

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,600

Open access books available

137,000

International authors and editors

170M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

From ERL to MBZIRC: Development of An Aerial-Ground Robotic Team for Search and Rescue

*Barbara Arbanas, Frano Petric, Ana Batinović, Marsela Polić,
Ivo Vatavuk, Lovro Marković, Marko Car, Ivan Hrabar,
Antun Ivanović and Stjepan Bogdan*

Abstract

This chapter describes the efforts of the LARICS team in the 2019 European Robotics League (ERL) Emergency Robots and the 2020 Mohamed Bin Zayed International Robotics Challenge (MBZIRC) robotics competitions. We focus on the implementation of hardware and software modules that enable the deployment of aerial-ground robotic teams in unstructured environments for joint missions. In addition to the overall system specification, we outline the main algorithms for operation in such conditions: autonomous exploration of unknown environments and detection of objects of interest. Analysis of the results shows the success of the developed system in the competition arena of two of the largest outdoor robotics challenges. Throughout the chapter, we highlight the evolution of the robotic system based on the experience gained in the ERL competition. We conclude the chapter with key findings and additional improvement ideas to advance the state of the art in search and rescue applications of heterogeneous robotic teams.

Keywords: search and rescue robotics, aerial robotics, autonomous exploration, MBZIRC, ERL

1. Introduction

Search and rescue (SAR) problems for collaborative multi-robot systems (MRS) have been an interesting research topic for several decades [1–3]. The attractiveness of the domain stems from the variety of problems it incorporates, including mapping and situational awareness, monitoring and surveillance, establishing communication networks, or cooperative decision making. All these aspects make SAR a very difficult problem to solve. Despite the considerable difficulties, conducting SAR operations with autonomous MRS offers many advantages. A capable robotic search and rescue team can replace humans as first responders in disaster areas, map and inspect the site, and make it safe for the approach of human rescue teams.

To facilitate and accelerate progress in the field of search and rescue, many robotics competitions have been held over the years. In recent years, many of them

have begun to incorporate multi-domain challenges involving robots from different domains, including aerial, ground, and underwater robots. At the forefront of this concept is the euRathlon [4], the world's first multi-domain (air, land and sea) multi-robot SAR competition. In a recreated disaster scenario inspired by the 2011 Fukushima nuclear power plant accident, the euRathlon 2015 Grand Challenge required teams of robots to work together to map the area, find missing workers and contain a leak. As a continuation of this initiative, a series of the European Robotics League Emergency Robots [5] (ERL-ER) was organized as part of the European Robotics League. Some changes were made to increase the level and difficulty of the challenges, including more complex manipulation tasks and even more collaboration between robots from different domains. Three ERL-ER competitions were held: Piombino 2017, Seville 2019 and La Spezia 2019.

In parallel, there is another branch of multi-domain competitions that gathers many teams worldwide, the Mohamed Bin Zayed International Robotics Challenge [6] (MBZIRC). MBZIRC focuses not only on SAR missions, but also on promoting the state of the art in robotics in general. The missions are very challenging from both control and algorithmic perspective. They often involve complex manipulation tasks, the need for rapid response and tracking of fast-moving objects, and SAR segments such as fire detection and extinguishing. This competition took place in 2017 and 2020 in Abu Dhabi.

In this chapter, we focus on our efforts in developing an autonomous aerial-ground search and rescue team during the last few years. Our system was formed in two robotics competitions, ERL-ER 2019 Local Tournament in Seville and MBZIRC 2020. In the first competition, we succeeded in setting up the autonomous system for search and rescue missions, including mapping the area and detecting objects of interest. The MBZIRC challenge involved a more complex scenario with three different missions, and we adapted and improved the developed system based on the results of the ERL competition. In this chapter, we detail the hardware and software design and the methodology used to solve the problems in the challenges. At the end of the chapter, we offer important conclusions and practical advice from the years of development of our robotic team to further advance the practical application of autonomous MRS to these types of problems.

The chapter is organized as follows. The next section surveys the state of the art in SAR problems for multi-robot collaborative systems. Section 3 describes the mechanical design of the robotic aerial and ground platforms. Section 4 details the software design for both platforms, including the modules used and the organization of the overall software architecture. Section 5 provides insight into the development of autonomous exploration and mapping of unknown areas using an Unmanned Aerial Vehicle (UAV) or Unmanned Ground Vehicle (UGV) with a LiDAR. In Section 6 we give an insight into the object detection methods we have developed for the ERL and MBZIRC competitions. In Section 7 we analyze the performance of the described system and its components during the competition runs, followed by conclusions and lessons learned as the last section.

2. Related work

Robotics research in the field of SAR has, to a large degree, been driven by competitions and challenges, first of which were introduced in late 90s [7], followed by later establishment of euRathlon [4], DARPA challenge [8], ERL Emergency [5] and MBZIRC [6].

Through the fundamental research in the field of robotics that is required to meet the demanding scenarios, the competitions and challenges are also driving the

development of standardized benchmarks for SAR robots, with the US National Institute of Standards and Technology (NIST) leading the way [9]. NIST recognizes the urban search and rescue scenarios as a great tool to measure how intelligent autonomous robots are by looking at several components of an autonomous search and rescue robot or robot team.

The first component of a SAR system is the ability to traverse difficult terrain, which is usually solved by using a heterogeneous robotics team consisting of several robots with different capabilities. For the land-based SAR, there are various formations that are used, such as physically interacting UGV-UGV teams [10] where one robot can carry the other one, physically interacting UAV-UGV teams [11] where the UAV can carry the UGV over obstacles and the UGV can carry the UAV to save energy, or teams where only information is shared [12]. In our work for ERL and MBZIRC, we focus on the information sharing team consisting of a UGV and at least one UAV.

To traverse the terrain and find objects or people, the SAR robotic team needs to be able to perceive the surroundings. While some applications may require acoustic and thermal sensing, our robotic team is equipped with 2D and 3D LiDARs for mapping and navigation and cameras (both RGB and RGBD) for detecting and locating people or objects. For some detection tasks, we used deep neural networks [13], while some objects could be recognized solely by color, similar to [14]. 2D and 3D maps of the environment were built using Google Cartographer [15], which was adapted to use information from the GPS and to also serve as a pose sensor for UAV control [16]. The navigation stack for the UGV is based on the TEB planner [17], while the UAVs rely on the TOPP-RA approach [18].

Generating waypoints for navigation planners is an essential aspect of robot team autonomy. In our work, several methods were implemented: (a) waypoint generation using a 2D lawnmower pattern in relatively small areas of known size; (b) Levvy flight 2D waypoint generation for large areas of known size [19]; (c) autonomous 2D [20] and 3D exploration [21] for areas of unknown size. In this work, we briefly discuss our autonomous exploration approach for ERL and MBZIRC competitions, which was later extended to the planner described in [21], based on the exploration tool called 3D-FBET, work of Zhu et al. [22].

The final component of the autonomous SAR system is collaboration between multiple agents, including humans, which is constantly a topic of great interest in SAR challenges and projects [23–25]. While we have developed a framework for collaboration and coordination in heterogeneous robot teams [11, 26], due to nature of challenges in ERL and MBZIRC, the developed approaches were not deployed in the field.

3. Mechanical design of the robotic platforms

In this section, we describe the hardware setup of the robotic platforms used for the search and rescue missions in the ERL and MBZIRC competitions. We outline the mechanical structure of the robots and the components installed on board the vehicles to enable the execution of the specified tasks.

3.1 Hardware setup of the UAV

For the hardware design of the UAV, we followed the requirements of the challenge specifications. The main purpose of the aerial platform is to autonomously map and explore the disaster area and search for objects of interest. Therefore, we developed a custom hardware configuration to support this purpose, including the

use of appropriate sensors for localization and vision-based algorithms. The mechanical design of the UAV was the same for the ERL and MBZIRC search and rescue missions.

The frame of the aerial platform (**Figure 1**) consists of four arms, a body, and two legs with skis, all made of carbon. The vehicle is equipped with Flame 60A 12S ESCs which are driving T-motor P60 KV170 12S motors with 22-inch carbon propellers. The autopilot used is a Pixhawk 2.0, and the maximum takeoff weight of the vehicle is 12 kg with a 2 kg payload. The UAV is equipped with an Intel NUC i7/16GB computer running Ubuntu 18.04 LTS and ROS Melodic, which processes input from ZED Mini stereo camera, Smartek Vision camera and Velodyne VLP-16 Puck Lite 3D LiDAR. Image processing for the UAV is performed using convolutional neural networks running on Intel Neural Compute Stick 2. All components of the aerial platform are powered by two LiPo 12S 14000mAh batteries, which give the vehicle a flight time of up to 30 minutes. The summary of components on board of UAV is given in **Table 1**.

