eROBOTICA-VR Team Description Paper

Mikael P.B. Themoteo, Yago L.M. Silva, Pedro Lira, Felipe H.N. Nascimento, Fernando J. Mendiburu, Gabriela M.S. Araújo, Thiago de F.O. Araújo, Marcos R.A. Morais, and Antonio M.N. Lima

Department of Electrical Engineering - DEE
Universidade Federal de Campina Grande - UFCG
Rua Aprígio Veloso, 882 - Bairro Universitário
58429-900 Campina Grande, Paraíba, Brazil

Abstract. This paper describes the approach adopted by the eROBOTICA-VR team for participating in the Rescue Simulation League of the Robocup Rescue Competition held in the Robocup 2014.

Keywords: mobile robot exploration, mobile robot navigation, mobile robot localization, rescue simulation, victims detection, environment mapping

1 Introduction

Disaster management and urban search and rescue can be characterized as current, relevant and quite challenging research topics. Design and deploy an autonomous robot team for rescuing human victims in disaster scenarios is an open and quite complex problem for which several solutions have been proposed and tested within the scope of the RoboCup rescue leagues since 2001. The rules adopted for the Rescue Simulation League have evolved since 2001 however the fundamental idea of the competition is to create a controlled virtual environment that replicates, up to a certain similarity level, the difficulties present in a real life situation for allowing a fair comparison between the solutions proposed by the competing teams. The basic tool used in the rescue simulation league is USARSim (Unified System for Automation and Robot Simulation) which a high-fidelity simulation of robots and environments based on the Unreal Tournament game engine [2, 3].

Figure 1 shows the conceptual model for the competition environment. There are M victims to be found by a group of N robots controlled by a single operator base station. Human supervision is provided by the user interface. Although not clearly stated in the competitions rules, the robots instantiated in the virtual disaster scenario are just the hardware for sensing, actuating and communicating, all the computational power for the robots and the base station is emulated by the computational platform provided by the competing team. Although those CPUs are being emulated in the same computational platform no communication is allowed within the scope of the team computational platform. All the messages exchanged between robots and base station must necessarily flow wirelessly within the virtual disaster scenario. In the virtual disaster scenario, as in the real world, the wireless communications are subjected to noise and bounded by the wireless network coverage area.

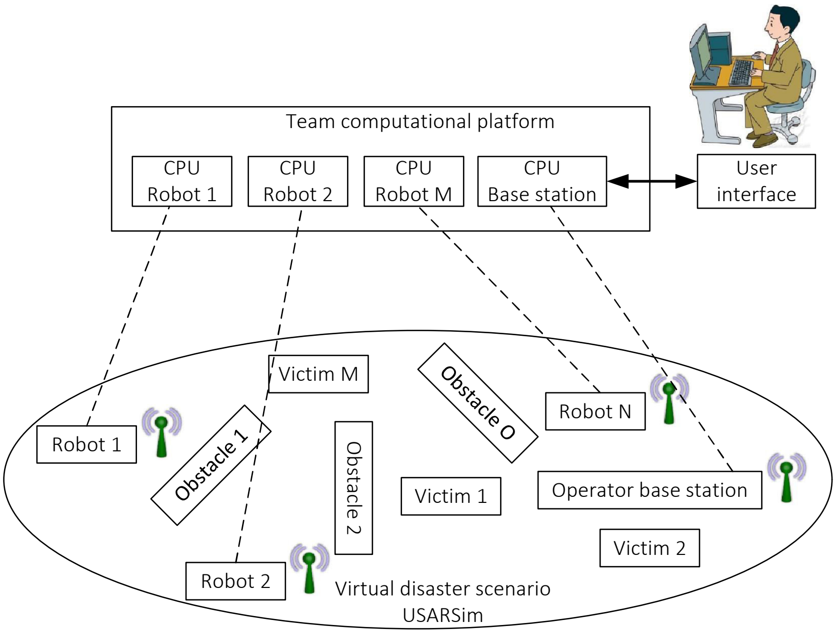


Fig. 1. Conceptual model for the competition environment. One human operator is allowed to supervise and to execute some restricted teleoperation tasks.

This paper presents the e-Robótica solution overview designed to the virtual rescue tasks and also discuss some the preliminary results obtained so far. The solution being proposed in this specific competition is based on the ideas already explored in the past [4, 5].

