

Towards heterogeneous robot teams for disaster mitigation: Results and Performance Metrics from RoboCup Rescue

Stephen Balakirsky

*Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD, USA
stephen@nist.gov*

Stefano Carpin

*School of Engineering
University of California
Merced, CA, USA
scarpin@ucmerced.edu*

Alexander Kleiner

*Institut für Informatik
University of Freiburg
79110 Freiburg, Germany
kleiner@informatik.uni-freiburg.de*

Michael Lewis

*School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
ml@sis.pitt.edu*

Arnoud Visser

*Informatica Instituut
Universiteit van Amsterdam
1098 SJ Amsterdam, the Netherlands
arnoud@science.uva.nl*

Jijun Wang

*School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
jiwang@sis.pitt.edu*

Vittorio Amos Ziparo*

*Dipartimento di informatica e Sistemistica
Università di Roma "La Sapienza"
00192 Rome, Italy
ziparo@dis.uniroma1.it*

Abstract

Urban Search And Rescue is a growing area of robotic research. The RoboCup Federation has recognized this, and has created the new Virtual Robots competition to complement its existing physical robot and agent competitions. In order to successfully compete in this competition, teams need to field multi-robot solutions that cooperatively explore and map an environment while searching for victims. This paper presents the results of the first annual RoboCup Rescue Virtual competition. It provides details on the metrics used to judge the contestants as well as summaries of the algorithms used by the top four teams. This allows readers to compare and contrast these effective approaches. Furthermore, the simulation engine itself is examined and real-world validation results on the engine and algorithms are offered.

*This work was conducted by the author while visiting the *Institut für Informatik* at the *University of Freiburg*.

1 INTRODUCTION

During rescue operations for disaster mitigation, cooperation is a must (Jennings et al., 1997). In general the problem is not solvable by a single agent, and a heterogeneous team that dynamically combines individual capabilities in order to solve the task is needed (Murphy et al., 2000). This requirement is due to the structural diversity of disaster areas, variety of evidence of human presence that sensors can perceive, and to the necessity of quickly and reliably examining targeted regions.

In the Urban Search And Rescue (USAR) context, there exists no standard robot platform capable of solving all the challenges offered by the environment. For example, there are places only reachable by climbing robots, spots only accessible through small openings, and regions only observable by aerial vehicles. Multi-robot teams not only offer the possibility to field such diverse capabilities. They also exhibit increased robustness due to redundancy (Dias et al., 2004), and superior performance thanks to parallel task execution (Arai et al., 2002). This latter aspect is as important as the first one, since the time to complete a rescue mission is literally of vital importance. However, designing a multi-robot system for rescue from a single robot perspective is not only tedious, but also prone to yield suboptimal solutions. The joint performance of a team depends on assembling the right mixture of individual robot capabilities, and thus has to be designed as a whole.

The RoboCup Rescue competitions provide a benchmark for evaluating robot platforms for their usability in disaster mitigation and are experiencing ever increasing popularity (Kitano and Tadokoro, 2001). Roughly speaking, the *league vision* can be paraphrased as the ability to deploy teams of robots that explore a devastated area and locate victims. Farsighted goals include the capability to identify hazards, provide structural support and more. Since its inception, RoboCup Rescue has been structured in two leagues, the Rescue Robot League and the Rescue Simulation League. Whereas the Rescue Robot League fosters the development of high-mobility platforms with adequate sensing capabilities, e.g. to identify human bodies under harsh conditions, the Rescue Simulation League promotes research in planning, learning, and information exchange in an inherently distributed rescue effort (Kitano and Tadokoro, 2001). It is envisioned that the research endeavors of the two communities will eventually converge to an intermediate point situated between the currently far apart research agendas. In order to achieve this goal the Rescue Simulation League has to move to adopt tools closer to those used in fieldable systems (Carpin et al., 2006b). For this reason the Rescue Simulation League has added a new competition (Carpin et al., 2007a). The Rescue Agents competition, historically the first one being introduced, focuses on the simulation of tens of agents with high level capabilities operating on a city-sized environment. The newly introduced Virtual Robots competition simulates, compared to the Rescue Agents competition, small teams of agents¹ with realistic capabilities operating on a block-sized scenario.

The Virtual Robot competition, which is the third competition running under the RoboCup Rescue Simulation League umbrella, was first held during the RoboCup competitions in 2006. It utilizes the USARSim framework to provide a development, testing, and competition environment that is based on a realistic depiction of conditions after a real disaster (Wang et al., 2005) such as an earthquake, a major fire, or a car wreck on a highway. Robots are simulated on the sensor and actuator level, making a transparent migration of code between real robots and their simulated counterparts possible. The simulation environment, described later on in this paper, allows evaluation of the

¹The maximal team-size during the competition in 2006 was 12 virtual robots.



Figure 1: Representative snapshot of a USARSim indoor (a) and outdoor (b) scene.

performance of large robot teams and their interactions. For example, whereas in the real robot competition there are usually only one or two robots deployed, in the Virtual Robot competition teams deployed up to twelve. Furthermore, the simulator provides accurate ground truth data allowing an objective evaluation of the robots' capabilities in terms of localization, exploration and navigation, e.g. avoidance of bumping. It has been previously stated (Carpin et al., 2007a)(Carpin et al., 2006b) that the Virtual Robot competition should serve the following goals:

- Provide a meeting point between the different research communities involved in the RoboCup Rescue Agents competition and the RoboCup Rescue Robot League. The two communities are attacking the same problem from opposite ends of the scale spectrum (city blocks vs. a small rubble area) and are currently far apart in techniques and concerns. The Virtual Robot competition offers close connections to the Rescue Robot League, as well as challenging scenarios for multi-agent research. The scenarios for the 2006 competition were chosen to highlight these connections. They were based on an outdoor accident scene, and an indoor fire/explosion at an office building. These scenarios included real-world challenges such as curbs, uneven terrain, multi-level terrain (i.e. the void space under a car), maze-like areas, stairs, tight spaces, and smoke. An exact copy of one of the RoboCup Rescue Robot League arenas was also included in the office space, and elements of other arenas were scattered throughout the environment. The area was far too large to be explored by a single agent in the time permitted (20 minutes) and thus the use of multi-agent teams was beneficial. Accommodations were provided in the worlds to assist less capable (in terms of mobility) robotic systems. For example, wheel-chair ramps were provided that allowed for alternative access around stairs. Snap shots of small sections of these environments may be seen in Figure 1.
- Lower entry barriers for newcomers. The development of a complete system performing search and rescue tasks can be overwhelming. The possibility to test and develop control systems using platforms and modules developed by others makes the startup phase easier. With this goal in mind, the open source strategy already embraced in the other competitions is fully supported in the Virtual Robot competition. Software from this year's top teams has already been posted on the web.

- Let people concentrate on what they can do better. Strictly connected to the former point, the free sharing of virtual robots, sensors, and control software allows people to focus on certain aspects of the problem (victim detection, cooperation, mapping, etc), without the need to acquire expensive resources or develop complete systems from scratch. In order to help people determine if they really can “do better,” performance metrics were applied to the competing systems.

The winning teams from the 2006 RoboCup Rescue Virtual Robot competition were:

First Place: Rescue Robots Freiburg, University of Freiburg, Germany

Second Place: Virtual IUB, International University Bremen, Germany

Third Place: UvA, University of Amsterdam, The Netherlands

Best Mapping: UvA, University of Amsterdam, The Netherlands

Best Human-Computer Interface: Steel, University of Pittsburgh, USA

In this paper we provide a detailed description of the performance metrics that were applied during the Virtual Robot competition, as well as a description of the solutions developed by the top performing teams. They were either autonomous or partially tele-operated by humans, and cooperated in order to achieve maximal performance. The paper documents the first step taken towards the quantitative evaluation of intelligent team performance in the context of emergency response. During this first step, mainly benchmarks for measuring the effectiveness in mapping and exploration were developed. We show that the simulation environment provides an ideal basis for evaluating heterogeneous teams. It allows collecting massively data from very different approaches while they are deployed within exactly same situations. We present detailed scores from each team, based on a comparison between data collected during the competition and absolute ground truth from the simulation system. These results provide insights regarding the strengths and weaknesses of each approach, and moreover, serve as basis for assessing the performance metrics that have been applied. We discuss lessons learned from the competition and conclude necessary improvements for further steps.

The remainder of the manuscript is structured as follows. The performance metrics, which were used to evaluate the multi-robot systems with respect to the goals of the competition, are presented first in Section 2. Then, the four best approaches that received an award at RoboCup Rescue Virtual competition 2006 are presented. The team sections are composed of a brief introduction, a description of the adopted robot platforms, followed by two sections focusing on the core issues of coordinated exploration and mapping, and finally some discussion. In particular, we start by presenting the winning Team *Rescue Robots Freiburg* (Section 3). This team explicitly addressed coordinated exploration and mapping based on autonomously released RFID tags which were used both as coordination points for exploration and as features for building topological maps of the environment. The description of *Virtual IUB* team, which reached the second place (Section 4), follows. This team focused on building a complete system of selfish robots based on state of the art approaches. In contrast, *UvA* team (Section 5) which reached the third place and won the *Best Mapping* award, concentrated on developing state of the art techniques for mapping. Finally, we present the *Steel* team (Section 6), which won *Best Human-Computer Interface* award and which was the only team among the “best” four to utilize a human operator. The scientific goal of this

team was to build a Human-Robot Interface allowing to coordinate the exploration of humans and robots. We close the paper with the details on the results (Section 7) and the lessons learned from the competition (Section 8).

