Special Issue: DARPA Subterranean (SubT) Challenge

Regular Article

A Heterogeneous Unmanned Ground Vehicle and Blimp Robot Team for Search and Rescue using Data-driven Autonomy and Communication-aware Navigation

Chen-Lung Lu[1,2,*], Jui-Te Huang[1,2,*], Ching-I Huang[1,2], Zi-Yan Liu[1,2], Chao-Chun Hsu[1,2], Yu-Yen Huang[1,2], Siao-Cing Huang[1,2], Po-Kai Chang[1,2], Zu Lin Ewe[1,2], Po-Jui Huang[1,2], Po-Lin Li[1,2], Bo-Hui Wang[1,2], Lai-Sum Yim[1,2], Sheng-Wei Huang[1,2], MingSian R. Bai[3] and Hsueh-Cheng Wang[1,2,4]

[1]National Chiao Tung University (NCTU)
[2]National Yang Ming Chiao Tung University (NYCU)
[3]National Tsing Hua University (NTHU)
[4]Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan.

Abstract: This paper describes the architecture and implementation of a heterogeneous team comprising unmanned ground vehicles and blimp robots capable of navigating unknown subterranean environments for search and rescue missions. The ground vehicles are equipped with a range of sensors for accurate perception, localization, and mapping. The blimps feature a long flight duration and collision tolerance when traversing uneven terrain. The design of the system was meant to satisfy the requirements of the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge in terms of perception capability and autonomy. To facilitate navigation through smoke-filled spaces, we employed novel millimeter wave radar to enable cross-modal representations for integration via deep reinforcement learning. The autonomy of the proposed scheme was augmented using simulations to train deep neural networks, thereby allowing the system to perform sequential decision-making for collision avoidance and navigation toward a specific goal. The navigation system was evaluated in the DARPA SubT Urban Circuit, and quantitative localization results and recovery strategy from failures was discussed. The proposed communication system comprises mesh WiFi with XBee (ZigBee network with XBee radios) and ultra-wideband (UWB) communication modules as well as spherical nodes that can be shot out like a cannonball and miniature cars deployed as mobile nodes. The propagation and radio signal strength index of various modules were evaluated using data collected during field tests in order to overcome the uncertainties of subterranean environments, including non-line-of-sight propagation, multipath propagation, and

*Chen-Lung Lu and Jui-Te Huang contributed equally.

Received: 16 January 2021; revised: 16 May 2021; accepted: 1 August 2021; published: 2 May 2022.

Correspondence: Hsueh-Cheng Wang, National Chiao Tung University (NCTU); National Yang Ming Chiao Tung University (NYCU); Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan., Email: hchengwang@g2.nctu.edu.tw

fading reception. We also discuss the lessons learned during this project and reflect on future plans.

Supplementary Material
For a supplementary video visit: https://vimeo.com/501143422
For the open-source dataset visit: https://arg-nctu.github.io/projects/team_nctu_subt.html

Keywords: subterranean robotics, learning, navigation, rotorcraft

1. Introduction

Search and rescue (SAR) robots are generally used to improve situational awareness and explore potentially dangerous areas in the event of disaster. SAR robots have been developed for operations on the ground, in the air, and in water (Delmerico et al., 2019). Legged robotic systems have also been developed to perform complex missions on particularly uneven terrain. Unmanned ground vehicles (UGVs) and quadrotor drones were used in the collaborative mapping of damaged buildings after the 2011 Tohoku earthquake in Japan (Michael et al., 2012). Unmanned aerial vehicles (UAVs) operating at low altitude provide reconnaissance with a wide field of view (FOV) (Gawel et al., 2018). In some instances, UAVs are flown ahead of UGVs to detect holes and other obstacles (Stentz et al., 2002), to map out possible ground routes (Guérin et al., 2015), or to assist UGVs in navigating through unknown environments (Kelly et al., 2006) (Dang et al., 2020).

Quadcopter UAVs (referred to as drones) are particularly popular for their speed, small airframes, and precise maneuverability over rough terrain. However, drones are limited in terms of flight time (16–40 min) (Sa et al., 2017) and highly prone to damage in the event of collision. UAVs can be attached to UGV platforms (Richardson et al., 2013) to reduce power consumption. One study developed a hybrid UAV/UGV vehicle (Fan et al., 2019) capable of ground travel (to reduce power consumption) as well as flight (to overcome difficult ground conditions). Rolling is far more efficient than flying in terms of navigation, autonomy, run time, and traveling distance.

Existing robot perception systems are highly susceptible to optical sensor degradation under smoke or foggy conditions, and the Defense Advanced Research Projects Agency (DARPA) Subterranean (SubT) Challenge is meant to simulate these conditions. Laser range finder (LiDAR) inputs tend to be unreliable, due to the sensitivity of cameras to variations in lighting. Degraded sensing due to low illumination and/or a featureless geometry can seriously undermine the localization performance of conventional sensors. Alternative approaches to dealing with visually degraded environments include thermal imaging (Shivakumar et al., 2020) for artifact segmentation, and millimeter-wave (mmWave) radar fusion (Kramer et al., 2020) for ego-velocity estimation. Radar data using self-supervised learning have been used to derive a traversable route (Broome et al., 2020). A dataset with raw radar samples in various scenes was proposed focusing on robot perception using millimeter-wave radar (Kramer et al., 2021). Other studies have combined ultra-wide-band (UWB) ranging measurements with inertial measurement units (IMUs) and LiDAR to facilitate localization in indoor environments. Ranging measurements from known UWB anchors combined with IMUs using extended Kalman filters was proposed for a robotic blimp (Mai et al., 2018). Localization in tunnel-like environments can also be improved by combining ranging measurements in known positions with LiDAR (Zhen and Scherer, 2019). UWB measurements can be considered an environmental "feature" for use in estimating robot trajectories relative to a map (Song et al., 2019).

Facilitating the situational awareness of operators at base stations is crucial to SAR operations in underground environments. Radio signal propagation can be affected by stochastic effects, including shadowing and fading under the effects of unstructured/irregular layouts. (Hsieh et al., 2008) constructed a radio signal strength indicator (RSSI) mapping scheme to plan exploration tasks involving multiple robots. A hybrid mesh network to extend communication distance was proposed (Nagatani et al., 2013), which integrated a high-power wireless communication device with a cable reel-out mechanism using a twisted-pair metal cable. Deployable and dynamic nodes have

(a) Tunnel Circuits. **(b)** Urban Circuits. **(c)** Shimen Reservoir.

Figure 1. Heterogeneous robot team, including an autonomous blimp and UGVs (Clearpath Husky and Jackal). The robot team was tested in the Tunnel and Urban circuits in the DARPA SubT Challenge as well as in local field-testing sites including artificial tunnels beneath Shimen Reservoir and Buddha's Hand (BH) ocean-formed cave.

also been used to extend the coverage of wireless signals from base stations; however, the placement of nodes in usable or optimal positions is often prevented by physical obstacles. Flocks of UAVs with nodes carried by ground vehicles is another approach to improve connectivity (Basu et al., 2004; Min et al., 2016). Regardless, constructing RSSI maps and maintaining connections among robots and nodes are main challenges pertaining to communication systems in SAR scenarios.

In the following, we outline our proposal for a heterogeneous team of robots, focusing on mobility, perception/autonomy, and communications:

1. **Low-cost autonomous blimps for flights of extended duration**: Blimps consume far less power than drones and are far more tolerant of collisions, both of which are primary considerations in SAR scenarios as well as the DARPA SubT Challenge (DARPA, 2019).
2. **Platform-agnostic mmWave sensor tower with learning-based algorithms for navigation**: In experiments, mmWave radar devices have proven highly effective as sensors in visually degraded environments. We developed a learning-based navigation policy network to facilitate range detection using mmWave radar or LiDAR.
3. **Deployable nodes for localization and communication**: Estimating the RSSI of XBee and UWB modules to construct RSSI maps is essential to communications in underground environments. We developed and field tested deployable nodes and evaluated them extensively in regular and irregular real-world environments.

This article extends our previous work (Huang et al., 2019) and (Huang et al., 2021) with an overview of the design and performance of heterogeneous robot team developed by the NCTU Team for the DARPA SubT Challenge. Section 2 summarizes the system architecture. Section 3 describes the design of the software infrastructure and the set of key algorithms developed for environmental perception and autonomous navigation. Section 4 describes the communication nodes and the RSSI maps. Section 5 reflects on the performance of the proposed system and lessons learned.

2. System Architecture

2.1. Problem Statements and Challenges

Navigating through subterranean environments, finding and locating designated artifacts, and then reporting the results imposes a number of daunting challenges, which can be categorized as follows:

- **Mobility:** Navigating through rough terrain within a limited time window. The duration of each run in the DARPA Tunnel and Urban Circuits competitions was one hour. The developed

vehicles required a certain moving speed to search the large environments, i.e., $>$ 2 km long competition sites.

- **Perception and Autonomy:** Ensuring robust navigation in the search for artifacts under degraded sensing conditions.

  1) Artifact Search: the artifacts in the Tunnel Circuit were all visible, including a survivor, a cell phone, a backpack, a drill, and an extinguisher to be located. The Urban Circuit included four visible artifacts (survivor, cell phone, backpack, and vent), and one artifact of CO_2 gas requiring olfactory sensing. Under the scoring scheme of the competition, every valid artifact report would earn the team one point, as long as the artifact type was correct and the reported location estimated by localization and mapping was within 5 m of the ground truth location (Euclidean distance).

  2) Localization: the simultaneous localization and mapping (SLAM) problem was represented is a factor graph (Kaess et al., 2012), including robot poses as states and the constraints that relate these hidden states. A SLAM algorithm was formulated as odometry u to connect consecutive mobile robot poses x_i, and artifact detections were represented as measurements m_j of the artifact landmarks or deployable node landmarks l_k. The constraints $m_j = \{r_j, \theta_j\}$ consisted of a range r and bearing θ. The implementations of loop closure were carried out in the lightweight and ground-optimized lidar odometry and mapping (LeGo-LOAM) library (Shan and Englot, 2018), where our artifact and deployable node landmarks were not used for loop closure.

  3) Autonomous navigation: multiple robot platforms were assumed to operate in static environments without other dynamic obstacles. We aim at developing decentralized collision avoidance control policies π_{fw} for forward exploration and goal navigation π_{goal}, where the parameters q of the policies were trained in the Virtual SubT Gazebo simulation.
- **Communication and Coordination:** Given any robot pose x_i and the base station position p, the RSSI map is a function $\psi : (p, x_i) \rightarrow \mathbb{R}$ that returns the estimated RSSI between p and x_i. Our mission was to maintain the RSSI above a certain level, and transmit robot pose x_i and the mapping $\mathcal{C}_i$ to the base station, where a human supervisor designated subgoal points $\{g_{r,1}, g_{r,2}, ..., g_{r,n}\}$ to each robot r for goal navigation. This paper focused on systematically evaluated radio propagation of the line of sight (LOS) RSSI gradient $G^{LOS}_{j,k}$ and nonline of sight (NLOS) $G^{NLOS}_{j,k}$ in the j subterranean environments $\{Tunnel, Cave, Corridor\}$ using the k mesh communication modules $\{UWB, XBee\}$.

2.2. System Architecture

As shown in Figure 2, the proposed system was designed to tackle the challenges of mobility, perception, autonomy, localization, and communication. Mobility is dealt with using a heterogeneous UGV-blimp with multimotor propulsion to carry a large payload of sensors through rough environments for an extended duration. Perception and autonomous operations rely on deep reinforcement learning (RL) for goal navigation and collision avoidance. We also developed deployable nodes for localization and communication based on analysis of radio propagation in subterranean environments.

Figure 3 presents a schematic illustration showing the computational hardware used in the proposed system. The system can be divided into the blimp (Duckiefloat), the UGV (Husky), the base station, and DARPA infrastructure. Duckiefloat and Husky were tasked with communications and perception. Note that Husky was also meant to operate autonomously using ranging and location sensors.

2.3. Robotic Blimp - Duckiefloat

The design of the proposed Duckiefloat robotic blimp was inspired by the resource-constrained Duckietown (Paull et al., 2017) and Duckiefly (Brand et al., 2018). Blimps have been widely used for transport and surveillance (Fukao et al., 2003). Small blimps equipped with high-capacity batteries

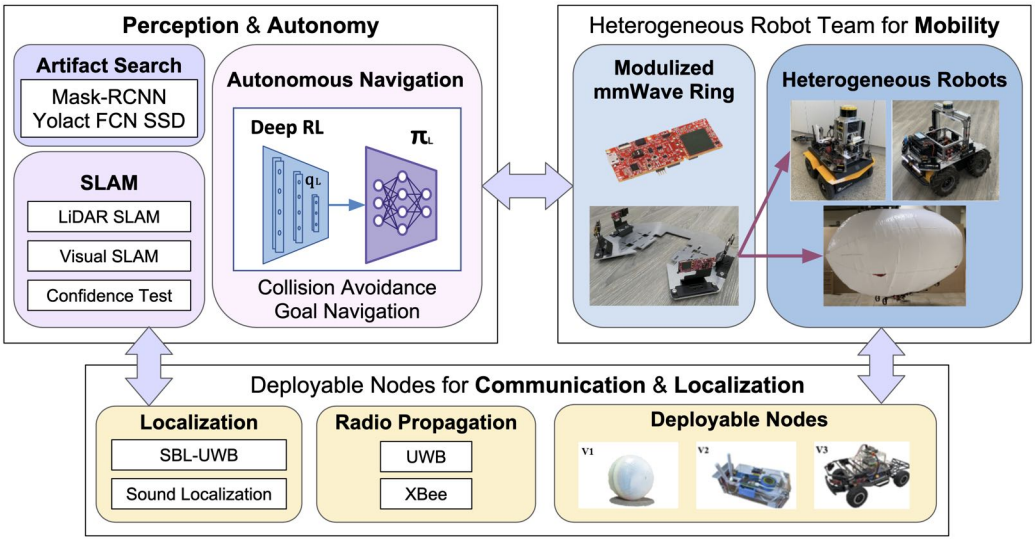


Figure 2. System architecture: (1) Proposed heterogeneous team of robots for mobility; (2) Algorithms used for perception and autonomy; and (3) Deployable nodes for communication and localization.

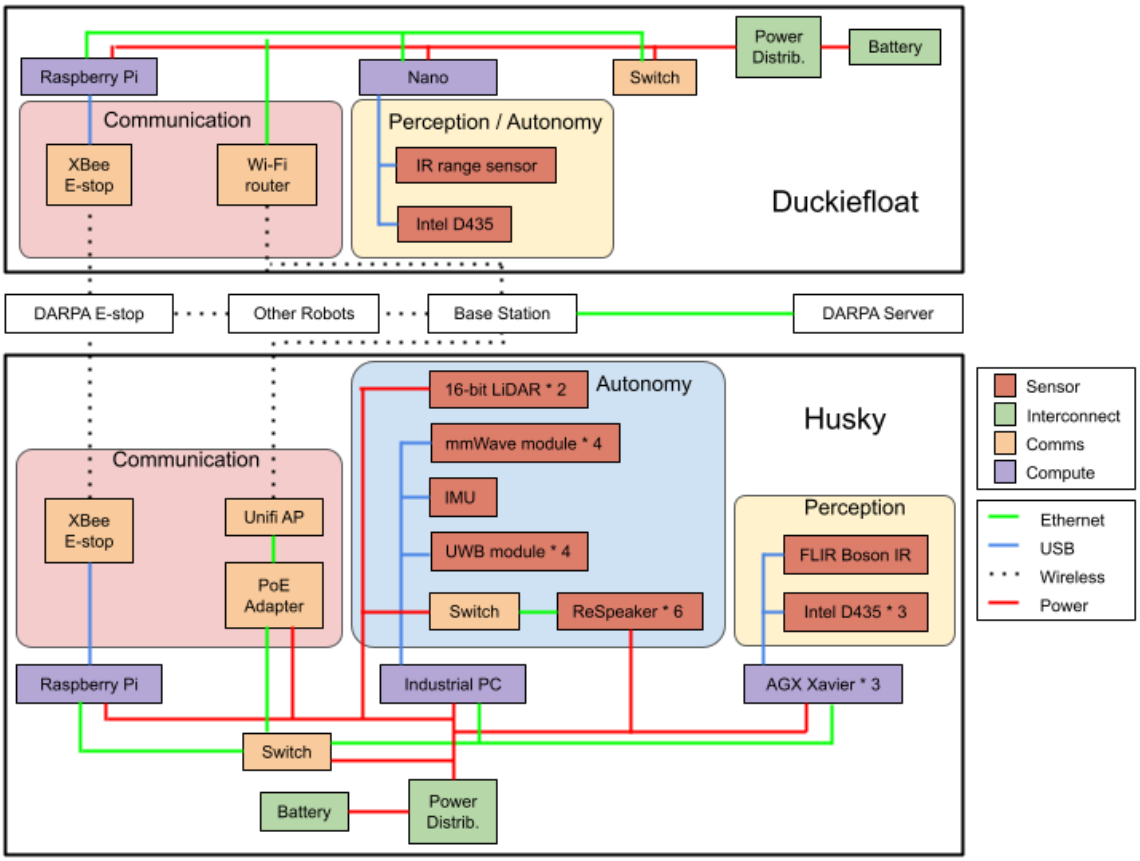


Figure 3. Computational hardware used in the Duckiefloat, Husky, base station, and DARPA modules. Note that the Husky and Duckiefloat play different roles in the heterogeneous robot team.

