Due date of deliverable: 30 April 2008

Actual submission date: 2 June 2008

Start date of the project: 1 May 2004

Organisation name of lead contractor for this deliverable: UJI

Revision: Revised version, June 2008

Dissemination level: PU
This report is the result of a number of workshops, meetings —formal and informal—, discussions —physical or email-based—, and information gathering, either obtained from different sources or kindly provided by a number of persons. The following list contains the names of those who —in one way or another— have contributed to its content along the last four years. Apologies are due to those who may feel their names are missing.

Diego Alonso
Nicolas Andreff
Benjamín Balaguer
Stephen Balakirsky
Sven Behnke
Antonio Bicchi
Andreas Birk
Andrea Bonarini
Fabio Bonsignorio
Herman Bruyninckx
Wolfram Burgard
Gilles Caprari
Stefano Carpin
Stefano Caselli
Enric Cervera
François Chaumette
Henrik Christensen
Joris de Schutter
Rüdiger Dillmann
Patrick Doherty
Etienne Dombre
Giulio Fontana
Roland Geraerts
Gerhard Grunwald
Jim P. Gunderson
Louise F. Gunderson
Martin Häggele
John Hallam
Patric Jensfelt
Dirk Kraft
Danica Kragic
Kostas Kyriakopoulos
Ville Kyrki
Christian Laugier
Steven LaValle

Frank Lingelbach
Ingo Lütkebohle
Ezio Malis
Raul Marín
Ali Marjovi
Lino Marques
Philippe Martinet
Mateo Matteucci
Gerard McKee
Claudio Melchiorri
Olivier Michel
Javier Mínguez
Rezia Molina
Antonio Morales
Ami Moshaiov
José Neira
Andreas Nuchter
Francisco Ortiz
Mark Overmars
Enrico Pagello
Michael Pardowitz
Kai Pfeiffer
Max Pfingsthorn
Mila Popovic
Domenico Prattichizzo
Ignacio Raño
Monica Reggiani
Pedro J. Sanz
Klaus Schilling
Roland Siegwart
Domenico Sorrenti
Stefano Stramigioli
Juan D. Tardos
Oliver Wulf
Cezary Zielinski

Also, the technical collaboration of Juan Carlos García, Patricio Nebot, Pablo Domínguez and Ester Martínez is greatly appreciated.

Angel Pasqual del Pobil
May 29, 2008.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction: Benchmarks in Robotics Research</td>
<td>5</td>
</tr>
<tr>
<td>Organized Workshops</td>
<td>8</td>
</tr>
<tr>
<td>EURON’06 Workshop on Benchmarks in Robotics Research</td>
<td>8</td>
</tr>
<tr>
<td>IROS’06 Workshop on Benchmarks in Robotics Research</td>
<td>12</td>
</tr>
<tr>
<td>EURON’07 Workshop on Experimental Practices and Benchmarking</td>
<td>15</td>
</tr>
<tr>
<td>IROS’07 Workshop on Performance Evaluation and Benchmarking for</td>
<td>17</td>
</tr>
<tr>
<td>Intelligent Robots</td>
<td></td>
</tr>
<tr>
<td>GEMBENCH’08 Open Forum on Good Experimental Methodology and Benchmarking</td>
<td>21</td>
</tr>
<tr>
<td>CogGEMBench’08 Workshop on Good Experimental Methodology &amp; Benchmarks in Cognitive Robotics</td>
<td>23</td>
</tr>
<tr>
<td>RSS’08 Workshop on Experimental Methodology and Benchmarking in Robotics</td>
<td>25</td>
</tr>
<tr>
<td>IROS’08 Workshop on Performance Evaluation and Benchmarking for</td>
<td>27</td>
</tr>
<tr>
<td>Intelligent Robots</td>
<td></td>
</tr>
<tr>
<td>Robot competitions and challenges</td>
<td>29</td>
</tr>
<tr>
<td>ROBOCUP SOCCER</td>
<td>29</td>
</tr>
<tr>
<td>ROBOCUP RESCUE</td>
<td>42</td>
</tr>
<tr>
<td>ROBOCUP@HOME</td>
<td>45</td>
</tr>
<tr>
<td>ROBOCUP JUNIOR</td>
<td>46</td>
</tr>
<tr>
<td>FIRA</td>
<td>47</td>
</tr>
<tr>
<td>DARPA Grand Challenge</td>
<td>57</td>
</tr>
<tr>
<td>DARPA Urban Challenge</td>
<td>61</td>
</tr>
<tr>
<td>EUROBOT</td>
<td>63</td>
</tr>
<tr>
<td>First International Cleaning Robot Contest</td>
<td>67</td>
</tr>
<tr>
<td>ELROB: European Land-ROBot trial</td>
<td>70</td>
</tr>
<tr>
<td>ICRA Robot Challenge</td>
<td>73</td>
</tr>
<tr>
<td>Benchmarking Initiatives in Europe</td>
<td>78</td>
</tr>
<tr>
<td>Motion planning</td>
<td>78</td>
</tr>
<tr>
<td>Manipulation and grasping</td>
<td>82</td>
</tr>
<tr>
<td>Visual servoing</td>
<td>84</td>
</tr>
<tr>
<td>Mobile robotics</td>
<td>88</td>
</tr>
<tr>
<td>Cognitive robotics</td>
<td>90</td>
</tr>
<tr>
<td>RAWSEEDS</td>
<td>92</td>
</tr>
<tr>
<td>RoSta</td>
<td>95</td>
</tr>
<tr>
<td>Experimental methodology in robotics research</td>
<td>97</td>
</tr>
<tr>
<td>Image processing and computer vision benchmarking in robotics</td>
<td>98</td>
</tr>
<tr>
<td>Networked robotics</td>
<td>99</td>
</tr>
<tr>
<td>European robotics research platforms</td>
<td>100</td>
</tr>
<tr>
<td>Related Workshops and Initiatives</td>
<td>101</td>
</tr>
<tr>
<td>IARP/IEEE/EURON Workshop on Dependable Robots in Human Environments</td>
<td>101</td>
</tr>
<tr>
<td>PerMIS: Performance Metrics for Intelligent Systems</td>
<td>103</td>
</tr>
<tr>
<td>Software Development in Robotics</td>
<td>105</td>
</tr>
<tr>
<td>Radish: The Robotics Data Set Repository</td>
<td>107</td>
</tr>
<tr>
<td>OpenSLAM: Open-source SLAM packages</td>
<td>109</td>
</tr>
<tr>
<td>References</td>
<td>....................................................................................................................</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Appendices</td>
<td>....................................................................................................................</td>
</tr>
<tr>
<td>Appendix I: Soccer competitions rules</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Appendix II: Robocup rescue general rules</td>
<td>.....................................................................</td>
</tr>
<tr>
<td>Appendix III: Hurosot challenges</td>
<td>..................................................................................</td>
</tr>
<tr>
<td>Appendix IV: DARPA Grand Challenge</td>
<td>...........................................................................</td>
</tr>
<tr>
<td>Appendix V: PerMIS’04 White Paper</td>
<td>.............................................................................</td>
</tr>
<tr>
<td>Appendix VI: Motion Planning Benchmarks</td>
<td>.......................................................................</td>
</tr>
<tr>
<td>Appendix VII: Models for Motion Planning</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Appendix VIII: Visual Servoing Benchmarks with the ACT</td>
<td>........................................</td>
</tr>
<tr>
<td>Appendix IX: Visual Servoing Simulation Experiments</td>
<td>........................................</td>
</tr>
<tr>
<td>Appendix X: Visually-Guided 3-Finger Grasping Experiments</td>
<td>........................................</td>
</tr>
<tr>
<td>Appendix XI: General Guidelines for Robotics Papers Involving Experiments</td>
<td>........................................</td>
</tr>
<tr>
<td>Appendix XII: Proceedings IROS’07 Workshop on Benchmarking for Intelligent Robots</td>
<td>..........................................................</td>
</tr>
</tbody>
</table>
INTRODUCTION
BENCHMARKS IN ROBOTICS RESEARCH

It was a well-known fact that the current practice of publishing research results in robotics made it extremely difficult not only to compare results of different approaches, but also to assess the quality of the research presented by the authors. Though for pure theoretical articles this may not be the case, typically when researchers claim that their particular algorithm or system is capable of achieving some performance, those claims are intrinsically unverifiable, either because it is their unique system or just because a lack of experimental details, including working hypothesis. Often papers published in robotics journals and generally considered as good would not meet the minimum requirements in domains in which good practice calls for the inclusion of a detailed section describing the materials and experimental methods that support the authors’ claims. This is, of course, partly due to the very nature of robotics research: reported results are tested by solving a limited set of specific examples on different types of scenarios, using different underlying software libraries, incompatible problem representations, and implemented by different people using different hardware, including computers, sensors, arms, grippers...

This state of affairs cannot be changed in the short term, but in the last four years some steps have been taken in the right direction by studying the ways in which research results in robotics can be better reported, assessed and compared. In this context, EURON has played an important role by fostering systematic benchmarking and good experimental practice in robotics research. The long-term benefits of these efforts are evident: not only will they foster the overall quality of research results but they will also improve publication opportunities for EU-based research, thereby increasing international visibility of European research and lead to rapid adoption of new research results by application developers and the robotics industry.

Curiously enough, the above described situation was compatible with the fact that some of the most popular organized events in robotics are related to comparative research: the different successful robot competitions that have been organized in the last years are a way of comparing the performance of the competing systems by means of very well-defined rules and metrics. The organization of these scientific competitions has proven a quick way to attract substantial research efforts and rapid to produce high-quality working solutions.

When considering the role of EURON in relation to these issues, trying to set up a task force to define a set of gold standards in robotics by itself was not considered as a feasible approach given the limited available resources. To mention just a well-known example: DARPA and NSF funded a study about a very particular field in robotics, namely human robot interaction (HRI). Even for this reduced field over sixty representatives from academia, government and industry participated in the study, and one of the recommendations regarding actions for the next 5 years concluded that the HRI field is still too new to set milestones or benchmarks [Burke at al. 04]. Even so, some grand challenges were proposed. Grand challenges are interesting as long-term goals, but they are usually vaguely described, resulting from a roadmap in the field, and not very useful for measuring progress or comparing results. Nevertheless, benchmarks could be conceived as a way of measuring progress toward a grand challenge.
Defining a benchmark —even a sound valid benchmark— could be an easy task, if it is just taken as an academic exercise. Defining a successful benchmark is something completely different. A benchmark can be considered as successful if the community extensively uses it in publications, conferences and reports as a way of measuring and comparing results. To put it in a few words: a benchmark is successful if and only if it is widely accepted by the community at which it is targeted.

This kind of success is somehow difficult to predict, but some the following considerations may help:

- its success is related to its quality, i.e. it really must serve its purpose.
- robotics is a too broad field and defining benchmarks for robotics in general does not seem to make sense, but rather benchmarks should focus on particular sub-domains, such as visual servoing, grasping, motion planning, ...
- it is easier that a benchmark is successful within a scientific (sub) community if it arises from the community itself instead of having it defined outside this community.

Reaching consensus does take time: proposing good benchmarks for the community to accept, is a long process that requires the concurse of many people in many subcommunities within robotics. Consequently, the role of EURON in this context was not that of defining benchmarks, but rather to propose and encourage:

- by convening the different sub-communities within robotics so that they become aware of the problem, and encourage them to make efforts to define these benchmarks, follow their work, help in disseminating their proposals so that they become publicly known.
- by helping in providing the required information about other efforts in the field, organizing workshops and meetings, coordinating discussions, etc

Since this work has been carried out in the framework of FET Beyond Robotics and IST Cognition Unit, we have been more concerned with non-industrial scenarios. This report builds on previous work developed in EURON I [Dillamnn 04] in which benchmarks in industry were discussed. In general this seems to be a different situation, since industry can provide the resources to measure whatever features they desire in a robot. In this sense they can not only develop their own benchmarks, but also they have even organized competitions: a famous example is the one held in March 1996, when Ford U.S.A. organized a competition for an order of 400 welding robots with the result that the KUKA robots could solve the benchmark problems considerably faster and smoother than the robots of the main competitor. KUKA won this contract and since then all Ford European plants become equipped with KUKA robots exclusively [Ford Competition].

In order to attain the above-mentioned goals, namely successful robotics benchmarks in the medium term, the following actions were identified:

a) Exhaustive survey: The first step was to make accessible to the community the state of the art in existing efforts in comparative research such as related initiatives in the U.S. and Japan, competitions, benchmarks, challenges, and conferences with relevant topics. Thus, information that was previously unavailable, or scattered in different sources and merged with irrelevant issues is make accessible to the community including a detailed description of the rules, metrics and procedures.
b) Increase awareness by organizing and participating in workshops, meetings, and discussions; encourage experts in the different subfields to get involved in the process of benchmark definition, and follow these efforts suggesting further actions.

c) Promote a set of benchmarking initiatives in some areas. These sub-areas are selected in different ways: as suggested by the research roadmap, by taking advantage of existing Special Interest Groups in EURON, or when experts propose benchmarking initiatives. These benchmarks should be operational and described in sufficient detail, as well as their associated metrics, and an independent measurement procedure, keeping always in mind that the fundamental goal is that they are widely accepted in the community at large.

More concretely, we planned a series of discussions and refinements in parallel actions, similar to the procedure described in [Burke et al. 04] as a continuous process of convergence towards consensus in order to ensure community wide acceptance:

- Identification of benchmarking initiatives.
- Establish working groups.
- Discussion: online, and physical —ad-hoc or taking advantage of other events.
- Dissemination of on-going initiatives through a report and the web site.
- Workshop on benchmarks at international events (IROS’06, IROS’07, RSS’08, IROS’08).

This report describes the up-to-date results of action (a) above in the form of the final version of an exhaustive, detailed survey and inventory of current existing efforts in comparative research: competitions, benchmarks, challenges, repositories, conferences, etc. This is the result of an on-going long process of information gathering, either obtained from different sources or kindly provided by a number of persons. Most of this information was previously unavailable, scattered in different sources and merged with irrelevant issues. After a process of selection and rewriting it is regularly updated and made available to the community in a web site, accessible through EURON main web site and also at http://www.robot.uji.es/benchmarks. Some technical material available in this website has also been included as appendices in this report for its interest for future benchmark development, since they include detailed description of the rules, metrics and procedures for current robot competitions and benchmarks.

Actions (b) and (c) have been addressed with a number of meetings, workshops, and discussions —physical or email-based. The main overall result is a considerable increase in the awareness of the importance of robotics benchmarking in Europe. This has resulted in a number of on-going initiatives in Europe towards defining benchmarking scenarios, the current results of which are included in this report. The degree of accomplished tasks varies among the different initiatives, some have been already available to the community for months on end, whereas others are more embryonic as described in the corresponding sections. Some of them are based on simulations only, and data sets are made available defining objects, robots and scenarios in standard format descriptions. When moving ahead beyond simulations into real hardware in the real world, computer data sets are not enough and various solutions are put forward. Some of them are based on specific hardware that is shared by remote access, whereas others describe experimental protocols to be shared in the verification of diverse approaches to the same problem.
EURON’06 Workshop on Benchmarks in Robotics Research

This was the main joint event in 2006 devoted to benchmarking in Europe. It was held in Palermo on March 18, 2006, taking advantage of the First European Robotics Symposium (EUROS) and the Annual EURON Meeting. It was well attended with over 25 participants.

The main purpose of the workshop was to provide an informal forum for participants to exchange their ideas about benchmarking in robotics, as well as presenting and discussing ongoing relevant initiatives.

The workshop was intended as a working meeting with the following structure:

- Series of a short talks
- Discussion after each talk
- Final discussion

Summary of talks

Speaker: Angel P. Del Pobil
Title: The need of Benchmarks in Robotics Research
Abstract:

Current practice of publishing research results in robotics makes it extremely difficult not only to compare results of different approaches, but also to assess the quality of the research presented by the authors. This state of affairs cannot be changed in the short term, but some steps can be taken in the right direction by studying the ways in which research results in robotics can be assessed and compared.

Speaker: Javier Minguez, University of Zaragoza, Spain
Title: Metrics to evaluate motion techniques from the resulting motion point of view (mobile robotics perspective)
Abstract:

Historically, many techniques have been designed to address autonomous collision-free motion (sensor-based motion with obstacle avoidance). For example (Khatib, O., 1986), (Borenstein, J. & Koren, Y., 1991), (Fox, D.; Burgard, W. & Thrun, S., 1997), (Simmons, R., 1996), (Fiorini, P. & Shiller, Z., 1998), (Minguez, J. & Montano, L., 2004) among many others. It is clear that under the same conditions each technique generates a different motion. Nevertheless, questions like: which is the most robust one? or which of them behaves better in a determined context or condition? They cannot be answered neither from a scientific nor technological point of view. In other words, once we face a mobile robotics application, selection of a motion technique among all the existing ones is a matter of specialists and not accessible to everybody. This is because there is no objective comparisons of methods neither quantitative (in terms robustness or action parameters of such as the time or the total distance travelled) nor qualitative (in terms of security of the motion). At present, there is only an experimental comparison (Manz, A.; Liscano, R. & Green, D., 1991). Nevertheless, this comparison is very old, and thus, it does not include the advances in this subject in the last 15 years. Furthermore it is based on the
observation and does not present a rigorous and objective methodology to address this objective. The scientific objective of this part is the development of a procedure to evaluate the motion techniques from a quantitative and/or qualitative perspective of the motion generated in different contexts.

Speaker: John Hallam, Syddansk Universitet, Denmark
Title: Experimental Robotics
Abstract:
The purpose of the talk was to promote good experimental methodology among roboticists. This would contribute to help pave the way for the widespread practice of benchmarking in robotics.

Speaker: Domenico G. Sorrenti, University of Milano-Bicocca, Italy
Title: Image processing and computer vision benchmarking in robotics
Abstract:
Computer vision (and image processing algorithms) is very important in robotics, as at the moment cameras seem both prospectively as low-cost as required and capable to supply rich data about the robot workspace. Therefore benchmarking of computer vision (and image processing algorithms) is important for roboticists; we think it is important to have a distributed awareness about the pros, cons and actual expectable capabilities of each algorithm in each situation. In some we probably have plenty of alternatives, in some just one and in other we have no viable algorithm for robotic use. The underlying issue is that generation of sensed data for our robots is done most of times by a processing pipeline of gigo type (garbage-in-garbage-out, at each step). We propose to prepare datasets, for relevant sub-domains, relevant algorithms, etc so that the community can ask new robotics algorithms, as described in the papers, to be validated with those datasets. In particular we are interested in omnidirectional vision and also in robustness to light changes, highlights, strong shadows, etc.

Speaker: Domenico Prattichizzo, Università di Siena - Italy
Title: Remote Visual Servoing Benchmarks Sharing the Same HW Setup
Abstract:
We describe current work aimed at the extension of our Automatic Control TeleLab so that it can be used as a tool for benchmarking visual servoing techniques by sharing remotely the same hardware set-up.

Speaker: Claudio Melchiorri, University of Genoa - Italy
Title: Benchmarks for Manipulation and Grasping
Abstract:
We present results of discussions and some proposals for benchmarking in the context of robot manipulation and grasping

Speaker: Javier Mínguez, University of Zaragoza, Spain
Title: Metrics to evaluate the motion techniques based on the biomedical response of the user (biomedical perspective)
Abstract:
In the healthcare or clinical fields, any proposed intervention (therapeutic or preventive) has to take into account the impact over the patient quality of life. This is not only to improve their health, but also the subjective perception of welfare. In this sense, the scenarios have to be conceived and validated explicitly in terms of compatibility with the users-patients welfare. Thus, for example, a motion technique can be robust and move the
vehicle with optimal trajectories in time, distance, etc. But when a robot guide or transports a human, a basic question is how does the user feel during the movement? In the development of healthcare technologies the user satisfaction is an essential specification. Therefore, it is required to measure and to evaluate systematically the quality of the user–device interaction and the reaction of the individual to a motion technique particularly. This fact has a vital importance in the growing demand of autonomous robots for human transportation, specially for those designed for aging users and/or that spend a great time using the vehicles. The objective is the development of a procedure to evaluate the reaction of the user to the motion generated by the vehicle. In biomedical sciences, one way to evaluate the human response to external events is to study the biomedical (physical and psychological) activity during the development of a task. Other methodologies include the filmed observation, speech techniques, etc. By using these methodologies, the objective is to develop a way to evaluate and thus to create a metric for the existing motion techniques obtained from the user reactions.

Speaker: Stefano Caselli, University of Parma, Italy.
Title: The case for repeated research in Robotics
Abstract:
The main point of my presentation is that repeated experimental research should be appreciated and properly rewarded. Among its benefits we have:

- Provide independent validation of published results
- Question and extend the range of tests performed
- Document and generalize research software
- Speed transfer into production systems
- Involve students in research

Discussions
The discussions focused on four axis:
1) Testbeds: some of the participants suggested to use EURON to have a repository similar to RADISH. Like this EURON will help to diffuse data sets that could be used for researchers to test their algorithms. Using the same data set would add homogeneity to the results, and thus would help to qualify and understand the benefits of a given published approach.
2) Benchmarks: it was acknowledge that an important initiative would be to create benchmarks for the different disciplines. This seemed to be a very difficult task that would need the cooperation of experts the areas to agree with them. It would be important to define why each benchmark is selected with rigorous criteria so that people understand the objective pursued. Otherwise the benchmark would not have credibility.
3) Method evaluations: some people proposed that methods could be evaluated by different working units that must to be centralized (EURON could coordinate this centralization). Like this, the units could define in each area which is the best protocol to evaluate the techniques. The result of this process should be tables of performance of techniques in different range conditions. This process centralized and well implemented would produce very good and homogeneous benchmarking in every discipline.
4) **Protocols for experimentation**: and interesting topic of discussion was the way that researches in robotics perform the experiments and the validity that they have. This is very important in other disciplines such as medicine and some ideas could be imported. It was suggested to create a protocol of “writing” the papers involving a given protocol to “experiment” with the robots. The idea was to implement this in the future papers trying to win homogeneity and rigorous validation processes. A starting point could be the next EUROS meeting.

**Conclusion**

Participants in general found the workshop very interesting in that it added significant insight to their perspective of benchmarking. The only thing that some people missed was the discussion in different working areas. That was planned but turned out to be impossible with an audience composed of many researches of very different topics. However, since small groups would generate very interesting discussions for interested and motivated people, who are the intended structure for more focused events.

Also it seems difficult to consider benchmarking in the same way for "low level" control issues and more "high level" planning and decision processes (as collision avoidance, vision, localization...). It was felt that implementation on a real experimental platform is required to compare real performances of "low level" control schemes (and not only Simulink simulation, even though obviously it gives significant information). But real implementation costs a lot in terms of equipments and time, and it requires very dedicated expertise. Another limitation is that it is robot dependant (frictions and other nonlinearities), thus in many cases the benchmark should be done on the same platform to get relevant and comparable results.
IROS’06 Workshop on Benchmarks in Robotics Research

This workshop was the second main meeting on benchmarks held in 2006. It was organized by A.P. del Pobil in conjunction with a major scientific conference—such as IROS’06—to maximize success and optimize the use of resources. It was a full day event that took place on October 10th, 2006. It was organized as a more formal event with a selection of full papers to guarantee the quality of the research. Eight papers were accepted and presented, authored by a total of 18 European researchers. Formal proceedings were issued and available as a 2007 deliverable document DR2.4.

The main purpose of this second workshop was to provide a forum for participants to exchange their on-going work and ideas about benchmarks in robotics research. It was intended as a forum for discussion, exchange of points of view, assessment of results and methods, and as a source of dissemination and promotion of benchmarks.

Summary of talks

Title: Why do we Need Benchmarks in Robotics Research?
Authors: Angel P. del Pobil
Speaker: Angel P. del Pobil
Abstract: This paper discusses the current state of affairs in measuring and comparing research results in robotics and describes current actions towards better benchmarking practice in the field.

Title: Robot Competitions - Ideal Benchmarks for Robotics
Authors: Sven Behnke
Speaker: Reimund Renner
Abstract: In this paper, I argue for the use of robotic competitions as benchmarks for robotics research. By providing a common task to be solved at a specific place and a specific time, competitions avoid some of the difficulties arising when evaluating robotics research in the own lab. Competitions also bring together multiple research groups working on the same problem. This fosters the exchange of ideas. I review two of the most popular robotics competitions, RoboCup and the DARPA Grand Challenge, and discuss some issues arising when designing robotics competitions.

Title: RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets
Authors: A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti and J. D. Tardos
Speaker: Juan Domingo Tardos
Abstract: The absence of standard benchmarks is an acknowledged problem in the field of robotics, and is doubly harmful to it. First, it prevents recognition of scientific and technical progress, thus discouraging research and development; second, it prevents new actors (and particularly companies) from entering the robotic sector, as heavy investments
are needed to compensate for that absence. The need for benchmarking in advanced robotics embraces a wide range of topics, from, e.g., dexterous manipulation to, e.g., emotional interfaces with humans. The RAWSEEDS project will focus on sensor fusion, localization, mapping and SLAM in autonomous mobile robotics. The project will provide a comprehensive Benchmarking Toolkit, including high-quality multisensorial data sets, well defined Benchmarking Problems (BPs) based on the data sets, state-of-the-art Benchmarking Solutions (BSs) in the form of algorithms, software, methodologies and instruments for the assessment of the BSs.

**Title:** Steps Towards the Automatic Evaluation of Robot Obstacle Avoidance Algorithms  
**Authors:** Ignacio Rañó and Javier Mínguez  
**Speaker:** Javier Mínguez  
**Abstract:**

This paper presents the first steps towards the evaluation of obstacle avoidance techniques for mobile robots. The idea is to create a methodology to evaluate the performance of the methods given a wide range of work conditions. The work conditions usually include scenarios with very different nature (dense, complex, cluttered, etc). The performance is measured in terms of robotic parameters (robustness, optimality, safety, etc). We describe in this paper the overall methodology that we intend to apply and the first steps in the scenario characterization.

**Title:** On Experimental Research in Sampling-based Motion Planning  
**Authors:** Roland Geraerts  
**Speaker:** Roland Geraerts  
**Abstract:**

Motion planning is one of the fundamental problems in robotics. The motion planning problem can be defined as finding a path between a start and goal placement of a robot in an environment with obstacles. Over the past fifteen years, many different researchers have studied sampling-based motion planning techniques such as the Probabilistic Roadmap Method (prm). This has led to many variants, each with its own merits. It is difficult to compare the different techniques because they were tested on different types of environments, using different underlying libraries, implemented by different people on different machines. We have provided a comparative study and analysis of a number of these techniques, all implemented in a single system and run on the same test environments and on the same computer. We encountered many difficulties and pitfalls during this study. We will identify them and discuss our solutions based on our experimental research over the past four years.

**Title:** Cross-Platform Software for Benchmarks on Visual Servoing  
**Authors:** Enric Cervera  
**Speaker:** Enric Cervera  
**Abstract:**

The field of visual servoing is now widely accepted as a modern, consolidated discipline for vision-based real-time robot control. Since experimental setups have become affordable, many results are published worldwide in conferences and journals, yet comparison among them becomes difficult due to the wide variety of systems and tasks. A need for a common framework arise, which should allow to compare control schemes, and provide a set of benchmarking tasks to the research community. This paper presents an object-oriented, cross-platform, network-ready environment for visual servoing simulation tasks. With flexibility and extensibility as the main design goals, tasks can be defined
either for cameras attached to moving Cartesian frames, or serial link manipulators. In order to provide fast feedback to the user, it includes real-time 3D rendering of the simulated scene. Task parameters and visual features can be freely chosen, and new features can be easily added to the framework. Output data is logged to disk files, which can be analyzed by any popular mathematical package. The simulator is built upon an agent-based framework, which makes possible the distribution across multiple networked computers. Moreover, it can be securely downloaded from a web server and automatically installed in a computer, running either Windows, Linux or Mac OS X operating systems, thus providing a set of common tools for a wide range of users and enabling the comparison and benchmarking or results. This simulator is now in alpha version, yet it has been extensively used by the students worldwide in an online course on visual servoing.

**Title:** Drawing a parallel: Benchmarking in Computer Vision and Robotics  
**Authors:** Danica Kragic, Ville Kyrki, Patric Jensfelt and Frank Lingelbach  
**Speaker:** Danica Kragic  
**Abstract:**  
In this paper, the current state of benchmarking and performance comparison in vision is reviewed, with the goal of discovering good benchmarking practices that could be also introduced in the robotics community. In short, what can researchers in robotics learn about benchmarking by taking a look at computer vision benchmarking procedures? An important common characteristic of both robotics and computer vision is that both are highly hardware and application dependent, and therefore many similar problems exist even though “pure” computer vision has still somewhat less variation. In both fields the tasks to be achieved are complex, making analytical performance prediction impossible in many cases thus leaving empirical study as the only available approach. For this reason, the test cases of the empirical studies, as well as the analysis of the results of experiments, are the most important considerations in benchmarking.

**Title:** Experiments in Visually-Guided 3-Finger Grasping  
**Authors:** Antonio Morales, Pedro J. Sanz and Angel P. del Pobil  
**Speaker:** Antonio Morales  
**Abstract:**  
We present an initiative that has been developed at Universitat Jaume I and offers a description of a set of experiments on visually-guided grasping of planar objects with a Barret hand. They are made available to the community as a set of standard experiments for defining benchmarks and associated performance metrics. An experimental protocol for a benchmark is proposed.
EURON’07 Workshop on Experimental Practices and Benchmarking

On March 28th, 2007 this workshop was held in the context of the 2007 EURON Annual Meeting in Chania (Greece). The workshop was co-organized by Angel P. del Pobil from Universitat Jaume I in Spain, John Hallam from the University of Southern Denmark, and Fabio Bonsignorio from Heron Robots (Italy).

After some preliminary discussions it was decided to focus the topic of the event on Experimental Practices and Benchmarking. The rationale was that a good definition for a benchmark is that it is an experiment with:

- an agreed interesting problem
- an agreed standard procedure, including agreement on:
  - choice of parameters
  - what to measure
  - how to measure it
  - how to analyse the results
- an agreed index of performance

Consequently, a good way to promote a culture for benchmarking in the research community is by promoting good experimental methodologies, since it was felt by many that a common body of knowledge concerning experimental methodology was missing in the robotics research community, i.e., a corpus of specifications on how to design and conduct good experiments in robotics. This fact is reflected in the frequent publication of scientific articles that report poor experimental work, few or no replications or tests, dubious comparison between algorithms, lack of suitable quantification of performance and its variability, and conclusions which, while may be correct, are unjustified by the reported experimental work. The goal of the workshop was to address this situation.

Program

The workshop was preceded by a plenary session on the same topic that served as an introduction. Since it was felt that the topic was not still mature enough, no formal presentations were held at the workshop, but instead it was arranged in a brainstorming style in which the main questions proposed in the plenary session were discussed in an informal atmosphere. The structure of the discussions was as follows:

- What makes a problem hard?
  - What are the key properties?
  - What are similar problems?
  - How is it hard?
  - Hard compared to what?
  - How does it scale?
  - How can ‘hard’ be quantified?

- Requirements for a proposed solution:
  - Give a Complete Method Description.
  - Identify Key Assumptions.
  - Identify key Parameters.
• Requirements for a demonstration:
  o Can I reproduce this?
  o Evaluate not Demonstrate.
  o Measures of performance.
  o Comparison with baseline methods.
  o Maybe you were lucky?
  o Maybe you were unlucky?
  o How does it fail?
• Generality of proposed solutions:
  o Replicability: ‘It worked in your lab. at least once under demo conditions’.
  o How well does it work (write other techniques)?
  o What are its limits?
  o What are its failure modes?
  o How does it scale?
  o ‘Will it work for me on my problem?’
  o What precisely may we conclude from the result?
• Actions to be taken:
  o Education:
    ▪ Courses in Methodology.
    ▪ Summer Schools / Workshops.
    ▪ Web materials on Methodology.
  o Reviewing guidelines.
  o Educate our own students.

**Actions**

The main planned action was to propose a *Special Interest Group* with the focus of increasing the quality of experimental methodology practiced in robotics by sharing good practice via educational workshops, summer schools, email discussion and web presentation; by providing assistance to journal and conference reviewers and editors concerning what constitutes experimental robotics and good practice in that sub-discipline; by encouraging the principled replication and comparison of results; and by encouraging the development and use of appropriate systems benchmarks and standard evaluation procedures.

Concretely, the following actions were considered:
• Make a document enumerating recommended quality criteria for Robotics Journal and Conference reviews, formulated as a reviewer checklist.
• Develop a Web site as part of the EURON web facility that includes information on planned meetings and activities, educational material on good experimental practices, etc.
• Organize and support workshops on good experimental methodology and Benchmarking at major robotics conferences
• Develop initiatives (meetings and information interchange) to collaborate with other experimental methodology, benchmarking and standards efforts.
• Develop a repository for tools and data sharing and a workflow environment for collaborative work over the Internet.
• Support the organization of summer/winter school(s) on experimental methods for robotics.
• Encourage and facilitate the publication of replications of robotics results, either in existing robotics journals or by establishing a high quality journal.
IROS’07 Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems

This workshop was the second main meeting on benchmarks held in 2007. It was organized by A.P. del Pobil in conjunction with a major scientific conference—such as IROS’07—to maximize success and optimize the use of resources. It was a full day event that took place on November 2, 2007. It was organized as a more formal event with a selection of full papers to guarantee the quality of the research. One invited paper on cognitive robotics together with six contributed papers were accepted and presented, authored by a total of 16 European researchers. Formal full-paper proceedings are included at the end of this document as Appendix XII.

The main purpose of this second IROS workshop was to provide a forum for participants to exchange their on-going work and ideas about performance evaluation and benchmarks in robotics research in general and in particular in high-level cognitive competencies encompassing knowledge representation, perception, control, and learning. It was intended as a forum for discussion, exchange of points of view, assessment of results and methods, and as a source of dissemination and promotion of benchmarks. It was co-organized with Rad Madhavan and Elena Messina from NIST, who are organizers of the PerMIS workshop series on Performance Metrics in Intelligent Systems.

The emphasis of the workshop was on cognitive solutions to practical robotics problems, so that cognitive approaches should enable an “intelligent” robot to behave appropriately in real-world scenarios in various application domains. In the context of this workshop, intelligence was understood as “the ability to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioural goals” (J. Albus, “Outline for a Theory of Intelligence”, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 21, No. 3, May/June 1991).

Summary of talks

Title: Synthetic Approach to Cognitive Systems: A Perspective from Cognitive Robotics.
Authors: Kaz Kawamura.
Speaker: Kaz Kawamura.
Abstract:
Today robotics technology is broadening its applications from factory to more general-purpose applications in domestic and public use, e.g., partner to the elderly, rehabilitations, search and rescue, etc. If robotics technology is to be successful in such unstructured, dynamic environments, it will need to meet new levels of robustness, physical dexterity and cognitive capability. This presentation discusses an emerging field called cognitive robotics from developmental point of view, i.e. "learning from building". Research topics and features of cognitive robotics are introduced. A case study of internal rehearsal for performance enhancement for cognitive robots will be introduced.
Title: Benchmarking Urban 6D SLAM.
Authors: Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernardo Wagner.
Speaker: Andreas Nuchter
Abstract: In the past many solutions for simultaneous localization and mapping (SLAM) have been presented. Recently these solutions have been extended to map large environments with six degrees of freedom (DoF) poses. To demonstrate the capabilities of these SLAM algorithms it is common practice to present the generated maps and successful loop closing. Unfortunately there is often no objective performance metric that allows comparing different approaches. This fact is attributed to the lack of ground truth data. For this reason we present a novel method that is able to generate this ground truth data based on reference maps. Further on, the resulting reference path is used to measure the absolute performance of different 6D SLAM algorithms building a large urban outdoor map.

Title: The Jacobs Test Arena for Security, Safety, and Rescue Robotics (SSRR).
Authors: Andreas Birk, Kaustubh Pathak, Jann Poppinga, Soren Schwertfeger, Max Pfingsthorn and Heiko Bulow.
Speaker: Andreas Birk
Abstract: The Jacobs University Bremen features a special test site for mobile robots. It consists of two large arenas with test elements, which are particularly suited to evaluate the performance of systems operating in challenging domains like Security, Safety, and Rescue Robotics (SSRR). The site is one of only six worldwide, which has been established in close cooperation with the US National Institute of Standards and Technology (NIST). This paper presents the Jacobs arenas and a set of metrics for evaluating the mobility, sensing capability, and onboard intelligence of robots. The tests are illustrated by using a Jacobs rescue robot, which is equipped with state of the art sensors. The relative strengths and weaknesses of these sensors are evaluated in a variety of situations; many of them are typically encountered in SSRR applications.

Title: Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems.
Authors: Benjamin Balaguer, Stefano Carpin, Stephen Balakirsky.
Speaker: Stefano Carpin
Abstract: Autonomous robotics has been plagued by the lack of quantitative comparisons between different solutions for the same problem. The situation arose due to a lack of theoretical background, recognized benchmarks, and the existence of a culture that is not oriented towards the free sharing of ready-to-use code for scientific research. In this paper we leverage a recent paradigm shift, and contrast different algorithms for Simultaneous Localization And Mapping (SLAM) readily available to the scientific community. In particular, we have run the same algorithms in two different settings. The first one is based on a P3AT robot operating inside a large building hosting office space and research labs. The second scenario is a virtual replication of the identical floor plan, implemented inside the USARSim simulation environment. In other words, the simulated scenario features the exact models of the environment and robot. The experimental setup offers a matrix where weaknesses and strengths of different SLAM algorithms can be contrasted in real and virtual environments, also outlining the degree to which the simulated results
can be extrapolated to measure or predict real world systems performance. We conclude that the availability of open source algorithm implementations, data sets, and simulation environments is the key to promote accelerated research in autonomous robotics. In particular, it appears that available SLAM implementations are robust and easy to use for environments like those used in our experiments, and therefore research efforts should be accordingly re-modulated.

Title: Reliability Testing for Embodied Autonomous Systems.
Authors: Louise F. Gunderson and Jim P. Gunderson
Speakers: Louise F. Gunderson and Jim P. Gunderson
Abstract:
Autonomous systems are intended to function effectively in dynamic and uncertain environments. Unfortunately, traditional testing methodologies are grounded in the assumptions of static environments and deterministic analysis. As a result, these methodologies fail to test the very characteristics that are critical for the deployment of autonomous systems. At the same time, there is more and more demand to provide quantitative benchmarks for autonomous systems.

New agile development methodologies, such as eXtreme Programming (XP), are becoming standard. XP, in particular, relies on constant automated testing to provide both the developers and the customers with some level of confidence that the systems are performing correctly and reliably. However, XP is not designed to handle the additional challenges posed by embedded systems. We provide an augmented XP methodology that is specifically designed to address the issues associated with the automated testing of embodied (i.e., robotic) autonomous systems in dynamic environments.

In this paper, we present the results of over four years of designing and developing tests for embodied autonomous systems. We use a brief case study to provide examples of several key pitfalls, and provide a high-level overview of the requirements that must be met to test these systems.

Authors: José L. Jiménez, Iñaki Rañó and Javier Mínguez.
Speaker: Iñaki Rañó
Abstract:
This paper will describe the advance in our project for benchmarking obstacle avoidance techniques for mobile robots. The core of the project is to create a methodology/software to evaluate the performance of the methods given a wide range of work conditions. These work conditions usually include scenarios with very different nature (dense, complex, cluttered, etc). The performance is measured in terms of robotic parameters (robustness, optimality, safety, etc). In the paper we will give an overview of the project and we will focus on the project analysis from a software engineering point of view. At this state the software design decisions are critical and could impede a proper later development; therefore we have developed a great effort in the analysis and design of the project.
Title: Good Experimental Methodologies in Robotics: State of the Art and Perspectives.
Authors: Fabio P. Bonsignorio, John Hallam, and Angel P. del Pobil.
Speaker: Fabio P. Bonsignorio

Abstract:
As the complexity of developed robotic and intelligent systems grows, it is more and more needed to define proper experimental approaches and benchmarking procedures. Trustable benchmarks are needed in order to allow the comparison of the many research results in service robotics research and enable their industrial application. On the other end, if robotics aims to be serious science replication of experiments deserves serious attention. It is necessary to be able to verify if and by which measure new procedures and algorithms proposed in research papers constitute a real advancement and can be used in new applications. New more successful implementations of concepts already presented in literature, but not implemented with exhaustive experimental methodology, risk to be ignored, if appropriate benchmarking procedures, allowing to compare the actual practical results with reference to standard accepted procedures, are not in place. Both replication and benchmarking are needed to foster a cumulative advancement of our knowledge of intelligent physical agents and even to correctly appreciate disruptive innovation in the science and technology of robots. Should we take inspiration from biology and medicine? In order to address these needs the European Union Network of Excellence on robotics EURON has funded a Special Interest Group on Good Experimental Methodology and benchmarking. This paper summarizes the state of the art in the field, the possible perspective activities identified so far, and the EURON GEM SIG challenges and ambitious plans.
GEMBENCH'08
EURON Workshop on Good Experimental Methodology and Benchmarking in Robotics

This two-day workshop was held on March 25 and 26, 2008 in Prague, Czech Republic collocated with the Second European Robotics Symposium (EUROS) and the 2008 Annual EURON Meeting. On this occasion emphasis in the Call for Contributions was on methodological aspects as a way of fostering good experimental practice and benchmarking. Nine European speakers presented their work or ideas with plenty of time for discussions, as listed in the Program below.

Motivation

All science proceeds from experiment, which motivates the creation of new theory and establishes the limits and validity of the existing theoretical basis. Individual branches of science conduct experiments differently, depending on the topic of investigation, but all have in common a body of knowledge concerning experimental methodology that specifies how to design and conduct `good' experiments in that discipline.

If robotics aims to be serious science, serious attention must be paid to experimental method. We can all point to published papers that report poor experimental work: few or no replications or tests; no, or dubious, comparison between algorithms; lack of suitable quantification of performance and its variability; conclusions which, while perhaps correct, are unjustified by the reported experimental work. It is time to address this. EURON GEM Special Interest Group focuses on increasing the quality of experimental methodology practiced in robotics. We believe this general aim can be achieved, for instance, by sharing good practice via educational workshops, summer schools, email discussion and web presentation; by providing assistance to journal and conference reviewers and editors concerning what constitutes experimental robotics and good practice in that sub-discipline; by encouraging the principled replication and comparison of results; and by encouraging the development and use of appropriate systems benchmarks and standard evaluation procedures.

Program

March 25th: Chair F Bonsignorio.

• Welcome and presentation of the workshop, F Bonsignorio.
• Open discussion about benchmarks and architectures for robustness and autonomy, H. Bruyninckx.
• Plenary discussion.
• Proposals for benchmarking SLAM, G. Fontana, M. Matteucci, J. Neira, D. Sorrenti.
• Motion Planning vs. Automated Planning in benchmarking, M. Reggiani, E. Pagello.
• Benchmarking mobile robots' motion, A. Marjovi, L. Marques.
• Plenary discussion.
March 26th: Chair A. P. del Pobil.

- Welcome and presentation of the workshop.
- The Hydra-Shiva concept for GEM and Benchmarking in robotics, A Moshaiov.
- Plenary discussion.
- RoSta - A Brief View Over Benchmarking Activities In Service Robotics, K. Pfeifer.
- GEM and Benchmarking in robotics, where we are? Serious? Science??, F. Bonsignorio, J. Hallam, A. P. del Pobil.
- Plenary discussion.
CogGEMBench'08 Workshop on Good Experimental Methodology & Benchmarks in Cognitive Robotics

This half-day workshop was held on April 1st, 2008 in Karlsruhe, Germany. It was originally submitted and accepted as part of the Program of the International Conference on Cognitive Systems (CogSys 2008), however the organization decided to cancel all workshops and replaced them by project presentations. We decided to hold the workshop one day before the conference in a nearby hotel room.

Motivation

As the complexity of current embodied cognitive systems grows, it is more and more necessary to define proper experimental approaches and benchmarking procedures. On the one hand, reliable benchmarks are called for in order to allow the comparison of the many research results in embodied cognitive systems, so that their potential application is eventually possible. On the other hand, if cognitive robotics aims to be regarded as serious science, replication of experiments deserves conscientious attention to verify if and by which measure new procedures and algorithms are real progress. New implementations of concepts in the literature, but not implemented with exhaustive experimental methodology, risk to be ignored, if appropriate benchmarking procedures are not in place, allowing to compare the actual practical results with reference to standard accepted procedures. Both replication and benchmarking are needed to foster a cumulative advancement of our knowledge of intelligent physical agents and even to correctly appreciate disruptive innovation in the science and technology of robots. In order to address these needs the European Robotics Network of Excellence (EURON) has funded a Special Interest Group on Good Experimental Methodology and Benchmarking. This workshop aims to provide a discussion forum on these topics and to identify guidelines for the future.

All science proceeds from experiment, which motivates the creation of new theory and establishes the limits and validity of the existing theoretical basis. Individual branches of science conduct experiments differently, depending on the topic of investigation, but all have in common a body of knowledge concerning experimental methodology that specifies how to design and conduct good experiments in that discipline. If cognitive robotics aims to be serious science, serious attention must be paid to experimental methods. During the last four years different initiatives have addressed benchmarking in a number of robotics areas such as visual servoing, motion planning, or SLAM. However, performance metrics for the cognitive aspects of embodied systems has been somehow neglected by both the robotics and cognitive systems communities. By getting together experts in both fields we plan to foster further progress in that direction. The research activities in this field are huge as it is the number of published papers. In order to allow the exploitation of the many results obtained it is at least necessary to be able to validate the results by replicating them or compare the results in terms of the selected performance criteria. Although some work is already carried on, many open issues are still ahead.

The emphasis of the workshop will be on cognitive solutions to both theoretical and practical problems. These cognitive approaches should enable an intelligent system to behave appropriately in real-world scenarios in various application domains. We also propose to discuss the distinction between autonomy and intelligence (if any) and how one influences the
other. We welcome any topic relevant to benchmarking and performance evaluation in the context of cognitive solutions to practical problems.

**Program**

- GEM and Benchmarking in robotics, F. Bonsignorio, J. Hallam, A. P. del Pobil.
- Quality Measures for Mapping: from Test Environments to Analysis Tools, Max. Pfingsthorn, Andreas Birk, Jacobs University Bremen, Germany
- Benchmarking in the DEXMART project, Gerhard Grunwalg, DLR, Germany
- Experiences in evaluating human-robot interaction - the COGNIRON project, Ingo Lütkebohle, University of Bielefeld, Germany
- Open Discussion
RSS’08 Workshop on Experimental Methodology and Benchmarking in Robotics Research

The 2008 Robotics Science and Systems Conference will be hosted by ETH Zurich, Switzerland, on June 25-28, 2008. The workshop proposal has been accepted to be held on June 28, and is co-organized by Angel P. del Pobil from Universitat Jaume I in Spain, John Hallam from the University of Southern Denmark, and Fabio Bonsignorio from Heron Robots (Italy).

As the complexity of current robotic and embodied intelligent systems grows, it is more and more necessary to define proper experimental approaches and benchmarking procedures. On the one hand, reliable benchmarks are called for in order to allow the comparison of the many research results in robotics research, so that their industrial application is eventually possible. On the other hand, if robotics aims to be regarded as serious science, replication of experiments deserves conscientious attention; it is necessary to be able to verify if and by which measure new procedures and algorithms proposed in research papers constitute a real advancement and can be used in new applications. New more successful implementations of concepts already presented in the literature, but not implemented with exhaustive experimental methodology, risk to be ignored, if appropriate benchmarking procedures are not in place, allowing comparing the actual practical results with reference to standard accepted procedures. Both replication and benchmarking are needed to foster a cumulative advancement of our knowledge of intelligent physical agents and even to correctly appreciate disruptive innovation in the science and technology of robots. Should we take inspiration from experimental practice in disciplines such as biology or medicine?

This workshop aims to provide a discussion forum on these topics and to identify guidelines for the future. The workshop will be organized in such a way as to generate fruitful discussions, it will consist of invited presentations (45 min. each) and regular presentations (25 min. each) with a significant amount of additional time for discussions. The primary audience of the workshop is intended to be researchers and practitioners both from academia and industry with an interest in experimental robotics. The workshop is also aimed at benchmarking and objectively evaluating performance of robots. Accordingly, it is envisioned to be useful for anyone who has an interest in quantitative performance evaluation of robots and/or robot algorithms. Some controversial issues will be discussed such as: measuring autonomy or information metrics of intelligent systems, or the concept itself of replicability or benchmarking of research results in robotics.

List of topics

- Good Experimental Methodology in Robotics Research
- Replication in embodied cognitive systems
- Methodological/experimental best practices
- Benchmark/comparison examples
- Benchmarking autonomy
- Information metrics of cognitive natural and artificial systems
- Benchmarking autonomy and robustness to changes in the environment/task
• Requirements, theories, architectures, models and methods that can be applied across multiple engineering and application domains
• Detailing and understanding better the requirements for robots in terms of performance, the approaches to meeting these requirements, the trade-offs in terms of performance
• The development of experimental scenarios to evaluate performance, demonstrate generality, and measure robustness
IROS’08 Workshop on Performance Evaluation and Benchmarking for Intelligent Robots

For third consecutive year, we organize a workshop together with the IROS conference. The 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008) will be held at the Acropolis Convention Center, Nice, France from September 22 to 26, 2008. The workshop proposal has been accepted and the event will take place on September 26, it is co-organized by Angel P. del Pobil, Raj Madhavan and Fabio Bonsignorio.

Objectives and Topics

Realization of purposive goals in complex environments, especially in applied robotics (e.g. service, manufacturing, and unmanned vehicle applications), impose stringent demands on robots in terms of being able to cope with and react to dynamic situations using limited on-board computational resources, incomplete and often corrupted sensor data. High-level cognitive competencies encompassing knowledge representation, perception, control, and learning are considered essential elements that will allow robots to perform complex tasks within highly uncertain environments. New more successful implementations of concepts already presented in literature, but not implemented with exhaustive experimental methodology, run the risk of being ignored, if appropriate benchmarking procedures, allowing to compare the actual practical results with reference to standard accepted procedures, are not in place.

It is a well-known fact that current robotics research makes it difficult not only to compare results of different approaches, but also to assess the quality of individual research work. Some steps have been taken to address this problem by studying the ways in which research results in robotics can be assessed and compared. In this context the European Robotics Research Network EURON has as one of its major goal the definition and promotion of benchmarks for robotics research. As part of this goal, it has funded a Special Interest Group on Good Experimental Methodology and Benchmarking in Robotics. Similarly, the Performance Metrics for Intelligent Systems (PerMIS) Workshop series has been dealing with similar issues in the context of intelligent systems. The main purpose of this workshop is to contribute to the progress of performance evaluation and benchmarking, focusing in intelligent robots and systems, by providing a forum for participants to exchange their on-going work and ideas in this regard. These objectives are pursued in a context where there is not a wide agreement on the concepts like ‘autonomy’, ‘cognition’ and ‘intelligence’.

The emphasis of the workshop will be, then, on cognitive solutions to practical problems. These cognitive approaches should enable an “intelligent” system to behave appropriately in real-world scenarios in various unstructured application domains. In the context of this workshop, we define intelligence as “the ability to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral goals” (J. Albus, “Outline for a Theory of Intelligence”, IEEE Trans. on Systems, Man, an Cybernetics, Vol. 21, No. 3, May/June 1991).
Another key issue will be a capability-led understanding of cognitive robots: how to define shared ontologies or dictionaries to discuss robotic cognitive systems in terms of their performance, relationships between different cognitive robotics capabilities, requirements, theories, architectures, models and methods that can be applied across multiple engineering and application domains, detailing and understanding better the requirements for robots in terms of performance, the approaches to meeting these requirements and the trade-offs in terms of performance. The proper definition of benchmarking is related to the problem of measuring capabilities of robots in a context in which, in many cases, the ‘robotics experiments’ themselves are difficult to ‘replicate’.

**Topics**

We welcome any topic relevant to benchmarking and performance evaluation in the context of cognitive solutions to practical problems, such as:

- Knowledge representation, perception (sensing), and learning
- Uncertainty management in robot navigation, path-planning and control
- Cognitive manipulation
- Benchmarking autonomy and robustness to changes in the environment/task
- Capability-led understanding of cognitive robots
- Shared ontologies to discuss robotic cognitive systems in terms of their performance capabilities
- Relationships between different cognitive robotics capabilities
- Requirements, theories, architectures, models and methods that can be applied across multiple engineering and application domains
- Detailing and understanding better the requirements for robots in terms of performance, the approaches to meeting these requirements, the trade-offs in terms of performance
- The development of experimental scenarios to evaluate performance, demonstrate generality, and measure robustness
- Benchmarking of sensory motor coordination
- Performance modeling of the relationship between a task and the environment where it is performed
- Relationship between benchmarking and replication of experiments with robots
ROBOT COMPETITIONS AND CHALLENGES

ROBOCUP SOCCER

Introduction

RoboCup is an international research and education initiative. It is an attempt to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined, as well as being used for integrated project-oriented education.

For this purpose, RoboCup chose to use soccer game as a primary domain, and organizes RoboCup: (The Robot World Cup Soccer Games and Conferences). In order for a robot team to actually perform a soccer game, various technologies must be incorporated including: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion. RoboCup is a task for a team of multiple fast-moving robots under a dynamic environment. RoboCup also offers a software platform for research on the software aspects of RoboCup.

While the soccer game is used as a standard problem where a broad-range of efforts will be concentrated and integrated, competition is only a part of RoboCup activity.

Activities of the RoboCup consist of:
- Technical Conferences.
- RoboCup International Competitions and Conferences.
- RoboCup Challenge Programs.
- Education Programs.
- Infrastructure Development.