2 PERFORMANCE METRICS AND SIMULATION

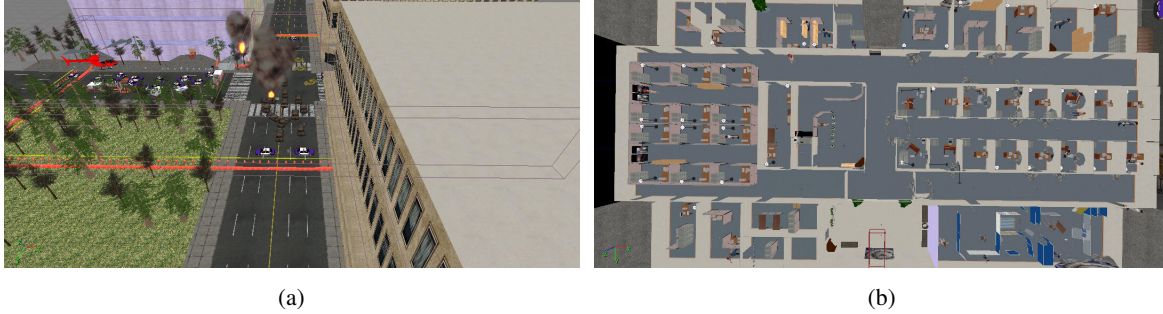


Figure 2: Overview of the simulated world used for the 2006 RoboCup Virtual Robot competition. The world consisted of outdoor (a) and indoor (b) areas.

In designing metrics for the Virtual Robot competition, the objectives of developing relevant metrics for the urban search and rescue community, providing metrics that could be applied to both virtual and real robots with little modification, and assuring that the metrics did not place an undo burden on competitors were all balanced. In order to maintain relevance, metrics from the Rescue Robot League that emphasized the competency of the robots in tasks that would be necessary at a disaster site were used as a starting point. These tasks included locating victims, providing information about the victims that had been located, and developing a comprehensive map of the explored environment. The environment (shown in Figure 2) for the competition consisted of an arena that was over $11,000 m^2$ in size and contained a multi-story office building (with office space and cubicals), outside streets, a park, and several accident scenarios. Participants needed to deploy teams of agents to fully explore the environment.



Figure 3: Maze environment used for real/virtual development and testing. (a) is the real world maze and (b) is its virtual representation.

While this was a fictitious world, researchers have modeled their real-world lab environments to assist in real/virtual testing and development. One such environment is shown in Figure 3. Figure 3.a shows an actual outdoor maze environment for robotic testing. Figure 3.b shows the simulated version of this environment. During testing, real robots cooperate with virtual robots to perform missions and behaviors. For example, a real robot may be exploring the maze in cooperation with

a virtual robot. The robots share map information and even see each other in their own respective representations of the real or virtual worlds.

The primary goal of the competition was to locate victims in the environment and determine their health status. However, what does it mean to “locate” a victim? How does one autonomously obtain health status? Several possible interpretations exist ranging from simply requiring a robot to be in proximity of a victim (e.g. drive by the victim) to requiring the robot to employ sensor processing to recognize that a victim is located near-by (e.g. recognize a human form in a camera image) and then examine that victim for visually apparent injuries. While recognizing a human form from a camera image is the solution most readily portable to a real hardware, it places an undo burden on both the competitors and the evaluation team. For the competitors, a robust image processing system would need to be developed that could recognize occluded human forms. No matter how exceptional the mapping and exploration features of a team were, failing to produce the image processing module would result in a losing effort. In addition, the evaluation team would need to develop an entire family of simulated human forms so that teams could not “cheat” by simply template matching on a small non-diverse set of victims. It was decided that robots should be required to be “aware” of the presence of a victim, but that requiring every team to have expertise in image processing was against the philosophy of lowering entry barriers. Therefore, a new type of sensor: a victim sensor, was introduced. To allow for the metrics to be portable to real hardware, this new sensor would need to be based on existing technology.

The victim sensor was based on Radio Frequency Identification Tag (RFID) technology. False alarm tags were scattered strategically in the environment, and each victim contained an embedded tag. At long range (10 m), a signal from the tag was readable when the tag was in the field of view (FOV) of the sensor. At closer range (6 m), the sensor would report that a victim or false alarm was present. At even closer range (5 m) the ID of the victim would be reported. Finally, at the closest range (2 m), the status of the victim (e.g. injured, conscious, bleeding, etc.) was available. Points were subtracted for reporting false alarms, and were awarded for various degrees of information collected from the victims. Bonus points were awarded for including an image of the victim with the report.

As the robots were exploring the environment, their poses (on 1 s intervals) and any collisions between the robots and victims were automatically logged. The pose information was fed into a program that automatically computed the amount of area that was covered by the robotic teams. This figure was normalized against the expected explored area for the particular run, and points were awarded accordingly. The collision information was used as an indication of suboptimal navigation strategies that should be penalized. Another parameter that was used to determine the overall score was the number of human operators that were needed to control the robots. The idea was borrowed from the Rescue Robot league with the intent of promoting the deployment of fully autonomous robot teams, or the development of sophisticated human-robot interfaces that allow a single operator to control many agents.

The final area that was judged during the competition was map quality. The map quality score was based on several components.

- Metric quality – The metric quality of a map was scored automatically by examining the reported locations of “scoring tags”. Scoring tags were single shot RFID tags (they could

only be read once). A requirement of the competition was for the teams to report the global coordinates of these tags at the conclusion of each run. The automatic scoring program then analyzed the deviation of the perceived locations from the actual locations.

- Multi-vehicle fusion – Teams were only permitted to turn in a single map file. Those teams that included the output from multiple robots in that single map were awarded bonus points.
- Attribution – One of the reasons to generate a map is to convey information. This information is often represented as attributes on the map. Points were awarded for including information on the location, name, and status of victims, the location of obstacles, the paths that the individual robots took, and the location of RFID scoring tags.
- Grouping – A higher order mapping task is to recognize that discrete elements of a map constitute larger features. For example the fact that a set of walls makes up a room, or a particular set of obstacles is really a car. Bonus points were awarded for annotating such groups on the map.
- Accuracy – An inaccurate map may make a first responder's job harder instead of easier. Points were assessed based on how accurately features and attributes were displayed on the map.
- Skeleton quality – A map skeleton reduces a complex map into a set of connected locations. For example, when representing a hallway with numerous doorways, a skeleton may have a line for the hallway and symbols along that line that represent the doors. A map may be inaccurate in terms of metric measurements (a hallway may be shown to be 20 m long instead of 15 m long), but may still present an accurate skeleton (there are three doors before the room with the victim). The category allowed the judges to award points based on how accurately a map skeleton was represented.
- Utility – One of the main objectives of providing a map was to create the ability for a first responder to utilize the map to determine which areas had been cleared, where hazards may be located, and where victims were trapped. Points were granted by the judges that reflected their feelings on this measure.

$$S = \frac{V_{ID} * 10 + V_{ST} * 10 + V_{LO} * 10 + t * M + E * 50 - C * 5 + B}{(1 + N)^2} \quad (1)$$

The above mentioned elements were numerically combined according to Equation 1. The meaning of the variables is discussed below. This equation represents a schema that took into account merit factors that concerned (1) victims discovery, (2) mapping, and (3) exploration. The exact point calculations for each factor are presented below.

1. 10 points were awarded for each reported victim ID (V_{ID}). An additional 10 points were granted if the victim's status (V_{ST}) was also provided. Properly localizing the victim in the map was rewarded with an additional 10 points (V_{LO}). At the referee's discretion, up to 20 bonus points were granted for additional information produced (B). For example, some teams managed to not only identify victims, but to also provide pictures taken with the robot's cameras. For this additional information teams were awarded with 15 bonus points.

2. Maps were awarded up to 50 points based on their quality (M), as previously described. The obtained score was then scaled by a factor ranging between 0 and 1 (t) that measured the map's metric accuracy. This accuracy was determined through the use of the RFID scoring tags.
3. Up to 50 points were available to reward exploration efforts (E). Using the logged position of every robot, the total amount of explored square meters (m^2) was determined and related to the desired amount of explored area. This desired amount was determined by the referees and was based on the competition environment. For example, in a run where 100 m^2 were required to be explored, a team exploring 50 m^2 would receive 25 points, while a team exploring 250 m^2 would receive 50 points, i.e. performances above the required value were leveled off.

On the penalization side, 5 points were deducted for each collision between a robot and a victim (C). Finally, the overall score was divided by $(1 + N)^2$, where N was the number of operators involved. So, completely autonomous teams, i.e. $N = 0$, incurred no scaling, while teams with a single operator had their score divided by 4. No team used more than one operator.

It should be noted that except for the map quality, all of the above components were automatically computed from the information logged during the competition. Therefore subjective opinions during the scoring stage were reduced to a minimum. In an ideal scenario, the scoring step would be completely automatic as is currently the case for the RoboCup Rescue Agents competition. In addition to assigning points to determine the overall best systems, the judges assigned winning teams in the special categories of map creation and human-machine interface. The map creation award was presented to the team that consistently scored the highest in the map quality assessment while the human-machine interface award recognized the team with the most innovative robot control console. These performance metrics were successfully applied to judging the 2006 Virtual Robot competition at RoboCup 2006.

2.1 USARSim FRAMEWORK

The current version of USARSim (Balakirsky et al., 2006) is based on the UnrealEngine2 game engine that was released by Epic Games as part of Unreal Tournament 2004.² This engine may be inexpensively obtained by purchasing the Unreal Tournament 2004 game. The engine handles most of the basic mechanics of simulation and includes modules for handling input, output (3D rendering, 2D drawing, and sound), networking, physics and dynamics. Multiplayer games use a client-server architecture in which the server maintains the reference state of the simulation while multiple clients perform the complex graphics computations needed to display their individual views. USARSim uses this feature to provide controllable camera views and the ability to control multiple robots. In addition to the simulation, a sophisticated graphical development environment and a variety of specialized tools are provided with the purchase of Unreal Tournament.

The USARSim framework³ builds on this game engine and consists of:

²Certain 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 or their institutions, nor does it imply that the software and tools identified are necessarily the best available for the purpose.

³The USARSim framework can be downloaded from <http://sourceforge.net/projects/usarsim>.

- standards that dictate how agent/game engine interaction is to occur,
- modifications to the game engine that permit this interaction,
- an Application Programmers Interface (API) that defines how to utilize these modifications to control an embodied agent in the environment,
- 3-D immersive test environments.

When an agent is instantiated through USARSim, three basic classes of objects are created that provide for the complete control of the agent. These include robots, sensors, and mission packages and are defined as part of the API to USARSim. For each class of objects there are class-conditional messages that enable a user to query the component's geography and configuration, send commands, and receive status and data. Permissible calls into the game engine and complete details on the API may be found in the USARSim Reference Manual (Wang and Balakirsky, 2007).