Table 1. Local field test environments.

Field Site	Total Length	Site Area	Development Learned from Fieldwork
18-Peak Mt. Dugout	20 m	15 m x 10 m	Envelop and Frame
Houli Tunnel	1.2 km	1200 m x 3.2 m	Multirotor Design
Indoor Corridor	120 m	75 m x 50 m	Wider Sensor FOV
Shimen Tunnel	4 km, 4 Levels	120 x 230	Airflow/Power Consumption
BH Cave	200 m	65 m x 28 m	Airflow/Power Consumption

(a) WWII-era bunker for testing. **(b)** Abandoned railway tunnel in Houli.

Figure 4. Field tests of robotic blimp before attempting the Tunnel circuit

can hover stably for an extended duration (Palossi et al., 2017), and such devices are highly tolerant of collisions, which is crucial when navigating through unstructured environments (Yao et al., 2017; Yao et al., 2019; Burri et al., 2013).

The buoyancy, mass effects, and aerodynamics of blimps are similar to those of submarines. Several model-based schemes have been devices to control the movement of blimps (González et al., 2009; Fedorenko and Krukhmalev, 2016). One study by (Ko et al., 2007) used model-free RL with a Gaussian-process model for yaw control. (Rottmann et al., 2007) used RL with Gaussian processes to approximate the Q function of altitude controls. (Hygounenc et al., 2004) used vision-based localization and mapping (SLAM) for blimp localization. (Mai et al., 2018) implemented a local pose estimator using three UWB radio sensors, one gyroscope on a blimp, and four additional UWB sensors in an indoor stadium.

Note that throughout the development of the proposed system (April 2019 to April 2021), we carried out field tests in multiple tunnels and caves. Table 1 summarizes the total length of the explorations, the area of the sites, and advancements derived from these expeditions.

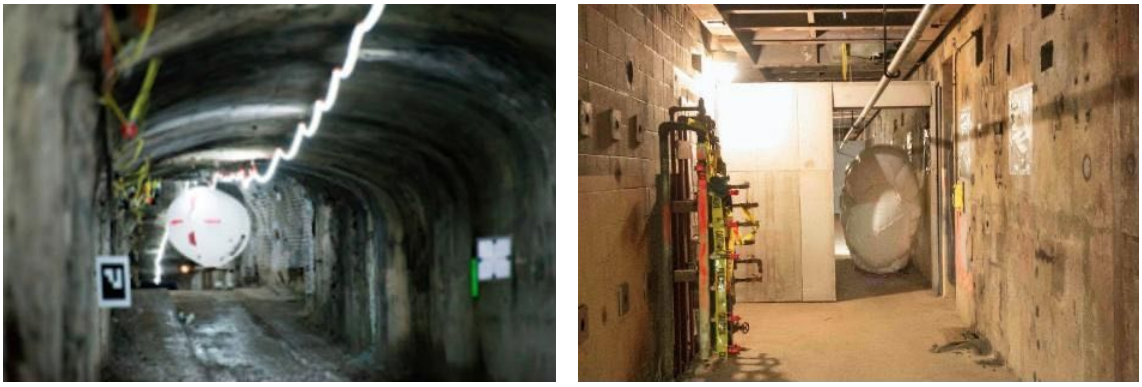
2.3.1. Envelope, Payload, and Frame Designs

The first field test in April 2019 was carried out in an abandoned WWII bunker, which included two intersections within a distance of approximately 20 m. The original route to deeper tunnels was closed due to safety considerations.

We had to limit the envelope of the blimp to deal with the tight passages in this subterranean environment; however, this had a negative impact on payload, which included a battery for mobility and illumination. Our objective was to power the blimp for one hour; i.e., the duration of each run in the DARPA Tunnel and Urban Circuits competitions. The robotic blimp in its DF-v1 configuration had all of the sensors, computing units, and motors attached directly to the envelope. This blimp succeeded in traversing the route using RC controls; however, the process was somewhat

Table 2. Envelop developments.

Circuit	Tunnel DF-v1	Urban DF-v2	Cave/Final DF-v3	Note
Envelope				floating (g)
1.2m x 1.2m x 2.5m	✓			1,600
1.5m x 1.5m x 2.5m		✓		2,600
1.8m x 0.8m x 2m		✓		950
1.2m x 1.5m x 2.5m			✓	2,000
Multirotor Designs				
4 DC motors, 3-inch propellers (4)	✓			
6 DC motors, 3-inch propellers (6)		✓		
4 DC, 2 Brushless, Servo motors (7)			✓	



(a) Tunnel Circuit **(b)** Urban Circuit

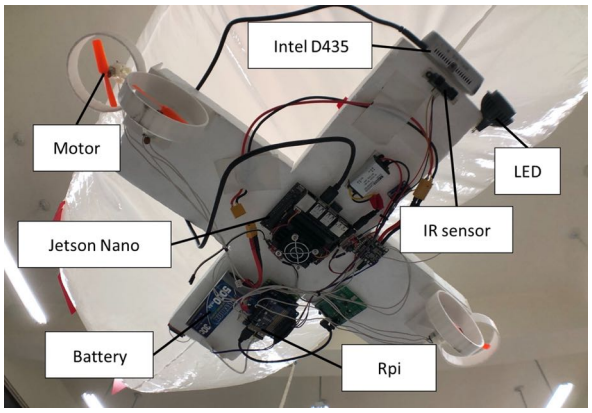
Figure 5. Envelope designs of duckiefloat in the Tunnel and Urban Circuit. The geometry of corridors varied more in the Urban Circuit than in the Tunnel Circuit. Thus, we modified the shape of Duckiefloat and removed some of the payload to allow passage through narrow openings.

awkward with the motors attached to the envelope. In an earlier version, we employed a cross-shaped Styrofoam platform slung beneath the envelop to carry sensors, motors, and controllers. For the final version, we fabricated a carbon-fiber frame using a three-dimensional (3D) printer, as shown in Figure 6. The robotic blimp measured 1.2 m × 1.2 m × 2.5 m, which was small enough to travel through the Tunnel circuit. When filled with helium gas, the blimp had a payload capacity of approximately 1600 g, which was sufficient to carry the electronic devices and batteries (Table 2). The geometry of tunnels in the coal mine did not vary. We observed a difference in atmospheric pressure between the bunker and the controlled indoor environments, which necessitated modifications to the envelope. We increased the dimensions of the envelope for the Urban Circuit to 1.5m × 1.5m × 2.5m to increase the payload capacity to 2500 g, which was necessary for additional devices including LiDAR.

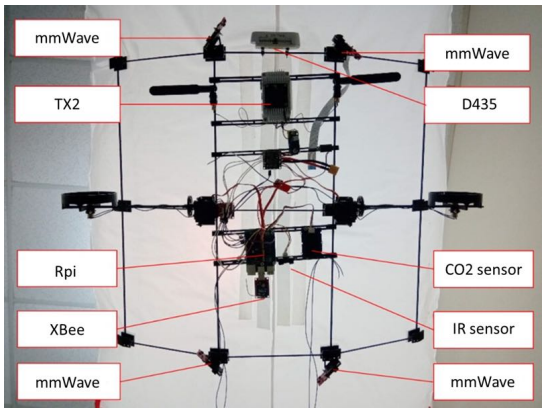
Note that this larger blimp would have been too large to pass through the small openings in the Urban Circuit Beta course in the DARPA competition. Thus, we had no choice but to modify the dimensions of the envelope to 0.8 m × 1.8 m × 2 m [Figure 5 (b)]. This reduced the payload to just 950 g, thereby necessitating the removal of several sensors, which in turn shifted the center of mass with a corresponding effect on blimp navigation and controls.

2.3.2. Multirotor Design

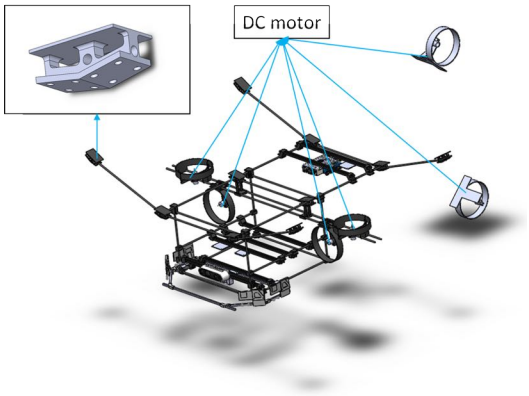
We eventually moved to the abandoned Houli railway tunnel measuring 1.2 km in length. Note that the tunnel is also a popular site for tourists and cyclists, which limited our access. Autonomous navigation was conducted along the tunnel for roughly 300 m. The blimp could have travelled farther; however, its progress was interrupted by passing pedestrians and cyclists. During this run,



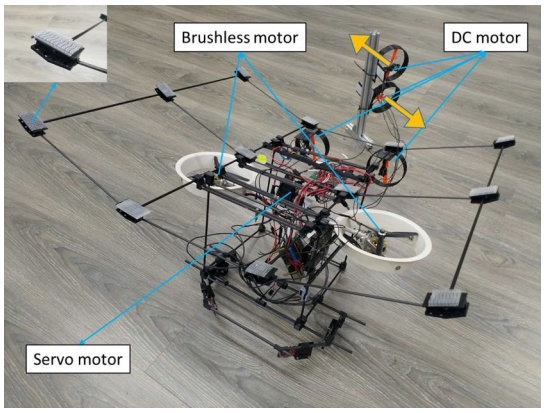
(a) DF-v1 motors, frame, and sensors.



(b) DF-v2 frame and sensors.



(c) DF-v2 multi-rotor designs.



(d) DF-v3 multi-rotor design.

Figure 6. Hardware comparison between the various versions of Duckiefloat. The tail motors increased control over heading. The DC motors in the DF-v2 design provided position control in the x axis and z axis as well as orientation control in the z axis (i.e., yaw). The brushless motors in the DF-v3 design provided hovering ability and minor positional control in the x axis. A servo motor was installed to adjust the angle of the brushless motors in order to balance hovering against forward movement.

the blimp had a number of minor collisions with the walls, which caused a slow leak but did not hinder operations. The propulsion system comprised two pairs of DC motors in a perpendicular configuration. One pair of motors was used for altitude control, and the other pair was used to control horizontal movement by functioning as a differential drive. Note that this scheme was hindered by the fact that any change in heading led to a corresponding increase in forward velocity. We also tried reversing using the forward motors, but this caused the motors to fail within a few test runs. To achieve precise control over yaw rotation, we attached two additional DC motors to the tail of the DF-v2 and DF-v3 blimps.

2.3.3. Sensors and Computation Units

The robotic blimp was required to follow the tunnel path and search for artifacts in a dark environment. We therefore required a lightweight camera with depth information operable in low-light conditions. We used the Intel RealSense D435 depth camera as our main sensor. Intel RealSense D435 also provided data for visual odometry to construct maps and localize the robot within the tunnel. A Raspberry Pi 3B with an Adafruit DC Motor Hat was used as the motor controller. A NVIDIA Jetson Nano was the main computing unit responsible for perception and

other high-level tasks. The robotic blimp was also equipped with an infrared sensor to measure the altitude of the blimp above the floor.

In the Tunnel Circuit, the robotic blimp traversed the main passage for roughly 50 m before becoming trapped in a branch segment. These experimental runs in the tunnel revealed a number of drawbacks to the four-motor configuration, including the inability to recover after becoming trapped. In preparing for the Urban Circuit, we prepared a general indoor environment on campus, including corridors and two intersections in an H pattern.

The DF-v1 robotic blimp operated in autonomous mode for 47 minutes during which it travelled approximately 500 m, including a successful autonomous traversal of a stairway. The robotic blimp was trapped eight times at corners and narrow paths, all of which required recovery by a human supervisor. These difficulties can be attributed to the limited FOV of the RGB-D camera. This motivated our design of mmWave radar devices for wide FOV sensing. Note that the UGVs were already equipped with mmWave radar devices; therefore, we briefly considered designing the robotic blimp to work directly with the UGVs.

2.3.4. Motor Design and Power Consumption for Airflow Conditions

Considering weight and power consumption, we initially selected DC motors over brushless motors; however, during the competition runs, in-tunnel airflow overpowered the motors. We therefore replaced the DC motors in the vertical pairs with brushless motors, and added a tilting motor to provide control in either the upward or forward directions. We then compared the DF-v2 and DF-v3 multirotor designs in terms of energy consumption by collecting flight data and logging battery voltage and current levels with the velocity set at 0.3 m/s. The average power consumption of the motors on the DF-v2 and DF-v3 were 15.81 W and 61.54 W, respectively. The power consumption of the DF-v3 was slightly higher than the rolling mode of the hybrid system (Fan et al., 2019), and roughly 4× more efficient than in flying mode. The DF-v2 using 6 DC motors consumed the least power; however, it was suitable only for environments without ambient airflow. DF-v3 used stronger brushless motors to operate under moderate airflow, resulting in greater power consumption. Operating sensors and running algorithms, i.e., ORB-SLAMv2 (Mur-Artal and Tardós, 2017) and cross-modal contrastive learning for representation (CM-CLR) (Huang et al., 2021) also consumed considerable power.

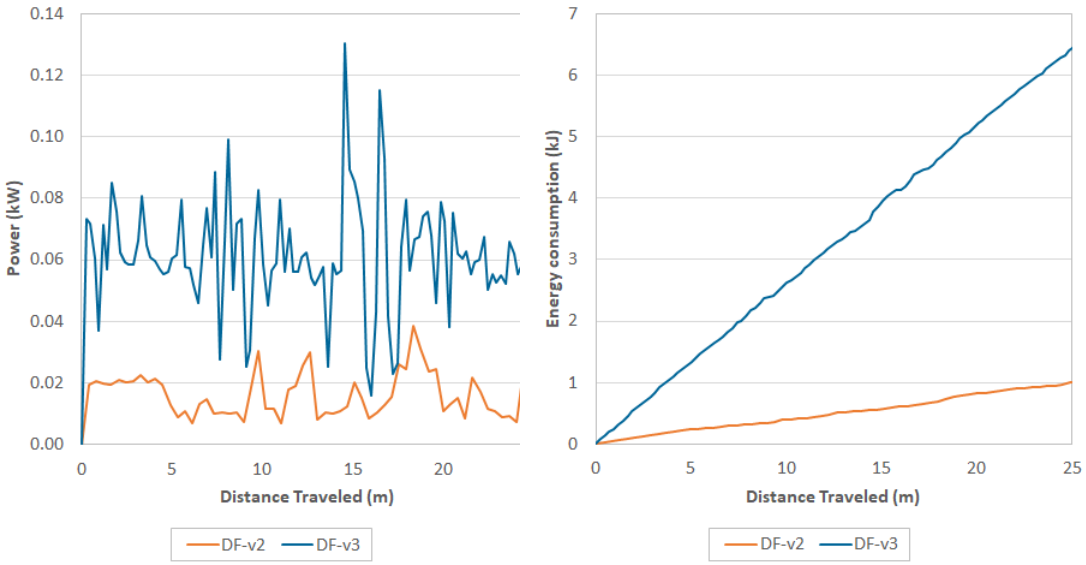


Figure 7. Comparison of power consumption by motors in proposed Duckiefloat.

Table 3. Comparisions of average power consumption (Watts).

Vehicle / Mobility Mode	Hovering	Flying	Rolling	Velocity
Rollocopter (Fan et al., 2019)	-	971.9 (66.3)	194.5 (17.6)	0.3 m/s
DF-v2		**121.04**	-	
Motors	2.75 (0.28)	15.81 (7.52)	-	0.41 m/s
D435 + ORB-SLAMv2 on TX2		105.23 (9.43)	-	
DF-v3		**229.76**	-	
Motors	2.32 (0.29)	61.54 (18.07)	-	0.24 m/s
4× mmWave + CM-CLR on TX2		168.25 (10.00)	-	

Table 4. Field test results of robotic blimps.

Env.	Vehicle	RC or Autonomous	Avg. Speed (m/s)	Avg. Travel Dist. w/o Collision (m)	Note
Maze 1	DF-v1	Autonomous	0.32	5.19	4 turns, 27 m
Maze 1	DF-v1	RC	0.37	6.75	4 turns, 27 m
Maze 2	DF-v3	RC	0.54	**10**	17 turns, 100 m

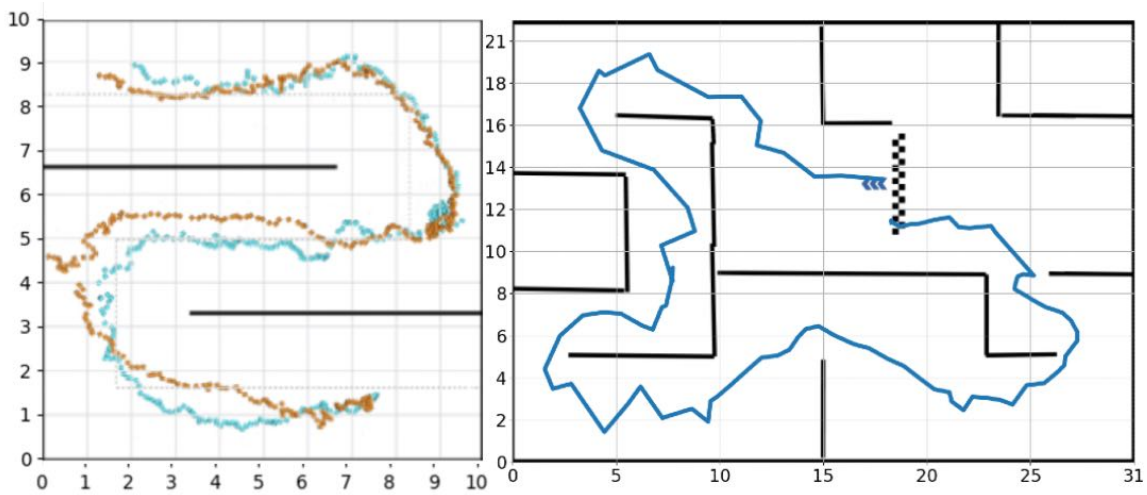


Figure 8. Trajectories of Duckiefloat in field tests through mazes: (left) S-shaped testing track setup showing trajectories of Duckiefloat version DF-v1 in autonomous mode (cyan) and RC mode (brown); (right) Trajectories of Duckiefloat version DF-v3 in RC mode in a longer (100 m) more complex maze.

