



RoboCupRescue 2017 - Virtual Robots Team Echoic (Iran) Team Description Paper

Elham Iravani, Saba Kadjodazadeh, Nicky Sadighi, Pantea Shamirzaei, Saba Kalaei,
Nika Sadat Mousavi Zarandi

Salaam Resaalat High School, Tehran, Iran
Rescue Robots Center of Youth Organization of Iranian Red Crescent Society, Golestan, Iran

`e_iravani@aut.ac.ir`

Abstract. This paper describes the approaches of the Echoic rescue team. The framework that used is ROS operating system. A novel scenario is designed for victim detection, path planning and navigation. Moreover, a decision making model designed for robots to have good multi agent tasks handling.

Keywords: virtual rescue robot, ROS framework, decision-making model, path-planning algorithm, SLAM, HOG-HAAR.

1 Introduction

The team members have a lot of experience in robotics and programming. Team leader has good experiences in national and international RoboCup competitions in Real Rescue, @Home and Humanoid robot and 2D Soccer simulation leagues in the past years. Furthermore, other members have good experiences in national programming and 2D Soccer Simulation competitions. This team has started to work in this league since last year and successfully participated in IranOpen 2016, RoboCup 2016, Leipzig and Iran-Open 2017.

We use some autonomous robots and a manual robot with P3AT model and a Quadrotor for complete arena covering especially outdoor. The number of autonomous robots depends on the mission. The robot has a laser scanner in order to navigate and generate a 2D map. We also have new methods for planning and victim detection.

According to the frontiers like exploration model for planning, which customized with some newly developed approach, the robots can explore autonomously very well.

During the mission, the robots produce two types of map including a 2D map and a grid map. A grid map is shared between the robots to let them know which locations have been occupied before and which not, to cover more area faster. Another 2D map is generated with details containing arena's shape and victims' location and robot paths. Finally, a good map is available.

For software development, we continued using ROS directly. ROS concepts help the developer to easily manage processes and keep the modules as simple as possible so debugging the whole process of a robot's behavior will not be a challenge. Lots of work is done in the open-source community of ROS in form of ROS packages therefore we could save lots of time and energy and focus on improving algorithms instead of writing everything from scratch. [1]

2 Map generation/printing

For SLAM and Navigation, we have used ROS packages from Gmapping [2] with a 2D laser scanner. The SLAM begins at the start and our exploration algorithm plans where to go next, based on whether a victim is detected or more exploration is needed. Figure 1 shows a map which is generated using our customized SLAM algorithm based on Gmapping by Echoic team.