Ardupilot's Copter firmware is a full-featured, open-source multicopter UAV controller. Copter is capable of handling the full range of flight requirements, from fast FPV racing to smooth aerial photography to fully autonomous complex missions that can be programmed via a range of compatible software ground stations. The entire package is designed to be safe, feature rich and open to custom applications. Copter firmware is loaded onto the Cube Black autopilot, which includes a 32-bit ARM Cortex M4 core processor with a fail-safe 32-bit coprocessor, 256 KB of RAM and 2 MB of flash memory. Integrated into the Cube are a triple redundant

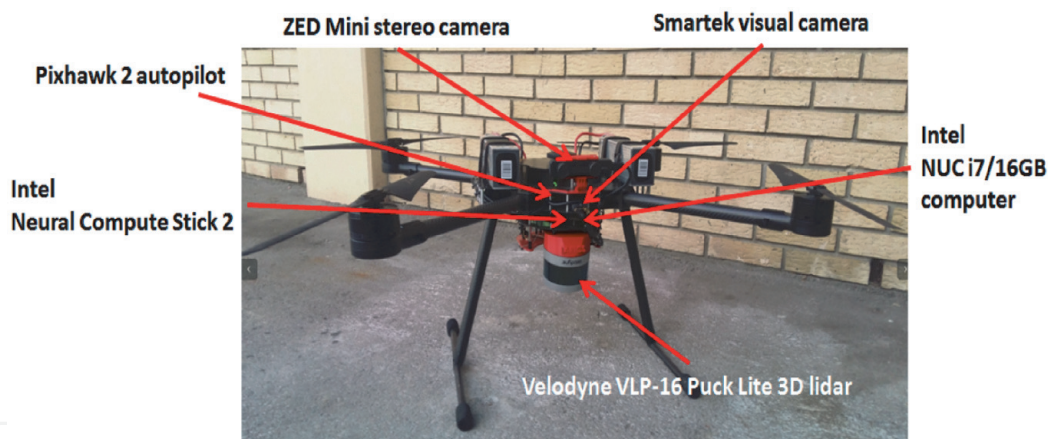


Figure 1. Front view of the aerial platform with all sensors and electronic devices on board.

Component	Specifications	Purpose
On-board computer	Intel NUC i7-8650U Quad Core 1.9GHz	Main computing unit
Autopilot	Pixhawk 2.0	Flight control, sensors (IMU, barometers)
Computing unit	Intel Neural Compute Stick 2	Neural network processing
GPS sensor	Ublox M8P GNSS receiver	Localization, navigation
LiDAR	Velodyne VLP-16 Puck Lite	Localization, mapping
Stereo camera	ZED Mini	Perception, object detection

Table 1. UAV hardware component summary. The same setup was used in ERL and MBZIRC competitions.

inertial measurement unit (IMU) containing accelerometer, gyroscope and magnetometer as well as two barometers, which are isolated and damped.

The carrier board provides an interface for various Pixhawk-compatible peripherals, as well as power for all components of the drone. On the top of the UAV is an external Global Positioning System (GPS) sensor, whose receiver is based on the high-precision Ublox M8P GNSS module. When paired with the external GPS and antenna via a telemetry module, the vehicle's GPS (Pixhawk) can achieve centimeter-level accuracy. The copter firmware uses an Extended Kalman Filter (EKF) algorithm to estimate the vehicle's position, velocity, and angular orientation based on gyroscope, accelerometer, compass, GPS, airspeed, and barometric pressure measurements.

The Velodyne VLP-16 Puck Lite features a rotating array of sixteen infrared laser emitter-detector pairs that rotate at 300 to 1200 RPM. Each of the 903 nm lasers is fired 18080 times per second, and distances are measured based on the time of flight for each pulse. The captured distance measurements can be used to create high-quality point clouds. Each LiDAR sweep covers a vertical angle of 30 degrees with a resolution of 2 degrees and any horizontal angle with a resolution of 0.1 degrees to 0.4 degrees, depending on the RPM setting.

The ZED Mini is a stereo camera that provides high-resolution images and accurate ambient depth measurements. It is designed for the most demanding applications, such as autonomous vehicle control, mobile mapping, aerial mapping, security, and surveillance. It is actually a device with two cameras 65 mm (2.56 inches) apart, designed to mimic the human eye.

The on-board computer used in the UAV is an 8th generation Intel NUC with a powerful Intel i7-8650U Quad Core 1.9GHz processor. The computer has an integrated Intel Iris Plus Graphics 640, a 16GB Kingston SODIMM RAM module and a 250GB Kingston A2000 M.2 SSD as storage unit. The required power supply range is 12-19 V. The operating system installed on the on-board computer is Linux Ubuntu 18.04 LTS with the middleware ROS Melodic. Since the on-board computer needs to be embedded in the body of the drone and to save some weight, the outer case is removed and the computer is mounted on the bottom of the UAV.

3.2 Hardware setup of the UGV

In ground search and rescue missions, the choice of the ground mobile platform is mostly dictated by the mission and terrain requirements. In our robotic team, a Husky A200 was chosen as a platform suitable for motion in rough terrain. The competition tasks directly dictated the choice of additional hardware components. The complexity of the MBZIRC tasks was significantly higher compared to the ERL competition, and the hardware design was therefore more elaborate, as explained in this section.

3.2.1 ERL search and rescue platform

The main task of the UGV in the ERL competition was to autonomously locate objects of interest. Our search and rescue platform is shown in **Figure 2**, with the autonomous navigation sensor setup: a Sick NAV350 LiDAR, an IMU, a GPS receiver, and a real-time kinematic positioning (RTK) unit for higher precision. For reliable detection of obstacles during movement, LiDAR was mounted on top of the vehicle body, near the front end of the vehicle, so that there is no shadowing from other UGV parts. GPS and RTK units were elevated with respect to the vehicle body to avoid potential signal blockage and interference. The task detection requirements allowed the use of the Intel RealSense D-435, a commercially available RGB-D

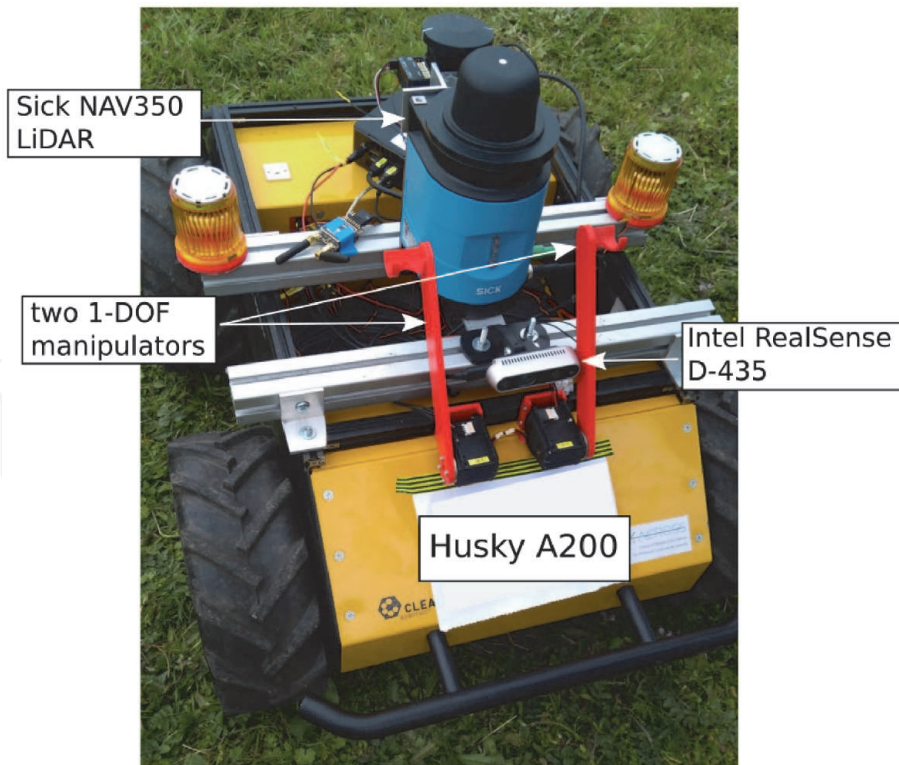


Figure 2.
Hardware setup of the UGV and its sensors and actuators used in ERL competition.

camera, instead of more precise and expensive alternatives. Thanks to recent technological advances, the camera is small enough to be mounted in the front part of the vehicle without interfering with other sensing components, such as LiDAR beams.

Another task of the UGV was to provide first aid kits. For this purpose, a simple 1-DOF manipulator was constructed, sufficient for safe transport and easy unloading of the package. As the ERL task required delivery of two kits, two such manipulators were mounted at the front of the vehicle, outside of the camera field of view. The manipulator links were designed and 3D printed from PLA plastic (shown in red in **Figure 2**) and driven by Dynamixel servo motors.

The UGV design described was quite simple, quick to implement, and task-specific, resulting in successful mission execution. However, in the long run, it became apparent that a more complex solution would be much more beneficial for mission versatility.

3.2.2 MBZIRC mobile manipulation platform

In the MBZIRC competition, we chose to design a mobile manipulator platform, and to rely on the dexterity of the arm for execution of more complex tasks, such as repetitive pick and place in the wall construction scenario. Still, each new task requires some adaptation of the manipulator tool. We equipped the vehicle with a lightweight 6-DOF manipulator, a Schunk Powerball LWA-4P. The arm has integrated joint drives, eliminating the need for additional external control units or power converters. The setup of our platform is shown in **Figure 3**.