It is envisioned that researchers will utilize this framework to perfect algorithms in the areas of:

- Autonomous multi-robot control,
- Human, multi-robot interfaces,
- True 3D mapping and exploration of environments by multi-robot teams,
- Development of novel mobility modes for obstacle traversal,
- Practice and development for real robots that will compete in the RoboCup Rescue Robot league.

Moreover, it is foreseeable that USARSim will also be valuable in robotics education contexts (Carpin et al., 2007b).

3 RESCUE ROBOTS FREIBURG

This section presents the Rescue Robots Freiburg team (Kleiner and Ziparo, 2006) for the Virtual Rescue Competition, describing their approach to explore unknown areas and to build a topological map augmented with victim locations. In contrast to their counterpart in the Real Robots Competition (Kleiner et al., 2006a), the goal here is to develop methods for mapping and exploration that are suitable for large teams of robots inhabiting wide environments. Therefore, these methods are designed to require low computational and representation costs, while not relying on the use of direct communication. The basic idea is, on the one hand, to perform grid-based exploration and planning, locally restricted to the close vicinity of the robot. On the other hand, to utilize RFID tags for the global coordination of the exploration (e.g. to leave information in the field by programming their memory). Note that these RFID tags were autonomously deployed by the robots and are different from those used by the competition system to calculate the team's score. The latter were prepared in order to be readable only once and hence unusable for team coordination and localization.

This section first introduces the simulated hardware used for the competition along with some real world validation. Then, the coordinated search method based on RFIDs and its embedded

navigation system is presented. Finally, the Simultaneous Localization And Mapping (SLAM) approach, which is based on the topological graph built on the autonomously deployed RFIDs, is presented.

3.1 ROBOT PLATFORM

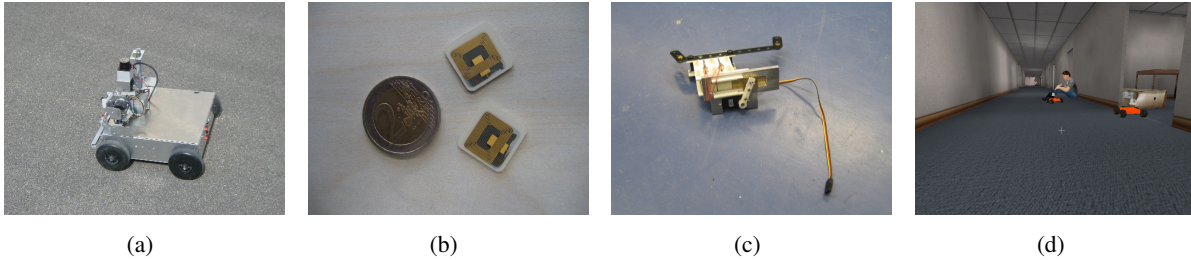


Figure 4: The 4WD rescue robot (a) and RFID tags utilized with this robot (b) with a hand crafted deploy device (c). A model of this robot simulated in the USARSim simulator within an office-like environment (d).

The robot model used during the virtual competition is based on the real robot *Zerg*, a platform which has been originally developed for the Rescue Robot league competitions (Kleiner et al., 2006a), shown in Figure 4.a. The robot utilizes Ario RFID chips from *Tagsys* (see Figure 4.b) with a size of $1.4 \times 1.4\text{cm}$, 2048Bit RAM, and a response frequency of 13.56MHz, implementing an anti-collision protocol, which allows the simultaneous detection of multiple RFIDs within range. A Medio S002 reader, likewise from *Tagsys*, is employed for the reading and writing of tags and allows to detect the tags within a range of approximately 30cm. The antenna of the reader is mounted in parallel to the ground, which allows it to detect any RFID tag lying beneath the robot. The active distribution of the tags is carried out by a custom built actuator, made from a magazine, maximally holding 100 tags, and a metal slider that can be moved by a conventional servo. With this platform, real-robot experiments have been conducted, demonstrating the feasibility for the real robot platform to autonomously deploy and detect RFID tags in a structured environment (Kleiner et al., 2006b).

The simulation model of the *Zerg* captures the same physical properties as the real one, e.g. a four wheel drive, a RFID tag release device, a RFID antenna, Inertial Measurement Unit (IMU), and LRF (see Figure 4.d). The sensors of the model are simulated with the same parameters as the real sensors, expect the real RFID reading and writing range. Without loss of generality, these ranges have been extended to two meters, since this parameter mainly depends on the size of the transmitter’s antenna, which can also be replaced.

3.2 NAVIGATION AND EXPLORATION

In this section, we present the coordination mechanism (Ziparo et al., 2007) which allows robots to explore an environment with low computational overhead and communication constraints. In particular, the computational costs do not increase with the number of robots. The key idea is that the robots plan paths and explore based on a local view of the environment, that maintains consistency through the use of indirect communication via RFID tags. To efficiently and reactively

navigate, robots continuously plan paths based on their local knowledge of the environment, which is maintained within an occupancy grid limited in size for allowing fast computation. In particular, in the competition implementation, the grid is restricted to a four meter side square. The occupancy grid is shifted based on the odometry and continuously updated based on new scans. This allows to overcome the accumulation of the odometry error when moving, while having some memory of the past. The exploration process periodically selects a target, as shown in the following, and continuously produces plans (based on A* search performed on the occupancy grid) to reach it. The continuous re-planning allows the robot to reactively avoid newly perceived obstacles or unforeseen situations caused by errors in path following. While navigating in the environment, robots maintain a local RFID set (LRS), which contains all the perceived RFIDs which are in the range of the occupancy grid. On the basis of this information, new RFIDs are released in the environment by the robots in order to maintain a predefined density of the tags. In order to avoid collisions with teammates, the local displacement between robots are synchronized via RFID tags. That is, if a robot knows the relative offset of a teammate with respect to a commonly perceived RFID (which is stored and time-stamped in the RFID itself), it can compute its relative position. This information is used to label the cells of the occupancy grid within a given range from the teammates as penalized. This will be taken into account at the planning level adding an extra cost for going through such locations.

The fundamental problem in the exploration task is how to select targets for the path planner in order to minimize overlapping of explored areas. This involves: i) choosing a set of target locations, ii) computing an utility value for each of them, and iii) selecting the best target, based on the utility value, for which the path planner can find a trajectory. First, a set of targets is identified extracting frontiers (Yamauchi, 1997) from the occupancy grid. These are then ordered based on an utility calculation which takes into account an inertial factor and the density of the paths (stored in the LRS) in the vicinity of the frontier. The inertial factor prefers frontiers ahead of the robot and prevents the robot from changing direction (which would result in an inefficient behavior) too often. If the robot would have full memory of all perceptions (i.e. a global occupancy grid), the inertial factor would be enough to allow a single robot to explore successfully. Due to the limitation of the local occupancy grid, the robot will forget the areas previously explored and thus will possibly go through already explored ones. In order to maintain a memory of the previously explored areas the robots store in the nearest RFID at writing distance the poses from their trajectories (discretized at a lower resolution with respect to the occupancy grid). It is worth noting that robots writing and reading from RFIDs, not only maintain memory of their own past but also that of the other robots implementing a form of indirect communication. Thus, neither multi-robot navigation nor exploration require direct communication. This feature is very useful in all those scenarios (e.g. disaster scenarios) where wireless communication may be limited or unavailable. A notable feature of the approach is that the computation costs do not increase with the number of robots. Thus, in principle, there is no limit, other than the physical one, to the number of robots composing the team.

3.3 LOCALIZATION AND MAPPING

The approach for localization and mapping (Ziparo et al., 2007) is based on the RFID tags distributed by the robots in the environment. For the purpose of navigation each robot tracks its own pose by integrating measurements from the wheel odometry and IMU sensor with a Kalman filter,

whereas it is assumed that the yaw angle of the IMU is aligned to magnetic north, i.e. that IMU measurements are supported by a compass. In order to communicate poses with other robots, e.g. for collision avoidance, or to report them to a central command post, poses are denoted by the identification of the closest RFID and the robot's local displacement from the RFID's location, estimated by the Kalman filter. Each time the robot passes the location of a RFID i , the Kalman filter is reset in order to estimate the displacement $d_{ij} = (\Delta x_{ij}, \Delta y_{ij})^T$ with covariance matrix Σ_{ij} to the subsequent location of RFID j .

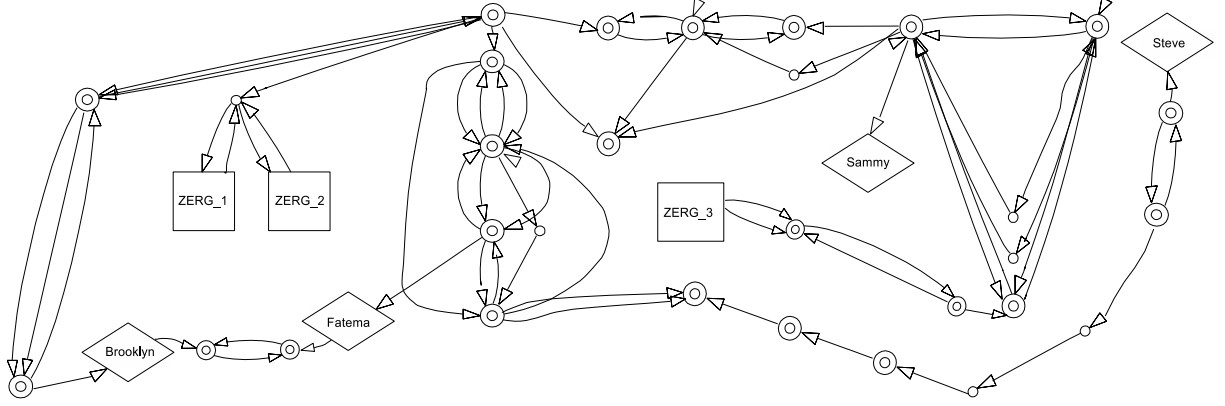


Figure 5: A part of the topological map of the disaster space generated by a team of robots: rectangles denote robot start locations, diamonds denote detected victim locations, and circles denote RFID locations.

