PoAReT Team Description Paper

Francesco Amigoni, Alain Caltieri, Riccardo Cipolleschi, Giovanni Conconi, Michele Giusto, Matteo Luperto, and Mladen Mazuran

Artificial Intelligence and Robotics Laboratory, Politecnico di Milano, Italy
`francesco.amigoni@polimi.it`

Abstract. This paper introduces the main features of the PoAReT (Politecnico di Milano Autonomous Robotic Rescue Team) system for the Virtual Robot competition of the Rescue Simulation League at RoboCup 2012.

1 Introduction

This paper summarizes the main features of the *PoAReT* (*Politecnico di Milano Autonomous Robotic Rescue Team*) team for the Virtual Robot competition of the Rescue Simulation League at RoboCup 2012. The system is developed by six MSc students in Computer Engineering at the Politecnico di Milano. Further information about the team, including a link to the source code and the list of members with their roles is available at `http://home.dei.polimi.it/amigoni/research/PoAReT.html`.

In developing PoAReT, we push along the *autonomy* axis, attempting to equip the robotic system with methods that enable its autonomous operation for extended periods. At the same time, the role of human operator is not neglected, but is empowered by the autonomous features of the system.

Besides the base station, our PoAReT system is composed of P3AT mobile platforms, equipped with laser range finders (with a 360° horizontal field of view around each robot, possibly obtained with two laser range finders) and with a camera. Laser range finders are used to build a geometrical map of the environment that is represented with two sets of line segments. The first set contains the line segments that represent (the edges of) perceived obstacles. The second set contains the line segments that represent the *frontiers*, namely the boundaries between the known and the unknown portions of the environment.

The main cycle of activities of the PoAReT system is: (a) building a geometrical map of the environment composed of line segments, (b) selecting the most convenient frontiers to reach, and (c) coordinating the allocation of robots to the frontiers. A distinguishing feature of our system is that it (d) maintains a *semantic map* of the environment that labels areas of the geometrical map with human-like names, like 'room' or 'corridor'. At the same time, the system performs (f) the detection of victims on the basis of the images returned by the onboard cameras and (g) the interaction with the human operator via the user interface. Exploiting semantic maps, the PoAReT system is expected to exhibit a high level of autonomy in selecting the interesting frontiers to reach and in allocating them to available robots. Also the interaction with the user could be improved by availability of semantic maps, enabling the interpretation of human-level commands like "always send two robots along corridors".

In the following of this paper, we first describe the overall architecture of the PoAReT system and then the individual modules that perform activities (a)-(g).

2 Overview of the PoAReT System

The architecture of our system is organized in two different types of processes, one related to the base station and one related to the mobile robots, to have a clear separation between their functionalities.

The base station embeds the user interface module. It displays data to the user and accepts commands from the user to control the spawned robots. The base station process can spawn new robots in USARSim: for each robot, a new independent process is created and started. The processes of the base station and of the robots communicate only through WSS and do not share any memory space, as required by the rules for the competition. A distance vector routing protocol [1] is implemented to deliver messages. This protocol calculates the routing tables for every robot in a dynamic and distributed way, considering the performance of a connection in terms of the number of hops for the messages. Messages can be delivered either in a non-reliable or in a reliable way, used for critical information and implemented with the usage of message queues. Although in principle there is no need to maintain a direct connection between robots and base station (robots explore autonomously and, when connection is re-established, they can report to the base station and share collected information with other robots), the routing protocol maintains indirect connectivity between robots and base station in order to extend the operative range of the human operator.

The robot process is structured in seven different modules, each one related to a high-level functionality: motion control, path planning, SLAM, semantic mapping, victim detection, exploration, and coordination. Almost all of these modules are threads that communicate through a queue system. The main of the above modules are described in the following sections. Here, we briefly discuss the motion control module, which is straightforward, given the locomotion model of P3AT, and the path planning module. Path planning can be invoked in three cases: to reach a position with a path that lies entirely in the known space (usually, the position is a frontier between known and unknown space), to reach a position with a path that traverses unknown parts of the environment, and to cover a known portion of the environment (e.g., a room) to search victims. The algorithms we use are A* in a graph whose vertices are the past poses of the robot and edges are the elementary (visibility) paths between them, RRT [2], and one of the algorithms for coverage [3], respectively.

Fig. 1 shows the PoAReT system architecture. The base station module is in green, while the mobile robot modules are in yellow.