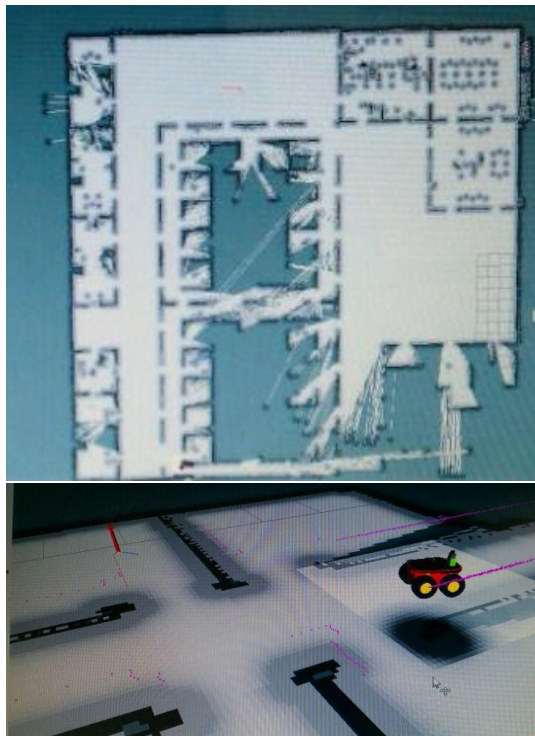


Fig. 1. Output map from the 2012 Arena

We developed our own method for planning the robot's motion where more exploration is needed to expand the map for unknown areas. The developed method is based on Frontier Exploration. Frontiers are regions on the boundary between open space and unexplored space. The Frontier Exploration is a strategy based on the robots moving to these frontiers, adding new information to its map and seeing into unexplored space, while constantly increasing its knowledge, in the quickest way possible. [3] We merge this method with some new concepts to generate more accurate frontiers algorithm to have optimized path planner. As mentioned before, robots have a shared grid map to know about occupied cell to move appropriate through the arena and explore more area.

The UAV robot flies over the walls during each mission and generates a 2D map using the bottom camera. When the operator marks the victim, he will take the robot to a higher point above the victim and an image is taken from that view. These images will be stitched into another aerial map, which is generated at the beginning of the mission with the same methods [4], by flying all over the mission's area at the beginning. Victim's position in the taken image is preserved by tracking SIFT features of the area in which the operator has marked the victim, hence it will be available in a final 2D map. Then it should convert the graphical maps to the form of the map that a UGV generates. Further processing can be done if the map has to be visually the same as the map generated by the UGV.

Another important factor to be mentioned is how the robots can share their location and have a common map, which is updated by each of the robots. Each robot has a GPS sensor that is filtered by odometry. So the locations of all robots are in the same pattern. In this way, the robots can have a common map that updates in each frame.

3 Autonomous Navigation

The navigation stack is simple and useful to use for autonomous navigation. It receives odometry and sensor streams information and send velocity commands to a mobile base. For using navigation stack, the robot must run ROS, containing a tf transform tree in place and it should receive sensor data and publish it by using the correct ROS message types. In addition, the Navigation Stack needs to be configured for the shape and dynamics of a robot to be performed at a high level.

By ROS nodes, we can control a mobile robot within a planner environment. The most important features, which make multiple robots to build a map cooperatively, are: a path planner, a graph based SLAM node and purely reactive obstacle avoidance. Using an exploration plugin for the navigation module allows the robot to explore the working space autonomously and create a shared map for navigation.

The Nearness Diagram Navigation [5] works by choosing a navigational situation in all circumstances and giving appropriate motion to each of them. This method even works in an environment with little space to move. The system uses the data of a laser scanner and the odometry of the vehicle. Then, for reaching the goal, it can use the information of the changes in the obstacle and free spaces in the model that is used by

the planner module to compute the course. At last, the avoidance module uses the information of the obstacles contained in the grid and information of this tactical planner to generate the motion (to reach a goal).

For executing the motion and restarting the process with a new sensorial measurement, we must use the vehicle controller.

After all these things, the design of the modules and the integration architecture should be addressed with emphasis in the planning module. [6]

4 Autonomous Exploration

4.1 Sensors for Navigation and Localization [7]

Laser Scanner.

A Scanning Laser Rangefinder is used in the robot. It is mostly used for 2D mapping but can also offer 3D data.

The Scanning Laser Rangefinder is a small, affordable and accurate laser scanner that is perfect for robotic applications. The Rangefinders are almost able to report ranges in a 240° arc (0.36° angular resolution) and in various ranges and accuracy and power consumption. The laser is used by robot for SLAM and Navigation task. The resolution, accuracy and range of view are the important properties that cause using laser.

Camera.

A camera is mounted in all robots and is used to detect both moving and fixed victims, map generating with UAV and control the robot by the operator. In UAV, it is used to generate aerial maps and calculate the optical flow for navigation purposes. It is also used for tracking visual features of where the victim is marked as described in section 5.

It is also used for our experiments on autonomous flight and victim detection of UAV. In manual flight, we use the camera to help the operator in control of the UAV when it should lower the robot's distance from where the victim is, avoiding collision with the walls.

Ultrasonic Range Finder.