Objective

RoboCup is used as a vehicle to promote robotics and AI research, by offering publicly appealing, but formidable challenges. One of the effective ways to promote engineering research, apart from specific application developments, is to set a significant long term goal. When the accomplishment of such a goal has significant social impact, it is called the grand challenge project. Building a robot to play soccer game itself, do not generate significant social and economic impact, but the accomplishment will certainly be considered as a major achievement of the field. This kind of project is called as a landmark project. RoboCup is a landmark project as well as a standard problem.
The ultimate goal of the RoboCup Initiative is: *by mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, complying with the official rule of the FIFA, against the winner of the most recent World Cup.*

This goal is proposed to be one of the grand challenges shared by robotics and AI community for the next 50 years. This goal may sound overly ambitious given the state of the art technology today. Nevertheless, it is important that such a long range goal is claimed and pursued. It took only 50 years from the Wright Brother's first aircraft to Apollo mission to send man to the moon and safely return them to the earth. Also, it took only 50 years, from the invention of digital computer to Deep Blue, which beat the human world champion in chess. Building humanoid soccer player requires equally a long period and extensive efforts of broad range of researchers, and the goal will not be met in any near term.

The RoboCup is designed to meet the need of handling real world complexities, though in a limited world, while maintaining an affordable problem size and research cost. RoboCup offers an integrated research task covering the broad areas of AI and robotics. Such areas include: real-time sensor fusion, reactive behaviour, strategy acquisition, learning, real-time planning, multi-agent systems, context recognition, vision, strategic decision-making, motor control, intelligent robot control, and many more.

**Rules**

The regulations provided constitute official regulations of RoboCup games, administered by The RoboCup Federation. The initial regulations were drafted in 1994. Several revisions were made from technical and logistical point of view. The basis of the current regulation was drafted in the 1996 version of the RoboCup regulations, and modified each year for RoboCup World Championship games. Currently rule revision for RoboCup-2005 is under discussion.

**RoboCup Policy of Rule changes**

RoboCup's rule changes in order to promote science and technology. It will be reviewed annually by the committee, and discussed with participants and other knowledgeable researchers in the field to draft out new rules. One example of major change in rule, which may happen in future, is the removal of all walls around the field. It is being planned to make a technology assessment when it can achieve the level of technology to do this, and set up a target year for such a rule change. These measures will be taken with the careful evaluation of the technology level achieved. However, in principle, RoboCup's rule will be continuously modified to make it closer to real world, rather than to impose an artificial set up to improve superficial performance.

**Rules for Each League**

The League Committee is the official body for determination of rules and other long-range issues for each league. Current rules and discussions are maintained by a website of each league committee.
Competitions

RoboCup has different kinds of tournaments and each one has different categories and rules in it, namely:

1. RoboCup Soccer:
   a. Simulation League.
   c. Middle Size Robot League (f-2000).
   d. Four-Legged Robot League.
   e. Humanoid League (from 2002).

2. RoboCup Rescue
   a. Rescue Simulation League
   b. Rescue Robot League

3. RoboCup@Home

4. RoboCup Junior
   a. Soccer Challenge
   b. Dance Challenge
   c. Rescue Challenge

RoboCup Junior is a project-oriented educational initiative that sponsors local, regional and international robotic events for young students. It is designed to introduce RoboCup to primary and secondary school children, as well as undergraduates who do not have the resources to get involved in the senior leagues. The focus in the junior league is on education, and we will not describe it further in this report.

History

The idea of robots playing soccer was first mentioned by Professor Alan Mackworth (University of British Columbia, Canada) in a paper entitled ``On Seeing Robots” presented at VI-92, 1992, and later published in a book Computer Vision: System, Theory, and Applications, in 1993. A series of papers on the Dynamo robot soccer project was published by his group.

Independently, a group of Japanese researchers organized a Workshop on Grand Challenges in Artificial Intelligence in October, 1992 in Tokyo, discussing possible grand challenge problems. This workshop led to serious discussions of using the game of soccer for promoting science and technology. A series of investigations were carried out, including a technology feasibility study, a social impact assessment, and a financial feasibility study. In addition, rules were drafted, as well as prototype development of soccer robots and simulator systems. As a result of these studies, it was concluded that the project was feasible and desirable. In June 1993, a group of researchers, including Minoru Asada, Yasuo Kuniyoshi, and Hiroaki Kitano, decided to launch a robotic competition, tentatively named the Robot J-League (J-League is the name of the newly established Japanese Professional soccer league). Within a month, however, they received overwhelming reactions from researchers outside of Japan, requesting that the initiative be extended as an international joint project. Accordingly, they renamed the project as the Robot World Cup Initiative, “RoboCup” for short.
Concurrent to this discussion, several researchers had already been using the game of soccer as a domain for their research. For example, Itsuki Noda, at ElectroTechnical Laboratory (ETL), a government research center in Japan, was conducting multi-agent research using soccer, and started the development of a dedicated simulator for soccer games. This simulator later became the official soccer server of RoboCup. Independently, Professor Minoru Asada's Lab at Osaka University, and Professor Manuela Veloso and her student Peter Stone at Carnegie Mellon University had been working on soccer playing robots. Without the participation of these early pioneers of the field, RoboCup could not have taken off.

In September 1993, the first public announcement of the initiative was made, and specific regulations were drafted. Accordingly, discussions on organizations and technical issues were held at numerous conferences and workshops, including AAAI-94, JSAI Symposium, and at various robotics society meetings.

Meanwhile, Noda's team at ETL announced the Soccer Server version 0 (LISP version), the first open system simulator for the soccer domain enabling multi-agent systems research, followed by version 1.0 of Soccer Server (C++ Version) which was distributed via the web. The first public demonstration of this simulator was made at IJCAI-95.

During the International Joint Conference on Artificial Intelligence (IJCAI-95) held at Montreal, Canada, August, 1995, the announcement was made to organize the First Robot World Cup Soccer Games and Conferences in conjunction with IJCAI-97 Nagoya. At the same time, the decision was made to organize Pre-RoboCup-96, in order to identify potential problems associated with organizing RoboCup on a large scale. The decision was made to provide two years of preparation and development time, so that initial group of researchers could start robot and simulation team development, as well as giving lead time for their funding schedules.

Pre-RoboCup-96 was held during the International Conference on Intelligence Robotics and Systems (IROS-96), Osaka, from November 4 - 8, 1996, with eight teams competing in a simulation league and demonstration of real robot for middle size league. While limited in scale, this competition was the first competition using soccer games for promotion of research and education.

The first official RoboCup games and conference was held in 1997 with great success. Over 40 teams participated (real and simulation combined), and over 5,000 spectators attended. The number of teams and spectators has been growing successfully. In Bremen (2006, Germany) 15,270 people were present in the exhibition halls from 14 to 18 June and 2,613 of them (440 participating teams from 35 countries) took an active part in the contests.

The 12th RoboCup International Competitions and Conferences, RoboCup-2008 Suzhou is to be held on July 14th - July 20th, 2008, Suzhou, China. The RoboCup-2009 will be held in Graz, Austria.
## Participation

<table>
<thead>
<tr>
<th>Place</th>
<th>Year</th>
<th>Participating teams</th>
<th>Countries/regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>2007</td>
<td>300</td>
<td>37</td>
</tr>
<tr>
<td>Bremen</td>
<td>2006</td>
<td>440</td>
<td>35</td>
</tr>
<tr>
<td>Osaka</td>
<td>2005</td>
<td>330</td>
<td>31</td>
</tr>
<tr>
<td>Lisbon</td>
<td>2004</td>
<td>346</td>
<td>37</td>
</tr>
<tr>
<td>Padova</td>
<td>2003</td>
<td>277</td>
<td>34</td>
</tr>
<tr>
<td>Fukuoka/Busan</td>
<td>2002</td>
<td>188</td>
<td>29</td>
</tr>
<tr>
<td>Seattle</td>
<td>2001</td>
<td>119</td>
<td>22</td>
</tr>
<tr>
<td>Melbourne</td>
<td>2000</td>
<td>110</td>
<td>19</td>
</tr>
<tr>
<td>Stockholm</td>
<td>1999</td>
<td>120</td>
<td>35</td>
</tr>
<tr>
<td>Paris</td>
<td>1998</td>
<td>63</td>
<td>20</td>
</tr>
<tr>
<td>Nagoya</td>
<td>1997</td>
<td>40</td>
<td>10</td>
</tr>
</tbody>
</table>

## Comments

### RoboCup Simulation League

RoboCup Simulation League consists of a number of competitions with simulated soccer matches as the main event.

**Categories:**

**i. 2D Soccer Competition**

In the RoboCup Simulation League teams of 11 autonomous software agents per side play each other using the RoboCup soccer server simulator, available from the official simulator website.

There are no actual robots in this league but spectators can watch the action on a large screen, which looks like a giant computer game.

Each simulated robot player may have its own play strategy and characteristic and every simulated team actually consists of a collection of programmes. Many computers are networked together in order for this competition to take place.

The games last for about 10 minutes, with each half being 5 minutes duration.

**Rules:**

For more information about the rules of this competition see Appendix I.

**ii. 3D Soccer Competition**

The 3D competition makes use of the simulator that is based on the simulation system introduced at the RoboCup 2003 symposium and the spades simulation middleware system introduced at the RoboCup 2002 symposium. It can be downloaded from sourceforge.

**Rules:**

For more information about the rules of this competition see Appendix I.
iii. 3D Soccer Competition Development

The idea of the development competition is to get more input from the community in order to speed up the development of the 3D simulator. Last year, there were three focus topics for the implementation of new ideas:

- Humanoid robot models and controllers
- Visualization
- Tools (offline trainer, robot model editor, log 2 flash converter, etc.)

If the proposed ideas get accepted by the technical committee, they will be incorporated into the official simulator.

One of the goals for future 3D soccer competitions is to have simulated robots with articulated bodies, for example like humanoid robots.

Rules:
For more information about the rules of this competition see Appendix I

**Small Size Robot League**

![Small size robot soccer](image)

Small size robot soccer is one of the RoboCup league divisions. Small size robot soccer, or F180 as it is otherwise known, focuses on the problem of intelligent multi-agent cooperation and control in a highly dynamic environment with a hybrid centralized/distributed system.

A small size robot soccer game takes place between two teams of five robots each. Each robot must conform to the dimensions as specified in the F180 rules, the robot must fit within a 180mm diameter cylinder. If a team is using the global vision system, each robot on that team must have height of 150 mm or less. In all cases, a robot must have height less than 225 mm. The robots play soccer on a green carpeted field that is 5m long by 3.5m wide with an orange golf ball. Robots come in two flavours, those with local on-board vision sensors and those with global vision. Global vision robots, by far the most common variety, use an overhead camera and off-field PC to identify and track the robots as they move around the field. The overhead camera is attached to a camera bar located 3m above the playing surface. Local vision robots have their sensing on the robot itself. The vision information is either processed on-board the robot or is transmitted back to the off-field PC for processing. An off-field PC is used to communication referee commands and, in the case of overhead vision, position information to the robots. Typically the off-field PC also performs most, if not all, of the processing required for coordination and control of the robots. Communications is
wireless and typically uses dedicated commercial FM transmitter/receiver units although at least one team has used IRDA successfully.

Building a successful team requires clever design, implementation and integration of many hardware and software sub-components into a robustly functioning whole making small size robot soccer a very interesting and challenging domain for research and education.

**Rules:**
For more information about the rules of this competition see Appendix I

**Middle Size Robot League**

Two teams of mid-sized robots with all sensors on-board play soccer on a field. Relevant objects are distinguished by colors. Communication among robots (if any) is supported on wireless communications. No external intervention by humans is allowed, except to insert or remove robots in/from the field.

**Other challenges**

**Ball Control and Planning**

Six to eight black obstacles (length/width 40 cm, height 60 cm) are put at arbitrary positions on the field. The ball is put on the middle of the penalty area line, and a robot inside the same goal. The robot should dribble the ball into the opposite goal within 90 seconds, while it avoids all obstacles. One point is awarded to the robot if the ball has passed the centre line, another point when a goal is scored.

Penalty points are given each time the robot or the ball touches an obstacle. The challenge is repeated three times with various setups. An extra point is awarded to the team with the fastest robot. In order to be eligible for this extra point the robot may not have touched any of the obstacles. In total a team can be awarded up to seven points for this challenge.

**Cooperative behaviour**

Teams should demonstrate cooperative behaviour between at least two robots. The selection of the activity to be performed is free, but it should last at most 90 seconds. A jury will evaluate the quality of cooperation and cooperative behaviour and will assign up to six points to each team.
Cooperative Mixed-Team Play

Teams should demonstrate cooperative mixed-team play between at least two robots from different teams. The selection of the activity to be performed is free, but it should last at most 90 seconds. A jury will evaluate the quality of cooperation and cooperative behaviour and will assign up to six points to each team.

Play with an arbitrary FIFA ball

The aim of this challenge is to encourage teams to improve their vision routines. This challenge is carried out with three different standard FIFA balls. A robot is placed on the field and the ball is placed in front of the robot for 5 seconds. Afterwards the ball is placed at an arbitrary position on the field. The robot has now 60 seconds to find the ball and to dribble it into a predefined goal. One point is awarded to the robot for correctly identifying the ball, i.e. the robot has found and touched the ball for the first time. A second point is awarded if the robot has scored a goal. In total this challenge is repeated three times with varying balls but always with the same robot. In total a team can be awarded up to six points for this challenge.

Play with arbitrary goals

A robot is placed in a random position within its own half of the field and a regular RoboCup ball is placed in a random position in the other half. The robot should kick the ball in the opposite goal (the one in the other half of the field with respect to its initial position). No assumptions should be made on the colors of the two goals. The organizer will cover the yellow and blue goals with a different color pattern (except for the red). The challenge is repeated three times with different initial positions of the robot and of the ball. Each attempt has a time limit of 60 seconds. One point is awarded to each score and an additional point is given to robots that score three times in less than 30 seconds each. In total a team can be awarded up to four points for this challenge.

Show scientific or engineering achievements

Teams are free to show one significant achievement each, and all the other team leaders, together with the TC members will judge them. Achievements in the present list are encouraged.

Passing

At the beginning, the ball is on the own penalty kick point. A player (#1) is on the center point. Another player (#2) is in the opposite half field. A still goalie (a black box, 50 cm wide) is in the middle of the opponent goal. Player #1 has to get the ball, pass it to player #2 which should score the goal. 1 point if the ball passes midline, 1 point if it is dribbled by player #1 over midline, 3 points if it hits player #2, 3 points if player #2 tries to score in the opponent goal (either by kicking or bringing the ball there, always following the rules), 2 points if the goal is scored. Time is also taken and used to rate the teams in case of same number of points.

Rules:

For more information about the rules of this competition see Appendix I
Four-Legged Robot League

One of the ultimate dreams in robotics is to create life-like robotics systems, such as humanoid robots and animal-like legged robots. Quadruped robots have major potential for future robotics, including entertainment applications for personal robots. However, a number of challenges exist before any such robot can be fielded in the real world. Robots have to be reasonably intelligent, maintain a certain level of agility, and be able to engage in some collaborative behaviours. RoboCup is an ideal challenge to foster robotics technologies for small personal and mobile robotics systems. Through the research and competition in this league, technology will be developed and will improve robot performance not only in entertainment but also in non-entertainment applications such as rescue robots and other dangerous jobs currently performed by humans, and that robots will perform these tasks in the future by working together.

In the league teams consisting of four Sony Aibo robots each play on a field of 5.4 m x 3.6 m. The robots operate fully autonomously, i.e. there is no external control, neither by humans nor by computers.

Other Challenges

The Open Challenge

This challenge is designed to encourage creativity within the Legged League, allowing teams to demonstrate interesting research in the field of autonomous systems. Each team will be given three minutes of time on the RoboCup field to demonstrate their research. Each team must distribute a technical description of their research (1-2 pages) before the round-robin starts. Teams who do not submit a description or propose a non-technical challenge will be ineligible to compete. The winner will be decided by a vote among the entrants. In particular:

- Each team must distribute a technical description of their research before the round-robin starts.
- Each team may use any number of Sony AIBO robots. Teams must arrange for their own robots.
- Teams have three minutes to demonstrate their research. This includes any time used for initial setup. Any demonstration deemed likely to require excessive time may be disallowed by the organizing committee.
• Teams may use extra objects on the field, as part of their demonstration. Robots other than the AIBOs may not be used.
• The demonstration must not mark or damage the field. Any demonstration deemed likely to mark or damage the field may be disallowed by the organizing committee.
• The demonstration may not use any off-board sensors or actuators, or modify the AIBO robots.
• The demonstration may use off-board sensors or actuators as long as the AIBO is still the focus of the challenge.
• The demonstration may use off board computing power connected over the wireless LAN. This is the only challenge in which off board computation is allowed.
• The demonstration may use off board human-computer interfaces. This is the only challenge in which off board interfaces, apart from the Game-Controller, are allowed.

The winner will be decided by a vote among the entrants using a Borda count (http://en.wikipedia.org/wiki/Borda_count). Each entering team will list their top 10 teams in order (excluding themselves). The teams are encouraged to evaluate the performance based on the following criteria: Technical strength, novelty, expected impact and relevance to RoboCup. At a time decided by the designated referee, within 30 minutes of the last demonstration if not otherwise specified, the captain of each team will provide the designated referee with their rankings. Each ranking is converted to points: ten points for the top ranked team, nine for the team ranked second and so on down to one point for the team ranked tenth. Any points awarded by a team to itself will be disregarded. The points awarded by the teams are summed and the team with the highest total score shall be the winner.

The Passing Challenge

This second challenge is intended to encourage teams to develop passing and catching skills. In this challenge each team will be required to provide three robots; all robots must be in the same coloured uniform (the decision on red or blue uniforms can be made by each team).

Each robot will be placed on the field inside a circle of radius 35 cm. The centre of the circles will be no closer then 80 cm and no further then 200 cm apart. The triangle formed by the circles will not be equilateral, i.e. the distances between robots will be different.

Initially the robots will be in the ‘set’ state for 15 seconds, this will enable them to localise. The robots will then be placed into ‘playing’ and given two minutes to pass the orange ball around. A pass will be regarded as successful when:
• The passing robot releases the ball from inside its circle and the catching robot stops/controls the ball inside its circle.
• A pass will be deemed partially successful if:
• The passing robot releases the ball from inside its circle and the catching robot touches the ball inside the circle but the ball then travels outside the circle.
• A pass is deemed unsuccessful if:
• Either robot makes contact with the ball when the ball is outside a circle or the ball exits the field.

Robots may pass between each other in any order, but will be rewarded for passing to a different robot then that which passed to it.
All normal game rules apply in the challenge, except when a ball leaves the field it will be replaced back in the closest circle. If a rule is violated then any pass resulting from this violation will receive no points.

The New Obstacle Avoidance Challenge

The purpose of this challenge is to successfully solve a series of problems. In summary robots must navigate from one goal to the other goal while avoiding obstacles and performing tasks.

This challenge will require the team to use two robots, one red and one blue. The blue robot will start in the blue goal while the red robot will start in the yellow goal. The task of each robot is to reach the opposing goal. “Reaching” means to have at least two feet in the opposite goal (that is two legs behind the goal line). Along the way the two robots will face a series of obstacles. The obstacles may include:

- Each other.
- Stationary robots - Robots (of either colour) will be placed at random locations in the standard UNSW stance. One of the robots may be placed as a goal keeper (i.e. inside a penalty area).
- Moving robots - Robots (of either colour) may be moving around the field. For example running between two beacons or moving between two points.
- A robot in a different uniform - At least one of the robots (stationary or moving) will be wearing a different uniform (color and/or pattern).

At the middle of the field the two robots will be asked to perform a simple cooperative task. In this case both robots will have to 'cross' the halfway line at the same time. The rule will be defined as 'the two robots can not be in the same half of the field for more than 3 seconds'.

Humanoid League

The Humanoid Leagues are special within RoboCup. The human-like embodiment of the robots leads to extraordinary tasks and challenges that are different from the other RoboCup leagues. In addition, humanoid robots are essential to fulfil the aim of beating the human world champion in soccer by the best humanoid team in the year 2050.

Humanoid robots show basic skills of soccer players, such as shooting a ball, or defending a goal. Relevant objects are distinguished by colors.

External intervention by humans is allowed, as some of the humanoid robots are tele-operated.

Strategy for the next years:

The robots are diverse: The size of humanoid robots that participated since the start of the league varies from 10cm to 2m. Robots of so different sizes can hardly play soccer together. For the next years one important aim is to bring the robots together in one league. The number of leagues is 2: Kidsize and Teensize.

Robots participating in the Humanoid League competitions must have a human-like body plan, as shown in the figure. They must consist of two legs, two arms, and one head,
which are attached to a trunk. The robots must be able to stand upright on their feet and to walk on their legs. The only allowed modes of locomotion are bipedal walking and running.

![Diagram of robot legs and torso](image)

The Footrace

In the TeenSize class, the footrace replaces soccer games until enough TeenSize teams are able to play soccer games. The footrace is done as a 1 vs. 1 competition between two teams. It takes place on a TeenSize field. Access to the field is given to both teams at least 10 minutes prior to the scheduled starting time.

A footrace match consists of five runs. A run goes from one touch line to the other touch line. Both teams place their robot in front on the border strip, outside the field, in front of the touch line. The robots must be in an upright standing posture. One team uses the yellow side of the field and the other team uses the blue side. Different robots might be used for multiple runs.

After the referee gives the start signal, both robots walk as fast as possible across their half of the field towards the opposite touch line. Robot handlers are not allowed to enter the field, unless the referee asks them to remove a robot.

If a robot touches the start line before the referee gave the start signal, the start is invalid. This robot receives a warning and the start is retaken. The warnings of a team accumulate within a match. Every third warning results in the technical winning of a run for the other team.

The robot which first crosses the goal line wins the run. Both feet must be outside the field on the border strip again. If a robot falls, it must get up by itself to continue the run.

If a robot leaves its half of the field, its run terminates at the position where it left the field. If 60s after the start signal no robot crossed the goal line, the referee decides which robot advanced more towards the goal line. This robot wins the run. If the referee cannot determine which of the robots advanced more the run ends in a draw. Both teams must place their robot in front of the start line within 60s after the end of a run.

The team which wins most of the runs wins the footrace match. If both teams win an equal number of runs, the match ends in a draw. To decide knock-out matches, the runs are continued until one team wins a run.
Challenges

- **Obstacle avoidance**

   Six black obstacles (cylinders of diameter 20 cm and height 90 cm) are put in the half of the field in front of the yellow goal, as defending players. The obstacles are put at arbitrary positions by the referees, just before the start of each trial, when the robot is already waiting at the starting position and are rearranged at each trial. The distance between the obstacles is at least 50 cm for KidSize and at least 100 cm for TeenSize. No obstacles are placed in the centre circle. The robot is placed at the centre mark. The robot should reach the goal, enter in it, by crossing at least once the line between two obstacles and by avoiding touching all obstacles.

   The trial ends without success, if the robot bumps on any obstacle or if the robot leaves the field. If the robot touches an obstacle a minus is given.

   The trial ends with success, if the robot touches the goal line. The teams are ranked by the time needed to complete the task and the number of minus.

   Poles are only re-arranged once the robot has actually made an effort to navigate around the poles and a minimum of 60 sec. must have elapsed.

- **Dribbling Around Poles**

   Two cyan poles are placed at the penalty kick marks of a TeenSize field. One magenta pole is placed at the center mark. The poles have a diameter of 20cm and a height of 90cm.

   - The ball and the robot start in one of the goal areas.
   - The task is to dribble the ball around the poles as indicated by the example trajectory.
   - The trial starts when the ball leaves the goal area.
   - The trial ends without success if the ball leaves the field or if the ball crosses the line between two poles too early. The trial also ends without success if the robot collides with a pole.
   - The ball must cross four times the line between a magenta pole and a cyan pole, in the indicated order.
   - The trial ends successfully if the ball reenters the goal area from where the robot started.

- **Passing**

   A KidSize field is marked with two additional white lines, which are parallel to the middle line and tangential to the centre circle. The ball is placed at a penalty kick mark. One robot is placed inside each goal area.

   - The task is to pass the ball between the robots back and forth.
   - The trial starts when the first robot leaves its goal area.
   - The trial ends without success if the ball leaves the field or stops inside the middle area. The trial also ends without success if one of the robots leaves the field or enters the middle area.
   - The ball has to cross the middle area five times, as indicated by the example trajectory.
   - The trial ends successfully if the receiving robot touches the ball after the fifth crossing.

Rules

For more information about the rules of this competition see Appendix I
ROBOCUP RESCUE

Disaster rescue is one of the most serious social issues which involves very large numbers of heterogeneous agents in a hostile environment. The main goal of the RoboCup Rescue project is to promote research and development in this socially significant domain at various levels involving multi-agent team work coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, a standard simulator and decision support systems, evaluation benchmarks for rescue strategies and robotic systems that are all integrated into a comprehensive systems in future.

Two projects and leagues, Simulation League and Real Robot League, are currently in use. The goal is that integration of these activities will create the digitally-empowered international rescue brigades in the future.

Objective

The RoboCupRescue -Simulation League- is an international testbed for the simulation of software agents and robots performing Urban Search And Rescue (USAR) missions. The main purpose of the RoboCupRescue Simulation Project is to provide emergency decision support through the integration of disaster information, prediction, planning, and human interface. Heterogeneous intelligent agents conduct search and rescue activities in this virtual disaster world. This problem introduces researchers to advanced and interdisciplinary research themes. In AI/Robotics research, for example, behaviour strategy (e.g. multi-agent planning, realtime/anytime planning, heterogeneity of agents, robust planning, mixed-initiative planning) is a challenging problem.

Road Map

June 1999: Version 0 simulator is open to public.
August 2001: 1st competition for research evaluation starts.
2003: Version 1 simulator with more realistic simulation and agent behaviour with PDA interface.
2005: Version 2 simulator partially in practical level as a decision support system.
2020: Realization of the RoboCup Rescue concept with digitally empowered rescue brigades.
2050: Autonomous robot rescue agent team saves human lives

Participation

More than 30 teams from all around the world (Portugal, Iran, Denmark...).


**History**

May 1998  Hiroaki Kitano and Satoshi Tadokoro meet at ICRA98 in Leuven discussing to start RoboCup Rescue.
Nov. 1998  Discussion about definite plan of research & competition starts.
Jun. 1999  Development of prototype simulator (version 0) starts.
Dec. 1999  RoboCup Rescue Symposium in Kobe.
Mar. 2000  Organized Session in Annual Conference of Information Processing Society Japan
May 2000  Japan Open 2000 Hakodate demonstration. Version 0 simulator is open to the public.
Aug. 2000  RoboFesta Kansai Pre-Festival demonstration
Aug. 2000  RoboCup World Cup 2000 Melbourne demonstration. Version 0 simulator is demonstrated and explained. Call for project participants starts.

**Comments**

**RoboCup Rescue Simulation League**

The RoboCup Rescue Simulation competition is held within three competitions:

1. **Agent competition**

   The agent competition is a modular large-scale disaster simulation in real time, to which multiple agent teams can connect in order to reduce human casualties and damage to buildings. Typical agent types are police forces, ambulance teams, and fire brigades. The rescue domain represents a real multi-agent scenario since most of the encountered problems cannot be solved by a single agent. For example, fire brigades depend on police forces to clear blocked roads in order to extinguish fires. Therefore, team cooperation and coordination is highly required in this domain. Moreover, the task is challenging due to the limited communication bandwidth, the agent's limited perception and the difficulty of predicting how disasters evolve over time.

2. **Infrastructure competition**

   The purpose of the infrastructure competition is to foster the development of software components for simulation, such as the simulation of fire spread. This is necessary, since in the RSL, teams compete against a disastrous environment rather than against opponents, as it is the case in other leagues. The teams are requested to provide their components under an Open Source policy for the next year's agent competition.

3. **Virtual Robots Competition**

   This competition is based upon the robot simulator USARSim, which is based on the game engine from the commercial computer game Unreal Tournament. USARSim allows high fidelity simulations of multi-robot systems. It currently offers the possibility to simulate commercial as well as self-developed robot platforms.
USARSim complements the current league in an ideal way with a realistic physics simulation of teams of robots operating within collapsed buildings. On the one hand, it offers the possibility to simulate search and rescue scenarios where every agent has capabilities comparable with those found on real robots, such as sensing with laser range finders or thermo sensors. On the other hand, it opens the door for investigating aspects of autonomous multi-robot cooperation within unknown and unstructured domains.

Rules

For more information about the rules of this competition see Appendix II

RoboCup Rescue Robot League

The RoboCup Rescue - Robot League - competition is an international evaluation conference for the RoboCup Rescue Robotics and Infrastructure Project research.

The RoboCup Rescue Robotics and Infrastructure Project studies future standards for robotic infrastructure built to support human welfare. Currently the NIST USAR arena has been used in several RoboCup Rescue and AIAA competitions.

A team of multiple (autonomous or tele-operated) robots moves inside this testbed, divided in 3 regions of increasing difficulty levels, searching for victims and building maps of the surrounding environment, to be transmitted and/or brought back by the robot(s) to the human operators.

The goal of annual RoboCup Rescue Robot League competitions is to increase awareness of the challenges involved in urban search and rescue (USAR) applications, provide objective evaluation of robotic implementations in representative environments, and promote collaboration between researchers.

The objective for each robot in the competition, and the incentive to traverse every corner of each arena, is to find simulated victims. Each simulated victim is a clothed mannequin emitting body heat and other signs of life, including motion (shifting, waving), sound (moaning, yelling, tapping), and/or carbon dioxide to simulate breathing. Particular combinations of these sensor signatures imply the victim’s state: unconscious, semi-conscious, or aware.

It requires robots to demonstrate their capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in unstructured environments. As robot teams begin demonstrating repeated successes against the obstacles posed in the arenas, the level of difficulty will be increased accordingly so that the arenas provide a stepping-stone from the laboratory to the real world. Meanwhile, the yearly competitions will provide direct comparison of robotic approaches, objective performance evaluation, and a public proving ground for field-able robotic systems that will ultimately be used to save lives.

The ultimate goal, of course, is to develop complete and effective robotic systems that can be deployed in the field to save lives.

Rules

For more information about the rules of this competition see Appendix II
ROBOCUP@HOME

Introduction

RoboCup@Home is a new league inside the RoboCup competitions that focuses on real-world applications and human-machine interaction with autonomous robots.

Objective

The aim is to foster the development of useful robotic applications that can assist humans in everyday life.

Everybody with an autonomous robot can participate. The @Home league consists of some tests and an open challenge to demonstrate the abilities of the robot. To participate in the open challenge the robot has to participate in at least one test. The competition is in a real world scenario.

The competition of RoboCup@Home consists of tests which the robots have to solve. The tests will change over the years to become more advanced and function as an overall quality measurement in desired areas. Performance measurement is based on a score derived from competition rules and evaluation by a jury.

The tests should:
- include human machine interaction.
- be socially relevant.
- be application directed/oriented.
- be scientifically challenging.
- be easy to set up and low in costs.
- be simple and have self-explaining rules.
- be interesting to watch.
- take a small amount of time.

The ultimate scenario is the real world itself. To build up the required technologies gradually a basic home environment is provided as a general scenario, consisting of a living room and a kitchen.

Rules

For more information about the rules of this competition see Appendix II
ROBOCUP JUNIOR

Introduction

RoboCup is an international effort whose purpose is to foster Artificial Intelligence (AI) and robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. As well, the initiative serves as a basis for project-oriented education.

Objective

RoboCup Junior offers several challenges, each emphasizing both cooperative and competitive aspects. For children, the Junior initiative provides an exciting introduction to the field of robotics, a new way to develop technical abilities through hands-on experience with electronics, hardware and software, and a highly motivating opportunity to learn about teamwork while sharing technology with friends. In contrast to the one-child-one-computer scenario typically seen today, RoboCup Junior provides a unique opportunity for participants with a variety of interests and strengths to work together as a team to achieve a common goal.

RoboCup Junior aims at bringing together many of these ideas, promoting project-oriented, team-based education, giving children with a variety of interests and abilities an opportunity to pick their own challenges while contributing to the progress of the whole.
FIRA

Introduction

FIRA robot soccer, a brainchild of Prof. Jong-Hwan Kim, KAIST, began in 1995, the first international championship was held at KAIST, Daejeon, Korea in 1996. The Federation of International Robot-soccer Association (FIRA) was established on June 5, 1997 during the micro-robot soccer tournament (MiroSot'97) held at KAIST in Daejeon, Korea. FIRA Robot World Cup is held every year on global-scale. FIRA Robot World Congress is held along with FIRA Cup, where all participating teams submit and present papers on their robots and schemes for sharing the expertise and technology for building the soccer robots.

Micro-Robot World Soccer Tournament (MiroSot) initiative gives a good arena for multi-agent research, dealing with research subjects such as cooperation protocol by distributed control, effective communication and fault tolerance, while having efficiency of cooperation, adaptation, robustness and being in real-time.

With the ever increase in number of robots in an industrial environment, scientists/technologists were often faced with issues on cooperation and coordination among different robots and their self-governance in a workspace. This has led to the developments in multi-robot cooperative autonomous systems. The opponents of multi-robot autonomous systems needed a model to test the theories being proposed to test its efficacy and efficiency. It is not a surprise that they started focusing on robot soccer. Robot soccer makes heavy demands in all the key areas of robot technology, mechanics, sensors and intelligence. And it does so in a competitive setting that people around the world can understand and enjoy.

The Micro-Robot World Cup Soccer Tournament (MiroSot) thus was given birth, and a new interdisciplinary research area emerged, where scientists and technologists from diverse fields like, robotics, intelligent control, communication, computer technology, sensor technology, image processing, mechatronics, artificial life, etc., work together to make the multi-robot systems a reality. The robots used in MiroSot are small in size (7.5cm x 7.5cm x 7.5cm), fully/semi autonomous and without any human operators.

MiroSot involves multiple robots that need to collaborate in an adversarial environment to achieve specific objectives. In multi-robot systems, other robots in addition to the uncertainty that may be inherent in the domain, can determine the environment's dynamics. They have dynamic environments as other robots intentionally affect the environment in unpredictable ways. The key aspect being the need for robots not only to control themselves, but also to track and control the ball which is a passive part of the environment. The interesting theoretical issue behind MiroSot experiments is the use of soccer as a prototype example of a complex, adaptive system. MiroSot is a new interdisciplinary research area, where scientists and technologists from diverse fields like, robotics, intelligent control, communication, computer technology, sensor technology, image processing, mechatronics, artificial life, etc., can work together to make the multi-robot systems a reality.
**FIRA history**

1995
- International Organizing Committee: Micro-Robot World Cup Soccer Tournament, initiated by Jong-Hwan Kim, KAIST, Korea

1996
- August: International Summer Camp for Rule Meeting
- November 9-12: The 1st MiroSot'96 held at KAIST, Korea
- June 1-5: The 2nd MiroSot'97 held at KAIST, Korea
- June 5: Federation of International Robot-soccer Association (FIRA) was established.

1997
- August-September: MiroSot World Tour: led by Jong-Hwan Kim (Austria, Brazil, Canada, Germany, Italy, Mexico, Spain, U.K., and U.S.A.)
- September 19, 1997: BBC telecast MiroSot game in the famous Blue Peters program for children

1998
- June 29-July 3: 1998 FIRA Cup France at the La cite des Sciences et de l'Industrie, Paris, France

1999
- August 4-8: 1999 FIRA Cup Brazil at the Gymnasium of Colegio Notre Dame, Campinas, Brazil

2000
- September 18-24: 2000 FIRA Cup Australia, Rockhampton, Australia

2001
- August 1-5: 2001 FIRA Cup China at the Science and Technology Museum, Beijing, China
- May 23-29: 2002 FIRA Cup Korea, 6 FIFA World Cup Cities (The Preliminary Games: Busan, Daegu, Daejeon, Gwangju, and Suwon, The finals: SETEC, Seoul)
- May 26-28: FIRA Intelligent Robot Exhibition, SETEC, Seoul, Korea

2003
- Sept. 28-Oct. 3: 2003 FIRA Cup Austria, Vienna, Austria, Korea

2004
- Sept. 28-Oct. 3: 2004 FIRA Cup BEXCO, Busan, Korea

2005
- Dec. 12-14: 2005 FIRA Cup Singapore

2006
- June 30-July 3: 2006 FIRA Cup Dortmund, Germany

2007
- June 14-17: 2007 Fira Cup San Francisco, USA

**Introduction to the FIRA CUP**

Robot soccer can be portrayed as a competition of advanced robot technology within a confined space. It offers a challenging arena to the young generation and researchers working with autonomous mobile robotic systems. It is hoped that FIRA’s flagship event, called the FIRA Robot World Cup (or the FIRA Cup in short), which started in 1996, together with many other FIRA events, will help generate interests in robotics in the young minds.

Through these events, FIRA hopes to help them better understand and appreciate, with interests, the scientific concepts and technological developments involved. FIRA believes that some of these interests will fuel scientific and engineering skills that ultimately develop into research outcomes to serve mankind in a variety of ways.
Ever since its establishment, FIRA has had venues for its annual FIRA Cup in Australia, Brazil, China, France and Korea. Making progress over successive years since 1996, FIRA Cup has now attained world recognition as a robot festival.

**Objectives**

- To take the spirit of science and technology to the young generation and laymen.
- To promote the development of autonomous multi-agent robotic system that can cooperate with each other and to contribute to the state-of-the-art technology improvement in this specialized field.
- To bring together skilled researchers and students from different backgrounds such as robotics, sensor fusion, intelligent control, communication, image processing, mechatronics, computer technology, artificial life, etc. into a new and growing interdisciplinary field of intelligent autonomous soccer-robots to play the game of soccer.
- To organize the FIRA Robot World Cup and Congress every year.
- To work together to establish the FIRA Robot World Cup as a Science and Technology World Cup.

**FIRA Cup History**

1. **MiroSot '96**

   The very idea of Robot Soccer was originated in 1995 and in the month of September 1995 Professor Jong-Hwan Kim of KAIST, Korea, formally initiated an International Organizing Committee (IOC) for Micro-Robot World Cup Soccer Tournament (MiroSot). A summer camp (pre-meeting) on MiroSot was held in KAIST during July 29 - August 4, 1996, in which 30 teams from 13 countries attended. The MiroSot game rules were given a clear shape in the same meeting. The first MiroSot'96 was held in KAIST, between November 9 and 12, 1996. Twenty three (23) teams from 10 countries participated in the same.

2. **MiroSot '97**

   The second MiroSot was held at KAIST, during June 1-5, 1997 with 22 teams from 9 countries. Related research results on robot soccer are available from the workshop proceedings of MiroSot'96 and MiroSot'97 edited by Jong-Hwan Kim. The *Journal of Robotics and Autonomous Systems* published a special issue on MiroSot'97.

3. **1998 FIRA Cup France**

   FIRA hosted the FIRA Robot World Cup France'98 in Paris at the *La cite des Sciences et de l'Industrie*, Paris, between June 29 and July 3, 1998. Competitions were held in MiroSot, NaroSot (Nano Robot World Cup Soccer Tournament) and S-KheperaSot (Single-Khepera World Cup Soccer Tournament). The FIRA Robot World Cup France'98 witnessed the smallest robots ever to play the game of soccer between the SOTY and the BEST teams from Korea. It was the NaroSot category with robots sized 4 (cm) x 4 (cm) x 5.5 (cm). Teams from Denmark and USA competed for the S-KheperaSot. In the main competition of MiroSot there were 16 teams. Regional
Championships were held to select 16 teams in MiroSot and 4 teams each in NaroSot and S-KheperaSot for the Robot World Cup France'98.

FIRA'98 saw very promising developments in vision technology. The MiroSot world champion Keys developed and demonstrated the superiority of their vision card that works at a speed of 60 frames in a second with a capability to recognize 255 colours. Their robots were very fast, moving at a speed of 2m/s. From 1996, the year the first MiroSot competitions were held, the vision technology and robot speeds have improved on a tremendous level. In 1996, most of the teams used vision cards working at a rate of 10 frames/second. The robot speeds were as low as 50 cm/s then.

The competitions saw a sea change in motor technology and in the area of cooperative robotics as well. The outcome of the games clearly indicated the role played by the powerful motors in winning a game. The game strategies as well found different levels of development among the participants. The interesting point noted was the teams with the same level of competence (by way of vision card and motor speeds) did show better team strategies during a game. On the other hand, when two teams with different levels of competence were competing, due to the inability of the weaker team to move as fast as that of the opponent, resulted in collisions among robots at a higher rate. As years go by, it is expected to narrow down these differences in robot competence.

4. 1999 FIRA Cup Brazil

The FIRA Robot World Cup Brazil'99 was held from August 4 to 8, at the gymnasium of Campinas's most traditional school, Colegio Notre Dame. Fifteen teams representing six countries from four continents, selected through regional robot soccer competitions, participated in two categories of competitions: NaroSot and MiroSot. A scientific workshop was held on the evening of August 6th for the exchange and discussion of the scientific issues behind robot soccer and the applications derived from it. An estimated 3,000 people, including professionals from all areas of engineering and computer science, university professors, and graduate, undergraduate, and high school students, attended FIRA'99.

5. FIRA Benchmark Competition '99

For the first time, FIRA held a benchmark competition along FIRA'99 and participated four teams. FIRA was aimed to:

- Set a rigorous scientific standard for research into robot soccer;
- Encourage teams to work on the same problems to allow comparison;
- Collect and publish data on robot control and ball control;
- Enable scientific analysis of the performance of teams worldwide;
- Enable any particular team to gauge its performance against these standards;
- Provide a simple baseline from which new scientific benchmarks can be defined.

Three benchmarks were conducted at the FIRA'99:

- **Benchmark 1**: Ball striking: To control a single robot to move from a given initial position to strike a stationary ball. This benchmark runs three times, each time with a different initial position. The robot has 1 minute to complete the task.
- **Benchmark 2**: Goal scoring: To control a robot to move from a given initial position to strike a stationary ball and score a goal. This benchmark runs three times, each time with a different initial position. The robot has 1 minute to complete the task.
• **Benchmark 3:** Passing between players and shooting: To control two robots starting at given initial positions such that Robot 1 strikes the ball once and Robot 2 strikes the moving ball once to make the ball pass over the goal line. This benchmark runs twice, each time with a different initial position. The robots have unlimited time to complete the task.

6. **2000 FIRA Cup Australia**

   The FIRA’2000 was held in August 2000, in conjunction with the 2000 Olympics games, in Rockhampton, Australia. Four FIRA regional championships were organized to seed the teams for the FIRA’2000 event.

7. **2001 FIRA Cup China**

   The FIRA Robot World Cup 2001 was held at the Science and Technology Museum in Beijing, China from August 1-5, 2001 along with the FIRA Robot World Congress. 65 teams took part in 8 categories.

8. **2002 FIRA Cup Korea**

   2002 FIRA Robot Soccer World Championship (2002 FIRA Cup) took place in FIFA World Cup cities from May 23 to 29, 2002 one week before the 2002 FIFA World CupTM. Along with the 2002 FIRA Cup, 2002 FIRA Robot World Congress and FIRA Intelligent Robot Exhibition also took place.

   The 2002 FIRA Cup attracted an overwhelming participation, both locally and from overseas. A total of 207 teams from 25 countries, with over 600 professors, scientists, engineers and university students, participated in this grand event, making it the largest ever in its short history. The preliminary stages of MiroSot and SimuroSot were held at the 5 FIFA World Cup cities. The finals of MiroSot and SimuroSot, together with the other categories such as HuroSot, were held at SETEC, Seoul, from 26 May 2002 to 28 May 2002. HuroSot made its debut in this year’s FIRA Cup, and was an interesting highlight of the event.

9. **2003 FIRA Cup Austria**

   2003 FIRA Robot Soccer World Cup was held in Vienna, Austria from 28 September to 3 October, 2003. There were 6 different competition categories: HuroSot, KheperaSot, MiroSot (Small, Middle and Large League), NaroSot, RoboSot and SimuroSot (Large and Middle League). 107 Teams from 22 countries participated in this Cup.

   Parallel to the FIRA World Cup the FIRA World Congress was organized. On the congress the latest insights regarding mobile, cooperative, intelligent robots were introduced and discussed. Particular emphasis was given to the applications and development in the area of entertainment, education, service, personal robots.

10. **2004 FIRA Cup Korea**

    2004 FIRA Cup was held in Busan, Korea from October 27-31, 2004. There were 6 different competition categories: HuroSot, KheperaSot, MiroSot (Small, Middle and Large League), NaroSot, RoboSot and SimuroSot (Large and Middle League). 96 Teams from 21 countries participated in this Cup. As in the previous year parallel to the FIRA World Cup the FIRA World Congress was organized.
11. 2005 FIRA Cup Singapore
2005 FIRA Cup was held in Orchard Hotel, Singapore from December 12-14, 2005. There were 8 different competition categories: HuroSot, KheperaSot, MiroSot, NaroSot, and SimuroSot. 83 Teams from 16 countries participated in this Cup.

12. 2006 FIRA Cup Germany
2006 FIRA Cup was held in Dortmund, Germany from 30th June to 3rd July. There were 4 different competition categories: HuroSot, MiroSot, NaroSot, and SimuroSot. 47 Teams from 17 countries participated in this Cup.

13. 2007 FIRA Cup USA
FIRA RoboWorld Cup 2007 was held in San Francisco, USA along with the RoboGames from 14th June to 17th June 2007. There were new categories in Huro Cup to make FIRA RoboWorld Cup more exciting: HuroCup, Robo Marahtone, AndroSot, KheperaSot. So the competition categories were: HuroCup HuroSot, Robo Marahtone, AndroSot, KheperaSot, MiroSot (Large League and Middle League), NaroSot / - RoboSot, SimuroSot (Large League and Middle League).
In this year some rules were changed. The new rules can be downloaded from here: http://www.fira.net/soccer/robosot/RoboSot.pdf.

14. 2008 FIRA Cup China
The 13th FIRA Roboworld Cup China 2008 will take place in Qingdao from 22-25 July, 2008. More information can be found here: http://www.firachina.net/

15 2008 FIRA European Cup Linz
FIRA European Cup, EUROBY2008, will be held from June 19 to 22, 2008 in Linz. Further information can be found in the official website for EUROBY2008: http://www.euroby2008.at/

**Competition Categories**

The FIRA Cup robot-soccer event has well-defined game rules. It is organized into several categories, including the Micro-Robot Soccer Tournament (MiroSot), the Simulated Robot Soccer Tournament (SimuroSot) and the Humanoid Robot Soccer Tournament (HuroSot). These games are played under the watchful eyes of a human referee and the participants who are the robot players’ managers and trainers. In MiroSot, participants need to devise good strategies using artificial intelligence (AI) techniques, and develop sharp sensing and precise real-time control for the physical robot-soccer players. These basic capabilities are needed for the robot-soccer players to cooperate and coordinate autonomously (i.e., with “human hands-off”), and are crucial to winning the game against an opponent team. As many who have witnessed a MiroSot game will testify, the excitement always runs high especially when two strong robot-soccer teams meet. During the match, the robot players autonomously tackle many unfamiliar situations that arise due to the different strategies, hardware and control software technologies employed in the opponent robot players. Like in a FIFA World Cup soccer match, no one knows for sure which team will win until the final whistle. In SimuroSot, the game is played on a computer between two teams. With no physical robot involved, the game is decidedly one of complex strategy development using advanced AI techniques. In HuroSot, a robot player is more human-like in that it has two legs, hence the term humanoid. Given the current state of the art, the participants are only expected to endow...
their humanoid robot with, for instance, the ability to walk steadily, avoid obstacles simulating stationary opponent players and take penalty shots, all under the remote guidance of its human trainer.

In the following paragraphs the different competition categories are described:

- HuroCup
- KheperaSot
- MiroSot
- NaroSot
- AndroSot
- RoboSot
- SimuroSot

**FIRA Cup Categories**

**Humanoid Robot World Cup Soccer Tournament (HuroCup).**

The all-round competition is the most important HuroCup event as it tests the versatility of a humanoid robot. The humanoid robot requisites are:

- A humanoid robot shall have two legs.
- The maximum size of the robots is 150 cm.
- The maximum weight of the robot is 30 kg.
- Remote Control or Auto Control
- Pitch: 340~430 cm x 250~350cm

HuroSot attempts to encourage research into the many areas of humanoid robotics, especially walking and balancing, complex motion planning, and human robot interaction. The HuroSot competition also emphasizes the development of flexible, robust, and versatile domain. Robots must compete in several events (e.g., robot dash, obstacle run, penalty kicks, lift and carry, weight lifting, marathon and basketball).

**Objective**

The goal of the HuroSot league is to encourage research in practical, autonomous, highly mobile, flexible, and versatile robotic platforms. Intended applications for these robots are, among others, search and rescue robots, care robots, etc. As a benchmark problem, the goal of the HuroSot league is to develop humanoid robots that can perform several tasks in complex environments.

**Challenges**

To reduce the steep learning curve toward fully autonomous soccer, the rules committee has developed 3 new challenges: marathon, weight lifting and basketball that will be added to the last year four challenges for physical agents: (a) robot forward/backward dash, (b) penalty kick, (c) obstacle run, and (d) lift and carry. These challenges are aimed at providing intermediate goals on the path to fully autonomous robots that can operate in difficult environments.

**Forward - Backward Dash**

The robot dash challenge is a sprint event for humanoid robots. The goal is for the robots to move as quickly as possible from a start line to the end line for a series of segments.
Penalty Kick
In this challenge, the robot must approach and kick a ball positioned somewhere in the ball area. A robot from a different team will act as goal keeper during this event.

Obstacle Run
This challenge is similar to the robot dash challenge. The robot must move from one end of the playing field to the other as quickly as possible. However, in this case, a number of obstacles are distributed over the playing field. The robot must navigate around the obstacles and reach the end zone.

Lift and Carry
The goal is to provide an event that requires robots to use active balancing. The robots will be fitted with a small basket. The robots walk for a specified distance at which time the referee places small heavy obstacles into the basket. The robot must compensate for the extra weight and continues to walk. The robot that can carry the most weight is declared the winner of the event.

Marathon
The marathon is an endurance race over 42.195m. The robot must follow a coloured track.

Weight Lifting
The goal of this event is to develop robots that can lift and balance heavy weights.

Basketball
This competition is another single robot event at the moment, but will be expanded to multiple players in the future. The robot must throw a ball into a coloured target.

Soccer
The soccer competition is the first team event in the HuroCup competition. It is a game of soccer played by teams of 3 players.

Rules
For more information about the rules of this competition see Appendix III

KheperaSot
The KheperaSot game shall be played by two teams, each consisting of one robot player and up two human team members. The robot will be fully autonomous with on board vision system. The human member team will only be allowed to place their robot on the field, start their robot at the beginning of each round at the position indicated by the referee before each round, start their robot when indicated by the referee and remove the robot from the field at conclusion of the match.

The ball is a yellow tennis ball, and the pitch should be 130x90cm.

Rules
For more information about the rules of this competition see Appendix I
**Micro Robot World Cup Soccer Tournament (MiroSot)**

A match shall be played by two teams, each consisting of three robots, one of which can be the goalkeeper. Three human team members, a "manager", a "coach" and a "trainer" shall, only be allowed on the stage. One host computer per team, mainly dedicated to vision processing and other location identifying, shall be used.

The size of each robot shall be limited to 7.5 cm x 7.5 cm x 7.5 cm. The height of the antenna shall not be considered in deciding a robot's size.

- Robot: 7.5 cm x 7.5 cm x 7.5 cm.
- Ball: an orange golf ball.
- Pitch: 150cm x 130cm for Small League and 220cm x 180cm for Middle League.

**Rules**

For more information about the rules of this competition see Appendix I

---

**NaroSot**

A match shall be played by two teams, each consisting of five robots, one of which can be the goalkeeper. Three human team members, a "manager", a "coach" and a "trainer" shall, only be allowed on the stage. One host computer per team, mainly dedicated to vision processing and other location identifying, shall be used.

The size of each robot shall be limited to 4 cm x 4 cm x 5.5 cm. The height of the antenna shall not be considered in deciding a robot's size.

- Robot: 4cm x 4cm x 5.5cm
- Ball: An orange ping-pong ball
- Pitch: 130cm x 90cm

**Rules**

For more information about the rules of this competition see Appendix I

---

**AndroSot**

A match shall be played by two teams, each consisting of three robots. One of the robots can be the goalkeeper. Each of the three human team members shall control one of the three robots.

- Robot Weight Range: 600g.
- Robot Dimensions: up to 50 cm.
- Pitch: 260cm x 220cm
- Robot Control Specifications: Remote Control

**Rules**

For more information about the rules of this competition see Appendix I
**RoboSot**

A match shall be played by two teams, each consisting of one to three robots. One of the robots can be the goalkeeper. Three human team members, a "manager", a "coach" and a "trainer" shall only be allowed on stage. The robots can be fully or semi-autonomous. In the semi-autonomous case, a host computer can be used to process the vision information from the cameras on-board the robots.

- Robot: 20cm x 20cm x No limit in height
- Ball: A yellow with light green tennis ball
- Pitch: 260cm x 220cm

**Rules**

For more information about the rules of this competition see Appendix I

**SimuroSot**

SimuroSot consists of a server which has the soccer game environments (playground, robots, score board, etc.) and two client programs with the game strategies. A 3D colour graphic screen displays the match. Teams can make their own strategies and compete with each other without hardware. The 3D simulation platform for 5 vs. 5 and 11 vs. 11 games are available at FIRA website.

**Rules**

For more information about the rules of this competition see Appendix I

**Participation**

There are participants from all around the world (Indonesia, Ecuador, South Africa, Spain...).
With the Grand Challenge the U.S. Defense Advanced Research Projects Agency (DARPA) continues with its policy of pursuing research and technology where the risk and payoff are both very high and where success may provide dramatic advances for traditional military roles and missions.

In the words of DARPA Director Anthony Tether; “DARPA has always attracted people who can look at problems differently and find creative ways to solve them, the Grand Challenge creates a way for us to reach out and find people who will help us advance the development of autonomous robotic ground vehicle technology”.