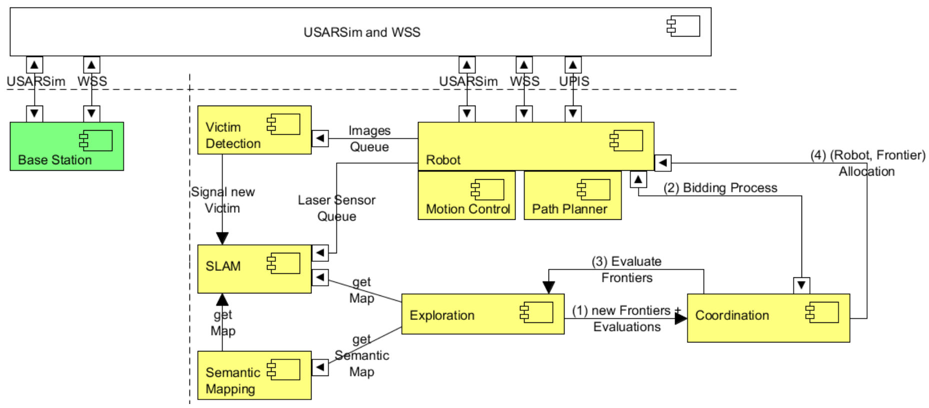

Fig. 1. PoAReT system architecture

3 SLAM

In our team, the simultaneous localization and mapping (SLAM) problem is tackled by adopting a feature-based method similar to that described in [4]. Differently of that approach, however, our system is based on the FastSLAM 2.0 paradigm [5] rather than on EKF, in order to get some important advantages, including: multiple data association, lower asymptotic time complexity, and full robots' path posterior (past poses are used for navigation when backtracking). From the other hand, FastSLAM has known problems with loop closures. However, these can be mostly ignored in search and rescue applications where the robots are not likely to pass through the same location more times.

The map is represented as a set of line segments that are extracted (using the split and merge algorithm [6]) from laser range scans received as input, together with odometric information about the poses where the scans have been taken. Line segments are thus treated as the features for the probabilistic filter. Use of line segments instead of more common grids provides a more compact representation of the environment that can be easily communicated to the base station.

The measurement step associates the line segments of a scan to the the linear features in the map (with respect to distance measures, such as those described in [7] or [8]) and executes an Iterative Closest Line (ICL) algorithm (like [8]) with constraints on the maximum rotation and on the maximum translation to provide a per-particle measure, which is then integrated in the particle filter. All the line segments of a scan are added to the filter; periodically a test is carried out to determine whether there is enough evidence to support the hypothesis of two previously associated line segments being in fact the same; if so, they are merged. The other steps of the FastSLAM procedure are straightforward.

Besides a line segment-based representation of the map, the SLAM module keeps also track of all the robots' poses, past and current, of the victims' locations (distance measurements to the victims are confirmed by the human operator), and of the line segments that constitute frontiers between known and unknown space (which are used by the exploration module).

4 Semantic Mapping

The semantic mapping module performs a semantic classification of places and works in parallel with the SLAM module with the aim of improving performance of the exploration and coordination modules. This module takes as input the line segment map of a discovered indoor environment (updated by the SLAM module) and tries to extract more information than the basic geometrical features, exploiting prior knowledge on the typical structure of buildings. Our approach extends that presented in [9] and [10] to line segment maps.

Firstly, the mapped area is divided into single rooms, identifying the area that belongs to each room and the doorways that divide the rooms. This is done applying an AdaBoost classifier to the scan provided by the laser sensor. As a result of this step, the parts of the environment are classified into two basic categories: room and doorway. With this information, the space portion marked as *room* is divided into different parts representing every single room separately. Later, each room is classified according to its own characteristic. A simple classification can label a room as a *small room*, a *large room*, or a *corridor*, while a more complex one, based on features of the room itself, could describe the function of the room as a bathroom or an office. The *corridor* and the *hallway* concepts, although very simple to understand for a human being, can be very important in a robotic rescue environment, because corridors and hallways can be explored first, leading the robots to discover many other rooms in a short amount of time.

The semantic map is represented as a graph, derived from the geometrical map, and is exploited by the exploration and coordination modules. For example, these modules can estimate from the semantic map how much time and effort are required to explore a portion of the environment, according to the presence of corridors and rooms. The semantic map is also used by the user interface module to give to the user an immediate and intuitive feedback on the behavior of the robots in the environment.

5 Exploration

The exploration module selects new frontiers to explore, in order to discover the largest possible amount of the environment within the time allowed in the competition. This module obtains from the SLAM and the semantic mapping modules the geometrical and the semantic maps, respectively. Then, it evaluates the frontiers by assigning them utilities and, finally, calls the coordination module to find an allocation of robots to the frontiers. The module also exposes a function that can be called by the coordination module and that evaluates a given set of frontiers from the point of view of a robot.

