SPQR Rescue Virtual Robots Team Description Paper

D. Calisi, G. Randelli, A. Valero, L. Iocchi, D. Nardi

Dipartimento di Informatica e Sistemistica
"Sapienza" Università di Roma
Via Ariosto 25, 00185 Roma, Italy
E-mail: `<lastname>@dis.uniroma1.it`

Abstract. In this paper we describe the technical characteristics of the rescue system developed by SPQR Virtual Team for RoboCup 2008. The system is composed by three ground robots P2AT and an UAV. We analyze the whole architecture, also focusing on the new features included. Some of these are the new interface and human-robot interaction layer, the quadrotor and its coordination with the other robots for the exploration task, the a priori map of the environment and a semi-distributed mapping technique. We also show the results of some experiments to evaluate the applicability of the system to real rescue scenarios.

1 Introduction

SPQR[1] is the group of the Department of Computer and Systems Science at Sapienza University of Rome in Italy, that is involved in RoboCup competitions since 1998 in different leagues (Middle-size 1998-2002, Four-legged since 2000, Real-rescue-robots 2003-2006, Virtual-rescue since 2006 and @Home in 2006). In 2007, our team got the third place in RoboCup Rescue Virtual Robots League in Atlanta (USA). All the research activities are carried out by SPQR team within the SIED Laboratory, which stands for "Intelligent Systems for Emergencies and Civil defense".

In this paper we describe the technical characteristics and capabilities of the Rescue Robot prepared by the SPQR Rescue Virtual Robots Team for Robocup Rescue 2008 competitions in China (Suzhou). Our system is able to:

- build a dense metric map;
- safely and autonomously navigate in the environment, even executing complex maneuvers;
- explore the environment;
- detect victims and their relevant signatures.

In the rest of this document, after the presentation of the team members, we describe in the next two sections the system characteristics, focusing on the new HRI system developed and the software architecture, based on our new OpenRDK[2] environment. Following sections deal with exploration and mapping techniques implemented, sensors equipment used in USARSim and finally some applications in real contexts.

[1] www.dis.uniroma1.it/~spqrv
[2] http://openrdk.sourceforge.net/

2 Team Members

- Daniele Nardi: Advisor
- Luca Iocchi: Advisor
- Daniele Calisi: Team Leader and Operator
- Luigi Del Negro
- Alessandro Dionisi
- Gabriele Randelli
- Alberto Valero Gomez

3 Operator Station Set-up and Break-down (10 minutes)

Our base station consists on a laptop PC, running under Linux. To ensure an effective control by the human operator, a new multi-robot graphic user interface has been developed, using the services offered by OpenRDK framework [1, 5]. The multi-robot interface allows the human operator to behave as a supervisor, that is to see the global map built by the robot team, to see interesting states of the robots, and to control the robots at different levels. Nevertheless, in case of failure the operator can have direct access to the robot and can give the robot speed commands through the keyboard.

4 Control Method and Human-Robot Interface

We use a single user control program which, interfacing with RDK framework, can connect to more than one robot, giving the operator the possibility of control and supervision of the robot activities. In particular, our robot team is composed by three ground robots P2AT equipped with a SICK Laser Range Finder (with no tilting platform) and one unmanned aerial vehicle (UAV) with an INS sensor and a Hokuyo laser. In our research we've been concentrating our efforts in developing a new HRI system able to manage the multi-robot-multi-user paradigm, and therefore to improve both the speed and quality of the operator's problem-solving performance and to improve efficiency by reducing the need for supervision. Moreover, we also maintain low communication bandwidths associated with semi-autonomous control, by requesting only the relevant sensory data, as well as optimize the amount of information transmitted and presented to the operator.