2.3.5. Mobility Tests in Controlled Maze Environments and Field Sites

Two indoor maze environments were set up to quantitatively assess the mobility of the robotic blimp through tunnels with uniform widths and heights but without intersections. Maze 1 included corridors of roughly 3 m in width passing through four turns over a total length of 27 m. The robotic blimp navigated through the maze in several runs in the RC mode and autonomous mode (described in Section 3.2.4). Maze 2 was located in a basement (23 m × 28 m) following corridors of roughly 3 m in width with cardboard walls measuring 1.78-m high. One lap including 20 turns was approximately 100 m in length, most of which were 90° turns and a few even sharper. Figure 8 illustrates the layouts of Maze 1 and 2, and the trajectories obtained using precalibrated UWB modules.

While navigating through the two maze environments in RC and autonomous modes, the blimp was kept in the middle of the route as much as possible. As shown in Table 4, average speed and average travelling distance without collision were used as mobility performance metrics. Overall, the

Table 5. Field environments and mobility tests of proposed robotic blimp.

Exp. Period	Locations	Dimensions Width x Height	Vehicle	RC/ Auto	Travel Dist <50	Travel Dist >200	Limitation
Tunnel Env.							
Apr. 2019	18-Peak Mt. Dugout	2.3 x 2.0	DF-v1	RC	✓		
Jul., 2019	Houli Tunnel	3.2 x 7	DF-v1	Auto		✓	Diff.-drive Motors
Aug., 2019	**Tunnel Circuit**	3.4 x 2.2	DF-v1	Auto	✓		Airflow
Urban Env.							
Sep., 2019	Indoor Corridor	2.7 x 2.7	DF-v1	Auto		✓	Sensor FOV
Feb. 2020	**Urban Circuit**	varies (min: 0.9 x 2)	DF-v2	Auto	✓		Entrance Dim.
Dec., 2020	Shimen Tunnel	1.9 x 2.1	DF-v3	RC	✓		Airflow
Cave Env.							
Apr., 2021	BH Cave	varies (min: 0.9 x 3)	DF-v3	RC	✓		Narrow Passage

human operator (in RC mode) slightly outperformed the autonomous system in terms of average speed and distance travelled without collision. Nevertheless, the differential drive on the DF-v1 prevented even the human operator from avoiding collisions at corners. The tail motors on the DF-v3 improved heading control, which significantly extended the average distance travelled without collision.

We also evaluated the blimps in real-world environments, as shown in Table 5. Two successful runs were conducted in Houli Tunnel and an indoor corridor. Both of these testing locations were comparable to the Tunnel Circuit, and airflow was negligible or nil.

In the Shimen Tunnel we tested DF-v3 equipped with stronger brushless motors. Under intermittent airflow, the brushless motors were unable to push the DF-v3 forward under the effects of external airflow. This might be due to the fact that the tunnel was so narrow that the envelope nearly filled the tunnel and thus was subjected to the full force of the wind. It appears that this robotic blimp is well-suited to environments with consistent geometries with minimum width and height of > 2.5 m and without excessive airflow.

2.4. Autonomous UGV

Two Husky UGVs (Clearpath Inc.) were equipped with an industrial PC (Advantech MIC-770: Intel i7- 8700 as CPU with a Quadro P2200 as GPU) to process information from sensors and controllers. Three NVIDIA Jetson Xavier kits were connected to RGB-D cameras (Intel RealSense D435) with horizontal FOV of 86 degrees and resolution of 640×480. We also attached an FLIR Boson 320 thermal camera calibrated using the RGB-D cameras. One Velodyne VLP-16 LiDAR device was installed for the Tunnel Event and two LiDAR (one installed vertically and one installed horizontally) for the Urban Event. We also included an MicroStrain 3DM-GX5-45 IMU for SLAM operations. All of the computing units were connected to form a network via the robot operating system (ROS). To enable system operations of up to two hours, we installed an additional 85-Ah lithium battery to supply the power requirements of the sensors, computing units, and two on-board LED lights.

2.5. Deployable Nodes

End-to-end solutions like LoRa (Liando et al., 2019) have drawbacks in NLOS implementations. This fact necessitated the establishment of a mesh network using WiFi and Xbee to provide sufficient coverage. Mesh WiFi provided high-bandwidth images to the human supervisor at the base station in real time, whereas the XBee module provided robust low-bandwidth coverage over a wider range. XBee is well-suited to emergency stop commands, monitoring the state of robots, and reporting artifacts. Robot states were determined using UWB ranging measurements, as it is more accurate than other wireless technologies, such as the Bluetooth, WiFi, and ZigBee. Recent work (Boroson

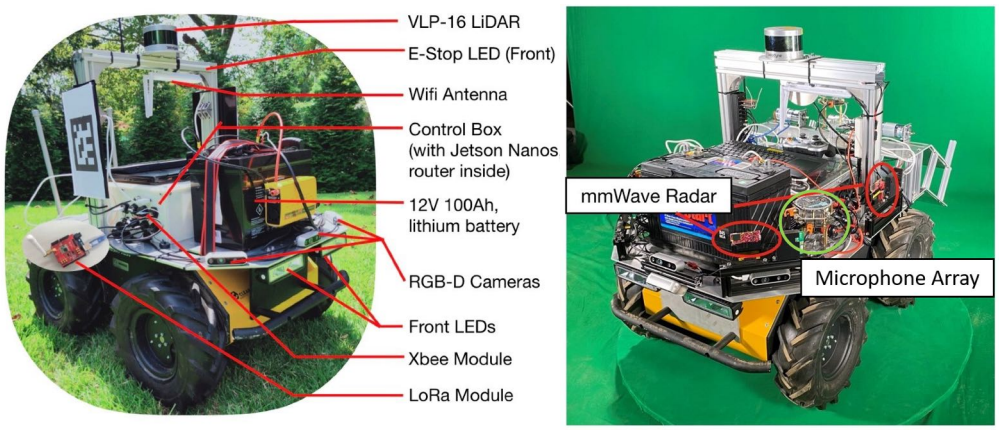


Figure 9. UGV Platforms equipped with Velodyne 3D LiDAR for the Tunnel Circuit and Urban Circuit. (a) LiDAR only and (b) LiDAR and mmWave radar

Table 6. The communication nodes include two deployable *anchor nodes*, and two self-deployed *dynamic nodes*.

Development Stages	**Tunnel**	**Urban**		**Cave/Final**
Deployable Nodes	tunnel-anchor-ball	urban-anchor-brick	urban-bbot	cave-racecar
Anchor/Dynamic	Anchor	Anchor	Dynamic	Dynamic
Shapes	Spherical	Cuboid	Spherical	Racecar
Dimensions (cm)	14	L27×W12×H10	24	L57×W30×H30
Weights (w/ Battery)	613g	1,917g	3,176g	3,365g
operating time	2 h	2 h	1 h	1.5 h
Deployments	Launch/Drop	Drop	Self-deployed	Self-deployed
N. on each UGV	4	6	-	-
Equipped Modules				
UWB	✓	✓	✓	✓
XBee	✓	✓	✓	✓
Wi-Fi AP		✓	✓	✓
Speaker		✓	✓	

et al., 2020) utilized UWB ranging between robots to detect loop-closure in multirobot SLAM scenarios using pose-graph optimization. Subsequently, an algorithm based on sparsely deployed UWB ranging beacons for large-scale SLAM were developed (Funabiki et al., 2020). UWB also allows direct data transmission between two robots as long as they are separated by less than 25 m. We incorporated the three modules within a communication node, thereby allowing the robots to extend their coverage back to the base station simply by deploying nodes along the way.

As shown in Table 6, our deployable communication nodes included static *anchor nodes* and movable *dynamic nodes*. For the Tunnel Circuit, we fabricated the nodes within a spherical shell (diameter = 25 cm) to be launched forward into an unknown area (like a cannonball) or dropped through tubes on either side. Each Husky UGV carried six nodes. In the competition run, the nodes dropped by the Husky were meant to roll toward the wall, as shown in Figure 10 (a); however, many of them ended up in less than ideal locations. For the Urban Circuit, we included a WiFi access point (WiFi AP) within nodes that had be redesigned as a brick. The deployment system on the Husky robot carried six of these nodes, as shown in Figure 10(b).

A limited number of deployable nodes would hinder scaling up to larger environments, such as the > 2-km long competition sites. We therefore also considered dynamic nodes, including a spherical robot [Figure 10(c)] designed to go down stairs in the Urban Circuit and a one-tenth scale race car

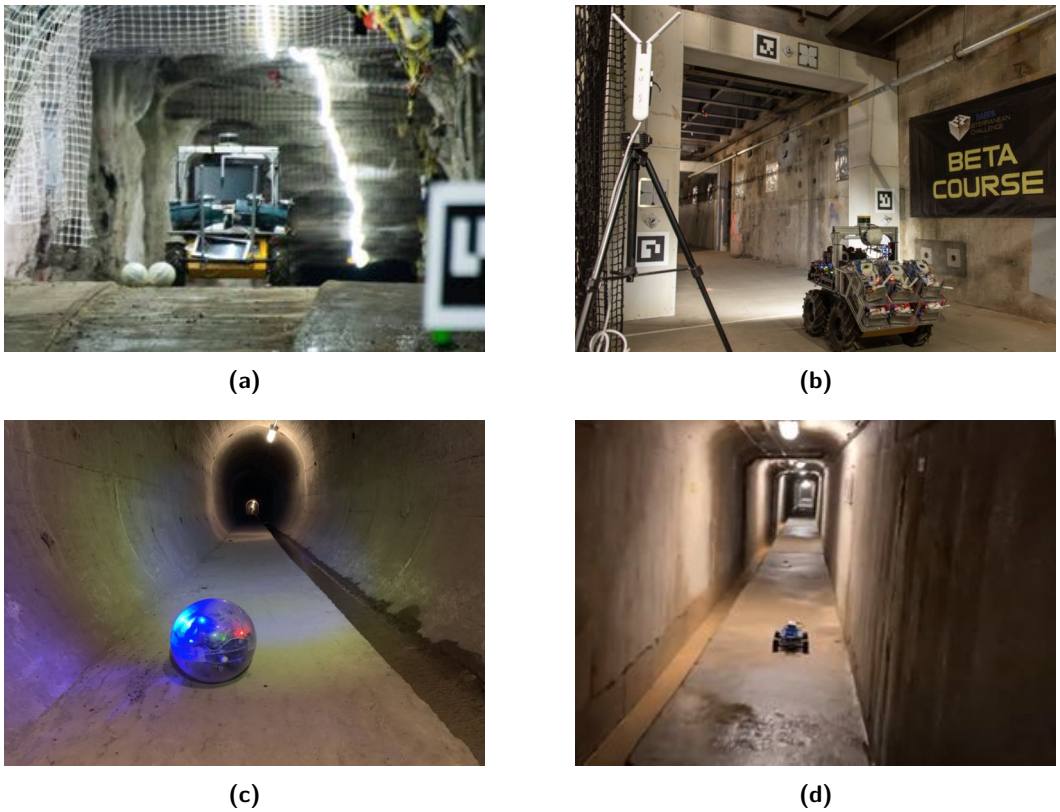


Figure 10. Communication nodes: (a) Spherical anchor node designed to be launched or dropped; (b) Deployment system on Husky robot carrying six brick-shaped anchor nodes; (c) Self-deployed spherical robot as movable dynamic node; and (d) Rapid self-deployed race car robot functioning as dynamic (movable) node.

to cover large areas with greater speed and maneuverability than those of the Husky and Jackal [Figure 10(d)]. Essentially, the movable dynamic nodes were meant to serve as "shuttles" for the physical transfer of data packets between nodes separated by a wide distance.

3. Perception and Autonomy

3.1. Artifact Search

3.1.1. Training Dataset

The Tunnel Circuit included a survivor, a cell phone, a backpack, a drill, and an extinguisher as artifacts to be located. We placed the artifacts in various places, including stairs, interior rooms, corridors, and a basement. RGB-D images were recorded using Intel RealSense D435 mounted on a handheld device or a Husky robot. The images of artifacts were affected by variations in illumination as well as the reflectivity of the object, the viewing angle, and distances to subject. The artifacts were labeled manually using the LabelMe tool (Wada, 2016; Russell et al., 2008). The data collection procedure for the Urban Circuit was similar to that of the Tunnel Circuit, with the exception that a FLIR Boson 320 thermal camera was mounted on the handheld device. The four visible artifacts were the survivor, cell phone, backpack, and vent. The total number of images and the corresponding paired annotations are presented in Table 7.

3.1.2. Semantic Segmentation Algorithms

In the Tunnel Circuit, we used fully convolutional networks (FCNs) (Long et al., 2015) within a perception pipeline to estimate the pixel-wise segmentation of artifacts. The resolution of input RGB

Table 7. Training datasets and models.

Datasets	Tunnel	Urban
N of Images/Anno.	12 664	4071
Artifacts		
Survivor	1545	1151
Cellphone	1435	918
Backpack	3348	910
Drill	3256	
Fire Ext.	3333	
Vent		1092
Models		
tunnel-fcn	✓	
urban-fcn		✓
urban-mask-rcnn		✓
urban-yolact		✓

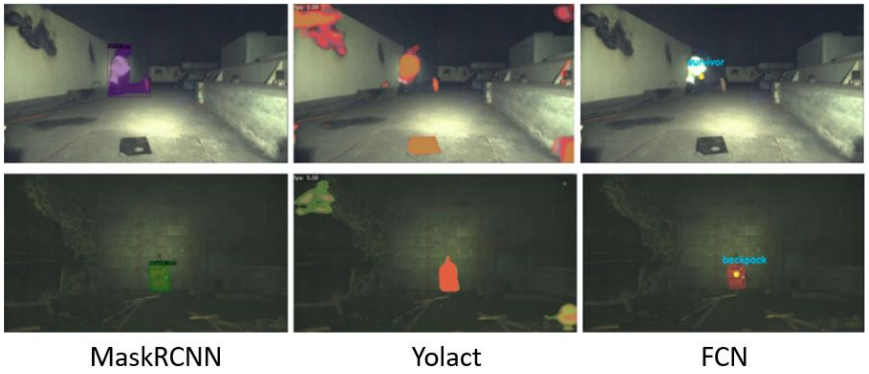


Figure 11. Samples of segmentation results from the SubT-Artifacts-6k Dataset. The frames per second (FPS) of running Mask-RCNN, Yolact, and FCN on each frame of resolution 544 × 1024 using a NVidia Jetson Xavier embedded computer are 1.0, 2.4, and 1.9, respectively.

images was maintained at a low level (480 × 640), considering the limited onboard computation. We found that the vanilla FCN performed poorly in dark environments, and the prediction masks tended to break down. We attempted a patch generative adversarial networks, but this did not improve prediction accuracy. For the Urban Circuit, we switched to Mask-RCNN to enhance accuracy (He et al., 2018) and Yolact to enhance efficiency (Bolya et al., 2019). As shown in Table 7 and Figure 11, we trained two model using the training datasets obtained using Mask-RCNN and Yolact.

3.1.3. SubT-Artifacts-6k Dataset: A Test Dataset using DARPA-released Logs

We manually retrieved image frames containing artifacts from the DARPA-released ROS bag files for labeling as a test dataset. We collected approximately 6,000 image frames. Table 8 lists the average precision (AP, based on the COCO Dataset protocol (Lin et al., 2014)) of Mask-RCNN and Yolact when applied to four classes of artifact in six different logs. Yolact was considerably faster than Mask-RCNN (Bolya et al., 2019), which operated 4 frame per second (FPS) and 1.7 FPS on the hardware of NVidia Jetson Xavier embedded computer, respectively). However, the AP predicted by Yolact was lower than that predicted by Mask-RCNN. Specifically, Mask-RCNN outperformed Yolact in detecting the survivor, backpack, and vent artifacts. The Yolact model was more sensitive to noise (see Figure 11) in the subterranean environments. The performance of Yolact was close to that of Mask-RCNN only in detecting the cell phones, which were largely disregarded by Mask-RCNN.

Table 8. Evaluations of the SubT-Artifacts-6k Datasets extracted from six DARPA-released ROS bag files. Average Precision (AP) was reported. Mask-RCNN outperformed Yolact in 3 artifacts (survivor, backpack, and vent).