In the exploration module, we employ an exploration strategy that tries to take advantage from the geometrical and semantic information gathered by the robots. We take inspiration from [11], where the authors achieve a good exploration performance by distinguishing if the robot is in a hallway or in a room. In our system, we integrate this semantic information into a framework, called Multi-Criteria Decision-Making (MCDM), that is described in [12]. MCDM is a flexible decision-theoretical approach: we can add or remove criteria and we can also modify the criteria weights at runtime.

Specifically, to calculate the utility of a frontier f for a robot r, we use the following criteria (similar to those used in [13]): the distance of r from f, the expected information gain at f (an estimate of how much new area r will perceive when at f), the probability of keeping the communication between r and another robot when r will be in f, and the battery level for r. Furthermore, we weight the utility of f with respect to the semantic information associated to f. For example, if f leads to a corridor, its utility will be increased.

6 Coordination

The coordination module is responsible of allocating tasks to the robots. As most of the other modules it is a thread. However, differently from other modules, it is only called whenever it becomes necessary to allocate tasks among robots. The mechanism we use to allocate tasks is market-based and sets up auctions in which tasks are auctioned to robots [14]. Auctions provide a well-known mean to bypass problems like unreliable wireless connections or robot malfunctions.

We consider three different tasks.

– **Reach frontiers**. As explained in the previous section, robots must explore the environment in order to discover it. Auctions that allocate these tasks work on frontiers. Any robot evaluates the frontiers that are auctioned (see Section 5) and submits bids accordingly. From these evaluations, frontiers are assigned to the best suited robots. Thanks to the semantic classification of places it is also possible to cope with the necessity to assigning the same frontier to more robots, for example when that frontier corresponds to a corridor that is expected to be connected to a large number of rooms.
– **Search for victims**. During the exploration, it is possible that an area that has been sensed by the laser range finders has not been completely covered by the cameras, due to the different fields of view of the two sensors. Hence, once an area is mapped, it has to be searched for victims. For this, we use a second kind of auction, where the tasks are the areas that must be be searched for victims by using the cameras. The bids for the areas are calculated according to the following criteria: the size of the area, the battery level of the robot, and the distance between the robot and the area.
– **Park close to victims**. The final task is strictly related to the competition rule that requires to park a robot close to a victim in order to have it scored. Hence, for every found victim, there is always a corresponding robot with the task to be parked enough close to the victim at the end of the time allowed in

the competition. In this case, auctions are used to allocate victims to robots (e.g., based on distance), thus allowing robots that have found victims to move again around to explore other areas of the environment. The only constraint is that a robot should be close to the assigned victim at the end of the competition.

7 Victim Detection

The victim detection module is responsible for searching victims inside the competition environment. It works by analysing images coming from the robots' cameras and classifying them according to the presence or absence of victims. In the first case, the victim detection module signals the human operator, who marks the location of the victim in the map.

We have chosen to implement a skin detector using HSV (Hue, Saturation, Value) color space, followed by a version of the Viola-Jones algorithm [15], a well-known image analysis method already used by many teams in previous editions of the competition. It is based on the value of simple features defined on the images, allowing object recognition (in this case, skin and facial recognition) after a training over sample images. The Viola-Jones algorithm uses a cascade of simple classifiers, each one of them is responsible to analyse a single feature and decide whether or not the image contains a victim according to that feature. This approach has been proved to be very fast and to reach high levels of accuracy. For implementing this module we use components already available through the OpenCV library framework [16].

8 User Interface

The PoAReT User Interface (UI) allows a single operator to control a relatively large group of robots in an easy way. It reduces the workload of the operator and increases her/his situation awareness. These two objectives are reached by our UI through a mixed-initiative approach [17].

The PoAReT UI allows a single operator to control the system in different ways (from the most to the least automated):

- It is possible to send high level commands to robots, like "explore along a direction" or "cover a room". The system handles autonomously such commands and behave intelligently to reach the setted goal, exploiting the semantic map.
- If the system is not able to handle the situation, the operator can take control of the single robot, controlling it by mean of waypoint targets. With this kind of approach only the basis of navigation is automated.
- In particularly dangerous or difficult situations, the operator can exclude all the automated processes and simply teleoperate the robot manually.

This multi-level approach, except the first level, has been previously explored by [18]. We enhance it with the introduction of high-level commands that limit the cognitive cost for the operator and exploit the full automation of the system.

The UI is also able to filter notifications arriving from the modules, based on the operator's preferences, her/his past behaviour, and situation parameters. The approach is a simplification of the architecture presented in [19] with the difference that the PoAReT UI tries to mantain the human operator always in the control loop, never treating her/him as a simple supervisor in order to preserve the situation awareness.