The DARPA Grand Challenge is a field test that requires autonomous robotic ground vehicles to successfully navigate a course: "It was an important step to have autonomous ground vehicles that can navigate and drive across open and difficult terrain from city to city. But the next big leap will be an autonomous vehicle that can navigate and operate in traffic, a far more complex challenge for a 'robotic' driver. So we are very excited to be moving from the desert to the city with our Urban Challenge".
The course will be nominally 60 miles in total distance, with a time objective of 6 hours. Prior to the Grand Challenge main event, there is a qualification, inspection and demonstration (QID) event. Teams undergo a series of tests to determine the ability of the systems to autonomously navigate and avoid obstacles, as well as thorough inspections to ensure that they meet safety and performance requirements. At the conclusion of the QID event DARPA announces the final field of teams that will compete in the Grand Challenge. DARPA awards a cash prize of $2 million to the winner - the team that most quickly completes the course in less than the 6-hour time limit. If no teams finish, no prize is awarded.

Objective

The DARPA Grand Challenge was created in response to a Congressional and DoD mandate, DARPA Grand Challenge is a field test intended to accelerate research and development in autonomous ground vehicles and promote innovative technical approaches that will enable the autonomous operation of unmanned ground combat vehicles (National Defense Authorization Act for Fiscal Year 2001, Congress mandated in Section 220 that “It shall be a goal of the Armed Forces to achieve the fielding of unmanned, remotely controlled technology such that... by 2015, one-third of the operational ground combat vehicles are unmanned.”). These autonomous ground vehicles will navigate from point to point in an intelligent manner to avoid or accommodate obstacles including nearby vehicles and other impediments. The DARPA Grand Challenge is a field test of autonomous ground vehicles over realistic terrain and sets specific performance goals for distance and speed. It draws widespread attention to the technology issues associated with autonomous vehicles and motivates entrants to overcome the obstacles to the realization of truly robust autonomous ground vehicles. The event challenges the most capable and innovative companies, institutions, and entrepreneurs in the United States and from around the world to produce breakthroughs in capability and performance.

The Grand Challenge brings together individuals and organizations from industry, the R&D community, government, the armed services, academia, students, backyard inventors, and automotive enthusiasts in the pursuit of a technological challenge. DARPA’s mission is to leverage ingenuity and research to develop transformational technologies that give armed forces a decisive edge.

The objective of this program is safe and correct autonomous driving capability in traffic at 20 mph. To do this, vehicles will demonstrate the following capabilities:

- Complete a mission defined by an ordered series of checkpoints in a complex route network. The vehicle will have 5 minutes to process a mission description before attempting the course.
- Interpret static lane markings (e.g., white and yellow lines) provided with the route network definition file and behave in accordance with applicable traffic laws and conventions. DARPA’s intent is for the RNDF lane boundary descriptors to match the physical lane markings on the ground. DARPA cannot ensure that this will be the case in all areas, and as such the RNDF shall take precedence over the physical ground markings in conflicting areas.
- Exhibit context-dependent speed control to ensure safe operation, including adherence to speed limits.
• Exhibit safe-following behaviour when approaching other vehicles from behind in a traffic lane. This includes maintaining a safe-following distance.
• Exhibit safe check-and-go behaviour when pulling around a stopped vehicle, pulling out of parking spot, moving through intersections, and in situations where collision is possible.
• Stay on the road and in a legal and appropriate travel lane while en route, including around sharp turns, through intersections, and while passing. The route network definition file will specify the GPS coordinates of the stop signs. The RNDF specifies the location of stop lines on the ground. On paved areas, each stop line will be represented by a painted stop line on the pavement. Physical stop signs, however, may or may not be present at the stop line locations.
• Navigate safely in areas where GPS signals are partially or entirely blocked.
• Follow paved and unpaved roads and stay in lane with very sparse or low accuracy GPS waypoints.
• Change lanes safely when legal and appropriate, such as when passing a vehicle or entering an opposing traffic lane to pass a stopped vehicle. Vehicles must not pass other vehicles queued at an intersection.
• Merge safely with traffic moving in one or more lanes after stopping at an intersection.
• Pull across one lane of moving traffic to merge with moving traffic in the opposing lane.
• Stop safely within 1 meter of the stop line at a stop sign intersection and proceed without excessive delay (less than 10 seconds) according to intersection precedence rules.
• Exhibit proper queue behaviour at an intersection, including stopping at a safe distance from other vehicles and stop-and-go procession to the stop line without excessive delay.
• Navigate toward a destination in a large, open area where minimal or no GPS points are provided, as in loading dock areas or parking lots. These areas may contain fixed obstacles such as parked vehicles and moving obstacles including other vehicles.
• Safely pull into and back out of a specified parking space in a parking lot.
• Safely execute one or more three-point turning manoeuvres to do an U-turn.
• Dynamically re-plan and execute the route to a destination if the primary route is blocked or impassable.

The following behaviours or capabilities are outside the scope of this program:
• Recognition of external traffic signals such as traffic lights and stop signs, through the use of sensors.
• Recognition of pedestrians and pedestrian avoidance.
• Behaviours necessary for highway driving such as high speed passing or high speed merge at an onramp. Speed limits for the Urban Challenge will be 30 mph or less.
• Driving in difficult off-road terrain is outside the scope of the program. Off-road navigation in an unpaved area, travel along roads with potholes, and travel along a dirt road are within scope.
**History**

**Grand Challenge 2004**

The Grand Challenge 2004 field test of autonomous ground vehicles ran from Barstow, California to Primm, Nevada offered a $1 million prize. From the qualifying round at the California Speedway, 15 finalists emerged to attempt the Grand Challenge. However, the prize went unclaimed as no vehicles were able to complete the difficult desert route.

**Grand Challenge 2005**

The Grand Challenge 2005 was held on October 8, 2005 in the desert Southwest. The Stanford Racing Team won the $2 million prize with the winning time of 6 hours, 53 minutes. A total of five teams completed the Grand Challenge course which was 132 miles over desert terrain.

**Participants**

The first official public event for the Grand Challenge came on February 22, 2003, when nearly 500 prospective participants gathered at the Petersen Automotive Museum in Los Angeles for a competitor’s conference. Later on, 106 teams submitted applications expressing their interest. Out of that initial group of applicants, 86 submitted technical papers by the October 14, 2003 deadline. After a rigorous evaluation of technical papers and select site visits, DARPA selected the final field of 25 teams. The Participants Conference was held in Anaheim, California on August 14, 2004.

In the Grand Challenge 2005, over 550 individuals from 42 states and 7 countries reviewed the preliminary rules, timetable for deadlines, and the qualification process. At the end, there were 23 finalists among 195 teams from 36 states and four foreign countries. These teams advanced to the final event by series of rigorous tests designed to assess their capability of completing the desert course. Four of them completed the course before the 10 hour limit. The first four robots entered the history books as being the first ground vehicle robots to travel a great distance at a relatively high speed within a specified time frame. The vehicle that completed the course in the shortest time was “Stanley” (Stanford racing team), with an average speed over the 131.6 mile desert course of 19.1mph.

**Rules**

For more information about the rules of this competition see Appendix IV.
DARPA URBAN CHALLENGE

The DARPA Urban Challenge is an autonomous vehicle research and development program with the goal of developing technology that will keep warfighters off the battlefield and out of harm’s way. The Urban Challenge features autonomous ground vehicles manoeuvring in a mock city environment, executing simulated military supply missions while merging into moving traffic, navigating traffic circles, negotiating busy intersections, and avoiding obstacles. This program is an outgrowth of two previous DARPA Grand Challenge autonomous vehicle competitions.

The program is conducted as a series of qualification steps leading to a competitive final event, scheduled to take place on November 3, 2007, in Victorville, California. DARPA is offering $2M for the fastest qualifying vehicle, and $1M and $500,000 for second and third place.

Teams from around the world were whittled down through a series of qualifying steps, beginning with technical papers and videos, then advancing to actual vehicle testing at team sites. Of the 89 teams to initially apply, 35 teams were invited to the National Qualification Event (NQE), a rigorous eight-day vehicle testing period. The NQE was co-located with the Final Event in Victorville, CA. DARPA transformed the roads of the former George AFB into an autonomous vehicle testing ground, laying over four miles of protective k-rail barriers in creating multiple test courses.

What is an autonomous ground vehicle?

An autonomous ground vehicle is a vehicle that navigates and drives entirely on its own with no human driver and no remote control. Through the use of various sensors and positioning systems, the vehicle determines all the characteristics of its environment required to enable it to carry out the task it has been assigned.

National Qualification Event

The NQE for the Urban Challenge was divided into three separate test areas, each with its own flavor and set of challenges:

- The NQE A test course required robots to safely merge into and out of two-way traffic in a tight, circulating course. Needless to say, this led to some hair-raising moments for some of the traffic drivers. Besides the complex timing and scoring being recorded by course officials, traffic drivers would alert officials to aggressive behaviour with an ever-popular horn blast. Amazingly, in eight days of testing, only one traffic vehicle was actually struck by a robotic vehicle, a testament to the progress of the teams and DARPA’s focus on safety.

- The meandering NQE B course tested robots on their ability to stay within a lane as they traversed this 2.8-mile course. One section, affectionately termed “The Gauntlet” required the robots to delicately manoeuvre through a series of parallel parked cars and road obstacles. A final test on the NQE B course required the robots to find an assigned parking spot between adjacent parked cars, then safely pull into and back out of the spot before proceeding on its mission.
• NQE C was traffic intensive, consisting of a series of four-way stop intersections for the robot to negotiate, each with its own arrangement of traffic. Robots had to recognize the other vehicles at these intersections, determine the order of precedence and then safely proceed through the intersection when it was their turn. For the second half of the NQE C course, various road blocks were emplaced and the robots were tested on their ability to recognize the road block, execute a U-turn and dynamically replan a new route to complete their mission.

Final Event
The course for the final event was communicated to the teams in the form of two files, analogous to a map and a specific mission. Upon announcing the finalist selections on November 1, teams were given the ‘map’ file of the final course (Route Network Definition File). However, each team didn’t receive their Mission Definition File, which lists the order of checkpoints they had to visit, until five minutes before they launched on race day. With this approach, the teams had no a priori knowledge of their missions, creating a truly autonomous driving test.

Awards Ceremony
This event was not just a timed race however – robots were also being judged on their ability to follow California driving rules. DARPA officials pored through reams of data throughout the night, analyzing each team’s infractions and elapsed run times.
At the awards ceremony the next morning, DARPA announced the winning order.

History
Urban Challenge 2007
The DARPA Urban Challenge was held on November 3, 2007, at the former George AFB in Victorville, Calif. Building on the success of the 2004 and 2005 Grand Challenges, this event required teams to build an autonomous vehicle capable of driving in traffic, performing complex manoeuvres such as merging, passing, parking and negotiating intersections. This event was truly groundbreaking as the first time autonomous vehicles have interacted with both manned and unmanned vehicle traffic in an urban environment. The winning order was: Tartan Racing (Pittsburgh, PA), Stanford Racing Team (Stanford, CA), Victor Tango (Blacksburg, VA).

The route
The road surface will range in quality from new pavement to potholes and broken pavement. Sections of dirt roads with low berms may also be encountered. The vehicle may negotiate sharp curbs, downed branches, traffic barrels, drains, hydrants, rocks, brush, construction equipment, concrete safety rails, power line poles, and other stationary items likely to be found in an urban environment. Vehicles will obey traffic laws as they negotiate traffic circles, intersections, and merge with moving traffic. Traffic on the route may be provided by manned vehicles, tele-operated vehicles, and other autonomous vehicles. Static vehicles may also be parked or stopped along the route.
EUROBOT

Introduction

Eurobot is an international amateur robotics contest open to teams of young people, organized either in student projects or in independent clubs. A team is composed of several people joining around a common project. Its rules are renewed each year.

Encouraging the practice of science in a friendly spirit, EUROBOT values are fair play, solidarity, technical knowledge sharing as well as creativity, both through techniques and project management more than competition. The contest aims at interesting the largest public to robotics and at encouraging the group practice of science by youth.

EUROBOT and its national qualifications are intended to take place in a friendly and sporting spirit. Thus, more than an engineering championship for young people, EUROBOT is a friendly pretext to technical imagination and exchange ideas, knowhow, hints and engineering knowledge around a common challenge. Creativity is at stake and interdisciplinarity requested. Technical and cultural enrichment is the goal. To conclude, a key goal of the organization is: “that everyone wins in knowledge, experience and friendship, those who are awarded of course, but also those who didn’t manage to make their robot work properly and were eliminated during the matches”.

Photo by: Enric Cervera at Eurobot’06 (Catania, Italy, 2 June 2006)
Objective

The object of Eurobot is to foster and develop interest in robotics in young people on an international scale.

It is proposed that the object includes the following:

- first and foremost, to develop activities Europe-wide.
- to promote and organize national qualifying rounds and international finals for the Eurobot robotics competition, with the help of the members.
- to provide assistance to members in setting-up and running projects relating to robotics competitions.
- to train new members on how to organize Eurobot competitions and to share experience gained on the ground with everyone.
- to be associated with demonstrations and events enabling clubs to display their robots, and thus contribute to spreading culture of science and technology, in particular for young people.
- to ensure quality and consistency of the Eurobot competition and label image at international level, insofar as its beneficiaries, partners, supporters and the media are concerned.
- generally speaking, to carry out all tasks necessary for the development of its activities and the dissemination of its expertise both within Europe and other countries, together with its interests and those of its object.

History

Eurobot 2005: Bowling

Objective

The team which will count the largest number of skittles of its colour laid down at the end of the match will be the winner.

It seems easy and quick to lay down its own skittles; however this is also possible to set up the ones of the opponent. With one or two robots per team, this is probably where the challenge is… Which strategy should the robot choose?

Date and Place

The Final of EUROBOT 2005 was held from Friday the 20th of May to Sunday the 22nd of May 2005 in Switzerland at Yverdon-les-Bains.
Final Ranking of Eurobot 2005

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Microb Technology</td>
</tr>
<tr>
<td>2</td>
<td>R-TEAM</td>
</tr>
<tr>
<td>3</td>
<td>RCVA</td>
</tr>
</tbody>
</table>

Eurobot 2006: Funny Golf

Objective
Two robots on a golf course!
The team which inserts the whitest balls in his holes at the end of the match will be the winner.
But be careful, it is also possible to insert black balls in the opponent’s holes to prevent him from scoring!

Date and Place
The Final of EUROBOT 2006 was held from June the 1st to June the 3rd 2006 in Catania, Italy.

Final Ranking of Eurobot 2006

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RCVA</td>
<td>France</td>
</tr>
<tr>
<td>2</td>
<td>TeamDare</td>
<td>Netherlands</td>
</tr>
<tr>
<td>3</td>
<td>BVP-M86</td>
<td>Serbia</td>
</tr>
</tbody>
</table>

Eurobot 2007: Robot Recycling Rally
**Objective**
Waste, Sort & Win! The robot which sorts the most waste into correct bins will be the winner.

**Date and Place**
The Final of EUROBOT 2007 was held from May the 16th to May the 20th 2007, in La Ferté-Bernard, France.

**Final Ranking of Eurobot 2007**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RCVA</td>
<td>France</td>
</tr>
<tr>
<td>2</td>
<td>Turag</td>
<td>Germany</td>
</tr>
<tr>
<td>3</td>
<td>ClubElek</td>
<td>France</td>
</tr>
</tbody>
</table>

**Eurobot 2008: Mission to Mars**

**Objective**
Find proofs of life and bring them to Earth... For analyse! The robot which will bring back to Earth the most living organisms in good conditions will be the winner.

**Date and Place**
The Final of EUROBOT 2008 was held from May the 21st to May the 25th 2008 in Heidelberg, Germany.

**Final Ranking of Eurobot 2008**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RCVA</td>
<td>France</td>
</tr>
<tr>
<td>2</td>
<td>Helb - inraci</td>
<td>Belgium</td>
</tr>
<tr>
<td>3</td>
<td>ISTIA - IUT d'Angers</td>
<td>France</td>
</tr>
</tbody>
</table>
FIRST INTERNATIONAL CLEANING ROBOTS CONTEST

Introduction

The contest took place on October 1-3, 2002, in Lausanne, Switzerland at the Swiss Federal Institute of Technology Lausanne (EPFL) jointly with IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS 2002). There were three categories: floor cleaning, window cleaning and ideas for housekeeping robots.

The general objective of this contest was to bring together representatives from the cleaning industry and appliance manufacturers as well as young researchers, future customers and users, and draw their attention to an emerging technology. Unfortunately, this event was discontinued in subsequent IROS conferences.

The goal of the floor cleaning category was to clean within 10 minutes 5x5m room as good as possible. The floor was covered with sugar. The robots competed in two rounds. The first qualification round consisted of matches within each of four randomly selected groups. The second round was finals of all winners from the first round. A total of 12 teams have participated.

The scoring was based on the area cleaned (number of cleaned tiles 50x50cm) multiplied by various "handicap factors". These factors promoted cheap sensors and self-designed robots and cleaning systems. There were also penalties for hitting the doll, which represented a baby lying on the floor and for restarting the robots (which got stuck in a corner or had some other problems).

Objectives

The event was a showcase for novel technologies and a catalyst for technology transfer. It helped to attract research potential and brain power to economically interesting applications.

For industrial sponsors, the event provided an excellent opportunity to access the creativeness, enthusiasm and technical ingenuity of young scientists and to gain their attention.

In parallel with the contest was an exhibition of industrial robotic cleaning technology. The contest, together with the exhibition, was an excellent way of introducing the general public to this new technology and its applications in a relaxed atmosphere, and without bias in favour of any particular approach.

Berta was the winner of the Cleaning contest 2002. It beat its competitors with 2D range finders, cameras and a lot of processing power using only a front bumper and a microcontroller.
BERTA

The bogie was made of aluminum U-shaped profiles, powered by two DC motors (1.5Amp/12VDC, 5500rpm with 20:1 gearbox). The brain of the robot was 16bit Hitachi microcontroller (14.745 MHZ clock, 4kB RAM, 128kB ROM (flash)). The only sensor that this robot utilizes was its front bumper. The cleaning method was based on a random walk algorithm.

Mechanical Design

The bogie was made of U-shaped aluminum profiles. Aluminum has proven itself to be a relatively light and quite durable material for such a construction. The bogie was a triangle with two geared motors mounted in the base, which was also the front of the robot. The opposite vertex provided support for a caster wheel. The motors were 1.5Amp/12VDC, 5500rpm with 20:1 gearbox (dimensions: 60 x 33 x 33mm).

The cleaning unit was a plastic box made with a hot-glue. The box was removable and it was attached to the robot by aluminum holder. Dry vacuuming was used for collection of dirt. The cleaning mechanism consisted of 2 salvaged 12V car vacuum cleaners and a box for collecting the dirt, equipped with 2 nozzles. An additional brush helped to collect dirt before it was sucked into the box.

The power source for all motors and electronics was 12V 7AH/20AH sealed rechargeable lead battery.

Both motors were powered by dual H-bridge and controlled via PWM (Pulse Width Modulation) directly from the main microcontroller. A feedback was provided by two encoders taken from a regular computer mouse. A non-linear PID controller was used for maintaining speed and direction.

Onboard Computer

Single microcontroller Hitachi H8/3048 - 14.745 MHz clock, 4kB RAM, 128kB ROM (flash). During software development was found, that it was more convenient to drive robot as a slave from a PC through a serial interface. The H8 processor was programmed to fulfill the simplest tasks only and the program itself ran in the PC. The PC allowed precise logging and easier changes in program. The same program was programmed to the H8 chip only after the software development was finished.

Sensor System

There are only two means of sensing the environment for the robot:

- bumper micro-switches
- virtual bumper based on correlation between applied power to the motors and resulting speed measured by encoders

The switches are very reliable and do not need any filtering. There are scanned every millisecond in the main processing loop where the change of state can be directly triggered.

The virtual bumper exploits the fact that the motors are not that powerful. If the maximum power (100% duty cycle of PWM) is applied to the motors for 1.5 second and robot is not moving then collision is registered. Again, this detection has proven to be quite reliable given the combination of the robot weight and motor power. From the view of the higher control software collision detected by the virtual bumper is indistinguishable from regular collision triggered by a micro-switch.
Obstacle Avoidance and Navigation Strategy

The obstacle avoidance is very simple, because a random walk (our coverage principle) does not require precise realization of the plan. When collision occurs robot tries to rollback its last action. If it collides while moving forward, it backs up 25cm and turns left or right for given value. The values are precomputed and stored in a table. If it collides while turning, it returns to its original position (the robot always turns on a spot), and backs up another 25cm. The last case, when the robot collides while reversing is not properly handled. It tries to turn, but better solution would be (based on experience in Lausanne) move forward say 10cm, and try to turn some smaller "escape" angle.

Coverage Principle

The presented strategy during the cleaning contest was a "random" walk. The rules were created by random generator, but for actual run they were coded into a table. Each command was defined distance how far should robot move straight and how much it should turn (with a sign determining direction left/right). This strategy was chosen for its simplicity and robustness (taken into account the environment and sensors used).

Performance Data

The performance was measured separately for cleaning, obstacle avoidance and for coverage. The test track for cleaning was on linoleum floor covered with confetti and sugar. The nozzles and the brush were then adjusted according to the immediate results. The obstacle avoidance was tested in normal office environment: chairs, tables, cables on the floor. It showed that simple bar with two micro-switches as a sensitive bumper is quite sufficient for given task. The maximum speed and position of the bumper was tuned during these experiments. The coverage was tested only under simulation. The main control loop was called 600000 times (10 minutes, 1ms cycle), and coverage ratio was based on the number of squares visited by robot. This was not perfect evaluation, but it was a good indicator if the strategy is getting better or worse.
Introduction

The European Land-Robot Trial (ELROB) is the first European robot trial providing an opportunity to demonstrate today’s state-of-the-art robotics. Two scenarios, focusing on both mobility and RSTA (Reconnaissance, Surveillance, and Target acquisition), allow participating teams to demonstrate their technical realizations of Unmanned Ground Vehicles (UGV). Both commercial and academic applicants from European countries are allowed to participate in the trial. The first instance of this annual event was organized by the German Federal Armed Forces (Bundeswehr).

Due to the reorientation of the Bundeswehr with an extended spectrum of tasks in conflict prevention and crisis management including the fight against international terrorism, the armed forces are heading for new demands. This holds also for most of the other European forces.

Robotics is one option for the intelligent substitution of personnel on highly dangerous and tedious operations. Such unmanned systems allow considerable improvement in the protection of soldiers. This applies to reconnaissance and combat operations as well as to handling or manipulating hazardous materials (e.g. NBC, mine detection). The employment of robot systems on the ground is imperative for protracted activities and/or for activities under threat.

Unmanned systems enable the soldier to enhance his protection considerably by substantially increasing the distance between him and the scene of operation. Furthermore, the employment of this technology allows accommodating the limited funds in the military sector and the increased need for opportunities of personnel cutbacks.

Against the background of this fundamental new situation and the permanent requirement for more economy of manpower and funds, the Bundeswehr and other European forces are consolidating and realigning their R&T activities in the area of robotics.

Basic Rules

Since ELROB is considered as a capability demonstration, there were only a few technical limitations implied on the participating teams, allowing a wide variety of possible solutions. There is, however, a predefined set of rules providing comparability among the participants, which is enforced by a team of seven judges:

- The vehicle must travel using traction with the ground.
- The vehicle’s maximum weight is limited to 3 tons.
- Participating vehicles must be unmanned.
- Only open to teams under European leadership
Following the concerns of the industry participants, no contest was conducted, and no official evaluation or ranking will be published. Thereby the overall goal of ELROB was to present a comprehensive overview about current developments and possibilities for the use of robotic capabilities in the context of military or civilian operations.

**Basic Scenarios**

The two scenarios are selected and constructed in order to provide a realistic training ground for real-world tasks, emphasizing military applications like exploration and surveillance. While the team controlling the robot is not allowed to enter the test site, all scenarios require remote or autonomous operation of the robot.

*Urban scenario*

The terrain includes urban obstacles like cars, stairs, narrow passes, and collapsed ceilings. The task is to search for and identify objects located around and inside buildings. The maximum distance for the urban scenario is approximately 500 metres.

*Non-Urban Scenario*

The terrain contains both paved and unpaved roads as well as ditches, fences and fire. Both natural objects like trees and stones, as well as artificially placed objects may block the path and have to be circumnavigated. The task is to follow a predefined route and to locate targets at certain points of interest. The maximum distance for the non-urban scenario is approximately 2000 meters.

The maximum operational area is not larger than 10x30 meters. The area may include paved regions, unpaved regions, trails, and off-road desert areas. In addition to the existing natural obstacles, the organizers might place obstacles (e.g. military equipment) in the operational area that may disable a vehicle if struck. These obstacles must be detected and circumnavigated by a vehicle to successfully complete the route.
More information can be found at the official sites:

- Official ELROB home page: http://www.elrob.org/
- Official Civilian ELROB home page: http://www.c-elrob.eu/
- Official Military ELROB home page: http://www.m-elrob.eu/

## History

### 1st European Land-Robot Trial 2006 (Military-ELROB)

The 1st ELROB was held from 15th to 18th of May 2006 in Hammelburg, (Germany) and participated 20 teams from 5 European countries, who gave an impressive performance for the spectators coming from 19 countries all over the world. Following the concerns of the industry participants, no contest was conducted, and no official evaluation or ranking will be published.

The scenarios were:

1. Non-urban: mobility in non-urban terrain.
2. Urban: tactical awareness in urban environment.
3. EOD/IED: Detection and removal of EOD/IED in urban terrain
4. EOD/UXO: UXO detection in non-urban terrain

The following issues with room for improvement have been identified:

- Ergonomics of the human-machine interface
- Communication in urban and non-urban domain under difficult conditions
- Mobility in non-urban terrain
- Agility in narrow urban structures
- Navigation and manoeuvring under difficult conditions
- Demonstration of stair-climbing capability
- Use of elevatable manipulators
- Movement and interaction inside buildings
- Recognition and circumnavigation of obstacles
- Moving on pathless terrain
- Manoeuvring at high inclination angles
- Communication and navigation without sight

### 2nd European Land-Robot Trial 2007 (Civilian-ELROB)

The 2nd ELROB was held from 13th to 16th of August 2007 in Monte Ceneri (Ticino, Switzerland) and participated 13 teams from 5 countries.

The scenarios were:

1. Non-urban: an NBC incident, search area for hazardous materials.
2. Urban: an incident with IEDs on a crowded marketplace, improve situation awareness.
3. Combined operation of UGV and UAV: fire and NBC incidents, search area for suspicious places.
4. Autonomous reconnaissance on urban and/or non-urban route: security patrol, do reconnaissance.

### 3rd European Land-Robot Trial 2008 (Military-ELROB)

The 3rd ELROB will be held from 30th of June to 3rd of July 2008 in Hammelburg (Germany). The scenarios will be similar to the Elrob 2006.
ICRA ROBOT CHALLENGE

Introduction

The ICRA 2008 was held in Pasadena, California, on May 19-23, 2008. A new robot event was introduced at the IEEE International Conference on Robotics and Automation, starting at ICRA 2008. The event consisted of three specific challenges with the overall theme of "space robotics". The main goal of the Challenge, however, was to showcase current research being done in all of the disciplines represented at ICRA and, over the coming years, to benchmark the progress that robotics, as a field, is making on real, hard, relevant problems.

First trial: The Sandbox

This event simulates the exploration of a small area of a planetary surface. There are a number of sub-challenges in the event, and teams should feel free to attempt as many or as few of these as they want. This event is intended to showcase autonomous systems that operate with a minimum of human intervention.

The Environment

The environment for this event is an area approximately 6m by 6m, covered in gravel to a depth of 10cm. The gravel will most likely be somewhere between "pea gravel" and 1/4 inch pieces. Rocks of varying sizes and compositions will be placed on the surface. A simulated lander will be at one end, with one or more ramps extending down to the surface. Robots will ideally start and end all missions on the lander platform. The ramps will be clearly marked with brightly-colored tape, and at least one of them will be free of obstacles. The surface sand will be smoothed between runs, removing all traces of previous robots.

The Event Elements

Onto the surface.

The robot must leave the lander platform, and make it onto the planetary surface. The robot must identify the ramps off of the platform, orient itself appropriately, and navigate down one of the ramps to the surface. The ramps might be obstructed by deflated "air bags", or might not have deployed properly. The system should recognize and deal with these situations appropriately. The operation will be considered successful if the robot manages to get two meters (measured at the closest point) away from the lander. The lander platform will be constructed to autonomously open (perhaps with some human intervention), and to include potential obstacles on the ramps. The event might also be set up to include obstacles at the bottom of the ramps that are either impassible, or must be navigated around.
Data collection.

The robot must autonomously navigate to a remote science station, at the other side of the planetary surface to collect a set of recorded science data, and then return to the lander. The science station and robot are only equipped with short range (simulated) communications, so the robot must be physically close to the science station or lander in order to transfer data. The robot will be given the location of the science station, in some coordinate frame, but will not be given a map of the environment. The goal is to retrieve the data and bring it back to the lander as quickly as possible.

Map the Environment.

The robot should build an accurate metric map of the environment in a fixed time period. The time period starts when the robot leaves the lander ramp, and the map must be built by the end of the time period. Building a map in the sandbox will be particularly difficult, since there will be a lot of wheel slip, and the surface is not planar. We will purposefully put rises and dips into the sand to make mapping challenging.

Extreme navigation.

Part of the sandbox will contain a number of extremely challenging physical obstacles, and a near-vertical cliff-face. Robots must traverse these obstacles to reach the bottom of the cliff, climb the cliff to the top (if possible), and then return to the lander.

Find the robot.

A previous mission has gone wrong, and one of our robots is missing. The goal is to find this robot, following it's tracks in the sand, and to bring it back to the lander. For robots that are incapable of picking up another robot, finding the other robot and touching it will be enough. To add difficulty to this problem, the lost robot can be made small, so that the finding robot cannot just follow the tracks (through narrow spaces), but must plan paths around obstacles, and find the trail again.

Bring back shiny things.

The robot must go out and find interesting objects, and bring them back to the lander. These objects could be unusual rocks, differently colored sand or gravel, alien plants, or man-made artifacts. We could even have the robot look for liquid water or buried treasure.

Back on the lander.

At the end of the mission, the robot must return to the lander platform.
Second trial:
The Robotic Planetary Contingency

This event simulates an unexpected problem occurring at a planetary habitat, where a robotic solution must be quickly developed and deployed, using only existing resources. The intent of this event is to develop versatile robotic systems and software that can be adapted quickly to address unexpected events. Since humans are present, a natural solution to realistic unexpected events would exploit human creativity and human-robot interaction.

The competition drives not only the development of versatile robotic hardware and onboard software, but also the design and development of programming and assembly tools capable of rapidly implementing a wide variety of capabilities. Since teleoperation is not precluded for this event, the development of effective user interfaces is another expected outcome.

The Environment and Event Parameters

The environment for this event will consist of two areas: the planetary surface and the habitat. The planetary surface will have the same specifications as for the Sandbox event. The habitat will represent the human-occupied structure from which the robots will be "launched" onto the planetary surface. In all scenarios, the human participants cannot exit the habitat. Robots must be placed in an airlock chamber and drive (or be driven) out onto the planetary surface. If a robot needs to return to the habitat, it must do so through the airlock chamber. The airlock will have a pair of sealing doors, making autonomy or wireless teleoperation the only options for robot control. The airlock will be 1.5m long, 1m wide, and 1m tall, with 1m by 1m doors at each end of the long dimension.

Teams will be allowed to use only what they can carry within a container with outside dimensions summing to less than 150cm, and weighing 25kg or less. For example, a container 70cm long, 50cm wide, and 30cm tall has a total dimension of 70+50+30 = 150cm, and would be within the size limits. These limitations are designed to represent the very real space and weight restrictions enforced on space missions, and to make the event more challenging. For convenience, we will also allow access to six standard domestic AC power outlets (United States standard NEMA 5-15, 110v, 15A, 60Hz).

The actual unexpected problem to be solved will be announced on the day of the competition, and can include anything that one can imagine happening on an extra-terrestrial habitation. The problems will be constrained to have likely robotic solution that fit the spirit of the competition. For example, you will not be required to have the robot travel 100km to the site of the problem, or to construct a 10-person emergency habitat from freshly-mined regolith. The scope of the task might vary from a short 1-hour task, to one taking from 4-6 hours. Specific tasks will be announced to all teams simultaneously, and they will work on their solutions independently.

Example Scenarios

To give you an idea of the sorts of tasks that we have in mind, here are a few examples. These tasks are meant to be representative, and to convey the spirit of the competition. We do not guarantee that any of these tasks will actually be assigned during the competition. On the other hand, we do not guarantee that they will not.
Antenna recovery.

An antenna outside the habitat has been knocked over during a Martian storm. The antenna is crucial to the guidance of a resupply transport, which is scheduled to arrive in 4 hours and an EVA is not safe. The team must develop a robot that can reach the antenna, grasp it and reattach it to its receptacle. The antenna is 10 m from the habitat, sitting on top of a 1m by 2m rectangular base that is 1m tall. The base is visible from the habitat. The antenna is a 1cm cylindrical rod 1m long that fits as a peg into a hole 2cm deep in the base. You have a spare antenna and base in the habitat that can be used for testing purposes.

Base station repair.

Sensors have discovered a tear in a thermal covering on the top of storage shed which contains the habitat's store of liquid nitrogen. The team has 4 hours before the Martian morning arrives and starts to dangerously heat the nitrogen. The team must develop a robot that can crawl on top of the structure, use the supplied patching material, and patch the hole by dispensing supplied glue. Unfortunately, the structure was not designed to support heavy weights, so the robot must weigh less than 5kg or risk collapsing the structure, with disastrous consequences.

Buggy Crisis.

a. Diagnosis. The habitat's moon buggy has sustained damage to its undercarriage during a particular rough outing with one of the more adventurous astronauts. It has stopped 20m from the base. The team must develop a robot and camera system that can squeeze in the 10cm clearance between the undercarriage and the terrain to determine the damage.

b. Repair. The damage to the buggy has been identified as a broken electrical wire. The team must develop a robot that can strip and splice the two ends of the wire so the buggy can limp back into its garage adjacent to the habitat.

Third trial: Human-Robot Interaction (HRI)

HRI has by now become a major research field in robotics. The ICRA'08 HRI Challenge aims at demonstrating a number of state-of-the-art platforms in HRI, as well as provides a realistic platform (the ICRA Conference) for evaluating the effectiveness of the interaction.

Rules and participation

In order to leave the floor open to any team working in HRI, there are no specific requirements neither on the shape and sensori-motor capabilities of the robot, or on the experimental context. We accept any robot: wheel-based platforms, as much as humanoids ones are welcome. We also accept WOZ type of experiments or video-based HRI experiments. And we leave it up to the teams to define the experimental context, i.e. the type of the interaction.

Thus, at this stage, the sole requirements we set for taking part in the HRI Challenge concern the robot's behaviour. The robot should be at least endowed with either of the following two capabilities:

- The ability to learn by interacting with a human (we say nothing about what the robot can learn; being more restrictive in the learning capabilities would constrain too much the hardware of the robot).

76
The ability to interact socially (this includes any of the following capabilities: detecting humans, detecting human activities, respond to humans; infer human intention, communicate verbally or in other manner, detect and respond to human emotion, ...).

**Evaluation:**

The effectiveness of a robot engaging in HRI must be evaluated by human users who got the chance to interact with the robot for a sufficiently long period of time. This challenge thus requires that the robots be highly interactive and run constantly throughout ICRA.

The robots' behaviour will be evaluated formally by a team composed of 10 official evaluators (10 experts in robotics, but not necessarily in HRI) and lay people (all the people attending the conference will be given a questionnaire to fill in). The questionnaires will be prepared by experts in HRI evaluation methodologies. The exact scoring system is yet to be defined, but it will at least encompass a score according to the two requirements on the robot's behaviour listed above.

More information can be found in the official webpage: http://icra.wustl.edu/events/
BENCHMARKING INITIATIVES
IN EUROPE

BENCHMARKS FOR MOTION PLANNING

This area is a mature field in robotics research: during the last three decades motion planning has established itself as an important area of robotics. In addition, it has traditionally only dealt with simulations, ignoring the problems of working with real robot implementations. These facts should make it particularly amenable for benchmark developing, since the only hardware involved is a computer. In spite of its progresses and maturity, motion planning has achieved limited success, so far, in terms of widespread penetration into industrial applications. One of the reasons might have been the difficulty in comparing the performance of the existing motion planning techniques and in assessing their suitability for the problem at hand. Consequently, the need for benchmarks is acknowledged in the community because usually it is difficult to compare the different techniques since they are tested by solving a limited set of specific examples on different types of scenes, using different underlying libraries, incompatible problem representations, and implemented by different people on different machines. This area has also been suggested by the EURON research roadmap, since motion planning is crucial both for advanced production systems and household robots.

There is a general agreement that efforts should therefore be devoted in the development and dissemination of open, standardized benchmarks and performance assessment methodologies. Even so, it is felt in the MP community that this is a very difficult issue because whether a method is good depends on a large number of factors, often defined by the application in which the method is being used.

This is, indeed, a long standing problem. Hwang and Ahuja [92], in their famous survey, already urged motion planning researchers to develop a set of realistic and non-pathological benchmark problems. This need was later highlighted by Gupta and del Pobil [98]. After a decade of significant advances in motion planning, no common set of benchmarks is available yet. Rather, the effectiveness of each tool is measured by means of a specific set of problems, which however cannot be shared by any other tool due to incompatible or proprietary formats and characteristics. A limited exception to this situation has been the alpha puzzle, a benchmark for narrow passage problems proposed by the Parasol motion planning group, though in the opinion of others, this would rather exemplify a disease called puzzlitis: according to which, the most obscure case within the domain is picked as a criteria for successful research, in order to show generality, even though the puzzle problem itself is very unlikely to ever be encountered in practice [Brooks 90], whereas other more practical issues are abstracted.

As an example of this state of affairs in the context of the probabilistic roadmap approach, which is a commonly used motion planning technique, Geraersts and Overmars [04], in a recent study, compared 12 sampling techniques which is a critical component of this approach. For the comparison they used a single environment on the same scenes. Their
results were surprising in the sense that techniques often performed differently than claimed by the designers. The study also showed how difficult it is to evaluate the quality of the techniques. It is expected that these results and other to come should help users in deciding which technique is suitable for their situation.

Another interesting work in this regard is that of Reggiani et al. [02]. It is a well know fact that randomized motion planners are greatly affected by the efficiency and robustness of algorithms for collision checking of robot configurations, even though the examples typically used tend to simplify this point. A number of powerful collision detection algorithms are currently available, but their relative capabilities and performance cannot be easily compared due to the many factors involved. In this work, collision detection packages were experimentally evaluated within the context of motion planning for rigid and articulated robots in 3D workspaces. Artificial and realistic problems were chosen as benchmarks to assess package behaviour with different object models.

An initiative toward the development of an interchange format for motion planning problems has been in progress. The participants in the discussions felt that the development of sharable benchmarks could not be achieved in just a few months but rather it was expected to be a long process. A primary goal in this process should be to identify input format requirements of existing tools and, based on this assessment, define common concepts and terminology.

Some considered that defining open, standardized benchmarks offers several benefits since they can give immediate insight when developing new planners, improve the understanding of strengths and weaknesses of existing planners, and they can be usefully exploited for educational purposes. Furthermore, benchmarks designed from industrial cases could inspire new research directions and contribute to increase the impact of planning tools.

It was agreed that given the usefulness and desirability of a set of benchmarks for the research community, the key issue of designing appropriate and sound benchmarks should be addressed. However, among the identified difficulties it can be mentioned that defining an adequate taxonomy for motion planning problems is quite complex because problems can be varied along several dimensions: problems could be classified either by workspace complexity (presence of narrow or wide corridors, geometric representation of obstacles), or C-space dimension, robot type (kinematic chains, closed kinematic chains, non-holonomic robots), etc

Assuming the availability of a set of benchmarks, still, there are other problems to be faced. Defining what should be compared is more tough than expected. Computation time for a given problem complexity has been the typical metrics for motion planning and collision detection algorithms, but the time required to find a solution is often less important than the solution quality. Moreover, quality is difficult to measure: the best solution could be the shortest, or the smoothest, or the most human-like depending on the problem requirements. Currently, other metrics are being considered such as those based on coverage and connectivity, rather than on the time it takes to solve some particular queries.

Another problem is that obtained results are vitiated by the large dependency of the tools on implementation details and exploited libraries, such as collision detection packages. A final issue regards the benchmark representation using the proprietary file format of the
tools: when converting a format into another some parameters might get changed causing the problem itself to change.

There are a number of initiatives aiming at defining a format for motion planning since the definition of sharable benchmarks cannot be achieved without a broadly accepted interchange format for the description of the robots, the workspace and the problem. Some of the requirements of a suitable interchange format were identified. The format should be human-readable (self-explaining and simple to use), flexible (able to describe several kind of robots and obstacles), validable (easy to check data consistency and coherence) and extendible (easy to scale to future needs).

Two remarkable experiences in defining an input format for motion planning applications have been developed: First, Utrecht University has proposed the input format to be used with their application Callisto, which is a library to visualize 3D virtual environments that is specifically suited to support research on Probabilistic Road Maps (PRM's). It consists of a visualization environment based on Maverik and it uses Solid for collision checking. The input format used to describe scenes in Callisto is based on XML. A DTD defines the structure and the tags of the XML files that describe the workspace, the robots and the problem.

An XML-based input format has also been proposed by the University of Parma. As the research group of Parma uses motion planning techniques for different applications, such as the control of mobile manipulators and programming by demonstration, the need of an easy to use, self-describing and flexible format to share problems among them raised. The structure, content and semantics of valid XML files describing the robots and the workspace are defined by a set of XML-Schema documents. A Java3D tool to visualize motion planning scenario that uses the XML format is also part of the system.

There has been an on-going discussion in the MP community as a first step toward format structure and requirements elicitation. The most challenging issue was recognized to be problem description. Currently, most of the formats specify the final goal through the definition of d.o.f. values for the robots. This has proven to lack in flexibility, indeed, it does not allow a higher level approach focusing on task goals instead of position goals.

The suitability of XML as the technology to define an interchange format was highlighted. The choice of a markup language is motivated by their common use in storage, transmission and exchange of information, as they allow the description in a standardized format of data or information contained in text. XML has already proven to conveniently describe various types of structured data, as demonstrated by a growing number of XML-based languages in a wide range of domains. XML documents are human-readable, self-described, easy to maintain while guaranteeing interoperability. Moreover, they can be easily extended and several tools to parse them are freely available. In particular, the structure of an XML document allows the reader to efficiently skip unsupported information. A comprehensive interchange format can be designed still guaranteeing interoperability with several applications as unsupported information (for example texture and color of geometric objects) can be easily ignored.

Another issue that has also been discussed is that of establishing a common geometric format to be used in the exchange format. As most of the currently available collision detection packages employ triangulated objects, a triangulated format could seem a sound
choice; but the problem of dealing with different geometric representations such as CSG or nurbs (the latter frequently used in industrial applications) would still remain open.

Assuming a unique format should be chosen, VRML was proposed as a suitable candidate format for geometry representation since a triangulated representation can be easily obtained from a VRML model. Moreover, VRML is widely supported by several tools, it is general and versatile to allow description of a wide variety of objects, it supports non-strictly-geometrical information such as texture and color, and a number of models are already freely available on the Web.

An initiative for a repository for motion planning benchmark problems is maintained by the University of Parma. It aims at including benchmarks designed by different research groups and documentation describing the file formats currently used by the available planning tools to define their problems. Also, links and useful information regarding motion planning research projects are available. The repository is intended to serve as the basis for further discussion on the requirements and the design of benchmarks in motion planning.

The data sets follow the so-called Motion Planning Markup Language (MPML), an XML-based input format that has been proposed by the University of Parma, in such a way that the structure, content and semantics of valid XML files describing the robots and the workspace are defined by a set of XML-Schema documents. A Java3D tool to visualize motion planning scenarios that uses the XML format is also part of the system.

The repository (which can be found in http://mpb.ce.unipr.it) includes data sets about: robots, workspaces and benchmark problems.

Similarly, the so-called Movie Models for Motion Planning is intended as a repository of motion planning benchmarks maintained by Utrecht University. It originated from work in the context of the Movie project in which motion planning techniques were developed and tested by using tools and 3D models of robots and scenes. Then they were opened to the community at large in model collections providing easy interface and search functionality, as well as useful information about the models, in such a way that new models could be easily added by researchers around the world. Specification formats for the robots and scenes are a critical issue for the widespread acceptance and use of the models. Current file formats are:

- XML file format of Callisto
- Virtual Reality Modeling Language (VRML and WRL)
- Extensible 3D (X3D)

This repository can be found in: http://www.give-lab.cs.uu.nl/movie/moviemodels/

**Also additional technical information** is included in this document as Appendix VI and VII and further details are described in the papers:

- *On Experimental Research in Sampling-based Motion Planning*, by Roland Geraerts, included in the document *Lecture Notes of the IROS’06 Workshop on Benchmarks in Robotics Research* (deliverable DR 2.4)

- *Motion Planning vs. Automated Planning in Benchmarking*, by M. Reggiani and E Pagello, in the *Lecture Notes of the IROS’07 Workshop on Benchmarks in Robotics Research* (Appendix XII)
BENCHMARKS FOR MANIPULATION AND GRASPING

The robotics community has always recognized as a key area that of the interaction of a robotic manipulation system with its environment. Although this is a very interesting and central issue, the involved problems, both at the technical and methodological level, still represent a limiting factor in a number of important cases not only in standard industrial applications but also in non-conventional uses (e.g. space, soft material, dexterous manipulation, etc.). Among the relevant research fields interested in this area, one can enumerate the following: advanced sensors (e.g. tactile, force/torque, vision, etc.), mechanical structures and devices (e.g. parallel, redundant, flexible, etc.), planning, control, optimization. Due to these reasons, the complexity to advance in the definition of benchmarks is acknowledged as a very difficult task, which is also the case for other areas of robotics, though manipulation and grasping is an area that has also been suggested by the EURON research roadmap, since it is crucial both for advanced production systems and household robots.

Nevertheless, the majority of experts in this field agree that such a definition would be desirable and less difficult to achieve if we try to define a benchmark for each elemental topic within this very large domain: such as mechanical design, control algorithms, sensors, artificial intelligence strategies, etc.

In particular, these tentative benchmarks should include well-defined tasks and rules, methodologies, design for experiments, calibration process for instrumental devices, etc. Moreover, the algorithmic performance should be tested on a variety of different hardware platforms and vice versa. For instance, a hand can perform better than another one, also with the same control algorithm, since it is more suitable for a given task, and so on.

Finally, some ideas about specific benchmarks have been proposed related with the following three tests:

- **Test-1.** Grasp and re-grasp of a given object (both regularly or no regularly shaped). The metrics here would be: Capability of disturbance rejection; Ability of fine manipulation; Complexity of the design; Force/Form closure capabilities.

- **Test-2.** After a pick and place operation is well-defined, the task is divided into two stages: (1) “picking phase”, and (2) “holding phase”. In (1) the evaluation concerns of measuring the statically applied picking forces to the grasped object (e.g. rigid, flexible, with different geometry, etc.). While, in (2) the measure of the maximum acceleration, in different moving directions, permitted before to lose the grasped object would be an interesting metric.

- **Test-3.** Take a set of objects and replace them as soon as possible. Here the metric would be time and statistics of failure.

There has been further recent progress based on the test 2 above, particularly in the context of visually guided grasping. This initiative has been developed at Universitat Jaume I and offers a description of a set of experiments on visually-guided grasping of planar objects with a Barret hand. They are made available to the community as a set of standard
experiments on which benchmarks and associated performance metrics can be proposed. A first experimental protocol for such a benchmark has been also proposed.

The goal of the experiments is to test different implemented procedures for visually-guided grasping. That is, by only using visual information about planar objects a set of 3-finger feasible grasps are determined and executed to lift the object. The experimental setup for the experiments consists of a stereo head with two gray-scale cameras and a robotic arm with a three-fingered Barrett Hand, which is commercially available and widely used for robotic grasping research.

Some problems are included as part of these experiments: such as constraining of a general three-finger grasp to the particular kinematics of the 4 d.o.f. Barrett Hand or the use of stereo images to locate an object placed in front of the robot, and to obtain the contour of such object.

A second step in this initiative puts forward a protocol based on the previous experiments, once the object is lifted successfully. The robot applies several sequences of shaking movements to the object consisting of rapid accelerations and decelerations of the hand. The result of this stability test, permit to asses the grasp quality in five different categories.

This initiative is described in: http://www.robot.uji.es/people/morales/experiments.

Also additional technical information is included in this document as Appendix X. Part of these ideas are going to be implemented in the context of the new FP7 project GRASP “Emergence of Cognitive Grasping through Emulation, Introspection, and Surprise” in which Antonio Morales from UJI (Spain) is one of the partners.

Benchmarks for manipulation and grasping is also an important issue other new FP7 projects such as DEXMART "DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills: A bridge from natural to artificial cognition". In this Integrated Project there is a complete workpackage devoted to benchmarking and experiments, as discussed by the WP leader Gerhard Grunwald from the DLR (Germany) at the CogGEMBench’08 Workshop on Good Experimental Methodology & Benchmarks in Cognitive Robotics in Karlsruhe.
BENCHMARKS FOR VISUAL SERVOING

Discussion

Visual servoing has been a viable method of robot manipulator control for more than a decade. It involves the use of one or more cameras and a computer vision system to control the position of the robot's end-effector relative to the workpiece as required by the task. It is a multi-disciplinary research area spanning computer vision, robotics, kinematics, dynamics, control and real-time systems.

Though this discipline has seen considerable development in recent years, the different methods offer tradeoffs in performance, and cannot solve all tasks that may confront a robot. There are a number of recurrent questions such as: is 2D better than 3D visual servoing? (and replace 2D and/or 3D by 2D1/2, stereo, point/line/circle ...) for which we have a certain amount of theoretical arguments. However, in practice, people seem to focus mainly on the overall behaviour of their method rather than on the effective accuracy or on the quantification of the behaviour. For instance, “my method yields straighter trajectories” but what does a straight trajectory means in visual servoing?

To date, there has been little research that explores the relative strengths and weaknesses of these methods. Thus, the experts agree that defining benchmarks, metrics, and measurement procedures in visual servoing is very interesting since many research papers present new visual features or new control schemes to realize the same basic task. Comparing these methods on a well-defined and universally recognized benchmark would help to judge the practical interests of these contributions.

But they also are aware of the fact that it would be very difficult or nearly impossible to specify a benchmark that turns out to be useful for all past and future works in visual servoing. On the contrary, benchmarking would be more amenable on free and web-available simulation software, since visual servoing algorithms are rather complex. Indeed, the control law itself may be rather simple to derive and implement once the geometric primitives are extracted from the image, even though obtaining the geometric primitives may not be easy. There exist many trackers, pose estimators, image treatments, not to say anything about robots and applications.

So, the problem is mainly to know what we want to measure:

- only the control law, then the trouble is how to have realistic inputs.
- the overall application, then the trouble is how to compare a mobile robot application with an assembly task, for instance.

Between these two extreme positions, we may want to split the benchmark into sub-benchmarks but under which criterion? Applications? Such as mobile robot, manipulator robot (and here we can split it again between anthropomorphic arms, Cartesian arms, parallel robots, etc.) Subtasks? (i.e. detection, tracking, control law, numerical implementation).
There is no clear answer since this is a multicriteria problem with strong overlappings and coupling between the criteria. For instance, detection algorithms depend on your application (outdoor mobile robot or microrobotics).

It is arguable that we can set-up a generic benchmark for visual servoing that is completely general and generic. We might be forced to restrict the benchmark to some specific application, but then the comparison would only hold for this application.

In a visual servoing system all the parts of the system should be considered, namely:
- matching
- tracking
- servoing

In each of these parts one should identify which information is measurable and which is not. Then, find the good metric to measure the performance of each sub-system and the performance of the whole system.

**Characteristics of benchmarks**

Benchmarks should include some datasets (which would specify at least the target and the robot configuration) available in a publicly available repository, together with free simulation software.

Researchers could use benchmarks and datasets to improve their algorithms. In order to avoid ad-hoc algorithms that just solve the problem, the benchmarks should evolve. Researchers should be able to propose new datasets that defeat standard algorithms.

What an ad-hoc dataset should be is also a tough question since visual servoing is definitely a continuous process over the whole workspace. Quantization of this workspace would generate non-neglectable artifacts. In the case of a benchmark on control only, then using the Java simulator or the Matlab Toolbox (see proposals below) would be a solution. However, we come back again to the relevance of the comparison and on how realistic is the simulation.

Some experts believe that well-defined tasks or rules should be preferred. However, there is not, at the moment, the slightest idea of what they could be.

It is widely agreed that it does not make sense to compare different approaches/algorithms running on different hardware (robots, hands etc). The hardware should be the same for a better comparison among algorithms. For an easy use by all the community, it is suggested to perform benchmarks/metrics/measure procedures in simulated tasks.

Only if the comparison is application oriented, then the hardware might be different, but then the benchmark is not restricted to visual servoing any more. Hence, the benchmark turns somehow into a robot competition under the constraint that vision is the only exteroceptive sensor, and opens to robotics in general.
If one only wants to restrict to algorithms, the hardware should be exactly the same, i.e. one single physical instance. It is not sufficient enough to say: “the hardware should be a Mitsubishi PA-10, with a Sony XCD-X700 camera” since the hand-eye position is not defined, and the lightning conditions may differ. Even two robots may have different accuracies depending on their age, mechanical assembly, low-level control algorithm version, and so on.

Thus, if a single hardware is proposed then, however, the comparison may be biased in the sense that the algorithm may be optimized (consciously or not) for this precise hardware and loose its generality which is, as for most experts, the beauty of visual servoing.

Due to the problems of using physical setups, an open simulation tool, which might be updated with all possible algorithms, would be necessary. Then, one could define a series of benchmarks to compare algorithms that work on the same data input and produce the same output.

**Benchmarks proposals**

To start with a minimal benchmark, one should consider the most basic task (6 dof control for a positioning task), consider different possible targets (planar, non planar), different initial and desired poses and different sources of perturbations. Afterwards, other tasks may be faced, as well as more complex situations.

