# MRL-VR Team Description Paper for Rescue Simulation League Virtual Robots RoboCup Competition 2018, Montreal - Canada

Sara Tohidi[1] and Mohammad H. Shayesteh[2]

Islamic Azad University of Qazvin,
Electrical, Information Technology & Computer Sciences Department,
Mechatronics Research Laboratory,
Qazvin, Iran
tohidi.sara.d@gmail.com
m.h.shayesteh@gmail.com

**Abstract.** In this paper, we describe MRL Virtual team description paper to take part in Rescue Simulation League Virtual Robot Competition. This year, we designed a new system to operate our four wheels robot using ROS framework [1]. We provided such capability for our wheels robot to explore on traversal area in the simulated unknown urban environment by navigating both manually and autonomously along with detecting victims in the allotted time.

**Key words:** ROS, Gazebo, Simulation, Multi-robot, Merged Map, SLAM, Exploration, Navigation, Traversal and non-Traversal area, 3D perception, Elevation Mapping, CostMap

## 1 Introduction

Mechatronics Research Laboratory (MRL) of Qazvin Azad University, which contains teams of students and academicians. The MRL has been working on autonomous robots since its establishment in 2002. MRL-VR team is one of the research group in the MRL that has been participating since 2006 in various competitions such as IranOpen, Kharazmi, and RoboCup. The MRL-VR team consists of M.Sc. and BC.s students in different fields such as Artificial Intelligence, Software, and Information Technology Engineers. Our major focus is on developing and preparing our system for robotic competitions. We have been the champion in 2013 and 2014 RoboCup competitions.

The competition environment has changed from USARSim to Gazebo/ROS, as discussed in the future of robot rescue simulation workshop. So our team has put a lot of effort for adapting to the new environment. Since last year we have been studying on our system and come up with a list of necessary changes as follows:

– Our navigation can be improved
– Our user interface can be improved

- Our victim detection can be improved
- Our obstacle detection can be improved
- We need a merged map module
- We need an exploration module

This year, we mainly focused on merging map, Exploration, Navigation, Victim Detection. We also improved our user interface.

## 2 Team Members

The team members and their contributions are as follows:

- Sara Tohidi : Merged Map, SLAM, Navigation, Exploration, Victim Detection, Obstacle Avoidance
- Mohammad H. Shayesteh : User Interface, Supervising, System Design

## 3 System Overview

With the latest changes in Virtual Robot league, we prepared and developed our system to use ROS modules such as SLAM, Navigation, Exploration, etc. directly according to the rules. Our system is based on Ubuntu (OS) and C++ language and uses rqt for GUI.

We use Pioneer 3AT model as a ground robot. The sensors to be used are determined as Hokuyo UTM-30LX model laser scanner, RGB-D camera, and Thermal camera.

## 4 SLAM & Merged Map

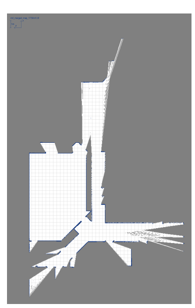

Fig. 1: Representation a merged 2-D Occupancy Grid map acquired by two wheels robots.

In the past years the use of Occupancy Grid Maps due to the need for key functions required for mobile robots such as localization, path planning, collision avoidance, have been increased [2]. The Simultaneous Localization and Mapping (SLAM) itself has been an active topic in the past decade. Since the Search and Rescue in simulated unknown urban scenarios of the Virtual Rescue Robot competition have been held by more than one wheel or air robot, the cooperation of them has been worthwhile. Therefore, provided a merged map node that updates by subscribing every presented robots map is required. This node creates a 2-D occupancy grid map Figure 1. Regarding this, by this way, we could make an efficient UI for our driver.

The Simultaneous Localization and Mapping (SLAM) addresses the problem of a robot navigating an unknown environment. Before navigating the environment, the robot seeks to localize itself using its sensors [3]. Due to this, using the gmapping package which provides a slam-gmapping node and creates a 2-D occupancy grid map [1] for all running wheels robots will be the best choice for


Fig. 2: Representation of running nodes for two wheels robots.

USAR environment to the best of our knowledge. We run one slam-gmapping node for every presented robot to gain their maps separately. The robot initials positions as blue colour and end positions as yellow colour should be marked on the final map due to the rule. We use hector geotiff package to write initials and end poses of the robots on the map.

The slam-gmapping for a single robot is not applicable anymore by raising the matter of cooperation between robots. Every node is a process that performs computation thus, running slam-gmapping node as many as presented robots perform more computation Figure 2. The multi-robot simultaneous localization and mapping (SLAM) enable teams of robots to build joint maps[4].

