Team-IMU Team Description Paper for RoboCup 2020 Virtual Rescue Simulation

Haluk Bayram, Şule Z. Aydın, Zeynep Yağbasa, Enis Çeri, Abdulhamit Bülbül

Istanbul Medeniyet University,

Department of Electrical and Electronics Engineering,

Field Robotics Laboratory and Robotics Club,

Istanbul, TURKEY

haluk.bayram@medeniyet.edu.tr

{aydinsulezeynep, zeynepyagbasa98, enisceri, emreb43}@gmail.com

Abstract. This paper presents a short description of the Team-IMU's plans for rescuing victims in ROBOCUP 2020 Virtual Rescue Simulation Competition. This will be the first time we participate in this competition. We focus on developing algorithms for enabling a heterogeneous multi-robot system cooperatively search and rescue victims. Since ground robots cannot navigate in rough terrains, aerial robots are added to the team. The aerial robots can provide visual information which is used for coordinating and localizing the ground robots. The multi-robot system has four main components: cooperative exploration and mapping, navigation, communication and victim detection. We validate the effectiveness of the algorithms to be developed using ROS and Gazebo.

1. Introduction

Natural disasters occur in many parts of the world. We cannot know the exact time when they happen. Unfortunately some of these disasters could be devastating. After such disasters, search and rescue (SAR) operations are required. SAR team members may not always reach victims due to dangerous conditions, such as leakage of radioactive materials and risk of explosion. Therefore, mobile robotic systems assist SAR teams. We, as TEAM IMU, aim to develop a heterogeneous multi-robot system for SAR operations.

TEAM IMU was established in 2018 and consists of members from Field Robotics Laboratory and Robotics Student Club at Istanbul Medeniyet University. TEAM IMU develops algorithms for various robotic problems and implements them on real robotic systems. Team leader, Haluk Bayram, works as a full-time professor in the department of Electrical and Electronics Engineering at Istanbul Medeniyet University. He is the director of Field Robotics Laboratory, working on multi-robot systems and developing task allocation, target localization and navigation algorithms for ground, aerial and surface robots. The publication list can be seen in [1]. The first competition TEAM IMU participated in is the TUBITAK International Unmanned Aerial Vehicle Competition in 2018. In this competition, the team participated in the fixed wing category with the HORIZON vehicle. TEAM IMU participated in the Rotary Wings Category in the TUBITAK International Unmanned Aerial Vehicle Competition in 2019. In the conceptual design phase, the team's VTOL design was placed seventh out of 187 teams competing. The aim of these contests is to develop Unmanned Aerial

Vehicles (UAVs) for civil use, which will help people especially in emergencies, such as fire or accident.

In this competition, we validate the correctness and efficiency of our algorithms using Robot Operating System (ROS) [2] and Gazebo simulation environment. ROS provides a framework with which roboticists can easily implement their algorithms on real robots. ROS is compatible with Gazebo, which enables us to simulate various robots in 3D environments. SAR simulations require realistic environments. For the competition, USARSim [3] was used as a simulator in the past competitions, and later Gazebo has started to be used. Over time, the USARSim environment was transferred to Gazebo. We also use USARGazebo [4] and RoboCup2019RVRL_Demo [5] to test our algorithms on ground and aerial robots. We have a heterogeneous multi-robot system as a SAR team consisting of differential drive ground robots and quadrotor robots. The robots are equipped with various sensors, such as camera, thermal camera, laser scanner. Depending on the requirements of the SAR task, the multi-robot system decides how many aerial and ground robots are needed and which sensors are required. The task distribution in the team is as follows:

- Supervising: Haluk Bayram
- Multi-robot coordination using aerial robots: Abdulhamit Bülbül, Enis Çeri
- Mapping, navigation and autonomous exploration using ground robots: Ş. Zeynep Aydın, Zeynep Yağbasa
- User interface: Abdulhamit Bülbül, Enis Çeri
- Victim detection with YOLO v3: Ş. Zeynep Aydın, Zeynep Yağbasa

2. System Overview

2.1. Robots

We have a robot team consisting of two different robots, Pioneer 3-AT [6] for land exploration and mapping, and hector_quadrotor [7] for aerial exploration and mapping. A sample scenario can be seen in Figure 1. The robots are equipped with RGB camera, thermal camera GPS, IMU and Hokuyo sensors. RGB and thermal cameras are used for victim detection. In addition, we add to Pioneer 3-AT robots a platform on which hector quadrotors can land and take off. We place unique AR tags [8] on these landing platforms. Aerial robots make use of the AR tags to localize and identify the ground robots. In this way, the robots can operate cooperatively. The task weights of the robots may vary depending on the characteristics of the disaster area, and the sensors on the robots are activated in line with these tasks.

