RoboCup Asia Pacific 2018 – Virtual Rescue Robot League Team Description

<Firebolts (Iran)>

Nicky Sadighi, Maryam Ahmadian fard, Ghazaleh Didari, Ghazal Laghaee

Salam farmanieh High school
NO.57,Hosseini St, Saeedi St, Niavaran Ave.
P.O. Box 1954618851
{sadighi.nicky, maryamahmadian733, didarighazaleh, laghaee.ghazal} @gmail.com
http://farmaniyeh.salamsch.com

Abstract. This paper is a short review of strategy designed by' Firebolts' team for being a part of the RoboCup Asia Pacific 2018 virtual rescue robot competition. 'Firebolts' is a team formed in Salam Farmanieh High school which has an experience in Cospace robot league for years and also in major rescue robot simulation league in spite of being students. As this year is the second participation of us in this league, the team mainly focuses on the important problem in rescue scenario which is human identification. In this paper, we introduced our system architecture which contains different algorithms we used in decision making, mapping, navigation, victim detection, and planning based on ROS framework and Gazebo.

Keywords: RoboCup Asia Pacific 2018 Virtual Robot Rescue, ROS, SLAM, Navigation, Multi Agent System, Human Recognition.

1 Introduction

In today's world, one of the major problems is rescuing people from natural and man-made disasters in a way that least first responders are injured. Every year many firefighters and rescue people are injured and killed, all of this could be prevented with appropriate rescue robots, searching for victims autonomously and saving them, and also giving a map of the disaster field. We live in an earthquake-prone country where there occurs in average 150 earthquakes with 5.4 in magnitude annually and one of our goals in participating in this competition is to present a suitable strategy to prove our cooperation with the earthquake victims of our country, especially the big earthquake that happened in Kermanshah last November. Another incident is the Plasco building which burnt and collapsed one year ago and if rescue robots had been sent inside the building instead of firefighters the number of fatality certainly would have decreased. All above factors motivated Salam Complex authorities to enhance their knowledge and research about virtual rescue robots and we have tried to deal with rescue issues by applying the knowledge and programming skills of our members.
'Firebolts' team members have a lot of experience in robotics and programming. Team leader has good experiences in national and international RoboCup competi-

tions in Virtual Rescue Robot and 2D Soccer simulation leagues in the past years. Furthermore, other members have good experiences in national programming, Virtual Rescue Robot and Cospace competitions. They wish to participate in RoboCup Asia Pacific virtual rescue robots league competitions which will be held this year. The focus of 'Firebolts' team in this year is on recognizing the victims and the members have tried to achieve this goal by presenting a comprehensive algorithm and reviewing available and related literatures and algorithms.

2 Team Members and Their Contribution

Nicky Sadighi: Autonomous Exploration, Navigation, RGBD Slam Multi agent Exploration
Maryam Ahmadian fard: RGBD Slam, Navigation, Autonomous Exploration, Decision Making
Ghazaleh Didari: Behavior, Victim Detection, Motion Detection, Heat Detection, Sound Detection, Quadrotor
Ghazal Laghaee: Powerpoint, Behavior, Aerial Control, Autonomous Exploration, Multi agent Exploration, GUI,Aerial Control, Terrestrial Control

3 System architecture

In Order to have best rescue algorithm in the maps, we use both aerial and terrestrial robots in this project. Both kinds of our robots have laser range finder sensor which is mostly used for mapping. Laser range finder is used to determine a domain from the robot to a selected target or obstacles. It is often used to create a map of the unknown environment and permit the robot to having a view of the field or benefit rangefinder capability.

Our aerial robots, also have a RGB-D camera. These cameras provide both color images and an estimate of each pixel's depth so with this camera, our robot can get more information about 3D environment to draw a better map. On the other hand our terrestrial robot also have Thermal camera. The thermal camera is a camera that can detect ambient temperature and body temperature. We used it in order to detect alive injured victims by detecting its heat which will be explained in the following section.

As we said earlier 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.