One of MBZIRC's challenges was a wall construction task in which the mobile manipulator had to pick up bricks from their stacks and transport them to the wall pattern, where they had to be arranged in a specified order. Again, a custom gripper was developed to magnetically grasp the ferromagnetic plates on the bricks, as shown in **Figure 4**. The gripper is lightweight, energy efficient and strong enough

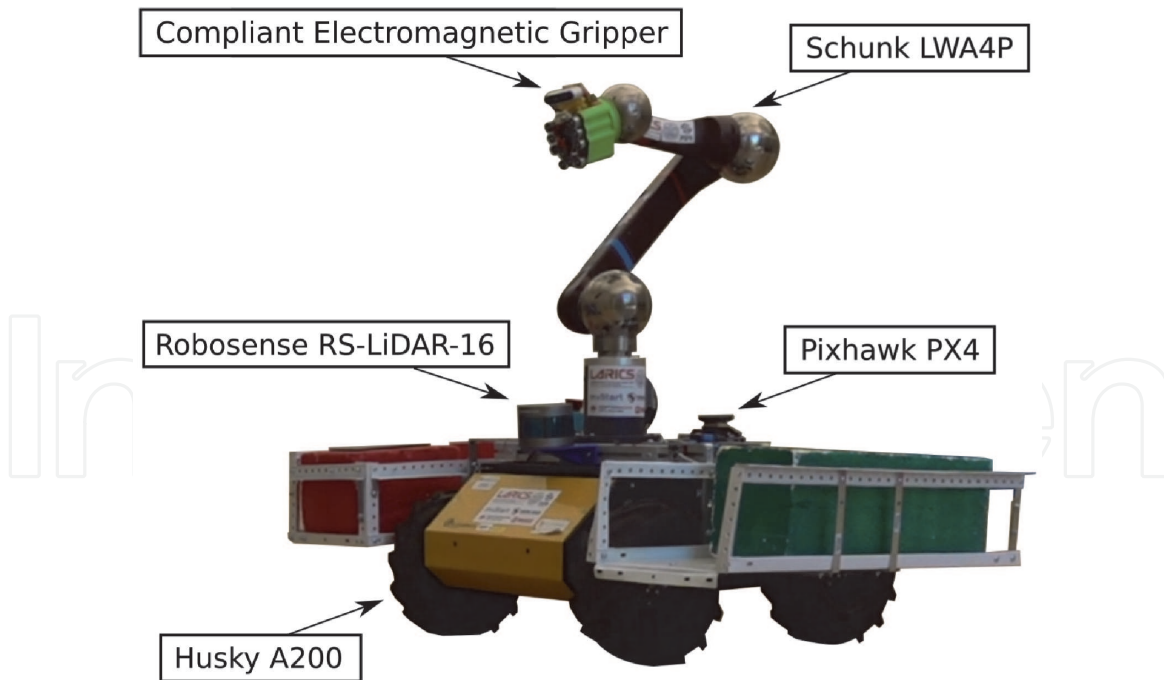


Figure 3.
Hardware setup of the UGV and its sensors, actuators, and electronic components used in MBZIRC competition.

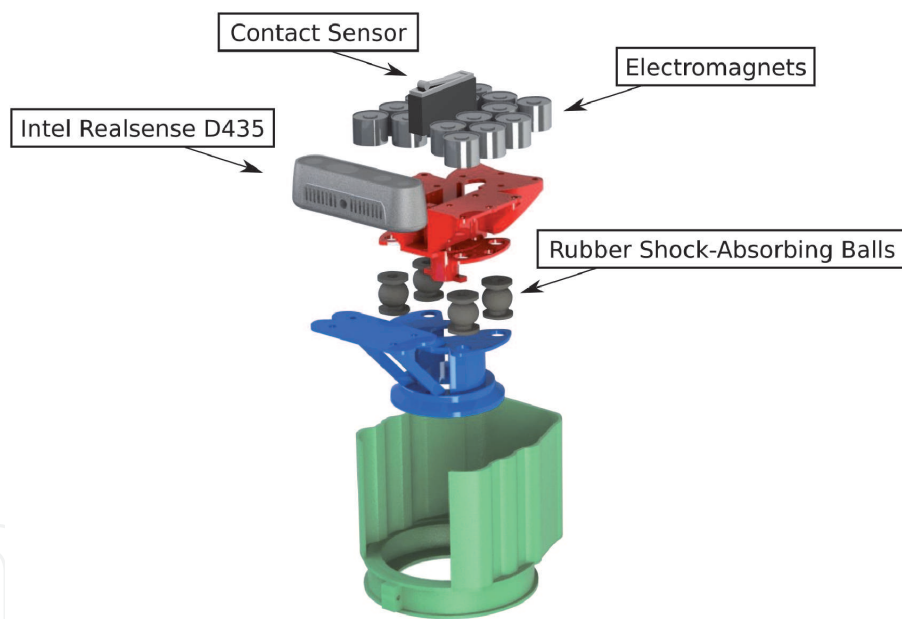


Figure 4.
Design of the passively compliant magnetic gripper for UGV brick manipulation tasks in MBZIRC challenge.

to lift up to 2 kg. The design is based on ten small electromagnets, and at least six magnets in contact are required to achieve sufficient force to lift the brick. Due to the rigidity of the mobile manipulator, any irregularity or misalignment on the contact surface could reduce the number of magnets in contact. This was solved with passive compliance, with four shock-absorbing rubber balls providing compensation for misalignment greater than 10° .

In this scenario, the RGB-D camera was used for detection, navigation, and manipulation. It was therefore mounted as an eye-in-hand on the gripper. For navigation, a 3D LiDAR (Robosense RS-LiDAR-16) was used instead of the 2D solution in ERL. As it turned out, mapping with 2D scans in realistic environments can miss some obstacles, such as those on the ground, below the beam. Other

Component	Specifications	Purpose
On-board	Intel NUC i7-8650U	Main computing unit
Computer	Quad Core 1.9GHz	
Autopilot	Pixhawk 2.0	Sensors (IMU, RC control)
Manipulator	ERL - two 1-DoF manipulators MBZIRC - SchunkPowerball LWA-4P	Object manipulation
GPS sensor	Ublox M8P GNSS receiver	Localization, navigation
LiDAR	ERL - Sick NAV350 LiDAR MBZIRC - Robosense RS-LiDAR-16	Localization, mapping
RGB-D camera	Intel RealSense D-435	Perception, object detection

Table 2.

UGV hardware component summary. The setups used in ERL and MBZIRC competitions.

hardware modifications included an additional battery pack and cargo baskets for transporting multiple bricks. The summary of the components on board the UGV, during the ERL and MBZIRC competitions, is shown in **Table 2**.

4. Software architecture and modules

Having defined the mechanical structure of the robotic platforms, in this section we address the design of the software modules implemented for each robot. Here we define the software that enables basic operations of the vehicles, such as control, localization, navigation, and trajectory planning, while challenge-specific tasks and more complex algorithms are described in the next section.

4.1 Software setup of the UAV

UAV platforms used for search and rescue missions are equipped with Ardupilot firmware. The firmware communicates with Micro Aerial Vehicle (MAV) messages using the MAVLink protocol. The Robot Operating System (ROS) is used to communicate with the on-board computer and the drone. The components used in the ROS environment are as follows:

- **MavRos**¹ is the most important component of the UAV software setup as it provides two-way communication between the vehicle firmware and the on-board computer. Its main task is to translate MAV to ROS messages and vice versa.
- **On-board UAV Controller** receives trajectory points and the current UAV state as input and produces a control output for the vehicle consisting of the desired rotation and thrust. The underlying implementation is a cascade of Proportional-Integral-Derivative (PID) controllers. The outer loop computes the velocity command based on the local position measurements, while the

¹ MAVLink extensible communication node for ROS with proxy for Ground Control Station. <http://wiki.ros.org/mavros>

inner loop calculates the desired orientation and thrust based on the velocity measurements.

- **UAV State Machine** serves as a position command multiplexer. It either computes commands based on the Radio Controlled (RC) controller input or forwards an external command. It also implements simple takeoff and landing procedures. The states are as follows:
 - HOLD GROUND: the UAV is on the ground and ready to arm and takeoff.
 - HOLD AIR: the UAV is in the air and ready to receive external commands.
 - JOY AIR: the UAV is in the air and is controlled by RC.
- **Trajectory Planner** receives a set of waypoints and outputs trajectory points that are sent to the on-board controller. The underlying implementation is a Time-Optimal Path Parametrization by Reachability Analysis (TOPP-RA) [18] which provides an optimal trajectory given the specified kinematic constraints.
- **Cartographer** is a method for Simultaneous Localization and Mapping (SLAM) used to determine the UAV pose in the map frame. Combined with a constant velocity model, the linear Kalman filter yields estimated UAV odometry that is used as feedback for the on-board UAV controller.