4.1 Interaction System

The HRI system manages the interaction among the agents (human operator or artificial agents) and the robots. The crucial point is how to integrate the operator within the autonomous processing in order to obtain the higher performance. Its core idea is to give the control of the robot to an artificial or to the human, most capable of accomplishing the task. A Rating Service Module and a set of Performance Indicators are employed to evaluate the agents *capabilities*. A task is allocated to an agent considering his/its performance in the scenario within which the robot is navigating. The scenario is given by the Scenario Recognition

Module. The Performance Rate is calculated by the *Rating Service Module*. The Performance Indicators are a set of metrics for measuring the performance of the robot making a precise task. This way of managing the interaction allows the system to identify situations in which the supervision of an operator is required, hypothesizing from the indicators what is not working properly. The *Task Allocation Module* must choose, among the different possibilities, the one that will drive to a better global performance. A more detailed explanation of the HRI system can be found in [8][9]. When the control is assigned to the operator the autonomy level is set to *tele-operation*, while when is given to an artificial agent it is set to *autonomy*. Both human and artificial can collaborate, resulting in the safe tele-operation and shared control modes. In the *safe tele-operation* mode the system prevents the robot from colliding with obstacles. In the *shared control* mode the operator sets a target point for the robot by directly clicking on the map, which the robot tries to reach. When working in shared control or autonomy, the operator can select from three agents (*Agent Mode Panel*): *slow*, *normal* and *speedy*, which have different pre-set maximum velocities and use different heuristics to explore the environment. The HRI system tracks in real-time their reputation and performance[8] to support the operator.

4.2 Operator Interface

Our interface is designed for controlling multiple robots in structured and partially unstructured environments. Some common examples are surveillance activities, scheduled operations in power plants (e.g., nuclear power plants to deal with radioactivity risks), rescue operations, and so on. Its scope is to be able to control a robot team in such situations, dealing mainly with exploration, navigation and mapping issues. The interface was developed analyzing the main difficulties and requirements of a single operator managing several agents concurrently [10]. Thus, its main purpose is to enhance the operator's performance of complex tasks, with a comprehensive global overview of the whole team, and supplying all the necessary tools to control each robot. The global information, is always visible on the screen, to allow a monitoring of the entire robot team, while controlling each individual robot. For the robot's individual information, we have decided not to overload the screen (and operator) with too much data.

The interface shown in figure 1 can be divided into two parts: the topmost panel is the *Active Robots Panel*, where the user can switch among the robots of the team, in order to directly interact with an individual unit. If a robot is added to the team the operator can easily connect with it. The rest of the window contains all the information relative to the selected robot and the robot team. To this point we have not yet integrated the UAV into the interface. At any case, we are durrently working on this interface, and important changes could be applied prior to the competition.

We're working on integrating with the new user interface the human body detection module which is present within RDK framework. Finally, the interface can now manage the a priori map, showing it overlapped on the map built by robots, thus allowing the operator to easily detect most difficult areas and better move the robots.

