RoboCup Rescue 2019
TDP Virtual Robot Simulation
ATR (US)

Jong-Hoon Kim, Xiangxu Lin, Nathan Kanyok, Alfred Shaker, Pavan Poudel, Irvin Cardenas, Nadia Karina Camacho Cabrera, HyunJae Jeong, and Gokarna P. Sharma

Kent State University, USA
{jkim72, xlin10, nkanyok, ashaker, ppoudel1, icardena, ncamacho, hjeong3, gsharma2}@kent.edu
http://www.atr.cs.kent.edu/

Abstract. ATR Kent is a research team from Kent State University, located in Kent, Ohio. The lab is oriented towards research in telerobotics, with emphasis on immersion and telepresence. The first robotics competition we participated in was the Disaster Response Challenge, which was a part of the World Robot Summit (WRS) 2018 in Tokyo, Japan. The 2019 RoboCup Rescue Simulation League presents us with an opportunity to expand our research in robot simulation and share our knowledge with fellow competitors, with the ultimate goal of innovating the field of robotics.

1 Introduction

ATR Kent, of Kent State University, is a competition team with backgrounds in Advanced Telerobotics Research. We focus on interaction and immersion within the realm of robotics, applying our methodology to Telebot-2, a disaster response robot which was developed for the WRS 2018 Disaster Response Competition (see Fig. 1 showing the participation at WRS 2018). The virtual robot simulation competition of RoboCup presents us with an opportunity to improve our simulation research.

In the past, the Advanced Telerobotics Research Lab has mainly focused on improving the control and operation of physical robots. Past projects include a teleoperation suit that allows an operator to remotely control the torso of a robot [11], and a Virtual Reality control interface for a remote robot's head. It is the case that the lab currently lacks having a strong simulation base. RoboCup gives ATR Kent an opportunity to establish research within the realm of robotics simulation, as well as improve the community with our unique approach to robotics.

Recent lab publications include: Telesuit: An Immersive User-Centric Telepresence Control Suit [11], Engage/Disengage: Control Triggers for Immersive Telepresence Robots [17], Towards an Ethical Framework for Robot Development [16]. and Performance Comparison of NVIDIA accelerators with SIMD, Associative, and Multi-core Processors for Air Traffic Management [20].

Fig. 1: ATR Kent at WRS 2018

2 Team Members

ATR Kent is comprised of all levels of researchers at the university level. Members range from being early in their undergraduate education to nearing the end of their PhD tenure. This gives the lab a dynamic range of skills and experiences. The ATR Kent RoboCup Simulation League team is composed as follows:

- Jong-Hoon Kim -*Advisor Professor*
- Gokarna P. Sharma -*Senior Adviser*
- XiangXu Lin - *Team Leader*
- Alfred Shaker - *Drones*
- Irvin Cardenas - *System Architecture*
- Nate Kanyok - *Ground Robots*
- Pavan Poudel - *Algorithms*
- Nadia Karina - *Victim Identification*
- Jared Butcher - *Victim Identification*
- HyunJae Jeong - *Electrical Engineering*

3 System Overview

Our research and testing are conducted on several machines, each running Ubuntu 16 [10], allowing for access to both ROS Kinetic [6] and Melodic [7]. The specific hardware varies from machine to machine. Thus, we are able to test our applications with several different levels of computing power.

The overall system architecture and the execution flow is illustrated in Fig. 2. The system consists of two sets of robots: *Ground Robots* and *Drones*. Both of the sets have the function of creating and exploring the map, as well as finding the victims. Due to the nature of drones, they can provide fast travel speed to produce faster mapping, with some compromises in accuracy. During the period while the drone(s) is exploring the map, it will also detect potential victims along the way and mark the location on the map. The ground robot(s) assignment is then to reach the potential location of victims after receiving the data provided by the drone on the map. It also prioritizes each location of the detected potential victims by drone and calculates which path to take and which location to go in order to spend the minimum amount of time to achieve maximum performance.

The following subsections describe each module in more details.

