RoboCup 2022
RoboCup Rescue Simulation League
Virtual Robot Competition

Rules Document

Technical Committee:
Mohammad Hossein Shayesteh, Amirreza Kabiri, Nicky Sadighi

Organization Committee:
Arnoud Visser, Masaru Shimizu, Francesco Amigoni, Josie Huges

Executive Committee:
Fatemeh Pahlevan Aghababa

version draft: Jan. 18, 2022

Abstract

This competition aims to provide a common benchmark to demonstrate scientific progress in the application of robotics to Urban Search and Rescue. The rules of this competition are loosely inspired by the rules of the RoboCup Rescue Robot League and the Agent Competition of the RoboCup Rescue Simulation League. As in the Rescue Robot League, a devastated area has to be explored for victims by a team of robots controlled by an operator. Compared to the Rescue Robot League, the focus is on exploring larger areas with multiple robots rather than the mobility of individual robots. As in the Agent Competition, the disaster situation is unknown before a competition run. The main difference between the two simulation competitions is our focus on realistic sensory and actuation and planning. This year, we had a server migration to ROS2 bringing a great opportunity for roboticists to learn ROS2 in a tested environment via challenges.

1 Introduction

The major technical goal is to encourage intuitive operator interfaces and autonomous and semi-autonomous algorithms that can be used to supervise and control multiple heterogeneous robots operating in challenging environments. Additionally, we aim to have a competition with a low barrier of entry for new teams and allow a variety of approaches. In this sense, the adoption of Gazebo as a simulation framework allows to easily leverage a large amount of code available in the ROS2 ecosystem[1]. Further, it allows teams to test their approach prior to the RoboCup event, which lowers the barrier of entry for new teams.

The challenges result from the environments that robots are deployed in, while the metric combines the number of victims found, the time required, and the area covered by the robots. Additionally, to foster the competition aspect, the scores are largely computable in real-time and can be displayed to an audience during each competition run but will not be announced till all the teams have completed their map runs.

Finally, the scores should reasonably reflect performance for relevant real-world problems modeled in the simulation. Here, the simulation aspect of our league has the advantage of reproducible comparisons since all activity can be logged, and ground truth data is readily available.

[1] See tutorials at https://robocup-rsvrl.github.io/index (It's for ROS2-Foxy and Gazebo 11)

The purpose of this competition is to provide a common benchmark to demonstrate scientific progress in the application of robotics to Urban Search and Rescue. The design and implementation of a RoboCup competition is an ongoing process that is possible thanks to many people worldwide volunteering a significant part of their time to this event. Contributions include improvements to the simulation engine, creating competition worlds, and running the competition itself. The generous contribution of all volunteers in the past is warmly acknowledged. They have enabled sustainable competition without excessive human effort and can remain viable. This is especially true this year, the fourth one after a big change in simulation platform, scenarios, and, consequently, in rules.

As usual, suggestions, constructive feedback, and volunteer work are welcome, appreciated, and needed.

All teams participating in the 2022 competition agree to follow the latest version of these rules.

2 RoboCup Rescue Virtual Robot Competition

Indoor and outdoor search and rescue scenarios may be encountered during the competition. Before the run, the teams will be given basic information about the scenario. This will include the location of the disaster (indoor/outdoor) and possible dangers. Teams will be required to search for victims located in different places in the arena. As there are two types of victims in each map including alive and dead victims, in order to count a victim as a successfully detected victim, at the end of each run, each team should park their robots "near" each alive victim and also provide a map which is generated by their robots explorations and this map should be marked with the location of dead detected victims (near is defined as somewhere in a radius of 1.5 meters around the victim, but Technical Committee has the final word on when a point is enough "close"). Another way to get points from finding victims is to place the victim's information, e.g., a numbered victim mark, on the map. The victim mark must be printed within 1.5 meters of the true coordinates of the victim. In this case, the robot that found the victim can continue to explore. The victim's information should be printed on the map for easy readability, but it is allowed that the victim's information can be submitted on a spreadsheet. Teams are asked to submit maps generated by their robots to the Technical Committee for scoring, comparing performance, and sharing information between teams. In 2019, the multi-floor was introduced, in which floors are connected via ramps. In this case, the team should provide an individual image for each floor, e.g., two images for a two-floor world. Scoring will be made according to the number of correctly detected victims, the time required to detect victims, and the amount of area covered by the robots. It should be noted that fault detections will cost penalty points.

There would be two types of victims: alive or dead victims. Alive victims are characterized by at least one of the following features:

(a) They have a *hot* skin that cannot be distinguished by the operator (human) eye looking at the images returned by cameras onboard robots but can be distinguished using thermal cameras[2].

(b) They move.

In missions with two types of victims, the number of found victims is calculated according to the number of successfully detected victims. Alive victims are successfully detected if they are discovered (seen) and a robot is parked close to each of them. Dead victims are successfully detected if they are discovered (seen), and their locations should also be marked near the victim on the map. It is also required that, when a victim is found, its status (alive or dead) is declared. If each of the above detections –the victim detection and the victim's status detection- are made autonomously, more points will be considered. The details of the scoring formula can be found in Section 2.3.