2 System Overview

The solution proposed by the e-Robótica addresses the different problems involved in the disaster scenario. Each robot has its own control program running on its specific processor which is assumed to have as much computational power and memory as required to handle its sensing, communicating and actuating subsystems. The robot control program will stay in one of the following operating modes:

- In the normal mode, the robot is located within the antenna coverage area of the base station and thus it may exchange information with the base station.
- In the remote mode the robot is located within the antenna coverage area of a neighbour robot and thus relies on it to exchange informations with the base station.
- In the isolated mode the robot is located outside the antenna coverage area of any other robot as well as the base station and thus must exhibit a completely autonomous behaviour.

There will be a specific set of behaviors [1] tailored for the specific robot operating mode, e.g., mapping, exploring, wandering, avoiding obstacles, and locating victims.

The base station has its own control program for interacting with all the robots and with an human agent. Similarly, it is assumed that there are no limits in terms of computational power and memory for processor that runs the based station control program. The base station will stay in one of the following operating modes:

- In the fully-autonomous mode the base station operates autonomously by exchanging information with the robots to maximize the sucess of the rescue mission in terms of the explored area and number of identified victims.
- In the semi-autonomous mode the base station operates exchanging information with the robots, however some high level decisions regarding the rescue mission are bounded by the actions of the human operator; a graphical user interface is provided to simplify such human interaction, specially for the victims identification task.

In both modes the software of the base station must keep track of an updated world representation (disaster scenario) by integrating the informations received from the robots. By such integration it is possible to avoid repeated exploration of the same area by more than one robot as well as to augment the knowledge of the partial world viewed by a robot by broadcasting information about the neighbouring areas to its current location.

To successfully implement such a complex system a proper framework is required. Such framework must allow the integration of heterogeneous tools, and provide ways to re-use already established technologies aiming to ease further development in the robotics field. Robot Operating System [33] encompass all of the previous features and thus was adopted as backbone of the eROBOTICAVR solution.

3 Communications and energy saving

The communication between the base station and the robots as well as between the robots will be provided by wireless data communication link. Thus it will be necessary to create a self-organizing and self-configuring mobile ad hoc network. However, it is assumed that no energy is used for transmitting or receiving data.

Sensing consumes the electrical energy provided by the robot's battery. On the other hand it is assumed that driving the robot around does not demand any power. Thus, it will necessary to implement a dynamic power management system by controlling the sensors usage. This is quite important to keep the robot alive during the entire rescue mission. For this paper, we have adopted energy-efficient motion planning for robot exploration [6–8] since it avoids repeated area coverage and allows to save energy.

4 Environment exploration and path planning

The disaster scenario is at some extent a real world problem which usually is highly nonlinear, multimodal and subjected to several complex constraints. Thus, metaheuristic optimization algorithms [9] seems to be a promising approach to design the control policy for environment exploration, path planning and obstacle avoidance [10–12]. Conflicting objectives like maximize the robot explored area and minimize its energy consumption must considered in the formulation of the optimization problem. On the other hand to favor collective behaviour all robots must aim to stay within the wireless coverage of ad hoc network to be, as long as it is possible, connected to the base station. In this context learning and autonomy are crucial attributes to included in the robot control policy [13, 14, 27], specially when the is outside of the wireless network coverage area.

5 Robot platforms

The disaster scenario may include indoor and outdoor environments. Thus, two robot platforms have been chosen to compose eROBOTICAVR multirobot team, i.e.: P3AT (Pioneer 3-AT is four-wheel, four-motor skid-steer robot for all-terrain operation) and a Kenaf platform (a 6-track mobile robot platform designed for uneven terrains).

The eROBOTICAVR team objective is to have a robot group as heterogeneous as possible, both in terms of robot models and in terms of type of sensors. Addition of different robot platforms is actually being implemented.

6 Localization and mapping

One of the main problems to be solved is robot localization and environment mapping within the disaster scenario. All the robots must keep track of its own localization by maintaining local maps of the environment. Based on those local robot maps, the base station may merge all of them into a global map [15–17, 28] of the disaster scenario and thus planning a more effective rescue action. To decide which type of SLAM to adopt, several implementations of EKF and Particle filters have been studied [18–21, 24]. The solution adopted for the eROBOTICAVR was based on grid mapping combined with Rao-Blackwellized particle filters [30].