Figure 1: Heterogeneous multi-robot team. (Left): A team of UGV and UAV are cooperatively performing their search and rescue task. (Right): A UAV is taking off on a UGV.

2.2. Software Architecture

The system consists of a base station and robots shown in Figure 1. The operator can monitor and send commands to the robots via a user interface on the base station. ROS packages are available for mapping, navigation and autonomous exploration. First, we implement these existing packages, then depending on their performance we plan to improve their performance. In mobile robots, SLAM-GMapping [9] is used for mapping, TEB Local Planner [10] algorithm and artificial potential function [11] is used for navigation, and the exploration algorithm to be applied based on Wavefront [12] algorithm is used for autonomous exploration. hector_slam [13] is used for mapping in quadrotors. A local map generated by each robot is transmitted to the base station and a global map is generated using Map-Merge Package [14] and this map is transmitted to each robot. The base station coordinates the multi-robot team. In case of communication loss, the robots continue their assigned task in a distributed manner.

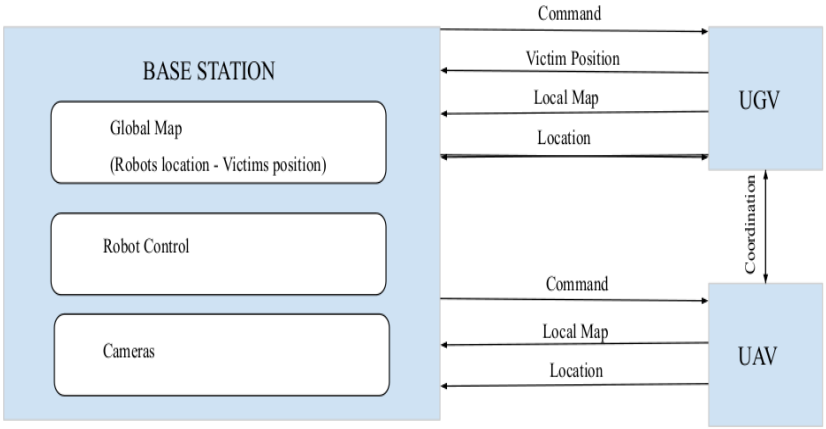

Figure 2: Overview of the system infrastructure

3. Heterogeneous Multi-Robot System

SAR tasks are performed under difficult conditions. A single robot may not overcome these tasks which require various sensory, computation or mechanical capabilities. Therefore we establish a multi-robot system, which speeds up the SAR task. Field knowledge is also required to adapt these robotic systems to disaster areas [15]. Considering the variability of disaster areas, two types of robots, ground and aerial robots, are used in the multi-robot system. Considering the field knowledge we have acquired (One of our team members, Zeynep Yagbasa, visited Elazig after the earthquake occurred on 24 January, 2020) and the problems of disaster area in the USARSim environment, we develop a flexible and robust algorithm which determines the number of robots in the robot team and which type of robot is predominantly used.

3.1. Ground Robots

Ground robots play a crucial role in the search and mapping tasks. Ground robots have been used throughout the RoboCup Virtual Simulation competitions. It is advantageous to use a ground robot in some indoor and outdoor environments where aerial robots have difficulty of flying due to weather conditions or limited flight space. As a ground robot, we choose the Pioneer 3-AT robot to use in indoor and outdoor environments. Each of these ground robots has odometry, laser scanner and camera and they are equipped with the same ROS based software packages. We develop a multi-agent control system for the ground robots, which has the following stages: following the leader, autonomous exploration and victim detection. According to the control model of the team, the next location of each robot is determined based on the location of other robots, the state of the environment and the tasks of the robots [16]. These ground robots perform the SAR task along with aerial robots while in

communication with the base station. However, there may be disaster areas where ground robots cannot operate. For instance, they may not be effective in environments, such as long indoor, long outdoor and maze. Based on the environment type, ground or aerial robots take main responsibility of the SAR task.