This sensor is placed around the autonomous robot for wall detection. These are only used as the confidence point to be sure that the robot can avoid the obstacle if another strategy cannot handle the condition.

4.2 Multi Agent Exploration

To explore the environment optimally it is necessary to have a good strategy to allocate the tasks appropriately to all robots. The first robot, which is placed in front of others, should scan the environment and see how much the angle of view is open to go forward.

Then a decision making model should run to assign an appropriate angle and time of initiate to each robot to start. [7]

In the next phase, a grid map is considered and it is accessible by all the robots. It is noticeable that this map is just for task allocation. Each robot should occupy the position it goes through. Thus, when there is more than one direction to go for each robot according to the path-planning algorithm, the robot should go through one, which has not been occupied before.

Robots use another strategy to have good Multi Agent system. Its objective is, to quickly recognize closed two-dimensional environments, without any access to references such as a Global Positioning System (GPS), to perform exploration of each unit, and the joint recognition. All robots communicate together with ROS, a system responsible for data collection, tracking and managing all collected information. Finally, it provides a basis for multi-agent robots, which allow recognition, mapping and information gathering in places where units are efficiently deployed the entire colony's ability. [8]

The most important parts of this algorithm are the graph and the number of the agents. The multi-agent path finding (MAPF) problem is a generalization of the single-agent path. Each agent has a unique start point and a unique goal point. Robots should find paths for all agents from the start point to the goal point. Robots cannot exit the circumscription. In addition, they cannot collide during their movements. [9]

5 Victim Detection

The main task of rescue robots is to find victims as many as possible. Moreover, the victim's location should be marked on the generated map autonomously. Actually detecting people by a moving robot with a monocular camera is still a very difficult and open problem. If the camera is stationary, then the problem can be simplified through background subtraction. A few algorithms can recognize people via RGB camera, which is not accurate enough. Nowadays, it is popular to detect people via RGB-D cameras like Kinect that is not attached to the introduced robots.

The most famous algorithm with RGB for human detection is Histograms of Oriented Gradient (HOG) algorithm. If there are not suitable dataset, it may result in many false positives and missed proper detections. This approach counts occurrences of gradient orientation in localized portions of an image. The key feature behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. [10]

First of all, a NN trained by a huge data set which provide from the previous area in gazebo to create positive and negative data. Five body gestures are trained. So, five HOG descriptor function are used in the next steps to detect the victim. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image. This normalization results in better invariance to changes in illumination and shadowing.

In comparison to another descriptor, HOG has some advantages because it is invariant to geometric and photometric transformations as it operates on local cells. Thus, the HOG descriptor is particularly suited for human detection in images as it shown in figure 2.



Fig. 2. HOG output for human detection [11]

Object Detection using HAAR feature-based cascade classifiers is the most effective object detection method. For detecting an object, we need to have positive and negative image. The positive images include the faces and the negative images contain environments (without humans). Therefore, the algorithm needs many positive and negative images to train the classifier. By using this algorithm, robots can detect faces. [12] We can use HAAR-like features for the structure of the object and HOG features for the edge. [13] Therefore, these two algorithms are fused and used for detecting purpose. A dataset of 5800 positive and 8200 negative images is created for training. In the tests the human detection is more accurate than face detection as it has 96% accurate in gazebo environment tests. The result of human detection using this approach is shown in figure 3.

