

PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS WORKSHOP

**National Institute of Standards and Technology, Gaithersburg, Maryland USA
August 21-23, 2006**

Co-Located with the
IEEE International
Workshop on Safety,
Security, and Rescue
Robotics



National Institute of Standards and Technology (NIST)

This year's workshop host was founded in 1901 as the first physical science research laboratory in the U.S. federal government. It currently is a non-regulatory federal agency within the U.S. Commerce Department's Technology Administration.

NIST's mission is to promote innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.



Commercial equipment and materials are identified on this CD in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology (NIST), nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Table of Contents

Foreword	4
Call for Papers	5
Program Committee	6
Plenary Addresses	7
Featured Presentations	10
Related Event: Response Robot Exercise	11
Robot Demonstrations	12
Exhibits, Posters, and Demos	13
Example Robot Test Methods	14
PerMIS Author Index	18
Acknowledgements	19

Technical Sessions

MON-AMI Autonomy and Intelligence

<i>Improving Knowledge for Intelligent Agents: Exploring Parallels in Ontological Analysis and Epigenetic Robotics</i> [G. Berg Cross]	20
<i>Intellectual Performance Using Dynamical Expert Knowledge in Seismic Environment</i> [V. Stefanuk]	34
<i>Reification: What is it, and Why Should I Care</i> [J. Gunderson, L. Gunderson]	39
<i>Characteristics of the Autonomy Levels for Unmanned Systems (ALFUS) Framework</i> [H. Huang]	47

MON-AM2 Performance Metrics

<i>Meaningful Metrics and Evaluation of Embodied, Situated, and Taskable Systems</i> [D. Gage]	52
<i>Fault-Tolerance Based Metrics for Evaluating System Performance in Multi-Robot Teams</i> [B. Kannan, L. Parker]	54
<i>Image Classification and Retrieval Using Elastic Shape Metrics</i> [S. Joshi, A. Srivastava]	62
<i>Performance Metrics for Operational Mars Rovers</i> [E. Tunstel]	69
<i>Traversability Metrics for Urban Search and Rescue Robots on Rough Terrain</i> [V. Molino, R. Madhavan, E. Messina, A. Downs, A. Jacoff, S. Balakirsky]	77

MON-PMI Performance Evaluation

Performance Evaluation of Integrated Vehicle-Based Safety Systems
[J. Ference, S. Szabo, W. Najm] 85

A Performance Evaluation Laboratory for Threat Detection Technologies
[R. Schrag] 90

USARSim: Providing a Framework for Multi-robot Performance Evaluation
[S. Balakirsky, C. Scrapper, S. Carpin, M. Lewis] 98

Performance Evaluation of a Terrain Traversability Learning Algorithm in the DARPA LAGR Program
[M. Shneier, W. Shackelford, T. Hong, T. Chang] 103

Quantitative Assessments of USARSim Accuracy
[S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, J. Wang] 111

Feedback and Weighting Mechanisms for Improved Learning in the Adaptive Simultaneous Perturbation Algorithm
[J. Spall] 119

TUE-AMI DARPA ASSIST Special Session

Overview of the First Advanced Technology Evaluations for ASSIST
[C. Schlenoff, B. Weiss, M. Steves, A. Virts, M. Shneier, M. Linegang] 125

A Two-Stage Approach to People and Vehicle Detection With HOG-Based SVM
[F. Han, Y. Shan, R. Cekander, H. Sawhney, R. Kumar] 133

Performance Metrics and Evaluation Issues for Continuous Activity Recognition
[D. Minnen, T. Westeyn, T. Starner, J. Ward, P. Lukowicz] 141

An Improved Stereo-based Visual Odometry System
[Z. Zhu, T. Oskiper, O. Naroditsky, S. Samarasekera, H. Sawhney, R. Kumar] 149

Technology Evaluations and Performance Metrics for Soldier-Worn Sensors for ASSIST
[B. Weiss, C. Schlenoff, M. Shneier, A. Virts] 157

Utility Assessments of Soldier-Worn Sensor Systems for ASSIST
[M. Steves] 165

Using an Ontology to Support Evaluation of Soldier-Worn Sensor Systems for ASSIST
[R. Washington, C. Manteuffel, C. White] 172

Evaluating Intelligent Systems for Complex Socio-technical Problems: Seeking Wicked Methods
[M. Linegang, J. Freeman] 179

TUE-PMI Performance Analysis

Memetics and Intelligent System
[R. Finkelstein] 187

An Information-based Cyber Infrastructure to Support Performance Analysis in Complex Systems
[M-S. Li, A. Deshmukh, A. Jones] 189

<i>Three-Dimensional Data Registration Based On Human Perception</i> [B. Brendle]	197
<i>Performance Analysis of Symbolic Road Recognition for On-road Driving</i> [M. Foedisch, C. Schlenoff, R. Madhavan]	205
<i>Control of Nonlinear Stochastic Systems</i> [V. Aksakalli, D. Ursu]	213

WED-AMI Autonomous Systems Evaluation: Testbeds & Tools

<i>Challenges in Autonomous System Development</i> [J. Connelly, W. Hong, R. Mahoney, Jr., D. Sparrow]	220
<i>Long Term Study of a Portable Field Robot in Urban Terrain</i> [C. Lundberg, H. Christensen, R. Reinhold]	225
<i>A Standardized Testing-Ground for Artificial Potential-Field based Motion Planning for Robot Collectives</i> [L-F. Lee, V. Krovi]	232
<i>A Testbed for Heterogeneous Autonomous Collaborative Agents</i> [S. Asundi, A. Waldrum, N. Fitz-Coy]	240
<i>Endurance Testing for Safety, Security, and Rescue Robots</i> [J. Kramer, R. Murphy]	247
<i>A Complete Simulation Environment for Measuring and Assessing Human-Robot Team Performance</i> [A. Freedy, E. Freedy, J. DeVisser, G. Weltman, M. Kalphat, D. Palmer, N. Coyeman]	255
<i>Development of an Evaluation Method for Acceptable Usability</i> [B. Stanton, B. Antonishek, J. Scholtz]	263
<i>Measuring Up as an Intelligent Robot - On the Use of High-Fidelity Simulations for Human-Robot Interaction Research</i> [A. Green, H. Huttenrauch, E. Topp]	268
<i>On-orbit Servicing: A Brief Survey</i> [A. Tatsch, N. Fitz-Coy, S. Gladun]]	276

Foreword

The 2006 Performance Metrics for Intelligent Systems (PerMIS) Workshop was held at the National Institute of Standards and Technology (NIST) in Gaithersburg, MD from August 21 - 23, 2006. Sixth in a series of workshops targeted at defining measures and methodologies of evaluating performance of intelligent systems, this workshop focused on applications of performance measures to practical problems in commercial, industrial, homeland security, and military applications. This year PerMIS was held in conjunction with the IEEE Safety, Security, and Rescue Robotics (SSRR) Workshop (at the same venue from August 22-24, 2006).

On the first day of the workshop, Prof. Henrik Christensen (Georgia Institute of Technology, USA/Royal Institute of Technology (KTH), Sweden) delivered a plenary address entitled *Evaluation of Robots for Human-Robot Interaction*. Over the course of the day, authors presented their talks under three technical sessions: *Autonomy and Intelligence*, *Performance Metrics*, and *Performance Evaluation*. The morning technical sessions also included two invited talks by Dr. Gary Berg-Cross and Dr. Douglas Gage. In the afternoon, workshop attendees traveled to the nearby Maryland Fire and Rescue Training Academy to observe Federal Emergency Management Agency (FEMA) Urban Search and Rescue (US&R) Task Force members putting a wide variety of robots through their paces in operational scenarios, test methods, and radiation sensor integrations.

The second day witnessed two plenary addresses by Prof. Shigeo Hirose (Tokyo Institute of Technology, Japan) and Prof. Hugh Durrant-Whyte (The University of Sydney, Australia), respectively. Prof. Hirose's address discussed *Development of Rescue and Demining Robots in Tokyo Institute of Technology*. Prof. Durrant-Whyte's address was on *Maximal Information Systems*. The morning technical session was a special session organized by Craig Schlenoff (NIST) on the *Defense Advanced Research Projects Agency (DARPA) ASSIST* program. The afternoon technical session, *Performance Analysis*, included an invited talk by Dr. Robert Finkelstein in addition to regular presentations.

The final day of PerMIS was a veritable intellectual feast book-ended by two plenary addresses. Dr. Martin Buehler (Boston Dynamics, USA) kicked off the day with a presentation on *Developing Dynamic Legged Robots - Towards Greater Mobility Without Falling Over*. Dr. James Albus (NIST) concluded the day – and the workshop – with a banquet address on *Building Brains for Thinking Machines*. The day also included two featured presentations by Mr. Chuck Shoemaker (Robotic Research, LLC and formerly with the Army Research Lab., USA) *Army Autonomous Tactical UGVs* and Dr. Mike Montemerlo (Stanford University, USA) *Winning the DARPA Grand Challenge* in addition to an *Emergency Responder Panel Discussion* moderated by Prof. G. Kemble Bennett (Texas A & M, USA) with the participation of US&R responders from several FEMA Task forces. The morning technical session entitled *Autonomous Systems Evaluation: Testbeds & Tools* included an invited talk by Dr. Dave Sparrow (Institute for Defense Analyses, USA). The attendees also saw demonstration of bomb disposal robots being operated by bomb squads from Maryland, Virginia, and Michigan, with emphasis on training procedures, performance test methods, operator interfaces, and deployment strategies.

Overall, there were thirty three regular presentations, four invited talks, two featured presentations, five plenary addresses and a panel discussion in addition to robot events, demos, and exercises! The attendees of the workshop consisted of researchers, students, practitioners from industry, academia, and government and proved to be an excellent forum for discussions and partnerships, dissemination of ideas, and future collaborations.

Selected papers from this year's PerMIS (and SSRR) are being published in a forthcoming special issue of the Journal of Field Robotics: *Quantitative Performance Evaluation of Robotic and Intelligent Systems*.

We thank NIST and DARPA for their support of the workshop in making it a great success. We are also extremely appreciative of the successful collaboration with the IEEE SSRR organizers, especially Adam Jacoff, the General Chair, for helping us jointly produce such interesting technical and social programs. We trust that you will find these proceedings of the 2006 PerMIS workshop to be a useful source of technical ideas and reference. We look forward to your participation next year!

Sincerely,

Raj Madhavan
Program Chair

Elena Messina
General Chair

PERMIS-2006

In the sixth workshop in a series targeted at defining measures and methodologies of evaluating performance of intelligent systems, we will focus on applications of performance measures to practical problems in commercial, industrial, homeland security, and military applications. Topic areas include, but are not limited to:

Defining and measuring aspects of a system:

- The level of autonomy
- Human-robot interaction
- Collaboration

Evaluating components within intelligent system

- Sensing and perception
- Knowledge representation, world models, ontologies
- Planning and control
- Learning and adaption
- Reasoning

Infrastructural support for performance evaluation

- Testbeds and competitions for intercomparisons
- Instrumentation and other measurement tools
- Simulation and modeling support

Technology readiness measures for intelligent systems

Applied performance measures, e.g.,

- Intelligent transportation systems
- Emergency response robots (search and rescue, bomb disposal)
- Homeland security systems
- De-mining robots
- Defense robotics
- Command and Control
- Hazardous environments (e.g., nuclear remediation)
- Industrial and manufacturing systems
- Space robotics
- Assistive devices

SPONSORS



The dress code for this event is business casual.

Emergency responders should wear their insignias so that researchers and developers may strike up conversations.

PROGRAM COMMITTEE

General Chair:

Elena Messina, NIST Intelligent Systems Division, USA

Program Chair:

Raj Madhavan (Oak Ridge National Laboratory/NIST, USA)

S. Balakirsky (NIST, USA)

G. Berg-Cross (Engineering, Management and Integration, USA)

S. Carpin (International University Bremen, Germany)

M. Fields (US Army Research Laboratory, USA)

M. Foedisch (NIST, USA)

K. Fregene (Honeywell, USA)

J. Gorman (NIST, USA)

J. Gunderson (Gamma Two, Inc., USA)

Z. Kootbally (NIST, USA)

T. Kramer (NIST, USA)

M. Lewis (University of Pittsburgh, USA)

L. Reeker (NIST, USA)

S. Roumeliotis (University of Minnesota, USA)

C. Schlenoff (NIST, USA)

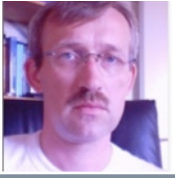
C. Scrapper (NIST, USA)

M. Shneier (NIST, USA)

Y. Wang (Technical University of Crete, Greece)

B. Weiss (NIST, USA)

PLENARY SPEAKER



**PROF. HENRIK
CHRISTENSEN**

Georgia Tech.
USA

Evaluation of Robots for Human-Robot Interaction

Mon. 08:30

ABSTRACT

Robotics is gradually maturing as a discipline which also implies an increased need for comparative R&D. At the same time robots are more and more deployed to serve as assistants to humans be it for search and rescue or as part of normal daily chores in the home. To enable evaluation of progress in research it is essential that rigorous methodologies for evaluation and performance characterization are adopted. Often a number of objections are put forward as to why such rigorous experimental protocols are not well suited for robotics. Some of the typical objections will be presented and discussed in the presentation. To illustrate the value and strategy of experimental evaluation two

example applications will be presented. Both applications are closely tied to robots that serve as assistants to people as part of daily operations. Experience from prior studies will also clearly illustrate the value of a careful design for evaluation and characterization of systems, which goes beyond the simple verification of theoretical models. Observations and lessons from an extensive set of studies will be summarized.

BIOGRAPHY

Henrik I Christensen is the Kuka Chair of Robotics and a Professor of Computing with the College of Computing, Georgia Institute of Technology. The appointment is part-time during 2006, which is a transition period from the earlier appointment at the Swedish Royal Institute of Technology, which included leadership of the Center for Autonomous Systems. He does research on mobile robotics, autonomous systems, computer vision, and biologically inspired robot systems. The overall emphasis is on a holistic approach to design of systems, including mathematically well defined methods for design, analysis and implementation of systems. A fundamental idea is that methods should be evaluated in realistic settings which involves an interesting scenario and a full systems context. He is involved in a large number of national and international projects. Dr. Christensen is a co-founder of the company Intelligent Machines and serve as a scientific advisor to Evolution Robotics. Research cooperation involves research labs and companies on three continents. In addition he has been actively involved in a number of community efforts in particular as the founding coordinator of the EU network of excellence in Robotics - EURON (2000-2006). Dr. Christensen is a fellow of the International Foundation of Robotics Research and served as an IEEE RAS distinguished lecturer (2004-2006). He also serves on the board of trustees of the Swedish STINT foundation.

PLENARY SPEAKER



**PROF. SHIGEO
HIROSE**

Tokyo Institute
of Technology
Japan

Rescue and De-mining Robots

Tue. 08:30

ABSTRACT

In this plenary talk, I will explain about our activities on rescue and demining robots. As for the robots for rescue operation, I will first explain my previous efforts on snake-like robots with slender and actively bending bodies. I will then show several types of snake-like "Soryu" robots which consist of three crawler-driven segments and their connecting joints. The Soryu has been adapted with a specific driving mechanisms to move inside narrow and winding paths among debris and is designed to protect against dust and water. A newly introduced crawler belt made of thin metal with rubber knobs will also be explained. I will also present a debris-inserting inspection camera, we are developing with a snake-like expandable rod mechanism.

In general, I will introduce our development process for these and other devices. We believe that the most effective rescue tools will be the ones which are widely used in our daily life. Based on this belief, we also paid special attention to the development of ordinary-life-embedded rescue devices. For example, automobile jack-up devices which can be used for rescue operations will be shown. As for the demining robots, I will explain about my preliminary efforts to develop walking-demining robots, and their tool-detachable foot mechanisms. I will explain about our latest activities on a practical demining vehicle named "Gryphon." It has a weight balanced arm with metal and ground penetrating radar and a 3D camera. It can measure the uneven ground and can drive the sensors along the surface of the ground. I will show the result of the experiments in several places such as in Croatia.

BIOGRAPHY

Shigeo Hirose was born in Tokyo in 1947. He received the B. E. degree with first class honors in Mechanical Engineering from Yokohama National University in 1971, and his M. E. and Dr. E. degrees in Control Engineering from the Tokyo Institute of Technology in 1973 and 1976, respectively. He was Research Associate and Associate Professor of the same university, and since 1992 he has been a Professor of Tokyo Institute of Technology, Department of Mechanical and Aerospace Engineering. He is a Fellow of IEEE, JSME and RSJ. His research interest is in the creative design of robotic mechanisms and their control. He has been awarded more than 30 academic prizes including the "Medal with Purple Ribbon" from the Japanese government (2006), the first Pioneer in Robotics and Automation Award (1999), and the Best Conference Paper Award (1995) from the IEEE Robotics & Automation Society.

PLENARY SPEAKER



**PROF. HUGH
DURRANT-WHYTE**

The University
of Sydney
Australia

Maximal Information Systems

Tue. 14:00

ABSTRACT

Information provides a quantitative metric for describing the value of individual systems components in autonomous systems tasks such as tracking, mapping and navigation, search and exploration; tasks in which the objective is information gain in some form. An information model is an abstraction of system capabilities in an anonymous form which allows a priori reasoning on the system itself. By construction, information measures have properties of composability and additivity and thus provides a natural means of modelling and describing large scale systems of systems.

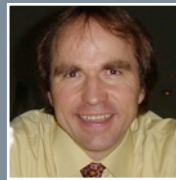
This talk will begin by describing how information measures arise naturally in

autonomous tracking, mapping and navigation, search and exploration tasks. It is then demonstrated that the performance of individual sensors and platforms can be modelled using these information measures and that system-level performance metrics can be computed. These ideas are illustrated in a series of tasks involving mixed air and ground autonomous systems. These include flight-tests of cooperative UAVs engaged in tracking and navigation tasks, mixed UAV, ground vehicles and human operatives, engaged in mapping and picture compilation operations, and operations involving multi-platform search in constrained environments. In each, it is shown how information provides both a performance metric and design objective underpinning large-scale systems of systems operation.

BIOGRAPHY

Hugh Durrant-Whyte received the B.Sc. in Nuclear Engineering from the University of London, U.K., in 1983, and the M.S.E. and Ph.D. degrees, both in Systems Engineering, from the University of Pennsylvania, U.S.A., in 1985 and 1986, respectively. From 1987 to 1995, he was a Senior Lecturer in Engineering Science, the University of Oxford, U.K. and a Fellow of Oriel College Oxford. From 1995 to 2002 he was Professor of Mechatronic Engineering at University of Sydney. In 2002 he was awarded an inaugural Australian Research Council (ARC) Federation Fellowship. He also now leads the ARC Centre of Excellence in Autonomous Systems. His research work focuses on autonomous vehicle navigation and decentralised data fusion methods. His work in applications includes automation in cargo handling, mining, defence, and marine systems. He has published over 300 technical papers and has won numerous awards and prizes for his work. He is a Fellow of the Academy of Technical Sciences, a Fellow of the IEEE and an IEEE Robotics Society Distinguished Lecturer.

PLENARY SPEAKER



**DR. MARTIN
BUEHLER**

Boston
Dynamics
USA

Dynamic Legged Robots

Wed. 08:30

ABSTRACT

Mobility can be an important contributor to robot intelligence, for gathering information, implementing decisions, and interacting with the environment. While wheeled and tracked robots have a relatively easy time moving around, we have to invest some intelligence first into legged robot design and control in order to harvest their potentially much greater mobility.

This talk will describe several recent legged robots that walk, run, balance, climb, carry loads, resist kicks and negotiate rough terrain with new levels of dynamic mobility, robustness, and performance. In the process we will encounter interesting issues related to the

system design, performance metrics, energy efficiency, and the experimental evaluation of these systems.

BIOGRAPHY

Martin Buehler received the M.Eng. and Ph.D. degrees in Electrical Engineering from Yale University in 1985 and 1990. His doctoral work focused on the design, control and analysis of juggling robots and the analysis of a hopping robot. After a Postdoc at MIT's leglab on dynamic legged locomotion, he joined McGill University, Montreal, in 1991 as an NSERC Junior Industrial Research Chair and a Scholar of the Canadian Institute for Advanced Research. He founded and headed the Ambulatory Robotics Lab, which produced one, four and six legged robots, including the ARL Monopods I and II, Scout I and II, CARL, PAW, RHex and AQUA, funded by major Canadian government, DARPA and industrial contracts and grants. In 2003 he received McGill's William Dawson Scholar Award. In the same year he moved on to become Director of Robotics at Boston Dynamics, Cambridge, USA. Dr. Buehler served as an Associate Editor of the IEEE Transactions on Robotics and Automation from 1998 - 2003, and is currently on the editorial boards of the International Journal of Robotics Research and the Journal of Field Robotics. He has supervised over 30 graduate students at McGill and has published over 100 papers on legged robot design and control, dynamic manipulation and motor control.d control, dynamic manipulation and motor control.

PLENARY SPEAKER



DR. JAMES
ALBUS

NIST
USA

Building
Brains for
Thinking
Machines

Wed. Banquet

ABSTRACT

In this talk, Dr. Albus will describe how research in computer science, control theory, and the neurosciences are converging towards intelligent systems that can mimic human performance in a broad range of applications. He will discuss current efforts to build machines that can perceive the environment, build an internal model of the external world, and use that model for decision-making, reasoning, planning, and real-time control of complex machines in uncertain, and potentially hostile, environments. He will suggest how system architectures designed for autonomous mobility systems are computationally similar in many respects to the human brain, and vice versa.

This work is part of a broad NIST program of research and engineering of intelligent systems to reduce costs and improve quality in manufacturing and construction, and to save lives of civilians on the highway and soldiers in combat. The research is conducted in collaboration with the Army Research Laboratory, DARPA, the Department of Transportation, and the U.S. manufacturing industry.

BIOGRAPHY

Dr. James S. Albus founded and led the Intelligent Systems Division at the National Institute of Standards and Technology for 20 years. He is currently a Senior NIST Fellow. Over a long and varied career Dr. Albus has made a number of scientific contributions. During the 1960's he designed electro-optical systems for more than 15 NASA spacecraft. During the 1970's, he developed a model of the cerebellum that after 30 years is still a leading theoretical model used by cerebellar neurophysiologists today. Based on that model, he invented the CMAC neural net, and co-invented the Real-time Control System (RCS). RCS is a reference model architecture for intelligent systems that has been used over the past 25 years for a number of systems including the NBS Automated Manufacturing Research Facility (AMRF), the NASA telerobotic servicer, a DARPA Multiple Autonomous Undersea Vehicle project, a nuclear Submarine Operational Automation System, a Post Office General Mail facility, a Bureau of Mines automated mining system, commercial open architecture machine tool controllers, and numerous advanced robotic projects, including the Army Research Lab Demo III Experimental Unmanned Ground vehicle. The latest version of the RCS architecture has been selected by the Army for the Autonomous Navigation Systems to be used on all Future Combat System ground vehicles, both manned and unmanned. He is also the inventor of the

NIST RoboCrane. He is currently working with DARPA and other government agencies on a concept for a National Program for Understanding the Mind, a.k.a "Decade of the Mind."

Dr. Albus has received numerous awards for his work in control theory including the NIST Applied Research Award, the Department of Commerce Gold and Silver Medals, the Industrial Research IR-100 award, the Presidential Rank Meritorious Executive, the Jacob Rabinow award, the Japanese Industrial Robot Association R&D Award, and the Joseph F. Engelberger Award for robotics technology. In 1998, he was named a "Hero of Manufacturing" by Fortune magazine.

Dr. Albus is the author of more than 180 scientific papers, journal articles, book chapters, and official government studies on intelligent systems and robotics. He has lectured extensively throughout the world and authored or co-authored five books:

- Engineering of Mind: An Introduction to the Science of Intelligent Systems - Wiley, 2001
- Intelligent Systems: Architecture, Design, and Control - Wiley, 2002
- The RCS Handbook: Tools for Real-Time Control Systems Software Development - Wiley, 2001
- Brains, Behavior, and Robotics - Byte/McGraw-Hill, 1981
- Peoples' Capitalism: The Economics of the Robot Revolution - New World Books, 1976

He is a member of the editorial board of the Wiley Series on Intelligent Systems serves on the editorial boards of six journals related to intelligent systems and robotics.

Dr. Albus received a B.S. in Physics from Wheaton College (Illinois) in 1957, a M.S. in Electrical Engineering from Ohio State University in 1958, and a Ph.D. in Electrical Engineering from University of Maryland (College Park) in 1972.

FEATURED PRESENTATIONS

Army Initiatives for Autonomous Tactical UGVs: The Last 10 Years WEDNESDAY 14:00

Mr. Charles Shoemaker, Robotic Research LLC, USA (formerly with the Army Research Laboratory)

Winning the DARPA Grand Challenge WEDNESDAY 14:45

Dr. Michael Montemerlo, Stanford University's Stanley Team, USA

Open Problems of Robot Technologies for Disaster Response THURSDAY 08:30

Prof. Satoshi Tadokoro, Tohoku University and International Rescue Systems, Japan

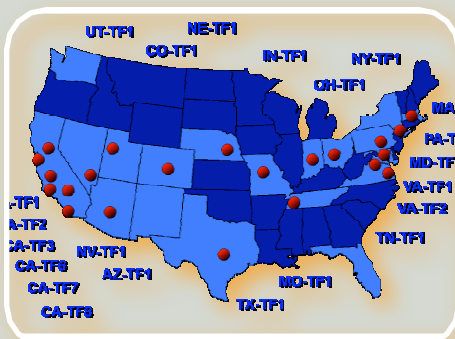
EMERGENCY RESPONDER PANEL DISCUSSION

Responder Experiences in the Field: Where Can Robots Help? WEDNESDAY 16:00

CHAIR: G. Kemble Bennett, Ph.D., P.E., Vice Chancellor and Dean of Engineering, Texas A&M University, USA

PANEL: US&R responders from several FEMA teams

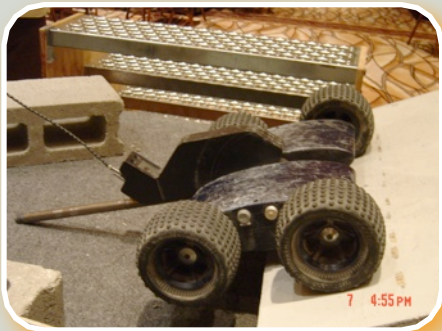
Due to the breadth and complexity of urban search and rescue (US&R) missions, and the diverse and evolving technologies present within robotic systems, the definition of performance requirements and associated test methods is an ambitious undertaking. Robot developers and emergency responders need to reach common understandings of the envisioned deployment scenarios, environmental conditions, and specific operational capabilities that are both desirable and possible for robots applied to US&R missions. Toward that end, NIST organizes events that bring emergency responders together with a broad variety of robots and the engineers that developed them to work within actual responder training facilities. These informal response robot evaluation exercises provide collaborative opportunities to experiment and practice, while refining stated requirements and performance objectives for robots intended for search and rescue tasks. This panel discussion will focus on responder perceptions regarding robot applicability, near-term opportunities for robots, and recent deployments that could have benefited from robotic technologies.



RELATED EVENT: RESPONSE ROBOT EXERCISE

MONDAY 16:00 - 18:00 FOR WORKSHOP VISITORS (SATURDAY-MONDAY FOR THOSE INVOLVED)

The third in a series of response robot exercises for FEMA US&R teams will be hosted at the Montgomery County Fire Rescue Training Academy in Rockville, Maryland (near NIST). This event will finalize the test methods targeted for the initial (Wave 1) set of standards as well as initiate experimentation with onboard payloads for chemical and radiological hazard detection. The three robot deployment categories selected by responders to be emphasized in Wave 1 are: ground peek robots that are small and throwable, wide-area ground survey robots that can traverse non-collapsed structures and provide remote situational awareness down-range, and aerial survey or loiter robots which in this case are rotary wing implementations. Robot developers take part in these multi-day exercises, which involve practicing operationally relevant US&R scenarios, to refine their understanding of responder requirements and deployment constraints. Buses from NIST will be provided so that attendees of SSRR and PERMIS can observe the final hours of this exercise.



BOMB SQUADS PRACTICE ROBOT DEPLOYMENTS

WEDNESDAY 17:00 - 18:30

Watch several civilian bomb squads deploy their robots in and around the test methods set up for all the other robot demos. See their operational methods and constraints. Discuss their needs.

Montgomery County, MD

Capt. Kevin Frazier

Fairfax County, VA

OFC Tom Eggers

Maryland State Police


Deputy Chief Jack Waldner

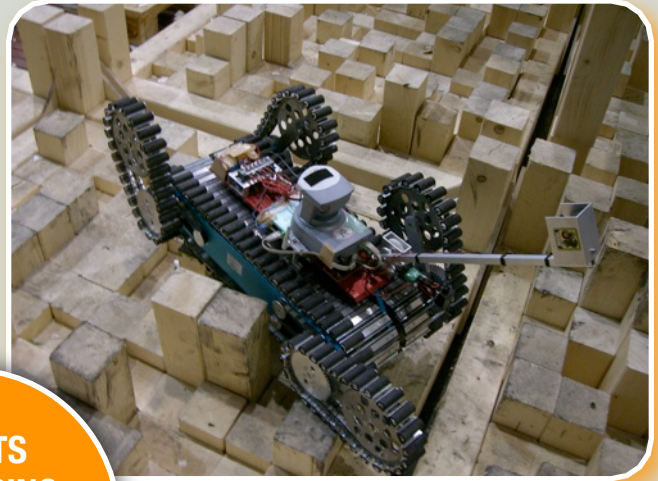
Michigan State Police

Lt. Shawn Stallworth



ROBOT DEMONSTRATIONS

DEMOS	RELATED EVENTS	LUNCH DEMOS	RECEPTION	BOMB SQUADS
	<p>Monday afternoon there will be a tour of the FEMA MD-TF1 training facility to watch urban search and rescue robots perform test methods and operational scenarios.</p>	<p>Every day the cafeteria will be filled with robot exhibits and performance test methods to host robot demos.</p>	<p>Tuesday evening there will be an exhibitor's reception in the cafeteria for both conferences, featuring robot demonstrations, appetizers, and a cash bar.</p>	<p>Wednesday afternoon there will be a realistic training event for local bomb squads practicing robot deployments in and around test methods in the cafeteria</p>



**ROBOTS
PRACTICING
EXAMPLE ROBOT
TEST METHODS**



EXHIBITS, POSTERS, AND DEMOS

Throughout the workshop, all exhibits and posters will be set up in the NIST cafeteria, along with some example robot test methods. Robot demonstrations will take place in and around these test methods during:

- Lunch hours each day
- Exhibitor's reception
(Tuesday 17:00 - 18:30)
- Bomb squad robots
(Wednesday 17:00 - 18:30)
- Coffee breaks

See the cafeteria layout for more information about exhibit booths, example robot test methods, and where to sit during lunches for best viewing.

The booth layout will be updated based on final registrations and set-up.

Robots and Associated Technologies:

- AirRobot (Germany)
- Applied Research Associates (USA)
- ARACAR (USA)
- Brno Univ. (Czech Republic)
- CRASAR (USA)
- Foster-Miller (USA)
- Fraunhofer AIS / Univ. of Osnabruck (Germany)
- Global Technical Systems (USA)
- HiBot (Japan)
- Idaho National Engineering Lab (USA)
- International Rescue System (Japan)
- Inuktun (Canada)
- iRobot (USA)
- Mesa Robotics (USA)
- NASA Goddard (USA)
- Non Lethal Solutions (USA)
- OmniTech (USA)
- Remington Technologies (USA)
- Remotec (USA)
- Skeyes Unlimited (USA)
- Telerob (Germany)
- Univ. of Electro-Communications (Japan)
- Univ. of Freiburg (Germany)
- Univ. of Massachusetts - Lowell (USA)
- Univ. of New South Wales (Australia)
- West Virginia High Tech Foundation (USA)
-

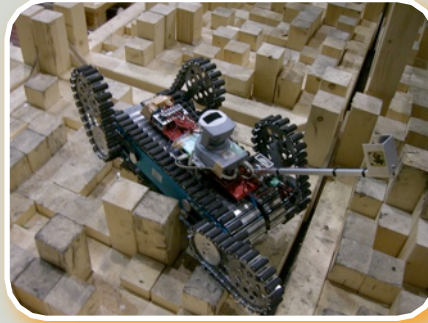
Sensors:

- Advanced Scientific Concepts (USA)
- Canesta (USA)
- CSEM (Switzerland)
- Envision Product Design CMOS X-ray (USA)
- Hokuyo (Japan)
- Multispectral Solutions (USA)
- RIKEN/Univ. of Tokyo (Japan)
- XRF Corporation (USA)

VISUAL ACUITY



STEP-FIELD DASH



DIRECTED PERCEPTION



ZIG-ZAG DASH



MANIPULATOR DEXTERITY



CONFINED SPACE DASH



CACHE PACKAGING



STAIRS, RAMPS, ETC.



EXAMPLE ROBOT TEST METHODS

The Department of Homeland Security, through the Science and Technology Directorate Standards Program, is developing performance standards for robots applied to urban search and rescue. NIST is leading this effort with collaboration from subject matter experts within FEMA US&R Task Forces and other response organizations, along with robot manufacturers and robot researchers intent on this application domain. The resulting standard test methods are being developed within the Homeland Security Applications Committee of ASTM International.

The various ASTM working groups developing these standard test methods will be meeting at NIST on Monday morning. All workshop attendees are welcome to participate in these meetings and to join the ASTM working groups.

Note: Please click on the paper title to view it in pdf.



08:15	Opening Remarks in Green Auditorium
08:30	Plenary Presentation in Green Auditorium: <i>Evaluation of Robots for Human-Robot Interaction</i> [Henrik Christensen]
09:30	Coffee Break
10:00	MON-AM1 Autonomy and Intelligence (Chairs: G. Berg-Cross and J. Gunderson) <ul style="list-style-type: none"> Improving Knowledge for Intelligent Agents: Exploring Parallels in Ontological Analysis and Epigenetic Robotics [G. Berg-Cross] (Invited) Intellectual Performance Using Dynamical Expert Knowledge in Seismic Environment [V. Stefanuk] Reification: What is it, and Why Should I Care? [J. Gunderson, L. Gunderson] Characteristics of the Autonomy Levels for Unmanned Systems (ALFUS) Framework [H. Huang]
11:30	MON-AM2 Performance Metrics (Chairs: D. Gage and S. Balakirsky) <ul style="list-style-type: none"> Meaningful Metrics and Evaluation of Embodied, Situated, and Taskable Systems [D. Gage] (Invited) Fault-Tolerance Based Metrics for Evaluating System Performance in Multi-Robot Teams [B. Kannan, L. Parker] Image Classification and Retrieval Using Elastic Shape Metrics [S. Joshi, A. Srivastava] Performance Metrics for Operational Mars Rovers [E. Tunstel] Traversability Metrics for Urban Search and Rescue Robots on Rough Terrain [V. Molino, R. Madhavan, E. Messina, A. Downs, A. Jacoff, S. Balakirsky]
13:00	Lunch in Cafeteria
14:00	MON-PM1 Performance Evaluation (Chairs: M. Lewis and R. Schrag) <ul style="list-style-type: none"> Performance Evaluation of Integrated Vehicle-Based Safety Systems [J. Ference, S. Szabo, W. Najm] A Performance Evaluation Laboratory for Threat Detection Technologies [R. Schrag] USARSim: Providing a Framework for Multi-robot Performance Evaluation [S. Balakirsky, C. Scrapper, S. Carpin, M. Lewis] Performance Evaluation of a Terrain Traversability Learning Algorithm in the DARPA LAGR Program [M. Shneier, W. Shackleford, T. Hong, T. Chang] Quantitative Assessments of USARSim Accuracy [S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, J. Wang] Feedback and Weighting Mechanisms for Improved Learning in the Adaptive Simultaneous Perturbation Algorithm [J. Spall]
16:00	All interested conference attendees take bus (10 min.) to MD-TF1 training academy to watch response robot exercise: <ul style="list-style-type: none"> Robots practicing operational scenarios Robots practicing test methods Radiation sensor integrations
18:00	Bus to Hotels





27
AUGUST
TUESDAY

08:15	Opening Remarks in Green Auditorium
08:30	Plenary Presentation in Green Auditorium: <i>Development of Rescue and Demining Robots</i> [Shigeo Hirose]
09:30	Coffee Break
10:00	<p>TUE-AM1 DARPA ASSIST Special Session (Chairs: C. Schlenoff and M. Linegang)</p> <ul style="list-style-type: none"> • Overview of the First Advanced Technology Evaluations for ASSIST [C. Schlenoff, B. Weiss, M. Steves, A. Virts, M. Shneier, M. Linegang] • A Two-Stage Approach to People and Vehicle Detection With HOG-Based SVM [F. Han, Y. Shan, R. Cekander, H. Sawhney, R. Kumar] • Performance Metrics and Evaluation Issues for Continuous Activity Recognition [D. Minnen, T. Westeyn, T. Starner, J. Ward, P. Lukowicz] • An Improved Stereo-based Visual Odometry System [Z. Zhu, T. Oskiper, O. Naroditsky, S. Samarasekera, H. Sawhney, R. Kumar] • Technology Evaluations and Performance Metrics for Soldier-Worn Sensors for ASSIST [B. Weiss, C. Schlenoff, M. Shneier, A. Virts] • Utility Assessments of Soldier-Worn Sensor Systems for ASSIST [M. Steves] • Using an Ontology to Support Evaluation of Soldier-Worn Sensor Systems for ASSIST [R. Washington, C. Manteuffel, C. White] • Evaluating Intelligent Systems for Complex Socio-technical Problems: Seeking Wicked Methods [M. Linegang, J. Freeman]
13:00	Lunch and Robot Demonstrations in Cafeteria and Courtyard
14:00	Plenary Presentation in Green Auditorium: <i>Maximal Information Systems</i> [Hugh Durrant-Whyte]
15:00	Coffee Break
15:30	<p>TUE-PM1 Performance Analysis (Chairs: B. Brendle and A. Jones)</p> <ul style="list-style-type: none"> • Memetics and Intelligent Systems [R. Finkelstein] (Invited) • An Information-based Cyber Infrastructure to Support Performance Analysis in Complex Systems [M-S. Li, A. Deshmukh, A. Jones] • Three-Dimensional Data Registration Based On Human Perception [B. Brendle] • Performance Analysis of Symbolic Road Recognition for On-road Driving [M. Foedisch, C. Schlenoff, R. Madhavan] • Control of Nonlinear Stochastic Systems [V. Aksakalli, D. Ursu]
17:00	<p>Exhibitors Reception in the Cafeteria and Courtyard</p> <ul style="list-style-type: none"> • Robot demonstrations • Example robot test methods • Posters
18:30	Bus to Hotels



08:15	Opening Remarks in Green Auditorium
08:30	Plenary Presentation in Green Auditorium: <i>Developing Dynamic Legged Robots</i> [Martin Buehler]
09:30	Coffee Break
10:00	<p>WED-AM1 Autonomous Systems Evaluation: Testbeds & Tools (A. Freedy and D. Sparrow)</p> <ul style="list-style-type: none"> Challenges in Autonomous System Development [J. Connelly, W. Hong, R. Mahoney, Jr., D. Sparrow] (Invited) Long Term Study of a Portable Field Robot in Urban Terrain [C. Lundberg, H. Christensen, R. Reinhold] A Standardized Testing-Ground for Artificial Potential-Field based Motion Planning for Robot Collectives [L-F. Lee, V. Krovj] A Testbed for Heterogeneous Autonomous Collaborative Agents [S. Asundi, A. Waldrum, N. Fitz-Coy] Endurance Testing for Safety, Security, and Rescue Robots [J. Kramer, R. Murphy] A Complete Simulation Environment for Measuring and Assessing Human-Robot Team Performance [A. Freedy, E. Freedy, J. DeVisser, G. Weltman, M. Kalphat, D. Palmer, N. Coyeman] Development of an Evaluation Method for Acceptable Usability [B. Stanton, B. Antonishek, J.Scholtz] Measuring Up as an Intelligent Robot - On the Use of High-Fidelity Simulations for Human-Robot Interaction Research [A. Green, H. Huttenrauch, E. Topp] On-orbit Servicing: A Brief Survey [A. Tatsch, N. Fitz-Coy, S. Gladun]
13:00	Lunch and Robot Demonstrations in Cafeteria and Courtyard
14:00	<p>Featured Presentations in Green Auditorium:</p> <ul style="list-style-type: none"> <i>Army Autonomous Tactical UGVs</i> [Chuck Shoemaker] <i>Winning the DARPA Grand Challenge</i> [Mike Montemerlo]
15:30	Coffee Break
16:00	<p>Emergency Responder Panel Discussion in Green Auditorium:</p> <ul style="list-style-type: none"> Chair: G. Kemble Bennett US&R Responders from Several FEMA Task Forces
17:00	<p>Local bomb squads deploy their robots in/around cafeteria</p> <ul style="list-style-type: none"> Robots practice training on test methods Operator interfaces and personal protective equipment Methods of deployment
18:30	Bus to Hotels
19:00	<p>Banquet for all attendees and responders at the Hilton Hotel (Gaithersburg)</p> <ul style="list-style-type: none"> Drinks then Dinner at 20:00 <i>Building Brains for Thinking Machines</i> [James Albus]

WATCH BOMB SQUAD ROBOTS IN/AROUND TEST METHODS

PERMIS

AUTHOR INDEX

Aksakalli, V.	TUE-PM1	Kramer, J.	WED-AM1	Ursu, D.	TUE-PM1
Antonishek B.	WED-AM1	Krovi, V.	WED-AM1	Virts, A.	TUE-AM1, TUE-AM1
Asundi, S.	WED-AM1	Kumar, R.	TUE-AM1, TUE-AM1	Waldrum, A.	WED-AM1
Balakirsky, S.	MON-AM2, MON-PM1	Lee, L-F.	WED-AM1	Wang, J.	MON-PM1
Berg-Cross, G.	MON-AM1	Li, M-S.	TUE-PM1	Ward, J.	TUE-AM1
Brendle, B.	TUE-PM1	Linegang, M.	TUE-AM1, TUE-AM1	Washington, R.	TUE-AM1
Carpin, S.	MON-PM1, MON-PM1	Lewis, M.	MON-PM1, MON-PM1	Weiss, B.	TUE-AM1, TUE-AM1
Cekander, R.	TUE-AM1	Lundberg, C.	WED-AM1	Weltman, G.	WED-AM1
Chang, T.	MON-PM1	Lukowicz, P.	TUE-AM1	Westeyn, T.	TUE-AM1
Christensen, H.	WED-AM1	Madhavan, R.	MON-AM2, TUE-PM1	White, C.	TUE-AM1
Connelly, J.	WED-AM1	Mahoney, Jr., R.	WED-AM1	Zhu, Z.	TUE-AM1
Coyeman, N.	WED-AM1	Manteuffel, C.	TUE-AM1		
Deshmukh, A.	TUE-PM1	Messina, E.	MON-AM2		
DeVisser, J.	WED-AM1	Minnen, D.	TUE-AM1		
Downs, A.	MON-AM2	Molino, V.	MON-AM2		
Ference, J.	MON-PM1	Murphy, R.	WED-AM1		
Finkelstein, R.	TUE-PM1	Najm, W.	MON-PM1		
Fitz-Coy, N.	WED-AM1, WED-AM1	Naroditsky, O.	TUE-AM1		
Foedisch, M.	TUE-PM1	Nevatia, Y.	MON-PM1		
Freed, A.	WED-AM1	Oskiper, T.	TUE-AM1		
Freed, E.	WED-AM1	Palmer, D.	WED-AM1		
Freeman, J.	TUE-AM1	Parker, L.	MON-AM2		
Gage, D.	MON-AM2	Reinhold, R.	WED-AM1		
Gladun, S.	WED-AM1	Samarasekara, S.	TUE-AM1		
Green, A.	WED-AM1	Sawhney, H.	TUE-AM1, TUE-AM1		
Gunderson, J.	MON-AM1	Schlenoff, C.	TUE-AM1, TUE-AM1, TUE-PM1		
Gunderson, L.	MON-AM1	Scholtz, J.	WED-AM1		
Han, F.	TUE-AM1	Schrag, R.	MON-PM1		
Hong, T-H.	MON-PM1	Scraper, C.	MON-PM1		
Hong, W.	WED-AM1	Shackleford, W.	MON-PM1		
Huang, H.	MON-AM1	Shan, Y.	TUE-AM1		
Huttenrauch, H.	WED-AM1	Shneier, M. ..	MON-PM1, TUE-AM1, TUE-AM1		
Jacoff, A.	MON-AM2	Spall, J.	MON-PM1		
Jones, A.	TUE-PM1	Sparrow, D.	WED-AM1		
Joshi, S.	MON-AM2	Srivastava, A.	MON-AM2		
Kalpath, M.	WED-AM1	Stanton, B.	WED-AM1		
Kannan, B.	MON-AM2	Starner, T.	TUE-AM1		
		Stefanuk, V.	MON-AM1		
		Steves, M.	TUE-AM1, TUE-AM1		
		Stoyanov, T.	MON-PM1		
		Szabo, S.	MON-PM1		
		Tatsch, A.	WED-AM1		
		Topp, E.	WED-AM1		
		Tunstel, E.	MON-AM2		

ACKNOWLEDGMENTS

These people provided essential support to make this event happen. Their ideas and efforts are very much appreciated.

Website and Proceedings

Debbie Russell (Chair)

Local Arrangements

Jeanenne Salvermoser (Chair)

Jennifer Peyton

Catherine Shupe

Exhibits and Test Methods

Brian Weiss (Chair)

Ann Marie Virts

Anthony Downs

Jeb Smith

Conference and Registration

Kathy Kilmer (Chair)

Teresa Vicente

Patrice Boulanger

Angela Ellis

Finance

Betty Mandel (Chair)

Facilities (Audio/Visual)

Hoyt Cox (Chair)

Dean Smith



Thank you
PerMIS
attendees!

Thank you
SSRR
attendees!

**Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Drive, MS-8230
Gaithersburg, MD 20899
<http://www.isd.mel.nist.gov/>**

Developing Knowledge for Intelligent Agents: Exploring Parallels in Ontological Analysis and Epigenetic Robotics

Gary Berg-Cross
EM&I

Abstract

1. Introduction

The overall PerMIS goal is to realize and measure both intelligence and autonomy. In pursuing this goal it is useful to think of intelligence as a phenomena realized by a cognitive system - one that can reason, using substantial amounts of appropriately represented knowledge. Thus central to progress towards the PerMIS goal is an adequate understanding and representation of knowledge that underlies an intelligent/cognitive system. However as argued by Berg-Cross (2004, 2006) designing effective semantically rich knowledge in dynamic domains remains difficult because of the lack of adequate model semantics and general design principles to engineer the underlying knowledge “represented” in intelligent systems. Engineering proper knowledge is hard because the real world does not have precisely defined states, because in meaningful situations the information available is limited, and because there is only partial predictability, since agents and the environment have their own dynamics. The challenge of dynamic, social and conceptual complexity can be seen in even seemingly reality based areas such as the geospatial realm. To describe geospatial information in support of cross-cutting missions from national security, law enforcement, health care, the environment, natural resources conservation and mobile robots we start with representations for descriptive attributes of geospatial concepts, and also the geometrical and positional aspects of these concepts. For example, in order to represent bus transportation systems we need realistic street information with bus stops imposed and more abstract timetables. This allows us to represent practical information about what stop on a bus route in is “near” the intersection of two streets. However, in event of a flood this information needs to be supplemented by elevation information and perhaps closeness to streams. Such features are not typically captured in geospatial data bases and precessable by GIS functions. Despite the importance and effort expended, on the whole, geospatial application models, supporting robots or humans, are functional weak and still have known logical and ontological flaws. For example, geospatial applications are typically not grounded in explicit reference ontologies with untangled taxonomic categories or abstract patterns relating entities. As a result they lack sufficient semantics to represent the information necessary to support geospatial tasks such as used by mobile robots and are typically very brittle with respect to making clear what conceptualization has been encoded, and tolerating other conceptualizations. How to avoid “brittle” representations of knowledge remains a major challenge for the development of robust intelligent systems. One problem is that alternative conceptualizations of knowledge various are not explicit enough to reflect both important distinctions and relations between concepts. As Frank (2001) notes, consistency constraints are placed on GIS DBs to assure that values incorporated in the database are consistent across the concepts a DD stores. Unfortunately in real situations the rules for such

consistency constraints are not very clear, in part because there are several levels of knowledge that needs to be represented and then unified as a system – a step not taken between ontologies. One approach for addressing this process, drawing heavily from the rational methods of philosophy, involves better conceptualization of knowledge using careful analysis of a problem. Such efforts typically manifest themselves in a highly structured model called a formal ontology. Ontological analysis extends some of the analysis that goes into engineering knowledge. However an issue for such efforts concerns an adequate basis for conceptualization given its dependence on inner perspective of the agents doing ontological analysis. An alternative approach to engineering knowledge is to develop it in a fashion similar to what happens in humans. A developmental approach considers the impact of embodiment philosophy on agent's and their knowledge which suggests a general way to address the problem of agent knowledge in a psychological realistic fashion. This might include the explicit role of beliefs, desires and intentions, something not incorporated in most theories of knowledge. A specific implementation of the embodiment philosophy, called epigenetic robotic uses cognitive principles to develop, adapt and learn through embedded robots interacting with a physical reality. This paper summarizes the ontological and developmental/evolutionary approaches which are currently separate, suggesting how they potential supporting one another as a rational-empirical approach to produce, validated, unified knowledge. Some implications for a philosophy of knowledge are drawn although the utility of the relationships are initial and speculative.

2. Quality Ontologies

Ontologies are appealing because they promise a grounding in semantics primitives but might be scoped for applications using information and data that has been "modeled" for older applications. For example, a spatial ontology might include very general categories of existence underlying geospatial object and events (e.g., existent item, spatial region, dependent part etc.). A problem as noted by several ontologists is that people now build what they call "ontologies without really knowing what ontological analysis goes into a quality ontology. Guarino (1998) provides important guidance to engineering an ontology using a "reality-based" conceptualization which is formalized commitment, language and intended models. My adaptation of Guarino ideas as an ontology forming process is shown in the Figure 1. Guarino's original ontology process stands in the center, and one can view particular domain ontology products arising from the same processes of conceptualization, commitment, expression in a language etc. Starting at the top of the core view, conceptualizations are agent's cognitive responses to "situation(s)" arising from a state of affairs of reality. Conceptualization is localized in an agent as a cognitive model that constrains the structure of what is comprehended as a piece of reality. Because of simplification this internal model is a partial comprehension and serves agents by simplifying and organizing attentional objects connected by perceived relations. As an example a cognitive model might provide constancy to routing knowledge allowing an agent's perceptual judgment to remain stable although particular situational features have changed. Object constancy, the ability to see an object as being one of constant size and shape despite variation in retinal-optical positions and distance from the observer, represents one example of human fundamental category invariants. Because this seems unaffected by culture it seems unlikely that object constancy depends

on learning: most of it has probably been "prepared" innately by evolution. Current development cognitive psychology suggests some starting points on invariants and more about these are discussed in the later section on a synthetic approach to grounding of this conceptualization.

The centrality of conceptualization means that useful/usable engineered knowledge of situations like geospatial problems can only be found by considering human experiences, cognition, abilities, and strategies, and integrating such conceptualization within our models of problem-solving. Application domain models do include some conceptualization and these are typically empirically driven, meaning they are generated by eliciting expert knowledge/opinion about a given domain of interest. However, this conceptualization is typically limited and lacks the broader analysis needed to resolve discrepancy between it and other conceptualizations. A particularly important conceptual difference concerns how psychological factors, (e.g. concepts of neighborhoods) are related to more concrete, physical features (e.g. topological features). Conceptualization follows strong cognitive bias that humans have based on perception of invariants, culture assumptions and social conventions.

Once conceptualization have been generated and validated they can be formalized in a language using some constraining commitments (see Figure 1) and this produces a model to represent the meaning of the conceptualization. That is, a formal structure employing conceptual relations over a domain space is used for a conceptualization aimed at accounting for the conceptualization's "meaning". In Guarino's formulation this is a general, intensional interpretation (Montague's intensional logic as opposed to an extensional definition of a particular state of affairs), with terms used to denote relevant conceptual relations. A crafted ontology is then a reflection of part of such a general/conceptual intensional model – the innermost part shown in the diagram. As part

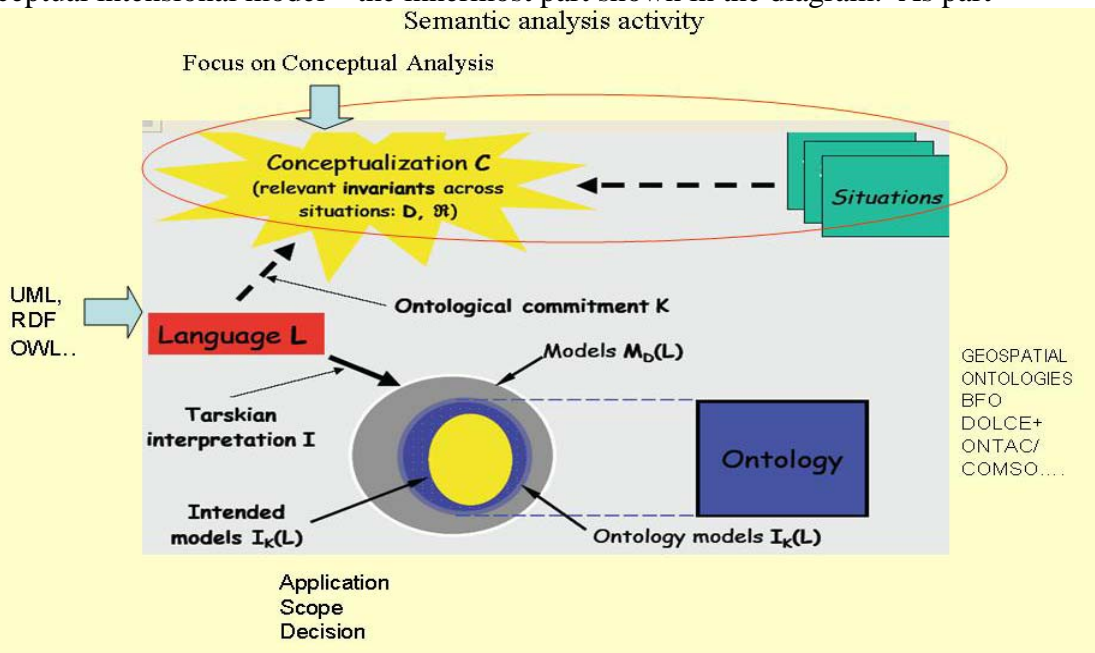


Figure 1 Adapted from Guarino "Making basic ontological assumptions: the DOLCE experience". -Ontolog presentation Feb. 2, 2006,

of conceptual analysis, I have located various activities that might be applied to a domain model like a geospatial model. Thus, we can start at the end of the diagram working with an existing foundational ontology like Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE+ Gangemi et al 2002) or the Basic Formal Ontology (BFO 2003) and place this as a foundation/reference point in a domain. However, we may have to use a different language to incorporate the ontology and to this properly we have to know the intensional model and the conceptualization as well as the situation that drove the conceptualization. The focused meaning of the formal model is derived from the mix of semantics in the formal language, but this formulation is good only if this matches the conceptualization. A key take away from this might be a warning about trusting ontologies that shortcut the process. In trying to make a domain model more rigorous we cannot concentrate just on representation issues or specific ontologies reuse. We don't make models better JUST by moving from entity-relation models to RDF or OWL. Such representation may be less important than a proper use of conceptualization and commitments to conceptualizations (explicit commitment on major ontological choices with clear branching points), since this drives the downstream semantic "products". Said another way, we have to consider judgments of domain coverage and its semantics based on formal relations. Also important for practical improvement is the strategy of using carefully crafted taxonomic backbone with a minimal number of general categories.

An ontology's semantic coverage addresses the extensional (from situations) and intensional aspects of a domain. Thus, domain models have conceptualization challenges covering both domain entity types and instances which are often not distinguished adequately. Both application models and ontologies as models should be trying to represent entities within the purpose "domain of discourse" - the situational chunk of the world that our model is about, and whose composition our formalisms should realize. We can see some of the more complex nature of meaning in seemingly tight concepts like "boundaries" as it is conceptualized in problem like a long border crossing. A useful starting point is to consider two senses of a border boundary:

1. an arbitrary physical border between tracts of land that are respectively claimed by two human social sovereignties.
2. a social border between those social sovereignties.

If I have modeled I should be able to answer quite a few questions about long borders. But one can think of exceptions to this way of defining long border in the first sense when we consider borders within social realities. Two sovereignties may CLAIM territory that intersects and thus we have a "disputed border" in which there may be little or no physical border BETWEEN the tracts. So perhaps we have a different category or a sub-type. Berg-Cross (2004) discussed some problems with tight, neat definitions of concepts based on Sowa's (2002) concept of "knowledge soup". Applying some of his problem types to the border issue I came up with the following problematic examples:

- Overgeneralizations: (classic example is Birds fly, but what about penguins?) Borders separate territories, but what if it is a disputed border? It is a border concept, but can't be used for separation.

- Abnormal conditions: You cross borders, but what if there is a fight at the border? What if there is a terrorist alert? Your passport has expired? There is an earthquake?
- Incomplete definitions: Is the concept of “no man’s land” included? Does it cover boundaries between things like the atmosphere and “outer space”?
- Conflicting defaults: You can see the Demasiado Corazon soap opera in Mexico, but not on US stations. Except you can get it via a satellite broadcast. Just as you can a TBS show in Mexico.

The point might be that when we try to build something like a formal ontology we are not typically including all this knowledge that humans know as adaptive agents working pragmatically based on interacting with the world. Even in infancy we seem to distinguish 2 types of object categorization. One is perceptual categorization, which is part of perceptual processing based on perceptual similarity of one object to another. As we develop we create perceptual schemas of what objects look like. Older infants develop a conceptual categorization that seems to be based on what objects do. One way of thinking of this more abstract type is as a restructuring of perceptual information into conceptual form. One basis for this is the experience of paths that objects take and the interactions among objects along these paths. Experience creates a simple mental model of a notion of kinds, such as animals, vehicles, furniture, plants etc. Underlying this kind of categorization seems to be functional roles played in events, rather than the physical appearance of the objects. Speculating, we might propose that evolution has selected us to be able to build schemas involving such broad categories to operate effectively in the world. Unanticipated applications come up all the time and we have to accommodate to the new demands. We can speculate about pragmatic based processes to do this. If some concept becomes the “expectation” part of a sufficiently reliable schema, then this concept is compared with the concepts at the basic level of abstractedness and a process tries to determine the common parts between the concepts. If the difference is less than a certain amount, then a new abstract concept might be added. Adding new entities to a domain model often extends the model and this provides many challenges to maintaining ontologies. Once a domain focus has been declared a particular model development may “overcommit” to cover more than needed (the extreme leads to the claim of “boiling the ocean”). Alternatively it can be undercommitted when it turns out that a model fails to represent relationships between entities that play a significant role. There is no a priori way to tell which problem a model may fall into and a reasonable approach to scoping might be to aim at one small degree of overcommitting since the added analysis and detail may provide contextual detail of use in the future or might reveal an error only seen through greater detail (Pisanelli et al, 2004). Another aspect of conceptualization in ontology arises when we consider the role of semantic primitives. It is recognized that primitive types and semantic relations used in a model/ontology should be grounded in some way and this should be taken into account in the intensional model. We can think of this as going through the cycle of Figure 1 from situation to ontology to form a core ontology of primitives. All models have to judge an “appropriate model focus” for what Pisanelli et al (2004) call modeling precision. Traditional application models have tended to develop shallower analysis and commitment than ontologies. This gets ever worse when they get represented in a weak formalism. The two problems go hand in hand. If

a modeler starts out with a formalism of certain expressability then the tendency is to conceptualize just to the extent of what the formalism can handle. One way to break out of this procrustean bed is to insist on adequate formalisms that can scale strategically to enhance domain semantics in a manageable way. We can ground work in 2 ways. First by using a cognitive stance toward invariant concepts in the conceptualization process which when done properly will support “intersubjective” validity and second by insisting on large scale reusability and empirical validity via situations as locally observed. The most expressive formalisms can be used to represent detailed accounts of intended meanings in an independently maintained reference ontologies which employ a cognitive stance such as a DOLCE’s top level ontology. Using this as a base, domain ontology can be scaled back to less formality in order to support easier comprehension and eventual computationally difficult services, such as data integration. The scaled versions are what Pisanelli et al (2004) call “lightweight” versions of reference ontologies and they will be more easily accepted in domain applications and these reference ontologies can be used in subsequent work. In terms of Guarino’s process a proper set of primitives simultaneously reflects both aspects of grounding. Taken together these provide some rational-empirical basis for work (Berg-Cross, 2004) which is often identified with pragmatism. Taking a pragmatic, cognitive stance is a key initial strategy leading to the types of epistemological methods needed to build a useful ontology or architecture. This view of grounding makes use of our understanding how human cognition isolates relevant invariants as the unified basis of perception, cognition and language.¹ Such invariant concepts as organizers of our experience are the essential tools of ontology building and can be supplemented by formal logic and other formalisms supporting knowledge representation and truth maintenance. Newell (1990) argued 15 years ago that the state-of-accumulated understanding in cognitive science could not adequately support and ground a unified cognitive theory. One question is how much we have advanced. Systematizing an approach to ontology development involves an advance unifying some of the field, but as seen in the above discussion major issues remain. An additional perspective on advancement is discussed in the following sections where the development approach to knowledge is outlined.

3. Embodied Agent, Situated Meaning and Developed Knowledge

Traditional robotics has explored a variety of techniques to establish perceptual and motor competence in negotiating the world. We have tested out various implementations and engineered knowledge by evaluating a mobile robot’s perception of bounded objects (walls etc.). For example robot exploration and map-learning problems, where the goal is a purely metrically accurate map, have often proved brittle when coping with the combination of low mechanical accuracy and sensory errors (Brooks 1985). Thus, on the whole, engineered general things like geospatial knowledge for imprecise spatial and temporal references such as near, far, around or with boundaries remains a challenge using traditional methods. To some these problems are due to the fact that engineered systems do not reflect the unification of embodied cognitive system that develop

¹ As noted by (Pisanelli et al, 2004) grounding is one of the more difficult epistemological principles to fulfill adequately, evidence because in its reckoning many discipline, such as philosophy, cognitive sciences, and linguistics are required.

capabilities that emerge in interaction with the natural and social worlds. The extreme view of embodied cognitive view proposes that intelligence is an emergent phenomenon. The premise of the embodied approach is that what we consider intelligent, flexible, and autonomous behavior only occurs in embodied agents which in turn are rooted in a “rich environment” within which they interact. To better understand the nature of this idea some general “facets of intelligence” of embodied cognitive systems can be articulated. Table 1 shows a list of facets proposed by Pfeifer and Scheier (1999) which serve to integrate agent problem solving along with other reasoning abilities that are psychologically plausible. Thus, as shown in the table, it is plausible to hypothesize that intelligence needs an incremental ability to learn from experience so that an agent performs more competently over time. One of the most important tenets for the organization of knowledge is the idea that the fundamental categorizing ability are built onto and emerge from sensorimotor interactions with the environment. To some this is the grounding for knowledge acquisition - letting the world serve as its own model which

Facet	Core Idea
Incremental Process	“prior” structures & functions bootstrap later structures & functions.
Central Role of Constraints	Early constraints promote realization of increased adaptability in a developing agent – given agents have degrees of freedom
Self organizing process	Self organizing via development and learning supplement innate mechanisms via interaction
Self exploration	Agents acquire control of their body dynamics via exploration
Categorization to aid sensorimotor coordination	Categorization is a foundational capability arising in response to sensorimotor interaction with the environment.
Value Systems	The saliency of environmental “features” is mediated by an agent value system, which modulates self organization and learning.
Social interaction	Interaction with other agents at different levels of cognitive maturity is important for cognitive development.

Table 1 Facets of Intelligence (adapted from Pfeifer and Scheier, 1999)

is discovered by an agent incrementally through development of its own models. Evidence for this includes the fact that children’s early words reflect the names of perceptual categories, models of the world, and this knowledge of underlying perceptual categorization develops from an iconic form to a more abstract categorical representation. Early knowledge acquisition is direct, via sensorimotor interactions with objects in the real world, which reflect the environment in which an embodied agent exists. This includes the essential ontological question of what exists in our “world”, and affordances for actions we can perform on these - what things are good for. Thus we eat nuts, but we lie on floors; we chase butterflies, but we draw with (don’t eat) a crayon. Children’s knowledge and their language reflects this categorization of entities which "afford" certain sensorimotor interactions. Floors afford walking and lying while nuts are for eating. But with continued embodied interactions simple concepts give way to more

complex distinctions. A child must learn to call some things "nuts" and other things "candy". This represents a fundamental ontological issue – we need to develop a consistent vocabulary reflecting presumed internal categories. Here a combined social and interaction seems to play a role. Such learning involves a degree of self organization, which stands in contrast to a rigid adherence to engineering intelligence into an agent rather than letting it develop in context. Embodied agents must deal with constraints and limitations. Given awareness of its own capabilities, an agent can organize a rational, practical approach to handle constraints. In turn, agent knowledge is further organized to reflect a pragmatic strategy of satisficing work arounds. Indeed reflection on one's own behavior in relation to expectations is a major source of learning arising from control that makes use of underlying value systems for motivating selection and beliefs, goals and intentions that rationalize behavior. As argued by Rao and Georgeff (1991) intentions are an integral part of an agent's mental state of an agent and play an important role in determining rational behavior as agents pursue goals. In turn the use of beliefs, goals and intentions (BDI) leads to higher functions. Thus, an agent can explain itself and can be told what to do in terms of beliefs, goals and intentions. These are particularly important in social interactions where shared beliefs and intentions play a role. A simple hypothesis about these abilities is that they serve evolutionary needs and the fact that human "intelligence" has evolved in a social setting and thus agent knowledge and abilities serve agent interactions which are important for the emergence of the higher forms of cognitive development. However, representing BDI as part of knowledge is hard and has not been part of the mainstream ontology development. A working hypothesis is that this may be one reason for the continuing problem with capturing semantics in our models. Sowa has a more structured discussion of a pragmatic cycle to handle this (see Berg-Cross 2004 and Sowa 2002) but an interesting question concerns whether these such dynamic theories of knowledge development (which seem to challenge the idea of "objective" knowledge) can be tested scientifically. An approach to this question is addressed in the next section on epigenetic robotics.

4. Developing Knowledge through Developmental/Epigenetic Robotics

One emerging area, called developmental robotics or epigenetic robotics, may provide a more empirical basis about how concepts are developed and provide robustness for such basic things as spatial/object concepts and reasoning. They have similar core ideas - combining developmental psychology & robotics, along with:

- embodiment of the systems;
- situatedness in physical and social environments;
- a prolonged developmental process through which varied and complex cognitive and perceptual structures emerge as a result of the embodied system interacting with its physical and social environment.

Developmental/epigenetic robotics takes inspiration from developmental psychology which it combines with mobile robotic abilities to show ongoing development of behavior in robotic systems. Thus, instead of modeling the surface behaviors of infants, it focuses on, for example, modeling more general causal mechanisms and variables

theorized to underlie the development of those behaviors. A starting point is the idea of a synthetic methodology, which understands “by building” an artifact for experiments. Experiments on children’s responses to stimuli, language used to describe shapes and patterns (e.g. pictures of bottles, trains, cats) are a good way to come into contact with the reality of invariants. A synthetic approach subsumes constructing a model (either a computer simulation or actual robot) aimed at addressing some phenomenon of interest (e.g. how a bird walks, how a baby recognizes an object on the floor), but also how to abstract general principles. Some general principles (emergence, sensorimotor grounding) for embodiment were described in the previous chapter. Particular hypotheses about the emergent can be empirically tested by building developmental agent architectures that are given the opportunity of adapting to the environments in which they are embedded. The process is well described in article Prince et al (2004):

The goal is to “specify the overall organization for constructing the epigenetic robot, and include architectures relating to *specific ontogenetic design* and *generic ontogenetic design*. In specific ontogenetic design, close use is made of knowledge of causal mechanisms and variables from psychological development. In generic ontogenetic design, the goal is to provide an ongoing emergence of behaviors in unstructured environments with less dependence on knowledge of psychological development.”...

“A key, we think, lies in viewing development as a task domain. Normally, as computer scientists and engineers, when we build a system, we think of specific task requirements and construct the system to fit those requirements... For example, in constructing a computer-vision system to detect and recognize objects, we may depend on the fact that the task has non-occluded objects, perhaps with specific colors, against a neutral colored background ... In the case of an ongoing emergence of behavior, we need to reshape our thinking about design. No longer are we striving towards task-specific design, rather we are engaging in *ontogenetic design*. We are designing systems that develop, and that have ongoing emerging behaviors. That is, our domain is now psychological development itself.”

The epigenetic approach might be described in four steps:

1. Create a formal model
 - Start with some ‘innate’ components/substrate (as previously discussed)
 - Consider the nature and demands of the environment
 - Add detail to mechanisms to allow “coding” as a program
2. Embody the design with robotic senses and actuators
 - Let development proceed from the substrate by an interaction between developing components & a dynamic environment
3. Along the developmental “path” temporary structures and processes may bridge to increasingly more complex cognitive structures (fitter ones) tuned to the environment by interactions with the environment (physical and social)
4. Test that the implemented formal model is able to parallel results observed in children

- Is yes, then we have more confidence in mechanisms proposed in the original theory
- If no, then revise hypothesized mechanisms and re-implement.

A main thrust of the epigenetic approach is that we don't try to directly engineer a task like visual segmentation, but instead utilize psychological development as the design goal for the robot and from this "development architecture" visual segmentation can emerge. An example of this work is to design a robot that learns to improve its target-oriented reaching based on initially having 2 substrates an accurate visual target fixation mechanism, and a reaching reflex. The robot's reaching reflex, is based on something we see in children, an Asymmetric Tonic Neck Reflex to generate an extension of the robot's arm, roughly in the direction of the robot's head turn. The robot's eye-gaze fixation control mechanism, separately tuned and not learned in this context, then develops to control the robot's head in order to accurately fixate on an object. In this way we have a method to have the robot learn to improve its target-oriented reaching proceeded in a series of steps. Like infants advancing in accurate fixation such agents should exhibit new, emergent, behaviors and to some degree an accumulation of knowledge and skills.

Developmental architecture refers to the overall organization used in constructing an epigenetic robot, and spans approaches using specific ontogenetic design which closely follow human psychological development, and approaches using generic ontogenetic design which have less dependence on principles of particular psychological development. Such a developmental architecture for such emergence has been engineered by Blank, Kumar and Meeden (2002). The main structure of the architecture which has both hierarchical and cyclical features is shown below in Figure 2. The hierarchical simplifies four distinct levels going from reactive behavior at the bottom to increasingly concept-driven behavior at the top. In the model each level builds "emergent" abstractions based on the representations formed and existing at the lower levels. To reflect multi-level learning the model is cyclical so that the subsequent levels can provide some feedback of constructed/discovered abstractions and conceptualizations to the next lower level. Blank et al (2002) seen this combination of hierarchy and feedback as essential to create an "understanding" of the world guiding how we behave in it. It is important to note that this understanding and the knowledge behind it is continually developing. Starting at the bottom, Level 0 is a substrate motor generator that models innate reflexes, which we can think of as the chiefly biological elements of the architectural infrastructure. These are represented as if-then rules handling perceptual situations – action interactions. Initially this is all the system "knows how to do." Level 1 included functionality to observe the sensor and motor values that are produced by Level 0. Level 1 controls the robotic agent based on abstractions it forms about the sensorimotor behavior. Level1 uses Self Organizing Maps (SOMs) to provide high-dimensional input vector to a particular cell in a low-dimensional matrix. The result is an approximation which effectively abstracts "similarity" from the pattern

via a family of concepts. Level 2 of the architecture provides an observation ability applied to the sensor/motor associations developed by Level 1 SOM. Level 2 observational experiences drives learning to predict what the next Level 1 state will be

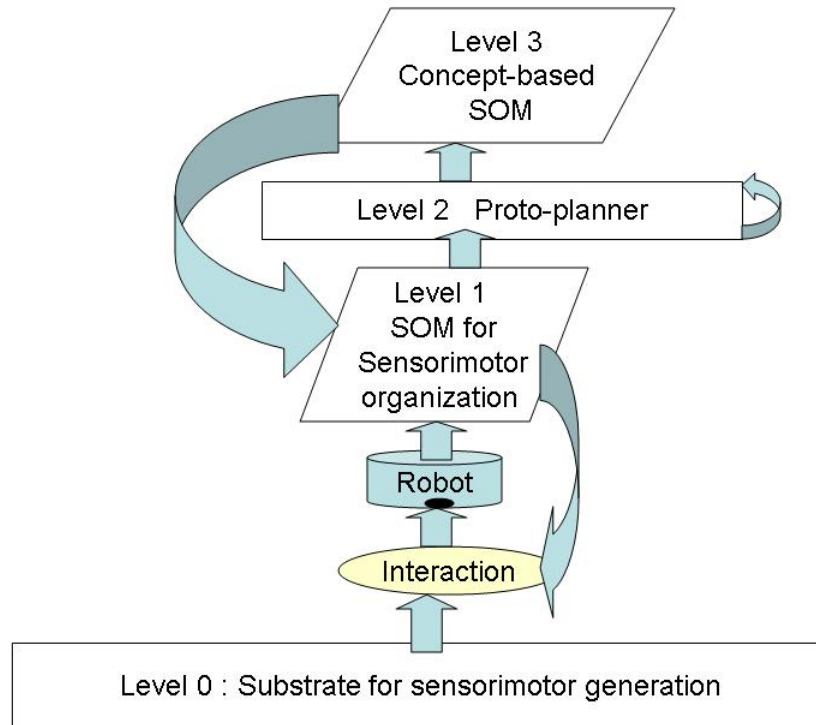


Figure 2 Architecture for developmental Agent. After Blank et al (2002)

given its current state. This prediction task enables Level 2 to use its re-current connections to recognize sequences of sensor/motor associations through time. Meeden (1994) has shown that this type of simple recurrent network will develop representations of multi-step behaviors, that might be termed “protoplans”. Note that these protoplans are not built in, but emerge from general capabilities that are “built in” as they develop through interaction with the environment. Level 3 uses a SNePs BDI architecture (Kumar 1996) that represents reason and can act on beliefs about conceptual entities of Level 2. Such belief representations for conceptual entities arise from social interactions with other agents as well as via concept discovery from lower level learning mechanisms. Taken as a whole, the result of the architecture for embodied agents is that simple reactive behavior can “develop” into time-dependent and planned behavior. The inference is that suitable knowledge has been abstracted to support such behavior although no direct knowledge was engineered in. Within the context of the Blank et al (2002) architecture this “knowledge” exists as conceptual entities arising from interactions with the world including other agents and supporting “plans” that arise from protoplans. It is suggestive that knowledge is embedded in just such a context.

5. Conclusions

Disciplined approaches to ontology development borrows from the rationalist tradition in philosophy while a synthetic-developmental approach using embodied agents provides an empirical way of develop knowledge. Together they provide an empirical-rational hybrid. While the efforts are seen as separate there as several ways that they may come

together and compliment one another as work progresses. As summarized earlier a cognitive stance for ontological analysis interprets real world “situations” using invariants. Developmental cognitive psychology and experimentation has provided some insights into the nature of these invariant, but because of the difficulty of manipulating underlying factors these are incomplete. Development is complex not just because of the number of factors, but the contingent nature of the process as understood by study of developmental dynamics of intelligence. By using a developmental architecture epigenetic robotics provides an empirical way to test out various hypotheses such as variation of substrates, learning processes etc. Embodied robotics also features a “diversity-compliance” principle to captures an essential aspect of the agent rationality but also the knowledge problem as we have been describing it. Agents problem solving includes constraints and limitations, which parallels problem solving for ontologists. A major research question is how embodied agents use invariants to exploit the implicit regularities of the world as a foundation for their knowledge. As we saw in Section 2, good ontologies exploit the notion of invariants. But as we have also seen in the previous section, agents develop more complex concepts from relatively simple sensorimotor interactions. What an agent can do or knows is not completely defined by a current situation. It can emerge. That is, early exploitation of abilities and invariants is superficial and is balanced by a divergent set of abilities and knowledge that emerges over time. The problem for ontology development, however, is that the field has largely been lacking a process for this emergence of richer concepts founded on interactions with the world. Instead of developing them from the ground up, the ontological development process has been more to develop top down, assuming the refined concepts are validated. One way to pursue bottom up development is to view particular task relevant knowledge as emergent supporting behavior based on a more general ontogenetic design. In this pursuit we may need to overcome some common ideas about knowledge:

- Over reliance on naive Information Processing (IP) models which gives too simplified a view of knowledge content as an “object” made of fixed concepts.
- Simplistic and inflexible use of top-down concepts which leads toward mechanical management rather iterative processes based on approximate and adductive principles. Further knowledge that is captured and management is often either too top down or bottom up oriented rather than integrated across all levels.
- Limited use of true knowledge management. Knowledge should be seen as part of a dynamic, rational-empirical system. Problems with linear methods of KM can be seen in the difficulty building enterprise models that integrate different levels of an organization. Entities mean different things at different levels and current methods may not allow very useful integration since with multiple levels there will be non-linear mapping of info between them.

Finally, the embodiment approach helps to add the BDI dimension to knowledge. Embodiment of agent’s and their knowledge suggests a general way to address the problem of agent knowledge in a psychological realistic fashion including the explicit role of beliefs, desires and intentions, something not incorporated in most theories of knowledge. As we have seen there are different notions of meanings coming out of these disciplines. One philosophical idea called realist semantics grounds knowledge in a states of affairs of real world and in Figure 1 something like this was alluded to in the

reference to “situation”. An alternative view is that cognitive semantics, part of the cognitive linguistics movement (Lakoff, 1998) which sees interaction with object structures (aka use) as the foundation on which to build the semantics of common terms. Cognitive semantic theories are similar to what we have seen in the previous section. It is built on the argument that the meaning of lexical information is conceptual. That is, the meaning of things like lexeme do not reference entity or relation in the "real world" but to a concept in the mind based on experiences with that entity or relation. An implication of this is that semantics is not objective and also that semantic knowledge is not isolatable from encyclopedic knowledge. Beyond that the meaning of agent knowledge is a cognitive model based on embodied interactions as we have seen in the developmental architecture previously described as a BDI model. This cognitive model has gestalt properties and utilizes basic-level categorization and basic-level primacy. It is more like that than an Aristotelian concept of essential properties. Agent thought uses prototypes and family resemblances as organizing structures and these conceptual structures can be described using cognitive models that have the above properties. One problem this causes is models are harder than concepts to fit into hierarchical structures.

References

Berg-Cross, Gary A Pragmatic Approach to Discussing Intelligence in Systems, Performance Metrics for Intelligent Systems (PerMIS) conference, 2004.

Berg-Cross Gary. Ontological Design, the Ultimate Design? Issues and Concepts, presented at The Emergences of Designs , The Washington Academy of Sciences Capital Science, March 25 – 26, 2006.

The BFO Ontology, <http://ontology.buffalo.edu/bfo/>

Blank, D.S., Kumar, D. and Meeden, L. (2002). **A Developmental Approach to Intelligence**. In *Proceedings of the Thirteenth Annual Midwest Artificial Intelligence and Cognitive Science Society Conference*, Edited by Sumali J. Conlon

Brooks R.A. Visual Map Making for a Mobile Robot, *IEEE Proc. International Conference on Robotics and Automation*, pp 824-829, 1985.

Frank, A. "Tiers of Ontology and Consistency Constraints in Geographical Information Systems," *International Journal of Geographical Information Science*, vol. 15, pp. 667-678, 2001.

Gangemi A., Guarino N., Masolo C., Oltramari, A., Schneider L., 2002. “Sweetening Ontologies with DOLCE”. In Gomez-Perez, A., Benjamins, V.R., (eds.), *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. Proceedings of the 13th International Conference (EKAW 2002)*. Lecture Notes in Computer Science 2473. Heidelberg: Springer, 166-181.

Guarino, N.: *Formal Ontology and Information Systems*. Proceedings of FOIS'98, Trento, Italy, 6--8 June 1998 (1998), pp. 3--10.

Meeden, L.. *Towards planning: Incremental investigations into adaptive robot control*. Ph.D. Dissertation, Indiana University. 1994.

Pfeifer, R. and Scheier, C.. *Understanding Intelligence*. MIT Press, Cambridge, MA. 1999.

Pisanelli, D. M. Gangemi, A. Massimo B., Catenacci C. "Coping with Medical Polysemy in the Semantic Web: the Role of Ontologies" MEDINFO 2004, M. Fieschi et al. (Eds) Amsterdam: IOS Press.

Prince, Christopher, Helder, N. Mislivec, E. and Hoolich, G. "Core Concepts in Epigenetic Robotics" paper presented at the University of Zurich, Aug. 23, 2004.

Rao, Anand and Georgeff, Michael Modeling Rational Agents within a BDI-Architecture, in Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, Fikes, and E. Sandewall, (Eds.) 1991.

Sowa, John F. "Representing Knowledge Soup In Language and Logic" A talk presented at the Conference on Knowledge and Logic, 15 June 2002.

Intellectual Performance using Dynamical Expert Knowledge in Seismic Environment

Vadim Stefanuk
Institute for Information Transmission Problems,
Russian Academy of Science,
Bolshoy Karetny per. 19
127051 Moscow, Russia
stefanuk@iitp.ru

Abstract—The paper introduces the notion of Dynamic Expert System. It shows that by restarting a common (static) expert system periodically it is possible to cope with dynamic environments. This quasi-static approach to dynamics is suitable if the environment is changing slowly enough in comparison with the inference engine operation and the user reaction time. Implementation of this scheme in "pure" Expert System shell meets no difficulties. However in practice some problems may occur due to the side effects in rules and attached procedures. These problems and their relation to classical AI issues are considered in details. The designed system has been applied to the task of seismology forecast, which contains the dynamical factors of both numerical and heuristic nature.

The resulting dynamic expert system never stops, occasionally interrogating the user when it suspects that some of the previously entered data are obsolete. In this sense the computer system behaves as an "alive creature."

I. INTRODUCTION

In eighties there was a certain interest to the reactive planning research. This interest also was supported by the general feeling of dissatisfaction with classical planners when facing with dynamic, time dependent reality [1].

In several application domains for advanced diagnosis strategies the same property hold as for reactive planning, though it is not the same problem [2]. An example application in [2] is the intention recognition in cooperative help to recognize the user's goal and intentions.

Anytime property of a real-time planning system was introduced to mean that the system can make a quick, dumb solution if that requested, but still is able to reason intelligently if given more time. The ERE planner [3] has the anytime property.

There was also research in real-time expert systems [3]. Real-time expert system must often work in limited time and adapt to a changing environment, its input being usually a sensor data.

Several researchers noted that "real-time" traditionally is taken to imply "need to be fast". However in reactive planning or real-time expert systems it is not necessarily so. In a number of applications the main issue is that the time for planning or

decision making is limited, but the limit might not necessarily be tight.

The problems of dynamics are being overcome in various ways. Thus, in reactive planning the execution system can be guided by a highly adaptable plan that either hand-coded or generated at compiler time [1].

The notion of reactive diagnosis to denote diagnosis problems where time limitation and adaptation to changes are important factors was introduced in [2]. In conclusion to the paper its author says in particular: "A more intricate problem in the notion of adaptiveness, how can we expect to reuse results from earlier proofs when the circumstances change? The author plans to investigate these questions for the particular application domain of intelligent help".

Present paper addresses similar questions in a straightforward manner through the notion of Dynamic Expert Systems. The goals of our paper are:

- to propose a new architecture for the reactive or Dynamic Expert System;
- to study the problems of combination of time-dependence and fuzziness in case of changing environment;
- to consider certain classical AI issues such as the frame problem, side-effects and fuzzy inference in the way they appear in our version of Dynamic Expert Systems;
- to show an example application of the proposed architecture in a real problem domain.

Common Expert System shells are usually oriented to problems for which the Knowledge Base (KB) and Data Base (DB) items, after they entered by the Knowledge Engineer, do not subject to changes during the whole session of interaction with the user. Though the user may occasionally modify DB items, the different sessions are essentially independent. Such a system may be called *static expert system* as its knowledge base is a fixed repository of information.

Yet, in a number of real applications DB and KB may considerably vary within one session of user interaction with ES. For example, during one session of the earthquake forecast in a certain area some new evidences may arrive and some old

evidences may become obsolete, both processes influencing which rules in KB are to be relevant.

Below we will discuss the proposed architecture and the problem domain, which was used to demonstrate its advantages. As our quasi-static system is built upon a traditional static expert system, in the following paragraph we will described briefly the static expert system which was used as the basis for the dynamical architecture.

IF (AND (A1:=V1) (A2:=V2) ... (An:=Vn)) THEN (Am:=Vm)

Fig. 1. The format of a "pure" rule

```
( (|subgoal processed| |yes|
  (with-window *instruction-window*
    (send *instruction-window* :clear-screen)
    (send *query-window* :clear-screen)
    (send (histo-window1) :clear-screen)
    (send (histo-window2) :clear-screen)
    (FORMAT (histo-window1) "*** Middle-term forecast~% the
      week of earthquake:~%")
    (histo |the week of earthquake| (histo-window1))
    (FORMAT (histo-window2) "*** Short-term forecast~% the
      hour of earthquake:~%")
    (histo |the hour of earthquake| (histo-window2))
    (wait-key)
    (send *query-window* :clear-screen)
    (FORMAT *query-window* "*** Expected power: ~A")
    (wait-key)))
  (|subgoal chosen| |monthly probability and power forecast|)
  (|total time estimation| |yes|)
  (|power estimation| |yes|)
)
```

Fig. 2. A rule with attached procedures

However the system ZNATOK acquires its efficiency and usefulness trough an extensive usage of various attached procedures (see example in Fig. 2).

Besides these traditional attached procedures like those shown in Figure 2 in the ZNATOK Expert System Shell, there is a number of so-called *control procedures*, influencing the performance the inference engine itself. This feature provides the necessary flexibility to the static shell ZNATOK that by now was used in a number of applications in medicine, tutoring, civil engineering, project evaluation [6], [7] and etc..

III. TREATMENT OF FUZZY VALUES IN ZNATOK

Before going further note that ZNATOK was able to cope with fuzziness.

The fuzzy concepts, met in the assertions (rules) and in the data, are treated in the system ZNATOK in a traditional way.

II. STATIC EXPERT SYSTEM ZNATOK

Our static expert system shell ZNATOK resembles the shell described in [5]. It is made as a cardboard rule-based system, which uses attributes for storing data and a simple stack arrangement as the inference engine.

A typical "pure" rule is shown in Fig. 1.

The main idea is to accumulate the evidences in favor of a certain hypothetical fact in the process of functioning of Inference Engine of the ES in order to decide what hypothesis is to be preferred.

In the paper [8] an axiomatic approach was proposed to create a regular way to produce formulae for combining evidences under various formal constrains. This approach found a support in [9] where it was somewhat emphasized. We will not go into details here pointing only that the combination formulae used in ZNATOK reminds the one used in MYCIN.

Many examples might be found in the area of earthquake forecast for long, middle and short term predictions.

The next rule (see Fig. 3) shows how the attribute "the hour of earthquake " receives a fuzzy value. Besides that, a fuzzy value for an attribute may be directly entered by the user in the way demonstrated in Fig. 4.

```

(|the hour of earthquake|
                                     (:F  ( |in 18 hours| 0.0)
                                     ( |in 20 hours| 0.7)
                                     ( |in 22 hours| 0.8)
                                     ( |in 24 hours| 0.9)
                                     ( |in 26 hours| 0.8)
                                     ( |in 28 hours| 0.7)
                                     ( |in 30 hours| 0.0)
                                     )
(|the earthquake will be in a day| |yes|))

```

Fig. 3. A rule, which assigns fuzzy values

***** System SEISMO ***** @AIPRO Moscow ****22:55

the level of water in wells

```

lowered
> stable

```



Fig. 4. A fuzzy value might be directly entered by the user

IV. QUASI-STATIC SYSTEM ARCHITECTURE

After some modification the described static architecture of ZNATOK type may be used in a quasi-static mode. This new mode results from a repeating restart of the static shell within one session of user interaction with the ES. During each restart cycle virtually all the data used (and all the rules involved) may be reconsidered, if of course some grounds for such reconsideration are found in the ES Knowledge Base.

It appears the proposed quasi-static schema resolves the classical Frame Problem: how to remove all the consequences of a datum if the datum is not valid any more. At least it is true for a "pure" ES shell (an analog of the "pure Lisp") with the "pure" rules of Figure 1, in which side-effects are absent (see however a figure below).

In case of a dynamic environment the attached procedures cause major problems due to possible side-effects related to them.

V. SIDE-EFFECTS HANDLING

Side-effects in a computer system may be subdivided into two classes: external and internal ones.

The notion of internal to the system side-effects is slightly more sophisticated and it was studied in [10] in connection with the recursion removal problem. The internal side-effects are related to the evaluation of attributes, rules, procedures and other forms that get their values in the normal run of the system [7].

One might find it useful to consider *relative side-effects*, relevant to the pairs of expressions (or forms) evaluated sequentially [10]. These are special cases of side-effects. We

say that the form F1 produces a relative side-effect on the form if and only if the evaluation of the form F1 may change the result of evaluation of F2.

When a ES shell of ZNATOK is being applied to a concrete static domain the Knowledge Engineer has to be careful using the side-effects to achieve a desired behavior of the system. This task is not simple and requires from the KE a deep knowledge of the system architecture and the problem domain.

This task becomes even more complex when the system is used in the quasi-static mode. In the quasi-static mode the number of possible side-effects are doubled. For each relative side-effect that the form A1 produces on the form A2 in the static case, in the quasi-static case one has to consider in addition a possibility of relative side-effect which the form A2 produces on A1 at the next restart cycle.

From the other side, used properly side-effects present a very powerful way to achieve flexibility and efficiency. Some hints for this may be found in [10].

One of the important goal of Knowledge Engineer is to achieve an *operational consistency* by avoiding undesirable side-effects. A practical way to check the operational consistency is to run the dynamic shell in a number of static environments. In a static environment the quasi-static ES should behave exactly as the original static ES.

It is obvious from the description of quasi-static system architecture that the Inference Engine must be a deterministic one. Otherwise the operational consistency is unobtainable.

To achieve both operational consistency and efficiency it is necessary to introduce types for the different attributes used in the ES. For this purpose there three types DYNAMIC, STATIC and FUZZY are used in the system. The DYNAMIC

type serves to limit the number of attributes reconsidered from cycle to cycle. FUZZY type saves efforts in evaluation of fuzzy variables by distinguishing fuzzy and ordinary variables.

VI. SEISMOLOGY DATA

In the present paper we are not considering the very tough problem of direct knowledge acquisition, i.e. the direct transfer of the knowledge of experts on seismology to the ES Knowledge Base. Thus, we avoid consideration of the problem of special "logic of time and space" [11] used by many experts. Both the time and the space are treated as regular physical phenomena in our system.

However, it turned out to be impossible to avoid the use of fuzzy concepts as they constitute the basis for almost all the assertions of Experts concerning the earthquake forecast. In particular, the fuzziness permits a seismologist to express a certain imprecision in measuring time intervals.

The earthquake forecast involves both observations based on the laws of physics and the observations of a heuristic nature. The law of linear time accumulation of the tension in rocks may be taken as an example of purely physical phenomenon. It may be easily taken into account by a corresponding attached procedure for computation of the tension, which gradually accumulates with time.

Here is a typical heuristic observation that may be found in the relevant literature: "As a rule, on the eve of a strong earthquake a predecessor occurs consisting in a local displacement. Yet on the eve of an average earthquake it is the whole area that is displaced."

From this and many similar examples found in the seismic literature it follows that the fuzziness and time dependence are intrinsic properties of the problem area.

VII. IMPLEMENTATION AND RESULTS

The quasi-static ES shell ZNATOK 2.0 was implemented in Common Lisp. It is very compact and runs practically on any PC.

The study of relevant literature on seismology and consultations with practitioners in the domain let us outline the most essential requirements for a working prototype of ES for earthquake forecast domain. Actually it is this application that has led us to the development of the quasi-static approach to the reactive expert systems. The quasi-static approach is quite applicable here as the process leading to an earth quake develops very slowly in comparison to the rate of performance of the inference engine of the Expert System.

The main menu to which the user may address many times (during one session) to choose a problem of interest related to the earthquake forecast is demonstrated in Fig. 5.

```
***** System SEISMO ***** @AIPRO Moscow ****22:51
```

```
Choose subgoal:  
> estimation of the time of a strong earthquake  
   estimation of strength of expected earthquake  
   forecast of probability and the power  
   monitoring  
   possibility of a forecast  
   the end of the session (exit)
```

Fig. 5. Starting menu of the seismic forecast system

The result given by the Expert System may be presented in a pseudo-graphical form (see Fig. 6), which is a convenient way to follow changes in the seismic situation in the seismic observation site.

In practice this dynamic expert system never stops. It behaves as an *alive creature*. Automatically it shows new results of forecast in accordance with its internal calendar. Occasionally it asks the user about some new data to replace the obsolete ones from the system point of view. Or else, the user (or a sensor, in the monitoring mode) may interrupt the

system and the user may click a new entry in the main menu (see Figure 5).

VIII. CONCLUSION

A practical way to make a reactive (dynamic) ES is proposed and its limitations and problems are studied. The quasi-static shell differs from conventional static one in the possibility of a frequent restart provided that the static data accumulated in DB from cycle to cycle are remembered. The architecture was

```
***** System SEISMO ***** @AIPRO Moscow ****23:15
```


Reification: What is it, and Why Should I Care?

Gunderson, J. P. and Gunderson L. F.

Gamma Two, Inc.

1733 York Street

Denver, CO, USA

jgunders@gamma-two.com

lgunders@gamma-two.com

Abstract—In this paper we present the idea of a Reification Engine, which bridges the gap between the sensor and symbolic levels for a cognitive system deployed in a robotic chassis. When any intelligent system is embedded, it can no longer deliberate only about crisp, clean symbols. It must somehow derive these symbols from ‘messy’ sensor data. This means that those symbols are now only representations of objects, events, and behaviors in the real world. To achieve its goals, the embedded cognitive system must quickly and effectively reason about the things that those symbols represent. We argue that a major problem with the top-down and bottom-up approach to resolving this issue may be due to the absence of a critical sub-component – which we are calling a Reification Engine. We present the reasons for our conclusions, and lay out the functional specification for this engine. Finally, we discuss ongoing work in which the prototype Reification Engine is embedded into a field robot, designed to function in a hazardous materials response role.

Keywords: *Cognition, Robotics, Artificial Intelligence, Sensors and Symbols, Biologically Inspired Robotics, Reification, Reification Engine.*

I. INTRODUCTION

Where is my robot? You know - the one that acts like the ones in the movies; the one that I just tell what to do, and it goes out and does it. If it has problems, it overcomes them; if something in the world changes, it deals with the changes. The robot that we can trust to do the dirty, dangerous jobs out in the real world - where is that robot? What is preventing us from building and deploying robots like this?

While there are a number of non-trivial and necessary hardware issues, the critical problem does not seem to be hardware related. We have many examples of small, simple systems that will (more or less) vacuum a floor, mow a lawn, or pick up discarded soda cans in an office. But these systems have a hard time dealing with new situations, like a tee shirt tossed on the floor or the neighbor’s cat sunning itself in the yard. We also have lots of teleoperated systems, from Predator aircraft to deep sea submersibles; from bomb disposal robots to remote controlled inspection systems. These systems can deal with changes to the world and significant obstacles: provided that one or more humans are in the loop to tell the robot what to do.

So what happens when a person takes over the joystick and looks through the low-resolution, narrow field of view camera of a perimeter-patrol security robot? Suddenly, where the

robot was confounded by simple obstacles and easy to fix situations, the teleoperated system is able to achieve its goals and complete its mission. This is despite the fact that in place of a tight sensor-effector loop, we now have a long delay between taking an action and seeing the results (very long in the case of NASA’s Mars rovers). We have the same sensor data, we have the same effector capabilities, we have added a massive delay – yet the system performs better. Of course, it is easy to say that the human is just ‘more intelligent’ (whatever that means), but that does not really answer the question. What is it that the human operator brings to the system?

We believe that a major component of the answer is the ability to reify: the ability to turn sensory data into symbolic information that can be used to reason about the situation and then to turn a symbolic solution back into sensor/effector actions that achieve a goal. This bridging process from sensor to symbol and back is the focus of this paper.

Since it is the addition of a human to the system that seems to enable success, we draw heavily from current research into what biological systems (primarily vertebrates) do to succeed the world, and how they do what they do. We look at research into cognition on a symbolic level and research into the physiology of biological entities on a physical (sensor/effector) level. From these investigations we derive a computational model of reification, and an infrastructure to support the mechanism. Finally, we outline an architecture that we are developing to add a Reification Engine to existing robotic systems.

II. COGNITION AND PHYSIOLOGY

There has long been a gulf between artificial intelligence researchers who focus on deliberative symbol manipulation and those who focus on embedding control systems into robots. Much of this gulf has been ascribed to the different approaches: top-down versus bottom-up. The general consensus has been that as the two ends work towards the middle, the gulf will narrow and narrow until it disappears. However, recent research has suggested that the gulf may not be bridgeable by work from either side; rather it may require a specific research approach that is different from either the sensor-based or the symbolic domains.

From the point of view of the deliberative approach, a symbol manipulation system is developed, and it is outside the

scope of the symbol system to recognize the physical and perceptual characteristics that define the thing referred to by the symbol. From the viewpoint of the embedded systems approach, the crucial task is the recognition of physical and perceptual cues, while mapping those cues onto a symbol system is outside the scope of the research.

Underlying both these beliefs is the assumption that once the core research was addressed, it would just be a matter of pushing the research frontier towards the opposing viewpoint until they met. If one continues the bottom-up (or top-down) approach long enough, eventually one gets to the top (or bottom) and the complete problem is solved (See Figure 1.A).

Both the top-down and the bottom-up approaches have made great strides towards the complete solution. However, there seems to be a gap that neither has been able to cross. It is clear that both the sensor/effector-to-symbol pathway and the symbol-to-sensor/effector pathway are necessary to support deployed intelligent systems. We argue that the gap can be bridged by the bidirectional process of reification.

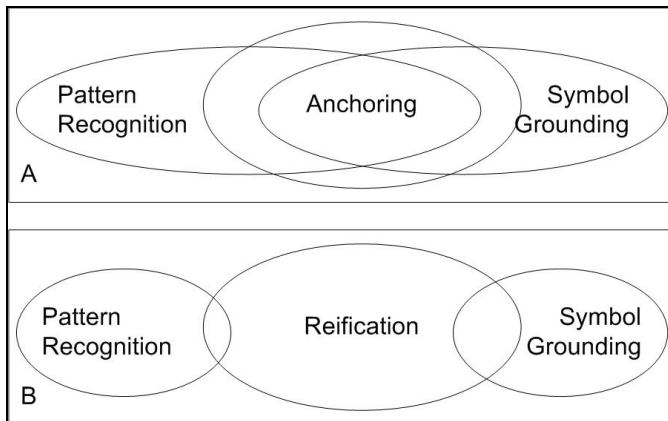


Figure 1 - Possible relationships of Pattern Recognition, Symbol Grounding, and Reification. In A, the problem of anchoring symbols to sensor/action patterns should be approachable by either top-down or bottom-up improvements. However, in B the problem cannot be solved by either top-down or bottom-up approaches, since there is no area of overlap. Rather, a third approach is required; one that solves the reification problem first, which then provides the bridge between symbol and sensors.

A. Bidirectional Approach

Reification is defined as “the process of regarding or treating an abstraction or idea as if it had concrete or material existence” [2]. In this case, it is the translation between the external world of concrete objects and the internal world of symbols. For a purely deliberative system that is manipulating abstract strings such as ‘block’ and ‘red’, these abstract symbols have no meaning other than the allowed manipulations in the symbol system. However, if these symbols are meant to refer to real-world objects or characteristics (i.e., if the things referred to have concrete or material existence) then these symbols must be reified to be effectively used. This is significant, since when one is deploying a robotic system into the real world, it is critical

that the system be able to respond quickly and correctly to rapidly changing world states. This requirement is echoed in the physical structure of biological systems, where much of the ‘higher cognition’ is performed outside the critical sense – react loop.

In a recent paper by Coradeschi and Saffiotti [8] the argument is made that the Symbol Grounding problem, as presented by Harnad [14] had features in common with Pattern Recognition. Coradeschi and Saffiotti argue that these two problems have an area of overlap as shown in Figure 1.A, which also overlapped with the anchoring problem. However, we believe that it is more likely that there is in fact no such area of overlap, and that the process of reification spans the gap between these two domains, as in Figure 1.B.

In recent research the terms ‘symbol grounding’ and ‘symbol anchoring’ have been used to describe the process as well. These terms are generally meant to capture one half of the reification process [14]. They are often used in the context of a lower ontology, in which symbols are defined in terms of other symbols, which are defined in terms of yet other symbols. To achieve some correspondence between the ontology and the external domain that it describes, some of these symbols must have a linkage to the ‘real world.’ This is the anchoring, or grounding, of the base symbols. This process is analogous to the terminal symbols in a formal grammar. However, it is only one half of the complete reification process.

There are two primary information flows that must be maintained to effectively connect symbols to objects: one is the flow from objects in the physical world onto the symbols, the second is from the symbols onto the objects. This problem is compounded by the fact that a symbol system typically does not have direct access to the objects in the physical world except via the mediation of the perceptual system. In short, this is the problem that we propose to solve by using the Reification Engine

B. Bi-directional mapping

To be effective, the Reification Engine must be capable of answering two fundamental questions:

1. What is this thing that is being perceived; and
2. How will the thing that corresponds to this symbol be recognized?

The system needs the ability to recognize the things in the real world that correspond to the symbols in the internal model. This is shown graphically in Figure 2.

The first necessary function is the mapping of symbols onto things. If the deliberative system has a reachable goal to achieve and a collection of operators that it can apply to modify the world, it can (with sufficient time and computational resources) find a sequence of actions or set of behaviors to achieve that goal. This has been a solved problem since the earliest days of artificial intelligence research.

However, for a robotic system to achieve this goal in the real world, that system must be capable of finding the things

in the real world needed to achieve each of the actions. It is one thing to produce the step 'Pick up the Red Block,' it is quite another to actually find the red block in the real world and grasp it.

The second necessary function is the ability to map things onto symbols. If one has a robot tasked to deliver mail around the office, it needs to be capable of noticing the stairs as stairs, not as a series of parallel lines on a level floor. Without this ability, it is not possible for the perceptual system to recognize exogenous changes to the world, which must be recognized either to take opportunistic advantage of conditions, or to avoid problems which crop up after the plan has been put into effect.

These two basic functions seem to be features common to almost all vertebrate brains¹. So it seems reasonable to begin by looking at the research into primitive vertebrate cognition.

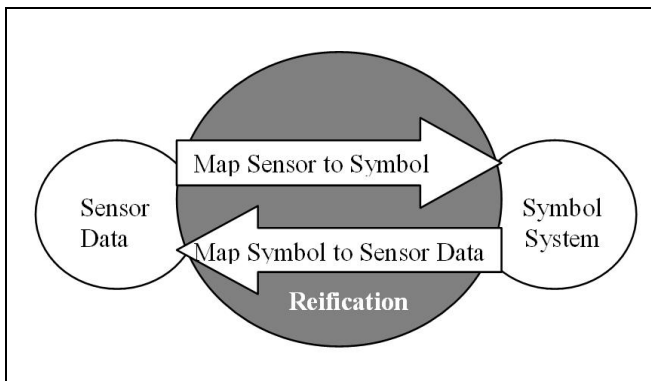


Figure 2 - Reification provides a bi-directional mapping between the symbol system used by the deliberative processing system and the sensor based system.

III. REIFICATION AND PRAEFERENCE IN BIOLOGICAL ENTITIES

For any species to survive, the members of that species must be able to sense and manipulate their environment so as to find food, avoid predators, and reproduce. In the case of vertebrate species, these survival mechanisms require the ability to map sensory data onto a neuronal representation, and to take the resulting behavior choices and map those onto motor actions. They must perform this bidirectional mapping between the sensory-motor systems and the (admittedly primitive) deliberative system continuously, efficiently, and in real time.

Discussing only the problem of finding food, they must be able to discover how their perceptions of the environment relate to the presence of food. For extremely simple, non-vertebrate species (e.g., amoebas) this might be a purely reactive mapping between chemical sensors on the surface

¹ We will restrict this analysis to terrestrial/amphibian vertebrates. Due to differential evolutionary pressures in oceanic environment, fish have taken a different evolutionary path (fish are weird - JPG).

and a gradient ascent behavior. However, for more complex (e.g., vertebrate) species, there is a mapping between the perception of sensory information, and some neuronal representation that is manipulated to assure survival. This is the process of reification.

Conversely, this vertebrate organism must be able, after sensing hunger, to know what features of the environment to use in the search for food. Current research [12] indicates that this is done by priming the sensory cortex with the sensations to expect after taking goal directed action. This is the process of preaffference. Both of these processes are discussed in more detail below.

While it is clear that humans can reify, it has been argued that more primitive biological entities are simply "hard wired" reactive systems. However, it can easily be argued that, in a changing environment, an organism that relies only on an inherited reactive system will be at a disadvantage to one that can reify. If this is true, one would expect to see reification in very primitive organisms. This leads to the question "How complex does a brain have to be before it can reify?"



Figure 3 - The Tiger Salamander has a brain structure that is extremely simple, but has all the core functionality of all land vertebrate brains. This makes it ideal for analyzing the 'bare minimum' needed for functional intelligence in the real world.

Salamanders (See Figure 3) have been used for decades by scientists researching brain function. While the nervous systems of all vertebrates have a common structural plan, the salamanders and their allied species have preserved a type of brain structure which closely resembles that of the most primitive amphibians [15]. These brains have most of the critical functional areas that are shared by all vertebrates, yet their brains are simple enough to allow clear research results.

For example, amphibians do have specialized, hard-wired “prey recognition” cells, which allow for the recognition of an object as potential prey [19]. This would suggest that they have the structure of a hard-wired reactive system. However, they also can also be trained to recognize a new scent, which implies that they are capable of reification of new sensory input [10]. It is interesting to note that this reification of new stimulus occurs without a cerebral cortex. The reification methodology described in this paper is guided by the example of these very primitive brains.

A. Simple Preference

Tiger salamanders also are capable of preference. Research has been done into the ways in which perception occurs in the brain [11][16]. One of the many interesting results of this research is that the electrical activity of the brain is chaotic, with information carried by a spatial pattern of amplitude modulation. Meaning is assigned to these patterns through the process of learning. This allows the brain to predict the outcome of behavior as preference. This preference makes it easier for the chaotic neuronal landscape to capture expected stimuli. This, in turn, makes it possible for primitive organisms, like the tiger salamander, to predict how actions taken will change their relations to the environment [11].

B. More Advanced Brains

Of course, it might be that primitive brains have some mechanism that is not present in more advanced brains, and so what we (as humans) do is different somehow. One of the reasons that the Tiger Salamander brain was chosen as a model, is that the core functions of all mammalian brains (including ours) have the same structural components as this primitive brain.

It is clear that humans have some sort of a reification mechanism. Artists have long known that we interpret visual images into familiar (if distorted) representations. One practice to overcome this mapping from the distal (external) image to a distorted proximal (internal) image is to copy a drawing by inverting it, and then drawing the upside down image. This allows the artist to duplicate what is actually there, rather than the interpreted image. Psychologists and philosophers have addressed this non-conscious automatic mechanism for much longer than artificial intelligence has been a discipline:

“We do not see patches of color, but trees and houses; we hear, not indescribable sound, but voices and violins.” [17]

It is clear that, in humans, the conscious mind deals not with low-level sensor data, but with symbols. It is also clear that when we look for things in the environment we do not look for “three orthogonal rectilinear surfaces of similar dimension, with a reflective electromagnetic signal with a wavelength of approximately 640 nanometers.” Instead we look for the ‘red block,’ and some non-conscious mechanism

translates this into the sensory/perceptual indicators that can be used to recognize the block when we see it.

IV. COMPUTATIONAL MODEL OF REIFICATION

The Summer 2004 issue of AI Magazine is devoted to cognitive vision, and the introductory article [6], stresses the need for the integration of the vision subsystem into the deliberative components in a bidirectional fashion – both the ability to map sensory data onto the symbol system, and the ability to map the expected state of the symbolic model into a form that the perceptual system can use to confirm or deny the expected state of the world. This is the function of a Reification Engine.

Many robotics researchers (See [1][3]) for an overview) have used biologically inspired models. While much of the current research focuses on hand-tuned, hard-coded mechanisms that are both task and hardware specific, there is a clear need for a more general model for the process of bridging the gap between sensor and symbol. Below, we sketch out a computational model.

A. Conceptual Model

The conceptual model of reification is fairly straight forward. If reification is the process of making a concept concrete, then each thing that is reified must have a conceptual part (a symbol), and a concrete part. In effect, reification is the establishment of a formal relationship between the symbol, and the perceptual cues that are used to identify the existence of the thing represented by the symbol.

$$\text{Exemplar} = (\text{symbol}, \{\text{cues}, \text{weights}\}, \{\text{actions}\}, \{\text{values}\}) \quad (1)$$

From this it is clear that an Exemplar is identified by the cues that it provides to the perceptual system (e.g., how it is recognized), and by the actions it affords (e.g., what can we do with it). Thus, when the perceptual sub-system encounters a collection of sensory-data, it can identify the Exemplar that is associated with the data, and if the deliberative system determines that a specific action is needed to complete a task, all Exemplars that can provide that action can be located, and from their cues, the necessary preference can be set up in the perceptual sub-system. This supports the needed bidirectional mapping needed to bridge the gap between sensor and symbol from either direction.

It should be noted that these cues and weights are identical to the cues and weights used in the lens model proposed by Egon Brunswik in his work on perception [5]. In this model a judgment is made about an object by using the cues or features of that object and weights placed on those cues. These weights are updated *a posteriori* by the ecological validity of the judgment. More information about the lens model and its applications can be found in [7].

Exemplar: The Exemplar is the encapsulation of the bridge between the perceptual and sensory attributes and affordances of a reified object and the symbolic representation of the same object in the symbol system.

Symbol: The symbol is the lexical tag used by the symbol system to refer to the object. It acts as a key into the symbol system, and provides the linkage that enables the bi-directional mapping between the symbol system and the perceptual system.

{Cues,weights}: The cues and weights are the recursive multi-modal sensor signatures that can be used by the Reification Engine to identify the symbol that describes the object in the sensory data field. Each cue has an associated weight, and they both are used to assess the confidence in the identification of a perceived object with a specific symbol. It should be noted that that for a specific object the cue set will be a subset of all the possible cues. The cues and weights are also used to pre-load the perceptual system to enable the system to ‘look for’ an expected object.

{Actions}: Objects are not always passive entities in a complex environment. Rather, just as they have static attributes, they can also have dynamic attributes. For example, in recognizing prey, the Tiger Salamander uses color, scent and relative motion patterns. Thus a twig of approximately the correct color and shape that is moved in the correct pattern will cause the salamander to strike at the twig, until the additional sensory cues such as taste and texture will cause the salamander to spit the twig out. This is the same mechanism used daily by people fishing with artificial lures. So the identifying action patterns of the object are also a critical component of the signature of the object.

{Values}: A hallmark of intelligent behavior is to not repeat mistakes. The mechanism that allows us to quickly dismiss candidate solutions which previously resulted in unacceptable outcomes must have some analog in the cognitive models of our robotic systems. While much of this mechanism may operate at a conscious level, it is clear that portions of the mechanism function non-consciously. Antonio Damasio argues that a critical aspect is the non-conscious assignment of values to the actions and objects manipulated by our conscious minds [9]. The Value set is used in the Exemplar to store these value assignments.

B. Exemplar Library

The Exemplar Library is a database of Exemplars. It is used as the repository for objects ‘known’ to the system, and in fact it defines the knowledge of the system. The Reification Engine maintains the library, and pulls instances of the Exemplars (See Figure 4) which can be loaded into the perception/action system to enable preference, or, in the case of sensory data that is not classified, the Reification Engine can pull additional Exemplar instances in an attempt to find a match. Since each Reification Engine is embedded into a specific hardware system, the cues utilized by each engine will be dependent on the sensors and actuators available to the system. It would be pointless to include detailed visual cues for an object if the robot were not equipped with a visual system. It would be equally pointless to include latitude and longitude information for a waypoint unless the robot had a GPS or some equivalent positioning system. So each set of cues will

be idiosyncratic to the specific hardware on the system. However, the mechanisms used by each Reification Engine would be similar on both platforms.

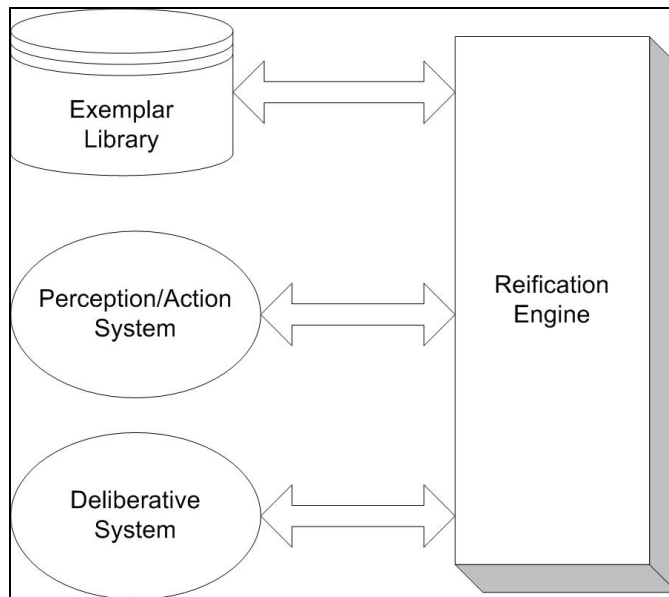


Figure 4 – Generalized architecture of a Reification Engine. It provides support for both the deliberative system and the Perception/Action system, and enables bidirectional mapping between the two.

The Exemplar Library corresponds in many ways to the association cortex (cortices) of the brain. This distributed cortex functions as the bridge between the various modal sensory cortices which are structurally mapped to their sensors, and the more symbolic processing systems of the semantic maps. The association cortex is the mechanism that allows us to recognize a ‘dog’ from any of its various sensor signatures (e.g., vision, sound, tactile, presumably scent) [20].

C. A note about hierarchical classification of things

Much of the research from the cognitive sciences and psychology suggests that biological entities (humans, at least) seem to build complex hierarchical classification structures which are used to both recognize objects in the real world, and to enable the grounding of symbols which have little or no physical presence. Stevan Harnad argues that the representation of zebra is simply the conjunction of the grounded symbols for horse and stripes [14]. However, it is unclear where in the process of recognition this occurs. Is the Reification Engine loaded with every grounded symbol, as well as every possible combination of symbols (effectively the power set of the symbols), or does the Reification Engine work only with those grounded symbols, and the responsibility for manipulating the possible combinations of percepts belongs to the deliberative, symbol processing system. We take this latter view, which allows the complex machinery of truth maintenance, and generalization to function external to the time critical, sensor-driven reification

process. This is also in line with the neurophysiology data that suggests that classification and generalization does not occur in the posterior areas of the brain, rather they belong to the higher brain functions, associated with the frontal cortex.

V. DESIGN OF A REIFICATION ENGINE

The basic design presented above needs to be both encoded and embedded into a system to be analyzed. In this section we present some of the technical approaches as well as some of the challenges that must be addressed.

We assume that the Reification Engine is part of an embedded system. If the system has no embedded components, if it has no direct interface with a dynamic, uncertain environment, there is no need for reification. We further assume that it is also a part of a deliberative, goal directed system. If there is no need for symbolic processing, then there is no need for a bridge between the perception/action system and the symbol system.

While these two criteria may not obtain in some specific cases, in practice, most functional embedded systems require some level of goal directed behavior (unless they are simple toys), and most goal directed systems have some need to maintain an interface to data which is external to their symbol system. Even a system as apparently simple as an automated vacuum cleaner has problems dealing with its environment. In a recent article from The Wall Street Journal rating commercially available robotic vacuum cleaners, the conclusion was:

"[O]ur tests on three popular ones found major disappointments with all of them with regards to doing a decent job of cleaning, which should be the only true criterion of having your vacuuming automated." [21]

The author went on to say that one system "was like a drunk driver, banging into my furniture and spinning about aggressively without any real plan in mind." Clearly, from the point of view of the end user, even something as simple as vacuuming a floor has a significant deliberative component, which, if absent causes the system to fail at its primary task.

Given that we have a task with both embedded aspects and deliberative requirements, matching the hardware to the software is going to be a major component of the design process. The Reification Engine is intended to be the bridge between the perception/action system and the deliberative system; therefore it must have one foot on each shore. However, a competing design goal is to make the engine as general as possible. Any particular instantiation of the engine will be tightly coupled to its particular hardware and goals, but the same engine core will be used across many different instantiations.

A. Hardware

The design for the Reification Engine is based on the assumption that the intelligent system will be a significantly parallel device, utilizing a heterogeneous set of computational

devices. This will closely model the functional units in the vertebrate brain. Just as functional units in the brain vary in computational power and interconnectivity, the computational devices will vary in architecture and computational power. Most of the computational power will be allocated to the Reification Engine, which loosely corresponds to the associational cortices, which receive both raw data and preprocessed data from specific modal cortices.

B. Execution Monitor

Let us imagine an autonomous hazardous materials (HAZMAT) response vehicle, equipped with LIDAR, vision, GPS, and a whole boatload of other sensors (See Figure 5, which is our low cost field robotics test chassis.) Tasked with the responsibility to place a specific area under observation and report on any release or detection of toxic materials, it consults its topological and hypsographic databases, and decides to travel down the hill, cross the river on the existing bridge, and position itself just below a ridgeline, where it can safely observe its assigned area. (O.K., so maybe it is a little advanced).

As it completes its travel down the hillside, a constantly running execution monitor has been checking position, orientation and velocity, and signals that the action has completed successfully. The next action is to locate and orient on the bridge so that it can cross the river. Based on the expected end state of the previous action, the system has an expectation of the current configuration of the world, and from this an expectation of what its sensor data should be like.

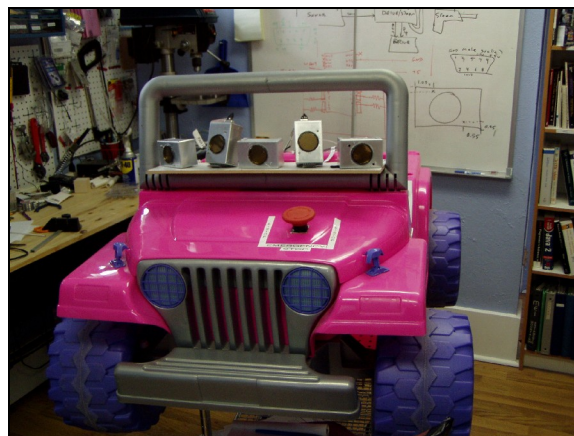


Figure 5 - Simple, low-cost, field robot test chassis. Currently undergoing sonar based reification testing.

Since it expects the bridge to be in front of it, it can preload the perception/action system with what the bridge should look like (to the available sensors). Rather than operating without any *a priori* knowledge, and treating the sensor data as a blank slate, the system preloads the expected cues that would indicate the bridge is right where it is expected to be. This allows the system to reduce the cognitive load to a presumptive testing strategy. If the sensory data supports the

presumptive hypothesis, no further symbolic or deliberative processing is needed, the HAZMAT response vehicle has a valid plan, knows that it has executed correctly so far, and it knows what to do next. So it does it, completing the transition of the bridge, and preloading the next expected state. Only if the presumptive tests fail will the higher level (expensive) cognition systems be brought back into the loop.

This is similar to the everyday experience of suddenly realizing that you have completed the drive home, even though your last conscious memory of driving was 5 miles and fifteen minutes ago. Both the work by Freeman (already cited) and work by the psychologists Bargh and Chartrand indicate that the vast majority of deliberative action is processed not by the expensive, conscious, cognitive systems but by non-conscious systems [4].

C. Perception/Action System

The perception/action system must be designed to integrate with the Reification Engine. To do so it must support both directions of the bidirectional mapping. It must have hooks that enable the Reification Engine to submit a collection of ‘look for this’ cues derived from the expectation of what the world should look like. It must be able to take a significant number of these cues and attempt to match them on a loosely parallel way, and signal either that one or more pattern has hit, or that none of the expectations are met by the current sensory data.

The second component is the ability to communicate to the Reification Engine, that ‘I have found something with these cues – are you interested?’ If an intelligent system is going to be capable of opportunistic re-planning and responding to unexpected changes in the world, it must be capable of noticing and identifying those unexpected world states.

D. Deliberative System

The deliberative system must also be designed to integrate with the Reification Engine. It must be capable of producing a symbolic mapping of expectations as the result of applying actions to the world. These actions will result in changes to the world, which should result in corresponding changes to the systems perception of the world. The Reification Engine can take the symbolic expectations and map those (via the library) into the sensory expectations required for preference. The deliberative system must also accept responses from the reification system indicating that the world is changing pretty much as expected, or that something unexpected occurred. In this case the deliberative system can determine whether the deviations are significant or if the current action sequence can continue to be executed.

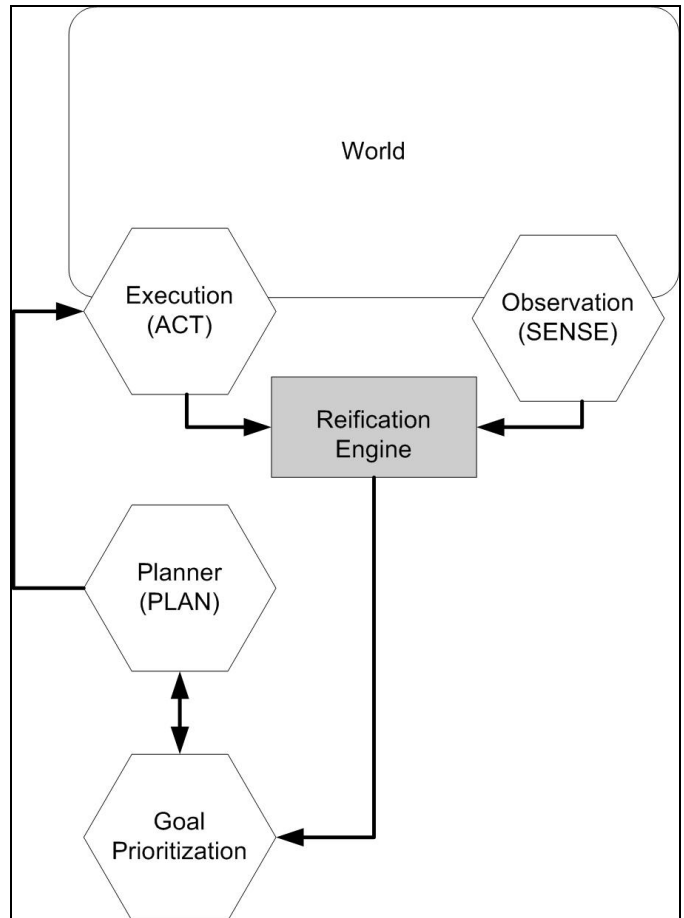


Figure 6 - The high-level Control System of the HAZMAT AGV

VI. FUNCTIONAL DESCRIPTION

The following example is taken from ongoing work assessing the performance of an autonomous ground vehicle (AGV). The system is based on a probability-aware planning and execution system which is designed to function as the deliberative component of a complete intelligent system [13].

The general functional model (See Figure 6) is a goal assessment loop which makes calls to a perception action component to determine whether any system goals are unmet. If any such goals are found, the deliberative system updates its symbolic world model, using current information provided by the perception/action component. In the following discussion, this information is mediated by the Reification Engine, although this capability was not available in the original analysis. The general model is a modification of the Elementary Loop of Functioning (ELF), proposed by Meystel and Albus [18].

Once the symbolic model is updated, the probability-aware planning component searches for and selects an action sequence which has the highest assessed probability of achieving the goals. This action sequence is made up of individual actions, and the expected salient world state that should result from the successful execution of the action. The

execution system emits the first of these actions to the perception/action component for execution by the system.

Since the deliberative component has only symbolic representations of the world state, the Reification Engine is responsible for mapping the symbolic expected state into a sensor based representation that can be utilized by the perception/action component directly.

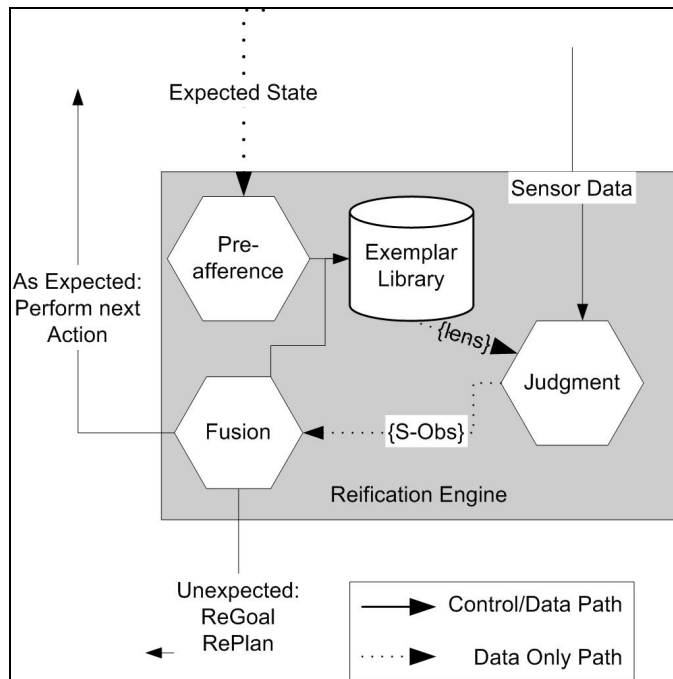


Figure 7 - Internal details of the Reification Engine.

VII. CONCLUSION

In this paper we present the concept of a Reification Engine, which we believe is a necessary component of any intelligent embedded system. The Reification Engine provides a bidirectional mapping between the crisp, symbolic aspects of deliberative cognition, and the dynamic and imprecise aspects of perception and action in the real world. We argue that there is a gap between the symbolic level and the sensor/effector level which cannot be bridged without this ability to provide bidirectional mapping, and that this lack has been a significant obstacle to the deployment of true embedded intelligent systems.

We outline the necessary functions of the Reification Engine and from these requirements detail the specifications of the data structures and operations of the engine. From these specifications we discuss the implementation details of the Reification Engine, and the high level design of a complete, integrated intelligent embedded system. This is placed in the context of a field robot, tasked with providing support to HAZMAT incident responders.

REFERENCES

- [1] Albus, J. S. and Meystel, M. M. *Engineering of Mind*. New York: Wiley and Sons, 2001, ch. 2, pp. 21-55.
- [2] *American Heritage Dictionary of the English Language*. New York: American Heritage Publishing Co., Inc., 1969, pg. 1097
- [3] Arkin, R. C. *Behavior based Robotics*. Cambridge, MA: The MIT Press, 2000, ch. 2, pp.31-63.
- [4] Bargh, J. A. and Chartrand, T. L. "The Unbearable Automaticity of Being," *American Psychologist*, vol. 54, no. 7, pp. 462 – 479, 1999.
- [5] Brunswik, E. *Perception and the Representative Design of Psychological Experiment*, Berkeley, CA: University of California Press, 1947.
- [6] Christensen, H. I. "Cognitive Vision," *AI Magazine*, vol. 25, no. 2, pp. 8-9, 2004.
- [7] Cooksey, R.W., *Social Judgment Theory*. San Diego, CA: Academic Press, 1996.
- [8] Coradeschi, S. and Saffiotti, A. "An Introduction to the Anchoring Problem," *Robotics and Autonomous Systems*, vol. 43, no. 2-3, pp. 85-96, 2003.
- [9] Damasio, A. *The Feeling of What Happens*. New York: Harcourt, Inc. 1999, ch. 6, pp. 168-194.
- [10] Dorries, K.M., White, J., and Kauer J.S. "Rapid classical conditioning of odor response in a physiological model for olfactory research, the tiger salamander," *Chemical Senses*, vol. 22, pp. 277-286, 1997.
- [11] Freeman, W. J. *How Brains Make Up Their Minds*. New York: Columbia University Press, 2000.
- [12] Freeman, W. J. "Perception of time and causation through the kinesthesia of intentional action," *Cognitive Processing*, vol.1, pp. 18-34, 2000.
- [13] Gunderson, J. P. *Probability-Aware Planning and Execution System*. Ph.D. Dissertation, University of Virginia, 2003, ch.5 pp. 59-83.
- [14] Harnad, S. "The Symbol Grounding problem", *Physica D*, vol. 42, pp. 335-346, 1990.
- [15] Herrick, C.J. *The Brain of the Tiger Salamander*. Chicago: The University of Chicago Press, 1948.
- [16] Kay, L. M. and Freeman W. J. "Bidirectional Processing in the Olfactory-Limbic Axis during Olfactory Behavior," *Behavioral Neuroscience*, vol.112, no. 3, pp. 541-553, 1998.
- [17] Lewis, C. I. *Mind and the World-Order*. New York: Scribner's, 1929, pg. 119.
- [18] Meystel, M. M. and Albus, J. S. *Intelligent Systems*. New York: Wiley and Sons, 2002, ch. 4, pp. 158-187.
- [19] Roth, G., "Neural Mechanisms of Prey Recognition: An Example in Amphibians" in *Predator-Prey Relationships: Perspectives and Approaches from the Study of Lower Vertebrates*. Edited by Feder, M. E. and Lauder, G.V., The University of Chicago Press, 1986.
- [20] Solms, M. and Turnbull, O. *The Brain and the Inner World*. New York: Other Press, 2002, pp. 30-31
- [21] Swisher, K. "The Only Thing Latest Robot Vacuums Can't Do is Clean," *The Wall Street Journal*, 15 July, 2004, pp. D1.

The Autonomy Levels For Unmanned Systems (ALFUS) Framework

Interim Results

Hui-Min Huang

National Institute of Standards and Technology
100 Bureau Drive, Mail Stop 8230
Gaithersburg, Maryland 20899
hui-min.huang@nist.gov

Abstract

Unmanned systems have become more and more widely used. Yet, various practitioners have different perceptions and different expectations of the systems. They also have different definitions for the term autonomy and different concepts about how it should be measured. We have been developing a framework called Autonomy Levels for Unmanned Systems (ALFUS) that aims at providing a common reference for the communication and the evaluation of the autonomy capabilities of unmanned systems. The framework is under development. This paper is a work-in-progress report on some key areas.

Keywords

autonomy, robot, environment, human independence, mission, task, unmanned system

I. INTRODUCTION

A wide range of applications has been either exploring the feasibility of, or actually employing, unmanned systems (UMS). Examples include aerial reconnaissance, bomb detection and disposal, combat support, urban search and rescue, physical security, and intelligent transportation systems [1, 2, 3, 4, 5, 6, 7, 8]. Figure 1 shows a robot searching for victims through a collapse scene.

Practitioners across those application domains have different perceptions and different expectations of the systems. In addition, the term autonomy has been interpreted differently in different areas. The methods with which autonomy is measured lack consistency. For example, Air Force Research Laboratory (AFRL) has established Autonomous Control Levels (ACL) [9]. The Army Science Board has described a set of levels of autonomy [10]. Many programs need only remotely controlled UMSs. For other programs, fully autonomous operations were required. Traditionally, autonomy has been perceived as the amount of the human interaction required. However, when analyzing the requirements for UMS, one must consider what kinds of tasks or missions are planned for the UMS and in what kinds of environments the UMS will operate. We have been coordinating a cross Government and industry, ad hoc

working group on developing a comprehensive framework for autonomy. The framework is called Autonomy Levels for Unmanned Systems (ALFUS) [11]. ALFUS aims at defining key autonomy issues, providing a commonly understood framework for communicating UMS autonomy issues, and evaluating UMS autonomy capabilities. Some early concepts have been reported since 2003 [12, 13]. This paper is a work-in-progress report stressing some key areas.



Figure 1: Robot searching through a wood pile for victims

II. ALFUS FRAMEWORK

The ALFUS Framework has been under development since 2003. The ALFUS framework:

- includes a generic model that can be instantiated for program specific models
- contains a metrics based model for autonomy levels that is flexible, quantified, and with smooth transitions
- employs multiple layers of abstraction in expressing autonomy requirements and capabilities
- is applicable to UMSs with various kinds of configurations
- is extendable as a general performance metrics model for unmanned systems.

Thus, the model is designed to apply to single UMS low-level operational behavior as well as multiple-UMS, high-level missions.

The first effort for this group was to define a set of terms. The group has reached consensus on defining UMS as: “An electro-mechanical system, with no human operator aboard, that is able to exert its power to perform designed missions. May be mobile or stationary. Includes categories of unmanned ground vehicles (UGV), unmanned aerial vehicles (UAV), unmanned underwater vehicles (UUV), unmanned surface vehicles (USV), unattended munitions (UM), and unattended ground sensors (UGS). Missiles, rockets, and their submunitions, and artillery are not considered unmanned systems [1, 14].”

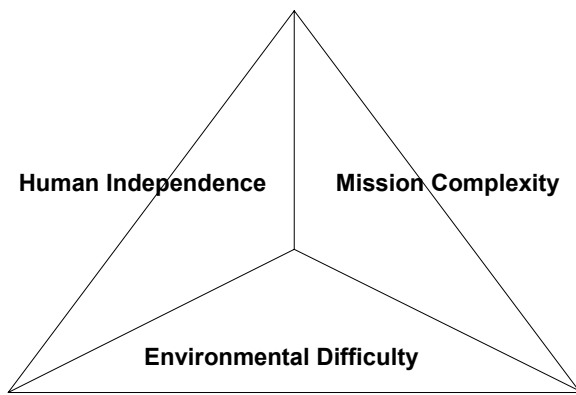


Figure 2: ALFUS Metric Overview

The group defined autonomy as “UMS’s own ability of sensing, perceiving, analyzing, communicating, planning, decision-making, and acting, to achieve its goals as assigned by its human operator(s) through designed HRI [14].” This emphasizes that, in the ALFUS model, the more the robots are able to serve human purposes, the higher the autonomy level would be. It is also proposed by the group that this model is to be called intelligent autonomy.

A fundamental concept for the ALFUS framework is that human interactions, types of tasks, teaming of UMSs and humans, and operating environment are the essential issues that need to be accounted for when characterizing UMS. This understanding leads to the three-aspect view in ALFUS for characterizing the autonomy of unmanned systems. The three aspects are:

- mission complexity
- human independence
- environmental difficulty

as shown in **Error! Reference source not found.** Each of the aspects contains a set of metrics, which will be described later in this paper.

The ALFUS Framework also emphasizes a generic model that can be instantiated for a program specific autonomy model. Figure 3 depicts the framework, which includes the following items:

- Terms and their definitions. Standard terms are defined to facilitate communicating UMS autonomy and ALFUS framework description.
- Detailed model. The aforementioned metrics form the detailed model of ALFUS. The metrics are applied to the UMS and scores are accumulated as the autonomy level of the UMS. This application process will be described, in detail in a later section.
- Executive model. Metrics descriptions are summarized to form this conceptual, high level autonomy model. While the detailed model facilitates technical development and evaluation of UMSs, the executive model facilitates communications among users and program managers.

While the generic model covers many types of UMS, individual programs would derive specific ALFUS models according to the programs’ emphases and particular needs. This paper focuses on the detailed model and how it may be applied.

III. ALFUS DETAILED MODEL METRICS

Efforts have been dedicated to the foundation of the ALFUS framework, the metrics. There have been many iterations due to the following challenges:

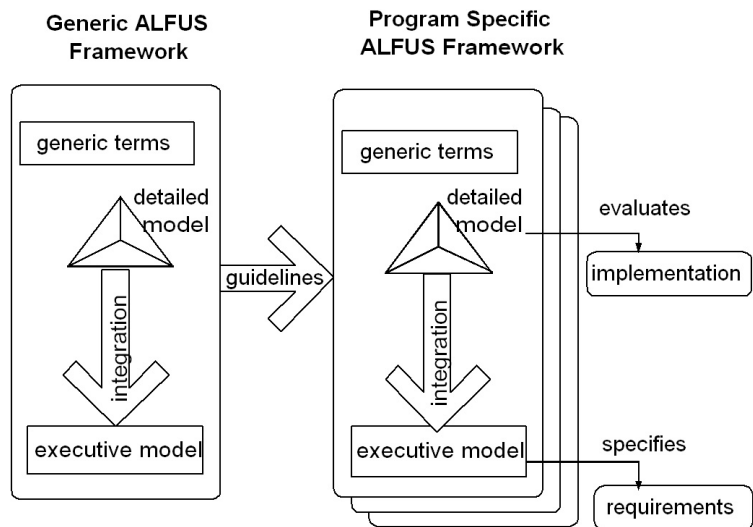


Figure 3: The ALFUS Framework

- Some of the metrics seem to be subjective in nature.
- Some of the other metrics are still being actively studied by the corresponding research communities and the measures/scales are not well defined yet.
- Participants might have been familiar with different analysis methods which might have resulted in overlapping, gaps, varied emphases, or different understanding on the contributed metrics.

The following sections describe the latest metrics.

A Human Independence

Following the model shown in **Error! Reference source not found.**, we propose that the following metrics facilitate in-depth analyses of the human independence (HI) requirements for UMSs:

1. Operator interaction time. How much time does an operator have to interact with the UMS, relative to the whole mission execution and completion time?
2. Mission Planning ratio. What percentage of a mission is to be planned by an operator and by the UMS?
3. Level of interaction. Does the operator only have to assign a mission? Does he have to also assign strategic goals and/or tactical goals of the mission? Does she/he have to provide detailed plans? Auto-piloting?

4. UMS initiation. How well is the UMS able to communicate to the operator? Is the UMS able to identify and communicate to the appropriate operator with proper information, such as a problem report and at the proper time? Does the UMS only respond to operator's requests? Does the UMS wait for proper input before it can proceed with its mission execution?
5. Operator workload. What is the workload for the operator during the UMS performance of missions? Is the operator highly stressed? Is the operator fully occupied but handling the tasks comfortably?
6. Training. What levels of training are required to operate the UMS? Does it take a highly skilled operator? Would a novice be able to operate the UMS?

B Mission Complexity

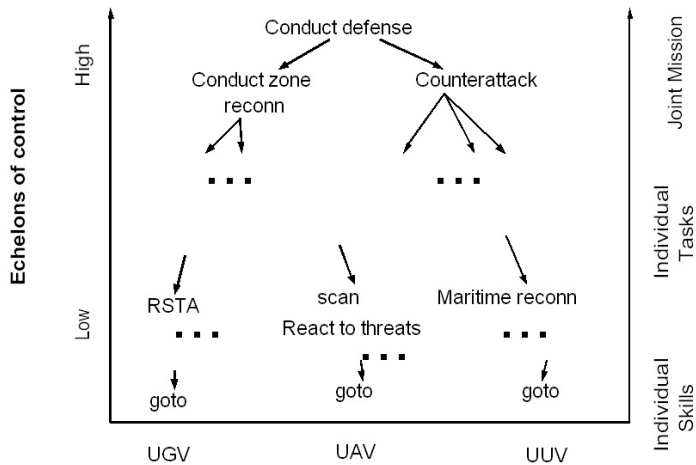
In analyzing the complexity of UMS missions, the following metrics should be included:

1. Task decomposition, or task integration in the reversed direction. What is the width and depth of task decomposition for a mission? A full-scale decomposition of a military mission could include the following levels: battalion, company, platoon, vehicle, skills, primitive, and actuator. There could even be levels higher than battalion. Multiple types of vehicles could be involved to conduct joint missions at these high levels. Figure 4 illustrates the point.

A simplified task decomposition model considers only three levels, namely, group tasks, vehicle tasks, and skills. In this model, the tasks that are at levels lower than skills implicitly affect the degrees of complexity of the corresponding skills. The similar argument can also be made for the high-level tasks.

2. Type of tasks. Would the task be mission level, groupings of high risk, highly complex tasks, single subsystem tasks, or actuator tasks? How many vehicle functions are involved-- Mobility, System C4 (Command, Control, Communications, Computers), Lethality, Survivability, Tactical C4, ISR (Intelligence, Surveillance, and Reconnaissance), and Support? How many vehicles are involved? How many subtasks or skills are needed?

3. Complexity of tasks. Is the mission coupled with other missions? What is the level of uncertainty of the mission? What is the required level of precision? What are the rules of engagement (in military situations)? The following factors facilitate the evaluation of the complexity: (i) numbers of transitions and state and



Multiple Types of Unmanned Systems
Figure 4: Task decomposition and integration

their ratios, depth/breadth of search tree (ii) numbers of concurrent tasks.