Circuit	Tunnel	Tunnel	Tunnel	Urban	Urban	Urban	Summary
ROS Bag File	sr_B_route1	sr_B_route2	ex_B_route1	a_lvl_1	a_lvl_2	b_lvl_2	
No. of Artifacts	10	9	12	7	5	10	
Survivor (N)	4	1	2	3	1	3	
Frames w/ Artifact	168	323	177	102	248	91	
Mask RCNN	0.291	0.199	0.823	0.562	0.806	0.724	**0.566**
Yolact	0.083	0	0.107	0.032	0.004	0.004	0.004
Cell phone (N)	1	1	2	1	-	2	
Frames w/ Artifact	66	84	37	200	-	95	
Mask RCNN	0.984	0.839	0	0	-	0.518	0.468
Yolact	0.924	0.678	0.194	0.025	-	0.582	0.480
Backpack (N)	3	4	1	1	3	3	
Frames w/ Artifact	97	113	106	26	468	65	
Mask RCNN	0.81	0.99	0.836	0	0.899	0.676	**0.589**
Yolact	0.58	0.292	0.415	0	0.738	0.097	0.337
Vent (N)	-	-	-	2	1	2	
Frames w/ Artifact	-	-	-	78	65	136	
Mask RCNN	-	-	-	0.735	0.954	0.621	**0.770**
Yolact	-	-	-	0.245	0.43	0.217	0.297

3.2. Autonomous Navigation

3.2.1. Sim-to-real Deep Reinforcement Learning

Researchers have long sought to develop high-quality simulation environments for learning-based navigation. The goal has been to provide high-dimensional input (raw images) rather than low-dimensional inputs. (Sadeghi and Levine, 2016) used simulations exclusively in the training of policies for navigation within virtual environments built using the 3D program Blender, which included rendered images with randomize textures and lighting to create a set of visually diverse scenes. A Q-function model was used to predict robot actions based on camera observations. A realistic 3D simulation framework (AI2-THOR) was developed with the Unity 3D physics engine (Zhu et al., 2017). A target-driven visual navigation model was then trained using high-dimensionality image inputs to provide end-to-end prediction from pixel information to actions. Another work (Jaritz et al., 2018) took advantage of the game engine World Rally Championship 6 , using A3C (Mnih et al., 2016) to learn car control in a stochastic, realistic environment in the self-supervised mode. The agent took 84×84 front view images and the speed as inputs, while the gas, brake, and handbrake were used as outputs. The Habitat framework (Savva et al., 2019) provides a photorealisitic environment Matterport3D (Chang et al., 2017) and Gibson (Xia et al., 2018). Point-goal navigation (Anderson et al., 2018) was carried out based on a deep reinforcement learning (DRL) model trained using proximal policy optimization (PPO) (Schulman et al., 2017). Habitat used "embodied" AI to enable actions in environments, rather than in media. Cross-dataset generalization experiments revealed that the depth sensor was superior to RGB and RGB-D sensors in terms of generalizing across datasets.

Researchers have recently begun training deep networks using pregenerated occupancy maps, motion commands, or mapless end-to-end approaches (covering range inputs to actions) for UGVs. A planner for search and rescue exploration was trained using a 64×64 2D local occupancy grid as an input for a neural network (Niroui et al., 2019). They modified 2D Stage simulation (usually used for multirobot problems) to train the A3C network for outputs of goal frontier actions. A data-driven planner to learn motion commands from local geometry obtained using a 2D laser range finder was proposed (Pfeiffer et al., 2017). Their convolutional neural network (CNN) processed 1080

Table 9. Overview of the deep RL developments using Virtual-Subt gazebo simulator and several algorithms, including DDPG (Lillicrap et al., 2016), RDPG (Heess et al., 2015), and D4PG (Barth-Maron et al., 2018), to optimize rewards to navigate in long tunnel, cave or goal navigation.

Models	Range Inputs	Additional Inputs	Network Backbone	Reward Settings	Note
urban-rdpg-fw-v1	1×241		RDPG	Forward	Competition Runs
urban-rdpg-fw-v2	1×241	Odom 1×2	RDPG	Forward	In (Huang et al., 2021)
urban-ddpg-fw	1×241	Odom 1×2	DDPG	Forward	In (Huang et al., 2021)
cave-ddpg-goal	4×241	Goal 10×3	DDPG	Goal	
cave-d4pg-goal	4×241	Goal 10×3	D4PG	Goal	

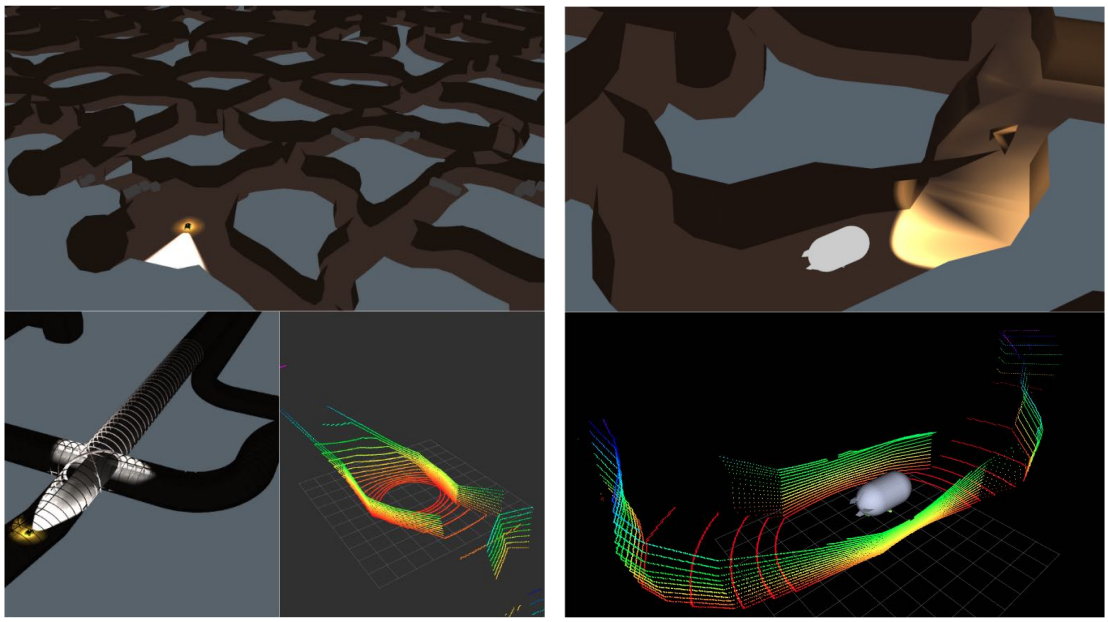
dimensions of laser inputs via convolutional layers with two residual building blocks. The fully connected (FC) layers then combined the extracted features and the target position. Their model proved highly capable of avoiding obstacles on the road (even unseen objects) to reach the final destination. A modified neural network was subsequently developed to down sample 1080 LiDAR measurement into 36 values via minpooling (Pfeiffer et al., 2018).

Unlike the model proposed in previous work (Pfeiffer et al., 2017), the model in the current study was simplified to include only FC layers in order to avoid over fitting. It has been shown that only a 10-dimensional range finds results to decrease the difference between virtual and real environments (Tai et al., 2017). Asynchronous deep deterministic policy gradient (Lillicrap et al., 2016; Gu et al., 2017) was then used to train the actor and critic models in a V-REP simulator. The multirobot collision avoidance problem was tackled in a decentralized scenario (Fan et al., 2020), in which the observation space was obtained from the last three consecutive frames with 512 range values acquired using a 2D laser range finder, resulting in 1536 dimensions. The actions were performed in a continuous space comprising translational and rotational velocities. PPO was used to train the model to enable the direct mapping of raw laser inputs to control outputs from the stage. They determined that a generalized policy for robots can be used in traversing simulated environments as well as real-world human-crowded environments. The use of an DRL agent navigating through an indoor social environment was proposed (Everett et al., 2021). The algorithm determined the optimal path using a value network trained with RL, using the position and velocity of the robot and surrounding neighbors (e.g., pedestrians) as inputs. Instead of predicting an optimal path, the RL agent selected an optimal control command with respect to the state.

Recent work (Kahn et al., 2020) used images as inputs to train a model for path prediction. A planner was used to help the robot traverse through tall grass and reach a goal deemed as untraversable, using classic SLAM in conjunction with planning. A data-efficient end-to-end learning method for goal-conditioned visual navigation was proposed (Manderson et al., 2020). They applied their system to underwater autonomous vehicles navigating through a coral-rich area to collect relevant data while avoiding collisions. The conditional network allowed additional goal information inputs to guide the agent robot to desired destinations.

3.2.2. Navigation with DRL

Our robot was designed to navigate through unknown, static environments that are partially filled with smoke. We modeled the robot control problem as a partially observable Markov decision process (POMDP), which is defined as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma\}$, where $\mathcal{S}$ is the state of the environment, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is the transition function, $p(s_{t+1}|s_t, a_t)$ with initial state $p_0(s_0)$, $\mathcal{R}$ is the reward function $r(s_t, a_t)$, and γ is a discount factor. The state received by the agent is used with a control policy to generate an action. The transition function then leads the agent to the next state where it receives a reward. In POMDP, the agent cannot directly observe the full state s_t; i.e., it can only receive the observed underlying state $p(o_t|s_t)$. As shown in Table 9, we used several RL algorithms to train the agent, and trained two policies for the navigation problem. The first one was designed to maximize exploration rewards and the second one was designed to navigate to a specified goal.



(a) UGV in Virtual-Subt gazebo simulator. **(b)** Duckiefloat in Virtual-Subt gazebo simulator.

Figure 12. (a) UGV training control policies in Gazebo cave and tunnel simulation; (b) Duckiefloat model integrated within simulator to train and evaluate control policies

For the Urban Circuit, we used the Recurrent Deterministic Policy Gradient (RDPG) (Heess et al., 2015) algorithm to train an agent navigating long corridors, tunnels, and caves. The agent was optimized to control the robot through access to history $h_t = (o_1, a_1, o_2, a_2, \ldots, a_{t-1}, o_t)$, whereas the goal of the agent was to set a deterministic policy $\pi(h_t) \rightarrow \mathbb{R}^2$ capable of mapping the observation history to linear angular actions and maximizing the expected discounted reward. The objective function of the agent was defined as follows:

$$J = \mathbb{E}_\tau \left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \right],$$

where $\tau = (s_1, o_1, a_1, s_2, \ldots)$ refer to trajectories. Further details related to reward, action, observation, and training environment can be found in our previous paper (Huang et al., 2021). In the Urban Circuit competition, we used the control policy trained via the virtual-subt gazebo simulator with the RDPG algorithm.

To generalize the proposed method for Duckiefloat, we also integrated the 3D model of Duckiefloat and an open source blimp model (ootang2018, 2020) within the Virtual-Subt gazebo simulator, as shown in Figure 12(b). However the complex dynamics of Duckiefloat prevented the RL agent from learning a robust control policy for navigation. Further works need to investigate the robot dynamics affects on the RL algorithms.

We also trained a control policy capable of navigating to a designated goal. To train a generalized model, we used the Virtual-SubT gazebo cave environment with tunnels of diverse size and shape as well as ramps, as shown in Figure 12. We integrated the Gazebo simulator with the API of OpenAI Gym to process observation data from the agent, calculate rewards, set actions, and reset the agent. The important issues and settings are elaborated in the following:

- Observations: The field-of-view of LiDAR points covered 240° (from 120° to −120°) with resolution of 1°. We concatenated four consecutive frames of range data and 10 frames indicating the relative position of the goal within a single observation space.

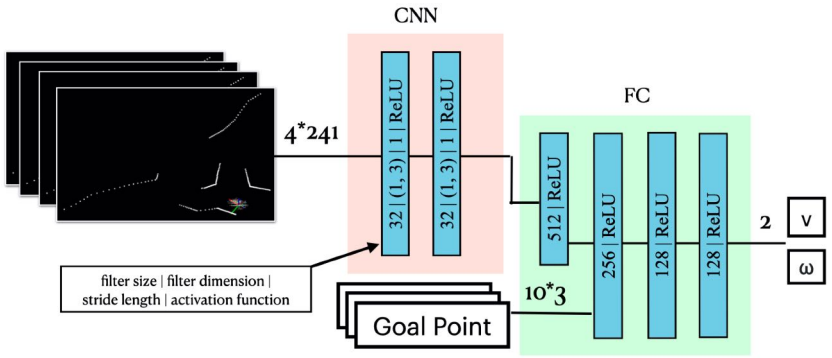


Figure 13. Policy network for goal point navigation showing the CNN feature extraction stage and the fully connected stage. We stacked four consective frames of point clouds and 10 goal points as inputs to provide the agent a sense of past trajectories. The subsequent CNN layers were used to extract essential features, after which goal points were added to the fully-connected layers to compute the optimal command.

- Actions: Actions were designated as linear and angular. Linear actions were limited to $[0, 1]$ and angular actions were limited to $[-1, 1]$.
- Reward: To ensure the robot avoided collisions and reached the goal, we established the following dense reward function: (1) relative position toward the goal, (2) reaching the goal, and (3) penalty for collision.

$$r\_goal = \begin{cases} 0.2 & if \quad Toward\ the\ goal \\ -0.2 & else, \end{cases}$$

$$r\_reach = \begin{cases} 100 & if \quad Reach\ the\ goal \\ 0 & else, \end{cases}$$

$$r\_collision = \begin{cases} -10 & if \quad Collision \\ 0 & else, \end{cases}$$

$$r = r\_goal + r\_reach + r\_collision.$$

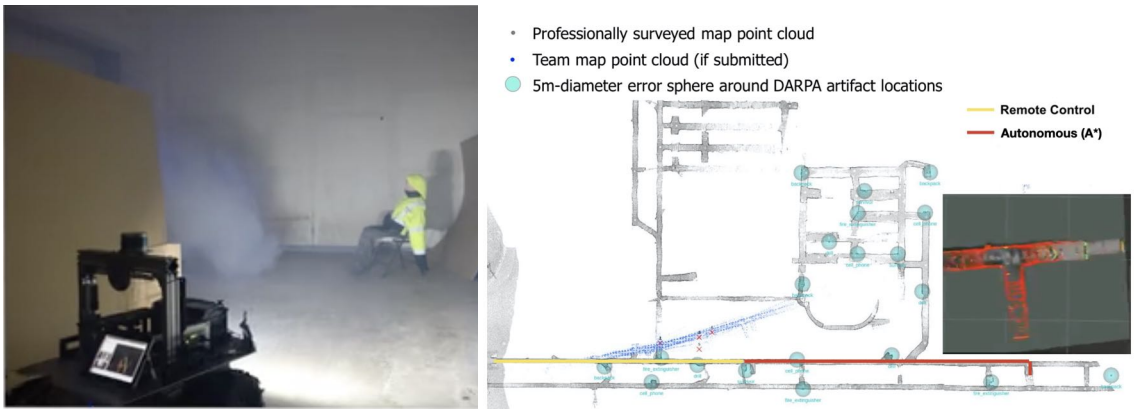
Goal point navigation was first trained using the Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al., 2016) using an actor-critic structure to train a deterministic policy and a Q-value estimator. The objective of the actor network was to maximize the Q value. To further improve navigation performance, we also trained a model using the Distributed Distributional Deep Deterministic Policy Gradient (D4PG) algorithm (Barth-Maron et al., 2018) using a distributional critic network, in which the output of the critic network becomes a probability distribution of Q values to enable robust value estimation.

For both algorithms, the actor networks included two 1D convolutional layers for the extraction of features from LiDAR points. The flattened outputs of the convolutional layers were then concatenated with the goal positions from the previous 10 frames for the subsequent fully-connected layers. A visualization of the network is presented in Figure 13.

Figure 15 presents the trajectory of the robot in the Alpha Course. According to a wheel odometer, the robot traversed 277.53 m before becoming trapped on a ledge, due to the limited FOV of the LiDAR sensor. In the Beta Course, the trained policy was too conservative to allow passage through narrow passages encountered at the beginning, which resulted in the robot circling around in an open area without exploring.

3.2.3. Navigation through Smoke using mmWave Radar

The Tunnel Circuit presented challenging environmental conditions, including long, unexplored tunnels filled with smoke capable of interfering with cameras and LiDAR devices. To overcome these



(a) Navigation through smoke **(b)** Wall following

Figure 14. (a) Navigation policy using mmWave sensor and CM-CLR to navigate through smoke-filled environments in the Tunnel Circuit (Huang et al., 2021); (b) Wall following strategy using the A* algorithm to navigate through the Tunnel Circuit.

difficulties, we employed lightweight inexpensive mmWave radar. We observed a notable reduction in performance when using mmWave radar alone, due to low spatial resolution and the effects of noise. Thus, we employed the cross-modal contrastive learning for representation (CM-CLR) method to maximize agreement between radar data and LiDAR data in the training stage. This was implemented in a reinforcement learning framework, and compared with cross-modal generative reconstruction and other baseline approaches. Our proposed end-to-end DRL policy with contrastive learning proved highly successful in navigating through a smoke-filled maze. The proposed scheme outperformed generative reconstruction methods, which tended to generate noisy artifact walls or obstacles. We refer the readers to our recent work (Huang et al., 2021) for details.

3.2.4. Classic Approaches for Navigation as Baselines

We implemented a policy based on a map-localize-plan to follow the tunnel as the baseline for exploration. We built an occupancy grid map by parsing the point cloud gathered from LiDAR inputs. The map was then used for a local planner, sweeping from -90 to 90 degrees with a radius of 2 m. A goal point was then set to the found traversible point via A* search.