Only robots validated before the competition will be allowed to be used. The list of accepted robots and sensors is reported in this document (Section 6). Note that not every combination is possible. The sensor load will be examined, and the Technical Committee reserves the right to disallow any unrealistic combination of robots and sensors.

Several mobile robots will form robot teams, plus a base station (a robot that does not move) that provides an interface to the operator.

[2] https://github.com/m-shimizu/Samples_Gazebo_ROS/

2.1 Simulation Environment

In 2022, after two years of virtual collaborations, we hope we can have the competition in person with the rise of vaccinations. However, we will monitor the situation closely, and if needed, we can have a hybrid of remote and in-person competition. Simulations will run in a ROSject. The ROSject is a cloud virtual computing platform provided by Construct[3]. The ROSject can run Ubuntu 20.04, ROS-melodic, and Gazebo 11 like a PC. The Technical Committee creates a ROSject that has everything needed for competition[4]. The teams can copy it and use it instantly[5], so they can start developing their robot client software[6] after the Technical Committee releases the ROSject for the competition.

During the remote competition, the Technical committee uses online conferencing systems such as ZOOM and MEET to control the competition's progress remotely. The Technical Committee runs one ROSject as a game server PC. On the other hand, the team runs its ROSject as a game client PC. The server PC's IP address will be announced to the team at the start of the setup time. (This server-client type ROSjects run may change into a single ROSject.) The Technical committee can run games parallelly up to three teams.

Each team has 20 minutes to set up with its assigned server. The run starts at the scheduled time. If a team is not ready, time will start anyway. Each competition run lasts about 20 minutes (exact time to be announced before the run). Each team will run its client code on its own ROSject.

Connection to the server ROSject happens using direct ROS control, and each team sends commands according to the general ROS-Gazebo interface[7]. The server ROSject runs Gazebo 11 and robot_state_publisher ROS node. The team knows the IP address of the server. To ensure the fairness of the competition and the fact that teams are not cheating in communicating with the server, the Technical Committee reserves the right to inspect the team's code at any time to ensure that the client controlling robots exchange with the simulator only data compatible with realistic situations. Reading data from sensors mounted on robots is realistic, sending commands to robot actuators is realistic. The Technical Committee reserves the right to decide about interpreting the term “realistic”.

The robots will be instantiated in the world at the prescribed start time. Starting poses of the robots forming a team will be provided as follows. Before the run, the Technical Committee will execute a launch file built from a template provided by the team (including the robot configurations the team would like to use) and modified with the starting poses of the robots (decided by the Technical Committee). Alternatively, the Technical Committee can provide the list of starting poses of the robots by other means (e.g., on paper), but the information should be used only for robots spawning.

All robots must be spawned at the start of a run, though teams can decide to activate them at their convenience.

2.2 Challenges

The rescue virtual robots league's main challenge is finding victims in the shortest possible time. All robots must search the unknown environment in a group of robots and mark the important information of each victim on a 2D-Map. In the following, the main challenges are discussed.

2.2.1 Victim Search Challenge

The victim search mission is a standard rescue virtual robot competition where all teams get points based on the performance metrics according to section 2.3. This challenge is to assess teams’ innovation and approaches in implementing Victim detection algorithms.

---

[3] https://www.theconstructsim.com/
[4] https://robocup-rsvrl.github.io/ and https://github.com/RoboCup-RSVRL/RoboCup2022RVRL_Demo
[5] https://www.youtube.com/watch?v=79TRqALia2s&feature=youtu.be
[6] https://www.theconstructsim.com/how-to-test-your-ros-programs/
[7] See https://robocup-rsvrl.github.io/installation/getting-started/https://github.com/nkoenig/pioneer3at_demo for spawning multiple robots with different namespaces.

2.2.2 Mapping Challenge

Teams are asked to submit the merged map of the world at the end of each run. They need to prepare a map merging algorithm that supports multi-level floors and handles distortion. The mapping challenge is to compete for the remarkable information on the submitted map during the victim search mission. The teams place the great information for the rescue mission on the highly accurate map in an easy-to-read manner: victim status information (alive, dead, partial burial, entombed) and danger information (high temperature, poor visibility and image).

2.2.3 Exploration Challenge

The exploration challenge is to assess teams' innovation and approaches in map navigation and exploration as well as multi-robot communication. For instance, how each robot decides what would be it's next target point for exploration purposes and how robots communicate with each other.

2.3 Performance Metrics

For scoring purposes, a team member is counted as a human operator who is supposed to do the following:

- Starts a robot, enters initial points,
- Actively drives a robot around,
- Stops a robot before the run is over (for example, to prevent it from bumping into victims),
- It is involved in any way in the victim recognition process.
- Hands the final map to the technical committee

Each team can have only one human operator for each run.