4. Decision space structure. What are the knowledge requirements--number of knowledge types and associated confidence levels, such as signals, entities, events, images, maps, laws of physics, and cultural values? What are the temporal and spatial resolutions for task execution? What are the required safety and risk levels? What are the rates of changes of tasks?
5. Collaboration level. The highest level of collaboration for a UMS team would be mission level collaboration and parenthetical understanding of mission intent. Detailed factors include number of communication channels, types of data exchanged, frequency of the data exchange, and synchronized vs. asynchronized operations.
6. Dynamic planning. The UMS's capability to perform planning onboard and in real-time indicates how it might be able to handle dynamic and changing environments and missions.
7. Analysis. Capability of values/cost and benefit/risk analysis.
8. Situation Awareness (SA): The highest level SA is omniscience. Below, the SA metric scale goes, from high to low: at the strategic level, at the tactical level, and at the internal level. At each level, the SA metric is further divided into, from high to low: projection, comprehension, and perception.

C Environmental Difficulty

UMS autonomy includes finding task solutions, navigation and others, for every environmental situation. The solutions should be characterized with respect to their difficulty levels. For example, if a UMS needs to cross a bridge, a solution may or may not exist depending on the width of the bridge. Even if the bridge is crossable, execution difficulty varies widely in different situations. Therefore, the task solution should be indexed with the difficulty level. The level of difficulty could be measured as:

- beyond the UMS's physical capability: for example, when the bridge is narrower than the width of the UMS. In other words, the identified, "apparent solution" is actually not a solution.
- restrictive: when there is clearance but requires high level perception, planning, and execution capabilities of the UMS.
- unrestrictive: open space and does not require advanced capabilities.

Environmental difficulty is evaluated with the concept of solution ratio, which is the ratio of the number of total possible choices a UMS can make and the number of

solutions that meet the mission/task objectives. Individual programs can set their own thresholds on the difficulty, based on cost/benefit/risk factors, and determine whether to accept or reject a "solution."

IV. APPLICABILITY ISSUES

The application of the metrics to the targeted UMS is illustrated in Figure 5. Individual metrics are applied to mission tasks and scores are averaged. One of the challenges during the development effort is to generate quantifiable scales for all the metrics, including those metrics that are rather subjective in nature.

Metric weight is another important factor. Since the metrics are developed for general purposes, weights should be used to ensure that the metrics are applied appropriately. Individual, weighted scores are added and averaged to derive a final score, which will be the autonomy level for the UMS.

The ALFUS framework is scalable. It can be applied to vehicle subsystems as well as large teams of vehicles. Figure 6 illustrates that a metric scoring form can be applied to UMSs with various configurations.

The ALFUS framework is extensible. The metrics can be used to measure not only for autonomy measures but also for general performance measure, once the metric scales are modified to reflect the performance requirements.

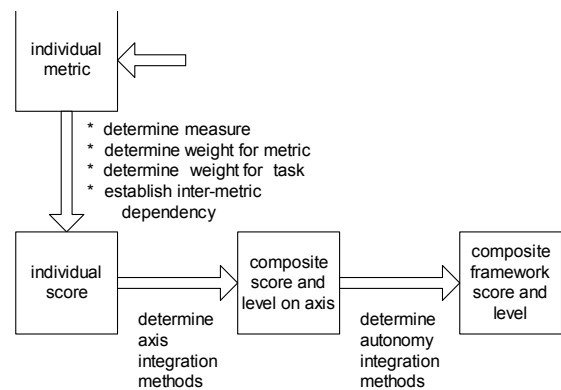


Figure 5: ALFUS Application Process

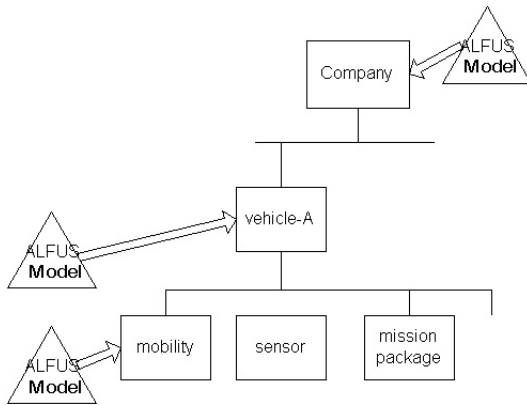


Figure 6: Applying a metric scoring form to different UMS Configurations

V. SUMMARY

The key concepts and critical elements of the ALFUS model are presented. The ALFUS framework applies metrics to unmanned systems for characterizing their autonomy capabilities. The framework is intended to be both scalable and extensible. It intends to provide detailed measures as well as high-level definitions for the UMS autonomy. ALFUS is an ongoing project with some interim results published

References

- [1] Joint Robotics Program Master Plan FY2005, OUSD, Pentagon, Washington, D. C.
- [2] Arthur, K., "Making UAVs Tactically Smarter," Proceedings of SPIE, Volume 5804, Unmanned Ground Vehicle Technology, Orlando, Florida, March 2005.
- [3] Neely, H.E. III, "Multimodal interaction techniques for situational awareness and command of robotic combat entities," 2004 IEEE Aerospace Conference Proceedings, Big Sky, MT, USA, March 2004.
- [4] http://www.nist.gov/public_affairs/techbeat/tb2005_1007.htm#dhs
- [5] Carroll, D. M., et al., "Development and testing for physical security robots," Proceedings of SPIE, Volume 5804, Unmanned Ground Vehicle Technology, Orlando, Florida, March 2005.
- [6] Stormont, D.P., "Autonomous rescue robot swarms for first responders," 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Orlando, FL, USA, March 2005.
- [7] Jacoff, A., et al., *Test Arenas and Performance Metrics for Urban Search and Rescue Robots*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003.
- [8] Da Silva, BC, et al., "ITSUMO: An intelligent transportation system for urban mobility," Innovative Internet Community Systems Lecture Notes in Computer Science 3473: 224-235 2006
- [9] Bruce T. Clough, "Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway?" Proceedings of the Performance Metrics for Intelligent Systems Workshop, Gaithersburg, Maryland, 2002.
- [10] Army Science Board, Ad Hoc Study on Human Robot Interface Issues, Arlington, Virginia, 2002.
- [11] http://www.isd.mel.nist.gov/projects/autonomy_levels/
- [12] Hui-Min Huang, Elena Messina, James Albus, "Autonomy Level Specification for Intelligent Autonomous Vehicles: Interim Progress Report," 2003 PerMIS Workshop, Gaithersburg, MD.
- [13] Hui-Min Huang, Elena Messina, Ralph English, Robert Wade, James Albus, and Brian Novak, "Autonomy Measures for Robots," Proceedings of the 2004 ASME International Mechanical Engineering Congress & Exposition, Anaheim, California, November 2004
- [14] *Autonomy Levels for Unmanned Systems Framework, Volume I: Terminology, Version 1.1*, Huang, H. Ed., NIST Special Publication 1011, National Institute of Standards and Technology, Gaithersburg, MD, September 2004.

ABSTRACT

Meaningful Metrics and Evaluation of Embodied, Situated, and Taskable Systems

Douglas W. Gage
XPM Technologies

Because, even after decades of technology development, our systems are not yet capable of operating fully autonomously, one obvious metric is some characterization of the amount and types of human participation required to achieve system functionality. One approach being pursued is to define an ordered set of “levels” of autonomy. This paper argues that, while the amount of intervention required for successful performance of a specific task in a specific environment is a valuable metric, at the system level it is necessary to focus on the definition and implementation of multiple specific modes of user interaction and intervention. While “levels” of autonomy can and should be measured during system operation, it is the specific system operating modes which must be defined and developed.

The key challenge standing in the way of fully autonomous systems is perception. Our systems are often unable to abstract from their sensor inputs representations of those salient features of the environments in which they are situated required for the performance of their tasks. For example, the ability to retrace a previously traveled route (without GPS) depends on “learning” key features (e.g., landmarks) along the route, and being able to detect and identify them in various conditions of lighting, weather, sensor calibration, sensor point of view, and so forth. Given some definition of “successful operation,” we desire a system which can “operate successfully” in the broadest possible range of environments and conditions. This is a “domain-referenced” metric (over what range of experimental input conditions do we succeed), rather than the usual “range-referenced” metric (what is the value of the measured outcome for a given input condition).

The evaluation of a system's performance in terms of metrics involves two components: measurement and scoring. Management considerations focus on the development and execution of a Test Plan – the identification of a test site, installation of instrumentation, hiring of observers, and running a series of discrete tests under contrived conditions. The measurements resulting from the experimental trials then flow into the scoring process. Management cares a lot about scoring, scores are often used in ways that can have profound financial consequences. How will the results of individual experimental measurements be combined, and what will be the thresholds for ultimate project success and interim acceptable progress?

While “bottom line” aggregated scoring metrics may be important for management purposes -- e.g., to decide whether a development project is making enough progress to justify continuation -- unaggregated measured data can serve other critical needs. First, while the project office needs to be able to claim that the system can work in XX% of tactically relevant environments, what the user of the deployed system really cares about is whether the system can perform THIS

task, in THIS environment, under THESE conditions. The user needs to have a reasonable expectation of what the system could do if deployed, needs to know how to tell the system what to do, and needs to be able to determine both what it is doing while it is doing it and what it actually accomplished when it is finished. Unaggregated measured data can also support the continuing development process -- knowing exactly what the system did under exactly what conditions makes it easier to know how to improve the system.

The fact that our systems will ultimately be deployed in a huge diversity of environments and conditions, and that system performance often depends sensitively on the details of environment and condition, suggests a strategy that integrates continuous performance evaluation tightly into the development process for situated systems. The system should incorporate continuous logging of its sensor inputs, internal states, and behaviors. This should be done throughout the complete lifetime of the system -- throughout its deployment as well as during initial development. This will provide users and developers with hard data to support system adaptation and on which to base discrete product improvements -- “we could have reduced operator interventions 43% over the past 3 months in Iraq if our system could discriminate between X and Y.”

Autonomous robots -- embodied, situated, and taskable machines -- are notoriously hard to debug, and one reason is because developers have traditionally not invested up front in the tools to log the data that could make debugging trivial. With the cost of sensors, processing, and (especially) memory storage continuing to decline rapidly, it is time to make the wholesale logging of data standard practice.

Fault-Tolerance Based Metrics for Evaluating System Performance in Multi-Robot Teams

Balajee Kannan and Lynne E. Parker

Distributed Intelligence Laboratory, Department of Computer Science

The University of Tennessee, Knoxville, TN 37996-3450

Email: {balajee, parker}@cs.utk.edu

Abstract—The failure-prone complex operating environment of a standard multi-robot application dictates some amount of fault-tolerance to be incorporated into the system. Being able to identify the extent of fault-tolerance in a system would be a useful analysis tool for the designer. Unfortunately, it is difficult to quantify system fault-tolerance on its own. A more tangible metric for evaluation is the “effectiveness” [8] measure of fault-tolerance. *Effectiveness* is measured by identifying the influence of fault-tolerance towards overall system performance. In this paper, we explore the significance of the relationship between fault-tolerance and system performance, and develop metrics to measure fault-tolerance within the context of system performance. A main focus of our approach is to capture the effect of intelligence, reasoning, or learning on the effective fault-tolerance of the system, rather than relying purely on measures of redundancy. The developed metrics are designed to be application independent and can be used to evaluate and/or compare different fault-diagnostic architectures. We show the utility of the designed metrics by applying them to a sample complex heterogeneous multi-robot team application and evaluating the effective fault-tolerance exhibited by the system.

I. INTRODUCTION

To scale the use of robots from simple tasks in static environments to large-scale applications, they must be able to effectively and robustly coordinate many different functionalities. Multi-robot teams applied to complex applications will typically require robot team members to perform multiple tasks such as planning, mapping, localization, formation-keeping, information sharing, and so forth. These functionalities are especially useful for applications involving dynamic environments such as urban search and rescue and future combat systems.

However, the nature of these operating environments are such that faults often develop during the course of regular action. A fault can cause the robot(s) to lose functionality, which in turn may lead to a drop in the overall performance of the system. Hence, it is important for these systems to exhibit some fault-tolerance, or the ability to diagnose and recover from encountered faults.

In the last decade, several researchers have studied fault-tolerance for robotic systems (e.g., [15], [1], [16], [5], [12], [9], [17], [14]). However, still missing from this research are standard metrics for evaluating new and existing multi-robot fault-tolerance methods. In the absence of an accepted metric, it is difficult for a designer to calculate the true measure of a system capability. This is especially true when attempting to

compare two different fault-tolerant strategies, and determining which strategy can achieve the best performance.

One possible way of measuring fault-tolerance is by defining the redundancy in a system, perhaps achieved through interchangeable components that can substitute for each other if one (or more) of the components fail. Most multi-robot applications are distributed in nature, and when robots are homogeneous, they can provide a natural redundancy to each other. However, while redundancy by itself is a useful measure, it is incomplete as an evaluation metric, since a system can also be effectively fault-tolerant through reasoning methods other than redundancy. Thus, it is preferred to have a metric that can measure the effective fault-tolerance as it influences overall system performance in achieving the tasks of the application. Based on this analysis of *effective* fault tolerance, this paper addresses the following problem: *Define a metric for calculating the usefulness of fault-tolerance towards system performance in multi-robot teams.*

The rest of this paper is organized as follows. We first present a brief review of the related work and discuss how the existing methods for evaluation are insufficient for multi-robot systems in Section II. Section III formally defines the above problem and details the derivation of the proposed metrics. In order to evaluate the validity of the metrics, we apply them to a set of experimental results obtained from a physical robot implementation of a sample complex heterogeneous application [10] in Section IV. We discuss the potential scope and significance of the new metrics in Section V and offer concluding remarks and comments on our ongoing work in Section VI.

II. RELATED WORK

The concept of metrics for quantifying performance is not new. In 1994, Cavallaro and Walker [4] recognized the lack of standard metrics and discussed the applicability of protocols based on NASA and military standards. Evans and Messina in [6] analyze the importance of defining universally accepted performance metrics for intelligent systems. The analysis outlines current efforts to develop standardized testing and evaluation strategies and argues the need for industry accepted metrics for inter-comparison of results and to avoid duplication of work. Extending the analysis of Evans and Messina, Pouchard in [22] explores metrics specific to the software agent perspective. Both sets of authors extend a

challenge to the research community to actively work towards the process of developing standard metrics.

Traditional engineering methods that address fault tolerance predominantly deal with reliability analysis of systems and components. *Reliability* is defined as the probability with which a system will perform its specified function/task without failure under stated environmental conditions over a required lifetime. Based on this concept, Carlson and Murphy extensively analyze failure data for mobile robots in [3]. Using MTBF (Mean Time Between Failures) as a representation for average time to the next failure, *reliability* for mobile robots is calculated. The MTBF metric is defined as:

$$MTBF = \frac{\text{No. of hours robot is in use}}{\text{No. of failures encountered}} \quad (1)$$

Other metrics used for evaluation include MTTR (Mean Time Taken to Repair) and *Availability*, which measures the impact of failure on an application or project. These metrics are defined as:

$$MTTR = \frac{\text{No. of hours spent repairing}}{\text{No. of repairs}} \quad (2)$$

$$Availability = \frac{MTBF}{MTTR + MTBF} \cdot 100\% \quad (3)$$

The resulting study illustrates that the reliability among mobile robots is low, with failures occurring at regular time intervals, mainly due to the operating platform. This study is very helpful in providing a detailed analysis of the component failure rate in mobile robots, and in highlighting the complexity of the operating environment as a significant determining factor for failures. However, it does not capture other types of fault tolerance that may be present in a system. It is also difficult to compare the merits of differing robot team control architectures purely using the standard manufacturing metrics of MTBF and MTTR.

In our work on metrics, we want to capture the notion of reasoning and intelligence as it affects the fault tolerance of a system. As our earlier work shows [21], [20], ultimately, multi-robot systems should be able to intelligently handle failures, and thus improve over time. Hence, it is important for any performance metric for a multi-robot system to measure the extent of intelligence exhibited by the system. Recently, there has been a renewed interest in exploring the problem of metrics for intelligent systems. Lee et al. [13], propose an engineering based approach for measuring system intelligence. In this method, learning is used to theoretically measure system intelligence through a formal analysis of system software architecture and hardware configurations. Other related works include Yavnai’s [23] approach for measuring *autonomy* for intelligent systems and Finkelstein’s Analytical Hierarchy Process (AHP [7]) for measuring system intelligence.

Unfortunately, existing work does not apply or extend these measures to help evaluate system fault-tolerance. In fact, relatively little research has addressed the issue of metrics specific to the field of fault-tolerance in multi-robot teams. Most existing architectures are evaluated purely based on task-specific

or architecture-specific quantities [19]. The consequences of such an evaluation are that the general characteristics of fault-tolerance, robustness, and so forth, are not explicitly identified, and instead are hidden in the application-specific measures.

The most promising work related to our objectives is the work of Hamilton, et al. [8]. Their approach outlines a metric for calculating “effective” fault-tolerance for single robot manipulators by combining the observed fault-tolerance with a performance/cost rating. The measure has two distinct terms: the first is based on a fault-tolerance rating and the second term is derived from a performance/cost value, as follows:

$$eff = k_1(f)^2 + k_2(p)^2 \quad (4)$$

where *eff* is the calculated measure, *f* is the fault-tolerance rating, *p* is the performance/cost rating, and *k*₁ and *k*₂ are normalizing constants. Here, fault-tolerance is calculated as $f = m/n$, where *m* is number of tolerable subsystem failures and *n* is number of available subsystems. The performance/cost rating is given by $p = (S + R + C)/3$, where *S* is performance speed, *R* is recovery time rating, and *C* is the cost measure. The authors evaluated their metrics on a number of multiprocessor control architectures.

This proposed metric has a few shortcomings that restrict its applicability for the multi-robot domain. First, the system calculates the effect of robustness purely based on redundancy, and thus does not capture the use of intelligence or reasoning to compensate for failure. Our prior work on developing and evaluating fault-diagnostic architectures for multi-robot systems [20], [21] identifies online learning from faults as an integral component of successful fault-tolerant systems. Hence, it is imperative for a evaluation strategy to quantify learning as part of the fault-tolerance measure. Also, as mentioned in the previous section, most multi-robot systems are task-centric rather than robot-centric. Hence, it is easier to evaluate the system if the metrics focus on task performance. In this paper, we attempt to extend the concept of “effective” evaluation of fault-tolerance to multi-robot systems. The newly proposed metrics are task-centric and include measures to identify system intelligence or learning. We introduce our measures in the next section.

III. PROBLEM DEFINITION

Based on our earlier work on developing turn-key solutions¹ for fault-diagnosis [20], we evaluate system performance based on the following terms:

- 1) **Efficiency** — ability of the system to best utilize the available resources,
- 2) **Robustness** to noise — ability of the system to identify and recover from faults, and
- 3) **Learning** — ability to adapt to the changes in the environment by autonomously extracting and integrating

¹A *turn-key* solution, as defined by Carlson and Murphy [2], is one that can be implemented on different applications without the need for significant modifications.

useful system information during the course of task execution.

We now formally define the problem as follows. Given:

- An autonomous robot team $R = \{R_1, R_2, R_3, \dots, R_n\}$.
- A pre-defined set of tasks to be executed by the robot team $T = \{T_1, T_2, T_3, \dots, T_m\}$, where each task T_j is executed by a separate robot R_i .

We assume:

- The task assignment is pre-defined by means of a set of pairings $\langle R_i, T_j \rangle$. An individual task T_j is executed by the specific robot R_i .
- Faults can occur naturally during task execution or can be artificially introduced into the system.
- Faults are broadly categorized into three (3) types: *known*, which are faults the designer can anticipate based on experience, application type and operating environment; *unknown*, which are faults not anticipated by the designer, but which can be diagnosed by the system based on experience and sparse information available during execution; and *undiagnosable*, which are faults that cannot be classified autonomously and need human intervention. In addition to diagnosis, the system can autonomously recover from *known* and *unknown* faults, whereas human assistance is required for it to recover from *undiagnosable* faults. The number of faults in each category are represented as f_{known}^i , $f_{unknown}^i$, and $f_{undiagnosable}^i$.
- The robots have three (3) functionally significant operating states: *Normal* state, in which a robot focuses all its system resources and operating time towards completing the assigned task; *Fault* state, in which a robot spends all available time and resources in attempting to identify the source of the encountered fault; and *Recovery* state, in which a robot spends its resources and operating time in executing the recovery action for the diagnosed fault.
- Once assigned to a robot, a task can have two possible outcomes: task success or task failure. Task success is defined as the ability of the robot to successfully complete its assigned task. A task failure is defined as the inability of the robot to complete its assigned task in the presence of faults.
- If a robot (R_j) fails to complete a task (T_j), then based on the system design, the system can either assign task T_j to a different robot R_i , re-assign T_j to the task queue of robot R_j , or remove task T_j from the system task list.
- Every task-assignment, $\langle R_i, T_j \rangle$, is considered a *task attempt* and is evaluated separately towards overall system performance.
- Based on the importance of the task the designer builds a task-utility table, such as that shown in Table I, in which the summation of the terms ($\sum u$ and $\sum c$) are normalized between ranges of $[0, 1]$.

A. Measuring System Performance

In developing our metric, we first define the total number of faults for the i^{th} attempt of task T_j as the summation of all

encountered faults during the course of task execution. That is, $F_j^i = f_{known_j}^i + f_{unknown_j}^i + f_{undiagnosable_j}^i$.

Successful completion of task T_j is measured by means of a success metric, A_j . An award is associated with every successfully completed task, given by the utility component u_j .

$$A_j = u_j \quad (5)$$

Then, the system level measure of success (A) is calculated as:

$$A = \sum_{j:T_j \in X} u_j \quad (6)$$

where $X = \{T_j \mid \text{Task } T_j \in T \text{ was successfully completed}\}$. That is, the system level measure of success is the sum of the utilities of the tasks that were successfully completed.

Similarly, we associate a task failure metric, B_j^i , for each unsuccessful attempt of task T_j by a robot. The punishment associated with a failed task attempt is given by the cost component for task failure, c_j . On the other hand, as the performance is closely tied with the robot's ability to recover from faults, every failed task has a robustness component associated with it. The effect of the task failure metric towards performance is discounted by the extent of the robustness in the task, i.e., the higher the robustness, the lower the value of the task failure. We define ρ_j^i as the measure of robustness for the i^{th} attempt of task T_j and is given by

$$\rho_j^i = \frac{f_{known_j}^i + f_{unknown_j}^i}{F_j^i} \quad (7)$$

That is, ρ_j^i gives the fraction of the faults that the system could successfully recover from.

Based on equation 7, the task failure metric for the i^{th} attempt of task T_j is:

$$B_j^i = c_j * (1 - \rho_j^i) \quad (8)$$

Grouping all failed attempts of a task T_j , we get the combined task failure metric (B_j) for a task T_j as:

$$B_j = (c_j * q_j) * \sum_{i=1}^{q_j} (1 - \rho_j^i) \quad (9)$$

where q_j is total number of failed attempts of task T_j . The upper bound of q is application specific and needs to be determined by the designer before implementation.

TABLE I
UTILITY-COST TABLE FOR SYSTEM TASKS

Task	Utility	Cost for task failure
T_1	u_1	c_1
T_2	u_2	c_2
...
T_m	u_m	c_m

Simplifying,

$$B_j = (c_j * q_j) * (q_j - \sum_{i=1}^{q_j} \rho_j^i) \quad (10)$$

Extending equation 10 across all task failures, gives:

$$B = \sum_{j:T_j \in Y} (c_j * q_j) * (q_j - \sum_{i=1}^{q_j} \rho_j^i) \quad (11)$$

where $Y = \{T_j \mid \text{Task } T_j \in T \text{ failed}\}$

Finally, the measure of performance can be obtained by subtracting the cost associated with a task failure from the utility for successful task completion, i.e.,

$$P = A - B \quad (12)$$

Substituting for A and B from equations 6 and 11 respectively, we obtain our desired effective performance metric:

$$P = \sum_{j:T_j \in X} u_j - \sum_{j:T_j \in Y} (c_j * q_j) * (q_j - \sum_{i=1}^{q_j} \rho_j^i) \quad (13)$$

P provides the designer with a measure of the system's effective performance. The measure results in P values in the range $(-\infty, 1]$. A value of 1 indicates an optimal system performance whereas, P approaching $-\infty$ indicates a total system failure. However, P by itself does not provide the all the information necessary for validation. Hence, we need to identify additional metrics that help give a complete picture of the system.

B. Measuring Fault-tolerance

In addition to outlining a measure for performance, we are interested in identifying the fault-tolerance in the system. Based on Murphy and Carlson's observation from the previous section, we measure the system fault-tolerance in terms of robustness, efficiency and learning. These components provide a good metric for identifying the extent and usefulness of fault-tolerance towards improving overall system performance.

Combining individual task robustness measures, system robustness can be represented as:

$$\rho_s = \sum_{j:T_j \in Y} \sum_{i=1}^{q_j} \rho_j^i \quad (14)$$

A high value of ρ_s (an average system exhibits a ρ_s value close to 1.5) indicates a highly robust system and a ρ_s value of 0 indicates a system with no robustness to faults.

In order to define the system efficiency metric (ϵ), we need to measure the total time (t_j) spent by a robot on a successfully completed task, T_j . This is given by the summation of time spent in Normal (t_{Normal}), Fault (t_{Fault}) and Recovery ($t_{Recovery}$) states for that attempt, i.e.,

$$t_j = t_{Normal_j} + t_{Fault_j} + t_{Recovery_j} \quad (15)$$

Then, we can define ϵ as:

$$\epsilon = \sum_{j:T_j \in X} \frac{t_{Normal_j}}{t_j} \quad (16)$$

Similar to the robustness measure, a more efficient system has a higher value of ϵ and an inefficient system has ϵ near 0. The influence of learning towards system performance can be measured as an empirical quantity. Comparing system performances with and without learning gives us a good estimate of the learning in the system.

$$\delta = P - P' \quad (17)$$

where P is a system with learning and P' is a system with no learning.

Finally, based on the above definitions for robustness, efficiency and learning, we can represent system level effective fault-tolerance as an unordered triple given by (ρ, ϵ, δ) .

IV. EVALUATION OF METRICS

To give a better understanding of the range of values for the metrics, we apply them to the following simple example scenarios.

A. Scenario 1

Consider a sample multi-robot application comprised of 10 individual tasks to be completed by a team of 10 functionally similar robots. We make the assumptions that the robots encounter one failure per task and the task/utility weights are evenly distributed. Then we define these measures as follows:

$$\forall i. u_i = c_i \quad (18)$$

$$\sum_{i=1}^{10} u_i = \sum_{i=1}^{10} c_i = 1 \quad (19)$$

The time spent by the robot in Normal operation mode is assumed to be t secs. Also, as it takes a very small fraction of time to diagnose task failure from the time a fault is discovered, we assume this time to be negligible and ignore it.

TABLE II
EVALUATION TABLE FOR SCENARIO 1

System	Case	P	ρ	ϵ	δ
S_1	Best case	1	0	10	0
	Average case	0.5	0	5	0
	Worst case	-1	0	0	0
S_2	Best case	1	0	10	0
	Average case	0.4	0	7.5	0
	Worst case	-0.4	0	5	0

To best illustrate the variations in the values, we choose three specific cases to evaluate, namely:

- 1) Best-case, where the system encounters no failures,

- 2) Average-case, where the system encounters at least one failure in half the number of executed tasks, and
- 3) Worst-case, where there is at least one failure in all cases.

Table II illustrates the values obtained for two different hypothetical architectural implementations – one with no built-in fault-tolerance (S_1) and another with some redundancy-based fault-tolerance (S_2). For this scenario, the task/utility weights are evenly distributed, i.e., $u_1 = c_1, u_2 = c_2, \dots$. When a fault is encountered during task execution in the first architecture, robot(s) do not have the capability to recover and report a failed task. In the case of the second architecture, if and when a failure occurs, the task is assumed to have failed and is reassigned to another team member for execution. The task reassignment continues until all robots in the team have had an opportunity to complete the task. We further make the assumption that on average it takes $\frac{n}{2}$ attempts to successfully recover from an encountered fault. Finally, we assume there is 50% probability of the system successfully recovering from an encountered error.

Looking at the values for system performance in Table II we can infer that, for the average-case, the architecture with zero fault-tolerance (S_1) performs better than the architecture with some fault-tolerance (S_2). For this specific application, as the cost/utility values are equivalent, the inability of the system to handle failures does not significantly affect system performance (P) for the average-case. In fact, the time and resources spent in fault-diagnosis and recovery by S_2 adversely affects its performance. However, with increasing number of faults the ability of the system to handle failures becomes important. This is indicated by the worst-case performance of the two architectures. Table II shows S_2 edging S_1 as the number of failures in the system increases. On the other hand, a system with a higher task completion rate will have a higher value for efficiency (ϵ), as reflected in Table II. Finally, a δ value of zero highlights the fact that neither system exhibits any kind of learning.

B. Scenario 2

Consider an alternate multi-robot application comprised of 10 individual tasks to be completed by a team of 10 functionally similar robots. Similar to the above scenario, we make the assumptions that the robots encounter one failure per task. In contrast to Scenario 1, the task/utility weights are not evenly distributed with a higher utility associated for task success than the cost for a task-failure, as given by:

$$\forall i, j. u_i = u_j \quad (20)$$

$$\forall i, j. c_i = c_j \quad (21)$$

$$\sum_{i=1}^{10} u_i > \sum_{i=1}^{10} c_i \quad (22)$$

$$\sum_{i=1}^{10} u_i = 1 \quad (23)$$

$$\sum_{i=1}^{10} c_i = 0.5 \quad (24)$$

The time spent by the robot in Normal operation mode is assumed to be t secs. Also, as it takes a very small fraction of time to diagnose task failure from the time a fault is discovered, we assume this time to be negligible and ignore it.

To maintain consistency, we choose the same three specific cases to evaluate that we discussed above, namely:

- 1) Best-case, where the system encounters no failures,
- 2) Average-case, where the system encounters at least one failure in half the number of executed tasks, and
- 3) Worst-case, where there is at least one failure in all cases.

TABLE III
EVALUATION TABLE FOR SCENARIO 2

System	Case	P	ρ	ϵ	δ
S_1	Best case	1	0	10	0
	Average case	0.25	0	5	0
	Worst case	-0.5	0	0	0
S_2	Best case	1	0	10	0
	Average case	0.575	0	7.5	0
	Worst case	0.15	0	5	0

The difference in the performance of the two systems S_1 and S_2 is highlighted Table III. Unlike in the previous scenario, system S_2 has a consistently higher performance rating than system S_1 . As more emphasis is placed on task success than on system failure (i.e., higher utility value), the ability to recover from failures and to complete the assigned task directly impacts system performance (P). Similar to the previous scenario, system S_2 has higher efficiency (ρ) values than system S_1 . Hence, comparing the values of P , ρ , and δ for the two systems, we can say S_2 is a more suitable architecture for the application.

C. Scenario 3

Finally, we apply the metrics to the experimental results obtained for the physical robot implementation of a complex heterogeneous application [10]. This test application is a large-scale locate-and-protect mission involving a large team of physical heterogeneous robots. The robot team has a very strict set of goals/tasks: to autonomously explore and map a single story in a large indoor environment, detect a valued object, deploy a sensor network and use the network to track intruders within the building.

The composition of the team shown in Figure 1 consisted of three classes of robots: Four (4) mapping robots equipped

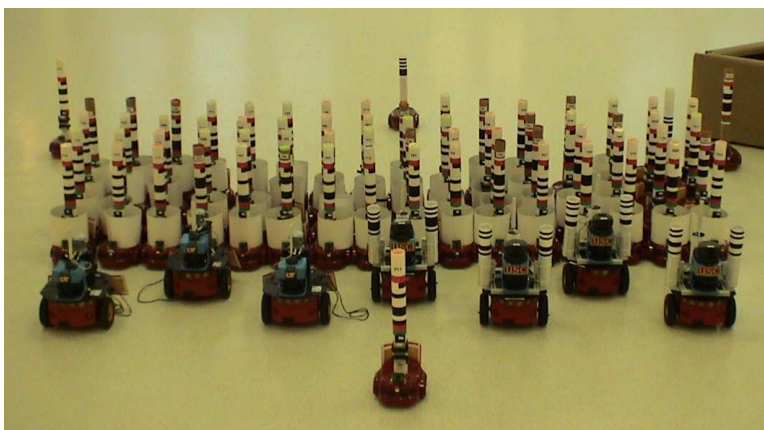


Fig. 1. The heterogeneous robot team — mapper, helper and simple robots.

with scanning laser range-finders and a unique fiducial; three (3) helper robots equipped with scanning laser range-finders and cameras; and a large number (approximately 70) of sensor-limited robots equipped with a microphone and a crude camera. All of the robots had 802.11 WiFi, and a modified ad-hoc routing package (AODV) was used to ensure network connectivity.

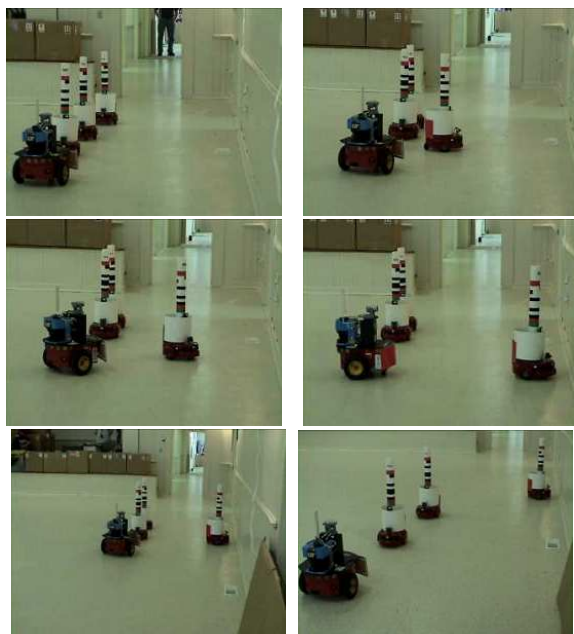


Fig. 2. Deployment of a sensor robot using assistive navigation: the lead robot first guides and then directs the sensor robot into position (read left to right, top to bottom).

In order to perform the task of locate-and-protect, the sensor-limited mobile robots had to be moved into deployment positions that were optimal for serving as a sensor network. Because these sensor-limited robots could not navigate safely on their own, complex heterogeneous teaming behaviors were used that allowed the small group of helper robots to deploy the sensor-limited robots (typically, 1–4 of these simple robots

at a time) to their planned destinations using a combination of robot chaining and vision-based marker detection for autonomous tele-operation [21]. Table IV shows the relation between the individual modules and the defined set of tasks. Figure 2 shows these robots performing one such deployment task. The scenario involves a complex combination of cooperative and single-robot behaviors, including laser-based localization, path planning, obstacle avoidance, vision-based autonomous tele-operation, simple vision-based following, and wireless *ad hoc* mobile communication in a heavily cluttered environment, leading to a wide variety of faults.

To handle the encountered faults, a fault diagnostic system based on causal model methodology was implemented (see [20] for more details). The experiments consisted of repeated deployments of 1, 2, or 3 simple robots per team. Over the course of the experiment, various failures were encountered, some of which were expected and some that were totally unexpected. If a deployment failed on one experiment, the consequences of that failure were not corrected, except on rare occasions. Thus, the data collected incorporates propagation of error from one experiment to the next. In these experiments, a

TABLE IV
TASK MODULE RELATIONSHIP TABLE FOR SCENARIO 3

Task	Modules
Go_to_goal Task	Localization, Path_planning, Navigation
Deployment Task	Marker Detection, Communication
Recharging Task	Localization, Path_planning, Navigation, Marker Detection, Communication
Follow_the_leader Task	Marker Detection
Return_home Task	Localization, Path_planning, Navigation

TABLE V
OVERALL SYSTEM SUCCESS RATE, AVERAGED OVER 45 TRIALS

Module	Subsystem Success Rate	Experimental Success Rate
Localization	0.83	
Path Planning	0.99 (est.)	
Navigation	0.95 (est.)	
Follow_the_leader	0.78 (est.)	
Marker Detection	0.98	
Communication	0.91	
Complete System	0.54 (est.)	0.67 (2-robot depl.) 0.48 (1-robot depl.) 0.59 (combined over all trials)
Helper Robot returning home		0.91 (over all trials)

total of 61 simple robot deployments were attempted. The experimental data showed an overall deployment success rate of 60% - 90%, depending upon the environmental characteristics. In other words, for each attempt at deploying a simple robot, 60% - 90% of those robots successfully reached their planned deployment position. Table V depicts the probability of success of each individual module in this implementation and the overall system probability, based upon the experimental results. The probability values are used to calculate individual and collective task robustness.

TABLE VI
UTILITY-COST TABLE FOR SCENARIO 3

Task	Utility	Cost for task failure
Go_to_goal Task	0.15	0.08
Deployment Task	0.15	0.08
Recharging Task	0.15	0.08
Follow_the_leader Task	0.15	0.08
Return_home Task	0.3	0.18

TABLE VII
EVALUATION TABLE FOR SCENARIO 3

System	P	ρ	δ
SDR	-5.4283	3.976	0

In order to better understand the quality of performance of the described system, we apply our metric on the obtained results. During the evaluation process certain constraints had to be accounted for, most important of which was incorporating the disparity in the task/utility value associated with helper and sensor-limited robots. This is shown in Table VI.

For these experiments, ρ is a measure based on the total probability of task success. Task success is given by the product of the success probabilities of individual sub-modules

for the specified task. Also, as the faults propagate from one run to the next, the entire set of trials (61) is considered as a single continuous experiment. From the collected experimental values in Tables VI and V, we get

$$A = ((.83 * .99 * .95) * .15) + ((.98 * .91) * .15) + \dots$$

$$B = ((.08 * 61) * (1 - (.83 * .99 * .95))) + ((.08 * 61) * (1 - (.98 * .91))) + \dots$$

Table VII shows the evaluated values for system performance and fault-tolerance. The system displays a high amount of robustness. However, the performance metric indicates a negative value, which shows that for the concerned application the implemented fault-tolerance does not optimize system performance. An alternate technique could potentially be used to further improve performance. The Table also indicates a total lack of any learning in the fault-tolerance design², which is consistent with the analysis that was performed separately [20]. On the other hand, the lack of learning does not indicate a failure of the system to learn, instead it merely highlights that the system was not designed to be a learning system and hence its failure to adapt to unexpected changes during the course of task-execution. Instead, an adaptive method is needed to enable the robot team to use its experience to update and extend its causal model to enable the team, over time, to better recover from faults when they occur.

In our other ongoing work [20], we are developing an extended causal model methodology, called LeaF, that enables the system to learn from experience and adapt its model, thereby improving its ability to diagnose and recover from unexpected faults. The unique aspect of the proposed architecture is its ability to extract useful information from previously encountered faults. Specifically, LeaF is designed as a distributed model that uses one or more partial causal models for representing the various faults in the system. The model has an *a priori* base set of assumptions about behaviors and availability of resources. Fault detection is defined as the ability of the agent(s) to recognize when an assumption becomes invalid. Given this invalid assumption, fault diagnosis is defined as the ability of the system to identify the resource behavior or sensor that is responsible for the failure. Prior knowledge about the expected behavior provides a comparison monitor for the subsequent actions of the system. In addition, a modified case-based learning algorithm is used to adapt and categorize a new fault and add it to the causal model for future use. At a higher level, the entire process of fault representation and diagnosis can be viewed as a fully connected graph, with nodes representing faults and edges highlighting the relation between the faults. Ultimately the goal is to build a cross-architecture system capable of learning from its own faults and those of other team members, making it a domain- and application-independent architecture.

²Since the experimental results did not have information regarding the time spent in handling faults, we do not calculate the efficiency metric for this system.

V. DISCUSSION

In the previous section, we have detailed distinct and separate measures for calculating system performance and fault-tolerance. In justification, when measured separately neither one of the two measures provide a complete assessment of the application in use. Using only system performance, we do not get a fair idea regarding the extent of fault-tolerance in the system. On the other hand, fault-tolerance by itself is not a strong enough measure for evaluating systems. However, the two metrics when viewed in context with each other helps the designer compare and contrast performances of different architectures in order to select the most appropriate one for the application in question. The ability to compare systems can help identify potential shortcomings, leading to the development of more refined and effective solutions. This also reduces the amount of time and resources spent in duplicating existing work.

VI. CONCLUSIONS AND FUTURE WORK

As new techniques in fault-tolerance are being explored [20], existing methods do not provide a complete measure of system performance for multi-robot teams. Hence, there is a need for a more generic evaluation method for multi-robot systems. In this paper, we present an evaluation metric to measure the extent of fault-tolerance towards system improvement over a period of time. Furthermore, we evaluated a large-scale multi-robot application based on the defined metrics. Specifically, the research provides a qualitative measure for identifying system fault-tolerance in terms of efficiency, robustness and the extent of learning. To the best of our knowledge, this is the first metric that attempts to evaluate the quality of learning towards understanding system level fault-tolerance.

As part of our ongoing research, we plan to apply the metrics to our newly proposed fault-tolerance architecture, LeaF (Learning-based Fault diagnosis), and compare the results with those of other existing architectures, such as CMM [11], SFX-EH [18], and so forth. The observations will help us further evaluate refine our approach.

ACKNOWLEDGMENTS

Parts of this research were sponsored by DARPA/IPTO's Software for Distributed Robotics program, Science Applications International Corporation, and the University of Tennessee's Center for Information Technology Research. This paper does not reflect the position or policy of the U.S. Government and no official endorsement should be inferred.

REFERENCES

- [1] J. Bongard and H. Lipson. Automated damage diagnosis and recovery for remote robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3545–3550, 2004.
- [2] J. Carlson and R. R. Murphy. Reliability analysis of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [3] J. Carlson and R. R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437, June 2005.
- [4] J. Cavallaro and I. Walker. A survey of NASA and military standards on fault tolerance and reliability applied to robotics. In *Proceedings of AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space*, pages 282–286, March 1994.
- [5] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robust positioning with a multi-agent robotic system. In *Proceedings of IJCAI-93 Workshop on Dynamically Interacting Robots*, pages 118–123, 1993.
- [6] J. Evans and E. Messina. Performance metrics for intelligent systems. In *Performance Metrics for Intelligent Systems (PerMIS) Proceedings*, volume Part II, August 2000.
- [7] R. Finkelstein. A method for evaluating IQ of intelligent systems. In *Performance Metrics for Intelligent Systems (PerMIS) Proceedings*, volume Part II, August 2000.
- [8] D. Hamilton, I. Walker, and J. Bennett. Fault tolerance versus performance metrics for robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3073–3080, 1996.
- [9] B. Horling, V. Lesser, R. Vincent, A. Bazzan, and P. Xuan. Diagnosis as an integral part of multi-agent adaptability. In *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 211–219, 2000.
- [10] A. Howard, L. E. Parker, and G. S. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment, and detection. *International Journal of Robotics Research*, 2006.
- [11] E. Hudlická and V. R. Lesser. Modeling and diagnosing problem-solving system behavior. *IEEE Transactions on Systems, Man, and Cybernetics*, 17:407–419, 1987.
- [12] G. A. Kaminka and M. Tambe. What is wrong with us? Improving robustness through social diagnosis. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pages 97–104, 1998.
- [13] S. Lee, W. Bang, and Z. Bien. Measure of system intelligence: An engineering perspective. In *Performance Metrics for Intelligent Systems (PerMIS) Proceedings*, volume Part II, August 2000.
- [14] M. Long, R. R. Murphy, and L. E. Parker. Distributed multi-agent diagnosis and recovery from sensor failures. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2506–2513, 2003.
- [15] S. Mahdavi and P. Bentley. An evolutionary approach to damage recovery of robot motion with muscles. In *European Conference on Artificial Life (ECAL)*, pages 248–255, 2003.
- [16] M. J. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Animals to Animats 2: Proceedings of the second international conference on simulation of adaptive behaviour*, pages 432–441. MIT Press, 1993.
- [17] R. Murphy and D. Hershberger. Classifying and recovering from sensing failures in autonomous mobile robots. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI)*, volume 2, pages 922–929, 1996.
- [18] R. R. Murphy and D. Hershberger. Handling sensing failures in autonomous mobile robots. *The International Journal of Robotics Research*, 18:382–400, 1999.
- [19] L. E. Parker. Evaluating success in autonomous multi-robot teams: Experiences from ALLIANCE architecture implementations. *Journal of Theoretical and Experimental Artificial Intelligence*, 13:95–98, 2001.
- [20] L. E. Parker and B. Kannan. Adaptive causal models for fault diagnosis and recovery in multi-robot teams. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006, to appear.
- [21] L. E. Parker, B. Kannan, F. Tang, and M. Bailey. Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 1016–1022, 2004.
- [22] L. Pouchard. Metrics for intelligence: a perspective from software agents. In *Performance Metrics for Intelligent Systems (PerMIS) Proceedings*, volume Part II, August 2000.
- [23] A. Yavnai. Metrics for system autonomy. Part I: Metrics definition. In *Performance Metrics for Intelligent Systems (PerMIS) Proceedings*, volume Part II, August 2000.

Image Classification and Retrieval Using Elastic Shape Metrics

Shantanu H. Joshi
Department of Electrical Engineering
Florida State University
Tallahassee, FL-32310
joshi@eng.fsu.edu

Anuj Srivastava
Department of Statistics
Florida State University
Tallahassee, FL-32306
anuj@stat.fsu.edu

Abstract—This paper presents a shape-based approach for automatic classification and retrieval of imaged objects. The shape-distance used in clustering is an intrinsic elastic metric on a nonlinear, infinite-dimensional shape space, obtained using geodesic lengths defined on the manifold. This analysis is landmark free, does not require embedding shapes in \mathbb{R}^2 , and uses ODEs for flows (as opposed to PDEs). Clustering is performed in a hierarchical fashion. At any level of the hierarchy, clusters are generated using a minimum dispersion criterion and a MCMC-type search algorithm is employed to ensure near-optimal configurations. The Hierarchical clustering potentially forms an efficient ($\mathcal{O}(\log(n))$ searches) tool for retrieval from large shape databases. Examples are presented for demonstrating these tools using shapes from the ETH-80 shape database.

Keywords: *shape clustering, shape classification, image retrieval*

I. INTRODUCTION

Unsupervised learning of visual object features is an important task in machine vision applications such as medical imaging, automatic surveillance, biometrics, and military target recognition. The imaged objects can be characterized in many ways: according to their colors, textures, shapes, movements, and locations. Of late, shape has been used as an important discriminant for identification and recognition of objects from images. Indeed, it is a desirable goal for an intelligent system to have automated tools for classifying and clustering objects according to the shapes of their boundaries.

A. Past Shape-based Image Retrieval

In general, there have been numerous approaches for including shapes in conjunction with color, intensity and textures for image indexing and retrieval. Many techniques, including Fourier descriptors [18], [17], Wavelet descriptors [20], chain codes, polygonal approximations [19], and moment descriptors [21] have been proposed and used in

various applications. Cortelazzo et al. [16] use chain codes for trademark image shape descriptions and string matching techniques. Jain and Vailaya [12] propose a representation scheme based on histograms of edge directions of shapes. A different approach by Mokhtarian et al. [14] uses curvature scale space methods for robust image retrieval from the Surrey fish dataset [13]. A majority of these methods have focused on the limited goal of fast shape matching and retrieval from large databases. Simple metrics using either Fourier or moment descriptors, or scale-space shape representations, may prove sufficient for retrieving shapes from a database. However they lack the tools and the framework for more advanced analysis, especially if one requires building probability models using the retrieved results.

B. Past Shape Analysis Methods

To address the above difficulties, and seek a a full statistical framework, Klassen, Srivastava et al. [2] adopt a geometric approach to parameterize curves by their arc lengths, and use their angle functions to represent and analyze shapes. Using the representations and metrics described in [2], Srivastava et al. [5] describe techniques for clustering, learning, and testing of planar shapes. One major limitation of this approach is that all curves are parameterized by arc length, and the resulting transformations from one shape into another are restricted to *bending only*. Local stretching or shrinking of shapes is not allowed. Mio and Srivastava [3] resolve this issue by introducing a representation that allows both bending and stretching of curves to compare and match shapes. It has been demonstrated in [3], that geodesics resulting from this approach seem more natural as interesting features, such as corners, are better preserved, thus leading to improved metrics in the shape space. We use the approach presented in [3] to represent and analyze shapes of closed curves. The basic idea is to represent



Fig. 1. Example of a geodesic between a pair of shapes.

these curves as parameterized functions, not necessarily arc-length, with appropriate constraints, and define a non-linear manifold \mathcal{C} of closed curves. To remove similarity transformations, one forms a quotient space $\mathcal{S} = \mathcal{C}/S$, where S is the space of similarity transformations. Shapes of closed curves are analyzed as elements of \mathcal{S} . The following section describes the shape representation scheme and briefly explains the construction of geodesics between any two given shapes on \mathcal{S} .

C. Elastic Shape Representation Scheme

Let β be a parameterized curve of interest, of length l , and $\alpha = 2\pi\beta/l$ be its re-scaled version. We will assume $\alpha: [0, 2\pi] \rightarrow \mathbb{R}^2$ to be a smooth, non-singular, orientation-preserving, parametric curve in the sense that $\dot{\alpha}(s) \neq 0$, $\forall s \in [0, 2\pi]$. Define the velocity vector of the curve as $\dot{\alpha}(s) = e^{\phi(s)} e^{j\theta(s)}$, where $\phi: [0, 2\pi] \rightarrow \mathbb{R}$ and $\theta: [0, 2\pi] \rightarrow \mathbb{R}$ are smooth, and $j = \sqrt{-1}$. The function ϕ is the *speed* of α and measures the rate of stretching and compression, whereas θ is the angle made by $\dot{\alpha}(s)$ with the X -axis and denotes bending. We will represent α via the pair $\nu \equiv (\phi, \theta)$, and denote by \mathcal{H} the collection of all such pairs. In order to make the shape representation invariant to rigid motions and uniform scalings, we restrict shape representatives to pairs (ϕ, θ) satisfying the conditions;

$$\mathcal{C} = \left\{ (\phi, \theta) : \int_0^{2\pi} e^{\phi(t)} dt = 2\pi, \frac{1}{2\pi} \int_0^{2\pi} \theta(t) e^{\phi(t)} dt = \pi, \int_0^{2\pi} e^{\phi(t)} e^{j\theta(t)} dt = 0 \right\} \subset \mathcal{H},$$

where \mathcal{C} is called the *pre-shape space* of planar elastic strings.

Remark: Note that the pair (ϕ, θ) represents the shape of β , and thus ignores its placement, orientation, and scale. Shape deformations are studied using geodesics in the shape space \mathcal{S} connecting them. Given two shapes ν_1 and ν_2 , computing a geodesic involves finding a tangent direction $g \equiv (h, f)$, such that the exponential map [1], $\mathbf{exp}_{\nu_1}(g) = \nu_2$. This is also represented by the geodesic flow $\Psi_1(\nu_1, g) = \nu_2$. Figure 1 shows such a geodesic between two shapes. Shape geodesics are computed under the following Riemannian metric [3]: Given $(\phi, \theta) \in \mathcal{C}$, let h_i and f_i , $i = 1, 2$ be

tangent to \mathcal{C} at (ϕ, θ) . For $a, b > 0$, define

$$\begin{aligned} \langle (h_1, f_1), (h_2, f_2) \rangle_{(\phi, \theta)} &= a \int_0^{2\pi} h_1(s) h_2(s) e^{\phi(s)} ds + \\ &+ b \int_0^{2\pi} f_1(s) f_2(s) e^{\phi(s)} ds. \end{aligned} \quad (1)$$

The parameters a and b control the *tension* and *rigidity* in the metric. The geodesic distance, (used as the shape metric) is now given by

$$d(\nu_1, \nu_2) \triangleq \|(h, f)\|_{(\phi, \theta)} = \sqrt{\langle (h, f), (h, f) \rangle_{(\phi, \theta)}}$$

The remainder of the paper is organized as follows. Section II outlines a clustering algorithm using the geodesic lengths discussed above. The results and the performance of the clustering algorithm are demonstrated in Section III followed by the conclusion.

II. SHAPE CLUSTERING

Classical clustering algorithms on Euclidean spaces generally fall into two main categories: partitional and hierarchical [8]. Assuming that the desired number k of clusters is known, partitional algorithms typically seek to minimize a cost function Q_k associated with a given partition of the data set into k clusters. Usually, the total variance of a clustering is a widely used cost function. Hierarchical algorithms, in turn, take a bottom-up approach. If the data set contains n points, the clustering process is initialized with n clusters, each consisting of a single point. The clusters are then merged successively according to some criterion until the number of clusters is reduced to k . Commonly used metrics include the distance of the means of the clusters, the minimum distance between elements of clusters, and the average distance between elements of the clusters. In this paper, we choose a value of k beforehand.

A. Minimum-Variance Clustering

Consider the problem of clustering n shapes (in \mathcal{S}) into k clusters. To motivate our algorithm, we begin with a discussion of a classical clustering procedure for points in Euclidean spaces, which uses the minimization of the total variance of clusters as a clustering criterion. More precisely, consider a data set with n points $\{y_1, y_2, \dots, y_n\}$ with each $y_i \in \mathbb{R}^d$. If a collection $C = \{C_i, 1 \leq i \leq k\}$ of subsets of \mathbb{R}^d partitions the data into k clusters, the total variance of C is defined by $Q(C) = \sum_{i=1}^k \sum_{y \in C_i} \|y - \mu_i\|^2$, where μ_i is the mean of data points in C_i . The term $\sum_{y \in C_i} \|y - \mu_i\|^2$ can be interpreted as the total variance of the cluster C_i . The total variance is used instead of the (average) variance to avoid placing a bias on large clusters, but when the data is fairly uniformly scattered, the difference is not

significant and either term can be used. The widely used *k-Means Clustering Algorithm* is based on a similar clustering criterion (see e.g. [8]). The *soft k-Means Algorithm* is a variant that uses ideas of simulated annealing to improve convergence [9], [7]. These ideas can be extended to shape clustering using $d(\nu, \mu_i)^2$ instead of $\|y - \mu_i\|^2$, where $d(\cdot, \cdot)$ is the geodesic length and μ_i is the Karcher mean [5] of a cluster C_i on the shape space.

Clustering algorithms that involve finding means of clusters are only meaningful in metric spaces where means can be defined and computed. However, the calculation of Karcher means of large shape clusters is a computationally demanding operation. Therefore, it is desirable to replace quantities involving the calculation of means by approximations that can be derived directly from distances between the corresponding data points. Hence, we propose a variation that replaces $d(\nu, \mu_i)^2$ with the average distance-square $V_i(\nu)$ from ν to elements of C_i . If n_i is the size of C_i , then $V_i(\nu) = \frac{1}{n_i} \sum_{\nu' \in C_i} d(\nu, \nu')^2$. The cost Q associated with a partition C can be expressed as

$$Q(C) = \sum_{i=1}^k \frac{2}{n_i} \left(\sum_{\nu_a \in C_i} \sum_{b < a, \nu_b \in C_i} d(\nu_a, \nu_b)^2 \right). \quad (2)$$

If the average distance-square within the clusters is used, the scale factor in each term is modified to $\frac{2}{n_i(n_i-1)}$. In either case, we seek configurations that minimize Q , i.e., $C^* = \operatorname{argmin} Q(C)$. In this paper we have used the latter cost function.

B. Clustering Algorithm

We will minimize the clustering cost using a Markov chain Monte Carlo (MCMC) search process on the configuration space. The basic idea is to start with a configuration of k clusters and keep on reducing Q by re-arranging shapes amongst the clusters. The re-arrangement is performed in a stochastic fashion using two kinds of moves. These moves are performed with probability proportional to negative exponential of the Q value of the resulting configuration.

- 1) **Move a shape:** Here we select a shape randomly and re-assign it to another cluster. Let $Q_j^{(i)}$ be the clustering cost when a shape ν_j is re-assigned to the cluster C_i keeping all other clusters fixed. If ν_j is not a singleton, i.e. not the only element in its cluster, then the transfer of ν_j to cluster C_i is performed with the probability:

$$P_M(j, i; T) = \frac{\exp(-Q_j^{(i)}/T)}{\sum_{i=1}^k \exp(-Q_j^{(i)}/T)}, \quad i = 1, 2, \dots, k$$

Here T plays the role of temperature as in simulated annealing. Note that moving ν_j to any other cluster

is disallowed if it is a singleton in order to fix the number of clusters at k .

- 2) **Swap two shapes:** Here we select two shapes from two different clusters and swap them. Let $Q^{(1)}$ and $Q^{(2)}$ be the Q -values of the original configuration (before swapping) and the new configuration (after swapping), respectively. Then, swapping is performed with the probability:

$$P_S(T) = \frac{\exp(-Q^{(2)}/T)}{\sum_{i=1}^2 \exp(-Q^{(i)}/T)}.$$

Additional types of moves can also be used to improve the search over the configuration space although their computational cost becomes a factor too. In view of the computational simplicity of moving a shape and swapping two shapes, we have restricted the algorithm to these two simple moves.

In order to seek global optimization, we have adopted a simulated annealing approach. That is, we start with a high value of T and reduce it slowly as the algorithm search for configurations with smaller dispersions. Additionally, the moves are performed according to a Metropolis-Hastings algorithm (see [6] for reference), i.e. candidates are proposed randomly and accepted according to certain probabilities (P_M and P_S above). Although simulated annealing and the random nature of the search help in getting out of local minima, the convergence to a global minimum is difficult to establish. As described in [6], the output of this algorithm is a Markov chain but is neither homogeneous nor convergent to a stationary chain. If the temperature T is decreased slowly, then the chain is guaranteed to converge to a global minimum. However, it is difficult to make an explicit choice of the required rate of decrease in T and instead we rely on empirical studies to justify this algorithm. It is important to note that once the pairwise distances are computed, they are not computed again in the iterations. Secondly, unlike k -mean clustering mean shapes are not used here. These factors make Algorithm 1 efficient and effective in clustering diverse shapes.

We have applied Algorithm 1 to organize a collection of $n = 3270$ shapes (not shown) from the ETH-80 shape database [10] into 25 clusters. Figure 2 shows a few sample images of common objects, and their shape representations from the ETH-80 dataset. Shown in Figure 3(a) are a few samples from the 25 clusters. The elastic metric used in computing pairwise distances for the clusters shown in Fig. 3 assumes the values of $a = b = 1$ in Eqn. 1.

In each run of Algorithm 1, we keep the configuration with minimum Q value. Figure 3(b) shows an evolution of the search process where the Q values are plotted against the iteration index. Figure 3(c) shows a histogram of the

Algorithm 1: For n shapes and k clusters initialize by randomly distributing n shapes among k clusters. Set a high initial temperature T .

- 1) Compute pairwise geodesic distances between all n shapes. This requires $n(n-1)/2$ geodesic computations.
 - 2) With equal probabilities pick one of two moves:
 - a) **Move a shape:**
 - i) Pick a shape ν_j randomly. If it is not a singleton in its cluster then compute $Q_j^{(i)}$ for all $i = 1, 2, \dots, k$.
 - ii) Compute the probability $P_M(j, i; T)$ for all $i = 1, \dots, k$ and re-assign ν_j to a cluster chosen according to the probability P_M .
 - b) **Swap two shapes:**
 - i) Select two clusters randomly, and select a shape from each of them.
 - ii) Compute the probability $P_S(T)$ and swap the two shapes according to that probability.
 - 3) Update temperature using $T = T/\beta$ and return to Step 2. We have used $\beta = 1.0001$ in our experiments.
-

best Q values obtained in 100 such runs, each starting from a random initial configuration. It must be noted that 80% of these runs result in configurations that are quite close to the optimal. Once pairwise distances are computed, it takes approximately 40 seconds to perform 45,000 steps of Algorithm 1 in the matlab environment. The success of Algorithm 1 in clustering these diverse shapes is visible in these results as similar shapes have been clustered together.

C. Hierarchical Classification

An important goal of this paper is to organize large databases of shapes in a fashion that allows for efficient searches. One way of accomplishing this is by organizing shapes in a tree structure, such that shapes are refined regularly as we move down the tree. In other words, objects are organized (clustered) according to coarser differences (in their shapes) at top levels and finer differences at lower levels. This is accomplished in a bottom up construction as follows: start with all the shapes at the bottom level and cluster them according to Algorithm 1 for a pre-determined k . Then, compute a mean shape for each cluster and at the next level cluster these mean shapes according to Algorithm 1. Applying this idea repeatedly, one obtains a tree organization of shapes in which shapes change from coarse to fine as we move down the tree. Critical to this organization is the notion of the mean of shapes for which we utilize Karcher means.

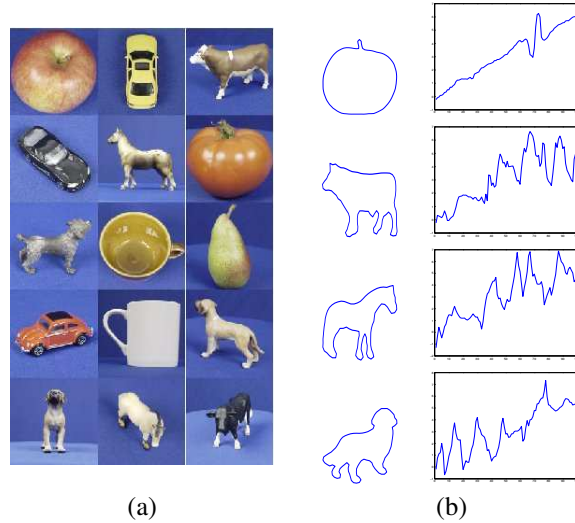


Fig. 2. (a) Examples of images from the ETH-80 dataset. (b) Examples of a few shapes and their angle functions.

We follow the procedure above to generate an example of a tree structure (Fig. 4) obtained for 3270 shapes selected from the ETH-80 database. It is interesting to study the variations in shapes as we follow a path bottom level, these 300 shapes are clustered in $k = 25$ clusters, with the clusters denoted by the indices of their element shapes. Computing the means of each these clusters, we obtain shapes to be clustered at the next level. Repeating the clustering for $k = 8$ clusters we obtain the next level and their mean shapes. In this example, we have chosen to organize shapes in six levels with a single shape at the top. The choice of parameters such as the number of levels, and the number of clusters at each level, depends on the required search speed and performance. It is interesting to study the variations in shapes as we follow a path from top to bottom in this tree. This hierarchical representation of shapes can be effectively used to compare highly dissimilar shapes at a low resolution while allowing similar shapes to be compared at a higher resolution.

III. RETRIEVAL PERFORMANCE AND RESULTS

A logical way to retrieve searches from the hierarchical database is to start at the top, compare the query with the shapes at each level, and proceed down the branch that leads to the best match. At any level of the tree, there is a number, say k , of possible shapes, and our goal is to find the shape that matches the query ν the best. This can be performed using $k - 1$ nearest-neighbor tests leading to the selection of the best hypothesis. In the current implementation, we have assumed a simplification that the covariance matrices for all hypotheses at all levels are identity and only the mean shapes are needed to organize the database. For identity

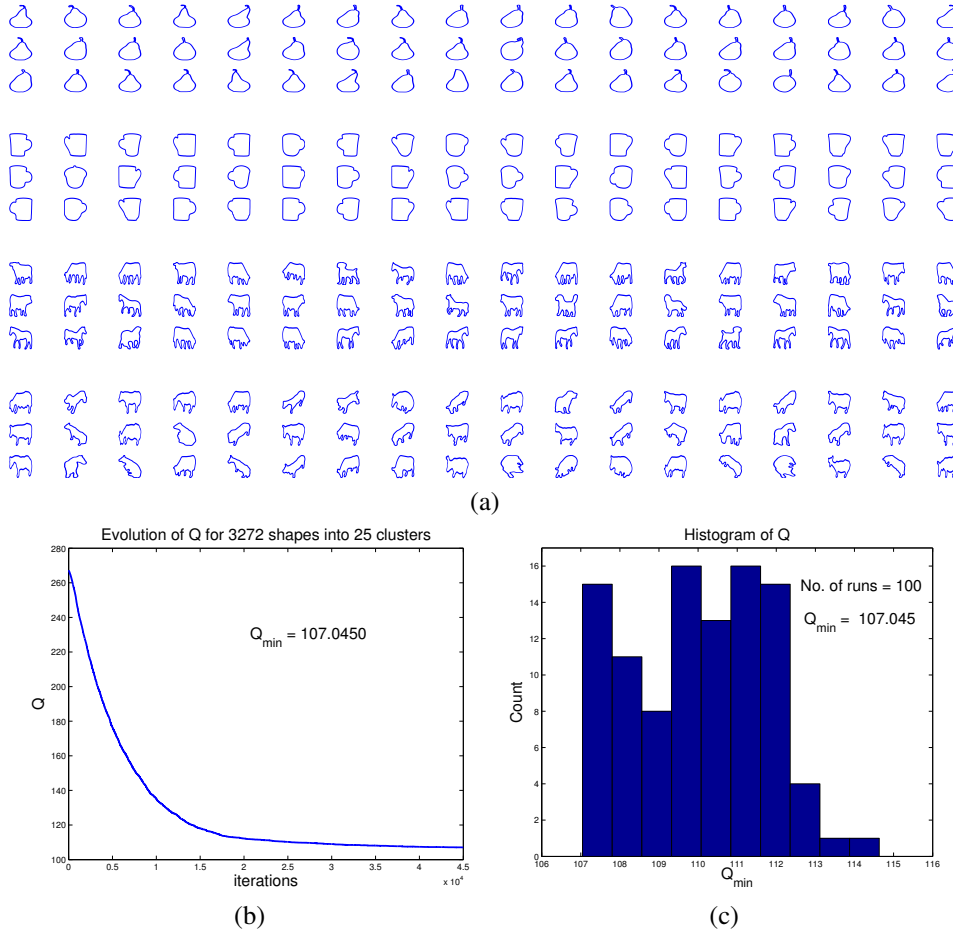


Fig. 3. (a) Examples of shapes from clusters 8,14,16,23 of the ETH-80 database. (b) Sample evolution of Algorithm 1 for the configuration in (a). (c) Histogram of $Q(C^*)$ for 100 runs.

covariances, the task of finding the best match at any level reduces to finding the nearest meanshape at that level. Let μ_i be the given shapes at a level and let x_i be the Fourier vector that encode tangent direction from ν to μ_i . Then, the nearest shape is indexed by $\hat{i} = \operatorname{argmin}_i \|x_i\|$. Proceed down the tree following the nearest shape $\mu_{\hat{i}}$ at each level. This continues until we reach the last level and have found an overall match to the given query. We have implemented this idea using test images from the ETH database. For each test image, we first extract the contour, compute its shape representation as $\nu \in \mathcal{S}$, and follow the tree, shown in Fig. 4, for retrieving similar shapes from the database.

Fig. 5 presents some pictorial examples from this experiment. Shown in the left panels are the original images and in the second left panels their automatically extracted contours. The third column shows five nearest shapes retrieved in response to the query. Finally, the last panel states the time taken for the hierarchical search. In this

experiment, retrieval performance is defined with respect to the original labels, e.g., apple, car, pear, etc. Shown in Fig. 6 are plots of retrieval performances, measured using two different quantities. The first quantity is the precision rate, defined as the ratio of number of relevant shapes retrieved, i.e., shapes from the correct class, to the total number of shapes retrieved. Ideally, this quantity should be one, or quite close to one. The second quantity, called the recall rate, is the ratio of number of relevant shapes retrieved to the total number of shapes in that class in the database. Fig. 6(a) shows average variation of precision rate plotted against the number of shapes retrieved, for four different classes –apple, car, pear, and tomato. As these curves indicate, the retrieval performance of apple falls quickly while that for the other classes remains high. The reason for a low-retrieval performance of apple shapes is their close resemblance in shape to tomatoes. Fig. 6(b) shows plots of recall rate plotted against the number of

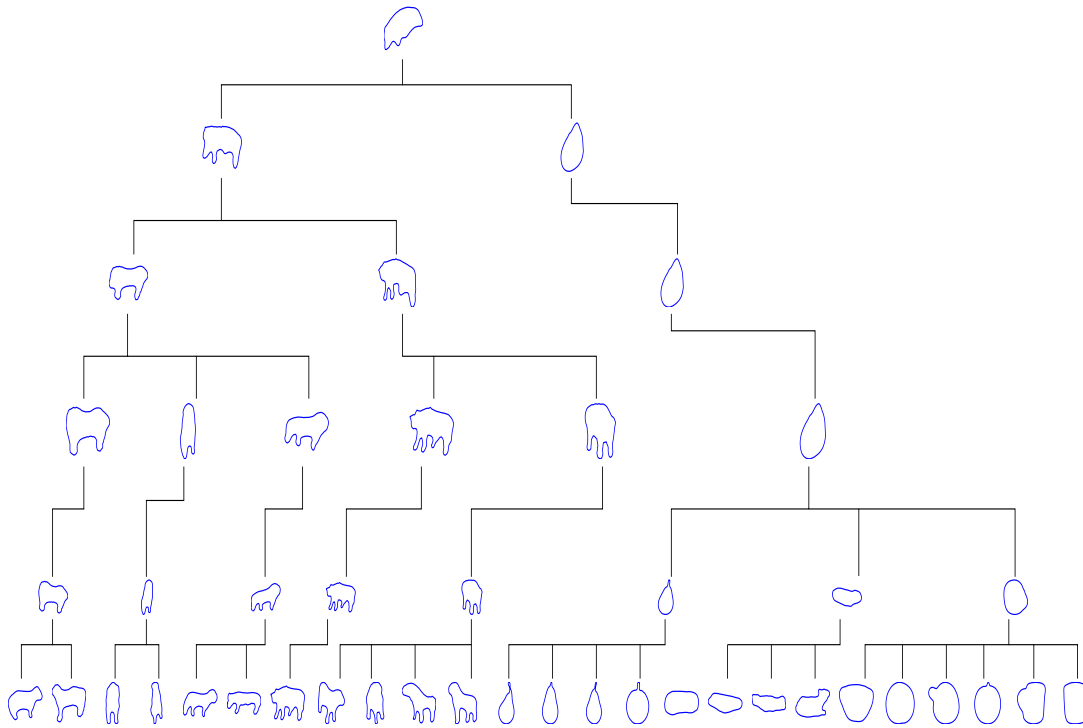


Fig. 4. Hierarchical Organization of 3270 shapes from the ETH-80 database.

shapes retrieved, and Fig. 6(c) plots precision rate against the recall rate, for the same four classes.

IV. CONCLUSION

We have presented a hierarchical organization of shapes based upon an elastic shape-distance metric which utilizes the Riemannian structure of the shape space. Clustering is performed efficiently by minimizing the pair-wise average variance within the clusters and can be used in clustering of shape databases of objects. Hierarchical clustering reduces the search and test times for shape queries against large databases. This has enormous potential for systems which use shape based object retrieval.

REFERENCES

- [1] William M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, Inc., 1986.
- [2] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2004.
- [3] W. Mio and A. Srivastava. Elastic-string models for representation and analysis of planar shapes. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2004.
- [4] A. Srivastava, A. Jain, S. Joshi, and D. Kaziska. Statistical shape models using elastic-string representations. In *Proceedings of Asian Conference on Computer Vision*, 2005.
- [5] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning and testing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):590–602, 2005.
- [6] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Text in Stat., 1999.
- [7] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, November 1998.
- [8] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.
- [9] D. E. Brown and C. L. Huntley. A practical application of simulated annealing to clustering. Technical Report IPC-TR-91-003, Institute for Parallel Computing, University of Virginia, Charlottesville, VA, 1991.
- [10] B. Leibe and B. Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2003.
- [11] A. Vailaya and M. Figueiredo and A. Jain and H. Zhang. Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing*, 10(1):117-130, 2001.
- [12] A. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, August 1996.
- [13] F. Mokhtarian, S. Abbasi and J. Kittler. Efficient and robust shape retrieval by shape content through curvature scale space. *Proceedings of First International Conference on Image Database and MultiSearch*, 1996.
- [14] F. Mokhtarian and A. Mackworth A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(8):789–805, 1992.
- [15] Sharvit, D. and Chan, J. and Tek, H. and Kimia, B.B. Symmetry-

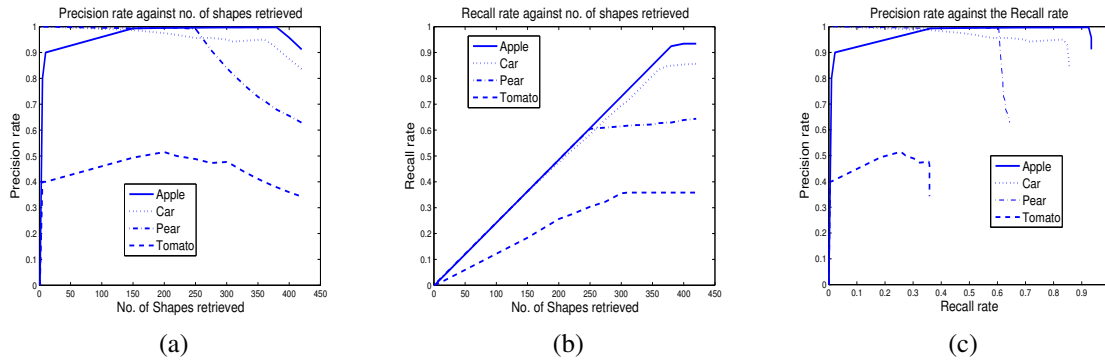


Fig. 6. (a) Precision rate versus number of shapes retrieved, (b) recall rate versus number retrieved, and (c) precision rate versus recall rate.

Shape Image	Shape Contour	Retrieved shapes from hierarchy

Fig. 5. Examples of shape retrieval using hierarchical organization.

based indexing of image databases. *Content-Based Access of Image and Video Libraries*, 1998

- [16] G. Cortelazzo and G. A. Mian and G. Vezzi and P. Zamperoni. Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27(8):1005–1018,1994.
- [17] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computer*, C-21, 269–281, 1972.
- [18] E. Persoon and K. S. Fu. Shape discrimination using Fourier descriptors. *IEEE Trans. Systems, Man, and Cybernetics*,(7):170–179, Mar. 1977.
- [19] J. Gary and R. Mehrotra. Shape similarity-based retrieval in image database systems. *In Proc. of SPIE*, 1662:2–8, 1992.
- [20] D. G. Shen and Horace. H. S. Ip. Discriminative Wavelet shape Descriptors for Invariant recognition of 2D patterns, *Pattern Recognition*, 32(2):151–166, February, 1999.
- [21] M. R. Teague. Image analysis via the general theory of moments. *J. Optical Soc. of America*,70(8):920–930, 1980.

Performance Metrics for Operational Mars Rovers

Edward Tunstel

Jet Propulsion Laboratory / California Institute of Technology

4800 Oak Grove Drive

Pasadena, CA, USA

tunstel@robotics.jpl.nasa.gov

Abstract— The concept of operational performance metrics is explored with a focus on mobility and robotic arm autonomy exercised on the NASA Mars Exploration Rovers (MER) surface mission. This space flight mission has been underway for over 2.5 years since January 2004. Autonomy functions of surface navigation, short-distance approach to surface science targets, and robotic placement of arm-mounted instruments on science targets are considered. Operational metrics that measure performance of these functions relative to system requirements are advocated. The metrics are computed using telemetry from the rovers' operations on Mars and applied to rate their performance during their respective missions. The metrics are applied using an existing methodology to aggregate multiple metrics into a composite performance score. Its formulation is augmented to accommodate importance weights that add flexibility in use of the metrics by different potential end-users e.g., sponsors, program managers, systems engineers, and technologists.

Keywords: *operational performance metrics, Mars rovers, MER, space robotics.*

I. INTRODUCTION

Robotic autonomy is increasingly required to achieve aspects of overall success for planetary surface missions. Current and near term missions include the NASA Mars Exploration Rovers (MER) mission and Mars Science Laboratory (MSL) mission planned for 2009, as well as the 2011 ExoMars mission of the European Space Agency. With MER, NASA landed twin rovers, named *Spirit* and *Opportunity*, on Mars in January 2004 (Fig. 1). These rovers were explicitly required to use mobility and robotic arm positioning functionality to achieve exploration mission objectives by serving as surrogate robotic field geologists for a science team on Earth. MSL and ExoMars may be required to do the same, and likely more, with greater demand on autonomy and lifetime.

MER now represents the longest deployment of planetary surface robots and a new benchmark in planetary robotic autonomy. As such, it is important to capture and document the rovers' performance in ways that facilitate relative evaluation of similar technologies. A previous study [1] initiated the groundwork necessary to establish mobility and related robotic arm autonomy used during MER surface operations as a state-of-the-art baseline useful for relative technology assessments. A set of performance metrics was introduced and used to apply a technology assessment algorithm for comparing current and future technologies with respect to impact on mission science return.

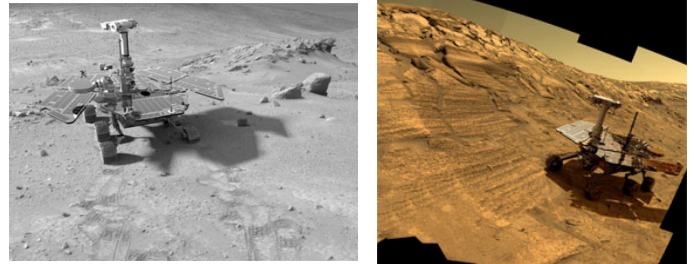


Figure 1. *Spirit* (left) in “Columbia Hills” and *Opportunity* (right) in “Endurance Crater” on Mars (Special-effects images created using photo-realistic rover models & image mosaics acquired during their missions. Rover model size approximated based on size of rover tracks in actual mosaic).

This paper focuses more intently on measuring rover navigation and robotic arm performance independent of impact on science return. In particular, the focus is on metrics for *operational* performance of deployed rovers as opposed to metrics for robot systems that are in experimental phases of development, verification, or validation. The intent is to sensitize the community to performance measurement issues of intelligent systems that are deployed (or decommissioned) and the need for associated metrics. This is particularly important now that we have a definitive benchmark for long-term planetary rover performance in the MER mission.

We begin in Section II with a discussion of related work on performance evaluation for similar robotic systems, pointing out distinctions between developmental and operational performance measurement. Section III suggests properties for operational metrics in relation to end-users, available mission data, performance references, and system levels at which performance is measured. A set of metrics for MER robotic autonomy, and the methodology used to apply them, is presented in Section IV. The metrics are computed based on actual telemetry data from *Spirit* and *Opportunity* on Mars. To apply the metrics, we adopt an existing methodology proposed for comparative performance evaluation and used as a decision aid to guide technology development investment decisions [2]. The method is applicable to systems in development and in operation. In Section V, the method is used to compute composite performance scores for each rover. The existing method is augmented in Section V to accommodate importance-weights for individual metrics. An example is included to illustrate how this enables user perspectives to influence calculation of the composite score. Remarks and conclusions are given in Section VI.

II. RELATED WORK

The importance of ways and means to measure performance is clear at this stage of intelligent systems development. Many relevant metrics driven by various motivations have been proposed in the literature, but it remains difficult to identify existing metrics that are generally applicable or otherwise relevant to specific systems.

Nehmzow [6] presented a quantitative measure of robot-environment interaction that estimates how sensitive a robot's trajectory is to changes in initial conditions. This is a straightforward concept that could be useful for measuring navigation or path planning performance but it is not directly applicable to a deployed operational system such as the MER vehicles. It is more useful for comparative analyses wherein runs can be repeated from different initial conditions, i.e., via experiments. Since neither Mars nor the MER mission is an empirical domain, there are no data from repeated runs.

Metrics that are more directly related to field mobile robots have also been proposed. Albus presented a set of performance measures for intelligent unmanned military scout vehicles that are based on metrics developed for measuring human performance in scholastic aptitude, athletic ability, and task performance [7]. The metrics are generally applicable and measurement is supported by test procedures, methodical ground-truth characterization of test environments, scenarios involving human drivers, and rich performance observability by high precision instruments and humans on site. Frost et al [8] described metrics and measurements used for testing iRobot Corporation's PackBot system. A set of application specific metrics are discussed for mobility, durability, situational awareness, communications, deployment, modularity, and endurance; the paper omits quantitative metric values and formulations defining the metrics. Related experimental evaluation work is described in [9] where the goal is to be able to predict the interaction between a mobile robot and complex terrain for the purpose of predicting the probability that a robot can traverse a given terrain. Experiments were conducted to investigate ways to develop, verify, and validate mobility models for different obstacle geometries. As such, it focused on performance of particular mechanical mobility systems. Recent efforts of similar focus prescribe mobility metrics for planetary rovers [10]. For operational performance assessment, we are interested in measures of functional autonomy enabled by the combination of mechanical capability and software-based intelligence.

Other metrics directly related to planetary rovers have been proposed. Sukhatme and Bekey [11] focused on evaluation and mission-oriented performance and proposed two generally applicable navigation metrics based on integrated time and energy. Application of the metrics call for calculating the probability of a rover successfully traversing a particular distance from its start location after some value of elapsed time or energy consumed. The approach depends on the availability of results, obtained from certain test campaigns prior to a mission, which would allow determination of the

required probabilities. Such test campaigns could be done during verification and validation, and ideally would establish some mapping between test-terrain characteristics and time/energy costs of traversing test-terrains. The probabilities and test-terrains would allow prediction of rover navigation performance during a mission on terrain similar to the test-terrains. Others have also invested the necessary effort to conduct test campaigns in the laboratory and in the field from which metrics for rover performance have been formulated or derived [12-15]. A common thread in these efforts is pursuit of a statistically significant number of trials needed to predict rover performance using metrics related to localization accuracy, traverse distance, energy efficiency, and degree of autonomy. Some are probabilistic and generally applicable to systems for which associated distributions are available. Where feasible, they account for resource and environmental factors related to sensor suite and terrain traversability.

Evaluation via large numbers of experimental trials is highly recommended. Unfortunately, the occurrence of such tests campaigns, or total trials sufficient to derive probabilities, is not always accommodated by project schedules and budgets during rover development phases [16]. When substantial pre-deployment testing (facilitating post-deployment performance measurement) is afforded by a project it is not always performed using a high-fidelity prototype of the deployed robotic system. While the metrics are broadly applicable, their key parameters are not always consistently observable on an actual planetary mission. In such cases, the above approaches and their by-products are of limited applicability for measuring operational planetary rover performance.

Most of the metrics mentioned above were motivated by, or primarily applicable to, systems in development or under experimental evaluation. They are representative of many metrics proposed for robotic systems for which the objective is to establish specifications of capability or enable prediction of system performance. In such situations, researchers often enjoy luxuries of controlled environments, high observability and broad selection of preferred measurables, and the abilities to reset or re-run trials and visually observe the robot system as it performs. For operational and, particularly, remote systems such as planetary rovers, some or all of these luxuries are rarely available. Such distinctions between *developmental* or *experimental* performance metrics and metrics that could be applied to *operational* systems should not be ignored.

A prime objective of operational rovers is achieving mission success, which to date has focused on science data collection and discovery. While rovers are instrumented with sensors that produce substantial amounts of engineering telemetry, the telemetry is not always tailored to support calculation of metrics used during development. Engineering telemetry content is typically defined for visibility into critical mission-long concerns of safety, operability, and survivability. The formulation of performance metrics for operational rovers is governed by the nature and availability of such telemetry. In some cases the configuration of mission components and operations facilitate use of developmental metrics. Take, for

example, the NASA Mars Pathfinder (MPF) mission of 1997, which deployed a lander and the rover *Sojourner* on Mars. A stereo camera on the lander was used routinely to establish ground truth and derive rover localization updates via direct observation of the rover on the terrain. This enabled mission operators on Earth to apply operational metrics for rover pose estimation accuracy and related navigation performance [17, 18]. The MER vehicles operate independent of instrumented landers or other *in situ* means of externally observing their motions; hence, metrics that require ground truth are more difficult (but not impossible) to apply.

III. DESIRABLE PROPERTIES OF OPERATIONAL METRICS

The performances of the *Spirit* and *Opportunity* rovers will serve as a benchmark that will influence future rover missions (e.g., MSL, ExoMars) and human-rover missions, as well as funding allocated to develop the necessary mission-enabling technology. Quantifying their robotic autonomy performance is important to a variety of customers or stakeholders including sponsors, program managers, flight systems engineers and technology developers. So, how should we define quantifiable measures to rate performance of rover autonomy functionality during post-deployment operation? This section offers thoughts on desirable properties for operational metrics, and the next section presents a set of metrics formulated with those properties in mind.

The MER mission architecture is a complex system [19] and isolation of robotic autonomy performance within the larger mission system is nontrivial. Acknowledging this and the spectrum of potential customers, it is desirable to formulate metrics that are *simple* and easy to interpret. Moreover, since the available set of engineering telemetry from the rover constrains what metrics can be formulated, operational performance metrics should be *functions of telemetry* or derived data products produced during operations.

To determine what the rover performance should be measured against, i.e., what establishes how well it performed on a mission, one should consider what the system was designed to do and what constrained its design. Tools for automatically encoding the intent of commanded rover actions could also be quite useful for providing a means of comparison against actual execution. Operational systems deployed in the real world have at least one thing in common. They are designed to meet a set of requirements subject to certain constraints (budget, schedule, etc). By virtue of the fact that they are deployed, they have met the imposed requirements (although this is not always the case). With good engineering, their performance is typically as good as the requirements and constraints would allow; and they could probably be better in certain respects or improved upon under different requirements and constraints. So, to first order, operational system performance should be measured with respect to *required* performance, i.e., *requirements-based*.

Requirements serve as a logical datum for operational performance. However, when requirements-based metrics cannot be formulated due to lack of correlation between

available telemetry and parameters specifying required performance, an alternative is to use as-tested performance as a reference. For example, engineers' assessments of *Sojourner's* autonomous navigation performance on Mars reported that it generally equaled or exceeded the performance observed during tests conducted on Earth prior to the MPF mission [17, 20]. Performance in pre-deployment verification or validation tests [11, 14] could also be used as a viable operational performance reference when: (a) the test cases are good representations of mission scenarios, (b) the test article is of high-fidelity with respect to the flight rover, or (c) when the test metrics can be computed using mission telemetry. Another alternative is to quantify performance relative to that of other robots or systems performing similar functions. For example, Albus suggests performance metrics for military unmanned ground vehicles be formulated relative to manned ground vehicle performance (accounting for terrain difficulty and other factors) [7]. This is similar to requirements-based performance for systems whose requirements are defined in terms of human driver performance.

Finally, operational performance metrics for rover autonomy should, in some way, measure *functional performance* of autonomous tasks. Measuring performance at this level is one way to manage the system complexity while still focusing on key activities that involve autonomy and directly contribute to mission success. MER surface robotic autonomy scenarios employ task-level functions including *Navigation*, *Approach* to science targets, and *Instrument Placement* onto science targets. Software and mechanical capability enables robotic execution of these functions. Robotic tasks include low-level safeguarded motion control and autonomous navigation functions of varying complexity for traversing the Martian surface, as well as robotic arm motion control for accurate placement of science instruments onto rocks and soil. Combined and repeated use of these capabilities enables acquisition of desired high priority science measurements, and thus contributes directly to mission success.

IV. PERFORMANCE METRICS FOR OPERATIONAL ROVERS

Each MER computer is a 20 MHz RAD-6000 processor (radiation-hardened version of a PowerPC chip) running the VxWorks real-time operating system, with 128 MB of DRAM and 256 MB flash memory and EEPROM. The rovers' primary source of power is a solar panel via which various amounts of solar energy are available depending on climate, regional atmospheric conditions, and levels of dust-cover settled on solar cells. The mobility and robotic arm software runs onboard the rovers' computers to consistently perform integral parts of various exploration tasks. Robotic tasks are specified in command loads uplinked to the rovers by engineers who plan their daily robotic activities on Earth given desired science activities prescribed by a team of scientists. Given a set of command sequences that would implement all exploration activities for a given day, *Spirit* and *Opportunity* set out to autonomously perform the necessary operational functions. Recalling the autonomy functions of

Navigation, Approach, and Instrument Placement, we desire a set of simple performance metrics for these activities.

Per discussion in the previous section, the goal thus far has been to formulate operational performance metrics that are:

- simple (and interpretable by diverse users/customers);
- focused on functional performance of autonomous tasks;
- supported by mission telemetry; and
- requirements-based where feasible.

Requirements-based operational performance metrics are presented and computed in this section using telemetry returned by *Spirit* and *Opportunity* over hundreds of sols (Martian days). Other non-requirements-based metrics that are useful for measuring science rover performance and are supported by MER telemetry are listed in section IV.D. The set of metrics in this paper is by no means complete but only a subset of many that could be applied to operational rover systems. They serve as a baseline set to be refined and expanded as needed in future work. While any reasonable set of performance metrics could be used for rover evaluation purposes, any relative comparisons with other rover systems are only proper if they employ the same metrics.

A. Methodology

Rodriguez and Weisbin [2] introduced a performance comparison methodology wherein primitive performance metrics are computed using base-2 logarithms of information theory, an idea drawn from similar approaches of complexity theory [22]. The method is sufficiently general to apply to many different domains. We adopt the approach to handle operational metrics due to several nice properties; e.g., it:

1. uses relative measures defined by ratios of system performance to a prescribed reference performance (e.g., other systems, requirements, etc), which can be arbitrarily selected by users;
2. handles disparate metrics with different units and scales by virtue of these dimensionless performance ratios;
3. expresses scores (via base-2 logs of dimensionless ratios) in a well understood unit, the binary bit, as a unifying dimension for relative performance;
4. generalizes quite easily to combine multiple metrics as an aggregated composite score that reflects the influence of each individual metric.

For our problem, we exploit the flexibility allowed by the first property above by choosing required system performance as the reference with respect to which the rovers' operational performance is evaluated, thus establishing requirements-based metrics. The basic analytical expression of the base-2 logarithmic performance metric [2], using required performance as a reference, is

$$\log_2 \left[p(m,r) / p(m,req) \right] \quad (1)$$

where $p(m,r)$ represents the performance of rover r for primitive metric m and $p(m,req)$ is the required performance

for primitive metric m . (Note that if values of metric m are interpreted such that smaller numbers imply better performance, then the reciprocal of the ratio in expression (1) is used). Let $P(m,r)$ represent the performance ratio, i.e., the argument to the base-2 logarithm in (1), for a single primitive metric m . If we employ a set of N primitive metrics of rover r 's performance, we can denote the N performance ratios as $P(m_i,r)$, $i = 1,2,\dots,N$. It can be shown [2] that a composite score, $Score(r)$, representing the overall performance of rover r considering the N metrics can be expressed as

$$Score(r) = \frac{1}{2} \sum_{i=1}^N \log_2 \left[P^2(m_i,r) \right]. \quad (2)$$

In the following subsections we present primitive operational metrics for navigation, approach, and instrument placement and compute corresponding performance ratios, P , for *Spirit* (S) and *Opportunity* (O).

B. Autonomous Navigation

The MER vehicles execute a navigation algorithm called GESTALT (Grid-based Estimation of Surface Traversability Applied to Local Terrain), which is documented in [23]. GESTALT performs stereo vision-based perception, local terrain hazard mapping, traversability assessment, and incremental goal-directed path selection through its onboard traversability map. In addition to these forms of autonomy for navigation, the rovers' onboard software performs visual odometry (on command) and reactive fault protection to achieve self-localized and safeguarded mobility.

The rovers were required to traverse a total accumulated path length of at least 600 meters (with a goal of reaching 1 km) over the course of their primary 90-sol missions. In addition, they were required to be capable of safely navigating at a low average rate of 35 m/hr autonomously in rocky terrain (~7% rock abundance) to designated positions on the surface. The required rate of autonomous traverse was derived from an early desire to be able to navigate 100 m in a sol. It was recognized that mission-specific issues such as operational risk concerns or terrain conditions would influence the degree to which such requirements were met.

1) *Total Traverse Distance*: Total distance traversed by the rovers is used here as a primitive metric to be measured relative to the distance required for mission success. At the end of their primary missions, both rovers exceeded the traverse distance success criterion of 600 m. Since MER operations continued well beyond the primary 90-sol duration into several extended missions we use the 1 km goal (which became an engineering objective of the first extended mission) as a reference distance, $p(TT, req)$.

MER traverses are commanded using a variety of mobility modes ranging from sequenced segments of driving motions without hazard avoidance enabled (i.e., "blind") to full autonomous navigation, including visual odometry on occasion. Since our focus in this paper is on robotic

autonomy we consider total distance traversed autonomously (i.e., using vision-based hazard detection and avoidance or visual odometry versus manually sequenced drive primitives with autonomy disabled). July 21, 2006 was sol 906 for *Spirit* and sol 885 for *Opportunity*. By that date, the total traverse distances for *Spirit* and *Opportunity* were 6876 m and 8603 m, respectively. *Spirit* traversed 3126 m (45%) autonomously, and *Opportunity* 2397 m (27%) for total traverse distance performance ratios of $P_{TT,S} = 3.126$ and $P_{TT,O} = 2.397$.

2) *Terrain-based Autonomous Navigation Speed*: The average rate at which a rover can traverse autonomously depends on the traversability of the terrain over which such a measurement applies (among other things). A given rover may traverse flat and hazard-free terrain at a faster average rate than it would a sloped and rocky terrain due to the increased deliberation required in the latter case. Therefore, a metric for traverse rate might ideally account for terrain type in some meaningful way. A primitive metric that magnifies the speed impact with terrain difficulty such that benign to very difficult terrain has a nil to double effect is:

$$(AverageAutonavSpeed)(1+\alpha) \quad (3)$$

where $\alpha \in [0,1]$ is the percentage rock abundance of the local or regional terrain traversed. This terrain parameter could be any normalized measure of terrain difficulty, in general, and could include additional regional measures of roughness, slope, traversability, trafficability, etc.

As mentioned above, the MER requirement was to navigate at 35 m/hr autonomously in rocky terrain of ~7% rock abundance. *Spirit's* average and maximum autonomous traverse rates at its Mars landing site (Gusev crater, ~7% rock abundance [24]) thus far are 15.06 m/hr and 34.35 m/hr, respectively. The same traverse rates for *Opportunity* at its landing site (Meridiani Planum) thus far are 22.26 m/hr and 36.0 m/hr, respectively. The Meridiani site is roughly devoid of rocks and observations suggest a rock abundance of only a few percent [24]; we use 3% here. This terrain-based autonomous navigation speed metric, expression (3), yields the following performance ratios for the average rates and rock abundances above: $P_{TAS,S} = 0.43$ and $P_{TAS,O} = 0.61$.

C. Approach and Instrument Placement

In addition to traversing from place to place, science rovers must deploy instruments in contact with or in proximity to reachable rocks and soil specimens. Therefore, an instrument positioning system with the ability to perform precision placement of instruments from mobile platforms is essential [25]. *Spirit* and *Opportunity* perform this function using a five degree-of-freedom (DOF) robotic arm known as the Instrument Deployment Device (IDD). It is mounted in a frontal area beneath the rovers' solar panel. Its end-effector is a rotary turret to which science instruments are mounted, and the remaining 4 DOFs are used to place the instruments onto science targets within its kinematic work volume (~0.14 m³). Rover mobility is used to approach a position offset from a

science target such that the target is within the work volume of the IDD. A successful target approach is typically followed by placement of instruments onto the target using the IDD.

1) *Approachability*: An approach traverse refers to a one on the order of 10 m or less that is intended to terminate with a specific science target within the IDD work volume. The science target is selected and designated by mission operators in stereo imagery acquired prior to the approach. Each rover was required to be capable of approaching a reachable science target in a single command cycle whenever the rover was within 2 m of the target at the start of a sol. However, depending on approach distance from a target, complexity of terrain between rover and target, and other considerations, target approach executions do not always succeed on first attempts. On occasion, more than one sol is needed to reach certain targets, particularly when approach distances are longer than 2 m. An appropriate figure of merit for performance of approach traverses considers distance to targets and the number of sols that were necessary to reach the targets. We employ such a measure as an average approach distance achieved (d_{app}) per unit sol needed (n_{sols}) during successful approaches to N targets as follows.

$$Approachability = \frac{1}{N} \sum_{i=1}^N \left(\frac{d_{app}}{n_{sols}} \right)_i \quad (4)$$

This primitive metric is evaluated relative to the required $p(APP,req) = (d_{app} / n_{sols}) = 2$ m/sol. The Approachability for *Spirit* and *Opportunity* was determined for a set of $N = 40$ approach traverses and found to be 5.85 m/sol for *Spirit* and 4.97 m/sol for *Opportunity*. These values yield the following performance ratios: $P_{APP,S} = 2.92$ and $P_{APP,O} = 2.48$.

2) *Positioning Accuracy and Repeatability*: Instrument placement is achieved autonomously (including switching from one instrument to another) by realizing a combination of kinematic configurations that are pre-taught and/or newly commanded. Motions are determined via onboard calculation of inverse kinematics and position error compensation (due to mechanical compliance of an as-built flexible link assembly and gravity effects of a given rover attitude). Original MER operational guidelines required human confirmation of the rover position prior to each IDD use, making each approach and instrument placement take at least two sols, but software upgrades made in summer of 2006 will make approach and placement possible in the same sol. The requirements on placement performance hold in either case.

Positioning accuracy herein refers to the arm's ability to position and orient its tools at a specified absolute location in its workspace. Repeatability refers to the difference between the initial and final positions and orientations of a tool when moved back and forth between an initial position and a designated position. These attributes of instrument placement performance are used directly as primitive metrics to be evaluated against required absolute positioning accuracy and

repeatability. The MER IDD was required to be capable of achieving a position accuracy of 5 mm in free space within its dexterous workspace. The IDD repeatability was required to be 4 mm in position [25]. Performance results from *Spirit* and *Opportunity* during surface operations revealed an absolute positioning accuracy of 0.8 mm, and a repeatability of approximately 1 mm [26]. This performance was derived based on telemetry and stereo image range data evaluations of 422 placements of all instruments by *Spirit* and 439 placements of all instruments by *Opportunity* on rock, soil, and rover-mounted targets [25].

Performance ratios for instrument placement positioning are thus: $P_{IPP,S} = P_{IPP,O} = 6.25$; and the ratios for instrument placement repeatability are: $P_{IPR,S} = P_{IPR,O} = 4.0$.

D. Other Rover Performance Metrics

Additional MER-focused performance metrics were presented in recent work [1]. While supported by MER telemetry, they are not easily related to quantified MER requirements as formulated. They are therefore not requirements-based but meaningful nonetheless. They could be used in comparative evaluations relative to alternative references such as as-tested performance, results of pre-deployment test campaigns, or even past, present, or future systems as in [1]. The metrics are listed in Table I for completeness; details can be found in [1].

TABLE I
ADDITIONAL ROVER PERFORMANCE METRICS

<p>Autonomous Traverse Speed Ratio:</p> $ATSR = \frac{\text{AverageAutonavSpeed}}{\text{MaximumAutonavSpeed}}$	<p>Navigation Step Time:</p> <p><i>Time required to perceive local terrain, detect hazards, select hazard-free path, and execute 35cm drive (nominal) along selected path.</i></p>
<p>Percent Autonomous Traverse:</p> $PAT = \left(\frac{d_{\text{auto}} + d_{\text{visod}}}{d_{\text{blind}} + d_{\text{auto}} + d_{\text{visod}}} \right) * 100$	<p>Mean Self-Localizations Per Sol:</p> <p><i>Mean number of position updates per traverse sol with visual odometry enabled</i></p>
<p>Mobile Manipulability:</p> <p><i>Ratio of dexterous arm workspace volume to mobile platform volume</i></p> $\text{MobileManipulability} = \frac{V_{\text{manip}}}{V_{\text{mobile}}}$	

Observe that metrics presented above and listed in Table I do not include other meaningful performance related factors such as available resources (power, computing, etc). While such resources should play a role in system evaluation, the task of correctly correlating resources to execution of isolated robotic autonomy activities is not well supported by the complex MER architecture. The difficulty is compounded by the wide variability of daily resources due to dependence on solar energy, and susceptibility to thermal effects and other daily environmental impacts. The system executes many tasks other than those related to robotic autonomy. Mapping resource usage into simple metrics is nontrivial and would have to be a subject for work of a larger scope than reported

here. Note, however, that the performance evaluation methodology adopted here readily handles resource metrics as well as it does performance metrics; in fact, it does so in an identical fashion, both conceptually and computationally [2].

Also worthy of mention in the context of this paper is ongoing mission-focused work, under the NASA/JPL Mars Technology Program in support of the upcoming 2009 MSL rover mission, to validate autonomous navigation, visual odometry, and visual target tracking performance [27]. The validation efforts will facilitate flight mission infusion of new technologies by running the controlled test campaigns needed to produce extensive experimental performance data for benchmarking autonomy algorithms [28]. Prototype rover systems are being used to test and validate software functionality against relevant MSL requirements in multiple terrain types that are generally more challenging than those encountered by the MER vehicles. Performance metrics used to date for this experimental validation work include:

- Success rate in reaching navigation waypoints
- Total distance travelled to reach desired waypoints
- Navigation steps required to reach desired waypoints
- Comparison of wheel to visual odometry state estimates
- Number of terrain features tracked by visual odometry
- Visual target tracking accuracy
- Visual target tracking reliability.

These metrics are also not easily related to quantified MER requirements. However, they stand a good chance of being used as requirements-based metrics for MSL operational performance since they are being used to validate algorithms against MSL requirements in extensive test campaigns.

V. COMBINING PERFORMANCE METRICS

A given set of numerically evaluated metrics can be aggregated in a variety of ways to compute an overall performance score. We apply Eq. (2) to compute composite performance scores for *Spirit* and *Opportunity*, $Score(S)$ and $Score(O)$, representing their overall performance considering the $N = 5$ metrics presented in Section IV. These are listed in Table II along with the sets of performance ratios arrived at in Section IV. It is not surprising that the scores are practically equal given that we are comparing performance of two essentially identical systems to a common set of requirements.

TABLE II
PERFORMANCE SCORES FOR MER ROBOTIC AUTONOMY

Operational Performance Metric	Performance Ratios	
	<i>Spirit</i>	<i>Opportunity</i>
Total Traverse Distance	3.126	2.397
Terrain-based Autonomous Nav. Speed	0.43	0.61
Approachability	2.92	2.48
Instrument Placement Position Accuracy	6.25	6.25
Instrument Placement Repeatability	4.00	4.00
SCORE (bits)	6.62	6.50

Recall that the use of the binary logarithm to compute metrics and overall scores yields results in units of bits [2].

Such scores can be interpreted as the number of multiples of 2 that the reference performance would have to be multiplied by to achieve the performance of the system being evaluated. In our case, using required performance as the reference for comparison, the scores in Table II tell us the following. *Spirit's* and *Opportunity's* operational performance thus far (based on the metrics applied) has been nearly 7 multiples of 2 better than required performance.

This methodology differs from one of the more commonly used approaches to aggregating primitive scores, which is to use a linear sum of weighted metrics wherein numerical weights impose degrees of importance on each metric [29, 30]. Applications of weighted linear combination techniques have been criticized for presupposing the existence of good combination models and involving arbitrary assignment of weights [11]. On the contrary, flexibility in assignment of importance weights is a desirable feature in our case since it allows the allocation of importance to be tailored by different users of the approach. Sponsors, program managers, flight systems engineers and technology developers would be free to use different weight assignments that, from their perspectives, reflect the relative importance of each metric for the mission, technologies, robotic system, or task performance.

We believe the base-2 logarithmic approach serves as a good model with nice properties that could be enhanced by the expressiveness that importance-weights enable. As formulated in Section IV.A, however, it embeds an implicit assumption that each of the N metrics is weighted equally. We can impose importance-weights on the formulation by simply introducing scalar multiples, w_i , $i = 1, 2, \dots, N$, in the summation of Eq. (2) as follows

$$\text{Score}(r | W) = \frac{N}{2} \sum_{i=1}^N w_i \log_2 \left[P^2(m_i, r) \right] \quad (5)$$

to represent an overall score for rover r considering the N metrics, and given a set of weights $W = \{w_1, w_2, \dots, w_N\}$. In this manner, each $w_i \in W$ is specified by the user or performance evaluator to individually prescribe relative importance to each metric. Relative assignment of importance weights can be ensured by constraining the sum of all w_i to equal unity, so that each weight induces some part in the N -part overall performance measure.

A. Example

While *Spirit* and *Opportunity* are essentially of the same hardware and software design, their respective modes of operation and missions on Mars have differed sufficiently for us to view them as the same system performing different missions. Regional surface locations (terrain, climate, topography, etc), the way they were used by the science team for exploration, and the software functional configurations were different for each rover [31, 32]. However, their functional capabilities of navigation, approach, and instrument placement remained similar (until recent aging-related

degradations in performance of different respective hardware components). In light of the differences in character of each mission, and for the purpose of illustration, we suggest two possible sets of importance-weights, W_s and W_o , and compute corresponding performance scores. These are sets of subjective weights prescribed to *Spirit* and *Opportunity* performance based on qualities of their respective missions, i.e., the importance or utility of navigation, approach, and instrument placement for the conduct of each mission [33].

Reasonable subjective conclusions about *Spirit's* mission are that total traverse distance (beyond the 1 km objective for its extended mission) was initially very important for reaching distant hills of high science priority but not as important afterwards. Navigation speed was not as important since there were many interesting science targets to explore en route to the hills. In addition, the terrain was challenging at various locations and particularly so while traversing the hills. The ability to approach science targets most efficiently was important given the large number of reachable targets on the terrain. Finally, IDD placement accuracy and repeatability (perhaps more so) were both important for investigating the science targets. *Opportunity's* mission has relied on long traverse distances since locations of highest priority science targets were few and far in between. Traverse speed was important for reaching the next high-priority location, but autonomous navigation was not as important since the terrain was essentially obstacle-free. The high-priority science locations tended to be craters of various sizes with targets on sloped terrain, many of which were small soil targets versus larger rock targets. Safe, efficient, and accurate approaches were very important as a result, as was accurate and repeatable placement of arm-mounted instruments. Plausible sets of weights for these two mission assessments are listed in Table III along with associated weighted scores per Eq. (5) and using performance ratios from Table II. The effect of importance-weights is apparent in the enhanced score for *Opportunity's* performance relative to its non-weighted score in Table II. *Spirit's* weighted score is about the same as in Table II given W_s . While this example is posed after the fact, such weight selections could be made *a priori* based on expected mission qualities.

TABLE III

EXAMPLE OF WEIGHTED PERFORMANCE SCORES FOR MER ROBOTIC AUTONOMY BASED ON "MISSION CHARACTER"

Operational Performance Metric	Importance Weights	
	W_s	W_o
Total Traverse Distance	0.30	0.35
Terrain-based Autonomous Nav. Speed	0.20	0.05
Approachability	0.15	0.20
Instrument Placement Position Accuracy	0.15	0.20
Instrument Placement Repeatability	0.20	0.20
(bits)	Score($S W_s$)	Score($O W_o$)
	6.39	7.97

VI. SUMMARY AND CONCLUSIONS

Performance measurement of operational rover systems is generally more difficult than performance measurement of developmental rover systems. Some of the primary reasons, depending on the deployment situation, may be lack of controlled conditions (such as test cases and environments), limited sensor-based or visual observability of performance, and lack of infrastructure for ground truth. Improvements may be possible when systems are developed to facilitate post-deployment performance measurement or when it is embraced as a design goal. This paper highlights some of the issues and presents ideas for formulating and applying operational performance metrics using the MER vehicles as a case study. Simple metrics were presented that are supported by available mission telemetry and interpreted with respect to system requirements as a baseline performance reference. In addition, a means for diverse end-users to tailor performance evaluations by assigning importance-weights to individual metrics is suggested. By grounding performance metrics to requirements, similar systems that were each designed to meet some intersecting set of requirements could be compared on more common grounds. The motivation is to keep metrics simple since they are often used as inputs to high-level decision processes as made, for example, by sponsors for granting of funds, program managers for program planning or development, systems engineers for technology infusion, or technologists for engineering trade-offs.

ACKNOWLEDGEMENT

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration and funded by the NASA Mars Exploration Program, Science Mission Directorate. Contributions of the MER Development, MER Mission Operations, and Athena Science Teams are acknowledged as well as contributions from Drs. Mark Maimone and Ashitey Trebi-Ollennu to formulation of MER metrics that motivated this work.

REFERENCES

- [1] E. Tunstel, A. Howard, M. Maimone and A. Trebi-Ollennu, "Mars Exploration Rover Baseline for Flight Rover Autonomy Technology Assessment," 8th Intl. Symp. on Artificial Intelligence, Robotics, and Automation in Space, Munich, Germany, 2005. (http://robotics.estec.esa.int/AUTOLINKS/i-SAIRAS/isairas2005/session_07a/4_tunstel_7a.pdf)
- [2] G. Rodriguez and C.R. Weisbin, "A New Method to Evaluate Human-Robot System Performance," *Autonomous Robots*, vol. 14 nos. 2-3, March-May 2003, pp.165-178.
- [6] U. Nehmzow, "Quantitative Analysis of Robot-Environment Interaction: Towards "scientific mobile robotics"," *Intl. Journal of Robotics and Autonomous Systems*, Vol. 44, 2003, pp. 55-68.
- [7] Albus, J.S., "Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles, Proc. Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD, August 2002.
- [8] T. Frost, C. Norman, S. Pratt, B. Yamauchi, B. McBride and G. Peri, "Derived Performance Metrics and Measurements Compared to Field Experience for the PackBot," Proc. Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD, August 2002.
- [9] B. McBride, R. Longoria and E. Krotkov, "Measurement and Prediction of the Off-Road Mobility of Small, Robotic Ground Vehicles," Proc. Perf. Metrics for Intelligent Systems Workshop, Gaithersburg, MD, 2003.
- [10] N. Patel and A. Ellery, "Performance Evaluation of Autonomous Mars Mini-Rovers," Proc. TAROS 2004, Towards Autonomous Robotic Systems, University of Essex, Report CSM-415, Technical Report Series, Dept. of Computer Science, Chichester, Essex, UK, Sept. 2004, pp. 133-140.
- [11] G.S. Sukhatme and G.A. Bekey, "Multicriteria Evaluation of a Planetary Rover," Proc. IEEE International Conference on Robotics and Automation, Minneapolis MN, April 1996.
- [12] E.P. Krotkov, R.G. Simmons and W.L. Whittaker, "AMBLER: Performance of a Six-Legged Planetary Rover," *Acta Astronautica*, vol. 35, no. 1, 1995, pp 75-81.
- [13] E. Gat, "Towards Principled Experimental Study of Autonomous Mobile Robots, *Autonomous Robots*, vol. 2, 1995, pp. 179-189.
- [14] L. Matthies, E. Gat, R. Harrison, et al, "Mars Microrover Navigation: Performance evaluation and enhancement," *Autonomous Robots*, vol. 2, no. 4, 1995, pp. 291-312.
- [15] C.R. Weisbin, G. Rodriguez, P.S Schenker, et al, "Autonomous Rover Technology for Mars Sample Return," 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, The Netherlands, June 1999, pp. 1-10.
- [16] E. Tunstel, "Autonomous Mobility Software Validation Challenges for Planetary Surface Missions," Intl. Conf. on Space Mission Challenges for Information Technology (SMC-IT), Pasadena, CA, July, 2003, pp. 167-173.
- [17] A. Mishkin, J.C. Morrison, T.T. Nguyen, et al., "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover," Proc. IEEE Aerospace Conference, Aspen, CO, March 1998.
- [18] B. Wilcox and T.T. Nguyen, "Sojourner on Mars and Lessons Learned for Future Planetary Rovers," Proc. 28th ICES Intl. Conf. on Environmental Systems, SAE Society of Automotive Engineers, Danvers, MA, July 1998.
- [19] Proceedings of the 2005 IEEE Intl. Conf. on Systems, Man, and Cybernetics, vols. 1 and 2, October 2005, Waikoloa HI.
- [20] J. Matijevic, "Autonomous Navigation and the *Sojourner* Microrover," *Science*, vol. 280, no. 5362, April 17, 1998, pp. 454-455.
- [22] G. Nocolis and I. Prigogine, *Exploring Complexity: An Introduction*, W.H. Freeman & Co., New York, 1989.
- [23] J.J Biesiadecki and M.W. Maimone, "The Mars Exploration Rover Surface Mobility Flight Software: Driving ambition," *IEEE Aerospace Conf*, Big Sky, MT, 2006.
- [24] M.P. Golombek, R.E. Arvidson, J.F. Bell III, et al, "Assessment of Mars Exploration Rover Landing Site Predictions," 36th Lunar and Planetary Science, League City, TX, March 2005, Paper No. 1542.
- [25] A. Trebi-Ollennu, E.T. Baumgartner, C. Leger and R.G. Bonitz, "Robotic Arm In-Situ Operations for the Mars Exploration Rovers Surface Mission," IEEE Intl. Conf. on Systems, Man, and Cybernetics, October 2005, Waikoloa HI, pp. 1799-1806.
- [26] E.T. Baumgartner, et al, "The Mars Exploration Rover Instrument Positioning System," *IEEE Aerospace Conf*, Big Sky, MT, March 2005.
- [27] R. Volpe, "Rover Technology Development and Mission Infusion Beyond MER," IEEE Aerospace Conference, Big Sky, Montana, Mar. 2005.
- [28] W.S. Kim, R.D. Steele, A.I. Ansar, K. Ali and I. Nesnas, "Rover-Based Visual Target Tracking Validation and Mission Infusion," AIAA Space 2005, Long Beach, CA, Aug. 2005.
- [29] A. Freedy, J. McDonough, R. Jacobs, et al, "A Mixed Initiative Human-Robots Team Performance Assessment System for Use in Operational and Training Environments," Proc. Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD, 2004.
- [30] K.R. Leitzau, *Mars Micro Rover Performance Measurement and Testing*, M.S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1993.
- [31] C. Leger, A. Trebi-Ollennu, J. Wright, et al, "Mars Exploration Rover Surface Operations: Driving *Spirit* at Gusev Crater," Proc. IEEE Intl. Conference on Systems, Man, and Cybernetics, Waikoloa, HI, October 2005, pp. 1815-1822.
- [32] J. Biesiadecki, E. Baumgartner, R. Bonitz, et al, "Mars Exploration Rover Surface Operations: Driving *Opportunity* at Meridiani Planum," Proc. IEEE Intl. Conference on Systems, Man, and Cybernetics, Waikoloa, HI, October 2005, pp. 1823-1830.
- [33] J. Biesiadecki, C. Leger and M. Maimone, "Tradeoffs Between Directed and Autonomous Driving on the Mars Exploration Rovers," Proc. Intl. Symposium of Robotics Research, San Francisco CA, 12-15 October 2005.

Traversability Metrics For Urban Search and Rescue Robots On Rough Terrain

V. Molino, R. Madhavan[†], E. Messina, T. Downs, A. Jacoff, and S. Balakirsky

Intelligent Systems Division, National Institute of Standards and Technology (NIST)

Gaithersburg, MD 20899-8230, U.S.A.

Email: vmolino@cims.nyu.edu, {raj.madhavan, elena.messina, anthony.downs, adam.jacoff, stephen.balakirsky}@nist.gov

Abstract—Rough terrain, such as the rubble that we would expect to find in urban disaster areas, will likely impede robot mobility. The goal of this paper¹ is to find methods for quantifying the difficulty a robot should encounter traversing a region of rough terrain. We construct three metrics describing rough terrain robot mobility. In order to simplify the problem we assume that the rough terrain in question can be discretized in a certain manner and then we develop the metrics for this discretized version of the terrain. Two of these metrics reflect the difficulty a robot would have trying to move over the entire region of terrain, which is what we refer to as the coverability. The other metric describes the difficulty a robot would encounter attempting to move from some fixed point on the terrain to some other fixed point, which we call the crossability. We compute some coverability numbers for NIST step fields and briefly analyze the numerical data that are obtained.

Keywords: roughness, rough terrain, step field, traversability, coverability, crossability

I. INTRODUCTION

When a robot is to be deployed in an urban disaster area we should expect it to encounter many different types of terrain that will pose varying degrees of difficulty to its mobility. For instance, rubble will often be present in such environments and the various properties of the rubble will greatly influence a robot's motion capabilities. Some of the aspects of the terrain that will affect the mobility of robots attempting to traverse it include, but are not limited to, the following:

- How rough is the terrain, i.e., how large are the small scale height variations of the terrain?
- What is the terrain's composition, i.e., what is it made of?
- Is the terrain stable or are there loose sections of the terrain?

In order to effectively use robotic tools for urban search and rescue, we must first come up with an accurate and robust system for classifying the traversability of the different types of terrain the robots will be encountering. It is too difficult

to address all of the issues related to terrain traversability simultaneously so this paper will focus on classifying the traversability of terrain that is assumed to be uniform in composition and stable but has varying degrees of roughness.

We develop three different metrics for terrain traversability. Two of the metrics correspond to the difficulty a robot would have attempting to cover every part of the terrain. It is important to be able to measure such a quantity since a robot performing a search and rescue mission might have to cover all of the terrain in order to be sure that no victims are located in that region. The other metric corresponds to the difficulty a robot would have moving from a given point on the terrain to some other point. This is also a quantity we would like to be able to measure since we might have some information regarding where a victim is located so we may wish to send a robot directly to that location.

In Section II of this paper we briefly discuss some of the previous research that has been done in the field of rough terrain robot mobility. Section III defines some key terms and concepts that are needed in order to understand the work that is being described in this paper. In Section IV we develop two metrics for terrain coverability, which represents the difficulty a robot would have moving over every part of a region of rough terrain. Section V addresses terrain crossability, which is the difficulty encountered when a robot tries to move from some fixed point to another given point. Then, in Section VI we present some numerical results obtained by computing the coverability metrics for four different step fields. Finally, Section VII discusses some of the conclusions that can be drawn from the research that we have conducted.

II. PREVIOUS WORK

A fairly large amount of research has been performed in the area of robot mobility, with much of the work on rough terrain mobility being done in the past ten to fifteen years. For a detailed survey of previous work, one should read chapter one of [1]. While some of the results of this research are useful in the construction of traversability metrics, most of it is not directly applicable for several key reasons.

Most past research on rough terrain robot mobility has focused on a detailed analysis of a specific type of robot traversing a region of rough terrain. In particular, researchers have come up with relatively complex mathematical models

¹Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

[†]Research Staff Member, Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA.

describing the static and dynamic stability of an individual robot moving on rough terrain, often for the purposes of path planning. For examples of such work see [2] and [3]. While such detailed models for specific robots may prove to be quite accurate, they fail to display a satisfactory degree of generality for our purposes since they cannot be applied to a wide enough range of robot geometries. Therefore, these robot-specific models should not be used to construct traversability metrics, which should mostly be classifying the inherent properties of the rough terrain. As a result, we must construct simpler, more general models that take into account only a minimal amount of information pertaining to the robot and focus more heavily on terrain characteristics.

Also, many traditional models of rough terrain mobility used for path planning represent obstacles and open space in a binary format. As a result, every point on the terrain is either considered an obstacle, meaning it cannot be traveled over, or completely open, meaning it poses no difficulty to robot mobility. This is undesirable for our purposes since there should be a continuous scale between terrain that can be traversed very easily and impassable obstacles. For a survey of traditional path planning methods, one should consult [4].

Although very little research has been done on objectively classifying rough terrain traversability for mobile robots, there are a few notable papers on this topic. For example, both [5] and [6] discuss the construction of a so-called traversability index, which is meant to classify the difficulty a robot would encounter attempting to traverse a region of terrain. In both of these papers, fuzzy logic is used to obtain the traversability index.

III. DEFINITIONS: ROUGHNESS, TRAVERSABILITY, AND STEP FIELDS

A. Measuring Roughness

Roughness is defined to be a measure of the small-scale variations in the height of a physical surface. Hence, for the purposes of this paper, we shall let rough terrain refer to terrain that is uniform in composition and stable but may display significant small-scale height variations. We expect terrain roughness to be directly related to robot traversability [7].

There is no single universally-accepted method for quantifying the roughness of a surface and the different methods of roughness classification may be suitable for different purposes. For example, many researchers use statistical roughness parameters such as average roughness (R_a), root mean square roughness (R_q), or the maximum peak height (R_p) to describe a surface's roughness [8]. Others feel that the fractal dimension of a surface is a good way to numerically characterize its roughness, although, it only makes sense to speak of the fractal dimension of a surface if that surface displays some sort of self-similarity at different magnification scales [9]. Still others construct their own roughness indices for their own specific purposes. For example, the roughness of natural water channels is often specified by a number called Manning's n -value [10].

B. From Roughness to Traversability

Even if we can decide on the appropriate method for quantifying the roughness of some patch of terrain, we will still need to find a way to go from terrain roughness to terrain traversability. We know that terrain roughness should be related to robot traversability, but we do not know the exact nature of this relationship.

This leads us to ask precisely that we mean by traversability. If we want the traversability of a patch of rough terrain to correspond to the difficulty a robot would have covering every part of the terrain, as a robot would likely have to do if it were performing a search of the region, then perhaps some modified roughness parameter would be a suitable estimate for traversability. However, if we want the traversability of a patch of rough terrain to correspond to the difficulty a robot would encounter getting from some fixed point to some other point, as would likely be the case if the robot had information regarding the location of a victim in need of assistance, then we would expect roughness to be a very bad proxy for traversability. Since both the ability to cover all of the terrain and the ability to cross it (from some fixed point to some other fixed point) are important for urban search and rescue robots, we must come up with different metrics for terrain traversability representing these different goals.

Hence, we define the coverability of some region of rough terrain to be a measure of the difficulty a robot would have moving over every section of that region. Similarly, we define the crossability of some region of rough terrain from point p to point q to be a measure of the difficulty a robot would have moving from point p to point q . We will make these definitions more precise later in this paper, when we express them mathematically as functions of the terrain topography and certain dimensions of the robot that is traversing the terrain.

C. Step Fields as an Approximation to Rough Terrain

In order to test the capabilities of urban search and rescue robots moving across rough terrain, it is necessary to have a describable, reconfigurable, repeatable test apparatus to challenge robot mobility. To this end, the National Institute of Standards and Technology (NIST) developed random step fields. A random step field consists of an array of square wooden blocks cut to assorted cubic unit lengths (a unit being the post width) and arranged in different geometric patterns. When several of these step fields are configured into a sequential series or side by side into a "field", they provide an abstract but easily fabricated surrogate for rubble, debris, or other complex ground environments. A picture of a robot traversing a group of step fields can be found in Figure 1.

The facts that step fields are a standard test apparatus used for challenging robot mobility and that they form good surrogates for rubble, debris, or other challenging ground conditions make them an excellent place to begin our analysis of rough terrain. We note that assuming that our rough terrain has the structure of a step field is not overly restrictive since



Fig. 1. A robot traversing step fields.

we can always discretize our terrain into a rectangular grid in order to obtain a step field structure.

Before we proceed, it will be useful to establish some standard notation that can be used for performing calculations relating to step fields. If we fix the post width to be one unit then it is clear that a step field with m rows of blocks and n columns of blocks can be completely described by an $m \times n$ matrix of real numbers with each entry representing the height of a certain post. For any such matrix A we will let SF_A denote the associated step field. In general, the NIST step fields are constructed to have the same number of rows and columns so a NIST step field will usually correspond to a square matrix A . However, for the sake of generality, we will establish metrics that can be applied to any rectangular matrix.

IV. TERRAIN COVERABILITY

As mentioned in the previous section, the coverability of a certain region of terrain is defined to be some measure of the difficulty a robot would encounter if it were to move over the entire region. Since the robot must cover all the terrain, some sort of modified roughness parameter for the surface of the terrain should also serve as a relatively good estimate for coverability. In this section, we will introduce two different modified roughness parameters, each with its own strengths and weaknesses, that should serve as good metrics for coverability.

A. Modified Average Roughness as a Metric for Coverability

The most common parameter used to quantify the roughness of a surface is the average roughness, denoted by R_a . Originally, average roughness was used for two-dimensional, stylus-type profiling applications so average roughness is commonly defined by putting $R_a = \frac{1}{b-a} \int_a^b |\phi(x)| dx$ where the profile runs from $x = a$ to $x = b$ and $\phi(x)$ denotes the height of the profile relative to some best fitting line. For many applications this best fitting line is taken to be the horizontal mean line, i.e., the horizontal line with y -intercept $\bar{y} = \frac{1}{b-a} \int_a^b y(x) dx$ where $y(x)$ is the height of the profile at the point x . When this is the case, the formula for average roughness becomes $R_a = \frac{1}{b-a} \int_a^b |y(x) - \bar{y}| dx$.

It is not difficult to construct a three-dimensional definition of average roughness that is analogous to the two-dimensional definition that we have just described. Let S be a surface and

let $\phi(x, y)$ denote the height of the surface S relative to a best fitting plane, cylinder, sphere, or other smooth surface Ω . We then define the average roughness, R_a , by writing $R_a = \frac{1}{Area(\Omega)} \int \int_{\Omega} |\phi(x, y)| dx dy$. As in the two-dimensional case, this best fitting surface is sometimes taken to be a horizontal plane, depending on the nature of the surface S and the application that is being considered.

When dealing with step fields, we shall always assume that the best fitting smooth surface Ω is a horizontal plane since step fields are meant to represent obstacles occurring on flat ground². This assumption means that the formula for average roughness reduces to $R_a = \frac{1}{Area(\mathfrak{R})} \int \int_{\mathfrak{R}} |z(x, y) - \bar{z}| dx dy$ where \mathfrak{R} is the rectangular base of the step field, $z(x, y)$ is the height of the step field at the point (x, y) , and \bar{z} is the average height given by $\bar{z} = \frac{1}{Area(\mathfrak{R})} \int \int_{\mathfrak{R}} z(x, y) dx dy$. In fact, since we are considering our rough terrain to be a step field, we can simplify this formula much further. Let SF_A be an $m \times n$ step field with associated matrix A and let \mathfrak{R} denote the base of SF_A . If $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$ then we obtain

$$R_a = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \left| a_{ij} - \left(\frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n a_{kl} \right) \right| \quad (1)$$

Now, we must determine how we should alter the average roughness of a step field in order to have it more accurately reflect coverability. First of all, total roughness is going to be a much better estimate for coverability than average roughness since it will obviously be harder for a robot to cover a large patch of rough terrain than it would be to cover a smaller one of equal roughness. We will denote total roughness by TR so that we can write $TR = \sum_{i=1}^m \sum_{j=1}^n \left| a_{ij} - \left(\frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n a_{kl} \right) \right|$. Furthermore, we need to somehow take into account the dimensions of the robot covering the step field. For example, it should generally be easier for a larger robot to move over a post of height h than it would be for a smaller robot. Perhaps the most relevant dimension of the robot attempting to cover a step field is its wheel diameter if it is a wheeled vehicle or its track height if it is a tracked vehicle. Thus, letting d be the wheel diameter or track height of the robot in question, we consider the quantity $TR_d = \sum_{i=1}^m \sum_{j=1}^n \left| \frac{a_{ij} - \left(\frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n a_{kl} \right)}{d} \right|$. Finally, we do not expect the difficulty a robot would have moving over a step field to scale linearly in height. For example, it should be harder for a robot to move over one very tall post of height h than it would be to move over two smaller posts, each of size $\frac{h}{2}$. This leads us to define our first coverability parameter for a step field, which we shall denote by Cvr_1 , to be

$$Cvr_1 = \sum_{i=1}^m \sum_{j=1}^n \left| \frac{a_{ij} - \left(\frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n a_{kl} \right)}{d} \right|^{p_1} \quad (2)$$

²The main reason that we require Ω to be horizontal is that if we allowed Ω to be any arbitrary plane then an inclined plane lying on flat ground, i.e., a ramp, would have $R_a = 0$. This is undesirable when dealing with robot mobility since ramps are clearly more difficult for robots to traverse than flat ground, especially if the ramp is very steep.

where $p_1 > 1$ can be chosen appropriately for different robots and different applications.

B. Another Modified Roughness Parameter as a Metric for Coverability

For reasons that will be discussed a little later in this paper, average roughness and the coverability measure Cvr_1 that we derived from it have some inherent shortcomings. Hence, it is worth coming up with another coverability metric that is obtained from a different roughness parameter. It is worth noting that while average roughness and, in turn, Cvr_1 can be defined for an arbitrary surface S , the roughness parameter that we construct here only really makes sense for a surface that has been discretized in some way, as is the case when dealing with a step field³.

As mentioned previously, roughness is a measure of the small-scale height variations of a surface so it makes sense to consider a roughness parameter that is basically the sum of all of the height changes. In the case of a step field, there is a potential height change between any two neighboring posts. However, we must define precisely what we mean by two neighboring posts. There are two reasonable definitions that we could consider:

- Any given post has four neighbors, namely the posts directly above and below it and the posts directly to the left and the right of it. If a post is on the perimeter of the array then we consider its exterior neighbors to have a height of zero.
- Any given post has eight neighbors, namely the four neighbors listed above and the four posts that are located on its diagonals. Once again, posts on the perimeter of the array are considered to have exterior neighbors with height equal to zero.

Since a robot should effectively have a full 360 degree range of motion, it is better to use the second definition so we assume that each post has eight neighbors. Thus, for an $m \times n$ step field SF_A with associated matrix $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$ we can define a new total roughness parameter

$$\begin{aligned} \tilde{T}R &= 3 \sum_{k=1}^m (|a_{k1}| + |a_{kn}|) \\ &+ 3 \sum_{k=1}^n (|a_{1k}| + |a_{mk}|) \\ &- |a_{11}| - |a_{1n}| \\ &- |a_{1n}| - |a_{m1}| \\ &+ \sum_{i=1}^m \sum_{j=1}^{n-1} |a_{ij} - a_{i(j+1)}| \\ &+ \sum_{j=1}^n \sum_{i=1}^{m-1} |a_{ij} - a_{(i+1)j}| \end{aligned} \quad (3)$$

³Another case in which this parameter makes sense is when we have a triangulated surface.

$$\begin{aligned} &+ \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} |a_{ij} - a_{(i+1)(j+1)}| \\ &+ \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} |a_{i(j+1)} - a_{(i+1)j}| \end{aligned}$$

that is obtained by adding up all the height changes.

As before, we take into account the wheel diameter or track height d of the robot and we take into account the fact that coverability should not scale linearly with height in order to define another coverability parameter

$$\begin{aligned} Cvr_2 &= 3 \sum_{k=1}^m \left(\left| \frac{a_{k1}}{d} \right|^{p_2} + \left| \frac{a_{kn}}{d} \right|^{p_2} \right) \\ &+ 3 \sum_{k=1}^n \left(\left| \frac{a_{1k}}{d} \right|^{p_2} + \left| \frac{a_{mk}}{d} \right|^{p_2} \right) \\ &- \left| \frac{a_{11}}{d} \right|^{p_2} - \left| \frac{a_{1n}}{d} \right|^{p_2} \\ &- \left| \frac{a_{m1}}{d} \right|^{p_2} - \left| \frac{a_{mn}}{d} \right|^{p_2} \\ &+ \sum_{i=1}^m \sum_{j=1}^{n-1} \left| \frac{a_{ij} - a_{i(j+1)}}{d} \right|^{p_2} \\ &+ \sum_{j=1}^n \sum_{i=1}^{m-1} \left| \frac{a_{ij} - a_{(i+1)j}}{d} \right|^{p_2} \\ &+ \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \left| \frac{a_{ij} - a_{(i+1)(j+1)}}{d} \right|^{p_2} \\ &+ \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \left| \frac{a_{i(j+1)} - a_{(i+1)j}}{d} \right|^{p_2} \end{aligned} \quad (4)$$

where $p_2 > 1$ can be chosen appropriately for different robots and different applications.

C. Strengths and Weaknesses of Cvr_1 and Cvr_2

Neither Cvr_1 nor Cvr_2 serves as a perfect metric for the coverability of rough terrain and both have their own relative strengths and weaknesses. Here, we will discuss the advantages and shortcomings of both of these parameters.

The parameter Cvr_1 is closely related to R_a so it will have many of the same properties as average roughness. One nice quality of Cvr_1 is that it can be calculated for any surface S , even if the surface S has not been discretized. Furthermore, Cvr_1 will scale properly with partitions of the surface. For example, suppose that we take a step field SF_A and partition each post into four posts by cutting the dimensions of the base of each post in half in order to obtain a new step field $SF_{A'}$. It is not hard to see that the Cvr_1 values will be the same for both SF_A and $SF_{A'}$. This is good since SF_A and $SF_{A'}$ are effectively the same step field, at least as far as robot mobility is concerned. However, Cvr_1 has one large disadvantage in that it does not take into account the placement of the peaks

and valleys relative to each other⁴. For example consider step fields SF_A and SF_B with associated matrices

$$A = \begin{pmatrix} 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 9 & 9 & 0 & 0 & 9 & 9 \\ 9 & 9 & 0 & 0 & 9 & 9 \\ 9 & 9 & 0 & 0 & 9 & 9 \\ 0 & 0 & 9 & 9 & 0 & 0 \\ 0 & 0 & 9 & 9 & 0 & 0 \\ 0 & 0 & 9 & 9 & 0 & 0 \end{pmatrix}.$$

For most practical purposes we would consider SF_B to be rougher and more difficult to cover than SF_A , even though both step fields have the same R_a value and, in turn, the same Cvr_1 value.

Now, consider Cvr_2 . Unlike Cvr_1 , the parameter Cvr_2 has the undesirable properties that it can only be expressed for surfaces that are discretized in some way and that it does not scale properly with partitions of the surface. However, the main advantage that it has over Cvr_1 is that it does take into account the relative placement of the peaks and the valleys of the surface. For instance, if SF_A and SF_B are the two step fields with associated matrices A and B defined above then the Cvr_2 value for SF_B will be higher than the Cvr_2 value for SF_A .

It is worth noting one more key difference between Cvr_1 and Cvr_2 . Cvr_2 assumes that the ground surrounding the step field is flat and that the intersection of this flat ground with the rough step field may cause the robot some difficulty. In other words, Cvr_2 takes into account the problems that the robot may have when traveling along the perimeter of the step field. This is desirable, so long as the robot is expected to be affected by the outer perimeter of the field, as it clearly would be if it were entering or exiting the array of posts. On the other hand, Cvr_1 ignores the ground surrounding the step field so it does not consider the difficulty a robot may encounter on its outermost perimeter. As a result, Cvr_1 is more appropriate for a robot that starts off on the step field that it wishes to cover and can avoid any interaction with the outer edges of the posts along the perimeter. It would be quite easy to modify either Cvr_1 or Cvr_2 in order to ensure that they both do or both do not take into account the terrain surrounding the step field. Perhaps the easiest way to cause Cvr_1 to reflect the terrain surrounding the step field would be to augment the $m \times n$ matrix A by surrounding it by zeroes in order to obtain a new $(m+2) \times (n+2)$ matrix A' and then use Equation (2)

⁴If we did not require the best fitting plane Ω to be horizontal then Cvr_1 would probably reflect peak/valley placement a bit better. However, it would still not be perfect and this would introduce other problems so we maintain the requirement that Ω be horizontal.

on A' . For example, the matrix $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ would become

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 3 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Similarly, it is quite easy to make sure that Cvr_2 does not take the terrain surrounding the step field into account by simply removing the first two lines of equation 4, i.e., removing the terms $3 \sum_{k=1}^m (|\frac{a_{k1}}{d}|^{p_2} + |\frac{a_{kn}}{d}|^{p_2}) + 3 \sum_{k=1}^n (|\frac{a_{1k}}{d}|^{p_2} + |\frac{a_{mk}}{d}|^{p_2})$ and $-|\frac{a_{11}}{d}|^{p_2} - |\frac{a_{1n}}{d}|^{p_2} - |\frac{a_{m1}}{d}|^{p_2} - |\frac{a_{mn}}{d}|^{p_2}$. However, we choose to leave Cvr_1 not representing the surrounding terrain and Cvr_2 representing the surrounding terrain in order to emphasize the fact that we may want to include or not include the terrain surrounding the step field in our model, depending on the application.

In summary, neither Cvr_1 nor Cvr_2 perfectly reflects rough terrain coverability. As a result, both metrics may prove to be useful in different circumstances so we shall use them both to represent the coverability of step fields.

V. TERRAIN CROSSABILITY

In addition to the ability to cover a patch of rough terrain, it is important for urban search and rescue robots to be able to move directly from some given point of that terrain to some other given point, i.e., to be able to cross the terrain in some sense. In this section we will work on developing a metric for terrain crossability that will take as inputs a topographical map of the terrain, the start and finish locations, and certain robot dimensions. It is worth noting that in the model that we develop for a robot crossing a patch of rough terrain, we focus more on keeping the model general enough to apply to different types of robots than we do on making it very accurate for some fixed robot. There is obviously going to be a large trade-off between accuracy for specific robot geometries and the generality and simplicity of the model and we choose to err on the side of generality.

A. Why Coverability and Crossability Require Different Metrics

While modified roughness parameters serve as reasonable measures for the coverability of rough terrain, it is quite easy to see that the relationship between roughness and crossability is not so simple. Consider a step field SF_A with associated matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This step field can have arbitrarily high roughness parameters and arbitrarily high values for Cvr_1 and Cvr_2 by taking a , the height of the post in the center of the array, large enough. However, for many robots it will be quite easy to cross this step field by traveling around the center post. Thus, we see

that roughness and coverability do not necessarily correspond to crossability.

A better method for measuring the crossability of some region of rough terrain from point p to point q would be to measure the difficulty a robot would encounter trying to maneuver the least difficult path connecting p to q . However, in order to make all of this precise, we need to define what we mean by a path in a region of rough terrain and we need to define some cost function that can accurately reflect the difficulty of a path. Once we do this, the problem basically reduces to finding a least cost path through a graph, which is a well-studied problem and can be solved using path-planning algorithms such as A^* .

B. Defining a Path Through Rough Terrain

If we wish to find a least cost path through a region of rough terrain, we must first define what constitutes a path through the terrain. We begin by assuming that the state of the robot at any time can be described by giving a point in \mathbb{R}^2 representing the location of the center of mass of the robot in the xy -plane and a real number in the interval $[0, 2\pi)$ representing the direction that the robot is facing.

Next, we discretize the surface of the terrain in such a way that, without loss of generality, we can consider the region of rough terrain to be a step field⁵. This allows us to make the state space of the robot discrete as well by assuming that the xy -coordinates of the robot's center of mass always lie at the center of some post and by assuming that the robot is always facing in one of eight directions: north, northeast, east, southeast, south, southwest, west, or northwest. In accordance with standard mathematical convention, we will let east correspond to an angle of 0 radians, northeast correspond to an angle of $\frac{\pi}{4}$ radians, and so on. Hence, the state of the robot can always be expressed by a 3-tuple where the first two entries are the row and column of a post in the step field, respectively, and the third entry is a number in the set $\{\frac{t\pi}{4} : t = 0, 1, \dots, 7\}$.

Now, for each robot state we need to define the set of states to which the robot can move. Suppose that the robot starts off in state $(i, j, \frac{t\pi}{4})$. For, our purposes, it makes sense to assume that the robot can move to one of three new states:

- $(i, j, \frac{t'\pi}{4})$ where $t' = t - 1 \pmod{8}$. This corresponds to the robot turning 45 degrees in the clockwise direction. We will call this a move of type one.
- $(i, j, \frac{t''\pi}{4})$ where $t'' = t + 1 \pmod{8}$. This corresponds to the robot turning 45 degrees in the counterclockwise direction. We will call this a move of type two.
- $(i + f_1(t), j + f_2(t), \frac{t\pi}{4})$ where $f_1(0) = 0, f_2(0) = 1, f_1(1) = -1, f_2(1) = 1, f_1(2) = -1, f_2(2) = 0, f_1(3) = -1, f_2(3) = -1, f_1(4) = 0, f_2(4) = -1, f_1(5) = 1, f_2(5) = -1, f_1(6) = 1, f_2(6) = 0, f_1(7) = 1,$ and

⁵Once again, we could instead triangulate the terrain surface in order to make everything discrete. However, since we are focusing on step fields in this paper, we choose a discretization that allows us to reduce the problem to the case of a robot on a step field.

$f_2(7) = 1$. This corresponds to the robot moving forward to the next block. We will call this a move of type three.

With just these three options it is possible for a robot to get from any state to any other state in some finite number of moves⁶.

Finally, we are ready to define a path through a step field. An ordered set of 3-tuples of the form $(i, j, \frac{t\pi}{4})$ described above such that the first 3-tuple represents the specified starting state p , the last 3-tuple represents the specified finishing state q , and each 3-tuple can be obtained from the previous one by one of the three valid robot moves described in the preceding paragraph is said to be a path from p to q .

C. Constructing the Cost Function

As mentioned before, the crossability from state p to state q of a region of rough terrain should be the cost of the least cost path connecting p to q , where the cost of a path is the difficulty that a robot would encounter trying to follow it. Thus, in order to calculate crossability, we need to construct this cost function. In other words, we need to define the cost of performing moves of type one, two, and three.

As done by Iagnemma and Dubowsky, we assume that the three main aspects of the path that affect robot mobility are the roughness of the terrain encountered along that path, the amount of turning required to follow that path, and the length of that path [1]. It is clear that a robot will have more difficulty traveling along a path that takes it over very rough terrain than it will have traveling along a similar path over perfectly flat terrain. Furthermore, we expect that it should be more difficult for a robot to follow a path that requires a lot of turning, especially if that turning occurs over rough terrain, than it would be for the robot to follow a straight path. Finally, it makes sense that, *ceteris paribus*, it is easier for a robot to travel a shorter path than a longer one.

Also, we assume that there are two major properties of the robot that will affect the difficulty it encounters along the path. First, we expect that the robot's wheel diameter or track height, which we again denote by d , will have a relatively large effect on its ability to maneuver a given path. This makes sense since robots with larger wheels or tracks should have less difficulty going over a bump of size h or traveling a distance of length l than robots with smaller wheels or tracks. Next, we expect that the dimensions of the base of the robot will be relevant, where the base of the robot is defined to be the convex hull of the wheels or tracks when the robot is placed on flat ground. These dimensions will determine the region of terrain about the center of mass that should be considered when accounting for the rough terrain encountered along the path.