We implemented tunnel-following policy as the baseline for blimp exploration. Analysis of the point cloud gathered from the RGB-D camera was used to identify points of interest and project them onto the plane parallel to the ground. We then searched line segments in the image and classified them as right, left, or front wall. The slopes and intercepts of lines related to the various classes of wall were used to determine the state of the blimp. The state of the robot at time t is represented as $x_t = \langle d_t, \phi_t \rangle$,where d_t indicates the lateral distance between the blimp and the center of the tunnel at time t, and ϕ_t is the angle relative to the tunnel axis. A proportional-integral-derivative (PID) controller was used to control the state of the robot with target $d = 0$ and $\phi = 0$, thereby ensuring that the blimp would remain in the middle of the tunnel with its yaw angle parallel to the length of the tunnel. We used another PID controller for altitude control with the goal altitude set at 0.6m.

3.3. Simultaneously Localization and Mapping

Simultaneous localization and mapping (SLAM) has been studied extensively for nearly 30 years [For an overview (Durrant-Whyte and Bailey, 2006)]; here we surveyed SLAM studies for subterranean environments. An earlier work (Huber and Vandapel, 2003) used a 3D LiDAR device mounted on a cart to perform 3D mapping of an underground mine. They used graph optimization with a global consistency measure to detect and avoid erroneous but locally consistent matches. Subsequently

an integrated localization system for robots in underground environments using only IMUs was proposed (Xiong et al., 2009). Recently a benchmark dataset of STIX and Tunnel Circuit applying state-of-the-art algorithms was reported (Rogers et al., 2020). LiDAR-based SLAM algorithms, such as OmniMapper (Trevor et al., 2014) and Cartographer (Nüchter et al., 2017) have produced impressive results in terms of root mean square error over long trajectories, accurately localizing ground truth artifact landmarks within 5 m on a global map. Vision-based methods, such as ORB-SLAM2 (Mur-Artal and Tardós, 2017), do not perform well in dark environments in terms of tracking, due largely to the failure of relocalization to recover the path after tracking is lost.

For the Tunnel Circuit, we used the ORB-SLAM2 (Mur-Artal and Tardós, 2017) algorithm for onboard visual odometry. Consistent with the results in the literature, the algorithm was vulnerable to fast movements (motion blur under low light) and changes in illumination during on-site field tests. We therefore implemented fail-safe mechanisms to detect cases of failure (i.e., loss of visual odometry) and recover the system to the previous state. Recovery after tracking loss involved resuming from the most recent pose. SLAM performance at the base station was evaluated via minimal mapping to represent the surroundings of the robot by taking a slice of the point cloud gathered by the RGB-D camera. We assumed that the system would not have to remain fully autonomous throughout the challenge, as long as we could maintain communications. Vision-based odometry (ORB-SLAM2) with reinitialization proved effective in resuming localization and mapping; however, the precision was insufficient to earn us additional points, due to the fact that the estimated pose was often rotated at the time tracking was lost.

LiDAR-based SLAM has produced promising results; however, parameter tuning can be difficult, and optimal parameters cannot be determined *a priori* when dealing with unknown sites. Due to its incremental nature, LiDAR-based SLAM is ill-suited to long narrow walkways and featureless environments, such as the Beta course in the Urban Circuit. A single erroneous data association could be catastrophic, as it would render all subsequent measurements useless. Team CoStar introduced heterogeneous redundant odometry (HeRO) to derive estimates from multiple odometry algorithms running in parallel (Santamaria-Navarro et al., 2020; Bouman et al., 2020). Resilience is achieved by ensuring redundancy and initiating recovery. Confidence tests are used to identify common failures, such as (1) gaps in state updates, (2) rapid jumps, and (3) divergence in the rate of change in the position/velocity or covariance matrix.

3.4. Overall System Performance

Clearly, localization and mapping accuracy were critical to the overall score. Thus, we quantitatively evaluated the localization and mapping accuracy during competition runs on the Alpha course in the Urban Circuit, as shown in Figure 15. We also analyzed the reasons for SLAM failures following rapid jumps and assessed our recovery strategy.

Localization and mapping accuracy have been evaluated using a variety of metrics (Funabiki et al., 2020). We used the generalized iterative closest point (ICP) (Segal et al., 2009), which was previously implemented in the PCL Library (Rusu and Cousins, 2011), to match frame-by-frame the onboard LiDAR point cloud and the ground truth map provided by DARPA. We used the LiDAR-based LeGO-LOAM (Shan and Englot, 2018), which performed well in the structured environments encountered along the Alpha course. Mean odometry and mapping errors were 1.198 and 0.365, respectively. These results were within the 5-m error range stipulated in the rules. Nonetheless, while operating on bumpy terrain, our UGV experienced two rapid jumps leading to SLAM failure, as indicated by the two peaks (at 925 and 954 seconds) in Figure 16. The rapid jumps greatly increased the mean odometer error (7.904) and mean mapping error (2.513) to beyond the 5-m limit. Thus, we failed to earn points for detecting and identifying a survivor artifact adjacent to the end location.

We also performed offline analysis of confidence testing and the recovery process. We followed the method proposed for HeRO (Santamaria-Navarro et al., 2020; Bouman et al., 2020) to estimate two LeGO-LOAM algorithms running in parallel, where one was in real-time and one had the sensor

Table 10. Mean (μ) and standard deviation (σ) of odometry errors estimated by LeGO-LOAM before and after rapid jumps (R.J.), and after recovery.

	Metrics	(a) Before R.J.	(b) After R.J.	(c) Recovered
Odometry Errors	μ	1.198	7.904	**1.924**
	σ	0.948	11.448	1.361
Mapping Errors	μ	0.365	2.513	**0.479**
	σ	0.199	17.104	0.409

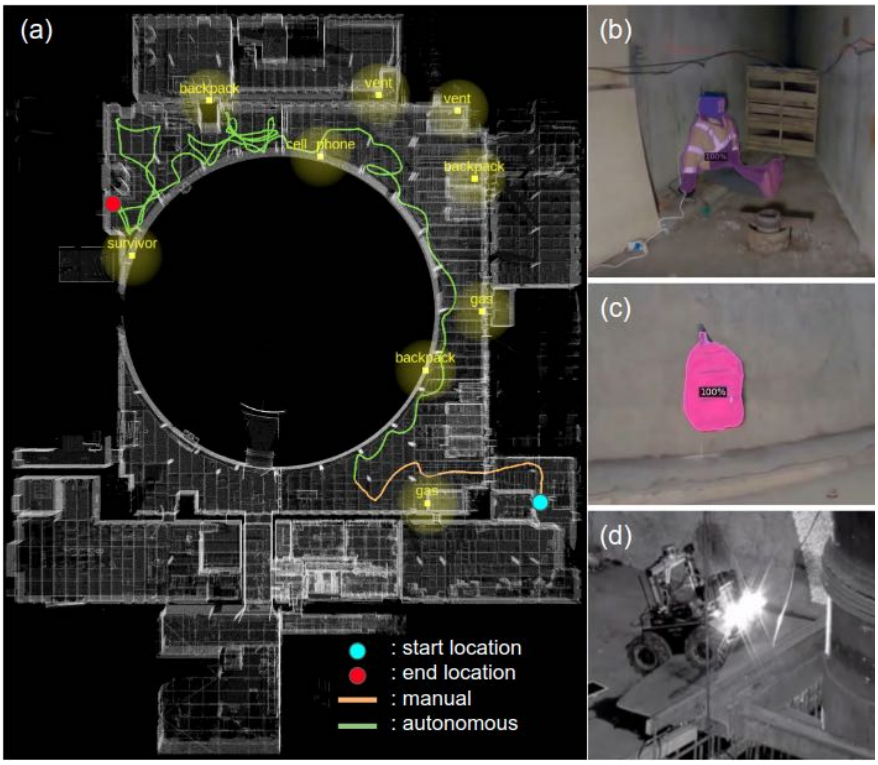


Figure 15. Overall performance in the DARPA Subterranean Challenge Urban Circuit Alpha Run at the Satsop Nuclear Power Plant in Elma, Washington: (a) Ground truth map with robot trajectory. The UGV autonomously traversed 277.53 m (green line) in 23 minutes using the proposed learning-based navigation system. The artifacts and 5-m scoring range are indicated by yellow dots; (b) (c) Artifacts correctly detected by our robot. At the end of the course, the UGV became stuck on a ledge [red dot in (a) and (d)]. The robot trajectory was plotted by matching the onboard LiDAR point cloud to the ground truth map provided by DARPA (white points) using the General Iterative Closet Point (ICP) algorithm for quantitative evaluation.

inputs purposely delayed by 10 seconds. For confidence tests associated with rapid jumps, we set the threshold of linear velocity at 5 m/s, as shown in Figure 16. We also observed other measurements (e.g., angular velocity), which did not provide a clear threshold for rapid jumps. The odometry and mapping errors after recovery were 1.924 and 0.479, respectively. This was sufficient for localization and mapping accuracy within the 5-m error range. Figure 17 compares the ground truth map (white) with the mapping estimates.

The rapid jumps revealed the shortcomings of the proposed navigation scheme using low-dimension inputs for the deep neural network, which were found more generalized from sim-to-real training. Training here was performed in the cave environment of the Virtual SubT Competition, which did not include low obstacles or ledges. The performance was reflected in the fact that

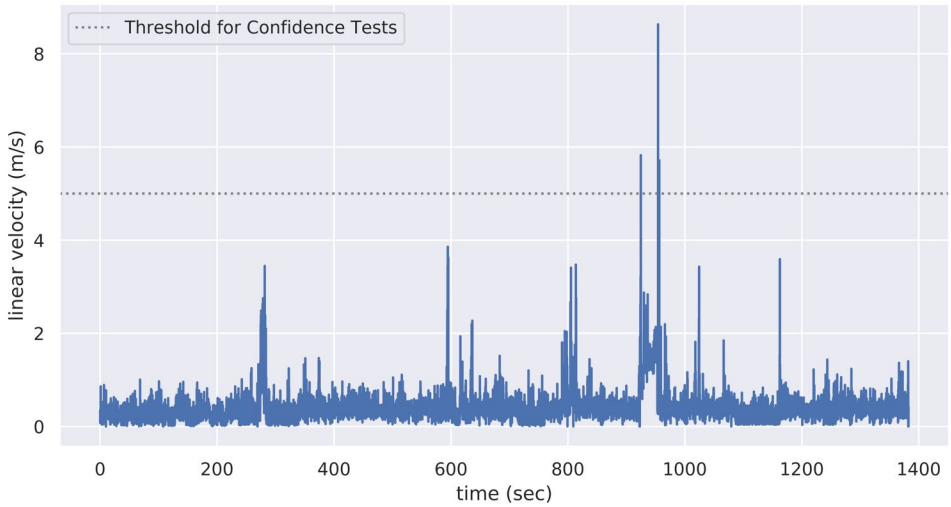
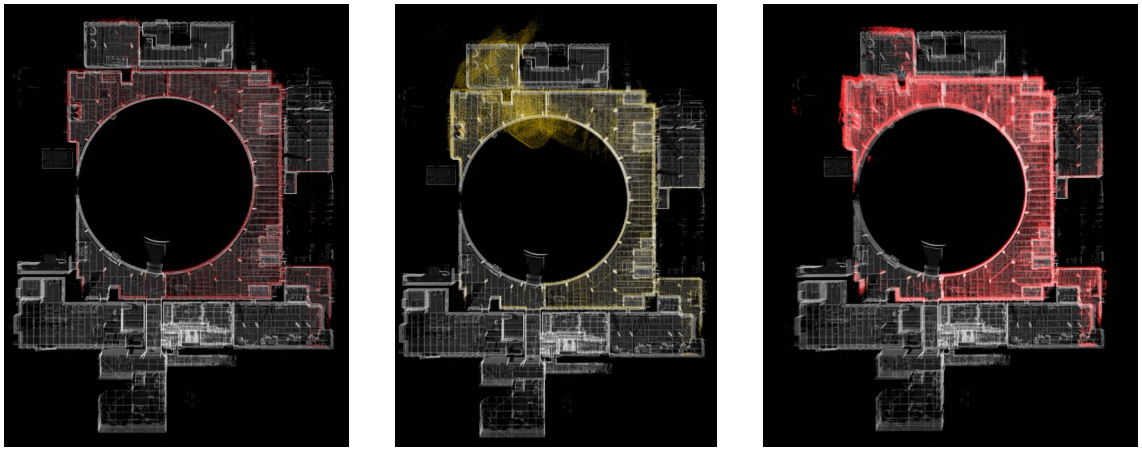


Figure 16. Linear velocities calculated from localization data obtained using LeGO-LOAM. The sudden change in velocity indicates rapid jumps.



(a) **(b)** **(c)**

Figure 17. Comparison of ground truth map (white) and maps generated using the LeGO-LOAM algorithm (colored): (a) Before rapid jump; (b) After rapid jump; (c) Map recovery based on confidence tests and corresponding recovery procedure.

the learning algorithm did not consider low obstacles or ledges, ultimately resulting in the UGV becoming stuck on a ledge [red dot in Figures 15(a) and 15(d)].

We also determined that our learning-based algorithm was too conservative to allow passage through narrow gates. During the competition, we used the urban-rdpg-fw-v1 model trained to avoid obstacles and keep moving as far as possible. The robot trajectory in Figure 15(a) shows that the UGV traversed within 5 m of six artifacts, but earned a point only for the red backpack [Figure 15(c)]. We did not earn any points for artifacts hidden in rooms or for tasks that required the robot to pass through narrow gates. This was sufficient to convince us of the need for multiple-objective DRL training and goal navigation models.

Table 11. Evaluation of goal navigation models passing narrow gates using D4PG (cave-d4pg-goal) and DDPG (cave-ddpg-goal) models. Gate width (GateW), success rate of passing through (Succ.), UGV-Jackal turned around and not passing (Trapped), and the number of times a human supervisor had to intervene to prevent a collision with the gate. The human intervention could be automatically triggered by an array of tactile sensors around the robot.

Models	GateW (cm)	Succ.	Trapped	Intervention
D4PG (cave-d4pg-goal)	140	1.00	0.00	0.00
	110	0.67	0.11	0.22
	100	0.56	0.33	0.11
	90	0.44	0.33	0.22
DDPG (cave-ddpg-goal)	140	0.67	0.00	0.33
	110	0.56	0.00	0.44
	100	0.33	0.00	0.67
	90	0.22	0.00	0.78

In a separate experiment, we quantitatively evaluated the success rate in passing through gates of various widths (140–90 cm). We then evaluated two goal navigation models (cave-d4pg-goal model and cave-ddpg-goal), using a Clearpath Jackal UGV in an indoor corridor. A human supervisor provided a goal point located 10 m from the UGV with a gate in between. The initial position of the robot was randomly assigned and nine trials were conducted for each condition. We measured the number of times the UGV succeeded in passing through the gate (Success), the number of times the UGV turned around without passing the gate (Trapped), and the number of times a human supervisor had to intervene to prevent a collision with the gate (Intervention). The results in Table 11 indicate that the success rate decreased with the width of the gate, regardless of the model used for navigation. The D4PG model outperformed DDPG by enabling the UGV to navigate through the narrowest gate (90 cm) with a success rate of 0.44. Its success can be attributed to the use of N-step returns, and extensions upon the DDPG that a distributional updates combined with the use of multiple distributed workers writing into the same replay buffer (Barth-Maron et al., 2018). Overall, learning-based multiobjective navigation was unable to match conventional methods in the piano moving problem (Schwartz and Sharir, 1983). Nonetheless, the learning approach was useful in some situations in which the map-localize-plan approach failed due to an inability to observe all geometric relationships in a scene. We refer readers to our work on navigating with mmWave radar and DRL under adverse environmental conditions (Huang et al., 2021).

4. Deployable Nodes for Localization and Communication

4.1. SBL-UWB Localization

A number of studies (Song et al., 2019; Zhen and Scherer, 2019) have reported on the challenges of LiDAR-based SLAM in long featureless tunnels, due to geometrically featureless long corridors. Despite recent efforts to leverage UWB ranging through the use of fusion, most systems require the predeployment of UWB modules, which is not feasible for SAR missions.

The short baseline (SBL) method is used in the navigation and localization of autonomous underwater vehicles (AUVs). In SBL, acoustic range sensors on a support vehicle with access to GPS (e.g., a boat) are used to localize the scout vehicle (AUV), as shown in Figure 18(a). In this study, we implemented SBL-UWB to obtain the relative location of a single robot, a deployable node, and/or a heterogeneous team of robots.

We equipped a large Husky robot with LiDAR-based SLAM to function as a support robot to enable localization of scout robots relative to itself using the onboard UWB module. As demonstrated in previous studies (Rogers et al., 2020) LiDAR-based SLAM could be robust in most scenarios except some cases (such as rapid jumps, see Section 3.4), LiDAR-based SLAM is robust in most

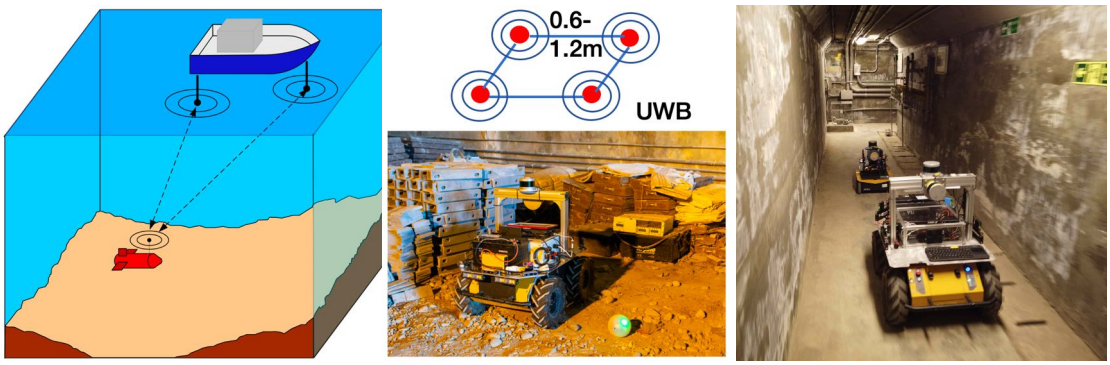


Figure 18. (a) Short baseline (SBL) is used with acoustics for localizing autonomous underwater vehicle (AUV). Image courtesy of Liam Paull. (b) SBL is used with UWB measurements to obtain relative location between a robot and a deployable node. (c) SBL-UWB could be used for a support vehicle (Husky) and a scout vehicle (Jackal).