Considering that matching and tracking algorithms are available and provide a set of image points, and a reference image is available, the problem is to position an eye-in-hand camera from a random starting position. The benchmarks could be:

- Measuring the success of the algorithms (convergence with respect to several starting positions).
- Measuring the computational cost of the algorithms.
- Measuring the behaviour of the algorithms.

Existing software packages could be possibly customized to allow such benchmarks, e.g. MATLAB Robotics Toolbox, ViSP (C++) from Irisa, or JaViSS (Java) from Jaume-I University.

The MATLAB Robotics Toolbox has been developed by Peter Corke (CSIRO – Australia) [Corke 96]. It provides many functions that are useful in robotics such as kinematics, dynamics, and trajectory generation. The Toolbox is useful for simulation as well as analyzing results from experiments with real robots. The current release includes a Simulink block library, which brings Toolbox functionality into the Simulink environment. There are also a number of demos ranging from simple forward dynamics to image-based visual servoing.

ViSP (Visual Servoing Platform) is modular software that allows fast development of eye-in-hand image-based visual servoing applications [Marchand 99]. Various issues have to be considered in the design of such application: among these issues we find the control of camera motions and the tracking of visual features. The ViSP environment features a wide class of control skills as well as a library of real-time tracking processes.
If physical robots were going to be considered in a further step, visual servoing schemes should be defined taking explicitly into account the robot kinematics. Hence, it would be possible to test various algorithms on various robots: a kind of plug-and-play interfaces so that one can “plug” a “foreign” robot on the PC doing visual servoing as easily as with the robot itself. This would become a kind of visual servoing interconnection standard, essentially.

In this context, a visual servoing simulation environment called JaViSS has been developed at Universitat Jaume I. It is called JaViSS and written in Java with graphical rendering making extensive use of Java 3D API. The calculus engine is implemented with the Colt libraries. Though JaViSS currently runs on a single computer, it has been heavily designed in a distributed manner, using the agent-based JADE platform. Finally, 3D models have been created with AC3D, and loaded with the Java 3D loader J3D-VRML97. Manipulator kinematics code is based on the Robotics Toolbox for Matlab by Peter Corke. It is intended as a tool to simplify the testing and comparison of different visual servoing approaches since it models free-motion cameras as well as cameras attached to a manipulator, with its kinematics. Different visual primitives can be modeled, to compare the behaviour of the visual task with regard to the choice of primitives.

This initiative is located at: http://www.robot.uji.es/research/projects/javiss. Also additional technical information is included in this document as an appendix and a complete account can be found in the paper included in the deliverable document Lecture Notes of the IROS’06 Workshop on Benchmarks in Robotics Research (deliverable DR 2.4):

Cross-Platform Software for Benchmarks on Visual Servoing, by Enric Cervera

Similarly, the University of Siena in Italy has developed an Automatic Control Telelab with the idea of offering support to real-time configuration and observation of experiments, as well as playback access to acquired data, from remote computer linked to a collection site through the Internet. The underlying philosophy is that distributed data acquisition makes possible the collection of data from remote environments and can also improve collaboration among geographically dispersed scientific communities by distributing scientific results more quickly and less expensively than most other methods. The on-going initiative offers the resources already available at the Automatic Control Telelab —both software and hardware—to the research community in order to extend them for implementing visual servoing benchmarks to be used in a remote way. Since it is the same hardware set-up that is used, comparison across different methods is easier.

At present, five diverse experiments for remote control are available and two competitions based on this framework are proposed as a way of benchmarking.

This initiative can be found at: http://act.dii.unisi.it. Also additional technical information is included in this document as Appendix VIII.
Mobile robotics has become in the last fifteen years a major area in robotics research. As suggested by the EURON research roadmap, its importance is strategic not only for advanced production systems but especially for household and service robots. We have already mention that 95% of the data sets contained in the US Radish repository are sensor-logs and maps relevant to researchers in mobile robotics. In Europe, an initiative called RAWSEEDS has recently started aiming at stimulating and supporting progress in autonomous mobile robotics by providing a comprehensive, high-quality benchmarking toolkit for SLAM (Simultaneous Localization And Map building). More details are included in the specific section.

Work aiming at the definition of metrics to evaluate motion techniques from two points of view has been proposed under the coordination of Javier Mínguez (University of Zaragoza). The first initiative takes mobile robotics perspective i.e. the point of view of the resulting motion. Historically, many techniques have been designed to address autonomous collision-free motion (sensor-based motion with obstacle avoidance). It is clear that under the same conditions each technique generates a different motion. Nevertheless, questions like: which is the most robust one? or which of them behaves better in a determined context or condition? Cannot be answered neither from a scientific nor technological point of view. In other words, once we face a mobile robotics application, selection of a motion technique among all the existing ones is a matter of specialists and not accessible to everybody. This is because there are no objective comparisons of methods either quantitative (in terms of robustness or action parameters such as the time or the total traveled distance) or qualitative (in terms of safety of the motion). The scientific objective of this initiative is the development of a procedure to evaluate the motion techniques from the quantitative and/or qualitative perspective of the motion generated in different contexts.

The second initiative aims at defining metrics to evaluate motion techniques from a biomedical perspective, i.e. based on the biomedical response of the user. Research in emotional computation explores computation in developments where human emotion is an essential concept. In the healthcare or clinical fields, any proposed intervention (therapeutic or preventive) has to take into account the impact over the patient quality of life. This is not only to improve their health, but also the subjective perception of welfare. In this sense, the scenarios have to be conceived and validated explicitly in terms of compatibility with the users-patients welfare. Thus, for example, a motion technique can be robust and move the vehicle with optimal trajectories in time, distance, etc. But when a robot guide or transports a human, a basic question is how the user feels during the movement. In the development of healthcare technologies the user satisfaction is an essential specification. Therefore, it is required to measure and to evaluate systematically the quality of the user–device interaction and the reaction of the individual to a particular motion technique. This fact has a vital importance in the growing demand of autonomous robots for human transportation for the present applications of transportation, specially for those designed for aging users and/or that spend a great time using the vehicles. The objective of this initiative is the development of a procedure to evaluate the reaction of the user to the motion generated by the vehicle. In biomedical sciences, one way to evaluate the human response to external events is to study the biomedical (physical and psychological) activity during the development of a task. Other methodologies include the filmed observation, speech techniques, etc. By using these
methodologies, the objective is to develop a way to evaluate and thus to create a metric for the existing motion techniques obtained from the user reactions. There is no work related with this issue, although this result could be very interesting for any robotic application of automatic transportation.

Further technical details can be found in two papers by Javier Mínguez and co-workers:

- *Steps Towards the Automatic Evaluation of Robot Obstacle Avoidance Algorithms*, by Ignacio Rañó and Javier Mínguez, *Lecture Notes of the IROS’06 Workshop on Benchmarks in Robotics Research (deliverable DR 2.4)*


Also, Ali Marjovi and Lino Marques from the Institute for Systems and Robotics at the University of Coimbra in Portugal have pointed out that there does not exist a set of benchmarks globally accepted to assess the performance of motion control algorithms. They have proposed a number of possibilities for benchmarking robot motion control algorithms at the GEMBENCH’08 EURON Workshop on Good Experimental Methodology and Benchmarking in Robotics.

In the last months there have been an increasing interest for benchmarking in the context of SLAM methods (Simultaneous Localization and Mapping). Apart from OpenSLAM and ongoing work in the context of RAWSEEDS —both described below— Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernardo Wagner from the University of Osnabrück in Germany have worked on benchmarking urban 6D SLAM. In the past many solutions SLAM have been presented. Recently these solutions have been extended to map large environments with six degrees of freedom poses. To demonstrate the capabilities of these SLAM algorithms it is common practice to present the generated maps and successful loop closing. Unfortunately there is often no objective performance metric that allows comparing different approaches. This fact is attributed to the lack of ground truth data. For this reason they developed a novel method that is able to generate this ground truth data based on reference maps. Further on, the resulting reference path is used to measure the absolute performance of different 6D SLAM algorithms building a large urban outdoor map.

Further technical details can be found in:

- *Benchmarking Urban 6D SLAM*, by Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernardo Wagner, *Lecture Notes of the IROS’07 Workshop on Benchmarks in Robotics Research (Appendix XII)*

Benchmarking in mobile robotics is not only limited to wheeled robots, snake locomotion has been addressed as in:

BENCHMARKS FOR COGNITIVE ROBOTICS

Both the EURON roadmap and the FP7 ICT Challenge 2 on Cognitive Systems, Interaction and Robotics suggest the importance on benchmarking Cognitive Robotic Systems. In particular ICT Challenge 2 specifically addresses the idea of autonomy: Artificial systems that can achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; adapt, within reasonable constraints, to changing service and performance requirements, without the need for external re-programming, re-configuring, or re-adjusting.

In the first half of 2007 a number of informal meetings and discussions on benchmarks for cognitive robots were held, taking advantage of related events. Several actions were considered resulting in the organization of a number of events specifically focused on cognitive or intelligent robots, namely:

- IROS’07 Workshop on Performance Evaluation and Benchmarking for Intelligent Robots
- CogGEMBench’08 Workshop on Good Experimental Methodology & Benchmarks in Cognitive Robotics
- IROS’08 Workshop on Performance Evaluation and Benchmarking for Intelligent Robots

They have all been already described above.

The common motivation of these workshops was to promote a capability-led understanding of cognitive robots: how to define shared ontologies to discuss robotic cognitive systems in terms of their performance, relationships between different cognitive robotics capabilities, requirements, theories, architectures, models and methods that can be applied across multiple engineering and application domains, detailing and understanding better the requirements for robots in terms of performance, the approaches to meeting these requirements and the trade-offs in terms of performance. Another important point to discuss was the distinction between autonomy and intelligence (if any) and how one influences the other.

Rat’s Life: a cognitive robotics benchmark

In this regard, the ICEA project (funded by IST Cognitive Systems) has organized a contest to promote research results and stimulate further interest in bio-inspired cognitive robotics. The participation to the contest is open to anyone and free of charge. Contestants can download a free version of the Webots software for simulating a robotic scenario where two rat robots compete against each other for survival in a maze-like environment. The developed robot controllers can be transferred in real e-puck robots roaming an interactive LEGO maze.

Rat's Life is a complete cognitive robotics benchmark that was carefully designed to be easily reproducible in a research lab with limited resources.

The real setup relies on two e-puck robots, some LEGO bricks while the simulation setup relies on the Webots robot simulation software. This benchmark is a survival game where two robots compete against each other for resources in an unknown maze. Like the rats in
cognitive animal experimentation, the e-puck robots look for feeders which allow them to live longer than their opponent. Once a feeder is reached by a robot, the robot draws energy from it and the feeder becomes unavailable for a while. Hence, the robot has to further explore the maze, searching for other feeders while remembering the way back to the first ones. This allows them to be able to refuel easily again and again and hopefully live longer than their opponent.

The e-puck robots are equipped with several sensors, including a camera, 8 distance sensors and a 3D accelerometer. Developing a robot controller requires basic programming skills in Java language. Different approaches can be investigated and combined with each other, including robot control, image processing, landmark based navigation, self localization and mapping (SLAM), game strategy, autonomy management, AI, learning, etc.

The contest started on January 7th, 2008 and terminates on June 30th, 2008. During this period, the contest remains open to new competitors. Every business day, a contest round is executed so that every competitor will face at least one opponent and the general ranking is updated. All the matches of a round are immediately visible online from the contest web site.

Participation to the Rat's Life contest is totally free of charge and doesn't require the purchase of any product or service. To enter the contest, the participant has to register on the website, download and install the demo version of the Webots software, program your robot controller and submit your code on the contest web site.

For more information: http://www.ratslife.org/

The RAWSEEDS project is an SSA (Specific Support Action) in the EU 6th Frame Program aiming at stimulating and supporting progress in autonomous robotics by providing a comprehensive, high-quality benchmarking toolkit for SLAM (Simultaneous Localization And Map building).

One of the most substantial limitations to the development of mobile autonomous robotics is the sheer arduousness and cost of performing repeatable and reliable tests of systems and algorithms. In addition, it is difficult to quantitatively assess the performance of a system in ways which are meaningful to people outside the group who designed the system; and it is often impossible to compare the results obtained with different solutions or by different research teams. This has a stifling effect on the whole robotics field, especially where industrial research policies are concerned: no one is happy to operate in a field where marketable applications abound, but heavy investments are needed to simply check if an idea is a good one, before any design, engineering or industrial effort can even begin.

This absence of standard benchmarks is a widely acknowledged problem in the robotics field, and this is doubly harmful to it: firstly, because it prevents recognition of scientific and technical progress, thus discouraging research and development; and secondly, because it prevents new actors (and particularly SMEs) from entering the robotic sector, as heavy investments are needed to compensate for that absence.

The problem described above has two causes: first, the experiments needed to test an algorithm or robotic system are extremely difficult and costly to set up; second, the unavailability of benchmarks to quantitatively evaluate the performances of such algorithms and systems makes the experiments nearly useless for groups different from the one which performed them in the first place. Rawseeds strives to tackle and eliminate both causes of the problem. The benchmarking toolkit that RAWSEEDS will create includes: high-quality multisensor data sets, benchmark problems based on them, state-of-the-art solutions to these problems in the form of algorithms and software, and methodologies for the assessment of algorithms. Rawseeds’ datasets are sets of time-synced data streams, generated by the sensors aboard a robot platform when it moves through an environment. The datasets are gathered in real-world locations.

For researchers, Rawseeds will speed up considerably the production of innovative, high-quality results (i.e., algorithms, software and complete robotic applications). If, as we hope, the Rawseeds Benchmarking Toolkit will be internationally acknowledged as a standard benchmark for the evaluation of algorithms and software systems (in the fields of mapping, localization and SLAM), its use will let research groups evaluate each other’s work and compare it with their own. The best solutions will be easily and rapidly singled out. Highly innovative but “far out” ideas will no more be neglected for the sheer cost of testing out them, as that cost will be heavily cut down. Even the smallest research groups will have the opportunity to publicize the results of their work, using the Rawseeds website: so, if worthwhile, this work will be immediately acknowledged by all the robotics community.
For groups with an interest into autonomous robotics but not yet involved, Rawseeds will represent the force that will drive them in. Knowing that their ideas and solutions can be tested with a small expense of time and money, the main reason for them to wait will disappear. This is especially true for companies (and even more for SMEs, i.e. Small and Medium Enterprises), both because they are especially sensible to cost issues and because they will have access to an easy mean to compare the performance of their (possible) products with those of competitors already entered into robotics or with state-of-the-art research. Moreover, Rawseeds will let companies make their achievements known to all the robotics community.

The presence on the Rawseeds website of a corpus of already working algorithms means that no one will ever need to “start from scratch”.

**Benchmarking Toolkit**

Rawseeds will generate and publish two categories of structured benchmarks:

- **Benchmark Problems** (BPs), defined as the union of: (i) the detailed and unambiguous description of a task; (ii) a collection of multisensorial data, gathered through experimental activity, to be used as the input for the execution of the task; (iii) a rating methodology for the evaluation of the results of the task execution. The application of the given methodology to the output of an algorithm or piece of software designed to solve a Benchmark Problem produces a set of scores that can be used to assess the performance of the algorithm or compare it with other algorithms.

- **Benchmark Solutions** (BSs), defined as the union of: (i) a BP; (ii) the detailed description of an algorithm for the solution of the BP (possibly including the source code of its implementation and/or executable code); (iii) the complete output of the algorithm when applied to the BP; (iv) the scores associated to this output, calculated with the methodology specified in the BP.

The set of sensor data (or dataset) associated to each BP is subject to a validation procedure prior to publication, to ensure absence of defects (such as data drop-outs) and correct time-syncing between separate sensor data streams. In addition to that, each dataset includes the corresponding ground truth, time-synced with the sensor data.

The complete set of BPs and BSs published by Rawseeds is called the Rawseeds Benchmarking Toolkit.

The main use of a BP is to test existing algorithms and compare their performance with that of alternative algorithms. The fact that a common ground for comparison exists is assured by the use of the rating methodology defined by the BP itself.

A BS can be used in many ways, as it is possible to:

- compare the scores obtained by the algorithm included in the BS with the scores obtained by another algorithm applied to the same BP (since the rating methodology is defined by the BP, and so can be applied to different BSs associated to the same BP);
- use the output of the algorithm included in the BS to get pre-processed input data for higher level algorithms to be tested (e.g. planners);
• use the algorithm and/or the software of the BS as a “building block” to design a multi-layer system for the processing of sensor data;
• use the algorithm included in the BS as a starting point for the design of new algorithms.

The Rawseeds Project will generate the BPs and a set of BSs based on state-of-the-art robotics algorithms, but users of the Rawseeds website are encouraged to contribute to Rawseeds by submitting their own Benchmark Solutions for publication. We hope that a vital community of Rawseeds users and contributors will build up, using our Forum to communicate. Rawseeds includes mechanisms to safeguard the intellectual property of the contributors.

It is also possible for Rawseeds users to submit new Benchmark Problems for publication. However, they will be accepted only if the associated datasets have been subjected to the same exacting validation procedures that we applied on our own datasets, and if an associated ground truth with sufficient accuracy is present.

For further information: http://rawseeds.elet.polimi.it

**Additional technical information** can be found in the papers:

- **RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets**, by A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti and J. D. Tardos, included in the deliverable *Lecture Notes of the IROS’06 Workshop on Benchmarks in Robotics Research* (deliverable DR 2.4)

ROSTA: ROBOT STANDARDS AND REFERENCE ARCHITECTURES

The RoSta project is a two-year Advanced Robotics CA (coordination action) in the EU 6th Frame Program. The project started in January 2007 and will end in December 2008.

The objective of RoSta is to proactively take the initiative on the definition of formal standards and the establishment of de facto standards in the field of robotics, especially service robotics. The project does not aim at a broad coverage of topics, which might lend themselves towards a standardization, but rather its goal is to take the initiative in the formulation of standards in a very few, selected key topics which have the highest possible impact. These topics are at the core of robotics research and development, and therefore have the potential to form the root of a whole chain of standard defining activities going far beyond the specific activities of RoSta. More specifically the technological objectives are to coordinate a set of actions initiating and preparing a set of standard defining activities on the following topics of advanced robotics:

a) Creation of a glossary/ontology for mobile manipulation and service robots.

b) Specification of reference architecture for mobile manipulation and service robots.

c) Specification of a middleware for mobile manipulation and service robots.

d) Formulation of benchmarks (of components, methods, middleware and architectures) for mobile manipulation and service robots.

Each line of activity will result either in:

- an action plan for a standard defining activity or
- an action plan and a recommendation/proposal to the European Commission for a supported activity (e.g. a open-source project with significant financial support in FP7) or
- an action plan for a community driven open-source activity with seed-money, for example, to run a project office or alike.

To translate the above activities into a standard defining activity, a liaison with IEEE Standard has been established. Recommendations or action plans developed in the above activities will lead to the establishment of so called Study Groups, which then initiate an official standard defining activity.

As part of its objective (d) above RoSta has carried out a bibliographical research to identify all benchmarks used to evaluate mobile manipulation and service robots. The objective of this research was to identify the state of the art of benchmarking in this specific as well as closely related areas and evaluate if and where there is a need for a standardization activity. For the emerging scientific and commercial field of mobile manipulation and service robots good experimental practice and representative benchmarking is vital. In order to prevent the reinvention of the wheel in every lab when it comes to functionalities necessary to build a mobile robot there needs to be a practical way to identify the best promising scientific and algorithmic approaches for single components and complex systems. But therefore these components and systems need to be tested and evaluated in a reproducible way to generate comparable results. This can help to narrow down and speed up the development and researchers can focus on their individual competences again and combine their solutions with...
other solutions available within the community. As a result of this mobile manipulation and service robots might reach the consumable product stage quicker. Potential customers can easier compare available robots and identify the best machine for their individual demand. The main sources used to identify the benchmarks were the Internet, meetings with experts with both robotic and benchmarking background and reports of EU funded projects. The results from these three areas were sorted and briefly summarized. It was found that there are benchmarks for components and subsystems of mobile manipulation and service robots and also competitions in different areas but there is nothing like a whole system benchmark that produces comparable results.

The resulting Report on State of the Art on Benchmarks for Mobile Manipulation and Service Robots briefly describes the identified benchmarks and competitions, evaluates them in terms of precision and complexity, points out areas in which benchmarks are missing and gives a first idea on how these areas could be filled.

For further information: http://www.robot-standards.org

Additional details can be found in:

EXPERIMENTAL METHODOLOGY IN ROBOTICS RESEARCH

This initiative aims at promoting better experimental practice in robotics research and help pave the way for the widespread practice of benchmarking in robotics. Often, the current practice of publishing research results in robotics makes it extremely difficult not only to compare results of different approaches, but also to assess the quality of the research presented by the authors. Frequently, when researchers claim that their particular algorithm or system is capable of achieving some performance, those claims are intrinsically unverifiable, either because it is their unique system or just because a lack of experimental details, including working hypothesis. Often papers published in robotics journals and generally considered as good would not meet the minimum requirements in domains in which good practice calls for the inclusion of a detailed section describing the materials and experimental methods that support the authors' claims. This topic raised interesting discussions about the way that researches in robotics perform the experiments and the validity that they have. This is very important in other disciplines such as medicine and some ideas could be imported. It was suggested to create a protocol of “writing” the papers involving a given protocol to “experiment” with the robots. The idea was to implement this in the future papers trying to win homogeneity and rigorous validation processes.

This initiative has advanced considerably in 2007 and 2008 with increasing support from EURON community. Two major actions have taken place:

- The topic of the EURON 2007 and 2008 workshops at the Annual Meeting this year included Experimental Practices and benchmarking. A full account of the topics discussed in these workshops can be found above in this document.
- A proposal for a Special Interest Group (SIG) in EURON was accepted with the focus of increasing the quality of experimental methodology practiced in robotics by sharing good practice via educational workshops, summer schools, email discussion and web presentation; by providing assistance to journal and conference reviewers and editors concerning what constitutes experimental robotics and good practice in that sub-discipline; by encouraging the principled replication and comparison of results; and by encouraging the development and use of appropriate systems benchmarks and standard evaluation procedures.

A number of SIG meetings were held to work on these issues, namely:

- Benicassim, Spain, 9/1/2008, collocated with RISE08
- Genova, Italy, 31/1/2008,
- Zaragoza, Spain, 4/3/2008
- Prague, 28/3/2008, as part of the EURON Annual meeting
- Valencia, Spain, 30/4/2008

The main result of these efforts was a document entitled General Guidelines for Robotics Papers Involving Experiments enumerating recommended quality criteria for Robotics Journal/Conference reviews that the community expects a proper high-quality experimental robotics paper have to satisfy. This is formulated as a reviewer checklist and circulated to Journal Editors and Conference Programme Committees. It is included here as Appendix XI. It has already been sent to the Editor in Chief of the highest impact robotics journal: the IEEE Transactions on Robotics. The TRO Senior Editorial Board discussed it in its last meeting in Pasadena (USA) in May 2008. We plan to sending to other journal editors.
Computer vision and image processing algorithms are very important in robotics, as at the moment cameras seem both prospectively as low-cost as required and capable to supply rich data about the robot workspace. Therefore benchmarking of computer vision and image processing algorithms is important for roboticists; it is important to have a distributed awareness about the pros, cons and actual expectable capabilities of each algorithm in each situation. In some we probably have plenty of alternatives, in some just one and in other we have no viable algorithm for robotic use. The underlying issue is that generation of sensed data for robots is done most of the time by a pipeline processing of gigo type (garbage-in-garbage-out, at each step). This initiative aims at preparing datasets, for relevant sub-domains, relevant algorithms, etc. so that the community can ask new robotics algorithms, as described in the papers, to be validated with those datasets. This initiative has been put forward by Domenico Sorrenti, (Universita Milano-Bicocca), with two initial subareas of interest:

- omnidirectional vision
- robustness to light changes, highlights, strong shadows, etc.

At the moment this initiative is in the process of applying for obtaining specific funding. Of related interest is the paper included in the deliverable *Lecture Notes of the IROS’06 Workshop on Benchmarks in Robotics Research (DR 2.4): Drawing a parallel: Benchmarking in Computer Vision and Robotics*, by Danica Kragic, Ville Kyrki, Patric Jensfelt and Frank Lingelbach.
BENCHMARKS FOR NETWORKED ROBOTICS

This area is a comparatively young field in which many new ideas and researchers are being incorporated and, at the same time, it is an area of increasing importance and rapid expansion in terms of people and funding as suggested by the EURON research roadmap.

Defining benchmarks and their associated metrics is perceived by experts in the field as both very interesting and certainly possible, since there are so many proposals for system designs of networked robots, that it would be mandatory to compare performances. Otherwise everybody would get lost in all the available design proposals. This initiative was received as very timely as this area is expanding to accommodate or integrate with other areas as well, such as ambient intelligence.

These benchmarks should include both datasets available in a public repository together with well-defined tasks and associated rules to be performed in a way similar to robot competitions. Also, suitability of the one—datasets—or the other—competitions—mainly depends on the given tasks. It is felt that the issue deserves further discussion. The emphasis should be in the comparison of different approaches/algorithms running on different hardware. Interest is increasing to form a working group or forum on networked robotics to deal with these issues.

Since networked robotics covers a number of areas, so a number of different benchmarks might be appropriate—e.g. online robots. In this sub-area one performance-oriented benchmark would be the time required for a remote robot system to respond to an instruction from an operator. A server could be set up that aimed to ‘attack’ a remote robot site with instructions (good and bad) and get a measure of reaction/return to the operator’s site. Such a site could be set up for different types of networked robotics scenarios. In this domain, performance is very much influenced by link bandwidth and inherent link delays, a fact that should clearly be taken into account by the benchmarking procedure in order not to render the validity of the benchmark useless.

This was also the topic of one of the workshops at the last ICRA’08 in Pasadena, entitled: Network Robot Systems: benchmarks and platforms toward Human-Robot Interaction, as a continuation of a successful discussion in the workshop in IROS2007 for the establishment of common testbeds or benchmarks, to extend this discussion to bring together the technological aspects of 'network robot systems' (such as a common testbed) with the human-centered aspects of Human-Robot Interaction.

For further details see: http://www.irc.atr.jp/icra08_nrs_workshop/
EUROPEAN ROBOTICS RESEARCH PLATFORMS

The idea of European Robotics Research Centers or Platforms had been discussed in the EURON community in the last years, as already reported in the 2006 second version of this document. Since service robots integrate a very large quantity of cutting-edge technologies, integration and maintenance of such systems is therefore a key challenge that has to be addressed for successful R&D in this field. In this context some discussions took place regarding the installation of some EU Robotics Centers that will integrate and maintain the most sophisticated platforms and offer them for experimentation to the European robotics research community. A simple way of ensuring similar conditions for comparing research could be that these centers would offer as a service to the community the possibility of using their platforms for benchmarking. A simple alternative could be to access remotely some hardware equipments that are made available to the community at large. This is for example the case of the initiative of the University of Siena for remote visual servoing benchmarks that has been described above.

Under the name European Robotics Research Platforms (ERRP) those consortia that provide the best proposal for an ERRP under certain specifications and boundary conditions would be funded. The motivations behind these ERRP are:

- Major steps forward in fundamental research in robotics require the availability of ever more expensive hardware and software platforms, as well as the integration of research results from ever more complementary disciplines.
- The investments in such platforms and human research power are too high for most research institutes.
- Too few potentially good researchers have access to such platforms.
- The European manufacturers of advanced robotics platforms miss too many opportunities to profit from the best research brains in Europe.

These ERRPs should provide, both, the infrastructure and the stimulus, to bring the best European robotics hardware and the best European brains in robotics together, with the minimum of technical and/or logistic constraints, in order to achieve major scientific and technological breakthroughs that can maintain/improve/create a leading role for Europe in some selected robotics application domains.

Needless to say, an ERRP would offer an ideal environment for conducting controlled experiments with:

- an agreed interesting problem.
- an agreed standard procedure, including agreement on:
  - choice of parameters.
  - what to measure.
  - how to measure it.
  - how to analyze the results.
- an agreed index of performance.

That is to say, for conducting benchmarks in robotics research.
A related interesting effort is the series of workshops on *Technical Challenges for Dependable Robots in Human Environments* sponsored by IARP and the IEEE Robotics and Automation Society. The last three editions held in 2005, 2007 and 2008 were co-sponsored by EURON. The organization aims to provide privileged interactions and ample room for discussion, with a limited number of attendees by invitation only, and a simple track program.

Since IARP, IEEE/RAS and EURON have similar interests and objectives regarding the dependability and usability of robots designed to operate in human environments, the three organizations joined together in 2005 to co-sponsor this workshop examining the technical and application issues of dependable robotics. In order to make advances in the application of new Human Centered robots, it is essential that they gain widespread acceptance from the community at large. This acceptance will only be achieved if, as well as meeting their functional requirements; robots are also dependable and usable. The research and development needed to achieve truly dependable robots is wide ranging and multidisciplinary. Research areas that can contribute to achieve dependable robots in human environments include Human Robot Interaction, intrinsically safe mechanisms, systems engineering, sensor interpretation, control, advanced locomotion, validation and verification, along with many other areas. However there are two features of Human Centered dependable robots that determine the challenge faced by the technologies. The first is that, unlike industrial robots, Human Centered robots work within the same space as humans and, indeed, are often designed to physically touch and interact with humans. This makes partitioning approaches to safety impossible to contemplate. Other methods for insuring the safety of users, participants and bystanders must be achieved. The second area defining the challenge is that Human Centered robots must be operated in the same environments as humans, rather than in ones that have been constructed to accommodate the robot. This potentially presents difficult navigation and locomotion challenges but, more importantly, means that these robots have to deal with uncertainty in their environment and in their task. A critical part of answering the dependability challenge presented by human interaction and human environments is the development of truly intelligent robots that can interact with their environment in a flexible manner and, ultimately, negotiate with users and bystanders.

The development of safety technology in the framework of various international standards is inevitable for making practical intelligent robots in human environments. It is also true; however, that only promoting current standard-oriented technology will bring the limitation to the future robotics of multidiscipline and wide applicability. The term dependability in itself, which was introduced to robotics at the first venue of the workshop, must cover overall intelligent machine characteristics: not only the current technology of reliability/safety enhancement but also future prospects of machine adaptability and autonomy for user satisfaction. This workshop series aims to discuss the present and future of dependability technology and resolve the issues on how dependable robots in human environments can be designed and should behave.
The main research topics in the workshop are:

- Definitions and evaluation of robot dependability
- Performance metrics
- Robust human-robot communication and interaction
- Architectures supporting robust autonomy
- Specification & verification of robot software
- Robust sensing and control
- Fault detection, diagnosis and exception handling
- Human factors
- Networked operations
- Robust cooperation between robots and between humans and robots

More information can be found at the official websites:
http://www.is.aist.go.jp/iarp2005/
http://wwwrob.brindisi.enea.it/iarp/dep07/
http://cs.stanford.edu/group/manips/iarp
PERMIS: PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS

Another relevant American effort is the PerMIS workshop series on Performance Metrics for Intelligent Systems sponsored by NIST, the IEEE and ACM. The workshop started in 2000 and in August 2007 the seventh edition took place. This workshop series is targeted at defining measures and methodologies of evaluating performance of intelligent systems. The 2008 edition will be held in August in Washington DC and Angel P. del Pobil and Fabio Bonsignorio are members of the Program Committee (only four out of its 21 members are from outside the USA).

The goals of PerMIS are to extend the body of knowledge pertaining to how to define, characterize, measure the intelligence of systems with emphasis on the theoretical developments in the area of performance measures and definitions relevant to intelligence and autonomy, complemented by a focus on experimental results applied to domain-specific systems.

This workshop seems more oriented towards knowledge, since it emphasizes metrics for evaluating the advanced methods of constructing the actionable knowledge from the data and information. Sensor development and data extraction with subsequent information acquisition is considered only one part of enabling the decision support processes accompanying the process of actuation. Discovery and disambiguation of the system of interacting agents coordinated within the set of goals is considered the essence of knowledge construction required for the successful decision making and performing the expected functions of the system. It is considered to be critical to be able to define the quality of the knowledge – its uncertainty, resolution, precision, accuracy, etc. since when constructing a model of the world, the system has to be able to weigh the relative merits of various inputs (clues) it receives as well as knowledge it already has.

Some fundamental ideas considered as focal themes at this workshop are:
- Models and Similarity Measures for Image Recognition.
- Models and Similarity Measures for Text Interpretation.
- Models and Similarity Measures for Situation Analysis.
- Algorithms and Processes of Generalization.
- Architectures of Intellect-like Computational Processes.
- Search for Exploring Bodies of Data, Information, and Knowledge.
- Hypotheses Generation and Disambiguation.

The seventh 2007 workshop focused on the interplay between autonomy and intelligence, i.e. how does autonomy influence intelligence and vice versa. The cardinal theme of this year’s workshop will be identifying and quantifying contributions of functional intelligence towards achieving success. Topic areas include:
- Defining and measuring aspects of a system:
  - The level of autonomy.
  - Human-robot interaction.
  - Collaboration.
• Evaluating components within intelligent systems:
  o Sensing and perception.
  o Knowledge representation, world models, ontologies.
  o Planning and control.
  o Learning and adapting.
  o Reasoning.

• Infrastructural support for performance evaluation:
  o Testbeds and competitions for intercomparisons.
  o Instrumentation and other measurement tools.
  o Simulation and modeling support.

• Technology readiness measures for intelligent systems.

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

For further information: the PerMis'04 White Paper is included at the end of this document as an Appendix V. Also more information can be found at the official websites:
http://www.isd.mel.nist.gov/PerMIS_2006/
http://www.isd.mel.nist.gov/PerMIS_2007/
http://www.isd.mel.nist.gov/PerMIS_2008/
SOFTWARE DEVELOPMENT IN ROBOTICS

The development of robotic systems has always been a great challenge due to the complex and heterogeneous technological issues involved. The process of bringing intelligence to a system requires strongly tighten capabilities of sensing, processing, and acting. In this scenario, software plays a key role as it is the medium to embody intelligence in the machine. The fast evolution of computing, sensor, and actuator hardware has made more difficult the problem of developing effective and efficient robotic software systems. There is an evident connection between widespread benchmarking and software development practice in robotics, since the former will only be possible if highly modular systems and applications are developed, systems and applications become interoperable, and robotics software is easily reusable.

The Robotics community is getting aware of the importance that software development principles have in building advanced robotics systems and there has been some activity in this regard, such as interesting initiatives in the field of open source development, distributed middleware, and standard architectures. An especially important event has been a series of three workshops on Software Development in Robotics held in conjunction with ICRA (2005, 2007 and 2008).

The workshops were intended to create a forum where researchers, practitioners, and professionals could discuss about principles and practice in the use of advanced software development techniques for building robotic systems by bringing together researchers from two different communities: Software Engineering and Robotics. The rationale was that Software engineering is today the Achilles heel of robotics, and it is reasonable to expect that in the near future major contributions in robotics will come from the software engineers. The claim was that what is still missing in robotics is a common language among researchers on this topic and a clear understanding of the state of the art of research in Software Engineering.

Consolidated software engineering techniques have proved their effectiveness in a variety of application domains and could be adopted to build robotics systems more effective. On the other hand, robotics software development is a valuable benchmark to assess the power and discover the limits of advanced software engineering techniques when used to design, implement, and test applications that control physical equipments interacting with the real world. The synergy between Robotics and Software Engineering is strategic. Their mutual benefit is not merely to make software systems bigger, faster, cheaper, but rather to make it possible to build and evolve new software systems. A number of new application areas are emerging from the robotic field, which strongly build on computer engineering technologies: Distributed Robotics, Internet Robotics, Rescue Robotics, Educational Robotics, Human-Robot Interaction.

The aim of this workshop series was two-fold:

1.- To point to the strategic role of advanced software development techniques as the basis of new robotic systems.
2.- To identify strategic directions for the Software Engineering research in order to meet the requirements of real world robotic systems.

These topics were discussed from both the Software Engineering and Robotics perspective. Nevertheless, the workshops did not present yet another software development process built on proprietary technologies or robotic projects. Instead, it aimed at showing how the state-of-the-art software development practice in robotics can meet the principles of software engineering.

The intended results of the workshops were a practical program of research, standardization, and public relations focused on how software development techniques are actually practiced in Robotics and in highlighting strategic directions to improve the synergy between Robotics and Software Engineering.

The following topics were discussed from the double perspective of the robotic expert and of the software engineering expert. The goal was to follow a roadmap that illustrates advanced techniques building modular, interoperable and reusable software applications and to discuss their applicability to the development of robotic systems:

- Crafting modular systems and applications.
- Integration of robotic systems and applications.
- Reuse of robotic systems and applications.
- Description of work in progress, innovative ideas, field-based studies, related to the adoption of Software.
- Engineering techniques in robotic software development.

The 2008 workshop reflects an increased awareness within the Robotics community for the importance of developing robotic software principles for large and complex robotics systems. As an emerging research field, robot software development is generating a growing body of scientific literature and industrial developments. Nevertheless, the field is still characterized by the lack of a sound and comprehensive body of concepts that has been widely adopted. As a consequence, it is rather difficult to understand, assess, and compare the existing approaches. In turn, this limits our ability to fully exploit them in practice, and to further promote the research work on robot software development.

This 2008 edition focused on a peculiar aspect of robot software development, that is design of real-time robot behavior. Usually, robot software design builds on high level abstractions such as concurrent execution, instantaneous computation, zero delay, and perfect communication between components and/or between components and the external environment. Real-time behavior is then achieved by experimentation and measurement on specific platforms in order to adjust design parameters. This is a typical scenario that limits the reusability of valuable robotic systems. The objective of this workshop was to identify real-time software requirements for robotic applications, to compare existing approaches, software environments and tools, and to discuss why current practice does not address the problem of designing reusable robot software in a satisfactory way.

The expected result is a practical program of research and public relations focused on the way that software development techniques are actually practiced in Robotics and a roadmap that indicates the strategic directions to pursue the synergy between Robotics and Software Engineering.

More information can be found at the official websites at ICRA 2005, 2007, 2008.
Radish: The Robotics Data Set Repository

Radish is an American initiative created by Andrew Howard and Nick Roy and intended as a repository of standard data sets for the robotics community. It started in May 2003. The goal is to facilitate the development, evaluation and comparison of robotics algorithms. The current focus is clearly on localization and mapping, although they expected that Radish would ultimately expand to reflect the interests of the broader robotics community.

Researchers are invited to download and make use of the available data sets, and also to make their own contributions to the repository. New data sets can be submitted through a submissions page and there also exists a mailing list for subscribers to receive news and notification of new file uploads.

Any file format is acceptable as long as it is sufficiently documented, but the preferred format seems to be that of CARMEN, the Carnegie Mellon Robot Navigation Toolkit which is an open-source collection of software for mobile robot control that provides basic navigation primitives including: base and sensor control, logging, obstacle avoidance, localization, path planning, and mapping.

Presently, the Radish repository contains a collection of some 39 data sets falling into four categories (the figure shows a sample):

- Logs of odometry, laser and sonar data taken from real robots.
- Logs of all sorts of sensor data taken from simulated robots.
- Environment maps generated by robots.
- Environment maps generated by hand (i.e., re-touched floor-plans).

Some examples of available data sets at Radish:

- Name: acapulco_convention_centre
  Desc: The site of the 2004 AAAI Robot Challenge

- Name: ap_hill_07b
  Desc: Multi-robot exploration and mapping, Fort AP Hill

- Name: ut_austin_aces3
  Desc: 3rd floor of ACES building on UT Austin campus.
Some examples of available data sets at Radish

The only exception is a data set from the NASA Robonaut during human teleoperation to perform 45 grasps trials. The provided data contains readings from various force and tactile sensors on the arm and hand on Robonaut used for grasping. The data is in MATLAB format.

Although no information about downloads is available, there is evidence that suggests that Radish is not being very active, with the number of uploads per year (i.e. contributed data sets) consistently decreasing, with only a few persons contributing since Jan. 2005 and a mailing list made up of 11 people.

More information can be found at the official website: http://radish.sourceforge.net
OPENSLAM: OPEN-SOURCE SLAM PACKAGES

A related initiative is OpenSLAM, a collection of open-source SLAM packages available at the OpenSLAM web-site since January 2007. Its goal is to provide a platform for SLAM researchers which give them the possibility to publish their algorithms. OpenSLAM provides to every interested SLAM researcher a subversion (svn) repository and a small webpage in order to publish and promote their work. In the repository, only the authors have full access to the files; other users are restricted to read-only access. OpenSLAM does not really aim to provide a repository for the daily development process of early SLAM implementations. Published algorithm should have a certain degree of robustness. OpenSLAM does not force the authors to give away the copyright for their code, but only require that the algorithms are provided as source code and that the authors allow the users to use and modify the source code for their own research. Any commercial application, redistribution, etc has to be arranged between users and authors individually.

- **CAS Robot Navigation Toolbox.** It's a GNU GPL licensed Matlab toolbox for robot localization and mapping. It is made for research and education and independent on the type(s) of feature and type(s) of sensors.

- **DP-SLAM** aims to achieve truly simultaneous localization and mapping without landmarks, it is compatible with techniques that correct maps when a loop is closed, and is accurate enough that no special loop closing techniques are required in most cases.

- **GMapping** is a highly efficient Rao-Blackwellized particle filter to learn grid maps from laser range data.

- **GridSLAM** is an easy to use and understand Rao-Blackwellized particle filter to learn grid maps from laser range data.

- **SLAM Package of Tim Bailey.** This package is a collection of implemented SLAM approaches by Tim Bailey. The code is written in MatLab and performs EFK, UKF, FastSLAM 1, and FastSLAM2.

- **CEKF-SLAM** is a Compressed Extended Kalman Filter-based SLAM simulator written under Matlab.

- **Thin Junction Tree Filters for SLAM.** This is a software package that implements a filtering technique that maintains a tractable approximation of the belief state as a thin junction tree.

- **TORO - Tree-based netwORk Optimizer-** is an optimization approach for constraint-network. It provides a highly efficient, gradient descent-based error minimization procedure. There is a 2D and a 3D version of TORO available.

More information can be found at the official website: [http://openslam.org/](http://openslam.org/)
REFERENCES

Alpha Puzzle, http://parasol.tamu.edu/groups/amatogroup/benchmarks/mp/alpha/
Burke, J.L., Murphy, R.R., Rogers, E., Lumelsky V.J., Scholtz, J., "Final Report for the
DARPA/NSF Interdisciplinary Study on Human-Robot Interaction", IEEE Trans. on
Systems, Man, and Cybernetics Part C, 34, 103-112, 2004
Cleaning Robot Contest, Oct 2002, Lausanne, Switzerland,
Ford Competition, http://www.op.dlr.de/FF-DR-ER/research/irobot/optmp.html
Geraersts, R., Overmars, M.H., “Sampling Techniques for Probabilistic Roadmap
Robotic Assistants”, accepted for publication in “Springer Tracts in Advanced
Robotics (STAR) Series”, Springer-Verlag, Heidelberg,
J. Baltes, “A Benchmark Suite for Mobile Robots”, IEEE International Conference on
Intelligent Robots and Systems (IROS'00), Takamatsu, Japan, 2000
IEEE Int. Conf. on Robotics and Automation, , Pages 3224-3229, Detroit, 1999.
MOVIE, http://www.give-lab.cs.uu.nl/movie/moviemodels/about.php
Proceedings of Third International Conference and Exposition on Robotics for
Reggiani, M. "A Proposal for a Motion Planning Markup Language", Technical report,
Reggiani, M. Mazzoli, S. Caselli, “An Experimental Evaluation of Collision Detection
Packages for Robot Motion Planning”, IEEE International Conference on
RoboCup Federation, http://www.rescuesystem.org/robocuprescue/
S. Hanks, M. E. Pollack, P. R. Cohen, “Benchmarks, Test Beds, Controlled Experimentation,
and the Design of Agent Architectures”, AI Magazine Volume 14, Issue 4, Winter 93
The Federation of International Robot-soccer Association (FIRA),
http://www.fira.net./about/overview.html
APPENDICES
APPENDIX I:

SOCCER COMPETITIONS RULES
SOCCER COMPETITIONS RULES:

The rules of each robot soccer competition are based in the FIFA laws and each competition makes its own modifications. Although having their own modifications the main rules are basically the same in all competitions changing only the field of play, the ball, the number of players, the players’ equipment and the duration of the match and other rules depending on the competition. The rules are classified in:

1. The Field of Play
2. The Ball
3. The Number of Players
4. The Players’ Equipment
5. The Referee
6. The Assistant Referees
7. The Duration of the Match
8. The Start and Restart of Play
9. The Ball In and Out of Play
10. The Method of Scoring
11. Offside
12. Fouls and Misconduct
13. Free Kicks
14. The Penalty Kick
15. The Throw-In
16. The Goal Kick
17. The Corner Kick

1. The Field of Play

This rule differs in each competition. This rule includes:

a) dimensions
b) field markings
   a. safety boundary
   b. ad panels
   c. restart spots
c) the goal area
d) the penalty area
e) flag posts
   a. corner flag posts
f) the corner arc
g) goals
h) safety
i) others
   a. surface
   b. lighting
2. The Ball
   a) Qualities and measurements
   b) Replacement of a defective ball

3. The Number of Players
   a) Players
      a. Incapable players
   b) Official competitions
   c) Other matches
   d) All matches
   e) Substitution procedure
   f) Infringements/sanctions
   g) Restart of play
   h) Players and substitutes sent off
   i) Decisions of the international F.A. Board

4. The Players Equipment
   a) Safety
      a. Jamming
      b. Exclusion
   b) Basic equipment
      a. Design guideline
      b. Space occupied by a robot
      c. Robot size
      d. Robot shape
      e. Robot weight
      f. Overweight robots
      g. Robot colours
      h. Robot markers
      i. Colour markers
      j. Number markers
      k. Top markers
      l. Communications
      m. Sensing systems
      n. Ball handling mechanisms
      o. Technical inspection
   c) Shin guards
      a. robustness
   d) Goalkeepers
   e) Infringements/sanctions
      a. Repair of robots
   f) Restart of play

5. The Referee
   a) The authority of the referee
   b) Powers and duties
      a. Referee box
      b. Play without referee box
   c) Decisions of the referee
6. The Assistant Referees
   a) Duties
   b) Assistance

7. The Duration of the Match
   a) Periods of play
   b) Half time interval
   c) Allowance for time lost
   d) Penalty kick
   e) Extra time
   f) Abandoned match

8. The Start and Restart of Play
   a) Preliminaries
      a. Start delay
      b. Remote start
   b) Kick-off
   c) Procedure for kick-off
      a. Positioning of robots
      b. Manual start
   d) Infringements/sanctions for kick-off
   e) Dropped ball
      a. Game stuck
   f) Procedure for dropped ball
   g) Infringements/sanctions for dropped ball
   h) Special circumstances

9. The Ball In and Out of Play
   a) Ball out of play
      a. Dead call
      b. Continuation after dead call
   b) Ball in play

10. The Method of Scoring
    a) Goal scored
    b) Winning team
    c) Competition rules

11. Offside
    a) Offside position
    b) Offence
    c) No offence
    d) Infringements/sanctions

Note: this law does currently not apply in any robotic tournament
12. Fouls and Misconduct

a) Direct free kick
b) Penalty kick
c) Indirect free kick
   a. Ball holding
   b. Pushing
c. Kicking
d. Illegal defense
e. Illegal attack
f. Manual interference
g. Remote interference
h. Unsportsmanlike behavior
d) Disciplinary sanctions
e) Cautionable offences
   a. Yellow cards
f) Sending-off offences
   a. Temporary sent-off
   b. Sending-off offences
g) Notes
   a. General principles
   b. Stopping the ball
c. Dribbling the ball
d. Kicking the ball

13. Free Kicks

a) Types of free kicks
   a. Touching
b) The direct free kick
c) The indirect free kick
d) Position of free kick
   a. Procedure
e) Infringements/sanctions

14. The Penalty Kick

a) Position of the ball and the players
b) The referee
c) Procedure
d) Infringements/sanctions

15. The Throw In

a) Procedure
b) Infringements/sanctions
16. The Goal Kick
   a) Procedure
   b) Infringements/sanctions

17. The Corner Kick
   a) Procedure
   b) Infringements/sanctions
APPENDIX II:

ROBOCUP RESCUE GENERAL RULES
ROBOCUP RESCUE GENERAL RULES

TEAM SETUP FOR MISSIONS

To maintain an ambitious schedule of missions, and ensure that team setup for each mission is timely and efficient, the following rules apply:

- Teams may trade time slots with other teams if mutually beneficial – but both team leaders must notify the chair at least one mission prior to the negotiated mission start time
- Failure to be ready for any scheduled mission scores (0) for that mission
- Teams should have their robots and operator equipment on a rolling cart at least 15 minutes prior to their mission.
- Teams should wait in the team preparation area until a league official asks you to approach the operator station.
- Teams will have 10 minutes to move into position and set up at the operator station (while the previous team exits from an adjacent area)
- Teams must demonstrate all functional robot sensing, localization, and mapping capabilities to the Judge prior to the start each mission

COMPETITION ROUNDS AND MISSIONS

To provide multiple search missions for each team, require modest robot stamina, encourage easy set-up/break-down, and allow some chance of failure without consequence, the competition has the following format:

- Each mission lasts 20 minutes (plus 10 minutes for set-up)
- Each competitive round consists of 2 or 3 missions
- In preliminary rounds, you will likely be able to drop one mission score
- A pre-determined number of teams with the highest scores advance to the next round of competition (or the chair may apply a minimum threshold score for advancement based on overall scoring results)
- The number of competitive rounds and missions per round may change depending upon days available and number of teams

NOTE: The team that wins should demonstrate effective and reliable implementations over several 20 minute missions (more than 1 hour of operation over a couple of days)
SQUARE LAYOUT AND START POINTS

COLUMN LAYOUT AND START POINTS
MISSION START POINTS

To similarly test mapping and planning capabilities, while allowing robots better access to their intended arenas, each round of competition will proceed as follows:

- **First mission of every round**: Teams must begin at START POINT [A] and must negotiate the Yellow arena.
- **Middle mission of every round (if any)**: Teams must begin at START POINT [B], between the Yellow and Orange arenas. Teams may enter either arena.
- **Last mission of every round**: Teams may choose their start point, START POINT [A], [B], or [C], but may not repeat their previous start point.

NOTE: After starting, all teams must follow the rule of Advancing & Retreating

ADVANCING & RETREATING

To promote collaboration between robots, and deter parallel teleoperation in separate arenas, the following rules apply:

Advancing to more difficult arenas:

- Robots are always free to advance to the next most difficult arena, but they must earn it by leaving the simpler arena through the door on the far side of the arena from their start point.
- Robots may always advance without the entire team of robots

Retreating to simpler arenas:

- Robots are always free to retreat to a simpler arena already negotiated during the current mission
- Retreating to a simpler arena not successfully negotiated during the current mission must be done as a team (all robots gather at the mission start point)
before entering the simpler arena). Teams may need to use “RESETs” if necessary to retrieve robots stuck in the more difficult arena. Once retreating, robots may retreat as far as they can without the entire team of robots.

**RESETS**

Operator can call ‘RESET’
- Judge returns robot to starting point
- Time continues to run
- Penalty: add one operator in score

Self-Reset
- Robot can return to starting zone by itself for operator repair
- Operator can continue setup during mission time
- Penalty: none

‘Out of Bounds RESET’
- Occurs when a robot leaves both the ‘HOT’ and ‘WARM’ zones
- Imposed at the discretion of the judge
- Penalty: add one operator in score

**NEGOTIATING ARENAS TO FIND VICTIMS**

Since the arenas are small compared to a building, and there are several ways to thwart the intention of the arena design, the following rules apply:
- Robots must pass under crossbars or through other obvious portals when traveling through arenas
  - Robots must enter the same area as the victim for identification. No victim identifications allowed:
    - Over maze walls
    - Through glass walls
    - Through mesh walls or netting
    - Looking over obstacles (not walls) or through access holes is encouraged
- Robots must surmount an elevated level to identify victims on that level, unless the robot is looking up/down through access holes such as:
  - Elevated floor holes
  - Into box obstacles
- Knowing (seeing) a victim is there does not mean you have identified that victim or any particular signs of life. Keep searching for a way to get into the same room, or onto the same level, as the victim.
SIMULATED VICTIMS

Operator must SHOW all perceived signs of life to the Judge
- Judges (with the operator) note the validity of the call based on the information shown in the operator interface
- Referees (with the robot) note the order that the victims are found and what signs of life are available on any given victim

Victims are not counted twice, even if found by a different robot

SIGNS OF LIFE:
- Form: Shape, colour ...
- Motion: Moving appendages...
- Heat: Body heat (heating blankets)
- Sound: Voice, beacons, tapping
- Chemical: CO2 emissions

SITUATIONS:
- Surface
- Trapped
- Void
- Entombed

STATES:
- Aware
- Semi-conscious
- Unconscious
PERFORMANCE METRIC

The intent of the performance metric for this competition is to encourage certain desirable robot capabilities while discouraging certain unhelpful team strategies. There are (50) points available for each victim found:

- (20) points for Mapping reward map QUALITY and accurate LOCATION of victims and features
- (15) points for Mobility reward capabilities required to identify the victim SITUATION and for advanced mobility required to read the VICTIM TAG.
- (15) points for Sensing reward individual sensor capabilities and for correctly identifying the victim STATE

NOTE: Points may be deducted for errant identifications, so be sure of what you are reporting. Would you send in a human rescuer based on the information you’re reporting?… that’s the question.
MAP QUALITY

MAP QUALITY (10 of 50 pts per victim)
Refers to the paper-based map of the arenas submitted to the Incident Commander (Judge) within two minutes after the end of your mission time expires. All maps should indicate the following:

- Victim LOCATION clearly marked with reference to VICTIM DATA SHEET. Pertinent features (doors, windows, stairs, collapses, etc.)
- DO NOT start with a line denoting the perimeter of the arena
- Hint: Could an audience member use your map?…does it have all the necessary information on it?