By doing this, each robot successively builds a topological map consisting of vertices that are RFID tags, starting location, and victim locations, and edges that are connections between them (see Figure 5 for an example). These graphs are exchanged between robots if they are within communication range. Due to the unique identification of RFID tags, graphs from multiple robots are easily merged into a single graph consisting of the unification of all vertices and edges, whereas the overall displacement D_{ij} between two vertices i and j is computed from the weighted average of performed measurements of all robots: $D_{ij} = \frac{1}{C} \sum_k \sum_{k_{ij}} d_{k_{ij}} \Sigma_{k_{ij}}^{-1}$, whereas $C = \sum_k \sum_{k_{ij}} \Sigma_{k_{ij}}^{-1}$ and k_{ij} indicates the measurement between node i and j performed by robot k . Note that if there does not exist a measurement between two nodes, the elements of the corresponding covariance matrix are set to zero.

With the method described above, multiple robots are able to generate a map of locally consistent displacement estimates between RFID landmarks. This approach is motivated from the idea that human beings can utilize this graph in order to locate victims by successively following RFIDs, which can be measured with a mobile RFID reader. However, there also exist more sophisticated methods that allow the generation of globally consistent graphs, as for example the method from Lu and Milios (Lu and Milios, 1997), which has been successfully adopted for optimizing RFID graphs within former work (Kleiner et al., 2006b).

3.4 DISCUSSION

In this section we presented the Rescue Robots Freiburg team that adopted efficient distributed methods without direct communication suitable for large teams of robots acting in unknown and wide environments. The approaches are all based on RFIDs which allow for efficient coordinated exploration and SLAM. In the former case, RFIDs serve as a distributed common knowledge, while in the latter case, they are used to build abstract topological maps and to solve data association issues. The approach presented here performed successfully during the competition, although, as with many local search algorithms, the coordinated exploration suffered from local minima. Later work on the topic (Ziparo et al., 2007) shows how to extend the presented method (through planned restarts of the local search) for improving the performance of the system.

4 VIRTUAL IUB

The main goal of the Virtual IUB team was putting together a fully autonomous multi-robot system integrating state-of-the-art algorithms and techniques to attack problems relevant to the USAR domain. Due to the tight time constraints faced while developing the team, during a rescue mission each robot acts individually without trying any coordination with the others. As explained later, cooperation occurs only when the mission is over and robots merge their individual partial results. Accordingly to our former experience in the Robocup Rescue Robot League (Birk et al., 2005), the core topics addressed were distributed exploration and navigation, cooperative map building and localization, and victim identification. Each of these topics has received its fair share of research attention in the past, but examples of fully functioning systems incorporating all of them are scarce. This lack substantiates the difficulty of integrating all these algorithmic results into a single working system. For these reasons the major effort of the Virtual IUB team was in system integration rather than the development of new algorithmic techniques. Having this objective in mind, existing available software was reused whenever possible. The four fundamental levels of competence identified during the design stage are *Navigation*, *Mapping*, *Exploration*, and *Victim Detection*.

4.1 ROBOT PLATFORM

The mobile platform selected by the VirtualIUB team was the Pioneer P3AT (see Figure 6.a), a four wheels all-terrain differential drive robot produced by MobileRobots Inc. (MobileRobotsInc., 2006). The basic platform is equipped with 16 sonar sensors (8 in the front and 8 on the back, 5 meters range). Odometry sensors are provided as well. We customized this elementary base placing one proximity range finder (20 meters range, 180 degrees field of view, 180 readings), a camera mounted on a pan-tilt unit (120×180 pixels resolution), and a IMU measuring *roll*, *pitch*, and *yaw*. Following the competitions' goals, the robot is also equipped with a *Victim sensor*, i.e. a device capable of reading RFID tags placed on the victims.

Although the research group does not own any of these robots, preliminary validation tests for the used sensors were conducted, in order to close the loop between simulation and reality. Accordingly to initial experiments detailed in (Carpin et al., 2006a) and (Carpin et al., 2006b), it was substantiated that while working with proximity sensors and cameras, it is possible to extrapolate results obtained within the USARSim environment to reality. For example, Figure 7 shows the

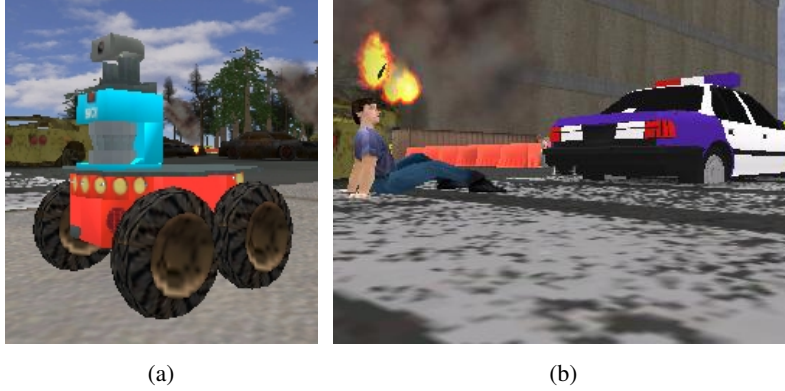


Figure 6: (a) VirtualIUB used up to eight P3AT robots during the virtual robots competition. (b) One of the victims detected during the competition.

results of the same Hough transform (Duda and Hart, 1972) algorithm run on real data (left) and data collected from the corresponding simulated environment (right).

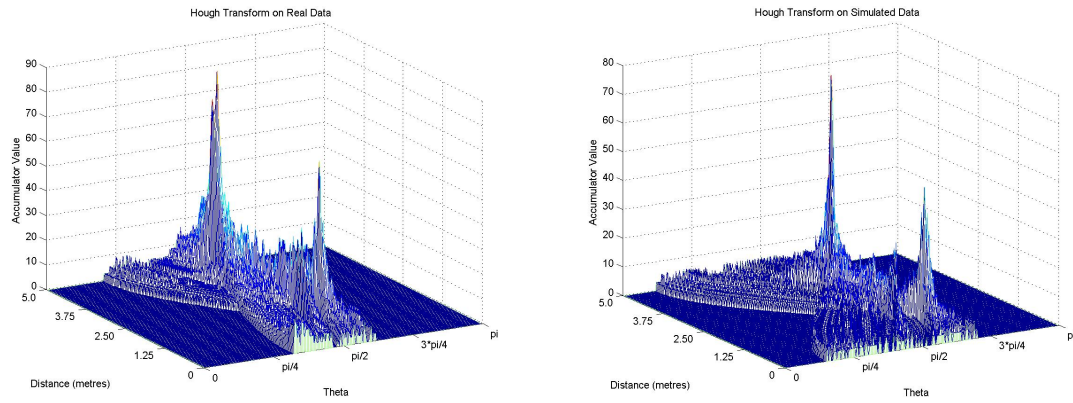


Figure 7: Hough transform on real (left) and simulated (right) data

4.2 NAVIGATION AND EXPLORATION

From a software point of view, the four fundamental tasks previously outlined are implemented by four separate modules running in parallel. Two additional modules have the exclusive role of polling sensors and driving actuators, in order to reduce concurrency problems (see Figure 8). Each robot in the team implements the same architecture.

The *Navigation* block is fundamentally a safety component designed to avoid collision with the environment. This goal is achieved by reading the sonar sensors. If no obstacle is closer than a specified safety threshold, no action is taken. Otherwise, a repulsive vector field is computed and a suitable command is sent to the module controlling the the actuators. Commands generated by the Navigation block prevail over any other command.

The *Exploration* block is the core component of the whole system. Exploration builds upon the popular idea of frontier based exploration introduced in (Yamauchi, 1997). At a fixed frequency Exploration gets a snapshot of the most recent map produced by Mapping (described in the next

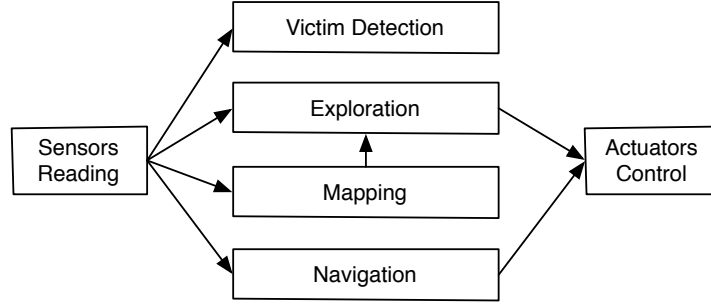


Figure 8: Six different blocks implement the required functionalities to complete a USAR mission. Arrows indicate the data flow.

subsection). Exploration assumes that a grid map is available and that each grid cell is associated with a state that can be either *free*, *occupied* or *unknown*. Exploration extracts all available *frontiers* from the currently available map. A cell is defined to be a frontier cell if it is free, and it is adjacent to a cell whose state is unknown. A frontier is a set of contiguous frontier cells. Once all frontiers have been extracted, the frontier closest to the robot is determined and a navigation function to the frontier is computed. Navigation functions over grids are convenient ways to encode feedback motion plans (LaValle, 2006). The exploration module then sends commands to the Actuators Control module in order to perform a gradient descent over the navigation map to reach the selected frontier. Even if during this path the Navigation module overwrites the commands generated by the Exploration module, the navigation function is still valid and the gradient descent can be resumed, in accordance with the concept of feedback motion plan. If during the motion towards a selected frontier a victim is detected, the Exploration block computes a new navigation function having as its goal the detected victim, rather than the previously identified frontier. Considering the goals of the competition and the ultimate purpose of USAR missions, this greedy choice is always appropriate.

4.3 MAPPING AND VICTIM IDENTIFICATION

In order to perform a meaningful exploration of the unknown environment, some sort of spatial representation is needed to decide where to move next in an informed way. This functionality is implemented through the level of competence named *Mapping*. Rather than implementing a SLAM algorithm from scratch, the *gmap* SLAM algorithm presented in (Grisetti et al., 2005) was adopted⁴. The *gmap* algorithm was developed and tested on a platform similar to the P3AT robot used by the VirtualIUB team, and therefore its integration was straightforward. The map can be updated at every cycle just by providing the most recent odometry and laser data. Gmap divides the environment to be mapped in cells and assigns to every cell a value p_{occ} , i.e. the probability that the cell is occupied. Initially all cells are set to a value encoding lack of knowledge about the status of the cell (also called *unknown* state). The Mapping block never acts on the actuators but just uses data coming from the laser and odometry sensors. This is a design choice, i.e. exploration is never guided to directions that would improve the resulting map in terms of accuracy. Each robot instead strives to get cover as much space as possible.

