RoboCup 2019 Virtual Rescue Robot Firebolts (Iran) Team Description Paper

Nicky Sadighi, Maryam Ahmadian fard, Ghazal Laghaee, Ghazaleh Didari, and Saba Kadkhodazadeh

sadighi.nicky@gmail.com
http://www.github.com/FireboltsVR

Abstract. This paper is a description of the strategies developed by the *Firebolts* team for participating in the RoboCup 2019 Rescue Simulation League Virtual Robot Competition. The framework used by this team is ROS. Several algorithms are designed for mapping, victim detection, multi robot exploration and decision making.

Keywords: RoboCup 2019 Rescue Simulation League Virtual Robot Competition · ROS framework · SLAM · Navigation · Human Recognition · Multi Agent System.

1 Introduction

Firebolts Virtual Rescue team is a team first formed in 2018 and has participated in the Robocup Asia Pacific 2018 competitions, winning the first place. This team has improved drastically, starting off from a very simple base code. *Firebolts* team members have a lot of experience in National and International Robocup competitions. Our main motivation for taking part in these competitions is to help improving autonomous exploration and rescue, so that every day less rescuers need to risk their lives going into ruins after a natural or man-made disaster. This year we have mainly focused on efficient victim detection, marking and also connecting the robots in order to explore more systematically.

Team members and their contributions:

- Victim Detection: Maryam Ahmadian fard, Saba Kadkhodazadeh
- Autonomous Exploration: Nicky Sadighi, Saba Kadkhodazadeh, Ghazal Laghaee
- Quadrotor: Ghazaleh Didari
- Navigation: Nicky Sadighi, Maryam Ahmadian fard
- User Interface: Ghazal Laghaee
- RGBD Slam Multi agent Exploration: Nicky Sadighi

2 System Architecture

Our main software modules are Navigation, Exploration, Localization, Mapping, User Interface, Victim Detection and Marking. Each robot is provided with these utilities. Our robots are currently not able to communicate with each other directly, although it is our goal to add this option to our code. We use two types of robots, firstly the Pioneer3AT with the Hokuyo laser scanner and an RGBD camera and odometry sensors; and secondly the Quadrotor. Our system is directly implemented in ROS which is a system in which tools and information are put to users and Gazebo for simulating the urban research as rescue scenarios.

2.1 Mapping

As the robot moves in the unknown environment, it should generate a map for itself to navigate and the rescue forces to use it afterwards.A Robots navigation would be much difficult in case of not having a map because the robot may choose a pre-explored target and spend its time on just some specific part of the map or could not reach its decided location due to wrong path planning. In robotic mapping and navigation, simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. We are currently using the Gmapping algorithm used in the ROS Slam package [1] with enhanced parameters in order to optimize the map precision. Using the Gmapping algorithm, robots can draw a map of the environment using the data that the laser and camera gathers from the environment. For our Quadrotors we use Hector Mapping.

2.2 Shared Map

As you can see in the above explanations, each of our robots alone generates a map of the environment. As we know the robots work in a group and that is the reason we emphasize the division of task which depends on the right sharing of the maps drawn by each robot, which causes the need of creating a shared map. We use the Multi Robot Map merger package from ROS repository to publish the map between our robots. The Map Merger ROS node retrieves maps from the Robots and attempts to merge them into a larger map using a modified version of the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) [2]. Using this package, our system can prevent the loss of time cost and prevent to search the environment more than once by sharing the map between the robots. Our output is shown in (see Fig. 1).

Fig. 1: Firebolts Map from Robocupap 2018 Final Round

2.3 Autonomous Navigation

One of the most important goals of this league is to accomplish a full autonomous navigation system so as to become needless of operators. The Navigation Stack takes in information from odometry and sensor streams and outputs velocity commands to send to a mobile base. [3] The main features of our navigation algorithm are a purely reactive obstacle avoidance and a simple path planner node that allows multiple robots to cooperatively build a map. By using an exploration plugin for the navigation module, one or more robots can autonomously explore their working space and create a shared map for navigation on all robots.

2.4 Victim Detection

Using the RGBD camera, robots are able to receive images of their environment. An algorithm is needed to detect if part of the surrounding is a victim. We developed a code based on the HOG algorithm [4–6]. Given a database of correct and false samples of humans, it learns the right shape of victims and compares it to each image frame it receives from the camera. Of course the problem of noisy and crowded places remains, where we cant explicitly tell the victim apart, we are still working on resolving this issue . Another way of detecting the victims by their voice, but this way is not used yet in any competition. Alive and dead victims should be treated differently when found. There are three ways to understand if a victim is still alive, by their voice, their motion and their heat; Voice detection is still not used in Robocup competitions, for motion detection we have designed an algorithm that makes the robot stand still and compare two images of the victim, if they are different it means that the victim is moving and hence alive. Our robots use a thermal camera to detect the heat of alive victims. Our output is shown in (see Fig. 2).

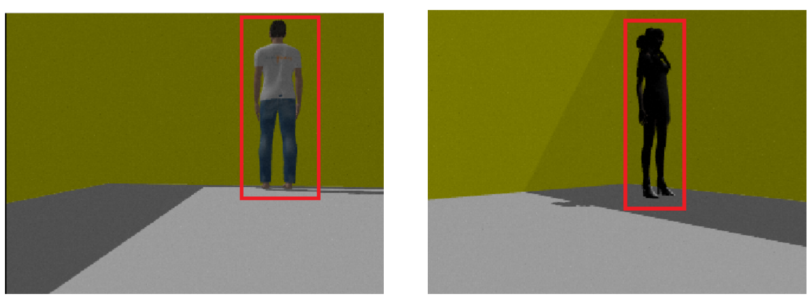

Fig. 2: HOG sample data

2.5 Marking

After robots have discovered a victim, it is important to specify its place in the map they have generated so that rescue teams can find the victim later. For this purpose, we mark the position of the victims in the map using a graphical algorithm.

2.6 Controller

We use the *Firebolts* user interface to save the map ,start each robots autonomous exploration, and switch to manual mode whenever desired. We also use a PS4 controller for manually driving the robots in case they are stuck or to separate their paths if they are initially close to each other.

3 Conclusion

In this paper we have given a short introduction on our system architecture. Last year we enhanced the navigation of separate robots, this year we are aiming to minimize the exploration time using a shared map for each robots navigation stack. Also we are trying to improve our victim detection and marking algorithms.

4 Acknowledgement

We wish to thank our families for their support, our teachers, and also sponsors for their financial support without which, all of this would have not been possible.

References

1. http://wiki.ros.org/gmapping
2. Li-Chee-Ming, Julien, and Costas Armenakis. "Feasibility study of using the RoboEarth cloud engine for rapid mapping and tracking with small unmanned aerial systems." The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40.1 (2014): 219.
3. http://wiki.ros.org/navigation
4. 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.
5. D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection." In Computer Vision and Pattern Recognition. CVPRIEEE, Vol. 1., 2005.
6. C. Xia and S. F. Sun and P. Chen and H. Luo and F. M. Dong, Haar-like and HOG fusion basedobject tracking. In Pacific Rim Conference on Multimedia, pp. 173-182, 2014.