In **Figure 5** we have illustrated the interaction of the above components within the UAV software architecture on a NUC computer. This figure refers to the case where the UAV state machine is in the HOLD AIR state. The pipeline begins with inputs from the LiDAR and IMU that are fed into the Cartographer SLAM

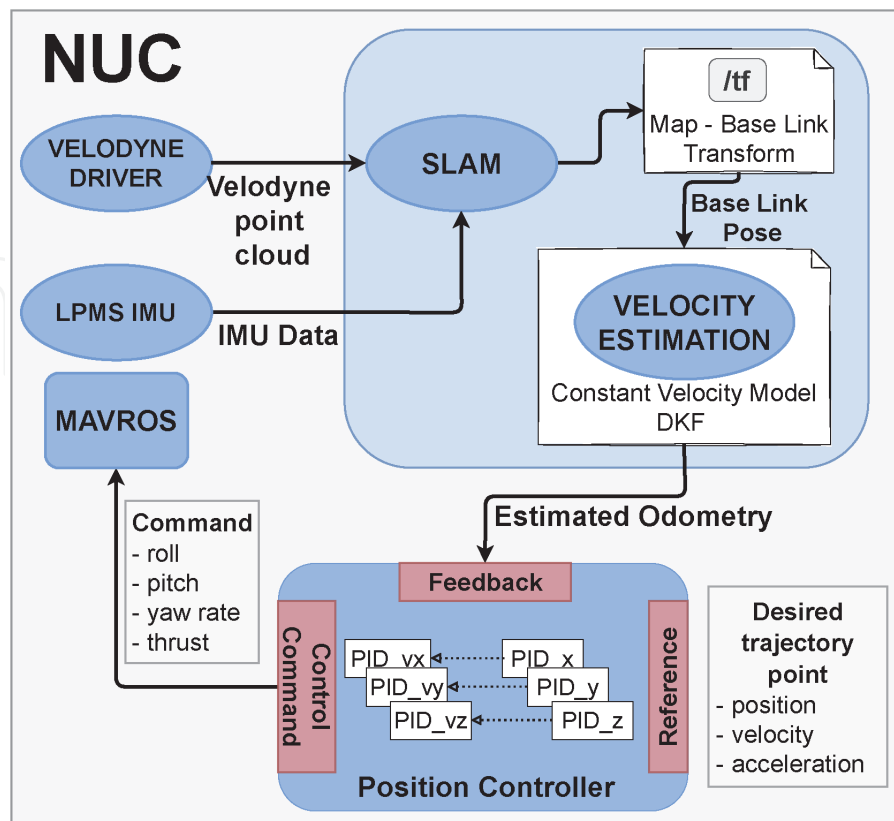


Figure 5. Layout and the interaction of the UAV software components running on the on-board NUC computer.

algorithm, which creates the map of the environment and locates the vehicle in the map frame. Next, using the Constant Velocity Model, the estimated odometry of the vehicle is calculated and fed to the UAV position controller. Based on the reference generated with the trajectory planner, the controller generates control outputs of the desired vehicle orientation and thrust and sends them to the MavRos. These messages are then forwarded to the lower level UAV controllers implemented in the firmware. If georeferencing is needed, GPS measurements can also be forwarded to the Cartographer.

4.2 Software setup of the UGV

The main software modules used for UGV control in the competition search and rescue missions are those for localisation, motion and manipulation:

- **Cartographer SLAM** is the module used for simultaneous mapping and localization of the vehicle.
- **Move Base** is a ROS package used for ground vehicle navigation.
- **Manipulation module** for controlling two simple 1-DoF manipulators used to drop aid packages.²

The Google Cartographer package was used to create a map of the unknown environment and localize the mobile base within it. During the ERL-ER competition, a two-dimensional version of the Cartographer's SLAM algorithm was used, utilizing 2D laser scan data from the Sick NAV350 LiDAR. This approach results in only detecting obstacles that are aligned with the LiDAR, and thus led to problems with lower obstacles, more specifically the lower part of the obstacles as shown in **Figure 14**. The approach we took to overcome this issue was solved using a large safety margin around all the detected obstacles. Later, during our work for the MBZIRC competition, we adopted a different approach using Cartographer's 3D SLAM algorithm, and a 3D LiDAR.

Ground vehicle motion planning was done with the Move Base package. Since this package uses a 2D cost map, the 2D Cartographer map could be used for safe motion planning with obstacle avoidance. However, with the Cartographer's 3D SLAM, we filtered the 3D submaps based on their height, to obtain a 2D cost map. Points below the lower threshold are filtered out to avoid classifying the ground as an obstacle. Similarly, points that are above the maximum height of the robot are removed as they do not represent a real obstacle for platform navigation. However, ground points that are far away from the robot may be perceived as false-positive obstacles if the ground is not completely flat. This issue was resolved by creating newer submaps, generated using the 30 most recent laser scans.

The navigation during ERL was based on manually defined waypoints, selected in the map by the user. Move Base package would then autonomously perform the path planning with obstacle avoidance. This semi-autonomous planning system was improved for the MBZIRC competition, by implementing a higher-level planning algorithm that generates desired waypoints autonomously.

Thanks to the Husky A200 ability to rotate in place, the most commonly used navigation planner the DWAP, which heavily relies on this capability. However, such motion causes significant vibrations, and the with the addition of the robotic

² Used only in ERL competition.

arm for the MBZIRC competition, this problem is further exacerbated due to the increase in the overall mass and heightening of the center of mass position. Therefore, in the MBZIRC competition we switched to the Timed-Elastic Band (TEB) planner [17] for car-like robots, which uses online trajectory optimization to create a plan constrained by the minimum turning radius.

5. Evolution of the autonomous exploration system

Exploration and mapping of unknown environments is a fundamental task in robotics. It can be used in many different applications such as inspection, surveillance, 3D reconstruction, and search and rescue. Searching for an object of interest in an unknown environment can be formulated as an iterative process consisting of a map update, selection of a next goal, and navigation to that target. The process is complete when the object of interest is found.

Analogously, the search strategy in the exploration of an unknown environment is similar, but with different objectives. In exploration, the main goal is to create a map of the environment, while in search, the focus is on finding the object of interest. Although search is a central task in many search and rescue scenarios, we use an exploration strategy adapted for the search application.

Typical exploration methods are based on frontiers [27] and are used in both 2D and 3D space. In contrast to 2D exploration and mapping strategies, mapping large environments in 3D requires a considerable amount of memory and computational effort.

To create a map of the environment and locate a robot in it, we use a submap-based graph SLAM method Google Cartographer [15]. The map consists of submaps created from a sequence of sensor scans through scan matching, fusion with IMU and odometry. The map is then used for exploration and robot navigation.

5.1 From manual to autonomous exploration

In the ERL competition, some of the objectives were to locate missing workers, entrances to the building, and damage to structures while creating a map of the environment. The proposed layout for the ERL competition is an area of approximately 200 m by 30 m with obstacles. Since the ERL arena may contain GNSS - degraded areas, we used a SLAM algorithm for map generation and robot localisation. We adopt Cartographer SLAM to generate a 3D map onboard the robot in real time. Objects of interest are detected during semi-autonomous navigation, where an operator manually selects desired waypoints for the robot to reach autonomously. Both 2D and 3D maps of the environment are generated, and the positions of objects of interest are marked on the map.

In the MBZIRC challenge, we consider autonomous exploration and search of a UAV in an unstructured outdoor environment. In this challenge, we used only one UAV because it can fly over the area quickly and has the advantage of bird's eye view. The goal is to explore and map the environment while searching for the object of interest (a fire). Initially, the area is unknown, while it is assumed that the boundaries of the area are known. Since the area for exploration and mapping is limited, the exploration is finished when the entire area is covered or when the objects of interest are found.

In order to explore an area, create a map and find the objects of interest, the search strategy is formulated. The search strategy algorithm generates waypoints for trajectory planning considering the boundaries of the environment. In addition,

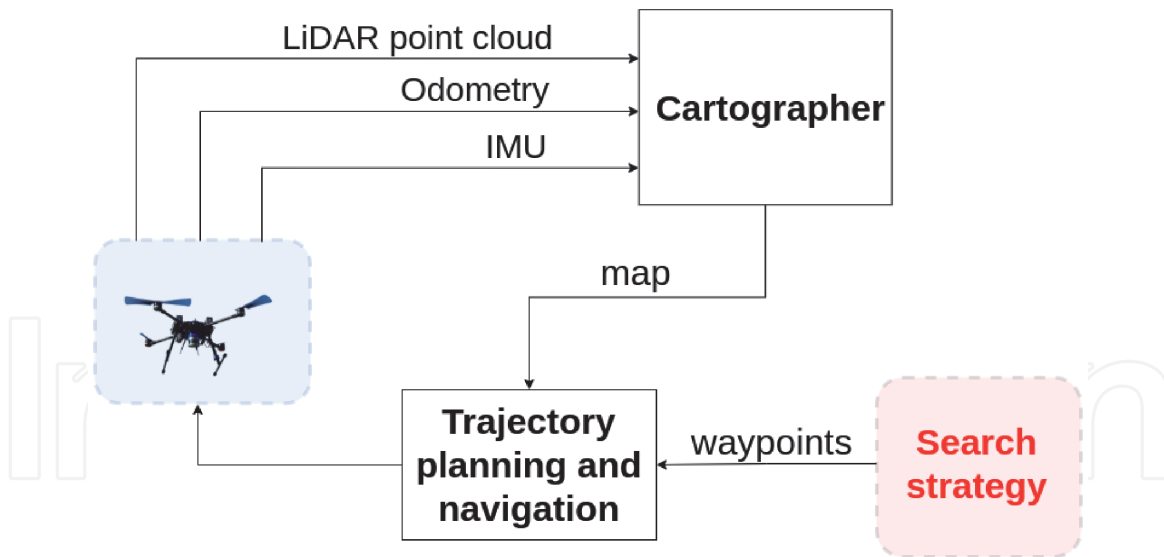


Figure 6.