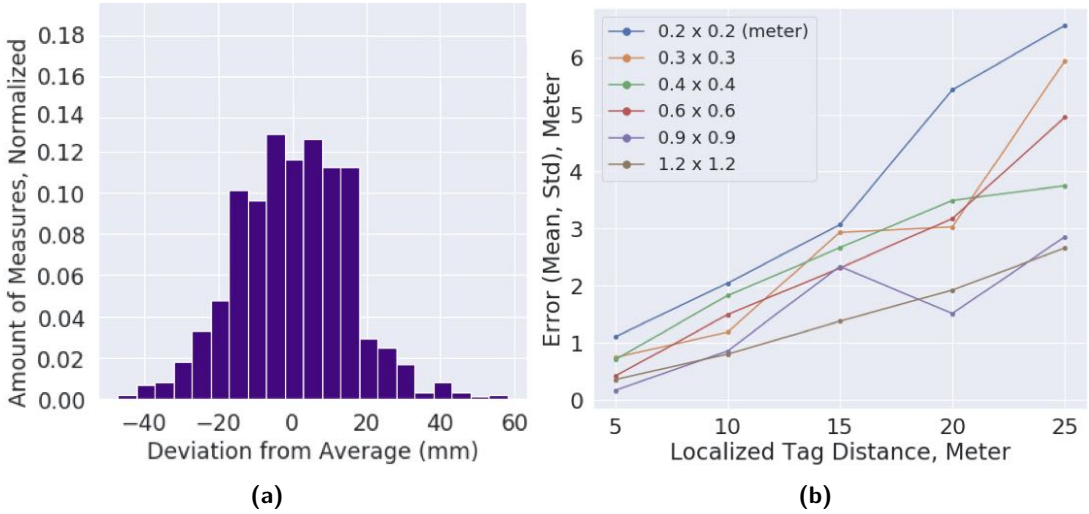


Figure 19. We tested (a) UWB sensor characteristic: range measurements distribution, and (b) the average errors of the SBL-UWB evaluated using various baseline length (from 0.2 to 1.2 m) and ranges (from 5 to 25 m). The error is caused by angular deviations.

scenarios other than sudden bounces (see Section 3.4). Thus, the resilience of the proposed SBL-UWB method to environmental variations (e.g., illumination) means that it could be used as a reference for confidence tests in the assessment of SLAM algorithms.

4.1.1. Sensor Characteristics and Baseline Length

Rather than Wi-Fi or Bluetooth, we selected UWB modules (Pozyx) as range sensors, which use transmission signals over multiple frequency bands (3.1 to 10.6 GHz) with a large bandwidth spread (499.2 to 1331.2 MHz). We first evaluated the range measurement distribution in a basement environment, where 1D LOS ranging was within 40 mm [see Figure 19(a)].

Baseline length largely determines the accuracy of SBL-UWB. Baseline length and measurement accuracy are constrained by the size of the robot; therefore, we performed an experiment to estimate the 2D localization errors of SBL-UWB. Four UWB modules with baseline lengths of 0.2–1.2 m were

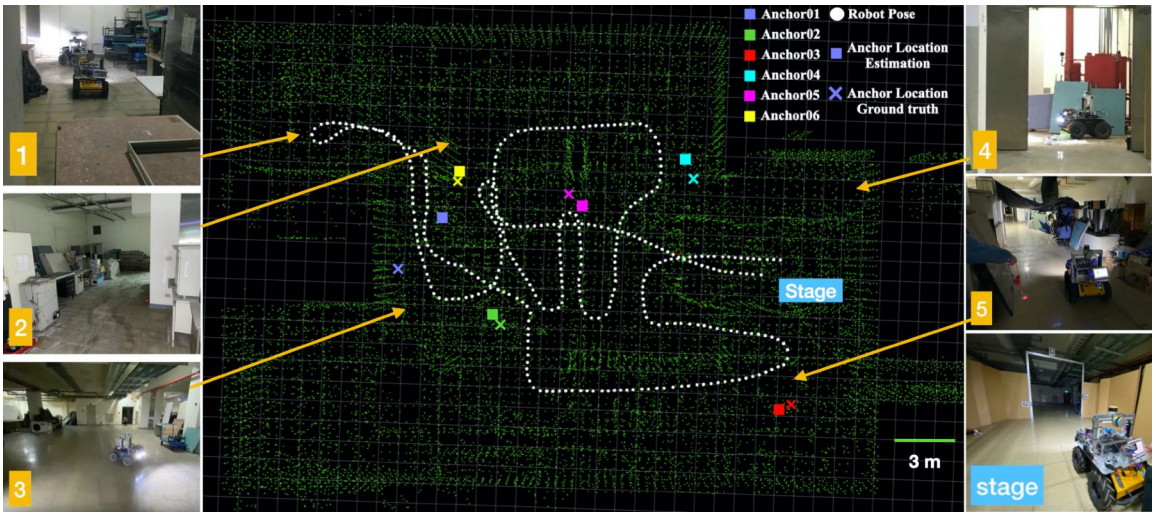


Figure 20. Local deployment of six SBL-UWB nodes in a test environment with the ground truth positions in a map frame designed in accordance with the reference frame defined by DARPA. The nodes were located using SBL-UWB along the trajectory of the robot, as estimated using LeGO-LOAM in the Circuit event. The average error in localizing the six nodes was 1.08m.

mounted on a frame. A tag was then mounted on a robot performing SLAM, while moving away from the frame. Localization errors were measured at a range of 5–25 m with respect to the frame. Multidimensional scaling was applied to the range estimates and range priors from tags on the support vehicle in order to estimate the relative position of the UWB tags. Our results revealed that baseline length was inversely proportional to estimation errors, as shown in Figure 19(b). When the baseline length exceeded 0.9 m, the error was roughly 10%, proportional to distance. The errors were caused by angular deviations.

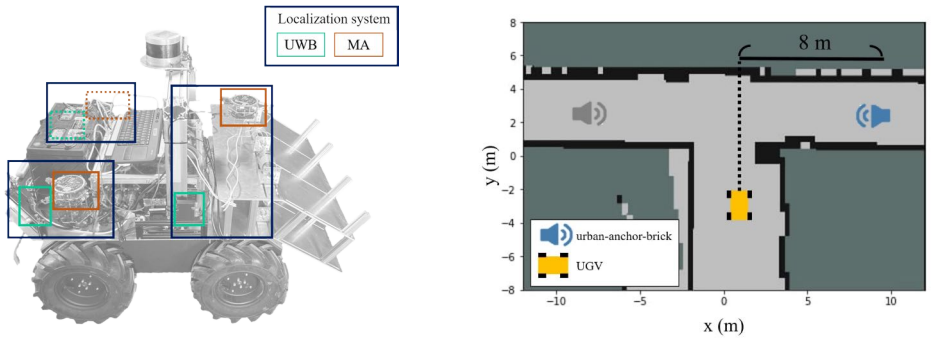
4.1.2. Localization Estimation to Deployable Nodes

Four UWB modules were installed on the support vehicle to estimate the position of the UWB module installed on the scout vehicle. The UWB modules on the support robot were placed at the four vertices of a square measuring 0.6 × 0.6 m. The estimated error based on baseline length was roughly 20%, proportional to distance. Accuracy of this degree may be sufficient to detect large divergences in results from the SLAM algorithm. It is also possible to execute a homing action back to a specific deployable node for recalibration. In the current study, we limited our analysis to a 2D plane; however, this approach could be extended to 3D if more UWB modules were installed.

We then evaluated the SBL-UWB setup on the Husky UGV. The testing environment was a basement floor, which included several pathways with multiple turns along narrow corridors cluttered with partitions, tables, chairs, furniture. Six deployable nodes (urban-anchor-brick) were also placed in the basement environment. The UGVs were initialized in the staging area and localized at the nodes using SBL-UWB, while navigating through the environment. Figure 20 presents the estimated position and ground truth position obtained from LiDAR SLAM. The average error of the six nodes was 1.08 m.

4.1.3. Sound Source Localization (SSL)

The deployable nodes included a speaker source, to be localized by our UGV. Seeed's ReSpeaker Core v2.0 with eight-channel ADC is utlized as a six-circular microphone array and two loopbacks. Three microphone arrays were installed as shown in Figure 21. The captured signals from the microphone array $\mathbf{x}$ were used to find the potential sources. Multiple Signal Classification (MUSIC) based methods have been suggested (Grondin and Michaud, 2019).



(a) Placement of localization system on UGV **(b)** Setup of sound localization for navigation

Figure 21. (a) Sound source localization using three sets of microphone arrays and UWB modules. The settings were designed to avoid the self-noise in between the sound source and the robot. (b) We found the SSL accuracy 7.78° did not meet our requirement for landmark loop closure, but we found it useful as a homing mechanism for a robot to return to a node. The UGV was tasked with approaching the goal at each one of sides without colliding using the localization system for guidance.

The implementation of SSL (Bai et al., 2013) was achieved using the MUSIC-based algorithm in the frequency-domain with thetime-harmonic dependence, $exp(j\omega t)$, where $j = \sqrt{-1}$, ω denoted the angular frequency, and t denoted the time variable. We considered N point sources and M sensors. For a wide-band audio signal in the audible range, the pressure field generated by a point source was expressed as

$$p(\mathbf{r_m}) = \frac{s_n(\omega)e^{jkr_{mn}}}{r_{mn}}, m = 1, ..., M; n = 1, ..., N, \tag{1}$$

where $r_{mn} = \|\mathbf{r_m} - \mathbf{r_n}\|$ with $\mathbf{r_m}$ and $\mathbf{r_n}$ being the position vectors of the m^{th} sensor and the n^{th} source. The symbol $s_n(\omega)$ denotes the Fourier transform of the n^{th} source amplitude and $k = \omega/c$ being the wave number with c being the speed of sound. The following frequency-domain array signal model can be established:

$$\mathbf{x} = \mathbf{Gs} + \mathbf{n}, \tag{2}$$

where $\mathbf{x} = [p(\mathbf{r_1}) \cdots p(\mathbf{r_M})]^T \in C^M$ is the array data vector, $\mathbf{s} = [s_1(\omega) \cdots s_N(\omega)]^T \in C^N$ is the source amplitude vector, $[\mathbf{G}]_{nm} = e^{j\,k\,r_{mn}}/r_{mn}$, $m = 1, ..., M$; $n = 1, ..., N$ is the steering matrix based on the point-source model, and $\mathbf{n}$ is the uncorrelated additive noise vector. The MUSIC algorithm relies on the separation of two orthogonal subspaces, the signal subspace and the noise subspace, constructed using the eigenvalue decomposition (Schmidt, 1986) of the spatial correlation matrix:

$$\mathbf{R_{xx}} = \mathbf{U\Lambda U}^H = \sum_{i=1}^{M} \lambda_i \mathbf{e}_i \mathbf{e}_i{}^H, \tag{3}$$

where $\mathbf{U}$ is a unitary matrix comprised of orthonormal eigenvectors $\mathbf{e}_i^{'}s$, $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_M$ as its diagonal entries, the superscript "H" denotes Hermitian transpose. N indicates the number of independent sources determined by the effective rank of the spatial correlation matrix. Note that $span\{\mathbf{e}_1, \cdots, \mathbf{e}_N\}$ is referred to as the "signal subspace" and $span\{\mathbf{e}_{N+1}, \cdots, \mathbf{e}_M\}$ is referred to as the "noise subspace". In light of the orthogonality of these two subspaces, a projection matrix onto the noise subspace can be constructed as follows:

$$\mathbf{P}_N = \sum_{i=1}^{M} \mathbf{e}_i \mathbf{e}_i{}^H = \mathbf{I} - \mathbf{U}_s \mathbf{U}_s{}^H, \tag{4}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{U}_s = [\mathbf{e}_1, \cdots, \mathbf{e}_N]$. The sources can be located by finding the peaks in the MUSIC pseudospectrum:

$$S_{MUSIC}(\mathbf{r}) = \frac{1}{\mathbf{a}^H(\mathbf{r})\mathbf{P}_N\mathbf{a}(\mathbf{r})}. \tag{5}$$

The vector, located at $\mathbf{a}(\mathbf{r})$ represents the steering vector corresponding to point source, $\mathbf{r}$. Note that the steering vector is based on the point source model, which necessitates a grid search.

The MUSIC-based algorithm was integrated within the UGV via microphone arrays. We disregarded elevation angle θ in order to facilitate real-time computation, and the direction of arrival (DOA) estimates were obtained using azimuth angle resolution of only 5° ($\theta = 0°$). The precision of the sound localization algorithm was evaluated in an indoor rectangular area measuring 7.4 m × 7.6 m. With one sound source placed by a distance of 1 to 5 m at the azimuth angles, 0° or 90° by turns, the average errors in estimating the angle of the sound source was 7.78°. We neglected the elevation angle θ for real-time computation considerations, and the DOA estimation was performed only on the resolution of azimuth angle by 5°. Here, SSL accuracy did not meet our requirement for landmark loop closure, but we found it useful as a homing mechanism for a robot to return to a node. We conducted an evaluation of the robot tasked with turning into the path where the node was located. The rate of success in recognizing the correct direction was 0.96 (52/54). Both of the errors involved nodes on the left side of the T junction, which may be due to some other interferences being detected in the lab environments.

4.2. Subterranean Radio Propagation Analysis

4.2.1. Goals and Metrics

Communications must be maintained at all times among heterogeneous robots, anchor nodes, and human supervisors. Without a clear understanding of radio propagation characteristics in SubT environments, there is no way to determine the optimal deployment strategy, and particularly in locations with a complex geometric structure. Thus, we collected radio propagation data from XBee and UWB modules in various underground environments, including a long artificial tunnel, a natural sea cave, and a baseline indoor corridor. Our objective was to provide quantitative measures for use in making decisions pertaining to node deployment:

- *Task-dependent thresholding*: The established RSSI thresholds to designate two ranges describing situations as "detectable (Det.)" or "stable (Sta.)". "Stable" indicates that transmission data can be decoded accurately to guarantee an XBee transmission rate of 255 byte per second and with minimal UWB localization error. The values of RSSI are dependent on the device of the company and without unit. The RSSI of detectable and stable ranges of XBee are from −90 to −40 and from −70 to −40, respectively. Besides, the RSSI of the detectable and stable ranges of UWB are from −110 to −80 and from −90 to −80, respectively. Samples of normalized RSSI measured in the Shimen Tunnel were plotted in Figure 22. Nonetheless, measurements within the detectable range are susceptible to frame loss and/or missing data and prone to outlier UWB range estimates. Coordinating the sharing of maps among robots and the base station requires stable ranges. Transmit robot poses and artifact reports within detectable range are required with multiple attempts.
- *Coverage*: The RSSI values of XBee and UWB modules were collected by a UGV traversing various passages. The locations of our urban-node-bricks nodes were recorded at the time of deployment, whereupon the RSSI between the robot and each node was recorded. We then analyzed the RSSI gradients to estimate the coverage of each node along the recorded trajectory.
- *LOS vs. NLOS*: We examined the RSSI gradients of nodes placed at specified distances from a 90 degree turn, including the immediate RSSI drop at the intersection (G_{turn}) and the average RSSI gradients G after the turn. NLOS propagation while making a turn could potentially cause a high G_{turn} in narrow tunnels.

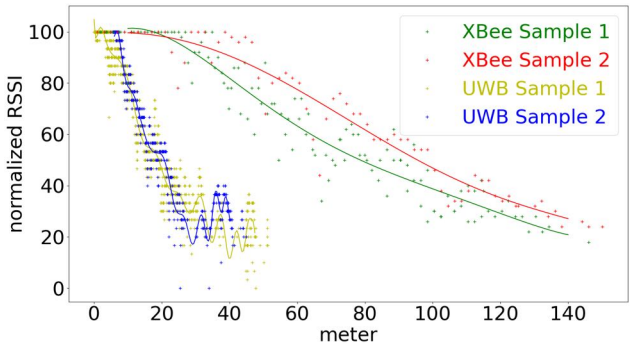


Figure 22. Signal strength measurements over various distances in Shimen Tunnel under LOS conditions. The radio signal strength indexes (RSSI) of XBee and UWB were normalized to a scale of 0 to 100.