We are going to provide a multi-robot SLAM which is more efficient for our system performance. Yet the multi-robot SLAM has different algorithms to manage its functionality and indeed it needs more time and a lot of effort.

## 5 Obstacle Avoidance

Autonomous robots need to get a higher-level knowledge of their surrounding to navigate and explore in an unstructured environment. However, there are a few open-source packages available which help them to get knowledge. None of them has been mentioning the determination of traversal from non-traversal area to the best of our knowledge.

We represent a robust method which makes a robot to navigate on traversal area. In this case, we use the RGB-D camera to create an elevation map. So far -to the best of our knowledge-, there are two ways to convert 3D point cloud structure to an elevation map, one of them is efficient probabilistic 3D Mapping Framework based on Octrees OctoMap [5] and the other one is ROBOT-CENTRIC ELEVATION MAPPING WITH UNCERTAINTY ESTIMATES [6]. Our method provides a global elevation map which is based on OctoMap.

A graph traversal based algorithm for obstacle detection [7] represents an algorithm to compute the Traversable/Obstacle Map. We are able to use this algorithm on the global elevation map to create the 2D occupancy grid map for robot costmap. The static map layer which is filled with OccupancyGrid data incorporates all data from an external source. This is one of the options of constmap_2d which provides the robot costmap.

By using this algorithm we can measure the maximum physical slope that the robot can successfully handle as follows:

$$C\ (u,\ v,\ p,\ q) = \mid Y_{max}(u,v)Y_{max}(p,q) \mid / \parallel l(u,v)(p,q) \parallel$$

Where C is an approximation of the slope between the two cells computed in world coordinates. the numerator describes the change in elevation, and the denominator describes the distance between the two cells over which the change occurs.The two cells (u, v) and (p, q) is defined to be traversable if the C value

[1] http://wiki.ros.org/gmapping

of them is lower than the threshold. Also, the Breadth-First-Traversal algorithm used for searching cells in the map and measures its threshold between every couple of them to determine whether they are traversable or occupy for navigating Figure 3.

Fig. 3: Representation of traversal area detected by rgb-d camera.

The only problem with this module is that it takes a lot of time to do the Breadth-First search on the global scale as the map extend. This is one of our future tasks to improve this module, and we will work on it for the next computation.

## 6 Navigation & Exploration

Navigating through the simulated unknown Urban environment is the vital task of the robots in the Virtual Rescue Robot (VRR) competition. Therefore, a 2-D navigation stack takes information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile robot[2]. Besides this task, they must collect information about the position of two types of victims: alive or dead victims (according to the latest changes of the rule). In order to achieve these goals, a highly modular system has been developed.

[2] http://wiki.ros.org/navigation

It consists of several layers such as behaviour control, global path planner and trajectory generation. The high-level behaviour of the robot chooses between two scenarios moving towards a potential victim and exploring the environment. The global planner finds a path towards goal points with respect to the maximum coverage and minimum distance travelled such that we considered the potential of the map. the presented method significantly increases the performance of the robot. The TebLocalPlanner plugin used to generates local path-following to follow the global path. This approach called "Timed-Elastic-Band" [8, 9] which implements an online optimal local trajectory planner for navigation and control of mobile robots as a plugin for the ROS navigation package.

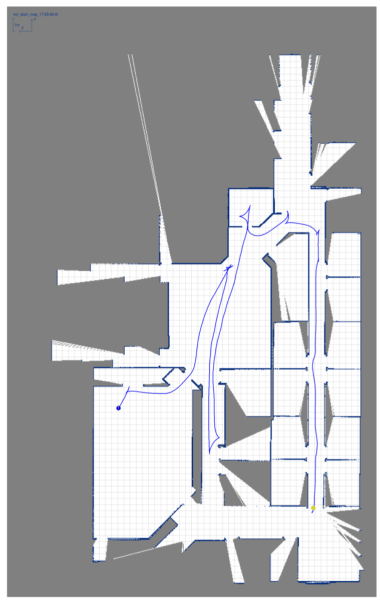

Fig. 4: Representation of running exploration node for one wheels robot.

The hector exploration planner is a planner that can both plan paths to goal points and generate goals to explore unknown environments [10, 11]. To the best of our knowledge, this is one of the two implemented modules for exploration in ROS which is efficient in rescue robot competitions. We use one of its functionalities for our wheel robots to generate goals between known and unknown area. Then, the implemented method handles it to reach every goal Figure 4.

