SOS CS Team Description Paper RoboCup Asia Pacific 2018 Rescue Virtual Robot League

Hannah Faghihi,Negar Rajabi

Tehran Polytechnic, No. 424, Hafez Ave., Tehran, Iran. P. O. Box
{faghihihannah,negar.271999}@gmail.com

Abstract. This paper outlines the method used by team SOS CS for participation in the 2018 RoboCup Asia Pacific Rescue Simulation league, Virtual Reality competition. In order to aid victims in a simulated area, we address our methods focused on the autonomous exploration of disaster sites. Based on ROS framework and using Gazebo as a simulation environment, our system provides a solution for the addressed problem. For these competitions, our team mainly focuses on the enhancement of autonomously explored area by our multi-robot system.

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

1 Introduction

In situations such as earthquake tornado and etc using autonomous robots can be more efficient. By using these robots the situation can be managed more accurately. Also having humans to aid the victims can be very dangerous, therefore we use the robots to avoid further injuries. Team SOS CS of MCS department of AUT has been established this year. We hope to present our best in our first year of competition.

2 System Architecture

Our robots system directly performs on ROS and Gazebo.by using a reliable perception system we can reach the purpose of the robust navigation and exploration. Therefore we enhanced some parameters of SLAM package. This year our main focus is on multi-robot exploration and task planning. [1]

3 Navigation

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 and planning to navigate through which path, until we find a victim, then we will decide what to do next. 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 makes SLAM one of the most critical challenges in Rescue Simulation. For these purposes we used this combination of this packages from ROS Navigation Stack.

1. Gmapping
   the gmapping package provides laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called slam-gmapping. Using slam-gmapping, you can create a 2-D occupancy grid map (like a building floorplan) from laser and pose data collected by a mobile robot.[2]
2. AMCL
   with using amcl which is a probabilistic localization system for a robot moving in 2D we can reach an optimized and fast solution for localization part. It implements the adaptive (or KLD-sampling) Monte Carlo localization approach (as described by Dieter Fox) which uses a particle filter to track the pose of a robot against a known map.[3]
3. Hector
   hector mapping is a SLAM approach that can be used without odometry as well as on platforms that exhibit roll/pitch motion (of the sensor, the platform or both). It leverages the high update rate of modern LIDAR systems like the Hokuyo UTM-30LX and provides 2D pose estimates at scan rate of the sensors (40Hz for the UTM-30LX). While the system does not provide an explicit loop closing ability, it is sufficiently accurate for many real world scenarios. The system has successfully
4. **Move Base**
   The move-base package provides an implementation of an action that, given a goal in the world, will attempt to reach it with a mobile base. The move-base node links together a global and local planner to accomplish its global navigation task. It supports any global planner adhering to the nav-core: BaseGlobalPlanner interface specified in the nav-core package and any local planner adhering to the nav-core: BaseLocalPlanner interface specified in the nav-core package. The move-base node also maintains two costmaps, one for the global planner, and one for a local planner that are used to accomplish navigation tasks. [4]
5. **Odometry**
   Odometry is the use of data from motion sensors to estimate the change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location. This method is sensitive to errors due to the integration of velocity measurements over time

to give position estimates. Rapid and accurate data collection, instrument calibration, and processing are required in most cases for odometry to be used effectively.

4 Map Merging

Each robot is capable of building a map on its own. They can also interact with other robots and share their knowledge of the map. When robots are not in the communication range no interaction is made. However, if they are in the communication range and are also able to communicate but dont have their relative location, one of the robots receives another robots sensor data and try to estimate the first robots location using its own map. In another case, hypothesis generation phase determines a location hypothesis. The robots communicate and try to meet. if the robots don't meet in the determined location the hypothesis is rejected and they continue with the hypothesis generation phase. If they meet, they can determine their relative locations and share their map so that the coordinated exploration will be performed. The feature of this interaction is its transitivity. This feature will lead a group of robots to merge their maps and make a cluster. Each cluster has a head that gathers the robots coordination, therefore, it is not necessary to have a direct connection between all of the clusters robots. There are two approaches to the problem of merging different maps from robots and building a globally shared map. when the coordination and orientation of robots are known the map of each robot will be put in the right place. Otherwise, by using a template matching algorithm, the overlapping areas in maps will be found to be merged based on their similarities.[5]