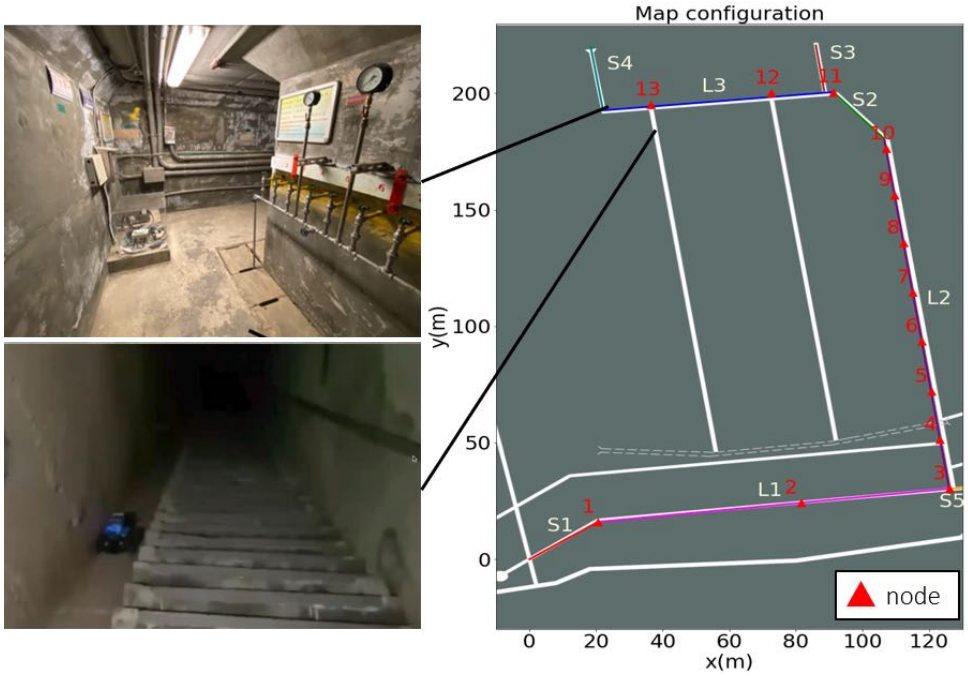


Figure 23. Shimen Tunnel, an artificial structure under the Shimen Reservoir, was built with regular corridor geometry. There were 13 nodes deployed, shown as red triangles. Average RSSI gradients were calculated over the segments of the main tunnel (L1 to L3) and the branches (S1 to S4).

4.2.2. Shimen Tunnel

Field tests were conducted in artificial tunnels beneath the Shimen Reservoir measuring roughly 8 km on four levels. The tunnels are used for regular public safety inspections. This network includes straight tunnels and narrow constrained passages, and the four levels are connected by stairs. For the sake of safety, data collection was conducted only in the main branch of the first level, which is equipped with 4G communication infrastructure, as shown in Figure 23. The cross section of corridors was consistent along the length of the passages; however, a notable lack of features provided challenges for mapping using existing SLAM algorithms. Technical drawings of the construction site were used as a ground truth map.

XBee Coverage. As shown in Table 12, the average LOS ranges of XBee were similar to the baseline indoor corridor. The standard deviation of the RSSI gradient in Shimen Tunnel was 0.21,

Table 12. Summary of coverage and normalized RSSI gradients (G, m^{-1}) of UWB and XBee modules under artificial tunnel, natural sea cave, and a baseline indoor corridor environments (Engineering Building 5 in NYCU). LOS: line of sight; NLOS: non line of sight; Cov.: coverage; Det.: detectable RSSI range; Sta.: stable RSSI range. G_{turn} refers to the larger degradation of the RSSI while the UGV turned into a branch and the communication became NLOS. RSSI gradients were normalized to a scale of 0 to 100, based on detectable RSSI range.

		LOS			NLOS		
Comm. Module	**Environment**	**Det. Cov.(m)**	**Sta. Cov.(m)**	**Avg. G.**	**Std. G.**	**G_{turn}**	**Avg. G.**
XBee	Shimen Tunnel	121.95	73.17	0.82	0.21	8	7.93
	BH Cave	42.92	25.75	2.33	0.70	11.04	9.68
	Indoor Corridor	125	75	0.80	0.14	12.86	4.10
UWB	Shimen Tunnel	27.47	9.07	3.64	1.61	20	-
	BH Cave	21.01	6.93	4.76	1.38	33.5	-
	Indoor Corridor	39.21	12.94	2.55	0.59	15.67	-

which was lower than in the BH Cave (0.70), indicating that similar scene geometries produced similar radio propagation characteristics. Taken together, these results indicate that in unknown environments, it is feasible to measure the coverage around entrances as long as the geometry is consistent. RSSI gradients and coverage can then be used to build a mesh network using multiple deployable nodes.

*XBee NLOS.*We determined that turns significantly degrade RSSI (G_{turn}) in NLOS communication systems, where G_{turn} and $Avg.G.$ were around 8. The reported NLOS RSSI gradient in Shimen Tunnel (7.93) exceeded that in the general indoor corridor (4.10), which may be caused by the narrower corridor geometry. We found that the signals were still able to propagate through the NLOS passage after the turn, which suggests that nodes do not necessarily have to be placed at all intersections. Rather, the decision of whether to deploy a node should be based on whether NLOS signals fall off faster than LOS signals after the turn.

UWB. We observed higher LOS and NLOS RSSI gradients, indicating shorter coverage. Specifically, the UWB appeared not to propagate in NLOS situations, due to a large G_{turn}. We also observed that UWB was not usable for localization outside its stable range (i.e., −110 to −90 dBm).

4.2.3. BH Cave

XBee Coverage. Unlike the Shimen Tunnel, the geometry of BH Cave is highly irregular, as shown 24. As a result, the detectable coverage and stable coverage were both shorter than in the Indoor Corridor and Shimen Tunnel. We also observed larger standard deviation in RSSI gradients (0.70), indicating that radio propagation varied considerably among the various areas within BH Cave, which greatly hindered modeling.

XBee NLOS. BH Cave includes several narrow twisting branches, which produce far more NLOS situations than in the other two environments. The G_{turn} was 11.04, higher than in Shimen Tunnel. Average RSSI gradients of NLOS (9.68) were higher than in the other two locations.

UWB. In the cave, the average detectable coverage was 21.01 m and the average stable coverage was 6.93 m. Both were shorter than the other two environments. When using UWB for localization, it is crucial that the robot remain within the range of stable coverage. As in the other two environments, NLOS communication was not available, due to a large G_{turn}. Thus, nodes should be placed at all intersections to maintain the link.

4.3. Coordinating Vehicles, Nodes, and Human Supervisors

The infrastructure used to coordinate vehicles, nodes, and human supervisors was based on our previous work in Duckietown (Paull et al., 2017), in which intervehicle communication was decentralized and perception based. In this paper, we created an ROS core for each robot, and

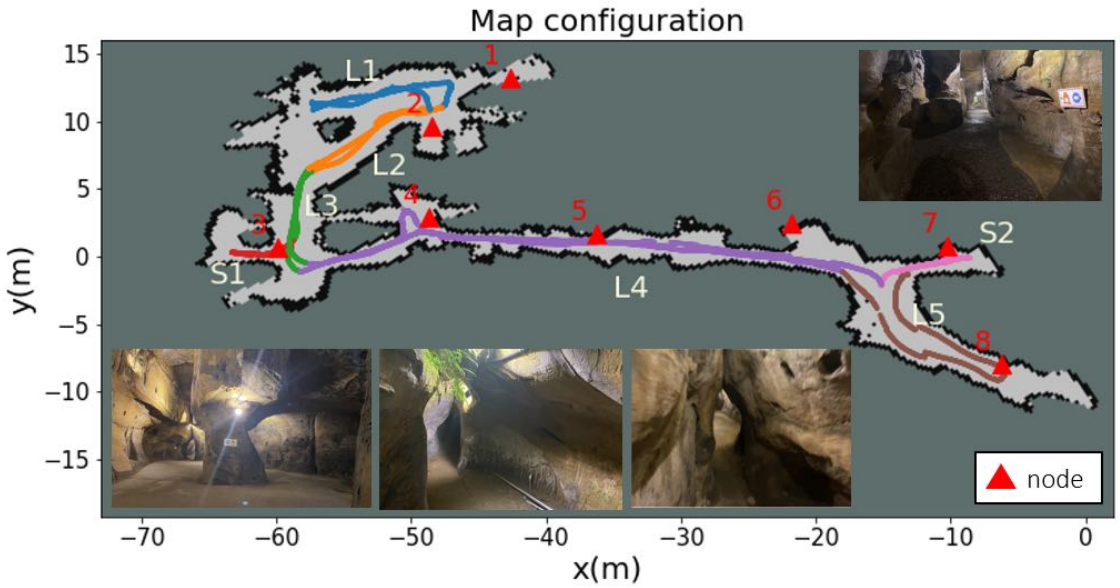


Figure 24. BH sea cave featuring irregular walls and small tapering caves. The deployed nodes are marked as red triangles. The robot followed the main branch (labeled L), explored the twig segments (labeled S), and returned to the base station.

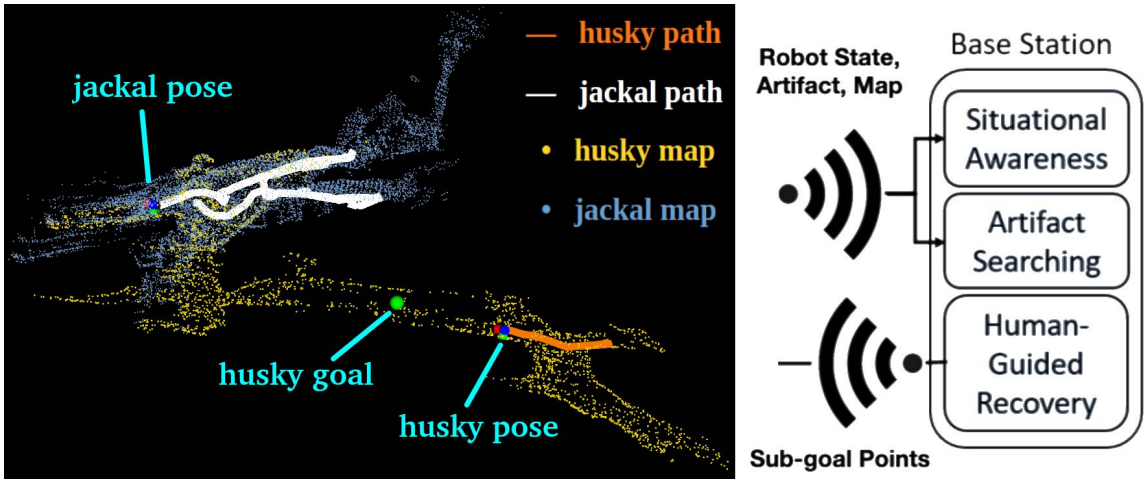


Figure 25. Coordination between vehicles and human supervisor via deployable nodes. Each vehicle broadcast robot states, artifacts found, and map (point cloud). The human supervisor interactively provided subgoals via GUI, and selecting locations of dropping deployable nodes. Although all robots could receive the packets from other robots, the high-level decisions were currently only made by the human supervisor.

communication was handled by XBee. Figure 25 illustrates the scheme used to coordinate multiple robots and human supervisors with each vehicle decentralized. Coordination was performed by a human supervisor (at a base station) using a graphic user interface to assign subgoal points pertaining to each vehicle. Mapping results sent back from multiple robots were used by the human supervisor to assign subgoals in unexplored regions, whereupon the DRL goal navigation model was used to perform autonomous navigation. The human supervisor was also responsible for selecting locations at which to drop deployable nodes. During exercises, each vehicle used several 250 byte packages to broadcast (at intervals of one second) information performing to its operating status,

the artifacts it found, and a corresponding map. All of the robots received updates about the status of other robots; however, all decisions were made by human supervisors.

5. Discussion

5.1. Limitations and Lessons Learned

Our experience in this project taught us a number of lessons pertaining to the mobility of Duckiefloat blimps, the control of Duckiefloat blimps in simulations, the use of mmWave sensors for localization of perception, sim-to-real deep reinforcement learning, communication modules, and radio propagation. It appears that blimps are well-suited to tunnel environments with a regular corridor geometry, due to their excellent energy efficiency, long flight time, and high tolerance for collisions. However, they are difficult to maneuver in tight spaces and susceptible to strong air currents, which prompted us to add tail motors for heading control and employ more powerful brushless motors.

An effective control system depends on an accurate model of blimp dynamics. Our attempt at establishing a 6-degrees of freedom nonlinear mathematical model ran into difficulties due to imprecision in parameter estimates, which prevented smooth operations in the virtual environment. We also had to accept the fact that without an extensive background in aerodynamics and fluid mechanics, we would be unable to derive the precise data required for this type of modeling.

Our use of mmWave radar for collision avoidance proved highly robust to smoke and other obscurants in SubT environments (Huang et al., 2021). Single-chip mmWave radar devices are light enough and energy efficient enough for aerial vehicles and provide sufficient FOV for effective navigation. Our experience in the Urban Circuit convinced us that heterogeneous robots can learn navigation via transfer learning based on mmWave radar data. Nonetheless, further research will be required to determine whether mmWave radar is also applicable to SLAM.

LIDAR-based SLAM proved effective in most situations (as long as confidence tests were performed); however, vision-based SLAM was susceptible to variations in illumination. The efficacy of SBL-UWB in determining the relative position of robots with respect to one another means that it could be used in cooperative multirobot missions. It should also be possible to compare a variety of SLAM algorithms using various parameter sets to optimize SLAM for specific subterranean environments. The best combination (i.e., with the highest confidence) could then be used to ensure failure-free SLAM in subsequent areas. The use of semantic segmentation in the search for artifacts does not pose difficulties as long as we know where to look. However, implementing this type of system over a wide FOV may require multiple cameras and heavy computational resources. In this study, we established benchmarks for state-of-the-art semantic segmentation algorithms [i.e. Mask-RCNN (He et al., 2018) and Yolact (Bolya et al., 2019)] with DARPA-released logs. Besides developing more robust algorithms, a possible future research direction is having the robots equipped with a 360 degrees, 3D visual coverage camera system. The simulation environments provided by the virtual track facilitated our development of a navigation system based on deep reinforcement learning. Our current DRL scheme enables collision avoidance with low-level control without the need for an external high-level planner. We believe that DRL is promising for real-world use, alone or in conjunction with frontier search algorithms.

Nodes must be deployed wisely; therefore, we suggest further analysis of radio propagation in SubT environments and the sharing of datasets as a practical baseline for further developments. Our results demonstrated the importance of finding testing sites that are similar to the competition environment in order to gain insights and avoid potential pitfalls. It is also important to establish plans and schedules for the development and revision of hardware to ensure that subsequent software development is not delayed. Finally, it should be noted that a number of our project milestones were too ambitious for the tight schedule imposed by the impending competition. The Cave Circuit was canceled due to the COVID-19 outbreak; however, we still learned a great deal about long-term preparation and development, as evidenced by our organization of subsystems (perception, autonomy, SLAM, communication, and infrastructure). We also learned the importance of focusing on the fact that the metric of success was the number of points scored.

5.2. Reflection on the Competition

There was considerable debate regarding whether or not a system team should also participate in virtual challenges. We appreciate the efforts of the organizers in providing simulation platforms to facilitate the developments of sim-to-real deep reinforcement learning algorithms. Virtual challenges were a viable solution during the COVID-19 outbreak; however, we considered our long-term research goal to deploy actual robots in the field and therefore opted not to participate in the virtual track. Our strategy in the Tunnel and Urban challenges were based on scoring protocols. The fact that we did not participate in the STIX event meant that we had to overcome a steep learning curve in dealing with the logistics for the Tunnel Circuit. Even after working hard to overcome those issues, the Urban Circuit was an arduous challenge. We look toward the final event with some trepidation, due to the fact that it will include all three environments (tunnel, urban, and cave). We were impressed by the definition and execution of rules by DARPA. Considering that the competition sites are unknown to participants prior to the events, it should be difficult to figure out how much information to disclose to teams. We were particularly concerned about the width of passages at the Urban Circuit site. We were made aware of the fact that the width of passages would vary considerably from large open areas to narrow constrained doorways. To quote from the information package: "For the Urban Circuit, it is expected that up to 50% of the competition course could be inaccessible to systems that cannot traverse these passages." During the competition run however, we ran into trouble with a narrow gate next to the starting gate, where high airflow greatly affected the capabilities of the robots. Clearly, the accessibility of the course did not match the description. We would appreciate it if DARPA could provide accurate site descriptions pertaining to corridor geometry and airflow.

5.3. Future plans

Subsequent research will focus on advancing real-world reinforcement learning (RWRL). Current navigation policies trained using DRL algorithms proved effective in navigating the UGV under our testing scenarios; however, further work will be required to extend them to a wider range of environments. We will also continue our work on transfer learning to train policies for blimp simulation. Further advances in RWRL impose a number of challenges, including reward unspecified (Fu et al., 2018), multiobjective, trained from (Fu et al., 2020), partially observable tasks (Dulac-Arnold et al., 2019).We are also planning to include communication-aware, i.e., RSSI inputs to DRL formulation, and develop policies for the dispersal of nodes with rewards for maximizing coverage.

We will also begin work on a high-level planner using PyRobot (Murali et al., 2019) for the development of systems that place greater emphasis on task-level autonomy. For example, the human supervisor should be able to employ a state-machine structure in the assignment of tasks, as follows: "Robot 1, move to the next intersection, drop a deployable node, switch to mmWave mode for smoke-filled environment, and return to the node." The robot should be given the ability to assess situations and make reasonable decisions pertaining to the next course of action.

We are currently involved in the development of a modular sensor tower, autonomous-operations box, and battery box, to facilitate the deployment of modules to heterogeneous robots. We hope that these developments will be available for future projects. It was a privilege to participate in the DARPA SubT Challenge in developing technologies capable of reaching dangerous places where humans cannot go. We have begun making connections and acquired research funds to work on systems for the inspection of public facilities, communications systems for nuclear research facilities, and robotic systems to assist frontline medical staff in isolation wards.