Figure 2 shows the UI in action. Clockwise from the top-left of the image: camera thumbnails (for monitoring and selecting robots), teleoperation commands (for directly controlling the selected robot), information about the selected robot (camera view, battery level, connected robots), interactive map (used for issuing control actions and integrated with both semantic mapping and victim detection). Finally, the message manager that shows, sorts, and filters messages from the robots is displayed at the bottom of the image.

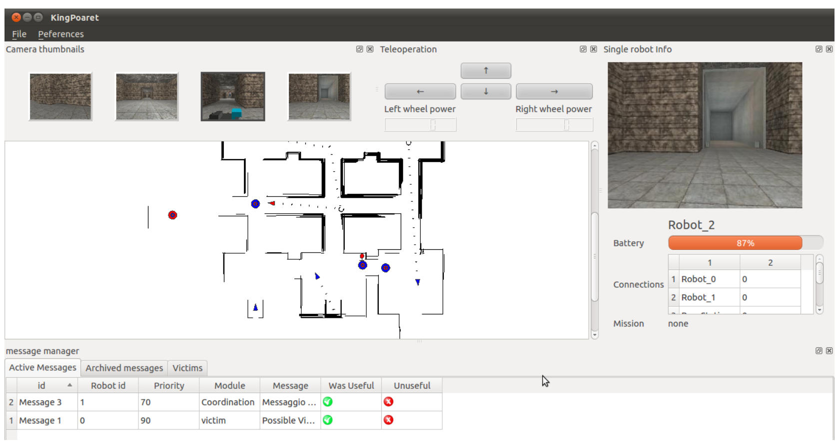

Fig. 2. PoAReT user interface

9 Conclusion

This paper has surveyed the main features of the PoAReT system for the Virtual Robot competition of the Rescue Simulation League at RoboCup 2012. Among the most innovative contributions, we remark the building of a geometrical map based on line segments and the building of a semantic map that can better inform the exploration and the coordination processes and that can raise the abstraction level at which human operators give commands to the robotic system. Overall, our contributions are intended to increase autonomy of robotic systems for rescue operations.

Acknowledgement

The PoAReT team gratefully thanks the Fondazione Banca del Monte di Lombardia for the financial support.

References

1. Comer, D.: Internetworking with TCP/IP - Volume 1. Addison-Wesley (2006)
2. LaValle, A., Kuffner, J.: Rapidly-exploring random trees: Progress and prospects. In Donald, B., Lynch, K., Rus, D., eds.: Algorithmic and Computational Robotics: New Directions. (2001) 293–308
3. Choset, H.: Coverage for robotics: A survey of recent results. Annals of Mathematics and Artificial Intelligence **31**(1-4) (2001) 113–126
4. Garulli, A., Giannitrapani, A., Rossi, A., Vicino, A.: Simultaneous localization and map building using linear features. In: Proc. ECMR. (2005) 44–49
5. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press (2005)
6. Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. Autonomous Robots **23**(2) (2007) 97–111
7. Elseberg, J., Creed, R., Lakaemper, R.: A line segment based system for 2D global mapping. In: Proc. ICRA. (2010) 3924–3931
8. Li, Q., Griffiths, J.: Iterative closest geometric objects registration. Computers & Mathematics with Applications **40**(10-11) (2000) 1171–1188
9. Pronobis, A., Martinez Mozos, O., Caputo, B., Jensfelt, P.: Multi-modal semantic place classification. International Journal of Robotics Research **29**(2-3) (2009) 298–320
10. Martinez Mozos, O.: Semantic Labeling of Places with Mobile Robots. Springer-Verlag (2010)
11. C. Stachniss, O.M.M., Burgard, W.: Speeding up multi-robot exploration by considering semantic place information. In: Proc. ICRA. (2006) 1692–1697
12. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. Autonomous Robots **31**(4) (2011) 401–417
13. Visser, A., Slamet, B.: Including communication success in the estimation of information gain for multi-robot exploration. In: Proc. WiOPT. (2008) 680–687
14. Zlot, R., Stentz, A., Dias, M.B., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proc. ICRA. (2002) 3016–3023
15. Viola, P., Jones, J.J.: Robust real-time face detection. International Journal of Computer Vision **57**(2) (2004) 137–154
16. OpenCV (Open Source Computer Vision): `http://opencv.willowgarage.com/wiki/`.
17. Wang, J., Lewis, M.: Human control for cooperating robot teams. In: Proc. HRI. (2007) 9–16
18. Nevatia, Y., Stoyanov, T., Rathnam, R., Pfingsthorn, M., Markov, S., Ambrus, R., Birk, A.: Augmented autonomy: Improving human-robot team performance in urban search and rescue. In: Proc. IROS. (2008) 2103–2108
19. Petersen, K., von Stryk, O.: Towards a general communication concept for human supervision of autonomous robot teams. In: Proc. ACHI. (2011) 228–235