⁴its implementation is freely downloadable from the authors' website

Finally, the *Victim detection* module is in charge of taking pictures of victims once the robot is close enough. While this task may seem simple enough not to justify a separate module, this design choice was made from a long term perspective of dropping the Victim sensor and grounding it exclusively on image processing. Figure 6.b shows one of the pictures detected by the Victim Detection module during the competition.

4.4 DISCUSSION

During every run robots performed in a *selfish* way, i.e. they did not try share information continuously. This choice was made mainly from lack of time to implement cooperative behaviors. Instead, cooperation was triggered at the end of the mission. When the mission ended, each robot sent its map to a pre-specified *master robot* who merged them together. Although in the past the research group developed different approaches for map merging (Carpin et al., 2005)(Birk and Carpin, 2006), precise knowledge of the initial robot positions allows the use of much simpler approaches. Knowing the initial position and orientation of each robot, the master robot can compute suitable rotations and translations to *patch* the various maps together. Figure 9 shows a map obtained after combining together 8 maps.

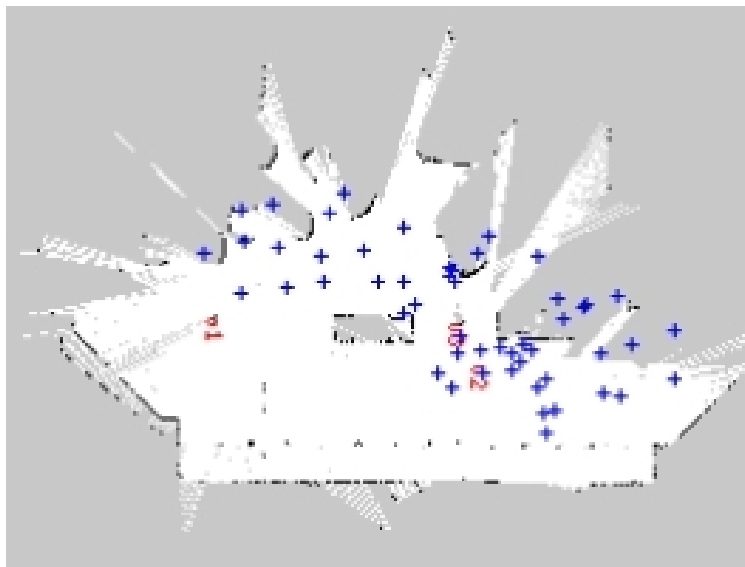


Figure 9: A combined map generated by eight robots. The map shows also victims' position (insicated as vx) and the detected RFID tags as crosses.

Not surprisingly, the main drawback of the selfish approach emerged during the exploration stage. Certain areas were covered multiple times, while other were simply ignored. Moreover, the lack of coordination occasionally generated disruptive interactions between the robots, mostly when trying to negotiate narrow passages. It then appears evident that the next step to undertake in order to refine the system will be the integration of cooperation mechanisms.

5 UvA RESCUE

The approach of the UvA Rescue team differentiated from the other teams in the sense that the focus was on solving the simultaneous localization and mapping problem for multiple robots in a rescue setting. A high quality map is valuable to a human first responder to assess the situation and determine the actions to be taken. A high quality map is also valuable for a team of robots. A shared map is a prerequisite to extend the planning of the team's joint actions from purely reactive via locally optimal towards globally optimal. The UvA Rescue team decided that by first concentrating on the simultaneous localization and mapping (SLAM) problem, later extensions towards complex multi-agent exploration should be possible.

5.1 ROBOT PLATFORM

The robotic platform utilized by the UvA Rescue team during the Virtual Robot competition is equivalent with the P3AT robot selected by the VirtualIUB team, except that no camera was used. The same robot was used by the SPQR team in the Rescue Robot league (Calisi et al., 2006). For the SLAM approach of the UvA the choice of the robotic platform was of minor importance, because the method is independent of the actual movements and odometry measurements of the robot. The SLAM approach fully relies on the measurements collected by the laser scanner, and is applicable to any robot which can carry a laser scanner (the *SICK LMS 200* weights 4.5 kg).

5.2 NAVIGATION AND EXPLORATION

Although autonomous exploration is an important research area, the UvA rescue team did not focus their attention on complex exploration algorithms. Instead a small set of reactive and robust exploration behaviors were designed. The control between the behaviors is managed with a state machine. During the competition four behaviors were used, but the set could easily be extended with more complex behaviors. The default behavior is *Explore*. During this behavior the robots try to move straight ahead as long as possible. When the sonar or laser range scanners detect an obstacle in front of the robot, it will stop, move backwards for a random duration, turn left or right randomly and move ahead again. The other behaviors are as reactive as *Explore*.

Simplistic as this small set of behaviors may seem, it is excellent for covering long distances (as can be seen from table 7) with multiple robots through corridors and open spaces, while collecting detailed information of the victims encountered (see table 2). The randomness in the behaviors has the effect that the robots spread over the area to be explored, without explicit coordination.

5.3 LOCALIZATION AND MAPPING

The mapping algorithm of the UvA Rescue team is based on the manifold approach (Howard et al., 2006). Globally, the manifold relies on a graph structure that grows with the amount of explored area. Nodes are added to the graph to represent local properties of newly explored areas. Links represent navigable paths from one node to the next.

The UvA Rescue team takes no information about the actual movement of the robot into account in creating the links. All information about displacements is derived from the estimates from scan matching. The displacement is estimated by comparing the current laser scan with laser scans recorded shortly before, stored in nearby nodes of the graph. In principle the scan matcher can also perform a comparison with measurements elsewhere in the graph, but such a comparison is only made under specific circumstances (for instance during loop closure, as illustrated in Figure 10). At the moment that the displacement becomes so large that the confidence in the match between the current scan and the previous scan drops, a new node is created to store the scan and a new link is created with the displacement. A new part of the map is learned.

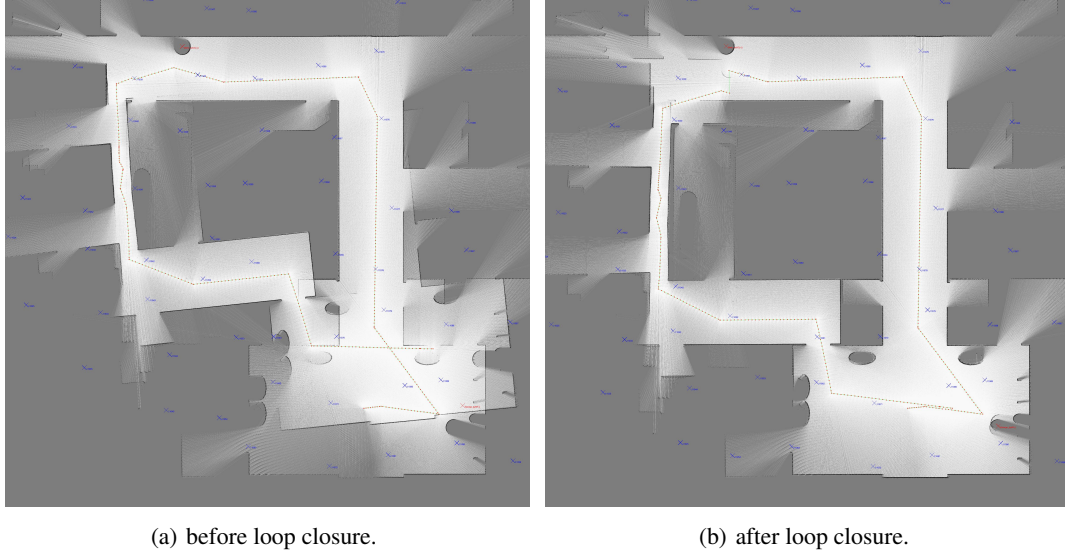


Figure 10: Loop-closing. The robot starts at the lobby at the bottom right and moves up. Then the robot turns left several times until it returns in the bottom right and re-observes a particular landmark and detects the loop.

As long as the confidence is high enough, the information on the map is sufficient and no further learning is needed. The map is just used to get an accurate estimate of the current location. The localization algorithm of the UvA Rescue team maintains a single hypothesis about where the robot currently is and does an iterative search around that location when new measurement data arrives. For each point the correspondence between the current measurement data and the previous measurement data is calculated. The point with the best correspondence is selected as the center of a new iterative search, until the search converges. Important here is the measure for the correspondence. Lu and Milios (Lu and Milios, 1994) have set the de-facto standard with their Iterative Dual Correspondence (IDC) algorithm, but afterward many other approaches have been proposed. The UvA Rescue team has selected the Weighted Scan Matching (WSM) algorithm (Pfister et al., 2007), which works fast and accurate in circumstances where dense measurements are available and the initial estimate can be trusted⁵.

The graph structure allows to have multiple disconnected maps in memory. In the context of SLAM for multiple robots, this allows to communicate the maps and to have one disconnected map for each robot. Additionally, it is possible to start a new disconnected map when a robot loses track

⁵MatLab code of the Weighted Scan Matching algorithm was made available by the authors

of its location, for example after falling down stairs.

When there seems to be an overlap between two disconnected maps, this hypothesis can be checked by scan matching. The displacement and correspondence between the measurements of two nearby points in each overlapping region are calculated. When the correspondence is good, a loop closing operation may be performed to refit all points on the two maps for improved accuracy. An example of a merged map is shown in (Slamet and Pfingsthorn, 2006).

5.4 DISCUSSION

To further validate our results, we applied the algorithm to the real world data available in the Robotics Data Set Repository. Inside this repository we selected a recent dataset collected in a home environment provided by Ben Kröse⁶. The home environment (see Figure 11.a-c) is a difficult environment for mapping. The living-room is full with furniture. One wall is completely made of glass. The bed is low, only the irregularly shaped cushions are visible to the laser scanner. The table in the kitchen is too high for the laser scanner, only table-legs and the back of the chairs are visible. The raw data acquired during a tour through the environment is given in Figure 11.e. The laser scan measurements are indicated in red; the driven path based on odometry information is given in black. The manifold algorithm as used in the competition is able to create a detailed map (see Figure 11.f) of the home, without using the available odometry data. The driven path (based on laser scan information) is illustrated in red; this is in correspondence with the odometry measurements.