7 Victims detection

The human rescue can only start when the robot detects a victim and provides her localization. Thus, detection of victims in the disaster scenario is one of the most important tasks for a fully autonomous robotic solution. Vision, sound, distance and thermal principles have been proposed to detect victims [22–25]. Locally normalized histogram of oriented gradient (HoG) descriptors have been used for detecting humans either in regular scenarios [31] and in simulated disaster scenarios [32]. The victim detection solution adopted in eROBOTICAVR is not completely automatic, indeed the preprocessing of the image streams is done with a detector that exploits HoG descriptors to identify the potential disaster victims but the final decision about human identification is delegated to the human agent that interacts with the base station.

8 User interface

The user interface must manage the information coming from the base station and from all robots deployed in the disaster scenario allowing to visualize the current status of the world representation. Besides, this user interface must show the robot current poses and battery lifetime but mainly the estimated locations of the identified victims. In the normal and remote modes the interaction of the robots with the environment is not completely autonomous. Indeed, an human operator may interferes with the robot behavior such that less time be spent on victims search, collision avoidance, and unwanted closed loop path.

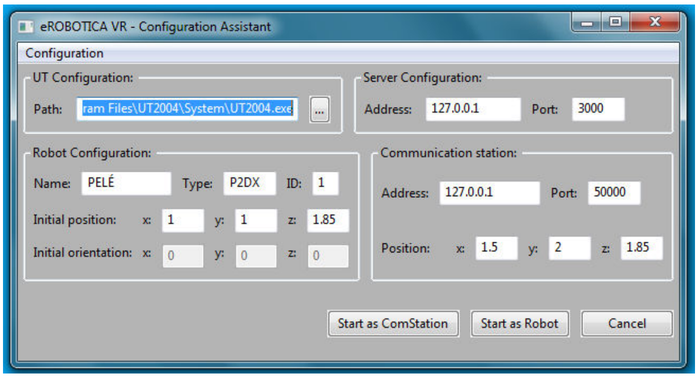
9 Preliminary test results

Several preliminary tests have been conducted with proposed eROBOTICA solution. Here we presented some selected results to demonstrate the feasibility of our proposed approach. So far, the eROBOTICAVR software has been written in C++ for the Microsoft Visual Studio. However, we are currently evaluating the possibility of using high level programming environments like, for instance, Julia[26].

Figure 2 shows a snapshot of our configuration interface running on a desktop computer in which the USARSim environment runs under Microsoft Windows XP operating system. On the other hand, Figure 3 shows, from the left to right, the robot controller, the laser range readings, the sonar range readings and a differential drive mobile robot instantiated in the virtual rescue scenario. The video showing the team skills can be found at the eROBOTICA channel by clicking here http://youtu.be/c6ZmU3HqGTQ.

10 Preliminary conclusions

In this paper, we have described the main aspects of the eROBOTICAVR approach for the Virtual Rescue Simulation competition held in Robocup 2014.



Path: ram Files\UT2004\System\UT2004.exe
Address: 127.0.0.1 Port: 3000
Name: PELÉ Type: P2DX ID: 1
Initial position: x: 1 y: 1 z: 1.85
Initial orientation: x: 0 y: 0 z: 0
Address: 127.0.0.1 Port: 50000
Position: x: 1.5 y: 2 z: 1.85

Fig. 2. Snapshot of the configuration interface for launching the robots and base station in virtual rescue scenario.