In order to define the cost function, it is useful to first define a bit of simplified notation. Suppose that the current robot state is $(i, j, \frac{t\pi}{4})$. We let $F_1(i, j, \frac{t\pi}{4})$ be the set of posts in the step field that come in contact with the base of the robot as the robot performs a move of type one. Similarly,

⁶Here, we assume that the robot can turn without moving its center of mass. This assumption is reasonable for the case of a skid steered robot but not particularly accurate for other steering designs.

let $F_2(i, j, \frac{t\pi}{4})$ and $F_3(i, j, \frac{t\pi}{4})$ be the sets of posts that the robot's base contacts as it performs a move of type two or three, respectively. Thus, $|F_m|$ denotes the number of posts that are in the set F_m for $m = 1, 2, 3$.

Now, we are ready to define the costs of performing moves of type one, two and three. Again, suppose that the robot is presently in state $(i, j, \frac{t\pi}{4})$. Let $cost_m(i, j, \frac{t\pi}{4})$, where $m \in \{1, 2, 3\}$, denote the cost of making a move of type m . We then say that

$$cost_m = \alpha_m \sum_{(k,l) \in F_m} \left| \frac{a_{kl} - \frac{1}{|F_m|} \sum_{(r,s) \in F_m} a_{rs}}{d} \right|^{\beta_m} + \frac{\gamma_m}{d} \quad (5)$$

where $\alpha_m > 0$, $\beta_m > 1$, and $\gamma_m > 0$ can all be chosen for different robots and different situations. Note that the symmetry between moves of type one and type two tells us that we should require $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$, and $\gamma_1 = \gamma_2$. The cost of any given path can now be computed by just summing the costs of all of the individual moves that are required to follow that path.

By defining cost in this manner, we take into account all of the primary factors that we said should affect robot mobility including roughness, turning, path length, and robot dimensions. In particular, the

$\alpha_m \sum_{(k,l) \in F_m} \left| \frac{a_{kl} - \frac{1}{|F_m|} \sum_{(r,s) \in F_m} a_{rs}}{d} \right|^{\beta_m}$ terms ensure that paths requiring the robot to move over large amounts of rough terrain and paths involving lots of turning on rough terrain will be considered more difficult than similar paths occurring on flat ground⁷. Additionally, the $\frac{\gamma_m}{d}$ terms imply that paths that are longer and paths that require a lot of turning will cost more than shorter and straighter paths over the same terrain.

D. Calculating Crossability by Finding the Least Cost Path

We have already determined that a reasonable estimate for the crossability of a step field from state p to state q is the cost of the least cost path connecting p to q . Hence, we formally define the crossability from state p to state q , denoted by $Crs_{p \rightarrow q}$, to be the cost of the least cost path going from state p to state q . Now that we have defined what constitutes a path over rough terrain and described how to calculate the cost of following such a path, the only thing that remains to be done is to explain the method in which we find the least cost path.

First, we construct the vertex set V of a digraph $G = (V, E)$ by letting each possible robot state be a vertex in a graph. Next, we construct the directed edge set E by saying that there is an edge directed from a vertex v_1 to another vertex v_2 if and only if the robot can get to the state corresponding to v_2 from the state corresponding to v_1 by performing a robot move of type one, two, or three. Finally, to each directed edge we associate the cost of performing the robot move that corresponds to that edge. In this way, we have reduced the problem to finding a least cost path through a digraph, which is a problem that has

⁷We use a modified R_α parameter to represent roughness. We could have instead used a modified TR parameter, where TR is as described in section IV.

been studied in great detail. An existing algorithm, such as A^* , can be used to find the cost of the least cost path through such a digraph and this number can then be used to represent the crossability of the terrain.

VI. NUMERICAL RESULTS

In this section, we compute the coverability parameters Cvr_1 and Cvr_2 for several step fields produced by a NIST random step field generator⁸. Also, we will discuss the results that are obtained and see if the coverability values agree with our expectations and intuition. We have not yet performed any crossability calculations, but this is something that we would like to do in the near future.

The four step fields for which we compute coverability parameters can be seen in Figure 2. The digits 0, 1, 2, 3, and 4 represent posts of height $1\frac{3}{4}''$, $3\frac{1}{2}''$, $7''$, $10\frac{1}{2}''$, and $14''$, respectively. Furthermore, the step fields are surrounded by borders of height $3\frac{1}{2}''$, which we treat as two extra rows and columns for each step field so that all of the associated matrices for these step fields are 13×13 . Note that the four step fields in Figure 2 are representative of the four different random step field layouts that NIST generates. The layout of SF_1 is known as the flat box layout, where the adjective flat describes the fact that there are no posts of size 4 and only four posts of size 3 and the terms box refers to the fact that those posts of size 3 make up a square box. Similarly, SF_2 is known as the flat cross layout since the four posts of size 3 form a cross. SF_3 is called the diagonal layout since there is a hill made up of posts of size 4 running across the diagonal of the field. Finally, the layout depicted in SF_4 is the hill layout and it is characterized by the column of posts of size 4 located in the middle of the step field.

The results of the calculations for the step fields shown in Figure 2 can be found in Table I (rounded to the nearest thousandth). For these calculations, we used $p_1 = p_2 = 2$ and assumed that $d = 7''$, i.e., the wheel diameter corresponds to about two post widths. The coverability values that were obtained agree roughly with our expectations in the sense that the diagonal step field (SF_3) and the hill step field (SF_4) yield significantly larger coverability numbers than the two flat step fields (SF_1 and SF_2) do. It is worth noting that the two coverability metrics Cvr_1 and Cvr_2 produce different relative orderings of the coverabilities of these four step fields with Cvr_1 implying that SF_4 is more difficult to cover than SF_3 and Cvr_2 implying the exact opposite. This reiterates the fact that Cvr_1 and Cvr_2 are very different parameters, each with its own advantages and disadvantages.

While it is useful to see what coverability values we obtain for a few actual step fields, these numerical results do not accurately test the coverability metrics and more detailed experiments should be conducted to suit this purpose. For example, one could run an experiment requiring subjects to

⁸A NIST random step field generator creates an eleven by eleven matrix where the entries of the matrix all lie in the set $\{0, 1, 2, 3, 4\}$ and the matrix is subject to certain rules. For example, the height difference between two horizontal or vertical neighbors can not exceed 2.

drive various robots over several step fields with the goal of covering each field and then have the subject rank the step fields in terms of difficulty to cover. Then, the relative difficulties of covering the step fields as ranked by the subjects can be compared to the relative coverability difficulties produced by the metrics Cvr_1 and Cvr_2 .

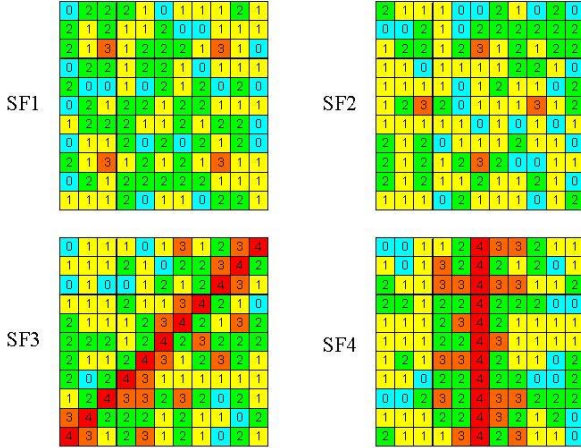


Fig. 2. Step Fields

TABLE I
COVERABILITY VALUES FOR SF_1 , SF_2 , SF_3 , AND SF_4

	Cvr_1	Cvr_2
SF_1	14.013	143.875
SF_2	12.764	137.625
SF_3	37.246	227.750
SF_4	37.956	191.500

VII. CONCLUSIONS

Urban search and rescue robots are likely to encounter many difficult terrain conditions as they perform their required tasks. In particular, they will have to be able to move over and across terrain that displays a lot of small scale height variation, which we refer to as rough terrain. Thus, it is important to have metrics that describe the difficulty a robot would have attempting to traverse different regions of rough terrain. To this end, we have developed three different metrics for the traversability of rough terrain that take as inputs a topographical map of the terrain and some minimal information about the robot's dimensions. These metrics are derived for the case when the region of rough terrain under investigation is a step field, which is not an overly restrictive assumption since we can always discretize the terrain in order to give it a step field structure.

Two of these metrics, which we have denoted by Cvr_1 and Cvr_2 , tell us how difficult it would be for a robot to cover, i.e., move over ever part of, a step field. They are effectively modified roughness parameters that are scaled by the wheel diameter or track height of the robot in question in order to make them dimensionless and in order to account for the effect of the size of the robot on traversability. Both Cvr_1 and Cvr_2

have their relative strengths and weaknesses so we use both of these quantities to describe coverability.

The third metric describes how difficult it would be for a robot to move from some point on a step field to some other point. In order to derive this metric, we discretized the state space of the robot and constructed a cost function that approximates how difficult it is for the robot to move from one state to another. Then, we defined the crossability from state p to state q , denoted by $Crs_{p \rightarrow q}$, to be the cost of the least cost path connecting p to q so that $Crs_{p \rightarrow q}$ describes the difficulty a robot would have moving from state p to state q .

Finally, we performed some computations and determined Cvr_1 and Cvr_2 values for four different step fields produced by a NIST random step field generator. The numerical results make intuitive sense but more detailed experiments should be performed in order to better test these coverability metrics. No crossability values have been calculated at this point, but this is an area in which we would like to devote more attention sometime in the near future.

ACKNOWLEDGMENTS

The first author would like to thank the United States Department of Homeland Security. His research towards this paper was performed on appointment as a U.S. Department of Homeland Security (DHS) Fellow from the Courant Institute of Mathematical Sciences at New York University under the DHS Scholarship and Fellowship Program, a program administered by the Oak Ridge Institute for Science and Education (ORISE) for DHS through an interagency agreement with the U.S. Department of Energy (DOE). ORISE is managed by Oak Ridge Associated Universities under DOE contract number DE-AC05-00OR22750. All opinions expressed in this paper are the authors' and do not necessarily reflect the policies and views of DHS, DOE, or ORISE.

In addition, he wishes to thank NIST for hosting his summer research and Elena Messina for the excellent guidance and mentoring that she provided.

REFERENCES

- [1] Iagnemma K. and Dubowsky S., *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Applications to Planetary Rovers*. Springer-Verlag, Berlin
- [2] Shiller Z., *Obstacle Traversal for Space Exploration*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation
- [3] Iagnemma K., Rzepniewski A., Dubowsky S., Huntsberger T., Pirjanian P. and Schenker P., *Mobile Robot Kinematic Reconfigurability for Rough Terrain*, Proceedings of the 2000 SPIE symposium on sensor fusion and decentralized control in robotic systems., Volume 4196
- [4] Latombe J.C., *Robot Motion Planning*. Kluwer Academic Publishers, Norwell
- [5] Seraji H., *Traversability Index: A New Concept for Planetary Rovers*, Proceedings of the 1999 IEEE International Conference on Robotics and Automation
- [6] Howard A., Seraji H., Tunstel E., *A Rule-Based Fuzzy Traversability Index for Mobile Robot Navigation*, Proceedings of the 2001 IEEE International Conference on Robotics and Automation
- [7] Bekker G., *Introduction to Terrain-Vehicle Systems*. University of Michigan Press, Ann Arbor
- [8] Cohen D., *Glossary of Surface Texture Parameters*, www.michmet.com/Michigan
- [9] Brown S., *A Note on the Description of Surface Roughness Using Fractal Dimension*, Geophysical Research Letters, Volume 14, Issue 11, P. 1095-1098
- [10] <http://www.camnl.wr.usgs.gov/sws/fieldmethods/Indirects/nvalues/>

Performance Evaluation of Integrated Vehicle-Based Safety Systems

Jack J. Ference

National Highway Traffic Safety
Administration
Washington, DC 20590
jack.ference@dot.gov

Sandor Szabo

National Institute of Standards
and Technology
Gaithersburg, MD 20899
sandor.szabo@nist.gov

Wassim G. Najm

Volpe National Transportation
Systems Center
Cambridge, MA 02142
wassim.najm@volpe.dot.gov

Abstract—¹This paper describes a program to develop and test an integrated crash warning system that addresses rear-end, lane change, and roadway departure crashes for passenger cars and heavy commercial trucks. One of the goals of this program is to facilitate the deployment of integrated crash warning systems by creating performance specifications and objective test procedures, and estimating potential safety benefits for integrated safety systems. In support of this goal, equations for the safety benefits estimation methodology are introduced and test scenarios derived from national crash database statistics are delineated. The approach, performance metrics and independent measurement system used to conduct objective tests are also discussed.

Keywords: *warning system, crash prevention, objective test, performance measurement*

I. INTRODUCTION

Rear-end, lane change, and roadway departure crashes account for approximately 3.6 million police-reported crashes each year on U.S. roadways. These three crash types result in about 27,500 of the Nation's 42,000 annual traffic fatalities and contribute to a considerable economic loss due to injuries, property damage, and decreased productivity. Studies conducted by the U.S. Department of Transportation (U.S. DOT) indicate that a substantial percentage of the 3.6 million target crashes could be prevented annually by widespread deployment of integrated crash warning systems that would warn drivers of imminent crash situations and prompt them to take corrective action [1 – 3].

In November of 2005, the U.S. DOT entered into a cooperative research agreement with an industry team to develop and test an integrated, vehicle-based, crash warning system that addresses rear-end, lane change and roadway departure crashes [4]. The four-year, two-phase program that will be carried out under this agreement is known as the Integrated Vehicle-Based Safety System (IVBSS) program.

During Phase I of the program, individual crash warning subsystems will be enhanced; the integrated system will be designed, and component subsystems will be combined with a driver-vehicle interface (DVI) into a prototype vehicle. The prototype vehicle will undergo a series of tests aimed at verifying that the integrated system meets the performance requirements and is safe for use by unescorted volunteer drivers for extended periods.

In Phase II, the deployment fleets will be constructed; volunteer drivers and truck fleets will be recruited, and the field operational test (FOT) will be implemented. Volunteer drivers and employees of the truck fleets will use the project vehicles as their own personal vehicle to drive as they normally would for a period of approximately one month. The field test will last approximately one year. Data will be collected on the driver/vehicle/system performance and the driving environment using on-board data acquisition systems (DASs).

Objective tests will be developed to verify the performance of the integrated system installed on the fleet of passenger cars and heavy commercial trucks during Phase I. These verification tests consist of controlled scenarios and procedures, typically conducted on test tracks or pre-defined routes on public roads. Results from these tests will help refine the design and construction of the prototype vehicles, and ensure deployment readiness for the field test.

As part of the IVBSS program, an independent evaluation will be performed to estimate potential safety benefits, determine driver and truck fleet acceptance, and characterize the capability and performance of the integrated system used in the field test. In addition to numerical and video data collected from the on-board DASs, subjective data will be gathered from field participants through surveys and focus groups.

II. BENEFITS ESTIMATION

IVBSS technologies have the potential to reduce the number of motor vehicle crashes and severity of crash-related injury. Prior to wide-scale deployment in the U.S. vehicle fleet, these safety benefits can be estimated using data collected from field tests of deployment-ready systems. Safety is ideally measured from actual crash data; however, such data are rare or non-existent during the conduct of field tests since a wide

¹ No approval or endorsement of any commercial product by the National Institute of Standards and Technology or the National Highway Traffic Safety Administration is intended or implied. This publication was prepared by United States Government employees as part of their official duties and is not subject to copyright.

exposure is required to ensure adequate crash data. The scope of field tests is typically limited to a few instrumented vehicles driven by volunteer subjects for a relatively short period. A methodology has been formulated to predict safety benefits utilizing non-crash, driver/vehicle/system performance data collected from encounters with various driving conflicts during the FOT [5].

Safety benefits are measured by estimating the number of crashes that might be avoided and the total harm that might be reduced due to full deployment of integrated systems. These two measures of safety benefits can be translated into monetary savings in terms of crash economic cost [6]. The number of crashes avoided is used to project monetary savings in crash economic cost due to property-damage only. Savings in injury-related economic costs are estimated by multiplying the total harm reduction factor with the cost of all injuries. The total harm reduction factor encompasses reductions in injuries due to crashes avoided and lower-severity of injuries from crashes not avoided.

The annual number of target crashes that might be avoided with full deployment of an integrated system, N_a , is:

$$N_a = \sum_{i=1}^n N_{wo}(S_i) \times E(S_i) \quad (1)$$

n ≡ Number of applicable pre-crash scenarios, S_i

$N_{wo}(S_i)$ ≡ Annual number of target crashes preceded by S_i prior to full deployment

$E(S_i)$ ≡ System effectiveness in avoiding target crashes preceded by S_i

Target crashes consist of vehicular dynamic scenarios and crash contributing factors that the system is designed to address. Pre-crash scenarios refer to vehicle orientations, dynamics, and movements that happen immediately prior to a target crash, as well as the critical event that makes the crash imminent [7]. $N_{wo}(S_i)$ can be obtained from national crash databases such as the National Automotive Sampling System/General Estimates System (GES) and Crashworthiness Data System (CDS) databases. $E(S_i)$ is expressed as:

$$E(S_i) = 1 - \frac{P_w(C|S_i)}{P_{wo}(C|S_i)} \times \frac{P_w(S_i)}{P_{wo}(S_i)} \quad (2)$$

$P_w(C/S_i)$ ≡ Probability of a crash with IVBSS assistance given that S_i has been encountered

$P_{wo}(C/S_i)$ ≡ Probability of a crash without IVBSS assistance given that S_i has been encountered

$P_w(S_i)$ ≡ Probability of an S_i encounter with IVBSS assistance

$P_{wo}(S_i)$ ≡ Probability of an S_i encounter without IVBSS

The ratios $\frac{P_w(C|S_i)}{P_{wo}(C|S_i)}$ and $\frac{P_w(S_i)}{P_{wo}(S_i)}$ are known

respectively as the crash prevention ratio and scenario exposure ratio. The prevention ratio can be obtained from computer simulations of kinematical models with representative random variables (e.g., Monte Carlo simulation), using naturalistic driving data from the FOT and experimental data from the system design phase. The exposure ratio can be obtained from FOT data by counting the

number of conflicts encountered and normalizing by the number of vehicle miles traveled with and without the integrated warning system engaged.

The methodology described above depends on the identification of driving conflicts from driving situations recorded during the conduct of field tests. These conflicts are defined in a similar way as pre-crash scenarios. It should be noted that these driving conflicts, S_i , must be quantified [8]. Some encounters with driving conflicts might be of benign nature, which typically occur in normal driving conditions where immediate and intense driver response to prevent a potential collision may not be required. Thus, boundaries need to be established between benign encounters with driving conflicts (i.e., normal driving situations) and safety-critical encounters with driving conflicts (i.e., near-crashes). Such boundary quantification allows accurate and consistent data reduction by retaining pertinent information on encounters with true critical conflicts obtained in FOTs.

A second benefit estimate is the annual reduction in total harm with full system deployment, H_r , which is obtained as follows:

$$H_r = \sum_{i=1}^n H_{wo}(S_i) \times R(S_i) \quad (3)$$

$H_{wo}(S_i)$ ≡ Annual total harm from target crashes preceded by S_i prior to full deployment

$R(S_i)$ ≡ System effectiveness in reducing total harm from target crashes preceded S_i

$H_{wo}(S_i)$ is determined from the following total harm equation:

$$H = \sum_{m=0}^6 w(m) \times O(m) \quad (4)$$

m ≡ Injury severity level

$w(m)$ ≡ Unit cost of injury severity level m

$O(m)$ ≡ Number of occupants with injury severity level m

Injury severity level, m , is based on the Abbreviated Injury Scale (AIS) used by the medical community. Level 0 refers to an uninjured person while level 6 denotes a fatal injury. Levels 1 through 5 indicate respectively a minor, moderate, serious, severe, or critical injury. The U.S. DOT has estimated the unit cost of each injury severity level, $w(m)$, in terms of economic cost based on year 2000 dollar value [6]. $R(S_i)$ in Equation (3) is determined from:

$$R(S_i) = 1 - E'(S_i) \times \frac{\bar{H}_w(S_i)}{\bar{H}_{wo}(S_i)} \quad (5)$$

The variables $E'(S_i)$, $\bar{H}_w(S_i)$ and $\bar{H}_{wo}(S_i)$ are computed from the following equations:

$$E'(S_i) = 1 - E(S_i) \quad (6)$$

$$\frac{\bar{H}_w(S_i)}{\bar{H}_{wo}(S_i)} = \frac{\sum_{k=1}^{\ell} P_w(\Delta v_k | S_i) \times \bar{H}(\Delta v_k)}{\sum_{k=1}^{\ell} P_{wo}(\Delta v_k | S_i) \times \bar{H}(\Delta v_k)} \quad (7)$$

Δv_k ≡ Change in speed in bin k that a vehicle undergoes as a consequence of crashing

$P_{wo}(\Delta v_k/S_i)$ ≡ Probability of Δv_k given that a crash has occurred during an S_i encounter without IVBSS assistance

$\bar{H}(\Delta v_k)$ ≡ Average harm per crash (harm unit) with Δv_k

Equation (7) assumes that vehicle crashworthiness (e.g., crash protection offered by vehicles), distribution of vehicle weights, and vehicle occupancy remain the same with and without IVBSS assistance. Therefore, the reduction of injury severity would occur due to lower closing speeds at impact (smaller Δv) if drivers were assisted by IVBSS technologies. The values of $\bar{H}(\Delta v_k)$ can be derived from national crash databases such as the CDS [5]. The parameters $P_w(\Delta v_k/S_i)$ and $P_{wo}(\Delta v_k/S_i)$ can be obtained from the same process used to estimate $P_{wo}(C/S_i)$ and $P_w(C/S_i)$. For instance, Monte Carlo simulations yield a number of crashes along with vehicle speeds at impact that can then be converted to values of Δv using simple models.

III. TEST SCENARIOS

Test scenarios are based on the most frequent pre-crash scenarios and most prevalent driving conditions at the time of the crash. Individual test scenarios are presented for rear-end, lane change, and roadway departure crashes based on 2003 GES statistics. Moreover, scenarios are suggested for integrated system applications.

A. Rear-End Scenarios

The following four scenarios are proposed as a basis for testing the rear-end crash warning function:

1. Host vehicle (vehicle equipped with an integrated warning system) changes lanes and approaches a stopped lead vehicle.
2. Host vehicle is moving at constant speed and approaches a lead vehicle moving at lower constant speed.
3. Host vehicle is closely following a lead vehicle at constant speed and then lead vehicle suddenly decelerates.
4. Host vehicle is moving at constant speed and approaches a stopped lead vehicle.

These scenarios mainly occur in daylight, clear weather, and on straight and level roadways. The most frequent speed limit is 35 mph.

B. Lane-Change Scenarios

The following four scenarios are proposed as a basis for testing the lane change crash warning function:

1. Host vehicle changes lanes (constant longitudinal speed) to the right and encroaches on another vehicle in the adjacent lane.
2. Host vehicle passes (changing lanes with longitudinal acceleration) to the left and encroaches on another vehicle in the adjacent lane.
3. Host vehicle turns to the left and encroaches on another vehicle in the adjacent lane.

4. Host vehicle drifts (changing lanes with small lateral speed) to the right and encroaches on another vehicle in the adjacent lane.

C. Roadway Departure Scenarios

The following five scenarios are proposed as a basis for testing the roadway departure crash warning function:

1. Host vehicle is going straight and departs road edge to the right.
2. Host vehicle is going straight and departs road edge to the left.
3. Host vehicle is negotiating a curve and departs road edge to the right.
4. Host vehicle is negotiating a curve and loses control due to excessive speed on the curve.
5. Host vehicle is turning left at an intersection and departs road edge to the right.

D. Integrated Scenarios

Using the sets of test scenarios from individual crash types listed above, the following integrated scenarios are suggested:

1. Host vehicle is moving at constant speed and approaches a lead vehicle moving at lower constant speed. Host vehicle then attempts to pass to the left adjacent lane that is occupied by another vehicle.
2. Host vehicle is moving at constant speed and approaches a stopped lead vehicle. Host vehicle then attempts to change lanes to the right adjacent lane that is occupied by another vehicle.
3. Host vehicle drifts and is about to depart to the right adjacent lane that is occupied by another vehicle.
4. Host vehicle drifts and is about to depart to the left adjacent lane that is occupied by another vehicle.
5. Host vehicle is closely following a lead vehicle on a straight road, both driving too fast for the upcoming curve. Lead vehicle then suddenly decelerates.

IV. OBJECTIVE TEST PROGRAM

The U.S. DOT has planned an extensive program for testing the integrated system with the following purposes in mind:

- Verifying warning system performance prior to building a fleet of equipped vehicles and conducting the field test
- Determining how well the integrated system addresses each crash scenario
- Conducting preliminary research for possible safety rating programs to be used by the public for buying safer cars

The majority of the test activities take place during Phase I of the IVBSS program. The program will test the four warning functions: rear-end, road departure, lane change and integrated, on passenger and heavy commercial vehicles, and on test track and on-road environments. Track-based tests focus on correctness and timing performance in controlled, ideal conditions. Road-based tests examine performance in

real-world conditions and primarily focus on measuring false alarm rates.

V. OBJECTIVE TESTS AND PERFORMANCE METRICS

Objective tests should, as much as possible, remove subjective analysis from the evaluation of system performance. The tests strive toward objectivity by:

- Defining metrics for measuring performance
- Conducting tests under controlled conditions
- Measuring conditions and performance variables using an independent measurement system

Metrics are the ruler, or scale, for objectively evaluating performance. Metrics typically consist of equations of several variables which, when evaluated, produce the performance measurement. Values for the variables may come from assumptions about the driver's response, from previous experiments and from measurements taken in real-time during a test run. Objective tests generate data to evaluate the correctness and timing of warnings. The response of the system for a given test is classified true positive (TP), false positive (FP) and false negative (FN) according to criteria listed in Table 1. The functional requirements dictate when the system should and should not issue a warning.

Table 1 Warning classifications.

Functional Requirement:	System shall warn	System shall not warn
System warned	TP	FP
System did not warn	FN	TN

The following equations define various effectiveness metrics used to summarize the warning system response ($\sum TP$ means the sum of all true positive warnings for a particular test or set of tests):

$$\text{True\%} = \frac{\sum TP}{\sum (TP + FN)} \times 100$$

$$\text{False\%} = \frac{\sum FP}{\sum (TP + FP)} \times 100 \quad (8)$$

$$\text{Missed\%} = \frac{\sum FN}{\sum (TP + FN)} \times 100$$

Metrics such as crash prevention boundaries (CPB) are used to determine if a warning provides sufficient time or distance for the driver to react to the warning, and to respond by either braking or steering. A CPB for the forward collision scenario specifies the minimum longitudinal range for a warning:

$$r_w = v_f t_r + \frac{v_f^2 - v_l^2}{-2(a_f - a_l)} \quad (9)$$

Where:

v_f = measured following vehicle forward velocity (m/s)

v_l = measured lead vehicle forward velocity (m/s)

t_r = assumed driver reaction time (s)

a_l = measured lead vehicle acceleration (braking is negative) (m/s²)

a_f = assumed following vehicle acceleration to avoid collision (m/s²)

Similarly for a roadway departure on a straight road, the minimum lateral range for a warning is:

$$r_w = v_{lat} t_r + \frac{v_{lat}^2}{-2a_{lat}} \quad (10)$$

Where:

v_{lat} = measured lateral velocity (positive toward road edge) (m/s)

t_r = assumed driver reaction time (s)

a_{lat} = assumed lateral acceleration to avoid departure (negative away from road edge) (m/s²)

An example application of the CPB metric for evaluating the performance of a road departure crash warning system appears in [9].

VI. PERFORMANCE MEASUREMENT SYSTEM

During system testing, evaluators will use an independent measurement system (IMS) developed by the National Institute of Standards and Technology (NIST) to:

- Support detailed analysis of conditions surrounding a warning or lack of warning
- Provide ground truth reference for measuring system performance
- Provide data and sensor redundancy for test verification purposes

The IMS developed for the roadway departure crash warning system (RDCWS) FOT includes calibrated cameras that enable evaluators to measure ranges to adjacent obstacles and to the road edge at distances up to 4 m [10]. NIST plans to extend the IMS in order to measure range and range-rate to forward-collision obstacles. The minimum requirements for the range measurement system include (desirable capability in parentheses):

- Range out to at least 60 m (100 m)
- 180° (360°) horizontal field of view (FOV) at 0.5° (0.25°) resolution
- 10 Hz (30 Hz) FOV update

A dual-head, laser-range scanner system that meets these requirements is currently being evaluated. The evaluation includes static characterization (stationary sensor and targets) and dynamic characterization (moving sensor and moving targets).

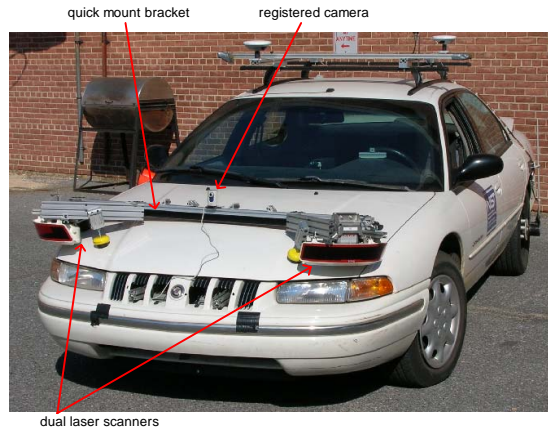


Figure 1 NIST/DOT test bed vehicle with dual-head laser-range scanner mounted on front hood.

VII. SUMMARY

This paper introduced the IVBSS program, a new U.S. DOT safety initiative to build and field test integrated crash warning systems designed to prevent rear-end, lane change, and roadway departure crashes for passenger cars and heavy commercial trucks. The goal of this program is to accelerate the deployment of integrated crash warning technologies by providing government and industry stakeholders' relevant information regarding system performance specifications, objective test procedures, potential safety benefits, and driver acceptance.

In support of the program goals and objectives, a methodology to estimate safety benefits using non-crash, driver/vehicle/system performance data gathered from a field test conducted in a naturalistic driving environment was developed. A set of crash scenarios, which serve multiple activities ranging from system design, objective testing, to safety benefits estimation, were also defined. Objective tests used to ensure that the IVBSS prototype vehicles meet performance requirements and are ready for use by laypersons in the field test were described.

Over the next four years, the IVBSS program will produce integrated system functional requirements, performance specifications, objective test procedures, and a fleet of passenger cars and heavy commercial trucks fitted with IVBSS technologies. In addition, a large database that will characterize driver/vehicle performance on public roads with and without the integrated safety system will be created. This database will be mined to estimate potential safety benefits, driver and truck fleet acceptance, and performance capability and maturity of the technologies. Interim and final program results will be published in public reports that will be available on NHTSA's website, <http://www.nhtsa.dot.gov>.

REFERENCES

- [1] National Highway Traffic Safety Administration, "Preliminary Assessment of Crash Avoidance Systems Benefits", Version II, Chapter 3, NHTSA Benefits Working Group, Washington, DC, December 1996.
- [2] D. Pomerleau and J. Everson, "Run-Off-Road Collision Avoidance Using IVHS Countermeasures – Final Report", DOT HS 809 170, December 1999.
- [3] S. Talmadge, R. Chu, C. Eberhard, K. Jordan, and P. Moffa, "Development of Performance Specifications for Collision Avoidance Systems for Lane Change Crashes", DOT HS 809 414, August 2001.
- [4] <http://www.its.dot.gov/ivbss/index.htm>
- [5] W.G. Najm, M.P. daSilva, and C.J. Wiacek, "Estimation of Crash Injury Severity Reduction for Intelligent Vehicle Safety Systems", Paper No. 2000-01-1354, SAE 2000 World Congress, Detroit, MI, March 2000.
- [6] L. Blincoe, A. Seay, E. Zaloshnja, T. Miller, E. Romano, S. Luchter, and R. Spicer, "The Economic Impact of Motor Vehicle Crashes, 2000". DOT HS 809 446, May 2002.
- [7] W.G. Najm, B. Sen, J.D. Smith, and B.N. Campbell, "Analysis of Light Vehicle Crashes and Pre-Crash Scenarios Based on the 2000 General Estimates System". DOT-VNTSC-NHTSA 02 04, DOT HS 809 573, February 2003.
- [8] D.L. Smith, W.G. Najm, and R.A. Glassco, "The Feasibility of Driver Judgment as a Basis for Crash Avoidance Database Development". Paper No. 02-3695, Transportation Research Record 1784, TRB 2002 Annual Meeting, Washington, D.C., January 2002.
- [9] S. Szabo and B. Wilson, "Application of a Crash Prevention Boundary Metric to a Road Departure Warning System", Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, National Institute of Standards and Technology, Gaithersburg, MD, August 24 - 26, 2004.
- [10] <http://www.isd.mel.nist.gov/documents/szabo/PerMIS04.pdf>
- [10] S. Szabo and R. Norcross, "An Independent Measurement System for Performance Evaluation of Road Departure Crash Warning Systems". National Institute of Standards and Technology Internal Report 7287, January 3, 2006

A Performance Evaluation Laboratory for Automated Threat Detection Technologies

Robert C. Schrag
Information Extraction and Transport, Inc.
1911 N Fort Myer Dr, Suite 600
Arlington, VA 22209 USA
rschrag@iet.com

Abstract—We describe a performance evaluation laboratory (PE Lab) appropriate for exploratory assessment of automated threat detection technologies. The PE Lab’s main components are a dataset generator and a hypothesis scorer. The dataset generator operates over an artificial world in which threat and non-threat individual and group actors exploit infrastructure elements—either productively (in a non-threat mode) or destructively (in a threat mode). The dataset generator creates synthetic transaction- and higher-level evidence reports (used to challenge technologies) and scenario ground truth (used in scoring technologies’ outputs). Threat detection technologies process evidence to hypothesize threat events, groups, individuals, and individual aliases. The scorer compares the technologies’ output hypotheses to ground truth, generalizing traditional information retrieval metrics (precision, recall, F-value, area under curve) to accommodate partial matching over structured hypotheses with weighted attributes.

The generator is parameterized so that problem difficulties may be varied along multiple dimensions (*e.g.*, dataset and population size, signal-to-noise ratio, evidence observability and corruption levels). We describe and illustrate, using a case study, our methodology for constraint-based experiment design and analysis to identify which among varied dataset characteristics most influence a given technology’s performance on a given detection task.

Keywords: *threat detection, performance evaluation laboratory, synthetic data, parameterized data generator, constraint-based experiment design, hypothesis scoring, non-parametric significance analysis*

I. INTRODUCTION

Threat detection by sifting high-volume data streams for indicators has been likened to the problem of recognizing a complete, “threat” needle by selecting from among many haystack-sized piles of threat and non-threat needle pieces [1]. Under this analogy, problem difficulty may vary depending on factors such as how many stacks there are, how many threat and non-threat needles are distributed among them, and how like are threat and non-threat needles. Our goal in developing a performance evaluation laboratory (PE Lab) is to understand how variation along dimensions like these can affect the performance of a threat detection technology.

As the haystack analogy suggests, many characteristics that contribute to threat detection’s difficulty may be modeled simply using convenient abstractions of

real-world phenomena. Our primary concern is to identify well-performing regions of an information fusion approach—*e.g.*, its power to resolve ambiguities arising from partial, potentially corrupted, and temporally overlapping evidence fragments. We deliberately aim to drive the evaluated technology toward explicit representations of and reasoning about structured data and connections between entities and events. Abstraction serves to factor out issues inessential to this, and we model key relationships among threat and non-threat actors, events, and evidence characteristics approximating qualitative real-world relationships and quantitative values. We also factor out user interaction—*e.g.*, evidence visualization and mixed-initiative hypothesis development—so that technology evaluation is in principle entirely automated (although in practice we have not yet required hands-off execution for detection technologies).

We have followed these principles in developing the PE Lab during a multi-year, multi-contractor Government research program. The PE Lab’s synthetic datasets and associated experimental capabilities (for hypothesis scoring, experiment design, and results analysis) have served in several program-wide technology evaluations.

In subsequent sections, we describe the following.

- Abstract challenge problem domain
- PE Lab components
- Experiment design addressing problem space performance influences
- Related work
- Impact
- Conclusion

II. ABSTRACT CHALLENGE PROBLEM DOMAIN

We deal exclusively with synthetic datasets. These have the advantage for evaluation that (synthetic) ground truth is readily available for scoring. An artificial world mitigates privacy and security classification concerns and is tunable, supporting systematic experimentation. An abstract world facilitates parameterized overlap between threat and non-threat activities and de-emphasizes knowledge representation and reasoning requirements in comparison

to (threat) signal detection requirements, consistent with the funding program’s goals.

Figure 1 exhibits some real-world motivation behind the abstract, artificial world challenge problem domain we have developed.

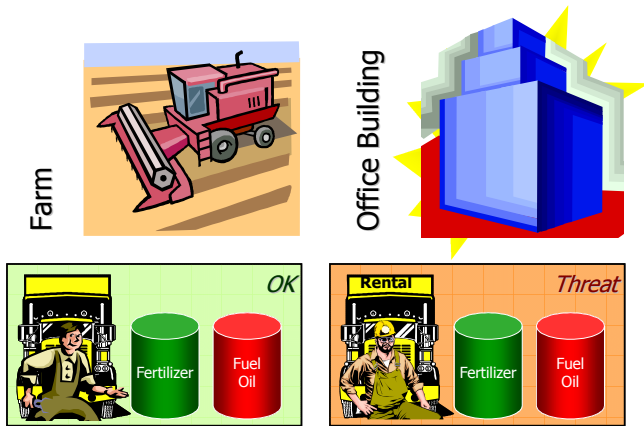


Figure 1: Real-world motivation for challenge problem

On the left-hand side of Figure 1, “Farmer Fred” buys fertilizer and fuel oil and transports these *via* truck to his farm. He applies the fertilizer using his tractor which (along with his truck) burns the fuel oil. (Fred is an honest, hard-working man.) On the right-hand side, “Demolition Dan” acquires the same resources but mixes them into a slurry that he transports (*via* rental truck) to the basement of an office building. (Dan is up to no good.)

Figure 2 illustrates our artificial world’s abstractions.

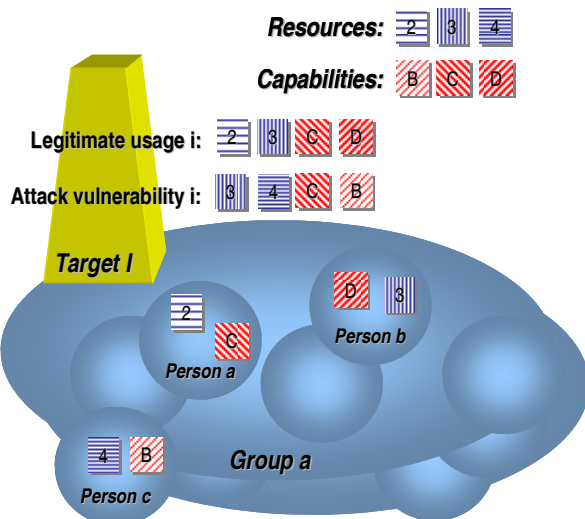


Figure 2: Artificial world abstractions

In the artificial world, capabilities (like farming and demolition) and resources (like fertilizer and fuel oil) are mapped to abstract elements that individuals can possess intrinsically or acquire. Infrastructure elements (like office buildings) are mapped to “targets” that support both legitimate/productive and destructive modes of use or

“exploitation.” Non-threat and threat individuals (like Fred and Dan) each may belong to any of various groups whose members collaborate towards different goals. Exploitations play out the general scheme of Figure 3.

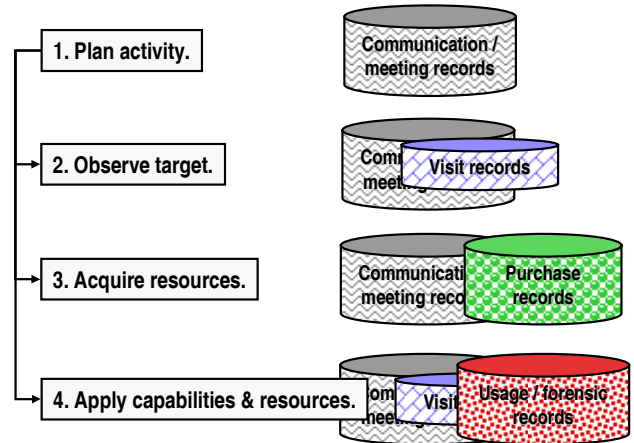


Figure 3: Generic exploitation scheme

The basic exploitation scheme on the left-hand side of Figure 3 unfolds through several levels of task decomposition, bottoming out in transactions with record types as indicated on the right-hand side. Alternative methods to accomplish each subtask (at each level) may be selected differently depending on whether threat or non-threat teams are undertaking threat or non-threat exploitations.

The challenge (depicted in Figure 4) to threat detection technology is to identify and report threat *cases*—top-level objects with attributes and values summarizing extant threat phenomena at a level sufficient for scoring. The case types that are detection objectives include threat actors (groups, individuals, and their aliases) and (ideally, impending) threat events/attacks. To perform this challenge, an automated threat detector is given information about the underlying artificial world that is relatively complete (excepting only a few, novel modes) and about events and actors that is only partial—per settings of “observability” parameters, as depicted notionally in Figure 4.

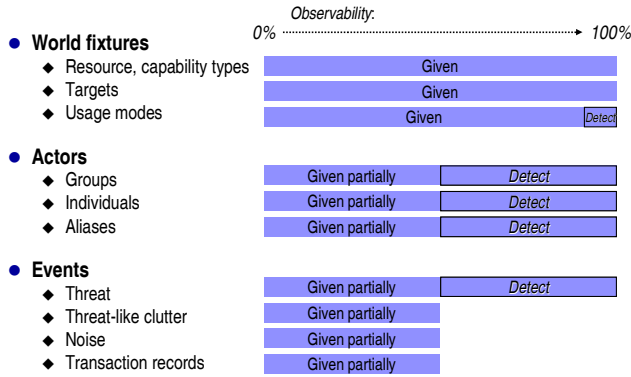


Figure 4: Threat detection challenge

We further describe the artificial world problem domain as follows.

- *Individuals* have permanent capabilities and can acquire resources as necessary to exploit a target in one of its modes.
- Both *resources* and *capabilities* are abstract enumerations.
- *Modes* are sets of capabilities and resources. *Vulnerability modes* are exploited by threat actors, *productivity modes* by both threat and non-threat actors.
- *Groups* are collections of individuals. Only *threat individuals* belong to *threat groups*. Both threat and *non-threat* individuals can belong to *non-threat groups*. Threat *groups* have designated exploitation modes—*vulnerability modes* for threat groups and *productivity modes* for both group types. A group can exploit a target that exhibits one of its modes.
- Groups have subgroups—*exploitation teams*—that focus on particular exploitation modes for which a team has qualified members.
- Individual group and team members tend to share abstract *social/demographic attributes*.
- To exploit a target, a team must acquire the required resources and apply the required resources and capabilities to the target (as illustrated in Figure 3).
- *Noise events* masking threat activity occur at several levels. We refer to non-threat exploitations as *clutter*. *Structured noise events* share intermediate structure with exploitations. *Transaction noise events* are atomic.

In this world, inter-connections abound. Modes overlap with respect to capabilities and resources (as suggested in Figure 1). Groups overlap with respect to modes, as do targets. Individuals overlap with respect to teams and groups and with respect to capabilities. Exploitations overlap in time with each other and with noise events. All of these inter-connections contribute to threat detection difficulty.

III. PE LAB ARCHITECTURE

The PE Lab includes key components summarized in Figure 5, where the following graphical conventions hold.

Square-cornered boxes represent products/artifacts. Round-cornered boxes represent processes. The threat detection process (presumed to use “link discovery” technology and referred to herein as LD) is realized separately by each detection technology developer and is rendered 3-dimensionally to highlight its status outside of the PE Lab proper (as the technology under test). Solid arrows represent flow of products/artifacts. Dotted arrows represent the flow of control information.

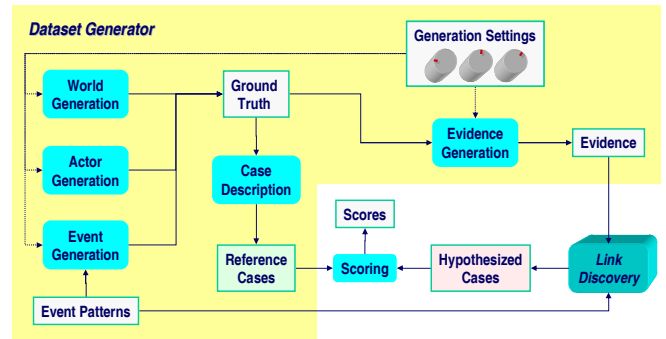


Figure 5: PE Lab overview

Generator parameter settings control creation of ground truth regarding the artificial world’s fixtures, actors, and events. Event generation follows event patterns covering both threat and non-threat/background activity. Evidence generation imposes partial observability and corruption (according to parameter settings) to create evidence from ground truth. Reference cases summarize ground truth for scoring. Detection uses published or learned patterns and evidence to detect threat entities, returning hypothesized cases. Scoring compares reference and hypothesized cases to yield scores. The collection of PE Lab artifacts making up a dataset—evidence, ground truth, and reference cases—supports experimentation.

A. Dataset Generator

In the real world, people typically interact simultaneously in several different social spheres associated with (*e.g.*) work, family, faith, neighborhood, sports/hobbies, civic involvement, shopping, and other relationships. People interact to coordinate times and locations for all of their activities, negotiate inter-activity constraints, and travel as necessary to interact. To make large dataset generation efficient, we have abstracted away such details, modeling all group activities with the same abstraction (the exploitation pattern), allowing individuals to participate in arbitrarily many activities simultaneously, and assuming that all activities take place in a single location (*e.g.*, a metropolitan area).

The PE Lab’s event generation language supports hierarchical task composition with combinational logic constructs for conditional execution and parallel and serial constructs that can be used independently or combined with iteration. Built-in probability distributions support

non-deterministic method selection and stochastic parameterization of optional task likelihoods and inter-task delays. The event pattern language is integrated with the underlying, general-purpose programming language (where the simulation world’s objects are defined) so that the patterns can manipulate world objects. Execution is efficient—we usually can generate a dataset involving 100,000 individuals and two million atomic transaction events in about twenty minutes using conventional hardware.

B. Hypothesis Scoring

Figure 6 depicts the generic hypothesis scoring scheme. Schrag and Takikawa [3] present a more thorough review.

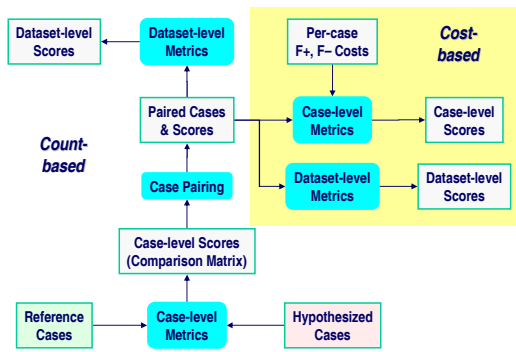


Figure 6: Scoring detail

The reference cases are from ground truth, the hypothesized cases from LD. We first compute scores for count-based metrics (precision, recall, and F-value), then for cost-based metrics.

Case objects have significant structure, and we want to credit LD for approximate matches. We assign weights to hypotheses’ different attributes to compute case object match quality and define weighted and thresholded versions of the count-based metrics. We compare all possible reference-hypothesis pairs based on their attribute values and store computed scores for case-level metrics in a matrix. Our dataset-level metrics require that we associate each hypothesized case with no more than one reference case (and *vice versa*). A given hypothesized case may approximately match many reference cases (and *vice versa*), so we apply an optimization algorithm to select pairs in the matrix that lead to good dataset-level scores. Using the scores for count-based metrics and specified per-case costs of false-positive (F+) and false-negative (F-) reporting, we additionally compute scores for cost-based metrics. These may be appealed to (rather than F-value) to provide an alternative, linear formulation of hypotheses’ utility.

The PE Lab also implements methods to address LD’s incremental output of threat event hypotheses (*i.e.*, alerting). Schrag *et al.* [4] present details on alert scoring.

IV. PE LAB-BASED EXPERIMENTATION

We summarize our experimental approach as follows.

1. Collapse many (fine) problem space parameters into a few dimensions with discrete (coarse) difficulty settings.
2. Specify a mix of experimental datasets that maximizes diversity over the difficulty settings.
3. Exercise participating detection technology configurations over datasets in the mix.
4. Score technologies’ output hypotheses.
5. Determine the statistical significance of apparent problem space performance influences by technology and detection objective.

We discuss these steps in subsequent subsections.

A. Problem Space Discretization

The coarse problem space dimensions are summarized in Table 1.

Group Connectivity	How many groups an individual belongs to
Noise, Clutter	How much threat masking
Dataset Size	How many observable transactions
Population Size	How many individuals
Pattern Complexity	Minimalistic vs. richer threat event modeling
Observability	How likely observations are
Corruption	How corrupted observations are
Aliasing	How frequently aliases are used
Event Confusability	How like are threat and non-threat activities
Target Duty Level	How busy targets are
Individual Duty Level	How busy individuals are

Table 1: Coarse problem space dimensions

Each coarse dimension corresponds to one or more fine parameters. For some dimensions, we discretize the fine parameters based on quantitative annual or semi-annual performance goals (set by the funding program)—see Table 2. For other dimensions we chose to explore, we discretize into difficulty settings such as Easy, Fair, Hard—see Table 3 for an example. We apply a stop-light color-coding over the discretized settings: lighter, greener colors for easier settings; darker, redder colors for harder.

Population Size	Y1	Y2.5	Y3
Number of individuals	~1,000	~10,000	~100,000
Mean threat group membership	20	80	80
Dev. threat group membership	5	20	20
Number of capabilities	50	100	150
Number of resources	50	100	150

Dataset Size	Y1	Y2.5	Y3
Number of observable transactions	N / A	~100,000	~1,000,000

Noise, Clutter	Y1	Y2.5	Y3
Threat-to-clutter event ratio	0.08	0.008	0.0008
Structured event SNR	0.08	0.008	0.0008
Transaction event SNR	0.08	0.008	0.0008
Individual SNR	0.4	0.08	0.008
Group SNR	0.8	0.16	0.016

Table 2: Fine parameter discretizations per annual performance goals

Group Connectivity	None		Easy		Fair		Hard	
Individual status:	Threat	Non-threat	Threat	Non-threat	Threat	Non-threat	Threat	Non-threat
	Mean groups per individual	1	1	2	4	4	6	6
Dev. groups per individual	0	0	1	2	2	3	3	4

Table 3: Style for other discretizations

B. Dataset Mix Specification

Several factors make effective experimentation challenging in this context. The evaluation dataset mix is scoped to occupy a few solid weeks of coordinated program effort. Processing is not always hands-off, with several disparate component developers sometimes manually handling intermediate results within a single technology configuration. A star experimental design with fixed baseline settings and single-dimension departures might serve individual technology configurations with single detection objectives, but—with each dataset—we must test multiple configurations over multiple objectives. What’s easier for one technology/objective combination might be harder for another. At evaluation time, we have somewhat sparse prior performance data from dry-run activities. We need an experiment that effectively tests over multiple baselines simultaneously, so we choose a diversity-maximizing, fractional factorial design.

We take the following steps, discussed below, to maximize diversity.

1. Specify cross-dimension settings constraints that ensure well dataset generation.
2. Perform constraint satisfaction to develop an initial dataset mix.
3. Perturb the initial mix in hill-climbing to optimize the experiment’s coverage.

1) *Dataset Specification Constraints*: Table 4 indicates some prohibited coarse setting combinations and associated rationale. *E.g.*, the Fat setting (corresponding to rich threat event modeling—resulting in more atomic transactions per threat event) results in too few threat events when the signal-to-noise ratio used is too low for the dataset size.

Noise, Clutter	Dataset Size	Population Size	Pattern Complexity	Rationale for prohibited combinations of settings listed in rows
	Y2.5	Y3		People doing too few things during a simulation
Y3	Y2.5			Too few threat events (maybe none)
Y1	Y3		Thin	Too many threat events to score practically
Y3	Y3		Fat	Too few threat events (maybe none)
Y2.5	Y2.5		Fat	
	Y3	Y2.5	Fat	Too many time ticks for incremental threat detection to be viable (currently)
	Y3	Y1	Fat	
	Y2.5	Y1	Fat	

Table 4: Constraints

Other combinations of coarse settings over these dimensions have been verified to generate well datasets. The coarse discretizations themselves assure compatibilities at the fine parameter level. For example, the various signal-to-noise (threat-to-non-threat) ratios (SNRs) for a given coarse setting in Table 2 are coordinated so that there are enough individuals to satisfy the generator’s minimum group size requirement. The discretization process thus factors out such fine, numerical constraints (whose violation would raise run-time exceptions), so that coarse constraint satisfaction over symbolic domains is sufficient for the dataset mix specification/experiment design.

2) *Constraint Satisfaction*: The constraint satisfaction problem is challenging in that we want a number of dataset specifications ranging over settings pools that have been fixed for each dimension. The pool for Group Connectivity, *e.g.*, includes six instances each for the tokens None, Easy, Fair, and Hard.¹ We have implemented an algorithm to specify a dataset mix respecting both the constraints and the pools.

3) *Coverage Optimization*: With an initial mix in hand, we perform a hill-climbing random walk over the space of well datasets, swapping any two datasets’ like settings along a given dimension whenever this decreases the maximum number of like settings shared across all datasets.

C. Detection Technology Exercise

Technology developers receive the test datasets in database form and are required to return ranked hypotheses in the scorer’s input format for each of the detection objectives noted in Figure 4.

D. Hypothesis Scoring Relativization

To compare dimension influences across different datasets requires comparable scores. As explained below, our default (absolute) scoring method credits hypothesis content that is patently manifest in datasets to different extents. Comparability requires a (relative) scoring method that factors this content out.

Evidence provided to LD (as illustrated in Figure 4) includes partial top-level case descriptions for some instances of the detection object types (threat event, group, individual, and alias association). These descriptions, corresponding to a legacy intelligence database, afford starting places for the detection process. The completeness, consistency, and transparency of these descriptions with respect to ground truth depend on settings for the Observability, Corruption, and Alias dimensions. In absolute scoring, LD gets credit for

¹ The experiment included 24 datasets developed directly from the pool specifications, plus one dataset deliberately designed to include what we believed were the easiest settings for all the dimensions.

reporting detection objects whether the same information appears in evidence or not.

In relative scoring, the detection task may be re-interpreted as, “Find unknown and correct misreported threat objects and their attributes.” Let a stand for LD’s absolute score, and let p stand for the score for returning exactly all and only the top-level threat case content provided in evidence. We use p as a baseline in computing the relative score $r = (a - p) / (1 - p)$. See Figure 7.

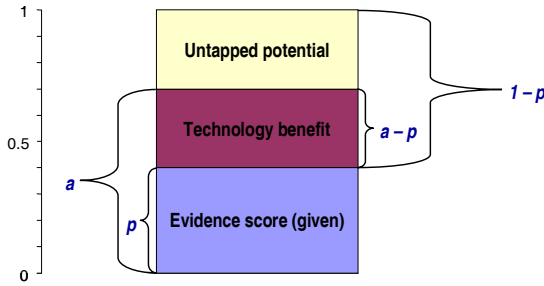


Figure 7: Relative scoring $r = (a - p) / (1 - p)$

Note that r can be negative—if LD does not perform as well as the baseline. Note also that the relative score rewards LD for any improvements to top-level threat case content provided in evidence—for corrected corruptions or resolved aliases.

E. Performance Influence Identification

Because of the coarse discretizations and constraints, our experiment design must be unbalanced (*i.e.*, have unequal numbers of settings within and across dimensions). This requires us to invent novel techniques to identify performance-influencing dataset characteristics, rather than, *e.g.*, applying ANOVA over coefficient means among regression fits.

Relative scores support ranking experimental datasets by LD’s performance for a given objective. Under this ranking, we expect the settings for a dataset dimension with significant performance influence to tend to exhibit the expected difficulty ordering—*e.g.*, “Easy, Fair, Hard” or “Y1, Y2.5, Y3.”² To determine the significance of the settings ordering actually observed, we first compute its distance to the ideal ordering. As Table 5 illustrates, we sum the distances between the two orderings of settings tokens with like rank for the same type—yielding in the example an aggregate distance of 32.

² This example is taken from an earlier evaluation in which the Observability dimension was discretized into just the three settings “Easy, Hard, Covert.”

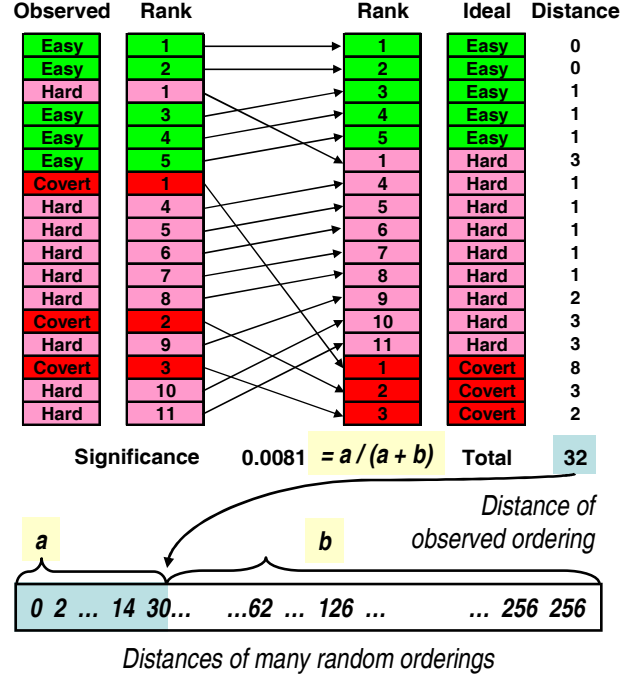


Table 5: Ranked settings significance testing

To determine the extent to which the observed ordering is significant with respect to the ideal—the extent to which the observed could have arisen strictly by chance, with lower values indicating greater significance—we similarly compute distances (represented in the abbreviated vector at the bottom of Table 5) to the ideal from a sufficient number $N = a + b$ of randomly generated token orderings, counting the number of times a the observed ordering is at least as close and reporting significance as a / N . The significance computation thus accounts both for the closeness of the observed ordering to the ideal and for variability of settings among the datasets.

By way of a case study, we include Table 6, covering results for a selected technology configuration with the group detection objective, (to provide membership lists for all of the threat groups). Table 6 covers an additional dimension (not included in Table 1) relevant to the technology configuration: Observed 2-way-comms per Individual. The 21 datasets actually processed using the selected technology³ are sorted by group detection performance (noted lower right). Each dataset dimension column is headed by an idealized settings order. Under the dimension name, significance is plotted on a log scale.

³ These results were developed in the context of a technology integration experiment; a different technology was used to process the remaining datasets.

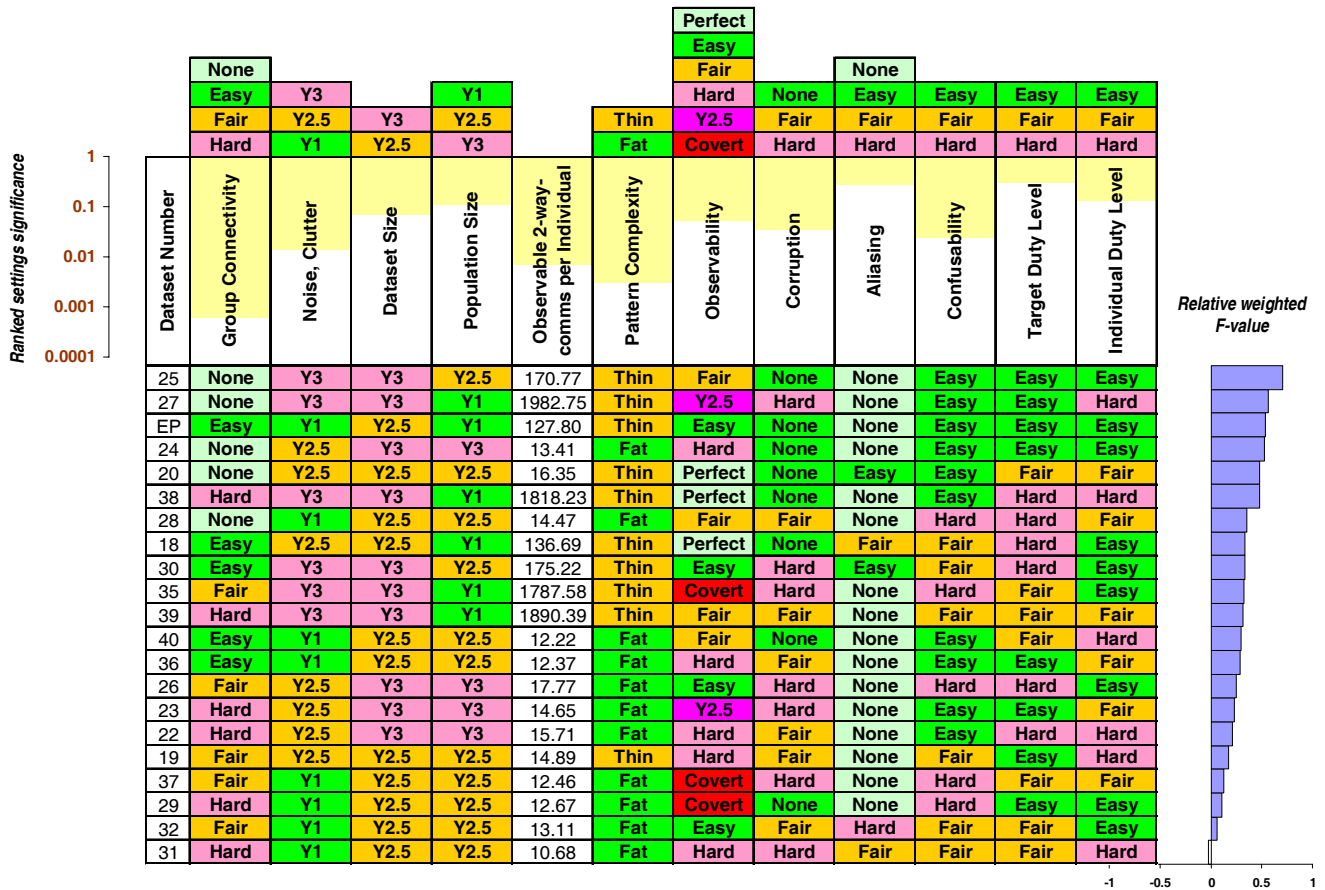


Table 6: Performance influence case study (group detection)

With a scoring option in effect to resolve aliases automatically from ground truth, Group Connectivity is the most significant influence: chance probability = 0.0006. (Without this option, Aliasing is.) We split the dataset mix along this dimension to continue analysis, with results shown below.

Group Connectivity at 0.0006 significance:
 = None: (No dimension of convincing significance)
 = Easy: (No dimension of convincing significance)
 = Fair or Hard: Observed 2-way-comms per Individual at 0.0005 significance

The dimensions noted are all relevant to group detection both intuitively and in the group detector’s implementation. Supporting detailed analyses are available from the author.

V. RELATED WORK

The PE lab’s dataset generation uses an artificial world abstraction style inspired by that of the Hats simulator of Morrison *et al.* [2]. A key difference is that the PE Lab is

structured deliberately to emphasize exploratory experimentation, as described in Section IV.

VI. IMPACT

The PE Lab supports advanced threat detection technology development in several ways.

As reported here, we assess technical progress through program-wide evaluation and identify particular problem characteristics most influential to a technology’s performance. Besides assisting individual technologists, this process can identify alternative technologies’ relative strengths and elucidate potentially advantageous combinations.

Within a functional architecture (such as the blackboard architecture schematized in Figure 8), we can employ the PE Lab to validate assumptions about the performance of a downstream component (or blackboard knowledge source—KS) based on that of an upstream one.

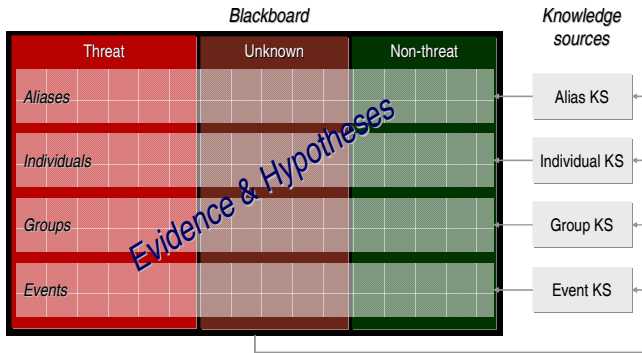


Figure 8: Blackboard-based component integration

Suppose, *e.g.*, that a group detector depends on an alias resolver to deliver sufficiently de-aliased evidence about individuals. If the resolver is not yet performing at a goal level meeting the detector’s input specs, we can still ascertain validity of performance claims for the latter by stubbing the former with a direct feed of evidence having per-spec de-aliasing. This can help to pinpoint performance gaps among functional components early during development.

In the future, we hope to facilitate such exploratory experimentation *via* a PE Lab-based component test harness and a program-wide commitment to automated (*i.e.*, hands-off) component execution. This has the potential to institutionalize the evaluation/experimentation process as a near-continuous loop in which experiments result in performance feedback to technology developers and developers respond to performance deficits with updated component versions. It also would enhance opportunities for large-scale experimentation. We could:

- Develop confidence intervals regarding a component’s performance over many datasets generated with the same parameter settings. We believe the significance results reported in Section IV-E are sufficiently dramatic to be considered solid, but it would be nice to know how much performance variation there is for different components in different problem space regions.
- Learn (problem space region-specific) relationships between component parameter settings (*e.g.*, likelihood ratio thresholds) and performance means and variances. This could help to elucidate which among alternative components with given functionality might perform best in given circumstances.
- Conduct more and finer star-design experiments to isolate the performance influences of particular dimensions or finer parameter settings.

VII. CONCLUSION

Threat detection research programs provide unique opportunities to develop and address challenge problems and associated evaluation metrics. It has been our privilege to work with members of the research community in defining and refining the PE Lab’s dataset generator, hypothesis scoring methods, and experimental designs. All of these

elements have evolved together, benefiting from significant community input along the way.

ACKNOWLEDGEMENT

We thank our colleagues who have contributed to the PE Lab’s design, implementation, and application, notably (with apologies to anyone we overlook) Jaffar Adibi, Chris Boner, Dan Bostwick, Hans Chalupsky, Hans Dettmar, Jim Eilbert, Paul Goger, Seth Greenblatt, Ian Harrison, Dan Hunter, David Jensen, Andrew Moore, Kendra Moore, David Page, Nick Pioch, Ben Rode, Ted Senator, Jeff Schneider, Sam Steingold, Masami Takikawa, Bob Washburn, Andre Valente, and Didier Vergamini.

REFERENCES

- [1] M. Goldszmidt and D. Jensen (editors). Recommendations Report: DARPA Workshop on Knowledge Discovery, Data Mining, and Machine Learning (KDD-ML). Arlington, VA: Defense Advanced Research Projects Agency. 1998.
- [2] C. Morrison, P.R. Cohen, G.W. King, J. Moody, and A. Hannon. Simulating Terrorist Threat in the Hats Simulator. In Proceedings of the First International Conference on Intelligence Analysis. MITRE Corp., 2005.
- [3] R. Schrag and M. Takikawa. Scoring Hypotheses from Threat Detection Technologies. AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection. 2006.
- [4] R. Schrag, M. Takikawa, P. Goger, and J. Eilbert. Scoring Alerts from Threat Detection Technologies. AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection. 2006.

USARSim: Providing a Framework for Multi-robot Performance Evaluation

S. Balakirsky, C. Scrapper
NIST
Gaithersburg, MD, USA
stephen.balakirsky@nist.gov,
chris.scrapper@nist.gov

S. Carpin
International University
Bremen
Bremen, Germany
s.carpin@iu-bremen.de

M. Lewis
University of Pittsburgh
Pittsburgh, PA, USA
ml@sis.pitt.edu

Abstract: Research efforts in urban search and rescue robotics have grown substantially in recent years. Two important robotic competitions (a robot physical league and a high-level infrastructure simulation league) were established in 2001 under the RoboCup umbrella to foster collaboration amongst institutions and to provide benchmark test environments for system evaluation. While these leagues play valuable roles, a significant gap existed between simulating disaster infrastructure and implementing agent behaviors on real hardware. In this paper we describe a software simulation framework intended to be a bridge between these communities. The framework allows for the realistic modeling of robots, sensors, and actuators, as well as complex, unstructured, dynamic environments. Multiple heterogeneous agents can be concurrently placed in the simulation environment thus allowing for team or group evaluations. This paper presents a description of the simulation along with results from the RoboCup 2006 Virtual Robot Competition in which it was used and a roadmap of the framework's future directions.

Keywords: performance evaluation, simulation, USARSim, robotics

I. INTRODUCTION

Research in robotics for Urban Search and Rescue (USAR) has recently experienced vigorous development. USAR offers a unique combination of engineering and scientific challenges in a socially relevant application domain [4]. The broad spectrum of relevant topics attracts the attention of a wide group of researchers, with expertise as diverse as advanced locomotion systems, sensor fusion techniques, cooperative multi-agent planning, human-robot interfaces and more.

The contest schema adopted by the RoboCup Rescue community, with the distinction between the real robots competition and the simulation competition, captures the two extremes of this growing community. The real robots competition is pushing the state of the art in robot mobility by challenging teams to perform in a room sized environment. These operations include tasks such as:

- autonomously negotiating compromised and collapsed structures,
- finding victims and ascertaining their condition,
- producing practical maps of victim locations,
- delivering sustenance and communications to victims,
- identifying hazards, and
- providing structural shoring.

The simulation competition's main purpose, by contrast, is to provide emergency decision support by integrating disaster information, prediction, planning, and human interfaces. The Version 0 simulator included simulations of building collapses, road blockages, spreading fire, and traffic. The competing teams must deploy scarce resources to address a dynamic disaster spreading over multiple city blocks. Both competition settings allow teams to be objectively evaluated in a challenging and realistic environment while providing a test arena for the development of performance metrics and standards for mobile robots.

Looking back at past RoboCup events, tremendous progress has been shown in both the real and simulation competitions. In 2002, the real rescue robots competition was described as a competition where the complexity of the problem caused most researchers to use tele-operated robots [1]. In the simulation competition, emphasis was placed on the inter-agent communication models adopted [7]. The huge gap between these two extremes is evident.

Only two years later [5], the real robot competition saw the advent of teams with three dimensional mapping software, intelligent perception, and the first team with a fully autonomous multi-robot system. Within the simulation competition, teams exhibited cooperative behaviors, special agent programming languages, and learning components. With these strong gains, it is evident that relevant techniques will soon begin to migrate between the competitions. Nevertheless, certain logistic obstacles still prevent a seamless and profitable percolation of ideas and knowledge.

At RoboCup 2005, USARSim was selected as the software infrastructure for a new competition that fits between the physical and agent competitions. During the 2006 competition, eight teams from four continents competed in an indoor/outdoor city block sized virtual arena.

The remainder of this paper broken down as follows: Section 2 describes a short overview of the USARSim framework; Section 3 outlines the competition and performance metrics under which the teams were judged; finally, Section 4 presents lessons learned and tentative

overview of rule and procedure changes for next year's competition.

II. USARSim FRAMEWORK

The current version of USARSim is based on the UnrealEngine2 game engine that was released by Epic Games as part of Unreal Tournament 2004¹. This engine may be inexpensively obtained by purchasing the Unreal Tournament 2004 game. The engine handles most of the basic mechanics of simulation and includes modules for handling input, output (3D rendering, 2D drawing, and sound), networking, physics and dynamics. Multiplayer games use a client-server architecture in which the server maintains the reference state of the simulation while multiple clients perform the complex graphics computations needed to display their individual views. USARSim uses this feature to provide controllable camera views and the ability to control multiple robots. In addition to the simulation, a sophisticated graphical development environment and a variety of specialized tools are provided with the purchase of Unreal Tournament.

The USARSim framework builds on this game engine and consists of

- standards that dictate how agent/game engine interaction is to occur,
- modifications to the game engine that permit this interaction
- an Application Programmer's Interface (API) that defines how to utilize these modifications to control an embodied agent in the environment
- 3-D immersive test environments.

In order to provide a standardized external interface, all units of measurement used in the USARSim API meet the International System of Units (SI) standard conventions. SI Units are a General Conference on Weights and Measures developed convention that is built on the modern metric system, and is recognized internationally. For coordinate systems, USARSim leverages the previous efforts of the Society of Automotive Engineers, who published a set of standards for vehicle dynamics called *Vehicle Dynamic Terminology* [6]. This set of standards is recognized as the American National Standard for vehicle dynamics and contains a comprehensive set of standards that describes vehicle dynamics through illustrated pictures of coordinate systems, definitions, and formal mathematical representations of the dynamics. Finally, the messaging protocol, including

¹ Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

the primitives, syntax, and the semantics are defined as part of the Application Programmer's Interface (API).

When an agent is instantiated through USARSim, three basic classes of objects are created that provide for the complete control of the agent. These include robots, sensors, and mission packages and are defined as part of the API to USARSim. For each class of objects there are class-conditional messages that enable a user to query the component's geography and configuration, send commands, and receive status and data. Permissible calls into the game engine and complete details on the API may be found in the USARSim Reference Manual².

It is envisioned that researchers will utilize this framework to perfect algorithms in the areas of:

- Autonomous multi-robot control
- Human, multi-robot interfaces
- True 3D mapping and exploration of environments by multi-robot teams
- Development of novel mobility modes for obstacle traversal
- Practice and development for real robots that will compete in the RoboCup Rescue Physical League

III. VIRTUAL ROBOT COMPETITION

The RoboCup Rescue Virtual Competition is the third competition running under the RoboCupRescue Simulation League umbrella. It utilizes the USARSim framework to provide a development, testing, and competition environment that is based on a realistic depiction of a disaster scenario. It has been previously stated [2,3] that the Virtual Robots competition should serve the following goals:

- Provide a meeting point between the different research communities involved in the RoboCupRescue Simulation league and the RoboCupRescue Robot league. The two communities are attacking the same problem from opposite ends of the scale spectrum (city blocks vs. a small rubble area) and are currently far apart in techniques and concerns. The Virtual Competition offers close connections to the Robot league, as well as challenging scenarios for multi-agent research. The scenarios for the 2006 competition were chosen to highlight these connections. They were based on an outdoor accident scene, and an indoor fire/explosion at an office building. These scenarios included real-world challenges such as curbs, uneven terrain, multi-level terrain (i.e. the void space under a car), maze-like areas, stairs, tight spaces, and smoke. An

² The reference manual may be found in the file releases area of the USARSim sourceforge site (http://sourceforge.net/project/showfiles.php?group_id=145394&package_id=180746&release_id=407397)

exact copy of one of the RobCupRescue Robot league arenas was also included in the office space, and elements of other arenas were scattered throughout the environment. The area was far too large to be explored by a single agent in the time permitted (20 minutes) and thus the use of multi-agent teams was beneficial. Accommodations were provided in the worlds to assist less capable (in terms of mobility) robotic systems. For example, wheelchair ramps were provided that allowed for alternative access around stairs. Snap shots of small sections of these environments may be seen in Figure 1.



Figure 1: Representative snapshot of a USARSim indoor (a) and outdoor (b) scene.

- Lower entry barriers for newcomers. The development of a complete system performing search and rescue tasks can be overwhelming. The possibility to test and develop control systems using platforms and modules developed by others makes the startup phase easier. With this goal in mind, the open source strategy already embraced in the other competitions is fully supported in the

RobocupRescue Simulation league. Software from this year's top teams has already been posted on the web.

- Let people concentrate on what they can do better. Strictly connected to the former point, the free sharing of virtual robots, sensors, and control software allows people to focus on certain aspects of the problem (victim detection, cooperation, mapping, etc), without the need to acquire expensive resources or develop complete systems from scratch. In order to help people determine if they really can "do better," performance metrics were applied to the competing systems.

IV. PERFORMANCE METRICS

For the 2006 competition, it was decided that performance would be measured in terms of information provided about victims, amount of area explored, and map quality. The primary goal of the competition was to locate victims in the environment. However, what does it mean to "locate" a victim? Several interpretations exist ranging from simply requiring a robot to be in proximity of a victim (e.g. drive by the victim) to requiring the robot to employ sensor processing to recognize that a victim is located near-by (e.g. recognize a human form in a camera image). It was decided that robots should be required to be "aware" of the presence of a victim, but that requiring every team to have expertise in image processing was against the philosophy of lowering entry barriers. Therefore, a new type of sensor: a victim sensor, was introduced.

This sensor was based on Radio Frequency Identification Tag (RFID) technology. False alarm tags were scattered strategically in the environment, and each victim contained an embedded tag. At long range (10 m), a signal from the tag was readable when the tag was in the field of view (FOV) of the sensor. At closer range (6 m), the sensor would report that a victim or false alarm was present. At even closer range (5 m) the ID of the victim would be reported. Finally, at the closest range (2 m), the status of the victim was available. Points were subtracted for reporting false alarms, and were awarded for various degrees of information collected from the victims. Bonus points were awarded for including an image of the victim with the report.

As the robots were exploring the environment, their poses (on a 1 s interval) and any collisions between the robots and victims were automatically logged. The pose information was fed into a program that automatically computed the amount of area that was covered by the robotic teams. This figure was normalized against the expected explored area for the particular run, and points were awarded accordingly. The collision information was used as an indication of suboptimal navigation strategies that should be penalized. Another parameter that was used to determine the overall score was the

number of human operators that were needed to control the robots. The idea was borrowed from the Physical Robots competition with the intent of promoting the deployment of fully autonomous robot teams, or the development of sophisticated human-robot interfaces that allow a single operator to control many agents.

The final area that was judged during the competition was map quality. The map quality score was based on several components.

- Metric quality – The metric quality of a map was scored automatically by examining the reported locations of “scoring tags”. Scoring tags are RFID tags that report their relative location to a robot and then disappear. A requirement of the competition was for the teams to report the global coordinates of these tags at the conclusion of each run. The automatic scoring program then analyzed the deviation of the perceived locations from the actual locations.
- Multi-vehicle fusion – Teams were only permitted to turn in a single map file. Those teams that included the output from multiple robots in that single map were awarded bonus points.
- Attribution – One of the reasons to generate a map is to convey information. This information is often represented as attributes on the map. Points were awarded for including information on the location, name, and status of victims, the location of obstacles, the paths that the individual robots took, and the location of RFID scoring tags.
- Grouping – A higher order mapping task is to recognize that discrete elements of a map constitute larger features. For example the fact that a set of walls makes up a room, or a particular set of obstacles is really a car. Bonus points were awarded for annotating such groups on the map.
- Accuracy – An inaccurate map may make a first responder’s job harder instead of easier. Points were assessed based on how accurately features and attributes were displayed on the map.
- Skeleton quality – A map skeleton reduces a complex map into a set of connected locations. For example, when representing a hallway with numerous doorways, a skeleton may have a line for the hallway and symbols along that line that represent the doors. A map may be inaccurate in terms of metric measurements (a hallway may be shown to be 20 m long instead of 15 m long), but may still present an accurate skeleton (there are three doors before the room with the victim). The category allowed the judges to award points based on how accurately a map skeleton was represented.
- Utility – One of the main objectives of providing a map was to create the ability for a first responder to utilize the map to determine which areas had been

cleared, where hazards may be located, and where victims were trapped. Points were granted by the judges that reflected their feelings on this measure.

The above mentioned elements were numerically combined according to a schema that took into account merit factors that concerned (1) victims’ discovery, (2) mapping, and (3) exploration. The exact point calculations for each factor are presented below.

1. 10 points were awarded for each reported victim ID. An additional 20 points were granted if the victim’s status was also provided. Properly localizing the victim in the map was rewarded with an additional 10 points. At the referee’s discretion, up to 20 bonus points were granted for additional information produced. For example, some teams managed to not only identify victims, but to also provide pictures taken with the robot’s cameras. For this additional information teams were awarded with 15 bonus points.
2. Maps were awarded up to 50 points based on their quality, as previously described. The obtained score was then scaled by a factor ranging between 0 and 1 that measured the map’s metric accuracy. This accuracy was determined through the use of the RFID scoring tags.
3. Up to 50 points were available to reward exploration efforts. Using the logged position of every robot, the total amount of explored square meters (m^2) was determined and related to the *desired* amount of explored area. This desired amount was determined by the referees and was based on the competition environment. For example, in a run where 100 m^2 were required to be explored, a team exploring 50 m^2 would receive 25 points, while a team exploring 250 m^2 would receive 50 points, i.e. performances above the required value were leveled off.

On the penalization side, 5 points were deducted for each collision between a robot and a victim. Finally, the overall score was divided by $(1+N)^2$, where N was the number of operators involved. So, completely autonomous teams, i.e. $N=0$, incurred no scaling, while teams with a single operator had their score divided by 4. No team used more than one operator.

It should be noted that except for the map quality, all of the above components were automatically computed from the information logged during the competition. Therefore subjective opinions during the scoring stage were reduced to the minimum. In an ideal scenario, the scoring step would be completely automatic as is currently the case for the RobocupRescue Simulation agent competition.

In addition to assigning points to determine the overall best systems, the judges assigned winning teams in the special categories of map creation and human-machine interface. The map creation award was presented to the team that consistently scored the highest in the map quality assessment while the human-machine interface award recognized the team with the most innovative robot control console.

The winning teams from the 2006 RoboCup Rescue Virtual Competition were:

First Place – Rescue Robots Freiburg, University of Freiburg, Germany

Second Place – Virtual IUB, International University Bremen, Germany

Third Place – UVA, University of Amsterdam, The Netherlands

Best Mapping – UVA, University of Amsterdam, The Netherlands

Best Human-Computer Interface – Steel, University of Pittsburgh, USA

V. COMPETITION FUTURE ASPECTS

As in all good competitions, the Rescue Virtual Competition must evolve in order to continue to challenge the competitors. In order to keep up with the competition changes, the metrics must evolve as well. While no firm decisions have been made about next year's competition, the following presents some of the current ideas.

- Modify victim discovery to require not only discovery, but will also require that a team provide a "data sheet" for each discovered victim. In order to receive full score, this sheet would need to include a map to the victim, information about the victim's status, and any hazards that exist along the route to the victim.
- The mapping requirement will remain the same. However, additional emphasis may be placed on including annotations on this map. Annotations should include hazards, "cleared areas", victim locations, and routes that robots took.
- Exploration may be based on the amount of area that a robot "clears". Where the definition of clearing an area means that all hazards and victims in the given area have been localized.
- Penalties will be assigned for victim bumping as well as reporting an area as clear that has victims or hazards. Inaccurate maps to victim locations will also be penalized.

VI. SUMMARY

This paper has presented results from the first annual RoboCup Rescue Virtual Competition that took place in June 2006 in Bremen Germany. The evaluation metrics for the competition were discussed, and possible modifications for the future were presented. Next year's competition will take place in Atlanta, GA. Everyone is invited to download the open source software (<http://sourceforge.net/projects/usarsim/>) and participate.

References

1. Asada, M. and Kaminka, G., "An Overview of RoboCup 2002 Fukuoka/Busan," *RoboCup 2002: Robot Soccer World Cup VI*, edited by G. Kaminka, P. Lima, and R. Rojas Lecture Notes in Artificial Intelligence (LNAI), Springer, 2002, pp. 1-7.
2. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., and Scrapper, C., "Bridging the gap between simulation and reality in urban search and rescue," *2006 RoboCup Symposium*, 2006.
3. Carpin, S., Wang, J., Lewis, M., Birk, A., and Jacoff, A., "High fidelity tools for rescue robotics: Results and perspectives," *2005 RoboCup Symposium*, 2005.
4. Kitano, H. and Tadokoro, S.. *RobocupRescue: A Grand Challenge for Multiagent and Intelligent Systems*. AI Magazine 1, 39-52. 2001.
Ref Type: Magazine Article
5. Lima, P. and Custódio, L., "RoboCup 2004 Overview," *RoboCup 2004: Robot Soccer World Cup VIII*, edited by D. Nardi, M. Riedmiller, and J. Santos-Victor Lecture Notes in Artificial Intelligence (LNAI), Springer, 2004, pp. 1-17.
6. Society of Automotive Engineers, "Vehicle Dynamics Terminology," SAE, J670e, 1976.
7. Tomoiki, T., "RoboCupRescue Simulation League," *RoboCup 2002: Robot Soccer World Cup VI*, edited by G. Kaminka, P. Lima, and R. Rojas Lecture Notes in Artificial Intelligence (LNAI), Springer, 2002, pp. 477-481.

Performance Evaluation of a Terrain Traversability Learning Algorithm in The DARPA LAGR Program

Michael Shneier, Will Shackelford, Tsai Hong and Tommy Chang
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
{firstname.lastname}@nist.gov

Abstract—The Defense Applied Research Projects Agency (DARPA) Learning Applied to Ground Vehicles (LAGR) program aims to develop algorithms for autonomous vehicle navigation that learn how to operate in complex terrain. For the LAGR program, The National Institute of Standards and Technology (NIST) has embedded learning into a control system architecture called 4D/RCS to enable the small robot used in the program to learn to navigate through a range of terrain types. This paper describes performance evaluation experiments on one of the algorithms developed under the program to learn terrain traversability. The algorithm uses color and texture to build models describing regions of terrain seen by the vehicle’s stereo cameras. Range measurements from stereo are used to assign traversability measures to the regions. The assumption is made that regions that look alike have similar traversability. Thus, regions that match one of the models inherit the traversability stored in the model. This allows all areas of images seen by the vehicle to be classified, and enables a path planner to determine a traversable path to the goal.

The algorithm is evaluated by comparison with ground truth generated by a human observer. A graphical user interface (GUI) was developed that displays an image and randomly generates a point to be classified. The human assigns a traversability label to the point, and the learning algorithm associates its own label with the point. When a large number of such points have been labeled across a sequence of images, the performance of the learning algorithm is determined in terms of error rates. The learning algorithm is outlined in the paper, and results of performance evaluation are described.

Keywords: Learning, performance evaluation, traversability, computer vision, robotics, LAGR

I. INTRODUCTION

The Defense Applied Research Projects Agency (DARPA) Learning Applied to Ground Vehicles (LAGR) program [1] aims to develop algorithms for autonomous vehicle navigation that learn how to operate in complex terrain. Over many years, the National Institute of Standards and Technology (NIST) has developed a reference model control system architecture called 4D/RCS that has been applied to many kinds of robot control, including autonomous vehicle control [2]. For the LAGR program, NIST has embedded learning into a 4D/RCS

controller to enable the small robot used in the program to learn to navigate through a range of terrain types [3]. The vehicle learns in several ways. These include learning by example, learning by experience, and learning how to optimize traversal. In this paper, we present a method of evaluating a learning algorithm used in LAGR that associates terrain appearance with traversability. The paper briefly describes the learning method and then focuses on the evaluation procedure. The approach is illustrated with examples taken from tests run by the LAGR evaluation team.

The appearance of regions in an image has been described in many ways, but most frequently in terms of color and/or texture. Ulrich and Nourbakhsh [4] used color imagery to learn the appearance of a set of locations to enable a robot to recognize where it is. A set of images was recorded at each location and served as descriptors for that location. Images were represented by a set of one-dimensional histograms in both HLS (hue, luminance, saturation) and normalized Red, Green, and Blue (RGB) color spaces. When the robot needed to recognize its location, it compared its current image with the set of images associated with locations. The location was recognized as that associated with the best-matching stored image.

In [5] the authors also addressed the issue of appearance-based obstacle detection using a single color camera and no range information. Their approach makes the assumptions that the ground is flat and that the region directly in front of the robot is ground. This region is characterized by color histograms and used as a model for ground. In the domain of road detection, a related approach is described in [6]. In principle, the method could be extended to deal with more classes, and our algorithm can be seen as one such extension that does not need to make the assumptions because of the availability of range information for regions close to the vehicle.

Learning has been applied to computer vision for a variety of applications, including traversability prediction. Wellington and Stentz [7] predicted the load-bearing surface under vegetation by extracting features from range data and associating them with the actual surface height measured

when the vehicle drove over the corresponding terrain. The system learned a mapping from terrain features to surface height using a technique called locally weighted regression. Learning was done in a map domain. We also use a map in the current work, although it is a two dimensional (2D) rather than a three dimensional (3D) map, and we also make use of the information gained when driving over terrain to update traversability estimates, although not as the primary source of traversability information. The models we construct are not based on range information, however, since this would prevent the extrapolation of the traversability prediction to regions where range is not available.

Howard et al. [8] presented a learning approach to determining terrain traversability based on fuzzy logic. A human expert was used to train a fuzzy terrain classifier based on terrain roughness and slope measures computed from stereo imagery. The fuzzy logic approach was also adopted by Shirkhodaie et al. [9], who applied a set of texture measures to windows of an image followed by a fuzzy classifier and region growing to locate traversable parts of the image.

Talukder and his colleagues [10] describe a system that attempts to classify terrain based on color and texture. Terrain is segmented using labels generated from a 3D obstacle detection algorithm. Each segment is described in terms of Gabor texture measures and color distributions. Based on color and texture, the segments are assigned to pre-existing classes. Each class is associated with an a priori traversability measure represented by a spring with known spring constant. We also make use of 3D obstacle detection in our work, but don't explicitly segment the data into regions. We model both background and obstacle classes using color and texture, but all models are created as the vehicle senses the world. Given that we have no prior knowledge of the type of terrain that may be encountered, it is usually not possible to pre-specify the classes. Similarly, the vehicle learns the traversability of the terrain by interacting with it, either by driving over it or generating a bumper hit.

II. THE LEARNING ALGORITHM

The learning process takes input in the form of labeled pixels with associated (x, y, z) positions. The labels are provided on a pixel-by-pixel basis by an obstacle detection algorithm that works on stereo data [11]. Given the labels and color characteristics of the pixels, the learning algorithm constructs color and texture models of traversable and non-traversable regions and uses them for terrain classification. The approach to model building is to make use of the labeled color data to describe regions in the environment around the vehicle and to associate a cost of traversing each region with its description. The terrain models are learned using an unsupervised scheme that makes use of both geometric and appearance information.

In our algorithm an assumption is made that terrain regions that look similar will have similar traversability. The learning works as follows (see [12]). The system constructs a map of a 40 m by 40 m region of terrain surrounding the vehicle, with

map cells of size 0.2 m by 0.2 m and the vehicle in the center of the map. The map is always oriented with one axis pointing north and the other east. The map scrolls under the vehicle as the vehicle moves, and cells that scroll off the end of the map are forgotten. Cells that move onto the map are cleared and made ready for new information.

The model-building algorithm takes as input the color image, the associated and registered range data (x, y, z) points), and the labels (GROUND and OBSTACLE) generated by the obstacle detection algorithm. Also associated with these data is the location and pose of the vehicle when the data were collected. When new data are received, the vehicle location and pose information are used to scroll the map so that the vehicle occupies the center cell of the map.

Points are projected into cells based on their 3D positions. Each cell receives all points that fall within the square region in the world determined by the location of the cell, regardless of the height of the point above the ground. The cell to which the point projects accumulates information that summarizes the characteristics of all points seen by this cell. This includes color, texture, and contrast properties of the projected points, as well as the number of OBSTACLE and GROUND points that have projected into the cell. Color is represented by ratios R/G, G/B, and intensity. The intensity and color ratios are represented by 8-bin histograms stored in a normalized form so that they can be viewed as probabilities of the occurrence of each ratio. Texture and contrast are computed using Local Binary Patterns (LBP) [13]. These patterns represent the relationships between pixels in a 3x3 neighborhood in the image, and their values range from 0 to 255. The texture measure is represented by a histogram with 8 bins, also normalized. Contrast is represented by a single number ranging from 0 to 1.

When a cell accumulates enough points it is ready to be considered as a model. We determine the sample size by requiring 95% confidence that the sample represents the true distribution. In order to build a model, we also require that 95% of the points projected into a cell have the same label (OBSTACLE or GROUND). If a cell is the first to accumulate enough points, its values are copied to instantiate the first model. Models have exactly the same structure as cells, so this is trivial. If there are already defined models, the cell is matched to the existing models to see if it can be merged or if a new model must be created. Matching is done by computing a weighted sum of the squared difference of the elements of the model and the cell. Cells that are similar enough are merged into existing models; otherwise, new models are constructed.

At this stage, there is a set of models representing regions whose appearance in the color images is distinct (Fig 1). Our interest is not so much in the appearance of the models, but in the traversability of the regions associated with them. Traversability is computed from a count of the number of GROUND and OBSTACLE points that have been projected into each cell, and accumulated into the model. Models are given traversability values computed as $N_{\text{OBSTACLE}} / (N_{\text{GROUND}} + N_{\text{OBSTACLE}})$.

+ N_{OBSTACLE}). These models correspond to regions learned by example.

Learning by experience is used to modify the models. As the vehicle travels, it moves from cell to cell in the map. If it is able to traverse a cell that has an associated model, the traversability of that model is increased. If it hits an obstacle in a cell, the traversability is decreased.

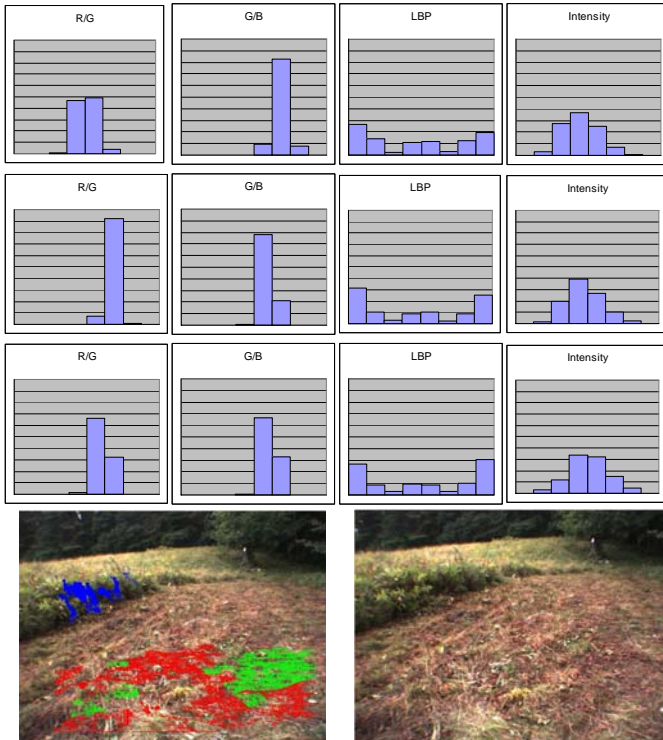


Fig 1. Examples of histograms used to construct models. Top row corresponds to the blue regions in the left image. Middle row corresponds to the green region. Bottom row corresponds to the red region. The blue region is not traversable, while the other two regions are traversable.

To classify a scene, only the color image is needed (no range data). A window is passed over the image and color, texture, and intensity histograms and a contrast value are computed as in model building. A comparison is made with the set of models, and the window is classified with the best matching model, if a sufficiently good match value is found. Regions that do not find good matches are left unclassified. Windows that match with models inherit the traversability measure associated with the model. In this way large portions of the image are classified (Fig 2).

The vehicle needs to know the locations of obstacle and ground regions, but has no stereo information during classification. To address this problem, the assumption is made that the ground is flat, i.e., that the pose of the vehicle defines a ground plane through the wheels. This allows windows that match with models to be mapped to 3D locations. Another assumption is that all obstacles (windows matching with models created from obstacle points) are

normal to the ground plane. This allows obstacle windows to be projected into the ground plane and thus to acquire 3D locations. Because of the ground plane assumption, the algorithm only processes the image from in front of the vehicle to a small distance above the horizon, to catch the obstacles but ignore the sky.

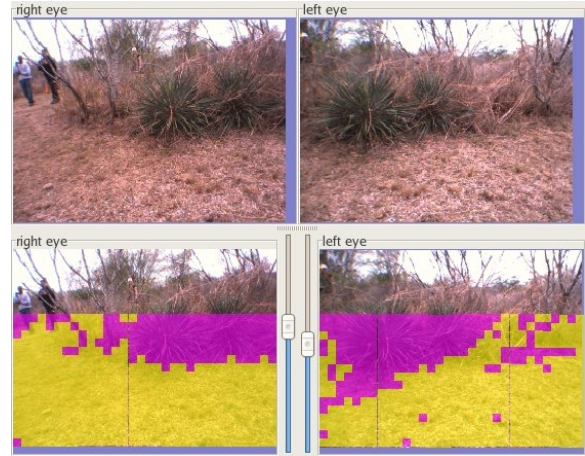


Fig 2. Top: Left and right eye views of a typical scene from Test 9. Bottom: Classification showing regions that are traversable in yellow, and not traversable in magenta.

III. EVALUATING THE ALGORITHM

The entire LAGR system was tested over the course of a year by a separate Government team using a vehicle functionally identical to the vehicles on which the software is developed. Tests occurred about once a month. Developers sent their control software on flash memory cards to the test facility. The software was loaded onto a vehicle which was commanded to travel from a start waypoint to a goal waypoint through an obstacle-rich environment. The environment was not seen in advance by the development teams. The Government team measured the performance of the system on multiple runs. To demonstrate learning, performance was expected to improve from run to run as the systems became familiar with the course. While these tests gave a good indication of how learning improved the overall performance, they did not provide evaluations of individual learning algorithms.

Evaluating the algorithm described in this paper requires determining how well the learned models enable the system to classify the degree of traversability of the terrain around the vehicle. The evaluation makes use of ground truth generated by one or more human observers who use a graphical tool to generate ground truth points against which the learning algorithm is compared.

Data sets used for the evaluation consist of log files generated during the tests conducted by the Government team. Log files contain the sequence of images collected by the two pairs of stereo cameras on the LAGR vehicle and information from the other sensors, including the navigation (GPS and

INS) sensors and bumper sensors (physical and IR bumpers). The NIST LAGR system performs exactly the same when playing back a log file as it did when it first ran the course, so long as no changes are made in the algorithms. Therefore, logged data is a good source for performance testing.

The ground truth is collected by a human stepping sequentially through the log file, and classifying one or more points from each image. A graphical tool is used to display the image and randomly select a point (Fig. 3). The point is highlighted for the user, who selects one of the labels Ground (G), Obstacle (O), or Unknown (U). The tool then writes a record to a file containing the frame number, coordinates of the selected point, and the label provided by the user. Note that the Unknown label is used for points that are neither ground nor obstacle (such as sky) as well as points where the human truly cannot decide between ground and obstacle (such as at the base of an obstacle that merges smoothly with the ground). When ground truth collection is complete, the file is available for evaluating the performance of the learning algorithm (or any other algorithm that assigns traversability labels to regions).

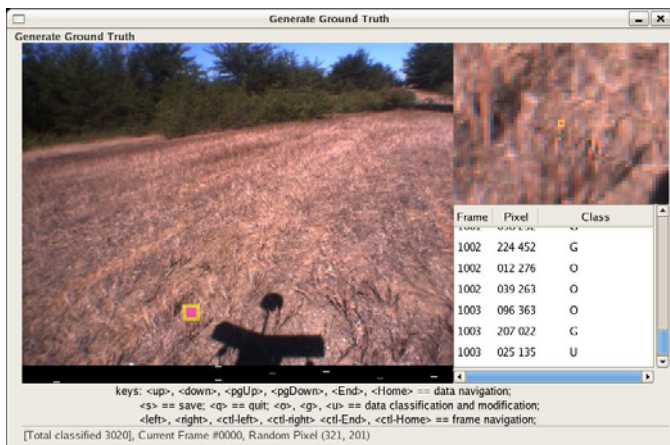


Fig. 3. The GUI for generating ground truth showing a frame from Test 7.

The learning algorithm reads the ground truth file and the log file. It processes the log file as it usually does when running on the vehicle. Each time it comes to an image frame for which ground truth is available, it classifies the points selected in the frame and writes out a file containing the ground truth it read in plus an entry giving the learned classification of the pixel in the ground truth file. When the entire log file has been processed, the output file contains an entry for each ground truth point that gives both the human's classification and the system's classification. Under the assumption that the human's classification is correct, an analysis can be conducted of the errors committed by the learning algorithm.

IV RESULTS

The evaluation was applied to a number of examples taken from data gathered by the LAGR evaluation team at locations in Virginia and Texas. Results are shown for these examples and an overall evaluation is given of the performance of the algorithm across all the data sets.

In the evaluations, the learning system starts out with no models. This is how the system typically starts, at least for the first test run at each location. As it reads the log file and the ground truth data, the learning program both creates the models and classifies the ground truth points. This means that early in the sequence of images, only a small number of models are available for classification. As more of the terrain is seen, more models are constructed, and the range of regions that can be classified increases. The algorithm learns very fast, however, often creating the first few models from the first frame or two of data. Since the terrain doesn't usually change abruptly, classification performs well from the start, particularly for points close to the vehicle.

Four sets of ground truth data were created by three different people using the GUI in Fig. 3. The data were taken from log files of three different tests. Test 6 was conducted in September, 2005 in Fort Belvoir, VA. Test 7 was also conducted at Fort Belvoir, in October, 2005. The course was very different, however. Test 9 was conducted in San Antonio, TX at the Southwest Research Institute's Small Robot Testbed.

A. Test 6

Test 6 included a run along a path through a slightly wooded area, ending in an open field. Two synthetic obstacles made out of orange plastic mesh were placed in the path of the vehicle (Fig. 4), with the goal being to learn that the first fence represented an obstacle and use that knowledge to avoid the second fence.



Fig. 4. A view of the first orange fence in Test 6.

The ground truth created for Test 6 consisted of approximately 3 points per frame, using the log file of the first test run. Because the human sometimes labeled a point as

Unknown, and because some of the points randomly selected for ground truth were in the sky, the actual number of usable points was closer to 2 per frame (there were 1,270 frames).

TABLE I shows a summary of the results of the evaluation. As can be seen, the algorithm labeled 87% of the points with the same class as the human. Of the incorrect labels, 30% arose from situations where the algorithm did not find a match with any model and labeled the points Unknown, 52% came from incorrectly labeling points as Obstacle instead of Ground, and 17% from labeling points as Ground instead of Obstacle.

TABLE I
Results for Test 6

Test 6, 2513 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
2197	317	87.4%	12.6%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
30%	52%	17%	

B. Test 7

The course for Test 7 began in an open field. The straight-line path would put the vehicle in a position that required a long detour through dense bushes. Traveling to the right of the straight-line path led to an easy route to the goal. The Government team placed an artificial barrier in the path to make it difficult to choose the right hand direction the first time the course was seen (Fig. 5). The idea was that the vehicle would fight its way through the bushes on the first run before reaching the goal, but would learn to recognize the barrier and select the right hand route on subsequent test runs. In fact, this is what the NIST vehicle did.



Fig. 5. A view of the Test 7 course from the vehicle (on the wrong side of the barrier).

The ground truth for Test 7 was created from the log file of the first test run. Two different people generated ground truth files. One selected 1 point per frame, resulting in a usable count of 702 points, while the other selected 3 points per frame, resulting in a usable count of 2195 points, where usable points are determined as described above for Test 6. Having different selections of points for the same data set enabled us to see if there was any significant variation between people's selection of labels and also let us see if a smaller number of points was as effective as a larger one.

As can be seen in TABLE II and TABLE III, the results for both the small sample size and the large one are very similar, indicating that it is not necessary to label large numbers of points. What was surprising was that the distribution of the errors was different. For the smaller set, the percentage of errors due to the learning algorithm not being able to identify the class of the point was 46%, whereas the corresponding percentage for the larger set was 71%. In the tests we have done, the distributions of errors with different random sets of points has not shown any obvious pattern.

TABLE II
Results for Test 7, User 1

Test 7, 702 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
592	110	84.5%	15.5%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
47%	34%	19%	

TABLE III
Results for Test 7, User 2

Test 7, 2195 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
1884	312	85.8%	14.2%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
71%	4%	25%	

C. Test 9

Test 9 was conducted in the desert in December, 2005. The terrain was vegetated with both woodland and grassland features. The vegetation was dry, and there was not much color difference between the vegetation and the ground (Fig. 6). The course ran along a mowed path through the terrain, but

there were other paths crossing the desired path which did not provide a traversable route to the goal. The Government test team expected the vehicles to explore the side paths on the first run, but learn that they were not productive and follow the preferred path on later runs. This is what the NIST vehicle did.



Fig. 6. A view of the terrain in Test 9.

The ground truth for Test 9 was created from the log file of the first run, using a single point from each frame and a total of only 176 frames. There were a total of 290 points to be classified. As can be seen in TABLE IV, the system performed a little worse in this low-color environment, but still respectably.

TABLE IV
Results for Test 9

Test 9, 290 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
232	58	80.3%	20.1%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
19%	21%	60%	

D. Cumulative Results

The results of all the performance evaluations are accumulated in TABLE V. As can be seen, 86% of the time the algorithm assigns similar labels to regions as do human observers.

TABLE V
Cumulative Results

Tests 6, 7, and 9, 5701 Ground Truth Points	
Number of points classified	5701
Number correct	4905
Number incorrect	797
Percentage correct	86%
Percentage incorrect	14%

IV EVALUATING ALGORITHM PARAMETERS

Another way of using the ground truth data is to investigate the effects of the model parameters. We use five parameters, and here we discuss the effects of selecting subsets of these parameters. We explored using only color (no intensity or texture), using color plus intensity with no texture, and not using color. There are two color components, R/G and G/B. We did not explore removing only one of them. Nor did we look at the effects of contrast. Some of the results were surprising.

TABLE VI

Effects on Classification of Changing Model Parameters

Test 7 Model Parameter Variation					
No Texture		No Color		Only Color	
% Correct	% Incorrect	% Correct	% Incorrect	% Correct	% Incorrect
83.52%	16.48%	53.26%	46.79%	86.25%	13.75%
Test 9 Model Parameter Variation					
No Texture		No Color		Only Color	
% Correct	% Incorrect	% Correct	% Incorrect	% Correct	% Incorrect
82.35%	17.99%	76.12%	24.22%	56.40%	43.94%

TABLE VI shows the classification success of the algorithm when it learns models with one or more features removed. It appears that removing texture has hardly any effect. The percentage of correct classifications for Test 7 goes down marginally (just over 2%), but the correct classification for Test 9 goes up (about 2%)! This is very surprising, since the data for Test 9 showed little color variation, so we assumed that the texture was providing most of the discrimination. It probably means that the texture measure we used is not suitable for this application (perhaps because it uses such a small neighborhood).

On the other hand, taking color out of the model features has a big impact, dropping the classification accuracy in Test 7 from about 86% to 53%. For Test 9 the accuracy also drops, but only from 80% to 76%. This is reasonable, since the data showed so little color variation.

Finally, if only color is used, the performance on Test 9 degrades considerably, from 80% to 56%. The performance on Test 7 actually goes up marginally, although probably not significantly. We can conclude that intensity plays a significant role in classification, especially in Test 9. Color is clearly important, but the use of the Local Binary Pattern operator is questionable.

V. CONCLUSION

Knowing the traversability of terrain is very important to a robot that navigates off-road like those in the DARPA LAGR project. At NIST, we have developed several methods of learning traversability for use in the LAGR program. In this paper, we discussed our method of evaluating the performance of an algorithm that learns to classify terrain as either traversable or not traversable based on models it builds using color and texture features of the terrain.

The performance evaluation is not specific to the particular algorithm shown in this paper. Once a human has generated a set of ground truth points, they can be used to evaluate any classification algorithm. It is straightforward to modify the number of classes the user has available to classify the points, although too many classes may lead to a higher rate of human error in classifying the points. The evaluation was also applied to the stereo obstacle detection algorithm that provides the input for the learning algorithm and in some sense determines the best performance that can be expected of it. The results showed that the obstacle detection algorithm agreed with human classification 91% of the time.

The random nature in which the points to be classified are selected has the advantage of preventing any bias in the way that the image sequence is sampled. It has a problem, however, in that it is not possible to say anything about the way the errors are distributed in the images. There is a significant difference between errors that congregate at the boundaries of regions and those that appear throughout the image. Usually, errors close to boundaries are less of a concern since they amount to a disagreement about where the boundary actually occurs. Thus, two algorithms with the same performance in terms of correct classifications could differ greatly in their utility. The method used in this paper cannot provide a distinction based on error locations, but a quick scan of images such as Fig 2 gives a good idea of the error distribution.

It should be pointed out that the results shown in this paper do not take into account some postprocessing that is done in the algorithm after an image frame is classified but before the results are sent to the planner. This involves removing singleton blocks (16x16 windows of pixels) classified as one type that lie within a region of the opposite type (e.g., a single non-traversable block within a traversable region as can be seen in Fig 2). Usually such blocks are the result of incorrect classification so removing them improves the overall performance of the algorithm. In one of the tests (Test 10), however, the vehicles had to make their way through a set of thin posts randomly placed in a field. By removing singleton

blocks, the locations of some of the posts that had been correctly recognized by the algorithm as not traversable were lost.

It is very helpful to be able to use the performance evaluation to tune the algorithm by determining the useful features and their relative contributions to the final classification. Our evaluation showed that the texture operator was not performing effectively and that using intensity as a feature is beneficial. We plan to explore alternative texture measures based on multiresolution Gabor filters as in [10] to see if they perform better.

Overall, the results show that the algorithm for learning traversability works well, with a high degree of agreement between its classifications and those of a human observer. This provides confidence that the algorithm will enhance the performance of the LAGR control system as a whole.

ACKNOWLEDGEMENT

The work described in this paper was conducted under a grant from the DARPA LAGR program. We are grateful for their support.

REFERENCES

- [1] Jackel, L. Learning Applied to Ground Robots (LAGR). <http://www.darpa.mil/ipto/programs/lagr/>. 2005.
- [2] J. S. Albus, H-M Huang, E. Messina, K Murphy, M Juberts, A. Lacaze, S. Balakirsky, M. O Shneier, T. Hong, H. Scott, J. Horst, F Proctor, W. Shackleford, S. Szabo, and R. Finkelstein, "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," National Institute of Standards and Technology, Gaithersburg, MD, NISTIR 6912, 2002.
- [3] J. Albus, R. Bostelman, T. Chang, T. Hong, W. Shackleford, and M. Shneier, "Learning in a Hierarchical Control System: 4D/RCS in the DARPA LAGR Program," *Journal of Field Robotics, Special Issue on Learning in Unstructured Environments (in press)*, 2006.
- [4] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Place Recognition for Topological Localization," IEEE International Conference on Robotics and Automation, 2000, pp. 1023-1029.
- [5] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," Proceedings of the AAAI National Conference on Artificial Intelligence, 2000.
- [6] Ceryen Tan, Tsai Hong, Michael Shneier, and Tommy Chang, "Color Model-Based Real-Time Learning for Road Following," Proceedings of the IEEE Intelligent

Transportation Systems Conference, 2006.

- [7] Carl Wellington and Anthony Stentz, "Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation," International Conference on Field and Service Robotics, 2003, pp. 49-54.
- [8] Ayanna Howard, Edward Tunstel, Dean Edwards, and Alan Carlson, "Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability," Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001, pp. 7-12.
- [9] A. Shirkhodaie, R. Amrani, N. Chawla, and T. Vicks, "Traversable Terrain Modeling and Performance Measurement of Mobile Robots," Performance Metrics for Intelligent Systems, PerMIS '04, 2004.
- [10] A. Talukder, R Manduchi, R. Castano, L. Matthies, A. Castano, and R. Hogg, "Autonomous Terrain Characterisation and Modelling for Dynamic Control of Unmanned Vehicles," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002, pp. 708-713.
- [11] T. Chang, T. Hong, S. Legowik, and M. Abrams, "Concealment and Obstacle Detection for Autonomous Driving," Proceedings of the Robotics & Applications Conference, Santa Barbara, CA, 1999, pp. 147-152.
- [12] Shneier.M., T. Chang, T. Hong, and W. Shackleford, "Learning Traversability Models for Autonomous Mobile Vehicles," *Autonomous Robots (submitted)*, 2006.
- [13] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29 1996, pp. 51-59.

Quantitative assessments of USARSim accuracy

S. Carpin, T. Stoyanov, Y. Nevatia
School of Engineering and Science
International University Bremen
Bremen, Germany

M. Lewis, J. Wang
Dept. of Information Sciences and
Telecommunications
University of Pittsburgh
Pittsburgh, USA

Abstract—Effective robotic simulation depends on accurate modeling of physics and the environment as well as the robot, itself. This paper describes validation studies examining feature extraction, WaveLan radio performance, and human interaction for the USARSim robotic simulation. All four feature extraction algorithms showed strong correspondences between data collected in simulation and from real robots. In each case data extracted from a well lit scene produced a closer match to data extracted from a simulated image than to camera data from a poorly lit scene. The radio simulation also performed well in validation showing levels of attenuation due to intervening walls that were similar to signal strengths measured in the modeled environment. The human-robot interaction experiments showed close correspondence between simulator and robots in performance affected by robot model, control mode and task difficulty.

I. INTRODUCTION

USARSim is a high fidelity robot simulator built on top of a commercial game engine [1] with a wide range of possible applications. USARSim is currently being used to investigate human robot interfaces (HRI), to develop and tune robot algorithms, and to study cooperative behaviors. USARSim has recently been adopted by the Robocup Federation [2][3] as the software infrastructure for a new Urban Search and Rescue (USAR) competition that models robots and environments from the USAR Robot League. It joins an earlier Robocup Rescue simulation that focuses on a higher level of logistics and emergency management. Although robot simulators have been widely used since the field's inception there remain widespread reservations about their usefulness. There are a variety of reasons behind these concerns. First, robot simulators have often offered application program interfaces that were inconsistent with those found on real robots. This made it difficult to move software between robot and simulator for code development and debugging which was often the primary purpose for using simulation. This problem has been largely overcome by hardware neutral middleware such as the widely used player/stage software [4][5]. A more damaging criticism concerns discrepancies that may be found between results obtained from simulation and those obtained with real robots. A prime tenet of modern behavior-based robotics [6] is that effective systems can be designed by eliminating internal representations and focusing instead on the direct relation between stimulus and action [7]. From this perspective a good simulation must simultaneously supply an accurate model of the robot's geometry and kinematics, accurate models of

sensors, an accurate model of the environment, and an accurate model of the robot's interaction with that environment. If any one of these constituents breaks down the simulation can no longer provide an adequate model of the process being studied. Simulation requirements were far more relaxed for an earlier generation of robots that relied on planning and many robot simulators still provide only schematic or 2D models of the environment and pay little attention to the physics of the interaction between robot and environment. USARSim, by contrast, provides detailed models of both the environment and the physics of interaction making accurate simulation for behavior-based robotics a possibility.

In this paper we provide a quantitative evaluation of the accuracy of USARSim, paying particular attention to the validation of robot performance, as well as the perceptual processes. Specifically, we define a set of perceptual tasks to be studied both in simulation and in reality, as well as metrics to compare the obtained results. The goal is to provide quantitative indices that indicate to which degree it is possible to extrapolate results obtained in simulation. Additional validation data are reported for disruption of radio communications and human control of robots. The overall USARSim architecture is described in section II, with an emphasis on the specific components devoted to perception and action. One of the tasks more relevant in mobile robotics is visual perception. Section III presents a set of algorithms commonly used for robotics oriented image processing, as well as performance indices. In multi-robot systems, inter-robot communications based on wireless channels play a relevant role, but up to now few simulators explicitly model aspects like signal degradation and the like. These topics are addressed in section IV. Section V presents data for two robots controlled by operators using two control modes showing correspondences in behavior between simulated and real robots. Finally, conclusions are offered in section VI.

II. USARSIM SOFTWARE ARCHITECTURE

USARSim uses Epic Games' UnrealEngine2 to provide a high fidelity simulator at low cost. The current release consists of models of standardized disaster environments, models of commercial and experimental robots, and sensor models. USARSim also provides users with the capability of building their own environments and robots. Its socket-based control API was designed to allow users to test their own control algorithms and user interfaces without additional pro-

gramming. USARSim includes detailed models of the NIST Reference Test Arenas for Autonomous Mobile Robots [8] and offers the possibility of providing more realistic challenges and significantly larger disaster environments.

The official release of USARSim available from (www.sourceforge.net/projects/usarsim) currently provides detailed models of eight robots including both experimental and commercial robots widely used in USAR competition. These models were constructed using the Karma physics engine [9], a rigid body simulation that computes physical interactions in realtime. A hierarchy of sensor classes have been defined to simulate sensor data. Sensors are defined by a set of attributes stored in a configuration file, for example, perception sensors are commonly specified by range, resolution, and field-of-view.

The scenes viewed from the simulated camera are acquired by attaching a spectator, a special kind of disembodied player, to the robot. USARSim provides two ways to simulate camera feedback: direct display and image server. Direct display uses the Unreal Client, itself, for video feedback, either as a separate sensor panel or embedded into the user interface. While this approach is the simplest, the Unreal Client provides a higher frame rate than is likely to be achieved in a real robotic system and is not accessible to the image processing routines often used in robotics. The image server intermittently captures scenes in raw or jpeg format from the Unreal Client and sends them over the network to the user interface. Using the image server, researchers can tune the properties of the camera, specifying the desired frame rate, image format, noise, and/or post processing needed to match the camera being simulated.

III. VALIDATION OF VISION IN USARSIM

Vision is one of the richest perceptual sources for both autonomous and remotely operated robots. A realistic simulator cannot therefore omit a realistic and quantitatively precise video simulation component. Within USARSim, video input is produced by directly grabbing images from the scene rendered by the visualization component of the game engine. Frames are provided to the robotic controller encoded as jpegs of different quality and with different resolutions. We have implemented four different image processing algorithms that require the fine tuning of several parameters. The parameter fine tuning phase has been performed exclusively in simulation and then the same algorithms have been run on real world images, to outline similarities and differences in performance.

A. Feature extraction algorithms

The four visual tasks implemented are described in the following subsections.

1) *Edge detection*: Edge detection has been implemented using the well known Canny edge detection operator. Given a grey scale picture, the image is first filtered with a Gaussian filter to remove noise. Then, a Sobel operator separates regions of high horizontal or vertical frequencies. Finally, the Canny operator is applied, leaving lines with a 1 pixel thickness, and

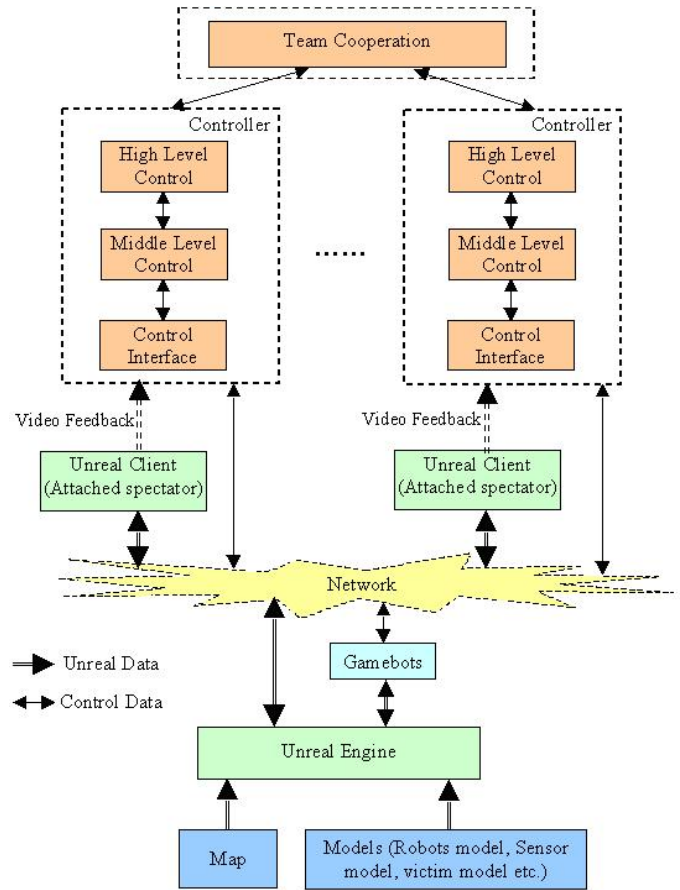


Fig. 1. System Architecture

a thresholding final pass provides a black and white image. Figure 2 illustrates these four steps.

2) *Template matching*: Template matching consists in finding whether (and where) a known given target template appears within a wider image. Template matching is very useful, for example, when beacons are scattered in the environment to help the robot recover from localization errors. For this operation, a simple template correlation was used. First, the two dimensional Fourier transform of the image is computed. Then the template image is transposed and padded to the size of the image. Next, the Fourier transform of the template is taken and multiplied with the transform of the image. The inverse transform of the result provides an image of the template convolution. We take the transpose of the template instead of the template itself because the algorithm needs to obtain the correlation of the two images and not the convolution. An example is show in figure 3. On the left is the template, followed by the inverted Sobel of the image and the final result. The darker regions are the locations in the image where the template is most probably located. In this example there are two distinct peaks, close to each other. Such variations occur when the size of the template does not exactly match that of the feature in the image, as this algorithm is not scale invariant. The usual practice to obtaining

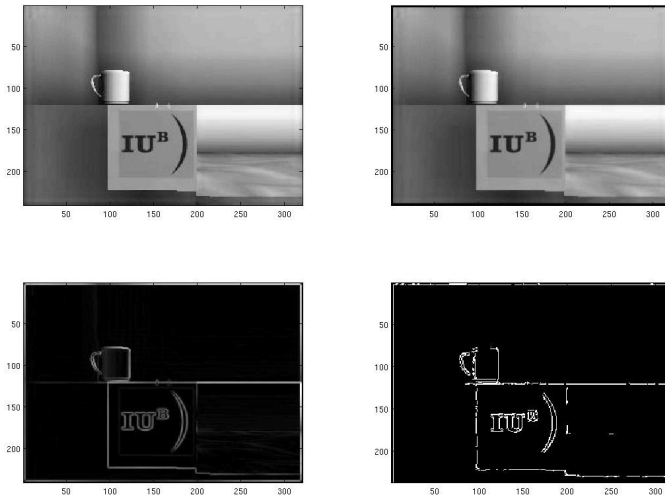


Fig. 2. The steps of the Canny edge detection operator

a scale invariant implementation involve generating a pyramid of possible templates of different sizes. A similar technique is used for obtaining rotation invariance, although in this case the problem is more complicated, due to the interpolation errors that occur when a digital image is rotated.

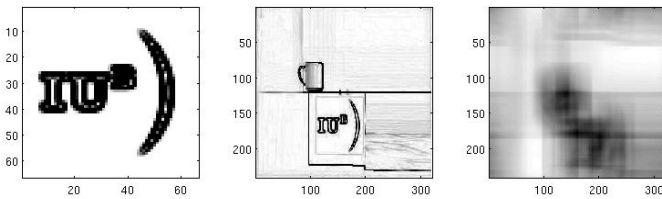


Fig. 3. Template matching. Picture of template (left), image with target feature(middle) and correlation(right)

3) *Snakes*: Active contours, also known as snakes, are one of the best performing feature extraction techniques available. The idea is the following: start with a number of points that encompass the target feature. The points form a contour with total energy

$$E_{snake} = \int_{s=0}^1 E_{int}(v(\mathbf{v})) + E_{im}(v(\mathbf{v})) + E_{con}(v(\mathbf{v})) ds \quad (1)$$

where E_{int} is the internal energy of the contour, E_{im} is the energy component from the image and E_{con} is the constraint energy. The internal energy is implemented as the average distance between each two neighboring snake points, the constraint energy is the curvature of every three consecutive snake points and the image energy is proportional to the value of the pixel that the snake point is currently occupying. On each iteration of the algorithm the snake points are moved to minimize the snake energy and eventually shrink the contour to that of the targeted feature. There are several methods to solve rigorously and implement the continuous solution of the snakes algorithm in a discrete space. We have embraced the solution known as *the greedy snakes algorithm*, that performs

a greedy search on points in the vicinity of each snake point. It computes a discretized version of equation 1 for each pixel in a 3 by 3 neighborhood and moves the snake point to the pixel that has the lowest value for E_{snake} . For the purposes of this algorithm the image energy is computed as the value of every pixel in a normalized, inverted Sobel edge transform of the original image. This implementation has a few inherent problems that sometimes lead to a complete failure of the algorithm. The first, and most serious shortcoming is that the snake points can get stuck at local minimums and stop moving. In general this is not that frequent, as if only one point moves this will likely trigger motion of other points and thus move the whole snake. To prevent cases when all points are stuck we have increased the size of the search window from 3 by 3 to 9 by 9 pixels, which has no considerable effect on the execution time, as the number of snake points is generally low. The second problem concerns the choice of weighting coefficients for each of the three components of the snake energy. Choosing a high value for the image energy makes snake points migrate to the closest edges and distort the original shape of the contour. Choosing too low a value on the other hand makes the contour static, because even small changes in the spacing between points and in the curvature have a huge impact on the total energy. Thus, choosing the proper constants becomes a tedious process that is specific for each image analyzed. Over a few tests constants that have a stable performance on the simulated images were chosen, again with the purpose of testing how well the tweaked algorithm would later perform on the real images.

4) *Optical character recognition*: Optical Character Recognition (OCR) is the problem of extracting text from raster images of text. There exist different algorithms to perform OCR. The one described here starts by properly aligning the text, so that all rows are parallel to the horizontal axis. This is achieved by computing the Hough transform of the text image and rotating it around an angle, equal to the most frequent Hough angle. Assuming we have a long enough text, all parallel lines that belong to characters will intersect in Hough space and thus the angle of rotation can be determined. The next step is to compute the vertical projection of the image and separate each element. This is possible, because of the white spaces between rows which are distinctly visible on the vertical projection of the image. Using a similar argument, we can compute the horizontal projection of each row and separate letters, also called *glyphs*. Individual letters are then cropped to ensure there are no extra white spaces. This algorithm is first performed on a learning image, which contains the whole character set to be recognized, in a known order. Thus, a database of characters and their respective glyphs is created. Subsequent text images are processed in the same way and for each character glyph a template matching is performed to find the character from the database that has the greatest similarity. An example is shown in figure 4: the original image, the image after thresholding and inverting, after rotation and after applying the noise reduction filter are displayed in sequence. This algorithm achieves a 100%

accuracy on images grabbed from the screen, but is susceptible to noise, as stray pixels, unless filtered, will be recognized as glyphs and matched against the database. The noise reduction filter was implemented to reduce salt and pepper noise and stray single pixels, but that has no effect on groups of noisy pixels. Filtering out such noise is very hard, as it is sometimes impossible to differentiate between a cluster of noisy pixels and a valid character.

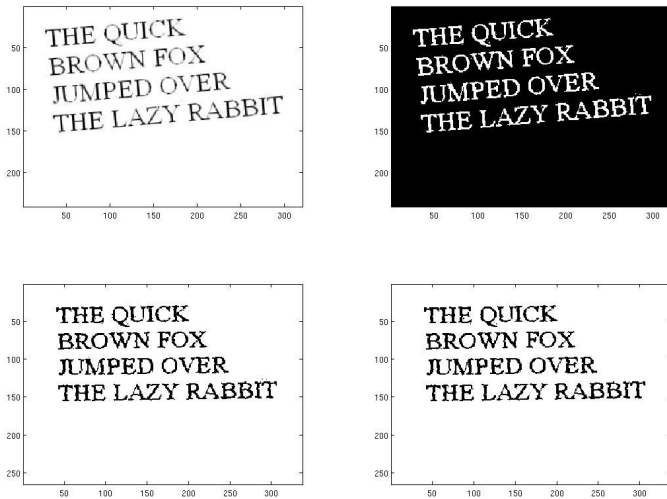


Fig. 4. The stages of Optical Character Recognition

All the above described algorithms have been implemented in Matlab closely following the descriptions found in [10] and [11].

B. Experimental setup and results

In order to compare the algorithm performance on corresponding simulated and real images, we have developed within USARSim a detailed model of a room environment and we have successively taken pictures from corresponding points of the virtual and real world. In order to test the algorithms under different boundary conditions, images with different light conditions were used.

Figure 5 presents the correlations between the edge images for eight test images. The autocorrelations of the simulated image in column 1 (dark blue) are comparable with those from the correlation between a well lit real world image and a simulated image (column 2, light blue). The same is true in most cases about the correlations of the simulation and the bad lit image, compared to the correlations of the well lit and bad lit image (columns 3 and 4, yellow and brown respectively). The slight deviations are mainly due to minor deviations of the positions of the camera when taking the images. It should be observed that the precise numerical value of the correlation is not the main aspect of this experiment. The relevant aspect is rather the gross scale similarity or discrepancy in the values.

Figure 6 presents the results for the distances (in pixels) between the actual position of the target feature (IUB logo displayed in figure 3 on the left) and the position estimated with template convolution. Except for the third and the seventh

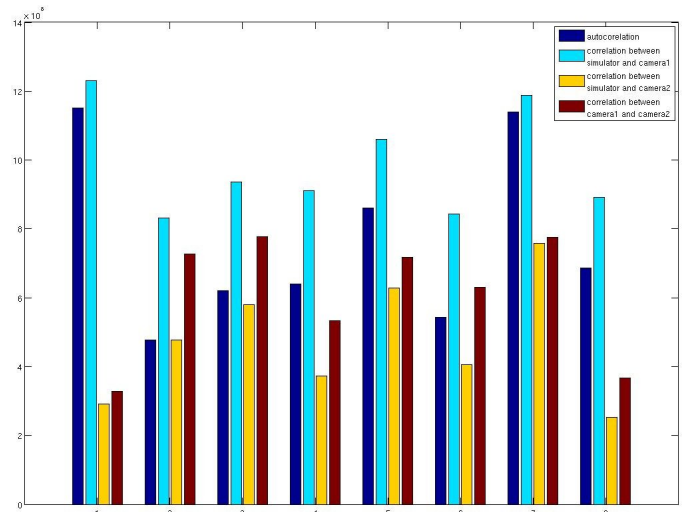


Fig. 5. Results for edge detection metric

image, the distances are below 100 pixels, which is about one and a half times the template size and a very good result. In most of the cases the results on the three images are very close, with the noticeable difference of image 6, where the well lit real image shows a much worse behavior than the other two. In most cases however, the performance is almost identical, as visual inspection of figure 7 (test image 1) confirms.

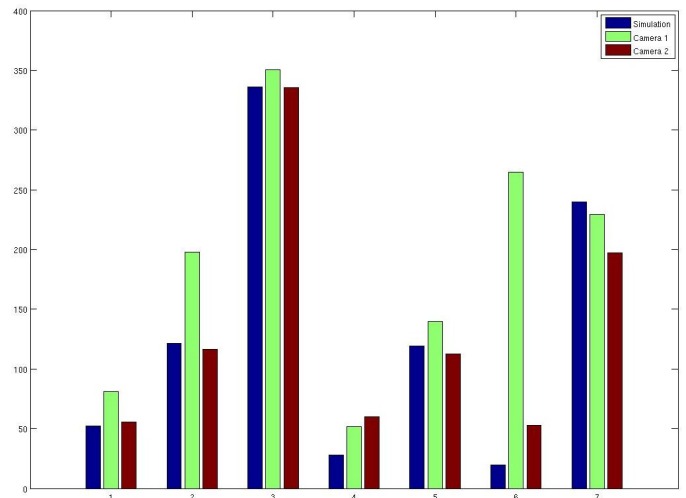


Fig. 6. Results for template convolution metric for simulation (blue), well lit conditions (green) and bad lit conditions (brown).

Figure 8 shows the average distance in pixels between snake points and target features for the three sets of images. The results show a maximum deviation of about 11 pixels, which is a good result, as well as some excellent performances on images 4 and 6 with average distance of about 3-4 pixels. The results for Image 6 are also presented in figure 9 (simulation) and figure 10 (real-world). Again, the performances on the three sets are comparable, and although the constants have been tweaked for the simulation, the real images sometimes outperform the simulated ones.

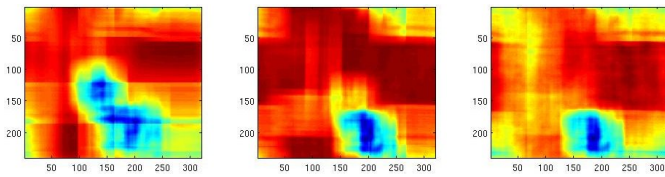


Fig. 7. Template convolution performed on simulator(left) and real world(right,middle)

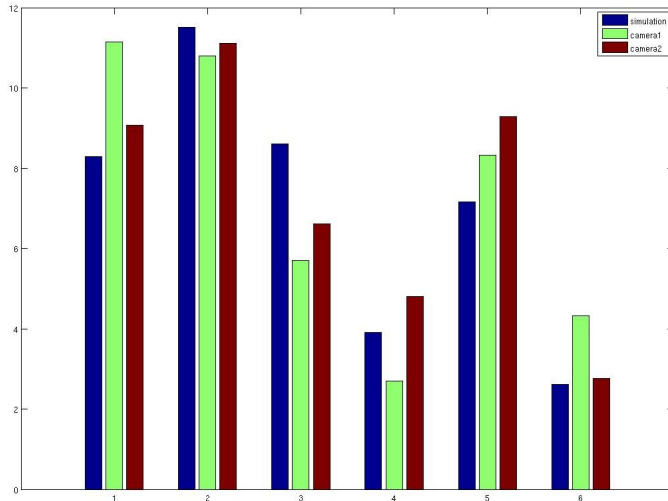


Fig. 8. results for the Active Contours metric

Finally, figure 11 presents the results of testing optical character recognition on two sets of images - one from the real model and one from the simulation. The figure measures roughly the percentage of recognized characters in each case. The success rate in both cases is pretty low, and noticeably lower in the case of the real world images. Inspecting the sample image in figure 12 gives a very good explanation for these low figures, i.e. the high level of noise. The figure shows the original images on the top - simulation on the left and real image on the right, as well as the images after filtering and rotation on the bottom. The bottom images exhibit a low quality and high fragmentation on the characters. This is due to the rigorous filtering that has removed most of the noise, but also parts of the characters. As the images from the real camera exhibit higher level of noise, they also have a lower quality after filtering and thus a lower success rate of recognition.

IV. WIRELESS SIMULATION

An important factor in the performance of multi robot teams is the communication between the agents. In complex environments offering little or no opportunity for implicit information exchange, explicit communication can greatly improve the performance of multi-agent teams. USARSim currently does not provide any kind of simulation of communication mechanism, thus allowing all robots to freely communicate regardless of their position in the environment. To include a more realistic scenario in future USARSim releases, we have developed and validated a preliminary software module that mimics wireless

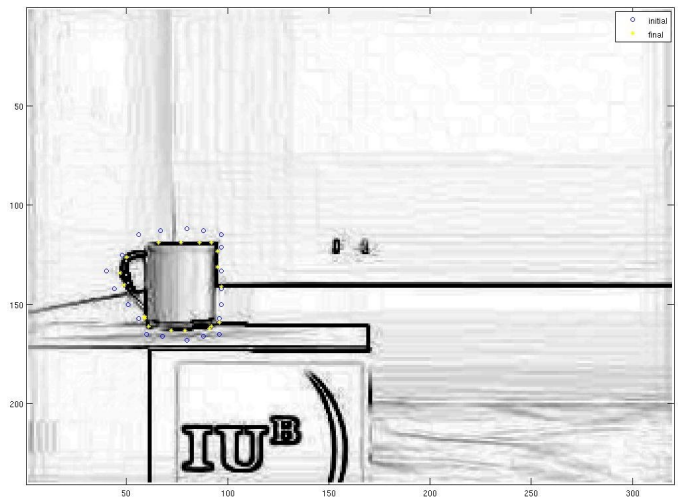


Fig. 9. Snake algorithm performed on simulated image

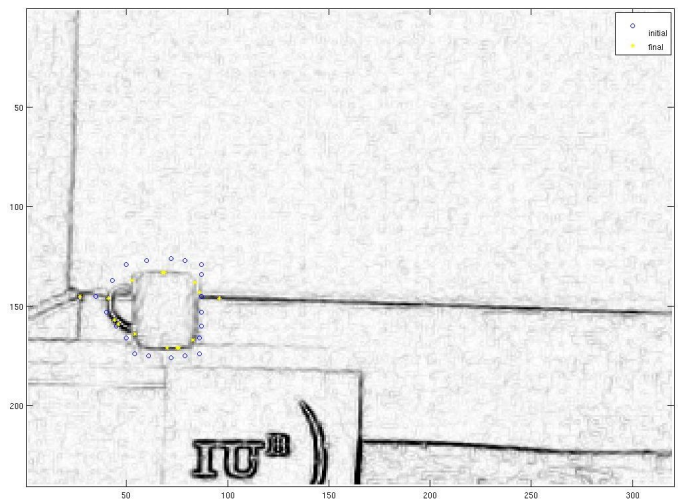


Fig. 10. Snake algorithm performed on real image

communication within simulated environments. As nowadays most robots use WaveLan cards to send messages to each other over wireless channels, the implementation of a WaveLan simulator for USARSim will greatly improve its accuracy as a tool to develop multi robot teams, hence making it even more attractive for the research community.

The simulation system consists of three modules. A so called *parser* component provides the infrastructure to compute the strength of a signal received by a receiver. A *server* component is used to dispatch messages from transmitters to receivers. Therefore if a the process controlling the simulated robot *A* desires to send a message to the process controlling the simulated robot *B*, it does not directly talk to it, but it rather asks the *server* to deliver a message. The server, upon inspection of the receiver signal strength, decides whether the message should be passed on or not. The third component, which will not be extensively described here, provides a one-to-one simulation of the socket API, so that communication software written within the simulator can be easily moved to

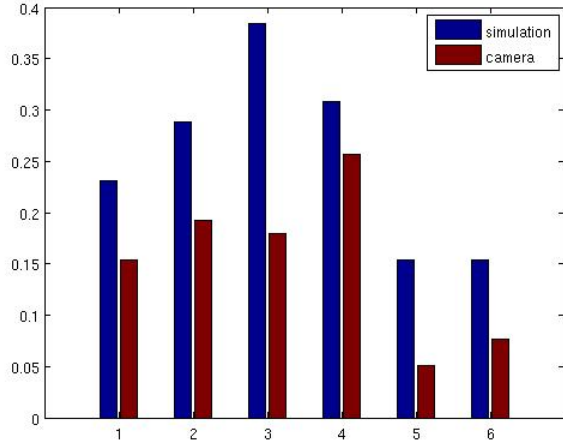


Fig. 11. Results of OCR metric

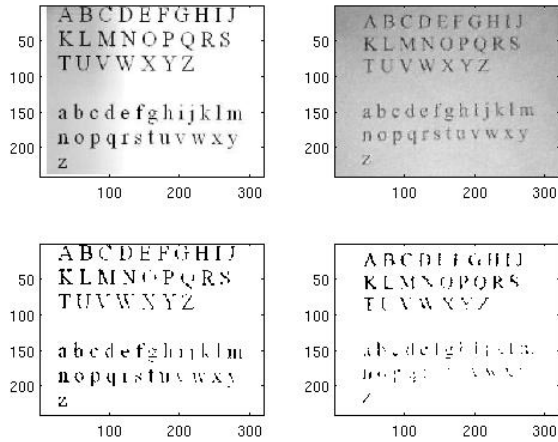


Fig. 12. OCR performed on simulator(left) and real world(right)

real robots.

A fundamental step for the simulation of wireless signals is the selection of a propagation model, i.e. a model describing how signals are propagated in the environment. Among the different ones proposed in the literature, we have selected the one presented in [12], also known as RADAR model. The model best predicts propagation within floors, accounting for the attenuation of the transmitted signal due to distance and traversed walls. The signal strength at a point at distance d from the emitter is modeled by the following equation

$$P(d) = P(d_0) - 10n \log\left(\frac{d}{d_0}\right) - \begin{cases} nW * WAF & nW < C \\ C * WAF & nW \geq C \end{cases} \quad (2)$$

$P(d_0)$ is the reference signal strength in dBm, nW is the number of obstructions between the transmitter and the receiver, and WAF is the so called *Wall Attenuation Factor*, i.e. an empirical factor accounting for the attenuation experienced

by the signal while traversing a wall. C is the maximum number of obstructions up to which the attenuation factor affects the path loss. Finally, n is a factor indicating the rate with which the path loss increases with distance. It is therefore evident that if one wants to use equation 2 to predict the received signal strength, it is necessary to know the relative positions between the transmitter and the receiver, as well as the number of walls. This later number, needed to determine the right nW value, is not computed on the fly every time the value for $P(d)$ is needed, but is rather deducted from a data structure obtained by preprocessing once the map of the environment. The preprocessing operation is performed by the *parser* subsystem. According to the technical specifications of commercially available wireless devices the minimum receiver sensitivity is -92dBm. Therefore when the *server* receives a request for a message to be dispatched, it passes it on only if the received signal strength is above this value.

A. Testing and validation

In order to evaluate the performance of the proposed wireless simulation system we have developed within Unreal the model of an existing building. The environment features three fixed base stations that can be modeled as well within the proposed framework. A preliminary step has been the experimental determination of the parameters in equation 2. These values are displayed in table I.

Parameter	Value
Wall attenuation factor (WAF)	7
Maximum number of obstructions (C)	4
Path Loss factor (n)	1
Reference distance (d_0)	2
Signal strength at d_0 (dBm)	-50

TABLE I

EXPERIMENTALLY DETERMINED PARAMETERS

Next, for different placements of transmitters and receivers we have

- measured the actual signal strength in the environment
- computed the value predicted by equation 2
- computed the signal strength with the simulation system.

