RoboCupRescue 2016 - Virtual Robots Team Echoic (Iran) Team Description Paper

Elham Iravani[1], Negar Rajabichavari, Sogand Shayanfard, Kiana Rajabi

Salaam Resaalat High School, Tehran, Iran
e_iravani@aut.ac.ir
www.echoicrescue.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 verification, a decision making model for multi agent exploration, path planning and navigation. Moreover, a decision making model designed for robots to start with their mission.

Keywords: virtual rescue robot, ROS framework, decision making model, thinning planning algorithm, SLAM.

1 Introduction

The team members have a lot of experience in robotics and programming. Team leader has good experiences in the national and international RoboCup competition 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.
Now, in this year we want to participate in virtual rescue robot league. We use some autonomous robots and a manual robot with P3AT model and a Quadrotor for complete arena covering. The number of autonomous robots is related to the round. The robot has laser scanner to navigate well and generate 2d map. We have new method for planning and also victim detection and verification. The robot prioritize victims that seen and give weight to them then the robot navigate through corridor to arrive to the victims in queue.

[1] Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.

At the start of mission by using a decision making model, each robot gets a proper direction and time to start the mission. During the mission, the robots produce two types of map including a 2D map and a grid map. A grid map share between the robots to let them know which locations occupied before and which not to cover more area faster. Another 2D map is generated with detail contains arena's shape and victims' location. Thus, at the end 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 would not be challenge. Lots of work is done in the open-source community of ROS in form of ROS packages then we could save lots of time and energy and focused on improving algorithms instead of writing everything from scratch. [1]

2 Team Members and Their Contribution

- **Elham Iravani:** SLAM, UAV and UGV Control, Autonomous Exploration, Multi Agent Exploration, Task Allocation, Victim Detection, Motion Detection, Decision Making.
- **Negar Rajabichavari:** SLAM, UGV Control, Autonomous Exploration.
- **Sogand Shayanfard:** UAV Control, Victim Detection, Motion Detection.
- **Kiana Rajabi:** Multi Agent Exploration, Task Allocation, Decision Making.

3 Map generation/printing

For SLAM and Navigation, we have used ROS packages from Hector team, hector_slam [3] and hector_navigation [4] with a 2D laser scanner. The SLAM begins at the start and our exploration algorithm plans where to go next, based on weather a victim is detected or more exploration is needed. The figure 1 shows a map which is generated using hector_slam algorithm.

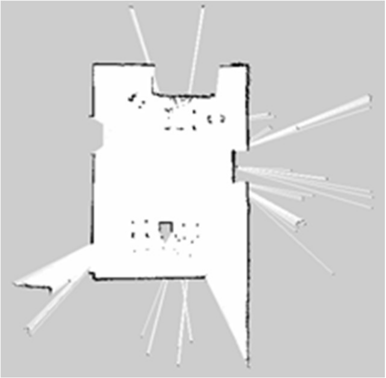

Fig. 1. Output map from the AUT robotic laboratory (using hector_slam package [3])

We use hector navigation to navigate inside the map area whenever is needed. In addition, we developed our own method for planning the robot motion where more exploration is needed to expand the map for unknown areas. The developed method is based on morphological thinning kernels which thins the extracted corridor and converts it to a skeleton so the planner is able to choose the next point along the middle line in each view the robot has in each frame. The output of this method is shown in figure 2. [2]

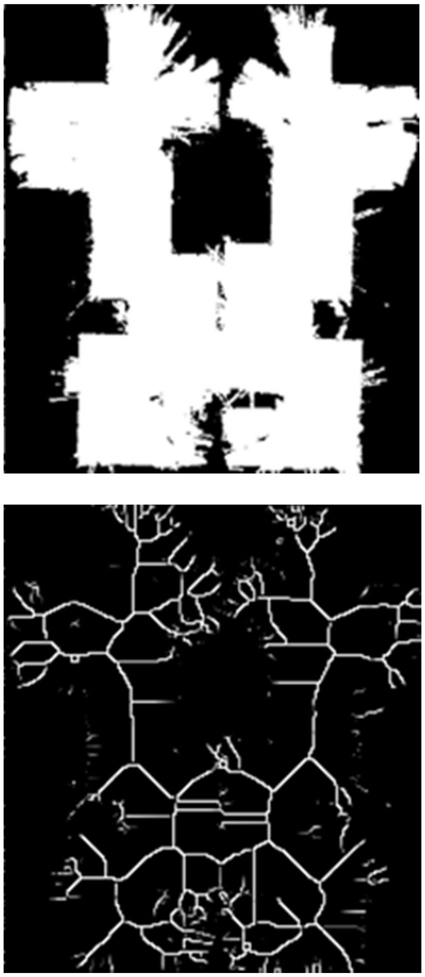

Fig. 2. Output of thinning algorithm for planning (map is related to yellow Real Rescue arena of RoboCup IranOpen)

The UAV robot flies over the walls during each mission and generates a map in 2D 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 [5], 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 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 generated map by the UGV.