SCORING

(10 Points) Fully automatic, robot sensor generated, accurate map of arena interiors automatically showing victim locations and reference to VICTIM DATA SHEET. Human labeling of obstacles and features only. No corrections of map
(5 Points) Robot sensor generated, human interpreted map of the arena interiors. Victim locations, obstacles and features may be hand written. Hand corrections may be discounted by the Judge if considered influenced by operator knowledge of the arena
(1 Point) Human generated map (hand or computer drawn) of the arenas or topological directional information to the victim.

VICTIM LOCATION

VICTIM LOCATION (10 of 50 pts per victim)
Refers to the Mapped location of a found victim, which should indicate any part of a found victim to within 1 cubic meter. Note that the Incident Commander (Judge) will USE your map to find these victims. If your map is not clear enough to follow, you will not score ANY points for that victim.

Hint: For accuracy, reference locations from easily identifiable arena features rather than from the start point.

SCORING

(10 points) Locating a victim to within 1 cubic meter
(5 points) Adjacent cube is called (not through walls)
(1 point) Any other cube is called
VICTIM SITUATION

VICTIM SITUATION (5 of 50 pts per victim)
Requires understanding the victim’s rescue needs by discerning what type of rescue SITUATION a victim is in:
- SURFACE: Entirely visible (head/torso and legs or baby)
- TRAPPED: Partially visible under light rubble (head/torso)
- VOID: Minimally visible in void under collapse `(legs or baby)
- ENTOMBED: Not visible without probing(arm, sound, heat, CO2)
- UNKNOWN

VICTIM STATE

VICTIM STATE (15 of 50 pts per victim)
- Requires identifying a victim’s sensor signatures and increasing confidence through multiple sensor signatures
- If you correctly identify at least (3) sensor signatures you may attempt to determine the victim’s STATE for bonus points.
  - FORM (+/-1 POINT)
  - MOTION (+/-1 POINT)
  - HEAT (+/-3 POINTS)
  - SOUND (+/-2 POINTS)
  - CO2 (+/-3 POINTS)
  - VICTIM STATE (5 POINTS)
    - AWARE: Fully conscious and moving (arm waving and/or yelling)
    - SEMI-CONSCIOUS: Not aware but may be moving (finger moving or moaning)
    - UNCONCIOUS: No motion, no sound (has heat and may have alarm and CO2)
    - UNKNOWN

VICTIM TAG

VICTIM TAG (+/-10 of 50 pts per victim)
All victims have VICTIM TAGS prominently displayed, but may not be easily visible. Achieving the proper viewing position may be an extreme test in mobility. If you can read the tag along with all your other signs of life, score (+10) points.

Note, however, that identification tags are also placed throughout the arenas as likely locations to search for victims (analogous to a rescue dog handler’s pointing motion).

Hint: Be careful not to be quick to identify VICTIM TAGS from a distance, because reporting a VICTIM TAG that is not associated with a victim will cost you (–10) points. So be sure you identify other signs of life in addition to the VICTIM TAG.
For this competition, VICTIM TAGS will display numbers.
EFFECTIVE ARENA MAPPING (Set-up Plan)
EFFECTIVE ARENA MAPPING (LADAR Map)

EXTREMELY GOOD AUTOMATIC MAPPING OF WALLS AND FEATURES, ADD VICTIM POSITIONS FOR MAXIMUM POSSIBLE SCORE

EFFECTIVE ARENA MAPPING (Plan and Map)
EFFECTIVE ARENA MAPPING (EXAMPLE)

THIS MAP, AUTOMATICALLY GENERATED, WOULD SCORE MAXIMUM POINTS!

PENALTIES

ARENA BUMPING
- Uncontrolled Bumping (-5 points per incident) Example: Undesirable contact with environment that does not result in damage
- Heavy Damage (-20 points per incident)
  Example: Undesirable shifting or damage to environment

VICTIM BUMPING
- Bumping Victim (-5 points per incident)
  Example: Contact with a victim’s torso, legs, or head (not hands or feet)
- Harming Victim (-20 points per incident)
  Example: Contact that clearly repositions or “harms” a victim

Penalties May Compound
- Example: Causing ‘Heavy Damage’ (-20 points) to arena which results in ‘Harming’ a victim (-20 points) = 40 point deduction
PROCEDURES, DEFINITIONS AND OTHER STUFF

SUGGESTED TEAM PROCEDURE

Victims are found by following all the steps below (in suggested order):
Determine:
1) VICTIM SITUATION (senor: surface, trapped, void, entombed)
2) VICTIM STATE (sensors: aware, semi-conscious, unconscious)
3) VICTIM TAG (operator)
4) VICTIM LOCATION (sensors and/or operator)
Then:
5) Map the VICTIM LOCATION
6) Complete the VICTIM data sheet
7) Notify the Incident Commander you have found a victim (identify the victim using the VICTIM TAG)
8) Show the Incident Commander a view of the TAG, SITUATION, all sensor readings leading to a STATE, and how you determined your LOCATION

ADMINISTRATIVE PROCEDURE

1) Teams submit all VICTIM data sheets
2) Teams submit MAP for each arena
3) Referees submit sheets detailing victims found and penalties assessed
4) Incident Commander (Judge) follows map to find victims and scores accordingly

DEFINITIONS: HOT ZONE

Yellow Arena
- 2-D maze with no flooring issues
- Arena weighting = 0.50

Orange Arena
- 3-D maze with variable household/office flooring
- Arena weighting = 0.75

Red Arena
- Totally unstructured and unstable
- Arena weighting = 1.00

NOTE: No team members allowed in the field of play once competition starts!
**DEFINITIONS: WARM ZONE**

Operator station
- Faces away from “Hot Zone”
- Only essential team operators should be present during a mission
- Everybody who enters the warm zone during a mission will count as an operator during that mission

Starting Point
- All team members may place and initialize the robot prior to the mission

**NOTE:** Any member of a team found in the “Warm Zone” during another team’s mission will be penalized at the discretion of the Chairs.

**DEFINITIONS: COLD ZONE**

- Contains TEAM PREPARATION ROOMS and STAGING AREA
- All team members that are not acting as operators, have the option to observe their mission and any subsequent team missions of a round.
- All observing team members must stay in the spectator areas and may not interact with their operator.
- Any team members that may wish to be additional operators during the course of their mission, must remain in the “Cold Zone” until requested.

**DEFINITIONS: OPERATORS**

The intent of this rule is to encourage an increase in the ratio of robots to operators by demonstrating bounded autonomy and high level management of multiple robots
- Any person present in the “Warm Zone” during a mission
- Any person who touches, interacts with, or controls the robot during a mission
- Any person who helps prepare the map or fill in the VICTIM sheet (see next page)
### VICTIM

#### TAG

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Indicator</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO₂</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### STATE

If you have (3) of the above indicators choose:

- **Aware**
- **Semi**
- **Unconscious**
- **Unknown**

### MAPPING

#### SURFACE

- Entirely visible: [+5]
- Partially visible under rubble: [+5]
- Partially visible in void: [+5]
- Entombed: Visible only probing: [+5]
- Unknown: [0]

### SITUATION

#### SURFACE

- Entirely visible: [+5]
- Partially visible under rubble: [+5]
- Partially visible in void: [+5]
- Entombed: Visible only probing: [+5]
- Unknown: [0]
DEFINITIONS: REFEREES

- Either organizing officials or non-competing team members
- Responsibilities
  - tracks the robot through the mission
  - notes victim identifications
  - assigns penalties (arena damage and victim harm)
- One referee per robot
- Must observe in a non-interference manner
- Fills in the REFEREE sheet

REFEREE

<table>
<thead>
<tr>
<th>ROBOT NAME</th>
<th>VIDEOGRAPHER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>VICTIMS FOUND</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>V-1</td>
</tr>
<tr>
<td>V-2</td>
</tr>
<tr>
<td>V-3</td>
</tr>
<tr>
<td>V-4</td>
</tr>
<tr>
<td>V-5</td>
</tr>
<tr>
<td>V-6</td>
</tr>
<tr>
<td>V-7</td>
</tr>
<tr>
<td>V-8</td>
</tr>
<tr>
<td>V-9</td>
</tr>
<tr>
<td>V-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YELLOW ARENA PENALTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUMPING (-5 PTS)</td>
</tr>
<tr>
<td>DAMAGE (-20 PTS)</td>
</tr>
<tr>
<td>HURTING (-5 PTS)</td>
</tr>
<tr>
<td>HARMING (-20 PTS)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORANGE ARENA PENALTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUMPING (-5 PTS)</td>
</tr>
<tr>
<td>DAMAGE (-20 PTS)</td>
</tr>
<tr>
<td>HURTING (-5 PTS)</td>
</tr>
<tr>
<td>HARMING (-20 PTS)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RED ARENA PENALTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUMPING (-5 PTS)</td>
</tr>
<tr>
<td>DAMAGE (-20 PTS)</td>
</tr>
<tr>
<td>HURTING (-5 PTS)</td>
</tr>
<tr>
<td>HARMING (-20 PTS)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REFEREE NAME</th>
<th>REFEREE TEAM</th>
<th>REFEREE SIGNATURE</th>
</tr>
</thead>
</table>
**DEFINITIONS: JUDGE**

- An organizing committee member
- Responsibilities during each mission
  - starts the official time
  - only official allowed to interact with the operator(s)
  - relays to the referees that a potential victim has been found
  - see JUDGE sheet
- Responsibilities after each mission
  - interprets the map to seek each victim
  - determines the victim location and map quality
  - calculates the score
  - see SCORE sheet
- Has final authority over any disputes

### JUDGE

#### VICTIMS FOUND

<table>
<thead>
<tr>
<th>V</th>
<th>BOT</th>
<th>ARENA</th>
<th>VICTIM IDS</th>
<th>NOTES</th>
<th>V</th>
<th>BOT</th>
<th>ARENA</th>
<th>VICTIM IDS</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-7</td>
<td>Y-O-R</td>
<td>F-M-H-S-C-TAG</td>
<td></td>
<td></td>
<td>V-17</td>
<td>Y-O-R</td>
<td>F-M-H-S-C-TAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V-10</td>
<td>Y-O-R</td>
<td>F-M-H-S-C-TAG</td>
<td></td>
<td></td>
<td>V-20</td>
<td>Y-O-R</td>
<td>F-M-H-S-C-TAG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX III:

HUROSOT CHALLENGES
HUROSOT CHALLENGES

Laws of the Game for Forward - Backward Robot Dash

In the forward - backward robot dash, the robot must walk first forward into the end zone and then return to the start zone by walking backwards.

RD-1 The Field of Play

RD-1.1. The dimensions of the playing field are at least 220cm by 180cm.
RD-1.2. The playing field consists of a race track that runs the whole width of the playing field.
RD-1.3. There are two zones on the ends of the race track. The zones are marked by lines parallel to the goal lines.
RD-1.4. The width of the zones is the width of the playing field and the length of the zones is at least 30cm.
RD-1.5. The length of the race track depends on the height of the robots. The racetrack is parallel to the side lines.
(a) The length of the racetrack for small robots as defined in Law-4.2 is 120cm.
(b) The length of the race track for medium robots as defined in Law-4.2 is 170cm.
(c) The length of the racetrack for large robots as defined in Law-4.2 is 200cm.
RD-1.6. All lines are marked with 5mm to 15mm wide white masking tape or are painted in white on the playing surface.
RD-1.7. Teams may place small coloured or infra red markers in the end zones to guide the robot as long as they do not interfere with other teams.

RD-2 Game Play

RD-2.1. At the beginning of the competition, all robots must be behind the start line (i.e., in Zone 1) of their respective categories (i.e., small, medium, or large).
RD-2.2. The referee will signal the start of the competition by blowing the whistle. After the referee blows the whistle, the robots walk forwards towards the end of segment 1 (i.e., zone 2).
RD-2.3. A robot is not allowed to leave the playing field as defined in RD-1. If a robot leaves the playing field, it must be moved back to the start zone.
RD-2.4. A robot has crossed the end line of one segment when either foot of the robot crosses the finish plane and touches the ground in the respective zone. The finish plane is the plane which intersects the playing field at a 90 degree angle at the back of the finish line.
RD-2.5. After the robot has crossed the finish line of the first segment (i.e., the robot has reached Zone 2), the robot must walk backwards towards the end line for segment 2 (i.e., Zone 1).
RD-2.6. A robot is walking backwards if the difference of the robot’s current orientation to its orientation when positioned at the start of the challenge in Zone 1 is at most 90 degrees in either direction.
RD-2.7. The robot must move forward towards in segment 1, and backwards in segment 2.
RD-2.8. Each robot may have at most one human handler associated with it.
RD-2.9. The human handlers are not allowed to interfere in any way with other robots, the referee, or other human handlers.
RD-2.10. A human handler may only enter the playing field or touch his/her robot with the permission of the referee.
RD-2.11. The handler shall remove his/her assigned robot as soon as possible from the respective end zone after it has crossed the finish line.
RD-2.12. Any robot that either leaves the playing field, breaks down, or falls over may be moved by a human helper and placed again behind the start line. This is subject to laws RD-2.9 and RD-2.10.
RD-2.13. The end of the competition is signaled by the referee by blowing the whistle a second time. The referee terminates the competition if
  • the maximum duration of the competition (10 minutes) has elapsed.
  • all robots have crossed the finish line of the backward segment,
  • no more active robots remain in the competition.

RD-3 Method of Scoring

RD-3.1. Robots are awarded points based on the last segment that the robot completed successfully as well as the order in which they crossed the end line of the last segment.
RD-3.2. All robots that have not crossed the finish line of the first segment are automatically awarded 0 points.
RD-3.3. The point allocation for robots is as follows:
  • The first place robot is declared the winner and is awarded 10 points.
  • The second place robot is awarded 8 points.
  • The third place robot is awarded 6 points.
  • The fourth, fifth, sixth, and seventh place robots are awarded 4, 3, 2, and 1 point respectively.
  • In case of a tie between n robots, the points accumulated for the n succeeding places are divided equally between the competitors. For example, in case two robots both cross the finish line in 2nd place, then the points for 2nd and 3rd place shall be divided between the two robots. In this case, each robot receives \((8 + 6)/2 = 7\) points.

RD-4 Fouls and Misconduct

RD-4.1. A robot is not allowed to interfere with another robot in any way.
RD-4.2. Light contact: Should the contact between two robots be light and infrequent, then the referee uses the following rules:
  (a) should a contact occur between two robots where one robot is deemed responsible by the referee for the offense (for example, the other robot is stationary), then the offending robot must be moved back behind the start line and continue with segment 1. The robot may then continue in the competition.
  (b) if both robots are moving, then the referee will have both robots moved behind the start line by the human handlers. Once both robots have been moved behind the start line, they may then continue in the competition by restarting segment 1.
RD-4.3. The referee may use the penalties described in Law-8 accordingly.
Dec-4.1. In 2006, the difficulty level of the robot dash has been increased.
  • The robot has to autonomously switch from walking forwards to walking backwards one it reaches the end of segment 1. No human intervention is allowed.
• A robot that falls during the backward leg has to be moved back to start line (i.e., it has to start with the forward leg again).
The motivation for these changes is to reward teams with more autonomy. Furthermore, it simplifies the running and organizing of the event.

Dec-4.2. If it is impossible for two or more robots to race concurrently (for example, too many entries for a single heat), then the robots shall race separately and their performance shall be compared based on time.

**Laws of the Game for Penalty Kicks**

**PK-1 The Field of Play**

PK-1.1. The dimensions of the playing field are at least 220cm by 180 cm.
PK-1.2. One side of the playing field contains a goal. This side of the playing field shall be called the goal side. The opposite side of the playing field is called the empty side. The two other sides are called side lines.
PK-1.3. The goal is 100cm wide and is placed on the goal side of the playing area with its center along the center line of the playing field.
PK-1.4. The penalty mark is 75cm away from the goal line as described in Law-1.12.
PK-1.5. The penalty area is specified by the triangle that extends from the penalty mark to the top left and top right corner points of the goal area.
PK-1.6. Unless otherwise stated, the field of play is as described in Law-1.

**PK-2 Game Play**

PK-2.1. One robot is designated the kicker, another robot is designated the goal keeper. All other robots must be positioned as described in Law-9.4e and must not interfere with the designated kicker or goal keeper in any way.
PK-2.2. The only robots allowed to move during a penalty kick are the designated kicker and goal keeper.
PK-2.3. Each robot may have at most one human handler associated with it.
PK-2.4. The human handlers must not interfere in any way with other robots, the referee, or other human handlers.
PK-2.5. A human handler may only enter the playing field or touch his/her robot with the permission of the referee.
PK-2.6. The kicker robot must be at the start point at the beginning of the competition. The start point for:
• small robots is 15cm behind the penalty mark for small robots in the center of the playing field.
• medium robots is 20cm behind the penalty mark for medium robots in the center of the playing field,
• large robots is 30cm behind the penalty mark for large robots in the center of the playing field.
PK-2.7. The designated goal keeper must be positioned so that a part of the robot touches the goal line at the start of the game.
PK-2.8. The referee will place a ball (as described in Law-2) at a random position inside of the penalty area.
PK-2.9. The designated kicker is not allowed to leave the playing field or enter the goal area.
PK-2.10. The designated goal keeper is not allowed to leave the goal area.
PK-2.11. The designated goal keeper must remain in a standard walking posture until the ball has been kicked.
PK-2.13. The penalty kick begins by the referee blowing a whistle.
PK-2.14. The end of the penalty kick is signaled by the referee by blowing the whistle a second time. The referee terminates the penalty kick if:
• a goal has been scored by the kicker,
• the ball moved outside of the playing field,
• a robot falls over or is immobilized by a technical defect,
• a robot leaves the playing field,
• the maximum duration of the competition (2 minutes) has elapsed.
• at least 1 minute has elapsed since the start of the competition and it is unlikely in the opinion of the referee that the kicker will score in the next minute,
PK-2.15. After the end of the penalty kick, the next robot is designated the kicker.
PK-2.16. After all other players have played against a robot as goal keeper, a new robot will be designated the goal keeper.

PK-3 Method of Scoring

PK-3.1. The number of rounds in the competition is dependent on the number of robots in the event. Each robot will take a penalty kick against all other robots and will play goal keeper against all other robots at least once. So, if there are five robots in the competition, each robot will take four penalty kicks and there will be 20 kicks in total. Each robot receives one point for each goal that was scored according to law Law-7.
PK-3.2. Any robot that has not scored a single goal is automatically awarded 0 points.
PK-3.3. Among the robots that have scored at least one goal:
• The robot with the most goals is declared the winner and is awarded 10 points.
• The robot with the second highest score (2nd place) is awarded 8 points.
• The robot with the third highest score is awarded 6 points.
• The fourth, fifth, sixth, and seventh highest scoring robot is awarded 4, 3, 2, and 1 point respectively.
• In case of a tie between n robots, the points accumulated for the n succeeding places are divided equally between the competitors.
For example, in case two robots both score three goals with two robots scoring more than three goals, then the points for 3rd and 4th place shall be divided between the two robots. In this case, each robot receives $(6 + 4)/2 = 5$ points.

Decisions
Dec-3.1. A robot may follow up on the ball, that is, should the first shot miss, but still be inside of the playing field, then the kicker can repeatedly approach the ball and kick toward the goal.
Dec-3.2. A robot may also re-approach the ball if it missed hitting the ball in the first attempt.
Dec-3.3. The goal keeper is not allowed to squat or try to block a large part of the goal until the ball has been kicked for the first time.
Dec-3.4. During the time between the start of the penalty kick and the time that the ball has been kicked, the robot may move freely in the goal area as long as it remains in a standard walking posture.
Laws of the Game for Obstacle Run

OR-1 The Field of Play

OR-1.1. The dimensions of the playing field are at least 220cm by 180cm.
OR-1.2. There is a finish line marked on one side of the playing field. This side of the
playing field is called the finish side. The opposite side of the playing field is called the
start side. The two other sides are called side lines.
OR-1.3. There is a 30cm zone behind the finish line, which is called the end zone.
OR-1.4. There are three start points marked on the playing field. The start points are in
the center of the playing field and the distance between the start points and the finish
line depends on the category of the robot.
(a) The start point for small robots as defined in Law-4.2 is 120 cm in front of the finish
line.
(b) The start point for medium robots as defined in Law-4.2 is 140 cm in front of the
finish line.
(c) The start point for large robots as defined in Law-4.2 is 165 cm in front of the finish
line.
OR-1.5. All lines and points are marked with approximately 1cm wide white masking
tape or are painted on the playing surface.
OR-1.6. Teams may place small coloured or infra red markers in the area behind the end
zone to guide the robot as long as they do not interfere with other teams.

OR-2 Obstacles

OR-2.1. An obstacle is a block or cylinder with a maximum length/width or in case of a
cylinder a diameter of approximately 30cm, and a height of approximately 30cm.
OR-2.2. The colour of the obstacle is red or blue.
OR-2.4. The referee or a person designated by the referee shall place five obstacles (see
OR-2) at random in the playing field.
OR-2.5. The obstacles may be placed anywhere on the playing field from the start point
to the finish line given the following constraints:
• a circular region with a radius of at least 30cm, 40cm, and 50cm for small, medium,
and large robots respectively around the starting point is free of obstacles,
• at least one of the obstacles shall be in the direct path of the robot to the finish line,
• there is at least one free path from the start point to the finish line. That is, a circle
with a diameter of 40cm can be moved from the start point to the finish line without
touching any obstacle. Note that this does not imply that the minimum distance between
two obstacles is at least 40cm. Some obstacles may be closer together than 40cm

OR-3 Game Play

OR-3.1. A single robot is designated the runner. All other robots must be outside of the
playing field.
OR-3.2. The only robot allowed to move during a run is the designated runner.
OR-3.3. Each robot may have at most one human handler associated with it.
OR-3.4. The human handlers are not allowed to interfere in any way with other robots,
the referee, or other human handlers.
OR-3.5. A human handler may only enter the playing field or touch his/her robot with
the permission of the referee.
OR-3.6. At the beginning of the competition, the designated runner must be at the start point for its respective category. The runner must face forward. (See OR-1.4).

OR-3.7. After the robot has been placed, the obstacles will be distributed by the referee according to OR-2.5.

OR-3.8. The referee will signal the start of the competition by blowing the whistle.

OR-3.9. A robot is not allowed to leave the playing field as defined in OR-1.

OR-3.10. A robot has crossed the finish line when either foot of the robot crosses the finish plane and touches the ground in the end zone. The finish plane is the plane which intersects the playing field at a 90 degree angle at the back of the finish line.

OR-3.11. The handler shall remove his/her assigned robot as soon as possible from the end zone after it has crossed the finish line.

OR-3.12. The end of the competition is signaled by the referee by blowing the whistle a second time. The referee terminates the competition if:

• the robot has crossed the finish line,
• the maximum duration of the competition (three minutes) has elapsed,
• the robot falls over or is immobilized by a technical defect,
• the robot leaves the playing field,
• the robot touches one of the obstacles (In case the obstacle was moved by the robot, the obstacle will be repositioned by the referee),
• at least two minutes have elapsed since the start of the competition and it is unlikely in the opinion of the referee that the robot will cross the finish line within the minute,

**OR-4 Method of Scoring**

OR-4.1. At the end of the run, another robot will be designated the runner.

OR-4.2. There are five rounds in the competition. Each round consists of all robots being designated the runner exactly once. Each robot receives one point for each run in which it manages to cross the finish line.

OR-4.3. Any robot that has not reached the finish line at least once is automatically awarded 0 points.

OR-4.4. Among the robots that have reached the finish line at least once:

• The robot with the most runs is declared the winner and is awarded 10 points.
• The robot with the second highest run number (2nd place) is awarded 8 points.
• The robot with the third highest run number is awarded 6 points.
• The robots with the fourth, fifth, sixth, and seventh highest run number are awarded 4, 3, 2, and 1 point respectively.
• In case of a tie between n robots, the points accumulated for the n succeeding places are divided equally between the competitors. For example, in case two robots both score three successful runs with no other robot scoring more runs, then the points for 1st and 2nd place shall be divided between the two robots. In this case, each robot receives \((10 + 8)/2 = 9\) points.
Laws of the Game for the Lift and Carry Competition

LC-1 The Field of Play

LC-1.1. The dimensions of the playing field are at least 190cm by 180cm.
LC-1.2. An approximately 1.00m by 1.00m wide uneven terrain is placed by the referee in the playing field (See Fig. 7.
LC-1.3. The uneven terrain consists of sheets of hard material such as corrugated plastic, corrugated cardboard, or wood.
LC-1.4. The thickness of a single sheet is between 5mm to 10mm.
LC-1.5. The uneven terrain is constructed by placing random cut-outs of the sheets on top of each other. The cut-outs may contain holes. The exact shape of the uneven terrain is determined by the local organizing chair.
LC-1.6. The sheets are colour coded, that is sheets at different heights have different colours as shown.
LC-1.7. Teams may place small coloured or infra red markers in the goal are as long as they do not interfere with other teams.

LC-2 Game Play

LC-2.1. A single robot is designated the runner. All other robots must be outside of the playing field.
LC-2.2. The only robot allowed to move during a run is the designated runner.
LC-2.3. The runner must be fitted with a small basket or, which is able to hold as many batteries as the team wants the robot to carry.
LC-2.4. The runner will be placed at one end of the uneven terrain in the middle of the terrain.
LC-2.5. At the beginning of the competition, the referee will place one weight in the basket attached to the robot.
LC-2.6. The referee will signal the start of the competition by blowing the whistle.
LC-2.7. After the referee gives the start signal, the robot must cross the uneven terrain to reach the other side.
LC-2.8. A robot is not allowed to leave the uneven terrain along the sides.
LC-2.9. Each robot may have at most one human handler associated with it.
LC-2.10. The human handlers are not allowed to interfere in any way with other robots, the referee, or other human handlers.
LC-2.11. A human handler may only enter the playing field or touch his/her robot with the permission of the referee.

LC-2.12. The end of the competition is signaled by the referee by blowing the whistle a second time. The referee terminates the competition if
• the robot has crossed the uneven terrain with a maximum load,
• the robot was unable to cross the uneven terrain within 2 minutes,
• the robot falls and is unable to get up on its own or is immobilized by a technical defect,
• the robot leaves the uneven terrain along the side lines,
• at least one minute has elapsed since the start of the competition and it is unlikely in the opinion of the referee that the robot will cross the finish line within the remaining time.

**LC-3 Method of Scoring**

LC-3.1. At the end of the run, another robot will be designated the runner.

LC-3.2. Any robot that has not crossed the uneven terrain at least once is automatically awarded 0 points.

LC-3.3. Among the robots that have crossed the uneven terrain at least once
• The robot able to carry the largest load is declared the winner and is awarded 10 points.
• The robot with the second highest load (2nd place) is awarded 8 points.
• The robot with the third highest load is awarded 6 points.
• The robots with the fourth, fifth, sixth, and seventh highest load are awarded 4, 3, 2, and 1 point respectively.
• In case multiple robots are able to carry the same weight, the points accumulated for the n succeeding places are divided equally between the competitors. For example, in case two robots both carry a load of three weights with no other robot able to carry a larger load, then the points for 1st and 2nd place shall be divided between the two robots. In this case, each robot receives \( \frac{10 + 8}{2} = 9 \) points.

**Decisions**

Dec-3.1. The technical committee decided that in addition to the restrictions given above, the uneven terrain includes the following simplifications:
• The uneven terrain will only contain steps which move one level up or down.
• The minimum distance between edges of the sheets is 3cm.

These additional constraints are targeted at simplifying the lift and carry competition over uneven terrain. teams.
APPENDIX IV:

DARPA GRAND CHALLENGE
## DARPA Grand Challenge

### Rules

The development of revolutionary technologies is a key objective of the Grand Challenge. Entrants are invited to communicate directly with DARPA regarding any rule that restricts their ability to demonstrate technical achievement and innovative solutions to intelligent autonomous ground vehicle behavior.

The Chief Judge has the authority to modify the rules at any time. Reasons for rules modifications include, but are not limited to, the accommodation of promising but unexpected technical approaches that would have been prohibited by the rules and the exclusion of approaches that seek to win without demonstrating the desired technical achievement in autonomous vehicle behavior that is the purpose of the Challenge. DARPA will announce any modifications to the rules with an e-mail to all entrants and a statement on the Grand Challenge website under “Rules”.

The Chief Judge may revise the schedule of the Challenge and provide interpretation of the rules at any time and in any manner that is required. The Chief Judge’s decisions regarding the rules are based on a number of factors, such as safety, legal compliance, fairness, Challenge goals, environmental protection, and efficient operations.

At any time prior to the National Qualification Event (NQE), requests for rules clarifications should be sent to: GrandChallenge@darpa.mil. DARPA will hold confidential any questions that are designated as team proprietary.

Decisions of the Chief Judge are final.

### 1 Eligibility

#### 1.1 Team Membership

A team is comprised of the individuals identified to DARPA on the team roster. Only these individuals are team members. Each team must designate a single individual to serve as the team leader. The team leader must be at least 21 years of age and must hold US citizenship on the date of application to the Grand Challenge, and must remain a citizen for the duration of the Grand Challenge. Proof of US Citizenship for the team leader must be provided with the application as described in the application instructions. DARPA representatives will verify these documents at the site visit.

For each team, the team leader will serve as the primary point of contact with DARPA. The team leader must sign the application, must provide a notarized signature on the Certification of Team Funding and Support and Site Visit Liability Statement, and must be present at the site visit, the National Qualification Event (NQE), and the Grand Challenge Event. The team leader will specify the team members and will determine the disposition of the prize should the team be successful. An individual may be the leader of only one team but team members may serve on multiple teams.

Team leadership may be transferred from the team leader to another eligible individual; there may be only one team leader at any time. Transfer of team leadership occurs when DARPA receives a notarized Change of Team Leader form. The form is available from DARPA and must be signed by the existing team leader and the new team leader. The new team leader must also submit proof of citizenship, a notarized Part 2B (Certification of Team Funding and Support), and a notarized Part 5B (Site Visit Liability Statement).
Although the number of individuals listed on the team roster is not expressly limited, DARPA will impose a limit on the number of team members allowed into designated areas at the NQE and the Grand Challenge Event. DARPA will communicate the limit to the team leaders upon notification of selection.

1.2 Non-US Participation and Sponsorship

Individuals holding foreign citizenship are eligible to participate in the Challenge on teams with a team leader who is a US citizen. Foreign corporations and non-governmental organizations may participate as team sponsors. Teams receiving funding or any form of support from foreign governments or foreign governmental organizations are not eligible to participate.

1.3 Team Funding and Support

The cost of developing, fielding, and insuring entered vehicles is the sole responsibility of the individual teams. DARPA will not provide funding for the purpose of Grand Challenge entry or participation.

Each team leader must sign and submit a notarized Certification of Team Funding and Support. This document contains the following certifications:

1. No funding used in the design, development, construction, or operation of the vehicle has been or will be charged to a grant, contract, or other transaction from the government, either directly through such work or indirectly through government-reimbursable research and development, government-funded Independent Research and Development (IR&D), overhead, or general and administrative accounts (known as G&A in the US). This restriction includes funding to pay for labor, travel, equipment leases, or other services that are applied directly to the design, development, construction, or operation of the Challenge vehicle.

2. No portion of the software or hardware used on the vehicle, including the vehicle itself, has been or will be paid for, wholly or in part, using government funding. This exclusion does not apply to government-funded software or hardware that is commercially available or openly available to all teams on June 8, 2004, and through the duration of the Grand Challenge.

3. No patented invention that was developed under government funding is part of the vehicle unless the patented invention is commercially available or is openly available to all Grand Challenge teams on June 8, 2004, and through the duration of the Grand Challenge.

4. Government-owned equipment or facilities have not been used and will not be used in the design, development, or operation of the vehicle unless the equipment or facilities are available to all teams.
This certification does not prohibit:

a) • the use of government-sponsored information such as Global Positioning System (GPS) signals, cartographic products, or government-developed numerical software routines that are openly available

b) • the use of any technologies that are commercially available to all teams

c) • the use of facilities, services, or equipment supplied by DARPA to teams for Grand Challenge

d) • the use of paid vacation time by US government employees and contract employees to support a Challenge team.

2 Challenge Vehicle Requirements

2.1 Autonomous Vehicle Behavior Requirement

Participating vehicles must demonstrate fully autonomous behavior and operation at all times during the NQE and Grand Challenge Event. Vehicles must be unmanned, and no animals are permitted onboard.

2.2 Vehicle Limitations

The entry must be a ground vehicle that is propelled and steered principally by traction with the ground. The type of ground contact devices (such as tires, treads, and legs) is not restricted. The vehicle must not damage the environment or infrastructure at the NQE or along the Grand Challenge route. Vehicle operation must conform to any regulations or restrictions imposed by the applicable land-use authority.

The vehicle must be able to pass through any underpasses encountered on the route. The clear opening of the smallest underpass will measure no less than 10 feet in width and 9 feet in height. Maximum vehicle weight is 20 tons; any team whose vehicle weighs more than 10 tons must provide its own off-road recovery capability. The vehicle must be able to travel on asphalt pavement without damaging the pavement surface.

2.3 Classified Data and Devices

No classified data or devices may be used by a team in preparation for or during the Grand Challenge.

2.4 Tethered Vehicle Systems

Only individual, independent, untethered ground vehicles are eligible to participate in the Grand Challenge.

A system comprising a single ground vehicle and one or more subsystems (such as sensors) that are physically tethered to that ground vehicle is permitted provided that the tethered subsystems are not propelled or maneuvered independently of the ground vehicle (as would, for example, an aircraft or steerable balloon or kite). Tethered subsystems that are specifically permitted include those that are rigid, telescoping, or on an articulating mast; and move only in response to relative wind and vehicle motion, such as balloons or kites. Tethered subsystems that are designed to extend more than 10 feet above the surface must be painted so as to enhance their visibility to helicopter pilots that may need to land near a Challenge vehicle. Entrants are advised that the Federal Aviation Administration, particularly in 14 CFR 101, regulates the operation of moored (tethered) balloons. Entrants are advised that the route may be adjacent to utility and power structures and high-voltage power lines.
2.5 Vehicle Identification Number

Each semifinalist team will be assigned a unique identification number that shall be displayed on its vehicle at least 12 inches in height on its sides, front, back, and top. The number should be either black or white and have a solid background that extends at least 3 inches larger than the number. The colour of the background should contrast with the number such that the number is clearly visible and distinguishable from other signage or symbols on the vehicle. A vehicle that can operate when flipped over shall also display the number on its underside.

Teams are allowed to obtain sponsorships and to display advertising if such advertisements are not considered inappropriate by the Officials. The DARPA Grand Challenge 2005 logo may be displayed on each vehicle.

2.6 Vehicle Safety

DARPA makes no representation as to the safety of any vehicle entered in the Grand Challenge notwithstanding any rule or the acceptance by DARPA of any application document, vehicle specification sheet, video demonstration, or any inspection or demonstration required as a condition of participating in the Grand Challenge.

2.6.1 Radiated Energy Safety Standards

2.6.1.1 Laser Safety Standards

All parties are directed to OSHA 29 CFR 1926.54 and OSHA Technical Manual (TED 1-0.15A), Section III - Chapter 6 (1999, January 20) for relevant laser safety standards. Challenge vehicles must comply with all applicable local, state, and federal laser safety regulations.

2.6.1.2 RF Radiation Standards

All parties are directed to OSHA 29 CFR 1910.97 (Non-ionizing Radiation) and Department of Defense Instruction 6055.11 (1995, February 21) for relevant RF safety standards. All Challenge vehicles must comply with all applicable local, state, and federal RF safety regulations.

2.6.1.3 Acoustic Safety Standards

All parties are directed to OSHA 29 CFR 1910.95 (Occupational Noise Control) and OSHA Technical Manual (TED 1-0.15A), Section III - Chapter 5 (1999, January 20) for relevant acoustic safety standards. All Challenge vehicles must comply with all applicable local, state, and federal acoustic safety regulations.
2.6.2 Wireless Emergency Stop (E-stop) Units

DARPA will supply each semifinalist team one government-owned E-stop system consisting of a controller and a vehicle receiver. It is the sole responsibility of the team to properly install the E-stop system in its vehicle. Detailed specifications for the integration of the E-stop system will be provided on the Grand Challenge website. Limited technical assistance for this installation will be available. DARPA shall not, however, incur any liability from the semifinalist’s use of this technical assistance. Use of this technical assistance is solely at the discretion of the team leader.

Semifinalists have 10 calendar days following receipt of the E-stop to notify DARPA that the unit is damaged or otherwise not in working condition. After that period, the semifinalist assumes responsibility for the E-stop, and DARPA will not be responsible for repairs to the E-stop or replacement of damaged units.

DARPA reserves the right, solely within its discretion and assuming equipment availability, to provide the team with a replacement unit. Each E-stop must be fully functional for the semifinalist to be eligible to participate in the NQE and Grand Challenge Event.

Each team shall return its E-stop to DARPA within 24 hours from the date of any of the following events:

- The vehicle is eliminated from participation in the Grand Challenge
- The vehicle is disqualified from the Grand Challenge
- The vehicle is withdrawn from the Grand Challenge
- Completion of the Grand Challenge

If any of these events occur during the NQE or Grand Challenge, the equipment shall be returned to the proper DARPA official on-site.

The E-stop system has three modes: a RUN mode, a PAUSE mode, and a DISABLE mode. Teams must integrate the E-stop equipment so that the vehicle responds to the E-stop outputs as follows:

- E-stop RUN mode enables the vehicle for autonomous movement.
- E-stop PAUSE mode brings the motion of the vehicle to a prompt stop, with brakes applied to hold the vehicle even if it is on a slope. The vehicle should be ready to resume forward motion when the E-stop re-enters RUN mode.
- E-stop DISABLE mode brings the vehicle to a prompt halt and shuts down all propulsion systems while actively applying and maintaining the brakes.

Specifications regarding size, weight, power, output voltage, current, connectors, and other relevant details will be furnished to semifinalists.

The required integration of the E-stop system enables the E-stop PAUSE mode to be cycled on or off so that the vehicle can be stopped and resumed during the Challenge. The E-stop DISABLE mode should be latched so that its state cannot be changed after initiation except by a manual unlatch switch.

The vehicle and its systems must not interfere with the proper functioning of the E-stop device. A demonstration of the wireless E-stop capability is required as part of the NQE. Teams should anticipate that their vehicle may receive the E-stop PAUSE signal numerous times during the Grand Challenge Event, and that the duration of any individual E-stop PAUSE event may be as long as several hours. Teams should ensure that all electrical connections to the E-stop are ruggedized and tested to provide assured electrical connectivity after exposure to adverse (damp or dusty) environmental conditions and a high vibration environment.
2.6.3 Manual Emergency Stop Unit

Each vehicle must be additionally equipped with an externally-actuated manual emergency stop capability. Activating the manual emergency stop must promptly bring the vehicle to a complete halt in the E-stop DISABLE mode. At least one actuator and its labeling must be easily visible and accessible from anywhere around the vehicle. The manual emergency stop must be easy to identify and activate safely, even if the vehicle is moving at a walking pace. The operation instructions for manual emergency stop actuators must be clearly labeled in English and Spanish. The instructions must not be interfered with by any other labeling or advertising. A demonstration of the manual emergency stop capability is required as part of the NQE.

2.6.4 Warning Devices

Each vehicle shall be equipped with audible and visual alarms that are activated according to the state of the E-stop system. The following is a summary of the required behavior of the alarms.

- E-stop PAUSE mode: No audible alarm. Visual alarm on.
- E-stop DISABLE mode: No audible alarm. No visual alarm.

2.6.4.1 Audible Warning–Vehicle Operating

Each vehicle shall produce an intermittent warning sound when, and only when, the vehicle is in E-stop RUN mode. The vehicle may not commence movement until the warning sound has been in operation for 5 seconds. The warning sound shall produce at least 85 dBA at 10 feet in front of the vehicle, and shall be loud enough to be clearly heard over the normal vehicle engine noise. The audible warning shall not produce sounds that can be confused with those of public-safety vehicles such as law-enforcement, fire, or ambulance.

2.6.4.2 Visual Warning–Vehicle Operating

Each vehicle shall display one or more flashing amber warning lights, the combination of which results in visibility 360 degrees azimuthally around the vehicle. The warning light shall operate when, and only when, the vehicle is in E-stop RUN or E-stop PAUSE mode. The vehicle may not commence movement until the warning light has been in operation for 5 seconds. The warning light(s) shall comply with SAE Class 1 standards for warning lights and shall not produce light(s) than can be confused with those of public safety vehicles such as law enforcement, fire, or ambulance.

2.6.4.3 Visual Warning–Vehicle Brake

Each vehicle shall display two or more steadily illuminated red warning lights on the rear of the vehicle and visible within a 90-degree cone that illuminates when, and only when, the vehicle’s dynamic braking system (not the parking brake) is activated. The purpose of this light is to indicate that the vehicle is braking. The placement of this light should be mounted high and sufficiently distant from the flashing amber warning lights to permit rapid recognition.
2.7 Towing Requirements

Each vehicle must be designed to facilitate removal from the route should the vehicle be disabled. The vehicle must have tow points front and rear, or if the vehicle design precludes towing, the vehicle must have hoist points. Wheeled or tracked vehicles must have a free-wheel mechanism that enables the wheels or tracks to spin freely in order to enable towing. The free-wheel mechanism must be readily accessible and clearly marked.

2.8 Position Determination Signals

Challenge vehicles may be equipped to receive and process electronic position-determination signals (such as GPS) that are openly or commercially available to all teams. Position-determination signals that are neither openly available nor commercially available to all teams are prohibited.

2.9 Wireless Signal Restrictions

All computing, intelligence, and sensor processing must be contained onboard the vehicle while on the NQE course or the Challenge route. Apart from the control and tracking signals from DARPA-provided systems and openly or commercially available navigation signals, the emission or reception of communication signals is prohibited. On-board wireless connections are prohibited. A vehicle may emit and receive signals to sense the environment. Vehicles may record video or other data on-board for review after the conclusion of the event. Any data recorded on the NQE course may not be shared among teams until the conclusion of the NQE. Any data recorded during the Grand Challenge Event may not be shared among teams until all vehicles have finished the route or have been disqualified.

Any wireless system used for vehicle movement or testing must be disconnected prior to the departure signal at the NQE and Grand Challenge Event. The wireless hardware must be easily accessible and capable of being inspected. This includes systems for monitoring, control, or intra-vehicle communication.

2.10 Vehicle Cooperation

Cooperation of any kind among vehicles on the NQE course or the Grand Challenge Event route is prohibited.

2.11 Environmental Impact

Any aspect of vehicle activity or operation that has an unacceptable impact on the environment is prohibited. These activities include destructive vehicle behavior, the use of abnormally hazardous substances or materials, and generally reckless operation. Potentially hazardous equipment or activities must be identified to DARPA for review in the vehicle specification sheet and at the site visit.
2.12 Pre-Challenge Testing

Testing of Challenge vehicles or components is the sole responsibility of each team. The use of public lands for this purpose is at the team’s own risk and must be in accordance with applicable local, state, and Federal guidelines.

3 Qualification Process

3.1 Overview

All steps of the qualification process must be completed by teams that wish to compete in the Grand Challenge Event. A team that has submitted parts 1 and 2 of the application by the deadline and has received acknowledgement from DARPA becomes a Grand Challenge entrant. A team must submit parts 3, 4, and 5 of the application by the deadline in order to remain an entrant. A team selected for the NQE become a semifinalist, and a team selected for the Grand Challenge Event is a finalist.

3.2 Visit

DARPA will review each team’s video demonstration and vehicle specification sheet submitted as part of the application. Applications will be evaluated on the basis of:

- Conformance with the rules
- Possession of a vehicle
- Possession of sensor equipment
- Possession of navigation equipment
- Capability of vehicle to complete the Grand Challenge Event route
- Demonstration of navigation and sensor capabilities necessary for completion of the Grand Challenge.

Instructions for the Video and the vehicle specification sheet are provided on the application form available on the Grand Challenge website.

On April 4, 2005 DARPA will notify all teams of the results of the review process based on the vehicle specification sheet and video demonstration. Selected entrants will be notified of DARPA’s intent to conduct a site visit. Only teams selected for a site visit will continue as Grand Challenge entrants.

3.3 Site Visit Procedure

Site visits will take place at an appropriate testing location in the United States specified by the entrant. The Site Visit Liability Statement (Part 5B of application) must be on file with DARPA before a site visit can be scheduled. Because of scheduling limitations, mandatory schedule dates and times will be set by DARPA. If rescheduling is necessary due to DARPA’s inability to keep the primary scheduled meeting, DARPA will work with the team to find a mutually agreeable new date. Inability to find a mutually agreeable new date may result in removal from further participation. Site visits are scheduled to take place May 2–15, 2005, with backup days May 16–21, 2005. The team leader and vehicle must be present at the site visit. The inspection team will verify the proof of US citizenship of the team leader. Site visit guidelines will be available on the Grand Challenge website.

Based on the results of the site visits, DARPA will select and invite teams to participate in the NQE. Teams that accept this invitation must submit a technical paper describing their vehicle. Teams that are not selected are no longer eligible for participation in DARPA Grand Challenge 2005.
3.4 Papers

A technical paper describing the vehicle of each semifinalist must be received at DARPA by August 15, 2005. A description of the subjects that must be addressed in the technical paper will be available on the Grand Challenge website. DARPA will withhold the technical papers until the conclusion of Grand Challenge 2005, at which time the papers will be made available to the public.

Other than the required technical paper and information already in the public domain, DARPA will not publicly release information regarding a team’s technical approach without permission from the team leader.

DARPA claims no intellectual property (IP) rights from entrants, semifinalists, finalists, or the winner. All trade secrets, copyrights, patent rights, and software rights will remain with each respective team.

3.5 Qualification Event (NQE)

The NQE will be held September 27, 2005 to October 6, 2005 at the California Speedway in Fontana, California. A detailed schedule will be published on the Grand Challenge website and instructions for semifinalists will be distributed.

Semifinalist teams will transport their vehicles to the California Speedway on September 27, 2005 for team check-in and the NQE start.

3.5.1 Inspection

The first phase of the NQE is a static technical and safety inspection of all vehicles to ensure compliance with all rules, to verify the details of vehicle operation described in the vehicle specification sheet, and to ensure safe vehicle operation. Any deviations will be identified to the team leader for immediate action to bring the vehicle into compliance. If a vehicle cannot be brought into compliance it may be disqualified.

3.5.2 Demonstration

The NQE is used to select finalists to compete in the Grand Challenge Event. Teams are given two opportunities on the NQE course. Additional opportunities are at the discretion of DARPA. A team’s final score is derived from its best two attempts.

Vehicle control procedures, autonomous vehicle requirements, and the route definition will be based upon those of the Grand Challenge Event. Details are provided in section 6.

3.5.3 Restrictions on Vehicle Operation

Operation of semifinalist vehicles is limited to DARPA-specified events and operation within DARPA-specified practice areas from the first day of the NQE until the vehicle is returned to the team.

Vehicles that may be selected as Grand Challenge finalists must remain at the NQE until October 6, 2005. Teams that choose not to participate in the Grand Challenge Event may remove their vehicles at any time. Teams may petition DARPA if major repairs are needed that require expertise that is only available offsite.
3.5.4 Security

3.5.4.1 Access Control

Grand Challenge semifinalist teams and DARPA-accredited media representatives will be issued access-control passes that are required for entry into controlled areas at the NQE.

3.5.4.2 Team Security

DARPA will control access to the garage area at the NQE but DARPA assumes no responsibility for the security of team equipment or supplies.

3.5.5 General Safety

A Safety Standard Operating Procedures (SOP) Manual will be distributed to each semifinalist team prior to the NQE. The SOP will provide specific instructions for the administration of activities as well as emergency procedures and instructions for handling other contingencies. Compliance with the SOP is mandatory whenever the team or its vehicle is within DARPA-controlled areas. Failure to comply with the SOP may result in disqualification.

4 Grand Challenge Event

On October 6, 2005, each finalist team will transport its vehicle to the Grand Challenge departure area to make final preparations for the Grand Challenge Event. On October 7, 2005, DARPA will host a meeting with teams to make final preparations for the start. The time and place for this meeting will be provided to the finalists at the NQE. Each team will receive a compact disc (CD) containing the RDDF at least 2 hours prior to the start of the event.

The first vehicle starts the route after first light on October 8, 2005.

DARPA maintains control of all vehicles for safety and operational purposes using the E-stop system. While vehicles are on the route, DARPA officials follow each vehicle in a dedicated control vehicle equipped with an E-stop transmitter.

4.1 Departure Area

When instructed to do so, each team must move its vehicle promptly to the start chute. Challenge vehicles start in sequential order at specified time intervals. Start order is announced at the end of NQE.

Each vehicle must be enabled for autonomous operation within 5 minutes after entering the start chute. Vehicles must be prepared to wait in E-stop PAUSE mode in the start chute for up to 1 hour without manual intervention.

Before each start, an official places the vehicle in E-stop PAUSE mode. At the designated start time an official switches the E-stop from PAUSE to RUN and the vehicle must depart the start area promptly after the mandatory 5 second delay for the audible alarm.
4.2 Vehicle Control

An official may place any vehicle in E-stop PAUSE mode for safety or operational reasons. The official later returns the vehicle to E-stop RUN mode so that it may continue. Time spent in E-stop PAUSE mode does not contribute to a team’s corrected time. If a vehicle does not progress within 10 minutes of resuming E-stop RUN mode, it may be disqualified.

If dangerous or destructive behavior by a vehicle is imminent, an official places the vehicle in E-stop PAUSE mode and the vehicle may be disqualified. If necessary to stop it, the official places the vehicle in E-stop DISABLE mode.

DARPA will take measures to stop a vehicle that does not respond to an E-stop command, even if these measures may result in damage to the vehicle.

4.3 Challenge Route

A team may not intervene in any aspect of vehicle operation or participate in vehicle tracking from the time the vehicle clears the start chute until it is returned to the team. A vehicle is returned to the team after it is disqualified from the event or after it clears the arrival line and is inspected. Refueling of vehicles is not permitted.

Teams may not operate any ground vehicles or position any team members along or near the route during the Grand Challenge Event except at designated viewing areas. Each vehicle must remain within the route boundary from its departure from the start chute to its arrival at the last waypoint.

If a vehicle is in E-stop RUN mode and the vehicle does not progress for longer than 10 minutes, it may be disqualified.

If DARPA officials determine that it is not possible for a vehicle on the route to finish in less than 10 hours while traveling at the maximum speed limit over the remaining segments of the route and allowing the vehicle to continue would hinder Grand Challenge operations, the vehicle may be disqualified.

4.3.1 Route Definition

The route definition data file (RDDF) is the official definition of the route and defines the corridor through which all vehicles are required to travel. The RDDF contains waypoints, lateral boundary offsets (LBO), and maximum speed limits. Vehicles may traverse any area within the route boundary but must detect and avoid obstacles therein. Navigability directly along the track line connecting successive waypoints is not guaranteed; vehicles must determine for themselves the best way to travel from one waypoint to the next while staying within the lateral boundaries.

Vehicles encounter the first waypoint after departing the start chute. The last waypoint is beyond the arrival line. The LBO is specified from any point on a track line and applies to the route segment defined by the associated waypoint to the next sequential waypoint. Boundaries may be marked with concrete barriers, plastic snow fencing, or by other similar means along limited portions of the route. Any vehicle that leaves the route may be disqualified.

A maximum speed limit is specified for each segment of the route. Any vehicle that exceeds the speed limit may be disqualified. A specified speed limit does not imply that it is a safe or achievable speed. Speed limits are specified in the RDDF and apply to the route segment defined by the associated waypoint to the next sequential waypoint. Between the start chutes and the first waypoint, vehicles may not exceed the speed limit of the first route segment. In the area where two route segments overlap, the least restrictive (i.e., higher) speed limit applies.
4.3.2 **RDDF Format**

The RDDF is a comma-delimited text file distributed on a PC-formatted CD and contains the following data fields: the number of each waypoint, the latitude of each waypoint, the longitude of each waypoint, the lateral boundary offset of each segment (feet), and the maximum speed of each segment (mph).

Each waypoint marks the start of a segment. The first waypoint is number 1 and subsequent waypoints follow in sequence. Latitude and longitude are specified in decimal degrees with seven decimal places. The applicable datum is WGS 84. The accuracy of the waypoint locations is +/- 15 cm. The 7th decimal figure does not connote an additional degree of accuracy. Segments with unspecified maximum speed are indicated by 999.

4.4 **Obstacles**

The vehicle must avoid collisions with any obstacle, moving or static, on the route. DARPA will place obstacles along the route to test obstacle avoidance capabilities. Vehicles that collide with any other vehicle or obstacle along the route may be disqualified. Incidental or non-damaging contact with obstacles may not result in disqualification.

4.5 **Intentional Interference and Damage**

Intentional interference with other vehicles is prohibited. Intentional interference is any activity that, in the opinion of the Chief Judge, is intended to degrade another vehicle’s ability to compete.

Any team responsible for the intentional damage of property that does not belong to that team may be disqualified. Intentional damage includes damage that occurs as a result of failure to prevent damage that could have been foreseen and includes damage that adversely and materially affects the performance of another team. The Chief Judge will have the final say in all matters involving damage.

4.6 **Improper Vehicle Contact**

A team may not make or cause physical contact with its vehicle after it has departed the start chute and before it is returned to the team. Contact with the vehicle may be permitted if the vehicle has been disqualified or overnight procedures have been enacted as determined by the officials. Physical contact includes indirect contact with tools and human-initiated contact using remotely controlled or electronic equipment.

4.7 **Jettisoning Material on the Route**

Except for normal by products of power generation, the intentional jettison of any material from a vehicle is prohibited and may result in disqualification. If a portion of a vehicle unintentionally falls from the vehicle while on the route, DARPA will notify that team, and the team is responsible to recover such debris once all qualified vehicles have cleared the affected area.