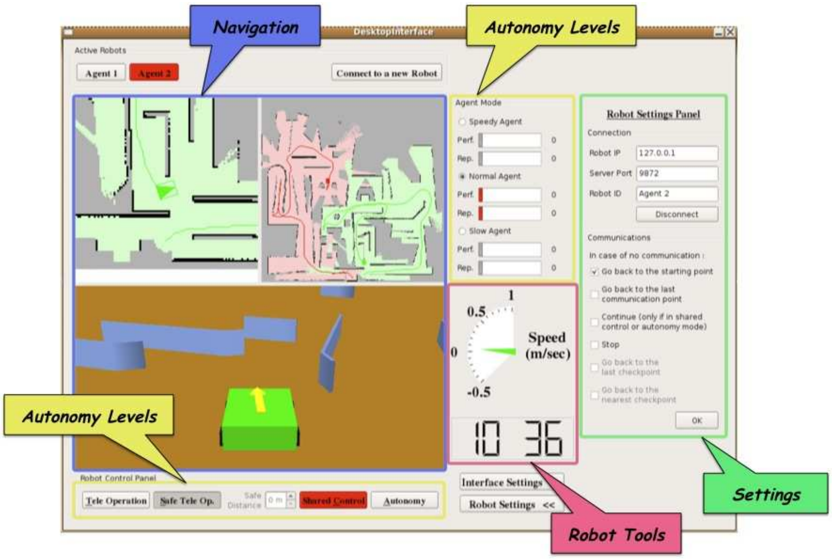

Fig. 1. SPQR Interface with all its main components. The blue rectangle stands for *navigation* functionalities, which include a local and global view of the map, as well as pseudo-3D view. The yellow boxes are for *robot control*, with four motion modes and three different heuristics for autonomous navigation. The red area is for *robot tools* and the green one for *settings*.

5 Software Architecture

The HRI described in section 4 is just the top of our software architecure, as it connects to our robotics development framework, called OpenRDK. The latter is based on a set of modular components that interact with each other via a shared repository. Each component is responsible for a basic functionality of the robotic platform, e.g., navigation, localization, mapping etc, thus acting as a middleware layer between the interface and the simulators (Player/Stage or USARSim) or the real robots. OpenRDK allows for easily interchange modules with similar functionality, switch from simulation to real robot and sensor, and successful team development. Modules are loosely connected to each other and can be scheduled independently and with different priorities. The modules interact and communicate with each other using a blackboard-type data repository, these shared data can be visualized and modified by the remote console and partially shared among different processes/computers.

6 Coordinated Multi-Robot Exploration

The exploration strategy with relative coordination aspects [4], depends on the autonomy level of the robots at a certain moment. We will refer to the same four modes introduced in section 4. In *Autonomy* mode robots follow a distributed coordination, where a task assignment strategy is employed to allocate robots to different tasks. Each robot maintains a structure containing the tasks known to all the agents. Each robot locally computes the current target points to reach, and verifies that they are not within the current tasks already known to the system. To compare the tasks a simple nearest neighbour technique is used. Each robot sends in broadcast the new tasks to all team mates, computes its utility function for all the tasks present in the system and broadcasts the function values to all other team mates. Each robot computes autonomously the best allocation of robots to targets and then execute the best task according to the chosen allocation. The best allocation is computed considering all possible assignment of robots to tasks, and choosing the one that maximises the sum of utility functions. These latter are based on specific parameters related to the task execution (e.g. distance to travel). When the *Shared Control* mode is selected, the robot waits for a goal point to be provided by the operator. When a goal point is provided, the robot will autonomously navigate towards the specified point using the approach described in. When it reaches the goal point provided by the human operator, it will stop waiting for another goal point. Finally, when the *Teleoperation/Safe Teleoperation* mode is selected the robot will not try to act proactively, waiting for low level commands from the operator. Each robot will then perform the task assignment strategy specified in the previous section, treating the goal point sent by the human operator as a high priority task. Task assignment also allows the human operator to select interesting areas for exploration, without assigning them to a particular robot. In this case, after reaching the human goal target, the robot will keep executing the frontier base exploration from the current position, as described in [6].

7 Map Generation and Printing

For building a consistent global map, we implemented a centralised coordinated SLAM approach that merges the local maps from all robots, while each robot builds its own local map integrating LRF output and encoder information. Together with the map, we can estimate the path taken by each robot during the exploration process. As for the UAV, at the moment it doesn't contribute to the map generation, because the Hokuyo laser is used for obstacle avoidance. Moreover, because of limited computational resource, it's impossible to have on-board image processing. So the UAV has a role only in exploration and victims detection tasks, especially in areas with bad mobility conditions for the UGVs. Both the global map and local ones can be displayed on the operator interface while the robots operate, together with the robot position in the environment[3, 2]. The map is finally converted into a bitmap image. On such a map the identified victims can be annotated to produce a final report (Figure 2).