Overall schematic diagram of autonomous exploration and search. The cartographer SLAM creates a map, that is used for the trajectory planning module. The generated trajectory navigates the robot towards the waypoint.

the strategy aims to explore the environment as quickly as possible while trying not to miss the objects of interest.

Waypoints can be generated in a variety of forms. For our scenario, we use the lawnmower pattern with the predefined horizontal and vertical size of an area and with the spacing value indicating how far apart the segments of the trajectory are. The lawnmower pattern is easy to implement and is suitable for exploration and search for the desired object.

The search strategy represents an input to the trajectory planning module, which generates the trajectory from the given waypoints. Objects of interest are detected during robot navigation. An overview of the proposed system is shown in **Figure 6**. The map provided by Cartographer is reliable and enables successful exploration while searching for fire.

In autonomous exploration, a suitable trajectory planner should be used to navigate the robot to the desired point. After the search algorithm computes the points in the specified arena, they are forwarded as waypoints to a trajectory planner. The robot starts following the planned trajectory and navigates to the next point. At the same time, the algorithm for detecting objects of interest runs in the background. For the safe navigation of the robot, we use the trajectory following solution described in [11].

6. Detection of objects of interest

Another major feature of the SAR systems is the ability to detect various objects of interest. In the ERL, the robots had to detect four different objects – a blocked entrance (blue rectangle), an unblocked entrance (green rectangle), damage to the building (red rectangles), and a missing worker (a dummy wearing a helmet and a safety vest). Some of the aforementioned objects are shown in the ERL arena in **Figure 7**. In the MBZIRC competition, one of the challenges was to locate and extinguish fires. Although there were variants of fires to be extinguished with water, we focused on the ground fires over which the robots had to throw blankets, as shown in **Figure 8**.

We used different strategies to detect objects of interest in these two competitions. For the ERL competition, we chose to use visual recognition using a neural



Figure 7. Objects of interest to be detected in ERL competition. Images display the missing worker mannequin, obstructed entrance (blue rectangle), and damages to the structures (red rectangles).



Figure 8. Ground fires to be found and covered with blankets during MBZIRC competition (Image source: <https://www.mbzirc.com/photo-album>).

network, for which a custom training dataset had to be created, even though the exact object was not known a-priori. To mitigate these issues and simplify the approach, we decided to use a color segmentation algorithm for object detection in the MBZIRC challenge. The simplicity of the objects to be found in this semi-structured challenge and the straightforwardness of the method and its application were the main reasons for this decision. Both methods and their properties are described in the rest of this section.

6.1 Neural network for ERL object of interest detection

The ERL search and rescue mission required recognition of several simple markers, and a human figure wearing a helmet and a vest, representing a missing worker. Even though the markers were rectangular and easily distinguished by color, we opted for the deep learning-based approach to detection because of the

complexity of human figure recognition. For this reason, we relied on the transfer learning approach, taking advantage of the pre-trained deep convolutional network based on the YOLO architecture. Since we only needed to recognize 4 object classes, but needed this in real-time on constrained computational resources, we opted for the tiny-yolo version of the network (**Figure 9**).

For our detection task, we created a small custom training dataset. A large number of images with simple rectangular objects can easily be created in different environments, with the limiting factor in the cost of manual labeling procedure. For the case of a mannequin representing the missing worker, the difficulty was in not knowing exactly what this object would look like. The final dataset consisted of around 150 images, out of which around 50 contained the human figure various poses and cluttered environments, and around 100 contained the rectangular markers.

Finally, we trained the network using DarkNet implementation, with the following parameters: $batch = 64$, $subdivision = 16$, $max_batches = 500200$ on a 4GB Nvidia GeForce GTX 980. To prevent overtraining, we saved the trained weights every 1000 iterations of learning, and in the end ran a forward pass on the test set for all of the saved stages. We chose the network weights based on the mAP score on the test set.

6.2 Color-based image filtering for object detection

The detection tasks in the MBZIRC competition were simpler compared to ERL, allowing for solution based on color segmentation. Hence, the fire detection in the third challenge of the MBZIRC competition was performed using color-based image filtering (**Figure 15**). The image received from the camera is filtered using pre-tuned Hue-Saturation-Value (HSV) thresholds. Contour detection is then performed on the filtered image, and contours with areas below a lower threshold are disregarded. If multiple contours remain, the one with the largest area is selected as the detected fire. The center of the contour is used as its position for the visual servoing procedure.

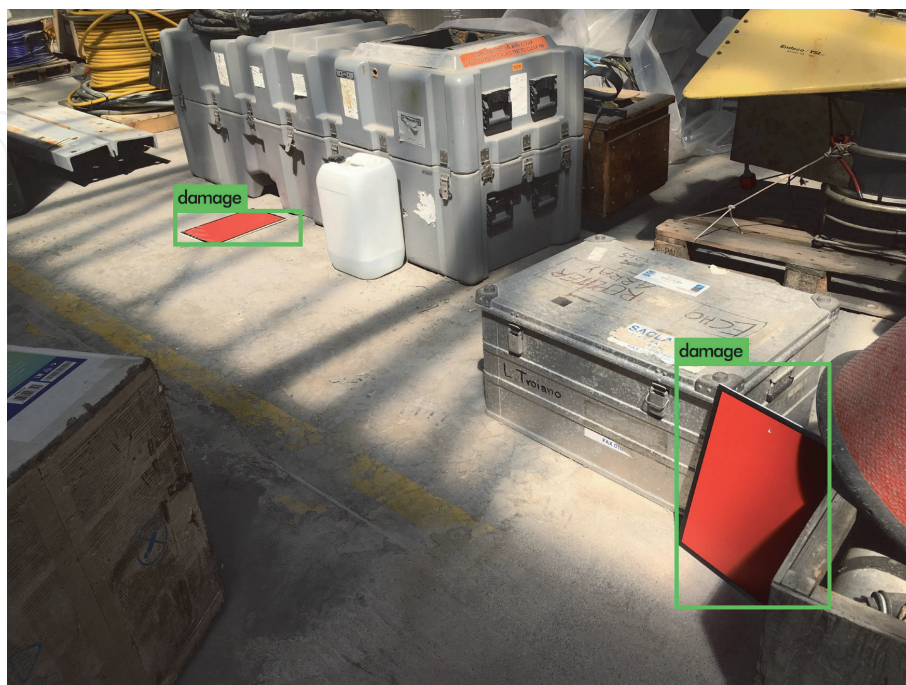


Figure 9.
Detection of the red rectangles using YOLOv3 real time object detection.

6.2.1 MBZIRC brick detection

Although this chapter focuses on SAR applications of MRS, one of the challenges during the MBZIRC competition was to achieve a more precise object manipulation in the wall building scenario that could be easily applied to the search and rescue domain. Our proposed solution involved precise 3D object detection that can be applied to other objects that need to be handled during emergency operations.

In the competition, the robots had to autonomously manipulate brick-shaped objects to build a wall. A ferromagnetic patch was attached to the top of the bricks for grasping. At the beginning of the challenge, the bricks were arranged as structured stacks, separated by color.

Initially, stacks of different colors are detected using a method similar to that used for fire detection, with different HSV thresholds used for red, green, blue and orange brick stacks. To distinguish individual bricks, ferromagnetic patches are detected within the convex hull of a detected brick stack. Contours resembling a rectangle are selected as candidate objects. To select a single brick for manipulation, a scoring scheme is used to calculate the desirability of each candidate. The desirability is calculated based on the area and image position of the candidate contour:

$$S = -w_x|x_p| + w_y y_p + w_A A \quad (1)$$

where A is the candidate area, and w_x , w_y and w_A are non-negative weights used to tune the scoring scheme and correspond to the x and y position in the image and the area, respectively. Switching from one patch candidate to another occurs only when the score of the new candidate is larger than that of the current one by some margin, to prevent rapid switching between candidates.

Our approach to determining the real-world pose of the detected magnetic patch, was to use an inverse of the Perspective-N-Point problem [28]. Assuming that the ferromagnetic patch is parallel to the ground, the positions of the endpoints of the detected rectangle in the image can be transformed into real positions using:

- Real-world camera position - from localization and manipulator kinematics.
- Height of the ferromagnetic patch - from the organized pointcloud obtained from the RealSense camera.
- Known width of the ferromagnetic patch.
- Camera calibration parameters.

7. Experimental results and evaluation

In this section, we present the results obtained during the challenge runs in the two robotics competitions. We analyze the performance of our robotic team in the autonomous mapping and exploration and object detection tasks.

7.1 Competition performance

Looking from the lens of the robotic competitions, we first present the results of the proposed UAV-UGV team in the competition arena. Here we provide challenge descriptions, and the scores obtained during the trials in the ERL-ER and MBZIRC.