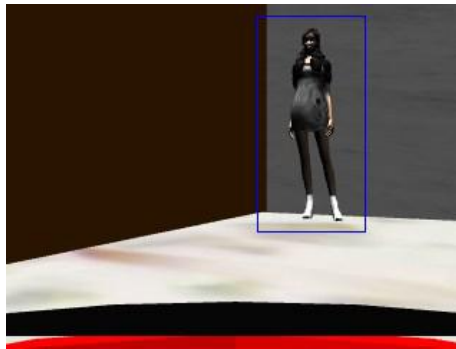


Fig. 3. Output of Echoic human detection (Victim in VRL2016)

6 Simulation

In robotic world, simulators have played an important role to represent the real world virtually and test the mechanisms and algorithms. Gazebo is a 3D simulator that can communicate to well-known Robotic Operating System (ROS). Figure 4 shows which important packages of ROS is used and for which purposes. [7]

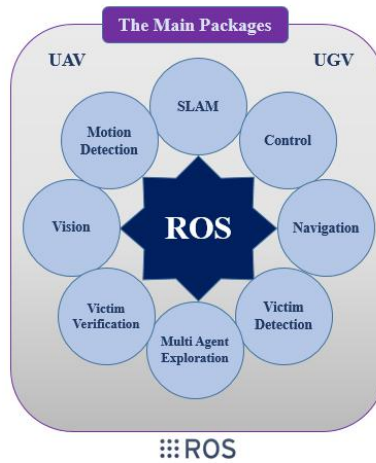


Fig. 4. Our main packages of ROS

The robots in the client session should connect to each other in order to make the multi agent exploration possible. They should get sensors' data from ROS server.

For creating complex robot behavior rapidly, we use SMACH, which is task-level architecture. SMACH is a library that uses old significances to create robot behavior. SMACH is an effective way when you want a robot to performance some complex plan. [14]

7 Innovation

There are some important experimental changes in comparison to last year. Last year in Leipzig, we can only have a manual robot which has a high-loaded model but a very good SLAM. So, the map was so good but the real time factor was so low. This year we can implement a good multi agent system with both autonomous and manual robots. So, a basic navigation model is implemented. For autonomous victim detection, a useful combined HOG-HAAR algorithm is used in the new system.

8 Conclusion

Our purpose of participating in this competition is far more than just winning awards; the most important thing that made us to take part in this competition is to achieve experience and knowledge about international society and virtual rescue league. The other thing that encouraged us is the idea of saving human's life; In Fact, all over the world the life of saviors is in danger. Therefore, it is a good reason to send the robots in the scene instead of humans. According to the human mercy, we are responsible for taking care about others life. Unfortunately, it cannot happen yet but we hope to see it in the future. If this happens, it can be a great step for the human society.

References

1. <http://wiki.ros.org>
2. <http://wiki.ros.org/gmapping>
3. B. Yamauchi, "A Frontier-Based Approach for Autonomous Exploration", In *CIRA '97 Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146-151, 1997.
4. <http://wiki.ros.org/mapstitch>
5. M., Javier, L. Montano. "Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios." In *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 1, pp. 45-59, 2004.
6. J., Minguez, "Integration of planning and reactive obstacle avoidance in autonomous sensor-based navigation." In *Intelligent Robots and Systems (IROS)*, IEEE, pp. 2486-2492, 2005.
7. E. Iravani, S. Shayanfard, N. Rajabi Chavari, K. Rajabi, etc., "Virtual Rescue Robot League Team Description Paper of Echoic (IRAN)", *RoboCup 2016, Leipzig, Germany*, 2016.
8. I. Hughes, G. Millán, etc. "Colony of Robots for Exploration Based on Multi-Agent System", In *International Journal of Computers Communications & Control*, Vol: 9, No. 6, 2014.
9. Zhou, Xin. "A Survey of Multi-Agent Path Finding Problem."
10. N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection.", In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Vol: 1, pp. 886-893, 2005.
11. D., Navneet, and B., Triggs. "Histograms of oriented gradients for human detection." In *Computer Vision and Pattern Recognition. CVPR IEEE*, Vol. 1., 2005.
12. http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
13. C., Xia, S. F., Sun, P., Chen, H., Luo, & F. M., Dong, "Haar-like and HOG fusion based object tracking." In *Pacific Rim Conference on Multimedia*, pp. 173-182, 2014.
14. <http://wiki.ros.org/smach>