SOS RS Team Description Paper
RoboCup 2018 Rescue Virtual Robot League

Seyed Mostafa Ahmadi, Mohammad Hossein Gohari Nejad, and Alireza Torabian

Tehran Polytechnic, No. 424, Hafez Ave., Tehran, Iran. P. O. Box
{s.mostafa.a96,mhgoharinejad,torabian.alireza}@gmail.com

Abstract. This paper describes the approach used by Team SOS RS for participation in the 2018 RoboCup Rescue Simulation league, Virtual Robot competition. We address our methods focused on the autonomous exploration of disaster sites in order to aid victims in a simulated area. Our system provides software solutions for the addressed problem based on ROS framework and uses Gazebo as a simulation environment. In 2017, the team mainly focuses on the full autonomy of system which will be possible by improving vision system, robust navigation, efficient exploration and SOS RS utility packages. Same as the first year of our participation in the Virtual Robot league, our platform and its major components will be released publicly.

Keywords: RoboCup, Virtual Robot, Gazebo Simulation, ROS, Navigation, Multi-Agent, Victim Detection

1 Introduction

Using autonomous robots in tough situations such as earthquakes, tornados, and urban disasters to aid victims is obviously safer, faster, and more efficient. RoboCup Virtual Robot league has challenged this problem in a simulated environment. Team SOS of CEIT department of AUT has been participating in 2D simulation league since 2002 with outstanding results. The Advent new simulation environment and ROS in the Virtual Robot league in RoboCup lead us to find a new branch of SOS team in order to work on more realistic and crucial challenges. Team SOS RS established in late 2017 driven by the goal of creating a team of heterogeneous autonomous robots. This is first competition which our team is going to participate.

For RoboCup 2018 our goal is to reach a stable base code. Major novel works of us are:

1. Development of various Convolutional Neural Networks to create robust victim detection system.
2. Development of a novel robotic team autonomy.
3. Using a heterogeneous team of robots, including UAV and different UGVs.
4. Development of a frontier-based exploration.
5. Release of SOS RS Controller, Human-Robot interface in order to provide a control panel for the human agent to control robot team.

2 System Architecture

In this section, we are going to describe new developments of the previous system. To reach the purpose of robust navigation and exploration, we need a reliable perception system. We have enhanced some parameters of SLAM packages we used before to fulfill our needs. We are planning to work on this part in future, for this year, we have focused on multi-robot exploration, task planning, and victim detection strategies.

2.1 Navigation

SLAM addresses the problem of building a map of an environment from a sequence of landmark measurements obtained from a moving robot. Since robot motion is subject to error, the mapping problem necessarily induces a robot localization problemhence the name SLAM. The ability to simultaneously localize a robot and accurately map its surroundings is considered by many to be a key prerequisite of truly autonomous robots [2,3,4]. Kalman filter-based algorithms require time quadratic in the number of landmarks to incorporate each sensor observation. Though EKF SLAM suffers from a $\mathrm{O(K^2)}$ complexity where K being the number of landmarks, In contrast, FastSLAM has an $\mathrm{O(M} \log K)$ complexity with M = const denoting the number of particles. Because of this, we decided to use the FastSLAM algorithm for simultaneous localization and mapping [5]. For using this approach we didn't satisfy of this single fact. According to "EKF SLAM vs. FastSLAM A Comparison" by Michael Calonder in EKF SLAM all densities involved in the calculation of the posterior, so this makes EKF SLAM more precise but it has a big overhead on memory and CPU utilization. So we developed our own FastSLAM to reach more real-time factor and we sacrificed accuracy for speed!

2.2 Map Merging

We use pair-wise relations between our robots. Each robot is capable of mapping the environment on its own and is also capable of communicating intermittently with other robots. When robots are in communication, they can share information and coordinate their exploration activities.

Each pair of robots can have four types of interactions:

1. **No interaction** When two robots are not in communication range.
2. **Hypothesis generation** The robots are in communication range and can communicate but they don't have their relative locations. In this state, one of the robots receives another robot's sensor data and try to estimate the first robot's location using its own map.
3. **Hypothesis verification** Robots can communicate and verify a location hypothesis determined in the hypothesis generation phase. In the case of uncertainty, the robots can try to meet. If the robots dont meet at the expected location, the hypothesis is rejected and they continue with the hypothesis generation phase.