Figure 11: Measurements collected in a home environment for the IROS 2006 Workshop 'From sensors to human spatial concepts'.

The UvA Rescue team has demonstrated that high quality maps can be generated both in the Virtual Robot competition as in a real home environment. The basic idea for solving the SLAM problem

⁶<http://cres.usc.edu/radish/view-one.php?name=cogniron>

is that two state-of-the-art algorithms are combined; merged into a new approach. The mapping part of the SLAM-algorithm is based on the Manifold approach (Howard et al., 2006). The localization part of the SLAM-algorithm is based on the Weighted Scan Matching algorithm (Pfister et al., 2007). This method was chosen after careful evaluation (Slamet and Pfingsthorn, 2006) of several scan matching techniques. The chosen algorithm outperforms the classical Iterative Dual Correspondence algorithm (Lu and Milios, 1994) used by Howard.

6 STEEL

In Urban Search And Rescue (USAR), human involvement is desirable because of the inherent uncertainty and dynamic features in the task. Under abnormal or unexpected conditions such as robot failure, collision with objects or resource conflicts human judgment may be needed to assist the system in solving problems. Due to the current limitations in sensor capabilities and pattern recognition people are also commonly required to provide services during normal operation. For instance, in the USAR practice (Casper and Murphy, 2003), field study (Burke et al., 2004), and RoboCup competitions (Yanco et al., 2004), victim recognition is still primarily based on human inspection. Thus for this team, as in real applications, humans and robot(s) must work together to accomplish the task. This implementation, however, goes a step farther by using a multi robot control system (MrCS) to allow robots to navigate autonomously and cooperatively.

The teamwork algorithms used in MrCS are general algorithms that have been shown to be effective in a range of domains (Tambe, 1997). encapsulated in reusable software proxies. The Machinetta (Scerri et al., 2004) proxies are implemented in Java and freely available on the web. Machinetta differs from many other “multiagent toolkits” in that it provides the coordination algorithms, e.g., algorithms for allocating tasks, as opposed to the infrastructure, e.g., APIs for reliable communication.

Most coordination decisions are made by the proxy, with only key decisions referred to human operators. Teams of proxies implement team oriented plans (TOPs) which describe joint activities to be performed in terms of the individual roles to be performed and any constraints between those roles. Typically, TOPs are instantiated dynamically from TOP templates at runtime when preconditions associated with the templates are filled. Constraints between these roles will specify interactions such as required execution ordering and whether one role can be performed if another is not currently being performed. Current versions of Machinetta include state-of-the-art algorithms for plan instantiation, role allocation, information sharing, task deconfliction and adjustable autonomy.

6.1 ROBOT PLATFORM

The robot platform utilized by this team is the experimental robot, Zerg, built by University of Freiburg that was described in Section 3.1. During the RoboCup Rescue competition, the robot was equipped with a front Laser Range Finder with 180 degree FOV and angular resolution of 1 degree, and an Inertial Measurement Unit (IMU) that measures the robot’s 3D orientation. A pan-tilt-zoom camera was mounted on the front of the chassis as well to provide the operator the visual feedback from the scene. The FOV of the camera ranged from 20 to 50 degrees.

6.2 NAVIGATION AND EXPLORATION

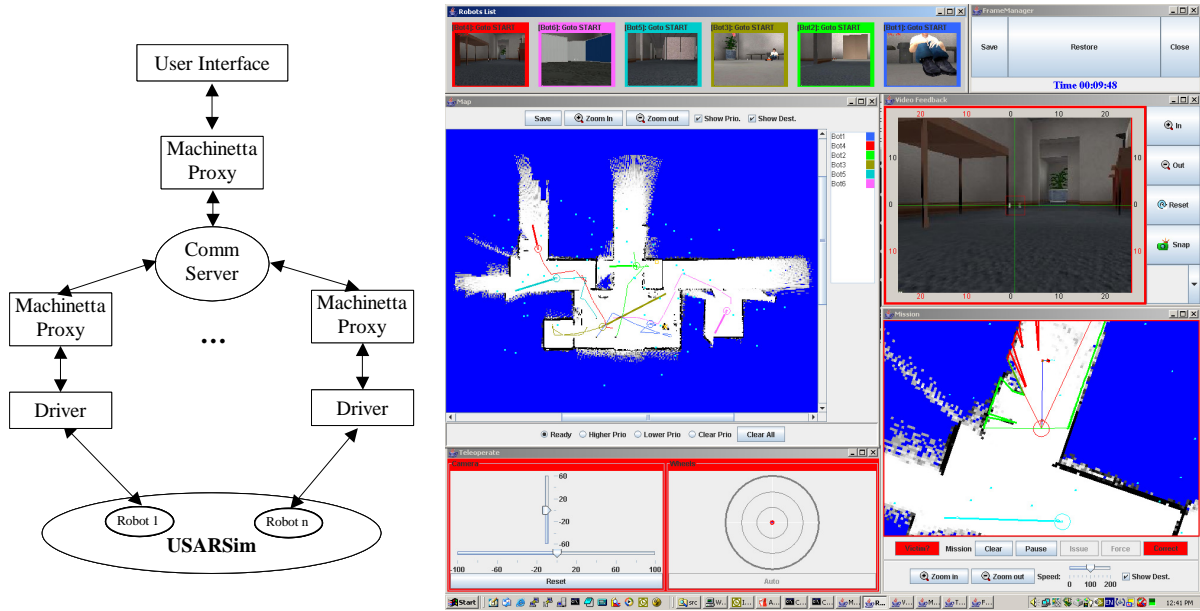


Figure 12: System Architecture (left) and User Interface (right) of MrCS

As a human-machine system, Steel’s multi-robot control approach accomplishes the task of navigation and exploration in a cooperative framework quite distinct from that of the other teams. Although Steel’s multi-robot mapping performance (Table 1) placed either first or second in the trials in which it competed, its conventional use of laser scans and odometry to construct a common map does not significantly advance the state of the art. The “localization and mapping” section is, therefore, omitted and the discussion of “navigation and exploration” focuses on cooperative mechanisms and human interaction that are novel to its approach.

The system architecture of MrCS is shown on the left side of Figure 12. Each robot connects with Machinetta through a robot driver that controls the robot on both low and intermediate levels. For low level control, it serves as a broker that interprets robot sensor data to provide local beliefs, and transfers the exploration plan into robot control commands such as wheel speed control. On the intermediate level, the driver analyzes sensor data to perceive robot state and environment and uses this perception to override control when it becomes necessary to ensure safe movement. When the robot is in an idle state the driver can use laser data to generate potential exploration plans. When a potential victim is detected the driver stops the robot and generates a plan to inspect it. Instead of executing these plans directly, however, they are sent to the Machinetta proxy to trigger TOPs. Using Machinetta’s role allocation algorithm, the robots and human cooperate to find the “best” robot to execute a plan.

The operator connects with Machinetta through a user interface agent. This agent collects the robot team’s beliefs and visually represents them on the interface. It also transfers the operator’s commands in the form of a Machinetta proxy’s beliefs and passes them to the proxies network to allow human in the loop cooperation. The operator is able to intervene with the robot team on three levels. On the lowest level, the operator takes over an individual robot’s autonomy to teleoperate it. On the intermediate level, the operator interacts with a robot via editing its exploration plan.

For instance, an operator is allowed to delete a robot's plan to force it to stop and regenerate plans, or issue a new plan (a series of waypoints) to change its exploration behavior. On the highest level, the operator can intervene with the entire robot team by altering the priorities associated with areas impacting the cost calculations for role allocations thus affecting the regions the robots will explore.

In the human robot team, the human always has the highest authority although the robot may alter its path slightly to avoid obstacles or dangerous poses. In the case in which a robot initiates the interaction the operator can either accept the robot's adjustment or change the robot's plan. One of the challenges in such mixed-initiative systems is that the user may fail to maintain situation awareness of the robot team and individual robot in switching control and therefore make wrong decisions. On the other hand, as the team size grows, the interruptions from the robots may overwhelm the operator's cognitive resources (McFarlane and Latorella, 2002) and the operator may be limited to reacting instead of proactively controlling the robots (Trouvain et al., 2003). These issues in user interface design are described below.

The user interface of MrCS is shown on the right side of Figure 12. The interface allows the user to resize the components or change the layout. Shown in the figure is a typical configuration from the competition in which a single operator controlled 6 robots. On the left upper and center are the robot list and team map panels that show the operator the team overview. The destination of each robot is displayed on the map to help the user monitor team performance. With it, the operator is also able to control regional priorities by drawing rectangles on the map. On the right center and bottom are the camera view and mission control panels that allow the operator to maintain situation awareness for an individual robot and edit its exploration plan. On the mission panel, the map and all the nearby robots and their destinations are represented to provide partial team awareness to facilitate the operator in switching context while moving control from one robot to another. At the bottom left is a teleoperation panel allowing the operator to teleoperate the robot. Interruptions from the robots were mitigated by using principles of etiquette in the design design (Nass, 2004). When the robot needs the operator's attention or help such as in sensing a victim or a precarious pose, instead of popping up a window, the system temporarily changes the mission panel's size and background color or flashes the robot's thumbnail to inform the operator that something needs to be checked. This silent form of alerting allows the operator work at his own pace and respond to the robots as he is able.

In this contest a very simple instantiation of cooperation that was limited to avoiding duplicate efforts in searching the same areas was used. The human robot team design successfully allowed one operator to control 6 robots to explore a moderately wide area and find a large number of victims (although this score was penalized for use of an operator) demonstrating the extent to which well designed human-in-the-loop control can improve effectiveness.

6.3 DISCUSSION

Validating USARsim for human-robot interaction (HRI) presents a complex problem because the performance of the human-robot system is jointly determined by the robot, the environment, the automation, and the interface. Because only the robot and its environment are officially part of the simulation, validation is necessarily limited to some particular definition of interface and automa-

tion. If, for example, sensor-based drift in estimation of yaw were poorly modeled it would not be apparent in validation using teleoperation yet could still produce highly discrepant results for a more automated control regime. Realizing the impossibility of “complete” validation standard HRI tasks were compared for a limited number of interfaces and definitions of automation. Positive results give some assurance that the simulation is physically accurate and that it remains valid for at least some interface and automation definitions.