3.1 Mapping

In order to have an autonomous robot which moves in an unknown environment, a need of the map for better exploration is clear. Robot's 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 couldn't reach its decided location due to wrong path planning. So, generating a map of the environment is one

of the most important problems which robot should solve. Simultaneous localization and mapping (SLAM) is a very convenient approach to this problem. SLAM can be used for both two-dimensional and three-dimensional maps. Slam 2D is used in our terrestrial robots and RGB-D Slam is used in our aerial robots. Our SLAM algorithm is based on GMapping algorithm. Using Gmapping, the robot can draw a map of the environment using the data that the laser and camera gathers from the environment.
When a robot automatically navigates through an unknown environment, it needs to have a map. To create our map, we use the Gmapping algorithm used in the ROS Slam package. Gmapping is an algorithm that uses laser data to locate the surroundings obstacles, walls, and open space by drawing a map. Gmapping is a Rao-Blackwellized particle filters that in which each particle carries an individual map of the environment and an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation[1].
Nodes that are currently available in Slam_GMapping package of ROS is used as a base package for our SLAM algorithm that can boost the robot system. Hector is the best relief and rescue robot in this field. The robot has a sophisticated program that can save more injuries to the robot. Hector has some ideas for solving the robustness problem that we used in our gmapping algorithm in order to make the algorithm more robust due to the noisy environments which we may face in this year's competitions.
It should be mentioned that the probability of robot's location while moving in the 2D map is calculated by AMCL. If the AMCL shows the sign (0,0,0), that means an error has occurred which is considered in order to prevent destruction of map.

3.2 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)

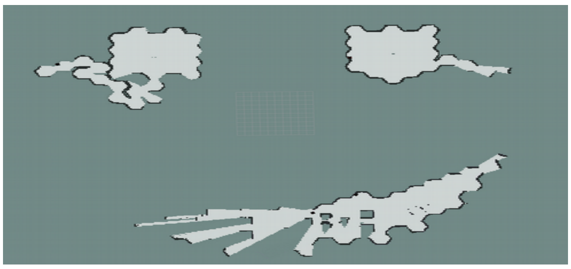


Fig. 1. Firebolts, shared map result from 3 robots from Iranopen 2018 "Round 1" map.

3.3 Autonomous Navigation

Navigation system is one kind of system that focuses on tracking and controlling the movement of a vehicle from one location to another. We use ROS Navigation stack as a base of our navigation system and we also add Odometry data to our system in order to estimate (rather than accurately determine) the relative position relative to a starting point.
Our robot uses the laser and camera that acts like eyes to give us information, and with the navigation package nodes, we can give the robot a program that moves on its own because controlling the robot using Joystick is a very difficult and time consuming task, also we are aware of the fact that every year autonomous systems must improve drastically because of the needs of human being. The two different exploration approach were used in order to help our multi agent system explore the whole map.

The first one is Frontier exploration approach which is a borderline-based exploration algorithm in an unknown environment. In this method, exploration algorithm is based on borderlines or in other words frontier and in our algorithm exploration point will be chosen based on the distance of the path and our remained time [3]. The robot can select three different frontier target based of the target distance to a robot which would be the nearest, farthest and random. In general, any time when the robot is in the outdoor environment, robot prefer frontier for the farthest spot while when the robot is in an indoor environment where there are many barriers, robot prefer the nearest frontier. It should be mentioned that in each of these situations each robot will choose a random target between a subset of far/near frontiers which makes a multi robot system do its work by spreading in the environment and prevention of repeated tasks especially in noisy environments.
Sometimes a robot is stuck, in these situations, the robot will choose the farthest frontier and if it is still stuck, whatever its task is will be changed to exploring the random frontier. For frontier exploration we have used FFD (Fast Frontier Detector), a novel approach for frontier detection which processes raw sensor readings, and thus only scans areas that could contain frontiers [4]. We use FFD because our robot thus borrow from FFD the ability to maintain knowledge of frontiers as fast as possible and its algorithm was easy to be understood and to be implemented by our knowledge.
Sometimes, due to the fact that the robot does not find a path to arrive at the victim by using all kind of path planning algorithms make us use a different method called wall following approach. In this approach, the robot first uses a laser to detect the wall and then pursue walls along the length of time. Also, the results was shown that in places full of barriers, the Wall follower method is better than Frontier. We implemented our wall following algorithm based on [5]. In this implementation, two cases i.e, if the left wall is present and if the right wall is present were taken into consideration.