5 Autonomous Exploration

In lack of direct connection Controlling robots manually is not quite possible. In the multi-agent scenario, Since each robot needs, at least one operator for reliable functioning the need of various human operators can be challenging. Therefore autonomous exploration has great importance. Since this year our focus is on autonomous exploration we decided to split the map and give each agent an area to explore. In the end, By using frontier-based exploration each robot has a detailed information of the explored area. The central idea behind frontier-based exploration is: To gain the newest information about the world, move to the boundary between open space and uncharted territory. Frontiers are regions on the boundary between open space and unexplored space. When a robot moves to a frontier, it can see into unexplored space and add the new information to its map. As a result, the mapped territory expands, pushing back the boundary between the known and the unknown. By moving to successive frontiers, the robot can constantly increase its knowledge of the world. By sharing the gathered information with other robots we can provide a detailed map for better exploration in sufficient time.

Fig. 1. Map Merging

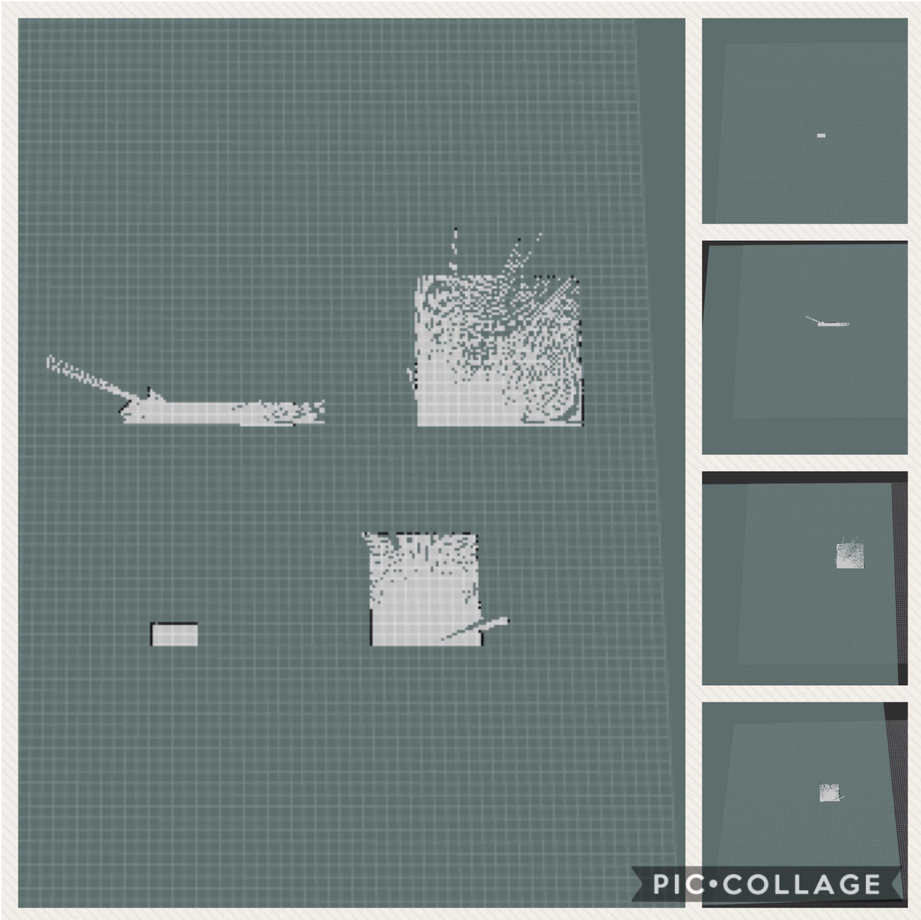

Fig. 2. Multi Robots

6 Human-Robot Interface

to allows the human operator to see robot state, map, camera feed, and switch between robot controllers. We use SOS RS Panel is our Control Panel visualizer which is a plugin for rqt.

7 Innovations

As our first year in competition, To overcome the exploration problem we have developed a new frontier based approach. For map merging problem, we used a new way of merging each robot's local map.

8 Conclusion and Future Works

All of the strategies will be improved and new features will be added:

1. Development of algorithms to create robust victim detection system.
2. Using PID control for navigation purpose.
3. Improving the autonomous state machine.
4. Development of Exploration, a new exploration system for autonomous efficient exploration of UGVs.
5. Improving Controller, Human-Robot interface in order to provide a control panel for a human agent to control robot team.

References

1. wiki.ros.org site
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 on 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. D. Fox, B. Stewart, K. Konolige, B. Limketkai, and J. Ko. Map merging for distributed robot navigation, 2003