A smokescreen or any other obscurant intentionally discharged from a vehicle is specifically prohibited.
4.8 Passing

DARPA officials determine when and where vehicles pass. No vehicle may intentionally operate to hinder another vehicle that is trying to pass it. The overtaking vehicle has the burden of responsibility for collision avoidance and must remain within the route boundary.

If the width of a route segment is sufficient for passing, DARPA officials may place a slow moving (impeding) vehicle in E-stop PAUSE mode to allow a faster (overtaking) vehicle to pass with its associated control vehicle. The overtaking vehicle must sense all stopped vehicles and navigate around them.

If the width of a route segment is insufficient for passing, and the impeding vehicle is moving, a DARPA official places the overtaking vehicle in E-stop PAUSE mode until there is sufficient room to pass.

If the width of a route segment is insufficient for passing and a vehicle is immobile and blocking the route such that no other vehicles can pass, DARPA officials place any approaching vehicles in E-stop PAUSE mode until the route is clear.

4.9 Overnight Operations

If necessary, the event will continue beyond October 8, 2005. DARPA will distribute procedures for overnight operations that address corrected time computation, vehicle shut down and restart, security, and safety.

4.10 Arrival Area

After a vehicle crosses the arrival line it is impounded for an inspection. Teams may not interact with their vehicle until it is released by a DARPA official.

4.11 Corrected Time

*Elapsed time* for each vehicle begins at the departure signal and ends when the entire vehicle clears the arrival line from the direction of the previous waypoint. Elapsed time will not be maintained for disqualified vehicles.

Corrected time is computed by subtracting corrections for time spent in E-stop PAUSE mode from the elapsed time.

A vehicle must have a corrected time of 10 hours or less to be eligible for the prize.

4.12 Disqualification

A disqualified vehicle may not continue on the route. DARPA will coordinate with the team to recover the vehicle from the route. Teams will enter the route area only when so directed by DARPA officials.
APPENDIX V:

PERMIS’04 WHITE PAPER
PERMIS’04 White Paper

PERMIS’04 Focuses on Measuring the Performance of Intelligent Systems.

A natural link and even lead emerges: not only to measure the Performance but also Measure the intelligence of Intelligent Systems and put both of them in correspondence. Measuring intelligence might become easy after a mathematical theory of intelligence is developed. However, for obvious reasons we need metrics before we can (and in order to) develop intelligent systems. One approach could be to list the properties of intelligence as far as we understand them and then to come up with qualitative measures for each. The overall model of intelligence might be obtained as a result of this problem-solving maneuver. Because there are a lot of these properties we need to come up with a hierarchy of them. Possibly this hierarchy (which, like intelligence itself, is not necessarily a strict hierarchy, but a heterarchy to some extent) could be a subject for the discussion.

A good starting point could be building a simplified provisional model of an "elementary intelligent agent (EIA)" (This model in its initial stage might include an internal model, sensors, perception, (re)cognition, behavior generator, actuators+ a hierarchy/heterarchy of these agents.) One can imagine an EIA even as a much simpler reduced set: perception and cognition are similar - just occurring at different hierarchical levels. A fundamental question: how are we going to represent EIA? Present possibilities include:

- pure mathematical model
- a descriptive model
- a model in terms of control theory
- linguistic model
- a model in terms of Information Technology

A next step could be characterization of the modules of EIA in terms of their relationships to the features of intelligence that are vaguely represented scientifically: concepts, emotions, thinking, learning, memories,... Can we come up with numerical characterization of each of these functions or at least a crisp descriptive characterization? How do they add to intelligence first, in terms of functions, second, in terms of efficiency, and third in satisfying our intuition about what is intelligence? Since various types of intelligence are being evaluated for humans, there similarly should be an array of types of intelligences for artificial systems.

The Goals of PERMIS’04

The goals of PerMIS’04 are to extend the body of knowledge pertaining to how to define, characterize, measure the intelligence of systems. There will be tracks emphasizing the theoretical developments in the area of performance measures and definitions relevant to intelligence and autonomy. These are complemented by tracks that focus on experimental results applied to domain-specific systems.

Examples of key themes for the theoretical tracks are listed here.
1. Evaluating Actionable Knowledge
This conference emphasizes metrics for evaluating the advanced methods of constructing the actionable knowledge from the data and information. The next problem would be: Actuation based on the Results of Intelligent Processing. Sensor development and data extraction with subsequent information acquisition (often called “getting the intelligence”) is only one part of enabling the Decision Support Processes accompanying the process of Actuation. Discovery and disambiguation of the system of interacting agents coordinated within the set of goals is the essence of knowledge construction required for the successful decision making and performing the expected functions of the system. It is critical to be able to define the quality of the knowledge – its uncertainty, resolution, precision, accuracy, etc. When constructing a model of the world, the system has to be able to weigh the relative merits of various inputs (clues) it receives as well as knowledge it already has.

2. Multiagent Intelligent Systems
Formulation or required behavior of multiple agents, and design of the action networks consistent with the realistic systems is the purpose in all domains of business and engineering application: from management support and drug discovery to robotics and military operations. Between data and actions there are processes of information extraction and knowledge construction required for behavior generation. Networks of knowledge, interwoven with intentional systems of goal oriented agents, ought to account for the available options in decision-making based upon knowledge, phenomenology and computational processes underlying the data and action. The knowledge repositories should not only contain the results of acquired data and information sets, they also must contain
- physical models of sensors or wave propagation and scattering
- chemical models of molecular interactions
- statistical models of object properties
- dynamical models of motion
- linguistic text models
- semiotic models of meaning
- cultural models of human behavior in the industry, or a society of interest.

3. Computations Pertained to Intellect
The challenge is to develop integrated intelligent computational systems capable of combining available data from the knowledge repositories, multiple disciplines, and the afforded sensors. Also corresponding to the stored knowledge, the models might be detailed or approximate, reflecting precisely known physical laws or uncertain intuitions about undiscovered phenomena or human nature. The integrated functioning of an intelligent system is not a one-time deal but a continuous loop of operations in which sensors and data collection are directed based on the current-moment results, models and actions are continuously refined. No wonder that the relevant systems have substantial affinity with the known architectures of Brain and Intelligence of living creatures. This resemblance will be explored at this conference.

4. Similar Algorithmic Structure
The conference focuses primarily upon areas important in the industrial environment, scientific research, business, and defense operations: integrated closed-loop operation of data acquisition→information extraction→knowledge construction→action. Components of this loop demonstrate similar algorithmic structure. All stages of this string employ tools and techniques of
• multiresolutional data, information, and knowledge analysis,
• entities discovery and recognition,
• exploratory large data arrays processing,
• signals and images analysis and interpretation,
• objects, scenes, and situation identification,
• design of efficient sensor systems,
• multimodal data fusion
• sensory and textual data fusion
• analysis of text messages
• natural language text interpretation
• integrated closed-loop operation

5. Domains of Application
These techniques are expected to be applied in the following domains
• intelligent transportation systems
• emergency response robots
• demining robots
• defense robotics
• command and control
• hazardous environment robots and control systems
• space robotics
• assistive devices
• automatic target recognition
• design of communication systems in the network-centric environment
• generation of the common operational picture in battlefield
• analysis of situations in business and industry
• gene profiling and development
• drug discovery
• manufacturing and process planning systems

6. Focal themes
The following fundamental ideas will be the focal themes at this conference:
• Models and Similarity Measures for Image Recognition
• Models and Similarity Measures for Text Interpretation
• Models and Similarity Measures for Situation Analysis
• Algorithms and Processes of Generalization
• Architectures of Intellect-like Computational Processes
• Search for Exploring Bodies of Data, Information, and Knowledge
• Hypotheses Generation and Disambiguation

7. Examples
The following are a few examples of major programs that rely on intelligent systems technologies and approaches. They are presented as a stimulus for workshop topics and discussion.

U. S. Army Future Combat Systems and Future Force Warrior
The Army is reinventing itself, aiming at lighter, more portable and lethal configurations. Robotic agents are a significant component within this new vision, dubbed Future
Combat Systems (FCS). Complementing this is the Future Force Warrior (FFW), described as “an integrated system of systems approach is being employed to support the Army transformation to a soldier-centric force. FFW notional concepts seek to create a lightweight, overwhelmingly lethal, fully integrated individual combat system, including weapon, head-to-toe individual protection, netted communications, soldier worn power sources, and enhanced human performance.”

Physical agents, such as robotic weapons and sensor platforms, will be ubiquitous in the future battlefield, significantly lowering the risks to our warfighters. These physical agents are to complement future manned systems and therefore they must be able to collaborate not only amongst themselves but also with their manned partners. Their roles will range from scout missions performing reconnaissance, surveillance, and target acquisition to urban rescue missions. Information from both local and remote sensor systems will be fused by intelligent agents and provided to the dismounted warfighter in a highly intuitive form to enhance rapid assimilation and action. An integrated, multi-agent (software and physical) intelligent combat system will facilitate increased mobility, survivability, sensor coverage, information flow, and situation awareness.

Logistics and Distributed, Modular Component-Based Systems

Of relevance to the military and to industry in general is the area of logistical planning. A key aspect of logistics planning and execution, involves reasoning about things, their properties, their relationships, and the activities in which these things participate. The things under consideration may generally be considered assets of one sort or another, and include equipment (e.g. radios, trucks, components, etc.), materiel (e.g. fuel, machine parts, food, etc.), facilities (e.g. road networks, airports, depots, etc.), varied types of organizations (i.e., civil, military and commercial organizations), and even individual people. In order to construct a computer system which participates in logistics planning and execution, it is necessary to have a mechanism to represent all the properties of assets required for logistics reasoning. Several aspects of these assets and their use in logistics systems make the problem hard: (1) The set of asset types and asset properties is very large. These asset properties must describe the forms and functions of each asset required for logistics reasoning. (2) The set of assets and properties evolves continuously over time as new models and types of equipment and materiel are continuously introduced, and older ones are retired. (3) Reasoning must be done over a range of granularities since varied amounts of detail are required at different locations and echelons in the logistics planning and execution processes. (4) Different portions of a logistic planning and execution system require different granularities specialized knowledge. [http://www.cougaar.org/] is one program that is focusing on development of an open source architecture for the construction of large-scale distributed agent-based applications.

Distributed and Multi-resolutional Knowledge Representation

A corresponding requirement for development of open, modular, intelligent logistical planning systems (and similar large scale systems) is the need for careful design of the knowledge representation. Assets must be carefully described; their properties updated as needed and made accessible to the concerned planning entities. As noted by the Cougaar program:
“Given these principles which factor knowledge of logistics properties and behavior, we have been able to allow the detailed asset representation of any particular asset to differ depending on the perspective or interests or needs of the using agent. Thus, the instantiated aspects (properties or attributes) of an asset change as references to that same asset move throughout the society.”

The following are a few examples of agent roles and the corresponding property groups of interest for a truck asset. A transoceanic shipping company, which ships assets such as trucks, needs to know the physical dimensions of the trucks, and what is loaded on the trucks (if anything) while they are to be moved. A truck repair shop needs to know what is wrong with the trucks they have been contracted to repair, and what resources (parts, repair equipment, mechanical skills, time) will be needed to fix each truck. A ground shipping company which owns and operates fleets of trucks needs to know such things as what cargo is on board its trucks, where its trucks are going, how much fuel is in its trucks’ fuel tanks, and when its trucks will be out of service for scheduled maintenance or repair.”

The different, parallel views of the same asset must be crafted to provide the right information for reasoning by the different interested systems. How to design, evaluate, and ensure the validity of the knowledge representation is an extremely significant aspect within the science of intelligent systems.

8. Workshop Format

Researchers from academia, commerce, and defense research centers will exchange ideas and program managers will inform on the directions of research and development. The workshop will consist of plenary lectures, panel discussions, and concurrent focused session where papers are presented and discussion is encouraged. Two social events are planned: a reception on Tuesday night and a banquet with special speaker on Wednesday night.
APPENDIX VI:

MOTION PLANNING BENCHMARKS
APPENDIX VI: MOTION PLANNING BENCHMARKS

This initiative is a repository for motion planning benchmark problems maintained by the University of Parma. It aims at including benchmarks designed by different research groups and documentation describing the file formats currently used by the available planning tools to define their problems. Also, links and useful information regarding motion planning research projects are available. The repository is intended to serve as the basis for further discussion on the requirements and the design of benchmarks in motion planning.

The data sets follow the so-called Motion Planning Markup Language (MPML), an XML-based input format that has been proposed by the University of Parma, in such a way that the structure, content and semantics of valid XML files describing the robots and the workspace are defined by a set of XML-Schema documents. A Java3D tool to visualize motion planning scenarios that uses the XML format is also part of the system.

The repository includes data sets about:
- robots
- workspaces
- benchmark problems

The workspace may contain an arbitrary number of obstacles which are described by their position in the workspace and their shape. The shape is described as a composition of one or more basic objects (see figure). Some representative examples are shown in the figure.

Different kinds of robots are supported, such as:
- mobile robot on the plane
- free-flying robot
- manipulator
- sequence of manipulators
- union of mobile robot and one or more manipulator sequences

They are described similarly in a hierarchical way (see figure). For instance, manipulators and kinematic chains are described as sequences of links. Some representative examples are shown in the figures.
Examples of workspaces

Examples of robots

Name: Manus Robot
Description: a six dof manipulator robot for wheelchairs

Name: Nomad 200
Description: the well known mobile robot

Name: Manus and Nomad 200
Description: Manus manipulator mounted on top of a Nomad 200 mobile robot

Structure of a robot
Examples of benchmark problems follow Hwang's taxonomy [Hwang 92], and comply with his requirements, that is, they must be realistic and non-pathological problems with at least one example satisfying the following criteria:
1. Number of obstacle $\geq 5$, some concave obstacles,
2. Solution paths nontrivial, must utilize all available dofs,
3. Narrow space at some point
4. A trap in the space requiring backtracking

Some representative examples of the available problems are shown in the figures.

Examples of benchmark problems

The environment also incorporates a 3D tool to visualize results; that is, a viewer for MPML which is decoupled of the application.

Structure of the MPML 3D Viewer
Some snapshots of the viewer are included:

More information can be found at the official website: http://mpb.ce.unipr.it
APPENDIX VII:

MODELS FOR MOTION PLANNING
The so-called Movie Models for Motion Planning is intended as a repository of motion planning benchmarks maintained by Utrecht University. It originated from work in the context of the Movie project in which motion planning techniques were developed and tested by using tools and 3D models of robots and scenes. Then they were opened to the community at large in model collections providing easy interface and search functionality, as well as useful information about the models, in such a way that new models could be easily added by researchers around the world.

Specification formats for the robots and scenes are a critical issue for the widespread acceptance and use of the models. The current file formats are:

- XML file format of Callisto.
- Virtual Reality Modeling Language (VRML and WRL).
- Extensible 3D (X3D).

Callisto is a library to visualize 3D virtual environments that is specifically suited to support research on Probabilistic Road Maps (PRM's). It consists of a visualization environment based on Maverik and it uses Solid for collision checking. The input format used to describe scenes in Callisto is based on XML. It has the following key features:

- Easy creation of an environment by using primitives: box, sphere, cylinder, cone, tetrahedron, poly lines, polygons and triangles.
- Supports polygons and poly lines.
- Objects can be translated, rotated and scaled using Euler angles and quaternions.
- Objects can be grouped so that they can be manipulated together.
- Groups can be grouped as well.
- Collision checks can be performed for a point, a line and between groups.
- Object/group clearance.
- Penetration depth.
- Closest pairs.
- Textures.
- Scenes can be loaded from disk using an easy to use XML format
- Support for loading VRML scenes.
- Animation support (by interpolation between key states).
- Camera animation

A DTD (Document Type Definition) defines the structure and the tags of the XML files that describe the workspace, the robots and the problem (a DTD in general defines the legal building blocks of an XML document, i.e. it defines the document structure with a list of legal elements). Further information about XML in Callisto can be found as an appendix at the end of this document.

Maverick is a publicly available virtual reality (VR) system. It enables rapid production of complex virtual environments as well as providing many functions that are valuable to anyone developing applications with 3D graphics or using 3D peripherals since it deals
primarily with graphical and spatial management. It is designed to support high-performance rendering, including large-model processing, customised representations of environments for different applications, and customisable techniques for interaction and navigation.

The **Maverick** micro-kernel implements a set of core services, and a framework that applications can use to build complete virtual environments and virtual reality interfaces. The supporting modules contain default methods for optimised display management including culling, spatial management, interaction and navigation, and control of VR input and output devices. It allows these default methods to be customised to operate directly on application data, so that optimal representations and algorithms can be employed. It offers a separate representation for the application data in contrast with conventional VR systems that need to import data into their own format, maverick avoids this by making use of the application's own internal data structures so that it can far more readily adapt (dynamically) to a wide range of application demands.

**Solid** is a software library for collision detection of geometric objects in 3D space. it contains operations for performing intersection tests and proximity queries on a wide variety of shape types, including: deformable triangle meshes, boxes, ellipsoids, and convex polyhedra. Since it exploits temporal coherence in a number of ways, it is especially useful for detecting collisions between objects that move smoothly over time. The motions of objects are controlled by the client application, and are not determined or affected by Solid.

**Virtual Reality Modelling Language** (VRML) is a hierarchical scene description language that defines the geometry and behaviour of a 3D scene or "world" and the way in which it is navigated by the user. VRML world files have the file extension .wrl (or .wrz for gzip compressed files) and require either a stand-alone application or web browser plug-in to be viewed.VRML is the predecessor of X3D, however the VRML97 specification and many VRML tools are still very useful and will remain so while developers update their products to support X3D.

**Extensible 3D** (X3D) is a scalable and open software standard for defining and communicating real-time, interactive 3D content for visual effects and behavioral modeling. It can be used across hardware devices and in a broad range of applications including CAD, visual simulation, medical visualization, GIS, entertainment, educational, and multimedia presentations. X3D provides both the XML-encoding and the Scene Authoring Interface (SAI) to enable both web and non-web applications to incorporate real-time 3D data, presentations and controls into non-3D content. X3D is the successor to the Virtual Reality Modeling Language (VRML). It improves upon VRML with new features, advanced APIs, additional data encoding formats, stricter conformance, and a componentized architecture using profiles that allows for a modular approach to supporting the standard and permits backward compatibility with legacy VRML data.

X3D is open source, i.e. so no licensing issues; it has been officially incorporated within the MPEG-4 multimedia standard. XML support makes it easy to expose 3D data to Web Services and distributed applications. It is compatible with the next generation of graphics files - e.g. Scalable Vector Graphics and 3D objects can be manipulated in C or C++, as well as Java.
The repository currently contains data sets describing 192 robots and objects, together with 75 scenes. They fall into several categories, namely:

- Articulated objects. Robots whose joints can move freely within certain joint limits: Fixed-base, robotic arms.
- Free-flying objects. Rigid robots that can translate and rotate freely.
- Non-holonomic objects. Robots that have additional constraints on the directions of motion: car-like, autonomous agents.
- Indoor scenes. Located, suited for, or taking place within a building.
- Open space scenes. Located, suited for, or taking place in open air, like hills, crates, etc.
- Puzzles. Scenes that represent a problem to be solved: Narrow passages, mazes.

In the following figures some examples of the different categories are shown:

Examples of free-flying objects

Examples of robots arms
Examples of car-like objects

Examples of in-door scenes
Examples of benchmark puzzle problems

The famous alpha puzzle
Example courtesy of Pekka Isto

Other benchmarks have been proposed such as the example shown in the right figure courtesy of Pekka Isto. Unfortunately, they use a proprietary format, and hence it is not available.

More information can be found at the official websites:
- About the Movie models:
  http://www.give-lab.cs.uu.nl/movie/moviemodels
- About Callisto:
  http://www.cs.uu.nl/~dennis/callisto/callisto.html
- About Maverick:
  http://aig.cs.man.ac.uk/maverik/
- About SOLID:
  http://www.dtecta.com/
- About Virtual Reality Modeling Language and Extensible 3D:
  http://www.web3d.org
APPENDIX VIII:

VISUAL SERVOING BENCHMARKS WITH THE ATC
APPENDIX VIII: REMOTE VISUAL SERVOING BENCHMARKS WITH THE AUTOMATIC CONTROL TELELAB

The University of Siena in Italy has developed an Automatic Control Telelab (ACT) with the idea of offering support to real-time configuration and observation of experiments, as well as playback access to acquired data, from remote computer linked to a collection site through the Internet. The underlying philosophy is that distributed data acquisition makes possible the collection of data from remote environments and can also improve collaboration among geographically dispersed scientific communities by distributing scientific results more quickly and less expensively than most other methods.

The on-going initiative offers the resources already available at the Automatic Control Telelab -both software and hardware- to the research community in order to extend them for implementing visual servoing benchmarks to be used in a remote way. Since it is the same hardware set-up that is used, comparison across different methods is easier.

At present, five diverse experiments for remote control are available, namely:
- DC Motor for position and speed control. (Linear - Stable)
- Water Tank for level and flow control. (Nonlinear - Stable)
- Magnetic Levitation System. (Nonlinear - Unstable)
- 2 DOF Helicopter. (MIMO - Nonlinear - Unstable)
- Lego Mobile Robot. (Mobile Robotics)
The ACT controller is based on Matlab-Simulink, making it very easy to use. In addition, the user can define his own controller model with only Matlab 5.0 (or higher) and Simulink. For the user convenience, a template model can be downloaded from the server. This Simulink model contains two subsystems, which refer to the controller and the reference as well. On-line tuning parameters can be easily inserted that can be modified while the experiment is running.

An easy-to-use experiment interface is provided having 5 parts:
1. Command Panel
2. Controller Parameters Panel
3. Reference Panel
4. Experiment Dynamics Window
5. Live Video Window

Currently, two competitions based on this framework are proposed as a way of benchmarking:
- Control of a magnetic levitation system (see figures)
- Control of a 2 DOF Helicopter

More information can be found at the official website: http://act.dii.unisi.it.
APPENDIX IX:

VISUAL SERVOING SIMULATION EXPERIMENTS
APPENDIX IX: VISUAL SERVOING SIMULATOR

A visual servoing simulation environment has been developed at Universitat Jaume I. It is called JaViSS and written in Java with graphical rendering making extensive use of Java 3D API. The calculus engine is implemented with the Colt libraries. Though JaViSS currently runs on a single computer, it has been heavily designed in a distributed manner, using the agent-based JADE platform. Finally, 3D models have been created with AC3D, and loaded with the Java 3D loader J3D-VRML97. Manipulator kinematics code is based on the Robotics Toolbox for Matlab by Peter Corke. It is intended as a tool to simplify the testing and comparison of different visual servoing approaches.

System requirements
- Windows and Mac OS X: Java JDK 1.4.2
- Linux: Java JDK 1.5 Release 2
- Java 3D (not installed by default in JDK)
- Java 3D update for Mac OS X

Running JaViSS
JaViSS is executed by means of Java Web Start, just by clicking on some links. The user interface is a window such as this:
The target object, camera view, and observer view, can be moved by the button panel at the bottom of the window. Features can be selected from a predefined set for each target object and by pressing the "Features..." button a dialog appears, where each feature is individually selected:

For the default target object, the available feature set consists of the center point, and two segments (defined by the upper and lower points), each defined either by two points or by the center, length and orientation. This is still in a preliminary stage and lacks some error checking (out of view points, joint limits, etc).

**Data log files**

The program stores some visual servoing data in text files on the current directory. Some gnuplot scripts to plot those data (for the default file names) can be obtained:

- Feature error
- Kinematic screw
- Image point trajectories (only when using image point features, obviously)
- Camera 3D trajectory

More information can be found at the official website:
http://www.robot.uji.es/research/projects/javiss
APPENDIX X:

VISUALLY-GUIDED 3-FINGER GRASPING EXPERIMENTS
APPENDIX X: VISUALLY-GUIDED
3-FINGER GRASPING EXPERIMENTS

This initiative has been developed at Universitat Jaume I and offers a description of a set of experiments on visually-guided grasping of planar objects with a Barret hand. They are made available to the community as a set of standard experiments on which benchmarks and associated performance metrics can be proposed. A first experimental protocol for such a benchmark is also proposed.

The goal of the experiments is to test different implemented procedures for visually-guided grasping. That is, by only using visual information about planar objects a set of 3-finger feasible grasps are determined and executed to lift the object. The experimental setup for the experiments consists of:

- A Bisight head consist of four mechanical d.o.f. (pan/tilt head , and two independent verge).
- Two gray-scale cameras enhanced wit two mechanized lenses that provided three additional optical d.o.f. (zoom, iris, focus) per camera.
- A seven d.o.f. Whole Arm Body (WAM) robot manipulator from Barrett Technologies.
- A three-fingered Barrett Hand. Each hand has 4 d.o.f and has a tactile load cell (ATI) integrated in each fingertip. These cells provide a 6-dimensional force/torque information.

It is worth noting that these experiments involve the solution of several problems. The main one is the constraining of a general three-finger grasp to the particular kinematics of the 4 d.o.f. Barrett Hand. Another relevant problem is the use of stereo images to locate an object placed in front of the robot, and to obtain the contour of such object.

The following steps must be followed for the execution of the experiments
1. An object is placed on a table in front of the robot.
2. The stereo camera system takes a pair of images from the object, and uses them to locate the object and to obtain the contour of each object.
3. The grasp synthesis algorithms compute a set of feasible grasps that meet the force-closure condition using the contour information obtained.
4. A human operator selects the desired computed grasp.
5. The robot tries to lift the object using the selected grasp.

The description of each experiment includes the pair of stereo images taken from the object (the detected contour is highlighted with a blue line for each image); an image of the contour with the grasp configuration executed; a video recording showing the execution of this grasp; and finally, a comment on the performance of the grasp attempt. A total of 67 experiments are described based on 14 different planar objects. Some of them are shown in the figure.

**Experimental Benchmarking of Grasp Reliability**

A second step in this initiative puts forward a protocol based on the previous experiments, that is, it assumes as starting point that a grasp has been chosen for an object that appears in the workspace of the robot. The robot tries to grasp and lift the object. If it succeeds in this operation it positions its arm in such a way that the hand and the object are hanging vertically.

If the object is lifted successfully, the robot applies several sequences of shaking movements to the object. These movements consist of rapid accelerations and decelerations of the hand that holds the object. A single sequence is composed of four different kinds of movements. Two describe balancing displacements of the object along two perpendicular horizontal axes of the object (X and Y). The third describes rotations around the vertical axis (Z). And, finally, the last applies a combination of the three previous movements.

After each sequence, the force sensors on the fingertips are used to check whether the object is still in contact with the fingers (contact checking), that is, the object is still held by the hand. If any of the contacts is lost, the object is assumed to have been dropped during the shaking and the test is finished, otherwise the test is continued. It is not unusual that only one finger has lost contact with the object but it is still retained by the other two fingers. However this is in most of the cases a precarious situation, so it is considered by default as if the object had been dropped.

The sequences are applied with increasing magnitudes of acceleration. If the object does not fall after these three tests it is returned to its original location and laid down there. The later in the process an object drops, the more stable the grip can be considered. The result of the stability test, i.e. the final state, will be the number of the state in which a contact checking has failed. The final state is a simple indication of the firmness of the initial grasp; these finishing states are grouped in five different categories. Two video segments showing examples of the procedure are included.

More information can be found at the official website:
http://www.robot.uji.es/people/morales/experiments
Experiments for grasping reliability
APPENDIX XI:

GENERAL GUIDELINES FOR ROBOTICS PAPERS INVOLVING EXPERIMENTS
General Guidelines for Robotics Papers Involving Experiments

John Hallam

Robotics papers come in many varieties, in a spectrum from purely theoretical to purely practical. For example, a paper may present a new theoretical advance; it may describe a new system concept; it may advance an argument based on discussion; it may present comparisons between a set of known techniques; it may do more than one of the foregoing. Most forms of robotic system performance measurement, evaluation, comparison, characterisation etc. involve practical experimentation, which must be carried out responsibly and reported well.

This document is not (except indirectly) an attempt to teach good experimental design and practice; there are excellent textbooks for that. Neither does it address ‘political’ issues, such as whether experimental work is necessary in a good paper in some subfield of our discipline.

Rather, it presents a structured set of questions intended to help reviewers recognise, and authors write, high quality reporting of replicable experimental work and in the process improve the standard of robotics papers internationally.

1. Is it an experimental paper?

An experimental paper is one for which results, discussion and/or conclusions depend crucially on experimental work. It uses experimental methods to answer a significant engineering or scientific question about a robotic (or robotics-related) system. To test whether a paper is experimental, consider whether the paper would be acceptable without the experimental work: if the answer is no, the paper is experimental in the context of this discussion.

Note that experiments may be conducted using simulation as a tool.

2. Are the system assumptions/hypotheses clear?

The assumptions or hypotheses necessary to the function of the system must be clearly stated. System limits must be identified.

3. Are the evaluation criteria spelled out explicitly?

An experimental paper should address an interesting engineering (or scientific) question. Such questions will generally concern the relationship between system or environment parameters and system performance metrics. The performance metrics being studied must be clearly and explicitly motivated, and the parameters or factors on which they depend must be identified. The criteria for “success” should be stated and, where necessary, justified.

4. What is being measured and how?

The performance criteria being studied must be measurable; the paper must identify measurements corresponding to each criterion and motivate the choice of measurements employed. The data types of measurements should be clearly given or obvious — categorial (e.g. yes/no), ordinal (e.g. rankings), or numerical.
5. **Do the methods and measurements match the criteria?**

   Measurement methods and choices must be clearly and explicitly described and, where appropriate, explained and justified. The paper must demonstrate (unless it is self-evident) that the chosen measurements actually measure the desired criteria and that the chosen measurement procedures generate correct data (for example, that implementations are plausibly correct).

6. **Is there enough information to reproduce the work?**

   It is fundamental to scientific experimentation that someone else can in principle repeat the work. The paper must contain a complete description of all methods and parameter settings, or point clearly to an accessible copy of that information (which should be supplied to the paper’s reviewers). Known standard methods need not be described, but any variations in their application must be noted. If benchmark procedures are used, they must be referenced, and any variations from the standard benchmark must be documented and justified.

7. **Do the results obtained give a fair and realistic picture of the system being studied?**

   Care must be taken to ensure that experiments are properly executed: factors affecting measured performance that are not the subject of study must be identified and controlled for. In particular, uncontrolled variations in the system or the environment must be identified and dealt with by elimination, grouping techniques or appropriate statistical methods. The task tackled by the system must neither be too easy or too hard for the system being studied (demonstrated for example by performance comparison with standard methods). Outlying measurement data may not be eliminated from analysis without justification and discussion.

8. **Are the drawn conclusions precise and valid?**

   The experimental conclusions must be consistent with the experimental question(s) the paper poses, the criteria employed and the results obtained. System limits must be presented or discussed as well as conditions of successful operation. Conclusions should be stated precisely. Those drawn from statistical analysis must be consistent with the statistical information presented with the results.
APPENDIX XII:

PROCEEDINGS IROS'07 WORKSHOP ON BENCHMARKING FOR INTELLIGENT ROBOTS
It is a well-known fact that current robotics research makes it difficult not only to compare results of different approaches, but also to assess the quality of individual work. Some steps have been taken to address this problem by studying the ways in which research results in robotics can be assessed and compared. In this context the European Robotics Research Network EURON has as one of its major goals the definition and promotion of benchmarks for robotics research. Similarly, the “Performance Metrics for Intelligent Systems (PerMIS)” Workshop series has been dealing with similar issues in the context of intelligent systems. The main purpose of this workshop is to contribute to the progress of performance evaluation and benchmarking, focusing in intelligent robots and systems, by providing a forum for participants to exchange their ongoing work and ideas in this regard.

The emphasis of the workshop will be on cognitive solutions to practical problems. These cognitive approaches should enable an “intelligent” system to behave appropriately in real-world scenarios in various application domains. We also propose to discuss the distinction between autonomy and intelligence (if any) and how one influences the other. We welcome any topic relevant to benchmarking and performance evaluation in the context of cognitive solutions to practical problems, such as:

- Knowledge representation, perception (sensing), and learning
- Uncertainty management in robot navigation, path-planning and control
- Cognitive manipulation
- Benchmarking autonomy and robustness to changes in the environment/task
- Capability-led understanding of cognitive robots
- Shared ontologies to discuss robotic cognitive systems in terms of their performance capabilities
- Relationships between different cognitive robotics capabilities
- Requirements, theories, architectures, models and methods that can be applied across multiple engineering and application domains
- Detailing and understanding better the requirements for robots in terms of performance, the approaches to meeting these requirements, the trade-offs in terms of performance
- The development of experimental scenarios to evaluate performance, demonstrate generality, and measure robustness

**Workshop Format**
The workshop will consist of invited presentations (45 min. each) and regular presentations (25 min. each). These will be included in the IROS 2007 Tutorials and Workshops DVD.

**About the Organizers**
The organizers of the proposed workshop have been actively involved in performance evaluation and benchmarking for robotics. Prof. del Pobil organized and chaired an IROS 2006 Workshop on “Benchmarks in Robotics”. Ms. Messina and Dr. Madhavan are General Chair and Program Chair, respectively, of the 2007 “Performance Metrics for Intelligent Systems (PerMIS)” Workshop series. The organizers are currently involved in several projects to define methods for measuring cognitive abilities for robots, both at the component and the systems level.

**Intended Audience**
The primary audience of the proposed workshop is intended to be researchers and practitioners both from academia and industry with an interest in cognitive robotics and how these approaches can be utilized in generating intelligent behaviors amidst uncertainty for robots in the service and commercial sectors. The workshop is also aimed at benchmarking and objectively evaluating performance of such robots. Accordingly, it is envisioned to be useful for anyone who has an interest in quantitative performance evaluation of robots and/or robot algorithms.

**Important Dates**
Confirmation of Participation & Abstract
August 7th, 2007
Submission of Final Papers
Sept. 7th, 2007

**Contact Details**
Angel del Pobil, Univ. Jaume I
pobil@icc.uji.es
Raj Madhavan, ORNL/NIST
raj.madhavan@nist.gov
Elena Messina, NIST
elena.messina@nist.gov
Program

*Synthetic Approach to Cognitive Systems: A Perspective from Cognitive Robotics,*
Kaz Kawamura

*Benchmarking Urban 6D SLAM,*
Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernardo Wagner

*The Jacobs Test Arena for Security, Safety, and Rescue Robotics (SSRR)*
Andreas Birk, Kaustubh Pathak, Jann Poppinga, Soren Schwertfeger, Max Pfingsthorn and Heiko Bulow

*Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems*
Benjamin Balaguer, Stefano Carpin, Stephen Balakirsky

*Reliability Testing for Embodied Autonomous Systems*
L. F. Gunderson and J. P. Gunderson

*Advances in the Framework for Automatic Evaluation of Obstacle Avoidance Methods*
J.L. Jimenez I. Rañó, J. Minguez

*Good Experimental Methodologies in Robotics: State of the Art and Perspectives,*
Fabio P. Bonsignorio, John Hallam, and Angel P. del Pobil
Introduction

Today robotics technology is broadening its applications from factory to more general-purpose applications in domestic and public use, e.g., partner to the elderly, rehabilitations, search and rescue, etc. If robotics technology is to be successful in such unstructured, dynamic environments, it will need to meet new levels of robustness, physical dexterity and cognitive capability. This presentation discusses an emerging field called cognitive robotics from a developmental point of view, i.e. “learning from building”. Research topics and features of cognitive robotics are introduced. A case study of internal rehearsal for performance enhancement for cognitive robots will be introduced.

Cognitive Robotics: An Overview

Cognitive Robotics is an emerging field of robotics. Currently there is no generally accepted definition of cognitive robotics since cognition like intelligence is difficult to define. However, the field of cognitive robotics generally considered to comprise the design and use of robots with human-like intelligence in perception, motor control and high-level cognition. To realize cognitive robots many overlapping disciplines are needed, e.g. robotics, artificial intelligence, cognitive science, neuroscience, biology, psychology, and cybernetics. Thus, attempting to tightly define the subject is not constructive as often its nature is amorphous, growing and a strict definition could exclude future relevant work.

The origin of modern cognitive robotics comes from the field of cybernetics, the study of control and communication in living organisms, machines and organizations. The term cybernetics was popularized by Norbert Wiener in his 1948 book. Cybernetics had a crucial influence on many important concepts such as goal-directed behavior generation, self organization and situated nature of intelligence, which are now commonly used in the intelligent robotics community.

As the opening paragraph of this section stated, the goal of cognitive robotics is to build a robot with human-level intelligence. There are two problems with this statement. First is “intelligence”. The term intelligence is used in a variety of situations and is difficult to quantify. Another problem is “human-like”. What do we mean by human? Do we mean an ordinary adult? How about children? Even a three-year child can show many levels of intelligence. This simple argument may be enough to show the problems with the term “human-level intelligence” to evaluate the performance of cognitive systems such as assistive robots (Figure 1).

Cognitive Robot and Internal Rehearsal

It is known that humans are able to have sensory experiences in the absence of external stimuli. It thus seemed reasonable to assume the existence of an ‘inner sense’ where sensory experiences and consequences of different behaviors may be anticipated in cognitive robots. The idea of the existence of such an inner sense does not necessary go against the theory of embedded intelligence advocated by a number of researchers who de-emphasize the role of internal world models and instead emphasize the situated and embodied nature of intelligence. An alternative to internal world models is the ‘simulation hypothesis’ by Hesslow [1] which accounts for the ‘inner world’ in terms of internal simulation of perception and behavior. Our approach may be termed as a “grounded internal simulation” utilizing one type of internal representation of perception and behavior (Figure 2).

Conclusion

Cognitive robotics is still an evolving field with many possible and exciting future directions. This talk presented a synthetic approach to realize cognition such as “Cognition, we want to say, requires both fluent real-world coupling and the capacity to improve such engagements by the use of de-coupled, off-line reasoning” [2] for robots and the difficulties in evaluating the performance of cognitive robots using assistive robots.

Reference:
Benchmarking Urban 6D SLAM

Oliver Wulf, Andreas Nüchter, Joachim Hertzberg, and Bernardo Wagner

Abstract—In the past many solutions for simultaneous localization and mapping (SLAM) have been presented. Recently these solutions have been extended to map large environments with six degrees of freedom (DoF) poses. To demonstrate the capabilities of these SLAM algorithms it is common practice to present the generated maps and successful loop closing. Unfortunately there is often no objective performance metric that allows to compare different approaches. This fact is attributed to the lack of ground truth data. For this reason we present a novel method that is able to generate this ground truth data based on reference maps. Further on, the resulting reference path is used to measure the absolute performance of different 6D SLAM algorithms building a large urban outdoor map.

I. INTRODUCTION

Algorithms for solving the robotic simultaneous localization and mapping (SLAM) problem are a key scientific issue in mobile robotics research. Solutions to SLAM are of core importance in providing mobile robots with the ability to operate with real autonomy. SLAM algorithms integrate robot action and sensor readings and exploit the fact that previously mapped areas are recognized. Global optimization methods yield consistent maps. Nevertheless, these consistent maps might be incorrect and therefore ground truth experiments have to be made. This paper presents ground truth experiments using a novel empiricism.

Popular mapping algorithms work with 3DoF pose estimates, i.e., robot poses are represented by three degrees of freedom $P = (x, y, \theta)$. For indoor environments this choice is appropriate, but a current trend for mapping outdoor environments are mapping algorithms that represent poses in 6DoF, i.e., 6D SLAM [17]. These algorithms consider the 6DoF pose $V = (x, y, z, \theta_x, \theta_y, \theta_z)$ of the mobile robot with 3 position coordinates and roll, pitch and yaw angles. Robot motion and localization on natural surfaces must regard these 6 degrees of freedom. Recently, 3D mapping of large environments received much attention, [4], [18], [24]. A framework for benchmarking these large experiments is still missing.

This paper evaluates algorithms and methods for autonomous mapping. A mobile robot, equipped with a fast 3D scanner gages the environment, while it is steered through a large urban environment. The maps generated by online and offline algorithms are compared to odometry based, gyro based and GPS based pose estimates. Ground truth is provided by a Monte Carlo Localization (MCL) using accurate reference maps.

A. Ground Truth Experiments

In doing experiments with ground truth reference, researchers aim to measure the objective performance of a dedicated algorithm. Based on this benchmark it is possible to give an experimental prove of the effectiveness of a new algorithm. Furthermore measuring the performance of algorithms allows to optimize the algorithm and to compare it to other existing solutions.

Benchmarking is a common scientific instrument. A good example for successful performance measurement in computer science is the computer vision community. There are several projects that aim at providing image data bases to other researchers [12] [23]. These image databases are supplemented by ground truth images and algorithms that calculate performance metrics. In doing so, the community is able to make progress and to document its progress in fields like image segmentation and object recognition.

Unfortunately this kind of performance measurement is not widely spread in the robotics community. Even though there are several ways of comparing the performance of robotic algorithms and systems, one basic step is to provide experimental data and results to other research groups. Up to now this is only done by small projects [14] [19] or...
individual researchers. Another way of comparing robotic systems are competitions like RoboCup [7], ELROB [8] or the Grand Challenge [5]. With this kind of competitions it is possible to measure the level of system integration and the engineering skills of a certain team, but it is not possible to measure the performance of a subsystem or a single algorithm.

Objective benchmarking of localization and mapping algorithms is only achieved by comparing of experimental results against reference data. The practical problem is the generation of this ground truth data. In computer vision, ground truth data is either available for synthetic images, or needs to be hand labeled. In case of mobile robot navigation one way of gathering ground truth data is the use of precise global positioning systems (RTK-GPS) [11]. Unfortunately, this data is only available in open outdoor environments and not for urban outdoor environments or indoor environments. Another possibility is to use complex external measurement setups.

Another benchmarking method for robotic algorithms comprises simulation. Realistic simulation enables researchers to perform experiments with defined conditions and to repeat these experiments. However, real life differs from simulation. Experiments, involving sophisticated sensors such as cameras or laser scanners can only be simulated up to a certain level of accuracy, e.g., capturing environments must regard surface properties such as material, local structures and reflexions. Therefore, using real robotic data sets is favored for benchmarking.

With this paper, we present a novel method of gathering ground truth data in indoor and urban outdoor environments. The procedure is making use of a highly accurate environment map (provided by the land registry office), a Monte Carlo Localization that matches sensor data against the reference map and a manual supervision step.

B. State of the Art in Metric Robotic Mapping

1) Planar Mapping: State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and uncertain perception models. Through integration of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here: Once detected, they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun [21] surveys existing techniques, i.e., maximum likelihood estimation, expectation maximization, extended Kalman filter or (sparsely extended) information filter SLAM. FastSLAM [22] approximates the posterior probabilities, i.e., robot poses, by particles.

SLAM in well-defined, planar indoor environments is considered solved. In principle probabilistic methods are extendable to 6DoF. However, to our knowledge no reliable feature extraction mechanisms nor methods for reducing the computational cost of multihypothesis tracking procedures like FastSLAM (which grows exponentially with the degrees of freedom) have been published.

2) Mapping Environments in 3D: An emerging research topic is 6D SLAM, i.e., while mapping the robot pose is represented with six degree of freedom. In previous work, we used a 3D laser range finder in a stop-scan-match-go-process to create a 3D map of the environment by merging several 3D scan into one coordinate system [17], [20]. Similar experiments have been made by Newman et al. [16]. A current trend in laser based 6D SLAM is to overcome stop- and go fashion of scan acquisition by rotating or pitching the scanner while moving [4], [24], [25]. In the most recent work Pfaff et al. [18] employ two rotating SICK scanners for data acquisition, odometry, IMU and DGPS positioning, a variant of the iterative closest point (ICP) algorithm and a loop closing procedure to map large urban environments in 3D.

Feature-based 6D SLAM methods are investigated by Udo Frese, who adapted his fast treemap algorithm to six degrees of freedom [10]. Among the category of feature based 6D SLAM are the visual SLAM methods, i.e., the MonoSLAM system of Davison et al. [6].

The remainder of the paper is structured as follows: Next, we describe the sensor system for generating large 3D maps and the two pairs of evaluated mapping algorithms. In section III we present the MCL based benchmarking technique. Then we present results from an experiment consisting of 924 3D scans. Section VI concludes.

II. GENERATION OF LARGE URBAN 3D MAPS

A. 3D Range Sensor

The sensor that has been employed for the experiments is a fast 3D laser range scanner, developed at the Leibniz Universität Hannover (see Fig. 2). As there is no commercial 3D laser scanner available that meets the requirements of mobile robots, it is common practice to assemble 3D sensors out of standard 2D laser range sensors and additional servo drives.

The specialties of our RTS/ScanDrive are a number of optimizations that are made to allow fast scanning. One mechanical optimization is the slip ring connection for power and data. This connection allows continuous 360° scanning without the accelerations and high power consumption that are typical for panning systems. Even more important than the mechanical and electrical improvements is the precise synchronization between the 2D laser data, servo drive data and the wheel odometry. Having this good synchronization, it is possible to compensate systematic measurement errors and to measure accurate 3D point clouds even with a moving robot. Detailed descriptions of these 3D scanning methods and optimizations are published in [27].

Having these optimizations described above the limiting
Thus the measurement time for 3D scans with 2 scanner makes use of two SICK LMS 2D laser scanners. For this reason we building faster SICK LMS 2xx based 3D scanners is the use with a SICK LMS 2xx sensor in one second. The only way of number of 13575 (75 \times 181) points that can be measured with a SICK LMS 2xx sensor in one second. The only way of building faster SICK LMS 2xx based 3D scanners is the use of multiple 2D measurement devices [24]. For this reason we first present the RTS/ScanDriveDuo with this paper. This 3D scanner makes use of two SICK LMS 291 2D laser scanners. Thus the measurement time for 3D scans with 2^o horizontal and 1^o vertical angle resolution is reduced to 1.2 sec. In this case one 3D scan measured in 1.2 sec consists of 32580 (180 \times 181) 3D points.

B. The Mobile Robot Erika

The mobile service robot Erika is build out of the Modular Robotics Toolkit (MoRob-Kit). The over all size (LxWxH) of Erika is 95x60x120cm. With its differential drive motors it is possible to drive up to 1.6m/s in indoor and urban outdoor environments. The battery capacity is designed to supply the electric wheelchair motors, sensors and a 700MHz Embedded PC for at least 2 hours or 5km.

In addition to the 3D laser scanner the mobile robot is equipped with wheel odometry, a 3 axis gyroscope and a low-cost SiRF III GPS receiver. The measured data of the wheel odometry and the gyroscope are fused to result in the OdometryGyro that is used as the internal sensor for both MCL and SLAM. In contrast to the odometry sensor the GPS receiver that has got no influence on neither the MCL nor the SLAM results. It is only logged to have another laser independent reference.

C. 6D SLAM with ICP based Scan Matching

We use the well-known Iterative Closest Points (ICP) algorithm [1] to calculate the transformation while the robot is acquiring a sequence of 3D scans. The ICP algorithm calculates iteratively the point correspondence. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation \((R,t)\) for minimizing the equation

\[
E(R,t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left| \left| m_i - (Rd_j + t) \right| \right|^2 , \tag{1}
\]

where \(N_m\) and \(N_d\) are the number of points in the model set \(M\) or data set \(D\), respectively and \(w_{i,j}\) are the weights for a point match. The weights are assigned as follows: \(w_{i,j} = 1\), if \(m_i\) is the closest point to \(d_j\) within a close limit, \(w_{i,j} = 0\) otherwise. The assumption is that in the last iteration the point correspondences are correct. In each iteration, the transformation is calculated by the quaternion based method of Horn [13].

To digitalize environments without occlusions, multiple 3D scans have to be registered. Consider a robot travelling along a path, and traversing \(n + 1\) 3D scan poses \(V_0, \ldots, V_n\). A first straightforward method for aligning several 3D scans taken from the poses \(V_0, \ldots, V_n\) is pairwise ICP, i.e., matching the scan taken from pose \(V_1\) against the scan from pose \(V_0\), matching the scan taken from \(V_2\) against the scan from pose \(V_1\), and so on. Here the model set \(M\) is formed the the 3D data from pose \(V_{i-1}\) and the data set \(D\) that of the pose \(V_i\) for all \(i \in [1, n]\). A second plausible method is to form of all previously acquired 3D scans a so called metascan and match the last acquired one against this metascan. This method is called metascan ICP. Here, the model set \(M\) consists of the union of the 3D scans from the poses \(V_{0}, \ldots, V_{n-1}\) and the data set \(D\) that of pose \(V_i\), for all \(i \in [1, n]\).

D. 6D SLAM with Global Relaxation

Both, pairwise ICP and metascan ICP correct the robot pose estimates, but registration errors sum up. SLAM algorithms use loop closing to bound these errors. If two estimated robot poses \(V_i\) and \(V_j\) are close enough, i.e., their distance falls below a threshold (here: 5 meter) then we assume these scans overlap and are matchable. To a graph, initially containing the sequence of all poses \((V_0, V_1), (V_1, V_2), \ldots, (V_{n-1}, V_n)\), the edge \((V_i, V_j)\) is added. While processing the scans with pairwise ICP or metascan matching, we detect closed loops using this simple distance criterion. Once detected, a 6DoF graph optimization algorithm for global relaxation based on the method of Lu and Milios [15] is employed, namely Lu and Milios style SLAM (LUM). This is a variant of GraphSLAM. Details of the 6DoF optimization, i.e., how the matrices have to be filled, can be found in [2], thus we give only a brief overview here:

Given a network with \(n + 1\) nodes \(X_0, \ldots, X_n\), representing the poses \(V_0, \ldots, V_n\), and the directed edges \(D_{i,j}\), we aim at estimating all poses optimally to build a consistent map of the environment. For simplicity, the approximation that the measurement equation is linear is made, i.e.,

\[
D_{i,j} = X_i - X_j \tag{2}
\]

An error function is formed such that minimization results in improved pose estimations:

\[
W = \sum_{(i,j)} (D_{i,j} - D_{i,j})^T C_{i,j}^{-1} (D_{i,j} - D_{i,j}). \tag{3}
\]
where $\bar{D}_{i,j} = D_{i,j} + \Delta D_{i,j}$ models random Gaussian noise added to the unknown exact pose $D_{i,j}$. This representation involves to resolve the non-linearities resulting from the additional roll and pitch angles by Taylor expansion. The covariance matrices $C_{i,j}$ describing the pose relations in the network are computed, based on the paired closest points. The error function eq. (3) has a quadratic form and is therefore solved in closed form by sparse Cholesky decomposition. The algorithm optimizes eq. (3) gradually by iterating the following three steps: First, for every network link the corresponding covariance is computed based on the point correspondencies of the scan matching. Then the error function (3) is minimized by solving a linear system of equations. In the third step, the local transformations are applied to the poses, resulting in improved pose estimates.

Using the global optimization, two more strategies have been implemented: In pairwise LUM, we use pairwise matching of scans for initially estimating the robot poses. After a loop has been closed, the global relaxation to all previously acquired scans is applied. In metascan LUM, every new scan is initially matched against all previously acquired scans. In both algorithms, global relaxation is started after a closed loop is detected. The relaxation considers all previously acquired scans.

E. Mapping Strategies

Fig. 3 depicts how the mapping strategies are interleaved. 6D SLAM is the result of a 6 DoF ICP algorithm combined with the extension of Lu/Milios Scan Matching to 6 DoF as global relaxation. The SLAM backend uses fast matrix computations exploiting the sparse structure of the corresponding SLAM graphs [3]. Using different path in Fig. 3 the different mapping strategies are created.

Animations of the four mapping strategies, pairwise ICP, metascan ICP, pairwise LUM, metascan LUM are given in the accompanying video and on the following web page: http://kos.informatik.uni-osnabrueck.de/download/6DSLAMbenchmarking. Note the maps presented in the video are rotated about 190°.

III. BENCHMARKING TECHNIQUE

This paper introduces a new benchmarking technique for SLAM algorithms. The benchmark is based on the final SLAM results and a reference position that is obtained independently of the SLAM algorithm under test.

As highly accurate RTK-GPS receivers can not be used in urban outdoor environments, we present a technique that is based on surveyed maps as they can be obtained from the German land registry offices. The process of generating this ground truth reference positions can be divided into a Monte Carlo Localization step that matches the sensor data to the highly accurate map and a manual supervision step to validate the MCL results.

As the SLAM algorithm under test and the MCL algorithm use the same sensor data, the SLAM results and the reference positions are not completely independent. But on the other hand, global localization algorithms and incremental localization and mapping algorithms work differently. Incremental mapping algorithms like odometry and SLAM can suffer from accumulating errors and drift effects. However, pure localization algorithms eliminate these errors by continuously matching to an accurate given map. For this reason the remaining error of the manually supervised reference position is at least an order of magnitude smaller than the discussed SLAM errors.

A. Reference Map

As part of their geo information system (GIS) the German land registration offices features surveyed data of all buildings within Germany. The information about these buildings
Liegnenschaftskarte (ALK)`. The vector format contains lines that represent the outer walls of solid buildings. Each line is represented by two points with northing and easting coordinates in a Gauss-Krüger coordinate system. The upper error bound of all points stored in the ALK is specified to be 4 cm. Up to now there are no further details about doors, windows or balconies available.

B. Monte Carlo Localization

The Monte Carlo Localization (MCL) is a commonly used localization algorithm that is based on particle filtering [9]. As the theory of MCL is well understood we focus on the sensor model that is used to match the 3D sensor data to the 2D reference map with this paper.

The key problem of matching a 3D laser scan to a 2D map is solved by using a method called Virtual 2D Scans [25]. The method splits up into two steps. The first step reduces the number of points in the 3D point cloud. The reduction step is based on the assumption that the reference map presents plain vertical walls. For this reason all 3D measurement points that do not belong to plain vertical surfaces need to be removed (Fig. 4). A sequence of 3D segmentation and classification algorithms that is used to do this reduction in urban outdoor environments is described in [26]. By this means the ground floor, vegetation and small objects are removed from the 3D data. Measurement points on the outer walls of buildings and on other unmapped vertical obstacles remain.