3.2. Aerial Robots

Aerial robots have become an important part of the SAR tasks. It is advantageous to use aerial robots in long and high indoor, and large outdoor environments because they move faster. As an aerial platform, we use the quadrotors equipped with camera, GPS, laser range finder and thermal sensors. In addition, each quadrotor is able to land and take off on the platform on the ground robots, so the team performs the SAR task cooperatively. In cases of communication loss among the ground robots, the quadrotors tracking the ground robots can help coordination among them. Based on the information obtained from the sensors, our system decides which robotic system can effectively complete the task.

3.3. Communication

Communication between robots is an important issue. In real life, unlimited communication is not possible. There may occur interruptions in communication or robots may be in an environment which blocks communication. Hence we aim to integrate a realistic communication infrastructure with Gazebo. For this purpose, we have begun to integrate the NS-3 network simulator [17] with ROS and Gazebo.

4. Mapping & Searching

4.1. Mapping

Mapping is essential for robots to localize themselves in the environment and explore an unknown environment. In our design, we implement GMapping [9] on the ground robots and hector_slam [13] on the aerial robots. A sample map shown in Figure 3 is generated by a single robot using GMapping. Each robot shares its local map with the base station. In turn, the base station generates a global map by merging the local maps the robots sent, then it broadcasts the resulting map to the robots. The global map is updated as long as the robots send their local maps.

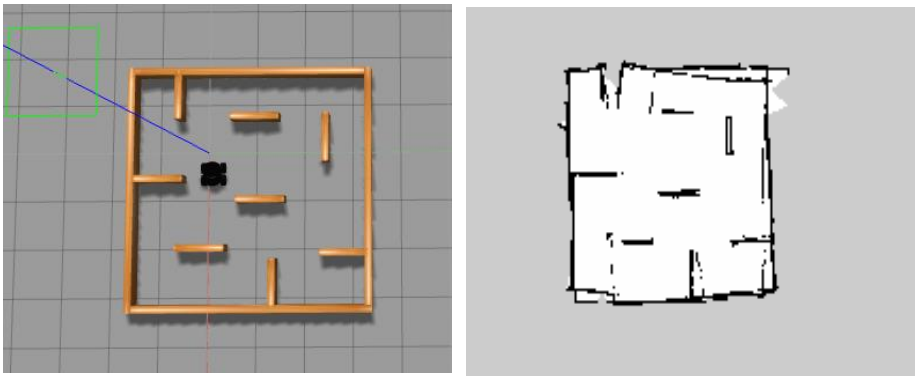

Figure 3: Mapping an environment using GMapping. (Left): A single ground robot is generating the map of the environment, (Right) The resulting map of the environment.

4.2. Navigation

To make the robots navigate to a location, we implement artificial potential function based navigation algorithm [11]. We modify the algorithm so that the robots execute the navigation algorithm in a coordinated or distributed manner. When all the robots

have all-to-all communication, the algorithm is executed in a coordinated manner. Otherwise, since the robots do not have access to all the other robots' state information, they use local information for navigation.

4.3. Cooperative Exploration and Search

Cooperative exploration is very important for the multi-robot team to complete their tasks in an unknown environment. The basic idea behind exploration is that the nearest frontier point is found as soon as possible for exploration by robots and then a path to this point is generated by the path planning algorithm. Exploration continues until no frontier points are left. The frontier points obtained after mapping with the frontier-based algorithm, widely used for exploration, are transferred to the queue data structure. Paths to these points are calculated by the path planning algorithm [18]. The Wavefront algorithm was developed based on the frontier-based algorithm. In this algorithm, instead of considering all frontier points transferred to the queue, less frontier points are taken into account as a result of an evaluation. The evaluation step reduces the time spent [19].

The Wavefront algorithm determines the frontier point after generating the map by Gmapping. Taking into account the regions, the algorithm labels the areas as occupied, free or unknown. The robot goes to this point according to the path planning algorithm. This process continues until there is no frontier point. With this algorithm, frontier points are found fast and the integrity of the discovery environment is ensured [12]. An autonomous exploration algorithm developed based on the Wavefront algorithm is used for ground robots.