3.3.1 Path Planning

We have our custom global and local planning algorithm. We have introduced the bases of our global planner algorithm. As well as what we have said for this planner we have used some tricks in local planning which mostly purposed stuck situations.

From now on we have two algorithms which combine Dijkstra and A* path planning algorithms but we hope that we could combine those with more algorithms. The robot first tries to plan a path by Dijkstra and then if the result would be a stuck state, the robot will try another plan which would be A*.

3.4 Victim detection

Exiting human detection algorithms use different descriptors for human recognition. Each descriptors depend on specific features. Psychological studies have shown that human perception mechanism is using shape feature to detect objects, so we consider this feature in our algorithm. The shape feature may have incorrect result in noisy and crowded places, so we may have to add some other features to it. Texture or color features are intended in some algorithms, combination of these two features could be useful for describing visual information about the human.
Although detecting dead victims is valuable, finding alive victims is much more important. Their weak point is that motion feature works well only in state of victims have motions. Since we have alive victims in virtual rescue competitions and finding them is very important to us, it is important that our recognition algorithm can diagnose different motions that could happen in different part of human body. But motion feature cannot be used alone because even alive victims might not be able to move or their movement will not be obvious for the algorithm.
We reviewed different kind of algorithms that use one or more of this kind of algorithms and by comparing their result in different scenarios and customizing it by our voice and heat data we proposed a model that considered almost all of the aforementioned features. Table 1 gives a summary of all heading levels.

Feature	Component	Approach
Shape feature	Edge, Shape	(PCA),(HOG),(DHG)
Motion feature	Optical flows	,(HOS),(LBP),(HOF) (LTP),(LUV)
Appearance feature	Color, Texture	,(NRCPB),(CSLBP) ,(LTP),(HAAR),(LID) (LBP),(HOG),(CHO)
Combination feature	_	,(HOG)+(LBP) ,(HOG)+(LUV) ,(HOG)+(HOF) ,(HOG)+(CHOG)

Table 1. comparing the different approach based on [6],[7],[8]

We used the HOG package for victim detection. Using the Histogram of oriented gradients and a database of human shapes, this package can detect if what it is seeing is a human or not. The package written by us also includes: Face detection, Shape detection, Motion detection and Heat detection.

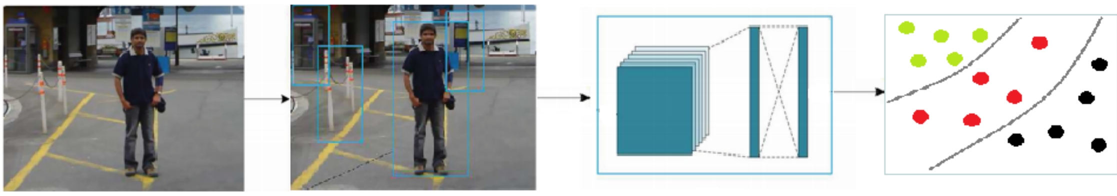
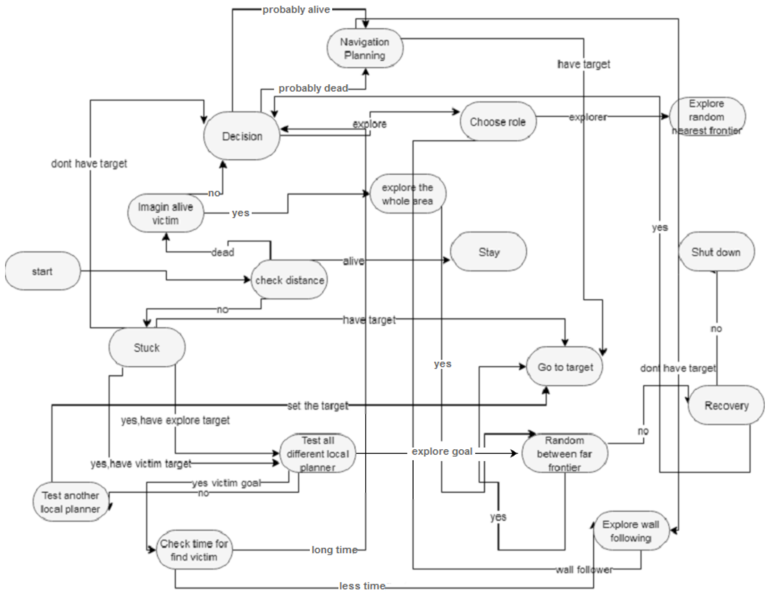


Fig. 2. block diagram of "Firebolts" victim detection algorithm(Green means alive victims, red means dead victims and black means non victim objects)

Fig. 3. HOG sample data.