Having this reduced 3D point cloud, the second step of the Virtual 2D Scan method is a parallel projection of the remaining 3D points onto the horizon plane. After this projection the z coordinate contains no information and can be removed. By this means, the Virtual 2D Scan has got the same data format as a regular 2D scan. Thus it can be used as input data of a regular 2D MCL algorithm. To reduce the computational complexity of the successive MCL algorithm the remaining measurement points are randomly down sampled. Experimental results show that less than 100198 results are needed for sufficient localization. Thus the average 3D point cloud with about 30000 measurement points is reduced to a Virtual 2D Scan with only 100 point without loosing information that is needed for localization in urban outdoor environments.

Due to the 2D nature of the reference map and the used 2D MCL algorithm it is only possible to estimate the 3DoF pose $P^{REF} = (x, y, \theta_z)$ of the robot. There is no reference information on the robots height $z$. Further the roll and pitch components $\theta_x, \theta_y$ of the 6DoF robot pose can not be estimated with this 2D method. These angles need to be measured and compensated with a gyro unit before the generation of the Virtual 2D Scans.

C. Manual Supervision

Unlike MCL algorithms used in fully autonomous navigation the generation of reference positions needs manual supervision. Even though the human supervisor is not able to identify the absolute accuracy of the estimated MCL position, it is possible to check the conditions that are needed for proper operation. If all of these conditions are fulfilled the MCL algorithm is able to find the true position of the robot in global coordinates.

There are several conditions that need to be checked to attest proper operation:

- At first the sensor data needs to be checked for a sufficient number of landmarks. Namely, walls as they are given in the reference map. In case of an open area without landmarks in the surrounding of the robot, occluded landmarks or insufficient Virtual 2D Scans the MCL results only depend on odometry and are therefore not accurate.

- The second step is to supervise the numerical condition of the particle filter. As a particle filter only presents a sampled belief an efficient distribution of the finite number of particles is essential for correct operation. For this reason the human supervisor needs to make sure that enough particles are located around the true position. The estimated position can be corrupt if particles are located around more than one maximum or around wrong local maxima.

- Finally, the human supervisor can valuate the overall soundness of the localization and mapping results. For this reason it is necessary to display the reference map with overlaid sensor data. As the sensor data is transformed with the MCL results, fatal matching errors can be detected by the supervisor.

D. Benchmark Criteria

Up to this point the MCL positions and SLAM positions are given in different coordinate systems. The MCL positions are given in the global Gauss-Krüger coordinate system of the reference map and the SLAM positions are given in a local coordinate system that is centered in the robots start position. To be able to compare the positioning results it is necessary to transform the SLAM positions into the global coordinate system based on the known start position.

Having the trusted MCL reference $P^{REF}$ and the SLAM measurement points are needed for sufficient localization. Thus the average 3D point cloud with about 30000 measurement points is reduced to a Virtual 2D Scan with only 100 point without loosing information that is needed for localization in urban outdoor environments.

Due to the 2D nature of the reference map and the used 2D MCL algorithm it is only possible to estimate the 3DoF pose $P^{REF} = (x, y, \theta_z)$ of the robot. There is no reference information on the robots height $z$. Further more the roll and pitch components $\theta_x, \theta_y$ of the 6DoF robot pose can not be estimated with this 2D method. These angles need to be measured and compensated with a gyro unit before the generation of the Virtual 2D Scans.

C. Manual Supervision

Unlike MCL algorithms used in fully autonomous navigation the generation of reference positions needs manual supervision. Even though the human supervisor is not able to identify the absolute accuracy of the estimated MCL position, it is possible to check the conditions that are needed for proper operation. If all of these conditions are fulfilled the MCL algorithm is able to find the true position of the robot in global coordinates.

There are several conditions that need to be checked to attest proper operation:

- At first the sensor data needs to be checked for a sufficient number of landmarks. Namely, walls as they are given in the reference map. In case of an open area without landmarks in the surrounding of the robot, occluded landmarks or insufficient Virtual 2D Scans the MCL results only depend on odometry and are therefore not accurate.

- The second step is to supervise the numerical condition of the particle filter. As a particle filter only presents a sampled belief an efficient distribution of the finite number of particles is essential for correct operation. For this reason the human supervisor needs to make sure that enough particles are located around the true position. The estimated position can be corrupt if particles are located around more than one maximum or around wrong local maxima.

- Finally, the human supervisor can valuate the overall soundness of the localization and mapping results. For this reason it is necessary to display the reference map with overlaid sensor data. As the sensor data is transformed with the MCL results, fatal matching errors can be detected by the supervisor.

D. Benchmark Criteria

Up to this point the MCL positions and SLAM positions are given in different coordinate systems. The MCL positions are given in the global Gauss-Krüger coordinate system of the reference map and the SLAM positions are given in a local coordinate system that is centered in the robots start position. To be able to compare the positioning results it is necessary to transform the SLAM positions into the global coordinate system based on the known start position.

Having the trusted MCL reference $P^{REF}$ and the SLAM
Fig. 5. Orientation errors. Comparing internal sensors measurements, GPS headings and metascan ICP matching with orientations computed by MCL localization. The x-axis represents the 3D scan index, roughly corresponding to the position at the robot path.

to calculate objective performance metrics based on position differences. The first metric based on the 2D Euclidean distance between the SLAM and MCL position

\[ e_i = \sqrt{(x_{i}^{\text{SLAM}} - x_{i}^{\text{REF}})^2 + (y_{i}^{\text{SLAM}} - y_{i}^{\text{REF}})^2}. \] (4)

The second metric is based on the difference between the SLAM und MCL orientation

\[ e_{\theta,i} = |\theta_{z,i}^{\text{SLAM}} - \theta_{z,i}^{\text{REF}}|. \] (5)

As the MCL position has got only 3DoF, the robots elevation, roll and pitch angle can not be tested.

To compare the performance of different SLAM algorithms on the same data set, it is possible to calculate scores like the standard deviation

\[ \sigma = \sqrt{\frac{1}{n+1} \sum_{k=0}^{n} e_i^2}, \] (6)

and the error maximum

\[ e_{\text{max}} = \max e_i. \] (7)

Of course these statistic tests can be done analogously on the orientation errors \( e_{\theta,i} \), resulting in the scores \( \sigma_{\theta} \) and \( e_{\theta,\text{max}} \).

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

The presented experiment has been carried out at the campus of the Leibniz Universität Hannover. The experimental robot platform that was used to collect the data was manually driven on the 1.242 km path closing a total of 5 small and large loops. On this path 924 full 3D scans have been collected at an average robot speed of 4 km/h and a maximum speed of 6 km/h. In addition to the 3D laser data wheel odometry and fused wheel/gyro odometry have been stored with a data rate of 10 Hz. And the position fixes of a low-cost GPS have been logged with 1 Hz.

B. Ground Truth Data

The section of the ALK that is used as the reference map contains 28 buildings represented by 413 line segments. To avoid huge coordinate numbers a constant offset of 5806400 m northing and 3548500 m easting is subtracted from all Gauss-Krueger coordinates. This offset corresponds to the position 52°23′58″ north, 9°42′41″ east in WGS84 coordinates.

The MCL reference positions are calculated online on the Pentium III 700 MHz processor included in the 3D sensor. The particle filter runs with 200 samples and a generous estimate of the sensor variance of 30 cm. This estimate includes the sensor range error, errors from scanning while moving and map uncertainties. The localization results are plotted as a solid gray line in Fig. 1.

The result of the offline manual observation is that the MCL positions can be used as reference positions for 3D scan indexes 1 to 197 and 242 to 924. On the other hand positions corresponding to 3D scan indexes 198 to 241 cannot be used as there are not enough landmarks visible to the 3D sensor (MCL error box in Fig. 1). Due to that particles diverge and the calculated position follows the drifting odometry. Starting with 3D scan 138 the Virtual 2D Scan contains new landmarks and thus the MCL converges quickly to the true position.

For that reason results from 3D scan indexes 198 to 241 are not considered in the following analysis.

C. Mapping Results

1) Mapping with Internal Sensors and GPS: Since all sensors are inaccurate the maps generated using internal sensors for pose estimation are of limited quality as has been demonstrated many times before. For odometry and the gyro based localization the error for orientation and position are potentially unbounded. However, since paths usually contain left and right turns, these errors partially balance. The GPS shows problems close to buildings, where the orientation is poorly estimated and the position error reaches its maximal value. Fig. 5 shows the orientation errors of the internal sensors in comparison to ICP scan matching.
2) Mapping with ICP: Mapping with ICP was done using two different methods, namely pairwise ICP and metascan ICP. The latter method outperforms pairwise ICP since it considers all previously acquired 3D scans leading to slower error accumulation. Fig. 6 shows the scan matching errors in comparison to methods using explicit loop closure that are described next.

3) Mapping with ICP and Global Relaxation: The performance of the methods pairwise LUM, metascan LUM have also been evaluated. As expected, loop closing reduces the position error at the positions, where the loop is closed to approximately zero, e.g., Fig. 6 at scan index 100, where the first loop was closed and at the indices 300–400 and 600–700. At these locations, the Lu/Milios style SLAM methods outperform the pairwise ICP and metascan ICP methods. However, pairwise LUM, and metascan LUM may also fail, if the loop cannot be closed. This case occurs in our experiment in the final part of the trajectory, i.e., when the scan index is greater than 700 (cf. Fig. 6 and Fig. 9). This last loop was not detected by the threshold method described in section II-D.

Finally, Tab. I and II compare all localization/mapping methods. Fig. 7 shows the final map generated with metascan LUM. The left part contains the first 720 3D scans that have been matched correctly, whereas the right part contains all scans including the errors, due to the undetected loop. Fig. 8 shows a 3D view of the scene including two close-up views.

D. Computational Requirements

Of the compared mapping methods only the internal sensor based and the pairwise ICP are online capable. Pairwise ICP using an octree based point reduction and kd-tree search are performed in less than 1.2 sec. using standard computing hardware. In metascan ICP, mapping the computing time for 200 scans increases with the number of scans; therefore, the scan matching time increases to 11.2 sec. for

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>POSITION ERRORS [M].</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>σ_e, max</td>
</tr>
<tr>
<td>Odometry</td>
<td>55.1 261.2</td>
</tr>
<tr>
<td>OdometryGyro</td>
<td>64.7 250.1</td>
</tr>
<tr>
<td>GPS</td>
<td>5.8 95.1</td>
</tr>
<tr>
<td>pairwise ICP</td>
<td>5.2 21.8</td>
</tr>
<tr>
<td>metascan ICP</td>
<td>3.8 13.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>ORIENTATION ERRORS [DEG]</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>σ_θ, max</td>
</tr>
<tr>
<td>Odometry</td>
<td>77.2 256.6</td>
</tr>
<tr>
<td>OdometryGyro</td>
<td>15.1 56.7</td>
</tr>
<tr>
<td>GPS</td>
<td>27.3 171.0</td>
</tr>
<tr>
<td>pairwise ICP</td>
<td>6.3 17.7</td>
</tr>
<tr>
<td>metascan ICP</td>
<td>2.4 11.8</td>
</tr>
<tr>
<td>metascan LUM</td>
<td>4.3 21.2</td>
</tr>
</tbody>
</table>
matching scan No. 920 with all previous ones, i.e., matching 32580 against 29 Mio. points.

Pairwise LUM and metascan LUM spend additional time on computing the point correspondences for scans represented by the nodes in the graph. Due to the iteration required by our GraphSLAM algorithm, both methods are not online capable [2]. The total map processing time was 207 min and 371 min, respectively. The largest portion of the computing time was spent by calculating closest points.

V. Justification of the Results

To validate our experimental methodology, i.e. generating ground truth reference positions using MCL as described, we match the acquired 3D scans with a 3D map generated from the 2D reference map. For this the 2D map is extrapolated as 3D map, by cuboids representing the boundaries of the buildings. Fig. 10 shows the final map with the point clouds representing the buildings.

This 3D map which is a 2D map extended by some height
is used for comparison using the following three strategies:

1) The ICP algorithm is used to match every single 3D scan with the point cloud based on the 2D map. As it turns out, this method can only be applied to the first 200 scans, since the map does not cover the whole robot path. However, in comparison MCL sucessfully deals with this problem by applying the motion model to the particles until scan matching is possible again. This method is referred to as \textit{map ICP (first part)}.

2) The ICP algorithm is used to match a 3D scan with the metascan consisting of the 3D points from the map and all previous acquired and registered 3D scans. The method will be called \textit{metascan map ICP}.

3) The previous method is used with the extention, that points classified as ground are not included, i.e., only \textit{metascan map ICP wo ground}. It is expected that this restriction results in better convergence of the ICP algorithm.

Fig. 11 and 12 compare the additional localization/mapping methods. Tab. III and IV gives quantitative results. It turns out that these justification methods are behave similar to MCL and produce comparable results, i.e., the MCL trajectory differs only by statistical nois from the trajectories produced by ICP scan matching using a 3D point cloud derived from the map.

VI. CONCLUSION AND FUTURE WORK

Benchmarking of algorithms and research in experimental methodology are topics that get more and more important in the field of robotics.
for SLAM in urban outdoor environments. The evaluation is based on a comparison of the final SLAM results and ground truth reference positions. In our case these reference positions are generated with a manually supervised Monte Carlo Localization working on surveyed reference maps. Having this reference positions it is possible to calculate objective benchmark scores that can be used to improve and compare evaluation results of SLAM in urban outdoor environments. The evaluation is based on a comparison of the final SLAM results and ground truth reference positions. In our case these reference positions are generated with a manually supervised Monte Carlo Localization working on surveyed reference maps. Having this reference positions it is possible to calculate objective benchmark scores that can be used to improve and compare algorithms. This evaluation technique is demonstrated with experimental data and four different 6D SLAM strategies. The experiment that contains 924 full 3D scans on a 1.2 km path was carried out on the campus of the Leibniz Universität Hannover. Needless to say that much work remains to be done. Future work will be done on two aspects: First, research in robotic benchmarking techniques needs to be emphasized. And second this ideas need to be spread out in the robotics community. To this end, we plan to cooperate with the Radish: The Robotics Data Set Repository [14] and the OpenSLAM [19] project.

VII. ACKNOWLEDGMENTS

The authors would like to thank Dorit Borrmann, Jan Elseberg and Kai Lingemann (University of Osnabrück) for joint development of the Lu/Milios style GraphSLAM with 6 degrees of freedom.

REFERENCES


for SLAM in urban outdoor environments. The evaluation is based on a comparison of the final SLAM results and ground truth reference positions. In our case these reference positions are generated with a manually supervised Monte Carlo Localization working on surveyed reference maps. Having this reference positions it is possible to calculate objective benchmark scores that can be used to improve and compare algorithms. This evaluation technique is demonstrated with experimental data and four different 6D SLAM strategies. The experiment that contains 924 full 3D scans on a 1.2 km path was carried out on the campus of the Leibniz Universität Hannover. Needless to say that much work remains to be done. Future work will be done on two aspects: First, research in robotic benchmarking techniques needs to be emphasized. And second this ideas need to be spread out in the robotics community. To this end, we plan to cooperate with the Radish: The Robotics Data Set Repository [14] and the OpenSLAM [19] project.

VII. ACKNOWLEDGMENTS

The authors would like to thank Dorit Borrmann, Jan Elseberg and Kai Lingemann (University of Osnabrück) for joint development of the Lu/Milios style GraphSLAM with 6 degrees of freedom.
The Jacobs Test Arena for Security, Safety, and Rescue Robotics (SSRR)

Andreas Birk, Kaustubh Pathak, Jann Poppinga, Sören Schwertfeger, Max Pfingsthorn and Heiko Bülow

Abstract—The Jacobs University Bremen features a special test site for mobile robots. It consists of two large arenas with test elements, which are particularly suited to evaluate the performance of systems operating in challenging domains like Security, Safety, and Rescue Robotics (SSRR). The site is one of only six worldwide, which has been established in close cooperation with the US National Institute of Standards and Technology (NIST). This paper presents the Jacobs arenas and a set of metrics for evaluating the mobility, sensing capability, and onboard intelligence of robots. The tests are illustrated by using a Jacobs rescue robot, which is equipped with state of the art sensors. The relative strengths and weaknesses of these sensors are evaluated in a variety of situations; many of them are typically encountered in SSRR applications.

I. INTRODUCTION

In addition to their well-established role as programmable tools for industrial automation, robots are increasingly used as autonomous intelligent systems in challenging environments that are not adapted for automation. These robots have to function without continuous supervision by a human operator. They have to be able to adapt to unforeseen circumstances. They have to deal with various working environments where they encounter an enormous variety of different objects with various types of surface materials, shapes, sizes, and so on. One such application domain is the field of search and rescue robotics. As discussed in detail in [1], rescue robotics is an important domain that can serve as an important milestone on the road to truly autonomous systems.

Existing systems have demonstrated their usefulness in the field and there is a sound, growing market for them. Any bit of autonomous intelligent functionality added can improve their overall usefulness leading up to full autonomy in the long term. But though this line of reasoning is valid on the general level, it is much more difficult to assess the real benefit of a particular implementation of a concrete function. Any additional intelligent capability increases the complexity, requires space and payload capacities, consumes energy, and increases the price. Customers, end users and developers have hence an interest in being able to properly measure the benefits of a particular function and to compare the benefits to the costs.

Task based performance testing, just like observing a mouse in maze, has been proposed for this purpose [2][3]. In addition to the temporary test arenas used at field tests and competitions like RoboCup events, there are six permanent test sites based on the work by the NIST. The following list gives an overview of their locations and the years of opening:

- National Institute of Standards and Technology: Gaithersburg, USA (2000)
- Jacobs University Bremen: Bremen, Germany (2004)

The performance evaluation aspects described in this paper are based on work at the test site at the Jacobs University Bremen. This test site consists of two parts as shown in figures 1 and 2. The first arena is available since spring 2004 at Jacobs [4]. It is based on a high-bay-racking system. This allows a large floor-space on a relatively small area. The first arena has a footprint of 5.60m by 4.70m and it is approximately 6m high. It has 3 main floors and several intermediate floors, which are interconnected. This testbed consists of three different levels, the so-called yellow, red, and the orange zone as motivated in detail in [5][3]. Yellow models an intact building structure with a normal office or home interior that has been mildly affected by a disaster. In the orange zone, the effects to the interior as well as to the building itself are much more severe. Last but not least, the red zone is a model of a pancake collapse with large amount of rubble and highly instable structures.

The second arena provides an additional collection of locomotion and sensing challenges that supplement the original arena. The second test arena is mainly organized in a flat, 2D fashion with a large floor area. It features test elements like maze-like structures, but also some local 3D obstacles like ramps, stairs and elevated floors. It also offers a large collection of random step fields, i.e., test elements that realistically simulate various forms of rubble. Both arenas are furthermore equipped with components to simulate various features of humans like according form, motion, sound or respiration.

II. LOCOMOTION

Mobility is the most fundamental aspect for SSR. The environment presented in the test scenarios has a typical selection of challenges that are to be expected in real scenarios. The simplest scenario element is built as a U-shaped track.
Fig. 1. The first Jacobs test arena with several interconnected floors on a compact footprint.

Fig. 2. The second Jacobs test arena with a large extended area.

(see Fig. 3) in which the robot has to start in the area denoted by S and proceed into the end area E. These scenes are designed based on a square Random Step Field of size 1.35 meters. Random Step Fields are standardized test elements developed by the NIST, which very realistically model the challenges of rubble and uneven terrain. The robot is always placed such that it faces the correct direction ± 45 degrees. The points A, B, C and D depict where obstacles/difficulties might be placed. Usually the robot has to reach the end area E as fast as possible. The robot might touch the walls and other obstacles softly but not ram them hard, which would lead to an abortion of this run with a score of zero.

Fig. 3(b) shows the dimensions of the standard U-shaped track. The width was chosen such that two random step fields fit in the U-shape. The timing line is two meters from the upper border of the U-shape. The lower border is at least two meters from the timing line to allow some variations in the start positions of the robots. A rugged robot or rugbot [6], [7], the search and rescue robot of the Jacobs Robotics Group, is used throughout this paper to demonstrate some of the challenges the test element cover. Fig. 4(a) shows the Rugbot driving toward the end area of an empty U-shape.

The list of locomotion challenges that can be combined to generate a complete test course for a robot is as follows:

- **Empty U-Shape** An empty U-Shape is used to test the most basic teleop and autonomy functions. A system that fails here cannot be expected to continue elsewhere in the performance evaluation.

- **Littered Ground** The U-Shape is filled with loose paper and small sticks. The paper can easily obscure the field of view (FOV) of important sensors while wheeled robots could have problems moving over small sticks.

- **A Short Ramp** A two sided ramp is placed in the U-Shape such that the top on the line between the points A and D, filling the whole passage. The angle of the ramp is approx. 15 degrees while the height is 20 cm. The challenge for e.g. an autonomous robot is to stay at the outer borders of the scene in order to be able to enter the ramp, to climb the ramp, to go over the top, and find its way to the end area.
(a) The task for the robot is to get from S to E U-shape without bumping into the walls or the specific obstacle which might be at A, B, C or D.

(b) The dimensions of the U-shape.

(c) Three random step fields in the U-shape.

Fig. 3. The standard test scene - U-shape.

- **Orange Random Step Fields** Three orange random step fields are placed in the U-shaped track as shown in Fig. 3(c).

- **Red Random Step Fields** Now three red random step fields are used. The reason for using this configuration is that it is even more difficult to make an aimed turn on this difficult terrain than just driving straight. Autonomous robots are not expected to cope with this scenario in the near future.

- **A Steep Ramp** A long and steep ramp as shown in the left image of Fig. 2 has to be crossed by driving the ramp up, turning and driving down again. Autonomous robots most likely will have to be guided into the right direction by placing walls around the entrance and turning corners. The difficulties in this scenario are the entering of the ramp as well as having enough friction to prevent sliding down.

- **Stairs** Climbing up stairs is an important topic in mobile robot locomotion. See Fig. 4(b) for a Rugbot climbing the stairs in the rescue arena.

### III. Mapping

Maps are an important mission delivery in SSRR applications, e.g., to guide first responders to a victim that is found. Also, it is often desirable to have some information whether a certain area is likely to be void of persons. For this purpose, proper map based exploration is a must. Furthermore, mapping is a central issue from the research perspective. As mentioned in the introduction, there is a tremendous potential for fruitful interaction between application oriented work and basic research in the field of SSRR. Especially, this field and therefore also the related performance evaluations based on the NIST arenas can play a major role in the development of autonomous intelligent robots in general.

Map quality as performance evaluation criterion is composed of two main elements: **coverage** and **precision**. The coverage component simply refers to the size of the map, i.e., the amount of area of the environment that is represented in the map. Precision refers to the related accuracy of this representation. With respect to precision there are two fundamentally different approaches. First, there is the option to solely take the topology of the representation into account. Second, a metric error measure between the representation and ground truth can be used. Fig. 6 shows two maps generated by the same robot in parallel. The first one shown on top is based on the classic occupancy grid algorithm [8][9] while the second one shown below uses a state of the art algorithm for simultaneous mapping and localization (SLAM) [10]. The second SLAM algorithm gives a precise metric representation of the environment at the cost of high computational requirements. The first occupancy grid algorithm is less computationally demanding, but it does not give a proper metric representation. Nevertheless, it provides at least a good topological representation of the environment.
As a reproducible metric for a test scenario a *Slalom* course can be chosen. The main criteria are the requirements regarding the capabilities of the robot and the comparability of the designed test arena. Metrics for a slalom course are straightforward and a corresponding test arena can easily be rebuild. The demands on the robots odometry are high due to the many rotations during the course. These reasons finally lead to a comparability between different robot teams. Figure 5 shows the slalom test element at Jacobs. In figure 6 two different maps generated by the Rugbot are shown.

Fig. 5. Slalom test with the robot at the start(left) and the robot traversing the corridor(right). Small sticks are scattered on the way to irritate the odometry of the robot.

![Sensors of the Rugbot](image)

**IV. PERCEPTION**

Rescue scenarios and therefore also our rescue arena pose different challenges to the kind of sensors usually found on a mobile robot. A set of tests, most of them placed in the U-shaped track, have been developed to measure how well the robot supports a human operator coping with these difficulties. Even more challenging is to pass these tests in autonomous mode. To illustrate the problems these challenges have been explored with a Rugbot (Fig. 7). As can be seen from the figure, it has a wide array of sensors, whose output can be seen in figure 8:

- a 3D time-of-flight (TOF) camera (SR: SwissRanger SR-3000 from CSEM),
- a stereo camera (STOC from Videre),
- a webcam,
- a pan-tilt-zoom-camera (from Panasonic), and
- two laser range finders (LRF) (both URG from Hokuyo).

**A. Dark Scene**

In the dark scene the U-shape is covered such that normal cameras provide no information. Points are being awarded if the robot traverses from the start to the end without bumping the arena, which indicates that the environment had been perceived in one form or the other. To cope with this challenge the robot should be equipped with a light or use active light sensors such as laser range finders. In the example given in Fig. 9 an additional heat source (victim) and a box have been placed in the cave. The IR image identifies the heat source. The LRF and the SR determine the spatial structure whereas the optical sensors webcam and Stereo receive only poor images.

**B. Glass**

In this scenario, a part of the middle-wall near the point A and the back wall between B and C are replaced by a transparent material (plexiglass). Many sensors have difficulties with this material, as can be seen in Fig. 10. The webcam is not able to detect the obstacle except for the case when light or objects are reflected. A benefit is the recognition of movement within the image by what a recognition of victims giving signs is possible. The LRF returns error beams (which signify free space). The TOF camera, which uses near IR light, does not detect the glass plate, only the reflection of its illumination unit. The far IR light as detected with the IR camera is absorbed by the glass such that victims behind glass can not easily be spotted this way. Echo-sound sensors, which have not been tested with this robot, are likely to provide the most reliable data with this scene.

**C. Mirror**

The mirror scenario is quite similar to the one above, except that the glass is replaced by a mirror. As could be expected, the result (Fig. 11) are similar to the ones with the plexiglass. Visible and near IR light is reflected, far IR light is not. For sensors, whose output is processed, this can be an advantage as well as a disadvantage: if a mirror is sensed unknowingly, it will produce faulty data. If, however, a mirror is used on purpose, it can be used to look around the corner or to widen the FOV (like in an omni-cam). The images in Fig. 12 show that this is also possible for the 3D sensors. Again echo-sounders might be able to sense mirrors without problems.

**D. Absorbtions**

In this scene black material is used to absorb the beams of typical laser range finders. The back wall from B to C
Fig. 6. The results of two different mapping algorithms running on the same robot, namely once an evidence grid (left) and once a state of the art SLAM algorithm (right). Being able to use either of the two approaches allows to trade processing speed for precision.

(a) Webcam image  (b) Pan-tilt-zoom camera  (c) Pan-tilt-zoom camera, zoomed in  (d) Pan-tilt-zoom camera, zoomed in more
(e) Pan-tilt-zoom camera, maximum zoom  (f) Pan-tilt-zoom camera, panned and tilted to show something else  (g) TOF camera, distance image  (h) TOF camera, intensity image
(i) Stereo camera, disparity image  (j) Stereo camera, greyscale image  (k) Laser range finder  (l) Infrared camera

Fig. 8. The output of all sensors in a scene which does not produce any errors as well as parts of the wall near A are covered with this material. Fig. 13 shows a different scene, where is robot is standing in front of a ramp covered with said material. A unique strength of the TOF camera can be seen there. Both the LRF and the stereo camera fail to deliver correct measurement. For the stereo camera, this is because it has difficulties identifying features its two images due to the repetitive pattern of the cover. For the LRF on the other hand, it is due to the absorptions of the material. This weakness of the 2D case would, of course, also carry over to a possible 3D extension of the LRF.

E. Featureless Scene

Robots relying on stereo cameras will have difficulties with featureless scenes. Those are tested by covering the walls with a plane, white, featureless material. Fig. 14 shows this specific weakness of the stereo camera in another scene: it hardly detects any distances. The TOF camera in contrast has no such weakness. It has just an error at the very edge of the ramp caused by the tape applied there.
Fig. 9. Sensor Data from the cave. The darkness is a challenge to all the sensors which depend on the presence of visible light.

(a) IR image – the simulated victim is clearly spotted
(b) LRF – the quadrangle in the middle correctly captures the shape of the far walls.
(c) TOF camera intensity – due to active sensing, the interior of the cave is visible to the SR.
(d) TOF camera distance – the shape of the cave is correctly measured by the SR.
(e) Stereo disparity image – as it uses visible light, the stereo camera suffers from the low contrast in its images.
(f) Stereo intensity – because of the darkness, only few details can be seen.
(g) Webcam – the image lacks contrast and has high noise level.

Fig. 10. Sensor Data from scenes involving plexiglass

(a) IR camera – The scene contains a heat blanket behind a pane of plexiglass (between the two boxes, recognizable by the reflection of a fluorescent tube in the top right corner). The part of the heat blanket protruding from behind the pane is visible in the IR image on the right, the lower part is not.
(b) LRF – The LRF sees straight through the plexiglass between the two stacks of boxes (visible by the reflections).
(c) TOF camera – When faced with plexiglass, the TOF camera produced a reflection on it (black spot in the middle of the image). This stems from the camera’s illumination unit and leads to erroneous measurements for the rest of the image. The image on the right shows a correct measurement without the plexiglass.

F. Range

Two additional tests are used to score the range of the sensors used. These scores are given for the closest distance to an obstacle where the robot still receives correct data and the farthest distance.

In Fig. 15, output is reproduced from sensors being very close to an obstacle, with two different distances. While the LRF can handle both situations well, the stereo camera already fails in the first set-up and the TOF cam fails in the second set-up. For the stereo cam, the inability to determine distance is due to the large difference in the images of the two lenses. The TOF camera, on the other hand, suffers from its active sensing. The error seen in Fig. IV-F is caused by excessive brightness of the ambient light. This phenomenon can also occur shortly in other situations (such as the first scene), where the camera is still able to automatically adjust the brightness after a short period of time.

In order to calculate the score for each sensor the minium
(a) Webcam image  (b) TOF camera – In the right image, the robot is clearly visible. But as this the distance image where color signifies distance, it should not. All parts of it should have roughly the same color. Also, none of the distances in the mirror are correct. The white/black spot is due to the reflection of the camera’s illumination unit.

(c) Stereo camera – Here, the distances in the mirror appear correct. Yet the mirror is not recognized as such, it looks like a square hole in the wall.

Fig. 11. Sensor Data with errors caused by a mirror

(a) Webcam – the obstacle around the corner is visible in the mirror.  (b) LRF – The two long groups of beams roughly between 11 and 12 o’clock are reflected by the mirror. The shorter beams in between detect the pole which is also visible in the mirror (Subfig. IV-C).

(c) Stereo camera – The pole is visible by its edges.  (d) TOF camera – distances are correctly measured.

Fig. 12. A mirror can be used to look around a corner, but only if it is recognized as such. For a computer, however, this is very difficult. It is more likely to mistake a mirror for a kind of window.

(a) Webcam – A ramp covered with plastic.  (b) LRF – The material absorbs the laser, causing the LRF to report free space in front.

(c) Stereo camera – The repetitive pattern of the material confuses SwissRanger outputs a gradient.  (d) TOF camera – Only the stereo camera such that only the edges are correctly measured.

Fig. 13. Sensor Data from scenes involving absorptions
distance is determined. This average distance over all sensors is then used in the following formula:

\[
\text{Score} = (20\text{cm} - \min(\text{dist}, 20\text{cm})) \times 20. \tag{1}
\]

This leads to a maximum score of 40 and a minimum score of zero at an average distance of 20 or more centimeters.

We also score the maximum usable range of a sensor. Some sensors have considerable difficulties with large distances. While the LRF reliably reports “error beams” for distances beyond its maximum of 4m, the TOF and the stereo camera produce errors, most notably the TOF camera. As it measures distances indirectly by measuring the phase shift of the emitted modulated light, the distance measurement wraps around after 7.5 m. For examples, refer to Fig. 16.

The score is calculated using the average maximum distance of all sensors as follows:

\[
\text{Score} = 40 \times \frac{\min(\text{dist}, 20m)}{20m} \tag{2}
\]

Again the maximum possible score at 20m distance is 40 and the minimum score is zero.

\section*{G. Perception performance of the Rugbot}

To illustrate the performance of the Rugbot a typical rescue scene is shown in Fig. 17: a human. This is, of course, the \textit{raison d’être} for the IR camera being onboard. (Fig. IV-G). The other sensors show different performances. The LRF does generally well, but again (see Sec. IV-D) has problems with the black material. The TOF camera delivers a dense image with errors at the tape. As this scene has too few features for the stereo camera, it can only give distance values for the edges. Additionally, it is confused by the repetitive pattern on the right edge of the image.

In Fig. 8 (page 5), an average scene for rescue robotics is shown: A hallway with a ramp and victim at a distance. This set-up poses no difficulties for any of the sensors. The geometry is correctly captured by the distance sensors, just the far end of the LRF is out of range and produces error beams between 12 and 1 o’clock. In the swiss ranger distance image even the shape of the far away box is captured. The out-of-range measurements in the top right corner are dropped. The stereo camera provides a relatively dense image with correct distances. The cameras deliver good images, especially the zoomed in picture is impressive, and in the image of the IR camera, even the relatively distant heat source is clearly visible.

To round off this summary, a collection of images taken by the Rugbot is shown in Fig. 18, where one of the sensors reported wrong information with respect to the ground truth.

Table I shows a summary of the sensor’s performance in the different scenes. A minus (-) means that the sensor performed very poorly, a zero (0) indicates that the sensors returned mostly correct readings while a plus (+) states that the sensor was not affected by the challenge at all. Regarding perception, one can say that on using sensor fusion, the Rugbot has enough sensor data to cope with most of the challenges quite well.

\section*{V. TESTING ROBOT AUTONOMY}

Due to the high difficulty level of the test arenas, autonomous operation of the robot is usually carried out in a sub-arena which is much simpler for locomotion. In particular, stairs and random step fields are currently not used, although ramps (pitch as well as roll) are used. The main criteria for evaluating autonomy are two-fold: (i) decision making and perception using sensors, and (ii) local and global motion planning. These are discussed in more detail next.
1) **Decision making based on sensor perception:** An autonomously moving robot has to make informed decisions such as in the following situations:

1) **Human detection:** The two common ways a human like a victim or an intruder can be detected are by using IR camera images to capture body heat, and through motion detection using camera images for a static robot. For the latter case, if the robot is equipped with more sophisticated algorithms to detect motion while the robot itself is moving, it gets assigned extra points. Human voice or whistle recognition or any other non-traditional techniques also accrue extra points.

2) **Obstacle detection using sensor-fusion:** In view of the pitfalls encountered by various sensors in different scenarios as described in Sec. IV, the robot should be able to fuse different sensor data, analyze them, and come up with the right identification of the obstacle type. The U-Shaped track along with obstacles of type described in Sec. IV can be used. Another ability is to distinguish a climbable ramp from an obstacle which is insurmountable and hence is to be avoided. Extra points are also awarded if the robot is able to avoid falling down ditches or table tops.

2) **Local and Global Planning:** Local planning involves a reactive approach to obstacle avoidance. The reactive
VI. MAPPING AS GENERAL PERFORMANCE CRITERION

One disadvantage of purely scene based performance evaluation is, that it only covers a part of the problems robots have to cope with, namely locomotion, mapping, perception and autonomy. For successfully completing real SSRR missions the following aspects are of importance, too.

The mobile robots fielded have to be robust enough to succeed in the difficult environment. Their energy source, which are most often batteries, has to last for the typical mission time which is often more than two hours. Another aspect is the communication range which can be quite limited if wireless systems are used. Multi-robot systems and how they coordinate their actions is another increasingly important topic as well as active rescue systems which not only observe but also help by, for example, enabling communication between the victim and the rescue personnel or by providing water or maybe pain relief. Also the interface between the robot and the operator (Graphical User Interface, GUI; Joysticks; etc.) as well as between the operator and the rescue personnel (victim reports; maps; PDA; etc.) is of great importance for the success of the mission.

Some of these aspects (locomotion, perception, mapping, autonomy, robustness, energy, multi-robots systems) can very well be measured if a complete test arena is available. This arena could be temporally one, for example during RoboCup, or a permanent one like the one at Jacobs University. A number of competitors will try to map as much of the arena as possible in the given time. The only criterion needed to measure the performance can then be the coverage and
precision of the generated map.

Whenever possible, an exact metric measure of precision like the mean squared error between the generated map and ground truth is desirable. But when the ground truth is not readily available or the generated map is too imprecise for a meaningful evaluation, a topology match can be used for precision. In doing so, the number of nodes and vertices in the largest common sub-graph of the topological map and an accurate environment representation can be used. In this case, the measure includes coverage at the same time. Mapping as an evaluation criterion automatically reflects robot capabilities in other aspects like locomotion and sensors. Good locomotion implies better coverage and hence a more extensive map of the area. The sensing and perception capabilities of the system are obviously reflected in any map. Some concrete state of the art sensors and related challenges are discussed in more detail later on in this paper. Directly related to the obstacle sensing is, of course, the task of mapping itself, which is an important research issue as well as mission delivery in search and rescue missions. In addition to the sensor related perception challenges, state of the art mapping algorithms differ significantly in their precision but also computational requirements. An overview of current approaches is given in [11]. Several techniques have been developed to solve the so-called simultaneous localization and mapping problem (SLAM), e.g., by using Kalman filter based approaches [12] or expectation maximization techniques [13].

Note that the majority of work on mapping is focused on 2D maps. There is an increasing amount of research on 3D mapping [14], [15], [16], [17] and, as shown in detail later on in this paper, mapping in the arenas requires 3D perception. From the practical viewpoint of an end user perspective as well as with respect to performance evaluation, 2D maps are desirable as final delivery, either by using 3D perception in 2D mapping or by projecting 3D structures onto 2D planes.

Map quality can also be used to assess the autonomous capabilities of a robot. The map coverage is directly linked to its exploration capacities [18], [19]. This includes navigation skills [20] and intelligent decision making to for example find safe terrain for driving [21], [22], [23], [24], [25], [26]. In general, maps play a core role for almost any algorithm for goal-oriented autonomous behaviors [27], [28], [29]. This holds also with respect to cooperation between intelligent robots and humans [30].

Finally, multi-robot cooperation is an important aspect for the evaluation of autonomous intelligent systems. Map quality is also here a reliable indicator. The better the robots cooperate, the larger the amount of coverage by the joined map [15], [31], [32], [33], [34], [35]. The maps again form the main basis for further algorithms like exploration, which in turn are improving the skills of the robots as reflected in even higher amounts of coverage. The multi-robot cooperation can also be used to increase robustness, which is then reflected in the precision of the maps [36], [37].

VII. Conclusion

The paper presented the Jacobs Test Site for Security, Safety, and Rescue Robotics (SSRR). The site consists of two arenas with test elements presented that can be used to evaluate mobile robot performance in a repeatable and comparable manner. The performance assessment covers aspects from locomotion, perception, mapping and autonomy. A Jacobs rescue robot was used to demonstrate the usage of the test elements.

Acknowledgments

The authors gratefully acknowledge the financial support of Deutsche Forschungsgemeinschaft (DFG) for their research. Please note the name-change of our institution. The Swiss Jacobs Foundation invests 200 Million Euro in International University Bremen (IUB) over a five-year period starting from 2007. To date this is the largest donation ever given in Europe by a private foundation to a science institution. In appreciation of the benefactors and to further promote the university’s unique profile in higher education and research, the boards of IUB have decided to change the university’s name to Jacobs University Bremen (Jacobs). Hence the two different names and abbreviations for the same institution may be found in this paper, especially in the references to previously published material.

References


Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems

Benjamin Balaguer, Stefano Carpin
School of Engineering
University of California, Merced, USA
E-mail: {scarpin,bbalaguer}@ucmerced.edu

Stephen Balakirsky
National Institute of Standards and Technology
E-mail: stephen@nist.gov

Abstract—Autonomous robotics has been plagued by the lack of quantitative comparisons between different solutions for the same problem. The situation arose due to a lack of theoretical background, recognized benchmarks, and the existence of a culture that is not oriented towards the free sharing of ready-to-use code for scientific research. In this paper we leverage a recent paradigm shift, and contrast different algorithms for Simultaneous Localization And Mapping (SLAM) readily available to the scientific community. In particular, we have run the same algorithms in two different settings. The first one is based on a P3AT robot operating inside a large building hosting office space and research labs. The second scenario is a virtual replication of the identical floor plan, implemented inside the USARSim simulation environment. In other words, the simulated scenario features the exact models of the environment and robot. The experimental setup offers a matrix where weaknesses and strengths of different SLAM algorithms can be contrasted in real and virtual environments, also outlining the degree to which the simulated results can be extrapolated to measure or predict real world systems performance. We conclude that the availability of open source algorithm implementations, data sets, and simulation environments is the key to promote accelerated research in autonomous robotics. In particular, it appears that available SLAM implementations are robust and easy to use for environments like those used in our experiments, and therefore research efforts should be accordingly re-modulated.

I. INTRODUCTION

One of the cornerstones of the scientific method is repeatability. Experimental tests confirming or disproving a certain theory should be carried out by different researchers and lead to the same conclusions, within the defined error bounds, provided that the operative conditions are the same. Robotics has not yet enjoyed such a rigorous approach, a fact that can be explained by a multitude of reasons. Robots are complicated systems composed of many interacting units, each of them characterized by its own behaviors and errors. Robots’ observed behaviors do not only depend on the software and hardware, but also on the surrounding environment. Evidently, even if two researchers buy the same robot and perform the same experiment with the same software, they are likely to observe very different findings. Another problem that has not helped in making the situation better, but that finds roots in the same issues, is the lack of a widely shared base of reusable code that is maintained and exploited by different research groups. While it is true that certain middle-ware softwares are enjoying significant popularity [1] [2], they are basically interfaces to gain portable access offered in section V.

In this paper we propose to start a systematic investigation and comparison of different algorithms for the well known simultaneous localization and mapping problem (SLAM). In particular, we present a juxtaposition between real world validation and experimental runs within the high fidelity USARSim simulator [5]. The use of a simulator is particularly appealing for quantitative comparisons of SLAM algorithms because it allows the generation of huge data sets without investing too many resources. Moreover, the simulator is highly configurable and allows one to specify different noise levels for the various sensors, thus permitting a careful evaluation of robustness with respect to noise sources.

The paper is organized as follows. In section II we provide an overview of the USARSim software, while in section III we illustrate and contrast three SLAM algorithms. Section IV reports on the experimental setup that was implemented and the results that were produced. Finally, conclusions are
II. THE USARSim FRAMEWORK

The current version of Urban Search and Rescue Simulation (USARSim) [6] is based on the UnrealEngine2 1 game engine that was released by Epic Games as part of Unreal Tournament 2004. The engine may be inexpensively obtained by purchasing the Unreal Tournament 2004 game. The USARSim extensions may then be freely downloaded from [7]. 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. USARSim uses these features to provide controllable camera views and the ability to operate 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 the Unreal 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
- models of several commercial and laboratory robots and effectors
- models of commonly used robotic sensors

While there exists quite a few robotic simulators, USARSim was chosen for many different reasons, the most important of which being its accuracy. Indeed, a lot of time and research is spent every year improving the robotic platforms, authenticating the sensors, building additional robots, sensors, and environments, and validating the physics engine. More specifically, [8] [9] [10] [11] [12] provide details about USARSim validation both quantitatively and qualitatively. Additionally, USARSim provides the same robotic interface as the real P3AT, allowing researchers to run two robots, one in USARSim and one in the real world, with a single input (e.g. a joystick).

A simple but effective command-and-message interface is used to interact with the USARSim robots: string commands are sent to the robot and string messages are sent by the robot. The USARSim interaction standards consist of items such as robot coordinate frame definitions and unit declarations while the API specifies the command vocabulary for robot/sensor control and feedback. Both of these items have become the de facto standard interfaces for use in the RoboCup Rescue Virtual Competition which utilizes USARSim to provide an annual Urban Search and Rescue competition. In 2007 this competition had participation from teams representing 5 countries.

Both laboratory and commercial vehicles with different mobile platforms (skid-steered, Ackerman-steered, omni drive, legged, humanoid, nautical, and aerial) are incorporated within USARSim. Additionally, a set of robotic arms and plan-tilt mechanisms can effortlessly be mounted and utilized on any robot. Figure 1 shows a small collection of some of the robots that USARSim has to offer. The list of available sensors and effectors is also quite extensive and includes range scanners, sonars, cameras, grippers, RFID tags, and INS sensors.

1Certain 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 authors, nor does it imply that the software tools identified are necessarily the best available for the purpose.

Fig. 1. A subset of available robots in the current version of USARSim.

Highly realistic environments are also provided with the USARSim release, ranging from simple planar mazes to multi-level collapsed structures. The environments encompass challenging robotic problems that cover many areas of research including mapping, planning, mobility, cooperation, communication, image-processing, and victim detection. Furthermore, the environments accommodate all the USARSim robotic platforms by providing indoor buildings, urban roads and highways, lakes and rivers, and large flying spaces. Example indoor and outdoor environments may be seen in Figure 2. In addition, an editor delivered for free with the game engine and the ability to import models simplifies the creation of worlds.

It is worthwhile to note that USARSim does not supply a robot controller. In other words, USARSim simply provides a well-defined interface to communicate with the robots and it is the researcher’s responsibility to appropriately use the interface to achieve desired results. Researchers do not have to write a controller from scratch, however, since several open source controllers may be freely downloaded. These include the community developed Mobility Open Architecture Simulation and Tools (MOAST) controller [13], the player middle-ware [1], and any of the winning controllers from previous RoboCup competitions. Winning controllers, from RoboCup 2006, may be found on the robocuprescue wiki [14]. A description of the winning algorithms may be found in [15].
III. SLAM ALGORITHMS

In this section we briefly describe the three algorithms that were compared. We selected three of the seven packages currently available on the OpenSlam website. The selection was mainly driven by the desire to compare algorithms with similar characteristics in terms of requested input data, rather than by the desire to perform a comprehensive comparison.

A. GMapping

The GMapping algorithm produces a grid map and takes a particle filter approach [16] [17]. In particular, each particle is associated with a possible map. The main challenge for the algorithm, therefore, is to reduce the number of particles, because of the significant overhead associated with each particle. The algorithm uses a so-called Rao-Blackwellizzed filter and its major contribution is in the definition proposal distributions and resampling techniques that allow one to decrease the number of particles without incurring in problems related to undersampling.

The GMapping implementation available on OpenSlam is coded in C++ and processes data logs encoded using the Carmen log format [18]. Basically, the algorithm requires time-stamped odometry (pose, translational and rotational velocities) and time-stamped readings from the SICK laser.

B. GridSlam

GridSlam also uses a Rao-Blackwellizzed filter, and is particularly aimed to mapping environments with loops, a problem known to be challenging [19]. GridSlam also tries to decrease the number of particles used in the filter, but takes a different approach from GMapping. In GridSlam, a model of the residual error from scan registration is learned on the fly and used to contract the number of particles.

The GridSlam implementation is also coded in C++ and, similarly to GMapping, processes data provided in the Carmen log format.

C. DPSlam

DPSlam also uses a particle filter to estimate the robot’s pose and maps [20]. However, unlike the former approaches that aim to carry along a restricted set of candidate maps, DPSlam exploits a peculiar map representation that allows one to track a huge set of candidates (thousands, according to the authors).

DPSlam is also implemented in C++ and requires the same data as the former algorithms (i.e. odometry and laser scans) although not encoded in the Carmen log format.

IV. EXPERIMENTAL SETUP AND RESULTS

The experimental setup aims to not only compare different SLAM algorithms, but also assess the fidelity of the simulation engine. In fact, if we are able to show that results extracted in the simulation environment can be safely extrapolated to real world scenarios, we have then installed a very powerful tool to generate a massive amount of test data with minimal effort. The methodology developed to conduct this twofold evaluation will be described shortly. For real world validation, we use a P3AT platform. The robot is equipped with odometry sensors and a SICK PLS range finder. A wireless-capable laptop is mounted on the robot, and the robot is controlled using the Player middleware [1]. The robot used in simulation is the corresponding P3AT model available in USARSim. The simulated robot is also controlled using Player. Data collection for the real robot took place in the hallway of the School of Engineering of UC Merced. For the simulated experiments, we developed a model of the same building, using the original blue prints provided by the architects. Figure 3 shows matching screenshots, in simulation and the real world, of the robots collecting data.

In order to keep a close alignment between the real world and the simulation, we avoided entering offices or research
labs, especially for experiments aiming at assessing the simulation accuracy. The reason for this constraint stem from the lack of detailed footprints of lab and office furniture. Data collection was performed in parallel. A control application gets input from a user via a joystick and then sends the same commands, i.e. rotation and translation speeds, to the two robots. The user does not directly see any of the robots, but rather controls them by observing the output coming from the SICK sensor equipped on the real robot. The described approach requires careful tuning of the robot model in simulation in order to match the performance of the real platform.

We preliminary compared the performance of the various algorithms while processing data coming from the real robot and from the simulation. Figures 4 and 5 show the output of the GMapping algorithm for data collected by the real robot and the simulator, respectively. Figures 6 and 7 show the same results for the GridSlam algorithm. Finally, figures 8 and 9 show the same results produced by the DPSlam algorithm.

It is important to stress that, in this set of tests, we did not strive to find the best fine tuning for the algorithms, but rather to assess the similarity between results produced in simulation and with the real robot. A few observations can be made from the figures:

- There is a good correspondence between maps produced in simulation and in reality by the GMapping algorithm. In both cases the produced map and the tracked path basically agree with ground truth.
- For the considered dataset, the DPSlam algorithm seems to fail both for real and simulated data. This does not mean that DPSlam is not working properly in general, but rather that both specific set of simulated and real
data exhibit some characteristic hard to deal with for this algorithm. Successive runs show good performance.

- GridSlam algorithm exhibits an intermediate behavior. The map produced with real world data shows an inconsistency on the lower left corner and an incorrectly-handled opening in the horizontal corridors (the opening is much wider in figure 6). The map produced with data coming from the simulator does not show these problems, though there is a problem with the vertical corridor since it appears to be wider than it is in reality (its width should be the same as the horizontal corridor).

The above results illustrate that there is a reasonable correspondence between results produced with simulated and real world data. Such correspondence is fairly strong for GMapping and DPSlam (in terms of success or failure) but less evident for GridSlam.

The next set of tests aims to measure the robustness of the three algorithms with respect to signal noise. Previously illustrated runs were obtained under the following simulated conditions. Readings from the SICK laser were affected by an additive noise with intensity 0.1% while the odometry was affected by Gaussian noise with 0 mean and 0.1 covariance. This means that every value $d_i$ returned by the sensor is altered accordingly to the following formula

$$d_i' = d_i(1 + 0.05x)$$

where $x$ is a random variable with uniform distribution over the interval $[-1, 1]$. Produced maps are illustrated in figures 10, 11 and 12.

- It is understood that Gaussian noise is highly suboptimal when it comes to reproduce data coming from odometry, a fact that needs to be better addressed within the USARSim framework. We then executed the three SLAM algorithms on a data set produced by a simulated robot whose SICK laser was affected by an additive noise of intensity 5%. This means that every value $d_i$ returned by the sensor is altered accordingly to the following formula

$$d_i' = d_i(1 + 0.05x')$$

where $x'$ is a random variable with uniform distribution over the interval $[-1, 1]$. Produced maps are illustrated in figures 13, 14 and 15.

The final set of tests was produced in a similar setting, with an additive noise of 10%. Resulting maps are illustrated in figures 16, 17 and 18.
V. CONCLUSIONS

Some general conclusions can be drawn. First, the use of the USARSim framework in order to compare different SLAM algorithms appears appropriate. Indeed, results obtained in simulation nicely translates to real robots. The performance of the simulated SICK laser is comparable to the real one, and an additive noise of 0.1% seems a reasonable choice. It appears necessary to develop a better model for odometry noise in order to accommodate the incremental nature of this disturbance. The three algorithms seem to be equally and reasonably robust to noise in the SICK laser. A level of 5% noise that visually appears much larger than anything observed in real world systems can still be dealt with by the algorithms. Both GMapping and DPSlam suffer from a lack of accuracy when noise is increased to 10%, but this is a level that is hardly ever observed in reality.

Comparing the different algorithms, GMapping emerged to be the more stable one in terms of performance, seldom221vision, related to image similarity, could be useful but have incuring in severe problems of map consistency. Along a different line for comparisons, DPSlam required the most time when processing batch data logs offline and, as acknowledged by the authors, is very demanding in terms of memory. It is important to note that we have not fine-tuned the algorithms, resulting in a possibly unsatisfactory set of parameters. Different sets of parameters could have easily produced different results. However, if the robotics community aims to a wide sharing of open source algorithms, the availability of easy to use and tune algorithms is a must.

An aspect that still seems to be underconsidered is the availability of well defined metrics for mapping algorithms to, for example, quantitatively measure the correlation between a produced map and the corresponding ground truth. Visual inspection still seems to be the most widely used means of performing such evaluation, but a more rigorous approach is needed. Tools coming from the field of computer
yet to enjoy significant popularity.

REFERENCES

**Reliability Testing for Embodied Autonomous Systems**

L. F. Gunderson and J. P. Gunderson

---

**Abstract**— Autonomous systems are intended to function effectively in dynamic and uncertain environments. Unfortunately, traditional testing methodologies are grounded in the assumptions of static environments and deterministic analysis. As a result, these methodologies fail to test the very characteristics that are critical for the deployment of autonomous systems. At the same time, there is more and more demand to provide quantitative benchmarks for autonomous systems.

New agile development methodologies, such as eXtreme programming (XP), are becoming standard. XP, in particular, relies on constant automated testing to provide both the developers and the customers with some level of confidence that the systems are performing correctly and reliably. However, XP is not designed to handle the additional challenges posed by embedded systems. We provide an augmented XP methodology that is specifically designed to address the issues associated with the automated testing of embodied (i.e., robotic) autonomous systems in dynamic environments.

In this paper, we present the results of over four years of designing and developing tests for embodied autonomous systems. We use a brief case study to provide examples of several key pitfalls, and provide a high-level overview of the requirements that must be met to test these systems.