5. Victim Detection

Victim detection is very important for search and rescue. While searching for victims, which is one of the main objectives of the contest, we use YOLO V3 [20] in which a trained neural network processes RGB images received from the camera to detect victims. In addition, thermal cameras are also used for the victim detection in dense smoke environments where RGB cameras do not provide sufficient information.

6. User Interface

A user interface in the base station is designed using QT and Python. With the user interface, the operator can manually select the number and types of robots according to the disaster scenario, can have access to the cameras of each robot and to the global map, and can monitor the locations of the robots and the detected victims on the map.

7. Conclusion & Future Work

In this article, we introduce the system we plan to design for the RoboCup Virtual Simulation competition. We aim to develop a heterogeneous multi-robot system consisting of aerial and ground robots to perform SAR tasks. We plan to make contributions in the multi-robot coordination and exploration. For mapping, we make use of existing ROS packages. We test and validate our algorithms using ROS and Gazebo. Furthermore, to make the communication realistic, we integrate NS3 network simulator with ROS and Gazebo.

References

1. Haluk Bayram, https://scholar.google.com/citations?user=HepCUqoAAAAJ
2. ROS/Introduction - ROS Wiki. http://wiki.ros.org/ROS/Introduction. (Accessed on 27.02.2020)

3. The Future Of Rescue Simulation Workshop, 29 FEB - MRT 4, 2016 . https://staff.fnwi.uva.nl/a.visser/activities/FutureOfRescue/. (Accessed on 27.02.2020)
4. m-shimizu, RoboCup Rescue Package: A package adding USARSim interface to Gazebo. https://github.com/m-shimizu/RoboCupRescuePackage. (Accessed on 26.02.2020)
5. RoboCup-RSVRL/RoboCup2019RVRL_Demo. https://github.com/RoboCup-RSVRL/RoboCup2019RVRL_Demo. (Accessed on 26.02.2020)
6. Robots / AMR_Pioneer_Compatible. http://wiki.ros.org/Robots/AMR_Pioneer_Compatible. (Accessed on 24.02.2020)
7. hector_quadrotor - ROS Wiki, http://wiki.ros.org/hector_quadrotor. (Accessed on 24.02.2020)
8. ar_track_alvar - ROS Wiki. ar_track_alvar, http://wiki.ros.org/ar_track_alvar.(Accessed on 27.02.2020)
9. gmapping - ROS Wiki. http://wiki.ros.org/gmapping. (Accessed on 26.02.2020)
10. teb_local_planner - ROS Wiki. http://wiki.ros.org/teb_local_planner. (Accessed on 26.02.2020)
11. Karagöz, C. Serkan, H. Işıl Bozma, and Daniel E. Koditschek. "Coordinated navigation of multiple independent disk-shaped robots." IEEE Transactions on Robotics 30.6 (2014): 1289-1304.
12. TANG, Chunhua, et al. Autonomous Indoor Mobile Robot Exploration Based on Wavefront Algorithm. In: International Conference on Intelligent Robotics and Applications. Springer, Cham, 2019. p. 338-348.
13. hector_slam - ROS Wiki. http://wiki.ros.org/hector_slam. (Accessed on 25.02.2020)
14. multirobot_map_merge - ROS Wiki. http://wiki.ros.org/multirobot_map_merge. (Accessed on 26.02.2020)
15. DELMERICO, Jeffrey, et al. The current state and future outlook of rescue robotics. Journal of Field Robotics, 2019, 36.7: 1171-1191.
16. KOZHEMYAKIN, Igor, et al. Multi-agent Control System of a Robot Group: Virtual and Physical Experiments. In: International Conference on Interactive Collaborative Robotics. Springer, Cham, 2019. p. 40-52.
17. Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator." Modeling and tools for network simulation. Springer, Berlin, Heidelberg, 2010. 15-34.
18. frontier_exploration - ROS Wiki. http://wiki.ros.org/frontier_exploration. (Accessed on 26.02.2020)
19. TOPIWALA, Anirudh; INANI, Pranav; KATHPAL, Abhishek. Frontier based exploration for autonomous robot. arXiv preprint arXiv:1806.03581, 2018.
20. Marko B., YOLO ROS: Real-Time Object Detection for ROS. https://github.com/leggedrobotics/darknet_ros. (Accessed on 25.02.2020)
21. ZHENG, Kaiyu. Ros navigation tuning guide. *arXiv preprint arXiv:1706.09068*, 2017.