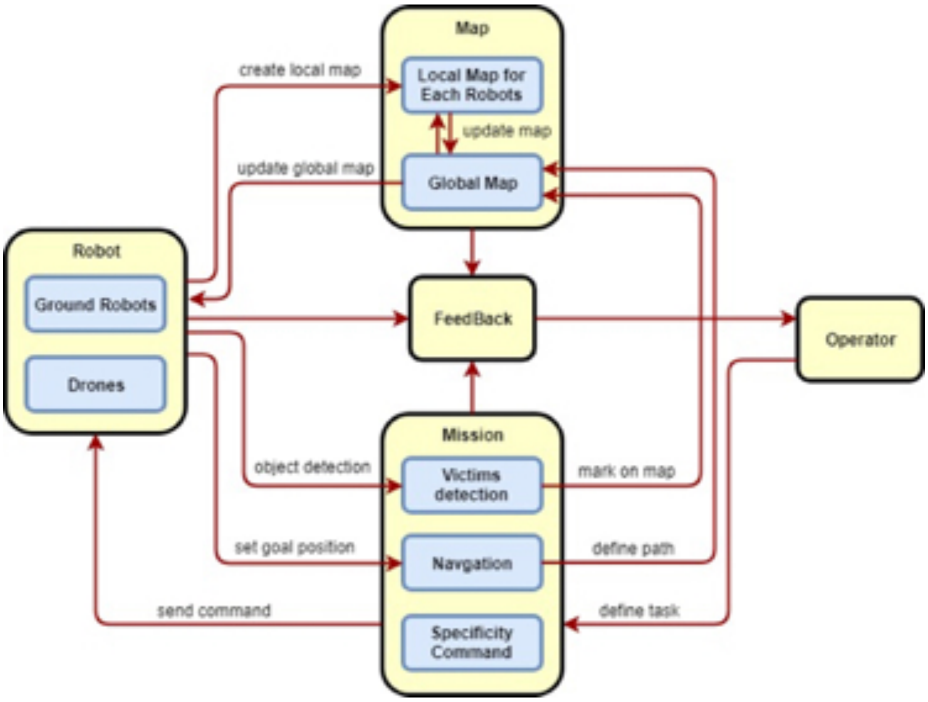

Fig. 2: System Architecture

3.1 Mapping

During the process of path planning, the robots continuously learn about the environment. Simultaneous Localization and Mapping (SLAM) [12] allows for a robot to navigate an environment efficiently by creating a local representation of itself on a 2D map.

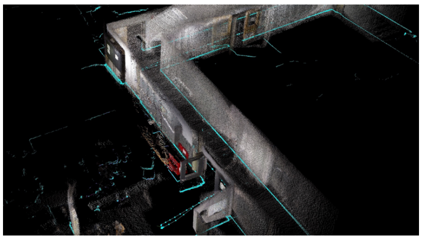

Fig. 3: RTAB-Mapping with 3D point cloud and 2D map

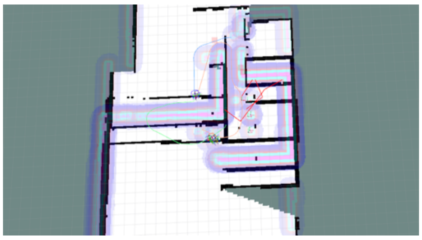

Fig. 4: rrt_exploration with 3 robots autonomously exploring the world

Mapping with Drone(s): Our robot team will be comprised of several different robots, with a range of form factors. Thus, it is essential to have separate mapping algorithms that correspond to the design of the robot. We are using PX4 [4], an open source flight control software for simulated and real drones, as well as other unmanned vehicles. We plan to use the OctoMap library to dynamically generate volumetric 3D environments from sensor data [3]. This data is then used by the simulated drone to navigate the environment safely and avoid collision. We use the Gazebo Rotors Simulator library and ROS [14]. MAVROS is the officially supported bridge between ROS and MAVLink protocols used by PX4 that we will use in this simulation. Additionally, we plan to use the *rtabmap_ros* (Real-Time Appearance-Based Mapping) as shown in Fig. 3 package for the 3D/2D mapping by drone. The *rtabmap_ros* package is a RGB-D SLAM approach based on a global loop closure detector with real-time constraints[9]. We can use the *rtabmap_ros* package to generate a 3D point clouds of the environment as well as to create a 2D occupancy grid map for the navigation. Then we will apply rrt_exploration Fig. 4 on top of given 2D map to calculate possible path for robot to autonomous travel inside the map.

Mapping with Ground Robot(s): As mentioned above, our multi-robot team requires separate modules depending on the robot's design. We plan to employ *P3AT* for our ground robot. The algorithms used for map production in these robots are as follows:

We are using rrt-exploration [21] as our initial approach for the ground robot mapping algorithm. It applies a Rapidly-Exploring Random Tree (RRT) algorithm to different robots, and then creates a separate map for each individual robot. The maps will then be merged via map_merger ROS package. Additionally, we plan to extend our methods by using the *explore_lite* [1] package available in ROS. The *explore_lite* package provides ROS node for autonomous exploration also based on the Frontier search algorithm [22]. Unlike other similar packages, *explore_lite* does not create its own *costmap*, which makes it more efficient and lightweight.

3.2 Map Integration

Following what we describe above, each robot generates a partial map during the exploration phase. Then we merge the maps to make a general map that is available for all robots. We are using the 'multirobot map merge' [15] package for ground robots to merge the separate maps into one, and 'map merge 3d' [2] package for drones. We can then cross compare these two type of merged maps to achieve better localization results.

3.3 Navigation

For the ground robot, we use 2d navigation stack [8] as the foundation, which take input from odometry, sensor message, and a goal pose to give a safe velocity command that are sent to a mobile base.

Drone navigation will be done using the PX4 software package that uses MAVROS, the official ROS package for interfacing with MAVLink. This allows for dynamic navigation based on the previously talked about dynamic octomapping. This package also includes tools for collision avoidance that we utilize to make sure the drone doesn't get stuck or damaged.