Validation testing has been completed at Carnegie Mellon’s replica of the NIST Orange Arena for the PER robot using both point-to-point and teleoperation control and for teleoperation of the Pioneer P2-AT (simulation)/P3-AT (robot). Participants controlled the real and simulated robots from a personal computer located in the Usability laboratory at the School of Information Sciences, University of Pittsburgh. For simulation trials the simulation of the Orange Arena ran on an adjacent PC. A standard interface developed for RoboCup USAR Robot League competition was used under all conditions. In testing, robots were repeatedly run along a narrow corridor with varying types of debris (wood floor, scattered papers, lava rocks). Robots followed either a straight path or a complex one that required avoiding obstacles. The sequence, timing, and magnitude of commands were recorded for each test. Participants were assigned to maneuver the robot with either direct teleoperation or waypoint (specified distance) modes of control. Figure 13 showing task completion times demonstrates the close agreement between simulated and real robots. Similar results were found for distribution of commands and pauses, changes in heading, and number of commands issued.

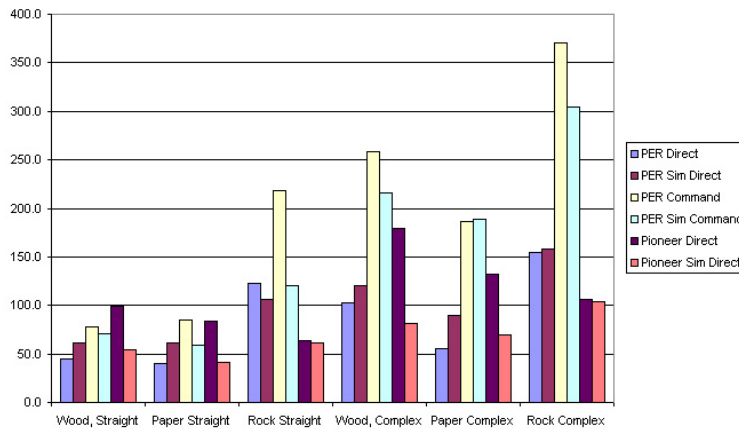


Figure 13: Task completion times.

7 COMPETITION RESULTS

This section overviews the results of the RoboCup Rescue Virtual competition 2006. The competition was held in three preliminary rounds, two semifinals and two finals. The scoring follows the metrics described in Section 2 and thus evaluates team performance at *victim discovery*, *mapping*, and *exploration*. Table 1 reports the overall scores for each team during the entire competition. In the following, the performance measure respectively for victim discovery, mapping and exploration are detailed.

	# operators	RRFreiburg	GROK	IUB	SPQR	STEEL	UvA
		0	1	0	1	1	0
Preliminary 1	Victim points	70	20	20	10	100	0
	Map points	-	16	19	-	22	24
	Exploration points	50	8	18	49	50	12
	Total points	120	44	57	59	172	36
	Operator corrected points	120	11	57	15	43	36
Preliminary 2	Victim points	20	10	20	55	70	5
	Map points	-	15	10	19	21	8
	Exploration points	50	15	26	48	44	26
	Total points	70	40	56	122	135	39
	Operator corrected points	70	10	56	30	34	39
Preliminary 3	Victim points	75	75	50	10	160	30
	Map points	-	31	24	7	27	-
	Exploration points	50	15	41	11	31	30
	Total points	125	120	115	28	218	60
	Operator corrected points	125	30	115	7	54	60
Semifinal 1	Victim points	170	140	260	203	285	150
	Map points	-	23	29	15	47	42
	Exploration points	50	5	38	16	22	44
	Total points	220	168	327	235	354	235
	Operator corrected points	220	42	327	59	89	235
Semifinal 2	Victim points	380	125	205	315	410	40
	Map points	12	39	26	36	45	9
	Exploration points	50	21	35	31	35	50
	Total points	442	185	267	382	489	99
	Operator corrected points	442	46	266	95	122	99
Final 1	Victim points	245	-	145	-	-	-
	Map points	37	-	48	-	-	-
	Exploration points	50	-	50	-	-	-
	Total points	332	-	243	-	-	-
Final 2	Victim points	270	-	135	-	-	-
	Map points	46	-	47	-	-	-
	Exploration points	50	-	34	-	-	-
	Total points	366	-	216	-	-	-

Table 1: Final results from RoboCup '06

		RRFreiburg	GROK	IUB	SPQR	STEEL	UvA
Preliminary 1	reported victims	5	2	1	1	2	0
	status bonus	20	0	10	0	20	0
	victim bonus (picture, etc)	-	-	-	-	-	-
Preliminary 2	reported victims	2	1	1	3	2	1
	status bonus	20	0	10	30	20	0
	victim bonus (picture, etc)	-	-	15	-	30	-
Preliminary 3	reported victims	5	3	3	1	5	3
	status bonus	50	0	0	0	50	0
	victim bonus (picture, etc)	-	45	30	-	60	-
Semifinal 1	reported victims	8	4	7	6	7	6
	status bonus	70	20	30	50	60	30
	victim bonus (picture, etc)	-	60	105	45	90	-
	localization bonus	80	20	70	48	70	60
Semifinal 2	reported victims	16	4	5	9	14	2
	status bonus	80	0	40	60	140	0
	victim bonus (picture, etc)	-	45	75	75	0	-
	localization bonus	160	40	50	90	140	20
Final 1	reported victims	10	-	3	-	-	-
	status bonus	60	-	30	-	-	-
	victim bonus (picture, etc)	-	-	60	-	-	-
	localization bonus	100	-	30	-	-	-
Final 2	reported victims	10	-	5	-	-	-
	status bonus	80	-	10	-	-	-
	victim bonus (picture, etc)	-	-	30	-	-	-
	localization bonus	100	-	50	-	-	-

Table 2: Victim results from RoboCup '06

Providing information about the victims is the primary task of the robot team. Therefore, the victim score is not bound to a maximum, but grows linearly with the number of victims found. Bonus points were awarded both for locating victims and providing extra information. Table 2 summarizes the victim information reported by each team. In particular, the table shows the number of victims identified, the bonus for reporting the status (i.e 10 points for each victim status), and extra information such as pictures (up to 20 points for each report) and accurate localization of the victims (up to 20 points). It is clear from Table 2 that RRFreiburg found at every round the largest number of victims. Still, STEEL received the largest amount of bonus points due to additional information they provided about the victims. The human operators could use the camera of the robot to get detailed information about the victims which was not available to the automated teams.

		RRFreiburg	GROK	IUB	SPQR	STEEL	UvA
Preliminary 1	# Robots	12	1	6	4	6	1
	Area [m2]	902	31	70	197	353	46
	Area/#robots	75,17	31	11,67	49,25	58,83	46
	Speed [m/s]	0,96	0,19	0,20	0,46	0,50	0,47
Preliminary 2	# Robots	12	1	4	4	6	8
	Area [m2]	550	61	105	191	174	104
	Area/#robots	45,83	61	26,25	47,75	29	13
	Speed [m/s]	1,09	0,55	0,31	0,44	0,29	0,19
Preliminary 3	# Robots	10	1	5	7	6	7
	Area [m2]	310	59	164	44	124	120
	Area/#robots	31	59	32,8	6,29	20,67	17,14
	Speed [m/s]	0,67	0,48	0,42	0,06	0,32	0,35
Semifinal 1	# Robots	8	1	6	4	6	6
	Area [m2]	579	27	227	96	134	262
	Area/#robots	72,38	27	37,83	24	22,33	43,67
	Speed [m/s]	2,62	0,33	0,35	0,30	0,27	0,51
Semifinal 2	# Robots	8	1	6	5	6	7
	Area [m2]	1276	82	139	123	139	286
	Area/#robots	159,5	82	23,17	24,6	23,17	40,86
	Speed [m/s]	2,61	0,66	0,21	0,21	0,38	0,48
Final 1	# Robots	8	-	8	-	-	-
	Area [m2]	1203	-	210	-	-	-
	Area/#robots	150,38	-	26,25	-	-	-
	Speed [m/s]	2,67	-	0,24	-	-	-
Final 2	# Robots	8	-	6	-	-	-
	Area [m2]	350	-	136	-	-	-
	Area/#robots	43,75	-	22,67	-	-	-
	Speed [m/s]	1,82	-	0,36	-	-	-

Table 3: Exploration Results from RoboCup '06

Exploration was evaluated based on the totally explored area of each team. These values were automatically computed from the logs of the server hosting the simulator. Table 7 gives an overview on the number of deployed robots, and area explored by each team. Furthermore, the average area explored by a single robot of each team is shown (*Area/#robots*). RRFreiburg demonstrated that it is possible to quickly explore a large area with a team of robots. On average, their robots drove 5 times as fast as the robots of the other teams. Detailed mapping of the disaster area is not possible at those speeds, but reliable navigation could be performed.

From the results it is clear that no team dominated in all the tasks. In particular, the winning team Rescue Robots Freiburg was the only team which addressed cooperative-exploration explicitly, whereas most of the “best” teams adopted a selfish approach. Their RFID-based exploration,

allowed to explore quickly and efficiently large environments at the cost of abstracting the environment representation to a topological map. On the other hand, teams such as UvA, focused on providing high quality merged maps using state of the art techniques which resulted in a third place and best mapping award. UvA thus had to spend more time assessing map quality, rather than exploring, which penalized them in the final scoring both for exploration and number of found victims. VirtualIUB, which reached the second place, mainly focused on building a complete system. They integrated several existing robotic approaches outperforming teams that underestimated some of the robotic challenges. The migration of well known robotic approaches towards USAR-Sim, in general, shows that Virtual Robot Competition is a meaningful test-bed. Finally, STEEL which won the best Human-Robot Interface award, was the only team in the “best” four that was not fully autonomous. It is clear from the results that their performance was impressive although they were strongly penalized by the scoring, for having an operator. In fact, their approach allowed their only operator to successfully coordinate up to seven robots.