Let v be the number of victims a team detected successfully (see Section 3 for the definition of successfully detected victims) and t be the completion time for the team, i.e., the time until either all V victims in the arena have been found or the maximum mission time T is exceeded. Note that, the number v_r of successfully detected victims is the sum of:

- the number of successfully detected dead victims (with a marked point on the map close to them) and of
- the number of successfully detected alive victims (which must have a robot parked close to each of them) which is represented by v_{ar}

The score of a team is calculated using the following formula:

$$Score = [\beta \times S + (1 - \beta) \times (\frac{v_r - 0.5 v_w}{V} + max(0, \frac{2 v_{ar} - v_{aw}}{2 V_a}))]\alpha$$

Where the indexes r, and w respectively represent successfully detection of the victim, fault detection of the victim, and the index a represents the alive victim, while α $(0.1 < \alpha < 2)$ is proportional to the difficulty level of the map, defined by the map designer, β is a constant $(0.1 < \beta < 1)$ that balances the weight of finding victims and of exploring the environment, and S is the ratio of the area explored by robots, which is measured as follows. Each map has a number N of "invisible portals"[8] that represent important points of the environment and that are defined by the map designer (for example, invisible portals are intersections of corridors, doorways, and other relevant points according to the environment[9]), but unknown to the teams. Explored area ratio S is calculated as *n*/N, where *n* is the number of invisible portals discovered by the robot out of N total invisible portals, representing the portion of the environment explored by robots of a team.

---

[8] Invisible portals mimic the idea of "map fiducials" used in the RoboCup Rescue Robot League (see rules for the 2015 competition at: http://wiki.ssrrsummerschool.org/doku.php?id=rrl-rules-2015.

[9] In future competitions, invisible portals could be automatically generated from a map. For example, considering an indoor map, invisible portals can be generated according to a Voronoi decomposition of the map.

To encourage safe robot behavior, any contact between a robot and an (alive) victim causes the team a 20% score penalty for that victim (instead of 10 points, the team gets 8 points after the first contact, 6 points after the second contact, ..., down to 0).

Besides, before each run, the Technical Committee can communicate to the teams that operators are not allowed to control the robots manually (*handoff period*) for *k* minutes, during which robots are supposed to explore automatically.

3 Technical Challenge Competition

Communication between mobile robots and between them and the base station is based on realistic models. In order to facilitate broader participation during the transition from USARSim to Gazebo, for the 2022 competition, the communication between robots will be assumed to be unconstrained (both in terms of connections and of bandwidth). However, teams will be required to run some experimental sessions with constrained communication, using the Wireless Communication Simulator (WCS) available at:

https://github.com/taherahmadi/WCS

In the competition at RoboCup 2022, the Technical Committee will not enforce any segregation of the code of different robots of the team. However, for the future, the Technical Committee is exploring these two possibilities:

(1) One ROS master running with Gazebo, separating robots, and delivering messages according to their namespaces (for example, with the ros_con gateway).

(2) One ROS master running with Gazebo and separate ROS masters for each robot (for example, using multimaster_fk).

This applies to physical robot-environment communications, which should be kept local. All other communications (operator-robot and robot-robot) will use the WCS mentioned above, which substitutes the Wireless Communication Server WSS and runs on the server machine, simulating constrained and intermittent wireless network links in a disaster setting, where multi-hop routing and autonomous behavior of robots are required. All communications between robots and a base station must go through WCS. As long as the WCS is not used, the teams may assume unlimited communication but should be aware that this situation will change next year. With WCS, the operator at the base station can send commands to a robot and will obtain measurements and video images from a robot only when that robot is in radio contact. The location of the base station and the wireless cutoff strength will be provided as a priori data and announced before each run. During the competition, all communications with the robots can be logged. Therefore, the Technical Committee will check for communications that bypass the WCS. Teams that violate this policy are immediately disqualified, and the reason for the disqualification will be posted on the web.

4 Open Source Policy

The winning teams must provide a fully functioning copy of their software to the organizers before the final ceremony. Failure to do so will result in team disqualification. All other teams are also requested to provide their code, though not before the awards event. The software will be posted on (or linked from) the competition web pages and wikis, giving proper credit to the authors.

All data logs collected during the competition can be made available on the web for public use, including, but not limited to, scholarly work devoted to performance evaluation and benchmarking

.

5 Summary

The competition intends to stimulate research in robotics that allows for autonomous and safe exploration of significant parts of the environment providing aid to first responders to rescue victims

6 Allowed Robots and Sensors

It is suggested to the teams to use combinations of the following robots (whose models are available in Gazebo) and the following sensors:
P3AT (odometry, camera, battery, sonar, gps, laser range finder, thermal camera)
Turtlebot (odometry, camera, battery, sonar, gps, laser range finder)
Quadrotor (camera, battery, gps, laser range finder)
Crawler_robot (camera, laser range finder)
Centaur_robot (camera, laser range finder)

Note: *In addition, the teams could send their robot models to the Technical Committee in order to be evaluated. In the case of the qualified models, it would be announced to the other teams, thus they can use the model if they want.*

Sensor data can have added noise.

Sensor load will be examined. The Technical Committee reserves the right to disallow any unrealistic combination of robots and sensors. Before the competition, the technical committee can publish many reasonable configurations used during the competition. Additionally, if, during the competition, unrealistic behavior is detected for a robot or a sensor, this device can be excluded for further usage during the rest of the competition.