7.1.1 ERL-ER competition score

The target scenario of ERL-ER is emergency response in an industrial environment affected by an earthquake. The UAV-UGV team serves as a first responder in this disaster-stricken area to secure the perimeter and enable the arrival of human rescue teams. The robots must search for two missing workers, find them as quickly as possible, and provide both with an emergency kit. The robots must also check the condition of the building after the earthquake. To do this, they output a detailed map of the surroundings to assess the safety of the area.

The competition took place in an outdoor arena of approximately $200m \times 30m$ with obstacles that the robots had to detect and navigate around. The competition arena also included a marquee that the aerial and ground robots had to enter. Rectangular red, green and blue A3-sized objects representing damage, blocked entrances and unobstructed entrances were distributed throughout the field.

At the tournament in Seville, five teams competed in a four-day competition. As specified in the rules of the competition, each team's final score was determined as the median value of all four attempts. **Table 3** shows the full results of all teams and their attempts, as well as the final score. At the expense of a lower score, we aimed for full autonomy during all trials. Therefore, the UAV, which was in an earlier stage of development, did not perform as well. However, the UGV did great, and we were able to use it to test the SLAM algorithm, as well as object detection. This tournament proved to be an excellent development and testing ground for the upcoming MBZIRC competition.

Other teams took the manual approach, such as teleoperation. For example, the user interface of Raptors was built in LabVIEW and the communication layer is based on the ROS software [30]. The system they developed supports the operator during teleoperation and during partial autonomy of the robots.

7.1.2 MBZIRC competition score

The MBZIRC competition involved three different challenges – tracking and capturing a target, building a wall, and disaster response in a fire scenario. Here we focus only on the latter, as it falls within the scope of SAR missions.

Challenge 3 of MBZIRC comprised a team of UAVs and a UGV working together to autonomously extinguish a series of simulated fires in an urban high-rise firefighting scenario. The arena, measuring $50m$ by $60m$, contained a tall structure (up to $20m$ high) simulating a high-rise building. Fires were simulated at various random locations on the floor of the arena (interior and exterior) and at various heights (ranging from $5m$ to $18m$) of the building to simulate a firefighting scenario

TEAM	Wed, Feb 20		Thu, Feb 21		Fri, Feb 22		Sat, Feb 23		median
	A	PB	A	PB	A	PB	A	PB	
LARICS	7	0	19	2	19	0	24	1	19
Raptors	0*	0	16	0	20	0	27	3	18
ENSTA Bretagne	0*	0	9	1	17	1	25	2	13
KAUST	0*	0	8	1	11	1	18	0	9.5
ENSTA Paris	0*	0	9	2	0*	1	19	1	4.5

Table 3.

Team results by day and the total score (median) for the ERL-RL Seville 2019 competition [29]. Column A represents team scores, PB penalized behaviors, while attempts marked with * had disqualifying behaviors (operating outside allowed limits or no map provided).

in a high-rise building. Teams were scored based on the number of tasks completed (fires extinguished), locations of fires extinguished, type of task completion (UAV, UGV, fire extinguishers, fire blankets), and speed of completion.

Our strategy for the challenge was to primarily focus on the ground fires, because they required minimal changes to the robot design. Therefore, we attempted to detect and extinguish the outdoor fires using a fire blanket deployed by the UAV. During the two trials, we managed to partially cover one of the two existing ground fires. This attempt was enough to place us in seventh place out of 20 participating teams. Although seemingly simple, the task proved very challenging for many teams, as shown by the results presented in **Table 4**.

An approach similar to ours was used by a team consisting Polytechnical University of Madrid, University Pablo Olavide and Poznan University of Technology [32]. In their localization, they also used a map-based approach and used 3D LiDAR scans in LOAM: Lidar Odometry and Mapping algorithm. A different approach was taken by the team from Czech Technical University in Prague, University of Pennsylvania and NYU [33] who used Global Navigation Satellite System (GNSS) for outdoor localization and a 2D LiDAR and stereo camera for indoor localization. For fire detection, both teams decided to use infrared cameras to detect heat sources. In the case of wall fires, this was a logical solution. Since we were not targeting these fires, our image-based detection worked just as well for the red floor fires. These results demonstrate the difficulty of SAR applications in real-world conditions in outdoor arenas and the potential for great progress in this area of research.

7.2 Autonomous mapping and exploration

In the field of autonomous mapping and exploration, we can observe an evolution of approaches. In the ERL competition, we explored the area manually (using an RC controller) or semi-autonomously (generating waypoints in the graphical interface and reaching them autonomously). For the MBZIRC challenge, we extended the method to autonomous exploration, where the robots could search the area and expand the map based on the autonomous exploration algorithm described in Section 5.

TEAM	Score	Time left	Rank
University of Seville, Tecnico Lisboa (IST Lisbon), CATEC	12.2625	0	1
Technical University of Denmark	10	0	2
University of New South Wales Sydney	10	0	2
Czech Technical University in Prague, University of Pennsylvania, NYU	7	0	4
Korea Advanced Institute of Science and Technology (KAIST)	7	0	4
University of Tokyo	5	120	6
UNIZG Faculty of Electrical Engineering and Computing	5	0	7
Polytechnic University of Madrid, University Pablo Olavide, Poznan University of Technology	5	0	7
Virginia Tech	4.5	0	9

Table 4.

The results of the best 9 out of 20 participating teams in the MBZIRC 2020 competition. These results were obtained in fully autonomous mode and therefore rank above manual approaches. The full results are available at [31].

The maps created during the ERL competition with UGV are shown in **Figure 10**, and 2D and 3D UAV maps are shown in **Figure 11**. In both the 2D and 3D maps, we see high-resolution detailed maps of the competition arena. Here, we demonstrate the ability of the proposed system to generate accurate maps during the mission runtime. However, in MBZIRC, we have gone a step further and combined the map generation with the autonomous waypoint selection method. The result of the obtained map can be seen in **Figure 12**.

The complete trajectory performed by an UAV is shown in **Figure 13**. The trajectory is obtained during the first experiment where we used manual navigation. The drift after loop closure was less than 0.3 m. We can conclude that Cartographer SLAM is useful when we navigate in the map and when we have a closed loop with trajectory planning and navigation. A more detailed analysis of Cartographer performance can be found in [16].

It can be observed that the accuracy of the final maps is not perfect compared to the mobile mapping technologies. This is due to the use of Cartographer SLAM, as it requires a feature-rich environment, a well-sampled IMU and loop closure to generate as accurate map as possible. In [16], it is shown that Cartographer accumulates more drift than other state-of-the-art strategies, but Cartographer can detect loop closure independently and the drift can be corrected. In other words, Cartographer accumulates a significant amount of drift (up to 2 meters), but upon returning near

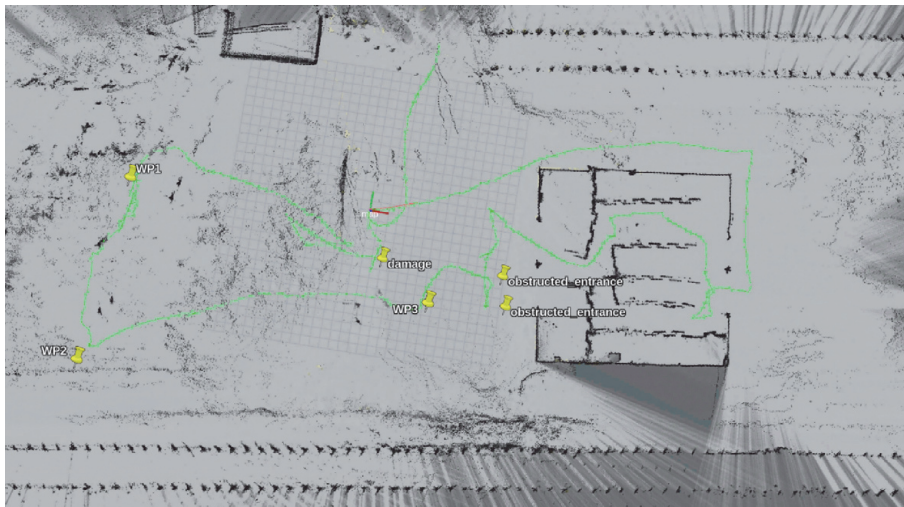


Figure 10.
2D map built by UGV during an ERL competition run.