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

4 Obstacle Avoidance

A suitable obstacle avoidance algorithm should be applied in order to move through the crowded environment. Vector Field Histogram (VFH) method is considered to implement in this paper. Johann Borenstein and Yoram Koren proposed this real time motion planning algorithm. [6]

According to VFH algorithm, we can have a statistical representation of the robot's environment. We should consider great emphasis on dealing with uncertainty from sensor and modeling errors in the histogram grid. VFH has an advantage in comparison to another algorithm which consider the dynamics and shape of the robot in order to returns steering commands. VFH can produce near optimal paths while it used as a local path planner not for global path optimality. The figure 3 shows the result of the histogram. Some certain angle of the laser scanner are used for this method.

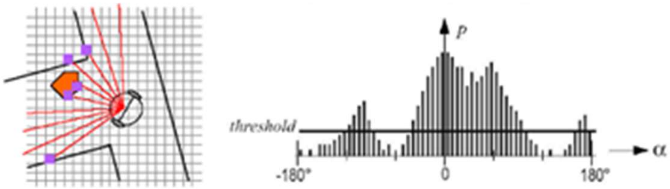

Fig. 3. VFH output [6]

5 Autonomous Exploration

5.1 Sensors for Navigation and Localization

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

The Scanning Laser Rangefinder is a small, affordable and accurate laser scanner that is perfect for robotic applications. The Rangefinders almost are 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.
This camera is mounted in all robots and is used to detect both movement and fixed victims, map generating with UAV and control the robot by the operator. In UAV, it is used to generate aerial maps and calculating 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 have used 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.

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

5.2 Multi Agent Exploration

To explore the environment optimally it is necessary to have a good strategy to allocate the tasks appropriately to the 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.

A back propagation neural network is considered to use as the decision making model. The inputs in this network is “the angle of view; the number of robots; robots’ dimension” and the output layer is contain “direction for each robot; time of start for each robot”. A function should be consider to weight the network according to distance and probability of collision between robots. It is obvious that the robot may start with a delay according to the route and their numbers.

In the next phase, a grid map is consider 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 go 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 is not occupied before.

6 Victim Identification

6.1 Victim Detection

The main task of rescue robots is to find as many as victims. Moreover, the victim's location should be marked in 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. For the first situation there are a few algorithms that can recognize people via RGB camera and it is so popular to detect people via RGB-D cameras like Kinect that is not attached to the introduced robots.

The most famous algorithm of human detection is Histograms of Oriented Gradient (HOG) algorithm. In the result it may get a lot of false positives and missed detections from it. 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. [7]

Each frame consider as an image which need to divide into the smaller parts. For pixels of each region, a histogram of gradient directions is compiled. Then, the descriptor is the concatenation of these histograms. 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. [7]

In comparison to another descriptors, 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 4.

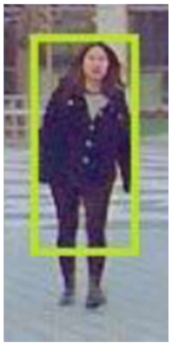

Fig. 4. Output of human detection using HOG (MIT Pedestrian Data Set [8])

6.2 Victim Motion Detection

Some victims may have some motions and robots should be able to detect those motions. The robots can do so by frame subtraction methods with its visual sensors

6.3 Victim Verification

We have designed a multi-sensor fusion platform which detects victims and plans what to do against each one in two phases: Detection and Verification. In the detection phase, robot looks for potential victim candidates and potential places which should be scrutinized more. Possible candidates are detected using RGB camera. These possibilities are calculated in different ROS Services which report their findings to a Verifier service. After all, the verifier plans if there is a real victim or more scrutiny is needed, and sends the related data to a planner service. The planner service has a priority list which is filled by candidates' data, like "vic1, X, Y, Z" which has been sent by the verifier. This service plans if the robot has certain victims to mark in the map, or it should navigate to another position to find a real victim. The victims can be verified with another sensors or the same one in a near distance to be sure about the detected victim. During all these phases, new candidates can be added to the list which also has fixed constraints like size, which avoids the robot from continuously switch between possible candidates. [2]

7 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 5 shows which important packages of ROS is used and for which purposes.

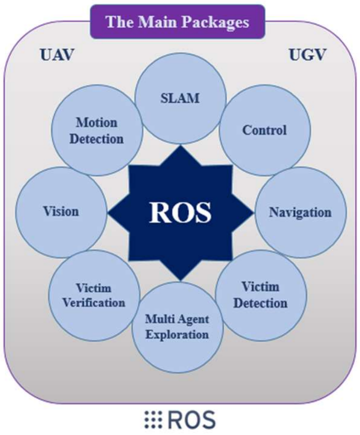

Fig. 5. Our main packages of ROS

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.

8 Conclusion

Our goal of participating in this competition is far more than just winning awards. As in virtual environment teams can develop and test the novel algorithms and strategies, this contest could be a very good benchmark for identifying our strengths and weaknesses. We can use this precious experience to improve our team and robots for the future competitions.

References

1. http://wiki.ros.org
2. E. Iravani, N. Khazaee Korghond, etc., "Real Rescue Robot League Team Description Paper of ARC- AUT (IRAN)", RoboCup 2015.
3. http://wiki.ros.org/hector_slam
4. http://wiki.ros.org/hector_navigation
5. http://wiki.ros.org/mapstitch
6. J. Borenstein, Y. Koren, "The vector field histogram-fast obstacle avoidance for mobilerobots". In *Robotics and Automation, IEEE Transactions on*, Vol: 7, pp.: 278–288. , 1991.
7. 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.
8. http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html