The results of these measurements and predictions are displayed in tables II, III and IV respectively.

Name	No of Walls [m]	Average [dBm]	Median [dBm]	3rd Quartile [dBm]
rtest1	1	-71.43	-71.2	-69.03
rtest2	1	-74.2	-74.05	-71.52
rtest3	0	-66.65	-67.04	-64.18
rtest4	2	-78.48	-77.7	-75.91

TABLE II

WIRELESS SIGNAL STRENGTH PREDICTED BY WAF MODEL

It can be observed that there is in general a good correspondence between the two predictions and the measured signals, although there are some obvious fluctuations. Large

Name	Average [dBm]	Std Dev [dBm]	Median [dBm]	3rd Quartile [dBm]
rtest1	-72.18	6.37	-68	-67
rtest2	-70.85	2.47	-71	-70
rtest3	-66.97	7.44	-63.5	-61
rtest4	-73.95	1.34	-74	-73

TABLE III
EXPERIMENTAL VALUES FOR WIRELESS SIGNAL STRENGTH

Name	Average [dBm]	Median [dBm]	3rd Quartile [dBm]
rtest1	-71.33	-71.09	-68.93
rtest2	-81.12	-80.42	-78.29
rtest3	-73.76	-73.6	-71.12
rtest4	-78.25	-77.46	-75.71

TABLE IV
WIRELESS SIGNAL STRENGTH VALUES FROM THE SIMULATOR

discrepancies between the results predicted by the simulator and those forecasted by the RADAR module are explained by the approximations introduced by the *parser* module.

V. HUMAN ROBOT INTERACTION

Validating USARsim for human-robot interaction (HRI) presents a complex problem because the performance of the human-robot system is jointly determined by the robot, the environment, the automation, and the interface. Because only the robot and its environment are officially part of the simulation, validation is necessarily limited to some particular definition of interface and automation. If, for example, sensor-based drift in estimation of yaw were poorly modeled it would not be apparent in validation using teleoperation yet could still produce highly discrepant results for a more automated control regime. Our validation efforts for HRI, therefore, sample two widely used control schemes [13], teleoperation and point-to-point control for two robots, the experimental PER [14] and the commercial Pioneer P2-AT (simulation)/P3-AT (robot) in order to provide an indication of the likely validity of the simulation for HRI across a range of configurations.

We have completed validation testing at Carnegie Mellon's replica of the NIST Orange Arena for the PER robot using both point-to-point and teleoperation control modes reported in [15] and have collected teleoperation data for the Pioneer reported in [3]. In these tests robots were run along a narrow corridor in either the simulation or the Orange Arena with three types of debris (wood floor, scattered papers, lava rocks) while the sequence, timing and magnitude of commands were recorded. In the first three trials, participants had to drive approximately three-meters, along an unobstructed path to an orange traffic cone. In the next three trials, obstacles were added to the environments, forcing the driver to negotiate at least three turns to reach the cone yielding a between groups design pairing each surface type with straight and complex paths.

The paper surface had little effect on either robot's operation. The rocky surface by contrast had a considerable impact, including a loss of traction and deflection of the robot. This was reflected by increases in the odometry and number of turn commands issued by the operators even for the straight course. A parallel spike in these metrics is recorded in the simulator data. As expected the complex course also led to more turning even on the wood floor. Figure 13 shows task times for real and simulated robots. Differences within conditions were low particularly for complex paths which are more likely to be influenced by human control suggesting that USARSim is likely to provide a valid tool for investigating HRI.

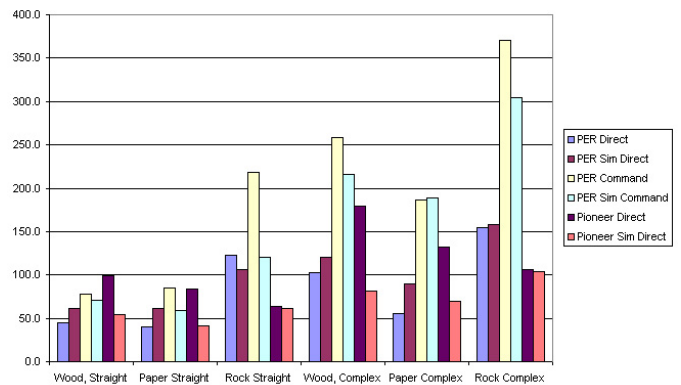


Fig. 13. Task Duration

The one metric on which the simulation and the physical robot consistently differed was proximity to the cone when teleoperating the PER (14). Operators using the physical robot reliably moved the robot to within 35cm from the cone, while the USARSim operators were usually closer to 80cm from the cone. It is unlikely that the simulation would have elicited more caution from the operators, so this result suggests that there could be a systematic distortion in depth perception, situation awareness, or strategy.

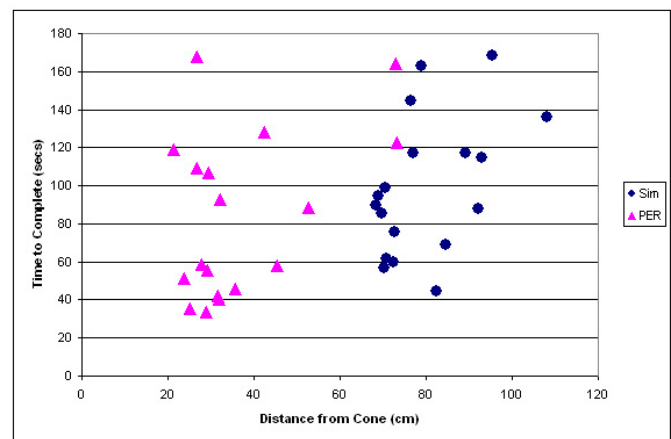


Fig. 14. Approach to Cone for Teleoperated PER

VI. CONCLUSIONS

This paper describes validation tests for feature extraction from simulated images, a radio propagation model, and tests involving human control. The feature extraction tests are especially important to validating the simulator because of the complexity of the visual imagery. The underlying game engine was explicitly designed to generate imagery that would appear realistic to human perception. This is, however, no guarantee that the information extracted from synthetic images would correspond to that extracted from real camera views. In fact, the clarity and lack of naturally occurring distortion in synthetic images might be expected to yield perfectly formed extractions where nothing might be found even in clear appearing real images. Our results are very encouraging because they show a close correspondence between information extracted from real and computer generated images at least under well lit conditions. Further validation will be required to determine whether this correspondence will extend to other illumination levels and extraction algorithms. The radio simulation, by contrast, provides a validated tool for approximating communications difficulties at USAR tasks for use with the simulator but does not reflect on the validity of the simulator itself. The driving tests showed that robots in simulation behaved in much the same way as real robots. The correspondence in performance for robots and simulation between control modes, terrain type, and task complexity suggest that the simulation is both physically accurate and presents similar challenges to human operators making it an appropriate tool for HRI research.

To draw valid conclusions from robotic simulations it is important to know the metrics which are consistent with the operation of the actual robot and those which are not. By collecting validation data for all entities within the simulation we hope to create a tool with which researchers can pick and choose manipulations and metrics that are likely to yield useful results. As our library of models and validation data expands we hope to begin incorporating more rugged and realistic robots, tasks and environments. Accurate modeling tracked robots which will be made possible by the release of UnrealEngine3 would be a major step in this direction.

REFERENCES

- [1] J. Wang, M. Lewis, and J. Gennari, "Usar: A game-based simulation for teleoperation," in *Proceedings of the IEEE International conference on systems, man and cybernatics*, 2003, pp. 493–497.
- [2] S. Carpin, J. Wang, M. Lewis, A. Birk, and A. Jacoff, "High fidelity tools for rescue robotics: results and perspectives," in *Robocup 2005: Robot Soccer World Cup IX*, ser. LNCS, 2006, pp. 301–311.
- [3] S. Carpin, M. Lewis, J. Wang, S. Balakirski, and C. Scrapper, "Bridging the gap between simulation and reality in urban search and rescue," in *Robocup 2006: Robot Soccer World Cup X*, ser. LNCS.
- [4] R. Vaughan, B. Gerkey, and A. Howard, "On device abstractions for portable, reusable robot code," in *Proceedings of the IEEE/RSJ IROS*, 2003, pp. 2421–2427.
- [5] "Player/stage project," <http://playerstage.sourceforge.net>, 2005.
- [6] R. Brooks, "A robust layered control systems for mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.
- [7] —, "Intelligence without reason," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1991, pp. 569–595.

- [8] A. Jacoff, E. Messina, and J. Evans, "Experiences in deploying test arenas for autonomous mobile robots," in *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS)*, 2001.
- [9] Mathengine, *Karma User Guide*. [Online]. Available: <http://udn.epicgames.com/Two/KarmaReference/KarmaUserGuide.pdf>
- [10] M. Nixon and A. Aguado, *Feature extraction and image processing*. Newnes press, 2002.
- [11] J. Parker, *Algorithms for image processing and computer vision*. Wiley Computer Publishing, 1997.
- [12] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM (2)*, 2000, pp. 775–784.
- [13] J. Crandall, M. Goodrich, D. Olsen, and C. Nielsen, "Validating human-robot interaction schemes in multi-tasking environments," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, no. 33(3), pp. 325–336, 2003.
- [14] I. Nourbakhsh, E. Hamner, E. Porter, B. Dunlavey, E. Ayoob, T. Hsiu, M. Lotter, and S. Shelly, "The design of a highly reliable robot for unmediated museum interaction," in *2005 IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005.
- [15] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin, "Validating usarsim for use in hri research," in *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting (HFES'05)*, 2005, pp. 457–461.

Feedback and Weighting Mechanisms for Improved Learning in the Adaptive Simultaneous Perturbation Algorithm

James C. Spall (james.spall@jhuapl.edu)

The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 U.S.A.

Abstract—It is known that a stochastic approximation (SA) analogue of the deterministic Newton-Raphson algorithm provides an asymptotically optimal or near-optimal form of stochastic search. However, directly determining the required Jacobian matrix (or Hessian matrix for optimization) has often been difficult or impossible in practice. This paper presents a general adaptive SA algorithm that is based on a simple method for estimating the Jacobian matrix while concurrently estimating the primary parameters of interest. Relative to prior methods for adaptively estimating the Jacobian matrix, the paper introduces two enhancements that generally improve the quality of the estimates for underlying Jacobian (Hessian) matrices, thereby improving the quality of the estimates for the primary parameters of interest. The first enhancement rests on a feedback process that uses previous Jacobian estimates to reduce the error in the current estimate. The second enhancement is based on the formation of an optimal weighting of “per-iteration” Jacobian estimates. Given its basis in the simultaneous perturbation mechanism, the algorithm requires only a small number of loss function or gradient measurements per iteration—independent of the problem dimension—to adaptively estimate the Jacobian matrix and parameters of primary interest. This paper provides the basic idea together with some analytical justification and a small-scale numerical evaluation.

Keywords—Stochastic optimization; Jacobian matrix; root-finding; stochastic approximation; simultaneous perturbation stochastic approximation (SPSA); adaptive estimation.

I. INTRODUCTION

Stochastic approximation (SA) represents an important class of stochastic search algorithms for purposes of minimizing loss functions and/or finding roots of multivariate equations in the face of noisy measurements. This paper presents an approach for accelerating the convergence of SA algorithms through two enhancements to the adaptive simultaneous perturbation SA (SPSA) approach in Spall (2000). This adaptive algorithm is a stochastic analogue of the famous Newton-Raphson algorithm of deterministic nonlinear programming. Both enhancements are aimed at improving the quality of the estimates for underlying Jacobian (Hessian) matrices, thereby improving the quality of the estimates for the primary parameters of interest.

The first enhancement improves the quality of the Jacobian estimates through a feedback process that uses the previous Jacobian estimates to reduce the error. The second enhancement improves the quality via the formation of an

optimal weighting of “per-iteration” Jacobian estimates.

The simultaneous perturbation idea of varying all the parameters in the problem together (rather than one-at-a-time) is used to form the per-iteration Jacobian estimates. This leads to a more efficient adaptive algorithm than traditional finite-difference methods. The results apply in both the gradient-free optimization (Kiefer-Wolfowitz) and stochastic root-finding (Robbins-Monro) SA settings. This paper introduces the basic ideas associated with the two enhancements and presents a small-scale numerical study.

The basic problem of interest will be the root-finding problem. That is, for a function $\mathbf{g}(\boldsymbol{\theta}): \mathbb{R}^p \rightarrow \mathbb{R}^p$, $p \geq 1$, we

are interested in finding a point $\boldsymbol{\theta}$ satisfying $\mathbf{g}(\boldsymbol{\theta}) = \mathbf{0}$. Of course, this problem is closely related to the optimization problem of minimizing a differentiable loss function $L = L(\boldsymbol{\theta})$ with respect to some parameter vector $\boldsymbol{\theta}$ via the equivalent problem of finding a point where $\mathbf{g}(\boldsymbol{\theta}) = \partial L / \partial \boldsymbol{\theta} =$

$\mathbf{0}$. Let $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ be a point satisfying $\mathbf{g}(\boldsymbol{\theta}) = \mathbf{0}$. The stochastic setting here allows for the use of only noisy values of \mathbf{g} and the estimation (versus exact calculation) of the associated $p \times p$ Jacobian matrix $\mathbf{H} = \mathbf{H}(\boldsymbol{\theta}) \equiv \partial \mathbf{g}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}^T$. Note that the Jacobian matrix is a Hessian matrix of L when \mathbf{g} represents the gradient of L . As described in Spall (2000), simultaneous perturbation ideas that are used for gradient estimation in Spall (1992) can also be used for the per-iteration Jacobian matrix estimation as part of an adaptive stochastic approximation algorithm.

Certainly others have looked at ways of enhancing the convergence of SA. A relatively recent review of many such methods is in Spall (2003, Sect. 4.5). For the root-finding setting, Ruppert (1985) and Wei (1987) develop stochastic Newton-like algorithms by forming a Jacobian estimates via finite differences of \mathbf{g} measurements. In the optimization setting (using noisy measurements of L), Fabian (1971) forms estimates of the gradient and Hessian by using, respectively, a finite-difference approximation and a set of differences of finite-difference approximations. This requires $O(p^2)$ loss function measurements for each update of the $\boldsymbol{\theta}$ estimate, which is extremely costly when p is large. There are also numerous means for adaptively estimating a Jacobian (especially Hessian) matrix in special SA estimation settings where one has detailed knowledge of the

Acknowledgments—This work was partially supported by U.S. Navy Contract N00024-03-D-6606.

underlying model (see, e.g., Macchi and Eweda, 1983; Yin and Zhu, 1992); while these are more efficient than the general adaptive approaches mentioned above, they are more restricted in their range of application.

Another approach aimed at achieving Newton-like convergence in a stochastic setting is iterate averaging (e.g., Polyak and Juditsky, 1992). While iterate averaging is conceptually appealing due to its ease of implementation, Spall (2003, Sect. 4.5) shows that iterate averaging often does not produce the expected efficiency gains due to the lag in realizing an SA iteration process that approximately bounces uniformly around the solution. Kushner and Yang (1995) (see also Kushner and Yin, 2003, p. 76) present a method using feedback that is slightly similar to that here to improve iterate averaging in certain cases, but this method will not fundamentally cope with the above-mentioned issue of a lag in the SA iterates. Hence, there is strong motivation to find theoretically justified and practically useful methods for building adaptive SA algorithms based on efficient estimates of the Jacobian matrix.

In particular, in the optimization case, only *four* noisy measurements of the loss function L are needed at each iteration to estimate both the gradient and Hessian for any dimension p . In the root-finding case, *three* noisy measurements of the root-finding function \mathbf{g} are needed at each iteration (for any p) to estimate the function and its Jacobian matrix. Although the adaptive SPSA method is a *relatively* simple approach, care is required in implementation just as in any other second-order-type approach (deterministic or stochastic); this includes the choice of initial condition and the choice of gain (“step size”) coefficients to avoid divergence.

Section II describes the general adaptive SPSA approach, including the form of the per-iteration Jacobian estimates. Section III decomposes the per-iteration estimates to expose the terms comprising the errors in the estimates and Section IV uses this decomposition to present the feedback-based term in the enhanced adaptive recursion. Section V derives the optimal weighting for the feedback-based per-iteration estimates with the aim of reducing the error in the cumulative Jacobian estimates. Section VI is a summary of a numerical study.

II. THE PER-ITERATION JACOBIAN (HESSIAN) ESTIMATE IN THE ADAPTIVE SPSA ALGORITHM

The algorithm here has two parallel recursions, with one of the recursions being a stochastic version of the Newton-Raphson method for estimating $\boldsymbol{\theta}$ and the other being a weighted average of per-iteration (feedback-based) Jacobian estimates to form a best current estimate of the Jacobian matrix:

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \bar{\mathbf{H}}_k^{-1} \mathbf{G}_k(\hat{\boldsymbol{\theta}}_k), \quad \bar{\mathbf{H}}_k = \mathbf{f}_k(\bar{\mathbf{H}}_k), \quad (2.1a)$$

$$\bar{\mathbf{H}}_k = (1 - w_k) \bar{\mathbf{H}}_{k-1} + w_k (\hat{\mathbf{H}}_k - \hat{\Psi}_k), \quad k = 0, 1, 2, \dots, \quad (2.1b)$$

where a_k is a non-negative scalar gain coefficient, $\mathbf{G}_k(\hat{\boldsymbol{\theta}}_k)$ is some unbiased or nearly unbiased estimate of $\mathbf{g}(\hat{\boldsymbol{\theta}}_k)$, $\mathbf{f}_k: \mathbb{R}^{p \times p} \rightarrow \{\text{invertible } p \times p \text{ matrices}\}$ is a mapping designed to cope with possible noninvertibility of $\bar{\mathbf{H}}_k$, $0 \leq w_k \leq 1$ is a weight to apply to the new input to the recursion for $\bar{\mathbf{H}}_k$, $\hat{\mathbf{H}}_k$ is a per-iteration estimate of $\mathbf{H} = \mathbf{H}(\boldsymbol{\theta})$, and $\hat{\Psi}_k$ is the

feedback-based adjustment that is aimed at improving the per-iteration estimate. The two recursions above are identical to those in Spall (2000) with the exception of the more general weighting w_k in the second recursion ($w_k = 1/(k+1)$ in Spall, 2000, equivalent to a recursive calculation of the sample mean of the per-iteration $\mathbf{H}(\boldsymbol{\theta})$ estimates) and the inclusion of the adjustment $\hat{\Psi}_k$. Note that at $k = 0$ in (2.1b), $\bar{\mathbf{H}}_{k-1} = \bar{\mathbf{H}}_{-1}$ may be used to reflect prior information on \mathbf{H} if $0 < w_0 < 1$; alternatively, $\bar{\mathbf{H}}_{-1}$ may be unspecified—and irrelevant—when $w_0 = 1$. Because $\hat{\mathbf{H}}_k$ is defined in Spall (2000), the essential aspects of the parallel recursions in (2.1a, b) that remain to be specified are w_k and $\hat{\Psi}_k$.

Given that $\bar{\mathbf{H}}_k$ may not be invertible (especially for small k), a simple mapping \mathbf{f}_k is to add a matrix $\delta_k \mathbf{I}_p$ to $\bar{\mathbf{H}}_k$, where $\delta_k > 0$ is small for large k and \mathbf{I}_p is a $p \times p$ identity matrix. In the case of optimization, where $\mathbf{g}(\boldsymbol{\theta})$ is a gradient and $\mathbf{H}(\boldsymbol{\theta})$ is a Hessian matrix, one may also wish to impose the requirement that the Hessian estimates be symmetric. (Bhatnagar, 2005, discusses Hessian estimation without imposing symmetry at each iteration.) In this case $\mathbf{f}_k: \mathbb{R}^{p \times p} \rightarrow \{\text{symmetric positive definite } p \times p \text{ matrices}\}$. Given that $\bar{\mathbf{H}}_k$ is forced to be symmetric (as considered in Sections III and IV), one useful form for \mathbf{f}_k when p is not too large is to take \mathbf{f}_k such that $\bar{\bar{\mathbf{H}}}_k = (\bar{\mathbf{H}}_k^T \bar{\mathbf{H}}_k + \delta_k \mathbf{I}_p)^{1/2} = (\bar{\mathbf{H}}_k \bar{\mathbf{H}}_k + \delta_k \mathbf{I}_p)^{1/2}$, where the indicated square root is the (unique) positive definite square root (e.g., `sqrtn` in Matlab) and $\delta_k > 0$ is some small number as above.

Let us now present the basic per-iteration Jacobian estimate $\hat{\mathbf{H}}_k$, as given in Spall (2000). The form of this estimate will motivate the feedback-based modification $\hat{\Psi}_k$ that is one of the main purposes of this paper. This feedback modification is introduced in Section III. As with the basic first-order SPSA algorithm, let c_k be a positive scalar such that $c_k \rightarrow 0$ as $k \rightarrow \infty$ and let $\Delta_k \equiv [\Delta_{k1}, \Delta_{k2}, \dots, \Delta_{kp}]^T$ be a user-generated mean-zero random vector with finite inverse moments; further conditions on c_k , Δ_k , and other relevant quantities are given in Spall (2000). These conditions are close to those of basic SPSA in Spall (1992) (e.g., Δ_k being a vector of independent Bernoulli ± 1 random variables satisfies the conditions on the perturbations, but a vector of uniformly or normally distributed random variables does not). Examples of valid gain sequences are given in Spall (2000); see also the numerical study in Section VII below for some specific instances.

The formula for $\hat{\mathbf{H}}_k$ at each iteration is

$$\hat{\mathbf{H}}_k = \begin{cases} \frac{\delta \mathbf{G}_k}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] \text{ for Jacobian or} \\ \frac{1}{2} \left\{ \frac{\delta \mathbf{G}_k}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] \right. \\ \left. + \left(\frac{\delta \mathbf{G}_k}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] \right)^T \right\} \text{ for Hessian,} \end{cases} \quad (2.2)$$

where

$$\delta \mathbf{G}_k = \mathbf{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k + c_k \Delta_k) - \mathbf{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k - c_k \Delta_k),$$

and, depending on the setting, the function $\mathbf{G}_k^{(1)}$ may or may not be the same as the function \mathbf{G}_k introduced in (2.1a). In particular, when forming a simultaneous perturbation (or even finite difference) estimate for $\mathbf{g}(\boldsymbol{\theta})$ based on values of the loss function $L(\boldsymbol{\theta})$, there are advantages to using a *one-sided* gradient approximation in order to reduce the total number of function evaluations (vs. the standard two-sided form that would typically be used to construct \mathbf{G}_k). In other cases, one may have direct unbiased measurements of $\mathbf{g}(\boldsymbol{\theta})$ (e.g., Chap. 5 of Spall, 2003), implying that $\mathbf{G}_k^{(1)} = \mathbf{G}_k$.

Note that all elements of $\hat{\boldsymbol{\theta}}_k$ are varied simultaneously (and randomly) in forming $\hat{\mathbf{H}}_k$, as opposed to the finite-difference forms in, for example, Fabian (1971) and Ruppert (1985), where the elements of $\boldsymbol{\theta}$ are changed deterministically one at a time. The symmetrizing operation in the second line of (2.2) (the multiple $1/2$ and the indicated sum) is convenient in the optimization case in order to maintain a symmetric Hessian estimate at each k . In the general root-finding case, where $\mathbf{H}(\boldsymbol{\theta})$ represents a Jacobian matrix, the symmetrizing operation should not typically be used (i.e., the first line of (2.2) applies).

The feedback method below rests on an error analysis for the elements of the estimate $\hat{\mathbf{H}}_k$. Suppose that \mathbf{g} is three-times continuously differentiable in a neighborhood of $\hat{\boldsymbol{\theta}}_k$. Then,

$$E(\delta\mathbf{G}_k | \hat{\boldsymbol{\theta}}_k, \Delta_k) = \mathbf{g}(\hat{\boldsymbol{\theta}}_k + c_k \Delta_k) - \mathbf{g}(\hat{\boldsymbol{\theta}}_k - c_k \Delta_k) + O(c_k^3), \quad (2.3)$$

where (2.3) follows easily (as in Spall, 1992, Lemma 1) by a Taylor series argument when forming a simultaneous perturbation estimate for $\mathbf{g}(\boldsymbol{\theta})$ from measurements of the loss function $L(\boldsymbol{\theta})$ (the $O(c_k^3)$ term is the difference of the two $O(c_k^3)$ bias terms in the gradient estimate) and (2.3) is immediate (with $O(c_k^3) = \mathbf{0}$) when $\mathbf{G}_k^{(1)}$ and \mathbf{G}_k represent direct unbiased measurements of $\mathbf{g}(\boldsymbol{\theta})$. Let δG_{ki} be the i^{th} component of $\delta\mathbf{G}_k$. In the Jacobian case, (2.2) implies that the ij^{th} element of $\hat{\mathbf{H}}_k$ is $\delta G_{ki} / (2c_k \Delta_{kj})$. Then, by an expansion of each of $\mathbf{g}(\hat{\boldsymbol{\theta}}_k \pm c_k \Delta_k)$, as appears in (2.3) for any i, j ,

$$E\left(\frac{\delta G_{ki}}{2c_k \Delta_{kj}} \middle| \hat{\boldsymbol{\theta}}_k, \Delta_k\right) = H_{ij}(\hat{\boldsymbol{\theta}}_k) + \sum_{\ell \neq j} H_{i\ell}(\hat{\boldsymbol{\theta}}_k) \frac{\Delta_{k\ell}}{\Delta_{kj}} + O(c_k^2), \quad (2.4)$$

where H_{ij} denotes the ij^{th} component of \mathbf{H} . As in (2.3), the expectation in (2.4) removes the noisy error associated with the difference in the i^{th} component of the \mathbf{g} measurements $\mathbf{G}_k^{(1)}$. In the case where exact \mathbf{g} values are available (i.e., $\mathbf{G}_k^{(1)} = \mathbf{g}$, such as when $\mathbf{G}_k^{(1)}$ is an exact value of the gradient of a log-likelihood function), then $\delta G_{ki} / (2c_k \Delta_{kj})$ itself (without the conditional expectation) is equal to the right-hand side of (2.4). Because $E(\Delta_{k\ell} / \Delta_{kj}) = 0$ for all $j \neq \ell$ by

the assumptions for Δ_k , it is known that the expectation of the second (summation) term on the right-hand side of (2.4) is 0 for all i, j , and k . Hence,

$$E\left(\frac{\delta G_{ki}}{2c_k \Delta_{kj}} \middle| \hat{\boldsymbol{\theta}}_k\right) = H_{ij}(\hat{\boldsymbol{\theta}}_k) + O(c_k^2),$$

implying that the Jacobian estimate $\hat{\mathbf{H}}_k$ is nearly unbiased with the bias disappearing at rate $O(c_k^2)$. Trivial modifications to the above show the same for the Hessian estimate in the bottom line of (2.2).

Note that the Jacobian estimate $\hat{\mathbf{H}}_k$ from (2.2) can be decomposed into four parts:

$$\hat{\mathbf{H}}_k = \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\Psi}_k + \text{noise} + O(c_k^2), \quad (2.5)$$

where $\boldsymbol{\Psi}_k$ is a $p \times p$ matrix of terms dependent on $\mathbf{H}(\hat{\boldsymbol{\theta}}_k)$, Δ_k , and, when only noisy L measurements are available, an additional perturbation vector $\tilde{\Delta}_k$; the noise is based on the remaining mean-zero error from the differences $\mathbf{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k \pm c_k \Delta_k) - \mathbf{g}(\hat{\boldsymbol{\theta}}_k \pm c_k \Delta_k)$; and the $O(c_k^2)$ bias is as shown in the last term on the right-hand side of (2.4). Note that $\boldsymbol{\Psi}_k$ represents the error due to the simultaneous perturbations (Δ_k and, if relevant, $\tilde{\Delta}_k$). The noise term is zero when $\mathbf{G}_k^{(1)} = \mathbf{g}$.

Much of the convergence and efficiency analysis in Spall (2000) will hold verbatim in analyzing the enhanced form here. In particular, under conditions for Theorems 1a and 1b in Spall (2000), it is known that $\hat{\boldsymbol{\theta}}_k \rightarrow \boldsymbol{\theta}^*$ a.s. in the setting of either L measurements or \mathbf{g} measurements. On the other hand, because the recursion (2.1b) differs from Spall (2000) due to the weighting and feedback, it is necessary to make a few changes to the convergence arguments showing convergence of $\bar{\mathbf{H}}_k$. This is the subject of a separate paper available from the author upon request.

III. CHARACTERIZATION OF ERROR IN JACOBIAN ESTIMATE

This section characterizes the $\boldsymbol{\Psi}_k$ term in (2.5) as a vehicle towards creating the feedback term $\hat{\boldsymbol{\Psi}}_k$. Subsection III.A considers the case where $\mathbf{G}_k^{(1)}$ is formed from possibly noisy values of L ; Subsection III.B considers the case where $\mathbf{G}_k^{(1)}$ is formed from possibly noisy values of \mathbf{g} .

A. Error for Estimates Based on Measurements of L

This subsection considers the problem of minimizing L ; hence \mathbf{H} represents a Hessian matrix and the symmetric estimate in the second line of (2.2) applies. When using only measurements of L (no direct measurements of \mathbf{g}), the core gradient approximation $\mathbf{G}_k(\hat{\boldsymbol{\theta}}_k)$ in (2.1a) requires two measurements, $y(\hat{\boldsymbol{\theta}}_k + c_k \Delta_k)$ and $y(\hat{\boldsymbol{\theta}}_k - c_k \Delta_k)$, representing noisy measurements of L at the two design levels $\hat{\boldsymbol{\theta}}_k \pm c_k \Delta_k$, where c_k and Δ_k are as defined above for $\hat{\mathbf{H}}_k$. These two measurements will be used to generate $\mathbf{G}_k(\hat{\boldsymbol{\theta}}_k)$ in the conventional SPSA manner, in addition to being employed toward generating the one-sided gradient approximations

$\mathbf{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k)$ used in forming $\hat{\mathbf{H}}_k$. Two additional measurements $y(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k)$ are used in generating the one-sided approximations as follows:

$$\mathbf{G}_k^{(1)}(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k) = \frac{y(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k) - y(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k)}{\tilde{c}_k} \begin{bmatrix} \tilde{\Delta}_{k1}^{-1} \\ \tilde{\Delta}_{k2}^{-1} \\ \vdots \\ \tilde{\Delta}_{kp}^{-1} \end{bmatrix},$$

with $\tilde{\boldsymbol{\Delta}}_k = [\tilde{\Delta}_{k1}, \tilde{\Delta}_{k2}, \dots, \tilde{\Delta}_{kp}]^T$ generated in the same statistical manner as $\boldsymbol{\Delta}_k$, but independently of $\boldsymbol{\Delta}_k$ (in particular, choosing $\tilde{\Delta}_{ki}$ as independent Bernoulli ± 1 random variables is a valid—but not necessary—choice), and with \tilde{c}_k satisfying conditions similar to c_k (although the numerical value of \tilde{c}_k may be best chosen larger than c_k ; see Spall, 2000).

Suppose that L is four times continuously differentiable. Let $\tilde{\varepsilon}_k^{(\pm)}$ and $\tilde{\varepsilon}_k^{(\pm)}$ be the measurement noises: $\tilde{\varepsilon}_k^{(\pm)} \equiv y(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k) - L(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k)$ and $\tilde{\varepsilon}_k^{(\pm)} \equiv y(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k) - L(\hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k)$. Then, the i^{th} component of $\mathbf{G}_k^{(1)}$ at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k$ is

$$\begin{aligned} G_{ki}^{(1)}(\boldsymbol{\theta}) &= \frac{L(\boldsymbol{\theta} + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k) - L(\boldsymbol{\theta}) + \tilde{\varepsilon}_k^{(\pm)} - \varepsilon_k^{(\pm)}}{\tilde{c}_k \tilde{\Delta}_{ki}} \\ &= \frac{\tilde{c}_k \mathbf{g}(\boldsymbol{\theta})^T \tilde{\boldsymbol{\Delta}}_k + \frac{1}{2} \tilde{c}_k^2 \tilde{\boldsymbol{\Delta}}_k^T \mathbf{H}(\boldsymbol{\theta}) \tilde{\boldsymbol{\Delta}}_k}{\tilde{c}_k \tilde{\Delta}_{ki}} \\ &\quad + \frac{\frac{1}{6} \tilde{c}_k^3 L'''(\bar{\boldsymbol{\theta}}_k^{(\pm)}) [\tilde{\boldsymbol{\Delta}}_k \otimes \tilde{\boldsymbol{\Delta}}_k \otimes \tilde{\boldsymbol{\Delta}}_k] + \tilde{\varepsilon}_k^{(\pm)} - \varepsilon_k^{(\pm)}}{\tilde{c}_k \tilde{\Delta}_{ki}}, \end{aligned}$$

where $L'''(\boldsymbol{\theta}) = \partial^3 L / \partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}^T$ denotes the $1 \times p^3$ row vector of all possible third derivatives of L , $\bar{\boldsymbol{\theta}}_k^{(\pm)}$ denotes a point on the line segment between $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \tilde{c}_k \tilde{\boldsymbol{\Delta}}_k$ (the superscript in $\bar{\boldsymbol{\theta}}_k^{(\pm)}$ pertains to whether $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_k \pm c_k \boldsymbol{\Delta}_k$), and \otimes denotes the Kronecker product. Note that it is sufficient to work with the first line of (2.2) in characterizing the error for the second line (relevant for the Hessian estimation here), as the second line is trivially constructed from the first line. Substituting the expansion for $G_{ki}^{(1)}(\boldsymbol{\theta})$ above into the first line of (2.2), the ij^{th} component of $\hat{\mathbf{H}}_k$ is

$$\begin{aligned} G_{ki}^{(1)}(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k) - G_{ki}^{(1)}(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k) &= \frac{2c_k \Delta_{kj}}{\tilde{c}_k \tilde{\Delta}_{ki} \tilde{\Delta}_{kj}} \tilde{\boldsymbol{\Delta}}_k \\ &\quad + \frac{\tilde{c}_k \tilde{\boldsymbol{\Delta}}_k^T [\mathbf{H}(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k) - \mathbf{H}(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k)] \tilde{\boldsymbol{\Delta}}_k}{2c_k \tilde{\Delta}_{ki} \tilde{\Delta}_{kj}} \\ &\quad + \frac{\tilde{\varepsilon}_k^{(+)} - \varepsilon_k^{(+)} - \tilde{\varepsilon}_k^{(-)} + \varepsilon_k^{(-)}}{2\tilde{c}_k c_k \tilde{\Delta}_{ki} \tilde{\Delta}_{kj}} + \frac{O(\tilde{c}_k^3)}{c_k}, \end{aligned} \quad (3.1)$$

where the probabilistic term $O(\tilde{c}_k^3)$ reflects the difference of third-order contributions in each of the two gradient

approximations. The numerator of the first term on the right-hand side of (3.1) can be written as

$$\mathbf{g}(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k) - \mathbf{g}(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k) = 2c_k \mathbf{H}(\hat{\boldsymbol{\theta}}_k) \boldsymbol{\Delta}_k + O(c_k^3) \quad (3.2)$$

Hence, from (3.1), we have

$$\begin{aligned} &\frac{G_{ki}^{(1)}(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k) - G_{ki}^{(1)}(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \Delta_{kj}} \\ &= H_{ij}(\hat{\boldsymbol{\theta}}_k) + \frac{1}{\tilde{\Delta}_{ki} \tilde{\Delta}_{kj}} \sum_{\ell=1}^p \sum_{m=1}^p H_{\ell m}(\hat{\boldsymbol{\theta}}_k) \tilde{\Delta}_{k\ell} \Delta_{km} \\ &\quad + O(\tilde{c}_k) + \frac{\tilde{\varepsilon}_k^{(+)} - \varepsilon_k^{(+)} - \tilde{\varepsilon}_k^{(-)} + \varepsilon_k^{(-)}}{2\tilde{c}_k c_k \tilde{\Delta}_{ki} \tilde{\Delta}_{kj}} + \frac{O(\tilde{c}_k^3)}{c_k} \end{aligned} \quad (3.3)$$

Note that the four expressions to the right of the first plus sign on the right-hand side of (3.3) represent the error in the estimate of $H_{ij}(\hat{\boldsymbol{\theta}}_k)$. The last three of these expressions either go to zero (almost surely, a.s.) with k (the two big- O expressions) or are based on the noise terms, $\tilde{\varepsilon}_k^{(\pm)}$ and $\varepsilon_k^{(\pm)}$, which we control through the choice of the w_k (Sect. 5). Hence, the focus in using feedback to improve the estimate for \mathbf{H} will be on the first of the four error expressions (the double-sum-based expression).

Let us define

$$\mathbf{D}_k = \boldsymbol{\Delta}_k [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] - \mathbf{I}_p,$$

together with a corresponding matrix $\tilde{\mathbf{D}}_k$ based on replacing all Δ_{ki} in \mathbf{D}_k with the corresponding $\tilde{\Delta}_{ki}$ (\mathbf{I}_p is the $p \times p$ identity matrix). Then, the matrix representation of (3.3) is

$$\begin{aligned} &\frac{\delta \mathbf{G}_k}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] \\ &= \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \tilde{\mathbf{D}}_k \mathbf{H}(\hat{\boldsymbol{\theta}}_k) \mathbf{D}_k + \tilde{\mathbf{D}}_k \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \mathbf{H}(\hat{\boldsymbol{\theta}}_k) \mathbf{D}_k + O(\tilde{c}_k) \\ &\quad + \frac{\tilde{\varepsilon}_k^{(+)} - \varepsilon_k^{(+)} - \tilde{\varepsilon}_k^{(-)} + \varepsilon_k^{(-)}}{2\tilde{c}_k c_k} \begin{bmatrix} \tilde{\Delta}_{k1}^{-1} \\ \tilde{\Delta}_{k2}^{-1} \\ \vdots \\ \tilde{\Delta}_{kp}^{-1} \end{bmatrix} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] + \frac{O(\tilde{c}_k^3)}{c_k}. \end{aligned} \quad (3.4)$$

Given that the term dependent on the noises is $O(\tilde{c}_k^{-1} c_k^{-1})$, we have

$$\hat{\mathbf{H}}_k = \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\Psi}_k^{(L)}(\mathbf{H}(\hat{\boldsymbol{\theta}}_k)) + O(\tilde{c}_k) + O(\tilde{c}_k^{-1} c_k^{-1}) + \frac{O(\tilde{c}_k^3)}{c_k}, \quad (3.5)$$

where from (2.2) (Hessian estimate in second line) and (3.4)

$$\begin{aligned} \boldsymbol{\Psi}_k^{(L)}(\mathbf{H}) &= \frac{1}{2} [\tilde{\mathbf{D}}_k \mathbf{H} \mathbf{D}_k + \tilde{\mathbf{D}}_k \mathbf{H} + \mathbf{H} \mathbf{D}_k] \\ &\quad + \frac{1}{2} [\tilde{\mathbf{D}}_k \mathbf{H} \mathbf{D}_k + \tilde{\mathbf{D}}_k \mathbf{H} + \mathbf{H} \mathbf{D}_k]^T \end{aligned} \quad (3.6)$$

(The superscript L in $\boldsymbol{\Psi}_k^{(L)}$ represents the dependence of this form on L measurements for creating the \mathbf{H} estimate, to be contrasted with $\boldsymbol{\Psi}_k^{(g)}$ in the next subsection, which is dependent on \mathbf{g} measurements.)

B. Error for Estimates Based on Values of \mathbf{g}

We now consider the case where direct (but possibly

noisy) values of \mathbf{g} are available. Hence, direct measurements $\mathbf{Y}_k(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) + \mathbf{e}_k(\boldsymbol{\theta})$ are used for \mathbf{G}_k in (2.1a) and for $\mathbf{G}_k^{(l)}$ in $\delta\mathbf{G}_k$ appearing in (2.2), where \mathbf{e}_k is a mean-zero noise term (not necessarily independent or identically distributed across k). The analysis in this case is easier than that in Subsection III.A as a consequence of having the direct measurements of \mathbf{g} . As in Subsection III.A, it is sufficient to work with the first line of (2.2) in characterizing the error for the second line (relevant for the Hessian estimation here). Using the expansion in (3.2), the i^{th} component of the first line of (2.2) is

$$\begin{aligned} & \frac{G_{ki}^{(l)}(\hat{\boldsymbol{\theta}}_k + c_k \Delta_k) - G_{ki}^{(l)}(\hat{\boldsymbol{\theta}}_k - c_k \Delta_k)}{2c_k \Delta_{kj}} \\ &= \frac{g_i(\hat{\boldsymbol{\theta}}_k + c_k \Delta_k) - g_i(\hat{\boldsymbol{\theta}}_k - c_k \Delta_k)}{2c_k \Delta_{kj}} + \frac{e_{ki}^{(+)} - e_{ki}^{(-)}}{2c_k \Delta_{kj}} \\ &= H_{ij}(\hat{\boldsymbol{\theta}}_k) + \sum_{\ell \neq j} H_{i\ell}(\hat{\boldsymbol{\theta}}_k) \frac{\Delta_{k\ell}}{\Delta_{kj}} + O(c_k^2) + \frac{e_{ki}^{(+)} - e_{ki}^{(-)}}{2c_k \Delta_{kj}}, \quad (3.7) \end{aligned}$$

where g_i is the i^{th} term of \mathbf{g} and the $e_{ki}^{(\pm)}$ represent the i^{th} components of the noise vectors $\mathbf{e}_k(\hat{\boldsymbol{\theta}}_k \pm c_k \Delta_k)$.

Note that the three expressions to the right of the first plus sign in the last line of (3.7) represent the error in the estimate of $H_{ij}(\hat{\boldsymbol{\theta}}_k)$. The last two of these expressions either go to zero with k (the big- O expression) or are based on the noise terms, $e_{ki}^{(\pm)}$, which we control through the choice of the w_k (Sect. V). Hence, the focus in using feedback to improve the estimate for \mathbf{H} will be on the first of the three error expressions (the summation-based expression).

The matrix representation of (3.7) is

$$\begin{aligned} \frac{\delta\mathbf{G}_k}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] &= \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \mathbf{H}(\hat{\boldsymbol{\theta}}_k) \mathbf{D}_k + O(c_k^2) \\ &+ \frac{e_k^{(+)} - e_k^{(-)}}{2c_k} [\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \dots, \Delta_{kp}^{-1}] \quad (3.8) \end{aligned}$$

Analogous to (3.5), we have

$$\hat{\mathbf{H}}_k = \mathbf{H}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\Psi}_k^{(\mathbf{g})}(\mathbf{H}(\hat{\boldsymbol{\theta}}_k)) + O(c_k^2) + O(c_k^{-1}), \quad (3.9)$$

where from (2.2) and (3.8)

$$\boldsymbol{\Psi}_k^{(\mathbf{g})}(\mathbf{H}) \equiv \begin{cases} \mathbf{H}\mathbf{D}_k & \text{for Jacobian or} \\ \frac{1}{2}\mathbf{H}\mathbf{D}_k + \frac{1}{2}\mathbf{D}_k^T \mathbf{H} & \text{for Hessian.} \end{cases} \quad (3.10)$$

IV. FEEDBACK-BASED ESTIMATE OF \mathbf{H} MATRIX

From the analysis in Section III, there are two key ways in which the quality of the estimate $\bar{\mathbf{H}}_k$ can be improved relative to the simple averaging of Spall (2000), where $w_k = 1/(k+1)$ and $\hat{\boldsymbol{\Psi}}_k = \mathbf{0}$ for all k in (2.1b). The first will be in setting $\hat{\boldsymbol{\Psi}}_k \neq \mathbf{0}$ through the use of feedback, as discussed in this section. The second will be in choosing the weights w_k in an asymptotically optimal manner, as discussed in Section V.

If \mathbf{H} were known, setting $\hat{\boldsymbol{\Psi}}_k$ equal to $\boldsymbol{\Psi}_k^{(\cdot)}(\mathbf{H}(\hat{\boldsymbol{\theta}}_k))$ would leave only the unavoidable errors due to the noise and the bias at each iteration, where $\boldsymbol{\Psi}_k^{(\cdot)}$ represents either $\boldsymbol{\Psi}_k^{(L)}$

or $\boldsymbol{\Psi}_k^{(\mathbf{g})}$, as appropriate (expressions (3.6) and (3.10), respectively). Unfortunately, of course, this relatively simple modification cannot be implemented because we do not know \mathbf{H} !

A variation on the idealized \mathbf{H} estimate of the previous paragraph is to use *estimates* of \mathbf{H} in place of the true \mathbf{H} . That is, the most recent *estimate* of $\mathbf{H}(\hat{\boldsymbol{\theta}}_k)$, as given by $\bar{\bar{\mathbf{H}}}_{k-1}$, replaces $\mathbf{H}(\hat{\boldsymbol{\theta}}_k)$ in forming $\hat{\boldsymbol{\Psi}}_k$. Therefore, the quantity $\hat{\boldsymbol{\Psi}}_k$ appearing in (2.1b) is given by

$$\hat{\boldsymbol{\Psi}}_k \equiv \begin{cases} \boldsymbol{\Psi}_k^{(L)}(\bar{\bar{\mathbf{H}}}_{k-1}) & \text{when } L \text{ measurements used,} \\ \boldsymbol{\Psi}_k^{(\mathbf{g})}(\bar{\bar{\mathbf{H}}}_{k-1}) & \text{when } \mathbf{g} \text{ measurements used.} \end{cases}$$

V. OPTIMAL WEIGHTING WITH NOISY INPUTS

A. General Form

As discussed above, the second way in which the accuracy of the \mathbf{H} estimate may be improved is through the optimal selection of weights w_k in (2.1b). We consider separately below the cases where $\mathbf{G}_k^{(l)}$ is formed from noisy values of L and noisy values of \mathbf{g} . We restrict ourselves to linear unbiased estimators for \mathbf{H} as represented in recursive form in (2.1b). Hence, the estimator in equivalent batch form for n total iterations is

$$\bar{\mathbf{H}}_n = \sum_{k=0}^n \omega_k (\hat{\mathbf{H}}_k - \hat{\boldsymbol{\Psi}}_k), \quad (5.1)$$

subject to $\omega_k \geq 0$ for all k and $\sum_{k=0}^n \omega_k = 1$. As shown in Spall (2000), $E(\hat{\mathbf{H}}_k) = \mathbf{H}(\boldsymbol{\theta}^*) + o(1)$ under the given conditions for convergence of $\hat{\boldsymbol{\theta}}_k$ to $\boldsymbol{\theta}^*$. Then, because $E(\hat{\boldsymbol{\Psi}}_k) = \mathbf{0}$ for all k , the form in (5.1) guarantees asymptotic unbiasedness for $\bar{\mathbf{H}}_n$ (i.e., $E(\bar{\mathbf{H}}_n) = \mathbf{H}(\boldsymbol{\theta}^*) + o(1)$) provided that the ω_k do not decay too quickly; see Section 6. Once the ω_k are determined, it is straightforward to determine the weights w_k appearing in the recursion (2.1b).

B. Weights Using Measurements of L

The asymptotically optimal weighting is driven by the asymptotic variances of the elements in $\hat{\mathbf{H}}_k$. The dominant contributor to each asymptotic variance is the $O(\tilde{c}_k^{-1} c_k^{-1})$ term on the right-hand side of (3.5), leading to a variance that is $O(\tilde{c}_k^{-2} c_k^{-2})$. The variances of the elements in the matrix in (3.5) are known to exist by Hölder's inequality when $\varepsilon_k^{(\pm)}$, $\tilde{\varepsilon}_k^{(\pm)}$, Δ_{ki}^{-1} , and $\tilde{\Delta}_{ki}^{-1}$ all have finite moments of order greater than 2 for all i and k . Hence, given that the noise terms $\varepsilon_k^{(\pm)}$ and $\tilde{\varepsilon}_k^{(\pm)}$ are uncorrelated across iterations (e.g., $\text{cov}(\varepsilon_k^{(+)}, \tilde{\varepsilon}_k^{(+)}) = 0$ and $\text{cov}(\varepsilon_j^{(+)}, \varepsilon_k^{(+)}) = 0$ for $j \neq k$), we are faced with finding the weights ω_k that minimize $\sum_{k=0}^n \omega_k^2 \tilde{c}_k^{-2} c_k^{-2}$ subject to the constraints on ω_k above. It is fairly straightforward to find the solution to this minimization problem (e.g., via the method of Lagrange multipliers), leading to optimal values $\omega_k =$

$\tilde{c}_k^2 c_k^2 / \sum_{i=0}^n \tilde{c}_i^2 c_i^2$ for all $0 \leq k \leq n$. The above weighting can be implemented recursively by letting

$$w_k = \frac{\tilde{c}_k^2 c_k^2}{\sum_{i=0}^k \tilde{c}_i^2 c_i^2}$$

in (2.1b).

C. Weights Using Measurements of \mathbf{g}

As in Subsection V.B, the asymptotically optimal weighting is driven by the asymptotic variances of the elements in $\hat{\mathbf{H}}_k$. From (3.9), the asymptotic variance of each element is $O(c_k^{-2})$ (i.e., the fourth term on the right-hand side of (3.9) is the dominant term). Hence, given that the noise terms $\varepsilon_k^{(\pm)}$ are uncorrelated across iterations, we are faced with finding the weights ω_k that minimize $\sum_{k=0}^n \omega_k^2 c_k^{-2}$ subject to the constraints on ω_k above. The solution to this minimization problem is $\omega_k = c_k^2 / \sum_{i=0}^n c_i^2$ for all $0 \leq k \leq n$. The above weighting can be implemented recursively by letting

$$w_k = \frac{c_k^2}{\sum_{i=0}^k c_i^2}$$

in (2.1b).

VI. NUMERICAL STUDY

Consider the fourth-order loss function used in numerical demonstrations of Spall (2000):

$$L(\boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{B}^T \mathbf{B} \boldsymbol{\theta} + 0.1 \sum_{i=1}^p (\mathbf{B}\boldsymbol{\theta})_i^3 + 0.01 \sum_{i=1}^p (\mathbf{B}\boldsymbol{\theta})_i^4, \quad (7.1)$$

where $(\cdot)_i$ represents the i^{th} component of the argument vector $\mathbf{B}\boldsymbol{\theta}$, and \mathbf{B} is such that $p\mathbf{B}$ is an upper triangular matrix of 1's (so elements below the diagonal are zero). Let $p = 10$. The minimum occurs at $\boldsymbol{\theta}^* = \mathbf{0}$ with $L(\boldsymbol{\theta}^*) = 0$; all runs are initialized at $\hat{\boldsymbol{\theta}}_0 = [0.2, 0.2, \dots, 0.2]^T$ (so $L(\hat{\boldsymbol{\theta}}_0) = 0.1565$). The measurement noises for the L measurements are dependent on $\boldsymbol{\theta}$ in the sense that $\varepsilon = \varepsilon(\boldsymbol{\theta}) = [\boldsymbol{\theta}^T, 1]V$, where V is an independent and identically distributed (i.i.d.) vector (across L or \mathbf{g} measurements) with distribution $N(\mathbf{0}, \sigma^2 \mathbf{I}_{11})$; hence, the distribution of the noise e in the gradient measurements is i.i.d. $N(\mathbf{0}, \sigma^2 \mathbf{I}_{10})$; we choose $\sigma = 0.01$ for the study below. All iterates are constrained to be in $\Theta = [-10, 10]^{10}$.

This study compares the standard adaptive SPSA method with the enhanced method here when direct (noisy) measurements of \mathbf{g} are available. This corresponds to the ‘‘2SG’’ (second-order stochastic gradient) setting of Spall (2000). Following practical guidelines in Spall (2000), an iteration is blocked if $\boldsymbol{\theta}$ moves too far (an indication of algorithm instability); in particular if $\|\hat{\boldsymbol{\theta}}_{k+1} - \hat{\boldsymbol{\theta}}_k\| \geq 1.0$, then the step is blocked and $\hat{\boldsymbol{\theta}}_{k+1}$ is reset to $\hat{\boldsymbol{\theta}}_k$. We used the standard forms for the gain sequences a_k and c_k : $a_k = a/(k+1+A)^\alpha$ and $c_k = c/(k+1)^\gamma$, where a , c , α , and γ are strictly positive and the stability constant $A \geq 0$ (see Spall, 2003, Sects. 4.4, 6.6, or 7.5 for further discussion of these gain forms).

The table below presents the results of this small-scale

study. In each row to the table, the standard and enhanced algorithms are run with the same gain sequences as indicated in the table. The gains are chosen to satisfy convergence conditions and the critical step-size coefficient a is approximately ‘‘tuned’’ to optimize the performance of the *standard* 2SG method for a given number of iterations and choice of gain coefficients α and γ (governing the decay rates of the two sequences). Each sample mean represents 40 independent runs. The indicated P -values are based on the standard two-sample t -test and represent the probability in a future experiment of the two sample means being at least as far apart as the observed sample means under the null hypothesis that the true means are identical. All indicated P -values are relatively small, consistent with the enhanced 2SG algorithm being statistically significantly better than the standard 2SG algorithm (i.e., rejecting the null hypothesis of equality of means).

Sample means for terminal values of normalized loss functions $L(\hat{\boldsymbol{\theta}}_k)/L(\hat{\boldsymbol{\theta}}_0)$; indicated P -values are for differences between sample means.

Number of iterations, n	a, A, α, c, γ	Standard 2SG	Enhanced 2SG	P -value
2000	100, 50, 1, 0.01, 0.49	0.013	0.00088	0.058
10,000	100, 50, 1, 0.01, 0.49	0.0024	0.00021	0.0016
10,000	20, 50, 1, 0.01, 1/6	0.0032	0.00037	0.039

VII. REFERENCES

- [1] Bhatnagar, S. (2005), ‘‘Adaptive Multivariate Three-Timescale Stochastic Approximation Algorithms for Simulation Based Optimization,’’ *ACM Transactions on Modeling and Computer Simulation*, vol. 15, pp. 74–107.
- [2] Chow, Y. S. and Teicher, H. (1988), *Probability Theory: Independence, Interchangeability, and Martingales* (2nd ed.), Springer-Verlag, New York.
- [3] Fabian, V. (1971), ‘‘Stochastic Approximation,’’ in *Optimizing Methods in Statistics* (J. S. Rustagi, ed.), Academic Press, New York, pp. 439–470.
- [4] Ruppert, D. (1985), ‘‘A Newton–Raphson Version of the Multivariate Robbins–Monro Procedure,’’ *Annals of Statistics*, vol. 13, pp. 236–245.
- [5] Kushner, H. J. and Yang, J. (1995), ‘‘Stochastic Approximation with Averaging and Feedback: Rapidly Convergent On-Line Algorithms,’’ *IEEE Transactions on Automatic Control*, vol. 40, pp. 24–34.
- [6] Kushner, H. J. and Yin, G. G. (2003), *Stochastic Approximation and Recursive Algorithms and Applications* (2nd ed.), Springer-Verlag, New York.
- [7] Macchi, O. and Eweda, E. (1983), ‘‘Second-Order Convergence Analysis of Stochastic Adaptive Linear Filtering,’’ *IEEE Transactions on Automatic Control*, vol. AC-28, pp. 76–85.
- [8] Polyak, B. T. and Juditsky, A. B. (1992), ‘‘Acceleration of Stochastic Approximation by Averaging,’’ *SIAM Journal on Control and Optimization*, vol. 30, pp. 838–855.
- [9] Ruppert, D. (1985), ‘‘A Newton–Raphson Version of the Multivariate Robbins–Monro Procedure,’’ *Annals of Statistics*, vol. 13, pp. 236–245.
- [10] Spall, J. C. (1992), ‘‘Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,’’ *IEEE Transactions on Automatic Control*, vol. 37, pp. 332–341.
- [11] Spall, J. C. (2000), ‘‘Adaptive Stochastic Approximation by the Simultaneous Perturbation Method,’’ *IEEE Transactions on Automatic Control*, vol. 45, pp. 1839–1853.
- [12] Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ.
- [13] Yin, G. and Zhu, Y. (1992), ‘‘Averaging Procedures in Adaptive Filtering: An Efficient Approach,’’ *IEEE Transactions on Automatic Control*, vol. 37, pp. 466–475.
- [14] Wei, C. Z. (1987), ‘‘Multivariate Adaptive Stochastic Approximation,’’ *Annals of Statistics*, vol. 15, pp. 1115–1130.

Overview of the First Advanced Technology Evaluations for ASSIST

Craig Schlenoff, Brian Weiss, Micky Steves, Ann Virts, Michael Shneier
National Institute of Standards and Technology
(NIST)
100 Bureau Drive, Stop 8230
Gaithersburg, MD, USA
{craig.schlenoff}, {brian.weiss},
{michelle.steves}, {ann.virts}, {michael.shneier}
@nist.gov

Michael Linegang
Aptima, Incorporated
1726 M Street, NW
Washington, D.C. 20036
{linegang@aptima.com}

Abstract—ASSIST (Advanced Soldier Sensor Information Systems Technology) is a DARPA-funded effort whose goal is to exploit soldier-worn sensors to augment the soldier's recall and reporting capability to enhance situation understanding. ASSIST is separated into two tasks; Task 1 focuses on the hardware and Task 2 focuses on the software. NIST's role in this program is to develop and implement evaluation procedures to characterize the performance of the software components developed under Task 2. This paper provides an overview of the ASSIST program, the evaluation procedures, the metrics that the evaluation procedures were addressing, and the technology being evaluated.

Keywords: *DARPA, ASSIST, soldier-worn sensors, evaluation methodology, elemental tests, vignette tests*

I. INTRODUCTION

The Advanced Soldier Sensor Information Systems and Technology (ASSIST) program is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program. The objective of the ASSIST program is to exploit soldier-worn sensors to augment a soldier's recall and reporting capability to enhance situational understanding in military operations in urban terrain (MOUT) environments. The program is split into two tasks:

- Task 1, named Baseline System Development, stresses active information capture and voice annotations exploitation. The resulting products from Task 1 will be prototype wearable capture units and the supporting operational software for processing, logging and retrieval.
- Task 2, named Advanced Technology Research, stresses passive collection and automated activity/object recognition. The results from this task will be the algorithms, software, and tools that will undergo system integration in later phases of the program.

The National Institute of Standards and Technology (NIST)

Intelligent Systems Division (ISD), along with NIST's subcontractors (Aptima and DCS Corporation), are funded to serve as the Independent Evaluation Team (IET) for Task 2. As the IET for Task 2, NIST is responsible for:

- Understanding the Task 2 contractor technologies
- Determining an approach for testing their technologies
- Identifying a Military Operations in Urban Terrains (MOUT) site to evaluate the technologies
- Devising and executing the tests
- Analyzing the data and documenting the outcome

Section II gives background on how the ASSIST system is expected to be used. Section III provides an overview of the technology that was tested. Section IV described the metrics and the testing methodology. Section V concludes the paper.

II. EXPECTED USE OF THE ASSIST SYSTEMS

Soldiers are often asked to perform missions that can take many hours. Examples of missions include presence patrols (where soldiers are tasked to make their presence known in an environment), search and reconnaissance missions, apprehending suspected insurgents, etc. After a mission is complete, the soldiers are typically asked to provide a report to their immediate supervisor describing the most important things that happened during the mission. This report is used to gather intelligence about the environment to allow for more informed planning for future missions. Soldiers usually provide this report based solely on their memory and still pictures that were taken during the mission, if a camera is available to and used by the soldier. These missions are often very stressful for the soldier and thus there are undoubtedly many instances in which important information is not made available in the report and thus not available for the planning of future missions.

The ASSIST program is addressing this challenge by instrumenting soldiers with sensors that they can wear directly

on their uniform. These sensors include still cameras, video cameras, Global Positioning Systems (GPS), Inertial Navigation Systems (INS), microphones, and accelerometers. These sensors continuously record what is going on around the soldier while on a mission. When soldiers return from their mission, the sensor data is run through a series of software systems which index the data and create an electronic chronicle of the events that happen throughout the time that the ASSIST system was recording. The electronic chronicle includes times that certain sounds or keywords were heard, the times when certain types of objects were seen, and times that the soldiers were in a specific location or performing certain actions.

With this information, soldiers can give reports without relying solely on their memory. The electronic chronicle will help jog the soldier's memory on things that happened that s/he did not recall during the reporting period, or possibly even make him/her aware of an important activity that s/he did not notice when out on the mission. On top of this, the multimedia information that is available in the electronic chronicle is available to the soldier to include in the report, which will provide substantially more information to the recipient of the report than the text alone.

III. TECHNOLOGIES UNDER TEST

Task 2 of the ASSIST program is developing a variety of soldier-worn sensors, data capture, data analysis, and information presentation technologies. Below is a listing of three of the general data types being captured and analyzed by ASSIST technologies. Within each data type, numerous "technology elements" are being applied to organize, process, and present that data. Some of the key technology elements being applied in the ASSIST program are listed below.

"Image/Video Data Analysis Capabilities"

- Object Detection / Image Classification – the ability to recognize and identify objects (e.g. identify vehicles, people, license plates, etc.) through analysis of video, imagery, and/or related data sources.
- Arabic Text Translation – the ability to detect, recognize and translate written Arabic text (e.g. in imagery data).
- Change Detection – the ability to identify changes over time in related data sources (e.g. identify differences in imagery of the same location at different times)

"Audio Data Analysis Capabilities"

- Sound Recognition / Speech Recognition – the ability to identify sound events (e.g. explosions, gunshots, vehicles, etc.) and recognize speech (e.g. keyword spotting, foreign language identification, etc.) in audio data.
- Shooter Localization / Shooter Classification – the ability to identify gunshots in the environment (e.g.

through analysis of audio data), including the type of weapon producing those shots, and the location of the shooter for those gunshots.

"Soldier Activity Data Analysis Capabilities"

- Soldier State Identification / Soldier Localization – the ability to identify a soldier's path of movement around an environment and characterize the actions taken by the soldier (e.g. running, walking, climbing stairs, etc.)

There is no single integrated ASSIST system at this point in the program's life-cycle. Instead, several university and corporate research and development organizations have formed into "research teams." Each organization is developing specific technology components, and these components are gradually being integrated as a "research team" system. The following sub-sections provide a brief overview of the specific technologies being developed by each research team.

A. IBM / Georgia Tech / MIT Team System

The IBM Team ("IBM") brings together three research and development organizations: IBM, Georgia Tech, and Massachusetts Institute of Technology (MIT). AWARE Technologies is also involved in a portion of Georgia Tech's research and development. IBM has an ASSIST suite that includes hardware and software in more than 10 technological areas. The long-term vision for IBM's ASSIST suite is a complete system that captures, analyzes, organizes, and archives data for users (soldier and intelligence operators) to review and search to enhance after-action reporting and intelligence exploitation capabilities.

The IBM team's technology includes:

- Soldier state identification (e.g., driving, walking, running, standing, sitting, situation assessment from cover, going upstairs, going downstairs, lying down, crawling, taking a knee, shaking hands, opening door, raising a weapon, dragging)
- Image Classification -Images captured by the soldier are labeled with one or more classes and subclasses (outdoors, indoors, sky, building, vegetation, people, soldier, commotion, weapon, car, civilian vehicle, military vehicle, face, license plate)
- Object Detection --The presence of an object (faces, clothing color (based on face detection) and license plates) is detected based on data from one or more sensors
- Speech Recognition and keyword extraction is performed on the soldier's speech (keywords include assault, contact, dead, fire, flash bang, go, grenades, incoming, insurgent, intel, intelligence, kill, move, report, shots, spot suspicious, target, update, weapons,

A4 mm round, AK47, Alpha, Bravo, C4, frag out, halt, IED, m16, RPG, SITREP, and tango)

- Identification of languages spoken in the environment (Arabic, English, French, German, Hindi, Japanese, Mandarin, and Spanish)
- Identification of “impulse audio” (single gunshot, machine gun, and explosions) and vehicles (light truck, transport sedan, transport van)
- Automatic Timeline Segmentation-- For the period of capture, the system automatically tags the timeline with appropriate labels (e.g. soldier was running from time x to time y, explosion detected at time z, etc.).

IBM’s ASSIST suite hardware includes cameras, microphones, GPS, accelerometers, compass and physiological sensors. The IBM ASSIST hardware suite can be seen in the Figure 1. A screenshot of their user interface is shown in Figure 2.



Figure 1: IBM’s Hardware



Figure 2: IBM’s User Interface

B. Sarnoff Team’s System

The Sarnoff ASSIST team also consists of three research and development organizations: Sarnoff Corporation, Carnegie

Mellon University, and Vanderbilt University. However, each of these three groups is focusing on unique technologies that will not be integrated with one another during this initial phase of the ASSIST project. As a result, each organization was treated as a separate team. The following sections discuss the systems from each team.

1) Sarnoff’s System

Sarnoff is developing a prototype system that captures data from stereo-vision cameras, GPS, and an inertial navigation system (INS). These data capture devices are carried on a backpack framework along with a laptop computer. The Sarnoff system applies software algorithms (e.g. landmark matching) to support Soldier State Identification, Soldier Localization, and Object Detection. The team has also developed mission-map viewing software to allow the soldier to visually relive their mission.

Sarnoff’s technology includes:

- Soldier localization – The ability to locate a person outdoors and indoors in GPS coordinates using Video INS, INS, landmark matching and GPS (where available)
- Object detection – The ability to identify people, vehicles, and weapons (no sub-classification)
- Mission map viewer –The ability to overlay wearer’s path on overhead map. Click on different points on path to retrieve visuals of what the wearer sees at that location. Move along path and dynamically view the world. Detected objects will be highlighted.

Sarnoff’s system can be seen in Figure 3 and their user interface can be seen in Figure 4.



Figure 3: Sarnoff’s Hardware

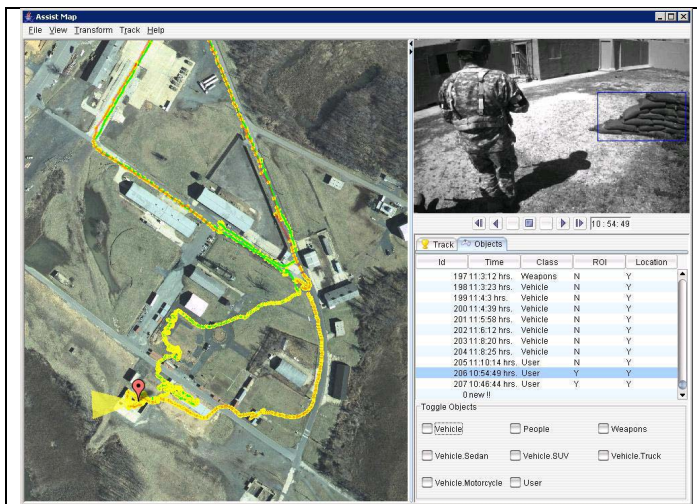


Figure 4: Sarnoff's User Interface

2) Carnegie Mellon University's System

Carnegie Mellon University (CMU) is developing Arabic text recognition and translation technologies. CMU's goal is to extract and translate Arabic text in pictures of the environment taken with a consumer-grade digital camera.

CMU's technology includes:

- Edge detection, layout analysis, and search algorithms to identify Arabic text in an image
- Optical character recognition software to extract the text from the image
- Statistical machine translation technology to translate Arabic to English.

CMU's user interface can be seen in Figure 5.



Figure 5: CMU's User Interface

3) Vanderbilt University's System

Vanderbilt University is developing shooter localization technology. Their technology seeks to locate a shooter,

determine bullet trajectory, and classify the type of weapon being fired. The current hardware suite consists of 10 acoustic localization sensors and 2 acoustic weapon classification sensors (currently, mounted on tripods, but will ultimately be worn by the warfighters).

Vanderbilt's technology includes:

- Shot localization - Determine the trajectory of shots from 50 m -300 m. Determine the shooter origin at short range of a shooter firing automatic rounds.
- Shot classification - Classify shots from an M16, AK-47, 50 caliber sniper rifle, M4, M240, and M249.
- Data display - Localization and classification data displayed on a single laptop.

Vanderbilt's user interface can be seen in Figure 6.

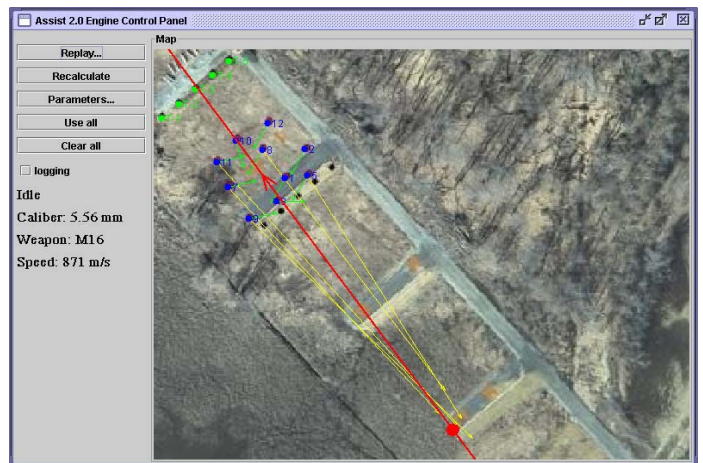


Figure 6: Vanderbilt's User Interface

C. University of Washington's System

The University of Washington ("UWash") team consists of the University of Washington, Intel Research Seattle, and Lupine Logic. This team is developing an integrated system that provides graphical and textual summaries of soldier activity over long periods of time. The system features a small, lightweight sensor pack that can be used up to eight hours for data collection in the current configuration. The system uses relational, hierarchical models of temporal data, and can be "trained" to recognize and distinguish different soldier activities.

Washington's technology includes:

- Soldier localization - GPS trace overlaid on overhead area image
- Soldier state identification - Identify activities of individual soldiers (indoor, outdoor, riding in vehicle, walking, running, standing, performing situation assessment from cover, going upstairs, going downstairs)
- Sound recognition - Manual review of audio data

during GPS trace

- Map/Mission viewer - Displays the wearer's path on an overhead map. Identifies soldier activities and audio events on a synchronized mission timeline.

Washington's system can be seen in Figure 7 and their user interface can be seen in Figure 8.



Figure 7: Washington's Hardware

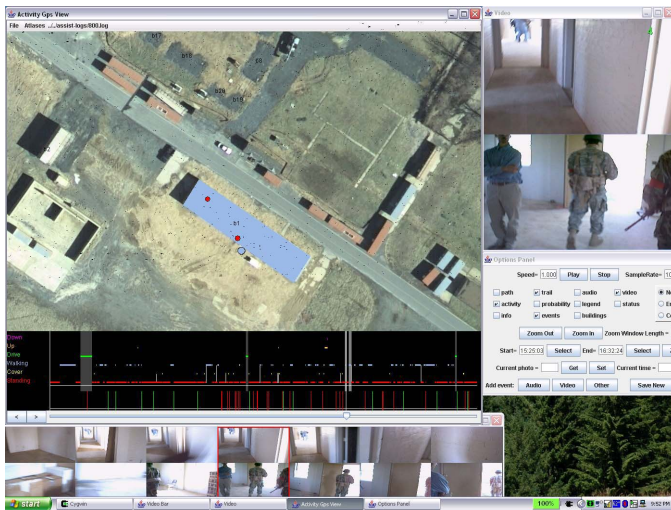


Figure 8: Washington's User Interface

IV. EXPERIMENTAL DESIGN

An experimental method was designed to evaluate the ASSIST technologies given their expected state of maturity at both 6 months and 12 months into the program. The IET attempted to design an evaluation approach that would scale well with the developing technologies, thus allowing valid assessments of technology performance improvements over time.

The ASSIST evaluations were intended as the first in a series of independent evaluations. As Per the ASSIST Broad

Agency Announcement (BAA) [1], the following three metrics were the focus for the Task 2 evaluation:

- 1) The accuracy of object/event/activity identification and labeling
- 2) The system's ability to improve its classification performance through learning
- 3) The utility of the system in enhancing operational effectiveness

The IET developed a two-part test methodology to produce these metrics. Metrics 1 and 2 were evaluated through "elemental tests," and metric 3 was evaluated through "vignette tests." In short, elemental tests were designed to measure the progressive development of ASSIST system technical capabilities; and vignette tests were designed to predict the impact these technologies will have on warfighter's performance in a variety of missions and job functions. In specifying the detailed procedures for each elemental and vignette test, the IET attempted to define evaluation strategies that would provide a reasonable level of difficulty for system and soldier performance at both the 6-month and 12-month evaluations.

A. Elemental Tests

Elemental tests were developed to test ASSIST technologies in an "ideal" environment, and allowed focused examination of specific system components. While these tests did not immerse the technologies in realistic military scenarios, they afforded the ability to modify certain variables in a controlled fashion to assess the impact of those variables on technology performance in a MOUT site environment. For example, to test CMU's Arabic text translation technologies, the IET established a method that varied the system's distance from Arabic signs, the angle at which the sign was viewed, and the amount of light in the environment. Similar variables were identified and manipulated in the other elemental tests. The five elemental tests are described below.

1) Shooter Localization

This test evaluated Vanderbilt's technology's ability to identify gunshots, the type of weapon producing those shots, and the source of those gunshots in an environment with some obstructions and minimal background noise. A "zero line" and four firing lines (≈ 50 m, ≈ 100 m, ≈ 200 m, ≈ 300 m) were marked on the firing range. The ASSIST system's acoustic sensors were placed around and behind the zero line, and randomly covered an area that was ~ 30 m x ~ 30 m. Five targets were set up behind the sensor region. Simple wooden-walled structures (single story and two story) with windows were constructed at the firing lines and in the sensor region to simulate the buildings and obstructions that would be found in a MOUT environment, and to provide unique shooter positions through windows, next to walls, and on

upper levels. Three to six shooter positions were specified at each firing line. The following variables were considered in the placement of shooter positions:

- Shooter positioning relative to walls at the firing line (within a window, next to a wall, from a clearing)
- Obstructions between the firing line and sensor field (Positions obstructed by walls that could occlude a bullet's muzzle blast and/or shockwave from a subset of the sensors)

2) Soldier State/Localization

The goal of the soldier state / localization elemental test was to determine the ASSIST systems' ability to localize a soldier in indoor and outdoor environments, and to characterize the motion of the soldier (e.g., running, walking, going inside a building, going up stairs, lying down, etc.). In the six-month evaluation, there were 6 test runs. Each test run exposed the system to a different level of difficulty for soldier state / localization identification. Run 1 was only outside in open areas. Run 2 was also outside but included some tight, GPS-hampered locations. Run 3 was both outside and inside, but did not force a change in elevation. Run 4 was predominantly inside and traversed two floors of a building. Run 5 involved a loop around a large portion of the MOUT complex, in which each action occurred for a longer period of time. Run 6 introduced a new part of the MOUT complex, and included much more driving and going up and down stairs. Each run required a soldier, shadowed by a researcher wearing the ASSIST system, to traverse a predefined path of waypoints in a scripted fashion.

101 waypoints were marked with two centimeters accuracy using differential GPS and surveying equipment. There were 42 indoor points across two different levels of buildings. There were 59 outdoor points, about 20 of which were placed next to walls and buildings, thus making it difficult to pick up a GPS signal. Poles were placed in orange cones at each waypoint. Colored signs attached to the poles indicated a letter for each waypoint in a run (e.g. A, then B, then C, etc.), gave a brief description of the action to be performed at the waypoint and on the way to the next waypoint (e.g. "lie down for 10 seconds then run," "drive," "go up stairs," "stand for 10 seconds then walk," etc.), and provided an arrow pointing to the next waypoint.

3) Object Image Classification

The goal of the object detection / image classification test was to evaluate the capabilities of the ASSIST systems to classify imagery based on the presence of various objects (e.g., people, vehicles, weapons, etc.) and states (outdoors and indoors).

The elemental test was designed to provide ample opportunities for the ASSIST systems to view the above list of objects and states. Prior to the evaluation, the courtyard area

of the MOUT site was chosen as the environment to conduct this elemental test. The ~45m square area contains 10-single story and two-double-story buildings. Each building had several doors and windows. Various pieces of furniture (e.g. chairs, desks, and tables) were distributed throughout the buildings. Approximately 50 waypoints were marked with two-centimeter accuracy using differential GPS and surveying equipment. The waypoints included a range of indoor, outdoor, ground-level, and upper-story locations (including positions in front of doorways, windows and other building features). These waypoints were used to mark the locations from which imagery would be captured by the ASSIST-wearer, and the locations of additional objects to be placed in the environment. Additional objects in the environment included vehicles (both civilian and military) with license plates (both US and Iraqi), people (soldiers and civilians dressed in simulated middle-eastern attire), weapons (both US military and foreign that were either carried by people or placed within the environment), IED materials (spools of wire, wire cutters, duct tape, etc.), simulated pipe bombs, Arabic signs, tires (both stacked vertically and resting against buildings), trash piles, barrels, boxes (various sizes) and sandbag piles, etc.

Imagery was collected from 25 viewpoints. The 25 viewpoints were distributed across 10 waypoints, each of which have multiple viewpoints to capture data from different orientations. Each team collected a single data set (image) at each of the 25 data collection viewpoints.

4) Sound Recognition

The goal of the sound recognition test was to evaluate the ASSIST system's ability to detect certain sounds in the environment.

To conduct this elemental test, the following sound events were scripted to occur in the environment at specified times relative to the start of a given evaluation run:

- A soldier fired blank rounds from one of three weapons: M240, M4, M107
- A soldier standing next to the ASSIST wearer spoke one of ten text phrase which incorporated some combination of the keywords listed above
- A person in the environment either spoke or played a digital voice recording of people speaking the languages listed above
- A soldier drove one of the vehicles specified above and either accelerated or decelerated past the ASSIST wearer.

There were 7 runs, each of increasing complexity. During the early runs, there was little or no ambient noise, the ASSIST wearer was stationary, there were no overlapping sounds, and most of the sounds in the environment occurred fairly close to the ASSIST wearer. During the later runs, there was a lot of ambient noise, the ASSIST wearer was moving, there were

overlapping sounds, and the sounds in the environment were moving to and from further distances from the ASSIST wearer. The last two runs in the evaluation incorporated the ASSIST wearer being in confined and indoor locations.

Ground truth locations of the ASSIST wearer and the sounds in the environment were measured based upon known points in the environment. Before the test, the locations of certain points in the environment were mapped out to specific GPS locations with two centimeters accuracy. These points were given letter tags. When stationary, the ASSIST wearer remained at a specific lettered point in the environment; when moving the ASSIST wearer moved between specific lettered points. Similarly, the sounds were generated at specific lettered locations, or moved between lettered locations.

5) Arabic Text Translation

The goal of the Arabic text elemental test was to evaluate the ASSIST system's ability to detect, recognize, and translate Arabic signs.

Three signs were placed in the environment at marked positions so that sets of images could be taken at known angles and distances from the signs. The first sign contained hand-printed characters, while the other two had machine-printed characters. One of the signs used a font known to be accepted by the optical character recognition (OCR) stage of the system.

The elemental test had three parts.

- *Sign Detection.* The signs were used to evaluate the ability of the system to extract text regions from signs.
- *Text Extraction.* The regions extracted from the signs were processed and the results evaluated. In addition, pictures of text were submitted to the OCR program. The output Arabic characters and words were compared with those on the signs. The fonts and point sizes of the text were controlled and were limited to those that the OCR system can handle.
- *Text Translation.* A set of Arabic words and sentences was input to the translation system in its preferred format and the resulting translations evaluated.

Note that in most cases, members of the research teams wore the technology, since the hardware at this stage was not intended to be hardened. Soldiers observed and guided the researchers in the elemental test activities to ensure a reasonable level of realism in the behaviors of the researcher wearing the technology.

B. Vignette Tests

The vignette tests were designed to assess the value of ASSIST systems in 1) infantry squad reporting of critical

information, events, and intelligence encountered during a mission, and 2) S2 (intelligence officer)/intelligence operations. These tests engaged soldiers in two realistic, albeit short, missions, where the ASSIST technologies were used to "shadow" the soldiers as they conducted the missions, and the S2 officer conducted debriefings post-mission. Additionally, a third vignette was employed to assess the contributions that ASSIST systems provided to another aspect of S2 responsibilities; data-gathering for a strategic product (actual production was not the focus here).

The scenario for Vignette 1 mimicked a presence patrol. The presence patrol included leaving a forward operating base (FOB) to patrol a local village, make the military presence known, and collect intelligence on the village and/or villagers before returning to the FOB. In Vignette 1, the soldiers were instructed to conduct a presence patrol in the market area of the village, and then conduct a deliberate search of the factory area.

The scenario for Vignette 2 focused on collecting intelligence about an Improvised Explosive Device explosion which had occurred overnight. The soldiers were instructed to gather detailed information about the IED event. Upon completion of that mission, they were to conduct a presence patrol in the market and factory areas of the village, while attempting to identify and/or detain several "gray list" and "black list" individuals.

As with the elemental tests, only the researchers wore the ASSIST systems unless otherwise requested. Each researcher was assigned a specific soldier to shadow during all parts of the mission.

After the vignettes were completed, the S2 was tasked with gathering data he would use to produce an intelligence report on the state of the village with respect to the upcoming election, including any related violence or unrest.

5) Soldier Test Procedures

For Vignettes 1 & 2, the following procedures were used:

- 1) A "simulated squad" of soldiers, comprised of two fire teams, with researcher 'shadows,' ran through an operationally-relevant scenario.
- 2) Upon completing the mission, the squad produced an after-action report, based the template provided.
- 3) Soldiers were asked to identify their information needs with respect to producing their report, e.g., information they would have preferred to include in their report but did not recall.
- 4) Each research team shared its processed data with the squad. Each soldier was asked to rate the importance of each information need and how well each ASSIST technology addressed each need

- 5) The soldiers participated in a semi-structured interview to get at more overall impressions from the exercise and ASSIST systems. The interview facilitator focused discussion on assessing if and how the after-action report produced by the squad would be different if the soldiers had been given access to ASSIST system functionality.

For Vignette 3, the following procedure was used:

- 1) The S2 was asked to identify information needs, e.g., information that would improve situation awareness, information about critical events, individuals, or situations, etc.
- 2) The S2 met with representatives of each research team to address the identified information needs.
- 3) The S2 was asked to rate the importance of each information need and how well the ASSIST system addressed each need.

Following the vignettes, the S2 participated in a semi-structured interview to capture his overall impressions of the ASSIST system capabilities and areas for improvement. The interview facilitator focused discussion on assessing if and how the S2's situation awareness and performance would be different if he were given access to ASSIST system functionality.

V. CONCLUSION

In this paper, we described the testing procedure that was implemented for Task 2 of the DARPA ASSIST program. The objective of the ASSIST program is to exploit soldier-worn sensors to augment a soldier's recall and reporting capability to enhance situational understanding in MOUT environments.

The following three metrics were the focus for the Task 2 evaluation:

- 1) The accuracy of object/event/activity identification and labeling
- 2) The system's ability to improve its classification performance through learning
- 3) The utility of the system in enhancing operational effectiveness

The IET developed a two-part test methodology to produce these metrics. Metrics 1 and 2 were evaluated through "elemental tests", and metric 3 was evaluated through "vignette tests". Elemental tests were designed to measure the progressive development of ASSIST system technical capabilities; and vignette tests were designed to predict the impact these technologies will have on warfighter's performance in a variety of missions and job functions. In specifying the detailed procedures for each elemental and vignette test, the IET attempted to define evaluation strategies that would provide a reasonable level of difficulty for system

and soldier performance at both the 6-month and 12-month evaluations.

The evaluation procedures described in this report were found to be very appropriate and successful at obtaining the information pertaining to the three metrics desired by DARPA. The separation of the technology evaluation (elemental test) from the utility tests (vignettes) allowed the IET to focus on these very important but also very different aspects separately, thus allowing for a better evaluation, from the IET's perspective.

The ASSIST program is expected to continue through at least 2009 and NIST expects to continue applying and refining these testing procedures as the project progresses.

ACKNOWLEDGEMENT

This work was supported by the Defense Advanced Research Projects Agency (DARPA) ASSIST program (Program Manager: Mari Maeda).

DISCLAIMER

Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

REFERENCES

1. DARPA, "Advanced Soldier Sensor Information System and Technology (ASSIST) Proposer Information Pamphlet," http://www.darpa.mil/ipto/solicitations/open/04-38_PI_P.htm, 2006.

A Two-Stage Approach to People and Vehicle Detection With HOG-Based SVM

Feng Han, Ying Shan, Ryan Cekander, Harpreet S. Sawhney, and Rakesh Kumar
Sarnoff Corporation
201 Washington Road
Princeton, NJ 08540, USA
{fhan, yshan, rcekander, hsawhney, rkumar}@sarnoff.com

Abstract—In this paper, we present a two-stage approach to robustly detect people and vehicles in static images using extended histogram of oriented gradient (HOG) and SVM for classification. The first stage is focus of attention generation, in which possible people and vehicle locations are hypothesized. This step uses stereo cue and generates potential target locations using some prior knowledge about what people and vehicle may look like in the depth map of the whole scene. The second stage is hypothesis verification. In this stage, all the hypothesis are verified by a strong classifier using extended HOG feature and SVM, which is robust to the wide range of variations of poses and viewpoints within people and vehicles. By adaptively combining the two stages, the final system achieves both speeding up and performance improvement. The system has been tested on some challenging datasets and illustrates good performance.

I. INTRODUCTION

Automatic object detection and classification is a key enabler for applications in robotics, navigation, surveillance, or automated personal assistance. On the other hand, automatic object detection is a difficult task. The main challenge is the amount of variation in visual appearance. An object detector must cope with both the variation within the object category and the diversity of visual imagery that exists in the world at large. For example, cars vary in size, shape, color, and in small details such as the headlights, grille, and tires. The lighting, surrounding scenery, and an object’s pose affect its appearance. A car detection algorithm must also distinguish cars from all other visual patterns that may occur in the world, such as similar looking rectangular objects.

The common approach to automatic object detection is shifting a search window over an input image and categorizing the object in the window with a classifier. To speed up the system without losing classification performance, one can exploit the following two characteristics common to most vision-based detection tasks: First, the vast majority of the analyzed patterns in an image belong to the background class. For example, the ratio of non-face to face patterns in the tests in [8] is about 50,000 to 1. Second, many of the background patterns can be easily distinguished from the objects. Based on these two observations, object detection is always carried out in a two-stage scheme as illustrated in Figure 1: First, all the regions in the image that potentially contain the target objects are identified. This is what we call “focus of attention’s mechanism”. Second, the selected regions are verified by a

classifier.

Numerous approaches to focus of attention generation have been proposed in the literature. Most of them fall into one of the following three categories: (1) knowledge-based, (2) stereo-based, and (3) motion-based. Knowledge-based methods make use of our knowledge about object shape and color as well as general information about the context. For instance, the prior knowledge that vehicles are symmetric about the vertical axis has been used in vehicle detection approaches using the intensity or edge map in [1], [2]. Stereo-based approaches usually employ the Inverse Perspective Mapping (IMP) [3] to estimate the locations of vehicles, people, and obstacles in images. One specific example is the work by Bertozzi et al. [4], in which the IMPs are computed from the left and right images respectively and compared with each other. Based on the comparison, the objects that were not on the ground plane can be easily found. With this information, the free space in the scene can be determined at the same time. Most motion-based methods detect objects such as vehicles, people, and obstacles using optical flow. However, generating a displacement vector for each pixel is time-consuming and also impractical for a real-time system. To attach this problem, some discrete methods use image features such as color blobs [5] or local intensity minima and maxima [6] as the basic unit and have produced some better results.

A number of different approaches to hypothesis verification that use some form of learning have been proposed in the literature. In these approaches, the characteristics of the object class are learned from a set of training images which should capture the intra-class variabilities. Usually, the variability of the non-object class is also modelled to improve performance. First, each training image is represented by a set of local or global features (e.g. Harr wavelet, SIFT, Shape Context) [8], [9], [16], [17] into some underlying configuration (e.g. “bag of features”, constellation model) [10], [11], [12], [13], [14]. Then, the decision boundary between the object and non-object classes is learned either by training a classifier (e.g., Adaboost, Support Vector Machine, Neural Network (NN)) or by modelling the probability distribution of the features in each class (e.g., using the Bayes rule assuming Gaussian distributions) [8], [11], [10]. These methods differ on the details of the features and decision functions, but more fundamentally they differ in how strictly the geometry

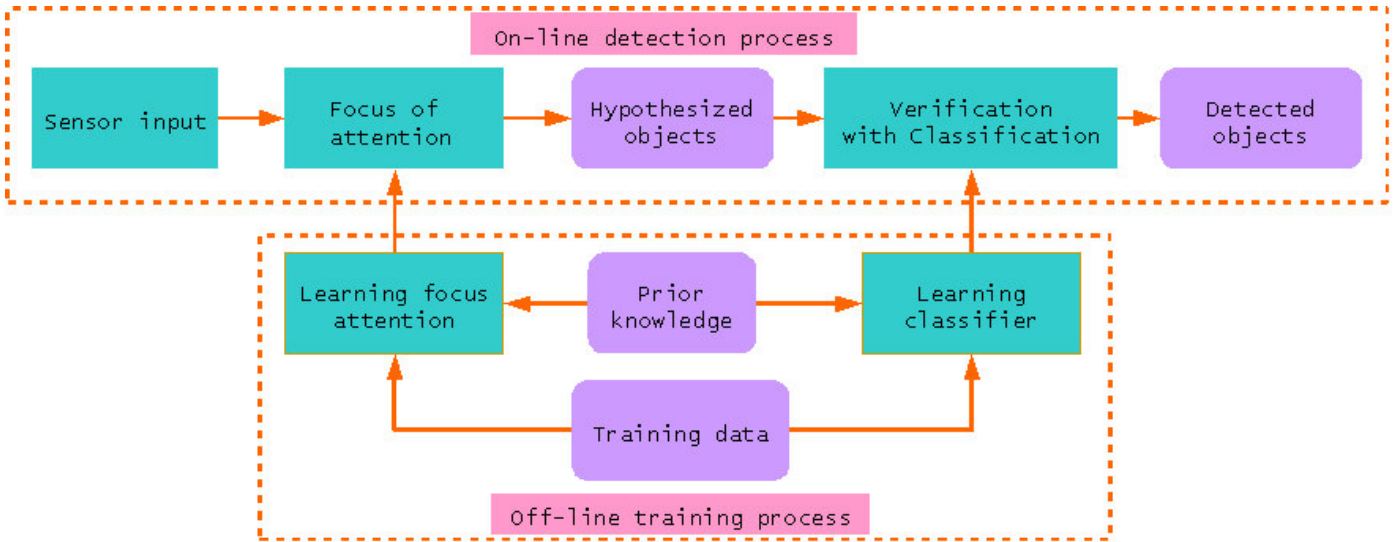


Fig. 1. Overview of the two-stage system for object detection.

of the configuration of parts constituting an object class is constrained.

Following the two-stage paradigm discussed above, we have built a system as illustrated in Figure 1 to stably detect standing people and vehicles over a wide range of viewpoints. The rest of paper is organized as follows: Section II presents focus of attention generation using stereo cue; Section III details the hypothesis verification with HOG-Based SVM; Section IV describes some implementation issues and a series of experiments; Section V concludes the paper.

II. FOCUS OF ATTENTION USING STEREO CUE

To generate focus of attention for the target objects in the scene, we use a stereo matching algorithm [7] to get the depth map of the scene. One example pair of left image and right image, and the depth map computed from this stereo pair are shown in Figure 2. In the depth map, red color implies closer points and blue to green implies further points.



Fig. 2. Compute depth map from the stereo images.

After getting the depth map of the scene, we can further align it with the ground plane with the help of the IMU attached with the stereo system, which gives us the pitch angle. Then we can remove the ground plane from the depth map. For the remaining depth map, we project it to the XZ plane and represent it with a uniform grid. For each cell in this grid, we compute the height and pixel density to get the ‘height map’ and ‘occupancy map’ as illustrated in Figure 3 (a) and (b) respectively. Then we compute the response of a predefined

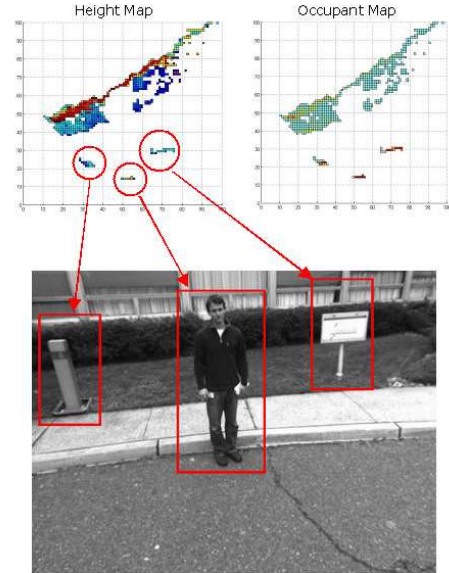


Fig. 3. Generate target hypothesis based on the height map and occupancy map.

adaptive Gaussian kernel on the ‘occupant map’. Finally, we choose peaks with local maximum response as the target object position hypothesis. An example result of this process is shown in Figure 3. Note that the algorithm has discovered relatively compact vertical objects.

III. CLASSIFICATION BY HOG-BASED SVM

We develop separate classifiers for each object class that are each specialized to one specific aspect or pose. For example, we have one classifier specialized to front/rear view of people and one that is specialized to side view of people. We apply these view-pose-based classifiers in parallel and then combine their results. If there are multiple detections at the same or

adjacent locations, the system selects the most likely one through non-maximum suppression.

We empirically determined the number of views/poses to model for each object. For people we use two view-based detectors: front/rear and side view, as shown in Figure 4. For cars we use eight detectors, which are specialized to each of the eight aspects shown in Figure 5.

Each of these detectors is not only specialized in orientation, but is trained to find the object only at a specified size within a rectangular image window. Therefore, to be able to detect the object at any position within an image, we re-apply the detectors for all possible positions of this rectangular window. Then to be able to detect the object at any size we iteratively resize the input image and re-apply the detectors in the same fashion to each resized image.



Fig. 4. Examples poses for people.

To build each view-pose-based classifier, we extend the histogram of oriented gradient (HOG) [15] representation and use support vector machines (SVM) as the classifier [20], [21]. Unlike some commonly used representations, the extended histogram of oriented gradient gives good generalization by grouping only perceptually similar images together. With a support vector machine, this gives rise to a decision function that discriminates object and non-object patterns reliably in images under different kinds of conditions and results good performance on some challenging datasets.



Fig. 5. Example viewpoints for vehicles.

A. Object Class Representation

1) *HOG feature*: Histogram of oriented gradient (HOG) is an adaptation of Lowe’s Scale Invariant Feature Transformation (SIFT) approach to wide baseline image matching [16] with local spatial histogramming and normalization. In this work, HOG is used to provide the underlying image patch descriptor for matching scale invariant key points. SIFT-style approaches perform remarkably well in this application.

A HOG feature is created by first computing the gradient magnitude and orientation at each image sample point in a region around an anchor point. The region is split into $N \times N$ subregions. An orientation histogram for each subregion is then formed by accumulating samples within the subregion, weighted by gradient magnitudes. Concatenating the histograms from all the subregions gives the final HOG feature vector as illustrated in Figure 6.

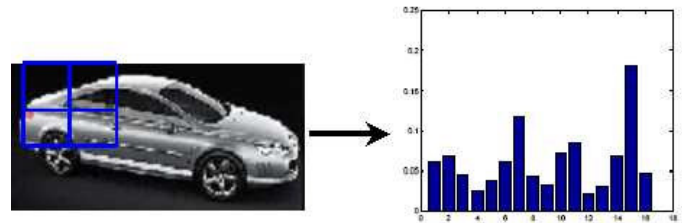


Fig. 6. HOG feature computation and structure.

2) *Extend HOG by Incorporating Spatial Locality*: The standard HOG feature only encodes the gradient orientation of one image patch, no matter where this orientation is from in this patch. Therefore, it is not discriminative enough if the spatial property of the underlying structure of the image patch is crucial. This is especially true for highly structured objects like vehicles. To incorporate the spatial property in HOG feature, we add one distance dimension to the angle dimension in the binning of all the pixels within each subregion. The distance is relative to the center of each subregion. The new binning process is illustrated in Figure 7.

3) *Dense Grid Representation*: Following [15], we divide the image window into small spatial regions, which consists of a number of subregions (or cells). For each cell we accumulate a local 1-D histogram of gradient directions over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram over somewhat larger spatial regions (or blocks) and using the results to normalize all of the cells in the block.

B. SVM Classifier

In this paper we choose the support vector machine [20], [21] as the classifying function. The Support Vector Machine

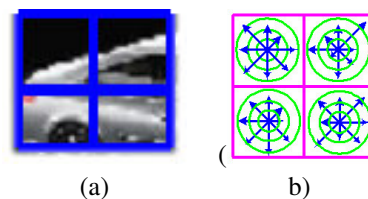


Fig. 7. Binning both distance and gradient direction for the pixels in each sub-region to compute the extended HOG feature. (a) sample image patch. (b) distance and direction ranges to do the binning.

(SVM) is a statistical learning method based on the structure risk minimization principle. Its efficiency has been proved in many pattern recognition applications [20], [22], [23]. In the binary classification case, the objective of the SVM is to find a best separating hyperplane with a maximum margin.

The form of a SVM classifier is:

$$y = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i K(x, x_i) + b\right),$$

where x is the feature vector of an observation example, $y \in \{+1, -1\}$ is a class label, x_i is the feature vector of the i^{th} training sample, N is the number of training samples, and $K(x, x_i)$ is the kernel function. Through the learning process, $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ is computed.

One distinct advantage this type of classifiers has over traditional neural networks is that support vector machines achieve better generalization performance. While neural networks such as multiple layer perceptrons (MLPs) can produce low error rate on training data, there is no guarantee that this will translate into good performance on test data. Multiple layer perceptrons minimize the mean squared error over the training data (empirical risk minimization) where support vector machines use an additional principal called structural risk minimization [21]. The purpose of structural risk minimization is to give an upper bound on the expected generalization error.

Compared with the popular Adaboost classifiers, SVM is slower in test stage. However, the training of SVM is much faster than that of Adaboost classifiers.

IV. EXPERIMENTS

A. Training

Our people training database contains images of 2000 standing people with various aspects, poses, and illumination conditions. Some of these images are from the public downloadable MIT people dataset and INRIA people dataset, while the rest are taken by ourselves. The resolution of each image is 64x128. For the vehicle training data, we collected 1000 images with 128x64 resolution and containing four types of vehicles (sedan, minivan/SUV, pick-up truck and U-Haul type truck) across a wide range of viewpoints. We also generate some rendered vehicle images, some of which are shown in Figure 8, by 3D vehicle models and use them as training data. Using this type of virtual training data is crucial since sometimes it is too time consuming or even impossible to get normal training data covering all possible pose-view variations for some object classes. The performance of vehicle classifier trained using these rendered images is tested in Section IV-D.



Fig. 8. Samples of rendered vehicle images.

One very important issue in the classifier training for one object class is how to select effective negative training samples. As negative training samples include all kinds of images, a prohibitively large set is needed in order to be representative, which would also require infeasible amount of computation in training. To alleviate this problem, a bootstrapping method, proposed by Sung and Poggio [24], is used to incrementally train the classifier as illustrated in Figure 9.

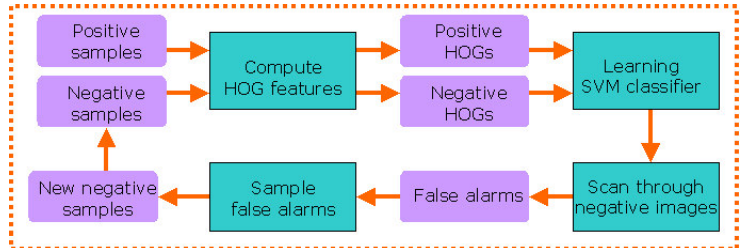


Fig. 9. The bootstrap training diagram.

All the view-pose-based SVM classifiers for each object class are trained separately, but with the same negative training samples. In this way, their outputs can compete with each other to remove multiple detections through non-maximum suppression.

B. Detection

As each specific view-pose-based classifier for every object class is designed on an image window with specific size (64x128 for people, 128x64 for vehicle), it implicitly requires that the to-be-detected objects lie roughly within a specific window in the testing images. To detect all the objects appearing at different scales in the test image, we build an image pyramid by successively up sampling and/or down sampling the test image by a factor of 1.2 till all the objects in the test image are scaled to the image window size at some layer in the pyramid.

C. Evaluation of Detection Results

Evaluation of detection results was performed using ROC curve analysis. The output required to generate such curves is a set of bounding boxes with corresponding “confidence” values, with large values indicating high confidence that the detection corresponds to an instance of the object class of interest. Figure 10 shows some example ROC curves, obtained by applying a set of thresholds to the confidence output by the SVM classifier. On the x-axis is plotted the average number of false alarms on one image; on the y-axis is detection rate. The ROC curve makes it easy to observe the tradeoff between the two; some thresholds may have high detection rate but more false alarms, while other thresholds may give more balanced performance.

To generate the ROC curves, we also need a criteria to evaluate the detection output. Judging each detection output by a method as either a true positive (object) or false positive (non-object) requires comparing the corresponding bounding box predicted by the method with ground truth bounding boxes

of objects in the test set. To be considered a correct detection, the area of overlap α_{ovlp} between the predicted bounding box B_p and ground truth bounding box B_{gt} was required to exceed 50% by the formula used in [25],

$$\alpha_{ovlp} = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}.$$

In [25], the threshold of 50% was set low to account for inaccuracies in bounding boxes in the ground truth data. This inaccuracy of ground truth is due to some ambiguities, for example defining the bounding box for a highly non-convex object, e.g. a side view of a motorbike or a car with an extended radio aerial.

D. Performance of People and Vehicle Classifier

To test the performance of the trained classifier for people, we apply it to one people dataset selected from INRIA people database and PSACAL database [26]. This people dataset consists of 800 images and most people in the images are standing or walking. The performance curve of the people classifier is shown in Figure 10 (a).

To test the performance of the trained classifier for vehicle, we apply it to two vehicle datasets. The first one is the UIUC dataset [11], which consists of 278 images of vehicles in side-view. The second one consists of 600 images selected from PSACAL database [26] and vehicles appear in any poses in the images. The performance curves of the vehicle classifier on these two datasets are shown in and Figure 10 (b) and (c) respectively.

In all the above testing, we search the whole image without using any focus of attention. Some typical results for these two classifiers on the three datasets are shown in Figure 11 and Figure 12 respectively.

We also test the performance of classifiers using the rendered images as training data. To do this, we use the rendered vehicle images to train a vehicle classifier and apply it to the selected PASCAL dataset. The performance curves of this classifier and then one using normal images as training data are shown in Figure 10 (d) together for comparison, from which we can see that the classifier using rendered images as training data can achieve compatible performance.

E. Performance of the final two-stage system

To test the performance of the two-stage system, we apply it on 100 images that contain both standing people and vehicles spanning a variety of viewpoints. To show the performance improvement achieved by incorporating the first stage of focus of attention generation by stereo cue, we compare the performance of the system by turning on and off the first stage. In Figure 13 (a), we show the two ROC curves corresponding to turning on and off the first stage in the system for people detection. From these two comparisons, we can clearly see that the focus of attention generation stage helps a lot to reduce false alarms. Figure 13 (b) shows the same case for vehicle detection. Some typical detection results in this testing are shown in Figure 14 and Figure 15 for people detection and vehicle detection respectively.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a two-stage approach to robustly detect standing people and vehicles over a wide range of viewpoints. The first stage is focus of attention generation, in which possible people and vehicle locations are hypothesized. This step uses stereo cue and generates potential target locations using some prior knowledge about what people and vehicle may appear in the depth map of the whole scene. The second stage is hypothesis verification. In this stage, all the hypothesis are verified by a strong classifier using extended HOG feature and SVM, which is robust to the wide range of variations of poses and viewpoints within people and vehicles.

Although the current two-stage system works reasonably well, the process to manually separate the training samples into pre-defined intra-class categories based on their view/pose is too time consuming and inherently ambiguous. In addition, the errors caused by improperly defined categories and incorrectly assigned labels will eventually be propagated into the final classifier and deter the object detection performance. Recently, we have proposed a novel computational framework that unifies automatic categorization, through training of a classifier for each intra-class exemplar, and the training of a strong classifier combining the individual exemplar-based classifiers with a single objective function [27]. We are working to incorporate the current classifiers into the unified framework to dramatically reduce the training time and improve the performance. Furthermore, we are also working on incorporating motion and video constraints in classification. Increasing the number of detectable object classes to a large set is also a goal.

REFERENCES

- [1] A. Kuehnle, "Symmetry-based recognition for vehicle rears," *Pattern Recognition Letters*, vol. 12, pp. 249258, 1991.
- [2] T. Zielke, M. Brauckmann and W. V. Seelen, "Intensity and edge-based symmetry detection with an application to carfollowing," *CVGIP:Image Understanding*, vol. 58, pp. 177190, 1993.
- [3] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological Cybernetics*, vol. 64, no. 3, pp. 177185, 1991.
- [4] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. on Image Processing*, vol. 7, pp. 6281, 1998.
- [5] B. Heisele and W. Ritter, "Obstacle detection based on color blob flow," *IEEE Intelligent Vehicles Symposium*, pp. 282286, 1995.
- [6] D. Koller, N. Heinze and H. Nagel, "Algorithm characterization of vehicle trajectories from image sequences by motion verbs," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 9095, 1991.
- [7] G. van der Wal and M. Hansen and M. Piacentino, "The Acadia Vision Processor", *International Workshop on Computer Architectures for Machine Perception*, September 2000.
- [8] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". *In Proc. CVPR*, pages 511-518, 2001.
- [9] P. Viola, M. Jones, and D. Snow. "Detecting pedestrians using patterns of motion and appearance". *In Proc. International Conference on Computer Vision*, volume 2, pages 734-741, 2003.
- [10] M. Weber, M. Welling, and P. Perona. "Unsupervised learning of models for recognition". *In Proc. ECCV*, pages 18-32, 2000.
- [11] S. Agarwal, A. Awan, and D. Roth. "Learning to detect objects in images via a sparse, part-based representation". *IEEE PAMI*, 26(11):1475-1490, Nov. 2004.
- [12] S. Agarwal and D. Roth. "Learning a sparse representation for object detection". *In Proc. ECCV*, volume 4, pages 113-130, 2002.

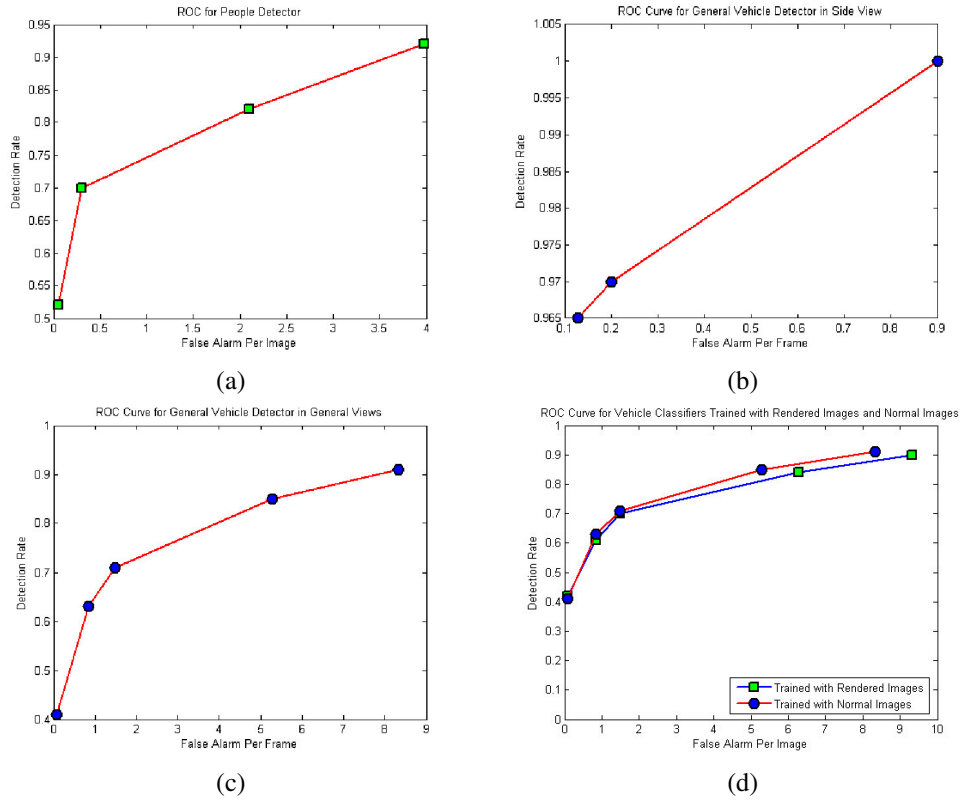


Fig. 10. (a) ROC curve for people detection without using stereo cue to generate focus of attention. (b), (c) ROC curves for vehicle detection on UIUC dataset and select dataset from PSACAL database respectively without using stereo cue to generate focus of attention. (d) ROC curves for vehicle detection on selected dataset from PASCAL database with two classifiers using normal vehicle images and rendered vehicle images as training data respectively.

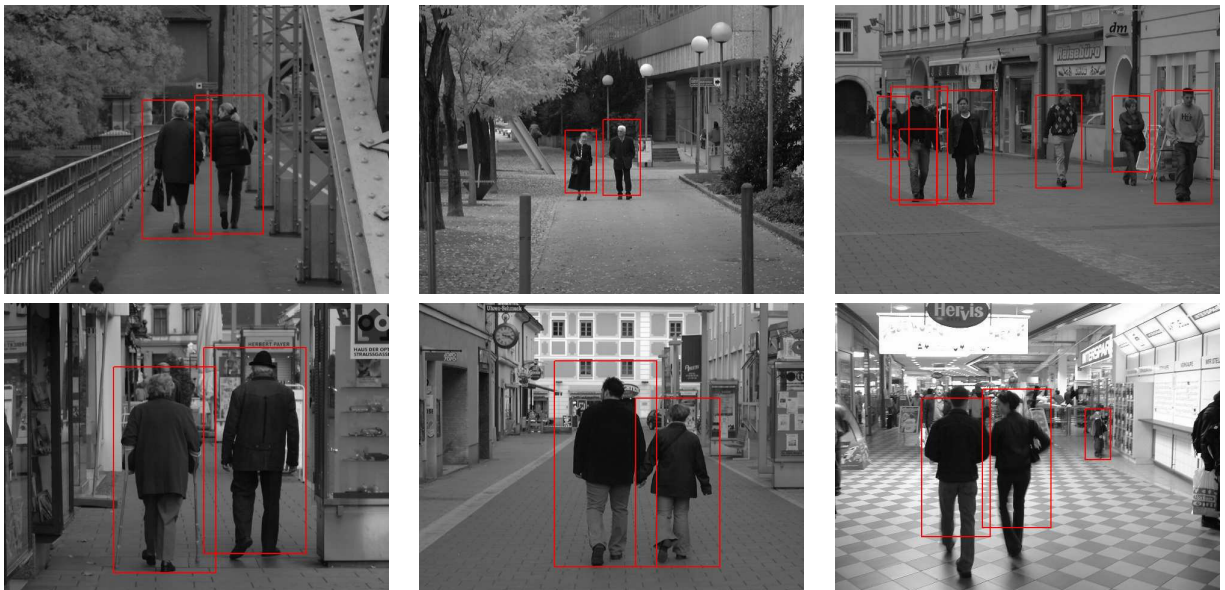


Fig. 11. People Detection Results Without focus of attention Stage

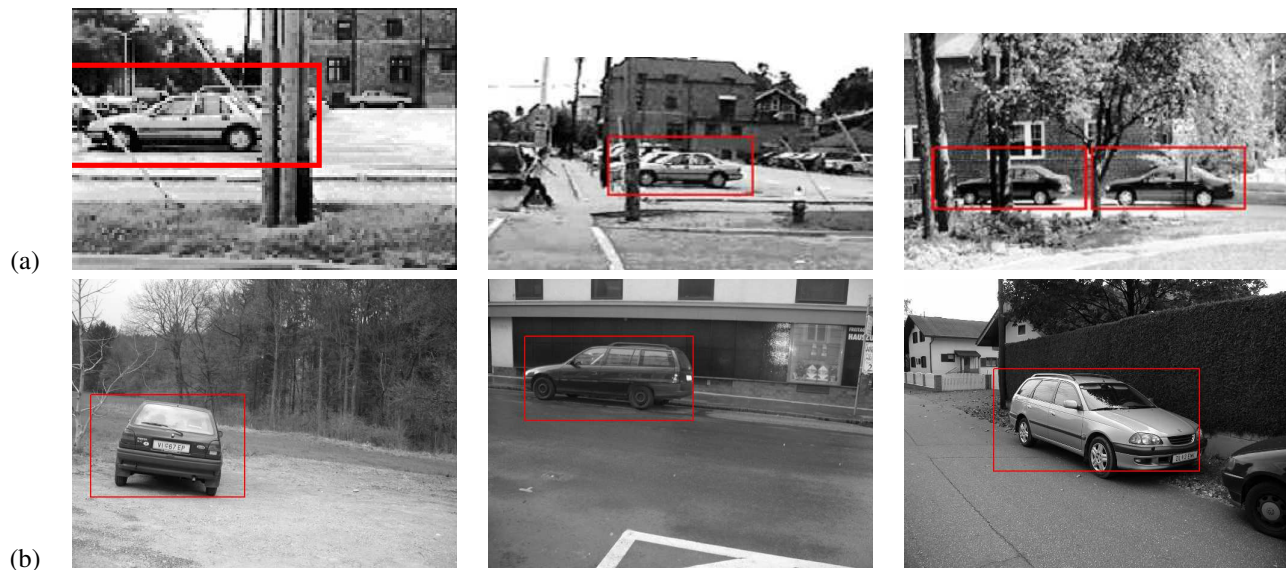


Fig. 12. Vehicle Detection Results Without focus of attention Stage. (a) Results on UIUC dataset. (b) Results on selected dataset from PASCAL database.

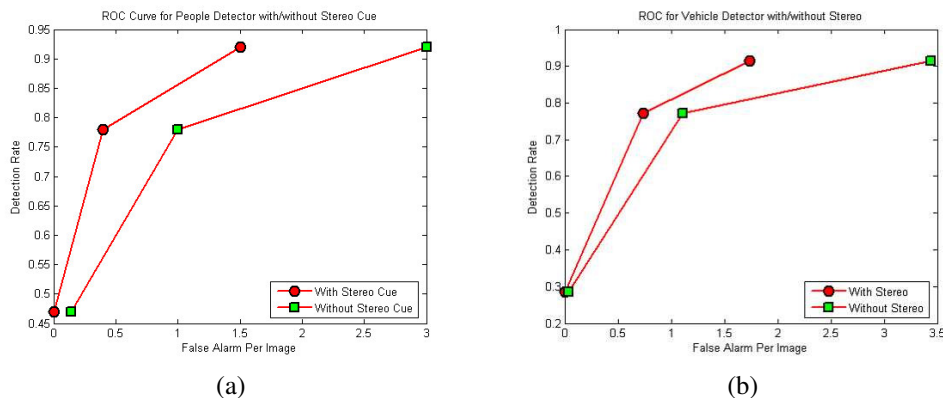


Fig. 13. (a) ROC curve for people detection with and without using stereo cue to generate focus of attention. (b) ROC curve for vehicle detection with and without using stereo cue to generate focus of attention.

[13] B. Leibe, E. Seemann, and B. Schiele. "Pedestrian detection in crowded scenes". In *CVPR*, pages 878-885, 2005

[14] B. Leibe, A. Leonardis, and B. Schiele. "Combined object categorization and segmentation with an implicit shape model". In *ECCV'04 Works. on Stats Learning in Comp. Vision*, pages 17-32, May 2004.

[15] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In *Proc. CVPR*, volume 1, pages 886-893, 2005.

[16] D. Lowe. "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision*, 60(2):91-110, Nov. 2004.

[17] S. Belongie, J. Malik, and J. Puzicha. "Shape matching and object recognition using shape contexts". *IEEE PAMI*, 24(4):509-522, 1998.

[18] W. T. Freeman and M. Roth. "Orientation histograms for hand gesture recognition". *Intl. Workshop on Automatic Face and Gesture Recognition*, IEEE Computer Society, Zurich, Switzerland, pages 296-301, June 1995.

[19] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. "Computer vision for computer games". *2nd International Conference on Automatic Face and Gesture Recognition*, Killington, VT, USA, pages 100-105, October 1996.

[20] E. Osuna, R. Freund, and F. Girosi. "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130-136.

[21] V. Vapnik, "The nature of statistical learning theory". *New York: Springer-Verlag*, 1995.

[22] B. Heisele, T. Serre, S. Prentice, and T. Poggio. "Hierarchical classification and feature reduction for fast face detection with support vector machines". *Pattern Recognition*, 36(9):2007-2017, Sep 2003.

[23] S. Romdhani, P. H. S. Torr, B. Scholkopf, and A. Blake. "Computationally efficient face detection". In *Proc. ICCV*, volume 1, pages 695-700, Jul 2001.

[24] K. K. Sung and T. Poggio. "Example-based learning for view-based human face detection", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39-51, 1998.

[25] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang. "The 2005 pascal visual object classes challenge". In *In Selected Proceedings of the First PASCAL Challenges Workshop*, LNAI, Springer-Verlag.

[26] <http://www.pascal-network.org/challenges/VOC/databases.html>.

[27] Y. Shan, F. Han, H. S. Sawhney, and Rakesh Kumar. "Learning Exemplar-Based Categorization for the Detection of Multi-View Multi-Pose Objects", *IEEE Conference on Computer Vision and Pattern Recognition*, NYC, 2006.

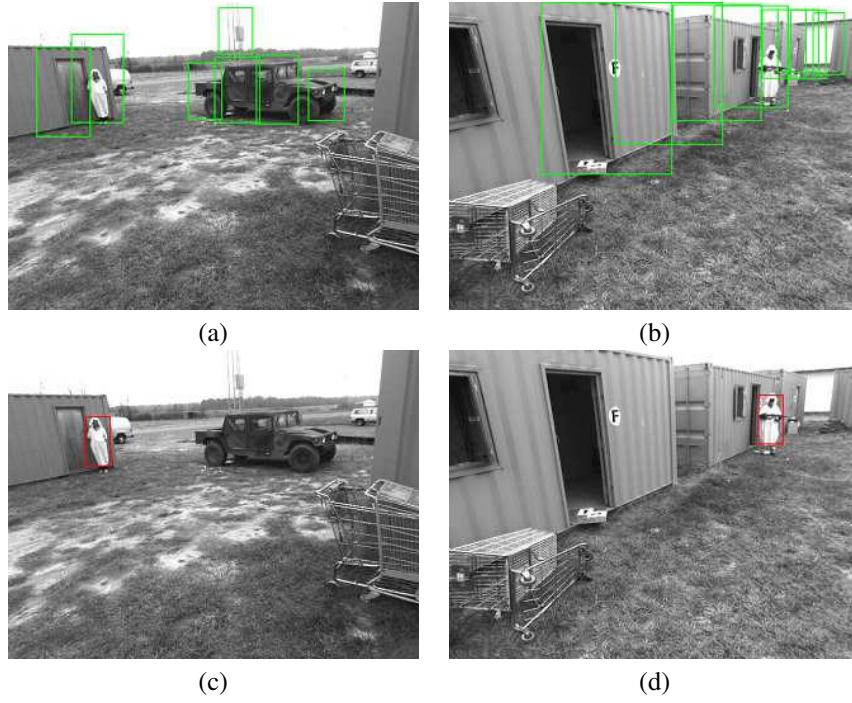


Fig. 14. (a),(b) Focus of attention for people by stereo cue. (c),(d) Final People Detection Results.

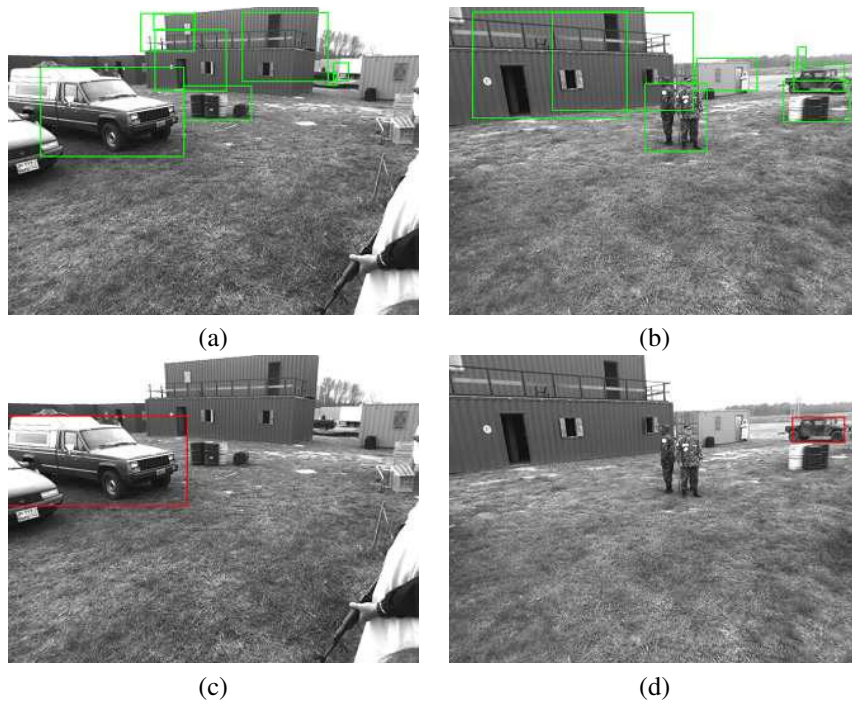


Fig. 15. (a), (b) Focus of attention for vehicle by stereo cue. (c), (d) Final Vehicle Detection Results.

Performance Metrics and Evaluation Issues for Continuous Activity Recognition

David Minnen, Tracy Westeyn,
and Thad Starner
GVU, College of Computing
Georgia Institute of Technology
{dminn,turtle,thad}@cc.gatech.edu

Jamie A. Ward
Swiss Federal Institute of Technology (ETH)
Wearable Computing Lab
ward@ife.ee.ethz.ch

Paul Lukowicz
University of Passau
Passau, Germany
paul.lukowicz@uni-passau.de

Abstract—In this paper we examine several factors that influence the evaluation of multi-class, continuous activity recognition. Currently, there is no standard metric for evaluating and comparing such systems, although many possible error formulations and performance metrics could be adapted from other domains. In order to make progress toward a standard metric appropriate for evaluating activity recognition, we outline the sources of errors in such systems, present different methods for detecting and labeling these errors, and compare existing metrics with a more nuanced performance visualization. We conclude with a discussion concerning the interpretation of the visualization for comparing recognition systems in different domains.

I. INTRODUCTION

Activity recognition describes the problem of detecting and identifying activities in time-varying sensor data. Throughout this paper, we will focus on human activities such as walking, running, driving, hammering, performing particular American Sign Language (ASL) signs, opening a door, *etc.* Our discussion of activity recognition metrics, however, extends to other behaviors and is not restricted to human activity.

The evaluation of an activity recognition system requires that the output of the system be compared to ground truth labels representing the actual activities performed during the time period of interest. This comparison is complicated, however, since activity recognition is temporally continuous (*i.e.*, both the label and the temporal boundaries of each activity must be determined) and since there are typically many different activities to be detected. We can contrast this situation with both isolated recognition, where the activity boundaries are predetermined, and with binary classification problems in which there are only two classes.

The evaluation is complicated further by the requirements of activity recognition systems in different domains. A sign language recognition system may be judged by how well the system recovers understandable utterances. The activities in this case are relatively dense (continuous signing) with a large number of classes (the signs). The exact time that each sign starts and stops is not as important as interpreting the meaning of the signed phrase. On the other hand, a computer vision system that monitors swimming pools for drowning victims has very different requirements. While drownings are rare, the system should never miss a potential incident. On the other

hand, some false alarms are acceptable since the lifeguard can quickly ascertain if someone is actually in danger.

Figure 1 provides a simple illustration of the evaluation problem that we are addressing. The diagram shows the ground truth events in the top row, followed by the activities predicted by three different recognition systems in the subsequent rows. The difficulty of evaluating such systems is illustrated because all three have equivalent performance according to a widely-used metric (frame-based accuracy, discussed in detail in Section V). We can see from the visualization, however, that each system makes different kinds of errors that may be more or less significant depending on the particular application domain. Thus, we argue that it is important to consider the assumptions underlying potential evaluation methodologies and that relying on summary statistics computed over coarse error categories can sometimes mislead performance analysis.

Many of the motivating and illustrative examples used in this paper are drawn from DARPA's Advanced Soldier Sensor Information System and Technology (ASSIST) project. In this project, soldiers utilize body-worn sensors to augment their after-action recall and reporting capability. Sensors include accelerometers, microphones, high resolution still cameras, video cameras, GPS, altimeter, and a digital compass. With ASSIST, the soldier may return from a five hour patrol and ask the system to display all the images taken immediately before he raised his weapon. The soldier may use these images to help generate a report of the important events that happened during the patrol. Working with NIST, DARPA, and subject matter experts, we identified 14 activities of interest: walking, running, driving, sitting, crawling, lying, walking up stairs, walking down stairs, situation assessment, a weapon up state, kneeling, shaking hands, opening a door, and standing still.

Obviously, activity recognition applications vary greatly, and system designers should be aware of the variety of tools available to evaluate these systems. In this paper we will review many of the common metrics currently in use and discuss a recently developed visualization method that has proven useful for our work with the ASSIST project.

The remainder of the paper is organized as follows. Section II outlines fundamental issues such as sources of error and other factors that complicate the evaluation of activity recognition results. Next, in Section III, we discuss three

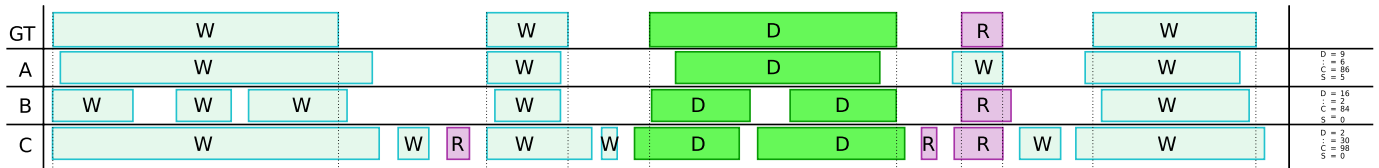


Fig. 1. Sample ground truth (GT) labels for a simple domain that includes walking (W), driving (D), running (R), and the null class (unlabeled). Hypothetical predicted labels are shown for three different recognition systems (A, B, and C). Each system has the same accuracy (66%), but the types of errors that each system produces varies dramatically. The total number of correct frames as well as deletion, insertion, and substitution errors (see Section II-B.2) is shown to the right of the output for each system.

different levels of analysis that can be used as the basis for evaluation. Error types are discussed in Section IV with common summary statistics presented in Section V. Finally, we introduce a recently developed method for performance visualization in Section VI and discuss how it might be used to evaluate and compare activity recognition systems.

II. ISSUES IN EVALUATING ACTIVITY RECOGNITION

A fundamental issue for evaluating activity recognition concerns the level of analysis used to calculate performance (see Figure 2). Each occurrence of an activity represents one *event*, which is a contiguous block of time during which the activity label is constant. Evaluation could measure whether each event is detected, whether the event is detected at the correct time, and how closely the predicted event boundaries correspond to the true start and stop times. We call evaluation at this level *event analysis*. See the top row of Figure 2 for an illustration of events detected at the correct time but with poor boundary alignment.

Alternatively, we can view the data as a series of equal-length time intervals and consider the activity being performed during each interval. For example, we could divide an hour of data into 3,600 seconds and label each one-second block according to the dominant activity during that time. Evaluation would then depend on the correspondence between the ground truth label and predicted label for each second. We call evaluation at this level *frame analysis*. Note that the temporal duration of each frame is arbitrary. As discussed in Section II-A.3, however, inappropriately large values can artificially affect the inherent error of the evaluation.

Finally, a hybrid approach divides the data into variable length segments. The segments are defined as maximal intervals within which both the predicted and true labels are

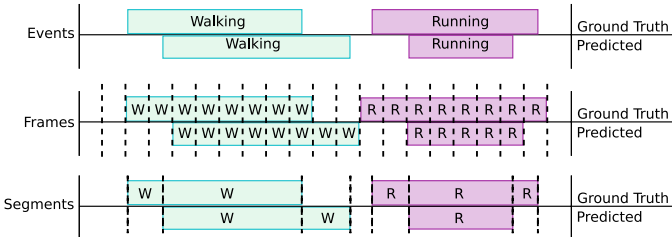


Fig. 2. An illustration of the three levels of analysis: events, frames, and segments.

constant. Thus the boundary of each segment coincides with a boundary of either a true or predicted label. Evaluation at this level is called *segment analysis* [1].

A. Intrinsic Sources of Error

Deviation between the ground truth and predicted labels can arise for many reasons. Some of the differences correspond to important errors in the recognition system and should be represented in the evaluation metric. Others arise due to inherent evaluation difficulties or semantic inconsistency and may mask the abilities of a particular recognition system. In this section, we outline sources of both kinds of errors.

1) *Errors in Ground Truth*: A fundamental problem for all evaluation systems concerns obtaining accurate ground truth data. For activity recognition, this includes correctly labeling each event and specifying accurate event boundaries at the desired temporal resolution. Generating ground truth is a notoriously difficult task and is of paramount importance for accurate evaluation. Errors often arise due to clock synchronization issues, limited human reaction time, and imprecise activity definitions. For example, if a subject transitions from walking to running, boundary errors may arise because of reaction delays (the evaluator must perceive the transition, recognize it, and then mark the transition time) and because of the subjective nature of the transition (does running start when the subject first leans in to run, the first time both feet leave the ground, as soon as his velocity exceed a particular threshold, or at some other point?). Worse yet, different evaluators may choose different transition definitions.

2) *Transition Effects*: Just as human evaluators may have difficulty labeling the precise temporal boundaries of each activity, these transition effects also lead to automatic recognition difficulties. Agreement between the performance metric and recognition system is vital. Should a strict division point be located during the transition? Should the transition be ignored (*i.e.*, it is recognized as neither the preceding nor following activity), or should it be explicitly recognized as a transition (*i.e.*, each transition is treated like an extra activity to be recognized)?

3) *Temporal Resolution*: Errors due to mismatched temporal resolution between the ground truth and predictions can further exacerbate boundary errors. Simply put, the recognition system should report activity boundaries at the same temporal resolution as the ground truth, else boundary errors are inevitable. As an extreme example, consider a true event

occurring from 47.3 to 102.5 seconds from the beginning of the data sequence. If this is used to evaluate a recognition system with minute-long frames, the minimum boundary error is 30.2 seconds (12.7 seconds at the beginning and 17.5 seconds at the end of the true event). This lower-bound can be easily reduced to 800ms by simply reporting predicted boundaries on a per second basis.

4) *Temporal Quantization*: In frame analysis, the atomic nature of each frame creates an additional complication. Regardless of the actual frame duration, an activity transition can occur within a single frame. In such cases, a single label must still be assigned to the frame. However, minor differences in the estimation of the boundary may change the overall label. In the extreme case when an inappropriately long duration is used, multiple activities might occur within a single frame. While decreasing the frame duration can reduce these errors, it cannot eliminate them entirely.

5) *Parallel Activity*: Our discussion of performance metrics for activity recognition is restricted to the case of temporally exclusive activities. That is, it is assumed that the subject is either walking or running or driving, *etc.*, but never performing multiple activities at the same time. This means that the recognition system should report one activity for each time interval. However, in actuality the subject may perform multiple activities in parallel. For example, the ASSIST evaluation was intended to follow the temporal exclusivity assumption, but at times the soldiers would walk with their weapons up, and naturally everyone would sit while driving. In such cases the correct label is inherently ambiguous. Methods for resolving such ambiguity and accounting for it during evaluation are discussed in Section VII.

B. Recognition Errors

In addition to the intrinsic errors just described, there are many sources of errors based on the events predicted by the recognitions system. These errors are typically much more serious and thus their classification is important for comparing different methods.

1) *Boundary Correspondence*: Even when the predicted label matches ground truth, the precise start and stop times may not align (see top of Figure 2). For event analysis, care must be taken to appropriately match predicted events with ground truth, a non-trivial correspondence problem in general. Several approaches for handling boundary mismatches are discussed in Section III.

2) *Label Correspondence*: Serious errors can occur when the predicted activity differs from the true activity at a particular time (a substitution error), when an activity is predicted when none actually occurred (an insertion error), when an activity is detected multiple times (also an insertion error), and when nothing is predicted when an activity really did occur (a deletion error). See Figure 1 for examples of each of these errors.

3) *Fragmentation*: The recognition system may erroneously divide an activity into multiple short intervals, especially when an activity has a very long duration. For example,

a subject may walk for 15 minutes, but slight irregularities may cause a prediction of three walking segments each lasting roughly five minutes (see System B in Figure 1). Tripping, stumbling, stepping over an obstacle, pausing, making a quick turn, or a myriad other minor deviations may be the source of such false division. For evaluation purposes, one must decide whether such irregularities constitute ground truth errors that should not count against the recognition system or if they are noteworthy mistakes. In general, this decision is highly contextual. If the predictions are used to answer questions such as “How many times did the soldier raise his weapon?” then fragmentation may cause serious over-estimation. On the other hand, for the purposes of answering “How much time was spent running?” fragmentation due to brief interruptions is a minor issue.

4) *Merging*: Merging is the opposite of fragmentation and occurs when two separate but closely occurring instances of the same activity are predicted as a single occurrence. For example, a subject may shake one person’s hand and then shake another in quick succession. A merge error occurs when the recognition system identifies this as a single, longer instance of shaking hands.

C. Domain Properties that Affect Evaluation

In addition to the sources of error presented in the previous section, several other issues complicate activity recognition evaluation. These include the existence of a “null” class, the semantic interpretation of the activity, the relative sparseness of the event (prior likelihood), and the cost of incorrectly classifying the event.

1) *Inclusion of a null class*: The null class represents the time during which none of the predefined activities occur. For example, if the system should report walking, running, and crawling, then the null class will include standing and driving but also scratching your head, playing ping-pong, petting your dog, and literally everything else that a person can do other than walking, running, and crawling.

In continuous recognition systems without a null class, insertion errors can occur when a long activity is falsely detected as several different, shorter activities. Similarly, a deletion error occurs when multiple different short activities are detected as a single long activity. Introduction of a null class complicates this by creating new kinds of insertion and deletion errors. Specifically, a known activity can be falsely detected during a period when an unknown (*i.e.*, null) activity occurs. Equivalently, a known activity can be deleted by incorrectly predicting null during the relevant time period.

Evaluation is further complicated because errors relative to the null class may be less serious than those relative to other classes. For instance, overextending the ending time of running when followed by null may be less critical than when running is followed by raising your weapon. In the latter case, the overextension of running would obscure the onset of an important event. This creates evaluation nuances that may mislead analysts if ignored.

2) *Semantic Differences*: Semantic differences between activities also affect evaluation. Activity classes can be divided into two categories according to whether they are “bounded” or “fluent.” Bounded activities are often gestural and tend to express a particular meaning or achieve a certain goal. For example, opening a door, signing a word in American Sign Language, and shooting a basketball are all examples of bounded activities. Erroneously detecting multiple occurrences of a bounded activity (*i.e.*, a merge or fragmentation error) is quite serious since it would reflect the interpretation that you, for example, opened several doors, signed multiple words, or shot many basketballs.

On the other hand, fluent activities typically have highly variable duration and are repetitive or periodic. For instance, standing, walking, running, and shaking hands are all fluent activities. Ideally, fluent activities should also be detected once and with accurate boundaries, however fragmentation is typically less serious. If a recognition system detects two instances of walking when only one longer instances occurs, it is still correct in that the subject was walking during both instances. This differs from a bounded activity in that the fragmentation does not imply an erroneous duplication of intent or action.

3) *Activity Frequency*: The frequency with which an event occurs can affect the interpretation of the evaluation. Consider a data sequence in which we are interested in detecting walking and standing. If standing occurs 90% of the time then many metrics may give misleading results. For instance, a very poor recognition system may simply recognize everything as standing, thereby achieving a high accuracy. Most would agree, however, that this recognizer would not generalize well and would have little practical value.

4) *Misclassification Cost*: In general, the cost associated with a labeling mistake may not be uniform across all activity pairs. For instance, it may be much more important for a system that recognizes “raising a weapon” and standing to correctly detect “raising a weapon,” even if standing accounts for the majority of the time. Similarly, in a system which recognizes kneeling, sitting, and running, mistaking an instance of kneeling for sitting may be much less severe than mistaking running for sitting.

III. LEVEL OF ANALYSIS

Each level of analysis represents the activity labels in a way that highlights different evaluation issues. In this section, we detail these differences and discuss how each representation facilitates detecting certain kinds of errors. See Figure 2 for a diagrammatic representation of the event, frame, and segment-based analysis.

A. Event Analysis

Event analysis takes the individual activity occurrence as the basic unit for comparison. Events are temporally distinct, contiguous blocks of time with a specific start time, end time, and label. Event analysis methods can be divided into two categories, those that consider the actual occurrence time

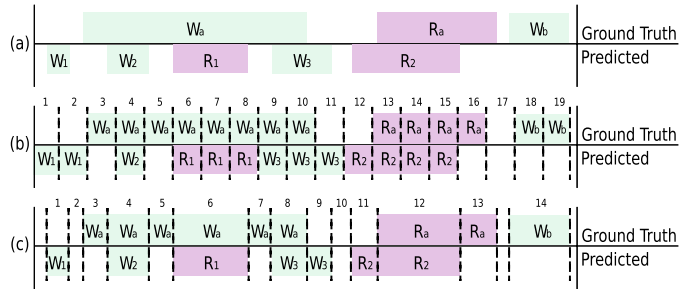


Fig. 3. Ground truth and predicted activity results demonstrating different types of errors for (a) event, (b) frame, and (c) segment analysis. The sample domain includes two known activities, walking (W) and running (R), as well as the null class, which is left unlabeled.

and those that only consider their order, but all carry the distinguishing characteristic of treating each event as an atomic structure. In the following sections, we discuss three different approaches for scoring events. The first only considers the order of the events when detecting errors, while the other two adopt different criteria for temporal matching.

1) *Levenshtein distance*: The Levenshtein distance (also known as the edit distance) computes the minimum number of modifications (insertions, deletions, and substitutions) needed to transform one string into another. This method can be used for activity recognition evaluation by treating each activity as a symbol, ignoring null labels, and representing the ground truth and predicted label sequences as two strings. Errors are detected by aligning the strings via dynamic programming, which will efficiently find the set of symbol correspondences that minimizes the number of errors. Use of this method arose in speech recognition where researchers were interested in measuring sentence-level accuracy rather than ensuring that each word or phoneme was detected at exactly the right time. Also, note that this method assumes that all of the activities are bounded. When fluent activities are involved, the detected errors may be less meaningful and thus mislead an analyst.

As an example of applying the Levenshtein distance to an evaluation problem, consider Figure 3. This diagram shows hypothetical ground truth and predicted events for a system that can detect walking, running, or null. Since the edit distance does not consider the temporal location of each event, it will match $W_1 \leftrightarrow W_A$, $R_1 \leftrightarrow R_A$, and $W_3 \leftrightarrow W_B$. W_2 and R_2 are then marked as insertion errors.

2) *Matching via temporal correspondence*: When it is important to consider the actual time during which an event was detected, a temporal correspondence method can be used. This method seeks to match each ground truth event with a predicted event based on temporal overlap. Many different match criteria are possible (see Figure 4), including:

- *midpoint overlap* A predicted event must span the midpoint of its matching ground truth event. This approach is often used to score word spotting systems in the speech recognition domain.
- *majority overlap* A predicted event is paired with a ground truth event if the overlap accounts for a majority

GT		Midpoint	Majority	Maximum
A		✓	✓	✓
B		✓	✗	✓
C		✗	✗	✓

Fig. 4. Illustration of three methods for temporal correspondence: midpoint, majority, and maximum overlap. The vertical, dashed line represents the midpoint of the ground truth label.

of the time in both events.

- *maximum overlap* Predicted and ground truth events are paired based on maximizing overlap. Although computing the optimal correspondence is NP-hard, greedy approaches work well in practice.

3) *Majority vote compared to ground truth*: When the results of an activity recognition system are used in an interactive tool, the severity of false positives may be greatly diminished. Because the system is interactive, the human user can easily detect and ignore false positives. In these cases, an asymmetric evaluation method may provide a better indication of real world performance. One such method evaluates predicted activities by allowing them to vote on the label of each ground truth event. Whichever predicted activity accounts for the majority of the time of each true event is taken as the overall label.

B. Frame Analysis

Frame-based methods take fixed-duration intervals as the basic, atomic unit. The specific duration of the frame can be determined based on the time-scale of the domain. For example, one second intervals may be appropriate for fine-grained human activities, while hour long frames may be more reasonable for predicting the highway traffic patterns, and annual frames are sufficient for many population statistics.

Since the duration and start time of each frame is independent of both the ground truth and predicted labels, they are always aligned, making it is easy to compare the labels. With frame-based analysis, all discrepancies between activity classes are substitution errors, predicting a known activity where the ground truth is null is an insertion, and erroneously predicting null is a deletion error.

For example, Figure 3b shows hypothetical ground truth and predicted frame labels for a simple domain. The predicted labels include four insertion errors (frames 1, 2, 11, and 12), three substitution errors (frames 6 – 8), and five deletion errors (frames 3, 5, 16, 18, and 19).