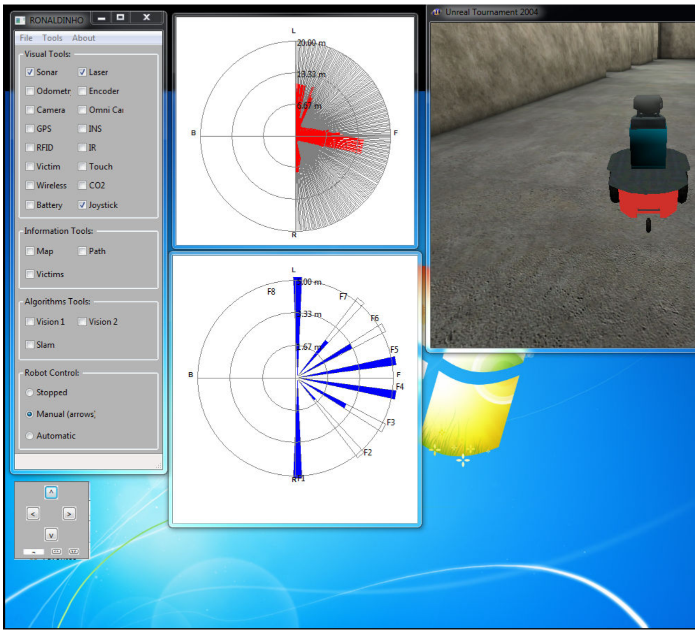
So far, the results obtained with proposed solution can be considered relatively good. However, its worth to mention that this is an ongoing work and several improvements are underway. Here follows a partial list of the foreseeable tasks: expand the number and the type of robot platforms, improve the performance of the navigation algorithm, improve the performance of the simultaneous localization and mapping algorithms and improve the quality of the visual victims detection algorithm.

11 Acknowledgments

The authors thank to PPgEE-UFCG, DEE-UFCG, CAPES and CNPq for the award of research grants and study fellowship during the course of these investigations.

References

1. R. C. Arkin, *Behavior-Based Rbotics*, MIT Press, 1998.
2. J. Wang and S. Balakirsky, “USARSim Manual v.3.1.1,” 2008.
3. J. Wang, M. Lewis, and S. Hughes, “Validating USARsim for use in HRI research,” Proceedings of the Human Factors and Ergonomics Society, 49 (2005) 457-461.
4. E. L. Colombini, A. S. Simões, T. F. O. Araújo, A. T. Neto, W. O. Guerra-Filho, G. Prado, J. P. Matsuura, and A. M. N. Lima, “Brazil-VR Team Description Paper,” Team Description Paper, 2009.
5. T. F. O. Araújo, A. T. Neto, A. M. N. Lima, W. O. Guerra-Filho, E. L. Colombini, and A. S. Simões, UFCG/ITA/UNESP Rescue Team Rescue Simulation Virtual Robots League RoboCup Brazil/LARC, Team Description Paper, 2008.
6. K. M. Thornburg and G. W. Thomas, “Robotic Exploration Utility for Urban Search and Rescue Tasks,” Journal of Computers, vol. 4, n. 10, Oct 2009.



Visual Tools:
☑ Sonar ☑ Laser
☐ Odometr ☐ Encoder
☐ Camera ☐ Omni Ca
☐ GPS ☐ INS
☐ RFID ☐ IR
☐ Victim ☐ Touch
☐ Wireless ☐ CO2
☐ Battery ☑ Joystick

Information Tools:
☐ Map ☐ Path
☐ Victims

Algorithms Tools:
☐ Vision 1 ☐ Vision 2
☐ Slam

Robot Control:
○ Stopped
● Manual (arrows)
○ Automatic

Fig. 3. Snapshot showing the robot controller interface, from left to right, the laser range readings, the sonar range readings and a differential drive mobile robot.

7. Y. Mei, Y.-H. Lu, C. S. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," Conf. Rec. IEEE/ICRA, pp. 505,511, 15-19 May 2006.
8. T. Stirling, S. Wischmann, and D. Floreano,"Energy-efficient indoor search by swarms of simulated flying robots without global information," Swarm Intelligence, vol. 4, no. 2, pp. 117-143, 2010.
9. X.-S. Yang (2011), Scholarpedia, 6(8):11472.
10. R. Simmons, "The curvature-velocity method for local obstacle avoidance," Con. Rec. IEEE/ICRA, vol. 4, pp. 3375-3382, Apr 1996.
11. J. W. Baxter and J. R. Bumby, "Fuzzy logic guidance and obstacle avoidance algorithms for autonomous vehicle control," Conf. Rec. Int. Workshop on Intell. Autonomous Vehicles, pp. 41-52, 1993.
12. J. Perez, J. Godoy, V. Milanes, J. Villagra, and E. Onieva, "Path following with backtracking based on fuzzy controllers for forward and reverse driving," Conf. Rec. IEEE/Intelligent Vehicles Symposium, pp. 1108,1113, 3-7 June 2012.
13. A. S. Simões, Aprendizado não-supervisionado em redes neurais pulsadas de base radial [online]. São Paulo: Escola Politécnica, Universidade de São Paulo, 2006. Tese de Doutorado em Sistemas Digitais. [Acessed 2014-01-29]. Available at: ¡http://www.teses.usp.br/teses/disponiveis/3/3141/tde-15092006-153353/¿.
14. E. Tunstel, "Mobile robot autonomy via hierarchical fuzzy behavior control," Conf. Rec. 1st World Automation Congress, pp. 837-842, 1996.
15. K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, "Map Merging for Distributed Robot Navigation," Conf. Rec. IEEE/IROS, pp. 212-217, 2003.