3.5 Behavior

In this section, we evaluate robot behavior which is one of the most important parts. In general, considering all the observed states in our robot behavior state machine (Fig 5), the most important ones are "Decision making" and "choose role". The first one make core decisions at any moments related to amount of exploration area and possibility of victim's existence which is the result of victim detection algorithm. In other words, if there is a high possibility of victim's existence the robot will follow it, otherwise, if the possibility of alive victim or existence of victim is low, the robot will explore the environment more accurately. The second state's obligation is choosing the robots' role which depends on the distance it explored. If the robot has traveled 40% of the way, by this state, replacing the role of wall follower robot and the frontier robot takes place, and after that, this state ensure us all the places in the map are checked. However, it must be considered that there should be a potential for displacement because if there is no wall in the environment, wall follower role cannot be dedicated to the robot. On the other hand, our strategy is to stand beside the alive victim at the time of detecting. The goal behind this strategy is that if the robot leaves the victim after finding it, the victim might lose his life and our human forces endanger their lives in vain. After identifying an alive victim and looking for it, the robot might confront a dead victim. In this case, it should carefully examine the environment and not leave and change its role unless it makes sure that the environment is empty of the alive victim. Besides, the robot might get stuck after identifying a victim and looking for it, which in this case first the local planer will changed and then if the change was not effective the role should be modified because the presence of the wall gives us a chance that the victim is behind it, hence the robot pursues the wall until it detects the victim.



3.6 Controller

We used GUI Firebolts and PS4 controller, that GUI Firebolts is designed by the group members for controlling and driving the robot so our operator can control the robot easier and also switch easily between manual and autonomous mode and easily save the maps of each individual robot and the merged map.

4 Conclusion

In the previous two year, we first learned various computer components, and then we learned how to design and implement algorithm, then we learned programming languages. We learned how to write programming in Java and C++. We tried to set a goal for ourselves to help our country with this. After a long study of the events that took place in our country, we decided to design a basic code that could be useful and have the basis of improvements in such a way that each year we could debug it and make it much better. In this paper we tried to introduce our system architecture, slam, navigation, victim detection and behavior we designed to the aforementioned purpose. This year, we have tried to focus on victim detection and improve this algorithm, and also a better autonomous robot exploration.e and easily save the maps of each individual robot and the merged map.

5 Acknowledgement

We thank the school staff who paid us a lot of attention and cost to learn and also provided us with a good teacher. Thanks to our good teacher for helping us on this path. Thanks to all the teachers who allowed us to participate fully in the programming class instead of attending their classes.

References

1. Grisettiyz, Giorgio, Cyrill Stachniss, and Wolfram Burgard. "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling." Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005.
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. Yamauchi, Brian. "A frontier-based approach for autonomous exploration." Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on. IEEE, 1997.
4. Keidar, Matan, and Gal A. Kaminka. "Efficient frontier detection for robot exploration." The International Journal of Robotics Research 33.2 (2014): 215-236
5. Nepali, Min Raj, et al. "A novel wall following Algorithm for mobile robots." International Journal of Robotics and Automation (IJRA) 5.2 (2014): 15.
6. Nguyen, Duc Thanh, Wanqing Li, and Philip O. Ogunbona. "Human detection from images and videos: A survey." Pattern Recognition 51 (2016): 148-175.
7. Chen, Xiaogang, et al. "Pedestrian detection with deep convolutional neural network." Asian Conference on Computer Vision. Springer, Cham, 2014.
8. Zhang, Shanshan, Christian Bauckhage, and Armin B. Cremers. "Informed haar-like features improve pedestrian detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.