C. Segment Analysis

A hybrid approach combining aspects of frame and event-level analysis uses segments as the basic unit for comparison. A segment is an interval of maximal duration in which both the ground truth and the predicted activities are constant. Thus, each segment may have a different duration, but there are no aliasing problems or ambiguities associated with event correspondences and boundary alignment (see Figure 3c).

In addition to avoiding event correspondence issues, one of the major benefits of a segment-based representation is that it simplifies incorporating contextual information for detecting different kinds of errors. Thus, we include underfill, overfill, merge, fragmentation, substitution-fragmentation, and substitution-merge as possible segment analysis labels. Note that while these error types exist at all levels of analysis, they are most easily detected using segments.

IV. CLASSIFICATION ERROR TYPES

Once a level of analysis has been selected, the ground truth and predicted activity labels can be compared in order to determine which errors were made. Before detailing common evaluation metrics, we will introduce the standard classification error types that are used to calculate the metrics and build the performance visualizations.

In isolated, binary classification problems, the two classes are often called the positive and negative class (as when a diagnostic test reports positive or negative for the presence of the disease or condition it detects) leading to the familiar four possible results: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

The definition of these four types of errors can be generalized to the multi-class situation by treating the null class as negative and all other classes as unique positive classes. Thus, a correct identification of any known class is a true positive, correctly detecting null is a true negative, failing to detect a known class is a false negative, and substituting a known class for null is a false positive.

In continuous recognition domains, researchers typically tally a different set of classification results that take into account the fact that there may not be a one-to-one correspondence of ground truth to predicted events (see Figure 5):

- 1) *Correct (C)* - sometimes called “Hits” (H); represents correct classification
- 2) *Substitutions (S)* - represent correct temporal detection but incorrect activity identification,
- 3) *Insertions (I)* - detection of an activity when none actually occurred; this can also occur when a long activity is partially detected multiple times (see Figure 3a, specifically W_1 and then R_1 and R_3 represent both kinds of insertion errors).
- 4) *Deletions (D)* - failure to detect an activity (*e.g.*, W_B in Figure 3a).
- 5) *Total Number of True Events (N)* - a useful variable for calculating statistics, though not strictly a classification result: $N = (C + D + S)$.

Finally, we can also tally the additional error types mentioned in Section III-C to provide a more nuanced summary of performance:

- 1) *Underfill (U)* - when an activity is correctly identified, underfill errors account for the time at the beginning and end of the activity that is not detected,
- 2) *Overfill (O)* - when an activity is correctly detected, overfill errors account for the time before and after

the standard measure for the percentage of correctly labeled frames. All divisions below true negatives represent errors which increase in severity farther down the column. The next two segments, overfill and underfill, represent difficulties in determining the boundaries of events. For systems dedicated to event-based recognition, these errors may be deemed inconsequential. Fragmentation and merge, on the other hand, may be considered more serious errors. These segments in the visualization provide a sense of the difficulty the recognizer has in detecting a whole event, uninterrupted by the null class. On the other hand, for applications that focus on spotting relatively sparse activities, these errors may also be considered acceptable. Errors represented by divisions below merge, including insertion, deletion, and all substitution errors, are considered serious errors since they would be classified as errors in both frame-based and event-based systems.

The visualization method allows the evaluator to compare the attributes of activity recognizers in the context of his application. For example, examining Figure 6 shows that recognizers A and B have about the same total number of errors from a frame-based perspective. However, recognizer B is superior if the intent is to make an event-based recognizer. Recognizer A has most of its errors concentrated under the serious error line (substitutions, insertions, and deletions), recognizer B, however, suffers most from overfill and underfill errors. Overfills and underfills of minor duration can be ignored. For example, most sign language recognition researchers would not bother to report overfill and underfill errors, focusing instead on accuracy rates at the event level.

For chronic care medical monitoring systems, where the goal is to capture relatively rare events for later study, overfills and underfills can not be ignored, but they may be tolerable. For example, a patient may wear a heart monitor for a month. His physician could then use a recognition system to reduce

the month of data to just the specific, rare events he wishes to examine, such as heart palpitations. Even though the system may not accurately determine the temporal boundaries of the palpitations, the physician can manually scan past the “fuzzy” transition boundaries to analyze the real events.

In other domains, the boundaries between events are important. For example, the ASSIST system can record hours of video for each soldier during a long patrol. Imagine a recognition system designed to help a commanding officer search the database for footage of “soldiers running” to splice into an intelligence debriefing video. If the footage starts with something other than running (an overfill) the viewer may be misled as to the topic. However, some fragmentation may be tolerable (e.g., the system might not return the full segment because of misclassifying some of the footage in the middle as null activity). Similarly, merges may be inconsequential for this application example (e.g., the system ignoring some null activity in the footage caused by a car passing in the foreground). Thus, in the figure above the designer of the video editing system may choose recognizers C and D over recognizer B and would certainly would prefer all three to recognizer A.

In addition to providing a means to evaluate and compare recognition systems, error division diagrams can also provide information useful for tuning a recognizer. For example, a relatively large number of overfills and merges may mean that a threshold is set too low, which leads to erroneous inclusion of nearby null frames into a real event. A system designer might also suspect that the boundaries on his ground truth labeling are too permissive. On the other hand, a relatively large number of underfills and fragmentations may mean that a threshold is too high or that a ground truth labeling is not permissive enough. Of course, the system designer must be careful that changing thresholds or labeling policies does not exacerbate the more serious substitution, deletion, and insertion errors.

If a system designer is interesting in building a system based on events, the visualization method still provides a valuable tool. Each graph shows the relative number of substitutions, deletions, and insertions, which can be a critical factor in comparing event based recognizers. For example, recognizers E-H in Figure 6 all have the same percentage of serious errors. Suppose we want to make a recognizer for just the most critical four activities of the fourteen we defined for the DARPA ASSIST program (*i.e.*, running, crawling, lying down, and weapon up). Soldiers we interviewed stated that if any of these four critical activities happened, it generally meant that the soldier was in trouble implying that the other soldiers should be alerted. We place the remaining ten activities into the null class (*i.e.*, the recognizer’s output of walk, sit, stand,*etc.* is mapped to null) and then trigger an alert whenever one of the four critical activities is detected. In such a case, recognizer E is superior to recognizer F. Recognizer F’s high deletion rate indicates that it would not trigger much more often than recognizer E, even if recognizer E would sometimes trigger because it thought the soldier was lying down even though



Fig. 6. Error division diagrams for eight recognizers showing, from top to bottom, the percentage of frames that were true positives, true negatives, overfills, underfills, fragmentations, merges, insertions, deletions, substitution-fragmentations, substitution-merges, and substitutions. The dark horizontal bar in each column indicates the division between severe errors and those that may be tolerable.

the soldier was crawling. Everything else being equal, one can imagine that a recognizer with a comparably high insertion rate could also be inferior to recognizer E. False alarms would be much more annoying to a soldier's friends than being alerted to the soldier lying down when he was actually crawling.

Error diagrams can also be used to help refine a recognition system. For example, in Figure 6 G and H, the dominance of the fragmentation and merge variants over pure substitutions may be an indication that some tuning may improve results. For example, in the ASSIST system previously described, "crawling" often involves "lying down" first. Perhaps the ground truth labeling ignored this fact, or perhaps the system is inserting crawling into the middle of a segment of lying down. The system might even be thrashing between the two classes because of the difficulty in distinguishing between them. Since the activities are similar and both result in the same alert action, the designer may want to make the classes equivalent or, at least, revisit his ground truth labeling. On the other hand, if the designer discovers that fundamentally unrelated classes are being substituted (for example, "running" often appears within segments of "lying down") he may inspect his models and labeling more carefully for errors.

A. Discussion

All of the above metrics can be used to summarize multi-class recognition performance. For metrics such as precision, recall, specificity, NPV, F-Measures, and likelihood ratio, each class can be evaluated as a binary class problem considering all other classes to be the "other" class. The results can be averaged across classes, or weighted by the amount of time each class represents in the data set. The first emphasizes the recognition of the activities, and the second emphasizes describing the time in the data set as accurately as possible. The accuracy expands on the idea of two classes and includes substitutions, insertions, and deletions. Error division diagrams provide yet more information for the designer's consideration. The visualization exposes the types of errors inherent to the accuracy metric which allows the system designer to choose a system based on the needs of the application. The designer has some information as to if false positives and false negatives are primarily due to confusion with the null class or with other classes. In addition, the error division diagrams simultaneously provide a sense of event, frame, and segment-based errors and provide an intuition as to how sparse activities are in the data space. Through the tracking of fragmentations and merges, error division diagrams also provide some intuition as to the temporal coherence of classes (if the recognizer is thrashing between two similar classes). Of course, the error division diagrams are not as compact as one or two number metrics, such as precision, recall, and accuracy rates.

VII. FUTURE WORK

We are currently investigating several ways to improve the evaluation methods described here. For example, when scoring the different error types, we currently give all errors the same weight regardless of which activities are involved. Instead, the

scores could incorporate the prior likelihood of the activity and the cost of making a particular mistake. For instance, if confusing running for walking is serious, it could be given a cost of 2.0, while confusing standing for walking only carries a weight of 0.4.

Overfill and underfill account for the often inconsequential mistake of correctly identifying an activity but failing to precisely detect the temporal boundaries. An equivalent issue exists for substitution errors, but this case is not handled by the current method. That is, if running is mistaken as walking, any boundary inconsistencies also count as insertion errors. These errors lead to asymmetry in the overall error calculation and may underestimate the performance of the system, especially if the substitution has low cost. To remedy this, we are exploring the use of "negative" overfill and underfill.

Although recognizing overfill and underfill as relatively minor errors is important, the current method does not consider the amount of over or underfill. Our goal is to only ignore small boundary errors, whereas the current approach allows arbitrarily large over or underfill segments.

VIII. CONCLUSION

We have described some of the factors that complicate the evaluation of a continuous activity recognition system, and we discussed metrics that are commonly used to describe system performance. In addition, we have examined the uses of a recent visualization tool that allows system designers to diagnose the performance of a recognizer and compare performance across recognizers with respect to the needs of event-based or frame-based applications. Continuous activity recognition applications span a wide range of domains, and the recognizers are currently difficult to characterize. We must strive to define tools that will allow the community to compare and communicate results as increasingly more researchers from diverse backgrounds join the field.

REFERENCES

- [1] J. Ward, P. Lukowicz, and G. Tröster, "Evaluating performance in continuous context recognition using event-driven error characterisation," in *Proceedings of LoCA 2006*, May 2006, pp. 239–255.
- [2] M. U. of South Carolina, "Diagnostic tests glossary," Web Article, June 2006. [Online]. Available: <http://www.musc.edu/dc/ice>
- [3] H. Müller, W. Müller, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," University of Geneva, Switzerland, Tech. Rep., 1999.
- [4] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.
- [5] J. Ward, P. Lukowicz, G. Tröster, and T. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *Pattern Analysis and Machine Intelligence (in press)*, 2006.

An Improved Stereo-based Visual Odometry System

Zhiwei Zhu, Taragay Oskiper, Oleg Naroditsky, Supun Samarasekera, Harpreet S. Sawhney and Rakesh Kumar
Sarnoff Corporation

201 Washington Road, Princeton, NJ 08540, USA

{zzhu, toskeeper, onaroditsky, ssamarasekera, hsawhney, rkumar}@sarnoff.com

Abstract—Conventional non-vision based navigation systems relying on purely GPS or inertial sensors can provide the 3D position or orientation of the user. However GPS is often not available in urban, forested regions and cannot be used indoors. An IMU sensor on its own will drift at the rate of T^3 . Vision-based navigation systems (Visual Odometry) provide an independent method to estimate position and orientation of the user/ system based on images captured by the moving user accurately. Vision based systems also provide information (e.g. images, 3D location of landmarks, detection of scene objects) about the scene that the user is looking at.

In this paper, we propose a set of techniques to improve the robustness and accuracy of previously developed techniques for stereo-based Visual Odometry. First, integrated with stereo constraints, a dynamic local landmark tracking technique is proposed to improve the feature matching. Second, during pose computation, a dynamic 3D reference selection technique is used to minimize the 3D reconstruction error and an outlier removal technique is used to reject outliers. Using these techniques, the error associated with each pose computation is minimized and the drift is reduced significantly. Finally, in order to further improve robustness, both IMU and GPS sensors are integrated with the Visual Odometry in an extended Kalman filtering framework. Compared to the original system developed in [1], which is sensitive to the outliers and does not work well over long distances, the improved system is significantly more accurate and robust over long-distance navigation both indoors and outdoors. The performance of the improved system is demonstrated through (ground truthed) real navigation tasks and we show that we are able to locate the user within 2 meters both indoors and outdoors for tracks ranging from 100 to 500 meters long.

I. INTRODUCTION

Precise navigation systems are very important for many applications in personal location and route planning assistance [2], autonomous robot navigation [3], unknown environment map building [4], etc. However, most of the available navigation systems do not function very well and fail frequently under certain circumstances. For example, GPS (Global Positioning System) is a widely used navigation system. However, it cannot work reliably once the satellite signals are blocked or unavailable in “GPS-denied” environments such as indoors, forests, urban areas, etc. Even when it works well, the GPS can only provide the location of the user, which is usually not sufficient to assist the user during navigation. For example, when a group of warfighters are performing a military task in an unknown environment, besides knowing the location of each warfighter, sharing what each warfighter is seeing, knowing where they are looking and what is happening in the scene is also important for them to cooperate with each other.

A vision-based navigation system has all these benefits. Specifically, it does not require any overly expensive equipment and it can independently estimate the position and 3D orientation accurately by using only image streams captured from one, two or more inexpensive video cameras. More-over it can be integrated with a GPS and IMU system to robustly recover both the location and the 3D gaze or orientation of the user under a wide range of environments and situations. Most importantly, the detailed information and imagery about the environment is recorded in real time. The imagery can be shared and analyzed to assist the user in reporting what is seen. Figure 1 shows a snapshot of our vision-based navigation GUI. As a person equipped with our navigation system travels, his position and viewpoint can be located precisely in the map as shown in the upper window of Figure 1 in real time. In addition, what is seen from that viewpoint and location is shown as a captured image in the lower-left window. At the same time, a smart image processing system analyzes the images, and detects objects of interest such as vehicles and people. Their locations are recorded in the database and can be seen in the map. Since all the videos and estimated pose information are stored in real time, they can be played and processed off-line to assist in future navigation tasks, mission reporting and mission rehearsal.

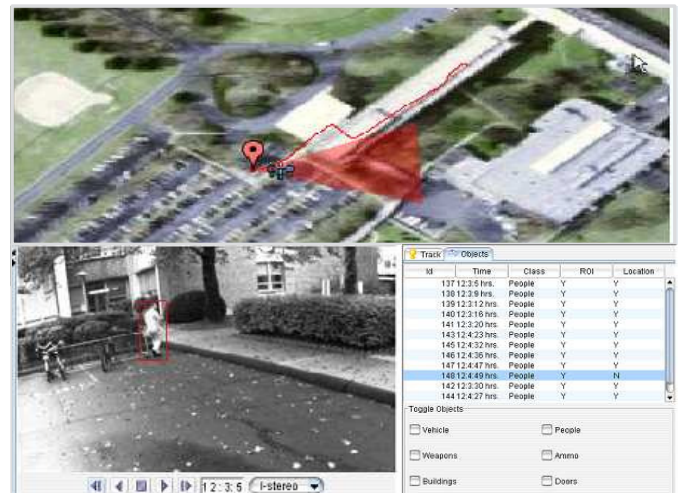


Fig. 1. A snapshot of our vision-based navigation GUI

II. RELATED WORK

A variety of efforts [5], [6], [7], [8], [1], [4], [3], [9], [10] have been made to build a navigation system using vision

approaches in the past few decades. In most approaches using computer vision techniques, a set of stationary feature points in the scene are tracked over a sequence of images. The position and orientation change of the camera is determined using the image locations of the tracked feature points. The motion estimation can be done using a monocular, binocular (stereo) and multi-camera configurations. In the stereo or multi-camera configurations, the 3D location of scene points can be estimated by the binocular disparity of the feature points. The estimated 3D point locations may then be used to solve by a 3D-2D motion estimation to track the motion of the camera. In a monocular configuration, both the relative motion of the camera and the 3D locations are estimated simultaneously. Because of this reason, visual odometry systems based on stereo [7], [8], [1], [4], [3], [9], [10] are more stable than monocular-based visual odometry systems [5], [6].

A visual odometry system can often drift over time due to many reasons, which include errors associated with stereo calibration or image quantization, poor-quality images, inaccurate feature positions, outliers, and finally the instability of motion estimation using noisy correspondences. In literature, most of the stereo-based visual odometry systems [7], [8], [1], [4], [3], [9], [10] try to compute the pose between each pair of image frames separately, which is referred to as the frame-by-frame approach. In [3], multi-frame tracking is performed to track a set of same features across a sequence of images, and pose is computed from the feature tracks. Compared to the traditional frame-by-frame approach, experiments show 27.7% reduction in navigation error when multi-frame tracking is performed. However, an issue for the multi-frame tracking approach is that it lacks a metric to stop tracking when the tracked feature points become insufficient for pose estimation in terms of either quantity or spatial distribution. In this paper, we proposed a dynamic local landmark tracking technique to automatically select an optimal set of tracked feature points across image frames for pose estimation.

During pose computation, most of the stereo-based visual odometry systems estimate the pose from the established 2D-3D feature correspondences [8], [1], [4], [3]. Since 3D coordinate of each feature point is reconstructed using the stereo based triangulation, it is essential that the error introduced during 3D reconstruction is minimized [3]. However, in [1], during stereo matching, no stereo geometric constraints are utilized to reduce the search region and a large amount of false stereo matches may be obtained. Therefore, in this paper, epipolar-geometry and disparity constraints are applied to reduce the search region for matching so that the 3D reconstruction error caused by false stereo matches can be minimized. In order to further reduce the 3D reconstruction error caused by depth, a dynamic 3D reference selection technique is also proposed to dynamically select the frame that produces less 3D reconstruction error as the reference frame during pose computation.

In reality, the scene always contains moving objects such as walking persons, moving vehicles, waving trees, etc. If the features in the moving objects are selected during pose

estimation, they can affect the accuracy of the resulting pose significantly unless they are detected and discarded as outliers. Therefore, in this paper, an outlier removal procedure is proposed to remove these outliers before performing pose estimation.

Using the above proposed improvements, our visual odometry alone is able to provide highly accurate pose of the camera. However, drift from the true trajectory due to accumulation of errors over time is inevitable in any relative measurement system. In addition, there are situations where the accuracy of our improved visual odometry system alone may degrade due to lack of features in the scene. For instance, the field-of-view of the cameras may be occupied by a textureless surface where feature detection is largely inhibited. Therefore, in order to further reduce the drift of visual odometry system, integration with other sensors such as GPS, Inertial Measurement Units (IMU) [11] or orientation sensors [12] are essential. Hence, in this paper, an Extended Kalman Filter (EKF) framework is proposed to integrate both IMU and GPS measurements with the visual odometry measurements and the robustness of the complete system is further improved.

III. SYSTEM OVERVIEW

Our proposed visual odometry system consists of two calibrated cameras that form a stereo system. The motion of the system is determined from the pairs of stereo images captured by the stereo cameras. Specifically, Figure 2 illustrates the flowchart of our proposed visual odometry system, and it loops in the following steps:

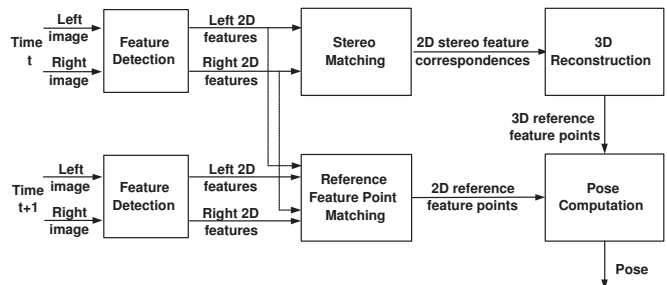


Fig. 2. The flowchart of the proposed visual odometry system

- **Step 1: Feature detection.** Natural feature points extracted from the images are utilized in our system to estimate the pose. The feature point-based pose estimation technique is robust to scale, viewpoint, illumination changes and partial occlusions. Given a pair of stereo images, our system starts with detecting a set of potential feature points for pose estimation. The Harris corner detector [13] is utilized to automatically extract a set of feature points from the left and right images respectively. Harris corners [13] are relatively stable under small to moderate image distortions.
- **Step 2: Stereo matching.** A stereo matching algorithm is used to find correspondences between the extracted feature points between left and right images. The 3D

coordinates of each feature point are obtained by triangulation using the feature correspondences. These obtained 3D feature points serve as the reference points for pose computation when a new pair of stereo images arrives.

- **Step 3: Feature tracking.** Once a new pair of stereo images is captured, the set of reference feature points is tracked in the left image and right image individually. Specifically, the tracking is done with a very similar technique as in stereo matching: feature detection followed by matching. For example, in order to track the reference feature points in the left image of a new stereo, Harris corners are first extracted from the left image to provide a set of potential feature points for matching. Then, each reference feature point is tracked by matching the extracted feature points between the left images of the stereo pairs.
- **Step 4: Pose computation.** After the 2D reference feature points in the new stereo pair are detected, a set of 2D/3D feature correspondences can be established and sent into the pose computation component to estimate the 3D pose of the new frame. Finally, the relative pose information between these two pairs of stereo images can be estimated.
- **Step 5: Updating the reference features.** When the system moves by a large motion step, most of the reference feature points will move out of the field-of-view of the camera quickly. Therefore, it is essential to update the reference feature points frequently so that there are always enough reference feature points for pose computation. As a result, after the pose is computed, new reference feature points that appear in the new stereo pair are detected by repeating the stereo matching in Step 2. Finally, the updated reference feature points are utilized to estimate the pose in the next pair of stereo images.

The whole process repeats to efficiently recover the full 3D track.

IV. FEATURE DETECTION AND TRACKING

Our system starts with feature detection and tracking. Precise and robust feature detection and tracking in the images is the prerequisite for the subsequent pose estimation accuracy. As a result, in order to maintain accurate navigation over long distances, the error introduced during the feature detection and matching must be minimized first.

A. Feature Detection

Extracting a set of reliable features for each pose computation is extremely challenging due to significant motion, and changes in the scene background, changes in camera viewpoint, changes in illumination and occlusion. In addition, as the camera moves, some features will move out of the field-of-view of the camera and new features will appear. The new and surviving features must be detected and tracked in subsequent frames.

In order to better handle the appearance and disappearance of the features, feature detection is performed in each image

frame. Since Harris corners [13] are relatively stable under small to moderate image distortions, Harris corner extraction is performed in every frame and the extracted Harris corners will serve as the feature points for the pose computation. Detailed implementation can be found in [1].

Once a set of feature points are extracted from each frame, they are matched to subsequent frames to find their correspondences. The matching procedure described in [1] is as follows: for each feature point x_i in the first frame, search a region in the second image around the location x_i for its correspondence x'_i . The search is based on the similarity of the local image patches centered around the points. In our implementation, normalized correlation over an 11×11 -pixel window is used to measure the similarity. Finally, a feature that produces the highest similarity score is considered as a match.

The matching technique described above is very easy to implement. However, it does not consider any geometric constraint or motion constraints. Therefore, a large percentage of false matches is obtained. In fact, our experiments show that there are around 40% false matches in the obtained stereo matches and around 30% false matches in the obtained next video frame matches.

In order to reduce the false matches, a set of techniques are proposed to improve the feature matching.

B. Constrained Stereo Feature Matching

Given a pair of stereo images, the task of stereo feature matching is to match the extracted feature points between the left and right images. In practice, the viewpoints of the left and right cameras in a stereo system are different and the image appearances of a same feature point may be significantly different in the left and right images. On the other hand, since all the feature points are corners, it is also very common for them to look similar to each other. As a result, false stereo matches may occur easily when using a large search region.

Fortunately, there are two geometric constraints can be applied to reduce the search space during the stereo feature matching: epipolar-geometry constraint and disparity constraint [14]. With the use of these two constraints, most of the geometrically infeasible feature matches are eliminated so that the number of false stereo matches is reduced dramatically.

Once a set of stereo feature matches are obtained from each stereo pair, the 3D coordinates of each feature can be reconstructed by triangulation [14]. Given an initial stereo pair, these features with 3D coordinates will serve as the reference points during pose computation. But first, they need to be tracked in the subsequent image frames.

C. Dynamic Local Landmark Tracking

In fact, the reference feature points are tracked in the subsequent left images and right images individually. Similar to the stereo matching, the feature tracking is performed with the Harris corner detection and matching technique. However, different from the stereo matching, the feature matching is performed between the left images or the right images of two different stereo pairs. Since the relative pose between these two

stereo pairs is not known, there are no geometric constraints that can be used during matching. As a result, falsely tracked features may occur frequently.

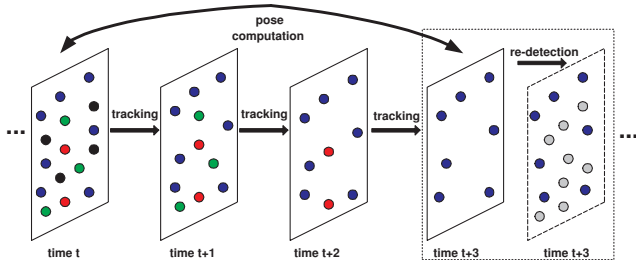


Fig. 3. The illustration of the dynamic local landmark tracking technique

In order to discard those falsely tracked reference feature points, a dynamic local landmark tracking technique as shown in Figure 3 is proposed:

- **Step 1: 3D reference frame initialization.** Given an initial stereo pair, the proposed stereo matching technique is activated to extract a set of 3D reference feature points. Once the set of 3D reference feature points are initialized, this stereo pair will be referred to as a 3D reference frame, and the reference feature points will be referred as local landmarks and tracked in the subsequent stereo images.
- **Step 2: 2D local landmark tracking.** Given a new stereo pair, the set of local landmarks are tracked. As the camera moves, some of the local landmarks will move out of the field-of-view of the cameras. However, besides the local landmarks that move out of the field-of-view of the camera, another set of local landmarks also lose tracking due to their non-distinctive intensity distributions. Local Landmarks with more distinctive intensity distributions survive longer during tracking. Hence, via tracking, the unstable local landmarks, which usually happen to be the false matches, are reduced significantly.
- **Step 3: Spatial distribution checking and pose computation.** The aim of local landmark tracking is to obtain a set of reliably tracked feature points that contains few false matches. Although the number of falsely tracked local landmarks will decrease as the tracking continues, more and more stable local landmarks will also move out of the field-of-view of the camera as the system moves. Since the spatial distribution of the tracked local landmarks is essential for accurate pose estimation, a metric is needed to measure the spatial distribution of the tracked local landmarks. Hence, when the set of tracked local landmarks become badly distributed, a set of new local landmarks is initialized.

A simple but effective metric as shown in Figure 4 is developed to measure the spatial distribution of the tracked local landmarks. Specifically, the left image of the 3D reference pair is divided into 10×10 grids. The total number of grids that contain local landmarks is used as a metric score.

The spatial distribution metric is computed for each

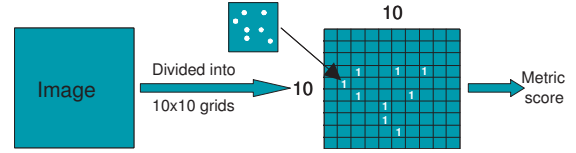


Fig. 4. The simple spatial distribution metric

frame. If the spatial distribution metric is greater than a predefined threshold, which is set to 50 empirically, the pose is computed from the 3D-2D correspondences of tracked local landmarks between the current stereo pair and the 3D reference frame.

- **Step 4: Updating 3D reference frame.** A new reference image is established, if the spatial distribution metric for matches between subsequent frames reaches less than a threshold. The stereo matching technique described above is re-activated to extract a set of new 3D reference feature points from the current pair of stereo images. At the same time, the current stereo pair will be updated as the 3D reference frame, and the new set of extracted reference feature points will be updated as local landmarks. These newly updated local landmarks are tracked in the subsequent stereo image frames, and the whole dynamical local landmark tracking technique repeats to recover the entire traveled 3D path.

With the use of dynamic local landmark tracking, a set of stable feature points that contains fewer false matches is obtained. Subsequently, a more accurate pose is computed from the tracked points.

V. ROBUST POSE ESTIMATION

Given the set of 2D-3D correspondences (x_i, X_i) of n reference features, the pose estimation algorithm is to find the rotation matrix R and the translation vector T that maps the world coordinate system to the camera coordinate system. In general, it can be done simply by minimizing the projection residuals in the image plane as follows:

$$\min \sum_i^n r_i^2 \quad (1)$$

where $r_i = \|x_i - x_i'\|$ and $x_i' = Proj(RX_i + T)$ ($Proj$ denoting the projection of a 3D point in the image plane).

Since the 3D coordinates in a 2D-3D feature correspondence are computed through the 3D reconstruction from the reference stereo pair, each feature will have a 3D reconstruction error. Hence, it is essential that the 3D position of each feature is reconstructed as accurate as possible so that the effect of the 3D reconstruction error on the pose estimation can be minimized. In this paper, a dynamic 3D reference selection technique is proposed to select the stereo pair that produces the least 3D reconstruction error caused by depth as the reference during pose computation.

In addition, since the above pose estimation technique considers all the features during minimization, a single outlier

can have a large effect on the estimated pose. In practice, the scene is not completely stationary so that it is common that some features will be located in the moving objects. Together with the falsely matched features, they can affect the resulting pose significantly. Therefore, outlier removal technique must be integrated to reduce the effect of outliers during pose estimation.

A. Dynamic 3D Reference Selection

During 3D reconstruction, the uncertainty of the reconstructed 3D coordinates of a feature point varies with the depth dramatically [14]. Usually, the points with large depth will have larger uncertainties than the points with small depth. Hence, when a point moves away from the camera, its associated 3D reconstruction error will become larger and larger.

With the proposed dynamic local landmark tracking, the 2D-3D feature correspondences are usually established between two stereo pairs that are more than 2 frames apart. Depending on the moving speed and the motion type of the system, for example, during the forward running-truck case, the distance of the camera between the selected stereo pairs can be several meters away in depth. Therefore, when a same feature point is reconstructed in both stereo pairs, its 3D reconstructed error in each stereo pair can be considerably different. Normally, the stereo pair of reference during pose computation is selected temporally, which means that the old stereo pair is always selected as the reference pair. Therefore, for the above forward running-truck case, the 3D reconstruction error is larger than the error when the new stereo pair is selected as reference for 3D reconstruction.

As a result, during the pose computation, the reference stereo pair needs to be selected dynamically in order to reduce the 3D reconstruction error introduced by depth. A simple technique is to dynamically select the reference stereo pair via the motion type of the system. Specifically, for the forward motion, the new stereo pair is selected as the reference frame; while for the backward motion, the old stereo pair is selected as the reference frame. Via the proposed technique, the stereo pair of reference for the pose computation can be selected dynamically so that the 3D reconstruction error can be reduced. As a result, a more accurate pose is computed.

B. Outlier Removal

In order to detect and reject the outliers in the set of 2D-3D feature correspondences, a Least Median of Squares (LMedS) approach is proposed. Specifically, the first step is to estimate a set of initial pose parameters as follows:

- From n 2D-3D feature correspondences, generate a set of hypotheses $H_i (i = 1, \dots, N)$ by drawing N random sub-samples of 3 different feature correspondences.
- For each hypotheses H_i consisting of 3 different feature correspondences, a set of pose parameters $P_i = (R_i, T_i)$ can be computed easily.
- For each set of estimated pose parameters P_i , the median of the squared residuals, denoted by Mr_i , is obtained

from the residuals $r_k (k = 1, \dots, n)$ with respect to the whole set of feature correspondences.

- From the N sets of estimated pose parameters $P_i (i = 1 : N)$, the one P_{min} that produces the minimal median Mr_{min} of the squared residuals is chosen as the final set of pose parameters.

The outliers are usually those that produce larger residuals given the correct pose parameters. With the proposed LMedS technique, the errors of the largest ranked half of feature correspondences are ignored in order to reduce the effects of the feature outliers during pose estimation. However, if the set of feature correspondences contains more than 50% outliers, it will fail to estimate the pose parameters correctly. Fortunately, with the use of our proposed feature detection and matching techniques, the outliers due to false feature matches in the obtained feature correspondences are usually less than 50%. As a result, unless the image is filled with moving objects, the overall outliers will not exceed 50%.

Once the set of initial pose parameters P_{min} is obtained, the standard deviation $\hat{\sigma}$ is computed from the obtained minimal median Mr_{min} as follows:

$$\hat{\sigma} = 1.4826[1 + 3/n]\sqrt{Mr_{min}} \quad (2)$$

where 1.4826 is a coefficient to achieve the same efficiency as a least-squares in the presence of only Gaussian noise, and $3/n$ is to compensate the effect of a small data set. Subsequently, the outlier rejection is done with the use of the standard deviation $\hat{\sigma}$. Specifically, the feature points whose residuals are more than $2\hat{\sigma}$ under the estimated pose parameter set P_{min} are rejected as outliers. With the proposed technique, most of outliers can be discarded successfully.

Finally, on the set of remaining 2D-3D feature correspondences, a pose refinement procedure with the use of nonlinear optimization technique starting from the estimated pose parameter set P_{min} is applied so that pose can be estimated accurately.

VI. VISUAL ODOMETRY, IMU AND GPS INTEGRATION

In order to increase the accuracy and robustness of the system and to reduce the long term drift as much as possible, we integrated our visual odometry system with a MEMS (Microelectromechanical Systems) based IMU and a GPS unit using the extended Kalman filter (EKF) framework. Similar to [5], we chose a ‘‘constant velocity, constant angular velocity’’ model for the filter dynamics. The state vector consists of 13 elements: \mathbf{T} , 3-vector representing position in navigation coordinates, \mathbf{q} , unit quaternion (4-vector) for attitude representation in navigation coordinates, \mathbf{v} , 3-vector for translational velocity in body coordinates, and $\boldsymbol{\omega}$, 3-vector for rotational velocity in body coordinates. Quaternion representation for attitude has several practical properties. Each component of the rotation matrix in quaternion is algebraic, eliminating the need for transcendental functions. It is also free of the singularities that are present with other representations and the prediction

equations are treated linearly. Based on this, the process model we adopted is given by

$$\begin{aligned} \mathbf{T}_k &= \mathbf{T}_{k-1} + \mathbf{R}^T(\mathbf{q}_{k-1})\mathbf{T}_{rel} \\ \mathbf{q}_k &= \mathbf{q}_{k-1} \otimes \mathbf{q}(\boldsymbol{\rho}_{rel}) \\ \boldsymbol{\omega}_k &= \boldsymbol{\omega}_{k-1} + \mathbf{n}_{w,k-1}\Delta t \\ \mathbf{v}_k &= \mathbf{v}_{k-1} + \mathbf{n}_{v,k-1}\Delta t \end{aligned} \quad (3)$$

where

$$\begin{aligned} \mathbf{T}_{rel} &= \mathbf{v}_{k-1}\Delta t + \mathbf{n}_{v,k-1}\frac{\Delta t^2}{2} \\ \boldsymbol{\rho}_{rel} &= \boldsymbol{\omega}_{k-1}\Delta t + \mathbf{n}_{w,k-1}\frac{\Delta t^2}{2} \end{aligned} \quad (4)$$

and \otimes is used to denote the quaternion product operation. Above, $\boldsymbol{\rho}$ is the rotation vector (representing the rotation axis) in the body frame, $\mathbf{R}(\mathbf{q})$ is the rotation matrix determined by the attitude quaternion \mathbf{q} in the navigation frame, and $\mathbf{q}(\boldsymbol{\rho})$ is the quaternion obtained from the rotation vector $\boldsymbol{\rho}$. Undetermined accelerations in both translational and angular velocity components are modeled by zero mean white Gaussian noise processes $\{\mathbf{n}_{v,k}\}$ and $\{\mathbf{n}_{w,k}\}$.

We observed that throughout most of the data sequences where the images contain lots of good features that are accurately tracked, there is a very high degree of agreement between the angular velocities computed by visual odometry alone and those available from the angular rate sensors (gyros) in the IMU. Therefore, in our system we chose to use the raw gyro readings from the IMU directly as measurements in the Kalman filter. The observations from visual odometry, IMU and GPS are used according to the following measurement model:

$$\begin{aligned} \mathbf{T}_k^{gps} &= \mathbf{T}_k + \mathbf{n}_k^{gps} \\ \mathbf{v}_k^{vo} &= \mathbf{v}_k + \mathbf{n}_{v,k}^{vo} \\ \boldsymbol{\omega}_k^{vo} &= \boldsymbol{\omega}_k + \mathbf{n}_{w,k}^{vo} \\ \boldsymbol{\omega}_k^{imu} &= \boldsymbol{\omega}_k + \mathbf{n}_{w,k}^{imu} . \end{aligned} \quad (5)$$

Here, \mathbf{v}_k^{vo} and $\boldsymbol{\omega}_k^{vo}$ are translational and angular velocity measurements provided by visual odometry (vo), $\boldsymbol{\omega}_k^{imu}$ is the gyro output provided by the IMU and \mathbf{T}_k^{gps} is the position measurement provided by the GPS unit. Uncertainty in the visual odometry pose estimates, represented by the noise components $\mathbf{n}_{v,k}^{vo}$ and $\mathbf{n}_{w,k}^{vo}$, is estimated based on the reprojection error covariance of image features through backward propagation [14]. The gyro noise errors are modeled with fixed standard deviation values that are much higher than those corresponding to the visual odometry noise when the pose estimates are good (which is the most common case in an outdoor situation) and are comparable in value or sometimes even less when vision based pose estimation is difficult due to brief instances which is mostly common in an indoor situation. This allows the filter to effectively combine the two measurements at each time update, relying more on the sensor with the better noise characteristics. In addition, to control the amount of vertical drift we use elevation measurements that are assumed to be constant except when there is staircase climbing. This assumes

the ground plane to be flat which is a good assumption except when the user moves from one floor to another inside a building. To accommodate local violations from this planar motion assumption due to kneeling, crouching, etc., 1 meter standard deviation for the measurement noise n_k^h is used

$$h_k = \mathbf{T}_k(2) + n_k^h . \quad (6)$$

In order to obtain the initial position and attitude of the camera in navigation coordinates, roll and pitch outputs from the IMU are used directly and heading is obtained by the aid of the first two GPS readings that are sufficiently spread apart. During filter operation bad measurements from all sensors are rejected using validation mechanisms based on Chi-square tests on the Kalman innovations. In addition, those measurements from visual odometry causing large accelerations are also discarded.

VII. EXPERIMENTAL RESULTS

A. System Setup

As shown in Figure 5, our system consists of a wearable lightweight backpack frame equipped with stereo cameras, IMU, a GPS unit and a laptop PC. The cameras produce grayscale 640 by 480 pixel images and are externally triggered at 30Hz. A custom synchronization unit allows very precise synchronization between camera frames and IMU measurements. The laptop PC is outfitted with RAID (redundant array of independent disks) configured at level 0 (striped set) that allows continuous streaming of uncompressed data to disk without frame-drops.

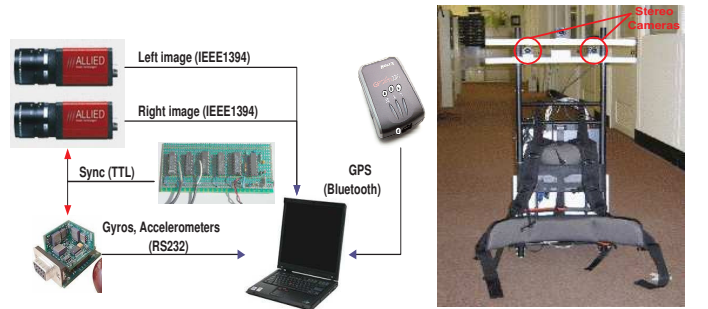


Fig. 5. The wearable backpack-mounted system

B. Performance of the Improved Visual Odometry

In order to evaluate the performance of our improved visual odometry system, we applied it to a set of collected video sequences with ground-truth. Specifically, each video sequence (containing both left and right images of the stereo pairs) was recorded in real-time while a user wearing our system was traveling along a set of predefined trajectories. Each trajectory is around several hundred meters long, and a set of key-points were set along the trajectories. During data collection, the user had to travel along the predefined trajectory and pass through every key-point. Both indoor and outdoor scenes are included in the predefined trajectories. In addition, the user was required to perform different maneuvers including

running, walking, lying down, crouching and sitting in a fast-moving truck during traveling. Figure 6 shows an example of the left images of the recorded stereo pairs.

The location of the key-points along each trajectory was measured by high-precision Differential GPS (DGPS). DGPS utilizes a differential correction technique to reduce the positional error so that it can achieve the positional measurement with up to 2cm accuracy. Hence, the 3D trajectories measured by DGPS will serve as the ground-truth to evaluate the performance of our improved visual odometry system. Figure 7 (a) shows a measured 3D trajectory of the user travelled by DGPS, which is around 106 meters long.

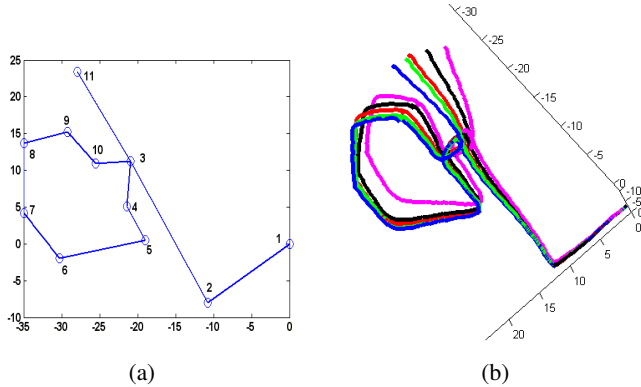


Fig. 7. The comparison between (a) the 3D trajectory measured by DGPS, and (b) the 3D trajectories estimated by our improved visual odometry under different improvements

However, the coordinate systems of the DGPS and the visual odometry are different so that the accuracy of the visual odometry cannot be measured directly. In order to avoid the coordinate alignment, a simple but effective error metric referred as relative localization accuracy is developed.

Specifically, the relative localization accuracy metric is computed as follows. First, the coordinates of the set of key-points along the 3D trajectories measured by both DGPS and our visual odometry system are collected. Second, select a key-point as a reference point, and compute the distance between the reference point and each of the remaining key-points one by one in both coordinate systems individually. Once all the distances are computed, they are summed in both coordinate systems individually. Subsequently, the summed distances from both coordinate systems are subtracted and divided by the number of remaining key-points to obtain an average error. Finally, the absolute value of the obtained average error is assigned to the reference point.

With the use of above metric, an error measurement can be derived to tell how well the system is able to provide location information relative to a given fixed location e.g. a street corner, or distance between two known intersections. Finally, the average of the relative localization accuracies over all the key-points is utilized to characterize the overall performance of our improved visual odometry system.

Figure 7 (b) shows an example of the measured 3D trajectories of the user by our improved visual odometry system under different improvements proposed in the previous sections. We

can see that the estimated 3D path looks more and more similar to the measured 3D path by DGPS visually as the proposed improvements are integrated one by one. In addition, Table I summarizes the relative localization accuracies of our improved visual odometry system under these improvements. Clearly, we can see that as each proposed improvement is integrated into the visual odometry system one by one, the average relative localization accuracy increases from 4 meters originally to less than 1 meter eventually.

TABLE I
THE RELATIVE LOCALIZATION ACCURACIES OF THE IMPROVED VISUAL ODOMETRY SYSTEM UNDER DIFFERENT IMPROVEMENTS (METERS)

	Original (magenta)	Dynamic reference selection (black)	Stereo constraint (red)	Dynamic local landmark tracking (green)	Outlier rejection (blue)
Min.	0.1480	0.1151	0.6679	0.4877	0.0648
Max.	10.4110	7.3310	3.8894	3.0959	2.0981
Med.	2.4633	2.2671	0.8933	0.8458	0.7907
Avg.	4.1788	2.8722	1.5710	1.2224	0.8311

C. Performance of the Integrated System

In certain situations such as poor illuminations or non-textured scenes, the captured images of the cameras fail to provide sufficient features for the pose estimation so that the visual odometry fails to work properly. For example, as shown in Figure 8 (a), the camera sees mostly white walls so that very few features concentrated in a small portion of the scene are extracted. As a result, the visual odometry alone cannot estimate pose accurately and will drift. In order to reduce the drift, the proposed EKF integration with IMU and GPS is activated.

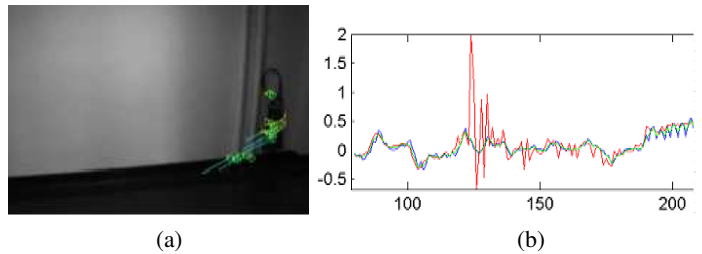


Fig. 8. IMU and visual odometry integration results: (a) sample frame from corridor sequence; (b) angular measurements

After integration, Figure 8 (b) shows the z-axis angle measurements from IMU (blue), visual odometry (red) as well as the filter output (green). Since the designed kalman filter is capable of producing the optimal output by combining with the best sensor measurement, as shown in Figure 8 (b), the filter output will automatically follow the visual odometry measurements closely at the beginning when the visual odometry is the most accurate, and then follows the IMU measurements mostly during the difficult portions of the sequence for the visual odometry.



Fig. 6. The randomly selected left images of a recorded video sequence. The images are frame 1th, 654th, 749th, 1739th, 1864th and 2447th from left to right respectively.

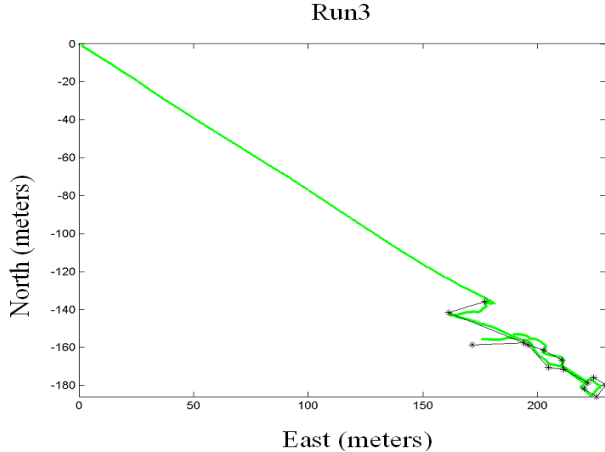


Fig. 9. The comparisons between the estimated trajectory of our integrated system (green) and the measured trajectory by DGPS (dark)

TABLE II
THE GLOBAL LOCALIZATION ACCURACIES OF SIX RUNS

Errors (meters)	run1	run2	run3	run4	run5	run6
Mean	2.79	0.84	1.56	1.20	1.73	1.99
Median	2.91	0.51	1.14	1.20	1.30	1.86
Total length (meters)	478	114	428	96	378	788

Figure 9 shows two measurements of a same trajectory by our integrated system and DGPS respectively. It illustrates clearly that they follow each other very well visually. After the filtering, since the coordinate systems of DGPS and our integrated system are same, the performance of our integrated system can be evaluated directly. Table II summarizes the average and median of the global localization accuracies obtained from six test runs performed, where the global localization accuracy is computed directly by using the position deviations of each key-point measured in DGPS and our integrated system. It shows that our integrated system can achieve less than 2-meter average global localization accuracy for 5 of the 6 runs. Note, run 3, 5 and 6 had significant portions, which were indoors, and no or very poor GPS was available there. The location accuracy of indoor points is also less than 2 meters in average.

VIII. CONCLUSION

In this paper, we have proposed a set of computer vision techniques to improve our earlier stereo-based visual odometry system [1] so that the drift can be reduced significantly over long-distance navigation. In addition, by integrating with IMU

and GPS measurements simultaneously, the robustness and accuracy of our integrated visual odometry system has been further improved. Real navigation tests show that our improved visual odometry system can achieve around 2-meter global localization accuracy for 100-meter to 500-meter long navigation experiments. We believe this can be further improved by using higher resolution cameras, global landmark matching and additional cameras to provide an even greater field of view.

REFERENCES

- [1] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conference on CVPR'04*, 2004.
- [2] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.
- [3] C. Olson, L. Matthies, M. Schoppers, and M. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, 2003.
- [4] M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotic and Automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [5] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *IEEE International Conference on Computer Vision*, 2003.
- [6] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *IEEE International Conference on Intelligent Robots and Systems*, 2004.
- [7] K. Sutherland and W. Thompson, "Inexact navigation," in *IEEE International Conference on Robotics and Automation*, 1993.
- [8] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *IEEE International Conference on Intelligent Robots and Systems*, 2002.
- [9] A. Johnson, J. Montgomery, and L. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *IEEE International Conference on Robotics and Automation*, 2005.
- [10] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," in *IEEE International Conference on Computer Vision Systems*, 2006.
- [11] S. Roumeliotis, A. Johnson, and J. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *IEEE International Conference on Robotics and Automation*, 2002.
- [12] R. Volpe, "Mars rover navigation results using sun sensor heading determination," in *IEEE International Conference on Intelligent Robot and Systems*, 1999.
- [13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988.
- [14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

Technology Evaluations and Performance Metrics for Soldier-Worn Sensors for ASSIST

Brian A. Weiss, Craig Schlenoff, Michael Shneier, and Ann Virts
National Institute of Standards and Technology (NIST)
100 Bureau Drive, Stop 8230
Gaithersburg, MD, USA
{brian.weiss}, {craig.schlenoff}, {michael.shneier}, {ann.virts} @nist.gov

Abstract—The DARPA-funded Advanced Soldier Sensor Information Systems and Technology (ASSIST) project is aimed at developing soldier-worn sensors and software to increase a soldier’s battlefield awareness during missions, provide them with data collection tools to augment their mission reporting capabilities following their field operations, and supply additional information to intelligence officers to enhance planning for future missions. The NIST-led Independent Evaluation Team is responsible for evaluating the ASSIST technologies developed by the Task 2 research teams. This paper discusses the overall Task 2 technologies of image/video, audio, and soldier activity data analysis capabilities with each participating research team’s technologies presented at deeper levels. After understanding the technologies, the five elemental tests (Arabic text translation, object detection/image recognition, shooter localization, soldier state/localization, and sound/speech recognition) are designed and implemented with metrics for the Baseline and Final Phase I Evaluations.

Keywords: DARPA, ASSIST, soldier-worn sensors, evaluations, performance metrics, elemental tests, object detection, image classification, shooter localization, sound recognition, speech recognition, soldier state, soldier localization

I. INTRODUCTION

The Advanced Soldier Sensor Information System and Technology (ASSIST) is a Defense Advanced Research Projects Agency (DARPA) supported research and development program. This effort intends to advance and exploit soldier-worn sensors to increase soldiers’ battlefield awareness during humanitarian and combat missions, provide enhanced data collection tools to augment mission reporting capabilities following field operations, and supply additional information to intelligence officers to improve mission planning all within military operations in urban terrain (MOUT) environments. [1] This program is split into two efforts:

- Task 1 emphasizes active information capture and voice annotations exploitation hardware. The resulting products will be prototype wearable capture units and the supporting software for processing, logging and retrieval.

- Task 2 stresses passive collection and automated activity/object recognition. The results from this task will be the software and tools that will undergo system integration in later program phases.

The National Institute of Standards and Technology (NIST), along with its subcontractors (Aptima, Inc. and DCS Corporation), was funded to serve as the Independent Evaluation Team (IET) for Task 2, Phase I. In this role, NIST was responsible for:

- Understanding the Task 2 research technologies
- Devising a testing approach for these technologies
- Identifying a MOUT site to evaluate these technologies
- Designing and executing the tests
- Developing performance metrics to analyze the data
- Documenting the test results

Section II presents the tested technologies along with the specific capabilities of the participating research teams, both at the 6-month (Baseline) and at the 12-month (Final Phase I) evaluations. Section III presents the elemental tests including enhancements that were made between the two evaluations and the performance metrics developed to evaluate the tested technologies. Section IV summarizes the paper.

II. TECHNOLOGIES FOR EVALUATION

Task 2 involves developing a range of data capture, analysis, and display technologies. These capabilities are broken down into three data type categories. Within each data type, several “technology elements” are applied to organize, process and present that data. Some of the key technology elements being applied in the ASSIST program are listed below:

“Image/Video Data Analysis Capabilities”

- Object detection/image classification – the ability to recognize objects (e.g. vehicles, people, etc.) through analysis of video, imagery, and/or related data
- Arabic text translation – the ability to detect, recognize, and translate written Arabic text through imagery analysis

“Audio Data Analysis Capabilities”

- Sound recognition/speech recognition – the ability to identify sound events (e.g. gunshots, vehicles, etc.) and recognize speech (e.g. keyword spotting, foreign language identification, etc.) in audio data
- Shooter localization – the ability to identify gunshots in the environment (e.g. through analysis of audio data), including the type of weapon producing those shots, and the location of the shooter

“Soldier Activity Data Analysis Capabilities”

- Soldier state identification/soldier localization – the ability to identify a soldier’s path of movement within an environment and characterize their actions (e.g. running, walking, climbing stairs, etc.)

Presently, there is no single integrated system within the ASSIST program. Instead, several universities and corporations have collaborated to form “research teams”. Each organization is developing specific technologies with these components being gradually integrated as a “research team” system. The following subsections provide overviews of the specific “research team” technologies and describe the progression of each team’s system from the Baseline Evaluation to the Final Phase I Evaluation.

A. IBM/MIT/Georgia Tech Team ASSIST Technologies

The IBM Team combines researchers from IBM, the Georgia Institute of Technology, and the Massachusetts Institute of Technology. The team’s long-term vision for their ASSIST suite is a comprehensive system that captures, analyzes, organizes, and archives data for users (soldiers and intelligence officers) to review and search records to augment military reporting and mission planning. The IBM team’s technologies include:

- Image classification – The presence of an object is detected based upon data from image and audio sensors and classified with one or more classes and subclasses. For the Baseline Evaluation, images were classified to contain the presence of *Outdoors*, *Indoors*, *Sky*, *Buildings*, *Vegetation*, *People*, *Weapons*, and *Vehicles*. The IBM team expanded their capabilities for the Final Phase I Evaluation to detect the presence of *Soldiers*, *Commotion*, *Vehicle_civil*, *Vehicle_military*, and *Cars* (in addition to their baseline capabilities).
- Object detection – Objects are detected and localized (a bounding box is created) to a specific region within an image. For the Baseline Evaluation, the IBM team detected and localized *Faces* and *License_plates*. During the Final Phase I Evaluation they could also detect and localize *Clothing_Color*.
- Sound recognition – Recorded audio from the environment that is classified as “non-speech

sounds” is further classified into the following: sounds from a car and large truck (Baseline Evaluation) plus single gunshots, machine gunfire, explosions, light trucks, sedans, and transport vans (Final Phase I Evaluation).

- Speech recognition – Keyword extraction is performed on a soldier’s speech (English). Keywords that are detected during the Baseline Evaluation include *insurgent*, *target*, *dead*, *shot*, *shots*, *suspicious*, *killed*, *kill*, *fire*, *incoming*, *contact*, *weapon*, *weapons*, *intelligence*, *Intel*, etc. Keywords that are added during the Final Phase I Evaluation include *update*, *A4 millimeter round*, *AK47*, *alpha*, *bravo*, *C4*, *frag-out*, *halt*, *IED*, *M16*, *RPG*, *SIT-REP*, and *tango*.
- Language identification – Capability to identify spoken English, German, Japanese, Mandarin, Spanish, and Hindi (Baseline Evaluation) and later Arabic and French (Final Phase I Evaluation).
- Soldier state identification – Capability to determine when a soldier is performing the following actions: standing, walking, running, driving, and lying down (Baseline Evaluation) along with opening doors, performing a situational assessment from cover, taking a knee, sitting, raising a weapon, shaking hands, crawling, going upstairs and going downstairs (Final Phase I Evaluation).

B. Sarnoff Team ASSIST Technologies

The Sarnoff ASSIST team is composed of three organizations: Sarnoff Corporation, Carnegie Mellon Institute, and Vanderbilt University. Each of these groups is focused on unique technologies that will not be integrated with one another during this phase of the project. This resulted in each organization being treated as a separate team. The following subsections discuss the technologies from these three groups.

1) *Sarnoff Corporation’s System*: Sarnoff is developing an ASSIST system to support soldier localization and object detection. These technologies are discussed further:

- Object detection – An object is detected and localized (specified with bounding boxes) to a region within an image. Sarnoff was able to detect *Vehicles* and *People* at the baseline plus *Faces*, *Weapons*, and *Vehicle_type* for the final evaluation.
- Soldier localization – Capability of locating (in GPS coordinates) the ASSIST-wearer both indoors and outdoors. This capability did not change between the evaluations, rather the software algorithms were refined following the Baseline Evaluation.

2) *Carnegie Mellon University’s (CMU) System*: CMU is developing technologies aimed at extracting and translating

Arabic text from images captured with a typical consumer-grade digital camera. Their technology operates in three stages:

- Arabic text is identified within an image through edge detection, layout analysis and search algorithms.
- Arabic text is extracted from an image using optical character recognition software.
- Arabic text is translated to English using statistical machine translation technology.

Again, there were no capability increases for this technology between the two evaluations, rather software refinements were made to improve each of the three technology stages.

3) *Vanderbilt University's System*: Vanderbilt University (also referred to as Vanderbilt) is developing a shooter localization technology that detects gunfire, determines bullet trajectory, localizes the shooter, classifies the bullet caliber and identifies the type of weapon being fired. Their current hardware suite consists of 10 acoustic sensors. The system's capabilities are below:

- Shot Localization – Determine bullet trajectories and shooter origins of short-range (≈ 30 meters) shots using single and multiple shooters along with determining trajectories of long-range (≈ 100 meters) shots (Baseline Evaluation) plus determining the trajectories of longer-range (200 meters to 300 meters) shots along with determining the trajectories and shooter origins of automatic fire at shorter ranges (Final Phase I Evaluation).
- Shot Classification – Classify shots from an M16, AK-47, and M107 (Baseline Evaluation) plus classifying shots from an M4, M240, and M249 (Final Phase I Evaluation).

C. University of Washington Team

The University of Washington (also referred to as Washington) team consists of the University of Washington, Intel Research Seattle, and Lupine Logic. This team's system is aimed at collecting soldier state data. Specifically, identifying whether a soldier is indoors, outdoors, riding in a vehicle, walking, running, standing, performing a situational assessment from cover, going upstairs and going downstairs at both evaluations. Again, no capability improvements were made between the two evaluations, rather software enhancements are made following Baseline Evaluation.

III. ELEMENTAL TESTS

The IET developed a two-part test methodology to produce the following three metrics (as stated per DARPA):

- Measure the accuracy of object/event/activity identification and labeling
- Measure the system's ability to improve its classification performance through learning
- Measure the utility of the system in enhancing operational effectiveness

The first two metrics are evaluated through "elemental tests" while the last metric is evaluated through "vignette tests". [2] Elemental tests are developed to test the ASSIST technologies in an "idealistic" environment and allow a focused examination of the specific components. Vignette tests immerse the technologies in realistic military scenarios to assess the systems in more practical, fast-paced, stressed conditions. This paper focuses on the elemental tests.

The elemental tests afford the ability to modify specific variables in a controlled manner to measure the impact of those variables on technology performance within a MOUT environment. Five elemental tests are developed:

- Arabic text translation
- Object detection/image classification
- Shooter localization
- Soldier state/localization
- Sound/speech recognition

Each of these elemental tests is discussed in detail in the following subsections.

A. Arabic Text Translation

The Arabic text translation elemental test was specifically designed to evaluate CMU's ASSIST ability to detect, recognize, and translate Arabic signs. Again, this elemental test seeks to evaluate the three key Arabic text translation capabilities:

- Identify Arabic text in an image
- Extract Arabic text from an image
- Translate Arabic text to English text

1) *Test Description – Baseline Evaluation*: A single approach was taken in evaluating these three capabilities. This was accomplished by placing six Arabic signs in the MOUT environment and having CMU collect imagery data at two distances (near and far) and five angles (30°, 60°, 90°, 105°, and 135° with 90° being a straight-on view of the sign). Distances were based upon the letter-size of the specific signs with the near distance corresponding to approximately 50 pixels per height of the smallest letter in the sign and the far distance corresponding to approximately 30 pixels per height of the smallest letter of the sign when using CMU's consumer-grade camera. Signs were placed both indoors and outdoors. The location of each sign placed in the environment along with their associated data collection points were

measured with two-centimeter accuracy.

The test began with the researcher collecting images of the signs from the various distances and angles. The test then proceeded through three successive stages whereby each was evaluated:

- Sign detection (step 1) – The signs placed in the environment were used to evaluate the ability of the system to extract regions of text.
- Text extraction (step 2) – The regions extracted from the signs in step 1 were processed to extract and localize Arabic characters and words.
- Text translation (step 3) – The output from step 2 was fed into the translation component and the English output was evaluated both quantitatively and qualitatively by a native Arabic speaker.

2) *Lessons Learned – Baseline Evaluation:* The testing approach taken during the Baseline Evaluation where technology performance of one step is dependent upon the technology performance of a previous step (i.e. successful text extraction that is dependent upon successful sign detection) made it impossible to accurately test the system's text extraction and translation capabilities. The test approach was modified for the Final Phase I Evaluation where the three individual steps of the system were evaluated separately in addition to conducting an overall (step successive) evaluation.

3) *Test Description – Final Phase I Evaluation:* To enable a comparison with the baseline, three signs were placed in the environment at marked positions so that sets of images could be taken at the same angles and appropriate distances. Of the three signs, one is a sign that was used during the baseline and setup at its original location while the other two signs have never been used before. Images were captured of these signs and the data is put through the three-step process. This was the overall test that enables direct comparison of the system's capabilities between the two evaluations. This process also allowed for the individual evaluation of sign detection (step 1).

In addition, text extraction (step 2) was separately evaluated by feeding Arabic letters and words in "ideal" fonts directly into the optical character recognition (OCR) program. In order to test the ability of the text extraction to deal with more complex backgrounds, two signs with textured backgrounds were used, two signs were input with an image as well as text, one sign included English numbers as well as Arabic text, and two signs had colored backgrounds.

The text translation component (step 3) of the system was tested in a similar way. Fifteen text files were created containing Arabic text taken from real signs. The files were encoded in the required format and input into the program one at a time. Once again, this provided an ideal situation for the

translation system, with no misspelled words, no extra characters, and no missing characters.

4) *Metrics for Evaluation:* The following metrics are identified and used to evaluate this technology:

- Text rows correctly extracted (%)
- Non-text regions found/false alarms (%)
- Characters correctly localized (%)
- Arabic words correct (%)
- English words correct (%)

B. Object Detection/Image Classification

The object detection/image classification elemental test evaluated the following capabilities of the IBM team's and Sarnoff's ASSIST systems:

- Presence detection of objects and states within imagery (IBM)
- Localized detection of objects within specific regions of imagery (IBM, Sarnoff)

1) *Test Description – Baseline Evaluation:* Prior to the evaluation, the ≈45 meters squared, courtyard (containing 10-single story and two double-story buildings) was setup with objects. Each building contained multiple doors and windows and is populated with various amounts of furniture (e.g. chairs, desks, tables, etc.). Approximately 50 waypoints (using two-centimeter accurate, differential GPS and surveying equipment) were marked to include a range of indoor, outdoor, ground-level, and upper-story locations (including positions in front of doorways, windows, etc.). These waypoints were used to denote imagery collection locations for the ASSIST-wearer, and the locations of additional objects to be placed in the environment.

Additional objects in the environment include vehicles (both civilian and military) with license plates (both US and Iraqi), people (soldiers and civilians dressed in simulated middle-eastern attire), weapons (both US military and foreign that were either carried by people or placed within the environment), Arabic signs, tires (both stacked vertically and resting against buildings), trash piles, barrels, sandbag piles, etc. The following variables were taken into account when selecting the locations of objects and imagery viewpoints:

- Position of ASSIST-wearer
 - Ground level
 - Upper level
- Position of ASSIST-wearer relative to object(s)
 - Both indoors
 - ASSIST-wearer indoors with objects outdoors
 - ASSIST-wearer outdoors with objects

- indoors
 - o Both outdoors
- Object(s) orientation relative to ASSIST-wearer
 - o Above, below, same level
 - o Head-on, angled, side-view, rear-view
- Object distance relative to ASSIST-wearer
 - o Near (<5 meters)
 - o Mid-range (<20 meters)
 - o Far (>20 meters)
- Object occlusion relative to ASSIST-wearer
 - o Entirely visible
 - o Partially occluded by other objects
- Background scene relative to object(s)
 - o Objects viewed with other objects close behind them vs. far away
 - o Objects viewed with objects behind them with similar colors vs. objects behind them with contrasting colors

Imagery was collected from 25 viewpoints that were distributed across 10 waypoints, most of which had multiple viewpoints at different orientations. Labeled doormats were placed at each waypoint to indicate the ASSIST-wearer's orientation for imagery collection. Each team collected a single image at each of the 25 viewpoints. The IET also collected imagery data from each viewpoint using its own consumer-grade, digital camera.

2) *Lessons Learned – Baseline Evaluation:* Several improvements were realized following the Baseline Evaluation. A greater quantity and diversity of objects (e.g. people in a wider range of attires, etc.) including clutter (e.g. wires hanging from buildings, more trash, etc.) should be added. The elemental test should also provide data collection points across a larger area of the MOUT. Another issue was that imagery collected from upper level locations allowed the ASSIST systems to capture data outside of the control area whereas ground locations only allowed imagery out to a very finite distance.

3) *Test Description – Final Phase I Evaluation:* This elemental test evolved to address the shortcomings of the first evaluation. First, the test area was expanded so that data collection viewpoints were added in both the courtyard and the warlord compound (≈100 meters x ≈60 meters with three, single-story buildings and a double-story building). Overall object density and diversity was increased as more objects (specifically, people and vehicles) were added to the environment (additional GPS waypoints were surveyed). Data viewpoints were also modified so that imagery was only collected from ground level to better control the viewing area.

4) *Metrics for Evaluation:* The imagery data that each team captured with its ASSIST system was used as both data for experimental analysis and as ground-truth. If an object could be viewed within a team's image, then it was evaluated

against the team's processed data (e.g. if a vehicle is visible in a team's image, then the team would be evaluated whether it could detect the vehicle or not). Likewise, if a human is not able to discern an object from viewing an image, then the team was not evaluated against that object. Output data includes:

- Positive identification (true positive) - an object was correctly identified
- Negative identification (false positive) – an object was identified that is not present
- Missed identification (false negative) – an object was not identified that is present
- Total instances of presence (total presence) – sum of positive identifications and missed identifications
- Total identifications – sum of positive identifications and negative identifications

The following metrics were applied based upon the output data:

- Positive identifications over total presence (%)
- Missed identifications over total presence (%)
- Positive identifications over total identifications (%)
- Negative identifications over total identifications (%)

C. Shooter Localization

The shooter localization elemental test evaluated the capabilities of Vanderbilt's ASSIST system to:

- Detect gunshots
- Calculate a bullet's trajectory
- Localize a shooter's origin
- Classify the caliber of bullet being fired
- Identify the specific weapon being fired

1) *Test Description – Baseline Evaluation:* This test was conducted at Aberdeen's outdoor firing range, due to restrictions against live fire at the MOUT site. A "zero line" and four firing lines (≈25 meter, ≈50 meter, ≈100 meter, and ≈200 meter) were marked on the range. The ASSIST system's acoustic sensors were placed on and behind the zero line, and randomly covered an area that was ≈30 meters squared. Five targets were set up behind the sensor region. Simple, wooden-walled structures (single-story and two-story) with windows were constructed at the firing lines and in the sensor region to simulate the buildings and obstructions that would be found in a MOUT environment, and to provide unique shooter positions through windows, next to walls, and on upper levels.

Five to seven shooter positions (both practice and test positions) were marked at each firing line. All positions on the firing range (sensors, targets, shooter positions and wall

corners) were localized to within two-centimeter accuracy using differential GPS. The following variables were considered in the placement of shooter positions:

- Shooter positioning relative to walls at the firing line
 - From a clearing
 - Next to a wall
 - From within a structure with the weapon's barrel protruding out of a window
- Obstructions between the firing line and sensor field
 - Positions obstructed by walls that could occlude a weapon's muzzle blast and/or shockwave from a subset of the sensors
 - Positions with clear line of sight to the sensors

A shot matrix was developed for 200+ shots with the following variables considered:

- Weapon and caliber [M16, M4, & M249 (5.56mm), AK47 & M240 (7.62mm), M107 (50 caliber)]
- Firing lines (≈ 25 m, ≈ 50 m, ≈ 100 m, ≈ 200 m)
- Shooter positions from the four firing lines
- Rounds per test (single shot vs. 3-round bursts)
- Number of shooters (single shooter vs. multiple shooters)
- Weapons fired by multiple shooters (same weapon vs. different weapon)
- Bullet trajectory (shots that crossed in between a majority of sensors vs. shots that passed by very few sensors near the perimeter)

Shots were fired at each of the four firing lines and data was collected.

2) *Lessons Learned – Baseline Evaluation:* Following the Baseline Evaluation, several enhancements were recognized that would improve the operational relevance and expand the complexity of this elemental test. First was that there is little operational relevance in testing from the ≈ 25 meter firing line. Additionally, shooters (particularly snipers) will typically fire from within structures where their weapon's barrel is not protruding out of a window/opening. Also, shooters will sometimes strafe up towards a target whereby they can see their bullets hit the ground in front of their target and adjust their trajectory accordingly.

3) *Test Description – Final Phase I Evaluation:* This later evaluation addressed all of the lessons learned from the Baseline Evaluation. First was the elimination of the ≈ 25 m firing line and the addition of the ≈ 300 m firing line. Second was to add a shooter position at each firing line from within one of the wooden structures that forced the weapon barrel to be recessed at least 1 m to 2 m from a window. Lastly, targets (additional to those placed behind the sensors) were placed in

front of the sensor region. The shot matrix was updated with ≈ 250 shots.

4) *Metrics for Evaluation:* The ASSIST system's output data was evaluated against the following three metric categories:

- Detection (broken down by firing line, shooter position, and weapon caliber plus variants to evaluate multiple shooter detections)
 - Shot detections over all shots fired (%)
 - Trajectory detections over all shots detected (%)
 - Shooter origin detections over all shots detected (%)
- Localization (broken down by firing line, shooter position, bullet caliber, and single shot vs. 3-round burst)
 - Shooter origin (m) – accuracy and precision
 - Trajectory angle (degrees) – accuracy and precision
 - Target crossing (m) – accuracy and precision
- Classification (broken down by firing line, specific weapon, and specific bullet caliber)
 - Correct shot detections by weapon (%)
 - Correct shot detections by bullet caliber (%)

D. Soldier/State Localization

The goal of the soldier state/localization elemental test was to evaluate the ASSIST systems' ability to localize a soldier within indoor and outdoor environments, and to characterize their actions. The IBM, Sarnoff, and Washington teams participated in this test, with each team outputting different information (see Section II for further detail).

1) *Test Description – Baseline Evaluation:* There were four test runs, each of which was performed twice. Each run exposed the system to a different level of difficulty for soldier state / localization identification. Each run required a soldier, shadowed by a researcher wearing the ASSIST system, to traverse a predefined path of waypoints in a scripted fashion. Run 1 was only outside in open areas. Run 2 was also outside but included some tight, GPS-restricted areas. Run 3 was both outside and inside, but did not force an elevation change. Run 4 was predominantly inside and traversed two floors of a building (one of the ground and the other elevated one story).

Approximately 60 waypoints were marked (including indoor, outdoor, ground-level and upper level points) with two-centimeter accuracy using differential GPS and surveying equipment. Poles were placed in cones at each waypoint. Signs attached to the poles indicated a letter for each waypoint in a run (e.g. A, then B, etc.), gave a brief description of the action to be performed at the waypoint and on the way to the

next waypoint (e.g. “lie down for 10 seconds then run”, “go up stairs”, etc.), and provided an arrow pointing to the next waypoint. The actions scripted were dictated by the superset of stated capabilities by all three teams’ ASSIST systems.

Before the execution of each run, the soldiers and the researchers walked the path of the run to become familiar with the route and actions. During the run, three observers captured the time that the ASSIST-wearer reached the waypoints and performed the specified actions. Observers also noted any inconsistencies in the actual actions of the ASSIST-wearer relative to the scripts. This data allowed the IET to accurately capture ground truth and measure the ASSIST system’s accuracy in localizing the ASSIST-wearer and identifying actions.

2) *Lessons Learned – Baseline Evaluation:* Several test concerns were noticed following this initial evaluation. Although each of the four runs was performed twice, the individual runs were relatively short in time and distance covered. Also, the range of actions was relatively limited.

3) *Test Description – Final Phase I Evaluation:* This elemental test was refined to address the concerns highlighted from the Baseline Evaluation. Instead of having each team perform the four original runs twice, the four original runs were performed in reverse and two additional runs were added (for a total of six runs). Performing the original four runs in reverse still provided a means of comparing data between the two evaluations. Run 5 involved a loop around a large portion of the MOUT complex, in which each action occurred for a longer period of time and distance. Run 6 (also run in a larger MOUT area) included much more driving and going up and down stairs. Also, each of the four original runs had some of their actions supplemented with more complex actions (e.g. raise weapon for 10 seconds, drag a sandbag, etc.). To account for these additional runs, more GPS waypoints were surveyed with the same two-centimeter accuracy.

4) *Metrics for Evaluation:* Soldier state accuracy was calculated by comparing ground truth times of ASSIST-wearer actions to the actions identified by each ASSIST system. All overlapping time periods were analyzed for correspondence. For example, if the ground truth showed that the ASSIST-wearer was walking from 0 seconds to 5 seconds and running from 5 seconds to 10 seconds, and the ASSIST system showed that the ASSIST-wearer was walking from 0 seconds to 7 seconds and running from 8 seconds to 10 seconds, the time periods were analyzed independently for correspondence. In this case, there would be a match from 0 seconds to 5 seconds and from 8 seconds to 10 seconds, with an incorrect detection from 5 seconds to 7 seconds. Specific state metrics include:

- Correctly identified movement vs. stationary (%)
- Correctly classified type of movement (%)

- Incorrectly classified type of movement (%)
- Unclassified soldier movements (%)
- % Correctly identified indoor vs. outdoor activity

Soldier localization accuracy was calculated by comparing the ground truth location of waypoints to locations returned by the ASSIST systems at specific times. Observers noted the exact time that the ASSIST-wearer reached each waypoint. These location and times were then compared to the data returned from the ASSIST system. To account for human error and non-exact clock time between systems a 4-second window (2 seconds before and 2 seconds after the exact time) were introduced when comparing the locations. The location returned by the ASSIST system that was closest to the ground truth location within this time window was used in the analysis. Specific localization metrics include:

- Accuracy (m) of mapping all soldier movement
- Accuracy (m) of mapping all outdoor movement
- Accuracy (m) of mapping all indoor movement

E. Sound / Speech Recognition

The goal of the sound recognition test was to evaluate the ASSIST system’s ability to detect certain sounds in the environment. Since only the IBM team has the ASSIST capabilities to perform this type of test, the sounds and speech presented in this test are aligned with the team’s technology.

1) *Test Description – Baseline Evaluation:* To conduct this elemental test, the following sound events were scripted to occur in the environment at specified times relative to the start of a given evaluation run:

- A soldier fired blank bullet rounds (5.56mm, 7.62mm, and 50 caliber)
- A soldier standing next to the ASSIST-wearer spoke one of ten text phrase which incorporated some combination of the team’s stated-capability keywords
- A person in the environment either spoke or played a digital voice recording of people speaking the stated-capability languages
- Vehicles were driven past the ASSIST-wearer while either accelerating or decelerating

The variables for this elemental test were as follows:

- Distance between the sound source and the ASSIST-wearer
- Speakers were stationary or moving (e.g. a person speaking a language in the environment could be stationary, walking away from the ASSIST-wearer, or walking towards the ASSIST-wearer).
- The level of ambient noise that was in the environment. For this condition, ambient noise was

either low (i.e. no additional ambient noise was added) or high (i.e. ambient noise was produced by a generator located ~7m from the ASSIST-wearer).

- ASSIST-wearer was stationary or moving
- Overlapping vs. non-overlapping sounds. In non-overlapping runs, each sound event was separated by a few seconds. In overlapping runs, multiple sounds occurred in the same time segment (e.g. a gunshot, a person speaking a language, etc.).

There were five runs of increasing complexity with each run performed twice. During the early runs, there was little or no ambient noise, the ASSIST-wearer was stationary, and there were no overlapping sounds. During the later runs, there was a lot of ambient noise, the ASSIST-wearer was moving, there were overlapping sounds, and the sounds in the environment were moving to and from the ASSIST-wearer.

Ground truth locations of the ASSIST-wearer and the sounds in the environment were measured based upon known points. Before the test, the locations of certain points in the environment were mapped out to two-centimeter GPS accuracy. When stationary, the ASSIST-wearer remained at one of these specified points in the environment; when moving, the ASSIST-wearer moved between these points. Similarly, the scripted sounds were generated at these locations or moved between them.

2) *Lessons Learned – Baseline Evaluation:* Following this evaluation, improvements were sought. The only realization was that the five runs were conducted in the same open environment. This meant that the environmental acoustics (potential presence of echoes, etc.) was not considered to be a variable.

3) *Test Description – Final Phase I Evaluation:* This later evaluation added two runs, each in a different part of the MOUT site as compared to the original five runs. This allowed the environmental acoustics to become an evaluation variable. The sixth run was outdoors, in a more confined area; closely surrounded by concrete walls. The seventh run was predominantly indoors. Additional keywords were also added to the soldier-spoken texts based upon the team's additional capabilities.

4) *Metrics for Evaluation:* This evaluation can be broken down into the following categories: sounds and speech recognition. The metrics applied for sound recognition:

- Correctly classified all sounds (%)
- Incorrectly classified all sounds (%)
- Unclassified sounds (%)
- Correctly classified sounds (broken down by vehicles, gunshots, foreign languages) (%)
- Incorrectly classified sounds (broken down by

vehicles, gunshots, foreign languages) (%)

- Unclassified sounds (broken down by vehicles, gunshots, foreign languages) (%)

The metrics applied for speech (keyword) recognition were:

- Correct keyword identifications (%)
- Missed keyword identifications (%)
- Incorrect keyword identifications (%)

IV. CONCLUSION

The IET successfully designed and implemented these five elemental tests for the DARPA ASSIST's Task 2, Phase I Baseline Evaluation and Final Phase I Evaluation. Metrics were consistently applied to the ASSIST teams' output elemental test data to achieve the DARPA-required high-level metrics:

- Measure the accuracy of object/event/activity identification and labeling (determined from data collected from each elemental test evaluation)
- Measure the system's ability to improve its classification performance through learning (demonstrated in comparing data between the baseline and final evaluations)

It is anticipated that the ASSIST program will continue for at least 3 more years and the NIST-led IET expects to continue to implement and improve upon its tests and metrics for future evaluations.

ACKNOWLEDGEMENT

This work was supported by the Defense Advanced Research Projects Agency (DARPA) ASSIST program (POC. Mari Maeda).

DISCLAIMER

Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

REFERENCES

- [1] DARPA, "Advanced Soldier Sensor Information System and Technology (ASSIST) Proposer Information Pamphlet," http://www.darpa.mil/ipto/solicitations/open/04-38_PIP.htm, 2006.
- [2] M. Steves "Utility Assessment of Soldier-Worn Sensors Systems for ASSIST", To appear Proceedings of Performance Metrics for Intelligent Systems Workshop, August 2006, Gaithersburg, MD.

Utility Assessments of Soldier-Worn Sensor Systems for ASSIST

Michelle Potts Steves

National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, Maryland, USA
msteves@nist.gov

Abstract—Utility assessments were performed for the Defense Advanced Research Projects Agency (DARPA) Advanced Soldier Sensor Information Systems and Technology (ASSIST) program [1]. This paper describes the field-based, formative methods used to assess utility for prototypical software designed to provide value to mission reporting for infantry and related intelligence operations. While results from these evaluations are not presented here, design considerations, evaluation procedures and metrics, as well as, lessons learned are described.

I. INTRODUCTION

The Advanced Soldier Sensor Information Systems and Technology (ASSIST) program [1] is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program. The objective of the ASSIST program is to exploit soldier-worn sensors to augment a soldier's recall and reporting capabilities to enhance mission reporting. The program consists of two parts, called tasks. Task 1, Baseline System Development, stresses active information capture and voice annotation exploitation. The resulting products from Task 1 are prototypical, wearable capture units and the supporting operational software for processing, logging, and information retrieval. Task 2 is Advanced Technology Research, it stresses passive collection and automated activity and object recognition. The anticipated results from this task are the algorithms, software, and tools that will undergo system integration in later phases of the program.

The National Institute of Standards and Technology (NIST), along with NIST subcontractors, Aptima and DCS Corporation, are funded to serve as the Independent Evaluation Team (IET) for Task 2. The IET is tasked with assessing the capabilities and developmental progress of the funded research systems. To that end, the IET identified two major assessment categories, technical performance and utility, that when combined would provide an overall assessment of value to the war fighter. Two very different types of tests were designed and administered to address these assessment objectives.

This paper focuses on the methods used by the IET to assess utility to the war fighter for these prototypical systems (Task 2). In a nutshell, utility was assessed by identifying soldiers' information needs with respect to mission reporting and intelligence product development, and then having the soldiers rate how well each information need was addressed by ASSIST contributions and the relative need for that

information. To date, two evaluations for ASSIST Task 2 teams have been performed, one in November 2005, a baseline test, and one in May 2006, a go/no-go (program) assessment. Each of these evaluation periods combined technical performance tests as well as assessments of utility. Descriptions of the technical performance tests can be found in [7]. This paper discusses how the IET designed and executed the assessments of utility provided by the ASSIST systems.

First, the paper presents some background on the ASSIST program and on assessing utility in prototypical software applications. Section III discusses the development and vetting of mission scenarios and environments, called vignettes. Section IV provides an overview of the test procedures used, while Section V presents a sampling of the metrics used and types of assessments made. Section VI concludes the paper with lessons learned.

II. BACKGROUND

A. Expected use of ASSIST systems

Soldiers perform missions of various types including presence patrols (where soldiers are tasked to make their presence known in an area), reconnaissance, apprehension of suspected insurgents, and so on. These missions can often be long and stressful. Regardless of the mission type, after a mission is complete, a unit, such as a platoon, typically provides a report or debriefing that describes any "events" encountered during the mission and collection for information requests that the unit was tasked with during the pre-mission briefing. Due to many factors, including human stress and fatigue, there are undoubtedly many instances in which important information is not captured in the report and thus is not available for use, such as planning for future missions.

The ASSIST program is addressing this challenge by supporting research in the instrumentation of a soldier with sensors that can be worn directly on a uniform. These sensors include microphones, video cameras, still cameras, Global Positioning Systems (GPS), Inertial Navigation Systems (INS), and accelerometers. The intent is to record continuously what is occurring around the soldier while on a mission. When soldiers return from a mission, the collected sensor data is run through a series of software systems which indexes it and creates an electronic chronicle of events that happened throughout the time that the ASSIST system was recording.

ASSIST, Task 2 system capabilities include:

- Image/video data analysis capabilities, including: object detection and image classification through analysis of video, imagery, and related data sources, Arabic text translation from image data, and change detection in related images over time
- Audio data analysis capabilities, including: sound and speech detection and recognition, and shooter localization and classification, e.g., origin of shots fired and identification of the weapon producing those shots
- Soldier activity data analysis capabilities, including mapping a soldier's path during a mission and activities along that mission path.

There is no single integrated ASSIST system at this point in the program's life-cycle. Instead, several university and corporate research and development organizations have formed into "research teams". Each organization is developing specific technology components, and these components are gradually being integrated as a "research team" system. See [5] for more detailed information on the program objectives and Task 2 system capabilities.

B. Evaluation and Utility Assessments

Several important benefits for research and development programs can be gleaned from evaluation. First, evaluations help program managers determine what progress is being made. These assessments can be used to obtain appropriate funding for programs that show progress and to determine alternative directions for programs that are less successful. Second, evaluations help researchers to see objectively how their software can help end users, and if necessary, help them refine their research goals. These more formative evaluations provide end-user feedback during pre-release software development phases geared towards showing current utility and utility enhancement opportunities.

In the early stages of software research and development, it would seem that effort spent on improving the utility of an application provides a greater long-term return than perfecting the overall usability of an application that may not provide increased utility, i.e., value, to the end-user. Given this premise for prototypical research software, it is appropriate to place an emphasis on assessing utility, i.e., the value the software application provides to the user, rather than focusing on training, user interface issues, and the like. However, *utility* and *usability* are certainly intertwined. Many have stated that usability today is multidimensional; it encompasses effectiveness, learnability, flexibility, and user attitudes towards the application, e.g., [2, 3, 4]. So, while we are certainly concerned with the eventual usability of the interface and usability issues that impact our assessments of utility of these prototypical applications, these other aspects of usability need only be "good enough" to assess the value the application provides currently and identify opportunities to increase the application's utility to the end user. With this in mind, the evaluations for this effort were designed to be more formative than summative. That is, the evaluations were

performed during design and development of the software applications with the intent of informing the design rather than summative or validation evaluations that are performed at the end of development [6].

The IET was interested in assessing utility to the war fighter; as such, we were concerned with impacts on both their processes and products. To reflect this perspective, user-centered metrics were employed. We attempted to identify metrics that would help assess such questions as: What information do infantry soldiers want and/or need after completing a mission in the field? How well are information needs met, both from the soldier perspective and the S2 perspective? What were the ASSIST contributions to mission reporting with respect to user-stated information needs?

III. VIGNETTE DEVELOPMENT

The vignette tests were designed to assess the value of ASSIST systems in 1) infantry squad reporting of critical information, events, and intelligence encountered during a mission, and 2) S2/intelligence operations. Additionally, the following design requirements were considered:

- Exercising the ASSIST technologies within their operating constraints
- Incorporating some environmental characteristics beyond the current system capabilities (i.e., establishing baseline system performance for comparison in future tests)
- Providing an operationally relevant environment within which infantry procedures could be executed safely but in a reasonably realistic manner
- Establishing a rich information environment (e.g., population dynamics, multi-sensory stimuli, realistically evolving terrain, etc.) to support intelligence collection and analysis procedures for the current mission
- Defining methods for the evolution of the information environment over time (i.e., supporting comparability of current missions with future missions)

A. Mission Scenarios

To support these assessment concerns, two types of scenarios were employed. The first type engaged soldiers in realistic, albeit short, missions, conducted at a Military Operations in Urban Terrain (MOUT) site, where the ASSIST technologies were used to "shadow" soldiers as they performed their missions, and an S2 officer conducted debriefings post-mission. The second type of test used ASSIST data collected from prior "missions" to assess the ASSIST contribution to another aspect of S2 responsibilities, specifically, data-gathering for a strategic product (actual production was not the focus here).

To date five scenarios have been developed and employed in ASSIST evaluations. Four of those scenarios involved data collection on the MOUT site, while the fifth used previously collected data. The scenarios are as follows:

- presence patrol with deliberate search (2)

- presence patrol leading to a cordon and search
- presence patrol and improvised explosive device (IED) site reconnaissance
- assessment of local situation with respect to an upcoming election

B. Mission Environments

To further facilitate obtaining an operationally-relevant test environment, the in-play area of the MOUT site was set-up with objects, persons, and background sounds, whose placement, behavior and occurrences were scripted. The purpose of this was to provide an environment that would exercise the different ASSIST systems' capabilities as they detect, identify and/or capture various types of information. The IET included all of these elements in the vignettes: foreign language speech, Arabic text, shots fired, vehicular sounds, soldier states, soldier locations (both inside and outside of buildings), objects of interest including vehicles, buildings, people, etc. In contrast, the soldiers' actions were not scripted as they moved through each exercise, with the intent of having the soldiers act according to their training and experience.

C. Vignette Vetting

The vignettes (each scenario with its supporting environment) were developed with the intent of exercising the ASSIST technologies within their operating constraints, while maintaining operationally relevant procedures in execution of those scenarios. Various Subject Matter Experts (SMEs) were consulted to ensure that the scenarios would be as accurate and realistic as possible while still allowing for maximum opportunities for system data capture and assessment. The IET consulted SMEs who provided the following perspectives, all with foreign deployment experience: commander, S2 (intelligence officer), platoon and squad leaders.

D. Details for an Example Vignette

Space constraints do not allow a detailed enumeration of all the particulars for each vignette. However, to provide some insight into the various aspects of a vignette, an overview is provided for one of the vignettes which mimicked a presence patrol.

The presence patrol included leaving a forward operating base (FOB) to patrol a local village, make the military presence known, and collect intelligence on the village and/or villagers before returning to the FOB. In this vignette, the soldiers were instructed via their pre-mission briefing, to conduct a presence patrol in the market area of the village and then conduct a deliberate search of the factory area.

During the vignette, the following activities occurred in the mission environment:

- the market area was crowded with shoppers
- a group of locals was engaged in a soccer match in the open space
- two mechanics worked on a car at the auto shop

- a group of factory workers finished eating lunch at a café and returned to work loading boxes on a truck
- two electricians strung wires around the village
- one insurgent covertly monitored the activities of the squad of soldiers
- foreign aid workers attended to an ill villager at the clinic
- a delivery man delivered packages
- a 6-vehicle convoy traveled by the village area
- a local villager rode around the area on his bicycle taking interest in all the activities
- Iraqi music was played at various locations throughout the village and an Arabic documentary movie was shown at the village café

The environment was envisioned to reflect a typical village setting in an area where U.S. forces were providing stability and reconstruction support. The environment included:

- Population groups, including simulated local villagers and shopkeepers, outside businesspersons and workers, foreign aid workers and contractors, soldiers, and insurgents. Actors were assigned to groups and given specific roles, e.g., "you are a local villager who operates a small clothing shop", "you live in another town but were hired by a cousin to operate an auto repair shop in this village". Actors were instructed to maintain an appropriate attitude relative to other groups, e.g., "you are friendly with everyone", "you like other villagers but dislike all the outsiders, foreigners, and soldiers that have come into your village", etc. Since language identification was a specific ASSIST capability, great care was taken to simulate a (near-)realistic language environment. Actors that were able to speak a foreign language were assigned to an appropriate group, e.g., three German speakers were assigned as foreign aid volunteers who worked at a health clinic. Those actors who did not speak a foreign language, were taught several phrases in Arabic, e.g., "I don't speak English", and assigned to roles that required minimal speech, for example, "you are a playing soccer with a group of friends". All actors were instructed to speak in a manner consistent with their role.
- Terrain, consisting of a market area with several shops, a factory building, a clinic, an auto repair shop, an open space used as a forum for gatherings and/or make-shift athletic field, and a construction zone. The terrain was populated with objects, signage, vehicles, and other "set dressing" objects. For example, a vehicle was placed on ramps next to the auto shop, with tires and car batteries stacked outside the shop.

IV. TESTING PROCEDURES

This section describes the test procedures for assessing the utility provided by the ASSIST system in infantry mission reporting and S2/intelligence operations, as well as the supporting activities used to facilitate a successful test.

A. Test preparation

In addition to defining the scenarios and populating the environment with objects and actors, further preparations were required prior to the start of vignette tests. The IET took steps to familiarize test participants with the test procedures prior to participation in these tests, ensure data capture for systems with immature user interfaces, and to capture of “ground truth” of what happened during each mission. These preparations are described in the following subsections.

1) *Test procedures familiarization*: Due to time and resource constraints during each week of testing, it was impractical to schedule a “practice” vignette test, as these tests ran 7-8 hours start to finish. In an ideal world, the IET would have scheduled a practice vignette so that all participants would have had direct experience to inform their understanding of the sequence of events, along with their roles and responsibilities during each aspect of a vignette test. To avoid having the first vignette degrade into a practice run, the IET took steps to prepare the test participants. Prior to the first vignette test, several activities were conducted that were intended to familiarize participants with various aspects of the test. They included:

- Research teams were given the mission report (MR) template. This provided the teams with an opportunity to see what types of information the squad would be asked to provide at the completion of each mission.
- Research teams were briefed on the general flow of the testing procedures, with the time allowances for each segment.
- The S2, Squad and Fire Team Leaders were briefed on the testing procedures.
- Squad and Fire Team Leaders were briefed on the mission report template, e.g., what information they would be expected to provide in their mission report.
- Research teams briefed the soldiers on the capabilities and information their systems could be expected to provide to augment after-mission reporting.
- Researcher-soldier shadow assignments were made and each pair was given training and practice time on 1) how the soldier would move and give direction to the ‘shadow’ and 2) how the researcher should ‘shadow’ the soldier – see next subsection, IV.A.2.

2) *Shadowing*: The research teams prepared “soldier-worn” systems, allowing each system to capture data from the soldier’s perspective. Some of these systems did not require any user interaction, but some did. To reduce data capture failure due to inadequate operator training for these prototype systems and possible system failure due to soldier-user of unhardened systems, a soldier-researcher shadowing tactic was employed to provide the best opportunity for the systems to gather their data during the mission. This procedure called for a researcher to “shadow”, i.e., follow closely, their assigned soldier throughout each vignette. Soldiers directed their shadow’s attention to pertinent data elements, allowing the researcher “shadow” to operate any manually-activated

capture features of the system. Use of this tactic reduced the probability of user error contributing to poor data capture, and reduced the time and expense for the researchers to provide good user interfaces as well as the need to harden their systems for soldier use.

The IET carefully developed the shadow assignments of the systems to soldiers within the fire teams to ensure that systems would be afforded as much opportunity as possible for exposure to activities, events, and/or objects that their systems were designed to detect. The assignments were designed to allow for maximum opportunities for the systems to collect data while maintaining the integrity of the vignette scenario in an operational setting. As mentioned in the previous section, a training session was provided to work on the mechanics of shadowing for each soldier-shadow pair prior to the vignette tests. Additionally, each pairing assignment was maintained for the entire week of testing.

3) *Ground-truth capture*: The IET captured “ground truth” data of what happened during each vignette mission. This data was captured to document the actual events during each mission, with time information, in the event that any questions were to arise regarding what actually happened or data capture opportunities each research team had during each vignette exercise. Multiple methods to record the ground truth were employed.

On the MOUT site, ground truth data was gathered in the following ways:

- Observers (IET) provided targeted data collection on specific events such as time of gunfire or explosions, convoy passage, a fire team entering a building or interacting with villagers. These observers were trained before each test week on their individual collection responsibilities.
- Video and audio capture of the mission from each fire team’s perspective.
- Elevated-perspective video capture of the squad’s movement through the MOUT site and inside buildings.
- Maps and still images of the environment that captured object and person placement at the start of each vignette

During the mission-reporting sessions following each mission, data was collected in the following ways:

- Audio recordings of the discussion were collected throughout each mission reporting session.
- An IET observer performed targeted data collection including noting information needs, soldier reactions during discussion and the semi-structured interviews.
- An audio recording synchronized with screen capture was collected for each research team’s interactions with the soldiers.
- All information needs identified by the soldiers were recorded by IET members.
- Each soldier was asked to rate each research team’s contribution with respect to each information need identified, both during production of the “naked”

mission report and while interacting with the research teams to review the mission information.

- Observer notes from the semi-structured interviews.
- Ontology data: Members of the IET noted which ASSIST data elements appeared to address the soldiers' information needs. These data elements were then tagged for use with the ontology.

B. ASSIST Utility in Infantry Squad Operations

For the vignettes in which a mission report was completed, the following procedures were used:

- A “simulated squad” of soldiers, comprised of two fire teams, with researcher ‘shadows’, ran through an operationally-relevant scenario on the MOUT site. The squad leader was provided with a pre-mission briefing. The squad leader was instructed to conduct the mission in the manner he deemed most appropriate.
- Upon completing the mission, the squad produced a mission report.
- Soldiers were asked to identify their information needs with respect to producing their MR, e.g., information they would have preferred to include in their MR but did not recall. ASSIST research team members were permitted to observe the soldiers identify their information needs.
- Each research team shared its processed data with the squad. Each soldier was asked to rate the importance of each information need and how well each ASSIST technology addressed each need. Additionally, the soldiers were encouraged to ask questions of the researchers to explore if and how the ASSIST systems produced data that might meet their previously-identified information needs, as well as, any new information needs that were uncovered during the ASSIST information reviews, e.g., newly-identified things that the soldiers would include in their reports.
- The soldiers participated in a semi-structured interview to get at more overall impressions from the exercise and ASSIST systems. The interview facilitator focused discussion on assessing if and how the mission report produced by the squad would be different if the soldiers had been given access to ASSIST system functionality.

C. ASSIST Utility in Intelligence Operations (S2 Level)

Additionally, an S2 (intelligence officer) evaluated ASSIST systems using the following procedures for vignettes in which a mission occurred:

- Following the mission, the S2 was provided with the pre-mission briefing (as appropriate) and the mission report produced by the squad. (Note: the S2 was not allowed to observe the actual mission.)
- The S2 was asked to identify information needs, e.g., information that would improve situation awareness, information about critical events, individuals, or situations, etc.

- The S2 interviewed a member of the squad¹. During this interview, the S2 was encouraged to discuss his information needs with the fire team leader. The S2 was asked to rate the importance of each information need and how well his interview with a soldier addressed each need.
- The S2 met with representatives of the research teams, and shared his information needs with the researchers.
- Each research team shared its processed data with the S2. The S2 was encouraged to discuss his information needs with the research teams and attempt to probe the ASSIST systems for relevant information. The S2 was asked to rate the importance of each information need and how well the ASSIST system addressed each need.

For vignettes in which a mission did not occur, rather the vignette revolved around an S2 tasking where the S2 is to use previously collected data, the following test procedure was used:

- The S2 was asked to give a description of his understanding (what he knew) regarding the content of the tasking.
- The S2 was asked to identify information needs, e.g., information that would improve situation awareness, information about critical events, individuals, or situations, and so on with respect to his tasking.
- The S2 met with representatives of each research team to address the identified information needs. This was a time-constrained review, appropriately 10 minutes were allowed to each team. Indexed methods of information retrieval were stressed as this is expected to be the preferred retrieval strategy in future use.
- The S2 was asked to rate the importance of each information need and how well the ASSIST system addressed each need.
- After having interacted with the ASSIST system, the S2 was asked to provide an updated description of his understanding regarding the content of the tasking.

V. ASSESSMENTS

A. Metrics

The vignettes were structured to allow the IET to assess the utility of the ASSIST technologies in enhancing operational effectiveness. Utility is assessed using the following categories: effectiveness, efficiency, and user satisfaction. More targeted assessments were then made within those categories, as noted below.

Effectiveness

- What information do infantry soldiers want and/or need after completing a mission in the field?
- How well are information needs met, both from the soldier perspective and the S2 perspective?

¹ In this case a fire team leader was used, as the squad leader was participating in the review of ASSIST contributions for the squad's information needs that was occurring simultaneously.

- What were the ASSIST contributions to mission reporting?
- What is the impact on situational awareness (after mission)?
- What are the users' perceptions of the impact ASSIST technologies will have on mission reporting (content & process)?
- What are the users' perceptions of the impact ASSIST technologies will have on soldier performance in the field?

Efficiency

- Was the post-processed data available when required?
- What are the users' perceptions of the impact ASSIST technologies will have on the time taken to produce mission reports?

User satisfaction

- How do ASSIST technologies impact information confidence levels?
- Overall
- User interface² and capability comments

Since the systems being evaluated under Task 2 of the ASSIST program are prototypes, several of the assessments deserve qualification. Some of the metrics above are meant to serve as guides for future development rather than attempting to inappropriately assess these prototype systems. For example, under the efficiency category, post-processing time was identified as an important metric. However, it was not timed how long post-processing took because these are prototype systems resulting in the researchers still exploring which algorithms produce the most effective results rather than having spent time optimizing a specific algorithm to run very efficiently. We used a much coarser measure of efficiency in this setting of whether the data was available when needed. This, more operational metric, is of course, situation-specific. However, we felt our approach identified important metrics while not inappropriately attempting to assess them.

B. Metrics-based Assessments

Use of the procedures outlined in the previous section provided the IET with quantitative and qualitative data that informed assessments of utility and opportunities to enhance utility for the ASSIST systems. These assessments were informed by questionnaire ratings, comments, and semi-structured interview discussions with the soldiers and S2. While actual results can not be reported here, the types of assessments made can. Assessments were made regarding product and process in the following areas:

1) Mission report content (product)

- Mission report content, soldier perspective – Assessments were made in the areas of effectiveness, efficiency and user satisfaction with respect to how

ASSIST systems might change quality and content detail in mission reports. Additionally, opportunities for value enhancement were identified.

- Mission report formatting, soldier perspective – Assessments were made in the areas of effectiveness, efficiency and user satisfaction with respect to how ASSIST might change mission report formatting, e.g., automatic generation of an after-mission report with annotated imagery included for “events”.
- Mission characterization, soldier perspective – Assessments were made regarding the soldiers' perception of the ASSIST systems impact on their observations about “what happened” on these missions. For example, how often did the soldiers modify the list of objects, events, and activities identified in their mission reports following very detailed reviews of the data provided by the ASSIST systems.
- Mission report content, S2 perspective – Assessments were made in the areas of effectiveness, efficiency and user satisfaction with respect to ASSIST systems might change the quality and content detail during the debriefing conducted by the S2. Having collected the S2's ratings of how well his information needs were met when he debriefed the fire team leader, the IET was able to compare these ratings with how well the ASSIST systems met the S2's information needs. Additionally, information confidence levels were assessed, and further opportunities to enhance value for an S2 were often identified.

2) Mission reporting (process)

- Mission reporting process, soldier and S2 perspectives – Assessments were made in the areas of effectiveness, efficiency and user satisfaction for both perspectives: the soldiers and the S2, with respect the predicted impact ASSIST will have on the mission reporting process, as well as opportunities to enhance utility.
- Intelligence analysis, S2 perspective – Assessments were made in the areas of effectiveness, efficiency and user satisfaction for the S2's perspective with regard to using ASSIST data for intelligence analysis.
- Pre-mission planning, soldier and S2 perspectives – Assessments were made in the areas of effectiveness, efficiency and user satisfaction for both the perspectives: the soldiers and the S2 with respect to the use of ASSIST technologies in planning for future missions.

It should be noted that, due to the sample size of soldier subjects in these evaluations, that the assessments drawn were not presented as being necessarily representative of soldiers in other infantry squads and environments and it would be inappropriate to generalize these responses to a great extent. That qualification given, the assessments made did provide formative feedback to the developers.

² Because the user interface was not the target of these evaluations, soldier comments on aspects of the user interfaces were reported only as a feedback to researchers to guide future development of the applicable systems.

VI. LESSONS LEARNED AND CONCLUSIONS

Formative evaluations for field trials of ASSIST technologies were designed and executed in this project. Challenges arose, some anticipated, some not, during this process. Provided below are some of those challenges and experiences.

- It was relatively difficult to obtain soldier participation for this study. Further, most of our SMEs recommended that we retain soldiers who were part of a cohesive squad with recent deployment experience.
- Due to time and resource constraints and the immaturity and lack of integration of the various ASSIST systems, we did not construct an evaluation that produced mission reports with and without ASSIST contributions. Therefore, no direct comparison of mission reports completed with and without the ASSIST technologies was possible. This might be possible in future evaluations.
- The soldier-shadowing tactic proved a useful device for reducing the chance of data capture failure and or system damage with these prototypical systems. We expect the effectiveness of this tactic to diminish as the systems mature and operational scenarios under which these systems are exercised become more intense and the 'shadow' has more difficulty following his assigned soldier.
- The vignette tests were designed to evaluate operational value as perceived by the war fighter rather than technical performance, therefore some soldiers answered questions while considering the *concept* of the technology not necessarily whether the technology worked or not, which clouded some of the data.
- The S2 took much longer to review data than the squad. In retrospect, this is because the S2 was attempting to put together pieces of information to understand relationships and societal dynamics as opposed to the squad's more straight-forward task of completing a report describing their mission and related events. Since we did not fully anticipate the S2 time requirements, we did not schedule sufficient time for the S2 to fully explore all the possible contributions the ASSIST data might have been able to make towards filling his information needs. As a result of the time constraints, the S2 employed a strategy typical to analysts in time-pressure situations of exploring topics that would yield the greatest increase in situational awareness using data sources that seemed to have the most potential for return.
- Following the first evaluation period in November 2005, SME feedback recommended increasing the environmental complexity significantly. The IET took steps to achieve greater environmental complexity for the May 2006 evaluation in the following areas: increased number of people in the environment with more realistic and intertwined relationships, more visual

and audio clutter, more vehicles and movement, and a more developed storyline. While this did increase the realism in the environment in which the ASSIST systems were exercised, it does make comparisons of system performance difficult to assess between the two assessment periods.

In summary, the NIST IET designed and executed formative evaluations to assess utility for ASSIST technologies in mission reporting. This paper presents the design considerations, the field-based evaluation methods used, types of assessments that were drawn, and lessons learned from the evaluations perform to-date.

ACKNOWLEDGEMENTS

This work was supported by DARPA.

Additionally the author would like to thank the following:

- Aberdeen Testing Center staff
- The soldiers who participated as subjects
- Those who contributed their subject matter expertise, especially with respect to military operational and tactical relevance.
- Other members of the IET, as they spent many hours procuring props, setting up equipment and environments, ensuring data capture, acting as observers, and the countless other tasks that contributed to the success of these evaluations.

DISCLAIMER

Certain commercial software and tools are identified in this paper in order to explain the work performed. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

REFERENCES

- [1] DARPA ASSIST, "Advanced Soldier Sensor Information System and Technology (ASSIST) Proposer Information Pamphlet," http://www.darpa.mil/ipto/solicitations/open/04-38_PIP.htm, 2006. (accessed 29 June 2006).
- [2] J. S. Dumas and J. C. Redish, *A Practical Guide to Usability Testing*. Intellect Publishing Co. Portland, OR USA. 1999.
- [3] G. Grinstein, A. Kobsa, C. Plaisant, B. Shneiderman, and J. T. Stasko, "Which comes first, usability or utility?" Proceedings of the IEEE Visualization 2003 Conference. Pp 11-12.
- [4] B. E. John, Evaluating usability evaluation techniques. *ACM Computing Surveys* 28(4es). 1996.
- [5] C. Schlenoff, "Overview of the First Advanced Technology Evaluations for ASSIST", Proceedings of Performance Metrics for Intelligent Systems (PerMIS) 2006, IEEE Press, Gaithersburg, Maryland, USA, August 2006.
- [6] M. Theofanos, and W. Quesenbery, Towards the Design of Effective Formative Test Reports. *Journal of Usability Studies* 1:27-45. 2005.
- [7] B. Weiss, M. Shneier, and A. Virts, "Technology Evaluations and Performance Metrics for Soldier-Worn Sensor Systems for ASSIST", Proceedings of Performance Metrics for Intelligent Systems (PerMIS) 2006, IEEE Press, Gaithersburg, Maryland, USA, August 2006.

Using an Ontology to Support Evaluation of Soldier-Worn Sensor Systems for ASSIST

Randolph Washington
DCS Corporation
1330 Braddock Pl
Alexandria, VA USA
rwashing@dcscorp.com

Christopher Manteuffel
DCS Corporation
1330 Braddock Pl
Alexandria, VA USA
cmanteuffel@dcscorp.com

Christopher White
DCS Corporation
1330 Braddock Pl
Alexandria, VA USA
cwhite@dcscorp.com

Abstract--ASSIST (Advanced Soldier Sensor Information Systems Technology) is a DARPA-funded effort whose goal is to exploit soldier-worn sensors to augment the soldier's recall and reporting capability to enhance situation understanding. An ontology is a data model that represents a domain and is used to reason about the concepts in that domain and the relations between them. It is a mechanism that enables a community to share a common conceptual model of a domain to facilitate communication of knowledge in that domain. This paper provides an overview of the development and use of an ASSIST ontology to support the evaluation of ASSIST technologies.

Keywords: *DARPA, ASSIST, Ontology, evaluation methodology*

I. INTRODUCTION

The Advanced Soldier Sensor Information Systems and Technology (ASSIST) program is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program. The objective of the ASSIST program is to exploit soldier-worn sensors to augment a soldier's recall and reporting capability to enhance situational understanding in military operations in urban terrain (MOUT) environments. The program is split into two tasks:

- Task 1, named Baseline System Development, stresses active information capture and voice annotations exploitation. The resulting products from Task 1 will be prototype wearable capture units and the supporting operational software for processing, logging and retrieval.
- Task 2, named Advanced Technology Research, stresses passive collection and automated activity/object recognition. The results from this task will be the algorithms, software, and tools that will undergo system integration in later phases of the program.

The National Institute of Standards and Technology (NIST) Intelligent Systems Division (ISD), along with NIST's subcontractors (Aptima and DCS Corporation), are funded to serve as the Independent Evaluation Team (IET) for Task 2. As the IET for Task 2, NIST is responsible for:

- Understanding the Task 2 contractor technologies
- Determining an approach for testing their technologies
- Identifying a MOUT site to evaluate the technologies
- Devising and executing the tests
- Analyzing the data and documenting the outcome

The following three metrics are the focus for the IET's Task 2 evaluation:

- 1) The accuracy of object/event/activity identification and labeling
- 2) The system's ability to improve its classification performance through learning
- 3) The utility of the system in enhancing operational effectiveness

To evaluate the ASSIST systems of the three Task 2 research teams, the IET developed a two-part test methodology to produce these metrics. Metrics 1 and 2 were evaluated through "elemental tests", and metric 3 was evaluated through "vignette tests". Elemental tests were designed to measure the progressive development of the ASSIST system's technical capabilities; and vignette tests were designed to predict the impact these technologies will have on warfighter's performance in a variety of missions and job functions. In specifying the detailed procedures for each elemental and vignette test, the IET attempted to define evaluation strategies that would provide a reasonable level of difficulty for system and soldier performance at both the 6-month

(November 2005) and 12-month (May 2006) evaluations.

This paper describes how an ontology is being used to support the evaluation of ASSIST technologies. Section II of this paper defines an ontology and identifies the objectives of the ASSIST ontology. Section III provides an overview of DCS' approach to developing the ASSIST Ontology. Section IV describes the use of the ASSIST Ontology during the 6-month and 12-month technology evaluations. Section V concludes the paper and identifies recommended future work in this area.

II. ASSIST ONTOLOGY OBJECTIVES

In computer science, an ontology is a data model that represents a domain and is used to reason about the concepts in that domain and the relations between them. It is a mechanism that enables a community to share a common conceptual model of a domain to facilitate communication of knowledge in that domain. A good ontology focuses on defining the meaning of concepts through its relationship with other concepts rather than merely enumerating concepts. An ontology together with a set of individual instances of the concepts constitutes a knowledgebase. Since a knowledgebase is strictly defined by the ontology, a knowledgebase can be easily understood by consumers of the knowledge. Therefore, ontologies are particularly useful in the exchange of knowledge between computer systems. The development of languages to build ontologies and populate a knowledgebase has recently been a very active area for standards organizations such as World Wide Web Consortium [1].

NIST has been active in the development and application of ontologies and had previously collaborated with DCS on the development of an ontology for defining the behavior of intelligent ground combat vehicle systems for the U.S. Army Tank Automotive Research Development, and Engineering Center. As the ASSIST program was focused on the automated acquisition and distribution of knowledge within the dismounted infantry domain, NIST believed that the development of an ontology for the ASSIST program would be beneficial and contracted with DCS to support the effort. The objectives of DCS' ASSIST Ontology effort were to:

1. Elicit and document the specific knowledge requirements in the ASSIST Ontology
2. Develop the ASSIST Ontology using formal knowledge representation techniques and standardized tools and representations

3. Apply the ASSIST ontology for evaluating the results of the developmental technology.

III. ASSIST ONTOLOGY DEVELOPMENT

DCS' approach to developing the ASSIST Ontology was to A) conduct an analysis of the information requirements in the ASSIST domain to identify the concepts and relationships to be included in the ontology; B) survey available ontology languages and tools and select the ones most appropriate for the ASSIST Ontology; and C) design an ontology that efficiently captures the key ASSIST Domain concepts for the purpose of performance evaluation of the ASSIST systems and implement the Ontology in the selected language. Each of these steps is described in the following paragraphs.

A. Information Requirements Analysis

In order to evaluate the information requirements for the ASSIST Ontology, DCS participated in discussion on potential applications of the ASSIST technologies NIST conducted with soldiers recently returned from Iraq. The focus of these discussions was identifying the kinds of information dismounted infantry soldiers were expected to recall at the end of a patrol through potentially hostile urban environments and identifying how this information was conveyed to other soldiers to support the development of an intelligence assessment of the area and to provide beneficial information for subsequent patrols. It was assumed that reports made by soldiers supported by ASSIST technology would be aggregated with reports from other soldiers on the same patrol as well as other patrols over periods of time. A key part of the analysis was identifying information that, when combined with other reports, would support the identifications of patterns of hostile or friendly activities.

In the course of these discussions there was a general assumption of the kinds of technologies to be applied to capture the information but no presumption of the format in which it was to be conveyed. The soldiers discussed how information is currently collected by soldiers (eyes, ears, cameras, GPS) and conveyed to squad/platoon leaders who in turn convey the information to battalion intelligence officers. The soldiers provided lists of concepts (i.e., objects, actions (both of the soldiers and external objects), and relations) they felt would be significant to be conveyed to the unit's intelligence officer. At the same time the soldiers were working with NIST to define evaluation vignettes for the ASSIST technologies. These vignettes included most of the

significant objects and actions identified by the soldiers. DCS analyzed the soldier's lists and the evaluation vignettes and generated a list of concepts that needed to be contained in the ASSIST ontology to enable a full description of the environment and activities of the vignettes from the ASSIST-wearing soldier's perspective.

B. Ontology Language Format and Tools Selection

DCS evaluated several current ontology development languages including openyc [2], KIF [3], UML [4], and Web Ontology Language (OWL) [5] for use in development of the ASSIST Ontology. Key criteria in the evaluation were descriptive ability, ease of human use, ease of computer use, reliance on open standards, and availability of associated open-source environments focused on knowledge capture and retrieval.

OWL, an XML-based markup language for publishing and sharing data using ontologies on the Internet, was selected as the best language for the ASSIST Ontology. OWL is a vocabulary extension of the Resource Description Framework (RDF) and is derived from the DAML+OIL (DARPA Agent Markup Language, Ontology Interchange Language) Web Ontology Language developed by DARPA. The OWL specification is maintained by the World Wide Web Consortium (W3C). OWL provides somewhat intuitive mechanisms for defining classes, properties, property restrictions, and individuals that are more than adequate to support the ASSIST domain. In addition, being based on XML it is simple to translate OWL documents to other formats using eXtensible Stylesheet Language Transformations (XSLT).

A number of open-source OWL editors including (Protégé-OWL, SWOOP, and OIEd) were evaluated for use in development of the ASSIST Ontology. Protégé [6] with the OWL-plugin-in was selected as the most suitable development environment due to the maturity of its ontology-editing GUI, the availability of a wide variety of plug-ins for data inferencing and visualization, and the ease of integrating user applications through Protégé's plug-in framework. Protégé and the OWL plug-ins are developed by Stanford University. DCS took advantage of this plug-in architecture by developing and integrating a custom query tab that supported advanced queries of ASSIST knowledgebases.

C. Ontology Development

Given the long list of important concepts and relationships identified during the information requirements analysis, there are many ways to organize these terms in an ontology. As the ASSIST technologies are primarily intended to aid soldiers observe things on the battlefield, it was decided to organize the ASSIST ontology around the concept of an observation. An **Observation** is a description of one or more related actions or objects, or lack thereof, which is made by the assisted soldier at a specific time. An observation may be supported by one or more multimedia files. An observation may also be related to an event which may be a part of another event.

Fig. 1 illustrates the top level of the ASSIST ontology. ASSIST concepts are shown as black boxes and ASSIST relationships are shown as blue lines between concepts. Asterisks by a blue arrow indicate a possible one-to-many relationship. A negative relationship between an observation and an action or object (i.e., *hasNotObservedObject* or *hasNonAction*) is necessary to report the observed lack of something (i.e., there were no children in the village).

The ontology provides many sub-concepts for **Action** and **Object** that allow the **Observation** to be very specific and allows other objects and data properties to be associated with the observed object. The primary driver for the creation of sub-concepts was the vignette descriptions. During the creation of the knowledgebases representing individual after-action reports by the soldiers generated during the evaluations, new concepts were added to the ASSIST ontology if suitable concepts did not exist to express all of the observations. In this fashion the ontology was iteratively improved as more and more of the after-action reports were described with it.

Fig. 2 shows a breakdown of the resulting major subclasses of **SpatialThing** and Fig. 3 shows the breakdown of the major resulting subclasses of **Action**. Note that **SpatialThing** has spatial relations to itself allowing a knowledgebase to identify the relative locations of objects. In addition, multiple observations of the same object could be correlated by associating a name with the object. Also note that object relationships can be chained to form complex descriptions such as "observed a human being named Joe, who wore a white Dishdasha and carried an AK-47, in front of the building with two floors that was located west of the village marketplace".

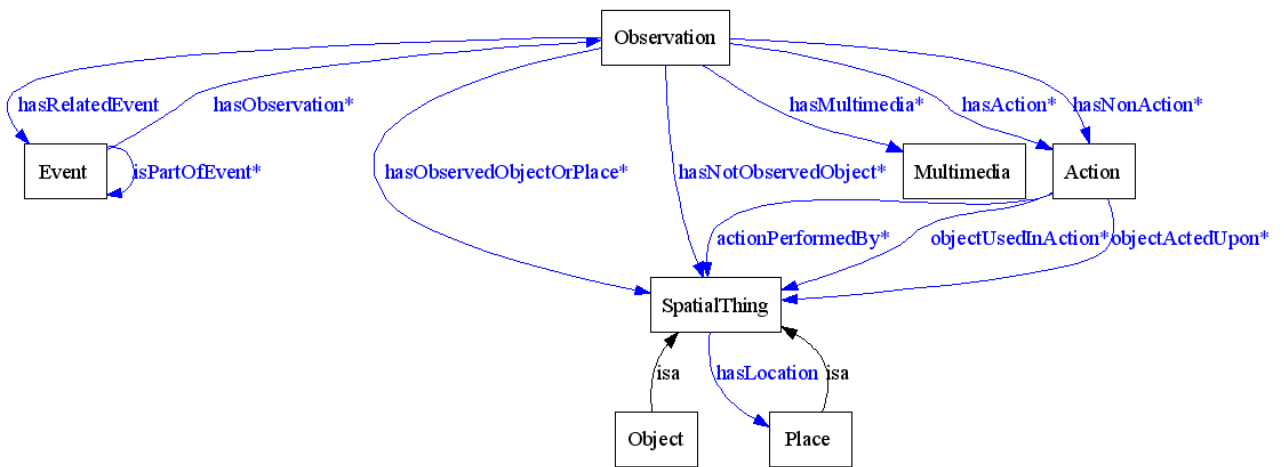


Fig. 1. Ontology Class Diagram

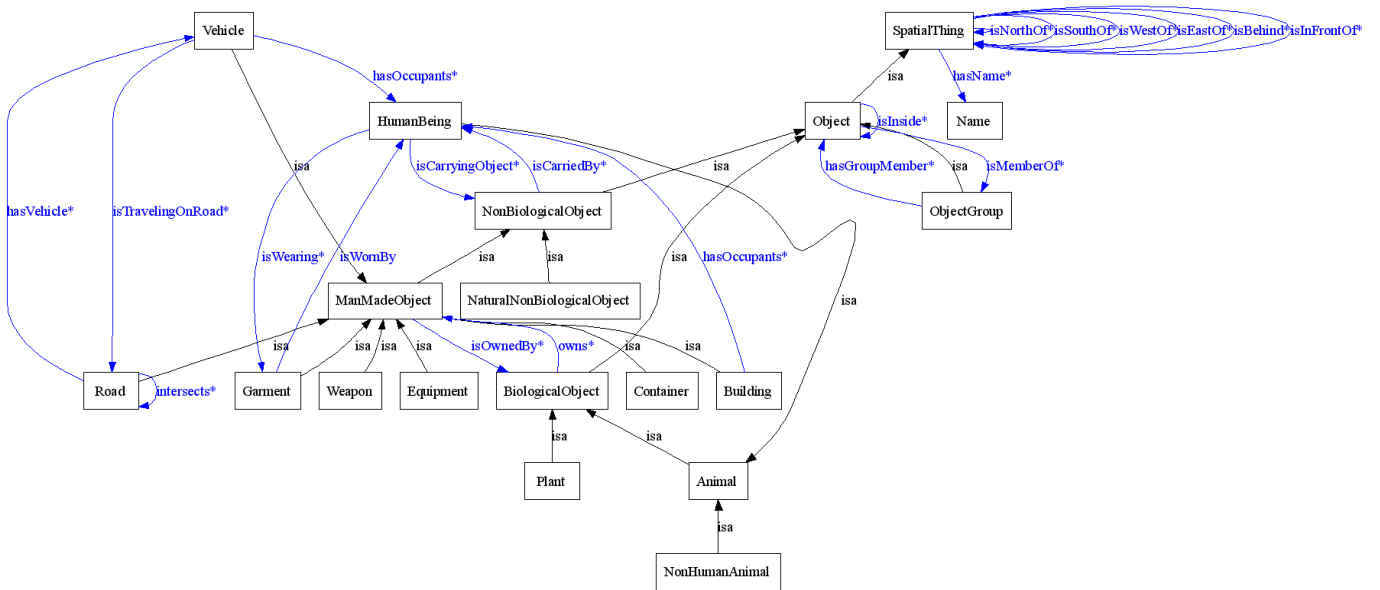


Fig. 2. Spatial Things Class Diagram

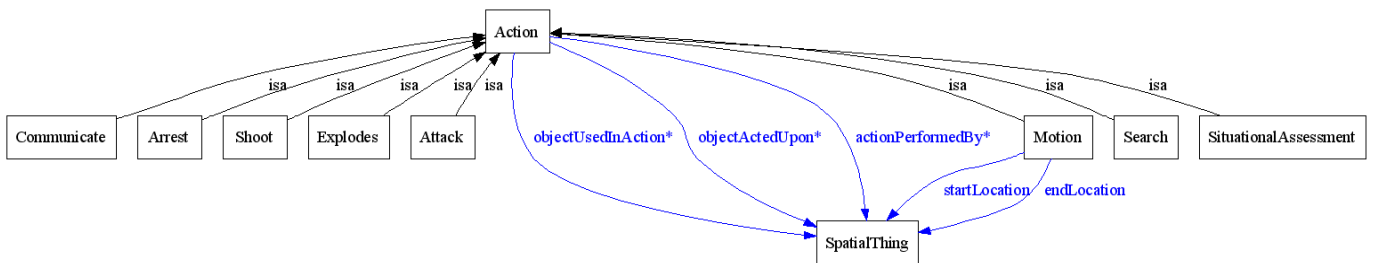


Fig. 3. Action Class Diagram

The ASSIST ontology was generated in the OWL language using Protégé. The resulting XML file can be found on the world-wide-web at www.isontology.org/ISOntology/ASSIST/Assistv10.owl. Knowledgebases reflecting observations made during evaluation vignettes were also created using Protégé.

IV. EVALUATION SUPPORT

The ASSIST ontology was used during the ASSIST technologies evaluation to quantify the amount of tactical information conveyed by soldiers during the vignette tests. This allowed the information content of after-action reports and intelligence officer debriefs generated without the aid of ASSIST technologies to be compared to the information content of the same reports and debriefs generated with the aid of each research teams ASSIST systems.

During the 6-month evaluation, two vignettes based on presence patrols were executed by a squad of two fire teams. After each vignette, each fire team generated an after-action report without access to the information from the ASSIST technologies. Each fire team then generated a list of additional information that they would have liked to have known but could not directly recall. This list of additional information requirements was provided to each ASSIST research team. One after another, each ASSIST research team was then given the opportunity to show the fire teams the information requested using their ASSIST system. DCS observers listened to each fire team generate the unassisted after-action report and recorded each of the observations stated by members of the fire team. The DCS observers then listened to each ASSIST research team as they addressed the fire team's information requests. Observations identified by the research team's ASSIST system that corresponded to the fire team's information requests were recorded by the DCS observers.

After execution of the vignettes, DCS generated ASSIST ontology knowledgebases documenting the unassisted after-action reports from each fire team for each vignette. DCS also generated separate ASSIST ontology knowledgebases documenting the additional observations provided by each research team's ASSIST system for each fire team for each vignette. Each of the 12 resulting knowledgebases was analyzed to quantify the number of observations, actions, observed objects, properties of each observed object, and multimedia files. The data showed that the soldiers were capable of recalling a lot of the militarily important observations from their exercises

and that the user interfaces of the vendors ASSIST systems were capable of detecting many of the militarily significant objects in the vignettes, but were not capable of describing the relationships between the objects, or describing the objects in as much detail as the soldiers were. It was also noted that as the ASSIST technology observations were recalled based on soldier inputs, these observations tended to include multimedia files (images and audio clips) of objects and actions already observed by the fire teams for the purpose of confirming the soldier's observations.

During the twelve-month evaluation, two vignettes based on presence patrols were again executed by a squad of two fire teams. After each vignette the squad generated an after-action report without access to the information from the ASSIST technologies. As the soldiers were provided two digital cameras which were not considered as part of the ASSIST technology during this evaluation, the unassisted after-action reports also included selected images from the digital cameras. After the unassisted after-action report was generated for each vignette it was given to a soldier acting as the unit intelligence officer. After the intelligence officer reviewed the report he interacted with one soldier from the squad to clarify information contained in the after-action report. The intelligence officer then generated a list of additional information that he would have liked to have known but could not get directly from the after-action report or through the clarifications of the soldier. This list of additional information requirements was provided to each ASSIST research team. One after another, each ASSIST research team was then given the opportunity to show the intelligence officer the information requested using their ASSIST system. DCS observers listened to the squad generate the unassisted after-action report for each vignette and recorded each of the observations stated by members of the squad. The DCS observers then listened to each ASSIST research team as they addressed the intelligence officer's information requests. Observations identified by the research teams that corresponded to the intelligence officer's information requests were recorded by the DCS observers.

After execution of the vignettes, DCS generated ASSIST ontology knowledgebases documenting the unassisted squad after-action reports for each vignette. DCS also generated separate ASSIST Ontology knowledgebases documenting the additional observations provided by each ASSIST research team for the intelligence officer for each vignette. Each of the eight resulting knowledgebases

was analyzed to quantify the number of observations, action, observed objects, properties of each observed object, and multimedia files. The data again showed that the soldiers were capable of recalling a lot of the militarily important details from their exercises and, given hand held digital cameras, had captured still images of many of the significant observed objects. The ASSIST systems provided by each research team were able to provide varying amounts of supplemental information in response to the intelligence officer requests for additional information. While each of these ASSIST systems had collected large quantities of data, the actual amount of information conveyed to the intelligence officer was limited by the significant amount of information contained in the squad's after-action report and the limited ability to search for information regarding specific objects and relationships between those objects with the ASSIST systems.

DCS also generated ground-truth ASSIST ontology knowledgebases to support the evaluations. These knowledgebases documented the test environment during a vignette rather than merely what the soldiers observed. This allowed the independent evaluation team to determine how much progress had been made from the 6-month to the 12 month evaluations in terms of environmental complexity and the ASSIST systems ability to characterize that complexity.

The number of objects of various categories placed in the MOUT site for the 12-month evaluation (May 2006) and 6-month evaluation (November 2005) are shown in Table 1. For comparison purposes Table 1 also lists the number of each category of objects expected to be found, on the average, for a site of the same area as the test site in the Iraq Governorates of Babylon and Baghdad based on a UN Development Program report ("Iraq Living Conditions Survey 2004"). As can be seen in the table, the total number of objects used in Vignette 1 in the May 2006 evaluation was approximately four times the number of objects used in the November 2005 evaluation. Also, the total number of objects used in Vignette 1 in the May 2006 evaluation was approximately 18 times the number of objects expected, on average, in an equivalent area in the rural Babylon Governorate and one-half the number of objects expected, on average, in an equivalent area in the very urban Baghdad Governorate.

	May 2006	November 2005	Babylon	Baghdad
People	50	14	3.45	108.41
Vehicles	23	8	0.11	5.49
Radio	3	0	0.29	12.69
Television	1	0	0.41	17.23
Video				
Player	1	0	0.12	7.20
Bicycle	1	0	0.04	1.33
Firearms	2	0	0.07	4.92
Computers	1	0	0.01	1.70
Total	81	22	4.49	158.97

Table 1 - Environmental Complexity

V. CONCLUSIONS

The ASSIST Ontology proved to be capable of supporting the manual generation of knowledgebases that represented the information contained in soldier after-action reports and the additional relevant information provided for each vignette by each of the research team's ASSIST systems. It also proved capable of manual generation of knowledgebases characterizing the test site to support analysis of environmental complexity during each evaluation and comparison to statistics from Iraq. In both cases it proved to be a relatively simple matter to extend the ASSIST ontology as needed by adding new object and action concepts when the appropriate concepts were not found in the original ontology.

While the generation of the unassisted after-action report knowledgebases was relatively straight forward because the soldiers articulated each observation, the generation of the knowledgebases representing the additional information provided by the ASSIST technologies was not as simple because it was not always easy to tell whether the information presented by the research teams was what the soldiers or intelligence officer had asked for or was not relevant. This made these knowledgebases and therefore comparison of the information contribution of each research team's ASSIST systems more subjective than DCS would have preferred.

When originally conceiving the ASSIST ontology, DCS had envisioned that the ASSIST systems would eventually be capable of automatically generating ASSIST ontology knowledgebases from the sensor data captured during a mission. This would allow each system's knowledgebases to be compared

directly to ground truth knowledgebases and allow direct comparison of the accuracy and completeness of the knowledgebases. However, the processing capabilities of each system were limited to recognizing a relatively small set of object classes in still or video imagery and recognizing a relatively small set words or sounds in audio data. This allowed the research teams to tag still or video images with metadata specifying the classes of objects contained in each image (i.e., a person) but did not allow them to uniquely identify those objects (i.e., the person named John Q. Smith from Silver Springs, MD). This inability to differentiate multiple observations of the same instance of an object class from multiple observations of different instances of the same object class prevented them from extracting specific instances of object classes from the sensor data and reporting their location and relationships to other objects over time. Although each still or video image could have been treated as an observation and stored in an ASSIST knowledgebase, this would have led to a huge amount of uncorrelated observations. As an example, a video stream taken while walking past the John Q. Smith on two successive days would have generated hundreds of observations of an object of class person each day rather than two observations of an object of class person both named John Q. Smith”).

Beyond its use for ASSIST technology evaluation, the ASSIST ontology has great potential to facilitate efficient storage and analysis of data from ASSIST technologies. Once the ASISST technologies have matured to the point where they can correlate specific objects over space and time, observations of these individual objects can be efficiently stored in ASISST knowledgebases. The ASSIST users can then annotate the observations of people with additional relationships such as names, family ties, business affiliations, and political affiliations to describe the social networks [7] of the communities observed either verbally as the imagery is taken or via a computer program after the completion of the mission. The ASSIST users can also annotate observations of manmade objects with additional

relationships such as “owned by” and “resides at” to define the relationships between the objects and the social network. These relationships can then form the basis for intuitive, relational queries into the ASSIST knowledgebase.

Relational queries, such as “show me a picture of everybody in the al-Lami family associated with the United Iraqi Alliance who was observed in Mosul on Monday and Tikrit on Tuesday”, offer tremendous advantages to using ontologies for data storage and analysis. Commercially available reasoners exist which are capable of handling queries of almost infinite complexity. Storing information in ontological knowledgebases allows such reasoners to determine objects, relationships and patterns that would otherwise be difficult for humans to identify. Therefore it is recommended that automated object identification and storage within ASSIST knowledgebases as well as efficient approaches for operator annotation be investigated in future phases of the ASSIST program.

ACKNOWLEDGEMENT

This work was supported by the National Institute of Standards and Technology (NIST) Intelligent Systems Division (POC. Craig Schlenoff)

REFERENCES

- [1] Nigel Shadbolt, Tim Berners-Lee and Wendy Hall, The Semantic Web Revisited. IEEE Intelligent Systems 21(3) pp. 96-101, May/June 2006
- [2] “The Cyc Foundation” <http://www.opencyc.org/>
- [3] Geneserth, M.R. and R.E. Fikes (Eds), Knowledge Interchange Format, version 3.0. Computer Science Department, Stanford University, Technical Report Logic-92-1, June 1992.
- [4] S. Ambler, The Object Primer. New York, NY: Cambridge University Press, 2001.
- [5] “OWL Web Ontology Language Overview” <http://www.w3.org/TR/owl-features/>
- [6] N. F. Noy, R. W. Fergerson, M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility. 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000. Also see <http://protege.stanford.edu/>
- [7] “Social Networks” http://en.wikipedia.org/wiki/Social_network

Evaluating Intelligent Systems for Complex Socio-technical Problems: Seeking Wicked Methods

Michael P. Linegang
Aptima, Inc.
1726 M Street NW; Suite 900
Washington, DC, USA
linegang@aptima.com

Jared T. Freeman
Aptima, Inc.
1726 M Street NW; Suite 900
Washington, DC, USA
freeman@aptima.com

Abstract— Many of today’s most advanced government-sponsored systems engineering efforts are developing technology to address complex socio-technical problems. This paper describes some of the challenges associated with traditional field-based evaluation methods for supporting these types of systems engineering efforts, and offers modeling and simulation as a tool to supplement field-evaluation. One current program, the DARPA ASSIST program is examined as a case study of the challenges associated with field evaluation for “wicked” socio-technical problems. We define requirements for a simulation environment to supplement field testing for this program, and describe how cognitive models and information visualization could be used to meet these requirements.

Keywords: *Wicked Problems, Socio-technical Systems, Modeling and Simulation, Cognitive Models, Cultural Models, Visualization.*

I. INTRODUCTION

Traditional systems engineering approaches attempt to define a fixed set of requirements early in the design of new technology. These requirements drive development and are the standard to which products are tested during evaluation. The assumption underlying this approach is that *the problem* will be solved if *the requirements* are met. But many of today’s most advanced government initiatives are grappling with the creation of intelligent systems to function in complex social settings (e.g., see [1]). In these situations, *the problem* is ill-defined and *the requirements* typically evolve with the product; the requirements thus cease to function as control measures. The challenge in these circumstances is to explore the problem space and discover requirements efficiently, rapidly, and early, and to test product concepts in the problem space before they are built, or even prototyped. This paper proposes ways to accomplish this using simulations and models of complex socio-technical systems. These techniques bolster the standard control mechanism for these systems engineering problems.

A. Wicked Problems

Complex, socio-technical systems often pose wicked problems, that is, problems that cannot be understood until they are solved, problems for which there is no one right

solution, and problems that change fundamentally with the imposition of any solution [2]. In these ill-structured settings, the solutions never truly solve the problem, but merely achieve some degree of balance in a complex system of competing forces.

Intelligent systems are being developed to address complex socio-technical problems. This paper references one program that is taking precisely this approach: the Defense Advanced Research Projects Agency (DARPA) Advanced Soldier Sensor Information Systems and Technology (ASSIST) program [1]. Its objective is to exploit soldier-worn sensors to augment a soldier’s recall and reporting capability and enhance situation understanding in military operations in urban terrain (MOUT) environments.

Figure 1 outlines a traditional systems engineering approach applied to a hypothetical wicked problem (shown in boxes with a white background). It also illustrates a new control mechanism, a “simulated wicked problem space”, as a mechanism to address some of the uncertainties associated with defining technology requirements for wicked problems (shown in the box with the gray background).

The hypothesis of this paper is that a simulated world can be used to represent our assumptions about wicked problems, and thus make explicit our understanding of them and of the requirements for addressing them. A simulated world can be a valuable control mechanism in systems engineering programs conducted to handle wicked problems. Specific benefits of this simulation-based assessment as a supplement to field testing would include an ability to assess the complex interactions between new technologies and forces in the problem space that are difficult to represent in a field test environment, to do so across a great range of scenarios, and to develop a deeper understanding of the fundamental dynamics of the problem space. The remainder of this paper describes this “simulated wicked problem space” concept as a supplemental control mechanism for an intelligent systems development effort, using the DARPA ASSIST program as a hypothetical backdrop.

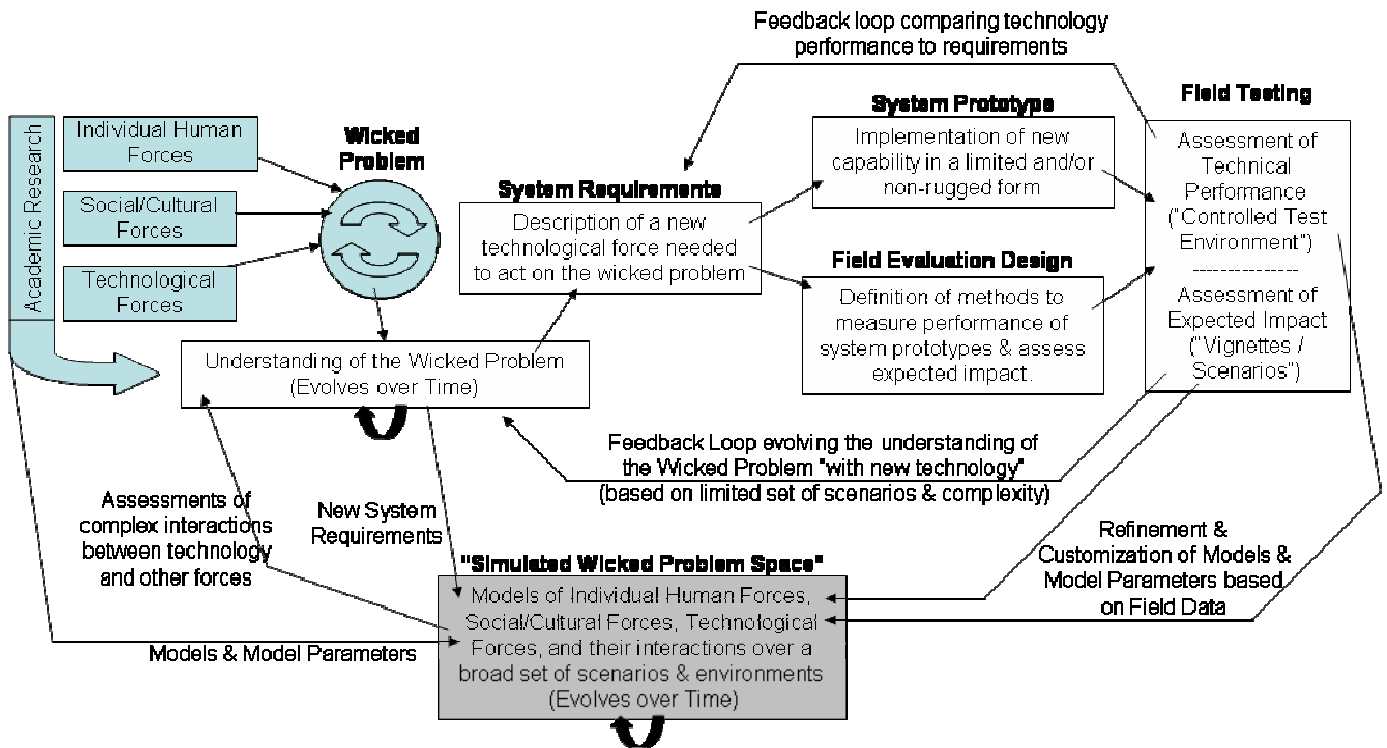


Fig. 1. Traditional systems engineering methods (white boxes) applied to a wicked problem (blue boxes), supplemented by a “simulated wicked problem space” (gray box)

II. DARPA ASSIST PROGRAM

The ASSIST program is a DARPA advanced technology research and development program. The objective of the ASSIST program is to enhance post-mission reports of sightings and events in MOUT operations by using data from soldier-worn sensors to supplement a soldier’s recall. The program is developing two classes of technology:

- The Baseline System: prototype wearable capture units and the supporting operational software for processing, logging and retrieving data. These typically require soldiers to take an active role in recording or capturing relevant data.
- Advanced Technology: passive wearable capture devices, algorithms, software, and tools that automatically collect data and recognize activities, objects, and/or events.

[3] provides additional details about the ASSIST program’s objectives and technologies.

A. ASSIST’s Wicked Problem Space

Enhancing situation understanding of an urban environment is a wicked problem. The urban environment contains a large number, many types, and varying density of entities with many ill-defined but critical relationships. A MOUT team may patrol a city with tens or hundreds of thousands of inhabitants, each allied with family, tribal, religious, political, and economic interests that drive them into cooperation or conflict with American troops. The city streets may be

crowded with cars and trucks, most on routine errands, but some supporting insurgent activities or worse, loaded with explosives. And the buildings house businesses, families, civilian organizations, international support organizations, and sometimes insurgent gangs planning attacks or fleeing patrols.