6. Conclusion

The aim of research into search and rescue robots is to develop systems capable of enhancing situational awareness without exposing humans to danger. The DARPA Subterranean (SubT) Challenge was meant to accelerate research in this area. The first two phases of the challenge were held in a mine tunnel and an abandoned nuclear power plant. Teams were tasked with locating

specific artifacts, while dealing with difficulties typical of subterranean environments, including austere navigation, degraded sensing, severed communications, and rough terrain. The problems can be broken down to those of autonomy, mobility, perception, and communication. The system we developed for these events was a heterogeneous team of robots, including a low-cost autonomous blimp with extended flight capability, a platform-agnostic mmWave sensor tower, learning-based algorithms for navigation through an obscured atmosphere, and deployable nodes for localization and communication.

The autonomous blimp, Duckiefloat, was meant to overcome extreme ground-level mobility issues. We developed various hardware configurations, including the envelope, rotors, and frame designs. The development of Duckiefloat was in using rolling bases and gradually solved the unique challenges in SubT environments. Preliminary testing prompted us to replace the Styrofoam airframe with carbon fiber struts. The final version of Duckiefloat was able to travel through narrow passages (85 cm in width). Adjustments were made to the drive system to enhance steadiness and hovering ability in the presence of airflow. Mobility tests were conducted in controlled maze environments as well as real-world locations, including an abandoned bunker, railway tunnels, indoor corridors, and a natural sea cave. Our results indicate that blimps do not provide a general solution; however, they could be invaluable in some scenarios. We also developed an autonomous UGV system based on a Husky with a proprietary sensor tower and deployable nodes for communication.

The ability to function under degraded sensing scenarios is crucial to the success of vehicles navigating autonomously through SubT environments. We developed a platform-agnostic mmWave sensor tower in conjunction with learning-based algorithms to achieve this end. Our navigation agent was based on DRL aimed at navigating to a designated goal point using point clouds as the input. Our DRL scheme also included reward functions and a network architecture. The DRL agent trained in the Gazebo virtual-subt-cave environment integrated with the OpenAI Gym API was able to navigate effectively under normal conditions. mmWave sensors were used to extend functionality to situations where cameras would be unable to function effectively, such as environments filled with smoke. We adopted cross-modal contrastive learning for representation (CM-CLR) to maximize agreement between mmWave data and LiDAR data. Our end-to-end DRL policy proved highly effective in navigating the robot through a smoke-filled maze achieving performance on par with generative reconstruction methods. The light weight of mmWave sensors also made them ideally for our dirigible. SLAM and deep learning-based semantic segmentation were used to locate artifacts. In a comparison of state-of-the-art semantic segmentation algorithms (MaskRCNN, Yolact and FCN), the MaskRCNN model was the most effective for large artifacts like survivors, backpack, and vent, whereas the Yolact model was overly sensitive to noise in SubT environments. Sudden bounces by the UGV had a profound effect on SLAM performance. Analysis of sudden bounces based on the linear velocity of the UGV provided a threshold for confidence tests to be used in recovering the map after such incidents.

Communication is the key to cooperation among robots. We developed deployable nodes to establish XBee and UWB mesh networks. In three representative environments, we measured the RSSI of the two systems from a UGV as it traversed the passages. We were able to establish RSSI thresholds to designate two ranges describing situations as “detectable” or “stable”. We also analyzed the RSSI gradient in line-of-sight and none-line-of-sight situations to elucidate the optimal dispersal of communication nodes. In a cave used for testing, the robots shared the maps they discovered and their current poses with each other and with the human supervisor. Note that the proposed nodes can also be used as anchors for localization. Finally, we developed a system for the localization of nodes and artifacts based on audible signals.

We hope that our research will inspire further work in subterranean robotics.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology (MOST) of Taiwan under Grants MOST 109-2634-F-009-019, MOST 110-2634-F-009-022, and MOST 109-2634-F-009-027,

through Pervasive Artificial Intelligence Research (PAIR) Labs under Grants MOST 109-2321-B-009-006, MOST 109- 2224-E-007-004, and MOST 109-2221-E-009-074, and in part by Qualcomm Taiwan University Research 2020 Program.

ORCID

Chen-Lung Lu https://orcid.org/0000-0002-0240-3079
Jui-Te Huang https://orcid.org/0000-0001-6527-5812
Ching-I Huang https://orcid.org/0000-0002-3987-259X
Zi-Yan Liu https://orcid.org/0000-0001-5434-2135
Chao-Chun Hsu https://orcid.org/0000-0002-7301-2884
Yu-Yen Huang https://orcid.org/0000-0001-7433-1687
Siao-Cing Huang https://orcid.org/0000-0003-3435-4387
Po-Kai Chang https://orcid.org/0000-0001-7481-4478
Zu Lin Ewe https://orcid.org/0000-0003-3515-9989
Po-Jui Huang https://orcid.org/0000-0003-0284-6934
Po-Lin Li https://orcid.org/0000-0002-9889-7957
Bo-Hui Wang https://orcid.org/0000-0001-6416-6286
Lai-Sum Yim https://orcid.org/0000-0002-8062-4221
Sheng-Wei Huang https://orcid.org/0000-0002-0507-8224
MingSian R. Bai https://orcid.org/0000-0002-8866-106X
Hsueh-Cheng Wang https://orcid.org/0000-0002-9419-4111

References

Anderson, P., Chang, A., Chaplot, D. S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M. *et al.* (2018). On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Bai, M. R., Ih, J.-G., and Benesty, J. (2013). *Acoustic array systems: theory, implementation, and application*. Wiley.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. (2018). Distributional policy gradients. In *International Conference on Learning Representations*.

Basu, P., Redi, J., and Shurbanov, V. (2004). Coordinated flocking of uavs for improved connectivity of mobile ground nodes. In *IEEE MILCOM 2004. Military Communications Conference, 2004*, volume 3, pages 1628–1634. IEEE.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). Yolact: Real-time instance segmentation.

Boroson, E. R., Hewitt, R., Ayanian, N., and de la Croix, J.-P. (2020). Inter-robot range measurements in pose graph optimization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4806–4813.

Bouman, A., Ginting, M. F., Alatur, N., Palieri, M., Fan, D. D., Touma, T., Pailevanian, T., Kim, S.-K., Otsu, K., Burdick, J. et al. (2020). Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. *arXiv preprint arXiv:2010.09259*.

Brand, I., Roy, J., Ray, A., Oberlin, J., and Oberlix, S. (2018). Pidrone: An autonomous educational drone using raspberry pi and python. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–7. IEEE.

Broome, M., Gadd, M., De Martini, D., and Newman, P. (2020). On the road: Route proposal from radar self-supervised by fuzzy lidar traversability. *AI*, 1(4):558–585.

Burri, M., Gasser, L., Käch, M., Krebs, M., Laube, S., Ledergerber, A., Meier, D., Michaud, R., Mosimann, L., Müri, L., et al. (2013). Design and control of a spherical omnidirectional blimp. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1873–1879. IEEE.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., and Hutter, M. (2020). Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388.

DARPA (2019). Darpa subterranean challenge.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., et al. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of real-world reinforcement learning.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110.

Everett, M., Chen, Y. F., and How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377.

Fan, D. D., Thakker, R., Bartlett, T., Miled, M. B., Kim, L., Theodorou, E., and Agha-mohammadi, A. (2019). Autonomous hybrid ground/aerial mobility in unknown environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3070–3077.

Fan, T., Long, P., Liu, W., and Pan, J. (2020). Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892.

Fedorenko, R. and Krukhmalev, V. (2016). Indoor autonomous airship control and navigation system. In *MATEC Web of Conferences*, volume 42, page 01006. EDP Sciences.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning.

Fu, J., Luo, K., and Levine, S. (2018). Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*.

Fukao, T., Fujitani, K., and Kanade, T. (2003). An autonomous blimp for a surveillance system. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 2, pages 1820–1825. IEEE.

Funabiki, N., Morrell, B., Nash, J., and Agha-mohammadi, A.-a. (2020). Range-aided pose-graph-based slam: Applications of deployable ranging beacons for unknown environment exploration. *IEEE Robotics and Automation Letters*, 6(1):48–55.

Gawel, A., Lin, Y., Koutros, T., Siegwart, R., and Cadena, C. (2018). Aerial-ground collaborative sensing: Third-person view for teleoperation. In *Safety, Security and Rescue Robotics (SSRR), IEEE International Symposium on*.

González, P., Burgard, W., Sanz Domínguez, R., and López Fernández, J. (2009). Developing a low-cost autonomous indoor blimp.

Grondin, F. and Michaud, F. (2019). Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations. *Robotics and Autonomous Systems*, 113:63–80.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE.

Guérin, F., Guinand, F., Brethé, J.-F., Pelvillain, H., et al. (2015). UAV-UGV cooperation for objects transportation in an industrial area. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 547–552. IEEE.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). Mask r-cnn.

Heess, N., Hunt, J. J., Lillicrap, T. P., and Silver, D. (2015). Memory-based control with recurrent neural networks.

Hsieh, M. A., Cowley, A., Kumar, V., and Taylor, C. J. (2008). Maintaining network connectivity and performance in robot teams. *Journal of field robotics*, 25(1-2):111–131.

Huang, J.-T., Lu, C.-L., Chang, P.-K., Huang, C.-I., Hsu, C.-C., Ewe, Z. L., Huang, P.-J., and Wang, H.-C. (2021). Cross-modal contrastive learning of representations for navigation using lightweight, low-cost millimeter wave radar for adverse environmental conditions.

Huang, Y., Lu, C., Chen, K., Ser, P., Huang, J., Shen, Y., Chen, P., Chang, P., Lee, S., and Wang, H. (2019). Duckiefloat: a collision-tolerant resource-constrained blimp for long-term autonomy in subterranean environments. *CoRR*, abs/1910.14275.

Huber, D. F. and Vandapel, N. (2003). Automatic 3D underground mine mapping. In *Field and Service Robotics*, pages 497–506. Springer.

Hygounenc, E., Jung, I.-K., Soueres, P., and Lacroix, S. (2004). The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *The International Journal of Robotics Research*, 23(4-5):473–511.

Jaritz, M., de Charette, R., Toromanoff, M., Perot, E., and Nashashibi, F. (2018). End-to-end race driving with deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2070–2075. IEEE.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *IInternational Journal of Robotics Research (IJRR)*, 31:217–236.

Kahn, G., Abbeel, P., and Levine, S. (2020). Badgr: An autonomous self-supervised learning-based navigation system.

Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T., et al. (2006). Toward reliable off road autonomous vehicles operating in challenging environments. *The International Journal of Robotics Research*, 25(5-6):449–483.

Ko, J., Klein, D. J., Fox, D., and Haehnel, D. (2007). Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 742–747. IEEE.

Kramer, A., Harlow, K., Williams, C., and Heckman, C. (2021). Coloradar: The direct 3D millimeter wave radar dataset. *arXiv preprint arXiv:2103.04510*.

Liando, J. C., Gamage, A., Tengourtius, A. W., and Li, M. (2019). Known and unknown facts of lora: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks (TOSN)*, 15(2):1–35.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In Bengio, Y. and LeCun, Y., editors, *ICLR*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Mai, V., Kamel, M., Krebs, M., Schaffner, A., Meier, D., Paull, L., and Siegwart, R. (2018). Local positioning system using uwb range measurements for an unmanned blimp. *IEEE Robotics and Automation Letters*, 3(4):2971–2978.

Manderson, T., Gamboa, J. C., Wapnick, S., Tremblay, J.-F., Meger, D., and Dudek, G. (2020). Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles. In *Proceedings of Robotics: Science and Systems*.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., and Tadokoro, S. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841.

Min, B.-C., Kim, Y., Lee, S., Jung, J.-W., and Matson, E. T. (2016). Finding the optimal location and allocation of relay robots for building a rapid end-to-end wireless communication. *Ad Hoc Networks*, 39:23–44.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Murali, A., Chen, T., Alwala, K. V., Gandhi, D., Pinto, L., Gupta, S., and Gupta, A. (2019). Pyrobot: An open-source robotics framework for research and benchmarking. *arXiv preprint arXiv:1906.08236*.

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., et al. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63.

Niroui, F., Zhang, K., Kashino, Z., and Nejat, G. (2019). Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, 4(2):610–617.

Nüchter, A., Bleier, M., Schauer, J., and Janotta, P. (2017). Improving Google's cartographer 3D mapping by continuous-time slam. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:543.

ootang2018 (2020). blimp simulator. https://github.com/ootang2018/blimp_simulator.

Palossi, D., Gomez, A., Draskovic, S., Keller, K., Benini, L., and Thiele, L. (2017). Self-sustainability in nanounmanned aerial vehicles: A blimp case study. In *Proceedings of the computing frontiers conference*, pages 79–88.

Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y. F., Choi, C., Dusek, J., Fang, Y., et al. (2017). Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1497–1504. IEEE.

Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y. F., Choi, C., Dusek, J., Fang, Y., Hoehener, D., Liu, S. Y., Novitzky, M., Okuyama, I. F., Pazis, J., Rosman, G., Varricchio, V., Wang, H. C., Yershov, D., Zhao, H., Benjamin, M., Carr, C., Zuber, M., Karaman, S., Frazzoli, E., Del Vecchio, D., Rus, D., How, J., Leonard, J., and Censi, A. (2017). Duckietown: An open, inexpensive and flexible platform for autonomy education and research. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1497–1504.

Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., and Cadena, C. (2017). From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In *2017 ieee international conference on robotics and automation (ICRA)*, pages 1527–1533. IEEE.

Pfeiffer, M., Shukla, S., Turchetta, M., Cadena, C., Krause, A., Siegwart, R., and Nieto, J. (2018). Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. *IEEE Robotics and Automation Letters*, 3(4):4423–4430.

Richardson, T. S., Jones, C. G., Likhoded, A., Sparks, E., Jordan, A., Cowling, I., and Willcox, S. (2013). Automated vision-based recovery of a rotary wing unmanned aerial vehicle onto a moving platform. *Journal of Field Robotics*, 30(5):667–684.

Rogers, J. G., Gregory, J. M., Fink, J., and Stump, E. (2020). Test your slam! the subt-tunnel dataset and metric for mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 955–961.

Rottmann, A., Zitterell, T., Burgard, W., Reindl, L., Scholl, C., et al. (2007). Towards an experimental autonomous blimp platform.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173.

Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

Sa, I., Kamel, M., Burri, M., Bloesch, M., Khanna, R., Popović, M., Nieto, J., and Siegwart, R. (2017). Build your own visual-inertial drone: A cost-effective and open-source autonomous drone. *IEEE Robotics & Automation Magazine*, 25(1):89–103.

Sadeghi, F. and Levine, S. (2016). Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*.

Santamaria-Navarro, A., Thakker, R., Fan, D. D., Morrell, B., and Agha-mohammadi, A.-a. (2020). Towards resilient autonomous navigation of drones. *arXiv preprint arXiv:2008.09679*.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., and Batra, D. (2019). Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwartz, J. T. and Sharir, M. (1983). On the piano movers' problem: III. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2(3):46–75.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA.

Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765.

Shivakumar, S. S., Rodrigues, N., Zhou, A., Miller, I. D., Kumar, V., and Taylor, C. J. (2020). Pst900: Rgb-thermal calibration, dataset and segmentation network. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9441–9447.

Song, Y., Guan, M., Tay, W. P., Law, C. L., and Wen, C. (2019). Uwb/lidar fusion for cooperative range-only slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6568–6574.

Stentz, T., Kelly, A., Herman, H., Rander, P., Amidi, O., and Mandelbaum, R. (2002). Integrated air/ground vehicle system for semi-autonomous off-road navigation. In *Proceedings of the AUVSI Unmanned Systems Conference*.

Tai, L., Paolo, G., and Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36. IEEE.

Trevor, A. J. B., Rogers, J. G., and Christensen, H. I. (2014). Omnimapper: A modular multimodal mapping framework. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1983–1990.

Wada, K. (2016). labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme.

Xia, F., Zamir, A. R., He, Z., Sax, A., Malik, J., and Savarese, S. (2018). Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079.

Xiong, C., Han, D., and Xiong, Y. (2009). An integrated localization system for robots in underground environments. *Industrial Robot: An International Journal*, 36(3):221–229.

Yao, N., Anaya, E., Tao, Q., Cho, S., Zheng, H., and Zhang, F. (2017). Monocular vision-based human following on miniature robotic blimp. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3244–3249. IEEE.

Yao, N.-s., Tao, Q.-y., Liu, W.-y., Liu, Z., Tian, Y., Wang, P.-y., Li, T., and Zhang, F. (2019). Autonomous flying blimp interaction with human in an indoor space. *Frontiers of Information Technology & Electronic Engineering*, 20(1):45–59.

Zhen, W. and Scherer, S. (2019). Estimating the localizability in tunnel-like environments using lidar and uwb. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4903–4908.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE.

How to cite this article: Lu, C.-L., Huang, J.-T., Huang, C.-I., Liu, Z.-Y., Hsu, C.-C., Huang, Y.-Y., Huang, S.-C., Chang, P.-K., Ewe, Z. L., Huang, P.-J., Li, P.-L., Wang, B.-H., Yim, L.-S., Huang, S.-W., Bai, M. R., & Wang, H.-C. (2022). A Heterogeneous Unmanned Ground Vehicle and Blimp Robot Team for Search and Rescue using Data-driven Autonomy and Communication-aware Navigation. *Field Robotics, 2*, 557–594.