Notice that while the global map is estimated by a centralised process each robot maintains a local map built autonomously. Therefore if a communication breakdown interrupts the link between one robot and the central station, the robot is still able to perform its tasks reasoning on its local map. The global map is used only by the human operator to monitor the mission execution and to control the robots. We also built an elevation map (used inside USARSim), in order to take care of the obstacles not seen by a laser rotating just on an horizontal plane. We implemented a technique based on two lasers acting on

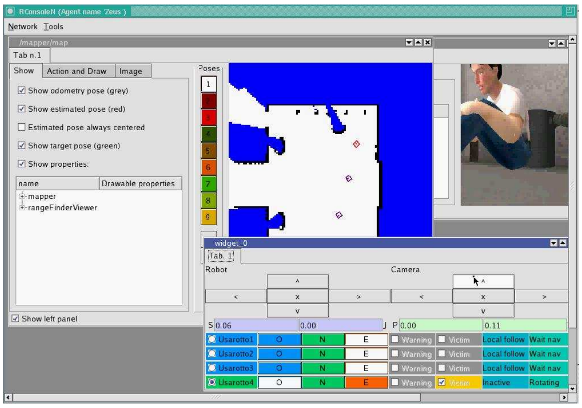

Fig. 2. An example of the generated map, including the victims found

the sagittal plane with different orientations to derive some information about obstacles heights. The information given by the elevation map allows to identify areas with too many mobility problems for UGVs and the chosen heuristics is to avoid to explore such areas. All obstacles with height higher than a set threshold are included in the map, identified by a different colour.

Fig. 3. Our AscTec UAV with the Hokuyo LRS installed on its top.

8 Sensors for Localization and Navigation

Safe navigation for ground robots is achieved using an integrated approach of two SICK Laser Range Finders (LRF). A single laser would be sufficient for planar maps, but to deal with 3D obstacles, such as stairs or pits, we need to use a second LRF, acting with different orientation onto the sagittal plane. As for the UAV, its localization is based on an INS sensor and we also aim at using a Hokuyo laser range finder for obstacle avoidance [7]. This set up has been also tested on a real AscTec quad-robot (see figure 3).

9 Team Training for Operations (Human Factor)

The Graphical User Interface is very user friendly, as it is developed for non expert usage. Not more than one week training for a computer expert user is necessary to command the entire functionalities of robots from the user interface.

10 Possibility for Practical Application to Real Disaster Site

The whole system is also implemented and tested on real robot units. We validated our approach with two mobile platforms: a P2DX equipped with an Hokuyo Laser Range finder, and a P2AT equipped with a SICK Laser Range Finder (fig.4).

Fig. 4. SPQR Team Real Robot

The experiments have been conducted in the arena set up in our lab.

Figure 5 shows the maps created by the robots during their mission. The environment to explore is 7×6 square meters, and the two robots completed the exploration in 10 minutes approximately. ¿From left to right it is possible to see the initial situation, a snapshot during the exploration process and the final map. The P2DX is represented with a circle and the P2AT is represented with a square. In the maps it is possible to see the current tasks the robots are allocated to (crosses in the map). Robots performed a coordinated supervised exploration. Giving high level advices the operator was able to efficiently control the system, nicely spreading the two robots.