Figure 2 represents this space abstractly. It illustrates three types of entities: 1) people, represented as circles, 2) vehicles, represented as white squares, and 3) buildings, represented as black squares. Many human, social, and cultural forces are at play in this environment, some externally visible, but many internally held or only manifested through complex behaviors. Figure 2 suggests some form of social or cultural complexity by representing three groups of individuals: Group 1 represented by shades of vertical green stripes, Group 2 represented by shades of red cross-hatching, and Group 3

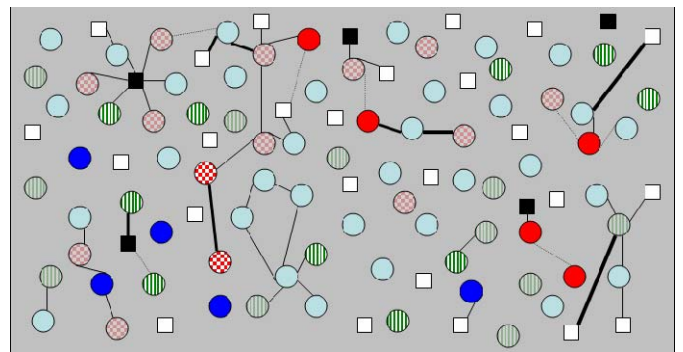


Fig. 2. Representation of entities and relationships in a MOUT environment

represented by shades of solid blue might represent three different political affiliations (and the color saturation in each circle suggests the strength of each individual's affiliation with their group). The environment results in frequent *interactions between entities*, with some interactions being highly overt (e.g. speaking to someone or interacting with an object) and others covert (e.g. passively listening to someone or observing an object). Figure 2 represents a range of interactions in the form of lines linking entities together, with dark, bold lines representing highly overt interactions between entities and lighter lines representing passive or covert interactions. This environment evolves as entities come and go, individual affiliations shift, and groups form and disband. Thus, our understanding of the environment must constantly be updated.

Identifying and categorizing the multiplicity of people, objects, relationships, interactions, events, and affiliations in a complex environment is a daunting task; and monitoring the changes in those variables over time is even more challenging. Further, the act of gathering the information necessary for gaining an understanding of this complex environment is likely to spawn unpredictable changes in the environment. The presence of American troops on the streets influences who hides, the attitudes of those who remain, and their behaviors. Observing the environment may actually **reduce** one's "understanding" of it if the processes used in gathering information significantly perturb the environment. The complexity of the MOUT environment, its dynamism, and the unpredictable effects of observing it make understanding it a wicked problem.

B. Evaluating ASSIST Technologies: Traditional Techniques

There are many potential technological "solutions" to this problem: some better, some worse, none perfect. But the complexity of the environment and the task of understanding it makes quantifying, or even qualifying, the goodness or badness of each solution a difficult challenge. The ASSIST program has completed two spirals through the traditional systems engineering approach (illustrated in the white nodes of Figure 1). We describe it here as a case study of traditional evaluation techniques.

An initial understanding of MOUT operations and the need for better understanding of the environment led to the specification of initial requirements for intelligent systems that are able to recognize specific classes of entities in the urban environment [1]. These requirements were interpreted by several engineering teams, and several prototype intelligent systems were developed [3].

The independent evaluation team (IET) for the ASSIST program developed and applied a two-pronged evaluation approach, effectively addressing some of the wickedness in this problem space [3]. One evaluation effort, called *elemental tests*, assessed levels of achievement for technology performance objectives in controlled environments [4]. The second assessed potential impact of technologies in less controlled environments, specifically, in *vignettes* involving

soldiers interacting with actors on MOUT sets [5].

This two-part approach provided clear, quantitative assessments of the prototype system's capabilities relative to the stated requirements (i.e. the elemental tests assessed the ability of the intelligent systems' to recognize and classify portions of an urban environment at a given point in time). The approach also provided significant qualitative (and some quantitative) assessment of the potential interactions of the new intelligent systems with a limited set of the other forces at work in the problem space (i.e. the vignette tests assessed the effect of incorporating the new technology into a semi-realistic mission scenario).

However, field test environments impose constraints on an evaluation that significantly limit the evaluator's ability to examine effects of new technologies on complex socio-technical interactions; issues that are critical in addressing wicked problems. Field assessment costs and other constraints force evaluators to attempt to "tame" a wicked problem; limiting the level of complexity and viewing it within a controlled setting (relative to the complexity of the actual environment). Even the "more wicked" vignette tests (involving soldiers interacting with actors on sets) require evaluators to manufacture as much "realistic complexity" as possible within the constraints of a budget and an unrealistic field test setting. And budget and timeline constraints typically limit the evaluation to a very small number of mission scenarios, with minimal ability to observe any secondary or tertiary effects of the technologies that might be manifested over longer periods of time. For example, four ~30-40 minute ASSIST vignettes were performed over a period of twelve months (2 vignettes at a 6-month test, 2 vignettes at a 12-month test), providing less than 4 hours of system-collected data from ~9 soldiers. The ASSIST system concept calls for every operational soldier to utilize the system on every patrol; meaning one 18-soldier patrol in a real world mission would produce double the dataset currently available for system evaluation purposes after 12 months. This is a very limited sample from which evaluators must extrapolate conclusions about how the system and the data it produces will affect soldier performance and understanding of the environment.

In spite of evaluators' best efforts to create semi-realistic complexity and determine the realistic effects of technology, field evaluations force evaluators to deal with reduced complexity as a fundamental constraint of the test. This lack of complexity likely conceals critical interaction effects of the new technology with the wicked problem space. [6] cautions that seemingly effective solutions for "tame" versions of a wicked problem can be deceptive, because "the wicked problem simply reasserts itself, perhaps in a different guise ... or, worse, sometimes the tame solution exacerbates the problem." The following sections consider the challenges associated with addressing complexity in field assessment.

III. THE FIELD EVALUATION "LEAP OF FAITH"

Referring to Figure 1, the primary limitation in field

evaluation methods is the limited nature of the feedback about the integration of new technology into its environment. Field evaluations can provide valuable and compelling insights into a limited set of interactions for a new technology. But it is unclear how stable the results will be when the new technology is introduced into the natural, more complex setting, or how those results might evolve over a large number of scenarios. Field evaluation results thus require investors to make a leap of faith, to extrapolate test results from a simple environment to a more complex one. Below, we use the ASSIST program to illustrate the limitations of field testing, and suggest how simulation might enhance and extend the results from ASSIST field testing.

The IET determined that ASSIST technologies displayed significant improvements over the course of multiple tests in their ability to recognize and categorize a variety of entities in the test environment (based on elemental tests that assessed automated recognition capabilities in controlled settings) [4]. The IET also evaluated the potential utility of ASSIST technologies for improving an intelligence officer's ability to understand the environment. These vignette-based tests found that, while intelligence officers needed human soldiers to help them navigate ASSIST data records to find key events, once found, those data provided a greater level of detail than soldier recall alone. ASSIST allowed intelligence officers to assess situations from an objective viewpoint, rather than rely on the soldier's interpretation of sightings [5]. This represents the results from two cycles through the traditional requirements definition, prototype/evaluation development, and evaluation feedback sequence.

These results suggest that ASSIST technologies are effective in classifying a MOUT environment, and that the introduction of ASSIST technologies as a supplement to individual soldier reports could allow an intelligence officer to gain a richer understanding of the environment. But the test environment required a great many assumptions and simplifications to achieve these results. It is unclear what effects from the use of new technology might arise as complexity is reintroduced. The following is a partial list of simplifications imposed by field evaluation constraints in the ASSIST tests:

- *Sparse sampling of potential missions:* ASSIST technologies were evaluated on a small subset of missions. How might their performance or their impact on soldier performance change given other mission objectives, procedures, and op-tempos?
- *Short mission length:* ASSIST was tested using less than four hours of vignettes. How might the intelligence officer's understanding of mission data evolve given longer, more numerous, or more diverse missions? How would the technologies contribute to the intelligence officer's understanding of the entities, relationships, and their affiliations as they evolve in the environment? How might soldier performance change as they develop new tactics, techniques, and procedures for using the ASSIST technologies? How might the dramatic increase in ASSIST system data available after multiple

missions impact the intelligence officer's utilization of the system?

- *Unrealistic environments:* ASSIST technologies were evaluated in missions conducted by soldiers working with relatively low levels of stress. The soldiers were not deprived of sleep or subjected to threats, for example. They were operating in a familiar setting (a military testing facility) during normal working hours. Stressors are expected to change the level of detail of soldier reports to the intelligence officer, the accuracy of those reports, and the level of conflict within and between reports. Will these effects occur in realistic settings? How will intelligence officers respond to them?
- *Absence of counter-measures:* ASSIST technology was operated without the degrading influence of enemy counter-measures. Is ASSIST technology vulnerable to hostile counter-measures? Can intelligence officers discern those effects when they are present?

Vignettes and field tests offer no easy answers for these challenges. An investor must choose between 1) a "leap of faith" that greater and different complexities will not significantly change the results, or 2) a costly investment in additional vignettes and field tests that will evaluate the technology in an environment that includes some of these additional complexities (but still a limited set). Option 1 is a relatively high risk approach since it offers no answers to these questions, but Option 2 is costly and unlikely to reduce the risk by a significant margin since it only adds a few new data points to the investor's portfolio.

An alternative approach is needed; one that can evaluate the technology in a much more complex setting, providing a more complete picture of expected interactions between the new technology and other aspects of the problem-space. We now describe a concept for using modeling and simulation to address some of this challenge.

IV. SIMULATION AS A SUPPLEMENT TO ASSIST FIELD TESTS

Modeling and constructive simulation has been used extensively to supplement experimental testing in domains such as team and organizational design (e.g. [7], [8], [9], [10], [11], [12]). Modeling and simulation has also been applied as a supplement to field evaluation methods early in the systems engineering life-cycle (e.g., [13]). Early life-cycle simulation offered a technique for assessing new tactics, techniques, and procedures (TTP's) and new command and control (C2) technologies at a point in the systems engineering when field testing was not possible or practical. The following sections describe a simulation environment that could provide a similar evaluation capability to assess aspects of the ASSIST problem that cannot easily be addressed through field testing.

To address the challenges of the ASSIST wicked problem, a simulated wicked problem-space could be created, based on several types of models:

- *Models of individuals, groups, and artifacts* (buildings, vehicles) to observe using ASSIST technologies. These

models must represent behaviors that are directly observable, and characteristics that drive the evolution of behavior indirectly, including social and cultural models of affiliation-related behaviors, and propensity for change of those behaviors.

- *Models of individual observers.* These models must represent the cognitive capabilities and resulting products of soldiers as they observe, recall, and interpret information. They must also represent the tactics, techniques, and procedures (the behaviors) required to use technologies to observe..
- *Models of new technologies.* These models must represent the capabilities of an intelligent system for capturing, organizing, and presenting data and descriptions of the environment.

The simulated wicked problem space would also require:

- *A simulation engine.* This engine must generate and measure interactions between these models, across a variety of model parameterizations, and use scenarios.
- *Measures.* The simulation requires a way to present the evaluator with some way to quantify the key concept driving this problem: understanding of the environment. Any single measure is likely to oversimplify the issue, but the simulation must offer some methods for the user to gain an understanding of the ‘level of understanding’ achieved by different observers in the environment.

As a tool to support continuous evolution in understanding of the problem, this simulation environment must support frequent modification, update, and reparameterization of models and measures, and the addition of new models and measures, to enhance the representation of complexity in the wicked problem-space. In a sense, the simulated wicked problem space must be a rapid prototyping tool, but instead of creating prototypes of the solution, it allows the user to create rapid prototypes of *the problem* (by gradually adding and adjusting the complexity of and interactions between the simulation models). For example, and ASSIST simulated wicked problem space would need to allow evaluators to rapidly add new complexities (e.g. simulate soldier behavior under stress, simulate degradations in system performance due to enemy counter-measures ...).

Two of the five simulation components, above, are commonplace. Simulation engines are readily available in flavors that include agent-based systems, blackboards, discrete event simulations, and others. Models of technology are routinely used in the design of new systems, such as those tested in ASSIST. (We note, though, that models are much less sophisticated when it comes to representing the complexity of the world in which these artifacts must operate). However, modeling of observers and the observed presents significant challenges; as does representing the “level of understanding”. In the following section, we articulate one approach to define these models based on cognitive and cultural theory, and describe how an abstract visualization might be used as a method for representing the level of understanding achieved by observers in the simulation.

A. Modeling the “Observed”

To model the people whom MOUT soldiers observe and with whom they interact, we draw on the concept that every person has a “repertoire” of identities [14]. The identify profile for an individual may include affiliations of family, tribe, religion, profession, politics, and other types. Each identity is associated with a belief system, that is, attitudes (e.g., hatred of American troops, respect for our troops) and corresponding propensity towards behaviors of interest (e.g., attacking troops, assisting troops). The question of which identity is dominant at a given moment is a function of 1) the propensity of the individual to choose an extreme vs. neutral identity in response to events and 2) the degree to which the context of an event is tied to an identity. For example, an individual prone to extreme responses, and witness to an American assault on insurgents near a tribal neighborhood school, may interpret the events through a tribal identity and respond defensively, violently to the American actions.

As any anthropologist will tell you, culture is not static. Thus, models of the observed must allow an individual to change the distribution of cultural affiliations over time. When many individuals shift away from a given identity (e.g., with a radical or insurgent group), the belief system of the remaining adherents tends either to soften by shifting towards the center (thus increasing the number of adherents), or to harden by shifting towards extremes (thus reducing the number of adherents to the most devoted). Models of multiple cultural identities are a promising area of research. We have developed such models using software agents, to explore methods of stabilizing failing states.

B. Modeling the “Observer”

To model the observers, our soldiers, we draw on the rich literature from cognitive psychology. In particular, we model the capabilities and constraints of people under stress to observe, recall (or report), and interpret what they recall.

Observation must be modeled as an activity driven both by goals (“On this mission, look for suspicious activity at the market.”) and knowledge of the patterns that meet the goals (e.g., The absence of women in the market is suspicious). [15]’s seminal studies of perception justify the effort to model search for patterns as a goal-driven activity. Decades of research concerning human pattern learning and pattern recognition argue for modeling pattern knowledge. The combined effects of modeling these phenomena is a bias towards observing what one is told to see, and what one knows. Unexpected and unfamiliar entities and events will tend not to be encoded (written to memory) during observation, even if they are in fact critically important.

In addition, observation capability sensitive to stress; people tend to narrow the focus of their attention under threat [16], for example, making peripheral events effectively unobservable. Thus, both stressors and sensitivity to them must be modeled.

Recall is dependent (logically) on observing events in the

first place and (psychologically) towards highly salient events [17]. Thus, it may be difficult for people to recall events that were not highly charged in some way. Further, recall is biased towards erroneously inferring things that are expected in a given context, but were in fact absent [18], [19].

Finally, decisions about (i.e., the interpretation of) recalled events are subject to a range of well-documented biases [20]. Their net effect is to cause observers to discount, and thus underreport evidence that conflicts with their prior beliefs (e.g., about what they expected to see), and to over-rely on, and thus over-report, evidence that confirms their beliefs. Biases of these sorts must be modeled.

A variety of methods exist for modeling these cognitive characteristics of individual observers, including ACT-R models for highly detailed, temporally accurate representation of behavior, through simpler, rougher rule-based models in JESS (the Java Expert System Shell), PROLOG and other languages may be adequate.

C. Representing “Level of Understanding”

Figure 2 provided an abstract representation of the dynamics found in ASSIST’s wicked problem space. Here we suggest that this type of abstraction of the problem might offer a useful means for visually qualifying and perhaps even quantifying the level of understanding achieved by different “observers” working in the ASSIST simulated wicked problem space.

To aid in this description, Figure 2 is repeated here as Figure 3. From a simulation perspective, Figure 3 might represent “ground truth” in the simulated world at any one point in time.

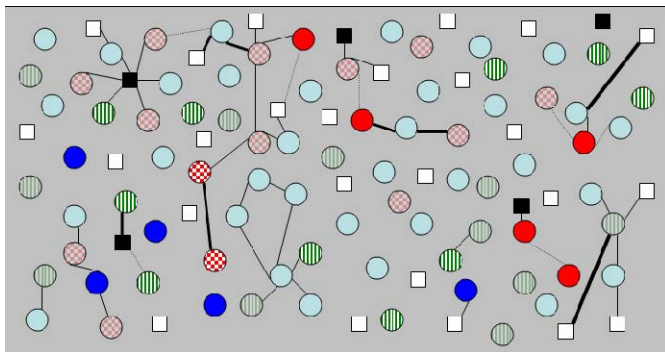


Fig. 3. Visualization of “ground truth” in a simulated MOUT environment

This figure shows the present state of the cultural affiliations and relationships for the “observed” models in the simulation. The state of this figure would evolve over time according to the dynamics of the cultural models driving the observed entities.

Figure 4 provides a visualization of a simulated ASSIST system’s “understanding” of this environment. This figure could be constructed based on a model of the program’s stated requirements for the system and refined based on results from field evaluations. This example represents results from an ASSIST technology model that is capable of correctly identifying 80% of the entities in the environment from Figure

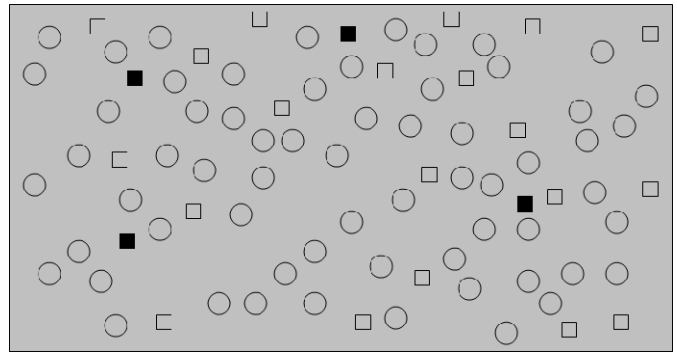


Fig. 4. Visualization of a simulated “system model’s understanding” of the MOUT environment

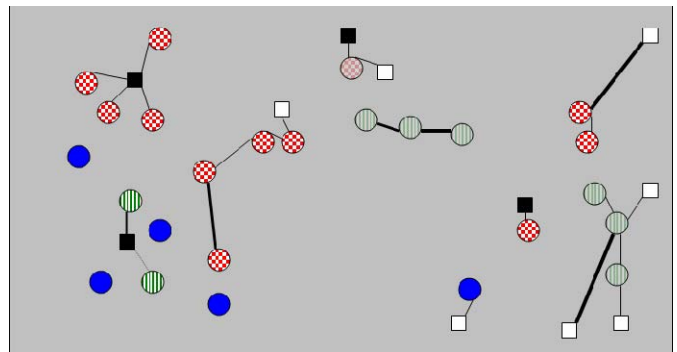


Fig. 5. Visualization of a simulated “human observer model’s understanding” of the MOUT environment

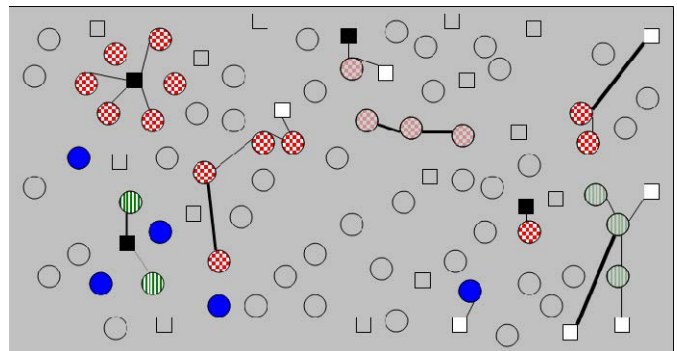


Fig. 5. Visualization of a simulated “high-level observer model’s understanding” of the MOUT environment

3, with a 0% false alarm rate. Taken by itself, Figure 4 offers insights into the system’s contribution to the level of understanding achieved in the environment. Specifically, this technology presents a relatively accurate representation of entities in the environment, but lacks crucial data attributes (e.g., the existence and intensity of relationships and interactions) that might lead to a higher level understanding of the environment.

Figure 5 is a visualization representing a report produced by the model of a human “observer soldier’s” for the environment from Figure 3. This figure could be constructed based cognitive models of human observation and recall skills, and refined based on results from vignette testing. This example is indicative of a model that provides much sparser

recall of entities (and in some instances identifies entities incorrectly), but provides a much richer recognition of critical relationships and interactions in the environment, and offers speculations about affiliations for entities. Figure 5 might represent one squad report's contribution to the level of understanding achieved in the environment.

In a simulation environment, multiple squad reports (variations of Figure 5) and multiple ASSIST system datasets (variations of Figure 4) could be constructed, based on interactions between different observer, observed, and system models over the course of multiple simulated missions. The net result would be a large database of "inputs" to a simulated intelligence officer "observer" model. This intelligence officer's level of understanding would be represented in a form like Figure 6. This figure is a visualization representing a "human observer model" that attempts to achieve an overall understanding of the environment (e.g. an intelligence officer) by combining soldier squad reports (e.g. Figure 5) with ASSIST system reports (e.g. Figure 4). This simulation model of the intelligence officer could be based on cognitive models for baseline decision-making, and tuned to reflect findings from vignette testing. For example, Figure 6 reflects a hypothetical result where the intelligence officer reviews ASSIST data, using it to make changes in the soldier report's assessments of individual entity affiliations.

Overall, these visualizations would provide a way to make qualitative assessments about the level of understanding different "observers" are able to achieve (i.e. Figures 4, 5, and 6) relative to an evolving ground truth situation (i.e. Figure 3). In this example, the observer represented by Figure 6 has successfully identified all the solid dark blue entities (which might represent clerics from one religious group), but has incorrectly identified a cluster of deep red hash-marked entities in the upper left corner (which might represent the mis-identification of a commingled group of students with different affiliations as a unified group with the same affiliation). In addition to qualitative comparison, this method could support quantification of levels of understanding by developing a scoring scheme based on the accuracy and completeness of one observer's visualization figure relative to the ground truth visualization figure.

These results could be useful for evaluating ASSIST technology interactions with other complexities in the environment, but perhaps more important, the simulation environment would support dialogue about the assumptions made in constructing the models, and allow project stakeholders to quickly modify those assumptions to determine their level of influence on the apparent results. As such, the simulated wicked problem space would provide systems engineering decision-makers with a new capability that would supplement and expand upon the feedback provided by field evaluation results.

V. SUMMARY

A socio-technical system is a complex web of interactions between individual humans, social, cultural, and technological

forces. Systems engineers must, increasingly, develop technologies that function within these systems. However, traditional field evaluation methods don't provide timely, decisive feedback concerning the adequacy of technology solutions. At best, they often advance our knowledge of the environment, lengthen the list of requirements technology must fulfill, and deepen our appreciation of the unpredicted effects of injecting one technology solution into a complex system. This is the very definition of a wicked problem. To respond, we need wicked methods. This paper offers simulation as an approach that can supplement traditional field evaluation methods by modeling the environment, technology capabilities, and technology effects early and iteratively. This accelerates and improves the feedback available in systems engineering efforts for addressing socio-technical problems.

ACKNOWLEDGEMENTS

This work is based on experiences gained working as part of the ASSIST IET, led by a team from the National Institute of Standards and Technology (NIST). The ASSIST program was supported by the Defense Advanced Research Projects Agency (DARPA) ASSIST program (POC. Mari Maeda). Any opinions, findings, conclusions, and/or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of NIST or DARPA.

REFERENCES

- [1] DARPA, "Advanced Soldier Sensor Information System and Technology (ASSIST) Proposer Information Pamphlet," http://www.darpa.mil/ipto/solicitations/open/04-38_PIP.htm, 2006.
- [2] H. Rittel and M. Webber; "Dilemmas in a General Theory of Planning" *Policy Sciences*, vol. 4, pp. 155-169, 1973.
- [3] C. Schlenoff, B. Weiss, M. Steves, A. Virts, M. Shneier, and M. Linegang "Overview of the First Advanced Technology Evaluations for ASSIST" in Proceedings of the Performance Metrics for Intelligent Systems Workshop, 2006.
- [4] B. Weiss, C. Schlenoff, M. Shneier, and A. Virts "Technology Evaluations and Performance Metrics for Soldier-Worn Sensor Systems for ASSIST" in Proceedings of the Performance Metrics for Intelligent Systems Workshop, 2006.
- [5] M. Steves (2006) "Utility Assessments of Soldier-Worn Sensor Systems for ASSIST" in Proceedings of the Performance Metrics for Intelligent Systems Workshop, 2006.
- [6] J. Conklin *Dialogue Mapping: Building shared Understanding of Wicked Problems*. Chichester: Wiley, 2005, ch. 1, pp. 1-25.
- [7] F. J. Diedrich, E. E. Entin, S. G. Hutchins, S. P. Hocevar, B. Rubineau, and J. MacMillan "When do organizations need to change (Part I)? Coping with incongruence" in Proceedings of the Command and Control Research and Technology Symposium, 2003.
- [8] E. E. Entin, F. J. Diedrich, D. L. Kleinman, W. G. Kemple, S. G. Hocevar, B. Rubineau, and D. Serfaty "When do organizations need to change (Part II)? Incongruence in action" in Proceedings of the Command and Control Research and Technology Symposium, 2003.
- [9] E. E. Entin, S. A. Weil, D. L. Kleinman, S. G. Hutchins, S. P. Hocevar, W. G. Kemple, et al. "Inducing Adaptation in Organizations: Concept and Experiment Design" in Proceedings of the Command and Control Research and Technology Symposium, 2004.
- [10] S. A. Weil, G. Levchuk, S. Downes-Martin, F. J. Diedrich, E. E. Entin, K. See, et al. "Supporting Organizational Change in Command and Control: Approaches and Metrics" in Proceedings of the International Command and Control Research and Technology Symposium, 2005.
- [11] S. A. Weil, W. G. Kemple, R. Grier, S. G. Hutchins, D. Kleinman, and D.

- Serfaty "Empirically-driven Analysis for Model-driven Experimentation: From Lab to Sea and Back Again (Part 1)" in Proceedings of the Command and Control Research and Technology Symposium. 2006.
- [12] Y. Levchuk, G. Levchuk, R. Grier, S. A. Weil, and D. Serfaty "Empirically-driven Analysis for Model-driven Experimentation: From Lab to Sea and Back Again (Part 2)" in Proceedings of the Command and Control Research and Technology Symposium, 2006.
- [13] Y. Levchuk, G. Levchuk, R. Grier, S. A. Weil, and D. Serfaty "Empirically-driven Analysis for Model-driven Experimentation: From Lab to Sea and Back Again (Part 2)" in Proceedings of the Command and Control Research and Technology Symposium, 2006.
- [14] S. Lovell, G. Levchuk, and M. Linegang "An Agent-based Approach to Evaluating the Impact of Technologies on C2" in Proceedings of the Command and Control Research and Technology Symposium, 2006.
- [15] U. Neisser *Cognition and reality*. San Francisco: Freeman, 1976.
- [16] P. Hancock "Human Factors in Homeland Security" in proceedings of Human Factors Research and Homeland Security: Current and Future Applications. 2005.
- [17] D. Redelmeier and D. Kahneman "Patients' memories of painful medical treatments: Real-time and retrospective evaluations of two minimally invasive procedures" *Pain*, vol. 66, pp. 3-8. 1996.
- [18] J. R. Anderson and G. H. Bower *Human associative memory*. Washington: Winston and Sons, 1973.
- [19] G. H. Bower, J. B. Black, and T. J. Turner "Scripts in memory for text." *Cognitive Psychology*, vol. 11, pp. 177-220. 1979.
- [20] D. Kahneman, P. Slovic, and A. Tversky *Judgment Under Uncertainty: Heuristics and Biases*. New York City: Cambridge University Press. 1982.

ABSTRACT

MEMETICS AND INTELLIGENT SYSTEMS

By

Dr. Robert Finkelstein, President
Robotic Technology Inc.
RobertFinkelstein@compuserve.com

The **conventional definition** of the meme is that it is a **self-reproducing and propagating information structure** analogous to a gene in biology. The meme is able to replicate using hosts and to influence behavior to promote replication. The word “meme” is a neologism described by Richard Dawkins in *The Selfish Gene* (1976), although it may have originated as “mneme,” a transliteration of the Greek word representing the notion of a unit of social evolution and introduced (1904) by German evolutionary biologist Richard Semon in *Die Mnemische Empfindungen in ihren Beziehungen zu den Originalenempfindungen* (English translation 1921).

The **meme is analogous to the gene**, with respect to the process of Darwinian natural selection, in that it is a unit of cultural inheritance which alters individual (psychological) and group (sociological) behavior – and ultimately evolution of the species. Memes and genes may have **cooperative or adversarial** relationships. The successful replication and propagation of the meme is independent of the usual social criteria of “good” or “bad” (e.g., truth, ethics, or science); good ideas can become extinct while bad ideas flourish.

Memes (like genes) do not have cognition or foresight – they (like genes) have **algorithms** which drive **natural selection**. The **evolutionary algorithm** generates complex entities from simple entities – a process for creating design out of chaos without the aid of mind. The **evolutionary process** for memes, as with genes, includes: **variation**, an abundance of different elements; **heredity or replication**, where elements can create copies or replicas of themselves; and **differential fitness**, where the number of copies of created elements varies depending on interactions among features of the elements (whatever makes an element different from other elements) and features of the environment in which it persists.

The meme conjecture is that the evolutionary algorithm (heredity, variation, selection) can run on different substrates (genes and memes), and that **humans are the product of not one but two replicators**: genes and memes. Memes might explain why humans exhibit non-genetic altruism and become **suicide bombers** (as well as other behavior adverse to genetic survival). **Examples of memes include**: ideas, tunes, poems, catch-phrases, fashion, technological processes (e.g., making arrowheads or nuclear warheads), fables, religion, graffiti, images, novels, movies, insurgent or terrorist culture, military culture (training, tactics, strategy, doctrine, policy).

In the **meme lifecycle**, the host transmits the meme intentionally or unintentionally via a transmission vector, and is then received and encoded by the host. The meme is then transmitted to a new host in any number of different ways, including stone engraving, speech, text, image,

observed behavior, and email. It can be seen that the meme lifecycle resembles Claude Shannon's iconic process for a general communications system.

Arriving at a **new potential host**, the meme is received and decoded. The potential host becomes an actual host if the meme satisfies certain selection and fitness criteria. The new host replicates and transmits the meme (perhaps with a different vector). Because there is always a great excess of potential memes over available receptive brains, the fitness criteria winnow the surviving memes. The **selection and fitness criteria** include such motivators and hooks as **threats** (hell or failure) and **rewards** (heaven or success), or the meme might be **beneficial** (in a practical way) or **entertaining** to the recipient, or consist of **appreciative direct feedback** to the recipient (providing emotional satisfaction). To be readily acceptable to the host, the meme should fit **existing constructs or belief systems** of the host, or be a paradigm to which the host is receptive. Memes also aggregate and reinforce in complexes (**memeplexes**) so that a suitable existing framework in the mind of the host is especially susceptible to a new meme which fits the framework. Suitable **storage capacity**, in memory or media, is necessary for the meme to persist, along with **enduring vectors** (e.g., the meme is literally chiseled in stone or reproduced in many, widely distributed copies of books or electronic media).

There is great potential **military worth** for memetics applications, including: Information Operations; Psychological Operations; Military Deception; Counter-propaganda; and Public Affairs. It could have a profound effect on **influencing potential adversary actions to avoid conflict**, as well as combat readiness and operational effectiveness. The Defense Advanced Research Projects Agency has sponsored preliminary analysis to determine whether memetics can become a quantitatively-based scientific discipline.

Intelligent machines could control the evolution, dissemination, and persistence of memes more precisely than Homo sapiens, engendering **memetic engineering**, which would be analogous to the genetic engineering of genes.

An Infrastructure to Support Performance Analysis in Complex Systems

Man-Sze Li
IC Focus,Ltd
London, UK
mqli@icfocus.co.uk

Abhijit Deshmukh
NSF
Arlington, VA
adeshmuk@nsf.gov

Al Jones
NIST
Gaithersburg, MD
jonesa@cme.nist.gov

Abstract: Systems research efforts have increased in a variety of disciplines. Despite these efforts, it is still difficult to predict long-term performance and to understand the relationship between the performance of the parts and the performance of the whole.

The traditional approach to dealing with system performance is based on the philosophy of Descartes, which involves three steps. First, you decompose the original global problem into independent, local sub-problems (the parts). Second, you find solutions to each local sub-problem, ignoring any interactions. Third, you recompose these local solutions to get the global solution (the whole).

Researchers in a number of fields have been quite successful at developing approaches to optimize the performance of the parts. They have not been, however, as successful with predicting the performance of the whole from those parts. We believe that there are two reasons for this. First, they ignore the underlying organizational structure of the system, which can impact its overall performance. Second, the overall system performance is impacted more by the interactions of the parts than it is by their individual performance. Those interactions are captured inherently in the information that they share. For a variety of reasons, that information is often not available.

In this paper, we review some of the modeling approaches that are used to estimate performance. We also review some of the recent network research and its relationship to system performance. We focus on a proposed vision of a new information infrastructure called the Interoperability Service Utility (ISU) and discuss how this infrastructure can help address the aforementioned two problems.

Keywords: complex systems, information, network, optimization, performance

I. INTRODUCTION

Systems research efforts have increased in a variety of disciplines. Despite these efforts, it is still difficult to predict long-term performance and to understand the relationship

between the performance of the parts and the performance of the whole. Researchers believe that these difficulties stem from the fact that these systems are somehow “complex” – without really knowing what that means.

The traditional approach to dealing with performance in engineered, complex systems is based on the philosophy of Descartes. It can be best described as a reductionism and it involves three steps. First, you decompose the original global problem into independent, local sub-problems. Second, you find solutions to each local sub-problem, ignoring all others. Third, you recompose these local solutions to get the solution to the global problem.

Researchers in operations research, artificial intelligence, and control theory have developed techniques to implement the first two steps. They have proposed sophisticated strategies that decompose the original problem into a number of local sub-problems. These strategies, which are based on principles from optimization theory, graph theory, and control theory, yield familiar tree and lattice structures. They have generated thousands of algorithms, models, and heuristics that produce optimal or near-optimal solutions to the resulting, local sub-problems. We summarize major approaches to their development in Section II.

In recent years, several researchers have begun to focus on the reconstruction of solutions to the local sub-problems into a solution to the original problem. These efforts rely on a thorough understanding of the underlying organizational network structure of the real system. We give a brief summary of some of the recent network-related research in Section III.

All of these reconstruction efforts rely on the timely, error-free, and meaningful exchange of information from a wide range of sensors and software applications. This exchange requires a new infrastructure, which we describe in Section IV. In Section V, we discuss some of its impacts on complex systems in particular, and science/engineering in general.

II. PERFORMANCE MODELS

The usual approach to understanding system performance is to build models. Typically, we model complex systems as

highly interconnected, multilayered, hierarchical networks of such models. The layering occurs in both temporal and spatial domains. The bottom layers model physical processes that transform or transport physical objects. As we move “up” the hierarchy, the layers model informational processes that affect those physical objects¹.

In the following sections, we describe several approaches for building such models. These models have many characteristics including discrete or continuous time, discrete or continuous state, linear or non-linear behavior, and deterministic or non-deterministic inputs and outputs. In addition, there are several, often conflicting, quantitative performance measures that drive the solutions to these models. For simplicity, we have classified these modeling approaches into two major categories: optimization approaches and simulation approaches. We describe three popular optimization approaches and two simulation approaches.

A. Optimization Approaches

Since the late 1940s, mathematical programming techniques have been applied extensively to solve optimization problems. Techniques include linear, non-linear, integer, mixed-integer, and dynamic programming. Until recently, the use of these approaches has been limited because many real-world problem optimization problems are NP-complete. A number of specialized algorithms, based on the structure of the associated mathematical representation, have been developed to solve many of these problems. In addition, advances in computational power and software engineering have produced numerical techniques for implementing those algorithms. Finally, several decomposition approaches have been proposed to reduce the computational complexity of individual problems [1- 4].

Neural networks, which attempt to mirror the learning and decision-making capabilities of human beings, have also been used to solve optimization problems. Supervised learning neural networks (SLNN) attempt to learn the right decisions from historical data by capturing the relationship between inputs and desired outputs. Back-propagation, which is the most popular SLNN [5, 6], applies the gradient-descent technique to ascertain that relationship. It is a popular technique when historical training data is available. A number of temporal learning approaches have been proposed when historical data is not available [7, 8].

Genetic algorithms (GA) are an optimization methodology based on a direct analogy to Darwinian natural selection and mutations. In principle, genetic algorithms encode a parallel search process through the solution space, with each process attempting coarse-grain hill climbing [9]. Induced changes and recombinations of these solutions are tested against an evaluation function to see which ones will survive to the next

¹ In manufacturing systems, these processes include machining, inspection, and assembly. In living systems, they include digestion, reproduction, and respiration.

generation. The success of genetic algorithms depends critically on the initial population of solutions. They have been most successful when combined with another technique that generates starting solutions close to the optimal.

B. Simulation Approaches

1) Discrete Event Simulation

Discrete event simulation (DES) models are mainly flow models that track the flow of entities through the factory. The task of the modeler is to determine the state variables that capture the desired behavior, events that change the values of those variables, and the logic associated with each event. Executing the logic associated with each event in a time-ordered sequence produces a simulation of the system. As each event occurs, it is removed from the sequence and the next event is activated. This continues until all the events have been processed. Statistics are gathered throughout the simulation and reported with performance measures (average delays, down time, and throughputs to name a few). Different probability distributions can be associated with each process to simulate natural variations [10].

DES does have two major drawbacks. First, one can only establish estimations of and correlations among variables and performance measures using statistical models. The underlying reasons for or causes of these estimations and correlations cannot be deduced from the models themselves; they must be inferred. Although critical to effective decision-making, understanding the difference between correlation and causality is not always easy. Consequently, erroneous causal inferences can be drawn based on the estimated correlations. Second, DES models allow us to evaluate the system performance for specific values of decision variables or control policies. They do not allow us to determine the stability of the system in any region or neighborhood of those values or policies. This is of critical importance in complex systems where system performance may be driven by hidden, causal relationships that may be highly non-linear. In such systems, small deviations from the optimal decision point can cause disproportionately large changes in the system performance. To better understand these causal relationships and their possible non-linear effects, we turn to system dynamics simulations.

2) System Dynamics Simulation

System dynamics is a method for studying the evolution of many real-world systems [11]. It can also be viewed as a conceptual approach to facilitate the understanding of complex problems [12]. Its central concept is that all the objects in a system interact through causal relationships. These relationships arise from feedback loops, where a change in one variable affects other variables over time; these variables, in turn, affect the original variable, and so on. System dynamics asserts that these relationships form a complex underlying

structure for any system. This structure may be empirically or theoretically discovered. It is through this discovery that the causal relationships become clear and predictions of the future behavior of the system become possible.

The creation of a complete dynamic model of a system requires the identification of the causal relationships that form the system's feedback loops [13]. Feedback loops can be either negative or positive. A negative feedback loop is a series of causal relationships that tend to move behavior towards a goal. In contrast, a positive feedback loop is self-reinforcing. It amplifies disturbances in the system to create high variations in behavior. Causal loop diagrams are important tools for representing the feedback structure of the systems. A causal loop diagram consists of variables connected by arrows denoting the causal influence among the variables.

From these causal loops, we can develop a stock and flow graphical structure. Stocks are accumulations of information or materials that characterize the state of the system. They generate the information upon which decisions and actions are based. They also create delays by accumulating the differences between the inflow and outflow of a process. Flows are rates that are added to or subtracted from a stock. This graphical description of the system can be mapped into a mathematical description of the system, usually differential or difference equations. These equations form the basis for the simulation, which advances using a predetermined time-step.

C. Integrated Approaches

Recently, two approaches to integrating such models have gained favor in the research community. The first integrates distributed simulation models; the second integrates simulation with other optimization techniques.

Distributed simulation models execute independently but interact with each other either interactively or sequentially. Interactive integration of discrete event models requires the interleaving of events from the different models. The High Level Architecture (HLA), which was developed by the Defense Department for integrating battlefield simulators, is the most common technique for achieving interactive integration today [14]. A number of manufacturing models have also been built using HLA including those described in [15, 16]. Sequential integration means that the simulation models are run one after the other. In [17] the authors have integrated a system dynamics model with a discrete event model to evaluate resource allocation decisions in a semiconductor company. In [18], the authors integrate multiple system dynamics models to capture the interactions of several critical infrastructures.

Recently, a number of researchers have proposed hybrid models that integrate simulation approaches with optimization approaches. In [19], the authors combine systems dynamics simulations with neural networks to improve performance in manufacturing supply chains. System Dynamics was used to model the supply chain behavior over time. Neural Networks

were utilized to detect the changes at a very early stage and predict their impact. Then, decomposition, linearization, and eigenvalue analysis were applied to make modifications to the information and materials flows to avoid any undesirable predicted behavior. In [20], the authors combine systems dynamics models, discrete event models, and non-linear programming to integrate hierarchical production planning with vendor managed inventory for a multi-product supply chain.

D. Remarks

Clearly, building system-level performance models will require the integration of many sub models, which are built using the aforementioned approaches. In linking numerous models together, researchers must pay special attention to the underlying organizational structure. That structure can have a dramatic impact on system-level performance. In Section III, we review some of the recent work in this area.

Ultimately, these models will be built and these approaches will be implemented in software applications. The information needed to run these applications will not be resident within the applications themselves. In fact, it may be resident in another computer system or data repository somewhere on the Internet. Managing, exchanging, and manipulating that information automatically and securely poses numerous problems. In Sections IV and V, we discuss the two emerging visions for an information infrastructure that can address these problems.

III. ORGANIZATIONAL NETWORK STRUCTURE

As we build larger and larger system-level models from local sub-models, an important relationship emerges between the underlying organizational structure of these models and system performance. That structure is typically modeled as a network of interconnected nodes. One need only study the spread of recent power failures, Internet viruses, and global diseases to understand the importance of the topology of that network on its performance. In this section, we review some of the recent research that relates the two.

A. Topologies

At one end of the spectrum we have ordered topologies such as chains, grids, lattices and fully-connected graphs. At the opposite end of the spectrum is the random graph [21], where the expected topology of an n -node random graph changes as a function of the number of edges, m . When m is small, the graph is likely to be fragmented into many small clusters of nodes, called components. As m increases, the components grow, at first by linking to isolated nodes and later by coalescing with other components. A phase transition occurs at $m = n/2$, where many clusters crosslink spontaneously to form a single giant component. For $m > n/2$, this giant component contains on the order of n nodes – the network

goes from being almost totally disconnected to almost totally connected. More precisely, the size of the giant component scales linearly with n , as $n \rightarrow \infty$, while its closest rival contains only about $\log n$ nodes. Furthermore, all nodes in the giant component are connected to each other by short paths and the maximum 'degree of separation' between any two nodes grows more slowly.

Although regular networks and random graphs are both useful idealizations, many real systems have network topologies between these two extremes. Watts and Strogatz [22] studied a simple model that can be tuned to model any such topology. Starting with a lattice structure, they replaced the original links by random ones with probability, p ($0 \leq p \leq 1$). They called the resulting structures 'small world' networks that have both small degrees of separation and high degrees of clustering. Barabasi and Reka [23] studied a particular kind of small-world network in which a very few nodes, hubs, were far more connected than others. They showed that the probability $P(k)$ that a node in the network connects with k other nodes was proportional to $k^{-\gamma}$. Networks that have this property are called scale free. The parameter can be thought of as the degree of clustering. For most networks studied to date, the parameter γ satisfies $2 < \gamma \leq 3$. In this form, essentially all graphs with a power law degree distribution were grouped together as scale-free. These networks have two important properties: random node failures have very little effect on connectivity or performance, but deliberate attacks on such a network's hubs can dismantle a network with alarming ease.

B. Impacts on Performance

When the network structure is ordered, the principal cause of complexity is the nonlinear dynamics of the nodes only [24]. We need not be concerned about additional complexity caused by the network structure itself. If the dynamical system at each node has stable fixed points and no other attractors, the network tends to lock into a static pattern. The intermediate case where each node has a stable limit cycle has turned out to be particularly fruitful especially in the study of biological systems. At the opposite extreme, suppose each node has a chaotic attractor. Few rules have emerged about the effect of coupling architecture on dynamics in this case.

When the network structure is random or small-world, much is known about the impact of structure on system performance when the behavior of the nodes is simple – power grids and the Internet are two such systems. Little is known about the impact of structure on performance when the individual nodes are non-linear dynamical systems - particularly when that structure changes over time.

IV. A NEW INFRASTRUCTURE

The Interoperability Service Utility (ISU) is one of the Grand Challenges identified in the Enterprise Interoperability Research Roadmap [25], developed under an extensive, open process of voluntary contributions coordinated by the

European Commission. The Roadmap's objective is to define and characterize the areas of research needed in the domain of enterprise interoperability. It is intended to be an input to the European Union's forthcoming seventh research framework program [26], FP7.

The roadmap envisions a diversity of continuously evolving ecosystems of enterprises. Interoperability of enterprises will be a key feature both within and across such ecosystems. Specifically, interoperability will be a *utility-like* capability for enterprises, a capability that is (1) available at (very) low cost; (2) universally or near-universally accessible; (3) "guaranteed" to a certain extent and at a certain level of performance, with a set of common rules; and, (4) not controlled or owned by any single private entity. By providing these capabilities, the ISU will fill an essential gap in a market that is concerned with promoting the "next big thing" rather than interconnecting the present "islands" of interoperability.

The ISU is conceived to be a basic infrastructure that will enable knowledge-oriented collaboration by supporting information exchange between diverse knowledge sources, software applications, and Web services. Conceptually, the ISU will constitute the next "layer" of open cyberspace, sitting atop the Internet and other evolving Web technologies. It will be independent of, rather than an extension to, particular enterprise software systems. Those systems, including the ones described above, would flow above the ISU and be provided by technology vendors.

The ISU is premised on several important assumptions including (1) application functionality is represented by and delivered as a service; (2) services may reside anywhere and be invoked on the fly; (3) the precise location of a service and means of access may not be pre-determined; (4) the number and variety of objects, devices and systems that need to communicate will continue to dramatically accelerate [27]; and (5) proprietary business intelligence and information assets should reside outside the ISU.

A. Some Potential Services

Information objects, ontologies and metadata repositories will be at the core of the ISU. Timely, error-free, and meaningful exchange of information will be among the minimum service guarantee and built into the design of the ISU (see below). A number of potential ISU services have been identified including

- Services that facilitate real-time information sharing and collaboration between enterprises, such as reasoning, searching, discovery, composition, assembly, and delivery of semantics automatically
- Services that leverage emerging Web technologies for enabling a new generation of information-based applications that can self-compose, self-declare, self-document, self-integrate, self-optimize, self-adapt, and self-heal, as encapsulated in the concept of Service-Oriented Knowledge Utility (SOKU) among others [28]

- Services that support knowledge creation, management, and acquisition to enable knowledge sharing between virtual organizations
- Services that help connect islands of interoperability by federating, orchestrating, or providing common e-business infrastructural capabilities such as digital signature management, certification, user profiling, identity management, and libraries of templates and interface specifications
- Services that support the use of mashup technologies such as verification of credentials; reputation management; assessment of e-business capabilities; assessment of collaboration capabilities; facilities for data sourcing, integrity, security and storage; contracting; registration and labeling; and payment facilities, among others

B. Some Important Issues

The vision of the ISU raises major performance issues at the service level and at the organizational level. First, the ISU is concerned primarily about preserving and propagating the meaning of the information. This contrasts dramatically with (1) the Internet, which is concerned only communication and (2) the Web, which is concerned only with presentation. However, the meaning of information is neither universal nor static. In a given real-world environment, the meaning of information is as much determined by conventions as by rules. With its “any-to-any” and “end-to-end” assumptions, the ISU cannot rely on specific conventions; nor can it unilaterally impose its own rules. More likely, they will need to be discovered or negotiated dynamically between the sender and receiver. This has important consequences for performance optimization, which now becomes, partially at least, a function of the openness and performance of the infrastructure.

Second, the primary entity exchanged via the ISU is information, as opposed to data, datagrams, or messages. This is important because a lot of research into interoperability has been concerned with the codification of business processes into pre-packaged software applications involved in manipulating, capturing and handling information, rather than the information itself. This static view is a consequence of the best practices of the solution providers, not of the business practices of end users. From the perspective of the end users, what is valuable and what may be shared is the information, not the IT systems that deal with that information. This means that system performance metrics and the techniques used to predict them are tied intrinsically to information – not information systems. Consequently, the software services that implement those techniques must understand information objects as the unit of exchange. The ability to do this may reside in either the services or the objects themselves. This latter capability raises the possibility of self-describing information objects. Defining the properties of such objects and developing methods for discovering them, will be critical to the successful implementation of the ISU.

Third, the ISU will evolve over time into a complex system in its own right. Therefore, it will be, in principle, subject to performance and organizational concerns described above - not just for technical purposes such as diagnostics, rollback, and recovery, but also for business purposes such as service pricing. There is, however, an added modeling complication because there are two fundamental entities in this system: information transactions and information objects. Although the “state” of a transaction is well understood in pure communications terms, what constitutes the “state” of an object as it propagates through the system is by no means clear. Moreover, the current values of these “states” are key determinants for the necessary services to be invoked. Crucially, unlike static systems, a shared state between the node (end system) and the network (the ISU infrastructure) cannot be assumed. In so far as the ISU services are not linear, they can alter the state of the information, possibly adding value to the information in business terms – possibly not.

Fourth, the ISU is expected to guarantee payload and message flow with a pre-defined, but possibly varying, quality of service, QoS. Traditional approaches, which are independent of the substance of the payload and the identities of the exchange parties, tend to focus on delivery modes, sequencing, queuing, and control states. To meet the varying QoS requirement, the ISU must be able to use (1) parameters associated with information integrity and quality and (2) knowledge about the identities of the transacting parties² to determine the transaction “routine” - including event notification, exception handling, failure recovery, and reporting. Furthermore, it must do this without infringement of the rules of privacy and data protection specified in the European Union’s Directive on Data Protection.

Fifth, for all practical purposes, the ISU would be a highly decentralized system of networked nodes performing a wide variety of services. Referring to the discussion in Section III, and recognizing the evolutionary nature of the ISU, the interesting question is, “What organizational structure best serves the main aims of the ISU, especially at the initial design phase?” For example, would the ISU be best served by a high degree of clustering which favors “centers of excellence” in service terms, or by more democratic, peer-to-peer transactions where mediating services “compete” on a case-by-case basis? In addition, since the value placed on the information is not identical between the transacting parties would the network evolve in alignment with the distribution of value across the nodes? For example, consider a structure in which large clusters of nodes are connected because of their common interest in particular collections of valuable information.

C. Design Implications

² Ascertaining that knowledge will be complicated because there will be no automatic alignment between the identity of a node and its location.

The above issues suggest that the ISU will evolve into a multi-layered, complex, non-linear, dynamic system. Consequently, its up-front design is vitally important. A set of design principles, modeled on the Internet, have been proposed for the ISU. These include: the end-to-end argument, preference of modular structures over hierarchical layering, transparency, minimum circumstances for message transactions, and scalability considerations. However, these principles are drawn from a communications paradigm, which is aligned with the reductionist problem-solving approach. We argue, based on our preceding discussions, that the ISU design principles must be drawn from an information paradigm, which must be aligned with a reconstructionist problem-solving approach. Therefore, more research is needed to develop such an approach, which we believe will come from synthesizing and distilling the research efforts and results from a variety of sciences.

V. HOW WILL THE ISU BE USED

The ISU described in the preceding section has the potential to become a kind of secure, artificial, nervous system for the engineered systems of the future [29]. It will provide the capability for optimal, integrated management of these systems. Given its broad scope and applicability, it has a wide spectrum of potential uses – one of which is managing complex systems.

A. Managing Complex Systems

Engineered complex systems, such as advanced manufacturing, service enterprises, power systems, smart structures, emergency response, and environmental control are undergoing significant changes due to the development of new sensors. These sensors, and the networks that integrate them, will provide enormous amounts of data that have the potential to make such systems much more visible. That potential will only be reached if the data can be appropriately mined, analyzed, and managed. If this happens, then it can be used to monitor, control, and improve system performance in real-time.

However, these issues are likely to present significant computational challenges. For instance, real-time control in chemical processing and other manufacturing plants pose global optimization problems that are NP-hard. Implementing even approximate solutions in real-time may require the resources of a high-end platform or a computational grid. Similar issues of real-time decision and control arise in operating the power grid, where detection of overloaded power lines must be followed with appropriate reactive control. Such controls may be possible if sensor data can be integrated at multiple levels with the optimization and modeling techniques described above.

B. Multi-scale, Computationally Intensive, Interoperable Models

One of the fundamental challenges of the kind of multi-scale modeling described in Section 2 stems from the need to maintain modularity and interoperability between models of differing scales. Consider a prototypical example from the field of computational fluid dynamics in which the solution to large sets of partial differential equations (PDEs) is obtained using Monte Carlo simulation. The computational complexity associated with these simulations is large, so, as described above, a decomposition approach is typically used. In this case, that approach is known as “domain decomposition” in which the entire domain is subdivided into multiple sub-domains which are assigned to separate processors. The computations are then coordinated by imposing algebraic boundary conditions to be satisfied by each processor/domain. Clearly, software imposing these algebraic conditions must interoperate with software simulating the PDEs on each processor.

There are many engineering disciplines in which such multi-scale models must interoperate. Researchers must define decomposition strategies and coordination protocols that make it possible for the resulting sub-models to inter-operate. The development of such strategies and protocols will be much easier with the ISU.

C. Multi-laboratory Collaborative Research

Many research institutions have invested in specialized laboratories that house expensive, one-of-a-kind equipment - such as the Large Hadron Collider at CERN. In the past, only the researchers at these institutions were able to use this equipment, which could generate terabytes of raw, experimental data. Resulting research papers typically contained only the analysis of that data, and not the data itself. Other researchers have found it nearly impossible to replicate those results without the original data and without knowledge of the original experimental setup.

Similarly, expensive power system equipment, like a FACTS (Flexible AC Transmission System) controller, is being studied in certain laboratories. Power market simulators have been developed in other laboratories across the country. The current Internet computing infrastructure does not allow experimental validation of new FACTS controls using market scenarios that overload the system.

Getting authorization to and gaining access to experimental data can be a logistical nightmare. One of the major benefits of the ISU will be removal of the access problems. The availability of such large data sets for global, unrestricted data will change dramatically the way science and engineering are done in the future.

D. Collaborative Model Development

New research in science and engineering often requires the use of predictive models, as well as experimental investigations. Although the collaborative physical

infrastructure discussed above provides linkages between physical laboratories, it is essential that the ISU also provide a suite of open-source software services that are not only portable, but also provide the basis for studying robustness of the optimization and modeling approaches described above. This capability will enable a new computational science of modeling and algorithms, which may, in turn, spawn the development of new and better approaches.

In addition, this provisioning of software services will allow greater collaboration among researchers working on software development. As a specific example, an automotive design engineer may be interested in predicting pollutant formation for certain pressures and temperatures. Using available benchmark data, quantum mechanical predictions, and software services available through ISU will provide optimized parameters for the engine.

E. Convergence of Critical Infrastructures and the ISU

The infusion of information technology into the critical infrastructures – communications, transportation, power, and water - has led to the concept of smart structures. They will be equipped with sensors, and software to allow them to monitor their health, compare performance with peers, adapt to changes, and sometimes take action to avoid catastrophes. The convergence of these critical infrastructures with the ISU is expected to accelerate the development of these smart structures.

The notion of networks has long been associated with the traditional infrastructures mentioned above. However, this notion of networks has expanded to the notion of ecosystems described in Section IV. These ecosystems form the life-blood of modern society, and the interdependencies between the various ecosystems is beginning to attract greater attention. Modeling, optimization, and simulation associated with these interdependent systems have become major research needs for engineering. The ISU is expected to provide the infrastructure to realize these needs.

VI. SUMMARY

In this paper, we focused on performance of complex systems. We briefly reviewed some of the optimization and simulation approaches to modeling performance of individual components in those systems and we gave a few examples of recent attempts to integrate such models. We argued that to build system-level models requires special attention to both organization and information. We summarized some of the recent work on small-world and scale-free organizational networks. We then presented a vision for a new informational infrastructure and discussed some of the potential uses of this infrastructure for modeling system performance. We also briefly described some potential impacts on scientific research.

DISCLAIMER

Commercial software products identified in this paper were used only for demonstration purposes. This use does not imply approval or endorsement by NIST or NSF, nor does it imply these products are necessarily the best available for the purpose. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or NIST.

REFERENCES

- [1] Davis, W. and A. Jones, "A real-time production scheduler for a stochastic manufacturing environment", *International Journal of Computer Integrated Manufacturing*, Vol. 1, No. 2, 101-112, 1988.
- [2] Benders, J., "Partitioning procedures for solving mixed-variables mathematical programming problems," *Numerische Mathematik*, Vol. 4, No. 3, 238-252, 1960.
- [3] Dantzig, G. and P. Wolfe, "Decomposition principles for linear programs," *Naval Research Logistics Quarterly*, Vol. 8, No. 1, 101-111, 1960.
- [4] Gershwin, S., "Hierarchical flow control: a framework for scheduling and planning discrete events in manufacturing systems," *Proceedings of IEEE Special Issue on Discrete Event Systems*, Vo. 77, 195-209, 1989.
- [5] Rumelhart, D. and McClelland, J., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press Cambridge, MA, USA, 1986.
- [6] Werbos, P., "Neurocontrol and supervised learning: An overview and evaluation," *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, Van Nostrand Reinhold Publication, New York, NY, USA, 65-89, 1995.
- [7] Rabelo, L., Sahinoglu, M., and Avula, X., "Flexible manufacturing systems scheduling using Q-Learning," *Proceedings of the World Congress on Neural Networks*, San Diego, California, I378-I385, 1994.
- [8] Zhang, W. and Dietterich, T., "High-performance job-shop scheduling with a time-delay network", *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, USA, 1025-1030, 1996.
- [9] Goldberg, D., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Menlo Park, CA, USA, 1988.
- [10] Law, A. and Kelton, W., *Simulation modeling & analysis*, 2nd edition, McGraw-Hill, New York, NY, USA, 1991.
- [11] Forrester, J., *Principles of systems*, Pegasus Communications, Inc., Williston, VT, USA, 1971.
- [12] Senge, P., *The Fifth Discipline*, Currency Doubleday, New York, NY, USA, 1994.
- [13] Sterman, J., *Business dynamics – systems thinking and modeling for a complex world*, McGraw Hill, New York, NY, USA, 2000.
- [14] Kuhl, F., Weatherly, R., and Dahmann, J., *Creating computer simulations: an introduction to the high level architecture*, Prentice Hall, Upper Saddle River, NJ, 1999.

- [15] Umeda, S., and Jones, A., "An integrated test-bed system for supply chain management", *Proceedings of the 1998 Winter Simulation Conference*, 1377-1385, 1998.
- [16] Riddick, F and McLean, C., "The IMS MISSION architecture for distributed manufacturing simulation", *Proceedings of the 2000 Winter Simulation Conference*, 2000.
- [17] Rabelo, L., Helal, M., Jones, A., and Min, H., "Enterprise Simulation: A hybrid approach", *International Journal of Computer Integrated Manufacturing*, to appear.
- [18] Min, H., Beyeler, W., Brown, T., Son, Y., and Jones, A., "Toward modeling and simulation of critical national infrastructure interdependencies", *IIE Transactions*, to appear.
- [19] Rabelo, L., Helal, M., and Jones, A., "Using Neural Networks to Monitor Supply Chain Behavior", IMDS, in review.
- [20] Venkateswaran, J., Son, Y., Jones, A., and Min, H., "A hybrid simulation approach to planning in a VMI supply chain", *International Journal of Simulation and Process Modeling on Supply Chain Modeling and Simulation*, to appear.
- [21] Erdős, P. and Rényi, A., "On the evolution of random graphs", *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17-61, 1960.
- [22] Watts, D., and Strogatz, S., "Collective dynamics of small-world networks". *Nature* **393**: 440-442, June, 1998..
- [23] Barabási, A., and Reka, A., "Emergence of scaling in random networks", *Science*, 286:509-512, October 15, 1999.
- [24] Strogatz, S., "Exploring Complex Networks", *Nature*, Vol. 410, 268-276, March, 2001.
- [25] M-S. Li, R. Cabral, G. Doumeings and K. Popplewell (editors). "Enterprise Interoperability Research Roadmap". Final Version (V4.0). European Commission, 2006.
http://cordis.europa.eu/ist/ict-ent-net/ei-roadmap_en.htm
- [26] European Commission. "Proposals for a Seventh Framework Programme (FP7) for research, 2007-2013, and for a Seventh Framework Programme of the European Atomic Energy Community (Euratom), 2007 to 2011", April 2005
<http://cordis.europa.eu/fp7/home.html>
- [27] ITU. "ITU 2005 Internet Report – The Internet of Things". November 2005.
http://www.itu.int/osg/spu/publications/internetofthings/ftp://ftp.cordis.europa.eu/pub/ist/docs/grids/ngg3_eg_final.pdf
- [28] European Commission. "Future for European Grids: GRIDs and Service Oriented Knowledge Utilities – Vision and Research Directions 2010 and Beyond". January 2006.
- [29] NSF Blue Ribbon Panel Report. "Revolutionizing Science and Engineering Through Cyberinfrastructure", D. Atkins, et al, January, 2003.

Three-Dimensional Data Registration Based On Human Perception

Bruce Brendle, Ph.D.
US Army RDECOM-TARDEC
AMSRD-TAR-R (MS:205)
Warren, MI 48397-5000
bruce.brendle@us.army.mil

Abstract—Registration, the process of transforming different sets of data into a common coordinate system, is often required to allow the comparison or integration of the data sets. Techniques such as the widely used Iterative Closest Point (ICP) algorithm have limited effectiveness on data sets that require significant transformation or that have large degrees of inconsistencies. This paper describes a biologically inspired algorithm for data registration that is based on two theories of human perception. The use of macro level registration based on these theories combined with micro level registration using the ICP algorithm provides enhanced registration of these challenging data sets. The new algorithm was tested extensively on simulated sensor images in several scenarios key to successful application to autonomous ground navigation. The excellent performance of the biologically inspired algorithm in these cases makes it a promising candidate for this field.

I. INTRODUCTION

Diverse applications ranging from medical imaging to computer vision make use of three-dimensional data. Often, multiple sets of data are acquired by sampling the same scene or object at different times, or from different perspectives, resulting in each data set having its own coordinate system. Registration, the process of transforming different sets of data into a common coordinate system, is then required to allow the comparison or integration of the data sets.

A basic rigid transformation between two coordinate systems, i and k , is composed of a translation and a rotation defined as:

$${}^i \mathbf{t}_k = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_k = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}$$

$$\begin{aligned} {}^i \mathbf{R}_k &= {}^i \mathbf{R}_{k,z(\phi)} {}^i \mathbf{R}_{k,y(\theta)} {}^i \mathbf{R}_{k,x(\psi)} \\ &= \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \\ &= \begin{bmatrix} \cos \phi \cos \theta & -\sin \phi \cos \theta & \sin \phi \sin \theta \\ \sin \phi \cos \theta & \cos \phi \cos \theta & \cos \phi \sin \theta \\ -\sin \theta & \cos \theta \sin \psi & \cos \theta \cos \psi \end{bmatrix} \end{aligned}$$

A data set comprised of m individual points is represented by a $m \times 3$ matrix, \mathbf{D} , with the following notation:

Frame of Reference $[\mathbf{D}]_{\text{Descriptor}}$ Operation on the Data Set

Actual data viewed from viewpoint i is represented as:

$${}^i \mathbf{D}_A = \begin{bmatrix} {}^i \mathbf{p}_{a_1} & {}^i \mathbf{p}_{a_2} & \cdots & {}^i \mathbf{p}_{a_m} \end{bmatrix} = \begin{bmatrix} a_{x_1} & a_{x_2} & \cdots & a_{x_m} \\ a_{y_1} & a_{y_2} & \cdots & a_{y_m} \\ a_{z_1} & a_{z_2} & \cdots & a_{z_m} \end{bmatrix}$$

A key fact that will become important later is that ${}^k \mathbf{D}_A$ the actual data viewed from viewpoint k , is not necessarily equal to ${}^i \mathbf{D}_A$ for several reasons. If viewpoints i and k are separated spatially, data that was occluded at viewpoint i may become visible at viewpoint k . Also, if viewpoints i and k are separated temporally, i.e. the data is captured at different times, new data may appear.

The forward kinematics relationship between the data sets is given by:

$${}^i \mathbf{D}_A = {}^i \mathbf{R}_k {}^k \mathbf{D}_A + {}^i \mathbf{T}_k \quad -1$$

This relationship is the foundational equation for data registration. The goal of registration is to determine ${}^i \mathbf{R}_k$ and ${}^i \mathbf{T}_k$ that will transform the data sets into a common coordinate system so that the data sets can be compared or integrated. A byproduct of accurate registration having great benefit to the autonomous navigation application is that the vehicle's 3D movement can be calculated from ${}^i \mathbf{R}_k$ and ${}^i \mathbf{T}_k$, allowing localization of the vehicle with respect to a known starting position. This is similar to dead reckoning based upon heading and motion sensors without its drawback of error being proportional to the amount of distance traveled.

Several issues make direct calculation of ${}^i\mathbf{R}_k$ and ${}^i\mathbf{T}_k$ impractical for real world applications. Two of the most significant issues are *inconsistencies in the data sets* and *noise in the measurements of the data sets*.

Inconsistencies may exist due to new data appearing when data capture points are separated temporally or, in the case of spatially separated capture points, when data becomes visible that had previously been occluded. Also, actual data is not available in many applications and we are left to deal with noisy measurements of the data, the size of the errors being related to the quality of the sensors used to measure the data.

II. DATA REGISTRATION APPROACHES

A review of published literature reveals that it is rich in the area of data registration. There are many published approaches to application-independent image registration [1]. The majority of the literature comes from the medical imaging community, whose emphasis is on accuracy of registration over speed of registration [2]. The autonomous ground navigation community is also addressing the issue of range image registration [3]. The notable difference in the work performed by this community is the emphasis on speed over accuracy as registration speed is required for high-speed navigation. An optimal approach to registration should have both performance characteristics, accuracy and speed.

These approaches can be classified by several characteristics including rigid or elastic, extrinsic or intrinsic, and feature-based or voxel-based. For the application of autonomous ground navigation, techniques that are rigid and intrinsic are most relevant. Of these approaches, voxel-based techniques are the most promising for applications requiring high-speed registration such as autonomous ground navigation as they avoid the processing time and error-potential of identifying features to be registered. The voxel-based technique most applicable to autonomous ground navigation is the Iterative Closest Point (ICP) algorithm.

III. ITERATIVE CLOSEST POINT ALGORITHM

Substituting measured or sensed data for the actual data in (1), the data sets for the image registration problem are the sensed data at viewpoints i and k , ${}^i\mathbf{D}_S$ and ${}^k\mathbf{D}_S$. The goal of registration is to find the three-dimensional transformation ${}^i\mathbf{D}_k$ (the combined rotation and translation) that minimizes the distance between the points in these sets, or more specifically:

$$\min_{\mathbf{T}, \mathbf{R}} \sum_{n=1}^m \left\| {}^i\mathbf{D}_{n,S} - ({}^i\mathbf{R}_k {}^k\mathbf{D}_{n,S} + {}^i\mathbf{T}_k) \right\|^2 \quad \square$$

The ICP algorithm solves (2) iteratively in the following steps: **S**

1. Establish a set of n closest points between the data sets ${}^i\mathbf{D}_S$ and ${}^k\mathbf{D}_S$.
2. Compute the incremental transformation, ${}^i\mathbf{R}_k$ and ${}^i\mathbf{T}_k$, using the set of closest points.
3. Apply the incremental transformation from Step 2 to ${}^k\mathbf{D}_S$

4. If relative changes in ${}^i\mathbf{R}_k$ and ${}^i\mathbf{T}_k$ are less than a threshold, terminate. Otherwise, repeat the procedure starting from Step 1.

The identification of closest points between two 3-D point sets required for the first step of the ICP algorithm can be accomplished by several methods including k-d trees and Voronoi polygons. The MATLAB function *dsearchn(X, XI)*, which returns the indices of the closest point in X for each point in XI was used for this research.

Likewise, the computation of ${}^i\mathbf{R}_k$ and ${}^i\mathbf{T}_k$ required for the second step of the algorithm can be performed by several methods including quaternions [4] and singular value decomposition [5].

Two major issues remain with the ICP approach. First, the amount of transformation possible with ICP is bounded and registration speed increases proportionally with the initial distance between images. The second issue is that inconsistencies in the data sets due to such real-world events as moving objects or occluded objects contribute directly to errors in registration accuracy.

IV. BIO-INSPIRED ALGORITHM

Fortunately, there exists an example of a system that handles these issues – the human perception system. This biological system, although not fully understood, has been modeled by physiologists and theories of its operation have been postulated and tested. This research project has applied recent theories in how humans register visual images during high speed eye movements and how they handle inconsistencies in the visual images to computer-based image registration. The resulting bio-inspired registration process has reduced dependency on iterative registration techniques such as ICP and is robust in cases of high degrees of inconsistency in the data sets. The process works well in all cases except in the extreme case in which there is no overlap between the data sets, making registration impossible by any method, and when there is not a sufficient difference in the data sets from which ${}^i\mathbf{R}_k$ and ${}^i\mathbf{T}_k$ can be calculated. An example of the latter case would be images from a sensor moving parallel to a flat surface, which are indistinguishable even by the human eye.

The proposed registration approach is based upon human use of extraretinal signals to estimate visual transformation and the assumption of stationarity. These concepts have been applied to macro level registration and then combined with the ICP algorithm for fine tuning in order to achieve accurate registration in cases where ICP alone does not perform well.

A. Saccadic Suppression

1) *Perception Theory*: A saccade is a rapid movement of the eye that results in the smearing of the image seen by the eye when the saccade occurs during the retinal integration time [6].

A similar phenomenon occurs while photographing high speed events. If the speed of the subject (relate this to a high-speed movement of the eye) is higher than the shutter speed of the camera (relate this to the integration time of the eye), the moving subject will appear blurry in the captured image.

To prevent this smearing, it has been suggested that the brain utilizes a *saccadic suppression* mechanism to shut off retinal processing during eye saccades [7] and that this suppression is triggered by extraretinal signals. The source of these signals is debatable and may come from extraocular muscles that measure actual movement of the eye [8], the efferent command that initiates the eye movement [9], or some combination of the two [10]. Regardless of the source, these signals can be used to estimate the required transformation between the pre- and post-saccadic images [11], allowing humans to accurately register the images. It is important to note that this mechanism accounts for transformation of images resulting solely from movement of the eye and does not account for inconsistencies in the images from such events as moving objects.

2) *Registration Equivalent*: Pre- and post-saccadic images can be equated to a series of medical images captured during a linear scan or to images captured from a sensor on a moving unmanned vehicle. In the same manner extraretinal signals are used to calculate the predicted transformation between the trans-saccadic images, positioning sensors on the medical imaging system or unmanned vehicle can be used to calculate an initial transformation between the sensor images.

Recall from (1) that the goal is to register two data sets, ${}^i\mathbf{D}_S$ and ${}^k\mathbf{D}_S$ captured from viewpoints i and k , respectively. The hypothesis is that we can transform ${}^i\mathbf{D}_S$ to a viewpoint, j , that is very close to the viewpoint k using noisy measurements of the transformation between the viewpoints, ${}^i\mathbf{R}_{k,S}$ and ${}^i\mathbf{T}_{k,S}$. This results in ${}^j\mathbf{D}_T$ with the subscript indicating a *transformed* data set. Next, a registration technique such as ICP can be used on the resulting data sets to remove any error in the measurements of ${}^i\mathbf{R}_{k,S}$ and ${}^i\mathbf{T}_{k,S}$.

Applying the kinematics relationship from (1), to the data sets of interest, we have:

$${}^i\mathbf{D}_S = {}^i\mathbf{R}_{k,S} {}^j\mathbf{D}_T + {}^i\mathbf{T}_{k,S} \quad - \square$$

The transformed data set ${}^j\mathbf{D}_T$ can then be calculated as:

$${}^j\mathbf{D}_T = {}^i\mathbf{R}_{k,S}^{-1} ({}^i\mathbf{D}_S - {}^i\mathbf{T}_{k,S}) \quad - \square$$

The registration of the resulting data set, ${}^j\mathbf{D}_T$, with ${}^k\mathbf{D}_S$ by means such as ICP provides the fine tuning rotation and translation parameters ${}^j\mathbf{R}_{k,I}$ and ${}^j\mathbf{T}_{k,I}$. Estimates of the desired total rotation and translation values can then be estimated as:

$${}^i\hat{\mathbf{R}}_{k,A} = {}^i\mathbf{R}_{j,S} {}^j\mathbf{R}_{k,I} \quad - \square$$

$${}^i\hat{\mathbf{T}}_{k,A} = {}^i\mathbf{T}_{j,S} + {}^i\mathbf{R}_{j,I} {}^j\mathbf{T}_{k,I} \quad - \square$$

Here the errors are limited to errors inherent in the registration technique. For accurate estimates of ${}^j\mathbf{R}_{k,A}$ and ${}^j\mathbf{T}_{k,A}$, the calculated values of the parameters ${}^j\mathbf{R}_{k,I}$ and ${}^j\mathbf{T}_{k,I}$ from the registration process must have negligible errors. However, *the registration of the data sets with negligible error is not trivial when there are inconsistencies in the data sets.*

B. Stationarity Assumption

1) *Perception Theory*: Registration of trans-saccadic images in the presence of moving objects or during the appearance of previously occluded objects is more complex. Although the extraretinal signals can be used to develop an estimate of the transformation between the images, the visual system must still account for unexpected changes in the images, which could be attributed to either moving objects or errors in the transformation estimate.

Image processing research in the area of Structure from Motion (SfM) has led to two hypotheses regarding the mechanism for registration in this case. A long-standing hypothesis, called the *rigidity assumption*, stated that the visual system would always choose the most rigid transformation, i.e. the one that required the least deformation [12]. An example of rigid transformation of a singular object would be a car moving parallel to an observer. Even though the car is moving, its size and shape remain the same.

Recent work has resulted in the postulation of a *stationarity assumption* [13] that would be considered along with rigidity in visual registration. Stationarity is a preference for objects to remain fixed in an allocentric, earth referenced coordinate system. An example of stationarity is the use of buildings as landmarks for navigation. As a person drives a car and these landmarks become occluded and then visible again, they assume that the buildings have remained stationary and they are able to estimate their position relative to these stationary objects. In weak stationarity, when multiple solutions are equally rigid, the visual system will select the one that is most stationary. In strong stationarity, the visual system will chose a stationary solution even if it is detectably non-rigid. Recent experiments support the hypothesis of strong stationarity [14].

2) *Registration Equivalent*: In the same manner that humans perceive visual images, an assumption of stationarity can be applied to the data sets ${}^j\mathbf{D}_T$ with ${}^k\mathbf{D}_S$ to compensate for any new data, changing data, or the appearance of previously occluded data. After translation of ${}^i\mathbf{D}_S$ from viewpoint i to viewpoint j , the data sets are within close enough proximity to each other that an assumption of, or preference for, stationarity would require corresponding data in the data sets that is not new, changed or previously occluded to be within some threshold, ε , of each other. To align the data sets using the assumption of stationarity, any points exceeding that

threshold should be excluded from the final registration process.

Prior to performing this distance check, however, it is first necessary to insure that the data sets are ordered by corresponding points, i.e. that each point $^j\mathbf{p}_{a_n,T}$ in $^j\mathbf{D}_T$, corresponds to $^k\mathbf{p}_{a_n,S}$ in $^k\mathbf{D}_S$ for all points n in the world coordinate system. This can be accomplished by identifying and removing any points in the data sets that are not within shared areas of the data sets, as defined by their extreme boundaries in the allocentric coordinate system.

The removal of these outlying points from their respective data sets results in data sets $^j\mathbf{D}_N$ and $^k\mathbf{D}_N$ that contain only points in a common area of the world coordinate system and that are ordered by corresponding points. Here the subscript N indicates data sets corresponding to an intersection of space in world coordinates.

It is a computationally inexpensive process to identify those points in the data sets where $\|{}^j\mathbf{p}_{a_n,N} - {}^k\mathbf{p}_{a_n,N}\| > \varepsilon$ and to remove those points from each of the data sets. This results in data sets $^j\mathbf{D}_C$ and $^k\mathbf{D}_C$ containing only the *consistent* components of the data sets. These data sets can now be finely aligned using a standard technique such as ICP.

C. Bio-Inspired Algorithm

The use of macro level registration based on theories of human perception combined with micro level registration using the ICP algorithm, allows (2) to be solved iteratively in the following steps (representative data sets are shown for illustrative purposes):

- 1) Data sets (range images) are captured at viewpoints i and k .
- 2) $^j\mathbf{D}_T$ is constructed from estimates of the rotation and translation between images using (4).
- 3a) Areas of each image, $^j\mathbf{D}_T$ and $^k\mathbf{D}_S$, that are not within a common area of the world are identified.
- 3b) These areas of the images are removed, resulting in data sets $^j\mathbf{D}_N$ and $^k\mathbf{D}_N$.
- 4a) Corresponding points in $^j\mathbf{D}_N$ and $^k\mathbf{D}_N$ that have a large Euclidean separation are identified as inconsistencies.
- 4b) The stationarity assumption is applied by removing these points from both data sets to create consistent data sets $^j\mathbf{D}_C$ and $^k\mathbf{D}_C$.
- 5) ICP is performed on the resulting data sets to determine $^j\mathbf{R}_{k,I}$ and $^j\mathbf{T}_{k,I}$.
- 6) The overall estimate of the transformation from viewpoint i to viewpoint k is estimated as:

$${}^i\hat{\mathbf{R}}_{k,A} = {}^i\mathbf{R}_{j,S} {}^j\mathbf{R}_{k,I}$$

$${}^i\hat{\mathbf{T}}_{k,A} = {}^i\mathbf{T}_{j,S} + {}^i\mathbf{R}_{j,I} {}^j\mathbf{T}_{k,I}$$

V. TEST RESULTS

Three conditions were tested to verify the performance of the bio-inspired registration algorithms in comparison to the standard ICP algorithm in scenarios relevant to autonomous ground navigation. Experiments were run to simulate images from a 3-D laser detection and ranging sensor mounted on a vehicle driving through an urban environment, on a stationary vehicle scanning a scene, and finally on a stationary vehicle observing a scene while experiencing periods of high occlusion. These scenarios also represent the most challenging areas for ICP; images with large amounts of transformation and images with a high degree of inconsistency.

Three performance characteristics were measured to compare performance. Convergence speed was calculated by determining the iteration at which the algorithm had converged to within 95% of its final values for every transformation parameter. Translation error was calculated by subtracting the Euclidean distance between the translation calculated by the algorithm from the actual translation between the data sets. Rotation error was calculated by subtracting the rotation parameters calculated by the algorithm from the actual rotation values.

A. Moving Vehicle

Vehicle motion was simulated by moving the sensor viewpoint in half meter increments in the x and y dimensions for a total translation of ten meters in both directions. Figure 1 illustrates the motion of the vehicle during the experiment. Test cases for data collection consisted of the starting image and the image captured at the current sensor location, e.g. images for Case 10 consisted of the starting image and the image captured after moving the vehicle 5 meters in both the x and y dimensions.

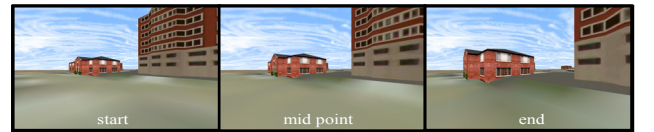


Fig. 1. Camera Images from Vehicle Motion Experiment

The biologically inspired registration algorithm outperformed the standard ICP algorithm for the moving vehicle experiment in all cases for registration speed and for registration accuracy as seen in Figures 2, 3, and 4. The major benefit of the new algorithm is seen when there is a large amount of transformation between images. The ICP algorithm stopped returning reasonable amounts of error when the separation between the images exceeded 4.5 meters in both the x and y directions. The biologically inspired algorithm continued providing accurate results up to the maximum tested separation of 10 meters in both the x and y directions.

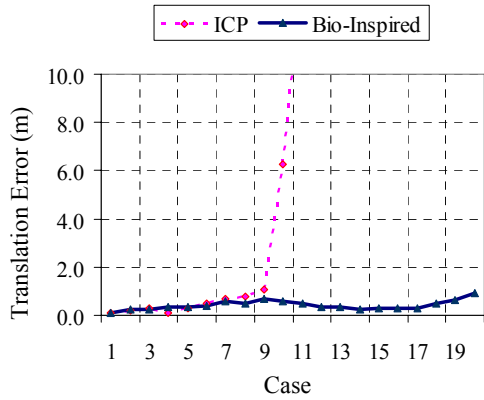


Fig. 2. Camera Images from Vehicle Motion Experiment

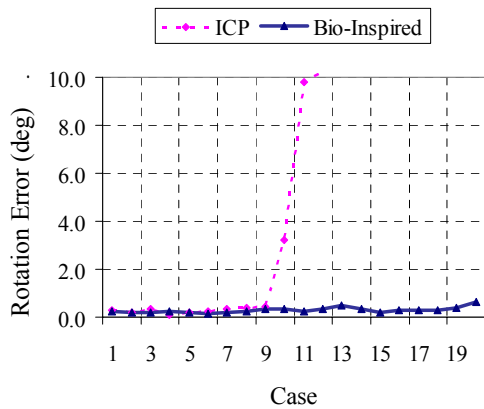


Fig. 3. Camera Images from Vehicle Motion Experiment

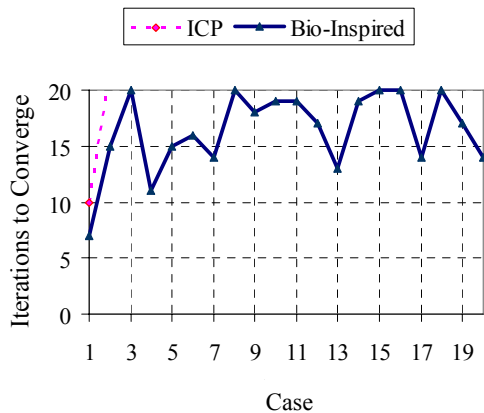


Fig. 4. Camera Images from Vehicle Motion Experiment

B. Scanning Sensor

Sensor scanning was simulated by fixing the sensor viewpoint and moving the sensor field of view in 0.5 degree increments for a total rotation of 10 degrees. Experiments were run with the sensor panning only, with the sensor tilting only, and then with the sensor panning and tilting. Figure 2

illustrates the sensor motion during each of these experiments.

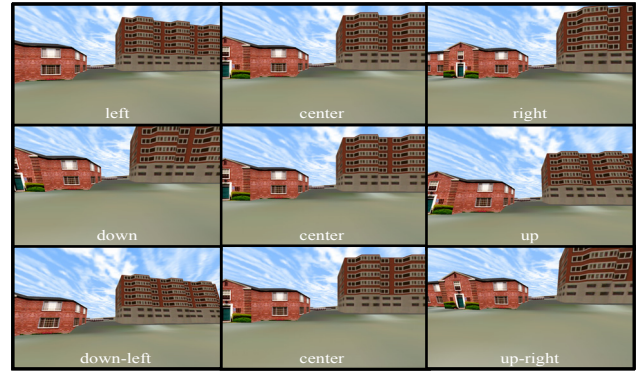
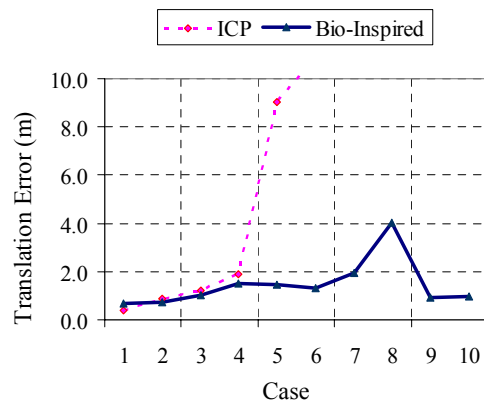


Fig. 5. Camera Images from Pan (top row), Tilt (second row) and Combined Pan and Tilt (bottom row) Experiments

Similar to the moving vehicle results, the biologically inspired algorithm proved much faster than the standard ICP algorithm during each of the sensor scanning experiments and outperformed the ICP algorithm for accuracy in all but two trials. For the experiments involving only panning of the sensor, the ICP algorithm was not able to accurately register the images for large amounts of panning, while the bio-inspired algorithm continued to perform well. Both algorithms were more sensitive to tilting than panning, which can be attributed to the comparatively smaller field of view of the sensor in elevation. Overall, the results were consistent, with the ICP algorithm experiencing significantly higher rates of error at relatively small amounts of sensor rotation, converging more slowly than the biologically inspired algorithm, and diverging at high amounts of rotation. Figures 6, 7, and 8 illustrate the results for the sensor pan and tilt experiment.

Fig. 6. Camera Images from Vehicle Motion Experiment



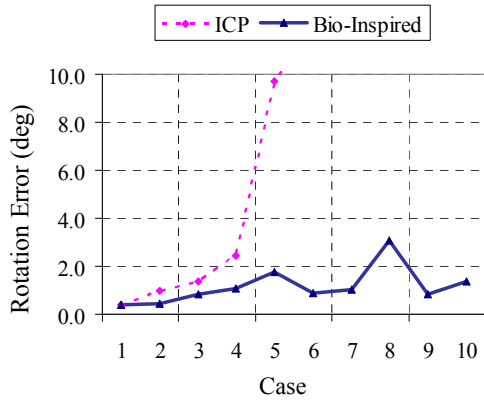


Fig. 7. Camera Images from Vehicle Motion Experiment

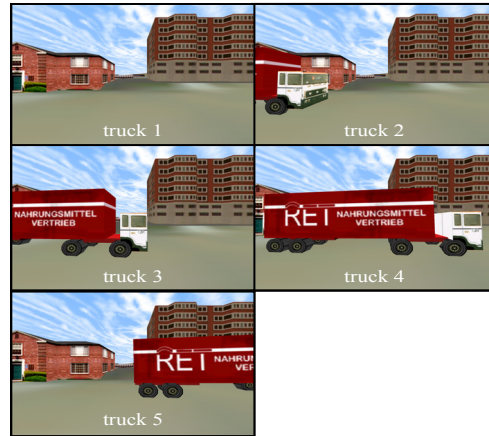


Fig. 9. Camera Images from Moving Object Experiment

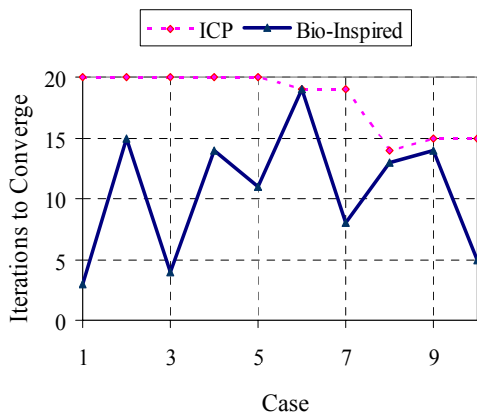


Fig. 8. Camera Images from Vehicle Motion Experiment

B. Sensor Occlusion

Sensor occlusion was simulated by fixing the sensor viewpoint and field of view and then moving a large object through the scene being observed by the sensor. Five images were captured of a truck moving through the scene in order to cover the range of cases from no occlusion to almost total occlusion. Figure 9 illustrates the images used during these experiments. Test cases for data collection consisted of the starting image, ‘truck 1’ and the image captured at the current truck location, e.g. images for Case 2 consisted of the starting image and the image labeled ‘truck 2’.

Sensor occlusion is the most difficult test for voxel based registration techniques. In several of the images tested, 50% or more of the pixels from the base image were occluded in the second image, a scenario which proved impossible for the standard ICP algorithm to address. The ICP algorithm demonstrated poor accuracy and diverged, rather than converged, on a solution. On the contrary, the biologically inspired algorithm performed rapidly and with exceptional accuracy in all cases.

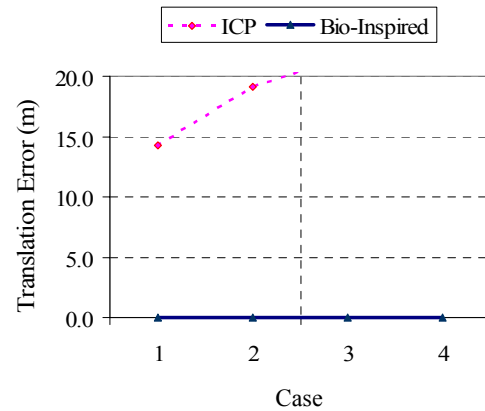


Fig. 10. Camera Images from Moving Object Experiment

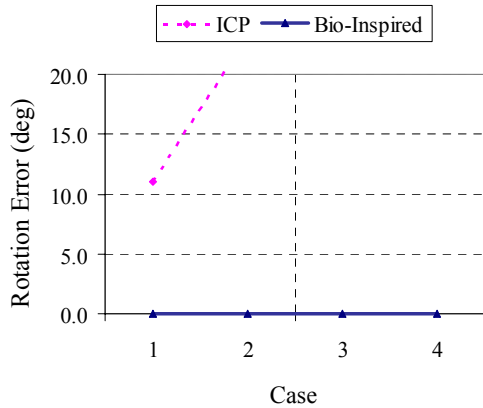


Fig. 11. Camera Images from Moving Object Experiment

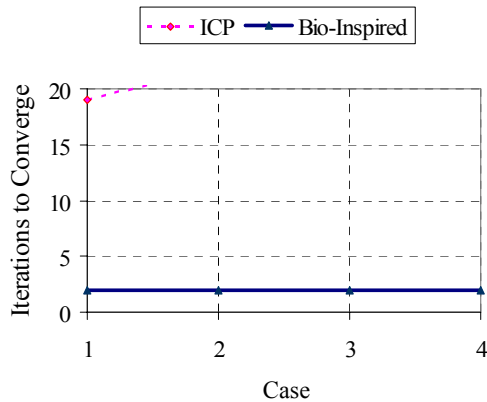


Fig. 12. Camera Images from Moving Object Experiment

D. Computational Efficiency

In addition to improved performance, the bio-inspired technique also improves computation time. On the computing platform used for this research, the combined computing time of the saccadic and stationarity corrections was only 10% of the computing time of a single ICP iteration. In addition, the ICP computing time on the resulting data sets was an average of 31% less per iteration than that required for data sets without the corrections. Together, this results in a decrease of the overall registration computation time.

VI. CONCLUSION

Registration of three-dimensional data sets, and image registration in particular, is a fundamental problem for a wide variety of applications ranging from medical imaging to autonomous ground navigation. Significant research has been conducted in this area, with the combined goal of achieving high-speed, precise registration of data sets. Particular difficulty persists in cases of noisy data sets requiring either significant transformation or those with a

large degree of inconsistency.

This research developed a biologically inspired algorithm based on two theories of human perception that addresses these difficulties and performed extensive testing on data sets associated with the application of autonomous ground navigation.

The biologically inspired algorithm outperformed the ICP algorithm in all cases, with the most significant benefits being found in cases involving a large amounts of translation and/or rotation between the images, for which the ICP algorithm returned unrealistic results, and for cases where there were large amounts of occlusion between the images, for which the ICP algorithm actually diverged rather converged to a solution. Both of these cases are extremely important to autonomous ground navigation. The excellent performance of the biologically inspired algorithm in these cases makes it a promising candidate for this field.

ACKNOWLEDGMENT

The author conducted the work described in this paper as part of his graduate studies and would like to thank Dr. Ka C. Cheok, his advisor, Mr. Scott Pletz, who provided the simulated camera and lidar images that served as the data sets for the autonomous navigation case study, and Dr. Raj Madhavan from the National Institute of Standards and Technology, who shared insight into his work with the Iterative Closest Point algorithm.

REFERENCES

- [1] L. G. Brown. "A Survey Of Image Registration Techniques", ACM Computing Surveys, 24(4):325-375, December 1992.
- [2] P. A. Van den Elsen, E.-J. D. Pol, and M. A. Viergever. "Medical Image Matching - A Review With Classification", IEEE Engineering Medical Biology Magazine, pages 26-39, March 1993.
- [3] R. Madhavan and E. Messina, "Iterative Registration of 3D LADAR Data for Autonomous Navigation", Proceedings of the IEEE Intelligent Vehicles Symposium, pages 186-191, June, 2003.
- [4] B. K. P. Horn. "Closed-Form Solution Of Absolute Orientation Using Unit Quaternions", Journal of the Optical Society of America, 4(4):629-642, 1987.
- [5] K. Arun, T. Huang, and S. Bolstein. "Least-Squares Fitting of Two 3-D Point Sets", IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5):698-700, 1987.
- [6] J. O'Regan, "Solving the 'Real' Mysteries of Visual Perception: The World as an Outside Memory", Canadian Journal of Psychology, pages 461-488, 1992.
- [7] F. Volkman, A. M. L. Schick, L. A. Riggs. "Time Course of Visual Inhibition During Voluntary Saccades", Journal of the Optical Society of America, 58, 562-569, 1968.
- [8] L. Marin, E. Marin, D. G. Pearce, "Visual Perception Of Direction When Voluntary Saccades Occur", Perception & Psychophysics, 5, 65-79, 1969.
- [9] E. Von Holst, H. Mittelstaedt. "The Principle Of Reafference: Interactions between the Central Nervous System and the Peripheral Organs", Perceptual processing: Stimulus equivalence and pattern recognition. New York: Appleton, 1971
- [10] B. Bridgeman, A. H. C. Van der Heijden, B.M. Velichkovsky, "A Theory Of Visual Stability Across Saccadic Eye Movements", Behavioral and Brain Sciences 17 (2): 247-292, 1994.
- [11] M. Wexler, "Anticipating The Three-Dimensional Consequences Of Eye Movements", Proceedings of the National Academy of Science, U. S. A. 102, 1246-1251, 2005.

- [12] H. Wallach, D. N. O'Connell. "The Kinetic Depth Effect", *Journal of Experimental Psychology*, 45: 205-217, 1953.
- [13] M. Wexler, F. Panerai, I. Lamouret and J. Droulez, "Self-Motion and the Perception of Stationary Objects", *Nature*, 409: 85-88, 2001.
- [14] M. Wexler, I. Lamouret and J. Droulez, "The Stationarity Hypothesis: An Allocentric Criterion in Visual Perception", *Vision Research*, 41:3023-3037, 2001.

Performance Analysis of Symbolic Road Recognition for On-road Driving

M. Foedisch, C. Schlenoff, and R. Madhavan[†]
National Institute of Standards and Technology (NIST)
Gaithersburg, MD 20899-8230, USA
{mike.foedisch, craig.schlenoff, raj.madhavan}@nist.gov

Abstract — Previous approaches to road sensing, namely road detection were based on segmenting the sensor data, i.e. color camera image, into road and non-road areas. Performance evaluation for such algorithms could be performed in a relatively straightforward fashion by comparing the algorithm’s result with ground truth. Ground truth for such an image-based evaluation approach could be limited to a geometrical structure describing the road area in the original image. However, the development of our new high-level road sensing approach, which is a model-based approach to road recognition, makes new demands to performance analysis and subsequent performance evaluation which would include comparison with ground truth. In this paper¹, we will briefly describe the new road recognition approach, show performance analysis results and discuss performance evaluation issues.

Keywords: *autonomous driving, model-based perception, road recognition.*

I. INTRODUCTION

Previous approaches to road sensing, namely road detection ([4], [5], [10]) were based on segmenting the sensor data, i.e. color camera image, into road and non-road areas. Performance evaluation for such algorithms could be performed in a relatively straightforward fashion by comparing the algorithm’s result with ground truth (see [7], [10]). Ground truth for such an image-based evaluation approach could be limited to a geometrical structure describing the road area in the original image. However, the development of our new high-level road sensing approach [3], which is a model-based approach to road recognition, makes new demands to performance analysis and subsequent performance evaluation which would include comparison with ground truth.

There are several approaches to performance evaluation which can be classified into the following general categories:

¹Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

[†]Research Staff Member, Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC, For the U.S. Department of Energy under contract DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC-05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

comparative evaluation compares the algorithms performance with similar other algorithms or a ground truth; for *analytic* evaluation the limits, computational complexity and theoretical optimality of the algorithm may be determined; the *performance* on test data and execution times with different parameters may measured; and finally the *appropriateness to the task* can be analyzed given the context of a particular application with its constraints (please refer to [7] for a more detailed discussion on the subject).

We present in this paper a two-level approach to performance analysis for a new road recognition approach providing symbolic descriptions of the road structure. The first level of performance analysis helps point out potentially problematic areas and real-time issues by analyzing the behavior of the tree search-based recognition approach. On the second level an actual performance evaluation is performed by comparing the symbolic results of the algorithm against (semi-) automatically extracted ground truth.

This paper is organized as follows: In Section II we briefly describe the new symbolic road recognition approach. In Section III we introduce the first level of performance analysis and discuss results. Finally, in Section IV we outline the second level which is actual performance evaluation employing ground truth and discuss issues with the automatic ground truth generation as well as results.

II. SYMBOLIC ROAD RECOGNITION APPROACH

Our previous work on road detection on color images demonstrated the advantages of using background knowledge (in terms of models) in order to improve the recognition results [5]. In the following, we will describe a new approach of a model-guided road recognition process [3] and will discuss the type of extracted features, the representation of models, the recognition process, and the representation of the resulting symbolic road data.

A. Feature Extraction

One important assumption of the new approach concerns the orientation of the vehicle on the road. A normal orientation, where the vehicle is limited to traverse only in lanes for which the legal driving direction agrees with the vehicle’s direction, provides a canonical form for the appearance of road on images and may therefore simplify the representation process. All other orientations of the vehicle do not comply with the

normal orientation. We can allow the limitation of a normal orientation if we assume that the autonomous system will be aware of when it leaves the normal orientation (e.g. due to avoidance of obstacles on the road).

Assuming normal orientation of the vehicle on the road, a simple set of features, which are easily extracted and well-understood, can be derived [2]. The features are based on “slices” of the road perpendicular to the direction of the vehicle. They can be extracted by applying one of several approaches for detecting the road area in images or road edge detecting algorithms (e.g. [1], [2], [5], [4], [8]).

Starting at the bottom image row, the left and right road edge points in each row are determined. A pair of road edge points described in both image and world coordinates (through camera calibration) describes one feature item. The process continues bottom-up row-by-row until the world coordinates of the road edges reach a given maximum distance in front of the vehicle (e.g. more than 55 m). Furthermore, additional data will be associated with a feature item, e.g. information about lane markings.

B. Model Representation

Figure 1 depicts our approach for representing road model primitives. A “slice” of road is described by its width (geometrical component) and lane structure in terms of number of lanes and their legal directions (topological component). This representation of road primitives is compatible with the type of feature data described in Section II-A.

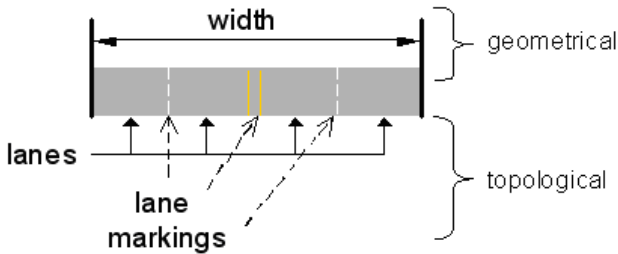


Fig. 1. Geometrical and topological representation of a “slice” of road.

A road type consists of an ordered group of primitive road model items. For such groups additional constraints apply. A road type might require a minimal and/or maximal lateral length or, in the case of road widening and narrowing, a certain monotonic behavior. Other constraints limit the connectivity between (primitive) road types, e.g. a two-lane road segment can connect to a three-lane road segment only through a transitional segment. Primitive road items and road types are organized hierarchically. Additionally, primitive model items are grouped by the type of driving environment, e.g. highway driving, rural road or urban road driving. Appropriate connectors describe transitions from one environment to another (e.g. a highway exit transfers the vehicle from highway driving to rural road driving).

C. Recognition Process

The goal of the recognition process is to find associations between feature items and (primitive) road models and eventually an interpretation of the scene. The application of a tree search algorithm spans potentially all possible associations of feature items and road models [6]. This process, however, is computationally expensive and must therefore be constrained. We define constraints on three different levels, the primitive associations level, the group level, and the symbolic-level interpretation.

On the primitive level, potential associations must comply with unary constraints. For example, in order to associate a feature item to a specific road model, the width of the road has to be similar for both entities. Whenever a feature item is associated with the same model as the previous feature item, group-related constraints apply. Assuming that feature items 1, 2, and 3 in Figure 2 are already associated with a model A, the association of feature item 4 to model A requires compliance of the extended *Group A* to group-related constraints. For example, in the case of a road widening (as part of an intersection) the group should comply with a certain monotonic behavior and the group’s length should be within the maximal length of the model. Assuming another situation where feature items 1-4 are already associated to model A, associating feature item 5 with model B would trigger additional constraints. Starting a new group B causes the previous group A to be closed. This, for example, requires compliance with the minimum length constraint.

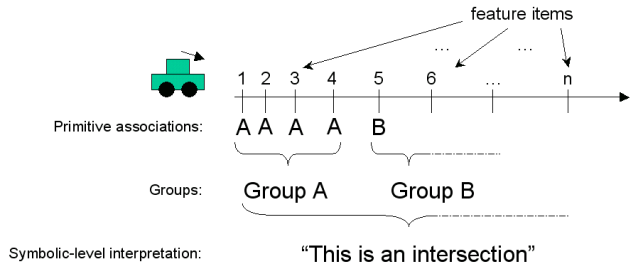


Fig. 2. Recognition process levels: primitive associations level, group level, and symbolic-level interpretation.

Finally, the set of (locally) consistent groups may allow a high-level interpretation of the scene. For example, the occurrence of a regular road segment, a widening segment, a narrowing segment, and another regular road segment (in this order) can be a strong indicator for the existence of an intersection.

Figure 3 depicts an example of a search tree used for the recognition process. On each level of the tree, one single feature item is associated with (potentially) all known models (within the current driving environment, see Section II-B). This potentially huge tree structure (considering all possibilities) will be reduced in numbers of nodes by the above described application of constraints. Branches in the resulting tree that show consistent associations of feature items to models from the root to a leaf of the tree represent surviving

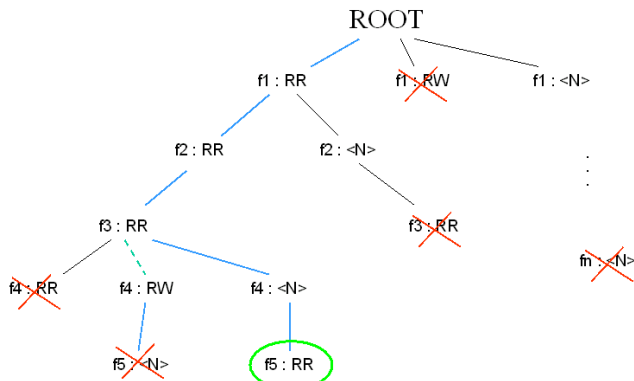


Fig. 3. Sample search tree. The feature items f1 - f5 are associated to the models RR (Regular Road), RW (Road Widening) and $\langle N \rangle$ (for noise). The branches of the search tree are being pruned whenever the associations are inconsistent on the local, group or global level. Paths reaching from the ROOT to one of the leaves are considered interpretations (e.g. blue path to the green circle).

interpretations.

We use the number of nodes and the number of interpretations as measures for performance analysis described in Section III.

D. Symbolic Representation of Road Structures

Figure 4 shows three examples of the symbolic description of our approach's recognition results. Each node describes one road segment's road type, e.g. node A1 in Figure 4(a) describes a straight road segment of a bi-directional two-lane road. The nodes also contain geometrical information such as the road width and segment length. Due to sensor limitations, however, geometrical measures give only a coarse impression and their interpretation should be considered carefully. The examples in Figure 4(b) and Figure 4(c) show more complex road structures. The occurrence of multiple road segments of several types is represented by a chain of nodes.

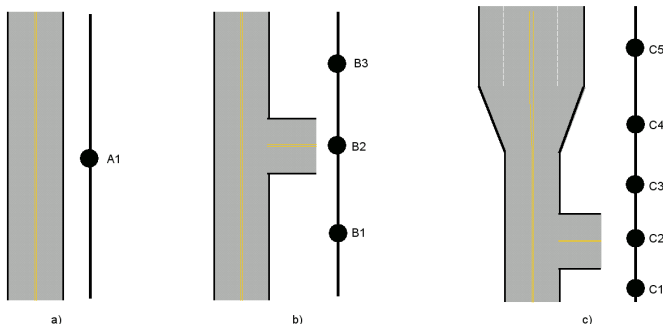


Fig. 4. Examples for symbolic description of recognition results. Each Node is of a certain type, e.g. *Regular Road, two lanes, bi-directional* (A1, B1, B3, C1, and C3), *T-Intersection, from right* (B2 and C2) etc.

III. PERFORMANCE ANALYSIS

As described in Section II-C, we use a constrained tree search approach for our high-level recognition process. Each

execution of tree search can be described by internal parameters describing the resulting search tree structure - the *number of nodes* and the *number of interpretations*. We use both values to gain a first impression of the recognition system's performance.

Figure 5 shows performance analysis results. Figure 5(a) shows the first frame of a test video sequence. In the background the original input image is depicted, in the upper right corner the result of the underlying road detection, in the lower center the most compressed representation of the symbolic results (for the left and right side of the road separately), and on the right side an iconic depiction of the symbolic results can be seen. The graph in Figure 5(b) shows the number of nodes (blue) and the number of surviving interpretations (yellow) for each frame of the test sequence.

From experiments we learned that a typical successful run of our system results in search trees of about a few hundred nodes and about one interpretation. The graph in Figure 5(b), however, shows (in the first half of the sequence) the occurrence of a magnitude higher number of nodes (> 2000) as well as sporadic lack of any interpretations. We consider these as clear indicators of problems with the recognition algorithm's performance for the following reasons:

- no interpretations mean lack of results and, therefore, complete failure of the algorithm;
- a high number of nodes is usually (from our experience) connected with failure or at least sub-optimal results;
- a high number of nodes also means a longer processing time which is usually an issue in real-time implementation.

We analyzed the algorithm's performance on the frames that showed no interpretations and/or a high number of nodes and we found out that in these cases most of the problems were due to a calibration issue. Figure 6 shows the performance analysis results for a second run after fixing some the discovered calibration problems. Compared to the results depicted in Figure 5(a), Figure 6(a) shows the correct symbolic result of a bi-directional two-lane road. The graph in Figure 6(b) appears now smoother with just a few problematic frames in the middle of the sequence where a high number of nodes and lack of interpretations point us to areas that need further investigation.

This fairly simple approach to performance analysis can be used to support further development of the algorithm by pointing out video frames that cause problems. An actual performance evaluation beyond mere heuristics, however, requires a more sophisticated approach and is described in the next section.

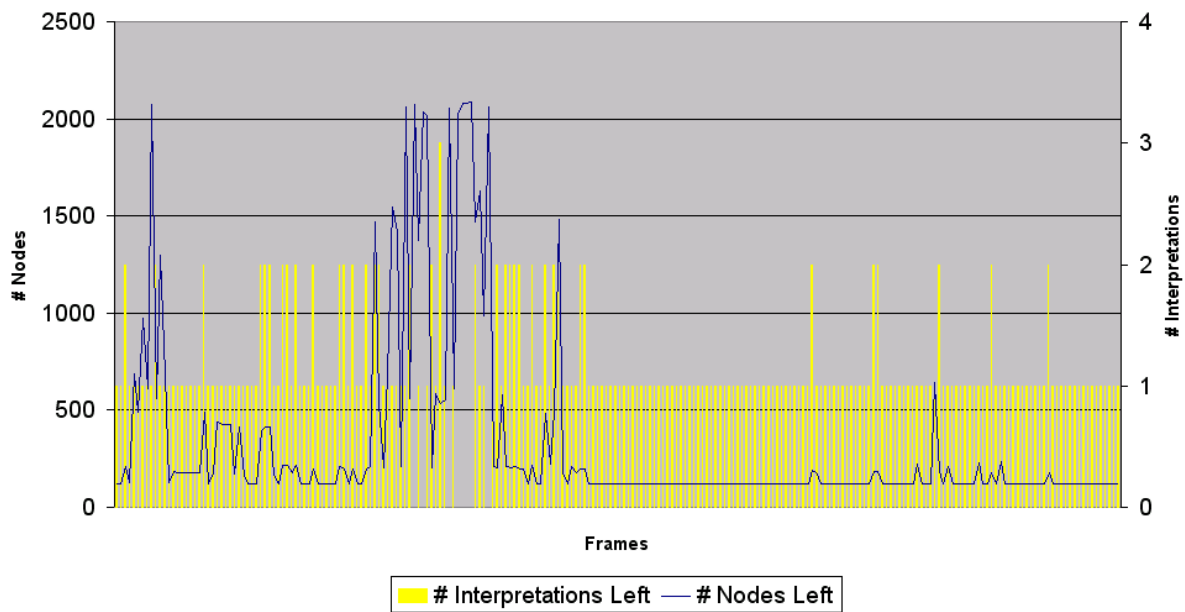
IV. PERFORMANCE EVALUATION

In order to evaluate the performance of an algorithm one needs a reference - the ground truth - to which the algorithm's results can be compared against. Considering our algorithm's results - chains of symbolic nodes - we need a repository of world data from which we can extract comparable structures. We decided to exploit an existing structure - the NIST Road Network Database (RNDB). In the following, we describe



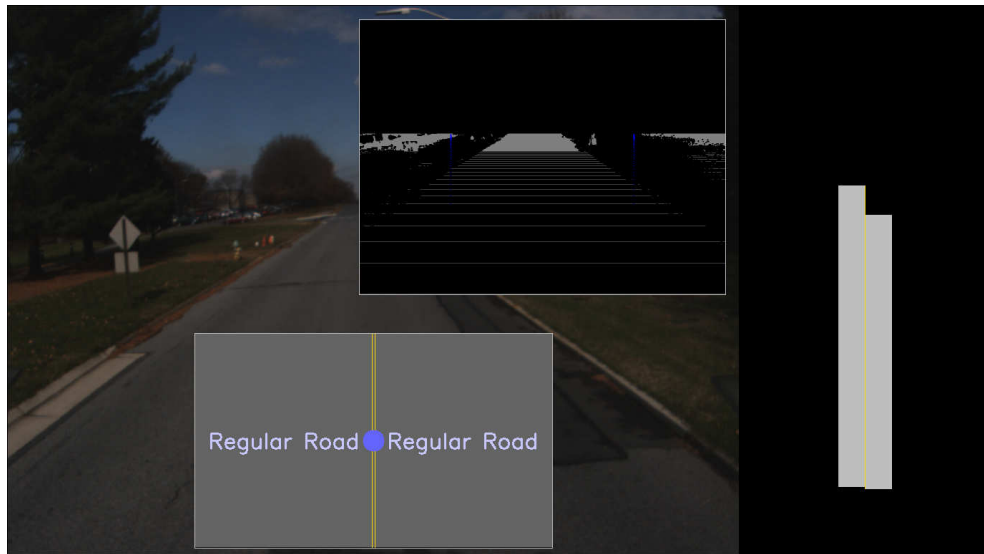
(a)

Internal parameters of search tree



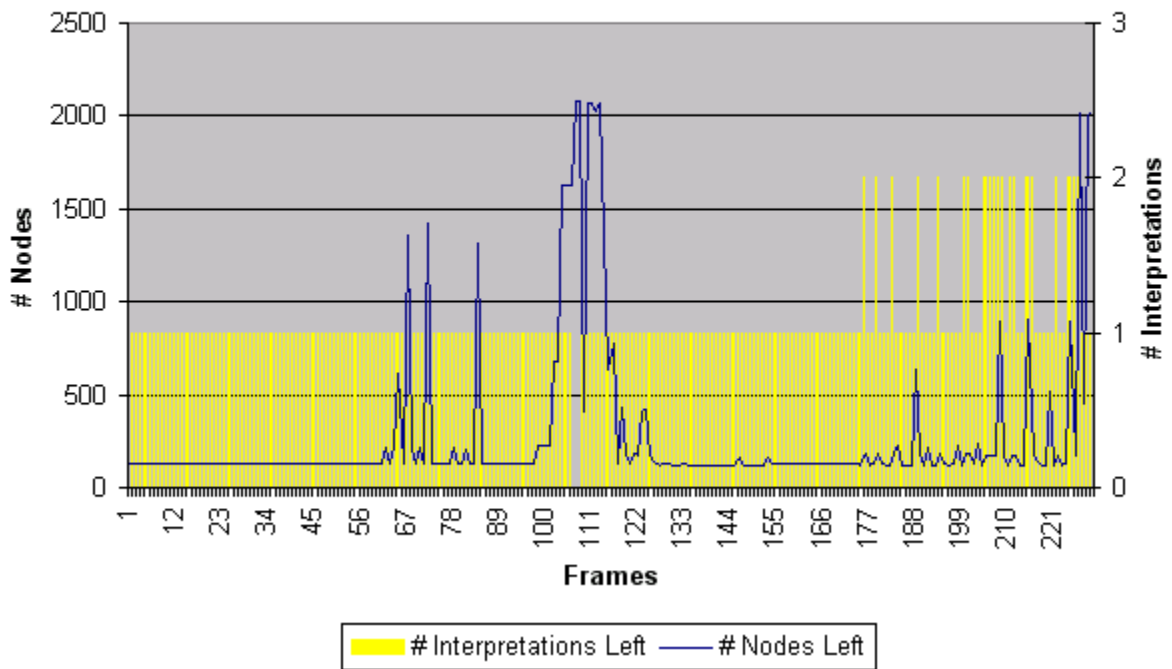
(b)

Fig. 5. (a) First run road recognition result for the first frame of the test sequence. The algorithm erroneously recognized an intersection on the left side of the road. (b) shows the number of nodes (depicted in blue) and the number of interpretations (yellow) for the first run (left road side only).



(a)

Internal parameters of search tree



(b)

Fig. 6. (a) Second run road recognition result for the first frame of the test sequence. There are no wrongly detected intersections anymore. (b) shows the number of nodes (depicted in blue) and the number of interpretations (yellow) for the second run (left road side only).

briefly the NIST Road Network Database, the extraction of ground truth from this database, and performance evaluation results.

A. NIST Road Network Database

In 2004, NIST embarked on an effort to create a Road Network Database (RNDB) structure for the purpose of informing an intelligent vehicle about the structure of the roadway to allow for better path planning and autonomous mobility during on-road driving. This database structure has been represented in a MySQL database [11], documented [9] and populated with detailed instances of roadways on the NIST campus. This section will briefly describe the RNDB and describe how it will be applied to the road recognition approaches described in this paper.

Some of the fundamental components of the Road Network Database are described below:

- *Junctions* - A junction is a generic term referring to two or more paths of transportation that come together or diverge, or a controlled point in a roadway, including lanes splits, forks in the road, merges, and intersections. Junctions are an abstract supertype in the sense that a junction must be one of the types listed above.
- *Intersections* - Intersections are a type of junction in which two or more separate roads come together.
- *Lane Junctions* - A lane junction is a location in a junction in which two or more lanes of traffic overlap.
- *Road* - A road is a stretch of travel lanes in which the name of the travel lanes does not change.
- *Road Segment* - A road segment is a uni-directional stretch of roadway bounded by intersections.
- *Road Element* - A road element is a uni-directional stretch of roadway bounded by any type of junction. Unlike road segments, road elements can be bounded by merging lanes, forks, etc.
- *Lane Cluster* - A lane cluster is a set of uni-directional lanes (with respect to flow of traffic) in which no physical attribute of those lanes change over the span of the lane segment.
- *Lane* - A lane is a single pathway of travel that is bounded by explicit or implicit lane marking. Lanes span the length of a lane cluster of which they are a part.
- *Lane Segment* - A lane segment is the most elemental portion of a road network captured by the database structure. Lane segments can be either straight line or constant curvature arcs. One or more lane segments compose a lane
- *Junction Lane Segments* - A junction lane segment is a constant curvature path through a portion of a lane junction.

For the purpose of road recognition system described in this paper, the two structures that are of most interest are the Road Segment and the Intersection. Figure 7 shows a sample roadway with one of the road segments shaded. There are two intersections shown, represented by black boxes with no lane markings.

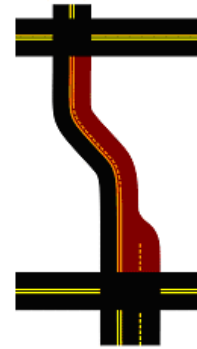


Fig. 7. Sample road network.

The Road Segment database structure contains information such as:

- The road that the road segment is a part of
- The adjacent intersections
- The length of the segment
- The class of road segment (interstate highway, beltway, country road, etc.)

Additional information can also be inferred by looking at other classes that this structure points to, including:

- Beginning and end point of the road segment
- Number of lanes

The database structured has been populated with data from the NIST campus using high-resolution LIDAR scans performed by an external organization. Through post processing, these LIDAR scans were tagged with information about roadways, parking lots, buildings, etc. This information was then converted into the RNDB format and used to populate the database.

Vehicles are localized on this road network using the *Global Positioning System* (GPS) data that is returned from their systems. Although this GPS data is often non-exact, one can still run an algorithm to find the closest road segment to the returned point (this is how off-the-shelf GPS navigation systems work). Since the road segments are defined by their known start and end point, this calculation is relatively trivial.

The Road Segment and Intersection structures in the RNDB correspond nicely to the road and intersection concepts used by the road recognition algorithms. As such, they should provide a nice representation approach for the algorithms.

B. Ground truth Extraction from RNDB

After localizing the vehicle's position within the road network, we need to extract the ground truth for the current frame of the video sequence. Figure 8 depicts the approach: From the vehicle's location and orientation a set of symbols describing the road in front of the vehicle is extracted.

Due to the limitations of the sensor, only parts of this symbolic structure are actually within the field of view. Therefore, we need to prune the structure at the maximum look-ahead distance which is known from camera calibration as being 55

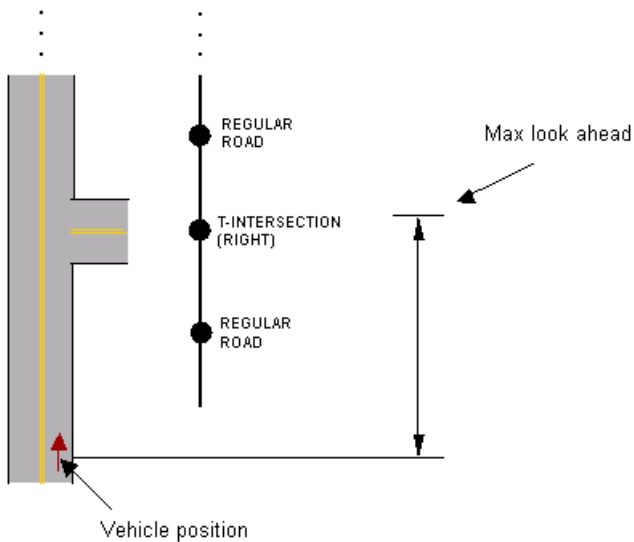


Fig. 8. Simple approach to ground truth extraction.

m. This, however, can only be a coarse estimate of the ground truth because several situations may change the maximum look-ahead distance, e.g.

- whenever the vehicle's orientation is not parallel with the ground, e.g. through tilting due to acceleration, deceleration, or terrain undulations.
- whenever the road's elevation in front of the vehicle differs from the road plane the vehicle sits on.

In the case of the example in Figure 8, the following symbolic road structure could be extracted as ground truth: (*REGULAR ROAD*, *INTERSECTION*). The second symbol (*INTERSECTION*), however, might or might not be part of the actually visible road on sensor data. Such situations require manual correction of the ground truth.

C. Performance Evaluation Results

Figure 9 shows the performance evaluation results for the second run on the video sequence from Section III.

Most of the frames show no classification error at all. The bigger block in the middle of the sequence shows an error of 50 %. This is due to the problem of ground truth generation described in the previous section - the ground truth contains information about an intersection that is actually not yet visible on the sensor data. There are two more peaks in the graph showing a classification error of 100 % (two frames in the middle) and 30 % (two peaks at the end of the sequence). These are good examples for the application of performance evaluation in order to find problematic situations that need further investigation.

In order to compare the performance of the algorithm on video sequences as a whole, we also calculate the minimum, maximum and average classification error throughout the video sequence. This allows for example to compare the performance of different versions (e.g. using different parameters) of the algorithm on the same input data.

The graph in Figure 10 shows the average classification error for the two runs from Section III. The improvement in the second run is reflected by an average error of half the size of the error in the first run.

V. CONCLUSION

We presented in this paper a two-level approach to performance analysis for a new road recognition approach providing symbolic descriptions of the road structure. The first level of performance analysis helps point out potentially problematic areas and real-time issues by analyzing the behaviour of the tree search based recognition approach. A high number of nodes and lack of interpretations in the resulting search tree are considered as indicators for such problematic areas. On the second level an actual performance evaluation is performed by comparing the symbolic results of the algorithm against a (semi-) automatically extracted ground truth. We pointed out situations where a manual correction of the ground truth is necessary. Both methods of performance analysis proved helpful for the ongoing further development of high-level road recognition for on-road driving. In order to allow comparison of different approaches to road sensing, more efforts are needed to bring together worldwide groups and to agree on common grounds for performance analysis in the future.

REFERENCES

- [1] R. Chapuis, R. Aufrere, and F. Chausse. Accurate Road Following and Reconstruction by Computer Vision. *IEEE Trans. on Intelligent Transportation Systems*, 3, 2002.
- [2] D. DeMenthon and L. Davis. Reconstruction of a Road by Local Image Matches and Global 3D Optimization. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1990.
- [3] M. Foedisch, C. Schlenoff, and M. Shneier. Towards an Approach for Knowledge-based Road Detection. In *Proceedings of the ACM Workshop on Research in Knowledge Representation for Autonomous Systems*, pages 1–8, 2005.
- [4] M. Foedisch and A. Takeuchi. Adaptive Real-Time Road Detection Using Neural Networks. In *Proc. of the Intl. Conf. on Intelligent Transportation Systems*, 2004.
- [5] M. Foedisch and A. Takeuchi. Adaptive Road Detection through Continuous Environment Learning. In *Proc. of the Applied Imagery Pattern Recognition Workshop*, 2004.
- [6] W. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, 1990.
- [7] T. Hong, A. Takeuchi, M. Foedisch, and M. Shneier. Performance evaluation of road detection and following systems. In *Proc. SPIE Vol. 5609, Mobile Robots XVII; Douglas W. Gage; Ed.*, 2004.
- [8] M. Luetzeler and E. Dickmanns. Road Recognition with MarVEye. In *Proc. of the IEEE Intl. Conf. on Intelligent Vehicles*, 1998.
- [9] C. Schlenoff, S. Balakirsky, T. Barbera, C. Scrapper, J. Ajot, E. Hui, and M. Paredes. The NIST Road Network Database: Version 1.0. Technical Report Internal Report 7136, NIST, 2004.
- [10] C. Tan, T. Hong, M. Foedisch, T. Chang, and M. Shneier. Performance Analysis of a new Road Following Algorithm Based on Color Models. In *Proc. of the SPIE Defense and Security Symp.*, 2006.
- [11] M. Widenius and D. Azmark. *MySQL Reference Manager*. O'Reilly & Associates, Incorporated, 2002.

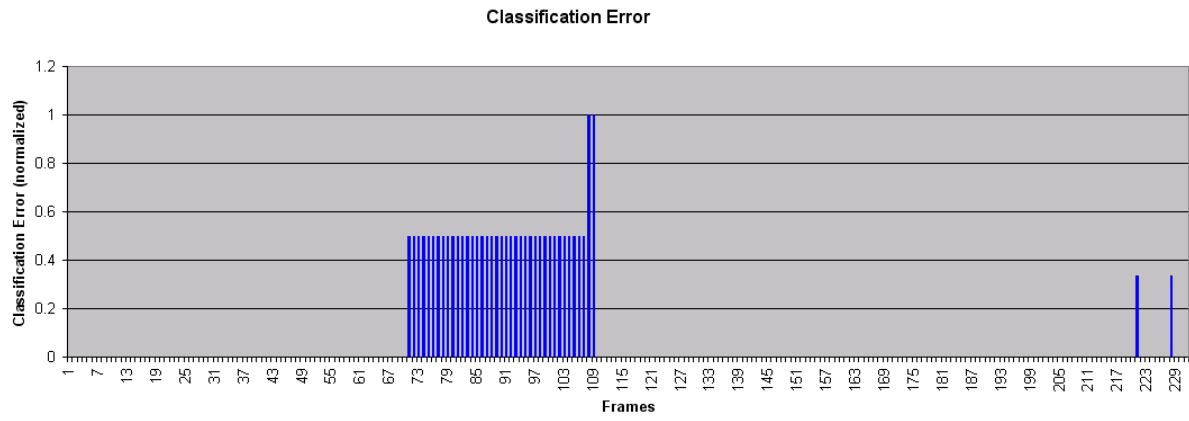


Fig. 9. Performance evaluation results for the second run on the test video sequence.



Fig. 10. Average classification error for the first and second run.

Control of Nonlinear Stochastic Systems: Model-Free versus Classical Controllers

Vural Aksakalli and Daniel Ursu

Abstract— Classical linear controllers are widely used in the control of nonlinear stochastic systems and thus there is concern about the ability of the controller to adequately regulate the system. An alternative approach to cope with such systems is to avoid the need to build the traditional “open-loop” model for the system. Through the avoidance of model, controllers can be built for arbitrarily complex nonlinear systems via neural-networks (NN’s) trained by simultaneous perturbation stochastic approximation (SPSA) so that only the output error (between the plant and target outputs) is needed. We discuss basic characteristics and limitations of both approaches and establish a framework for comparing the two in the control of nonlinear stochastic systems. We formally analyze this comparison in the case of linear quadratic controllers (LQR) and illustrate the comparison numerically on a simulated nonstationary multiple input, multiple output wastewater treatment system with stochastic effects. To the best of our knowledge, a comparison of the model-free approach to classical methods of control has not been done before.

I. INTRODUCTION

MODERN control engineering is expanding rapidly to fill the needs in complex and challenging systems for regulation and control. Such modern systems go well beyond the traditional electrical, mechanical, and aerospace systems that have been at the heart of control systems research for many years. Included in the kinds of modern systems for which control is needed are, to name a few, communications and transportation networks [1], biomedical systems (e.g., automated surgery and drug delivery [2]), and the control of financial markets [3]. Such modern systems do not typically lend themselves to easy representation via linear differential equations. Hence, the majority of the techniques that have been developed over the years to control linear systems may be inappropriate for coping with the control of many modern systems. Furthermore, despite the considerable efforts of many researchers and practitioners over many years, formal control techniques for most real-world nonlinear systems are unavailable [4]. Simply put, closed-form (or other “easy”) solutions to nonlinear problems are almost never available and hence linear methods are generally used. The question one faces

then is whether these linear methods are providing performance that is relatively poorer than possible with other feasible methods for nonlinear systems.

Spall and Cristion [5] make a significant advance in coping with nonlinear, stochastic systems by using neural network based controllers trained via simultaneous perturbation stochastic approximation (SPSA) so that the need to build the traditional open-loop model is avoided. The approach presented therein is based on using the output error of the system to directly train the NN controller without the need for a separate model (NN or other type) for the unknown process equations. Since it is assumed that the system dynamics are unknown, determining the gradient of the loss function in typical back-propagation type weight estimation algorithms is not feasible. To implement such a direct adaptive control, the authors propose simultaneous perturbation stochastic approximation for estimating the NN connection weights while the system is being controlled. In a related work, the authors demonstrate how such a model-free controller can be efficiently utilized to control a challenging nonlinear multiple input, multiple output (MIMO) stochastic wastewater treatment problem [6].

The model-free approach, although relatively new, has already been applied successfully in many real-world control problems. Applications include control of steel making processes [8], robotics [9], human factors systems [10], and bioreactor control [11]. However, a comparison with classical linear methods, theoretical or numerical, has not yet been conducted. The comparison with a classical linear method of control is appropriate as this is a default method given the paucity of usable nonlinear methods. Our goals in this paper are (1) to establish a framework for comparing model-free controllers to classical controllers, (2) formally analyze such a comparison in the case of linear quadratic controllers (LQR), and, (3) illustrate this comparison on an empirical basis in a challenging nonlinear control problem encountered in wastewater treatment systems. Our purpose is to provide some insight into the value of the model-free method and motivate further research in this direction.

The rest of this paper is organized as follows. Section II outlines model-free and classical controllers and briefly discusses their basic characteristics and limitations. Section III establishes general principles for comparing the two approaches. Section IV describes the waste-water treatment system and replicates the model-free controller in [6]. Linear system identification is performed in Section V. Section VI presents minimum variance and LQR controllers for the problem. Section VII summarizes our findings and relates them to those of Spall and Cristion [5,6] and Dochain and

Authors would like to thank Dr. James C. Spall of The John’s Hopkins University’s Applied Physics Laboratory for his constructive comments and suggestions.

V. Aksakalli is with the Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: ala@jhu.edu).

D. Ursu is with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: dursu1@jhu.edu).

Bastin [7]. Directions for future research are discussed in Section VIII.

II. MODEL-FREE & CLASSICAL CONTROLLERS: BASIC CHARACTERISTICS AND LIMITATIONS

A. Model-Free Controllers

We consider a discrete-time state vector given by

$$x_{k+1} = \phi_k(x_k, u_k, w_k), \quad (2.1)$$

where ϕ_k is a nonlinear, yet *unknown* function governing the system dynamics, u_k is the control input applied to the system at time k , and w_k is a random noise vector. Our focus will be on the case of direct measurements as in [5] and [6]. The goal here is to determine the control vectors $\{u_k\}$ such that the state values $\{x_k\}$ are as close as possible to a set of target vectors $\{t_k\}$. The information fed into the NN controller consists of the next target vector, M most recent state values, and N most recent controls. The output of the NN is then the value of the control u_k . Associated with this NN is a vector of connection weights $\theta_k \in \mathbf{R}^p$ that will be estimated. Our goal is to find θ_k^* that minimizes some loss function $L(\theta)$ measuring system performance. We will use the one-step-ahead quadratic tracking error below as the performance criterion:

$$L_k(\theta_k) = E \left[(x_{k+1} - t_{k+1})^T W_k (x_{k+1} - t_{k+1}) + u_k^T Z_k u_k \right] \quad (2.2)$$

where W_k and Z_k are positive semi-definite matrices specifying the relative weights on the deviations from the target values and the cost of large control values.

To find the optimal values of θ_k , the model-free controller uses a stochastic approximation of the form

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \hat{g}_k(\hat{\theta}_{k-1}), \quad (2.3)$$

where $\hat{g}_k(\hat{\theta}_{k-1})$ is the simultaneous perturbation approximation to $g_k(\hat{\theta}_{k-1}) = \partial L_k / \partial \theta_k$. The reader is referred to [1] for an in-depth discussion of the SPSA-based NN controllers. We now briefly discuss basic characteristics and limitations of this approach.

1. The use of “model-free” is to be taken literally in the sense that no hidden or implicit modeling is required, which eliminates the *system characterization and identification* processes, and thus the need to allocate time and resources to determine an adequate model of the underlying system and evaluate its validity.
2. Three major advantages of the model-free controllers are that they (i) tend to better handle changes in the underlying system as they are not tied to a prior model, (ii) require no open-loop training data, and (iii) tend to be more robust in the case of widely varying control inputs.
3. The model-free approach is appropriate for many practical systems, yet it is generally inappropriate for

systems where a reliable model of the system can be determined. This is primarily due to the fact that a controller designed using a reliable model will usually achieve optimal performance more quickly; also allowing theoretical analysis of issues such as stability and controllability in some cases. However, for systems where only a flawed (if any) model is available, such control approaches and analyses can lead to significantly suboptimal controllers and inaccurate stability and controllability analyses. It is such cases that the model-free approach should be considered. Nevertheless, partial prior knowledge of the system *can* be incorporated into the model-free framework via self-tuning methods for enhanced performance [5].

4. The model-free controller requires that the system under consideration be approximately stationary while an individual SPSA approximation is performed (the system dynamics can be nonstationary over longer time periods, however). A further restriction (which is typical of controllers relying on imperfect prior system knowledge) is that the system be able to tolerate suboptimal controls as the learning process takes place.
5. Success of the model-free approach in any particular application depends on the choice of the NN structure (e.g., number of hidden layers and nodes per layer, activation functions, number of prior state and target values used, etc.), and SPSA implementation methodology (such as gain sequence structures, gradient approximation averaging, smoothing, etc). Such issues need to be carefully addressed in practice for an effective implementation of this approach.

B. Classical Controllers

Discrete-time MIMO linear time-invariant systems are defined as $x_{k+1} = Ax_k + Bu_k$ (assuming direct state measurements); where A and B are matrices determined via a system identification process. Fundamental characteristics and limitations of classical linear controllers are briefly discussed below.

1. These controllers are widely used in practice due to their simplicity and availability of corresponding software tools and commercial products.
2. Given a nonlinear system, classical controllers can be used only on a “linearized” version of the system, giving good results at an equilibrium point about which the system behavior is approximately linear. However, this assumption of linearity is usually violated to a certain degree in many of today’s complex control systems.
3. Such controllers show poor and/or inadequate performance when process and/or measurement noise is present and in the case where the system varies in time.

III. A COMPARISON FRAMEWORK

The following general principles should be considered for a framework that compares a model-free controller to a particular classical controller on an empirical basis:

1. Both controllers first need to be fine-tuned for optimal performance associated with the range of problem instances under consideration (see [5,6,11,12] for model-free controller implementation guidelines and [13,14,15] for classical controllers). Both controllers should be evaluated under the same performance criteria to the extent possible.
2. Specifically, SPSA gain sequences need to be fine-tuned carefully. These sequences should satisfy certain regularity conditions [5,6]. If the system dynamics or the loss function is changing, constant gain coefficients should be used.
3. In case no model of any kind is available for designing a classical linear controller, one can simply perform linear regression on a set of open-loop training data [16, Chap. 7]. The linear model should also be evaluated to ensure that no significant violations of linearity assumption exist. However, if a reliable nonlinear model is available, then a simple first-order Taylor series expansion can be carried out for practical linearization.

IV. THE WASTEWATER TREATMENT PROBLEM AND THE MODEL-FREE CONTROLLER

The wastewater problem is described in [6] as follows: influent wastewater is first mixed (as determined by a controller) with a dilution substance to provide a mixture with a desired concentration of contaminants. This diluted mixture is then sent to a second tank at a controlled flow rate. In the second tank the mixture goes through an anaerobic digestion process, where the organic material in the mixture is converted by bacteria into byproducts such as methane. Therefore, the system consists of two controls (the mix of wastewater/dilution substance and the input flow rate) and two states (an effluent de-polluted water and methane gas, which is useful as a fuel). Since this system relies on biological processes, the dynamics are nonlinear and usually time-varying. Also, the system is subject to constraints (e.g., the input and output concentrations, the methane gas flow rate, and the input flow rate all must be greater than zero), which presents an additional challenge in developing a controller for the system.

The unknown process equations are assumed to be

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \begin{pmatrix} 1 + \mu_k T & 0 \\ -.3636T & 1 \end{pmatrix} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \begin{pmatrix} -Tx_{k,1} & 0 \\ -Tx_{k,2} & T \end{pmatrix} \begin{pmatrix} u_{k,1} \\ u_{k,1}u_{k,2} \end{pmatrix} + \begin{pmatrix} w_{k,1} \\ w_{k,2} \end{pmatrix} \quad (4.1a)$$

where the bacterial growth rate μ_k is given by

$$\mu_k = \frac{(.425 + .025 \sin(2\pi k/96))x_{k,2}}{.4 + x_{k,2}} \quad (4.1b)$$

where

- x_1 is the methane gas flow rate,
- x_2 is the substrate output concentration,
- u_1 is wastewater/dilution substance mix rate,
- u_2 is the input flow rate, and
- T is the sampling interval, which is .5 seconds.

We now replicate the problem environment and the model-free controller in [6]. The target sequence t_k is a periodic square wave with values (.97, .2) for the first 48 updates and (1, .1) for the second 48 updates¹. We assume independent noise terms $w_{k,1}$ and $w_{k,2} \sim N(0, \sigma^2 I)$ where $\sigma = .01$. The initial state is assumed to be $x_0 = (.5, 1.6375)$.

Note that the model-free controller has *no knowledge* of the above equations. The performance criteria used is the weighted root-mean-square (RMS) measurement:

$$\left[(x_{k+1} - t_{k+1})^T \mathbf{W} (x_{k+1} - t_{k+1}) + u_k^T \mathbf{Z} u_k \right]^{1/2} \quad (4.2)$$

with $\mathbf{W} = \text{diag}(.01, .99)$ and $\mathbf{Z} = \text{diag}(.001, .001)$ where $\text{diag}()$ denotes the square matrix whose diagonal entries are the given parameters and all off-diagonal entries are zero. Notice that (4.2) corresponds to the loss function (2.2) with the matrix $\mathbf{W}_k = \mathbf{W}$ and $\mathbf{Z}_k = \mathbf{Z}$. The values .01 and .99 reflect the relative emphasis of the controller on methane production and water purity, respectively. The control gains, on the other hand, are weighted less compared to deviations from the target values. The NN used contains two hidden layers with 20 nodes in the first hidden layer and 10 nodes in the second. All the hidden nodes have the scaled logistic function as the activation function. The inputs to the NN are the current and most recent states, the most recent control, and the target vector for the next state, yielding a total of eight input nodes. The output of the NN is then the next control values. Thus, there are a total of 412 weights in the NN, which will be updated by SPSA at each iteration. These weights are initialized with random values in [-.1, .1]. As for the SPSA implementation, a two-sided SPSA with constant gains is used (since the system is time-varying) where $a = .2$ and $c = .1$. SPSA is implemented without any gradient approximation averaging or smoothing.

Figures 2a and 2b show the state values versus target values for each target-state pair when the model-free controller is used. Notice that due to the relative importance of x_2 , x_1 is tracked with much less accuracy. It can yet be seen that x_2 is tracked quite closely. The discrepancy between the tracking errors of the two state variables is not just a result of the weight emphasis we have put on the RMS loss function via matrix \mathbf{W} . In fact, this discrepancy is built into the control system proposed by Dochain and Bastin [7], whose research also showed that the system exhibits

¹ Target values for x_2 in the first 48 updates are .13 in [2]. We use .2 to better illustrate the model-free controller's tracking capabilities.

preferential tracking of one state variable over the other. From a physical perspective, this can be explained by the wastewater system having been designed to prioritize the de-polluted water output over that of methane gas through the parameters proposed in [6]. Indeed, changing the weights of the weight matrix \mathbf{W} offers slightly different results, but not by much, regardless of the weights used.

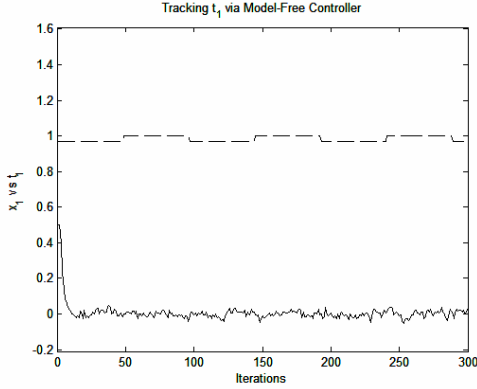


Fig. 2a: Model-free controller: tracking x_1 (solid line) versus t_1 (dashed line)

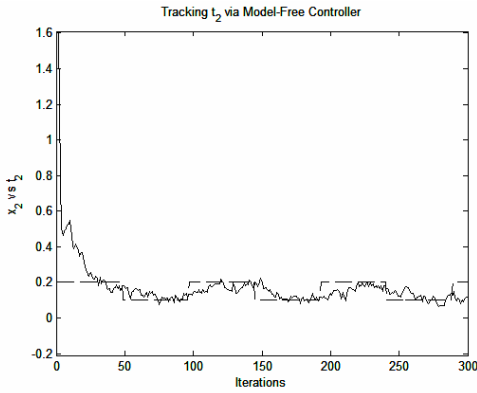


Fig. 2b: Model-free controller: tracking x_2 (solid line) versus t_2 (dashed line)

V. LINEAR SYSTEM IDENTIFICATION

Our goal is to design classical controllers for the wastewater treatment for comparison with the model-free controller. Thus, we first attempt to construct a linear model that adequately captures the relationship between the control inputs and system outputs. If the linear model predicts this relationship reasonably well, then the design and use of relatively simple classical linear controllers would be justified.

Due to the existence of the k term in (4.1b), a multivariate Taylor series expansion is not feasible. Therefore, we perform the system identification task via linear regression in two steps: collecting the data from which a model will be constructed and constructing an appropriate model from this data.