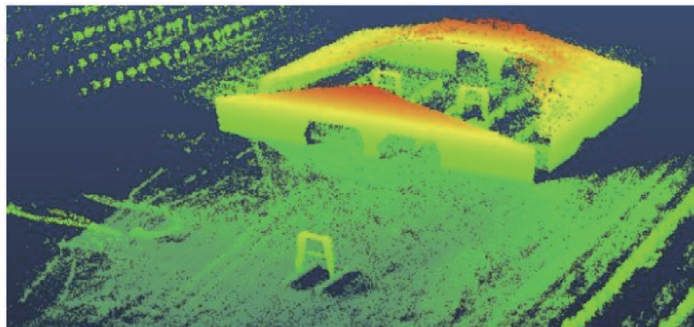
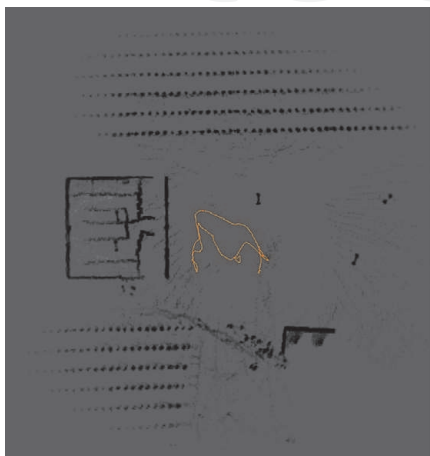


Figure 11.
2D (left) and 3D (right) map of the outdoor and indoor area of ERL arena generated by the UAV.

the start position, this drift is corrected (up to 0.2 meters) by closing the loop through the global SLAM.

In addition, closing the loop allows the correction of the cumulative drift, which provides higher pose and map accuracy during the flights. Due to the specific tasks performed in the challenges, closing the loop is not guaranteed to provide an accurate map. Future work will include significant efforts to improve map accuracy so that the map can be used not only in autonomous search and exploration tasks, but also in tasks that require a more accurate map, such as autonomous wall building, obstacle avoidance, and building inspection.

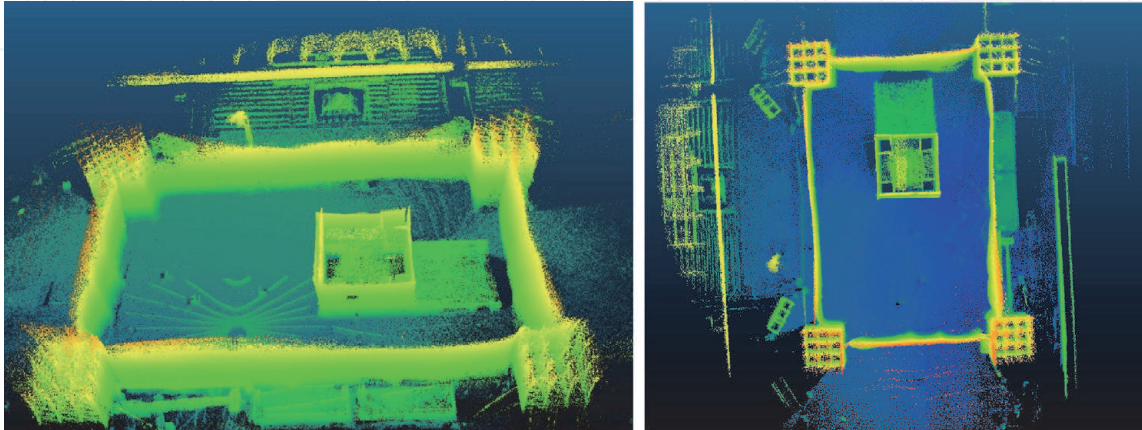


Figure 12.
Side view (left) and top view (right) of the 3D map of the outdoor MBZIRC area created with the UAV.

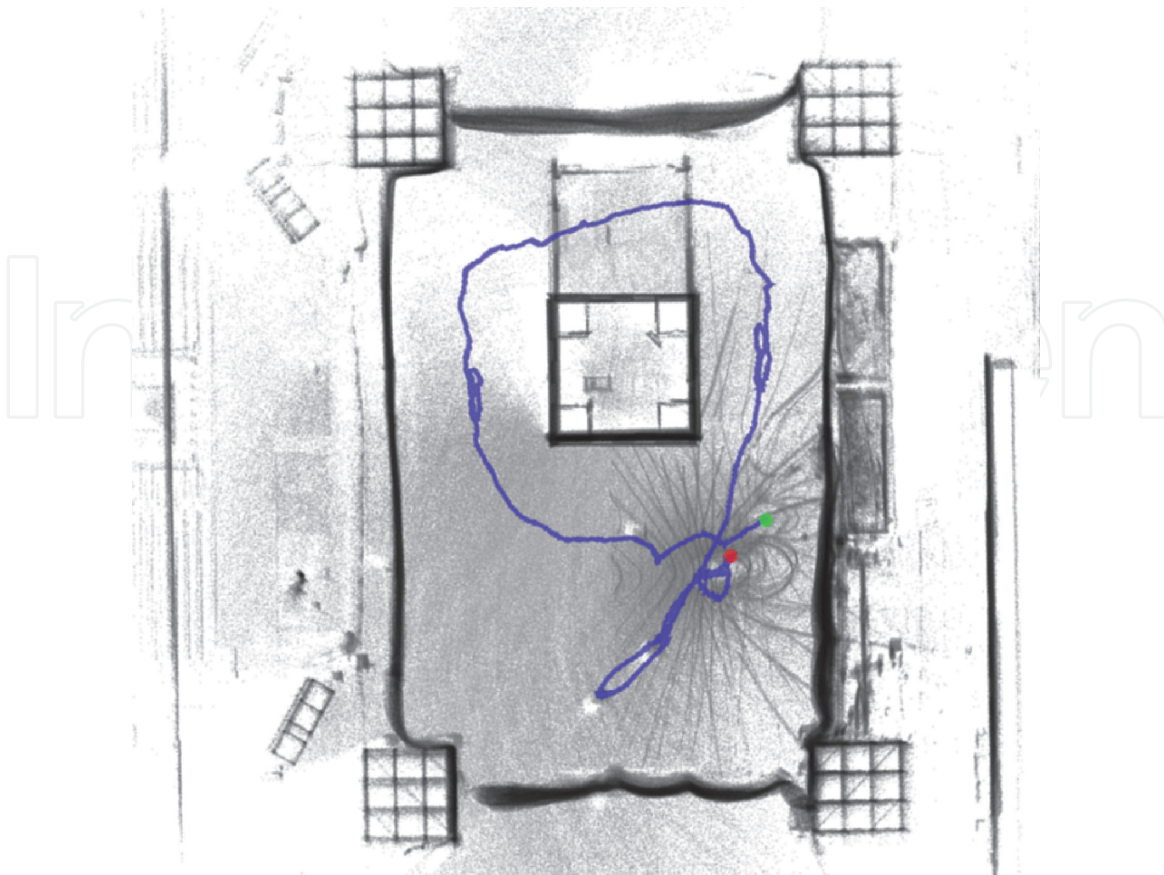


Figure 13.
A trajectory performed by an UAV in the MBZIRC challenge.

7.3 Detection of objects of interest

Next, we analyze the performance of the two object detection methods described in Section 6. In **Figure 14** we show several examples of detection of objects of interest by UGV using the algorithm described in 6 for ERL competition. These images were taken by a UGV on-board camera during several mission runs. The positions of the first three detected objects match those on the UGV map in **Figure 10**. We were able to detect the majority of the searched objects during the mission run. This mainly refers to the rectangular objects indicating blocked and unblocked entrances as well as damage to the building. However, due to the specifics of the missing worker mannequin, our neural network training set did not include the exact figure in any of the images. Therefore, we had difficulty performing this detection. This could be mitigated by training the network with a dataset containing a more accurate representation of the object being searched for, or by using a different approach. For example, color-based image segmentation might be appropriate since the safety vest has a very distinct orange color.

The output of the color-based fire detection algorithm in the MBZIRC competition is shown in **Figure 15**. The left side shows the original image, while the right side shows the image masked based on the predefined HSV threshold values. The green outline represents the object contour, while the yellow circle in the center represents the center of the object in the image. This approach provides a fast and efficient method for detecting simple objects of interest, which is certainly the case



Figure 14.
Image frames showing detections of objects of interest during ERL competition runs of the UGV.

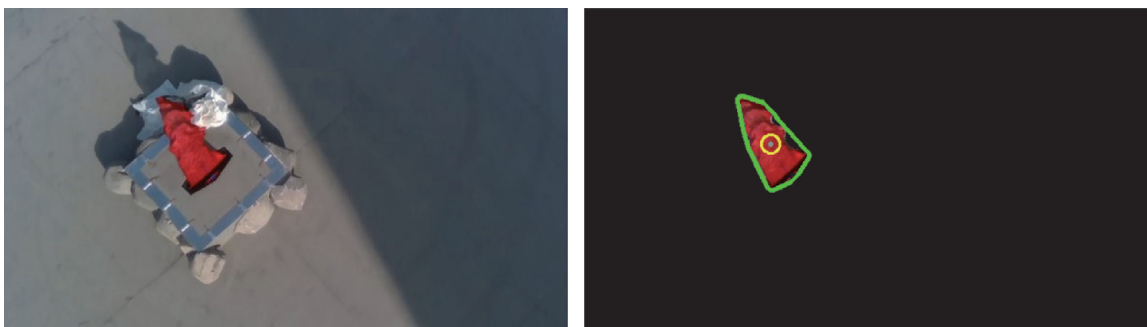


Figure 15.
Fire detection for the third MBZIRC challenge.

with a bright red fire. However, some care must be taken when setting the HSV thresholds, as the image colors are dependent on the illumination properties, which can change depending on outdoor lighting conditions.