16. P. Dinnissen, S. N. Givigi, and H. M. Schwartz, “Map merging of Multi-Robot SLAM using Reinforcement Learning,” Conf. Rec. IEEE/SMC, pp. 53,60, 14-17 Oct. 2012.
17. P. Dinnissen, “Using Reinforcement Learning in Multi-Robot SLAM”, MSc. Thesis, Carleton University: CA.
18. B. Slamet and M. Pfingsthorn, “ManifoldSLAM: a Multi-Agent Simultaneous Localization and Mapping System for the RoboCup Rescue Virtual Robots Competition,” MSc Thesis (Afstudeerscriptie), Universiteit Van Amsterdam, 2006.
19. P. Newman, “EKF based navigation and SLAM”. SLAM Summer School 2006, Oxford (2006).
20. K. Chanki, R. Sakthivel, and W. K. Chung, “Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem,” IEEE Transactions on Robotics and Automation, vol. 24, no. 4, pp. 808-820, Aug. 2008.
21. K. Arras, CAS Robot Navigation Toolbox, Available at: http://www.cas.kth.se/toolbox/.
22. C. Y. Wong, G. Seet, K. S. Siang, and W. C. Pang, “A framework for area coverage and the visual search for victims in USAR with a mobile robot,” Conf. Rec. IEEE/STUDENT, pp. 112-118, 20-21 Nov. 2010.
23. M. Z. Aziz, B. Mertsching, “Survivor search with autonomous UGVs using multimodal overt attention,” Conf. Rec. IEEE/SSRR, pp. 1,6, 26-30 July 2010.
24. J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, M. Andriluka, O. Schwahn, U. Klingauf, S. Roth, B. Schiele, and O. Stryk, “A semantic world model for urban search and rescue based on heterogeneous sensors,” Lecture Notes in Computer Science, J. Ruiz-del Solar, E. Chown, and P. Plüger, Eds. Springer Berlin Heidelberg, 2011, vol. 6556, pp. 180–193.
25. G. Liu, H. Tong, and R. Zhang, “An intelligent control architecture for search robot based on orthogonal perception information,” Conf. Rec. IEEE/FSKD, pp. 2348,2352, 29-31 May 2012.
26. The Julia Language, Available at: http://julialang.org/. Accessed in: February 2014.
27. B. Qing-yong, L. Shun-ming, S. Wei-yan, and A. Mu-jin, “A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments,” Conf. Rec. IEEE/AICI, vol.2, pp.257,261, 7-8 Nov. 2009
28. H. C. Lee, Y. J. Cho, and B. H. Lee, “Accurate map merging with virtual emphasis for multi-robot systems, Electronics Letters, vol. 49, no. 15, pp. 932934, 2013.
29. S. Kohlbrecher, O. V. Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” Conf. Rec. IEEE/SSRR, pp.155,160, 1-5 Nov. 2011.
30. G. Grisetti, C. Stachniss, and W. Burgard, “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters,” IEEE Transactions on Robotics, Volume 23, pages 34-46, 2007.
31. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” Conf. Rec. IEEE/CVPR, pp.886,893 vol. 1, 25-25 June 2005.
32. Y. Uzun, M. Balcilar, K. Mahmoodi, F. Davletov, M. F. Amasyali, and S. Yavuz, “Usage of HoG (histograms of oriented gradients) features for victim detection at disaster areas,” Conf. Rec. IEEE/ELECO, pp.535,538, 28-30 Nov. 2013.
33. Wikipedia, ”Robot Operating System — Wikipedia, The Free Encyclopedia, 2004, http://en.wikipedia.org/wiki/Robot_Operating_System,[Online; accessed 30-June-2013].