For data collection, open-loop training with random inputs was performed where the bounds on the control inputs are $[.09, .4]$ for u_1 and $[1.5, 3.0]$ for u_2 (as in [7]); with the system initialized at $x_0 = (.5, 1.6375)$. We generated

300 random controls within those bounds and evaluated the noisy state values when the process is subject to these controls, obtaining 300 random input-output pairs. Having generated the data, we fitted a first-order linear time invariant auto-regressive (ARX) model, which is given by

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \mathbf{B} \begin{pmatrix} u_{k,1} \\ u_{k,2} \end{pmatrix} \quad (5.1)$$

where the 2×2 matrices \mathbf{A} and \mathbf{B} are estimated using least squares regression. We chose the first-order model since it is simple and increasing the order did not significantly increase the model quality. For model evaluation, we computed the R^2 statistic associated with the regression, which revealed to be .98 for both x_1 and x_2 . Thus, a first-order linear model is quite good even though the underlying system is stochastic and nonlinear, which indicates validity of designing classical linear controllers for the wastewater problem. The least squares regression resulted in the following linear model:

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \begin{pmatrix} 1.0333 & .0907 \\ -.1786 & .8924 \end{pmatrix} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \begin{pmatrix} -.5204 & -.0007 \\ .7851 & .1172 \end{pmatrix} \begin{pmatrix} u_{k,1} \\ u_{k,2} \end{pmatrix} \quad (5.2)$$

VI. CLASSICAL CONTROLLERS FOR THE WASTEWATER TREATMENT PROBLEM

A. Minimum Variance Controller

We first attempt to design a minimum variance controller (MVC) for the wastewater treatment problem. The goal in MVC is to minimize one-step-ahead error. Thus, we would like the system outputs x_1 and x_2 (which are also the system's states) to match our target sequences t_1 and t_2 . In other words, $x_{k+1} = t_{k+1}$ with $x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k$; where \mathbf{B} is assumed to be nonsingular. Solving for u_k , we get

$$u_k^{mv} = \mathbf{B}^{-1} (t_{k+1} - \mathbf{A}x_k^{mv}) \quad (6.1)$$

Thus, the MVC changes the current input as a function of the one-step-ahead target output. The newly obtained u_k^{mv} and x_k^{mv} are then substituted into the original control system equation, yielding $x_{k+1} = \mathbf{A}x_k^{mv} + \mathbf{B}u_k^{mv}$. After implementing the MVC as described, we observed that this controller diverges very quickly. Dochain and Bastin [7] also attempt to design a MVC for their single-input single-output version of the same problem and their MVC also diverges, i.e, the system gains are not proper for following the target outputs t_1 and t_2 . Thus, we need to try alternative classical methods for an efficient control of this system.

B. Linear Linear Quadratic Regulator Controller (LQR)

Similar to model-free controllers, linear quadratic controllers involve minimization of a loss function measuring the difference between the system's outputs and target outputs. The performance criterion used in LQR is the following quadratic loss function:

$$J = \sum_{k=1}^{K-1} \begin{bmatrix} e_k^T & v_k^T \end{bmatrix} \mathbf{Q} \begin{bmatrix} e_k \\ v_k \end{bmatrix} + u_k^T \mathbf{R} u_k \quad (6.2)$$

Above, e_k is the control error (i.e., $e_k = x_k - t_k$), v_k is the cumulative error ($v_k = \sum_{i=1}^{k-1} e_i$), K is the number of iterations, and u_k is the control input. The goal is to determine the control sequence $\{u_k\}$ such that J is minimized. The matrices \mathbf{Q} and \mathbf{R} reflect the relative weights of control errors and the control gains. The above criterion poses interest to us because of its similarity to (2.2); the performance criterion of the model-free controller. The implementation of this algorithm is as follows [13, Chap. 8]: the loss function is first rewritten as

$$J = \sum_{k=0}^{K-1} [x_k^T \mathbf{P} x_k] \quad (6.3)$$

where \mathbf{P} is defined as the optimal steady-state matrix. For a linear system described by $x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k$, \mathbf{P} is given by

$$\begin{aligned} \mathbf{P} &= \mathbf{Q} + \mathbf{A}^T \mathbf{P} (\mathbf{I} + \mathbf{B} \mathbf{R}^T \mathbf{B} \mathbf{P})^{-1} \mathbf{A} \\ &= \mathbf{Q} + \mathbf{A}^T (\mathbf{P}^{-1} + \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T)^{-1} \mathbf{A} \end{aligned}$$

The above Riccati equation is solved iteratively until \mathbf{P} no longer changes values. The above expression further shows that \mathbf{P} is solely dependent on the state-space matrices \mathbf{A} and \mathbf{B} , and the matrices \mathbf{Q} and \mathbf{R} associated with the loss function. The steady-state gain matrix \mathbf{K} can then be written in terms of \mathbf{P} as:

$$\mathbf{K} = (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \quad (6.4)$$

The matrix \mathbf{K} optimizes the actual input, so the control law that minimizes (6.2) becomes:

$$u_k = -\mathbf{K}x_k \quad (6.5)$$

Once fed back into the original control problem, the control system can be stated as follows using (6.4) and (6.5):

$$x_{k+1} = [\mathbf{A} - \mathbf{B}(\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}] x_k \quad (6.6)$$

C. Model-Free Controllers versus LQR: A Formal Analysis

The LQR differs from the model-free controller in the sense that the former assumes a modeled control process whereas the latter does not. However, contrasting (6.2) to (2.2), it can be seen that both controllers attempt to minimize similar loss functions in an iterative manner. We now formalize this connection between the two controllers.

The objective in the model-free framework is to determine the control vector u_k that minimizes the one-step-ahead tracking error where $x_{k+1} = \phi_k(x_k, u_k, w_k)$. Assuming that the model-free framework uses constant gain and error matrices as in (4.2), we have

$$u_k^{mf} = u_k(\theta_k^*; x_k, x_{k-1}, \dots, x_{k-M+1}; u_{k-1}, u_{k-2}, \dots, u_{k-N+1}; t_{k+1}),$$

where

$$\theta_k^* = \arg \min_{\theta_k} \{(x_{k+1} - t_{k+1})^T \mathbf{W} (x_{k+1} - t_{k+1}) + u_k^T \mathbf{Z} u_k\}.$$

With a slight abuse of notation, we shall write

$$u_k^{mf} = \arg \min_{u_k} \{(x_{k+1} - t_{k+1})^T \mathbf{W} (x_{k+1} - t_{k+1}) + u_k^T \mathbf{Z} u_k\}. \quad (6.6)$$

Let the time-invariant linearization of ϕ be $\hat{x}_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k$ where \mathbf{A} and \mathbf{B} represent the linear system analogous to that of the LQR control. Define linear approximation residuals as

$$\begin{aligned} r_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= \phi_k(x_k, u_k, w_k) - (\mathbf{A}x_k + \mathbf{B}u_k) \end{aligned} \quad (6.7)$$

The control error e_{k+1} can now be written as

$$\begin{aligned} e_{k+1} &= x_{k+1} - t_{k+1} \\ &= \hat{x}_{k+1} + r_{k+1} - t_{k+1} \end{aligned}$$

Let $\hat{e}_{k+1} = \hat{x}_{k+1} + r_{k+1}$, which implies $e_{k+1} = \hat{e}_{k+1} - t_{k+1}$. Thus, equation (6.6) yields

$$\begin{aligned} u_k^{mf} &= \arg \min_{u_k} \{(\hat{e}_{k+1} - r_{k+1})^T \mathbf{W} (\hat{e}_{k+1} - r_{k+1}) + u_k^T \mathbf{Z} u_k\} \\ &= \arg \min_{u_k} \{\hat{e}_{k+1}^T \mathbf{W} \hat{e}_{k+1} + r_{k+1}^T \mathbf{W} r_{k+1} + 2\hat{e}_{k+1}^T \mathbf{W} r_{k+1} + u_k^T \mathbf{Z} u_k\} \end{aligned} \quad (6.8)$$

Now, to establish a fair comparison between the model-free controller and the LQR controller, let $\mathbf{R} = \mathbf{Z}$ and $\mathbf{Q} = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix}$ in (6.2) where \mathbf{V} specifies the relative weight of cumulative errors. Thus, the LQR control law can be expressed as:

$$\begin{aligned} \{u^{LQR}\} &= \arg \min_{\{u_k\}} \sum_{k=1}^{N-1} \left\{ \begin{bmatrix} \hat{e}_k^T \hat{v}_k^T \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \hat{e}_k \\ \hat{v}_k \end{bmatrix} + u_k^T \mathbf{Z} u_k \right\} \\ &= \arg \min_{\{u_k\}} \sum_{k=1}^{N-1} \{ \hat{e}_{k+1}^T \mathbf{W} \hat{e}_{k+1} + \hat{v}_{k+1}^T \mathbf{V} \hat{v}_{k+1} + u_k^T \mathbf{Z} u_k \} \end{aligned} \quad (6.9)$$

Comparing (6.8) to (6.9), we observe that the model-free controller performs minimization *at each iteration*, whereas LQR performs a *single* minimization over all the iterations; each with respect to their individual loss functions. This particular phenomenon is rather a design issue. Whether the control engineer chooses to minimize error at each iteration or prefers minimizing the total sum of errors over the entire control horizon for a given particular control problem depends rather on the nature of the system being controlled and/or the specific goals of the control process.

Now, suppose that the system under consideration is linear and time-invariant with no impact of noise. In other words, $\phi_k(x_k, u_k, w_k) = \mathbf{A}x_k + \mathbf{B}u_k$. Let $\bar{L}_k := e_{k+1}^T \mathbf{W} e_{k+1} + u_k^T \mathbf{Z} u_k$ and $\mathbf{V} = \mathbf{0}$. Thus, $r_{k+1} = 0$ and $\hat{e}_{k+1} = e_{k+1}$ for all k , which yields

$$\begin{aligned} u_k^{mf} &= \arg \min_{u_k} \bar{L}_k, \\ \{u^{LQR}\} &= \min_{\{u_k\}} \sum_{k=1}^{N-1} \bar{L}_k. \end{aligned}$$

Thus, in the case of a linear system without any stochastic effects, the residual terms in (6.8) vanish and the loss function of the model-free controller becomes *equivalent* to that of the LQR controller with $\mathbf{V} = \mathbf{0}$. That is, both the model-free and LQR controllers would be minimizing the same loss function, where the model-free controller again would be executed at each iteration, while the LQR would be executed over all the iterations. Furthermore, assuming

that all the target values are physically realizable and the NN structure in the model-free controller is capable of representing linear systems without any approximation errors, both controllers would yield the *same* control inputs, i.e., these two controllers would essentially be equivalent. Also notice that, in this particular case, the model-free controller would interestingly become a minimum variance controller as in Section VI.A.

Note that a fundamental advantage of the model-free controller in general is that it requires only one-step-ahead target values, as opposed to LQR that requires a priori knowledge of the *entire* target sequence; which is a desirable feature in real-time control.

D. LQR for the Wastewater Treatment Problem

In our LQR implementation, we chose $\mathbf{Q} = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix}$

with $\mathbf{W} = \text{diag}(.01, .99)$ and $\mathbf{V} = (.01)\mathbf{W}$ (placing more emphasis on the control errors relative to the accumulated control errors) and $\mathbf{R} = \text{diag}(.001, .001)$. Notice that our choice of \mathbf{Q} and \mathbf{R} coincides with the model-free controller loss function for a fair comparison. Figures 8a & 8b illustrate the LQR controller performance.

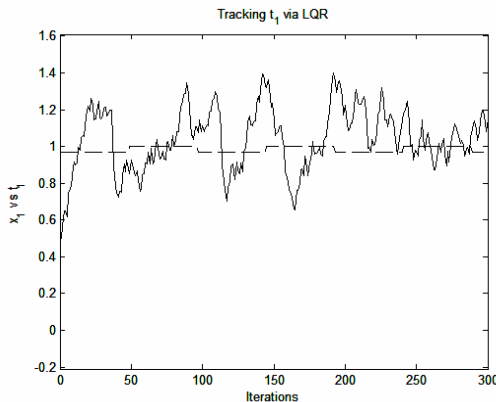


Fig. 8a: LQR controller: tracking x_1 (solid line) versus t_1 (dashed line)

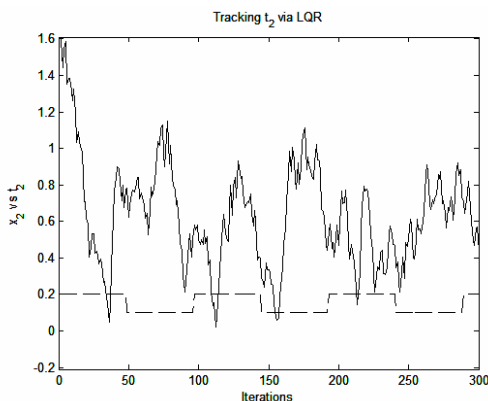


Fig. 8b: LQR controller: tracking x_2 (solid line) versus t_2 (dashed line)

As the above figures show, LQR outputs follow the target outputs to some degree, but not very closely; with t_1 being tracked somewhat better than t_2 . We tried fine-tuning the \mathbf{Q}

and \mathbf{R} matrices, yet did not observe any significant performance improvements. In fact, the LQR controller exhibited preferential tracking for different target values that we tried, with t_1 being tracked better than t_2 and vice versa. We hypothesize that this occurs because the matrices that define the loss function J were not fine-tuned throughout the simulation process, as the input kept oscillating. That would have assumed a controller of the *adaptive* type, and is beyond our scope due to the fact that the model-free controller is not of the adaptive type either. Moreover, we possess no intuition as to how the \mathbf{Q} and \mathbf{R} matrices should be automatically updated as a function of changing target values.

Analysis of the LQR controller output reveals that the state values are in the range of the target values, yet x_2 is a lot more amplified than x_1 . However, the LQR controller still behaves in a far worse fashion than the model-free controller. Since we have formulated both controllers to minimize similar loss functions, the difference between the behaviors of the two controllers can be attributed to the way each controller handles iteration error. The model-free controller updates itself after each iteration, thereby keeping the error between input and output to a minimum. On the contrary, the LQR algorithm sums the error over the whole simulation process, only attempting minimization at the very end. It is precisely this buildup in error that prevents the LQR controller from tracking as well as the model free SPSA controller. This may be particular to the nature of this MIMO system and how disturbances in one state variable affect the other state variable through relationships to be found in the system's state space. Therefore, in this application, the inability of the LQR controller to compensate for the error (between the actual output and target output) quickly enough actually penalizes it and forces the tracking to deteriorate.

VII. SUMMARY & CONCLUSIONS

In this paper, we discuss model-free and classical controllers for the control of nonlinear stochastic systems and briefly describe their basic characteristics and limitations. We present a comparison framework and a limited formal comparison in the case of LQR controllers. Specifically, we show that both controllers are governed by the same mathematical models; the difference being the way each controller handles error propagation. Furthermore, given a previously defined wastewater treatment problem by Dochain and Bastin [7], and a solution to the MIMO control of this system implemented by Spall and Cristion [6] through SPSA, we attempt to solve this problem through the use of classical control theory as well. We have found that given a well-defined linear quadratic regulator (LQR), we can achieve selectively good tracking of this problem as a coupled MIMO system. This implies that there is a class of controllers for which this problem is stable.

The implementation of the LQR controller for this

problem allowed us to study the interesting coupling between the two states of the system and observe some interesting comparisons between LQR and model-free controllers. These comparisons are more insightful because both controllers incorporate a minimization function, which tailors their respective outputs accordingly. However, LQR has one disadvantage over the model-free controller in that it is model based and thus constrained by the values of the state equations by which the model is described. Furthermore, it was observed that the summation of error on behalf of the LQR controller actually makes it perform worse than the model-free controller, which looks at error at every iteration of the system. This gives the model-free controller the flexibility to adapt to changes in a monitored system, its only limit being the definition of its loss function. A comparison of the two algorithms shows that both choose to track this MIMO system preferentially; that is, tracking of a certain variable is prioritized over the tracking of the other. However, each algorithm “chooses” to do so differently. The model-free approach matches the gains of the desired outputs but offsets one of them at the cost of following the other. The LQR regulator matches the overall value of the outputs well, but “chooses” to have smaller steady state error on one output, at the cost of the other, again an effect of error summation rather than adaptation per iteration.

Another distinct advantage of the model-free controller is that it can attempt to control systems whose internal processes cannot be observed because of real world constraints. The model-free controller will assume a solution as long as it is mathematically possible. While this is advantageous to the designer, it is a tool that must be used carefully. In control systems design, the state equations are designed based on measured parameters of sensors and the physical properties of the components. Thus a user of the model-free controller will have to choose the cost function for this algorithm very wisely, to make sure that, if used in a design tool, certain physical properties are met, such as controllability and stability of a physical system. But we cannot conclude here without mentioning that to some extent the LQR regulator has the same type of drawbacks. Although it is initially model based, the gains “ \mathbf{K} ” that it chooses to minimize a loss function are arbitrary, and it may very well be that in the real world, some of those gains are unattainable. So likewise, care must be chosen in its implementation.

VIII. DIRECTIONS FOR FUTURE RESEARCH

The comparison of the model-free controller to LQR can be extended to formally account for stochastic effects and/or incorporate linearization error for certain classes of nonlinear dynamical systems. The model-free controller can further be compared to other methods of classical control, such as linear quadratic Gaussian (LQG) controllers or control via pole-placement. In addition, the model-free controller can be compared to neural network-based

controllers as in [17], which would provide significant information on the relative value of utilizing truly model-free controllers versus first constructing a neural network representation of the system being controlled. These comparisons should be made generically to the extent possible, including effects of process and/or measurement noise.

REFERENCES

- [1] Friesz, T. L., Luque, J., Tobin, R. L., Wie B-W., “Dynamic network traffic assignment considered as a continuous time optimal control problem”, *Operations Research*, 37(6), pp. 893-901.
- [2] Taylor, H. R., Funda, J., Eldridge, B., Gomory, S., Gruben, K., LaRose, D., Talamini, M., Kavoussi, L., Anderson, J., “A telerobotic assistant for laparoscopic surgery”, *IEEE Engineering in Medicine and Biology Magazine*, 14(3), pp. 279-288.
- [3] Phillips, D. J., *Quantitative Analysis in Financial Markets*, NJ: Hackensack, World Scientific, 1999.
- [4] Khalil, H. K., *Nonlinear Systems*, 3rd Ed., NJ: Englewood Cliffs, Prentice Hall, 2002.
- [5] Spall, J. C. and Cristion, J. A., “Model-free control of nonlinear systems with discrete time measurements,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 1198–1210, 1998.
- [6] Spall, J. C. and Cristion, J. A., “A neural network controller for systems with unmodeled dynamics with application to wastewater treatment,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, pp. 369–375, 1997.
- [7] Dochain, D. and Bastin, G., “Adaptive identification and control algorithms for nonlinear bacterial growth systems,” *Automatica*, vol. 20, pp. 621–634, 1984.
- [8] Sadegh, P. and Spall, J. C., “Optimal sensor configuration for complex systems”, *Proceedings of the American Control Conference*, 1998.
- [9] Maeda, Y. and De Figueiredo, R. J. P., “Learning rules for neuro-controller via simultaneous perturbation”, *IEEE Transactions on Neural Networks*, vol. 8, pp. 1119–1130, 1997.
- [10] Song, J., Xu, Y., Yam, Y., and Nechyba, M. C., “Optimization of human control strategy with simultaneously perturbed stochastic approximation”, *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, 1998.
- [11] Vande Wouwer, A., Renotte, C., Bogaerts, P., and Remy, M., “Application of SPSA techniques in nonlinear system identification”, *European Control Conference*, 2001.
- [12] Ji, X.D. and FAMILONI, B.D., “A diagonal recurrent neural network-based hybrid direct adaptive SPSA control system,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 1469-1473, 1999.
- [13] Ogata, K., *Discrete-Time Control Systems*, 2nd Ed., NJ: Englewood Cliffs, Prentice Hall, 1987.
- [14] Astrom, J.K., *Computer Controlled Systems*, NJ: Englewood Cliffs, Prentice Hall, 1990.
- [15] Franklin, G.F., Powell, J.D., and Workman, M.L., *Digital Control of Dynamic Systems*, 3rd Ed., NJ: Englewood Cliffs, Prentice Hall, 1997.
- [16] Ljung, L., *System Identification: Theory For the User*, 2nd Ed., NJ: Englewood Cliffs, Prentice Hall, 1998.
- [17] Narendra, K. S. and Parthasarathy, K., “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, vol. 1, pp.4-26, 1990.

Challenges in Autonomous System Development

J. Connelly
Institute for Defense Analyses,
4850 Mark Center Drive,
Alexandria, VA, USA
jconnell@ida.org

W.S. Hong
Institute for Defense Analyses,
4850 Mark Center Drive,
Alexandria, VA, USA
whong@ida.org

R.B. Mahoney, Jr./D.A. Sparrow
Institute for Defense Analyses,
4850 Mark Center Drive,
Alexandria, VA, USA
rmahoney@ida.org
dsparrow@ida.org

Abstract—The field of autonomous vehicles sits at the intersection of artificial intelligence (AI) and robotics, combining decision-making with real-time control. Autonomous vehicles are desired for use in search and rescue, urban reconnaissance, mine detonation, supply convoys, and more. The general adage is to use robots for anything dull, dirty, dangerous or dumb. While a great deal of research has been done on autonomous systems, there are only a handful of fielded examples incorporating machine autonomy beyond the level of teleoperation, especially in outdoor/complex environments. In an attempt to assess and understand the current state of the art in autonomous vehicle development, a few areas where unsolved problems remain became clear. This paper outlines those areas and provides suggestions for the focus of science and technology research. The first step in evaluating the current state of autonomous vehicle development was to develop a definition of autonomy. A number of autonomy level classification systems were reviewed. The resulting working definitions and classification schemes used by the authors are summarized in the opening sections of the paper. The remainder of the report discusses current approaches and challenges in decision-making and real-time control for autonomous vehicles. Suggested research focus areas for near-, mid-, and long-term development are also presented.

I. INTRODUCTION

A. Definition of Autonomy

What is autonomy? According to Webster [1], it is “the quality or state of being self-governing”. However, in the field of autonomous vehicles and military applications, autonomy is usually thought of as something more synonymous with “independence” or “intelligence”.

The official Department of Defense (DoD) definition of “autonomous operation”, from the DoD Dictionary of Military Terms, provides an interesting perspective on the concept and separates it somewhat from just autonomous vehicles:

“In air defense, the mode of operation assumed by a unit after it has lost all communications with higher echelons. The unit commander assumes full responsibility for control of weapons and engagement of hostile targets.” [2]

This definition also highlights the fact that autonomy does not apply only to machines, but is already a working concept within the military chain of command. Therefore, when considering autonomy, the terms “Authority” and “Agent” instead of “human” and “computer” are suggested. In this way, the discussions are not limited to the hierarchy as it is

currently envisioned.

One interesting characterization of autonomy found was “[autonomy] is whatever we don’t know how to do yet. Once we know how to do it, we call it an algorithm.”¹ In fact, this is more widespread today than generally realized. Some functions taken for granted in cars or planes today make and execute decisions independently and thus may be considered autonomous subsystems, e.g. optimization of fuel and battery power consumption ratios in hybrid vehicles, air bags, and anti-lock brakes. However, because the whole car is not autonomous, there is a tendency to minimize the successes that have been attained thus far, and characterize them as “automatic” rather than “autonomous”.

What is the difference between “automatic” and “autonomous”? One distinction may be to say that something automatic has only one “choice” between two possible states, e.g. ‘on’ or ‘off’. Another classification would say that automatic systems take in only one input for making the decision. In either case, current air bags and anti-lock brakes would likely fall in the “automatic” instead of “autonomous” category. Autonomous systems could then be ones that process multiple inputs before acting, e.g. a braking system that considers both wheel slippage and speedometer measurements and only deploys if the car is traveling faster than 30mph. Alternatively, autonomous systems may be those that have more than two possible states, and so have to make more than an “on/off” choice.

The relative merits of differing distinctions between “autonomous” and “automatic” are hard to measure—there are continuing debates and the presence of counterexamples in any classification system or definition proposed to date. If the line between “automatic” and “autonomous” is drawn based on number of choices or whether the system is following rules instead of “making its own decisions”, then any current system would be considered “automatic”, not “autonomous”, because they are all deterministic in their decision making. This observation raises the question of whether any currently foreseeable (i.e. deterministic) system is truly autonomous; the ambiguity of the term may be why many sectors are choosing to use the term “unmanned” instead. However, in order to encourage research and development in useful

¹ Patrick Winston, former director of MIT’s Artificial Intelligence Laboratory, as quoted in “Autonomous Land Vehicles” by Dr. Hugh Durrant-Whyte.

near-term areas, one might use “autonomy” in an inclusive rather than restrictive sense. Therefore, we propose the following working definition of an autonomous system:

An autonomous system is one that makes and executes a decision to achieve a goal without full, direct human control.

Here “system” does not have to mean an entire vehicle; it could also mean a subsystem like the anti-lock brake system (ABS) example. By this definition, automatic is not distinct from autonomous, but is a subset instead. This inclusive definition dovetails nicely with the ongoing efforts to classify “levels of autonomy”. These levels would depend on such things as mission complexity or level of required human interaction. Automatic systems (single input to single output) would occupy the lower end of any autonomy scale.

In developing this working definition of autonomy, it became clear that there are two main areas of development for an autonomous vehicle: decision-making and real-time control. Generally speaking, the decision-making side corresponds to “autonomous” (or independence) and the real-time control corresponds to “vehicle” (or execution), although the line between the two can be a bit fuzzy at times. There is clearly some local decision-making that takes place within the realm of real-time control, such as in local navigation and obstacle avoidance. Otherwise, robots would run into an obstacle while trying to decide whether to go left or right around it. Similarly, these two categories cannot stand alone. Developers of autonomous vehicles cannot work on autonomy and computer processing separately from working on vehicle mechanics—the integration of these two areas into one physical system presents a significant challenge in and of itself. Not only does the computer equipment need to be able to physically withstand the operational environment onboard a moving vehicle, but it also needs to appropriately connect the algorithms to the incoming sensor data and decide which sensor information is needed in the first place.

Consistent with all three definitions above, “autonomy” is not a technology itself, but rather a capability enabled by supporting technologies. Dr. Durrant-Whyte divides these technologies into five categories: mobility, localization, navigation, planning, and communication [3]. Mobility includes the real-time control and mechanics of the vehicle itself. Localization incorporates sensors and software to identify the vehicle’s position, attitude, velocity, and acceleration. Navigation, to include local obstacle avoidance, combines decision-making and real-time control. Planning includes mission- and task-level decisions, waypoint generation, task allocation, etc. Communication involves all the links between the vehicle and teammates, operators, and command and control. These five categories summarize the main contributing technology areas for autonomous vehicles.

B. Autonomy Levels

There has been extensive work by others attempting to go beyond a working definition of autonomy to quantified autonomy levels [5–8]. From these various scales, four main categories were identified: piloted vehicle, authority in the

loop, authority on the loop, and authority out of the loop. These categories are based on work presented by Chad Hawthorne and Dave Scheidt at the Johns Hopkins University Applied Physics Lab [9]. These categories will differ qualitatively in engineering approach, test and evaluation activities, and demonstrated reliability prior to fielding.

II. DECISION-MAKING

Autonomous decision-making is an incredibly complex subject, especially given the fact that scientists do not fully understand how the human brain works and makes decisions. There appear to be two main categories of machine decision-making: reduction and learning. Figure 1 below reveals just how complex autonomous decision-making processes can be [10]. Indeed, one way to measure levels of autonomy would be to consider how many layers of the decision-making process portrayed are employed by the unmanned system.

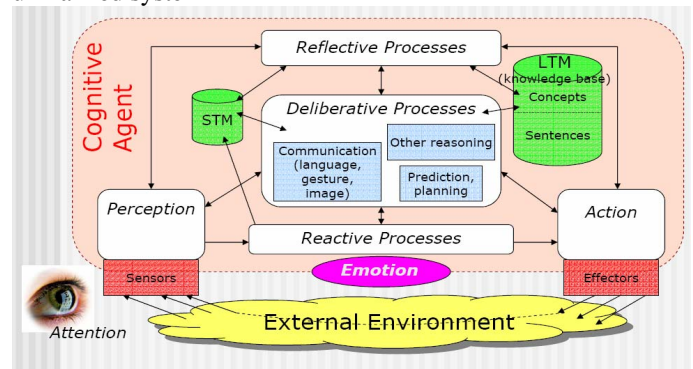


Fig. 1: Diagram of a Cognitive Agent

A. Reduction

As illustrated above, decision-making can involve much more than a simple binary selection. Humans incorporate a priori knowledge, context, and emotions when making decisions. In the reduction approach to autonomous decision-making, those elements are largely excluded. Instead, the problem is reduced to a simple, clearly-defined input-output mapping.

Automotive subsystems provide numerous examples of this approach. Anti-lock brake systems have long been the standard in American cars. Anti-lock brakes use a sensor that detects changes in wheel spin rate. When that sensor readout passes a certain threshold, the automatic brake is activated. There is a direct mapping of input to output, a clear rule for which action to take and only two choices for action: activate or not. ABS incorporate both the autonomous decision-making mentioned above and real-time control, in the pumping of the brake. From the definition of an autonomous system proposed above, the “goal” declared by the human driver is to stop; the ABS then decides how to accomplish that goal—whether the pumping is required in the situation—and then executes that decision, all more quickly than a human driver could.

A similar threshold sensor with a binary output option found in automobiles is the air bag. When deceleration is faster than a certain limit, the air bag deploys. Again, the air bag is an autonomous subsystem—although in this case, even more control is ceded to the computer, because the driver cannot override the decision to deploy just by lifting his foot up the way he can with his brakes.

An extension of this method is used in the new hybrid cars to optimize the ratio between fuel and battery power consumption as a function of speed, remaining battery life, etc. In this situation there are more dimensions than for ABS or airbags: the onboard computer needs to determine the optimal split between combustion and battery power and to execute the switching back and forth.

An example of the reduction approach that has already been applied in robotics is simple obstacle avoidance. The problem can be reduced to a binary output—“can I go straight or not?” There are only two output options and potentially only one required sensor. This is a very simplified method and would probably not detect things like cliffs or chain link fences, depending on the capabilities and sensitivity of the vehicle’s sensors, but it can be enough to successfully avoid obstacles in indoor or relatively uniform outdoor environments.

A variation on the reduction approach is the use of multi-robot systems. The concept is to give simple tasks/capabilities to each robot and connect them via a wireless network. By separating the overall mission into smaller subtasks, the complexity of the problem has been reduced to one that can be physically accomplished by current robots.

If the problem is more complex and a simple mapping is not obvious, researchers can conduct experiments, collect data, and write algorithms that characterize the domain within a given area (e.g. the flight envelope for an airplane autopilot). Then computing power can be employed to perform the bookkeeping and keep track of sensor data, the aerodynamic effects on the vehicle, etc.

A similar “bookkeeping” approach has been suggested for obstacle avoidance. Ideally, this method would help optimize a ground robot’s route—different maximum safe speeds could be connected with each terrain type, for example. However, building the database would be quite tedious and require some foreknowledge of the route. This approach degrades in changing terrain, and may require storage and processing beyond space/weight/power limits.

One issue with the reduction approach is that the “rules” given to the computer are only good within the given operational envelope—it is very difficult to cope with scenarios that fall outside the bounds of predicted patterns. For example, in 2001 a P-3 was involved in a mid-air collision with a Chinese aircraft and the pilot managed to land the plane safely [11]. To accomplish this feat, the pilot had to assess the plane’s changed response with enough speed and accuracy to prevent the plane from crashing. Current autopilots, such as that on the Global Hawk that recently landed safely after an engine flameout [12], may be able to

recover from types of in-flight failure that have standard responses that can be programmed in ahead of time. However, other types of failure may be too far outside the operational capability of the aircraft, requiring human-level experience, intuition, and rapid learning in order to successfully recover.

B. Learning

The other approach is to attack complex, incompletely characterized problems with superior computing power. The example of the P-3 pilot recovering from a midair collision is exactly the type of learning that the AI world is trying to recreate in order to tackle complex, incompletely-characterized problems. Missions beyond a certain level of complexity may never be possible without some leap in computer learning. For example, with the terrain database approach discussed above, it seems implausible to develop a database with any significant operational envelope for an uncertain or unknown outdoor environment. When the environment is structured or can be structured without disruption, it becomes possible to more fully characterize it, and achieve mission success with modest machine learning. Robots in manufacturing plants that follow lines or magnets in the floor are an example of such an application.

Robots need structure; that is how the variation and surprise can be restricted to levels that current processing power and algorithms can handle. Vehicle developers often find a way to bring structure to the environment and make it navigable for the unmanned systems. However, many of the environments in which users would like to send robots, such as natural disasters or military operations, cannot be structured ahead of time. If managing the environment is infeasible, it becomes increasingly important to develop learning capabilities so that robots can function in changing or unknown environments. The Defense Advanced Research Projects Agency (DARPA) has a number of programs focused on advancing machine learning and autonomous decision-making, such as Learning Applied to Ground Robots (LAGR) and Real World Reasoning (REAL). However, these programs are still in the early research phases and lie outside the scope of this report. To put the challenges these programs face in context, a first responder or soldier has 18 years or more of learning, before the task specific training starts.

C. Summary

Mission complexity for fully autonomous systems will be severely limited until significant AI developments are achieved. However, there are still a number of useful steps that could be taken, and it is in these areas that research and development would be most useful in the near-term. High payoff pursuits for near-term development include:

- Further characterizing the environment, i.e. quantifying and expanding the understood operational envelope for ground vehicles
- Increasing reliability of communication links in order to progress from tethered teleoperation to wireless
- Making sensible choices about the role and application

of autonomous vehicles and focusing development on those applications,

- Building machines robust enough to withstand less fine-toothed decision-making

III. REAL-TIME CONTROL

Real-time control concerns, in part, the physical aspects of an autonomous vehicle, as well as the translation from decision to action. Decision-making is still largely regarded as “science” and the real-time control is primarily considered “engineering”. However, this does not mean that all the unsolved problems are on the decision-making side and that successful real-time control is just a matter of working out some engineering details.

One continually difficult problem is local navigation and obstacle avoidance. Vehicles need to fuse and process sensor data at fast enough speeds and with enough accuracy to prevent running into things or getting stuck before higher-level decisions can be made. In a way, obstacle avoidance captures both real-time control and decision-making, albeit on the small-scale, local level. Current appropriate sensor packages are few and far between. While the problem may be “solved” in a performance sense, if the sensor that has been developed does not meet space, weight, power, and cost constraints, then that sensor is not a solution at all. Because the work done in this field is so application-specific, there appear to be numerous individual claims of solutions or successful demonstrations. Yet those successes do not readily translate to other programs or platforms. Therefore, it would be premature to consider such issues “solved” problems.

Much of the difficulty in developing autonomous vehicles capable of complex missions is that researchers don’t understand how humans make decisions or perform those same tasks. The same is true for some aspects of real-time control. The human hand is an incredibly complex array of sensors and interconnected effectors. The sensitivity of force sensors in our fingers is unparalleled. There is also a certain amount of local processing that takes place—for example, if a person touches a hot stove, his hand jerks away before the brain has even had time to register that the surface was hot. Similarly, if someone walks into a door frame, they don’t break a shoulder; they automatically start reducing the pressure applied at the point of contact. A robot, on the other hand, can snap an appendage off if it runs into a doorframe or tries to find a light switch and flip it on in a dark room. So there is a tradeoff between sensitivity and precision. The current sensor packages available for autonomous vehicles provide much less information to the decision-making algorithm than humans use on a regular basis. While building a humanoid robot may not be a primary interest for the military, this example highlights one of the significant limiting factors in the application of robots. Therefore, the best focus for development efforts is on tasks at which robots exceed human performance, rather than ones that just try to mimic humans.

A final challenge facing autonomous vehicle development from the real-time control side is systems integration. It is essential that all the components be mounted on board a mission-appropriate vehicle and that they survive the mission. Current sensor packages are generally too expensive or too bulky for practical applications—especially on ground vehicles. The vehicle also needs to be robust enough to protect all of the sensor and computing equipment when navigating in rough terrain. Similarly, a highly advanced sensor may be developed that would allow for significantly increased autonomy, but if that sensor requires a massive power supply, the vehicle would not be able to move very far from the base station. The systems integration challenge highlights a key issue in future autonomous vehicle development—specialization vs. generalization. While general programs and packages applicable across platforms appear to be the ideal, truly successful robots to date have been developed for specific missions. The specialized approach limits the systems integration issues, because the pieces are designed to go together more readily. While a common architecture or sensor platform may be on the research horizon, for the near-term, the field might be better served to focus development on more capable, task-specific vehicles.

IV. CONCLUSION

Research efforts in AI and cognitive computing have been largely theoretical or simulation-based. There is a disconnect between the field of robotics and the field of cognitive computing, especially when it comes to real-world implementation. Current artificial intelligence research is, by and large, not being designed for implementation on board a moving vehicle; yet robots will only be able to achieve a certain minimal level of complexity without integrating AI concepts and developments. If any significant advances beyond teleoperation are to be made in autonomous vehicle development, these two research fields need to come together and use advances from each area in the development of new vehicles.

Up to this point, autonomous vehicle development has been either highly application specific or too theoretical to apply on board an actual vehicle. There is a commonly-held hope that a single architecture or navigation method could be developed that would apply across platforms or applications, but that does not appear to be an option in the near term. Basic research should continue to provide new capabilities. However, it seems that there are many factors specific to each mission and/or environment that require specific development efforts for both the decision-making approach and the real-time control for each application. Thus far, the more useful a vehicle has been, the more specialized its development was.

Programs that focus on real-world implementation appear to be having more success. Their progress along the autonomy scale may be modest, but they are fielded and saving the lives of disaster victims or soldiers. However, DoD acquires

general-purpose equipment precisely because it is more difficult to anticipate operational needs than commercial ones. Therefore, the primary goal for the Department of Defense seems to be to increase the mission complexity and environmental variability in which unmanned vehicles are capable of performing. In this way human soldiers can be removed from dangerous, dull, dirty, and dumb situations. For DoD applications, at least, this motivates advances in cross platform commonality and in autonomy.

Basic AI research is still required, especially in the area of transfer learning—generalizing from a previous example to a novel situation. Until this trait of humans is more fully understood and accomplished in computers—or its effects mimicked—there will continue to be long training times and high costs, often resulting in brittle performance. In the slightly closer-term, integrating AI systems on board robotic platforms would yield enormous payoffs. We have only begun to focus on what happens with AI systems in real world environments. Incorporating context and intuition into machine systems—or at least modeling and understand their role in human decision-making processes well enough to assess the impact of their absence in autonomous vehicles warrants additional attention is both AI and psychology .

Outdoor obstacle avoidance remains a key issue for ground vehicles and is probably the area that already incorporates significant AI but also runs into the most problems due to the incredible variability of the terrain. Obstacle avoidance is much more straightforward for an unmanned aerial vehicle (UAV): there are far fewer obstacles above tree level and less variation in the environment. Indoor environments and highly structured outdoor environments such as those in agricultural applications are more tractable, specifically because structure has been imposed on the environment.

REFERENCES

[1] "Autonomy." *Merriam-Webster's Collegiate Dictionary*, 9th ed, 1986

[2] United States. Department of Defense. Defense Technical Information Center. DoD Dictionary of Military Terms. Jul. 2005 <<http://www.dtic.mil/doctrine/jel/doddic/dict/data/a/00599.html>>

[3] Durrant-Whyte, Hugh. "Autonomous Land Vehicles", Proc. IMechE. Vol. 219 Part I: J. Systems and Control Engineering, IMechE 2005

[4] Spence, Floyd D., National Defense Authorization Act for Fiscal Year 2001 (P.L. 106-398 Sec. 220)

[5] Clough, Bruce T. "Metrics, Schmetrics! How The Heck Do You Determine a UAV's Autonomy Anyway?", 2002. Jun. 2005 <http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/PerMIS_2002_Proceedings/Clough.pdf >

[6] United States National Institute of Standards and Technology, "ALFUS Framework", 2005. Jun. 2005 <http://www.isd.mel.nist.gov/projects/autonomy_levels/ >

[7] Sheridan, Thomas B. and Verplank, William L., "Human and Computer Control of Undersea Teleoperations", 1978

[8] IDA Rosetta Stone Paper

[9] Hawthorne, Chad and Scheidt, Dave, "Moving Emergent Behavior Algorithms from Simulation to Hardware: Command and Control of Autonomous UxV's", 10th International Command and Control Research and Technology Symposium, 2005

[10] U.S. Department of Defense, Defense Advanced Research Projects Agency, <<http://www.darpa.mil/ipto/briefings/IPTO-Overview.pdf>>

[11] Venik's Aviation, "Midair Collision Over China", 2001, 10 JAN 2006,

<<http://www.aeronautics.ru/news/news001/news031.htm>>

[12] Harvey, David S. "Global Hawk: Flameout Led To Automatic Afghan Alternate", SEP 2005, 10 JAN 2006 <<http://uvscanada.org/blog/?p=46>>

Long Term Study of a Portable Field Robot in Urban Terrain

Lundberg, C., Christensen, H.I
Centre for Autonomous Systems (CAS), KTH
S-100 44 Stockholm, Sweden
carl.lundberg@fhs.se, hic@nada.kth.se

Reinhold, R.
Länna Hammarby 6945
S- 761 93 Norrtälje, Sweden
info@rogerreinhold.se

Abstract— Military, police, fire brigade and rescue services need to evaluate robots as an aid to remove humans from risk, to perform more efficiently or at lower cost and to enable missions unsuited to humans. The benefits gained by using robots have to be valued against costs for acquisition, integration, training and maintenance as well as mission efficiency and reliability.

This study has investigated the benefits and costs experienced by an Army company specialized in urban operations while solving their tasks with the support of a PackBot Scout robot. Established research methods such as observations, exploratory testing, interviews and surveys were used for documentation of the study ranging over a period of two years. During the entire process great emphasis was placed on keeping the users' working conditions as normal as possible.

This paper describes the methodology used during the test, i.e. how the robot was embedded into the users' everyday activities and how this was evaluated. Experiences from the performed tests and evaluation approaches are discussed in order to serve as support for future field robot evaluations.

It is concluded that the taken approach can be recommended to research projects with a robot prototype with reasonable capability to work in a realistic setting in order to test its relevance and readiness.

I. INTRODUCTION

Robots also referred to as Unmanned Ground Vehicles (UGV), have been in extensive use for EOD (Explosives Ordnance Disposal, i.e. removal, disarmament and destruction of explosives) and mine clearance operations for quite some time. Recently, the use of robot technology has aspired to be incorporated in a number of other types of operations. Military, police, fire brigade and rescue services seek to evaluate robots as an aid to remove humans from risk, to perform more efficiently or at lower cost and to enable missions unsuited to humans.

Search and exploration is one of the most investigated applications for the next generation of field robotics. The ability to traverse and perceive premises is moreover the base for most robot applications. However, it is still unclear if current technology is mature enough to justify further implementation. Benefits gained by using robots have to be

valued against costs for acquisition, integration, training and maintenance as well as mission efficiency and reliability.

The aim of this project was to examine the advantages and disadvantages connected to the use of a scout robot on operator level as well as on higher levels in the organization. A long-term approach was chosen for the test, in order to diminish the initial bias connected with the introduction of a new product, give the test group time to modify their behaviors to the new circumstances and to form a mature opinion.

In this paper it is described how a PackBot Scout robot was implemented in an Army company and by what methods research was performed. The taken approach is discussed with the aim to serve as a support for further research with similar objective.

II. RELATED WORK

Various long-term user tests have been performed in the area of robotics previously. An early example is the integration of the SURBOT [1] for mobile surveillance in a nuclear power plant 1986. Later are the 17 month testing of the robot seal Paro amongst elderly [2], the 3 month testing of the fetch-and-carry robot CERO by a partially impaired person [3] and a number of long term tests of tour guide robots such as the RoboX9 serving for over 5 months at Expo02 [4]. The area of robot use in search and rescue has been subject to increased research since 9/11/2001, both in the USA [5] [6], and elsewhere [7]. Another example is the now substantial experience of long-term use of robots in space applications such as the NASA-rovers deployed on Mars [8]. Police SWAT-application has been analyzed regarding robot use [9], and finally, a considerable amount of work has been done in the military arena although it is not commonly published in detail [10].

The abovementioned examples have the implementation in a realistic setting over a period of time in common. But despite having the same overarching goal, the approaches may have to vary significantly depending on objective, resources and type of application. For example, evaluation amongst elderly will differ vastly from evaluation with search and rescue personnel. Likewise, there is a great variation on technical requisites to enable testing. The reliability demands are, for example, totally different if testing in a museum compared to

deployment on Mars. Hence, it is not obvious how to evaluate mobile robots. Established research methods may not fulfill the tele-presence and dynamics of mobile robots [6]. Research methods from other fields need to be trialed and perhaps modified or redeveloped in order to facilitate evaluation of robotics.

III. METHODOLOGY

A. Outline of Study

The project was a joint initiative between the Swedish Royal Institute of Technology (KTH), the National Defence College (FHS), the Swedish Defence Materiel Administration (FMV) and the Royal Life Guards Regiment of the Swedish Armed Forces (LG). The cooperation between the involved parties was initiated as members of military acquisition (FMV) and academic research (KTH and FHS) in cooperation together addressed the issue of to what extent current UGV-technology could be of benefit in urban field applications.

The project started with a user study and a number of small-scale robot tests with the purpose to observe the user and explore in what way robots could be of benefit in the users' activity. This research was carried out during five military exercises under a period of four months from fall 2005 to spring 2006. Data was gathered through field observation, informal interviews and participatory observation (Fig. 1). The robot tests included implementing and performing exploratory testing of the robot within group level as well as by having one of the researchers, who is an officer, participate as robot operator while the robot was used within company level [11].

Based on the results from the first phase it was decided to perform implementation and long-term testing on a larger scale during the following year of service for drafted soldiers doing military service. A number of the users' standard procedures were redesigned to include the robot as an aid. Once mastered by the operators the new procedures were demonstrated to the rest of the company. It was thereafter up to the officers of the company to use the robot as they saw fit in their training maneuvers. During the test period the robot system was part of the company as any of their other equipment.

PRE-STUDY	INTER-STUDY	POST-STUDY
Documentation Study Formal Interview Observation Pre-Testing	Observation Informal Interview Exploratory Testing Participation	Formal Interview Questionnaire Analysis Workshop
FY 2004/2005	FY 2005/2006	FY 2006

Figure 1. The various data collection methods deployed during the three phases of the test.

B. User Group and Test Facilities

The user group consisted of an Infantry Company for Military Operation in Urban Terrain (approx. 200 soldiers and 15 Armored Combat Vehicles). In addition 10 training officers were affiliated with the company during maneuvers. The maneuvers in which the robot trials took place included 200 to 6000 soldiers and lasted between three to six days.

All tests were carried out in facilities regularly used for police, fire brigade and military training. These consisted of deserted and partly destructed industrial and residential buildings and offered an environment similar to what can be expected in real operations. During the tests no adaptations or adjustments were done to the environment. The test period spanned over all season including all weather and a temperature span from -25°C to $+25^{\circ}\text{C}$.

C. Robot System

The IRobot PackBot Scout [11] used during the study was equipped with a number of accessories (Fig. 2). The same Direct Fire Weapon Effects Simulator that is used by the soldiers was mounted on the robot (DFWES-Saab BT46). The system consists of a laser mounted on the firearm and a sensor suit worn by the soldier. The system simulates direct fire when training with blank ammunition.

Two more payloads were developed, a flashlight for illumination in dark premises and a Claymore mine. The operator could trigger both by remote control. During the test the system also came to include extra batteries enabling a typical day's deployment, chargers, basic spare parts, a rope for lowering the robot into lower premises, a telescopic rod which could be attached vertically to the robot in order to detonate trip-wired explosives and protective cases for all the included equipment.

D. Robot Implementation

The long-term test started off with a handover of the robot system to the users. From then on, except during modifications and repairs, the robot was kept by the company during the 6 month test period. During the test all transport, maintenance and charging was carried out through the users' ordinary resources.

It was stated to the users that the robot system should be exposed to realistic stress and that any damage or wear due to normal use was a beneficial result to the study. The robots durability was said to be alike the users' radios or optical equipment.

In agreement with the Commander of the Royal Life Guards Regiment it was decided that one officer and two soldiers should be trained to operate the robot. They were previously non-practiced in robotics but were accustomed computer users and had some experience of RC-crafts. Unfortunately one of the soldiers was released from duty due to medical reasons after two months. Just as during the pre-study did one of the researchers act as operator in cases when the trained operators were not available. The original purpose of this was to

increase the prospect of robot use but it also proved to be a valuable occasion to perform participatory observation.

Three levels of operator training were defined; 1-*Basic Level*, 2- *Map and Search Level* and 3- *Tactical Level*. Training for the *Basic level* was done according to the scheme developed during the pre-study. This included briefing about the robot, basic driving and familiarization with the appearance through the cameras onboard, as well as performing simple missions. After the basic training, which could be done in less than one hour, the operators were able to continue practicing on their own.

Since the higher-level behaviors were not yet defined the rest of the training was in large extend performed in parallel with exploratory testing. The *Map and Search Level* incorporated the ability to make a sketch of explored premises and to search for persons or IEDs (Improvised Explosive Devices).

After having acquired personal skills in robot control the operators were trained to act in conjunction with other soldiers, i.e. the *Tactical Level*. First, this was done in pairs, where operator and assistant were trained to act as a team where the assistant handled the close up safety, transport, robot revival etc. (Fig. 2). Subsequently, after approximately seven days of practice, the pair was integrated into group, squad and platoon level performing their tasks in synchronization with other mission actives (Fig. 3). The information transfer from operator to team proved to be a crucial issue. While the two first training levels only include the operator and the assistant, the third level also requires adaptation from the group in which the robot operated.

The robot system was demonstrated to the rest of the company once the operators had acquired the necessary skills. The demonstrations were done to one platoon at a time and included a briefing about the system, safety issues and a



Figure 2. To the lower left the PackBot Scout robot, the robot assistant in the center and the robot operator in the upper right. The robot is about to drive up the staircase to the left, the assistant is ready to act in the most hazardous direction which is in this case considered to be the staircase. The operator is controlling the robot from a previously secured area.

demonstration of a search mission. It was emphasized that the robot use was a test and that some aspects could not be expected to have full effectiveness until some time had been given for tactical development and training.

After the demo the company was free to use the robot system as they pleased. Robot missions evolved to be initiated in two ways. Either the company commander submitted the robot system to one of the platoons in advance of the missions or the platoon or squad leaders requested assistance of the system during mission realization.

The formal organizational position of the robot operator was in the Company Command Squad. But at the end of the test period the company commander had set as a standard to submit the system to one of the platoons. Most often to the platoon which were to perform brake-in into the targeted building. Once a platoon's need of the robot had diminished it was released and the operator relocated to the medical evacuation post, which is aimed to be reachable by the entire company all the time.

During the training maneuvers the robot could be determined to be hit either by the DFWES-system or by the training officers. If judged to be destroyed it was returned to the combat vehicle used by the operator. The robot operator could then retrieve it as if it was a new robot. There was no limit set for how many times this could be done, i.e. the company had a fictitiously unlimited recourse of robots. However, in reality there were one or two systems in use at the time. The same safety guidelines developed during the first test phase were deployed during the long-term test [11].

E. Data collection and Analysis

A number of data collection methods were deployed during the three phases of the project (Fig. 1). The initial step of the user investigation was to study the training manuals and



Figure 3. From the left: the platoon leader, the robot operator, the group leader, and thereafter the soldiers. The platoon leader is using the robot-system to perform exploration around the corner. The group leader is observing the neighborhood and the soldiers are ready to act in hazardous directions.

instruction videos of the users. The researchers also participated in a national workshop and a NATO-workshop on urban warfare in order to deepen the knowledge in the field.

Observations were performed for two purposes. First, to gain knowledge about the users' environment and way of working. Second, to evaluate the use of the robot. These two processes were performed in parallel although the former was initiated at an earlier stage.

The users are accustomed to have both instructors and visitors amongst them during exercise. Armbands are used to distinguish observers from participants, which made it possible to have full access for observation during the maneuvers. For safety reasons hearing protector and eye protectors had to be worn. Documentation was done through video and photography.

An alternative way of observing the course of events was to listen to the radio communication between the platoon leaders and the company commander. This is an established way for the training offices to keep informed about the overall progress of the unit.

Participatory approaches were used when the trained operators were not available. This was possible since one of the researchers in the group is an army officer who was trained to operate the robot along with the operators from the company. During the participatory tests the researcher took the place of the operator soldier in obedience of regular rules and demands.

Three types of interviews were conducted during the project. From the start, and during the tests, participants were interviewed about their established procedures and about their experience of working with the robot. These interviews were done spontaneously whenever appropriate in the field and were held in an informal manner. In some cases these interviews were documented with video or notes but most commonly the data was written down at a later point of time.

Secondly, two officers were interviewed in advance of the long-term test regarding what applications they thought might be feasible for robot implementation. Notes were taken during the interview and the conclusions were verified with the participants within a couple of days.

Finally, after the long-term study, an even mix of 10 soldiers and officers were chosen for an in depth interview regarding their experiences from robot use. An anthropologist who had not participated in the field studies was recruited to perform the in-depth interviews. The interviews, which lasted a half to one hour, were semi-structured with one person at a time held in one of the users' classrooms. A number of topics were defined to be explored but the inquiry was open for modification and extension. Audio was recorded with an MP3-player and non-verbal cues etc. by notes. Each interview took six to eight hours to transcribe and another six to eight hours to analyze.

After the long-term test 40 of the most robot experienced soldiers (35) and officers (5) were selected for a questionnaire. The inquiry was designed to explore the participant's opinion about efficiency in different standard procedures, the robot's significance compared to other equipment, their ideas on

possible future use and whether or not they supported acquisition. The questionnaire contained both open-ended questions and Likert scale rated statements. It took the respondents between 20 minutes to one hour to fill in the form.

A number of issues have been investigated repeatedly through different methods along the project. An evaluation group has been organized to discuss and conclude the gained data. The group, which holds members from all the participating organizations, will conclude the outcome of the study in the fall of 2006.

IV. DISCUSSION

A. Implementation

A number of factors have influence on the results of long time studies. Some of these can be controlled to a desired state while others cannot, and therefore have to be regarded as an influence on the results.

Performing the tests in a relevant environment is of major importance. In this case the most realistic settings available were the large-scale maneuvers performed during soldier training. Still, these do not fully monitor all aspects of a real application. For example, the true impact of casualties is hard to reconstruct and considerations connected to these may be influenced. Further, the training maneuvers focus on the most difficult tasks rather than the most common in real missions, which is probable to offset the results. Despite these biases, a qualitative approach might be the best option while testing in large and complex settings such as operations including several hundred persons acting individually and dynamically on a mutual task. This, since there is no way such a complex setting can be kept constant enough to facilitate comparative quantitative measures or because the sought occasions do not occur with desired frequency.

Besides having a true test environment it is crucial for the user to have some kind of hardware-knowledge as a baseline. If the user is previously unfamiliar with robotics, his or her opinions will lack in relevance and realism.

An important quality of long-term testing is the decrease of bias connected to the introduction of a new product is diminished. It also gives the test group time to modify their behaviors to the new circumstances and to develop a mature opinion. In the beginning of the test period there were significant differences in the test persons' opinions of how the system should be used and what capabilities it could have. These anomalies were found to have decreased significantly during the test, which confirms the accuracy of the gained information.

The more developed the introduced system is, the more relevant are the rendered results likely to be. In conflict, there is a need to perform testing in early stages of the product design. An early introduction enables parallel development of the system and the tactics for use, if this includes changing well-established doctrines it might be a process taking

significant time. It also has to be considered that if the intended end users are incorporated in studies they will inevitably form an opinion about the tested system. Unsuccessful trials might have negative impact, which can be very hard to recover. Hence, the point of time when to perform testing is a strategic compromise. The PackBot Scout was found to be capable to perform well enough to serve as a basis for research around search, exploration/mapping and payload delivery. Issues regarding mobility, endurance, physical robustness, climate hardiness, radio reach, user interaction, tactics, organization and ethics concerning armed robots could be fruitfully investigated. The users did, if requested, have the ability to see past properties that restrained the system, for example the bulky operator laptop. On the other hand, implementation of the PackBot did not give the users an ability to discuss topics like autonomy or sensor data fusion with the same level of accuracy. Nor did the end-users have enough background knowledge to value the system in economical terms.

A long term study gives a good opportunity for extensive technical evaluation. Having the robot stationed with the Life Guard Regiment consumed eight PackBot batteries, an entire track system and an operator laptop. Wear, damage and breakdown has to be considered in the plan of the test. The project proved to build a cooperation framework between the participating organizations and to serve as a suitable way to initiate the use of robotics in the addressed fields. It also gave an opportunity to gain insights about the physical and mental recourses available amongst the users.

Previous experiments and demos had showed the military often to have a curious but somewhat skeptical approach to robotics [11, 12]. The intention of handing over the system to the users (instead of bringing it to appointed trials) was to give them a sense of responsibility and thereby increase commitment to use the robot. Still, experience from the study shows that implementation of a robot requires significant efforts in development and training of new behaviors. Specific support and coaching will be required to enable this. Simply providing a user with hardware will not be a sufficient measure for implementation in applications requiring qualified human-robot interaction.

The actual handover of the robot was not planned to be other than an informal delivery of the system. But the meeting turned out to be a kind of “kick-off” since more officers than expected attended, including the commanders responsible for the training of the concerned company. Also personnel from the Defence Material Administration and academia attended. The spontaneous meeting was beneficial to the project since it served as a starting ceremony for the trial. Performing ethnographic studies in large settings will call for a great deal of flexibility in order to perform tests and observations once the chance is given. In this case the users’ primary goal was to do conventional training of soldiers and officers. In several cases were planned robot trials cancelled or rescheduled. Being flexible and having backup plans will facilitate higher research efficiency. During the performed project few

occasions were given to gather representatives from the concerned organizations. Occasions like the handover kick-off proved to be rare, but important for test management. The attention from the superior officers during the handover was valuable since the attitudes of superiors have a big influence in strictly hierarchical organizations. The attitude of the higher-level officers was continually enthusiastic during the project. The lower level officers, who were the ones actually having contact with the robot, showed a pending attitude until they got to fully know the system abilities.

For hierarchy reasons it was decided to also train an officer in robot operation, although operation, of the robot would most likely be the task of a soldier. This is because the officers in most cases possess all the skills of the soldiers. Having an officer trained in addition denoted the presence of knowledge about the robot system’s capacities during tactical planning and briefing. The trained Captain had the positions of Second Company Commander and was thereby a key person in the control and ordering of the company.

Just as Jones et al. concluded for SWAT-police [9] the military to a large extent uses predefined scenarios as starting points for their behaviors. This need emerges from the high demand to perform synchronized with small means of communication. A predefined scenario mediates how the co-workers are likely to act even if the arisen situation differs from trained scenarios. Any new routines introduced need to be defined and trained accordingly. The demonstration of the robot system to the company proved to be a vital component from the scenario aspect. It showed the soldiers and officers how the system could be used for exploration which turned out to be the most common mission to be deployed. It is believed that the other capacities, developed after the demo, would also have been favored if demonstrated in the same way. Unfortunately this was not feasible due to practical reasons, instead the newer capabilities were shown in a smaller scale during the maneuvers.

Safety and reliability were two important issues during the trials. On numerous occasions, have safety issues brought deployment of systems to long lasting halts. Besides being important for safety, reliability is also of major importance for the addressed users. Again, just as Jones et al. concluded for the SWAT-police [9], the military is not interested in the introduction of any additional uncertainties in their activity. The need for reliance, together with the desire to move and respond to new situations swiftly, is often considered more important than reducing risk (at least during training maneuvers). This conception proved to be very strong amongst the military but it is indicated by the test that the use of robots, which can be sacrificed, could decrease both own and enemy casualties and thereby justify a change of doctrine. Out of a methodical point of view the use of robots can be encouraged by conventional means such as information, training and demonstrations. But it should also be remembered that the participants in the maneuvers are constantly being graded and that officers deciding to deploy a robot take an increased risk of failure, which might influence

their carrier. Establishing a tolerant, supportive and rewarding atmosphere during tests will influence the rate of deployment.

Military is currently in the process of drastically improving their way of evaluating the combat training. Video, GPS and increased use of weapon training systems (like DFVES-46) is being deployed. It is important for the robot developers to also integrate these systems since the addressed user, at least in this case, spends more time practicing than performing live missions.

B. Data Collection

Taking part of manuals and instruction videos was a good way to get to know the basics of the users' work procedures, learn their terminology etc. As might be the case for many vocations, the documentation did mainly cover the basics. The high-level skills were mediated by the officers once the soldiers had reached a higher level.

Observation and participation were important ways for gaining a holistic view of the user. It showed to be essential to attend the briefings in order to grasp the course of maneuvers. Observing the process from the enemy side was a beneficial way to get another point of view.

Monitoring a group as large as a company brought about a number of practical issues. The target of observation had to be constantly shifted in order to catch the overall situation. The constant observation shifts and the movements of the unit in turn caused logistical demands such as bringing clothing, safety gear, supplies and batteries for one or several days, arranging transport along with the combat vehicle convoys and managing accommodation.

The circumstances during the field studies made documentation difficult. Out of note taking, photography and video (including using a helmet mounted camera) photography proved to be the most valuable way for documentation. Real time note taking was often unpractical and video caught to little valuable information compared to the workload and distraction it imposed.

The initial interviews with the two officers did produce numerous suggestions of how the robot might be used in urban warfare. Most of these proved not to be feasible during the later trials. Similar was the case with the unstructured interviews made during the trial. Many of the rendered suggestions of how to deploy robots were unrealistic. Not until the end of the deployment phase when the final interviews were conducted, were the users experienced enough to reflect over robot deployment with more unity.

The unstructured interviews conducted during the maneuvers were an important way of getting to know the users and their activities. The short moments of conversations in the field did, on the other hand, not give much scope for deeper reflections.

The ten interviews in the end of the project served both as a recollection of the missions performed and for surveying the opinions about the system. Follow-up questions were used to verify the validity of the responses, which made it possible to examine in which areas the users had a well-grounded opinion.

The respondents chosen for interview showed a high level of cooperation and willingness to share knowledge. Taking in an interviewer who had not participated in the field study decreased the risk of bias. The interviewer's lower level of knowledge did not seem to cause frictions with the respondents. Instead it forced the respondents to be detailed and descriptive, which enabled additional discoveries.

The questionnaire aimed to document the performed robot missions as well as to investigate the questions from the final interviews over a higher number of respondents. The questionnaire and the ten interviews were designed and performed in parallel. Alike the final interviews, the questionnaires too indicated which topics the users could answer with validity. A pilot-test of the questionnaire would have revealed this and enabled a more feasible survey design. The mission-documentation part of the questionnaire gave a statistical supplement to the more detailed descriptions received through interviews.

V. CONCLUSION

It is concluded that the taken approach may well be used to investigate technical, tactical, ethical, organization and interaction issues concerning robot use in field applications. The ethnographic approach in combination with a long-term implementation rendered data that changed to be more uniform over time. It is reasonable to believe that the increase of unity indicates that end-result is valid and can serve as a foundation for future work.

Apart from producing valuable knowledge, the project did establish a cooperative relation between the involved organizations. It also served as the initial step in the introduction of robots for urban operations in the Swedish Defence, which is considered to be central for future robot implementation since military tactics need time to incorporate new technology.

ACKNOWLEDGEMENT

Coworkers from KTH would like to gratefully acknowledge the participation of 6th Urban Warfare Company of the Royal Life Guard Regiment and the contribution from anthropologist Roger Reinhold who performed the final interviews and participated in the analysis of the data collected during the project.

REFERENCES

- [1] White, J., Harvey, H., Farnstrom, K., "Testing of mobile surveillance robot at a nuclear power plant", *Proceedings of IEEE International Conference on Robotics and Automation*, March 1987.
- [2] Wada, K., Shibata, T., Saito, T., Sakamoto, K., Tanie, K., "Robot assisted activity at a health service facility for the aged for 17 months: an interim report of long-term experiment", *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, Nagoya, Japan, June 2005.
- [3] Huttenrauch, H., Eklundh, K.S., "Fetch-and-carry with CERO: observations from a long-term user study with a service robot", *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, Berlin, Germany, Sept. 2002.
- [4] Tomatis, N., Terrien, G., Piguët, R., Burnier, D., Bouabdallah, S., Arras, K.O., Siegwart, R., "Designing a secure and robust mobile interacting robot

- for the long term”, *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 2003.
- [5] Murphy, R.R., “Human-robot interaction in rescue robotics”, *IEEE Transaction on Systems, Man and Cybernetics*, Part C, May 2004.
- [6] Scholtz, J., Young, J., Drury, J.L., Yanco, H.A., “Evaluation of human-robot interaction awareness in search and rescue”, *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, New Orleans, USA, April 2004.
- [7] Matsuno, F., Tadokoro, S., “Rescue Robots and Systems in Japan”, *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, Aug. 2004.
- [8] Leger, P.C., Trebi-Ollennu, A., Wright, J.R., Maxwell, S.A., Bonitz, R.G., Biesiadecki, J.J., Hartman, F.R., Cooper, B.K., Baumgartner, E.T., Maimone, M.W., “Mars Exploration Rover surface operations: driving spirit at Gusev Crater”, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Waikoloa, Hawaii, USA, Oct. 2005.
- [9] Jones, H.L., Rock S.M., Burns D. & Steve Morris. “Autonomous robots in SWAT applications: Research, design, and operations challenges”. *Proceedings of AUVSI International Conference on Unmanned Vehicles*, Orlando, FL, July 2002.
- [10] John, A., “FCS Update”, *Unmanned Systems*, Volume 24, Number 2, Mars/Apr 2006.
- [11] Lundberg, C., Christensen, H., Hedstrom, A., “The Use of Robots in Harsh and Unstructured Field Applications”, *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, Nashville, TN, USA, August 2005.
- [12] Lundberg, C., Barck-Holst, C., Folkesson, J., Christensen, H.I., “PDA interface for a field robot”, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, Las Vegas, NV, USA, October 2003.

A Standardized Testing-Ground for Artificial Potential-Field based Motion Planning for Robot Collectives

Leng-Feng Lee and Venkat Krovi

Mechanical & Aerospace Engineering, State University of New York at Buffalo
318 Jarvis Hall, Buffalo NY 14260 USA
{llee3, vkrovi}@eng.buffalo.edu

Abstract— In this paper, we examine and evaluate artificial-potential field approaches for motion planning of robot collectives with formation-maintenance requirements. To this end, we demonstrate the practical use of construction of the Navigation Function (NF) to serve as the “standardized testing-ground.” The NF allows a designer to merge multiple local limited-range potential-functions uniformly into a well-behaved “potential space”-without creating multiple local-minima or irregular scaling at the obstacles. A MATLAB based Graphical User Interface (GUI) is created to aid the interactive creation of the “potential-space” for user-defined workspaces. Within this common test-ground, one is then able to systematically compare and evaluate the performance of various formation maintenance algorithms for robot collectives. In particular, we evaluate the performance of artificial-potential based formation-maintenance algorithms for wheeled mobile robot collectives with 3- and 10-members.

Keywords: *Robot Collectives, Motion Planning, Potential Field.*

I. INTRODUCTION

Ongoing revolutions in computing effectiveness and miniaturization of processors/sensors/actuators in the past decade have facilitated the deployment of networked distributed collectives of mobile robots in numerous applications from reconnaissance, foraging, herding to cooperative payload transport.

In recent years, the study of such *groups of multiple autonomous mobile robots exhibiting cooperative behavior* has emerged as an active and challenging research area. Groups in nature (from flocking birds, schooling fish to colonies of ants) appear to make use of a distributed control architecture in which individuals not only respond to their sensed environment (with limited ranges), but also respond to (or are constrained by) the behavior of their neighbors [1, 2]. Recent literature has identified the ability of relatively simple constraints such as: (1) attraction to neighbors up to a maximum distance, (2) repulsion from neighbors if too close, and (3) alignment or velocity matching with neighbors as playing the principal role in maintaining a group formation [1]. In addition, these constraints may also be employed to explain a ‘high-level emergent’ group behavior such as finding a food source, or move to higher temperature area, while avoiding obstacles (corresponding to a ‘gradient climbing’ problem).

Thus, there is a significant interest within the robotic community to better understand the biological imperative and exploit the same by incorporating similar principles in artificial robot collectives.

However, the diverse application arenas come with varying requirements for “cooperation”. Consider a case where a network of robots is to be used in a mapping/reconnaissance mission vs. one where teaming is desired for moving and manipulating large payloads. While deploying a group of robots (over a single robot) has distinct advantages in both cases, the cooperation and coordination requirements are significantly different. Considerable research attention has been focused on the former case where only loose formation maintenance is required [2, 3]. In this paper, however, we examine the latter case which has more stringent requirements on formation design and maintenance.

Artificial Potential Field (APF) approaches provide an intuitive way to model and analyze the behavior of group of robot with many desirable characteristics. In lieu of a deliberative/explicit trajectory or actuator-input computation (prior to the motion), the motion plan is reactively/implicitly specified in terms of the “dynamic-interaction behavior” of the robot system i.e. by how the robot interacts and responds to the sensed environment [4]. At the group level, the workspace is modeled as a “potential space” where the global minimum is at the destination, thus converting group-level motion-planning into a gradient decent problem. At an individual level, interactions of individuals with their neighbors and environment can also be modeled using potential functions, such that formation of the group can be maintained while achieving group motion. Despite these many advantages, APF approaches face the fundamental challenges in hierarchically combining these multiple levels of control within a common framework to realize truly decentralized yet scalable cooperation of such robot collectives.

In this paper, we examine and evaluate adaptation of potential field approaches for motion planning of robot collectives with emphasis on formation maintenance. *The crux is to select or create a “potential space” that can be used as a “standardized test-ground” for studying various formation-control algorithms. The Navigation Function (NF), in our*

opinion, is ideally suited for our implementation in that it allows us to combine the effects of multiple local potential approaches (with limited ranges-of-influence) to create a “well-behaved potential field”. There are numerous facets to this problem, as summarized pictorially in Fig. 1, that need to be carefully studied. However, in this paper, we focus our attention on sphere worlds with multiple point mass robots and stationary obstacles and target.

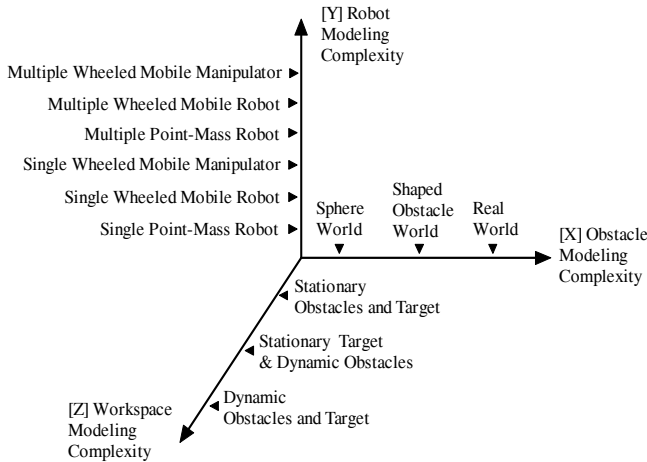


Fig. 1: Challenges entailed in multi-robot motion planning.

The rest of this paper is organized as follows: Section 2 surveys the variety of APF approaches and their key desirable features prior to introducing the navigation function. Section 3 recapitulates the principal formulation features of the navigation function within a convenient MATLAB Graphical User Interface (GUI) interface. Section 4 presents formulation, simulation results and comparative evaluation of potential-based formation- maintenance algorithms for groups of robots operating within such a test-ground. Section 5 concludes the paper with a discussion of the results.

II. LITERATURE REVIEW

In the past decades, *Artificial Potential Field (APF)* methods have gained popularity among robotic researchers especially in the mobile robot arena [5-10] due to its mathematical simplicity and elegance. Beginning in the early 80’s, Khatib’s work [7] focused on defining a potential field on the configuration space with a minimum at the goal configuration and potential hills at all obstacles. In such a potential field, the robot is attracted to the target while being repelled by obstacles in the workspace (The vector-sum determining the direction and speed of travel). Thus, the *gradient of the combined artificial potential* (from the multiple potential functions) serves as the input force driving the robot to its desired destination while avoiding collisions with obstacles.

Typically, the attractive potential field and the repulsive potential fields are formulated separately, and the total potential field of the workspace is obtained by linear superposition of the two fields. Examples of potential functions designed with this idea are Krogh’s GPF function [11], Khatib’s FIRAS function [7], Superquadric artificial

potential function [12], Ge and Cui’s New potential function [13], or the Harmonic potential function [8, 14]. These are called the *local potential approaches* since only local gradient information is needed. Such approaches are very attractive from a computational point of view since no processing is required prior to motion. Further, it is easy to specify a dynamic behavior that tends to avoid obstacles.

The main drawback of such a potential field approach is that when obstacles are present, the potential field may not be convex and local minima that can ‘trap’ the robot may exist at points away from the target. This local minimum is the result of unpredictable shape of total potential field after the superposition of attractive and repulsive potential fields. The second disadvantage of this approach is that it is difficult to predict the actual trajectory. In practice, one has to carefully pick the constants in the potential function in order to guarantee obstacle avoidance. Furthermore, the generated trajectories are usually far from being optimal by any measure.

Many of later approaches were developed to help overcome some of these limitations. Volpe and Khosla introduced a *Superquadric Artificial Potential Functions* [15], which model a wide range of shapes range from rectangles to ellipses. *Harmonic Potential Functions* [8] seeks to create a potential field without local minima that most potential field methods suffered from. Ge and Cui, proposed a new potential function that take into consideration of moving obstacles and moving target. The potential field is both a function of positions and velocities [6] of the robot, obstacles, and target. Another common problem found in most potential field methods, the GNRON problem (goals non-reachable with obstacle nearby) identified by Ge and Cui [13] and Volpe and Khosla [12], can be handled using their proposed potential fields respectively. Detailed studies of these local potential approaches, their limitations, and their characteristics were presented in [16].

Many of these limitations such as the multiple local minima of the irregular potential scaling at obstacles can be overcome by using the Navigation Function (NF), first introduced by Rimon and Koditschek in their series of papers [10, 17-19]. However, the NF is a ‘global’ strategy – constructing a configuration-space potential field free of local minima comes at the cost of losing the simplicity and computational advantages of the original local potential fields. Hence, while it may not well-suited for real-time applications, the NF based composite potential-field nevertheless retains many of the other desirable features and serves as an ideal testing-ground for evaluation of motion planning algorithms for robots collectives.

III. NAVIGATION FUNCTION (NF)

A. Properties of the Navigation Function

An NF is defined as follows: Let Ψ be a robot free configuration space, and let \mathbf{q}_{Tar} be a goal point in the

interior of Ψ . A map $\varphi: \Psi \rightarrow [0, 1]$ is a Navigation Function if it is [10]:

- (1) Smooth on Ψ , i.e., at least a \mathbb{C}^2 function.
- (2) Polar at \mathbf{q}_{Tar} , i.e., has a unique minimum at \mathbf{q}_{Tar} on the path-connected component of Ψ containing \mathbf{q}_{Tar} .
- (3) Admissible on Ψ , i.e., uniformly maximal on the boundary of Ψ .
- (4) A Morse Function.

Although a navigation function provide a workspace with a global minimum, an ‘‘essential’’ global convergence is not guaranteed. A global convergence means convergence from almost all initial configurations. In fact, it was shown that there exist at least as many saddle points as there are internal obstacles [10]. However, these spurious unstable equilibrium points need not cause any practical difficulties since only ‘‘few’’ such initial configurations. Further, for a group of robots with formation constraints, the chances of getting stuck on these saddle points is further decreased: if one robot stuck on the saddle point, other robots in the same formation will drive that robot out of the saddle point via the formation constraints.

B. Navigation function for a sphere world

For this paper, we focus on construction of an NF in a sphere world. The sphere world is a compact connected subset of E^n whose boundary is formed from the disjoint union of a finite number, $M + 1$, of $n - 1$ spheres. We largely follow the development in [10, 18] in the rest of the subsection.

Let $A(\mathbf{q}, \rho)$ denote a Euclidean n -dimensional disc with center at $\mathbf{q} \in E^n$, and radius ρ . A Euclidean Sphere World is formed by removing from a large n -dimensional disc, $A_0(0, \rho_0)$ (i.e. centered at $(0, 0)$ with radius ρ_0), M -numbered of disc-like ‘‘Hill’’, $A_j(\mathbf{q}_j, \rho_j)$ for $j = 1 \dots M$, called the obstacles. The bounded workspace $E^n - A_0$ is referred as the zeroth obstacle. The free configuration space (or simply, configuration space), Ψ that remains after removing all the internal obstacles from A_0 is:

$$\Psi = A_0 - \bigcup_{j=1}^M \text{Obstacle}_j \quad (1)$$

For Ψ to be a valid sphere world, the obstacles’ closure must be disjoint and be contained in the interior of A_0 . Thus, for this sphere world, there are $M + 1$ centers, q_i and radii ρ_i , for $i = 1 \dots M + 1$. Also, for this example, the Bounding Function is:

$$\beta_0(q) = -\|\mathbf{q} - \mathbf{q}_0\|^2 + \rho_0^2 \quad (2)$$

and spherical obstacle function given by:

$$\beta_j(q) = -\|\mathbf{q} - \mathbf{q}_j\|^2 + \rho_j^2, \text{ for } j = 1 \dots M. \quad (3)$$

These formulas are expressed in terms of the implicit

representation of the constituent shape, which assumed to be known. We are now ready to construct the NF for the sphere world which defined as:

$$\varphi_\kappa(\mathbf{q}) = \begin{cases} \left(\rho_\kappa \circ \sigma_1 \circ \frac{\gamma_\kappa}{\beta} \right)(\mathbf{q}) & \text{for } \beta > 0 \\ 1 & \text{for } \beta \leq 0 \end{cases} \quad (4)$$

where γ_κ is the distance-to-target function, given by:

$$\gamma_\kappa(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_{Tar}\|^{2\kappa} \quad (5)$$

In which \mathbf{q} denoted the position of the robot, \mathbf{q}_{Tar} denoted the position of the target, and $\|\mathbf{q} - \mathbf{q}_{Tar}\|$ is the Euclidean norm and $\kappa > 0$ is a control parameter. β in Eq.(4) denotes the product of obstacle function which is given by:

$$\beta = \prod_{i=0}^M \beta_i \quad (6)$$

where β_i is given in Eq.(2) and Eq.(3). On the other hand, $\sigma_\lambda(x) = x/(\lambda + x)$ and $\rho_\kappa(x) = x^{1/\kappa}$ are called the analytic switch function and sharpening function respectively, and they are both conditioning functions. For example, the analytic switch function $s(q, \lambda)$ performs the following operation:

$$s(\mathbf{q}, \lambda) = \left(\sigma_\lambda \circ \frac{\gamma}{\beta} \right)(\mathbf{q}) = \frac{\gamma(\mathbf{q})}{\lambda\beta(\mathbf{q}) + \gamma(\mathbf{q})} \quad (7)$$

Equation (7) has the following properties: it vanishes exactly at the zeros of γ , achieves its upper bound of unity exactly at the zeroes of β , and varies smoothly between the two elsewhere. Now using the sharpening function, combined with Eq.(7), we obtained the following NF in the sphere world:

$$\varphi_\kappa(\mathbf{q}) = \begin{cases} \frac{\|\mathbf{q} - \mathbf{q}_{Tar}\|^2}{\left[\|\mathbf{q} - \mathbf{q}_{Tar}\|^{2\kappa} + \beta(\mathbf{q}) \right]^{1/\kappa}} & \text{for } \beta > 0 \\ 1 & \text{for } \beta \leq 0 \end{cases} \quad (8)$$

The value of $\varphi_\kappa(\mathbf{q})$ varies between $[0, 1]$ for $\beta > 0$. β is the product of obstacle functions as defined in Eq.(6), $\beta \leq 0$ constitute the points ‘‘inside’’ the obstacle, and thus gives a maximum value of 1.

Here, we see that navigation function provides a solution to the local minima found in local potential field approaches; nevertheless it remains a global method. This means that it losses the advantages of local potential fields approach where only local information is needed. Further, while the repulsive potential function in local potential approaches have a limited range of influence (which is a desired feature), all obstacles in the navigation function contributes to the final shape of the potential field, no matter how far they located.

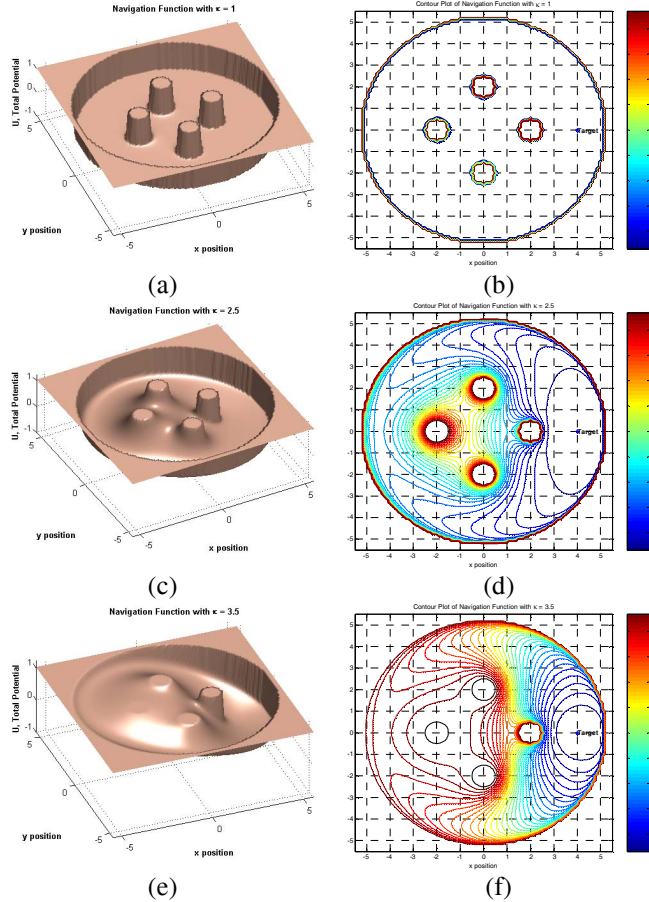


Fig. 2: The 3D visualization of a navigation function of a sphere world with four obstacles with (a) $\kappa=1.0$; (c) $\kappa=2.5$; and (e) $\kappa=3.5$, and their corresponding contour plots are given in (b), (d), and (f).

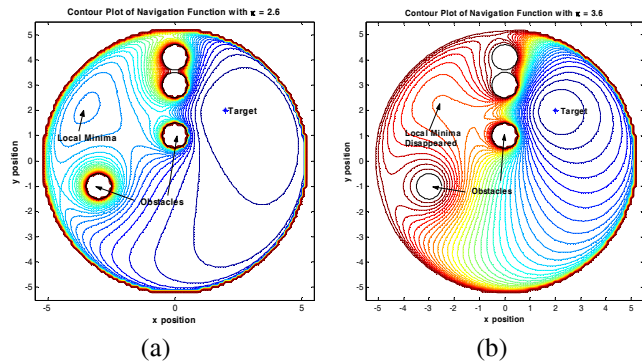


Fig. 3: Contour plot of a potential field for a workspace with four obstacles created using navigation function. In (a), local minima exist with a κ value of 2.6; and (b) Local minima disappeared as κ value increased to 3.6.

Further, Eq.(8) has the characteristic that it can provide a potential field with a unique minimum adjustable using a single tunable parameter κ . As an example, 3D potential field of a workspace with four obstacles and their contour are plotted in Fig. 2. In these figures, obstacles are located at $(2, 0)$, $(0, 2)$, $(-2, 0)$, and $(0, -2)$, with a radius of 0.5, and target is located at $(4, 0)$. Fig. 2(a)-(f) show that potential field

generated from navigation function has a smooth contour only with a proper selected κ value. For some arrangements of the obstacles, undesired local minima may exist at low value of κ . For example, Fig. 3(a) shows the contour plot where a κ value of 2.6 results in an undesirable local minima at $(-3.5, 2)$; which disappears as κ value is increased to 3.6 as shown in Fig. 3(b). Therefore, selecting a suitable κ value is critical but can be done offline. Hence, we develop a MATLAB GUI to aid the designer.

C. Design of a MATLAB GUI for Navigation Function

In the previous section, we have shown that an “adequate” κ value is necessary for finding a potential field without local minima – however, this value can vary significantly depending upon the workspace and the number, size and positioning of the various obstacles. Hence, it was useful to develop a tool to quickly determine the necessary κ value for a given workspace in order to subsequently evaluate motion planning algorithms for groups-of-robots. The interactive MATLAB-based Graphical User Interface (GUI), shown in Fig. 4, allow the designer to alter the size, number and location of various obstacles, change the κ value, visualize the contours, and interact with a 3D plot of the navigation function.

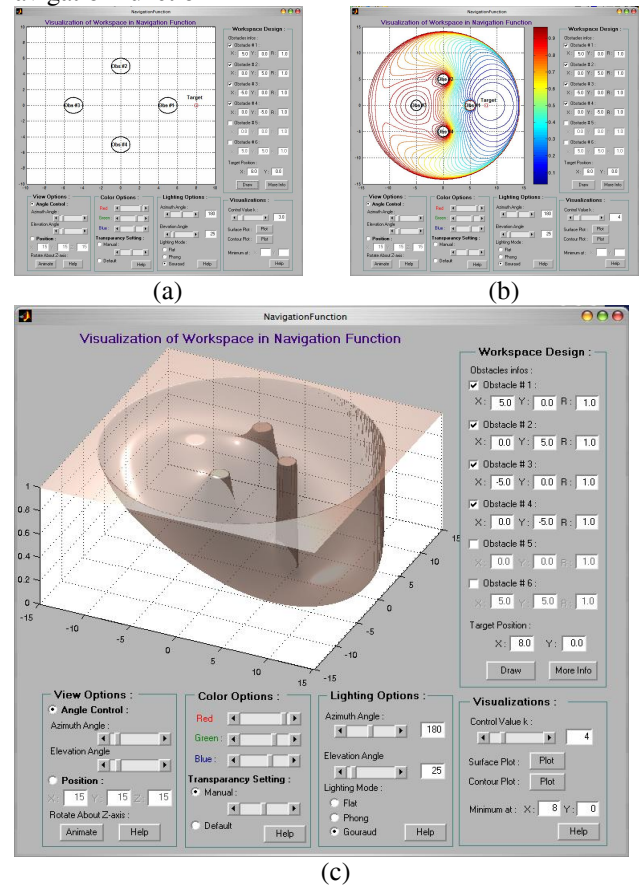


Fig. 4: The MATLAB GUI that allows a designer: (a) place obstacles of various sizes in the workspace; (b) visualize the contour of the navigation function of the workspace; and (c) visualize the potential field in 3D, and change κ value to obtain a smooth potential field with a unique minimum.

IV. FORMULATION AND SIMULATION RESULTS

In this paper, we implemented potential field approaches for motion planning of robot collectives. A “team” of differentially-driven nonholonomic wheeled mobile robots (NH-WMR) are shown in Fig. 5(a). Typically, the nonholonomic constraints of the individual wheels combine with the differentially driven architecture to limit the possible motion of such robots. Thus, any point along the wheel axle of the differentially driven wheels cannot move in the direction of the axis. However, all other points are not bound by this constraint which allows reduction of the NH-WMRs to an equivalent point mass, as is done in this paper.

A. Dynamic Equations Formulation

The dynamic equation of this group of n point-mass robot *without* formation maintenance is given by:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{u} - \mathbf{K}\dot{\mathbf{q}} \quad (9)$$

where $\mathbf{M} = \text{diag}\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n\}$, a $2n \times 2n$ diagonal mass matrix; $\mathbf{u} = -\mathbf{K}_f \nabla_{\mathbf{q}} U$, the input to the system is the negative gradient of the potential field U , and $\mathbf{K} = \text{diag}\{k_{x_1}, k_{y_1}, \dots, k_{x_n}, k_{y_n}\}$, also a $2n \times 2n$ diagonal control

matrix and $\nabla_{\mathbf{q}} U = \frac{\partial U}{\partial \mathbf{q}} = \left[\frac{\partial U}{\partial x_1}, \frac{\partial U}{\partial y_1}, \frac{\partial U}{\partial x_2}, \frac{\partial U}{\partial y_2}, \dots, \frac{\partial U}{\partial x_n}, \frac{\partial U}{\partial y_n} \right]^T$.

Equation (9) can also be written in the following matrix form:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{u} \quad (10)$$

with $\mathbf{u} = -\mathbf{K}_f \nabla_{\mathbf{q}} U$.

To include the formation maintenance in the dynamics equations, we note that this group of independent mobile robots moving together in formation and coupled together by constraint dynamics can alternatively be viewed as a constrained mechanical system. Hence, the dynamics of group of robots can be formulated as Lagrange Equation of the First Kind as:

$$\dot{\mathbf{q}} = \mathbf{v} \quad (11)$$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u}) - \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda} \quad (12)$$

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad (13)$$

where \mathbf{q} is the n -dimensional vector of generalized coordinates; \mathbf{v} is the n -dimensional vector of generalized velocities; $\mathbf{M}(\mathbf{q})$ is the $2n \times 2n$ dimensional inertia matrix; $\mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u})$ is the n -dimensional vector of external forces; \mathbf{u} is the vector of input forces, which is $-\mathbf{k}_f \nabla_{\mathbf{q}} U$; $\mathbf{C}(\mathbf{q}, t)$ is a m -dimensional vector of holonomic constraints; and $\mathbf{J}(\mathbf{q}) = \partial \mathbf{C}(\mathbf{q}) / \partial \mathbf{q}$ is the *Jacobian* matrix.

As a result, the formulation and computation of motion plans for such collectives in a potential field may be treated as being equivalent to simulating the forward dynamics of a constrained multi-body mechanical system. By doing so, we

can link and leverage the extensive literature on formulation and implementation of computational simulation of multibody systems [20-23]. Specifically, the constrained dynamics system may now be solved by using three methods: (i) direct Lagrange multiplier elimination approach [22, 24]; (ii) Penalty formulation approach [25]; or (iii) Constraint manifold projection approach [26-28]. In particular, the penalty formulation approach is very closely analogous to the potential field approach in [2, 3] and will be investigated in our work. A detailed treatment of the other two approaches can be found in [29].

In the penalty-formulation approach, the holonomic constraints are relaxed and replaced by linear/non-linear spring with dampers. This allows the constraint equations to be incorporated as an auxiliary dynamical system penalized by a large factor (as shown in Fig. 5(a)). Here, the Lagrange multipliers are explicitly approximated as the force of a virtual spring or damper based on the extent of the constraint violation and assumed spring stiffness and damping constant. The restoring force, which is proportional to the extent of the constraint violation, is expressed as:

$$\boldsymbol{\lambda} = \mathbf{K}_S \mathbf{C}(\mathbf{q}) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q}) \quad (14)$$

where \mathbf{K}_S is the spring constant term, and is given by $\mathbf{K}_S = \text{diag}\{K_{S1}, K_{S2}, \dots, K_{Sn}\}$, an $n \times n$ diagonal matrix; \mathbf{K}_D is the damping constant term, where $\mathbf{K}_D = \text{diag}\{K_{D1}, K_{D2}, \dots, K_{Dn}\}$, also an $n \times n$ diagonal matrix; and $\mathbf{C}(\mathbf{q})$ is the vector of constraint violation in the direction of the respective $\boldsymbol{\lambda}$. Substituting Eq.(14) into Eq.(12), the dynamic equation using penalty-formulation can be expressed as:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{M}^{-1}[\mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u}) - \mathbf{J}^T(\mathbf{K}_S \mathbf{C}(\mathbf{q}) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q}))] \end{bmatrix} \quad (15)$$

By doing this, the Lagrange multiplier is eliminated from the list of $n+m$ unknowns, leaving a system of $2n$ first order ODEs. On one hand, we note that this may creates a stiff dynamic equation with poor numerical conditioning, when a large penalty factor is selected. On the other hand, this spring force only approximates the true value of the constraint forces. Thus it may not be able to maintain tight formation, if a relative small penalty factor is selected. Note that with this formulation, we can also allow for shape change by letting:

$$\boldsymbol{\lambda} = \mathbf{K}_S \mathbf{C}(\mathbf{q}, t) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q}, t) \quad (16)$$

where the constraint matrix \mathbf{C} is now $\mathbf{C}(\mathbf{q}, t)$, a function of both position and time. Another advantage of approximating the Lagrange multiplier using the penalty approach is that fully decentralized formulation can be obtained without much effort. For example, the dynamics equation of a group of three point-mass robots, denote as A, B, and C ($n=3$), can be formulated using Eq.(16), and further

decentralized as:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{q}}_A \\ \ddot{\mathbf{q}}_A \end{bmatrix}_{4 \times 1} &= \begin{bmatrix} \mathbf{v}_A \\ \mathbf{M}_A^{-1} [\mathbf{E}_A \mathbf{u}_A - \mathbf{K}_A \dot{\mathbf{q}}_A - \mathbf{J}_A^T (\mathbf{K}_{S,A} \mathbf{C} + \mathbf{K}_{D,A} \dot{\mathbf{C}}_A)] \end{bmatrix} \\ \begin{bmatrix} \dot{\mathbf{q}}_B \\ \ddot{\mathbf{q}}_B \end{bmatrix}_{4 \times 1} &= \begin{bmatrix} \mathbf{v}_B \\ \mathbf{M}_B^{-1} [\mathbf{E}_B \mathbf{u}_B - \mathbf{K}_B \dot{\mathbf{q}}_B - \mathbf{J}_B^T (\mathbf{K}_{S,B} \mathbf{C} + \mathbf{K}_{D,B} \dot{\mathbf{C}}_B)] \end{bmatrix} \\ \begin{bmatrix} \dot{\mathbf{q}}_C \\ \ddot{\mathbf{q}}_C \end{bmatrix}_{4 \times 1} &= \begin{bmatrix} \mathbf{v}_C \\ \mathbf{M}_C^{-1} [\mathbf{E}_C \mathbf{u}_C - \mathbf{K}_C \dot{\mathbf{q}}_C - \mathbf{J}_C^T (\mathbf{K}_{S,C} \mathbf{C} + \mathbf{K}_{D,C} \dot{\mathbf{C}}_C)] \end{bmatrix} \end{aligned} \quad (17)$$

where $\mathbf{u}_i = \nabla_{\mathbf{q}_i} U$, $\dot{\mathbf{C}}_i = [\mathbf{J}_i][\dot{\mathbf{q}}_i]$, and $K_{S,i}, K_{D,i}$ with $i = A, B, C$ are the compliance matrices for springs and dampers respectively.

The three dynamic sub-systems shown in Eq.(17), can be simulated in a distributed manner if at every time step: (1) either the information pertaining to $\mathbf{C}(\mathbf{q})$, the extend of the constraint violation, is made available explicitly or (2) computed by exchanging information between the robots. The sole coupling between the three sub-parts is due to the Lagrange multipliers, which are now explicitly calculated using the virtual spring.

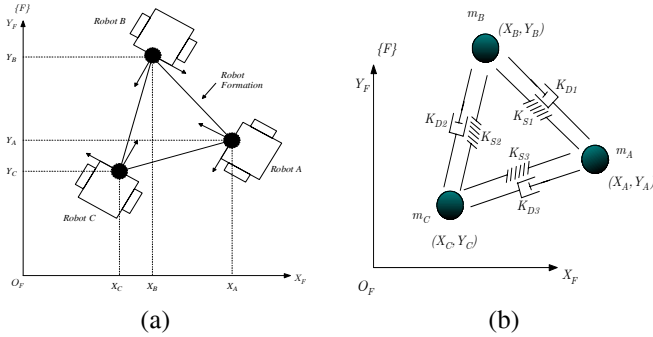


Fig. 5: (a) Three nonholonomic wheeled mobile robot in a triangular formation can be treated as three point mass robots by selecting a point away from the axle; and (b) The formation constraint is satisfied by approximating the Lagrange multiplier as springs and dampers.

B. Case studies

We performed three case studies using the formulation presented above on the common test course, shown in Fig. 6(a). Initially, we positioned three robots at $(-2, -3)$, $(-2, -4)$, and $(-2.866, -3.5)$ respectively with respect to an obstacle of radius 1.5 at the origin and the target located at $(2.5, 2.5)$. The potential field of the workspace, created using navigation function with $\kappa = 1.6$, is shown in Fig. 6(b).

We performed the simulation using MATLAB's fixed time-step solver, in consideration of actual implementation, where the information from sensors is evaluated in a specific fixed time-intervals. In particular, ODE5 (a fifth order Runge-Kutta Method) was chosen for its highest accuracy among other fixed time-step solvers. A fixed time step of 1×10^{-3} seconds was used for the total simulation time of 8 seconds.

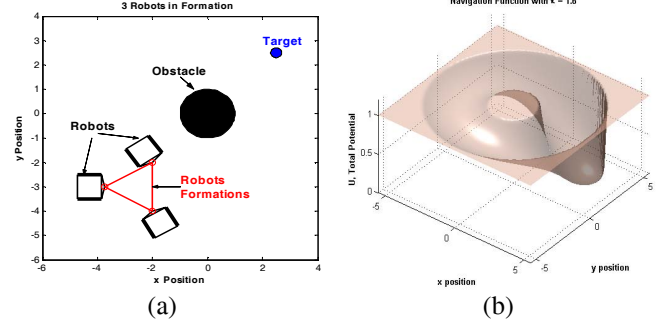


Fig. 6: (a) The simulation setup for case studies, shown here an obstacle between a group of three point-mass robots and the target location; and (b) the potential field of the workspace generated using the navigation function in 3D view.

As a baseline reference, the first case study is done using the dynamics equation *without incorporating the formation constraints*, as given in Eq.(10). The gradient information at each simulation step is obtained numerically for each robot. As the simulation shows, in the absence of the formation constraint the results depict 3 individual mobile robots converging to the common minimum. To quantify how well the formation is maintained, we study the total formation error, is given by:

$$\Delta_{Error} = \sqrt{\sum_{i=1}^M (c - \bar{c})_i^2} \quad (18)$$

where Δ_{Error} denote the total formation error, M denote number of holonomic constraints in the equation, c is the Euclidean distance of each holonomic constraint at each instant, and \bar{c} is the desired Euclidean distance for each holonomic constraint. Fig. 8 depicts the results of the second case study – where the constrained equations in Eq.(15) are simulated under identical conditions.

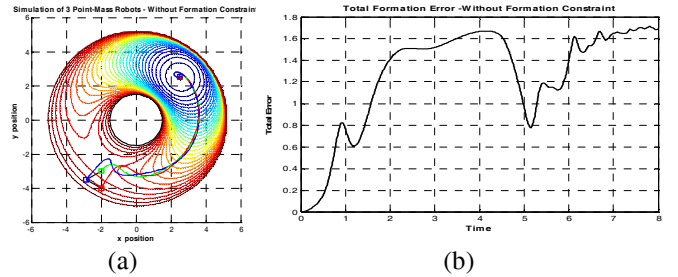


Fig. 7: (a) Simulation result showing the three robots traveling to the target by following the negative gradient of the potential, without formation maintenance; and (b) The total formation error throughout the simulation.

We notice that the formation error shows significant dependence on the value of the spring stiffness K_S and is order of 10^{-2} for $K_S = 100$. To further examine formation constraint maintenance, we study the parametric effect of raising the value of K_S . Fig. 8(b) shows the reduction in the total formation error as the value of K_S increases. We obtain a total formation error in the order of 10^{-3} with $K_S = 1000$,

with a simulation time-step of 10^{-3} . However, for value of $K_s > 1000$, the simulation become unstable as the equation stiffness increases requiring a reduction in the simulation step size. Several other studies, including formation-expansion and formation topological changes were also performed and discussed in [16].

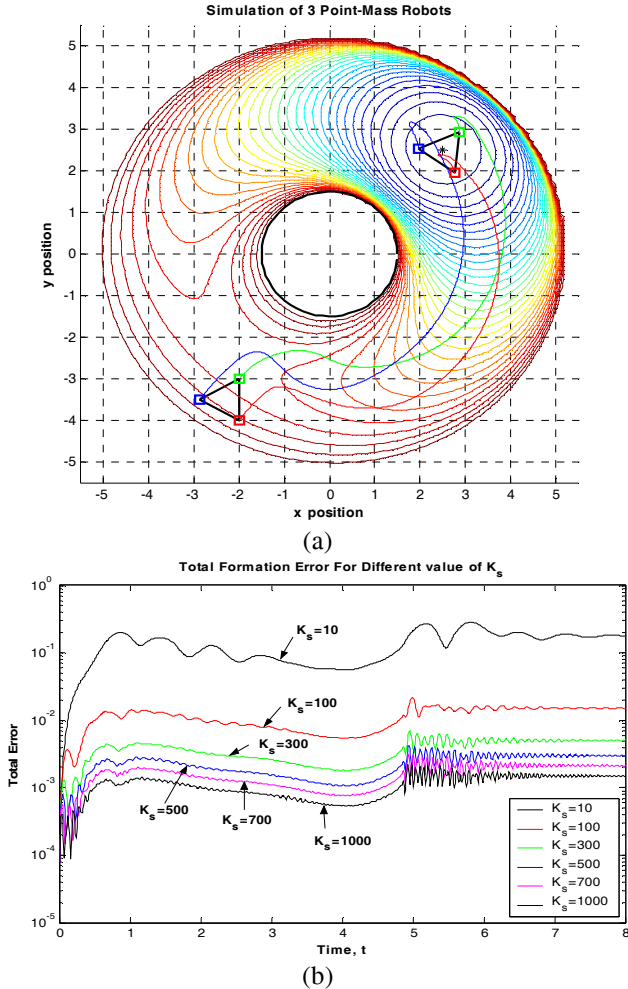


Fig. 8: (a) The simulation result showing the three robots in a triangular formation move from their initial position to the target position while maintaining formation; and (b) The total formation error for different value of spring constant K_s of the three robots in triangular formation case study.

For this paper, we also report a third case study, with 10 point-mass robots moving in formation that was performed to help study the scalability of this formulation. The initial locations of the robots are given by: (8, 11.46), (7, 9.73), (9, 9.73), (6, 8), (8, 8), (10, 8), (5, 6.26), (7, 6.26), (9, 6.26), (11, 6.26). The target is locate at (-7, -7), the obstacle is located at (0, 0) with a radius of 1.5, and the potential field is created using navigation function with $\kappa=1.7$. The sample simulation result, shown in Fig. 9(a) and the study of the parametric changes in spring constant K_s on the total formation error, shown in Fig. 9(b), are very similar to the previous case study.

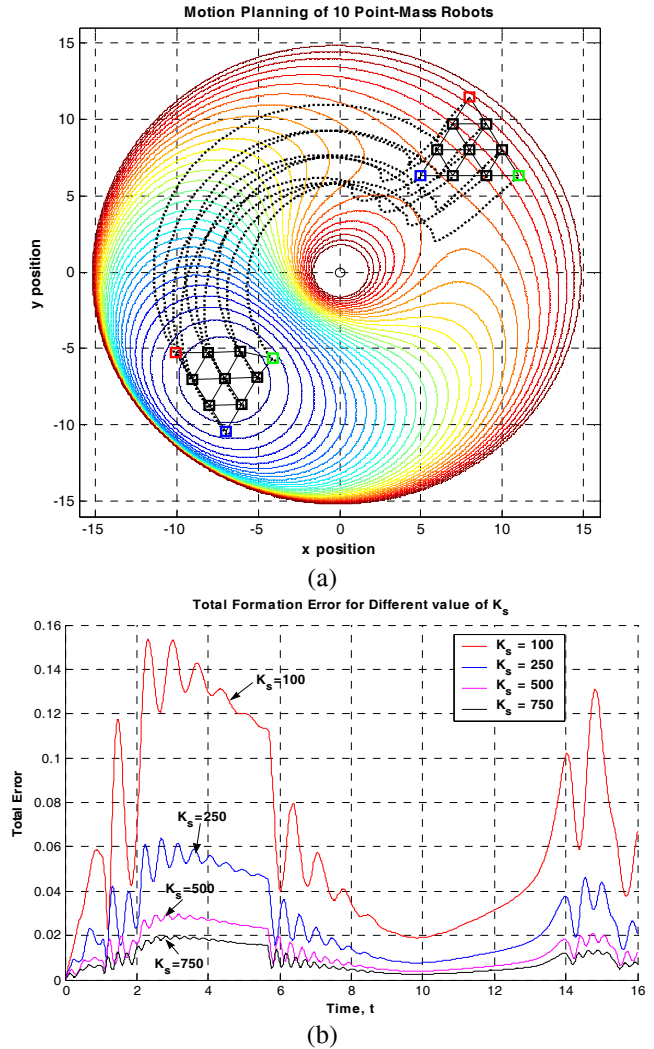


Fig. 9: (a) The simulation result of the 10 robots forming a interconnected triangular formation in a workspace with one obstacle; and (b) The total formation error plot for different value of K_s used in this simulation.

Hence, in practice, the selection of this spring constant value become critical as one tries to reconcile conflicting requirements for tight formation maintenance and avoiding ill-conditioned (stiff) formulations. Other approaches, such as constraint manifold projection [16, 29] exist but tend to be less amenable for decentralized implementations and/or more expensive in terms of computations-per-iteration. However, it is oftentimes more desirable to use these approaches in terms of their reduced overall formulation stiffness which makes for faster and stable overall computations.

IV. DISCUSSION & CONCLUSION

In this paper, we demonstrate the practical use of construction of a navigation function as the “standardized potential space” for evaluation of multi-robot cooperative payload transport algorithms. The navigation function provides a means to merge various limited-range

local-potential-fields into a “well-behaved” C-space potential function with a unique global minimum (that can be tuned using a parameter κ), and other useful characteristics. However, this comes at the cost of requiring complete knowledge of the workspace in order to construct the navigation function.

An interactive GUI interface was developed to alleviate the mathematical/computational complexity and tedium involved in merger of arbitrary local-limited-range potential functions into the navigation function. The tool allows us to quickly model the workspace and obstacle environment, select an appropriate κ value, and visualize the resulting navigation functions. Such standardization of testing-grounds allows algorithms for coordinated control of multi-robot collectives to be evaluated. This, for example, allowed us to study effects of potential/behavior-based constraints on performance of formation-maintenance operations. Other similar studies, leveraging this framework, are also currently underway.

ACKNOWLEDGEMENT

We gratefully acknowledge the support from The Research Foundation of State University of New York and National Science Foundation CAREER Award (IIS-0347653) for this research effort.

REFERENCES

- [1] Ogren, P., Egerstedt, M., and Hu., X., "A Control Lyapunov Function Approach to Multi-Agent Coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 847-851, 2002.
- [2] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1292-1302, 2004.
- [3] Ogren, P. and Leonard, N. E., "Obstacle Avoidance in Formation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, pp. 2492-2497.
- [4] Kumar, V., Zefran, M., and Ostrowski, J., "Intelligent Motion Planning and Control," in *Handbook of Industrial Robotics*, S. Nof, Ed.: John Wiley and Sons, 1999.
- [5] Cao, Y. U., Fukunaga, A., and Kahng, A., "Cooperative Mobile Robotics: Antecedents and Directions," *Autonomous Robots*, vol. 4, pp. 1-23, 1997.
- [6] Ge, S. S. and Cui, Y. J., "Dynamic Motion Planning for Mobile Robots Using Potential Field Method," *Autonomous Robots*, vol. 13, pp. 207-222, 2002.
- [7] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotic Research*, vol. 5, pp. 90-98, 1986.
- [8] Kim, J.-O. and Khosla, P. K., "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 338-349, 1992.
- [9] Leonard, N. E. and Fiorelli, E., "Virtual Leaders, Artificial Potentials and Coordinated Control of Groups," in *Proceedings of IEEE Conference on Decision and Control*, 2001, pp. 2968-2973.
- [10] Rimon, E. and Koditschek, D. E., "Exact Robot Navigation Using Artificial Potential Functions," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 501-518, 1992.
- [11] Krogh, B., "A Generalized Potential Field Approach to Obstacle Avoidance Control," in *Proceedings of ASME Conference of Robotic Research: The Next Five Years and Beyond*, 1984.
- [12] Volpe, R. and Khosla, P., "Manipulator Control with Superquadric Artificial Potential Functions: Theory and Experiments," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, pp. 1423-1436, 1990.
- [13] Ge, S. S. and Cui, Y. J., "New Potential Functions for Mobile Robot Path Planning," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 615-620, 2002.
- [14] Connolly, C. I., "Applications of Harmonic Functions to Robotics," in *Proceedings of IEEE International Symposium on Intelligent Control*, 1992, pp. 498-502.
- [15] Khosla, P. and Volpe, R., "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1988, pp. 1778-1784.
- [16] Lee, L.-F., "Decentralized Motion Planning Within an Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-Robot Collectives.," *Mechanical & Aerospace Engineering*, State University of New York at Buffalo, NY, Buffalo, 2004.
- [17] Koditschek, D. E. and Rimon, E., "Robot Navigation Functions on Manifolds with Boundary," *Advances in Applied Mathematics*, vol. 11, pp. 412-442, 1990.
- [18] Rimon, E. and Koditschek, D. E., "The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Sphere Worlds," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1989, pp. 21-26.
- [19] Rimon, E. and Koditschek, D. E., "The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds," *Transactions of the American Mathematical Society*, vol. 327, pp. 71-115, 1991.
- [20] Arnold, V., *Mathematical Methods of Classical Mechanics*, 2 ed. New York: Springer-Verlag, 1989.
- [21] Ascher, U., Chin, H., Petzold, L., and Reich, S., "Stabilization of Constrained Mechanical System with DOEs and Invariant Manifolds," *Mechanical Structures & Machines*, vol. 23, pp. 135-158, 1995.
- [22] García de Jalón, J. and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. New York: Springer-Verlag, 1994.
- [23] Haug, E. J., *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods: Allyn and Bacon Series in Engineering*, Prentice Hall, 1989.
- [24] Witkin, A., Gleicher, M., and Welch, W., "Interactive Dynamics," in *Proceedings of 1990 Symposium on Interactive 3D Graphics*, 1990, pp. 11-21.
- [25] Wang, J., Gosselin, C. M., and Cheng, L., "Modeling and Simulation of Robotic Systems with Closed Kinematic Chains using the Virtual Spring Approach," *Multibody System Dynamics*, vol. 7, pp. 145-170, 2000.
- [26] Khan, W. A., Tang, C. P., and Krovi, V., "Modular and Distributed Forward Dynamic Simulation of Constrained Mechanical Systems - A Comparative Study," *Mechanism and Machine Theory*, 2006. (In press)
- [27] Sarkar, N., Yun, X., and Kumar, V., "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots," *International Journal of Robotic Research*, vol. 13, pp. 55-69, 1994.
- [28] Yun, X. and Sarkar, N., "Unified Formulation of Robotic Systems with Holonomic and Nonholonomic Constraints," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 640-650, 1998.
- [29] Lee, L.-F., Bhatt, R. M., and Krovi, V., "Comparison of Alternate Methods for Distributed Motion Planning of Robot Collectives within a Potential-Field Framework," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005.

A Testbed for Heterogeneous Autonomous Collaborative Agents

Sharanabasaweshwara Asundi
asundi@ufl.edu

Andrew Waldrum
awaldrum@ufl.edu

Norman Fitz-Coy
nfc@ufl.edu

Space Systems Group
Department of Mechanical and Aerospace Engineering
P. O. Box 116205, University of Florida
Gainesville, FL 32611-6250, USA

Abstract – This paper presents lessons learned in the development of a testbed for the evaluation of heterogeneous, collaborative autonomous agents. Specifically, the paper address the design, fabrication, and preliminary evaluation of two wheeled robots, AWESIMO and LAWS-V, built on an E-MAXX Model 3906 RC monster truck. The heterogeneity of the vehicles, hence the system, arise from (i) the selected processor platforms, (ii) the modifications of the drive systems, and (iii) the selected sensor suites. Additionally, lessons learned in the application of on-board system health monitoring are also presented. Recommendations and the scope for future developments of these robots and the testbed are discussed.

I. INTRODUCTION

The AWESIMO (Autonomous Wheeled Experimental System for the Investigation of Multiple Objectives) and LAWS-V (Linux-based Autonomous Wireless-Capable Self-Diagnosing Vehicle) robots were motivated by the requirements of the HEROS (Heterogeneous Expert Robots for On-orbit Servicing) satellite project to have a testbed for verification/validation of subsystems and control methodologies. The HEROS (Fig. 1) project is a DoD funded study of cooperative heterogeneous autonomous specialized spacecraft performing advanced on-orbit operations. One of the goals of HEROS is to prove the feasibility of autonomous on-orbit operations, particularly servicing, by a team of cost effective cooperating heterogenous robotic satellites.

To help achieve this goal, monster truck based robots were conceived to test, among other things, real time health management and the feasibility of integrating cutting-edge range sensors with autonomous control in a small package.

In addition, the testbed has the ability to address some of the technical challenges of the US Army's FCS (Future Combat Systems) program, particularly in the area of Unmanned Ground Vehicle (UGV) Systems. The FCS program has three

classes of unmanned ground vehicles, one of which is the SUGV (Small Unmanned Ground Vehicle), a small, lightweight, man-portable UGV capable of conducting military operations in urban terrain tunnels, sewers and caves [7]. The two robots discussed in this paper have the capability of being an SUGV; both robots are well under 30 pounds, are capable of carrying up to 6 pounds of payload weight and have the potential to be controlled through vision.

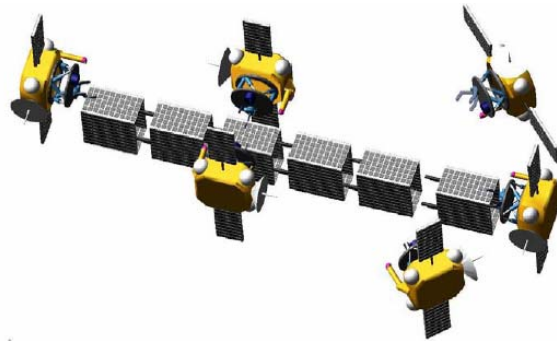


Fig. 1. HEROS – A rendition by Fred Leve

II. ROBOT SYSTEM ARCHITECTURE

A. Physical Platform

The autonomous collaborative robots, AWESIMO and LAWS-V (Fig. 2), were built on a customized E-MAXX Model 3906 RC monster truck platform. The rationale behind adopting RC monster trucks for the physical platform was to make the robots capable of traversing on different terrains with minimal investment in time and money for vehicle mobility. Originally, the monster trucks were equipped with two Twin Titan™ 550 motors that were powered by two 7-volt 6-cell Ni-MH batteries in series. The actuation on the trucks was made up of i) Traxxas 2055 high torque steering servo (80 oz/in of torque), ii) two-speed transmission servo

and iii) Traxxas EVX electronic speed control. Each monster truck was also equipped with WideMaxx™ 4-wheel independent suspension, efficient shaft drive 4WD, oil-filled shocks and a semi-tub molded chassis which made the E-MAXX platform ideal for multi-terrain application.



Fig. 2. LAWS-V and AWESIMO

The E-MAXX was designed for speeds in excess of 25MPH (440 inches per second), far above the speeds of 1-20 inches per second desired for our purposes. In an attempt to lower the operational speed of the vehicles, one of the Twin Titan™ 550 motors was disconnected on both robots. Additionally a 2.5:1 reduction gear was mounted on AWESIMO which brought down the speed of the robot by approximately 30%. This was not an ideal solution as the extra gearing introduced considerable backlash in the system and the increased torque mitigated the potential speed reduction. On LAWS-V the ground and VCC on one of the battery inputs was shorted to reduce the voltage to the speed controller from 14V to 7V. This reduced the vehicle speed by approximately 1/6th of its original minimum speed with concurrent increase in control sensitivity. This proved to be the most effective speed reduction technique.

The factory mounted shock springs on the monster trucks were not stiff enough to support the additional weight of the computer, sensors, and extra batteries used to power the onboard processor and sensors. These were replaced with Team Trinity's 8.5LB (136oz) heavy springs (TK5128 T/E-MAXX XX-TRA HEAVY).

B. System Hardware and Software

In an effort to build a heterogeneous testbed the robot platforms were built using different system architectures although each was interfaced with the actuation unit through the Mini SSC (Serial Servo Controller) II, described in sub section 3) below. The AWESIMO was built on an EPIA-M Mini-ITX Mainboard with a VIA C3TM E-Series x86 processor while the LAWS-V was built on a PC104 platform. LAWS-V was operated on Debian Linux and AWESIMO was operated on Gentoo Linux.

The AWESIMO robot was configured with a Mini-ITX

Mainboard single board processor unit having 128MB RAM and all the standard interfaces of a desktop PC - serial mouse, keyboard, monitor, USB, Ethernet & RS232. The processor was powered by a standard 12V supply unit connected to external power for testing purposes and by a PW-80-WV power supply using a 15V Li-Polymer battery for autonomous application. The Mini-ITX Mainboard was also made capable of wireless communication by mounting an 802.11g wireless LAN card on its motherboard.

On this platform, a 2006.0 Gentoo Linux operating system running the KDE windows manager was installed. The operating system default kernel was recompiled to support the wireless drivers. The Gentoo system took considerable time to install because each of its packages had to be compiled from source. Drivers to drive the Mini SSC II and a beta version (ver. 15.02.2005) of the Swiss Ranger driver were also loaded on the operating system. Programs were compiled using the GCC compiler and the AWESIMO robot was controlled remotely over a VNC connection (tightVNC).

The LAWS-V robot was configured with a PC104 Unit consisting of 4 modules: (i) MOPSLcd6 integrated CPU board having an Intel Pentium 266 MHz (32 KB cache), 2 standard RS232 serial ports, and 64 MB of SDRAM; (ii) V104 Vehicle Power Supply which with a 25W output, 6V to 40V DC input range, standard 5V and 12V and optional -5V and -12V outputs; (iii) Diamond-MM-16AT 16Bit Analog I/O Module with 16 single-ended (8 differential) analog inputs of 16-bit A/D resolution, a 100KHz maximum aggregate A/D sampling rate and 4 optional analog outputs & 8 dedicated digital I/Os with TTL compatibility; (iv) Kontron Dual Slot PCMCIA Adapter. The PC104 unit was powered by a standard 12V supply unit connected to external power for testing purposes and standard 11.1V Li-Poly battery pack provided power when operated autonomously. Although the order of arrangement of each module in the stack was not fixed, placing the power supply unit at the bottom and the PCMCIA adapter at the top worked best. The operating system was loaded into a laptop hard drive which was firmly fixed at the bottom of the PC104 stack with a hard layer of insulation separating it from the power supply unit. A PC104 peripheral interface board was also mounted to facilitate an LCD (or CRT) connection. We learned it was very important to have the hard drive connected to the actual hardware when installing the operating system and other software. For this purpose, an IDE to laptop hard drive adapter was acquired and a CDROM was connected along with the hard drive to load the operating system.

After configuring the basic hardware needed to make LAWS-V functional, it was then loaded with a Debian 3.0 Linux (Kernel 2.6.8) operating system having a command line interface. Some of the important packages on the operating system were the GCC compiler, a browser with SSL support, an SSH and a VNC server for remote communication and Tcl/Tk and Java support for Livingstone [3]. Once the operating system was up and running the driver module for the

PCMCIA adapter and the CISCO Aironet 340 series wireless LAN card were installed. The most significant and difficult part was loading the driver module for the analog I/O board. The analog I/O module from Diamond Systems Corporation requires that the operating system kernel source be available on the hard disk for driver compilation. The Debian Linux 3.0 was initially installed with a 2.4.X kernel, but due to difficulties in compiling a new kernel source on this operating system the Debian Linux 3.0 was reinstalled with a 2.6.x kernel. Over this, a new source of kernel 2.6.8 was compiled and deployed successfully. Then the driver module for the analog I/O board was compiled successfully with this kernel source. During autonomous operation when the robot got too far away from the wireless router it failed to send feedback to the computer it made a remote connection with. When SSH was used to make this remote connection the robot stalled during its operation. When the remote connection was made through VNC the robot seemed to perform pretty well although it was not able to send feedback to the computer it was connected to.

Building a processor unit on the LAWS-V robot was more cumbersome than building it on AWESIMO. A single board computer was mounted on the AWESIMO while LAWS-V had a stack of electronic boards put together to make up the processor unit. LAWS-V integrated a suite of sensors to make up its sensor unit while the AWESIMO had one powerful sensor. The AWESIMO had all the standard interfaces (mouse, keyboard, USB, monitor) inbuilt but adapters were required to connect the peripheral devices on LAWS-V.

The other significant hardware unit mounted on the two robots was the Mini SSC II. The Mini SSC II, a servo/motor driver, was equipped with a phone-style jack at one end and the other end went into one of the COM ports on the processor unit. Instructions were sent from the CPU unit to this device at 2400/9600 bps. The Mini SSC II drew 7 to 10 VDC at 10mA for its operation and drove the servos at 4.8 to 6.0 VDC with varying current.

Apart from the processor unit and the Mini SSC II an interface board was fabricated and mounted on the LAWS-V vehicle to normalize the connection pattern between the I/O board and the sensors. The interface board had a common 5V power out for the sensors which was supplied from the PC104 along with the ground. The interface board also acted as a channel for I/O signals from the sensors to the Analog board. The AWESIMO robot had only the Swiss Ranger (see below) as a sensor and so an interface board was not required.

C. Sensor Suite

In their current configurations, the sensor suite used for the two robots are very different. The AWESIMO robot is equipped with only one sensor, the Swiss Ranger, while the LAWS-V robot integrates a suite of several sensors including IR sensors for obstacle avoidance, Photoresistors for object tracking, bump switches, current sensors, fuel gauges for monitoring the battery capacitance, line tracker and a precision navigation compass.

1) Swiss Ranger – The Swiss Ranger (Fig. 3a), a 3D range sensor built on the time of flight principals and capable of detecting distance over a range of 0.05-7.5 meters at a resolution of 126X160 under ideal conditions, is mounted statically on the AWESIMO robot. For a detailed explanation of how the Swiss Ranger works, see Thierry, et al. [6]. Because the ranger information depends on the reflected light the practical distance under most lighting conditions is 3 meters. The ranger can be operated at a maximum sampling rate of 30Hz. The Swiss Ranger was primarily designed for MS Windows applications and thus, the drivers and testing software for the ranger in Linux are primitive with several bugs. One such bug included a switching of rows and columns in the data acquisition software output which skewed the images making objects look like noise.

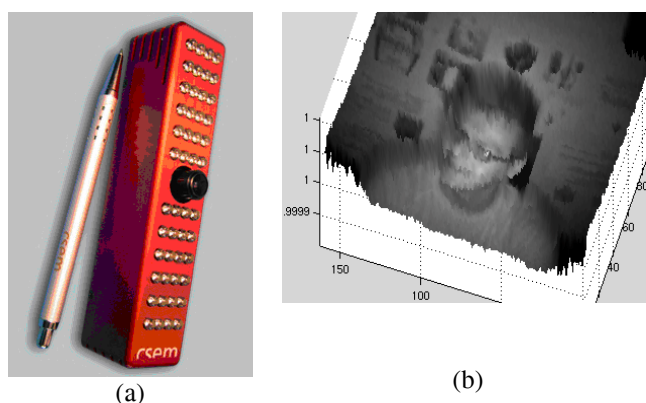


Fig. 3. a) Swiss Ranger b) 3-D Range Data

When too much modulated light is returned to the Ranger pixel from a target, the pixel saturates and gives the maximum distance value for those pixels. Additionally, when too little light is returned, the pixel returns a minimum distance and in between this value and the light required for good data, the ranger gives semi-random noise. These factors, giving distant ranges for close things, close ranges for distant things, guesses for values in between, and skewing 160X124 over a 124X160 display caused much delay in getting good data. Good initial images (Fig. 3b) were obtained by fixing the errors in the testing software and placing non-reflective objects in the field of view no less than 4 feet and no farther than 8 feet from the ranger. Additionally, a flat surface was placed behind these objects. These steps insured all objects in the field of view returned strong but not saturated signals. After good data had been taken and characterized, appropriate software filters were implemented and the restrictive testing parameters relaxed. Currently, no flat surface or range limits are required to achieve effective imaging though saturation of pixels from objects near the camera is still an issue.

2) IR Sensor Suite - The LAWS-V robot is equipped with three Sharp GP2D12 IR Sensors (Fig. 4a) on the front for obstacle avoidance. This sensor outputs an analog voltage and is captured by the PC104 analog I/O board that varies as a

function of the distance to an object obstructing the emitted infrared beam. The sensors on the left and right of the robot are mounted such that the emitted infrared beam is about 60 degrees to the path the robot is traversing.

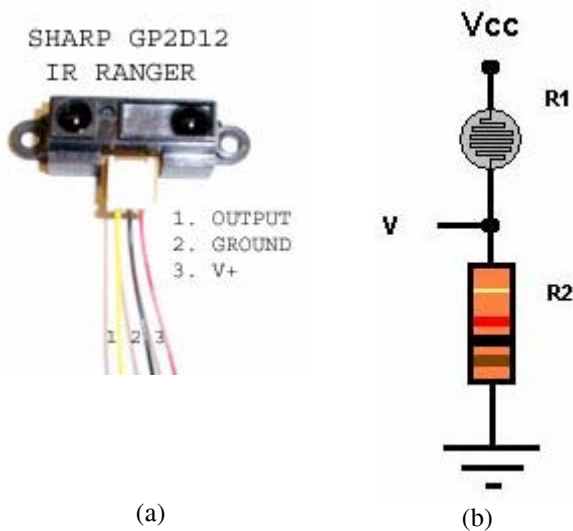


Fig. 4. a) IR Sensor b) Photoresistors Circuit

3) Photoresistors – Two sets of photoresistors (Fig. 4b) mounted on the front of the LAWS-V robot make it capable of following an object with a light source (preferably an LED light source). The photoresistors are sheathed in shrink tubes so the effects of ambient light are minimized. The change in voltage is captured by the onboard analog I/O board.

4) Bump Switches – Two bump switches (Fig. 5a) mounted on the front of the LAWS-V robot get activated when the robot collides into an obstacle. The bump switches, from Vex Robotics, output a digital “0” or a “1”. Springs were attached (glued) to the front of the bump switches to extend the reach of the switch and to keep the vehicle from getting very close to an obstacle.

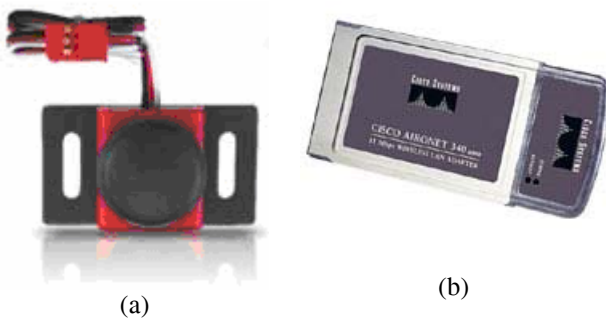


Fig. 5. a) Bump Switch b) Wireless LAN Card

5) Wireless LAN Card – The wireless LAN card on the LAWS-V robot, a Cisco Aironet 340 series PC Card Client Adapter is used as a communication device on the robot and as a sensor to sense the signal strength during autonomous

operation. It operates perfectly in a radius of 10-15 meters around the wireless router. The health management system on LAWS-V measures the signal strength as a state of the system through this device and associates this as a failure cause of a component. A more detailed discussion of the health management system appears in section IV of this paper.

6) Line Tracking Sensor – The LAWS-V robot is equipped with a line tracking sensor from Lynxmotion. The sensor is mounted on the axle carrying the front wheels. The tracker has 3 infrared reflective sensors which deliver 3 digital signals to enable the robot to follow a black line on a white background or vice versa. The line tracking sensor makes the LAWS-V robot capable of guided navigation.



Fig. 6. Lynxmotion Line Tracker

The LAWS-V robot is also equipped with a TCM2-20 precision 3-Axis orientation-sensing instrument from PNI Corporation with an RS232c interface. Work on integrating this sensor is ongoing.

III. OPERATIONAL MODES

Presently the robots are at similar levels of functionality but using very different means. The AWESIMO vehicle can traverse semi-randomly, avoiding obstacles along the way, and can do simple object tracking when a single object is fitted with a cube corner reflector. The LAWS-V robot is capable of guided, semi-guided and autonomous motion.

A. AWESIMO

AWESIMO presently employs a three state obstacle avoidance algorithm, setting distance limits ahead, to the left and right. An object tracking algorithm is also being implemented on the AWESIMO robot. As mentioned earlier, the Swiss Ranger’s pixels saturate and return high values if too much light is returned. This property of the Swiss Ranger can be used to track objects by placing a corner cube reflector on the object being tracked. The saturated pixels of the ranger indicate the object’s relative bearing while range information is taken from the area around the reflector, presumably the object being tracked. Implementation of object tracking is ongoing.

B. LAWS-V

The line tracker and the photoresistors make the LAWS-V robot capable of guided motion and IR sensor suite enables the robot to traverse autonomously through obstacles. The IR sensor and the photoresistors can be used in conjunction to do object (light) tracking. Three Sharp GP2D12 IR sensors mounted in the front assist the robot in autonomous traverse. The left and the right IR sensors are pointing away from the line of motion. This enables the robot to “see” objects towards its right and left. The IR sensors are calibrated but not dynamically. When the IR sensor detects an object right in the center of its pathway it backups up for a while and takes a random turn to continue its autonomous behavior. The random algorithm has proven adequate in enabling the robot to traverse autonomously through obstacles. The vehicle is also capable of ignoring an occasional misreported value by the IR sensors. More sophisticated obstacle avoidance algorithms are being investigated and are being integrated into the task specific path planning routine.

Object tracking, a semi-guided operational mode, on the LAWS-V robot is done by focusing a less divergent beam of LED light onto the two sets of photoresistors mounted at the front of the robot. The photoresistors are mounted with a pipe around them so that they are activated by a light source directly in front of them. The photoresistors are generally very sensitive to the ambient light. So in order to distinguish the tracking source of light from the ambient light the robot goes through a calibration phase. During the calibration phase the robot wanders around in four directions and then captures the light intensity as a voltage value. Subsequently the robot compares the intensity of the tracking source with the calibrated values and makes its decision of taking a left turn, right turn or keeping straight.

IV. HEALTH MANAGEMENT SYSTEM

The health management system on the robot consists of a state acquisition phase and a decision making phase. In the state acquisition phase, the voltage and current sensors will provide information on the state of the battery, the servos and the guidance sensors (IR sensors) to the decision making algorithm. In the decision making phase, the inference engine of Livingstone (L2) [4] kernel will use the L2 model of the robot to perform a diagnosis of its health.

Currently an L2 model of the robot, an inference engine of L2 and its real-time interface reside on the LAWS-V robot. L2’s real-time interface, called the RTAPI (Real Time Application Programming Interface), interfaces the application layer (program controlling the robots motion) with L2s inference engine. The application is thus rendered capable of interacting with sensor module for acquiring vehicle health related data and then perform a diagnosis using the component based model of LAWS-V.

A. Robot Model

For Livingstone, a model is a data structure that represents the real-world device, the faults of which Livingstone diagnoses [1]. An L2 model is made up of components and connections. Components represent parts of a system and are characterized by one or more I/O terminals and modes (nominal and fault). Connections represent the influence of one component over the other. An L2 model of the LAWS-V robot built using Stanley, a model-building GUI tool written in Tcl (Tool command language), is shown in Fig. 7. The model is made up of these components – a PC104 unit, a Mini SSC II, an analog I/O module, an interface board, a wireless card, a power supply unit, an LCD unit and the servos and IR sensors. The components being monitored for faulty behavior are the servos, the IR sensors and the LCD unit. The states of these components are simulated and the L2 engine is requested to diagnose the fault in the system.

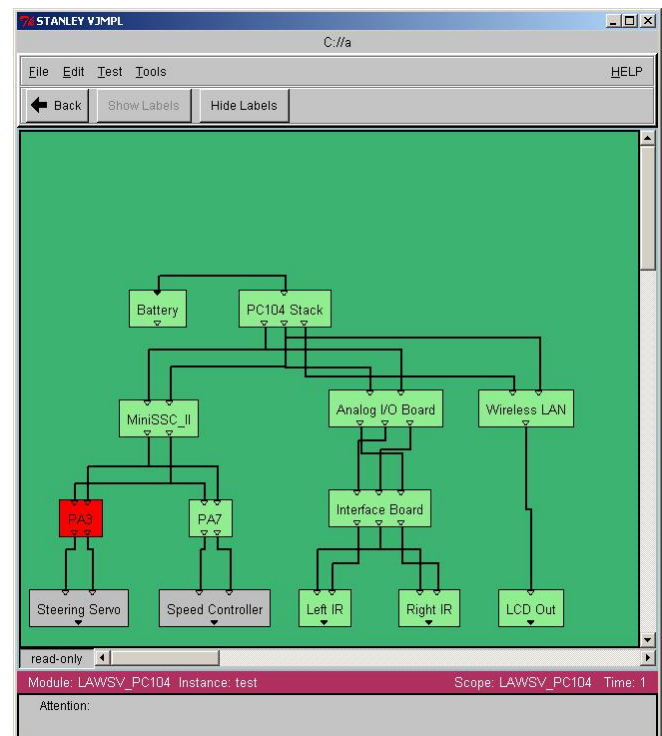


Fig. 7. Stanley model of the LAWS-V robot

B. Simulation Results

Four scenarios, where one or more components could fail, have been considered for the purposes of understanding model-based diagnosis. These scenarios reflect real-world situations the robot could encounter and how the L2 based fault diagnosis can be used to identify the failure.

1) Scenario I: Steering servo fails – The robot is performing an assigned task when steering servo fails and the robot is not able to turn in either direction. Sensor data is sent to the L2 engine and a diagnosis is requested. L2 presents the

probable causes of failure as candidates in a “Candidate Manager” window as shown in Fig. 8.

As seen by L2 the first two causes of failure are – (i) a faulty jumper on the Mini SSC II (ii) a faulty steering servo. Both these failure candidates are assigned the same rank. Apart from the above two causes L2 also associates the failure to a combination of – (iii) a faulty Mini SSC and a faulty speed controller (iv) a faulty Mini SSC and a faulty jumper (v) a faulty PC104, a faulty speed controller and a faulty interface board. Although the causes (iii), (iv) and (v) are less likely to occur and hence ranked lower L2 sees this as another possibility.

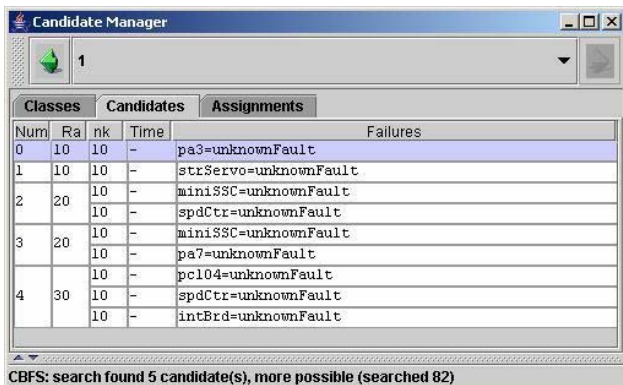


Fig. 8. L2 observations which indicate the probable causes for the failure of steering servo

2) Scenario II: Both Servos onboard fail – The robot is performing a task when steering servo and the speed controller fail. The results of the L2 diagnosis are shown in Fig. 9.

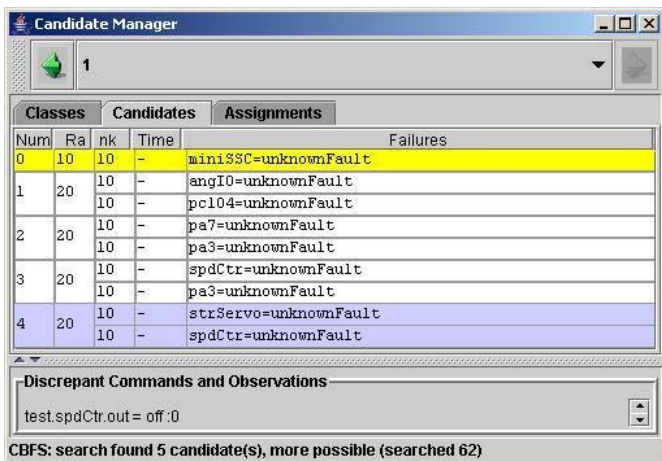


Fig. 9. L2 observations which indicate the probable causes for the failure of the servos

L2 identifies – i) a faulty Mini SSC II as the most probable cause of failure. It also identifies these – ii) faulty servos ii) faulty headers on the Mini SSC II iii) combination of a faulty servo and a faulty header as candidates of failure.

An additional cause listed by the L2 diagnosis – iv) a combination of a faulty PC104 unit and a faulty analog I/O

board. Although it’s less likely for such a case to exist L2 sees this as another candidate which could possibly cause a failure in the 2 servos.

3) Scenario III: Failure of both IR sensors – The Robot is performing an assigned task and it starts bumping into the side walls, which by design it is supposed to avoid. The results of the diagnosis are shown in Fig. 10.

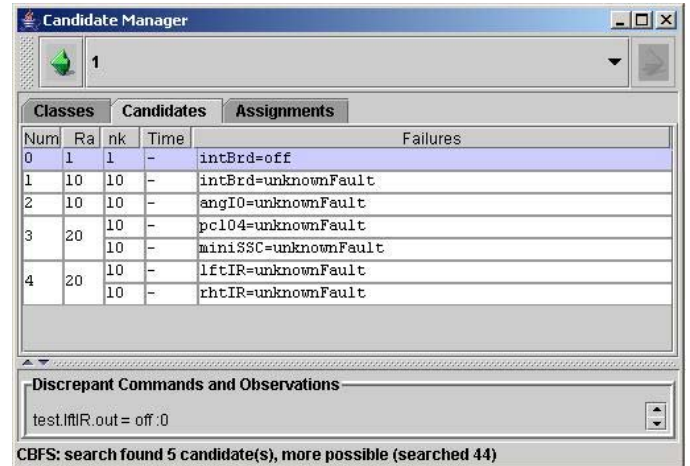


Fig. 10. L2 observations which indicate the probable causes for the failure of the IR sensors

L2 identifies – i) a turned off interface board as the most probable cause of failure. It ranks – ii) a faulty interface board and iii) a faulty analog I/O board below this. In addition L2 also identifies – iv) a combination of a faulty PC104 unit and a faulty Mini SSC II and v) faulty IR sensors as failure candidates.

4) Scenario IV – Failure of an IR sensor and the speed controller – The robot is performing an assigned task and the speed controller suddenly stops working. At the same time the robot is not able to see any obstacle on its left. The results of the diagnosis are shown in Fig. 11.

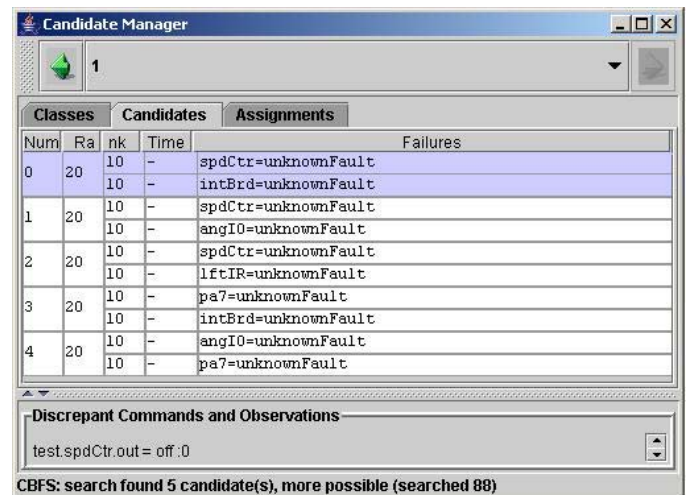


Fig. 11. L2 observations which indicate the probable causes for the failure of a servo and an IR sensor

The L2 engine performs a diagnosis and lists these – i) combination of a faulty speed controller and a faulty interface board ii) combination of a faulty speed controller and a faulty analog I/O board iii) combination of a faulty speed controller and a faulty IR iv) combination of a faulty jumper on the Mini SSC and a faulty interface board v) combination of a faulty jumper on the Mini SSC and faulty analog I/O board as the failure candidates.

V. CONCLUSION AND FUTURE SCOPE

Two heterogeneous mobile robots were developed by modifying E-MAXX model 3906 RC monster trucks. The robots represent autonomous collaborative agents in a testbed designed to evaluate elements of HEROS environment – cooperation, guidance, navigation and image-based control. In their current configurations, the robots are capable of operating in autonomous and guided modes and can communicate wirelessly to exchange critical information between themselves or to a ground station. Additionally, health monitoring based on the Livingstone diagnostic kernel is being integrated on the LAWS-V robot to demonstrate “intelligent self diagnostics”.

Future testbed developments involve demonstration of cooperative behavior between these robots (e.g., a leader-follower scenario in which one agent follows the other based on an electronic signature). Autonomous rendezvous and docking will also be demonstrated to show that autonomous servicing (e.g. refueling) of an agent is possible. To further demonstrate the servicing aspects, real time health monitoring

will be integrated into the robots so that they can self diagnose problems and then request servicing. Additional developments involve (i) efforts to perform simultaneous localization and mapping (SLAM) using the Swiss Ranger sensor on the AWESIMO robot and (ii) vision based guidance, navigation and control (e.g., feature point tracking).