8. Conclusions and lessons learned

This chapter presented the development of an autonomous aerial-ground search and rescue robotic team at the University of Zagreb, driven mainly by two large competitions: European Robotics League and Mohamed Bin Zayed International Robotics Challenge.

The hardware aspect of the robot team is developed considering specific requirements of the challenges, with the challenge-specific design decisions being more pronounced in our UGV design, ranging from simple 1DOF ejector arms to 6DOF mobile manipulator. On the other hand, our aerial platform did not require major hardware changes, as challenges did not require the UAV to interact with the environment on a similar scale to UGVs, but the trend is clearly changing and it can be expected that future challenges will require more and more interaction with the environment from the UAVs.

The biggest lesson to be learned from competition driven development, as evidenced by the software stack described in this chapter, is to design a modular system, both on the level of a single robot (interchangeable sensors and actuators that can easily be plugged into the existing control structure) but also on the level of a team. The modularity of our SAR robotic team is shown in using different sensors and algorithms for UGV mapping and navigation, and easily extending exploration strategies for the UAV.

Given the current state of our system, and the competitions being expected to become more and more challenging, the navigation through the unknown difficult environment is of utmost priority, which will require semantic segmentation of built 3D maps to differentiate traversable terrain from obstacles, both for the UGV and UAV. With the use of UAVs in SAR going from large open spaces to more confined urban areas, it is necessary to further develop the mapping and localization algorithms to provide a stable localization feedback for control and trajectory execution of UAVs in obstacle-rich environments.

Acknowledgements

This work has been supported by European Commission Horizon 2020 Programme through project under G. A. number 810321, named Twinning coordination action for spreading excellence in Aerial Robotics - AeRoTwin and by the Mohammed Bin Zayed International Robotics Challenge. The work of doctoral students Barbara Arbanas (DOK-2018-01), Ana Batinović (DOK-2018-09), Marko Car (DOK-2015-10), and Marsela Polić (DOK-2018-09) has been supported in part by the “Young researchers’ career development project–training of doctoral students” of the Croatian Science Foundation. DOK-2018-01 is financed by the European Union from the European Social Fund.

Abbreviations

ERL	European Robotics League
MBZIRC	Mohamed Bin Zayed International Robotics Challenge

UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
SAR	Search and Rescue
MRS	Multi-Robot Systems
RTK	Real-Time Kinematic positioning
TOPP-RA	Time-Optimal Path Parametrization by Reachability Analysis
SLAM	Simultaneous Localization and Mapping

IntechOpen

IntechOpen

Author details

Barbara Arbanas*, Frano Petric, Ana Batinović, Marsela Polić, Ivo Vatavuk, Lovro Marković, Marko Car, Ivan Hrabar, Antun Ivanović and Stjepan Bogdan
Laboratory for Robotics and Intelligent Control Systems, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

*Address all correspondence to: barbara.arbanas@fer.hr

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.
- [2] D. Drew, "Multi-agent systems for search and rescue applications," *Current Robotics Reports*, vol. 2, pp. 189–200, 06 2021.
- [3] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, L. M. Gambardella, R. Siegwart, and D. Scaramuzza, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.
- [4] A. F. T. Winfield, M. P. Franco, B. Brueggemann, A. Castro, M. C. Limon, G. Ferri, F. Ferreira, X. Liu, Y. Petillot, J. Roning, F. Schneider, E. Stengler, D. Sosa, and A. Viguria, "euRathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots," in *Towards Autonomous Robotic Systems*, pp. 351–363, Springer International Publishing, 2016.
- [5] A. F. T. Winfield, M. Palau Franco, B. Brueggemann, A. Castro, G. Ferri, F. Ferreira, X. Liu, Y. Petillot, J. Roning, F. Schneider, E. Stengler, D. Sosa, and A. Viguria, "euRathlon and ERL Emergency: A multi-domain multi-robot grand challenge for search and rescue robots," in *ROBOT 2017: Third Iberian Robotics Conference*, pp. 263–271, Springer International Publishing, 2018.
- [6] "MBZIRC competition." <https://www.mbzirc.com/>. Accessed: 2021-05-27.
- [7] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada, "Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, vol. 6, pp. 739–743 vol.6, 1999.
- [8] "DARPA robotics challenge." <https://www.darpa.mil/program/darpa-robotic-s-challenge>. Accessed: 2021-05-27.
- [9] A. Jacoff, E. Messina, and J. Evans, "A standard test course for urban search and rescue robots," *Performance Metrics for Intelligent Systems, Workshop — —* — NIST, 2000-08-01 2000.
- [10] G. D. Cubber, D. Doroftei, K. Rudin, K. Berns, A. Matos, D. Serrano, J. M. Sanchez, S. Govindaraj, J. Bedkowski, R. Roda, E. Silva, S. Ourevitch, R. Wagemans, V. Lobo, G. Cardoso, K. Chintamani, J. Gancet, P. Stupler, A. Nezhadfar, M. Tosa, H. Balta, J. Almeida, A. Martins, H. Ferreira, B. Ferreira, J. Alves, A. Dias, S. Fioravanti, D. Bertin, G. Moreno, J. Cordero, M. M. Marques, A. Grati, H. M. Chaudhary, B. Sheers, Y. Riobo, P. Letier, M. N. Jimenez, M. A. Esbri, P. Musialik, I. Badiola, R. Goncalves, A. Coelho, T. Pfister, K. Majek, M. Pelka, A. Maslowski, and R. Baptista, *Search and Rescue Robotics - From Theory to Practice*. InTech, Aug. 2017.
- [11] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, "Decentralized planning and control for UAV-UGV cooperative teams," *Autonomous Robots*, vol. 42, no. 8, pp. 1601–1618, 2018.
- [12] I. Noda and M. Hatayama, "Common frameworks of networking and information-sharing for advanced rescue systems," in *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 245–249, 2004.

- [13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.
- [14] M. Nieuwenhuisen, M. Beul, R. A. Rosu, J. Quenzel, D. Pavlichenko, S. Houben, and S. Behnke, "Collaborative object picking and delivery with a team of micro aerial vehicles at MBZIRC," in *2017 European Conference on Mobile Robots (ECMR)*, IEEE, Sept. 2017.
- [15] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, 2016.
- [16] R. Milijas, L. Markovic, A. Ivanovic, F. Petric, and S. Bogdan, "A comparison of lidar-based slam systems for control of unmanned aerial vehicles," 2020.
- [17] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, Feb. 2017.
- [18] H. Pham and Q. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Transactions on Robotics*, vol. 34, pp. 645–659, June 2018.
- [19] D. Puljiz, M. Varga, and S. Bogdan, "Stochastic search strategies in 2d using agents with limited perception," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 650–654, 2012.
- [20] A. Batinović, J. Oršulić, T. Petrović, and S. Bogdan, "Decentralized strategy for cooperative multi-robot exploration and mapping," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9682–9687, 2020.
- [21] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3d exploration," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4528–4535, 2021.
- [22] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3D frontier-based exploration tool for mavs," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 348–352, 2015.
- [23] D. Stormont, A. Bhatt, B. Boldt, S. Skousen, and M. Berkemeier, "Building better swarms through competition: lessons learned from the aaai/robocup rescue robot competition," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, pp. 2870–2875 vol.3, 2003.
- [24] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, "The sherpa project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–4, 2012.
- [25] N. Miskovic, M. Bibuli, A. Birk, M. Caccia, M. Egi, K. Grammer, A. Marroni, J. Neasham, A. Pascoal, A. Vasiljevic, and Z. Vukic, "Overview of the FP7 project "CADDY - Cognitive Autonomous Diving Buddy"," in *OCEANS 2015 - Genova*, IEEE, May 2015.
- [26] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, "Cooperative aerial-ground multi-robot system for automated construction tasks," *IEEE Robotics and Automation Letters*, vol. 5, pp. 798–805, Apr. 2020.
- [27] B. Yamauchi, "A frontier-based approach for autonomous exploration,"

in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.*, pp. 146–151, 1997.

[28] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, p. 381–395, June 1981.

[29] “ERL-ER seville 2019 competition results.” <https://sites.google.com/catec.aero/erl-emergency-2019/scores>. Accessed: 2021-07-01.

[30] A. Wegierska, K. Andrzejczak, M. Kujawinski, and G. Granosik, “Using labview and ros for planning and coordination of robot missions, the example of erl emergency robots and university rover challenge competitions,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 13, pp. 68–81, 09 2019.

[31] “MBZIRC competition results.” <https://www.mbzirc.com/winning-teams/2020/>. Accessed: 2021-07-01.

[32] S. Martinez-Rozas, R. Rey, D. Alejo, D. Acedo, J. A. Cobano, A. Rodriguez-Ramos, P. Campoy, L. Merino, and F. Caballero, “Skyeye team at MBZIRC 2020: A team of aerial and ground robots for GPS-denied autonomous fire extinguishing in an urban building scenario,” *CoRR*, vol. abs/2104.01834, 2021.

[33] V. Spurny, V. Pritzl, V. Walter, M. Petrlik, T. Baca, P. Stepan, D. Zaitlik, and M. Saska, “Autonomous firefighting inside buildings by an unmanned aerial vehicle,” *IEEE Access*, vol. 9, pp. 15872–15890, 2021.