8 CONCLUSIONS

This paper presented the outcome of the first Virtual Robot competition at RoboCup 2006. It discussed preliminary efforts of defining performance metrics in terms of mapping, localization, and exploration, for robot teams operating in the rescue scenario. The four best teams from the competition introduced their individual solutions. The wide spectrum of solutions, ranging from autonomy to teleoperation, and from detailed grid-based map representations to abstract topological map representations, shows clearly that there exists a real need for evaluating aspects of team collaboration in this domain, i.e. to identify methods for combining the individual team strengths in an ideal way.

The encouraging outcome of the competition gives rise to multiple research and development directions that will be pursued in the next contests. First of all, despite the already proved close correspondence between results obtained with simulated robots inside USARSim and with the corresponding physical robots, the necessity to obtain more validation data cannot be overemphasized. In particular, a standardized procedure to validate simulated robots is needed to avoid the addition of poorly validated models to the simulation environment.

While the competition provided an overall winner that can claim “best in search and rescue”, it proved to be difficult to discern the exact reasons behind the win. Was it better mapping, better mobility, or better strategy? To address this, next year’s competition will include a round of elemental tests that will provide incite into individual skills and competencies. The tests will include a mapping task, a mobility challenge, and a victim finding challenge. These individual tests will allow us to declare a best in class in each of the skills necessary for urban search and rescue.

Other additions to the competition are being made due to the necessity of closing the loop between simulation and reality. Therefore, the use of a software module to mimic wireless communication between agents is being introduced. When physical robots are fielded in a disaster scenario, their efforts are useless if collected information is not made available to human operators. A preliminary extension of USARSim implementing this idea has already been developed and is included in the official distribution. The new communication model accounts for the signal strength of wireless transmitters and spacial structure of buildings, thus denying communication over long distances

and communication through impenetrable walls. Hence, the new model requires teams to implement coordination strategies for sharing, and more importantly, reporting back their information to a central command post.

In addition to sharing communications, steps are being taken to encourage heterogeneous teams in this year's competition (all teams last year employed homogeneous teams of robots). To accomplish this a small payload flying platform and a slow, large payload, high-mobility platform are introduced. It is hoped that teams will construct heterogeneous groups of robots and apply each of their strong points to addressing the various challenges.

Another important milestone towards heterogeneous robot teams, which is currently under discussion within all Rescue leagues, is to introduce a standardized map representation. A common format could serve not only for comparisons between different approaches, but also as useful tool to exchange spatial information between heterogeneous agents, as well as to compare more rigorously results coming from simulation and real world systems. Furthermore, it makes possible to provide floor plans of building structures to rescue teams in advance, as is common practice during real rescue missions. Here, the long term goal is to introduce communication standards in the Virtual competition and to transfer them to the Rescue Robot league, after they have been extensively tested. Therefore, in future competitions all maps will be submitted to the judges in GEOTIF (a pixel format) or MIF (a vector format). This will allow the judges to more readily overlay the generated maps on ground truth to evaluate their quality, as well as to provide floor plans in advance.

On a more general perspective, it appears also that a systematic approach to team work evaluation is still lacking in the multi-robot research community. The use of a high fidelity simulator like USARSim opens the doors for more thorough investigation in this direction without the need of cumbersome or expensive experimental facilities. The RoboCup event, on the other side, offers the perfect venue to contrast different approaches in a competitive scenario and verify if the developed metrics capture strengths and weaknesses of the developed solutions.

Acknowledgments

The authors thank the many enthusiastic undergraduate and graduate students that contributed to the development of USARSim and of the various teams described in this paper. Mentar Mahmudi, Stefan Markov, Yashodan Nevatia, Ravi Rathnam, and Todor Stoyanov at IUB, Christian Dornhege and Dapeng Zhang at University of Freiburg, Max Pfingsthorn and Bayu Slamet at the Universiteit van Amsterdam, Chris Scrapper of the National Institute of Standards and Technology.

A part of the research reported here is performed in the context of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

- Arai, T., Pagello, E., and Parker, L. (2002). Guest editorial advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661.
- Balakirsky, S., Scrapper, C., Carpin, S., and Lewis, M. (2006). Usarsim: Providing a framework for multi-robot performance evaluation. In *Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, pages 98–102.
- Birk, A. and Carpin, S. (2006). Merging occupancy grids from multiple robots. *Proceedings of the IEEE*, 94(7):1384–1397.
- Birk, A., Carpin, S., Chonnaparamutt, W., Jucikas, V., Bastani, H., Delchev, I., Krivulev, I., Lee, S., Markov, S., and Pfeil, A. (2005). The iub 2005 rescue robot team. In *Robocup 2005: Robot Soccer World Cup IX*, Osaka, Japan. (Proceedings CD-ROM).
- Burke, J. L., Murphy, R. R., Coover, M. D., and Riddle, D. L. (2004). Moonlight in miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Journal of Human-Computer Interaction*, 19(1-2):85–116.
- Calisi, D., Censi, A., Farinelli, A., Iocchi, L., Nardi, D., and Tipaldi, G. (2006). Spqr real rescue team description paper. In *Robocup 2006: Robot Soccer World Cup X*, Bremen, Germany. (Proceedings CD-ROM).
- Carpin, S., Birk, A., and Jucikas, V. (2005). On map merging. *Robotics and autonomous systems*, 53(1):1–14.
- Carpin, S., Lewis, M., Wang, J., and Balakirsky, S. (2007a). Bridging the gap between simulation and reality in urban search and rescue. In *Robocup 2006: Robot Soccer World Cup X*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg. (to appear).
- Carpin, S., Lewis, M., Wang, J., Balarkirsky, S., and Scrapper, C. (2007b). Usarsim: a robot simulator for research and education. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1400–1405.
- Carpin, S., Stoyanov, T., Nevatia, Y., Lewis, M., and Wang, J. (2006a). Quantitative assessments of usarsim accuracy. In *Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, pages 111–118.
- Carpin, S., Wang, J., Lewis, M., Birk, A., and Jacoff, A. (2006b). High fidelity tools for rescue robotics: results and perspectives. In *Robocup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 301–311. Springer Berlin / Heidelberg.
- Casper, J. and Murphy, R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(3):367–385.
- Dias, M., Zlot, R., Zinck, M., and Stentz, A. (2004). Robust multirobot coordination in dynamic environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 4, pages 3435–3442.
- Duda, R. and Hart, P. (1972). Use of the hough transform to detect lines and curves in pictures. *Communications of the ACM*, 15:1–15.
- Grisetti, S., Stachniss, C., and Burgard, W. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2432 – 2437.

- Howard, A., Sukhatme, G. S., and Matarić, M. J. (2006). Multi-robot mapping using manifold representations. *Proc. of the IEEE*, 94(7):1360–1369.
- Jennings, J., Whelan, G., and Evans, W. (1997). Cooperative search and rescue with a team of mobile robots. In *Proc. of the IEEE Int. Conf. on Advanced Robotics (ICAR)*, pages 193–200.
- Kitano, H. and Tadokoro, S. (2001). Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52.
- Kleiner, A., Dornhege, C., Kuemmerle, R., Ruhnke, M., Steder, B., Nebel, B., Doherty, P., Wzorek, M., Rudol, P., Conte, G., Durante, S., and Lundstrom, D. (2006a). Robocuprescue - robot league team rescuerobots freiburg. In *Robocup 2006: Robot Soccer World Cup X*, Bremen, Germany. (Proceedings CD-ROM).
- Kleiner, A., Prediger, J., and Nebel, B. (2006b). Rfid technology-based exploration and slam for search and rescue. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4054–4059.
- Kleiner, A. and Ziparo, V. (2006). Robocuprescue - simulation league team rescuerobots freiburg (germany). In *Robocup 2006: Robot Soccer World Cup X*, Bremen, Germany. (Proceedings CD-ROM).
- LaValle, S. M. (2006). *Planning algorithms*. Cambridge Academic Press, Cambridge; New York.
- Lu, F. and Milios, E. (1994). Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994*, pages 935–938.
- Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Auton. Robots*, 4:333–349.
- McFarlane, D. C. and Latorella, K. A. (2002). The scope and importance of human interruption in human-computer interaction design. *Journal of Human-Computer Interaction*, 17(1):1–61.
- MobileRobotsInc. (2006). <http://www.mobilerobots.com>.
- Murphy, R., Casper, J., Micire, M., and Hyams, J. (2000). Mixed-initiative control of multiple heterogeneous robots for usar. Technical Report CRASAR-TR2000-11, Center for Robot Assisted Search & Rescue, University of South Florida, Tampa, FL.
- Nass, C. (2004). Etiquette equality: exhibitions and expectations of computer politeness. *Communications of the ACM*, 47(4):35–37.
- Pfister, S. T., Kriechbaum, K. L., Roumeliotis, S. I., and Burdick, J. W. (2007). A weighted range sensor matching algorithm for mobile robot displacement estimation. *IEEE Transactions on Robotics and Automation*. (to appear).
- Scerri, P., Xu, Y., Liao, E., Lai, G., Lewis, M., and Sycara, K. (2004). Coordinating large groups of wide area search munitions. In Grundel, D., Murphey, R., and Pandalos, P., editors, *Recent Developments in Cooperative Control and Optimization*, pages 451–480. World Scientific Publishing, Singapore.
- Slamet, B. and Pfingsthorn, M. (2006). Manifoldslam: a multi-agent simultaneous localization and mapping system for the robocup rescue virtual robots competition. Master’s thesis, Universiteit van Amsterdam.
- Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:88–124.

- Trouvain, B., Schlick, C., and Mevert, M. (2003). Comparison of a map- vs. camera-based user interface in a multi-robot navigation task. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3224–3231.
- Wang, J. and Balakirsky, S. (2007). Usarsim v 3.0. <http://sourceforge.net/projects/usarsim>.
- Wang, J., Lewis, M., Hughes, S., Koes, M., and Carpin, S. (2005). Validating usarsim for use in hri research. In *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, pages 457–461.
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '97)*, pages 146–151.
- Yanco, H. A., Drury, J. L., and Scholtz, J. (2004). Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Journal of Human-Computer Interaction*, 19(1-2):117–149.
- Ziparo, V., Kleiner, A., Nebel, B., and Nardi, D. (2007). Rfid-based exploration for large robot teams. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Rome, Italy. (to appear).