REFERENCES

- [1] Bajwa, A & Sweet, A., “The Livingstone Model of a Main Propulsion System” *Proceedings, IEEE Aerospace Conference IEEE, 2002.*
- [2] L. Matthies, Y. Xiong, R. Hogg, D. Zhu, A. Rankin, B. Kennedy, M. Hebert, R. Maclachlan, C. Won, T. Frost, G. Sukhatme, M. McHenry, and S. Goldberg. A portable, autonomous urban reconnaissance robot. *In Intelligent Autonomous Systems, Venice, Italy, July 2000.*
- [3] J. Kurien and P. Nayak, “Back to the future for consistency-based trajectory tracking.” *7th National Conference on Artificial Intelligence (AAAI 2000).*
- [4] Williams, B. C., and Nayak, P. P., “A model-based approach to reactive self-configuring systems.” *Proceedings, AAAI-1996.*
- [5] Thierry Oggier, Peter Seitz and Nicolas Blanc, “Miniaturized all-solid-state 3D camera for real-time range imaging” *Proceedings, PerMIS-2004*
- [6] “Army AL&T (Acquisition, Logistics & Technology)” January – February 2004 issue
- [7] <http://www.cse.lehigh.edu/~spletzer/gollum.html>

Endurance Testing for Safety, Security, and Rescue Robots

Jeffrey A. Kramer
University of South Florida
4202 E. Fowler Ave. ENB 118
Tampa, FL 33620
jkramer2@csee.usf.edu

Robin R. Murphy
University of South Florida
4202 E. Fowler Ave. ENB 118
Tampa, FL 33620
murphy@csee.usf.edu

Abstract— This paper investigates the role of endurance testing for rescue and safety robotics. Endurance testing is a form of acceptance testing that verifies whether the robot can operate correctly over the intended period of operation. A six-hour endurance test was developed for the commercially available ASR micro-VGTV Extreme rescue robot. The test uncovered failures consistent with those previously encountered in the field but under conditions that captured the source of the failures. In addition, the data captured identified subtle design and manufacturing issues. Based on these results, this paper proposes that endurance testing become an additional requirement for standards for rescue and safety robotics and that developers adopt endurance testing as a general design and manufacturing diagnostic method. Specific recommendations on endurance testing are presented.

I. INTRODUCTION

Technology readiness is a major concern for safety, security, and rescue (SSR) applications. The question comes down to “Is a robot really ready to be used in the field?” One aspect of readiness is how likely the robot is to fail during operations. Prior work by Carlson and Murphy [1][2] have determined that most mobile robots have a mean time before failure (MTBF) of 6 to 20 hours – an unexpectedly low number. Observations during field studies under high fidelity conditions and actual responses suggest that some of these failures are due to designers not having sufficient understanding of the environmental demands of the urban search and rescue (US&R) domain. However, some of the failures encountered appear to be due to manufacturing defects or subtle interactions between components. These failures are not dependent on the environment and theoretically could be uncovered prior to deployment.

In order to detect and prevent failures before fielding a robot, this paper investigates one method for uncovering defects in a robot or a model of robot: endurance testing. Endurance testing is a facet of acceptance testing, where all functions (and combinations of functions) of the robot are tested continuously. Acceptance testing is used in more mature fields, such as industrial manipulators and factory automation, to verify that the robot is without defects. The advantage is that endurance testing is able to uncover failures that would not immediately show up in a demonstration. For example,

some failures might take several “runs” before the problem emerged. Other failures may be the result of an unexpected use of a combination of functions or positions of the robots. Demonstrations often reflect expectations of how a robot will be used, but given that SSR robots are an emerging technology, new actual use patterns are being constantly uncovered. Therefore, endurance testing exercises the scope of possible uses (e.g., provides coverage of the functionality space) and establishes the reliability of the robot over time.

Endurance testing can be applied in two ways. First, it can be incorporated into standards such as NIST and JAUS as a component of a field acceptance test. Standards for SSR robots have largely focused on functionality and interoperability; that is, confirming that the robot is capable of doing certain things with a minimum level of competence. What is lacking is how to determine the reliability of the robot; that is, establishing that the robot is capable of doing certain things with a minimum level of competence for the expected operation time. Operation times have not been set yet by the SSR standards community, but the US Department of Defense in [9] [10] [11] [12] [13] [14] [15] [16] [17] has recommended a minimum MTBF of 96 hours for ground robots (including man-packable UGVs which have a dual-use as SSR robots) and this has been proposed as the target for US&R robots in [3]. Assuming that there is a standard for MTBF, a means of verifying that a system meets that standard is needed.

Second, endurance testing can also be used by robot developers for projecting and diagnosing failures before the robot model is considered completed. This can be done by collecting internal data from the robot and video of its actions while undergoing.

This paper presents a methodology for endurance testing, discusses the findings and ramifications of the endurance testing for an ASR micro-VGTV Extreme rescue robot, and proposes an endurance test for user acceptance and a diagnostic endurance test and data collection methodology for robot manufacturers. The paper is organized as follows. It discusses the limited related work in robot reliability. Next it describes the methodology for conducting a diagnostic

endurance test, followed by the findings organized by the failure taxonomy developed in [1][2], and a discussion of the ramifications of the findings. The paper concludes with a recommended endurance test procedure to be included in a user acceptance test suite and a recommended diagnostic endurance test methodology to be adopted by robot developers.

II. RELATED WORK

The work here is motivated in part by our prior work in how robots fail [1][2] and appears unique. Acceptance testing and quality control procedures for unmanned ground vehicles does not appear in the literature, though a survey paper by Cavallaro and Walker discusses the need for such practices [6], and some preliminary work has been done by ASTM International with funding from the Department of Homeland Security [7]. Portions of this paper appear in [5]; this paper refines the ideas put forth in that paper and explicitly discusses how the findings might be integrated into the SSR standards effort being led by NIST and JAUS.

This article uses the taxonomy developed by Carlson and Murphy [1][2] to define the failures and create the terminology that will be used through the rest of this paper. It also uses the data contained in those two studies to confirm the results of this study. The types of failures on the robot side are by major subsystem: effector, sensor, control system, power and communication. The types of failures on the human side are design and two types of human interaction failures – mistakes and slips. Mistakes are caused when you misunderstand the environment and do the wrong thing. Slips are caused when the operator tries to do the right thing, but is unsuccessful. Failures can also be classified as terminal and non-terminal, as well as field-repairable or not field-repairable. Terminal failures are those that completely interrupt the robot’s mission, while non-terminal failures degrade the ability of the robot to perform its mission. A failure is considered to be field-repairable if it is able to be repaired with tools that are commonly included with the robot in good environmental conditions. See Fig. 1 for the taxonomy of robot failures. [2]

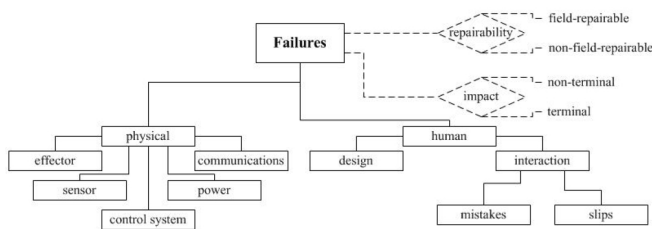


Fig. 1. Taxonomy of mobile robot failures used in analysis

III. METHODOLOGY

In order to perform an acceptance test on the robot, we go through a full-range exercise of the robot’s capabilities in a

contained and monitored environment. We tested the two robots in our study for 6 hours each. A simple test environment in the laboratory was set up. A square wooden frame was placed directly on a standard linoleum tile floor to form a robot-impassable barrier. The robot – see Fig. 2 – was operated via tether link and was connected directly to a PC running a control program written in Java. A camera on a tripod was set up to record the robot’s actions.



Fig. 2. The robot type tested

A simple test bed was set up directly on the laboratory floor. A frame was built to prevent the robot from running freely about the room. A standard digital video camera was set up to record the operation of the robot. Constructed of 4.5-foot long 2X4’s oriented with the width set vertically, the walls were tall enough to stop the robot from exiting the frame, yet short enough to allow the camera to film unobstructed. The floor under the frame was left bare to the linoleum tile.

The robot used in this study was an Inuktun MicroVGTV, a small man-packable robot. We used two separate robots in this implementation – robot 1 and robot 2. This robot is connected to its control interface via a tether. Its effector system is tracked. The Extreme was chosen for this project for two reasons. It is the only known platform marketed explicitly for urban search and rescue, and therefore considered a reasonable representative of SSR robots. Also, the Extreme, or an earlier version, have a known track record that allows comparison of failures detected by endurance testing to failures in the field. These robots been used in three disasters (World Trade Center[8], La Conchita Mudslides [3], and Hurricane Katrina) plus numerous field exercises [4].

The robot was put through a battery of actions designed to test all of its effectors and permutations of body shape. This made the robot travel through a full range of motion and provided data about the robot in all self-configurable states. The robot has the following active systems: a right track motor, a left track motor, a raise (body shape altering) motor, a tilt (camera angle) motor, a camera focus system, and lights.

All of these systems were tested both individually and together in all permutations. The equipped laser was not activated by the test set, as it is not eye safe and does not significantly impact the operation of the rest of the robot.

The robot's control interface was then connected to a PC running a control program written in Java. This program provided the commands to the robot and monitored the robot's state. It recorded the robot's state every second and stored it to a text file. The most applicable data types are listed in Table 1. Some of the data is not applicable to our testing, but all were collected for further analysis.

TABLE I
TYPES OF DATA COLLECTED AND SIGNIFICANCE

Data Collected	Significance
Time	Time (in seconds) of run duration
Raw Battery Value	A raw battery sensor value
Converted Battery Voltage	Actual battery voltage value
Percentage of Battery	Percentage of battery left
Focus Current	Current (in 0.1 A) drawn by focus
Left Track Speed	Commanded speed (from -255 to 255)
Right Track Speed	Commanded speed (from -255 to 255)
Left Track Current	Current (in 0.1 A) drawn by left track motor
Right Track Current	Current (in 0.1 A) drawn by right track motor
Light Current	Current (in 0.1 A) drawn by the light
Pitch of Frame	Robot's current orientation
Robot Temperature	The temperature inside the robot
Raise Current	Current (in 0.1 A) drawn by the raise motor
Tilt Current	Current (in 0.1 A) drawn by the tilt motor

The robot data was then analyzed using the WEKA data mining program, visualization software, and a custom data rendering system, as well as the camera data. By examining the tape, we were able to determine exactly when the robot experienced a failure. The failure point was then identified within the recorded data set as an additional entry to try and determine data clusters with. The data mining program provided data clustering as well as other statistical data analysis.

IV. FINDINGS

The testing uncovered several key facts: testing an individual robot is not sufficient, yet each individual that is tested can show a problem in the overall model's design, sensor data can be used to predict and possibly prevent failures, subtle problems can be very serious, and failures that show up in a simple lab test setup are typical of those that show up in the field. These robots failed in three of five ways outlined in the robot failures taxonomy – either a control failure, the most common (labeled A), a power failure (labeled B), or a communications failure (labeled C). A control failure occurred whenever the robot either did not respond to commands or failed to execute a command appropriately. A power failure occurred whenever the robot lost power or was

impaired in its performance by low power. A communications failure occurred whenever the control system could not communicate with the robot. Table 2 shows the frequency and types of errors, while Fig. 3 and 4 show the timeline for each robot and their respective failure points. The rest of the section covers each failure in order of occurrence then addresses higher level design issues.

TABLE II
TYPES AND NUMBERS OF FAILURES

Robot	Control Failure (A)	Effector Failure	Power Failure (B)	Communications Failure (C)	Sensing Failure
1	2	0	1	2	0
2	2	0	1	0	0
Total	4	0	2	2	0

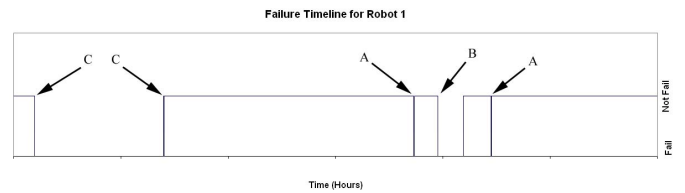


Fig. 3. Shows the timing and type of failures for Robot 1

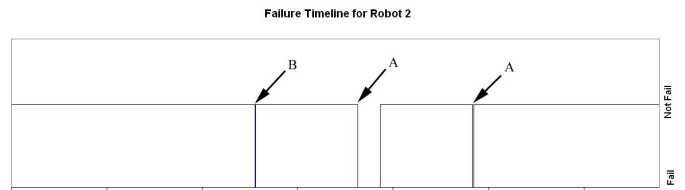


Fig. 4. Shows the timing and type of the failures for Robot 2

A. Control Failures

There were four control failures, two on Robot 1 and two on Robot 2. These failures can be further divided into two groups. The first group of failures, which happened once on each robot, could have been prevented. The imminent failures were presaged by a rapid change in pitch. The second group of failures, which happened once on each robot, show that the electronics have unmodeled behavior, as well as suggest a need for further manufacturer testing.

Two of these failures, both of which could have been prevented, were presaged by a rapid change in pitch, as shown in Figs. 5 and 6, followed by the robot flipping over. With better control software, the interface system could predict and possibly prevent human error in polymorphism, or a situation where the change in robot shape causes a system failure. This type of failure often happens when a human is using the robot – nothing on the robot failed, but

the robot is in a failure mode anyways. These failures were terminal, yet they required no repair. Figs. 5 and 6 illustrate several incipient failures – each a high spike on the pitch axis. If coupled with the robot’s control system appropriately, these high spikes can give warning to either the control system or to the human driver that they are operating in a precarious region. The clipped spikes that flip flop from -127 to 127 demonstrate a limitation of the sensor system – the robot is upside down at this time, yet the sensor cannot read the 180 degree position.



Fig. 5. Plot of pitch over time for robot 1 showing a failure presaged by fluctuations in pitch angle.

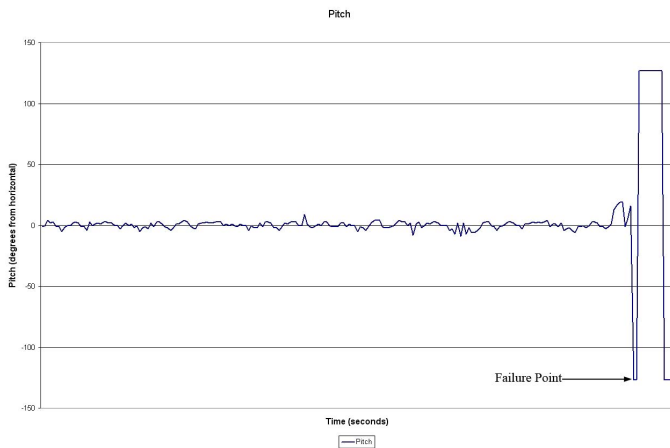


Fig. 6. Plot of pitch over time for robot 2 showing a failure presaged by fluctuations in pitch angle.

The two other control failures demonstrated that the electronics have unmodeled behavior, suggesting a need for more manufacturer testing. Robot 1 ran for 34 minutes and then lost control for 43 seconds. The software did not record any data during this time period, making the data points jump from 3508 seconds to 3551 seconds. The software automatically recovered, making this a field repairable, non-terminal error. Robot 2 was reset to begin testing after a tape change, but failed to respond to control – a terminal failure. Only by disconnecting the power and

letting the robot sit for 30 minutes was control restored. Both of these errors are unmodeled electrical problems. This shows a need for more manufacturer testing during development and before deployment. The testing method explained in this paper would constitute a reasonable factory acceptance test.

B. Power Failures

There were two power errors, one in each robot. Both of these failures reveal that we can and most likely should predict both battery loss and changes in performance. The robot’s batteries fail slowly, looking like minor power fluctuations at first – small blackouts in robot communication and minor video errors. These small errors were not immediately apparent to an external observer. The robots began to miss test steps, and then failed completely, moving only sporadically. By using the battery level data, we could have predicted the slowly growing errors in the rest of the system as shown in Fig. 7. The circled regions in Fig. 7 are places where the power to the robot failed, and therefore, the update to the control software stopped. The lower data spikes show where the motor was not being driven at all or being driven slowly.

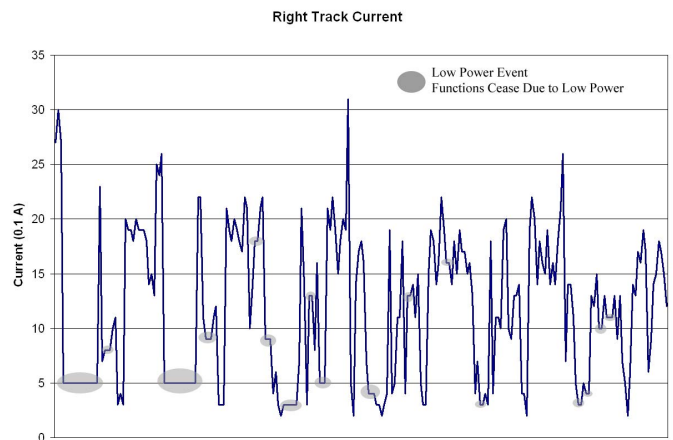


Fig. 7. Plot of right track current over time for Robot 2 showing low power events.

C. Communication Failures

Despite being tethered, this failure occurred twice in Robot 1 indicating poor design. The second failure also illustrates how serious problems can be very subtle. This error is usually due to the loss of wireless communications; however, this was caused by complex robot-environment interaction. Robot 1 ran for 12 minutes before failing. The robot backed up into one of the walls of the frame, bending the tether sharply upwards. This caused immediate communications failure – a terminal failure. Upon retrieving the robot, the tether was unconnected and reconnected, reestablishing communication. This type of failure is caused by poor design – the tether has been

reinforced against this type of failure, however, even in a controlled lab environment, it still occurs. In the cluttered environment of actual field use, this would occur even more often – the environment is more complex.

After the testing on Robot 1 resumed, it only took 15 seconds before the communications failed again. The tether had failed once more, revealing that the failure was worse than we believed – it was both terminal and non-field-repairable. This demonstrates how serious problems can be extremely subtle – a team in the field could have redeployed the robot, only to have it fail again in a short amount of time. It took several hours of lab time to repair the tether.

D. Design Issues

Out of the data recorded by the computer, two key design or manufacturing issues can be determined. The first design issue is due to an individual problem with the robot, while the second design issue appears to be a problem with the entire species of robot. These individual errors serve to highlight the variability in each robot, even within the same species. These robots are not mass-produced, therefore each has a potential for quirks and errors. Therefore, one cannot test just a single model in a line and accept all in good faith, as manufacturing errors do occur.

The first design or manufacturing issue demonstrates an individual problem with the robots. The track motors work at different levels during the entire set of data for Robot 1, while on Robot 2 the motors work at about the same level. Both motors on both robots have many long-term amperage spikes (shown in Figures 8a and 8b), showing that the track does not run smoothly – it binds relatively often. The current mean of the right track on Robot 1 is 1.0431 amps over the entire set of non-failure mode data. The current mean of the left track is 1.5828 amps, more than half an amp more for the same set of motions. This demonstrates that the left track runs harder than the right track – causing more wear on the motor, causing more thermal degradation, and making the motor fail earlier. Robot 2 does not have the same discrepancy, with the left track at 1.1754 amps and the right at 1.1252 amps. While both are slightly higher than the right track on Robot 1, they will both wear evenly over the life of the robot.

Another example of how testing can uncover individual problems with the robots is evidenced by the noise difference between Robot 1’s tilt motor and Robot 2’s. The data from the tilt motor on Robot 2 demonstrates that there is already a possible issue with that motor as the current draw from the motor is much noisier than the tilt motor on Robot 1. A noisier average current draw exposes electromechanical issues with that actuation. By comparing the two robots, one is able to determine possible future issues.

The second design or manufacturing issue is one that exists across the entire species, suggesting that there is a mechanical design flaw. There are two motors on the robot that run at a constant speed – the raise and tilt motors. The raise motor (the one that controls the height and shape of the robot) has an extremely noisy current draw in both robots (Figures 9a and 9b), showing a standard deviation of 0.9777 amps in Robot 1 and 0.7743 amps in the Robot 2. This illustrates a possible point of failure – a rough actuation can fail. Also, the high current spikes put extra stress on the motor – while the average current draw of the raise motor in Robot 1 was 1.3631 amps, it spiked up to 6.9 amps at least once during the run. Contrast this with the current draw of the tilt motor (the one that controls the angle of the camera head). It is in a well-protected system in regards to the motor motion, and it shows a standard deviation of only 0.1028 amps in Robot 1 and 0.2652 amps in Robot 2. The repeated binding and current spikes of the raise motor predicts extra wear and suggests that the design should be reevaluated. If it were not for the fact that the head of the robot is often misused as a handle, the tilt system would fail much less often than the raise system.

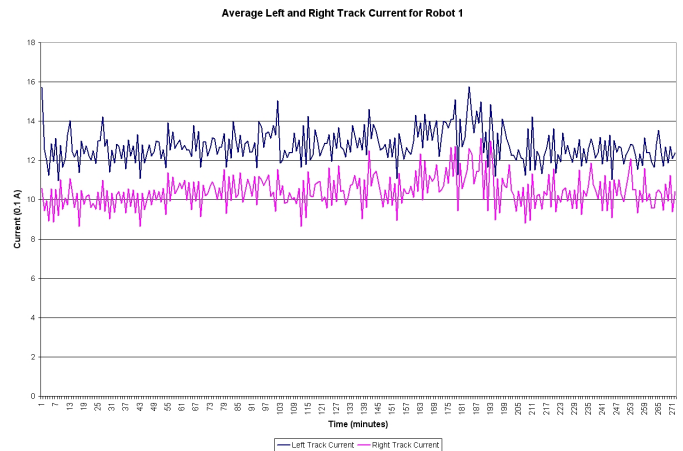


Fig. 8a. Left Track Current in Robot 1 is consistently higher, predicting earlier failure

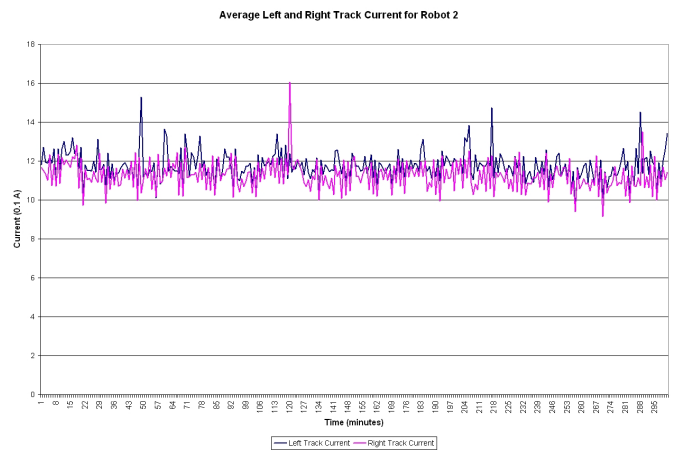


Fig. 8b. Both Left and Right Track Currents in Robot 2 are similar, predicting an even wear pattern

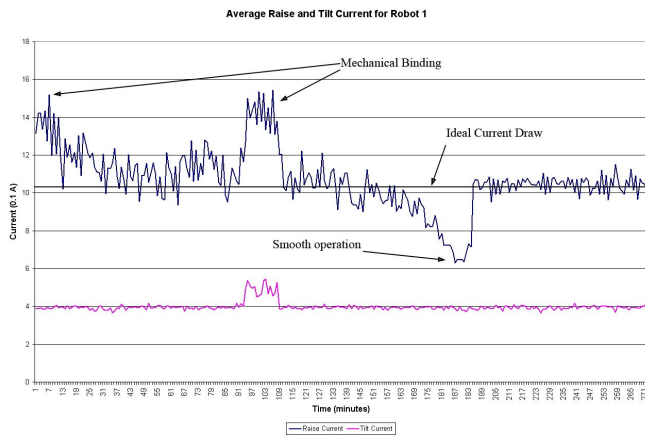


Fig. 9a. Deviations from ideal current indicate defects in the mechanical design.

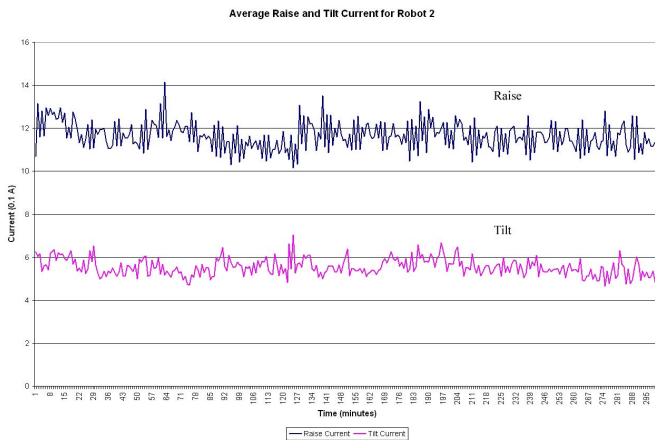


Fig. 9b. Robot 2 shows a cleaner current draw, yet it is still very noisy, indicating a model-specific, not individual robot, problem

V. DISCUSSION

The automated robot testing procedure revealed many of the same problems that plague the robot in the field, as well as some problems that are not typically seen. The observation of 8 failures in 12 hours equates to an MTBF of 1.5 hours, much less than the minimum 96 hours as established by TECO, part of the Maneuver Support Center at Ft. Leonard Wood. Also, this failure rate is much higher than that encountered in the field – nominally 6 to 20 hours MTBF [2]. The constant action of the robot versus the sporadic use during fieldwork contributes to the rapid development of failure modes- by repeatedly exercising the complete scope of actions, the robot is placed in valid configurations that may rarely occur in normal operations, thereby escaping detection. The three types of failures encountered – power (power loss), control, and communications (tether failure) – are typical of those found in the field.

The paper applied the acceptance test method to one model of robot; it is not expected that the strikingly low MTBF reflects

all robots. The results do confirm the value of acceptance testing, both as a diagnostic by the manufacturer and assurance by the customer. However, given the MTBF of 20 hours reported by Carlson and Murphy [1][2] for 7 different robot models, we believe the findings will largely generalize to other small unmanned ground vehicles. The findings from these tests suggest that for robots in general:

- The electronics of robots may have unmodeled behavior, suggesting that more manufacturer testing is necessary.
- It is possible (and strongly recommended) to predict battery loss and changes in performance.
- Even simple testing can uncover unforeseen problems. For example, even though the robot was tethered, there was still a communications failure due to complex robot-environment interaction and poor design of the tether.
- Given better sensors and control software, it is possible to predict (and possibly prevent) human error in polymorphic robots.

In addition, several observations are salient. Mobile robot manufacturers may not be using accepted industry practices when building and selling their robots. The unmodeled behavior in the control system, the tether failures, variations within individual robots as well as model wide problems suggest that more manufacturer testing is necessary. More and better testing would improve quality and catch design problems or errors in individual robots well before a robot is deployed and someone's life is in danger. The exhaustive method of acceptance testing as described above is standard in many fields – including software design and more mature industrial fields. This method of testing also provides data that can be used to show and diagnose long-term problems with a design. These failures could be predicted and perhaps prevented if manufacturers made better use of available sensors. The more proprioceptive sensors that are available, the more problems that can be diagnosed early or nascent failure states interrupted. The battery data can and should be used to predict battery loss and changes in performance. By monitoring certain sensor systems one can predict and possibly prevent human error in polymorphism.

VI. CONCLUSIONS

This paper has shown that even rudimentary endurance testing predicts how a robot will fail in the field and yields insights into the overall design. Future work is needed to further transfer concepts from quality control and systems testing to SSR robots; however, the results from this effort lead to two immediate recommendations described below.

Given the results of this investigation, it is recommended that endurance testing be adopted into the suite of user acceptance tests. It is recommended that the endurance testing procedure consist of the following:

- Testing of at least two robots selected at random from the batch. Note that given the custom-assembled, “small lots” production of many robots, this may not be sufficient to guarantee that robots built in different batches will have the same reliability. Additional research is needed to adapt statistical process control methods for sampling to accommodate the small lot phenomenon.
- An automated test sequence that exercises all functions and, if possible, all possible combinations of functions. It is important that the system be tested, rather than components. As seen in Sec.IV, an individual motor may have a high reliability but that reliability be truncated by the way it is mounted or used. A random choice of functions and ordering may be desirable because it may expose more unexpected interactions and also prevent a manufacturer from engineering a system that meets a specific test.
- Testing for the entire MTBF period. Ideally, the endurance test would be for the entire period of required reliable performance. In the case of the DoD robots, this is 96 hours without a failure, however, it does not appear that any MTBF standards have been adopted. This paper only ran the test for 6 hours due to the data collection and analysis demands, but longer testing focusing only whether the robot went for the desired period without a failure is straightforward.

In terms of diagnostic endurance testing, it is recommended that robot developers:

- Incorporate sensors into the design to provide diagnostic feedback. It is impossible to apply metrics without measurements. The addition of sensors to determine movement of parts, etc., also have the added benefit of providing more information that can be used by the control system. Additional research is needed in low-cost miniature sensors and what sensors are most informative.
- Collect internal and external video data during the endurance testing. The internal data is often insufficient to determine exactly what happened, therefore, external video is helpful.
- Analyze the data specifically to look for component failures and over design failures resulting from interactions between the components and/or the environment. More research is needed in how to analyze the data, perhaps through data mining or other AI techniques. The analysis conducted in this paper was by manual inspection and may have

missed other failures or design issues that could be captured through an exhaustive automated analysis procedure.

- Analyze the data to identify opportunities to improve performance, either through redesign or addition of software. As shown in the findings, a characteristic pattern of readings precedes a robot turning over. While the robot turning over is not a failure per se, there should be economic advantage in having a platform that can alert the user (or control software) that an undesirable event is about to occur.
- Create a “black box” data collection element that allows a window of operational data to be stored and accessed if needed after a failure in the field. This is similar to the black box recorders used in airplanes for accident investigations. It is also similar to the on-board diagnostic recorders used by the automotive and aerospace industry, which provide long-term databases of a vehicle’s performance.

ACKNOWLEDGEMENT

The work described in this paper was supported by a grant from the NSF Safety, Security, Rescue Research Center, an industry/university cooperative research center EEC-0443924.

REFERENCES

- [1] J. Carlson and R. Murphy. “How UGVs Physically Fail in the Field”, *IEEE Transactions on Robotics*, **Vol. 21, NO. 3**, 423-437, June 2005.
- [2] J. Carlson, R. Murphy, A. Nelson. “Follow-up Analysis of Mobile Robot Failures”, *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, 4987-4994, April 2004.
- [3] R. Murphy and S. Stover. “Rescue Robot Performance at 2005 La Conchita Mudslides”, submitted to “*ANS 2006: Sharing Solutions for Emergencies and Hazardous Environments*.”
- [4] R. Murphy and S. Stover. “Field studies of SSR technologies through training and response activities”. *Proceedings of SPIE. Unmanned Systems Technology VII*, volume 6230, pages 200-211, 2006.
- [5] J. Kramer and R. Murphy. “UGV Acceptance Testing”. *Proceedings of SPIE. Unmanned Systems Technology VII*, volume 6230, pages 200-211, 2006.
- [6] J.R. Cavallaro and I. D. Walker. “A Survey of NASA and Military Standards on Fault Tolerance and Reliability Applied to Robotics”. *Proceedings of the 1994 AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space*, 282–286, March 1994.
- [7] E. Messina, A. Jacoff, et. al. “Statement of Requirements for Urban Search and Rescue Robot Performance Standards – Preliminary Version”, http://www.isd.mel.nist.gov/US&R_Robot_Standards/, Accessed March 2006, Created May 2005.
- [8] M. Micire. “Analysis of the Robotic-Assisted Search and Rescue Response to the World Trade Center Disaster”, Masters Thesis, *University of South Florida*, July 2002.
- [9] “Final report for the chemical biological radiological nuclear (cbnr) sensor module with robotic platform limited objective experiment (loe)”. Final report, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 2003. From archives at www.jointrobotics.com
- [10] John G. Blitch. “Adaptive mobility for rescue robots”. In *Proceedings of SPIE. Sensors, and Command, Control, Communications, and Intelligence(C3I) Technologies for Homeland Defense and Law Enforcement II*, volume 5071, pages 315–321, 2003.
- [11] G. Boxley. “Teleoperational d-7 dozer concept evaluation program executive summary”. Executive Summary, US Army, Test and Evaluation

Coordination Office (TECO), Ft. Leonard Wood, 1999. From archives at www.jointrobotics.com.

[12] F.W. Cook. "Test report for the panther ii". Test report, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 1997. From archives at www.jointrobotics.com

[13] F.W. Cook. "Limited objective experimentation report for the deployable universal combat Earthmover (deuce)". Executive summary, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 2000. From archives at www.jointrobotics.com

[14] G. Piskulic. "Military police/ engineer urban robot (urbot) concept experimentation program (cep) report, engineer module". Concept experimentation program (cep) report, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 2000. From archives at www.jointrobotics.com

[15] G. Piskulic. "Military police cep for small robots in support of dismounted troops utilizing non-lethal weapons". Concept experimentation program (cep) report, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 2001. From archives at www.jointrobotics.com

[16] R.R. Walker and M.R. Scarlett. "Test report for the all-purpose remote transport system (arts) executive summary". Executive summary, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 1999. From archives at www.jointrobotics.com

[17] R. Weiss. "Final test report for the unmanned ground vehicle rapid obscurant platform (ugvrop) chemical corps concept exploration program (ugvrop -cep)". Concept experimentation program (cep) report, US Army, Test and Evaluation Coordination Office (TECO), Ft. Leonard Wood, 2001. From archives at www.jointrobotics.com

A Complete Simulation Environment for Measuring and Assessing Human-Robot Team Performance

A. Freedy, E. Freedy, J. E. DeVisser, G. Weltman
Perceptronics Solutions, Inc.
Sherman Oaks, California

M. Kalphat , D. Palmer, N. Coyeman
U.S. Army RDECOM-STTC
Orlando, Florida

ABSTRACT

This paper describes the in progress-development of a new simulation environment for performing experiments that will measure and assess the performance of mixed human-robot teams in a variety of military and non-military situations. The initial environment combines several novel components: (1) a Mixed Initiative Team Performance Assessment System (MITPAS); (2) a simulated robotic command and control system based on the U.S. Army's OneSAF simulation; and (3) a event-based test scenario characteristic of anticipated military operations. MITPAS consists of a methodology, tools and procedures to measure the performance of mixed manned and unmanned teams in both training and real world operational environments. It is directed toward supporting the operational and training needs of future military forces that will use mixed manned and unmanned forces for a broad variety of functions. Measurement of overall effectiveness in these mixed initiative systems will be essential in order to achieve optimal system performance levels. OneSAF is a generalized set of models and tools designed to supply semi-autonomous entities to a variety of simulations. Our event-based scenario reflects mid-term plans for DOD's Future Combat System (FCS) as well as current uses of robotic systems in combat. The purpose of the new simulation environment is to provide a self-contained, easy to use test bed for obtain meaningful measures of both human and unmanned system performance to be used to test proposed mixed initiative configurations and to identify new training requirements.

KEYWORDS: *mixed initiative teams, human-robot performance assessment, robotic training systems*

1. INTRODUCTION

Mixed initiative introduces a new and unique aspect to the psychology of team performance: the interaction of two cognitive systems -- human and autonomous unmanned robot [10]. In addition to the critical performance factors associated with human teams -- which include information exchange,

communication, supporting behavior and team leadership -- the mixed manned/unmanned team adds a number of challenging new dimensions. Foremost among these is the ability of the human team to manage, predict, collaborate and develop trust with unmanned systems that may sometimes exhibit fuzzy responses in unstructured and unpredictable environments [1] [2] [3] [4] [5] [6][7][11].

The major challenge in our work has been to develop system-specific measures of behavior on which to base assessment of the mixed initiative team performance. As reported by Freedy et al [8][9] a performance model for mixed initiative tasks was developed in the first Phase of the work. A brief summary of the performance model will be given here. The performance model represents a critical challenge since the measures must be unique to the information and decision environment associated with human-robotic teams and to directly link together behavioral processes important to mixed manned/unmanned tactical outcomes. The measures need to provide feedback for skill improvement in collaboration as well as adaptation to stress and workload, and they should help define the training needs themselves.

The Performance Model draws on five separate research areas that have been pursued independently in the past but which are being integrated in this project to establish meaningful criteria of overall performance. These research areas are:

- **Psychology of Team Performance** - Human team performance measurement in C3 information environments, performance variables, training evaluation and measuring team related expertise, management of workload and stress.
- **Unmanned Systems** - Principles of establishing performance metrics for autonomous systems
- **Mixed Initiative Systems** – Research and findings on the critical variables which affect human decision and control of autonomous systems
- **War Fighting Behavior** – Observations and measurements of combat team performance in war fighting tasks C3 tasks
- **Human/Robot Team Processes** - These processes represent the dimensions of the human interaction with the robotic elements

We have integrated and adapted theories and concepts in these areas to processes associated with manned/unmanned team performance and training. Most critical were variables related to the decision making behavior of the unmanned systems, such as behavior transparency to the human collaborators, human trust in robot decisions and human abilities to synergize the autonomy of robots so as to add to the capability of the total team. Issues such as behavior prediction, level of autonomy and acceptance of robots actions have also been examined and identified for possible high impact variable on total system performance.

2. SYSTEM PERFORMANCE MODEL

In accord with this approach, we have created a preliminary System Performance Model which captures the critical performance attributes of the distinct process of behavior composition environment. Our objective was to identify the dimensions of performance which contribute to effective outcomes of collaborative manned-unmanned tasks and, in particular, to formulate measures to evaluate training in processes that are unique to the collective team of humans and robots. Accordingly, we have built a taxonomy of specific processes which can be decomposed into explicit behavioral objectives side-by-side with measures of effectiveness based on actual outcomes. Our focus is on process measures that are closely linked to outcomes, because it is these measures that will provide the feedback necessary for training. The three levels of team processes critical to training evaluation and remediation are: (1) individual human; (2) team human; and (3) collective human/robot team.

We decomposed the processes into these three levels and developed taxonomy of measures for each level. We narrowed the performance measures to the simplest factor structure that adequately cover the dimension of teamwork as was found in previous investigators [2]. The actual Performance Model will consist of a multi-dimensional task process performance schema which will (1) aggregate the performance measures at each level, (2) provide for training feedback at each level, and (3) provide a multi-attribute discriminate function to determine an overall level of proficiency as well as a "pass-fail" score. The weights of the attributes will be established in simulations in which the linkage between specific task performance measures and outcomes can be estimated. There are two main types of measures: Measures of Performance (MOP) and

Measures of Effectiveness (MOE); these are defined separately below.

Measures of Performance (MOP). These are observable and derived measures of the operators' task skills, strategies, steps or procedures used to accomplish the task. They consist of the cognitive and interactive processes of the individual and team in collaborating together and controlling the robotic entities in a coordinate manner. MOP evaluates the human factor involved in a complex system. MOP was divided into 3 distinct classes of processes dimensions:

- **Human Team Processes** - These processes represent the dimensions of the human team interaction
- **UV Management and Control Processes** - These processes represent the tasks associated with real time control and monitoring of the autonomous entities
- **Human/Robot Team Processes** - These processes represent the dimensions of the human interaction with the robotic elements

Measures of Effectiveness (MOE). These measure the "goodness" of the composed behavior in quality and the execution of war-fighting tasks. MOEs are influenced by much more than human performance. These measures also contain variance accounted for by system design, the surrounding environment and luck [6]. The measure consists of the following dimensions

- **Mission Effectiveness** - Observable measures of the success of the mission as determined by objective military criteria.
- **Behavioral Effectiveness** - Measures of the dimension of behavioral effectiveness of the system in the battlefield

In our planned experimental studies we plan to reduce the measures to a manageable subset, as described below.

3. SIMULATED COMMAND AND CONTROL SYSTEM

We have adapted the OneSAF OTB simulation environment to represent a typical command and control system for a mixed human robot team. OneSAF-OTB is an extensive, DOD-developed simulation system that allows users to select from a large variety of previously-modeled vehicles and other battlefield entities and exercise them semi-autonomously in a variety of modeled terrains. Prior to a OneSAF exercise, the controller will specify the tactical behaviors of the various simulated entities by means of highly detailed graphical interface; during the course of the exercise, the entities will behave in accordance with the previously specified rules. We

have essentially converted this highly capable and flexible capability into a robotic command and control system by adapting it to the likely unit of action for the anticipated use of robotic forces in the military and severely restricting the available control inputs to make the user interfaces more amenable to real-time operation.

With respect to the military unit of action, our problem was that human-robot teams as envisioned in DOD's Future Combat System (FCS) do not yet exist; accordingly, tactical doctrine for human-robotic teams, which is normally derived from doctrinal and training publications, also does not exist. Our solution was to select a representative FCS element with a mix of human and robotic performers, select a tactically relevant scenario and then develop auditable threads of tactical documentation to create representative, surrogate standards of performance for scenario tasks. The representative unit of action we selected was a typical reconnaissance platoon, headed by a Platoon Leader, in which a Platoon Sergeant would be Responsible for controlling a unmanned ground vehicle

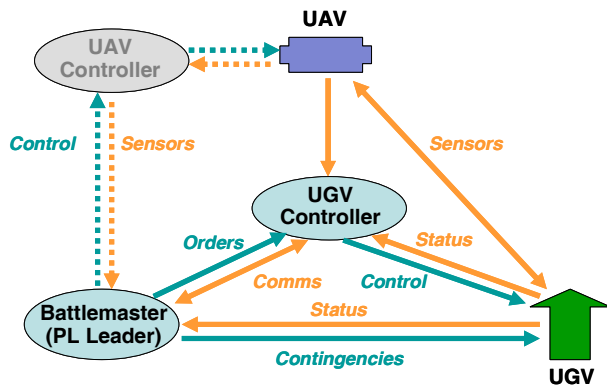


Figure 1 Command and Control Configuration

For our initial simulation environment and experimental studies of mixed initiative team performance, the command and control configuration is as shown in Figure 1. The experimenter, or Battlemaster, is primarily in charge of the experimental procedures and progress of the scenario, and also plays the Platoon Leader (PL) role. As the PL, he or she issues orders to and communicates with the Unmanned Ground Vehicle Controller (UGVC), who is the actual experimental subject.

The Battlemaster also controls the actions of an Unmanned Air Vehicle (UAV) through a surrogate UAV controller. Finally, the Battlemaster is able to monitor directly the status of the UGV being controlled by the experimental subject and to

introduce contingencies, such as malfunctions, during the course of a scenario exercise.

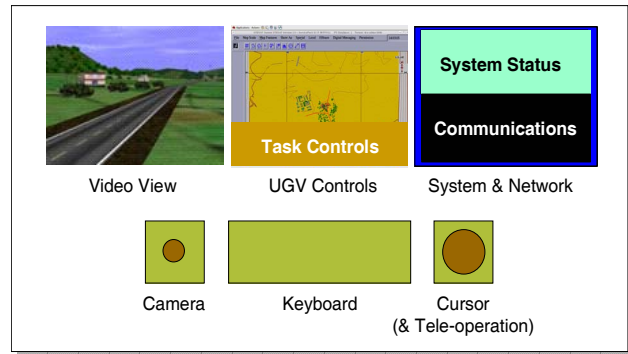


Figure 2 UGV Controller Station

The command and control stations used by the Battlemaster and the UGV controller are very similar. Figure 2 shows the UGV Controller station configuration. The station consists of three 19", 1280 x 1024 displays: the central display provides a continuous map view of the scenario area with OneSAF icons and the OneSAF control and system feedback mechanisms for the robotic vehicle; the right-hand display contains UGV system status information and text communications capabilities, and the left-hand display provides an optional 3D, real-time view from the vehicle for direct tele-operation of the UGV (termed 'stealth' operation in the OneSAF environment). The Battlemaster and UGV Controller stations also contain "instant messaging" capabilities for simulated C2 textual communication between the platoon leader and the UGVC. This is in conformance with modern C4I practice, in which text messages are preferred over voice communications, which are often confusing and costly in terms of bandwidth. Lastly, the set-up has the capability of introducing a secondary task to the UGV Controller and recording his or her responses on the exercise time line.

Figure 3 shows the central control screen with the map of the initial scenario area and an example of our OneSAF control interface. OneSAF allows for a large variety of task actions to be assigned to the simulated entities; these are typically defined in advance of the exercise itself. In order to make real-time operation of this interface feasible, we have defined a condensed set of nine mixed-initiative control actions available to the UGV Controller through the OneSAF task frame selection menu. These actions, shown as Move, Road Move, Halt, Occupy Position, etc., are sufficient for performance of the scenario and adequately represent the anticipated capabilities of unmanned system in the middle Level of Autonomy (LOA) range.

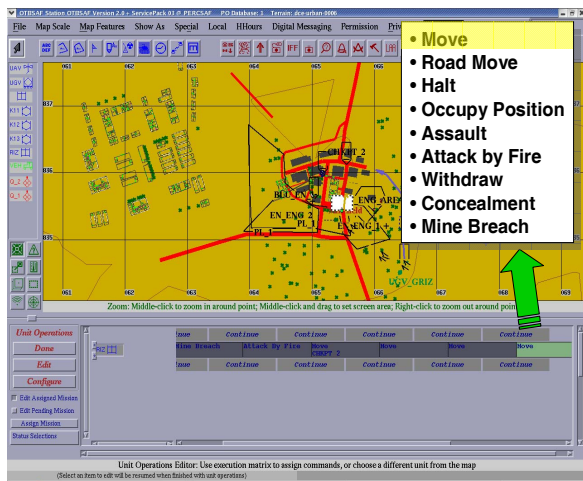


Figure 3 Control Interface

For our OneSAF simulation of a future UGV, we have initially selected the 'GRIZZLY' combat engineering vehicle, augmented with some additional tactical capabilities for our special purpose. The GRIZZLY is an armored vehicle designed to breach complex obstacles including mines, berms, wire, rubble, and tank-ditches. A modified Abrams M1 hull, it employs armors and many components derived from various members of the Abrams family of tanks. The GRIZZLY's obstacle clearing features include a full-width mine-clearing blade and a powered, extensible excavating arm. It mounts a 50 cal remote fired machine gun for self defense. In actuality, a crew of two operates the system. In the OneSAF version, it is capable of autonomous, intelligent operation and exhibits many of the characteristics of such systems. As such, this vehicle is sufficiently representative of capabilities anticipated in robotic ground vehicles of the FCS. While it may not physically resemble such future vehicles, this will not be apparent to our experimental subjects.

4. MIXED INITIATIVE SCENARIO

Development of a manned-unmanned scenario and selection of the terrain on which it would be played proceeded from analysis of our unit of action and the likely missions of future mixed initiative teams. Our development process is illustrated in Figure 5. We began with three sets of assumptions taken from available DOD forces: a Caspian Sea tactical vignette, an assessment of Future Combat Force Systems Capabilities and the Organization and Equipment anticipated for our unit of action. From this foundation we developed a generalized scenario and conducted an analysis of the activities expected from the various team members, which we would use

in our assessment of team performance. Finally, we matched scenario characteristics against available simulation data bases and selected for our initial locale the McKenna MOUT (Military Operations in Urban Terrain) data base, which has the necessary features of roads, forests and a small village or built-up area

The scenario area is shown in Figure 4 along with the locations corresponding to the currently planned experimental events. The military element is a Reconnaissance Platoon consisting of three M2 Bradley vehicles (V1, V2 & V3) shown at the lower left, an UGV shown at right flank and a UAV (not shown). The Platoon is in advance of an important supply convoy that is scheduled to come through the surveyed area within a set time. The mission of the platoon is to insure that the route is currently safe for passage and clear it if it is not.

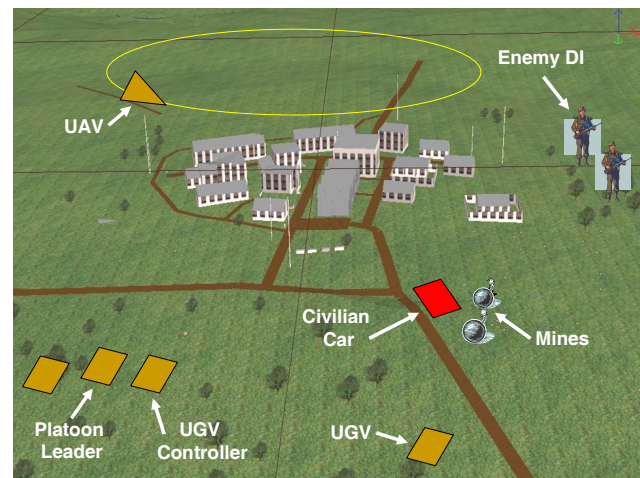


Figure 4 Initial Mixed Initiative Scenario Terrain

The Platoon Leader (PL) rides in V1 and issues commands. The UGV Controller (UGVC) rides in V2 and is responsible for the performance of the Unmanned Ground Vehicle (represented by a OneSAF Grizzly Combat Engineering Vehicle, as described above).

The planned pilot studies will require the subject UGV to perform a combination of several distinctive Tactical Events, typical of anticipated FCS mixed-initiative actions, which can then be examined with the MITPAS measures. The Tactical Events include:

- **Preparation.** The UGVC receives an order to move the UGV to a proscribed waypoint in a clump of trees east of an intersection. The UGVC creates an appropriate movement instruction and puts the UGV in motion.
- **Respond to Suspicious Activity.** The UGVC receives a message that the UAV has detected mines associated with a civilian vehicle parked at a road intersection south of the built-up area. The UGVC is ordered to proceed to the indicated point, determine if

mines are present, and destroy them if they are. The UGVC is warned that mined vehicles are typically guarded by a hidden squad of dismounted enemy.

- **Destroy Mine Threat.** The UGVC detects the mines and uses the anti-mine capabilities of the UGV to enter the mined area and destroy the mines.
- **React to Attack by Fire.** Following mine clearance, the UGVC is ordered to move the UGV north to a checkpoint further into the built-up area. As anticipated, the mined vehicle is in fact being watched over by a squad of enemy located to the east. They begin to fire on the UGV. The UAV helps pinpoint their location, and the UGVC engages them using the UGV's 50 cal remote fired machine gun.
- **Respond to UGV Malfunction.** The UGVC is then ordered to continue the UGV's movement further northward into the built-up area. While on the move the UGV encounters a typical robotic control problem in navigating around a building, causing it to perform repetitive or non-productive movements. The UGVC is required to intervene using Stealth or other means to return the UGV to a battle-ready condition.
- **Assessment.** The Platoon arrives at RP and performs assessment of operation. UGVC prepares an assessment report using a MITPAS form which we use as part of total measurement.

As we become more familiar with operation of this scenario, it will be possible to extend it by the addition of other Tactical Events, possibly including coordinated actions between the UGVC subject and another UGVC subject, thus extending the size of the experimental mixed-initiative team.

5. EXPERIMENTAL DESIGN

We are currently in the process of designing our initial experiments and conducting a series of pilot runs to test and refine the complete simulation environment. We are concerned with several factors: (1) experimental subjects, and the method we will use to train them; (2) independent variables, that will establish our experimental conditions; and (3) dependent variables, which will be derived from our MITPAS measures and will provide the basis of assessment; (4) experimental plan, the sequence of studies; and (5) analysis, how the various possible measures will lead to a useful assessment instrument.

Subjects. We are currently using project personnel for the pilot runs to test the system and experimental procedures and to select our final set of pilot

experimental values. We will then use outside subjects for the main pilot studies. It appears that considering the skills involved, male and female subjects with good gaming experience will be the most viable and the most representative of the future military UV operator population. Gamers are widely available in the Washington DC high school, junior college and university communities. It will be necessary to instruct the subjects in the objectives of the exercise and to train them in the use of the OneSAF interface and the other UGV control and communications capabilities. Based on current experience with the reduced task set, we believe this can be accomplished in about 2 hours.

Independent Variables. We have determined a candidate set of conditions under which the mixed initiative task can be performed in order to exercise the MITPAS measures and determine the information they will provide in the anticipated real-world, mixed-initiative environment. Selection was based on analysis of (1) the UV literature, (2) our proposed approach, (3) team member experience and (4) the current UV situation. The candidate set is described below; our plan is to investigate these variables during the preliminary pilot runs and to choose several of the more promising for the actual pilot experiment.

- **Control Emphasis.** The UGVC is instructed either to maneuver the UGV as fast as possible, commensurate with as few errors as feasible or to maneuver the UGV as precisely as possible with a timely response.
- **Time Stress.** In the Stress condition, mission execution is given a set time limit, and the UGVC is continually reminded as time is running out; in the No Stress condition, there is no mission deadline.
- **Mental Loading.** In the High condition, the UGVC is given a secondary task to perform that is logical within the scenario but unrelated to the main task; in the Low condition, there is no secondary task. Secondary tasks can be simple (such as responding to a message alert with a button push) or more complex.
- **Direct Control Penalty.** In the Penalty condition, the UGVC is told that taking direct control of the UGV (the 'stealth' mode) carries with it a tactical penalty. In the No Penalty condition, there is no such instruction. The justification for the penalty is that direct control consumes valuable bandwidth that is in short supply on the net-centric battlefield.
- **Control Action Set.** In the Small Set condition, the UGVC is given a highly condensed set of OneSAF task frames with which to control the UGV's actions, in the Large Set case, this is expanded to include more frames.
- **UGV Reliability.** In the High Reliability case, the UGV encounters relatively few operational problems,

in the Low Reliability Case, the number of problems is significantly higher.

- UGV Type. The Grizzly condition uses the presently selected OneSAF Grizzly model for the UGV representation, the Other condition uses another model selected from the available OneSAF inventory or one made up of specially assembled components, as feasible

Dependent Variables. We intend to collect data on: (1) objective measures of performance in the S&R Scenario and, (2) mixed-initiative team performance measures. The candidate *Objective Measures* are:

- Execution Efficiency. The relative proportion of time that the robot is executing and operator instruction as opposed to the time it is waiting for direction as a function of total mission time.
- Navigational Efficiency. The distance traveled by the robot from the start to the end of the mission as compared with the point-to-point distances summed along the pre-planned route.
- Average Speed. The average speed made by the robot along its path during the mission.
- Threat Reaction Time. The aggregate time required by the man-robot team to react to a threat after line-of-sight (or equivalent) is achieved.

The candidate *Mixed Initiative Team Performance Measures* are:

- Control Allocation. Optimal allocation of tasks between the Human and Autonomous UV function. “Who should do the task and when?” Does the operator take control where appropriate and let automation handle it when appropriate
- Human Robot Trust. The level of trust at which the human delegates control to the UVs. Is he over trusting or under trusting? Does he interrupt the UV unnecessarily due to lack of trust?
- Monitoring Feedback. Observation and keeping track of UV’s performance, including UV’s mistakes. Recognizing good and bad performance by UVs; recognizing UV system conditions and operational status.
- Task to Team Ratio. Ratio of the time allocated to the time allocated to war-fighting tasks to the management of team issues.
- Human Robot Coordination. How well team members work **synchronously** and **jointly with cooperation** and timely information **interchange** with others in managing the UVs.

- Mixed Initiative Efficiency. How the team modulates UV’s initiatives with human ones. Are they coordinated and smoothly integral?

There is also another set of ‘*internal*’ or ‘*system*’ measures associated with operation of the OneSAF interface itself, such as speed of UGV task assignment, ability to pre-program successive UGV tasks, ability to fully monitor system status, ability to use all the capabilities provided, station configuration variables (displays and controls), individual differences in robotic control capabilities, etc. While these measures are not central to the present mixed-initiative research, they are of some interest because they are likely to be representative of the factors that will arise in real-world UV control situations. We will try to record findings on an informal basis as a contribution to this body of knowledge.

Experimental Plan. Our experimental plan is structured in four parts as presented in the table below. Following is a preliminary description of each phase; the detailed test design will be produced in the coming months as we work through the initial phases.

- **Laboratory System Pilot Runs.** In the first phase, which is now under way, the test environment will be set up and validated.
- **Model Validation and Tuning.** The second phase will be devoted to test execution to collect data across the spectrum of operations in the scenario, expert observation and evaluation, and reduction of the measurement set.
- **Battle Operations in Simulation.** Phase three will validate the reduced measure set by applying it to a more complex set of scenario activities representative of FCS battlespace operations.
- **Field Operation with Live UVs.** As an option, we have proposed in a fourth phase to demonstrate the operation of the performance measurement system in a live environment using instrumented UVs operating on a tactical range.

Analysis. Our proposed analytical approach is based primarily on a mapping of measures into scenario events, in order to determine the effectiveness of the various measures in characterizing mixed initiative performance under typical mixed-initiative conditions. The table below shows an example of the process, in which measures are mapped into a set of events based on an earlier breakdown of a proposed scenario. The purpose here is to demonstrate the planned methodological approach, rather than provide the final listing that will be part of the full experimental design.

In addition, we are exploring the use of expert judgment to determine which set of MITPAS measures are most efficacious in representing mixed-initiative team performance. In this regard, we have developed a schema

employing a factor analytic approach to reducing and refining the set of measures to reflect underlying orthogonal performance dimensions: The scenarios, candidate measures and algorithms, and the OneSAF OTB virtual testbed provide a framework for a multi-stage data collection effort within which subjects with representative background, experience, training, and skill levels are asked to execute FCS missions as part of a human-robot team, as described above. Members of the combat development community, or equivalent groups, are asked to observe the trials (or recorded versions) and to provide subjective evaluations of the execution of the human-robot team in light of the experimental conditions, or independent variables. Accepting their expert judgment to be the reference standard for performance evaluation, a factor analysis process is employed to examine the value of the component and composite linear factor combinations of the candidate measures in accounting for observed performance. The intent is to identify a reduced set of orthogonal underlying composite measures to which a practically substantial proportion of the measure variance (in relation to expert subject judgment) can be allocated. As the experimental plan progresses, we will determine whether formal factor analysis or analyses can be used as part of this schema, or whether we need to employ less rigorous methods. In any case, the general approach of using expert judgment to validate the critical set of measures will form an important part of our analysis.

6. PILOT EXPERIMENTS

We have conducted a series of pilot experiments to test and refine the complete simulation environment and observe the behavior of the performance measures. Subjects received about two hours of training on proper usage of the OneSAF system by completing exercises in a training manual. The manual consists of a detailed account on UGV operation, UGV autonomous behavior, as well as a complete mission briefing. The experimenter provided any supplementary instruction as needed. Subjects who failed the final training test received additional training until they satisfactorily passed the final test. Following the training session, subjects participated in four trials, operating in the scenario described above. After each trial, the subject filled out a NASA TLX form.

The pilot experiments have fulfilled their objectives. The results have demonstrated successful end-to-end operation of the complete MITPAS system, including the simulation, the data acquisition and the measurement components. Accordingly, we have

judged the total simulation environment to be satisfactory for exercising the initial set of mixed initiative measurements. Also, we have found that our operator training procedures adequate for the initial trials, for both game-savvy and non-gamer subjects.

With regard to the subjects' in-task experiences, we found that the initial subjects:

- Understood their robotic mission very well;
- Developed a preference for assigning tasks to the UGV in a particular manner;
- Were careful about overriding the UGV with full manual control, and usually kept trying to accomplish the task with sequences of simple control actions;
- Became frustrated when the UGV would not respond well to commands; a typical real life occurrence.

The initial set of mixed initiative measures also behaved satisfactorily, showing consistency from run to run and providing some interesting preliminary insights into the mixed initiative team behavior. For example, we found that execution efficiency, defined as the ratio of execution time to total mission time, increase steadily with experience, as might be expected. Less intuitively, we found that execution efficiency was higher for both low and high amount of message traffic between the robot controller and his commander, and lower for intermediate traffic level. We are currently planning an initial set of studies focusing on the trust variable, which we believe to be a central and relatively poorly understood factor in mixed initiative team performance.

7. CONCLUSIONS

Colonel Jimmy Shiflett (USA, Ret), who now heads the FCS training development program at SAIC, Orlando, and who was formerly the person most responsible for US Army fielding of the ground breaking SIMNET virtually simulation system for combined arms training, has said, "Development of full procedures and training systems for human and robot teams is uncharted and unprecedented" [12]. Yet we know this is the wave of the future, and it is thus incumbent on us to establish precedent and begin to chart the territory. That is the purpose of the mixed initiative simulation environment described herein.

Our primary objective in the SBIR program is to provide a turnkey MITPAS software system running in the OneSAF-OTB environment, with automated data collection functions and containing protocols for evaluation of various manned/unmanned team configurations in selected event-based scenarios. Phase I validated the concept, the present Phase II effort will fully implement it.

Establishing the proper performance indicators for training and operations is a multidisciplinary as well as

multidimensional effort and will have benefits to training research as well as to operations. Research to date has been performed independently in five areas related to the present problem: human team performance; human-robot interaction; mixed initiative systems; metrics for autonomous UVs; and warfighting simulation and evaluation. We are applying and building on the existing knowledge and research findings in each area in order to synthesize a scientifically-sound performance measurement system. Specific resultant benefits will include: (1) critical team performance dimensions that can be generalized to a wide range of mixed initiative training; (2) criteria for assessing collective performance in manned/unmanned warfighting units; (3) performance measures for human supervision of autonomous decision making entities under stress; (4) OneSAF-OTB compatible software infrastructure and protocols for simulation and evaluation; and (5) embedded Training interoperability for ready transition to the Future Combat System (FCS).

8. REFERENCES

1. Albus, J.S., National "Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles" Per MIS Proceedings -2002
2. Cannon-Bower, J.A., and Salas, E., "Individual and Team Decision Making Under Stress: Theoretical Underpinning" in "Making Decisions Under Stress" Edited by J.A. Cannon-Bowers and E. Salas, Am Psych Assoc, WDC, 1998
3. Clough, B.T., "Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway?", Air Force Research Lab Wright Patterson. AFB Per MIS Proceedings -2002
4. De Visser, E., Freedy, A., Freedy, E., Weltman, G., & Parasuraman, R. A Comprehensive Methodology for Assessing Human-Robot Team Performance for Use in Training and Simulation Conference: In Proceedings of the 50th Human Factors and Ergonomics Society. In Press Meeting, San Francisco, CA. October 16-20. 2006
5. Drewes, P. and Franceshini, R. "Demonstration of a Systems Architecture for Live, Virtual, and Constructive Interoperation, SAIC Report, Orlando, 2003
6. Drewes, P., "Lessons Learned in Group Robotic Command and Control," Unmanned Systems Conference, Orlando, June 2002

7. Fong, T., Kaber, D., Lewis, M., Scholtz, J. Shultz, A., and Steinfeld, A. *Common Metrics for Human-Robot Interaction* IEEE International Conference on Intelligent Robots and Systems, Sendai, Japan 2004
8. Fong, T., Thorpe, C. and Baur, C., "Collaboration, Dialogue and Human Robot Interaction." Proc. 10th International Symposium on Robotic Research, Springer Verlag 2002
9. Freedy, A., McDonough, J.G., Freedy, E.T., Thayer, S.M., and Weltman, G., *Command Language for Composable War Fighting Behaviors of Autonomous Unmanned Vehicles in the Future Battle Space*, SBIR Phase I Final Technical Report, Perceptronics Solutions Contract No. DAAH01-03-C-R241, February 2004
10. Freedy, A., McDonough, J.G., Freedy, E.T., Jacobs, R., Thayer, S.M., and Weltman, G., *A Mixed Initiative Team Performance Assessment System (MITPAS) for Use in Training and Operational Environments*, SBIR Phase I Final Report, Perceptronics Solutions Contract No. N61339-04-C-0020, May 2004
11. Goldberg, S., *Informal Discussion*, Los Angeles, June, 2003
12. Kalphat, H. M., and Stahl, J., "STRICOM's Advanced Robotics Simulation STO: The Army Solution to Robotics M&S," Proceedings of the Eleventh Conference on Computer Generated Forces & Behavioral Representation, Orlando, FL, May 2002.
13. Shiflett, J., *Informal Discussion*, Orlando, August, 2005

9. ACKNOWLEDGEMENTS

1. The work reported here was funded under Contract No. N61339-04-0020 administered by US Army RDECOM-STTC, Orlando, FL
2. Michelle Kalphat serves as Chief Engineer, Intelligent Behaviors for Autonomous Systems & Simulation Technologies, US Army RDECOM-STTC, Orlando, FL
3. Donnie Palmer serves as Computer Scientist, Intelligent Behaviors for Autonomous Systems & Simulation Technologies, US Army RDECOM-STTC, Orlando, FL

Development of an Evaluation Method for Acceptable Usability

Brian Stanton
National Institute of Standards
and Technology
100 Bureau Drive
Gaithersburg, MD, USA
brian.stanton@nist.gov

Brian Antonishek
National Institute of Standards
and Technology
100 Bureau Drive
Gaithersburg, MD, USA
brian.antonishek@nist.gov

Jean Scholtzⁱ
Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352
jean.scholtz@pnl.gov

Abstract—The National Institute of Standards and Technology (NIST) conducted three workshops with First Responders to determine requirements for robots used in Urban Search and Rescue (USAR). These requirements [1] were further prioritized by the responders. NIST has now undertaken the task of developing evaluation methods and metrics for the requirements deemed to be the highest priority. Of these high priority requirements, a number of these addressed the human-system interface and interaction. In this paper, we explain the pilot testing that has led us to the current evaluation design and outline the next steps.

I. INTRODUCTION

Accepted measures of usability [2] are effectiveness, efficiency, and user satisfaction. Effectiveness is often defined as the percentage of users that can carry out a particular task. Efficiency is the time it takes users to complete the task. User satisfaction refers to users' ratings on one of a number of standard satisfaction questionnaires.

Acceptable usability was defined as one of the requirements for the human-system interaction in USAR robots. To measure this, it is first necessary to define the tasks that users need to carry out. Ground robots were selected as the focus for defining the first set of evaluations. Once we have defined evaluation methods for ground robots, we will need to define similar evaluations for aerial vehicles, underwater vehicles, and wall climbing vehicles.

II. DEFINITION OF TASKS

We selected navigation and camera manipulation as the first two essential tasks that an operator must be able to do using the human-system interface. These tasks are independent of the degree of autonomy that a particular robot has. We are only concerned with testing the interface – not the capabilities of the robot. Our evaluation of the user interface is but one piece of a package of evaluation methods, many of which are measuring the capabilities of the robot. The acceptable usability evaluation is not meant as a comparison between robots but as a comparison of how well a novice user does as compared to an expert user.

III. Acceptable Usability Test

A. Version One of the Acceptable Usability Test

The first version of the test was designed as a slalom course. This is shown in Figure 1. We set up a number of gates, some were wide and some were narrow gates. In addition, the gates differed in the distance they were apart and the offset of the gates. Our thought was that the gate width, distance between gates, and offset of the gates would differ depending on the size and turning radius of the robots.

We added camera manipulation to the evaluation by posting numbers and letters on the various gates and asking the operator to read as many as possible to us before going through the gate.

The measures we collected were: the time it took to navigate through the gates; the number of gates cleanly navigated through; and the number of letters and numbers the operator viewed. The latter was adjusted based on the capabilities of the robot. For example if a robot did not have a pan tilt camera and was low to the floor, then the operator was not expected to be able to see a letter posted on the very top of a cone.

This evaluation would be conducted after novices had completed the vendor suggested training. We piloted this test at a USAR workshop with both novices and experts in August, 2005. We asked both novices and experts to try out this first test. It should be noted that operators were not able to view the course while running the robots through. While we did not have enough users and robots present for any statistical data collection, we did discover that the novices could complete the course in about twice the time it took the experts to do this.



Figure 1. Version one of the acceptable usability test

There were several problems that we noted with the evaluation. One issue was the base on the cones. While the robot operators were able to navigate successfully through the upright portion of the cones, they often did not see the base of the cone and ran over it or moved the cone as they pushed up against the base. In addition it was difficult to place letters and numbers on different portions of the cones for the camera manipulation portion. Moreover, the operators could not read some of the numbers and letters if they were far enough away from the cone. So deciding on the “correct” number of letters and numbers depending on a robot’s cameras was difficult.

B. Version Two of the Acceptable Usability Test

In March 2006 we tried out a second design for the acceptable usability test. In place of the cones we previously used, we decided to use cardboard boxes to make rows. We did this to give us more flexibility in placing markings to use in the camera manipulation exercise and to prevent operators from being able to view the entire course at one time. We marked gates using red and white hazard tape and told the operators that they were only to go through the marked gates. We tried to place tape at various locations on the boxes so that all robots could see this using their cameras. Figure 2 shows the setup for this version of the test. The cones shown in the photo were used to designate the end of a row as we had to set this up in a large hotel space.

Again, we measured the time it took to navigate the course, the number of correct gates that the operator went through and whether the operator was able to traverse through the gate

cleanly.

We did not have enough robots or users for any statistical analysis. One problem with this setup was that it was easy for the robots to move the boxes. We needed to devise a way to easily replace the boxes in the correct position.



Figure 2. Version two of the acceptable usability test

C. Version Three of the Acceptable Usability Test

We had another opportunity to try out a version of the acceptable usability test during a responder workshop at Disaster City, Texas in April, 2006. This time, we decided to add the notion of situation awareness to the evaluation. We devised some gates that the robots would not fit through. We used cardboard boxes to make rows. In each row we had two gates. We marked the gate that the robots were supposed to go through using red and white hazard tape. In some instances these gates were too narrow for that robot. We asked the operator to determine this and if this was the case to traverse to the other gate and go through it.

As in the first version of the test, we measured the time it took to traverse the course (compared to the expert time), the number of gates that were cleanly navigated and the number of gates that the operator recognized as marked.

We had many more robots this time. In fact, due to the logistics of the workshop, we were unable to customize the gate width and the distance between rows to every robot. We decided to design the course for categories of robots, rather than using a percentage of the robot size as the criteria for the size and placement of the gates. Based on the robots taking part in this evaluation, we used two sizes of gates. For robots wider than 19 inches (48.3 cm), we used gates of 2 feet (61 cm) and narrow gates of 18 (45.7 cm) inches. For robots under 19 (48.3 cm) inches, we used gates of 20

inches (50.8 cm) and 14 inches (35.6 cm). In both configurations the rows were 48 inches (121.9 cm) apart.



Figure 3. Rows of boxes for the third version of the acceptable usability test



Figure 4. Markings on gates for third version of acceptable usability test

Figures 3 and 4 show the setup for the third version of the acceptable usability test. We conducted the test in low light. Although we were unable to measure the actual light in this instance, we plan to propose three lighting conditions for the final version of the acceptable usability test: low light with a range specified; complete darkness; and direct sunlight. For the direct sunlight case, it may only be necessary for the operator to be positioned in the sun and for the actual navigation maze to be in the low light condition.

Figure 4 shows how we marked the gates for the robots. We varied the markings so that they were low, medium, and high on the boxes. This gave all robots a good chance of seeing the markings.

Figure 5 shows the actual configuration we used. We did not have enough room for all eight rows of the maze so we made four rows and had the robots traverse it twice; once from the start and then returning through the maze. The second time through we marked different gates.

To ensure that we could easily reposition the boxes should they be moved by a robot, we taped the boxes together and we marked the various configurations on the floor so we could quickly replace boxes as well as change configurations easily. The space available to us for testing was in the theater building in Disaster City. Thus, like an actual theater, the floor was sloped. In this case the slope was 10°.

In addition one problem with the previous tests was that responders did not have a adequate training on the robots. While we did not have access to vendor supplied training, we did give the responders an opportunity to practice in the environment in a slalom course we set up next to the box course. Figure 5 shows the practice course responders used. We allowed the responders to traverse this course several times “eyes on.” When they felt comfortable, they traversed the course using the video feed only.

We collected two sets of data: the performance measures for navigating the course and ratings to a questionnaire that responders filled out after they completed the course. We do not have enough data for any statistical significance at this point, but we have enough data to help us refine the tests.

We asked the responders to provide information about their expertise with operating robots in general, and the robot used in the particular evaluation specifically. We also asked them to rate the difficulty of the exercise, how well they felt this evaluation would predict their performance with the user interface, and how well they felt this evaluation measured the user interface. We also asked them to rate specifics about the user interface and interactions with the robot they used for the specific test. We wanted to determine if there was a relationship between this and their actual performance. Table 1 shows the averages of the ratings concerning the evaluation methodology.



Figure 5. Practice course for the third version of the acceptable usability test

TABLE 1. Rating Questions

Question	Rating (1 is low; 7 is high)
Difficulty of the task	3.6
Predictor of how well you can perform US&R tasks	4.6
Indicator of the ease of use of the user interface?	5.2

The data from the questionnaire indicates that the responders feel we are on the right track with the evaluation. The ratings for an indicator of ease of use of the user interface are especially encouraging. The responders did not feel that the task was extremely difficult, despite the fact that six of the seven responders had no training on the specific robots other than the practice time we provided. One responder had several hours – most likely this occurred in other exercises earlier in the week. Two responders had no general training or practice with robots. Of the five others, the amount of

training/practice ranged from several hours to 3-4 years.

Table 2 shows the performance data from the responders along with their ratings of the particular robot interface used. The times for completion were quite variable between robots. The times for navigating the maze were less variable between users. The times it took the users of R2 to complete the maze were considerably more than the times it took the users of R3 to complete the task. The user ratings for ease of navigation also reflected this. The user who failed to accurately identify one gate as too narrow (R2) had a lower rating for ease of assessing gate width and camera manipulation than did the other user. Again, while we lack enough data for any statistical analysis at this time, we do think the trend is promising.

In addition to the novice performance we also had experts (representatives from the robot company or owners of the robots) drive the robots. This was done to determine if we could place a lower bound on the performance, such as novices should be able to complete the maze in three times the time required by the expert. Table 3 shows the comparison of the expert times and the average of the novice times. We had either two or three novices complete the maze for four different robots.

TABLE 3. Comparison of expert and novice times

Robot	Expert time	Average of novice time
Robot 1	11:05	5:07
Robot 2	6:14	5:06
Robot 3	4:20	16:04
Robot 4	6:04	15:27

TABLE 2. Data collection

Data	R1- user 1	R1- user 2	R2- user 1	R2- user 2	R3- user 1	R3- user 2	R3- user 3
Time	NA	NA	18 min	11 min	4:47 min	5:28 min	NA*
Number of gates called	NA	NA	6/6	5/6	6/6	6/6	4/4*
Ease of use	5	4	3	5	4.5	5	5
UI info	6	6	5	-	6	5	NA
Navigation	5	4	2	4	5	6	5
Camera manipulation	6	5	7	6	Camera not working – eyes on run		
Assessing Gate Width	5	5	6	4	Camera not working – eyes on run		

*This user did not complete the entire test due to a lack of time.

The results of the comparison of expert and novice times were rather surprising. There are several issues here. First, we need to ensure that the “experts” are really “experts.” We will need to develop requirements for operation time or some test of operational skill to ensure that we have true experts. However, we can also infer that Robot 1 and Robot 2 were relatively easy to operate and that Robot 3 and Robot 4 were much less intuitive.

IV. NEXT STEPS

We have not discussed how we will score the evaluation as it combines navigation, camera manipulation and situation awareness (determining which gates the robot will fit through). Our first thoughts were to award one point for each gate that the robot went through cleanly and another point for traversing each row cleanly. This would be a navigation score. The camera manipulation score would be computed based on the number of marked gates correctly identified by the operator. The situation awareness score would be the number of gates that the operator identified correctly as fitting through. In theory there is no reason that any of these tests could not be completed by an autonomous robot, assuming that the robot could automatically identify the hazard tape markings. There is an issue of the use of time in navigating the maze as this includes not only the time to drive, but the time to manipulate the cameras as needed and any time needed to make an assessment of the gate width. However, this is more of a true world measure that just time to navigate. Our biggest concern at this time is the identification of classes of robots by size and steering mechanism. We would like to identify a number of classes and design the test for those classes rather than using a percentage of each robots’ dimensions to configure the gates, row width and gate displacement.

Currently, we found no tele-operated robots with any additional features for situation awareness so we can consider eliminating this part of the test as it depends only on the operator’s ability to estimate the size of the robot. We are also considering developing a more rigorous camera manipulation test. This would consist of having the robot placed in an appropriately sized box with numbers and letters on the sides, top and bottom of the box. Without moving, the operator would be asked to move the camera(s) to read as many of the symbols as possible as quickly as possible. For each robot, we could calculate the total possible given the cameras. The metric would be computed using the ratio of those correctly read to those possible factoring in the time taken.

We will also vary the lighting conditions both in the maze (low light and completely dark) and the lighting conditions of the operator (direct sunlight with glare and low light). We

need to find a method for specifying the exact lighting conditions and duplicating those in another environment. We plan to run this evaluation with many more robots and with both expert and novice users, preferably who have had some basic training. We will collect the same type of data to determine if the performance measures correlate with the subjective ratings of the user interfaces. We are also considering having experts in human-robot interaction rate the quality of the interfaces and to determine if those ratings have any correlation with the performance measures.

ACKNOWLEDGEMENT

The authors would like to thank Elena Messina, Adam Jacoff, Brian Weiss, and Ann Virts for all their help and support. Our thanks also to the Responders who participated in all our pilot evaluations and to the Department of Homeland Security for providing the funding for this research.

REFERENCES

- [1] Preliminary Report
http://www.isd.mel.nist.gov/US&R_Robot_Standards/
accessed May 18, 2006
- [2] ISO 9241 part 11

ⁱ This research was conducted while Dr. Scholtz was at NIST.

Measuring Up as an Intelligent Robot – On the Use of High-Fidelity Simulations for Human-Robot Interaction Research

Anders Green

KTH School of Computer
Science and Communication
Royal Institute of Technology
100 44 Stockholm, Sweden
green@csc.kth.se

Helge Hüttenrauch

KTH School of Computer
Science and Communication
Royal Institute of Technology
100 44 Stockholm, Sweden
hehu@csc.kth.se

Elin A. Topp

KTH School of Computer
Science and Communication
Royal Institute of Technology
100 44 Stockholm, Sweden
topp@csc.kth.se

Abstract— In this article we describe and discuss how Hi-Fi simulation methods, or Wizard-of-Oz [25], can be employed in a user-centered approach to develop natural language user interfaces for robots with cognitive capabilities. By actively shaping the system both in terms of technology and in interface design the interest of both users and designers are involved in the process as stake-holders. We have analyzed two different simulation studies performed sequentially with different foci, but within the same overall scenario. The data obtained from Hi-Fi simulation studies is to a large extent qualitative, but data can also be used as a resource to be used for component development, e.g. as training data for speech recognizers. The collected data may also be used for quantitative evaluation depending on the experiment design.

I. INTRODUCTION

The creation of an interface for an autonomous robot is to some extent a very different undertaking than developing a graphical user interface for software application for the standard desktop computer.

In the European Cogniron project [1] robots with cognitive capabilities are investigated from different perspectives. The overall aim of the project is not to produce a single robot, instead several scenarios that explore and demonstrate different types of capabilities are investigated.

In this article we describe and discuss how Hi-Fi simulation methods, or Wizard-of-Oz [25], can be used in the research on human-robot interaction, where the specific aim is to explore and investigate interaction models for robots with cognitive capabilities. We have chosen to work with Hi-Fi simulations because we are dealing with technology for which users have very vague ideas, i.e. there are no cognitive service robots available on the market. Instead users' conceptions of service robots with cognitive capabilities are formed from how robots are presented in popular culture. Not even designers, i.e. roboticists, interface designers and usability experts etc, can fully predict in what manner the *robotic artifact* they are creating will be used and accepted. Maulsby et al [27] noted that "playing the role of an agent" enabled the designer to get

an in-depth understanding of what type of actions the system needed to provide. This is also appears to be the case for human-robot interaction, thus in order to understand how to design the interactive behavior of a cognitive companion robot we need to understand what this behavior should be like, i.e., formulated as a research challenge:

- *What are the requirements for the interactive behavior of a cognitive companion?*

This research challenge is not as straight-forward as it may seem. First of all the term "cognitive companion" is not well defined, but if we use the scenarios defined in the Cogniron project as a definition we end up with a system that should have the ability to use vision and audio to perceive and understand its environment, to engage in a continuous process of acquiring new knowledge and skills to learn tasks from users by interacting with them. The robot also needs to exhibit socially acceptable behaviors [1].



Fig. 1. The robot used in the two user studies in a scene from the first scenario.

The rest of this article is outlined as follows, first we describe the Wizard-of-Oz methodology and how this is can be used in user-centered approaches to design of human-robot interfaces. Then we describe, compare and discuss two different studies that explore the theme of the “Home tour” scenario.

A. Wizard-of-Oz methodology

High-fidelity simulation, or Wizard-of-Oz simulation, is a methodology used for simulation of high-level functions in an interactive system. The general idea is to simulate those parts of the system that require most effort in terms of development (like a natural language understanding module) or to assess the suitability of the chosen metaphor. One of the most common uses of Wizard-of-Oz is employed for finding out how users treat a system that uses natural language as an interface.

Wizard-of-Oz simulation methodology in its classical form, i.e., where one user interacts with one (desktop) computer in a lab environment, has been used since the 1970s. The term itself was first used by Kelley [25] two decades ago. Malhotra [26] used the method for simulation of natural language expert systems. The method has also been used for simulation of database question answering systems [7], [27] and uni-modal speech interfaces [3], but also for development of multi-modal interfaces [29], [30], [32].

The starting point for a Wizard-of-Oz study involves the construction of a prototype where some features of the system are for real and where some functions are simulated by one or more operators who control the system’s actions and responses.

A classical setup is to put a user in front of a desktop computer in one room and an operator, a wizard, in another room. The user is given a scenario by a test leader; a set of tasks to solve using the novel system and the interactions between the user and the system are recorded. Since the user often is unacquainted with systems of that particular kind, e.g. speech interfaces, or the task, the characteristics of the setup are that of a kind of a role-play, where the user tries to engage and act within the given scenario. Once the experiment is started the user is allowed to interact with the system in the same way as if the system was for real.

The wizard acts as the system’s high-level reasoning component responding to the users actions. During the experiment the test leader may intervene if the user gets into trouble related to the use of the (simulated) system. After the user has completed the scenario the test leader normally performs some kind of post session interview. At the end of this interview the test leader briefs the user that the system was in fact simulated and asks permission to use the data that has been collected during the experiment for research purposes. If the user does not give his/her permission the recordings are noted as unanswered and the data media is erased. If there is some kind of reward to be given to the user this will be handed over irrespectively of a negative or positive reply to the request to use data.

B. User-centered design process

Hi-fi simulation is to a large extent a method that provides qualitative data on human-robot interaction. Hence, we see it as primarily important in the very early stages of the design process. During the early phases of the system development methods that can be used to communicate the intended use of the product, e.g. mock-ups [9], [14] and scenarios with synthetic dialogues [17] are of great interest in order to inform design. When more mature prototypes have been implemented standard methods for usability evaluation [22] can be employed in order to ensure that the desired usability goals are met [8].

By now, the interaction model for a graphical user interface has become a de facto standard, and is thus known to virtually all users. The interaction model for a robot, however, is presented to the user as a *tabula rasa*, both in terms of lack of knowledge to predict the behavior of interface components and the type of tasks that the robot can perform. Normally a product that is released to the market has undergone several iterations of design and re-design. This process may be of lesser or greater complexity and the methods and work practices of may differ. However, given the fact that we in a sense are developing a “product”, we may still employ methods from user-centered software engineering.

By employing a user-centered approach we implicitly accept that we are not only observing The process of developing the user interface, e.g., by taking on the role of evaluating system and human performance along the lines of [35]. Instead we are actively shaping the system both in terms of technology [36] and in interface design [14], [20], hence both users and designers have a role as stake-holders in this process. Thus we need to identify values that we aim to support with our systems [12], as well as to understand that we as researchers are part of the process [4] and not merely evaluating a system that has been created by others.

C. Hi-Fi Simulation studies of Human-Robot Interaction

In the Human-Robot Interaction domain there are some compelling reasons for performing simulations rather extracting data from human-human communication.

In many of the envisioned scenarios human-robot communication is focused on communication about topics that are already known to the participants during conversation, making it unnecessary to address these explicitly using speech during the execution of a task. First of all there is the matter of the task that the robot is performing. The tasks that we address the robot with are either simple or focused on domains where explicit verbal instructions are rare or only sparsely used by humans, but seem very important for robots. For instance, explicit negotiation of easily understood tasks (e.g. fetching an object from a known location) [17], and communication concerning detection of humans and description of routes [13]. Secondly, even in scenarios where robots are intended to replace people, in dirty, dangerous or distant environments, the work allocation between actual workers and robots may differ (see [6] for an example of extreme use). This means

that the conversation between field workers, like divers or rescue workers would probably be of little direct use to when developing modules for spoken human-robot communication.

We may also perform simulation studies in order to investigate hypotheses concerning general aspects of human-robot interaction, such as social behaviors, e.g. studies of spatial positioning [40], [41] and collaboration [21].

As a method Hi-fi simulation supports a wide variety of activities where designers and users increase their understanding of a future system. According to [15] these activities can be categorized along the dimensions try-out, training and testing:

- *Try out and conceptualize design ideas*, i.e., to rapidly set-up and perform formative studies in order to conceptualize and try out new ideas for the interface that is being developed.
- *Training and education*, i.e., to train natural language components such as speech recognizers. It also provides an opportunity to educate interface designers by letting them to take on the role of the system [27], and thereby increase their understanding of the task.
- *Testing, evaluation and study of use*, using standard usability methods for the type of interface that is being evaluated, for instance using the “Common metrics” approach according to [35] or evaluation methods that are especially targeted for natural language user interfaces, e.g., PARADISE [39]. Last but not least, we include the study of systems in realistic usage contexts, i.e, observations that can be analyzed to increase our understanding of human-robot interaction in general.

II. WIZARD SIMULATION STRATEGIES

When employing Hi-Fi simulation techniques, such as Wizard-of-Oz, we need to establish what kind of data we are aiming to collect and how it should be analyzed. When we are simulating an autonomous robot with the ability to communicate in natural language we have identified a set of dimensions, in terms of behaviors and appearance, along which the system can vary:

- Degree of system realism: Are we going to simulate a “realistic” system? We must decide to what extent we are going to simulate system behavior so that they appear as realistic to the user. For instance, if we are simulating navigation capability we need to assure that the wizard does not “equip” the system with a navigation system that is able to plan ahead, seemingly inferring the intent of the user. Hence it is important to provide a set of constraints that bring some realism into the situation of use. This is what Maulsby [27] refers to being “true to the algorithm”.
- Degree of exploratory freedom: Are we going to simulate according to an “algorithm”? Decisions that are made along this dimension concern whether or not we should allow a completely free and explorative interaction style, or if we should restrict the task
- Behavioral mimicry: how “natural-like” should the system be in terms of appearance and interactive behaviors, i.e, to what extent should the system imitate “nature”, in

terms of appearance (e.g, using close resemblance with nature [31]) or interactive capability, e.g, allowing for a conversational style of interaction that is close to human capability.

A. Validity of Wizard-of-Oz simulation

Sometimes the Wizard-of-Oz method is criticized for not providing necessary data for evaluating practical dialogue. Wizard-of-Oz methods typically fail to involve users that bring real tasks to the system [23]. This should be seen in contrast to what is generally believed, namely that the Wizard-of-Oz method is an open-ended method for collection of data on user behavior. Allwood and Haglund [2] have noted that the wizard operator acting a scenario is involved in roles on different levels. Thus the researcher role involves acting as a system (wizard operator) and during sessions the wizard can take on different communicative roles like the sender role, the receiver role etc. Bell [5] notes that in a task scenario, like a travel agency dialogue, the wizard not only acts in a system role but in the role of a travel agent. In reality the behavior of people acting in a real situation may be quite different from what people do in a simulated scenario even if the user believes that she is interacting with a real system.

When simulating a multi-modal system with many different ways for the user of providing input the cognitive load of the operators increases. There is sometimes a mismatch between what needs to be simulated and what suits the cognitive and perceptual abilities of the wizards. This phenomenon was been noted by Fitts [10] who proposed a list describing what people do better than machines and what machines do better than people¹. In a simulation scenario, the problem of function allocation is twofold. First of all we need to consider the characteristics of the system we are simulating and secondly we need to think about the function allocation of the system we are using for the simulation. Following Sheridan [33] we see may see Fitts’ List as a set of accepted statements making up the foundation for how to reason when designing function allocation. According to Fitts people are better than machines at detecting small changes in the environment, perceiving patterns; improvising, memorizing, making judgments and reasoning inductively. Machines are better at responding quickly to signals, applying great force, storing and erasing information and reasoning deductively [33]. It is therefore important that we consider the strengths and weaknesses of machines and humans when designing and planning Wizard-of-Oz setups.

Ethical considerations

The use of the Wizard-of-Oz method provides a possible ethical dilemma for the researcher, since in the classical setup, the user is led to believe that the system is a real machine. When the truth is revealed afterwards, the test-leader normally asks for the permission to use the data. There are a number of studies that use deception; in fact it seems that the use of

¹The so-called MABA-MABA list.

the term Wizard-of-Oz is always connected to the practice of deceiving the user. Fraser and Gilbert [11] have argued against deceiving users on ethical grounds. This ethical dilemma might be solved by telling the user that the system is simulated, making the setup part of the role-play. The validity of the results from a study that is performed in this overt fashion may be questioned but this is dependent on the purpose with the study. Dahlbäck et al [7] argue that for some aspects of dialogue, like the type and frequency of anaphoric references, it is important that users are deceived rather than engaging in a role-play type of scenario. We should note that the work by Dahlbäck [7] and others in the late eighties was mostly done with systems that used text input. For spoken scenarios role-playing has been considered useful in multi-user scenarios [24] and elicitation of error handling strategies [34]. Human natural language performance is automated to a great extent and therefore we may assume that phenomena on the low-level like syntax, and vocabulary used probably are little affected by the fact that the user knows that the system is simulated or not.

However, using the Wizard-of-Oz method in the classical, deceptive manner, ensures the illusion of speaking with a computer. In a non-deceptive Wizard-of-Oz study, this illusion might be broken, something which may have an adverse effect on the result. However, it seems strange to use made up scenarios and role-play while maintaining the standpoint that we should deceive the user to the extent that the system is in fact real. In our view is that it is still an open question if deception is really necessary?

III. STUDYING THE “HOME TOUR SCENARIO”

The two studies that we are discussing in this paper both stem from work that is performed in the context of the European project Cogniron [1] where a set of scenarios has been defined. In the scenario we have worked with primarily, the user gives the robot a “tour” of the environment:

In this experiment, a robot discovers a home-like environment and builds up an understanding of it and of artifacts in it as taught by humans. This process is open-ended, i.e., it has no completion: the robot continues to learn as it faces new situations. A human shows and names specific locations, objects and artifacts, to the robot. The robot can engage in a dialogue in case of missing or ambiguous information.

This scenario can be characterized as kind of Co-operative Service Discovery and Configuration, stressing the way the user and robot is intended to engage in a joint effort to inform each other of relevant knowledge about the environment. The central themes that the scenario aims to address concerns two main types of information to be jointly discovered by the user and the robot:

- (i) the artifacts present in the environment (e.g. objects and locations) and,
- (ii) the actions that the robot can perform related to these artifacts.

The two studies that we discuss in this paper explore two key concepts. We have already introduced *Cooperative-Service Discovery and Configuration* which can be viewed as an abstraction of the scenario described above. In the second study the notion of Human-Augmented Mapping (HAM, [36]) was evaluated both in terms of interaction and from a technical point of view. In the following we will compare the set-up and the result of these studies. An overview is given in the Tables I and II.

The general purpose of the first study was to evaluate an extended dialogue model based on [19], [38] in order to provide a rich set of data for further analysis of human-robot communication. The first study was primarily aimed at exploration and not formal hypothesis testing. We did however put down a list of things that we believed that users would do in different situations during the study in order figure out what the wizards should do with the aim to provide a consistent behavior throughout the session.

In the second study, which can be characterized as a pilot study, the general aim was slightly different. We still have an explorative purpose based on the assumption that individuals have different preferences and strategies for how to present their everyday environment. The second study can also be seen as a technical test of a way of represent the environment. Using the notion of MacNamara [28]) we assume that individual strategies are based upon a common psychological representation the meaning and conventionalized use of the terminology that describe users environment. Thus, during the sessions, we use the communicated concepts to annotate the environment representation (i.e., nodes with the types location or region). This means that the role of the wizard is to decide whether the communicated concepts, e.g. phrases like “kitchen” and “my office”, can be categorized as locations and regions according to the environment representation. Study two thus aims to answer to questions:

- What strategies are used by individual users to *present* the environment?
- How can the information given by users be incorporated in a representation that can be used as a shared representation.

Both studies can be characterized as Hi-Fi simulations, the first study was performed as pure wizard study, i.e. users were not informed that the system was controlled by a wizard operators. The wizard operators were not completely hidden, instead the users were told that they were “technical support staff” that “monitored the experiment”.

When there are two wizard operators it is necessary to carefully consider the task allocation between wizards and to support their collaboration. In the first study there were two wizards, one controlling the dialogue (communicator) and one controlling the movements of the robot (navigator). The wizards stood close together during the session to support contextual awareness, meaning that they easily could get a glimpse of what the other wizard was doing while concentrating on the task. The setup can be seen in Figure 2.

In the second study the users were informed that the speech system was controlled by the accompanying operator. Whereas the first study was performed with 22 invited users (undergraduates from our campus) and thus did not know the environment, the 5 users in the second study were all staff, familiar with the lab environment. As familiarity with the environment was a requirement depending on the research questions posed in the second study lab staff was chosen. In future studies we aim to move the robot to an environment outside the lab so that users that have little experience with robotics can be studied.

The first study was performed in a single room in our lab (Figure 2), furnished like a living room to provide a home-like character. The second study was performed in a significantly larger environment, i.e., a whole floor of our lab comprising about 20 rooms (offices and common areas like kitchen, meeting room and printer area).

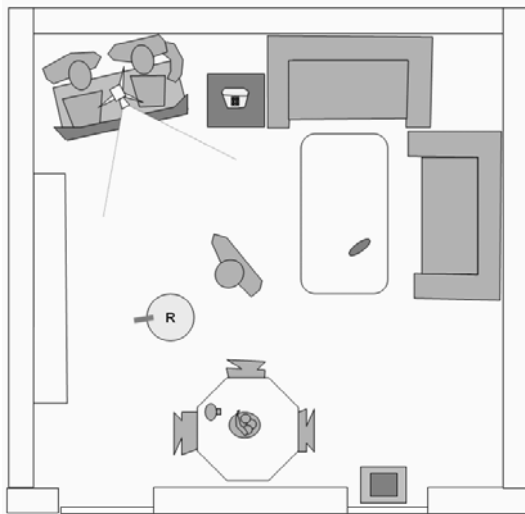


Fig. 2. Map showing the single room used in Study 1

a) Robot appearance, behavior and instructions to users:

In both studies we used an ActivMedia PeopleBot equipped with an on-board pan-tilt-camera (see Figure 1. In the first study the robot also was equipped with visible stereo microphones.

In the first study, in terms of behavior the movements (navigation and camera movements) of the robot was controlled by one wizard and the speech output was controlled by another. Since we were aware that a real robot navigation component would make the robot behave in a specific way we tried to mimic this behavior as far as it was possible. In the second study a real follow behavior was used to solve all tasks, except for error recovery, which was performed overtly as teleoperation.

In terms of interactive behavior the first study was more focused on providing a coherent and consistent dialogue behavior, i.e., the dialogue wizard tried to follow an “algorithm” using the terminology by [27]. Hence the subjects received instructions with clear examples of how they could use the

system. Our goal was not to test the system “in the wild” without informing the users about how the system was intended to work. Instead we very much instructed the users so that they would be able to use the system. Hence, the instructions were intentionally clear with exact phrasing. The navigation wizard, who also acted as the test leader, also showed the user how the robot worked by giving a small demonstration.

The robot had a dialogue model that accommodated for the actions:

- Greeting, e.g., phrases like “hello” and “my name is john”.
- Follow, e.g., phrases like “follow me”
- Demonstration, e.g., referential phrases and gestures like “this is an orange” + pointing gesture.
- Validation, e.g., phrases like “find the orange!”

For the second study the robot used two different ways of giving feedback on users’ input, dependent if the user provided information for labeling a *location* or a *region*. Thus if a location, e.g., “coffee machine” or “door to” a *region*, was presented by the user the robot would not move and immediately state that it had stored the location. If a region, e.g., “kitchen”, “my office”, was presented, the robot would state that it needed to “look around” and make a full 360 degree turn, before confirming that a *region* had been stored. The role of the wizard in was to select which behavior to use based on the wizards understanding if whether the concept used by the user could be interpreted as referencing a *location* or a *region*.

Users were told to use commands, i.e., follow me, *go to* <target>, *stop*, *turn* <left/right>, to control the robot and referential expressions like *this is* <term> together with deictic gestures. No terminology that could be directly related to the concepts tested was mentioned to the users, e.g., *location*, *region*, *place* etc. Instead phrases like “everything, that you think the robot needs to know” was used to guide the user. Since a real follow behavior was used special instructions was needed in order that the system would be able to track and follow the user (i.e., the user had to move about 1 meter to make the robot start following).

b) Data collection setup : In the first study we could collect data in a fairly controlled manner using a video camera placed in the location of the users. We also had web-cameras placed in each corner of the room. Mounted on-board the robot an audio recorder captured high quality stereo sound. Data from the laser range finder was also stored.

During the second study we used a hand-held camera and on-board camera to capture the “scene” of the user and the robot acting together in the environment. Since the environment comprised over twenty offices and common areas (e.g, kitchen, printer area) we could not cover the whole area with web cams or sensors.

In both studies we administered a questionnaire and interviewed the users after the end of the trial. In the first study we asked the users about who was controlling the robot before informing them about that they had been involved in a wizard study and asked again about their willingness

TABLE I
GENERAL PURPOSE AND SETUP OF USER STUDIES 1 AND 2

STUDY TYPE	Study 1: Single room	Study 2: Multiple rooms
Purpose	Data collection and exploration of interaction strategies related to a specific dialogue model [19], [38]	Investigate interaction strategies used by individual users to <i>present</i> the environment
Type, size of study	Explorative, 22 subjects (20+20 min)	Pilot-study with 5 subjects (20+25 min)
Key concept	Co-operative Service Discovery and Configuration	Human-Augmented Mapping (cf. [36])
Type of subjects	Undergraduate students	Staff that knew their environment
Environment	Single “living room”	20 rooms + kitchen

TABLE II
INTERACTION DIMENSIONS OF USER STUDIES 1 AND 2

INTERACTION	Study 1: Evaluation of Dialogue model	Study 2: Presentation strategy
Robot behavior and appearance	Dialogue model accomodating tasks: greet, follow, show (object, location), validate, close	Two behaviors: show (acknowledge) and show (look-around)
Instructions	User were provided with a written instruction with examples. Testleader gave a demonstration.	Users were told to use a small set of commands and gestures
Wizard-status	Users were informed about the role of robot operator	Pure wizard – user were told that wizard was an operator
Number and role of wizards	Two wizards (navigator and communicator) behind a screen (cf. [15])	One wizard accompanied user/robot
Data collection setup	<ul style="list-style-type: none"> • Video camera (overview). Onboard stereo audio (robot perspective). • Network web cameras in each corner. • Questionnaire • Post-trial interview 	<ul style="list-style-type: none"> • Hand-held camera (overview) / onboard camera (robot perspective) • Questionnaire • Post-trial interview (scripted) • Sketch session
Type of data	<ul style="list-style-type: none"> • Single audio/video track (Mini DV) • Web-cam images (~ 1 fps) • Data from laser range finder • Stereo audio (22 KHz) • Notes from interviews • Questionnaire data 	<ul style="list-style-type: none"> • Two video tracks with audio (Mini DV) • Notes from interviews • Data from laser range finder • Environment representation • Questionnaire data • Video from sketch session (Mini DV)

to participate given this new information. No user had any objections once we had revealed the truth and explained the reasons for our conduct. In the second study the subjects were informed that the system was partially simulated. The post-session interviews in the second experiment also included a sketch session where the user was instructed to draw a map on a white-board and explain how they had instructed the robot.

A. Findings from the studies

Several analyses have been performed using the data collected. An analysis of the communicative acts have been made and reported in [16]. We have also studied the role of posture and positioning [20] and active spatial influence on the part of the robot [18].

We performed an analysis of the miscommunication using the video recordings from the first 12 user sessions. These were transcribed and synchronized on the utterance level. We then printed out all dialogues and analyzed them by

marking utterances that could be interpreted as symptoms of miscommunication.

We found miscommunication that could be attributed to the users’ erroneous inferences about the system’s capability. For instance, users tried to hold up objects in front of the camera. This was considered to be an error according to the task model and an a repair was issued by the communicator wizard.

We also noted several types of problems related to feedback in our data. Providing relevant and timely feedback essential to maintaining an orderly and well managed dialogue. We have identified problems related to timing, i.e., feedback is *ill-timed*, something which may render it incoherent. Another problem that occurred in the material was lack of feedback, i.e., the robot does not respond to the user’s contribution before the user decides to make another contribution. Overlapping speech was also a problem in the corpus, i.e, the robot should have stopped the synthesized speech when the robot started to speak, something which lead to misunderstandings.

Another problem that can be described as an effect of the modality is miscommunication related to reference. In the manner the system was simulated we allowed for a "robust" object recognition system, meaning that the system would recognize any object given that it was small enough and placed on a flat surface. The information can be said to have been negotiated, but since there is no pointing capability apart from the general direction indicated by the front robot and the on-board camera, there is no way of indicating precisely which object has been detected.

Results from the second study have been reported in terms of a technical evaluation of the environment representation [37]. The strategies employed by the users were very different and no convergence in behavior could be established since there were only five subjects. The observed behaviors still need to be taken into account when constructing future systems. Some of the phenomena are provided below to give a sense of what went on during the sessions:

- Users' personal view of the environment affects the way it is introduced to the robot, e.g., a person not drinking coffee might not mention the coffee maker.
- Persons were pointed out to the robot as offices (with occupants in them) were passed. Bystanders that passed the robot were also pointed out by one user.
- Locations pointed out reflected an action oriented view of the environment, i.e., a coffee machine is mentioned as a place where the robot should get coffee, but the kitchen is not mentioned.
- Doorways were pointed out by subjects with knowledge about robot navigation
- After giving the follow-command, the users stood still and waited for the robot to move, something which was not anticipated. Hence it needs to be equipped with an active way of prompting the user to move (i.e., perform an act of *spatial prompting* [18]).

IV. DISCUSSION

When looking at the two different studies in a bird's eye perspective, the generic aspects of both studies can be contributed the context of use provided by the overall scenario, the Home Tour:

- The use of natural language using speech and gestures
- The initiative during the interaction lies with the user
- The behavior of the robot influences the user both in terms of communicative behavior and spatial adaptation.

In some respect the different foci of the studies affect the way studies have been setup. When mobility of both users and systems become a topic for investigation, the complexity of setting up the scenario increases, something which have an effect the way data can be collected.

Clearly the wizard-of-oz framework can be employed successfully for simulating a full fledged interactive robot system. There are several dimensions that have been added since the early work on screen-based applications [7], [25]–[27]. The dimension of collaborative behavior has always been in focus

of wizard-of-oz studies for natural language user interfaces. For human-robot interaction the situated use context is in focus, i.e., the human and robot share the same environment and partially share the same perceptual context.

Since a robot system allow several degrees of freedom the manner and ability for the wizards to control the robot system almost becomes research topic in itself. It is clear that to some extent we need to assure that we do not spend more resources on building a simulation environment than on building the real system. The use of a real robots also makes safety issues come in focus, hence we need to assure that we can maintain safety during the sessions. We also need to check that the more formal aspects of user studies, i.e., make certain that legal and ethical issues are compatible with the type of studies we will perform (see Walters et al [40] for a specific example from the UK).

V. CONCLUSIONS

In this article we have described the way in which Hi-Fi simulation studies can be used in the process of developing interactive service robots. When carefully designed, simulation studies will provide data about different aspects of human-robot interaction that would otherwise be unattainable until large effort had been spent on the creation of a working prototype. Thus, the data that we get from a Hi-Fi simulation study is to a large extent qualitative, but we may also collect data as a resource to be used for component development, e.g. as training data for speech recognizers. We may also use the collected data to perform quantitative evaluations using different performance metrics, e.g. time-to-completion for a defined task [35] or user satisfaction [39] with respect to interactivity.

The type of data we can get a wizard-of-oz type simulation study ranges from collection of data that can be analyzed in different ways.

- Data on language use and task strategies, especially spatial language (including gesture and speech) can be analyzed with methods from psychology, linguistics etc.
- Visualization of the whole interaction to enable the designer to conceptualize what the system could or should do in different realistic situations.
- Assessment of users' attitudes towards a future system or towards robots in general.

In the future our aim is to evaluate systems where tasks that now are handled by wizard operators gradually are replaced with real components. To succeed with this we need investigate how we can provide a system which *can* be managed by wizards, while providing the necessary exploratory freedom for users together with requirement of an appropriate degree of system realism.

VI. ACKNOWLEDGMENT

The work described in this paper was conducted within the EU Integrated Project COGNIRON ('The Cognitive Robot Companion' - www.cogniron.org) and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

REFERENCES

- [1] COGNIRON, Annex 1 - Description of Work, 2003. EU Sixth Framework Program FP6-IST-002020, <http://www.cogniron.org>.
- [2] J. Allwood and B. Haglund. Communicative Activity Analysis of a Wizard of Oz Experiment. Technical report, Department of Linguistics, Göteborg University, 1992.
- [3] G. Antoniol, R. Cattoni, M. Cettolo, and M. Federico. Robust Speech Understanding for Robot Telecontrol. In *Proceedings of the 6th International Conference on Advanced robotics*, pages 205–209, Tokyo, Japan, 1993.
- [4] L. J. Bannon and S. Boedker. *Designing Interaction: Psychology at the human-computer interface*, chapter Beyond the Interface: Encountering Artifacts in Use. Cambridge University Press, New York, 1991.
- [5] L. Bell. *Linguistic Adaptations in Spoken Human-Computer Dialogues: Empirical studies of User Behavior*. PhD Thesis, KTH Royal Institute of Technology, 2003. TRITA-TMH 2003:11.
- [6] J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 33(3):367–385, 2003.
- [7] N. Dahlbäck, A. Jönsson, and L. Ahrenberg. Wizard of Oz studies - why and how. *Knowledge-Based Systems*, 6(4):258–256, 1993.
- [8] P. Dario, E. Guglielmelli, and C. Laschi. Humanoids and personal robots: Design and experiments. *Journal of Robotic Systems*, 18(12):673–690, December 2001.
- [9] P. Ehn and M. Kyng. *Cardboard computers: mocking-it-up or hands-on the future*, pages 169–196. Lawrence Erlbaum Associates, Inc., 1992.
- [10] P. M. Fitts. *Human engineering for an effective air navigation and traffic control system*. National Research Council Committee on Aviation Psychology, Washington, DC, 1951.
- [11] N. M. Fraser and G. N. Gilbert. Simulating Speech Systems. *Computer Speech & Language*, 5(1):81–99, 1991.
- [12] B. Friedman and P. Kahn. The human-computer interaction handbook, chapter Human values, ethics, and design. Lawrence Erlbaum Associates, Mahwah, NJ, 2003. Eds: J. A. Jacko and A. Sears.
- [13] J. Fry, H. Asoh, and T. Matsui. Natural Dialogue with the JIJO-2 Office Robot. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:1278–1283, 1998.
- [14] A. Green, H. Hüttenrauch, M. Norman, L. Oestreicher, and K. Severinson Eklundh. User-Centered Design for Intelligent Service Robots. In *Proceedings of 9th IEEE International Workshop on Robot and Human Interactive Communication*, Osaka, Japan, 2000.
- [15] A. Green, H. Hüttenrauch, and K. Severinson Eklundh. Applying the Wizard-of-Oz framework to Cooperative Service Discovery and Configuration. In *13th IEEE International Workshop on Robot and Human Interactive Communication RO-MAN 2004*, pages 575–580, 20–22 Sept 2004.
- [16] A. Green, H. Hüttenrauch, E. A. Topp, and K. S. Eklundh. Developing a Contextualized Multimodal Corpus for Human-Robot Interaction. In *Proceedings of the Fifth international conference on Language Resources and Evaluation LREC2006*, 2006.
- [17] A. Green and K. Severinson Eklundh. Task-oriented Dialogue for CERO: a User-centered Approach. In *Proceedings of 10th IEEE International Workshop on Robot and Human Interactive Communication*, Bordeaux/Paris, September 2001.
- [18] A. Green, B. Wrede, K. S. Eklund, and S. Li. Integrating Miscommunication Analysis in the Natural Language Interface Design for a Service Robot. submitted to IROS2006, 2006.
- [19] A. Haasch, S. Hohenner, S. Huelwel, M. Kleinhagenbrock, S. Lang, I. Toptsis, G. A. Fink, J. Fritsch, B. Wrede, and G. Sagerer. BIRON – The Bielefeld Robot Companion. In *Proceedings of ASER 2004 - 2nd International Workshop on Advances in Service Robots*, Stuttgart, Germany, May 21 2004.
- [20] H. Hüttenrauch, A. Green, M. Norman, L. Oestreicher, and K. Eklundh Severinson. Involving Users in the Design of a Mobile Office Robot. *Systems, Man and Cybernetics, Part C: Applications and reviews*, 34(2):113–124, 2004.
- [21] H. Hüttenrauch and K. Severinson Eklundh. To Help or Not to Help a Service Robot. In *Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication RO-MAN'2003*, Millbrae CA, USA, 2003. IEEE.
- [22] ISO. International Standard ISO/IEC 9126. Information technology - Software product evaluation - Quality characteristics and guidelines for their use, International Organization for Standardization. 1991.
- [23] A. Jönsson and N. Dahlbäck. Distilling dialogues - A method using natural dialogue corpora for dialogue systems development. In *Proceedings of 6th Applied Natural Language Processing Conference*, pages 44–51, 2000.
- [24] K. Kanto, M. Cheadle, B. Gambäck, P. Hansen, H. K. Kristiina Jokinen, and J. Rissanen. Multi-Session Group Scenarios for Speech Interface Design. In C. Stephanidis and J. Jacko, editors, *Human-Computer Interaction: Theory and Practice (Part II)*, volume 2, pages 676–680. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2003.
- [25] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.
- [26] A. Malhotra. Design criteria for a knowledge-based English language system for management: an experimental analysis. Technical Report MAC TR-146, MIT, 1975.
- [27] D. Maulsby, S. Greenberg, and R. Mander. Prototyping an Intelligent Agent through Wizard of Oz. In *INTERCHI'93*, pages 277 – 282. ACM, April 1993.
- [28] T. McNamara. Mental representations of spatial relations. *Cognitive Psychology*, 18:86–121, 1986.
- [29] S. Oviatt. User-centered Modeling for Spoken Language and Multimodal Interfaces. *IEEE Multimedia*, 3(4):26–35, 1996.
- [30] D. Perzanowski, A. Schultz, W. Adams, E. Marsh, and M. Bugajska. Building a multimodal human-robot interface. *Intelligent Systems, IEEE [see also IEEE Expert]*, 16(1):16–21, 2001.
- [31] T. Saito, T. Shibata, K. Wada, and K. Tanie. Relationship between interaction with the mental commit robot and change of stress reaction of the elderly. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 1, pages 119–124, 16-20 July 2003.
- [32] D. Salber and J. Coutaz. Applying the Wizard of Oz Technique to the Study of Multimodal Systems. In *EWHCI*, pages 219–230. 1993.
- [33] T. B. Sheridan. Function allocation: algorithm, alchemy or apostasy? *International Journal of Human-Computer Studies*, 52(2):203–216, 2000.
- [34] G. Skantze. Exploring Human Error Handling Strategies: Implications for Spoken Dialogue Systems. In *Proceedings of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, pages 71–76, 2003.
- [35] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common Metrics for Human-Robot Interaction. In *In Proceedings of HRI2006 1st annual conference on Human-Robot Interaction*, Salt Lake City, UT, USA, March 2-3 2006. ACM.
- [36] E. A. Topp and H. I. Christensen. Tracking for Following and Passing Persons. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005)*, Edmonton, Alberta, August 2005.
- [37] E. A. Topp, H. Hüttenrauch, and H. I. C. K. Severinson Eklundh. Acquiring a shared environment representation. In *Proceedings of the 1st ACM Human Robot Interaction Conference, HRI2006*, Salt Lake City, Utah, USA, March 2006. (extended abstract).
- [38] I. Toptsis, S. Li, B. Wrede, and G. A. Fink. A Multi-modal Dialog System for a Mobile Robot. In *INTERSPEECH: 8th International Conference on Spoken Language Processing (ICSLP)*, volume 1, pages 273–276, Jeju, Korea, 2004.
- [39] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 271–280, Madrid, Spain, July 1997.
- [40] M. Walters, S. Woods, K. Koay, and K. Dautenhahn. Practical and methodological challenges in designing and conducting interaction studies with human subjects. In *Proceeding of AISB Symposium on Robot Companions Hard Problems and Open Challenges in Human-Robot Interaction*, pages 110–120, Hertfordshire UK, April 2005 2005.
- [41] M. L. Walters, K. Dautenhahn, K. L. Koay, C. Kaouri, R. te Boekhorst, C. L. Nehaniv, I. Werry, and D. Lee. Close encounters: Spatial distances between people and a robot of mechanistic appearance. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids2005)*, pages 450–455, Tsukuba, Japan, December 5-7 2005.

On-orbit Servicing: A Brief Survey

Andrew Tatsch
University of Florida
Mechanical and Aerospace
Engineering
231 MAE-A, PO Box 116250
Gainesville, FL, US
atatsch@ufl.edu

Norman Fitz-Coy
University of Florida
Mechanical and Aerospace
Engineering
231 MAE-A, PO Box 116250
Gainesville, FL, US
nfc@ufl.edu

Svetlana Gladun
The Boeing Company
Integrated Defense Systems
Huntington Beach, CA, US
Svetlana.A.Gladun@boeing.com

Abstract— A survey of on-orbit servicing (OOS) is presented. This survey touches on the history of servicing missions performed to-date and the evolution of the feasibility of a servicing infrastructure for the space industry. Furthermore, as a result of the absence of a servicing infrastructure in the space sector, an evaluation of the state of the art of OOS is achieved only by examining the level of sophistication of the individual subsystems inherent to servicing. Therefore, we outline the enabling technologies required for a fully functioning servicing architecture and evaluate the technology readiness level of each. Finally, current servicing concepts are presented and a fractionated space architecture for servicing being developed at the University of Florida is showcased.

Keywords: *on-orbit servicing, fractionation, responsive space*

I. INTRODUCTION

On-orbit servicing, that is the execution of assembly, repair, and maintenance as an in-space operation, has been around at least conceptually as early as the 1970s stemming from interest in large space structures. The first instance of OOS, the servicing of the Skylab space station, was performed in 1973 [1]. The early experiences with OOS required space suited humans performing the servicing tasks through EVA missions. On the other hand, early robotic OOS concepts were designed as monolithic spacecraft platforms that incorporated all of the subsystem technologies into a single unit, such as the Orbital Maneuvering Vehicle (OMV) [2]. The OMV was an important component of NASA's space station plans in the 1980s and it was intended as a short range robotic 'space tug' that could move payloads about in the vicinity of the Shuttle and Space Station. However, the high cost and lack of robustness of these early monolithic designs coupled with the immaturity of the subsystem technologies rendered OOS a non-profitable venture at that time, but the pursuit of more mature subsystem technologies continued due to other useful applications of these technologies, such as rendezvous and capture.

By the late 1980s into the early 1990s the development of subsystem technologies progressed to the point at which the financial implausibility became tantamount to OOS architecture development. Even with success of the STS-49

mission that serviced Intelsat VI in May 1992 and the STS-61 and STS-82 servicing of the Hubble Space Telescope in 1993 and 1997, all accomplished by human EVAs, the development of a robotic OOS infrastructure met unpopularity. Although the subsystem technologies had progressed, they were nowhere near the capabilities of a space-suited human and insufficient communications throughput excluded the possibility of a teleoperated OOS system. Furthermore, the financial analysis was conducted using net present value (NPV) projections, leading to inconclusive results.

Again in the late 1990s OSS was dealt a nearly fatal blow by the implementation of the "cheaper, better, faster" directive at NASA by Dan Goldin. The directive effectively rendered OOS unnecessary because now the replacement of spacecraft to be serviced was cost effective. However, just as the interest in large space structure gave birth to OOS, these structures resurrected the development of servicing architectures as plans for the International Space Station (ISS) materialized.

After the turn of the millennium, a steep resurgence of robotic OSS research precipitated. Partly due to exposure of "cheaper, better, faster" as an unsound means of doing business in the space sector, but primarily due to two aspects; new financial analysis that shed the NPV utilization, instead using cost independent and customer-centric perspectives showed that a market exists for a robotic OOS infrastructure. Second, as evidenced by the Space Robotics Technology Assessment Report [3] published by NASA in 2002, the technology readiness level (TRL) of OOS subsystems were nearing flight test status. These improved technologies together with the new financial analysis resulted in the immense interest in a robotic OOS infrastructure at the present time.

II. FEASIBILITY OF ON-ORBIT SERVICING

Identifying the feasibility of OOS requires both analysis from a fiscal perspective, but also analysis to quantify the number of servicing opportunities. To aid in the former, Sullivan and Akin [4] developed a database of only spacecraft failures from 1981-2001. Even though their analysis ignored other servicing applications, such as refueling, on-orbit

upgrades, and debris mitigation, this database indicated regular opportunities for satellite servicing, on the order of 10 to 20 opportunities annually.

In order to evaluate the fiscal viability of OOS, numerous cost benefits analyses have been published during the past decade [5] - [13]. Net present value (NPV) based financial analysis on the feasibility of OOS found that the uncertainties involved are too large and thus renders inconclusive results. Kreisel [8] has shown via a cost-independent analysis that a viable market exists for ISO based on the number of current and planned space assets with a 25%-50% assumed customer utilization of servicing. Additionally, Saleh, et. al. [9] - [11], presents analysis from a customer-centric perspective that promotes the value of OOS to the customer based on the inherent flexibility added to space assets by OOS capabilities. However, all of the cost-based or value-based analysis to date assumes the financial burden for development of an OOS infrastructure is carried only over the service life of the space assets.

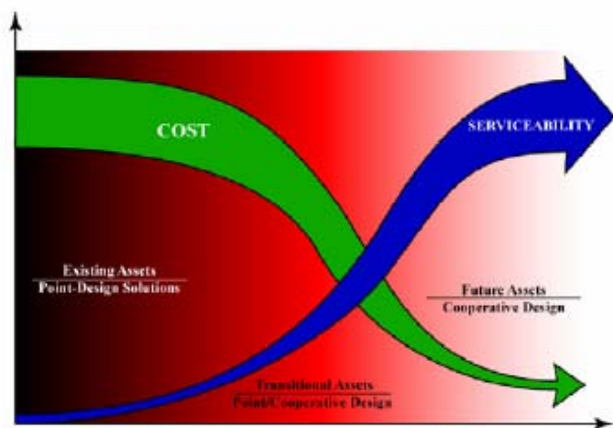


Fig. 1. Cost vs. Serviceability

From a broader perspective, if the cost of the development is prorated between the “current”, “transitional”, and “future” assets, coupled with the cost-savings incurred through cooperative development of future assets, a justification for OOS can be elucidated as shown in Fig. 1. Here, “Current Assets” refers to assets that are already in orbit (therefore requiring a point-design solution for servicing), “Transitional Assets” refers to assets that are too far along in their design and/or manufacturing process that a re-design is not financially plausible, and “Future Assets” refers to assets that can be cooperatively designed. When the time scale is expanded to include the entire developmental cycle necessary to achieve a mature low risk OOS infrastructure, the cost-to-serviceability trends can be described as shown in Fig. 1. Initially while the OOS infrastructure is being developed, the space assets will be dominated by “Existing Assets” and to a lesser extent “Transitional Assets”. During these initial phases the cost of servicing remains high, due to point-design solutions and technology development, with limited serviceability. However, as the technology matures and a

knowledge base acquired from experience is created, serviceability becomes enhanced and the costs begin to deflate. Furthermore, with established OOS technologies and cooperatively designed future assets (designed with intent to be serviced), a maximum serviceability to cost ratio can be realized. Therefore, if the major burden, that being the largest portion of cost with limited serviceability of assets, is prorated over the three segments of the development cycle, then OOS becomes justifiable.

III. OOS: STATE OF THE ART

In this section both current state of the art and experimental test bed sophistication are assessed for in-space assembly, repair, and maintenance functionalities. The implementations to-date of in-space servicing subsystems are emphasized while noting the progress possible from the experimental investigations.

A. Assembly and Repair

The current state of assembly and repair functionalities, with respect to real world implementations, is restricted to the Shuttle Remote Manipulator System (SRMS) and the Space Station Remote Manipulator System (SSRMS) aboard the International Space Station (ISS). Neither of the current systems possesses any autonomy as they are solely teleoperated on-board or from the ground. However, to date there have been on-orbit demonstrations of assembly subsystems such as the Robot Technology Experiment (ROTEX) [14], a German experiment flown by NASA, and Engineering Test Satellite (ETS-VII) [15], flown by the Japanese Aerospace Exploration Agency (JAXA) (formerly National Space Development Agency of Japan (NASDA)). The ROTEX was a robotic arm that flew in 1993 on Columbia as part of STS-55, and successfully completed multiple tasks that include replacement of a simulated Orbital Replacement Unit (ORU) and capture of a free-flying object via on-board and ground teleoperation and autonomous scripts. By accomplishing tasks from autonomous scripts during the experiment, ROTEX became the first autonomous space robotic system. The ETS-VII mission was composed of a resident space object (RSO), Orihime, and a chaser satellite, Hikoboshi, with a robotic manipulator arm. ETS-VII successfully demonstrated cooperative control of a robotic arm and satellite attitude, and simple examples of visual inspection, equipment exchange, refueling, and handling of a satellite.

The state of the art in assembly robotics, implemented via experimental test beds only, is the Skyworker robot at Carnegie Mellon University [16], NASA’s Robonaut [17], and Ranger at the University of Maryland [18]. Skyworker is an 11 DOF robot that walks across the structure it is assembling to mate new components to the existing structure. The current prototype allows for high-level command inputs that are then parsed and implemented on-board as motion control commands. Robonaut is a collaborative effort

between DARPA and NASA aimed at developing a humanoid robot capable of meeting the increasing requirements for extravehicular activity (EVA). Robonaut, composed of two dexterous arms and two five-fingered hands with teleoperational and autonomous capabilities, has already demonstrated assembly of complicated EVA electrical connectors and delicate capabilities such as soldering. Ranger is a teleoperated robot at the University of Maryland that completes assembly, maintenance, and human EVA assistance tasks in a neutral buoyancy tank. Ranger has demonstrated robotic replacement of an Orbital Replacement Unit (ORU), complete end-to-end electrical connector mate/de-mate, and two-arm coordinated control.

According to Space Robotics Technology Assessment Report, the expectation for in-space assembly under nominal research efforts is: “Robots that can autonomously mate components and do fine assembly, including making connections under careful human supervision.” It is perceived, given the current state of the art of robots like Robonaut, that robots will possess the mechanical capabilities equivalent to a space suited human, but barring a breakthrough in communications architectures, more aggressive expectation of space robotic assembly cannot be met for teleoperation, due to current low bandwidth/high latency communication. On the other hand, automating Robonaut and other highly dexterous robots is possible, but to make highly dexterous robots effective under autonomous operation, a system level design needs to be considered that designs the small components specifically for assembly by autonomous robotic systems. This requires significant redesign in current infrastructure, which was perceived to be financially impractical at the time of the Space Robotics Technology Assessment Report. However, the new space exploration mandate outlines strategic allocation of funding to achieve its aggressive goal set, and as a result it now may be possible to alter the current infrastructure with the intent to simplify automated robotic assembly.

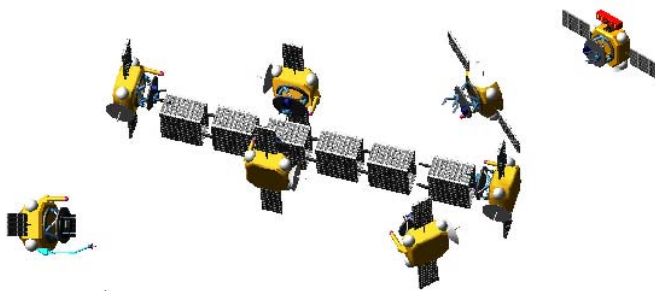


Fig. 2. Artist's depiction of the HEROS architecture

B. Maintenance

In-space maintenance consists of examining space structures for anomaly detection. The robots performing the inspection can be either free-flyers or manipulators capable of performing the subtasks of moving, such as navigation and obstacle avoidance, to examine the entire exterior of the

structure and anomaly detection via sensor interpretation. Currently, there are no operational space maintenance systems. However, there have been on-orbit investigations of subsystems, such as AERCam Sprint [19], which flew aboard Columbia (STS-87) in 1997. AERCam is a teleoperated EVA camera whose purpose was to merely investigate the feasibility of an autonomous, free-flying EVA camera system.

The state of the art in the area of inspection is the mini AERCam [20], Orbiter Boom Sensor System, OBSS [21], and the Supplemental Camera and Maneuvering Platform, SCAMP [22], at the University of Maryland. The mini AERCam project is a second generation version of the Sprint aimed at adding more complex capabilities while reducing the overall size of the prototype. A nanosatellite-class spherical free-flyer, the mini AERCam is only 7.5 inches in diameter and weighs a mere 10 pounds. Even though mini AERCam cutting edge hardware is not flight certified, making it far from implementable, the results obtained from orbital simulations and 5 DOF experimental test bed validations are invaluable to advancing the field of autonomous inspection agents.

The Orbiter Boom Sensor System (OBSS) is a manipulator based concept for inspection of the thermal protection surface (TPS) of the Shuttle. Consisting of a camera and a laser range sensor mounted on a 50' boom, the OBSS attaches directly to the SRMS, providing teleoperational inspection of nearly 75% of the Shuttle's TPS. In the Space Systems Laboratory at the University of Maryland, the Supplemental Camera and Maneuvering Platform (SCAMP) allows for investigations in free-flying camera applications. Operated in a neutral buoyancy tank, SCAMP provides near micro-gravity conditions for research into free-flying applications. SCAMP has demonstrated effective stereo video data interface and 3D navigation in the neutral buoyancy test bed.

For in-space inspection, NASA perceives the expectation for the next decade to be summarized as “autonomous robotic inspection of some of the exterior surfaces with sensory data filtered for potential anomaly before being stored or sent.” With more aggressive research effort, autonomous inspection and anomaly detection of most exterior surfaces of a target are realizable.

IV. OOS CONCEPTS

Several autonomous robotic servicers were/are under development. The robotic servicing spacecraft and concepts in the present work are dichotomized into two distinct classes. The first class will deal with single platform servicers, which are single spacecraft with *all* the individual subsystems required to perform servicing. The second class of robotic servicers will be fractionated space architectures that use a modular approach to perform the servicing subtasks.

A. Single Platform Servicing Architectures

Orbital Sciences Corporation received sponsorship from

NASA in 2001 to design, built, and test the Demonstration of Autonomous Rendezvous Technology (DART) [23], [24]. DART initially scheduled for flight in the fall 2004, was to be the first to locate and rendezvous with a satellite completely autonomously. Previously, astronauts had to control the vehicle via teleoperation in order to accomplish any rendezvous and servicing operations. After several delays DART was successfully launched on April 15, 2005 using an Orbital Sciences Pegasus Launch Vehicle and was scheduled to rendezvous and perform close proximity operations, “including station keeping, docking axis approach, circumnavigation, and a collision avoidance maneuver” [23, p. 1], via Advanced Video Guidance Sensor (AVGS). As reported by Spaceflight Now [25], DART suffered from problems with its guidance system from the start and, coming within 300 m of the target satellite, ran out of fuel, causing the autopilot to initiate the retirement segment of the mission. It was later reported by Space News [26] that DART has actually advanced further than originally thought, running into the target satellite and then maneuvering into the retirement orbit. The mission was partially successful, and a board to investigate the mishap was formed.

Experimental Small Satellite-10 (XSS-10) [27], [28] developed by the U.S. Air Force to evaluate future applications of micro-satellite technologies such as rendezvous, inspection, docking, and close-proximity maneuvering around orbiting satellites. Launched on January 29, 2003, the space robotics mission was pronounced a success. The flight experiment verified semiautonomous on-orbit rendezvous and inspection capabilities. The XSS-10 was the first demonstration of an autonomous inspection of another resident space object using a highly maneuverable micro-satellite. The flight experiment validated the design and operations of the micro-satellite’s autonomous operations algorithms, the integrated optical camera, and the star sensor design. The XSS-10 program team also verified the critical station keeping, maneuvering control, and logic guidance and control software necessary for autonomous navigation. The ground control capability, innovatively developed for XSS-10, enabled a small team to successfully interpret the real-time data and control of the spacecraft during its short mission [28].

Experimental Small Satellite-11 (XSS-11) [29], [30] managed by the U. S. Air Force Research Laboratory program and build by Lockheed Martin Co. is another mission that was successfully launched on April 11, 2005. XSS-11 is testing the autonomous technologies needed for the inspection and repair of the disabled satellites, such as approach and rendezvous maneuvers to several non-operational US satellites. XSS-11 will also “demonstrate technologies for military space surveillance” [30, p. 1]. The mission is scheduled to last approximately a year, with the rendezvous stage set to begin approximately six weeks after launch. Additional details of the XSS-11 mission are not readily available in the public domain.

Another on-going space robotics project is TEChnology

Satellite for demonstration and verification of Space systems (TECSAS) [31], [32] by the European Aeronautic Defense and Space Company (EADS), Babakin Space Center, and DLRRM. The mission consists of launching target and chaser satellites, the former equipped with a robotic arm and a docking mechanism, to verify robot’s capabilities for rendezvous and close-proximity operations on-orbit.

Also, the Phantom Works division of Boeing was selected to compete second phase of the Orbital Express [33], [34] project. Phase II consists of “finalize the design, develop and fabricate a prototype servicing satellite, the Autonomous Space Transport Robotic Operations satellite (ASTRO), and a surrogate serviceable satellite, NextSat, and conduct an on-orbit demonstration to validate the technical feasibility and mission utility of autonomous, robotic on-orbit satellite servicing” [33, p. 1]. It is also a goal to develop a standard upgradeable vehicle that would be able to be used for a variety of satellite servicing missions. The mission is set to launch in September of 2006.

Another space robotics project is Spacecraft for the Universal Modification of Orbits (SUMO) [35], sponsored by the Defense Advanced Research Projects Agency (DARPA) and implemented by the Naval Center for Space Technology. The servicing spacecraft is going to demonstrate “the integration of machine vision, robotics, mechanisms, and autonomous control algorithms to accomplish autonomous rendezvous and grapple of a variety of interfaces traceable to future spacecraft servicing operations” [35, p. 1]. A demonstration of the prototype was performed in December, 2005, while the launch of the SUMO spacecraft is set to occur sometime in 2008.

B. Fractionated Servicing Architectures

While the approach of using single monolithic service platforms may have its advantages, it severely lacks the responsiveness to address a wide class of on-orbit servicing missions. On the one hand, such an approach results in the development of complex spacecraft that are concisely surmised by the old adage “Jack of all trades, master of none.” On the other hand, utilization of a fractionated space architecture [36]-[38], where the architecture is decomposed into distinct modules that once “assembled” on-orbit possess the same functionality as the monolithic platforms of the previous section, creates a more responsive space environment.

The Heterogeneous Expert Robots for On-orbit Servicing (HEROS) architecture being developed at the University of Florida is an example of a fractionated space system (see Fig. 2). HEROS utilizes a fleet of smaller, adroit (possibly micro-satellite class vehicles) “expert” service platforms, where the expertise of the platform is determined by its subsystem functionalities, working symbiotically to achieve the servicing mission. As a result no single platform is required to perform all servicing tasks and thus, the individual platforms can be made less complex and more robust. Additional service robustness can be derived due to the

plug-n-play nature of fractionated systems from the ability to reconfigure the team based on the availability of service platforms; e.g., should a service platform fail, a replacement platform can be summoned. Of course, the cooperation between the servicing platforms increases the complexity of on-board computations, the inter-vehicle communication, and the dynamic interactions between the vehicles. Fig. 4 illustrates the hierarchical logic structure need for this type of approach.

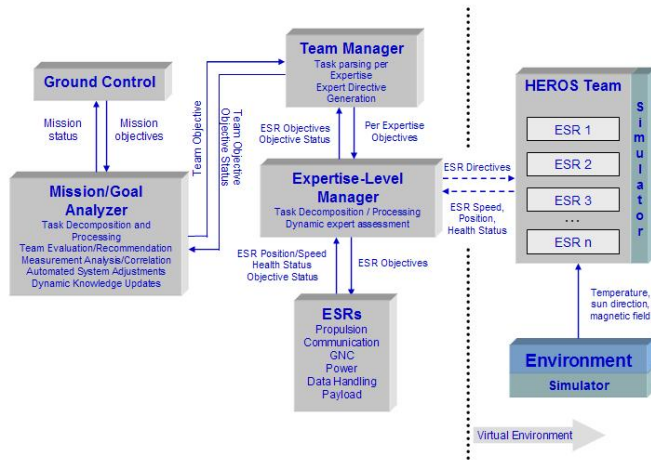


Fig. 3. HEROS organizational structure

The HEROS architecture utilizes this concept of strategic alliance to trade system robustness from within each spacecraft to a distribution across spacecraft platforms. For example, Fig. 3 shows a system that requires functionalities F_1 through F_n . In a conventional approach, the system would be constructed with redundant components and cross-strapped as shown in the figure. Such systems tend to be rather costly and thus are typically only produced on a limited basis (demonstrated in the figure with only two spacecraft). The HEROS approach, on the other hand, recognizes that not all functionalities are simultaneously required, thus vehicles are constructed with specialized capabilities. This simplifies the design of each vehicle and thus allows multiple copies of the vehicle to be constructed as shown in Fig. 3. Redundancy in this is accomplished through the availability of multiple vehicles as opposed to within the construction of a single vehicle. Furthermore, the HEROS architecture is more amenable to continuous system upgrades since only vehicles with the functionality to be upgraded are affected.

V. CONCLUSIONS

To date a variety of systems have been proposed and analyzed, yet most of the recent progress is either conceptual or awaiting a flight demonstration, with the exception of XSS-10, XSS-11, and ETS-VII. A robotic servicing architecture has been shown to be a feasible and profitable venture by recent analysis. Furthermore, the addition of

in-space serviceability has the potential to transform current space systems development and operations from slow and costly into responsive and cost efficient. Such transformation would enable the morphing of spacecraft into tactical assets capable of dealing with the ever growing uncertainties of space utilization.

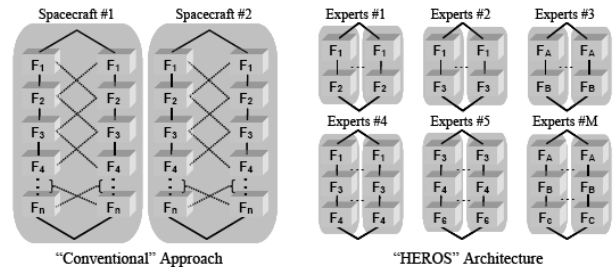


Fig. 4. Comparison of margins and redundancy of conventional and HEROS architectures

ACKNOWLEDGEMENT

This work was partially supported by DARPA Grant #00056302, and NASA CUIP.

REFERENCES

- [1] Skylab, May 15, 2006 (last accessed).
- [2] OMV, May 15, 2006 (last accessed).
- [3] L. Pedersen, D. Kortenkamp, D. Wettergreen, and I. Nourbakhsh, editors, NASA Exploration Team (NEXT), Space Robotics Technology Assessment Report, December 2002.
- [4] B.R. Sullivan, and D.L. Akin, "A Survey of Serviceable Spacecraft Failures," AIAA Space 2001 Conference and Exposition, Albuquerque, August 2001.
- [5] N. D. Davinic, S. Chappie, A. Arkus, and J. Greenberg, "Spacecraft Modular Architecture Design Study: Cost Benefit Analysis of On-Orbit Satellite Servicing," International Academy of Astronautics, IAA Paper 97-1.4.07, Oct. 1997.
- [6] C.M. Reynerson, "Spacecraft Modular Architecture for On-Orbit Servicing," AIAA Space Technology Conference and Exposition, Albuquerque, NM, Sept. 28-30, 1999 AIAA-1999-4473.
- [7] G. Leisman, A. Wallen, S. Kramer, and W. Murdock, "Analysis and Preliminary Design of On-Orbit Servicing Architectures for the GPS Constellation," AIAA Paper 99-4425, Sept. 1999.
- [8] J. Kreisel, "On-Orbit Servicing (OOS): Issues & Commercial Implications," PDF
- [9] J. Saleh, E. Lamassoure, and D. Hastings, and D. Newman, "Flexibility and the Value of On-Orbit Servicing: A New Customer-Centric Perspective," Proceeding of the 2001 Core Technologies for Space Systems Conference.
- [10] J. Saleh, E. Lamassoure, and D. Hastings, "Space Systems Flexibility Provided by On-Orbit Servicing: Part 1," Journal of Spacecraft and Rockets, Vol. 39, No. 4, July-August 2002, pp. 551-560.
- [11] J. Saleh, E. Lamassoure, and D. Hastings, "Space Systems Flexibility Provided by On-Orbit Servicing: Part 2," Journal of Spacecraft and Rockets, Vol. 39, No. 4, July-August 2002, pp. 561-570.
- [12] Madison, R.W., "Micro-satellite Based, On-orbit Servicing work at Air Force Research Laboratory," Kirtland AFB, NM: Air Force Research Laboratory.

- [13] Waltz, D. M., "On-Orbit Servicing of Space Systems," Krieger, Malabar, FL, 1993.
- [14] ROTEX, January 11, 2005 (last accessed).
- [15] O. Mitsushige, "ETS-VII: Achievements, Troubles and Future," Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), 2001.
- [16] P. Staritz, S. Skaff, C. Urmsen, and W.L. Whittaker, "Skyworker: a robot for assembly, inspection and maintenance of large scale orbital facilities," Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA '01), Vol. 4, May, 2001, pp. 4180-4185.
- [17] R. Necessary, Curator, Robonaut, December 5, 2005 (last accessed).
- [18] Space Systems Laboratory, University of Maryland, "Dexterous Robotics at the Space Systems Laboratory," Ranger, December 5, 2005 (last accessed).
- [19] AERCam Sprint, December 5, 2005 (last accessed).
- [20] Mini AERCam, December 5, 2005 (last accessed).
- [21] NASA's Implementation Plan for Space Shuttle Return to Flight and Beyond PDF, Vol. 1, Revision 2, April 26, 2004, pp. 1-21 - 1-31.
- [22] Space Systems Laboratory, University of Maryland, "The Supplemental Camera and Maneuvering Platform (SCAMP)," SCAMP, December 5, 2005 (last accessed).
- [23] Orbital Sciences Co., "Fact Sheet. DART - Demonstration of Autonomous Rendezvous Technology," DART posted June 2004, accessed June 2004.
- [24] NASA, "Demonstration of Autonomous Rendezvous Technology," Press Kit, October 2004.
- [25] J. Ray, "NASA autopilot test suffers crippling flaw," Ray posted April 2005, accessed June 2005.
- [26] B. Berger, "Fender Bender: NASA's DART Spacecraft Bumped Into Target Satellite," Berger posted 2005, accessed June 2005.
- [27] T. Davis, "XSS-10 Microsatellite Flight Demonstration Program," Davis accessed July 2005.
- [28] J. Banke, "Air Force XSS-10 Micro-Satellite Mission a Success," Banke posted January 2003, accessed July 2005.
- [29] R. Partch, "AFRL-003 - Experimental Satellite System (XSS-11)," November 2002.
- [30] J. Singer, and J. Bates, "U.S. Air Force, Critic Differ on XSS-11 Mission Objective," Singer posted April 2005, accessed June 2005.
- [31] K. Landzettel, B. Brunner, R. Lampariello, C. Preusche, D. Reintsema, and G. Hirzinger, "System Prerequisites and Operational Modes for On Orbit Servicing," International Symposium on Space Technology and Science, Miyazaki, Japan, May 30 - June 6, 2004.
- [32] G. Hirzinger, K. Landzettel, B. Brunner, M. Fischer, C. Preusche, D. Reintsema, A. Albu-Schaffer, G. Schreiber, and B-M. Steinmetz, "DLR's Robotics Technologies for On-Orbit Servicing," submitted to Advanced Robotics, Special Issue on "Service Robots in Space".
- [33] "DARPA Awards Orbital Express Demonstration to Boeing," Article posted March 2002, accessed June 2005.
- [34] "Orbital Express Space Operations Architecture," Article updated May 2005, accessed June 2005.
- [35] A.B. Bosse, W.J. Barnds, M.A. Brown, N.G. Creamer, A. Feerst, C. G. Henshaw, A.S. Hope, B.E. Kelm, P.A. Klein, F. Pipitone, B.E. Plourde, and B.P. Whalen, "SUMO: Spacecraft for Universal Modifications of Orbits," Proceedings of SPIE Defense and Security Symposium, Vol. 5419, April 2004.
- [36] O. Brown and P. Eremenko, "Fractionated Space Architectures: A Vision for Responsive Space," 4th Responsive Space Conference, RS4-2006-1002, Los Angeles, CA, April 24-27, 2006.
- [37] A. Tatsch, N. Fitz-Coy, and W. Edmonson, "Heterogeneous Expert Robots for On-Orbit Servicing (HEROS): A Robust and Sustainable Architecture for On-Orbit Servicing," NASA 2005 Flight Mechanics Symposium, Greenbelt, Maryland, October 18 - 20, 2005.
- [38] A. Tatsch, N. Fitz-Coy, and W. Edmonson, "Heterogeneous Expert Robots for On-Orbit Servicing: A New Paradigm," Infotech@Aerospace Conference, Arlington, Virginia, September 26 - 29, 2005.