3.4 Victim Detection

For victim detection, we will be using You Only Look Once (YOLO) [19] object detection algorithm to identify potential victims in the environment. It loads each image only once and applies a single neural network to the entire image. The network will then divide the image into regions, then predicts bounding boxes and probabilities for each region. We will implement this algorithm on both ground and drone robots. The drone will first give potential location of victims, and then the ground robot will go to each location to have a closer look at each object to determine if it is really a human or not.

For heat detection, we will be using color edge detection algorithms [13] through OpenCV over thermal images to identify whether the victim is alive or not. If the colors detected are not close enough to the typical colors values seen on the living, the victim will be declared dead.

For sound detection, we plan to run voice activity detection [23]. For every distinct sound we pick up, we plan to compare it with a trained model that will identify whether it is a voice or not. In order to train that model we will use several voices of different rangers to improve the voice activity's detection.

4 Robot Models

We plan to implement two different robots in our simulation environment: Pioneer 3-AT(p3at) for ground robot and hector quadrotor for drone. The p3at is a research robot developed by Pioneer, and serves as a platform for four-wheeled robot development [5]. Hector quadrotor is a simulated quadrotor that have complete modeling, control and simulation of quadrotor UAV systems[18].

Table 1: Robot List

Name	Type	Sensors	Amount
p3at	Differential	Hokuyo, thermal camera	TBD
hectorquadrotor	Drone	Hokuyo, thermal camera	TBD

5 Conclusion

We at ATR Kent are excited for the opportunity to participate in the 2019 RoboCup Rescue Simulation League. It serves as an excellent opportunity to improve our simulation core within the lab. Our knowledge from past competitions will be implemented into this new adventure. We hope to share our approach to robotics with other competing teams and accelerate the field of robotics.

References

1. explore_lite documentation, http://wiki.ros.org/explore_lite
2. map_merge_3d documentation, http://wiki.ros.org/map_merge_3d
3. Octomap models with ros, https://dev.px4.io/en/simulation/gazebo_octomap.html
4. Open source for drones, https://px4.io/
5. Pioneer 3-at documentation, https://cyberbotics.com/doc/guide/pioneer-3at
6. Ros kinetic, http://wiki.ros.org/kinetic
7. Ros melodic, http://wiki.ros.org/melodic
8. Ros navigation documentation, http://wiki.ros.org/navigation
9. rtabmap_ros documentation, http://wiki.ros.org/rtabmap_ros
10. Ubuntu, https://www.ubuntu.com/
11. Cardenas, I.S., Vitullo, K.A., Park, M., Kim, J.H., Benitez, M., Chen, C., Ohrn-McDaniels, L.: Telesuit: An immersive user-centric telepresence control suit. In: Proceedings of the 2019 ACM/IEEE International Conference on Human-Robot Interaction. HRI '19, ACM, New York, NY, USA
12. Durrant-Whyte, H.F., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Automat. Mag. **13**(2), 99–110 (2006)
13. Dutta, S., Chaudhuri, B.B.: A color edge detection algorithm in rgb color space. In: 2009 International Conference on Advances in Recent Technologies in Communication and Computing. pp. 337–340 (Oct 2009). https://doi.org/10.1109/ARTCom.2009.72
14. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: Robot Operating System (ROS): The Complete Reference (Volume 1), chap. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-26054-9$_2$3, http://dx.doi.org/10.1007/978-3-319-26054-9_23
15. Horner, J.: Map-merging for multi-robot system (2016), https://is.cuni.cz/webapps/zzp/detail/174125/
16. Kanyok, N., Kim, J.H.: Towards an ethical framework for robot development. SPRINT Industrial and Maintenance Robotics Conference (Nov 2018)

17. Kim, D.Y., Cardenas, I.S., Kim, J.H.: Engage/Disengage: Control Triggers for Immersive Telepresence Robots. In: Proceedings of the 5th International Conference on Human Agent Interaction - HAI '17. pp. 495–499. ACM Press, Bielefeld, Germany (2017). https://doi.org/10.1145/3125739.3132603, `http://dl.acm.org/citation.cfm?doid=3125739.3132603`
18. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehensive simulation of quadrotor uavs using ros and gazebo. In: 3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR). p. to appear (2012)
19. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
20. Shaker, A., Baker, J., Sharma, G., Yuan, M.: Performance comparison of nvidia accelerators with simd, associative, and multi-core processors for air traffic management. pp. 1–10 (08 2018). https://doi.org/10.1145/3229710.3229757
21. Umari, H., Mukhopadhyay, S.: Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1396–1402 (Sept 2017). https://doi.org/10.1109/IROS.2017.8202319
22. Verbiest, K., Berrabah, S.A., Colon, E.: Autonomous frontier based exploration for mobile robots. In: Liu, H., Kubota, N., Zhu, X., Dillmann, R., Zhou, D. (eds.) Intelligent Robotics and Applications. pp. 3–13. Springer International Publishing, Cham (2015)
23. Ying, D., Yan, Y., Dang, J., Soong, F.K.: Voice activity detection based on an unsupervised learning framework. IEEE Transactions on Audio, Speech, and Language Processing **19**(8), 2624–2633 (Nov 2011). https://doi.org/10.1109/TASL.2011.2125953