---

**I. eXtreme Programming**

In the last decade many methodologies have been proposed to improve the quality and reduce the cost of developing software. Among these tools, eXtreme Programming is becoming widely accepted as a methodology that supports the development of highly reliable software. For an overview see[2]. This is due, in part, to four specific practices: continuous testing, continuous code review (pair programming), continuous integration, and pragmatic coding standards. Most of the practices required by XP are the same whether they are applied to a pure software development project, or to an embodied system. However, the introduction of co-development of hardware, firmware, and software requires significant adjustment of some traditional XP practices. In specific, the application of continuous integration, and continuous testing require some re-evaluation in the hardware/firmware domain. According to the National Institutes of Standards and technology (NIST), inadequate testing infrastructures result in tens of billions of dollars in additional costs, and significant over-runs in development time[12]. As mentioned above, one of the core practices of XP is continuous integration and testing. XP relies on the use of automated test tools such as the xUnit family (JUnit for Java, CppUnit for C++, etc.) of programs. These tools allow the developer to produce detailed unit tests (tests at the function or method level) which can be automatically run, and the results evaluated, summarized, and displayed to the developer. It is not uncommon to have several hundred test cases, covering every method and execution path, run in a few seconds at the press of a button. This allows the developers to make significant changes to the code (re-factoring) while remaining confident that no unanticipated side effects have broken the code base. This capability also supports the nearly continuous integration of the new code into the build process.

While these techniques have been well developed for traditional software projects, their integration into embodied systems lags behind[15] [9][7][4][3]. In the current model of hardware/software co-development, this has frequently resulted in software being developed well in advance of the hardware that it is designed to control. As a result, the high-level software is often tested in isolation, and integration testing becomes a significant factor in total development cost and time to deployment. It would be extremely powerful to have the benefits of continuous testing when...
dealing with hardware and firmware, but in practice, the notion of unit testing needs to be extended to handle the new domains.

II. METHODOLOGY FOR TESTING EMBODIED SYSTEMS

In this section we discuss a methodology that we have developed for testing embedded systems. In the interests of clarity, the following definitions are used in this paper:

- **Software** is the high level code, for example the planning and reification system.
- **Firmware** is the lower level code that controls the hardware.
- **The interface** allows the passing of information and control between firmware and software modules.
- **The system** is the combination of the software, the firmware, the interfaces, and the hardware.
- **A scenario** is a preplanned sequence of inputs, for which the preferred output is known.
- **A simulation** is an unplanned or random sequence of inputs for which the preferred output is only known in a probabilistic way.

In traditional software deployment, every effort is made to abstract away from the specific hardware underlying the virtual machine. This is possible because, for the most part, the hardware is completely substitutable. In fact, the Apple/Windows/Linux wars can be considered an indirect result of a lack of substitutability. In embodied systems, the hardware is an integral part of the system, and the substitutability assumption can not be made. This lack of a (more or less) uniform platform causes the developers to be responsible for the production and testing of not only high level code but also lower level interface and firmware control code as shown in Figure 1.

The fact that the developers may be responsible all of these components leads to a complex process of co-development. The high level software must interact with the firmware, which must be written to interact with the high level software. This tight coupling significantly increases the chances of software failure. The potential of software failure can be mitigated by automated testing, but this requires testing both the high level software and the firmware[1]. We have developed a different way of thinking about testing in embodied systems.

The first improvement is the splitting of testing into static testing and dynamic testing. For software only systems, this would be the difference between regression testing (static) and full simulation testing (dynamic). The second improvement is a split between the testing of hardware, firmware, software, interfaces, and system as a whole. This framework is shown in the table below.

![Table 1: Testing Breakout](image)

The first test that happens is bench testing, when the individual components are tested for input and output parameters, often on the robotic chassis. Once the behavior of the component is known, the component can be “mocked up” in software (see Rainsberger[14] for more details on Mock objects). This allows the programmer to create and run regression tests on that component in a software environment such as jUnit.

The next set of static testing will be done on the higher level software (but as a good XP programmer, you already have a set of regression tests on your high level software). Once the firmware and software have been tested, then the interfaces between them can be tested. So far so good, but thus far only the behavior in a static environment has been tested, now, we need to test the system in the presence of
chaos, dropped messages, moving targets, moving obstacles, and other hazards.

Dynamic testing comes in four different flavors. For the purpose of this paper, we assume that the high level planning software has been run through simulation testing. However, in order to make the firmware and interface testing fit the XP framework, the tests take the form of pre-scribed scenarios, for which the appropriate behavior is known. This allows the programmer to have the same confidence in the system’s dynamic behavior that the regression tests give in the static behavior. It also would be a good idea to write scenario tests for the high level planner in addition to the probabilistic simulation tests. Last, but not least, on to system tests, where the robot is taken out and allowed perform in the real world. This is a much less frustrating experience, since at this point you have some confidence that the software is working correctly.

So, back to the example in the introduction, what went wrong? The bench test was probably done correctly as was a static test of the high level software. The system failed at the system test. Without the static and dynamic test of the firmware and interface it will take a significant amount of time to discover the mismatch between the output of a GPS board and the input of the high level software. By partitioning the tests, this error can be detected before the expensive field testing.

**A. Benefits of Partitioning the Tests**

So, what lift do we get from this partitioning? Ever since the introduction of functional decomposition, it has been clear that neatly decomposed software is easier to develop. One key reason for this is that software that has been decomposed into (relatively) independent modules has lower levels of coupling.

In this paper we are presenting a methodology for decomposing tests into roughly independent classes, and addressing cross product terms explicitly. This is driven by the need to effectively cover a large, complex space of test cases, in a domain that is not easy to exhaustively cover. From the test decomposition shown in Table 1, we can extract six classes of ‘software only’ tests. High level software has been extensively researched with respect to testing, and it is not uncommon to have two to three test cases for every function or method in the code. While the automated testing of firmware is not as well researched, it is expected that approximately the same test-to-function ratio will hold. Without any type of decomposition, the number of tests needed to cover the combination of software, firmware, and interfaces would be:

\[
\text{TestCount} = \text{Test}_f \times \text{Test}_i
\]

Where: Test\(_f\) is the number of firmware tests, and Test\(_i\) is the number of high-level software tests.

This results from the need to test each function with each combination of tests for the other classes. This is significant since it is not uncommon to encounter test libraries with thousands of individual automated tests. This can easily result in millions of test cases to assure coverage.

It is, of course, possible to reduce the rather large number of test cases by effectively partitioning the space. If partitioned perfectly, the total number of test cases reduces to the sum of the test cases in each class, a significant reduction. It is rarely possible to partition the test space perfectly, since there are interactions between the various software components.

As a result, we test the firmware and hardware classes as though they were partitioned perfectly, and then test the interface between the layers as a separate class of tests. The key condition for the interface testing is the ability to rely on the successful testing of the components on either side of the interface. In effect, given that the firmware is correct, and that the software is correct, it becomes possible to test the interface as an independent class of tests. Given the validity of this assumption, the total number of tests becomes:

\[
\text{TestCount} = \text{Test}_f + \text{Test}_i + \text{Test}_o
\]

Where: Test\(_f\) is the number of firmware tests, Test\(_i\) is the number of interface tests, and Test\(_o\) is the number of high-level software tests.

For the case where we have approximately 700 firmware tests, 1100 high-level software tests, and 200 interface tests; we go from 770,000 un-partitioned tests to 2000 partitioned tests.

**III. GENERAL TESTING GUIDELINES**

As mentioned above, the structural decomposition is into three layers: high-level software, firmware, and hardware. Each of these has characteristics that affect the way in which testing is done. In order to reduce the costs and delays associated with systems integration, we have developed several guidelines for the design of automated tests for the firmware layers in embodied systems.

This code is written in a requirements driven development environment. Practitioners of test driven development will notice that the following code guidelines will violate some of their rules. We have found that for embedded systems the constraints imposed by the combination of the requirements and the physical environment are limiting enough that the addition of test driven development may make the coding task impossible. So, the following are our “in house” testing rules.

- Some things cannot be tested
- Do NOT change the code to make testing easier
- Use public accessors for setting and testing variables
- Test physical constants
- Mock hardware components and interfaces. Note: this often requires duplicating hardware performance in testable software modules
- The static tests are unit tests, test unit level functions.

**A. General Partitioning Guidelines**

Another possible source of confusion is the line between static and dynamic testing, so here are our rules for that
partitioning. Static unit tests are traditional automated tests, in which the required behavior of a method or function is verified. These tests typically involve setting up a test harness for a method, invoking the methods with a specific input set, and confirming that the return data are the expected value(s). These tests are the staple of all unit testing frameworks. Dynamic unit tests are used to test the performance of a module under changing conditions. These tests are critical for embodied systems, such as autonomous robots, since they will almost always be deployed into dynamic and uncertain domains. Dynamic testing is done using a scenario generator. This software is built out of traditional unit testing components, which have been extended to allow the specification of a sequence of tests, which correspond to a requirement of dynamic behavior. For example, consider an autonomous ground vehicle that is expected to detect a dead-end, and avoid getting ‘trapped.’ This required behavior can best be tested by creating a scenario which consists of mocking the sensory data that would correspond to the robot entering the dead-end and detecting the blockage. This scenario would have an expected behavior that corresponds to reversing course, extricating the robot from the dead-end, and proceeding in a manner that would avoid the dead-end.

IV. CASE STUDY

A. Hardware

As a case study, we will use an autonomous ground vehicle that is currently under development.

In Figure 2, the autonomous ground vehicle (Kitty) is shown undergoing unconstrained environment testing at a recent robotics exposition. (Please note that the robot was moving towards the people when the photo was taken) While there is an emergency stop control system, during these tests the E-Stop was unnecessary.

Kitty has an architecture that is biologically inspired. Based on research into neurophysiology, the core structure uses a model based on the brainstem and functional units of simple terrestrial vertebrate nervous system.

In Figure 3, the UML deployment diagram for the brainstem of the vehicle is shown. Each of the major nodes (the cubes in the figure) is an independent processor module. These are real-time modules which are programmed in a subset of the Java Programming language, and which emulate the parallel processing capability of living systems. For the entire system for to perform reliably, each of these modules and their components must function correctly and the interactions between each module must perform correctly. The XP methodology relies on continuous testing to assure the developers that any new code that has been integrated is performing correctly, and to assure that the most recent changes have not caused any previous software to fail.
B. Static Tests

1) Hardware Bench tests: The first stage of testing is making sure that the hardware is doing what it is supposed to be doing. Without this foundation, it is extremely difficult to debug software failures. In traditional software-only development, this step is typically skipped, since the underlying hardware is assumed to be tested and functional. If the development is being embodied into a commercial, off-the-shelf robotic platform, this step is often provided by the platform manufacturer. However, if the embodied system is a custom, or in-house developed, platform some level of assurance must be provided that the hardware is functioning correctly. The second benefit of this bench testing is that it provides the ‘gold-standard’ data needed to successfully build software mock objects to use in the firmware testing.

2) Static Firmware Tests: The firmware acts as an abstraction layer between the hardware and the high-level software. In the past, firmware was frequently developed in hardware specific application languages, which generated obstacles to automated testing. Recently, there has been growing acceptance of higher level languages as the primary development tools for embedded micro-controllers, and low-level hardware interfaces. The use of a high-level language such as Java or C++ allows significant flexibility in testing firmware.

For example, Kitty’s brainstem is running on five independent Parallax Javelin[13] chips, microcontrollers which are programmed in a subset of the Java programming language. These chips provide Java wrappers for hardware components such as Universal Asynchronous Receiver/Transmitters (UARTs), timers, and motor control systems. When the embedded code is running on the actual chip, these hardware components are instantiated, and connected to the Java code.

During static testing, it is necessary to mock these components, enabling the embedded code to compile and execute. By building the mock components, we have the ability to verify the function of the firmware, without having to run the actual hardware[6]. This means that automated static test libraries can be developed and archived. They can then provide full regression tests at the firmware level.

As an example we have mocked the UART wrapper used by the Javelin chip. A UART is used to provide serial communications between independent devices. The firmware can instantiate a UART and assign it to any input/output (I/O) pin on the chip. The mocked UART has the ability to accept data from the firmware and store it into a transmission buffer, which is visible to the verification tests. This allows the static test to automatically verify that the order in which the data are transmitted is correct, and that each datum has the correct value. If, during development, a change is made that results in a software error affecting this information packet (perhaps the order of data is altered, or a data item is skipped) the automated tests will detect and flag the error.

It is important to note that the mocked hardware components should only have enough fidelity to enable testing the firmware. The focus is not on testing the hardware itself, that step was completed by the bench testing in the previous step.

The question under consideration is: “If the hardware is working correctly, will the current firmware code meet the requirements?”

3) Static Software Tests: The static software tests are traditional unit tests. Many of the resources previously cited provide coverage of the process of testing traditional software. However, it is important to note that by previously testing the firmware, the testing of the high-level software can be separated from the testing of the underlying firmware. As mentioned above, this partitioning significantly reduces both the number of tests needed to cover the high-level software, and reduces the complexity of those tests.

4) Static Interface Tests: The final leg of the static testing is to test the interfaces between the different components. It is entirely possible for two software components to each pass all their individual tests, yet fail to function as a complete system. In embodied systems this can become a major problem, since there are typically numerous independent microcontrollers, processors, and discrete
hardware components that must all interact to meet the system’s requirements.

As an example, consider the navigation and communications modules as shown in Figure 3. The individual tests on the firmware code running on each of these modules shows that the navigation board is correctly reading and packaging the GPS and compass data, and loading that data into the UART for transmission. In the same manner, the static tests on the communications board show that it is correctly receiving the data packet, and unpacking it into local storage for dissemination. However, even though all the tests are passing, the system fails in the field tests. This comes about because the navigation board and the communications board (developed by different teams) have an incompatibility. This was the cause of the failure of the Mars Climate Orbiter[11]. The use of automated interface testing can detect and flag these types of errors. In the case of the interface between the communication board and the navigation board, the tests would instantiate a copy of the interface according to the communication board and the navigation board, and then verify that the correct data is stored on the communications board.

This could be caught during the test writing process as a developer writes:

```
AssertEquals(COMM.HeightInFeet(),
             NAV.HeightInMeters());
```

In addition, during dynamic testing, the behavioral requirements would detect and flag the error.

C. Dynamic tests

The static tests on the firmware focused on the fixed behavior of the methods and functions. The dynamic tests address issues that occur during continued operations. These tests include verifying correct behavior in cases such as buffer over-runs, loss of communication, updating rapidly changing data, and run-time performance. These types of tests require extending the model of unit tests. In keeping with the notion of building firm foundations, dynamic tests of the firmware are the logical starting point.

1) Dynamic Firmware Tests: In the static testing of the firmware, it was necessary to create mock objects that corresponded to the structure of the hardware components that the firmware utilized. These objects must be extended to provide some of the dynamic behavior that the hardware objects display. As an example, in the case of the UART provided by the Javelin chip, there is a fixed size transmit/receive buffer. During static testing it was sufficient to mock the capacity of the buffer, however to support testing of buffer over-run behavior, it is necessary to mock a certain amount of the run-time behavior. The description of this run-time behavior may be described in the technical documentation of the hardware, or it may be necessary to discover the behavior by running additional bench tests. However, not knowing how the hardware performs under dynamic load means that the developers have a large class of errors that will only occur at run-time. A case in point was the behavior of the Pathfinder Mars Rover, which underwent system level resets (unfortunately, on Mars[5]) when a hardware priority inversion fault occurred.

The Dynamic firmware tests for the Navigation board include building a scenario which will continuously generate and export compass heading data, to confirm that the firmware does not get overwhelmed. Additional tests include updating the mocked data registers on the GPS module during the read by the firmware to verify that the data are consistent. In addition, general performance tests are run on the firmware loops, to verify that requirements of update rates are being met.

2) Dynamic Software Tests: The Dynamic software tests are a natural extension of the firmware tests. Since the hardware has been mocked to provide realistic dynamic behavior, and the firmware has been tested to verify that its dynamic behavior meets the requirements of the system, it is possible to develop similar tests of the dynamic behavior of the high-level software. In many embodied systems there are additional requirements for the high level software. These may include adaptive behaviors, learning, and autonomous behavior. These are aspects of the high-level software that require dynamic testing of a slightly different nature than the lower level firmware. For example, if the system is supposed to function autonomously, it may not do the same thing, in the same way, time after time. This may require dynamic tests which setup the same situation time after time, and record the behavior of the embodied system. No single test can necessarily answer the question “Is the system performing correctly?” Rather, it may require statistical analysis of the aggregated result before the automated test can be verified. If the system is supposed to learn from its actions in the environment, then we compound this verification problem. It is necessary to run a series of aggregate tests to establish the current behavior, allow the system to experience failures and learn from them, and then rerun the aggregate test to establish that the learning has occurred. Now the test harnesses must provide a complex and dynamic set of test cases, and measure the change in aggregate response distribution.

While it is certain that designing, developing, and, ironically, testing these test scenarios is a major undertaking; it is clear that attempting to do this in the field is far more time consuming, and physically challenging. (Consider setting up a physical test to introduce reliable, repeatable noise into a sonar sensor, versus pumping “noise” into the software mock of the sonar sensor. See the recent report by Tse et. al. [16] for an example in the manufacturing domain.)
3) Dynamic Interface Tests: The dynamic interface tests are extensions of the previous tests. They are more complex since they are testing the interactions between multiple components in the system. But the same types of tests that were run to verify the dynamic behavior of the individual components can be extended with the static interface tests to establish correct dynamic behavior.

D. System Level testing

Systems level testing is actually composed of two phases. The first is a general systems test which is run every time the robot turned on. The second is more interactive formal systems test protocol.

1) The Robot dance (systems self-test): The first phase is the something we call "the robot dance," which is run every time that the system is started. In the "robot dance" the hardware is self-tested by the machine and the success or failure of that hardware is announced in a way that the human can understand.

So Kitty, on startup, uses a voice output to say her name and software revision numbers, turns the steering to the left and right, activates the sonar sensors and announces the results, runs backwards and forwards, activates the compass and announces the result, and activates the GPS and announces the results. The steering and drive train tests explain the "robot dance" label. This sequence becomes a hardware regression test, although the complexity of the test is limited by the need for a human to observe and interpret the results. As we add new hardware, new tests are added to the dance. This test perhaps the most tedious of the set, because it happens just as you are ready to test all the new cool behaviors that have just been added and tested since you had the robot out last. However, this test is also essential. If you have completed all of the tests, you have an assurance that any anomalous results you may see in the final test sequence are the result of unexpected interactions and not testable software or hardware bugs.

2) Formal Systems tests: Finally, we come to the last tests. At this point, you are probably saying "Enough already, we've tested this to oblivion, what else could go wrong?" What we have not yet built is trust. This is an autonomous machine that we are proposing to send into real environments, we need to learn to trust that it will behave "well" even in situations for which we have not programmed it explicitly (See Gunderson and Gunderson [8]). This final testing takes the longest, but is also the most fun. However, if all of the previous testing has been done, we can be assured that what we are testing is the intended behaviors and not an artifact of errors in the software or hardware. That is a way to build trust in the system that you are about to send out into the real world.

V. CONCLUSIONS

The deployment of embodied systems such as robots into ‘real world’ environments is risky. Once a robot is removed from the tightly controlled environment of a laboratory or a factory floor, the range of situations and events that the system must handle increases dramatically. The experience of successfully ‘testing’ the robot in the lab, or in a controlled test environment, only to have it fail (sometimes dramatically) in the field is the norm. This unpleasantness has been experienced by teams from universities, commercial labs, and national governments. These failures can be avoided by incorporating a more formal test methodology, which takes advantage of both the recent advances in testing software, and new development methodologies such as agile development and XP. We present an extension of the XP methodology that has been designed to assist developers in producing reliable, well tested embodied systems.

The methodology is based on three core concepts:

1. Partitioning the space of tests into nearly independent classes;
2. Adding a separate class of tests for the cross product terms that result from the partitioning; and,
3. Extending the testing methodology to include both (traditional) static tests and dynamic tests.

These expanded tests make extensive use of the concept of ‘mocking,’ producing lightweight software components to replace either hardware components or heavyweight components that interfere with testing.

The result of applying these techniques enables the developers to test, in software, aspects of embodied systems that previously were tested in the field. This results in far more tests being run, and the tests being run far more frequently. This combination results is a much higher level of confidence in the ability of the embodied system to meet its design requirements when it is deployed.

REFERENCES

[9] H. Koehnemann and T. Lindquist, “Towards target-level testing and debugging tools for embedded software”. In Proceedings of the...


Advances in the Framework for
Automatic Evaluation of Obstacle Avoidance Methods

J.L. Jimenez  I. Raño
Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
irano@unizar.es

J. Minguez
I3A, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
jminguez@unizar.es

Abstract—This paper will describes the advance in our project for benchmarking obstacle avoidance techniques for mobile robots. The core of the project is to create a methodology/software to evaluate the performance of the methods given a wide range of work conditions. These work conditions usually include scenarios with very different nature (dense, complex, cluttered, etc). The performance is measured in terms of robotic parameters (robustness, optimality, safety, etc). In the paper we will give an overview of the project and we will focus on the project analysis from a software engineering point of view. At this state the software design decisions are critical and could impede a proper later development, therefore we have developed a great effort in the analysis and design of the project.

I. INTRODUCTION

There is currently great effort in the robotic community to find standards as a way to measure the quality of a wide range of technologies. Good examples are the RosTa EU project, the special group of interest raised by EURON and some other launched by the NIST. Many of these efforts are focused on creating benchmarks for a given area. Our project is concerned with the standarization but from the automatic evaluation point of view. Instead of working on benchmarks, we address an automatic evaluation system is being constructed for obstacle avoidance algorithms. On the one hand, benchmarking just will try to match the results of an algorithm with some expected output, a desired algorithm result. On the other hand, our automatic evaluation framework will provide the needed results to perform the benchmarking, but also will generate a comparison performance of the obstacle avoidance algorithms in very different situations.

In this direction, we have been working in the development of a software tool evaluate the obstacle avoidance algorithms in the context of service robotics. The objective of this paper is to present the first steps: the analysis and design of the problem from a software engineering perspective. The project is expected to grow up, hence this software engineering perspective is very important at this stage. Since this is currently an open research field, the application is expected to change as new ways of evaluation appear. In order to design a useful application in the future, including those desirable characteristics, it is important to follow good software engineering practices. An early implementation has been done and some tests have been performed as the final steps of the analysis of requirements and design of modules231modularity, flexibility and extensibility are mandatory as

and system. The rest of the paper is organised as follows: Section II introduces the evaluation system global analysis and its requirements. In Sections III, IV and V the analysis and design of the identified independent applications is presented. Some details of our early implementation and preliminary results are depicted in Section VI, while Section VII depicts some conclusions and next steps in our project on automatic evaluation system for obstacle avoidance.

II. OVERVIEW OF THE EVALUATION SYSTEM FRAMEWORK

In this section we will present an overview of the whole evaluation system framework to analyse and identify the possible modules and its requirements. Some aspects of the evaluation system has been introduced in [2]. The idea behind the application is to evaluate obstacle avoidance methods (measuring quantitative parameters of the solutions) given a wide range of working conditions (different scenarios). As can be seen from Figure 1 three independent modules can be clearly identified. The colored blocks denoted Module 1, Module 2 and Module 3 correspond to the scenario generator and characteriser part, the robot simulator and trajectory descriptor and the final analysis of the results module. They jointly from the complete automatic evaluation system and can be implemented as separate applications even some interaction exists between them.

Figure 2 draws the interaction between the three modules of the automatic evaluation system as a data flow diagram representation. Instead of keeping direct data flows between the modules, the applications can use two data repositories, namely the scenario files (obtained from the scenario generator) and the execution result files (provided by the robot simulator module). The user will introduce to the system some parameters like the amount of scenarios to generate, the number of test to run, and so on. In view of Figures 1 and 2 the automatic evaluation has been decided to be implemented as three independent applications.

Following [5] the requirement analysis of a software system can be viewed as a set of functional requirements, what the system should do, and a set of non-functional requirements, like efficiency, portability, extensibility and so on. As a ongoing research project some points are not currently defined or are prone to change, therefore software
non-functional desirable features. On the other hand the functional requirements of the system have to be defined before the analysis stage. The rest of this section is devoted to present the modules and its functional requirements.

a) Module 1: represented in Figure 1 is the scenario generator and characteriser. Its general purpose is to generate simulated environments (randomly or following some criteria) and compute descriptors measuring interesting properties from an obstacle avoidance point of view. The descriptors will be used to classify the different sets of environments according to its nature (dense, complex, cluttered, etc). This is a key module on the system since the obstacle avoidance algorithms must be tested on all the possible situations a robot will face up. A high level analysis shows that its functionality must include:

- It must be able to characterise the scenarios using the defined descriptors and compute their values.
- It must generate scenarios, randomly (for a given number of scenarios) or such that they descriptors cover a given range with a minimum number of them. While in the first case the user must only introduce the number of scenarios to be generated, the former must ask for a number of bins and minimum scenario number in each bin.
- It must handle scenario files and files with scenario lists.

b) Module 2: will compute the trajectories of an obstacle avoidance algorithm for a set of scenarios. It also should provide some performance measurements of the trajectories referred to an, in some sense, optimal path. The functional requirements found for this module are:

- It should be able to dynamically load any obstacle avoidance algorithm that matches a given function prototype.
- It must simulate a generic robot over a set of scenarios and characterise the robot motion according to some trajectory descriptors.

c) Module 3: builds the evaluation results based on the outputs provided by the two other modules. Given an obstacle avoidance algorithm, for each scenario descriptor and trajectory descriptor this module will generate a result table with the algorithm performance behavior. This module can be seen as the most general part of the system, since it can be used for other purposes where scalar results from two different features need to be drawn. The minimal functional requirements of this application are:

- It must build tables by crossing scenario descriptors and trajectory descriptors.
- It must be able to open and to create result files for an obstacle avoidance algorithm benchmark in a predefined format.

III. SCENARIO GENERATION AND CHARACTERISATION

In this section we present the application for scenario generation and characterisation (Module 2) analysis and design, while implementation and testing will be treated in a separated section. We choose to follow the OMT (Object Modelling Technique) methodology [3] to build the analysis diagrams of each application in the system and some UML (Universal Modelling Language) tools has been also used to build those diagrams. However, since UML [4] has been derived from OMT and they have many common elements. The structure of the sections describing the individual applications is the same for all three parts of the whole evaluation framework (see also Sections IV and V). First the analysis section focuses only on the static relations of data, not treating the dynamic and functional relations, because its smaller significance in our case. Then the design section includes the application architecture grouping the analysed
A. Analysis of the Scenario Generator and Characteriser Application

The key concept in this application is the scenario for a full technical description of scenarios see [2]. They will be characterised according to its qualitative features and will be used to carry out the robot motion simulations. A concept with such an importance must be included as a class to collect the scenario functionality on the application, the Scenario class. The scenarios will have associated sets of characteristics, the descriptors, and encoding models, the way scenarios are implemented in the application, that need to be included in the static class view. Therefore a Descriptor virtual class needs to be created to reflect the scenario characterisation. The general class represents the functionality of any descriptor prone to be implemented in the system, and collects the scenario characterisation functional requirement. Moreover, as the Descriptor class is virtual the concrete defined descriptors will be derived from it. This also fulfils the extensibility non functional requirement above mentioned. Since the calculation of some descriptors can be computationally expensive in terms of time and memory we decided to use the Proxy software pattern [1]. In this way the computation of the descriptor is delayed until it is really necessary, and once the numerical value is get it will be stored for later use. Figure 3 shows part of the UML class diagram generated on the analysis stage of the application. As can be seen from figure, currently only three descriptors have been implemented as Density, Clearness and Confinement derived classes.

The right class box on Figure 3 is the scenario implementation ScenarioImp virtual class. Since the computation of some descriptors can be quite difficult for some scenario representations the ScenarioImp class has been designed to allow different scenario representation on the application. Even not drawn in the figure two classes are derived from this virtual class, the discrete DiscreteImp and continuous ContinuousImp scenario implementations. While the later stores the environment as a set of obstacles defined by its geometrical primitives, the former consists on a binary occupancy grid with configurable size. This allows to compute the descriptors with different resolutions and algorithms for a given scenario. The capability of having different implementations introduces the need for an implementation converter in order to pass from one implementation to another. A Converter virtual class has also been designed using the Strategy software pattern. This pattern defines and encapsulates a family of algorithms, in our case for implementation conversion. From the Converter class any implementation converter can be derived. Currently only a continuous to discrete implementation algorithm is available.

Another functional requirement of this application stated in Section II is the ability to generate scenarios in different ways. A fixed number of random scenarios, for instance, could be desired to perform test on some algorithms. In other cases a set of scenarios need to be generated to cover all the possible value ranges of a given descriptor. A virtual class Generator, not reflected on the Figure 3, has been defined to fulfil the corresponding functional requirement of the application. Since the generation of scenarios can be performed in different ways, the Strategy software pattern has also been selected to design the generator class. Any instance of a generation algorithm must be derived from the Generator class. Moreover, as there are several implementations of the scenario, the virtual class has a method called FixImplementation() to select the implementation kind to be generated.

Up to this point the analysis has only included individual scenarios, however the evaluation system has to perform test on all possible scenario conditions. Therefore a scenario list ScenarioList abstract class needs also to be used to jointly store sets of scenarios. Any other scenario list can be derived from it. On the other side, the lists usually need to be acceded in a sequential way and an iterator software pattern is necessary. Besides this access method for the scenarios on the lists, another functional requirement is the capability of scenario indexing according to the value of some descriptor. A class for a scenario list index has been modelled to serve as an interface for scenario lists allowing to access groups of scenarios with some given ranges of descriptor values.

An important aspect of the scenario generator is data persistence, because all the computed descriptors and its scenarios must persist when the application finishes. To generate and characterise scenarios can be a resource consuming task. To allow this data persistence it was necessary to create a scenario storage file system. Since the internal form in which data is stored is not a part of the analysis stage, but just to take into account the persistence needs, two abstract classes have a reader and a writer been implemented, with some appropriate derived classes to store the scenario lists in a preliminary format. This structure allows for a change in the internal storage way, fulfilling also the flexibility non functional requirement.

The final requirement is to allow the user generate and handle the scenarios and sets of them through a Graphical User Interface. It is common practice to change an application GUI, and therefore it is important to have a small number of classes involved, to have a uncoupled...
the application is modelled as a `GUIElement` class, and are grouped hierarchically. The composite software pattern has been used since it allows to treat visual elements in the same way either being compound or not. It is a good practice to separate the user interaction in two parts: the visual part and the functional part, such that if any needs to be changed the other can be kept. The functional part are implemented through the application commands that represent the simplest actions a user can perform. The Command software pattern proses the creation of command objects to encapsulate and parametrise actions.

**B. Design of the Scenario Generator and Characteriser Application**

The previous section has depicted the five main concepts related to the scenario generation and characterisation application with its corresponding classes. A set of classes (Scenario, Scenario Implementation, Generator...) and its derived subclasses are grouped around the scenario concept. There are other classes related to the scenario list; the Scenario List itself, the Iterator, Reader, Writer and Indexes. Some others providing Graphical User Interface functionality and finally the command related classes. All of the above can be joined into a new class representing the application itself. Figure 4 groups and relates the main modules to build application, an arrow starting in one module means that it depends on the target arrow module. As can be seen all the modules, except the application, use services of the Scenario module while this one does not use any service of the rest of the layers. This makes the scenario module quite critical in case any change should be done, because all the layers could need an adaptation to the new module interface. The Scenario I/O layer is used by the Commands and GUI modules making it the second most critical module to changes.

![Diagram](image)

**Fig. 4. Scenario Generator and Characteriser Application Architecture**

The Commands and GUI layers interact with each other, since the GUI uses Commands functionalities and introduces information parameters from the end user to the commands. Finally, the higher layer, the Application is available to the user for the lower level Commands and GUI layers access.

**IV. ROBOT SIMULATOR AND TRAJECTORY EVALUATION**

This section presents the analysis and design performed on the simulation application, the so called module 1 in234 while safety is computed through comparison with the safest

**A. Analysis of the Robot Simulator and Trajectory Evaluation**

The simulated robot model has a sensorial and motor part. The sensor of the robot is assumed to be a proximity range sensor providing distance measures in a 180° range around the robot front, one for each degree. On the other hand, the robot always move forward with limited speeds and accelerations. As presented in Figure 5 a robot class models all those robot characteristics. We reflect in this class diagram takes the fact that the motion control is provided by an external library. Through the MovementCalculator class we provide an interface for the implementation of the dynamic library loading process and function call. As can be seen in the figure two classes are derived from this basic one that must take into account the way each operating system loads the dynamic libraries. Using this class heritage to load obstacle avoidance algorithms the non functional requirement of a multi-platform application is accomplished.

![Diagram](image)

**Fig. 5. Class Diagram for the Robot Simulator and Trajectory Evaluation**

The trajectory evaluation is performed by the computation of some trajectory descriptors also represented in Figure 5. The motion simulator obtains as output the robot trajectory that must be compared to some defined optimal paths to evaluate the trajectory generated by the loaded obstacle avoidance algorithm. The robot class includes a trajectory class which can be evaluated through the trajectory descriptors, modelled as a virtual TrajectoryDescriptor class. Currently three such derived descriptors have been defined and implemented, success, optimality and safety. The success descriptor has a boolean value indicating if the target position has been reached following the motion commands. Optimality is a comparison with the optimal path obtained using the visibility graph from the start to the end position,
path, obtained from the Voronoi diagram. Finally, as for the scenario characterisation application the results need to be stored, therefore the robot class also includes a result writer abstract class that allows to store descriptors in a format that can be changed in the future without affecting the rest of the classes.

B. Design of the Robot Simulator and Trajectory Evaluation

The robot simulator and trajectory evaluation application design is presented in Figure 6. As can be seen some of the application layers are Scenario and Scenario I/O, defined for the scenario generator but also used here. The design reflects that the only way to access the hard disk storage is through the I/O modules. An operating system layer has been added since the obstacle avoidance library, the Motion Calculator module, is OS dependant. The Robot layer is related with the Scenario, from which simulation data is extracted, with the Trajectories and Characterisation layer, the one that computes trajectories descriptors.

\[\text{Fig. 6. Simulation Application Architecture}\]

Since two of the designed modules on this trajectory evaluation application are directly dependent on the Scenario layer, as for the scenario generator and characteriser the classes related with the scenario class may change if the scenario is changed. Therefore the scenario layer must have a stable interface with the rest of classes.

V. RESULT ANALYSIS APPLICATION

This is the simplest application on the current implementation of the automatic evaluation system. Actually this application could be used to perform data analysis of any system with similar features, since it only builds result tables by crossing descriptor ranges. Besides the capability of building result tables the only functional requirement is to store them into a given format.

A. Analysis of the Application for Data Analysis

The key concept in this application is the result interpreter, an element that for a given result sequence and a scenario file is able to statistically summarise results in an automatic way. Since there are multiple table formats and programs to handle them we decided to use in out preliminary implementation a plain text format. However, it can be interesting to build binding with some office applications for data analysis. The main classes include the ResultReader abstract class that forms a base for the result file reader, the reading functionality for results not included on Figure 5. When the reader finished the process of data loading it generates an element of the ResultInterpreter hierarchy which is responsible of statistically cross the data. Once again the Result interpreter class is an abstract one allowing different data interpretation classes to be defined in later researching steps, that is providing extensibility to the application.

B. Design of the Application for Data Analysis

This is the smallest application to design, only an interpreter for the results needs to be added as represented in Figure 7, while the Scenario and Scenario I/O layers are again used. The main application layer uses the functionality of the results interpretation and the Results I/O, while the statistical interpretation module writes its own result files, making it independent of the Results I/O layer.

\[\text{Fig. 7. Analysis Application Architecture}\]

VI. IMPLEMENTATION ISSUES AND APPLICATION TESTING

Besides all the functional and non-functional requirement for our applications we should choose the proper development tools. Since the obstacle avoidance algorithms will usually be provided as compiled dynamic libraries and the evaluation problem is run on an intensive test basis, an efficient programming language must be selected. Another interesting requirement is to make a multi-platform project, that, as stated before, must be highly modular. On the other hand, both application analysis and design have been done Object Oriented and therefore the implementation language should support objects. We choose C++ as our implementation language since it fulfils all out requirements, but mainly it provides modularity and produces efficient programs. We also used a set of highly standard tools like the Standard Template library (STL) and GTK+ for the application parts that need a Graphical User Interface.

An early implementation of the whole framework has been performed contain more than 11,000 lines of C++ code. The performed tests over the system has been done in two steps; a component test and integration test. The component test where performed over all the classes and modules, and the joint application was also tested once the parts were integrated. Most of the test were black-box testing, where different inputs were provided to the elements and its outputs were checked with the expected output.
was fully tested the integration was performed and again
tested in a bottom-up way.

A. Scenario Generation and Characterisation

Figure 8 shows a final view of the application Graphical
User Interface. The left part of the main window shows the
working scenario list, while the right part is split in the
current scenario display at the bottom and the descriptors
sub-window, where descriptor values are displayed. The sce-
narios can be generated in different ways, the non-parametric
way just generates a random number of scenarios on a
scenario list. The parametric generator allows the user to
define bins over one descriptor and select a minimum number
of scenarios in each bin. Of course all the generated scenarios
or lists can be saved to the hard disk with an appropriate
format using the application.

B. Robot Simulator and Trajectory Evaluation

For the simulation application the optimal path between a
starting and target positions needs to be obtained. The current
implementation state includes a \( A^* \) algorithm to compute the
optimal path as a sequence of discrete cells, therefore using
the discrete implementation of the scenarios. The Voronoi
diagram on the discrete scenario implementation has been
used to get the safest path. Both, paths are used to compute
optimality and safety trajectory descriptors, which are also
evaluated using the discrete implementation by performing
an integration over the grid cells. Since the amount of
simulations to perform is big and done as a batch process,
no graphic interface has been designed for this application.

C. Evaluation Results

For the final evaluation results, we implemented a software
able to connect the different descriptors of the scenarios with
the performance parameters of the methods. This is dis-
played in the form of tables of performance for visualization
and comparison in between methods. Figure ?? shows the236
performance of one technique selected ???. In this case, one
can see the performance and evolution of the performance
descriptors as a function of the different scenarios measured
by density, clearness and confinement for example. In fact
this type of tables are a great help of researchers, engineers
and developers in order to asses their results and to search
for possible techniques to work in a given range conditions.

VII. CONCLUSIONS AND FURTHER WORK

This paper present the current state of the automatic
evaluation software for obstacle avoidance algorithms. As an
ongoing research project some aspects of the software frame-
work development could change. The initial idea is to create
an open source software system. A great effort has been
performed from a software engineering perspective to start
building an extensible and flexible framework in three mostly
independent modules. Extensibility and flexibility must be
key features on such a system. Some major aspects of the
analysis and design have been presented. Our framework
allows to include new descriptors to both, scenarios and
trajectories, that on the other hand need to be studied and
implemented to better evaluate obstacle avoidance mecha-
nisms.

The future steps include the definition and implementation
of new scenario descriptors, and an extensive scenario lists
generation as a test-bed for the algorithms. Even the system
has been developed to be multi-platform this feature has
not been tested yet, and maybe some small changes and
further development will be needed to actually have such
an application. A Developer’s Guide document and online
API documentation will be also created in order to help both
developers and users.

REFERENCES

[1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns:
Elements of Reusable Object-Oriented Software. Addison Wesley, 1998.
robot obstacle avoidance algorithms. In Workshop of Benchmarking
in Robotics, in the IEEE/RJS International Conference on Intelligent
Good Experimental Methodologies in Robotics: State of the Art and Perspectives

Fabio P. Bonsignorio, John Hallam, and Angel P. del Pobil, Members, IEEE

Abstract—As the complexity of developed robotic and intelligent systems grows, it is more and more needed to define proper experimental approaches and benchmarking procedures.

Trustable benchmarks are needed in order to allow the comparison of the many research results in service robotics research and enable their industrial application.

On the other end, if robotics aims to be serious science replication of experiments deserves serious attention.

It is necessary to be able to verify if and by which measure new procedures and algorithms proposed in research papers constitute a real advancement and can be used in new applications.

New more successful implementations of concepts already presented in literature, but not implemented with exhaustive experimental methodology, risk to be ignored, if appropriate benchmarking procedures, allowing to compare the actual practical results with reference to standard accepted procedures, are not in place.

Both replication and benchmarking are needed to foster a cumulative advancement of our knowledge of intelligent physical agents and even to correctly appreciate disruptive innovation in the science and technology of robots.

Should we take inspiration from biology and medicine?

In order to address these needs the European Union Network of Excellence on robotics EURON has funded a Special Interest Group on Good Experimental Methodology and benchmarking.

This paper summarize the state of the art in the field, the possible perspective activities identified so far, and the EURON GEM SIG challenges and ambitious plans.

1. INTRODUCTION

All science proceeds from experiment, which motivates the creation of new theory and establishes the limits and validity of the existing theoretical basis. Individual branches of science conduct experiments differently, depending on the topic of investigation, but all have in common a body of knowledge concerning experimental methodology that specifies how to design and conduct 'good' experiments in that discipline.

If robotics aims to be serious science, serious attention must be paid to experimental method. The research activities in the robotic fields are huge and it is huge the number of published papers.

In order to allow the exploitation of the many results obtained it is at least necessary to able to:
- validate the results by replicating them
- compare the results in term of the chosen performance criteria

Although some work is already carried on, a lot of open issues are still in front of us.

In section II is described the state of the art as regards the replication of experiments in robotics, in section III the situation in benchmarking is reviewed.

II. THE IMPORTANCE OF REPLICATION

Whether you see robotics as the science of intelligent physical agents ('embodied cognition') or as the branch of engineering that, through mechatronic integration, aims to build autonomous or semi-autonomous machines for many diverse tasks, it must be seen as a scientific quantitative discipline.

In any scientific discipline, from physics to engineering and medicine, the models of a system must be able to predict with a certain accuracy the evolution of the variables under study with a given input over time.

A 'good' model must give the same results in the same condition.

Results have to be 'replicable'.

An objection raised sometimes to this kind of considerations is that the robot which are developed are very different in kind, physical morphology, tasks, algorithms, sensors,
In particular, it is expected that an acceptable trial specify, with objectives, the data will be analyzed in relation to each of the primary and secondary hypotheses. The statistical section of the protocol is asked to define how experiment replication and a concept of 'falsification' of theory through experiments. There are different modulations of this concept [20,21,22,23,24], but whether we think we are in a cumulative phase in the development of a scientific field or in presence of a 'disruptive' creative paradigm shift [22], as somebody is claiming in nowadays robotics, a kind of widely accepted experimental methodology is needed in order to be able to ground the advancement of research on a shared quantitative language. On the other end, in different scientific fields like biology and medicine [17,18,19], there are well established experimental procedures to deal with the behaviors of complex systems, at present more complex, and less known, of those under development and study within the robotics community. They suggest that the huge diversity of the developed robots should not prevent us from implementing more reliable experimental procedures. A clinical trial protocol is the detailed written plan of a clinical experiment. It may be inspiring looking at the US NCI guidelines for drafting a clinical trial protocol, [18]: the emphasis on signaling 'adverse events', the definition of 'criteria for response assessment', the necessity of defining clearly principal and secondary hypotheses to be validated. The statistical section of the protocol is asked to define how the data will be analyzed in relation to each of the objectives. In particular it expect that an acceptable trial specify, with reference to the study objectives:

- Method of randomization and stratification
- Total sample size justified for adequate testing of primary and secondary hypotheses
- Error levels (alpha and beta)
- Differences to be detected for comparative studies
- Size of the confidence interval of the estimates.

It seems clear that in robotics the experimental methodology standards are currently in many cases weaker, and the syndrome 'it worked once, in my lab' could be more widespread than we may think.

As already noticed, a limit to replication is given by the huge variability of robot machine. Perhaps, following the biomedical analogy, we have to compare behaviors and performances of different 'animals'. Any way a wider capability to replicate research results is probably needed in order to allow a faster development of our field and to foster both cumulative progress and disruptive change.

III. BENCHMARKING IN ROBOTICS

Due to the huge diversity of robotic architectures and approaches and the intrinsic difficulty of measuring the performances of machines which aim to be flexible and autonomous, so far, there are benchmarks defined only for a limited subset of the tasks which can be performed by a robot [5].

There are benchmarks in machine vision and dependability of software systems which could probably be adapted to the robotic field.

There are some interesting conference tracks like NIST, IEEE and ACM Permis and the IARP, IEEE RAS series of workshop on Dependable Robots.

A. Conference tracks

The Permis workshop started in 2000 and in 2007 it reached the seventh edition. This workshops aim to define measures and methodologies for the evaluation of performance of intelligent systems.

The focus is to define quantitative metrics to be able to compare such things as: the level of autonomy, human-robot interaction, collaboration.

To provide methodologies to evaluate components of intelligent systems: sensing and perception, knowledge representation, world models, ontologies, planning and control, learning and adapting, reasoning.

Other topics are infrastructural support for performance evaluation, application specific performance measures.

In this context intelligence is defined as “the ability to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral goals” (J. Albus, “Outline for a Theory of Intelligence”, IEEE Trans. on Systems, Man, an Cybernetics, Vol. 21, No. 3, May/June 1991).

There are some connections between these workshop topics, those addressed in this paper and the scope of the Workshops on Technical Challenges for Dependable Robots in Human Environments co-sponsored by IARP and the IEEE Robotics and Automation Society. The 2005 workshop was co-sponsored by IARP, the EU network of excellence in robotics.

The concept of dependability requires that the capability of...
performing a set of task in a set of changing environments are performed with replicability with given measured performances.

Finally in both 2006 and 2007 IROS conference was organized a workshop on performance metrics of robots.

B. Performance Metrics
Performance metrics have been developed in various area of robotics for specific purposes. There are a number of initiatives devoted to define adequate performance metrics in specific subfields. Here below follows a non exhaustive list, whose main purpose is to exemplify the community attempts to cope with the benchmarking problems.

Radish, started in 2003, by Andrew Howard and Nick Roy is a repository of standard data sets with, currently, a main focus on localisation and mapping. The more common format CARMEN, the open source Carnegie Mellon Robot Navigation Toolkit, for mobile robots control that provides basic navigation primitives. At present it contains mostly logs of odometry, laser, sonar and other sensor data taken from real and simulated robots and environment maps created by robots or manually.

The Automatic Control Telelab (ACT) has been developed to support real-time configuration and observation of experiments in visual servoing remotely, as well as playback access to acquired data. Universitat Jaume I developed a visual servoing simulation environment called JaViSS. It is written in Java with graphical rendering. Manipulator kinematics code is based on the Robotics Toolbox for Matlab by Peter Corke. It is intended as a tool to simplify the testing and comparison of different visual servoing approaches. Another initiative at Universitat Jaume I offers a set of experiment data on visually-guided grasping of planar objects with a Barrett hand. This set of standard experiments allows the definition of benchmarks and associated performance metrics. This allows to compare different algorithms and implementation for visually-guided grasping.

The University of Parma is managing a repository storing benchmarks designed by different research groups and related documentation down to file format for motion planning. It stores data sets about robots, workspaces and benchmark problems, related to different kind of robots: mobile robots on the plane, free-flying robots, manipulators, sequence of manipulators, union of mobile robot and one or more manipulator sequences. An other shared repository of motion planning benchmarks, originated by Movie project, is Movie Models for Motion Planning maintained by Utrecht University.

It currently contains data sets related to 192 robots and objects and 75 scenes.

The RAWSEEDS project is an SSA (Specific Support Action) in the EU 6th Frame Program, providing a comprehensive, benchmarking toolkit for SLAM (Simultaneous Localization And Mapping). It will provide a web accessible repository storing standard data sets, based on different sensor sets, and related benchmarks, state-of-the-art solutions to SLAM problems in the form of algorithms, software, and methodologies for the validation of algorithms.

The NIST USAR (Urban Search And Rescue) 'after disaster' scenarios, ranked as yellow, orange and red, are used in RoboCup USAR. They provide an useful conventional reference scenario for USAR applications together with USARsim the open source simulation environment based on the Unreal Tournament gaming engine.

C. Research coordination activities
The RoSta project[4], started in 2007, is a two-year coordination action, within the EU 6th Framework Program. The objective of RoSta, which is linked also with standard related activities within IEEE, is to identify the action needed to start formal standards development and the establishment of de facto standards in service robotics. The short term aim is to select a few key topics, where standardization is already possible and whose expected impact is higher.

ALFUS (Autonomy Levels For Unmanned Systems) [8], is a similar US federal agencies ad hoc working group focused on unmanned system autonomy metrics. Its main objectives are to analyse the needs for autonomy metrics, the related methodologies and to develop standards.

D. Competitions and Challenges
RoboCup, [9] is probably the most famous competition in robotics. RoboCup is mostly focused on soccer game as a primary domain, and organizes the Robot World Cup Soccer Games and Conferences. Soccer is a very good testbed for multi (robot) agent technologies. New competitions in search and rescue, based on NIST scenarios, and home assistance have been added. A similar activity, born in 1997, is FIRA robot soccer league, a Korean KAIST initiative.

Eurobot is an international competition with chiefly educational purpose with rules renewed every year. The DARPA Grand Challenge is a famous competition for outdoor robot race on an about 200km circuit in the desert. The last edition was won by Stanford team, with a modified
version of a VW Tuaregh, and five teams were able to complete the race.

DARPA is now organizing the Urban Challenge where the robot has to cope with an urban traffic scenario.

An interesting cleaning robot competition was organized in 2002, in Lausanne, Switzerland jointly with IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS 2002). As an example the task of the floor cleaning section was to clean within 10 minutes 5x5m room covered with sugar.

The European Land-Robot Trial (ELROB), organized by the German Federal Armed Forces (Bundeswehr), is an outdoor robot demonstration with no real competitions or prizes, but otherwise similar to the DARPA Grand Challenge.

It focuses on mobility and RSTA (Reconnaissance, Surveillance, and Target acquisition). It took place in 2006 for the first time, in 2007 a civilian version was organized in Switzerland.

IV. DISCUSSION

Even from the limited survey above it is apparent that the bare replication of experiments and the quantitative comparison of research results in robotics raise many challenging issues.

This is due to the variety of applications, tasks, mechanical structures, sensor sets, actuators, control system, software architectures, required levels of flexibility and autonomy, and so on. When we are dealing we Human Robot Interaction in everiday settings also human psicology is involved.

On the other end, there are many initiative trying to define proper standards.

There are benchmarks in some specific areas like visual servoing, SLAM, motion planning, but there is still a lot of work to do.

Possibly we should identify a few limited and simpler tasks and related environments and develop benchmarks for those task that can be accepted and are by the community and then proceed extending the approach to more complex functions.

As told we should probably look to biology, medicine and 'soft' sciences for inspiration.

In [13] and [14], see fig. 1 and 2, from [14], and in other experimental works ‘entropy measures’ on the ‘sensory-motor’ coordination of different ‘robotics’ equipment have shown that information metrics can be used to classify, at least, and to get an insight on (semi) autonomous robotics devices, which show an 'emergent behavior', while, in [15], entropy measures are used to rank enviroment complexity, with reference to the navigation task, see fig. 3.

In [12] an approach integrating task and environment complexities is proposed.

HRI experimental research is sometime conducted by means of protocols deriving from psicology.

V. EURON GEM SIG

If we want to foster the further development of (service) robotics research and to enable the industrial exploitation of the many results already obtained, it is probably necessary to improve the common experimental prectices, looking at both replication of experiments and ojective performance evaluation.

In order to cope with these needs the EU EURON network
of excellence has funded a special interest group in replication and benchmarking with the objective of increasing the quality of experimental methodology practiced in robotics. This group is named GEM (Good Experimental Methodology) Special Interest Group.

We believe this general aim can be achieved by sharing good practice via educational workshops, summer schools, email discussion and web presentation by providing assistance to journal and conference reviewers and editors concerning what constitutes experimental robotics and good practice in that sub-discipline; by encouraging the principled replication and comparison of results; by encouraging the development and use of appropriate systems benchmarks and standard evaluation procedures.

There is a clear community interest in these issues, as it is shown by the summarized review of current activities, in response to the related email circulated on the Euron mailing list some 80-90 positive responses were received. The SIG is focused on experimental methodology with a special emphasis on replication and benchmarking. We are not underestimating the importance of purely theoretical research and related papers, we want to define clear rules for experimental papers. The SIG is studying the state of the art of experimental methodologies in robotics and AI comparing to biology, physics, medicine, psychology, social sciences. A document enumerating recommended criteria for Robotics Journal/Conference reviews, plus a one page summary (reporting standards, methodology standards, etc.) checklist to be adopted for reviews of experimental papers by the members and proposed to the community.

The group will support the adoption of this rules in EUROS, the biennial robotics conference organized by Euron and in other conference, possibly creating a special track. It will be developed a web site to collect and share information of the efforts in these directions. This will be a repository for tools and data sharing and a workflow environment for collaborative work over the Internet. The GEM SIG will organize/support workshops on the GEM and Benchmark at major conferences, develop initiatives (meetings and information interchange) to join efforts and results to identify new issues, applications and projects, including invitation of recognized domain expert, will organize invited sessions to present and disseminate results at major robotics, control, communications, and computer science conferences. A special attention will be given to identify typical cases to demonstrate the importance of the issues targeted by the SIG.

The organisation of summer/winter schools on experimental methods for robotics (high quality teaching on experimental design, methods, reporting etc. with examples of best practice and poor technique, preferably hands-on training as well as lecture material on basic statistical techniques.) will be supported.

It is also envisioned the establishing of the 'Journal of Replicated Robotics Results': a high quality open access web-based journal that encourages the publication of replications of published experimental results, rational reconstructions of systems, and similar.

VI. CONCLUSIONS AND FUTURE WORK

There is a widespread perception of the need of improving experimental practices in robotics, among many others world wide initiatives, the Euron SIG GEM is trying to address these needs.

It is thought that proper and widely accepted replication procedures and performance benchmarks are needed to allow the cumulative progress of robotic science and technologies and even to assess the value of new disruptive ideas.

REFERENCES

[17] https://www.ctnbestpractices.org/