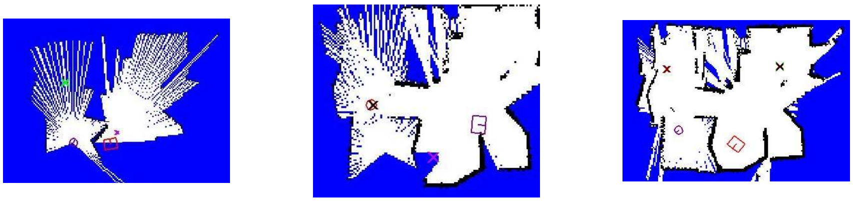

Fig. 5. Cooperative exploration sequence

Figure 6 shows the two maps of the single robot. These are the maps the two robots maintain locally. As it is possible to see the overlapping among the two maps is minimal, as it is desirable in a multi-robot exploration task. On the other hand, a bigger overlap between the two maps would have been beneficial for the cooperative SLAM process, and would have produced a better quality global map. In this work, we focused on minimising the exploration time rather than having a better quality map. We also organized a four-day experiment context, in order to evaluate the usability of the new interface, both in real and virtual environments. Subjects had to explore an unknown simulated environment controlling two robots, thus allowing to observe several parameters, like the ease of switching robot, the chosen exploration strategy, the amount of area explored, the ease of controlling the robot in clustered areas, and so on. Then they moved to the real environment, controlling just one P2AT in narrow spaces.

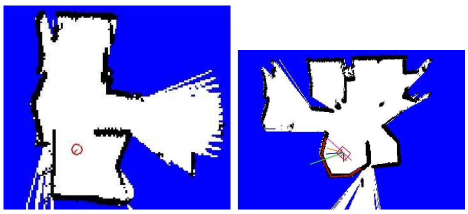

Fig. 6. Maps of each single robots: P2DX left and P2AT right

11 Conclusion

Among the future works that we have been attending at SIED Laboratory, we are focusing on coordination methods between UAV and the ground vehicles (UGV), at first considering just one UAV, later using a team of these vehicles, coordinated with the ground robots. Due to the strong constraints on UAV's payload, we have been analyzing scenarios where UGVs have the full equipment for victims recognition, while the aerial vehicle just a partial one. We also developed a new operator interface, comprehensive of a pro-active human-robot interaction system, that we are going to use for the first time in a RoboCup competition.

Further development on this interface will add a webcam module and a joystick controller. We're also planning to improve our map merging subsystem using a partially distributed algorithm, rather than a centralized one. Finally, we would like to develop a semi-automatic tool for report generation, to relieve the operator from this task.

References

1. Farinelli A., Grisetti G., and Iocchi L. SPQR-RDK: a modular framework for programming mobile robots. In D. Nardi et al., editors, *Proc. of Int. RoboCup Symposium 2004*, pages 653–660, Heidelberg, 2005. Springer Verlag. ISBN: 3-540-25046-8.
2. Stachniss C., Grisetti G., and Burgard W. Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loops. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
3. Stachniss C., Grisetti G., Haehnel D., and Burgard W.. Improved rao-blackwellized mapping by adaptive sampling and active loop-closure. In *In Proc. of the Workshop on Self-Organization of AdaptiVE behavior (SOAVE)*, 2004.
4. La Cesa S., Farinelli A., Iocchi L., Nardi D., Sbarigia M., and Zaratti M. Semi-Autonomous Coordinated Exploration in Rescue Scenarios. In *Proc. of Int. RoboCup Symposium 2007*, Atlanta, 2007.
5. Calisi D., Censi A., Iocchi A., and Nardi D. OpenRDK: a modular framework for robotics software development. *Technical Report*, 2008.
6. Calisi D., Censi A., Iocchi A., and Nardi D. Autonomous Exploration for Search and Rescue Robots. In *Proc. of Int. Conf. on Safety and Security Engineering (SAFE)*, 2007.
7. Hoffmann G. M., Huang H., Waslander S. L., and Tomlin C. J. Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment In *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*, South Carolina, 2007.
8. Valero A., Matia F., Mecella M., and Nardi D. Adaptative Huma-Robot Interaction for Mobile Robots. In *Proc. of the IEEE Int. Symposium on Robot and Human Interactive Communication*, Munchen, 2008.
9. Valero A., Matia F., Mecella M., and Nardi D. Pro-active Interaction for Semi-autonomous Mobile Robots. Introducing the Task Allocation Service. In *DVD of the IEEE ICRA Workshop: New Vistas and Challenges in Telerobotics*, California, 2008.
10. Humphrey C. M., Henk C., Sewell G., Williams B. W., and Adams J. A. Assessing the scalability of a multiple robot interface In *proc. of the Second ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, Virginia, USA, 2007.