By raising the matter of cooperation between robots, our system requires a robust feature for Performing multi-robot exploration. This will be our future task to develop our system for the next competition.

## 7 Victim Detection

The detection of victims in the disaster area and building map is the main task for every autonomous rescue robot. Therefore, it should be possible to give rescue virtual robots the capability to detect victims and landmarks autonomously. There will be two type of victims as rules have upgraded, dead and alive victims.The robots must find, identify, and report the location of as many victims in the allotted time.

For detecting alive and dead victims, the sensors like RGB-D and thermal cameras provide a global view of the environment. We designed two separated modules for detecting different victims. The thermal camera provides information to help identify temperature-related issues for detecting alive victims. The RGB-D camera provides RGB images whereby processing them we could identify dead victims.

There are many methods for detecting victims which have been developed lately. The hector team[3] has been developing various packages in ROS. The OpenCV library[4] itself provided so many features to use for detecting a variety of objects. In this year, we use hector heat detection package for detecting alive victims. Also, for detecting dead victims we use the RGB-D camera to do image processing. The RGB-D camera provides real-time RGB images and by gaining that information for Body-Parts-Detection package our wheels robot could be able to detect different body parts using Haar-like features [12]. Using this module, we can detect the following: Face, Eyes, Ears, Mouth, Nose, Upper Body, Lower Body, and Full Body[5].

At the end of every image processing for detecting a victim, robots should be able to mark a green circle for alive victims and a red circle for dead victims on the final map. By using hector object tracker we are able to add any object that we desire like alive and dead victims.

## 8 Conclusion

In this paper, we give an overview of what our team developed for this year and what our team is planning to do in future. We have re-designed our system to gain a feasible platform for this year competition. Our future task is to upgrade our merged map and SLAM to one multi-robot SLAM, and multi-robot Exploration based on machine learning topics to search in unknown environments more efficient and accurate.

[3] http://wiki.ros.org/tu-darmstadt-ros-pkg
[4] https://opencv.org
[5] https://github.com/prateekvjoshi/Body-Parts-Detection

## References


1. Quigley M., Gerkey B., Conley K., Faust J., Foote T., Leibs J., Berger E., Wheeler R., Andrew N. : ROS: an open-source Robot Operating System, ICRA Workshop on Open Source Software
2. S. Thrun, Learning occupancy grids with forward models, in Autonomous Robots. Springer, 2003.
3. Thrun, S. and Leonard, J.J., 2008. Simultaneous localization and mapping. In Springer handbook of robotics (pp. 871-889). Springer Berlin Heidelberg.
4. Thrun, S. and Liu, Y., 2005. Multi-robot SLAM with sparse extended information filers. In Robotics Research. The Eleventh International Symposium (pp. 254-266). Springer, Berlin, Heidelberg. Vancouver
5. Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C. and Burgard, W., 2010, May. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation (Vol. 2).
6. Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M. and Siegwart, R., 2014. Robot-centric elevation mapping with uncertainty estimates. In Mobile Service Robotics (pp. 433-440).
7. Kuthirummal, S., Das, A. and Samarasekera, S., 2011. A graph traversal based algorithm for obstacle detection using lidar or stereo. IEEE. In RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1579-1584).
8. P. van Turennout, G. Honderd, and L. J. van Schelven, Wall-following control of a mobile robot, in Robotics and Automation. IEEE, 1992.
9. S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk, Hector open source modules for autonomous mapping and navigation with rescue robots. RoboCup Symposium 2013, 2013.
10. Rsmann C., Feiten W., Wsch T., Hoffmann F. and Bertram. T.: Trajectory modification considering dynamic constraints of autonomous robots. Proc. 7th German Conference on Robotics, Germany, Munich, 2012, pp 7479.
11. Rsmann C., Feiten W., Wsch T., Hoffmann F. and Bertram. T.: Efficient trajectory optimization using a sparse model. Proc. IEEE European Conference on Mobile Robots, Spain, Barcelona, 2013, pp. 138143.
12. Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U. and von Stryk, O., 2013, June. Hector open source modules for autonomous mapping and navigation with rescue robots. In Robot Soccer World Cup (pp. 624-631). Springer, Berlin, Heidelberg.
13. Wirth, S. and Pellenz, J., 2007, September. Exploration transform: A stable exploring algorithm for robots in rescue environments. In Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on (pp. 1-5). IEEE.
14. Lienhart, R. and Maydt, J., 2002. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on (Vol. 1, pp. I-I). IEEE.