4. **Coordinated exploration** If the robots meet each other so they can determine their relative locations and share their maps and perform coordinated exploration. A nice feature of this interaction type is transitivity, i.e. if robot i and j can share their maps, and robot j and k can share their maps, then all three robots can build a combined map and make a cluster. Each cluster has a head that is responsible for data combination and robots coordination. So it's not necessary to have direct connections between all of the cluster's robots [6].

2.3 Autonomous Exploration

For this task we have created a distributed and non-centralized Algorithm based on the frontier based exploration. In existing frontier based algorithms we try to find frontier regions. A frontier region is the border between explored and not explored area. When using Occupancy grid data structure to save the map each element has three possible values: Free, Occupied, Unknown. As a result each frontier is defined as center element of an array of elements with their values being Unknown and having at least one neighbor element with Free value. After frontiers are found, each robot will assigns a cost to each frontier based on its distance from robot and then updated the cost based on the frontiers chosen by other robots.

2.4 Human-Robot Interface

SOS RS Panel is our Control Panel visualizer which is a plugin for rqt that allows the human operator to see robot state, map, camera feed, and switch between robot controllers. This year, we have updated the plugin, bug fixes have been done and it is publicly available on team's repository. Any contribution to this package is appreciated. See figure 6 for a screenshot.

3 Innovations

As our first year, we have used heterogeneous robots incorporation. Also, we developed a new frontier based approach to solve exploration problem. For map merging problem, we used a new way of merging each robot's local map.

4 Conclusion and Future Works

After all, these new contributions are under development and they need test and improvement. Besides we have in mind to move toward new challenges like 3D mapping, map merging without knowing initial pose and so on. As well we have to consider realistic wireless communication between mobile robots using distributed communication, which in near future will be added to the competition. We also want to develop a novel approach to solve exploration using victim made the sound problem.

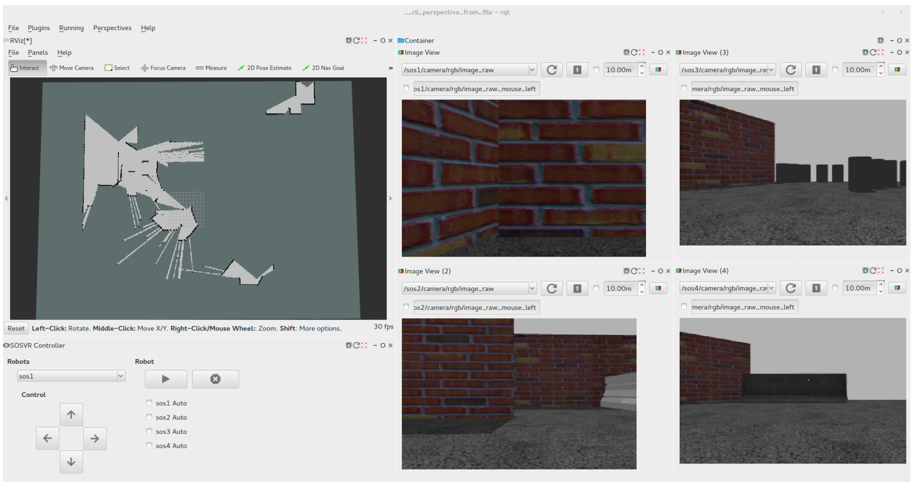

Fig. 1. Screen shot of SOS RS controller. The shared map is on the top left. Control buttons are at bottom left and camera feeds are shown in right half of panel.

References

1. M. Taherahmadi, S. Azami, M. GohariNejad, M. Ahmadi, and S. Shiry, (2017). SOSvr Team Description Paper: RoboCup 2017, Rescue Virtual Robot League.
2. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions of Robotics and Automation, 2001.
3. D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. AI-based Mobile Robots: Case studies of successful robot systems, MIT Press, 1998.
4. C. Thorpe, and H. Durrant-Whyte. Field robots. ISRR-2001.
5. M. Calonder. EKF SLAM vs. FastSLAM A Comparison, EPFL Computer Vision Lab, 2006
6. D. Fox, B. Stewart, K. Konolige, B. Limketkai, and J. Ko. Map merging for distributed robot navigation, 2003