

# A Low-Cost Approach for Real-Time Autonomous Large-Scale Exploration

Adam Seewald<sup>1</sup>, Marvin Chancán<sup>1</sup>, Connor M. McCann<sup>2</sup>, Seonghoon Noh<sup>1</sup>, Omeed Fallahi<sup>1</sup>, Hector Castillo<sup>1</sup>, Ian Abraham<sup>1</sup>, and Aaron M. Dollar<sup>1</sup>

**Abstract—**

**Index Terms**—Article submission, IEEE, IEEEtran, journal, L<sup>A</sup>T<sub>E</sub>X, paper, template, typesetting.

## I. INTRODUCTION

WIDELY used in cluttered environments [1]–[4], mobile robots can both substitute [5] and outperform humans in, e.g., areas that are too far or too dangerous to navigate [6]–[9]. In these areas, robots are often required to identify their surroundings by sensing the environment [10] and planning and executing complex trajectories [11], [12]. With little or no human intervention [13], this problem is known in the literature as autonomous exploration [11]. While successful in challenging indoor and outdoor environments [14], [15], autonomous exploration is especially useful in dynamic environments with no prior knowledge of the space to be covered [5], [16]. Despite recent advancements, autonomy is limited and costly in such environments. Many approaches that tackle autonomous exploration integrate commercial robots with sensing equipment that is both prohibitively expensive and difficult to maintain [8], [9], [14], [15], [17]–[20]. There is a wide range of methodologies for autonomous exploration at present [15], [21] nonetheless, which span from algorithmic foundations [15], [19], [22] to system-of-systems frameworks where, e.g., a multitude of robots integrate existing algorithms with sensors for real-time large-scale exploration [3], [7]–[9], [18]. Recent efforts in this direction include low-cost robots for exploration [17], [23], [24] but lack terrain adaptability [17] and capabilities [23], [24] often required to navigate outdoors in the real-world [2], [5].

Furthermore, in areas that are ambiguous or challenging to traverse—albeit autonomous—state-of-the-art approaches rely on humans for supervision and high-level decision-making [3], [7], [8]. As a result, robots often operate close to humans or require expensive network equipment, such as a mesh of communication devices [2], [3], [9], or existing network

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments.

<sup>1</sup>A. S., C. M., S. N., O. F., H. C., I. A., and A. M. D. are with the Department of Mechanical Engineering and Materials Science, Yale University, CT, USA. Email: [adam.seewald@yale.edu](mailto:adam.seewald@yale.edu);

<sup>2</sup>C. M. C. is with the School of Engineering and Applied Sciences, Harvard University, MA, USA.

Digital Object Identifier (DOI): see top of this page.

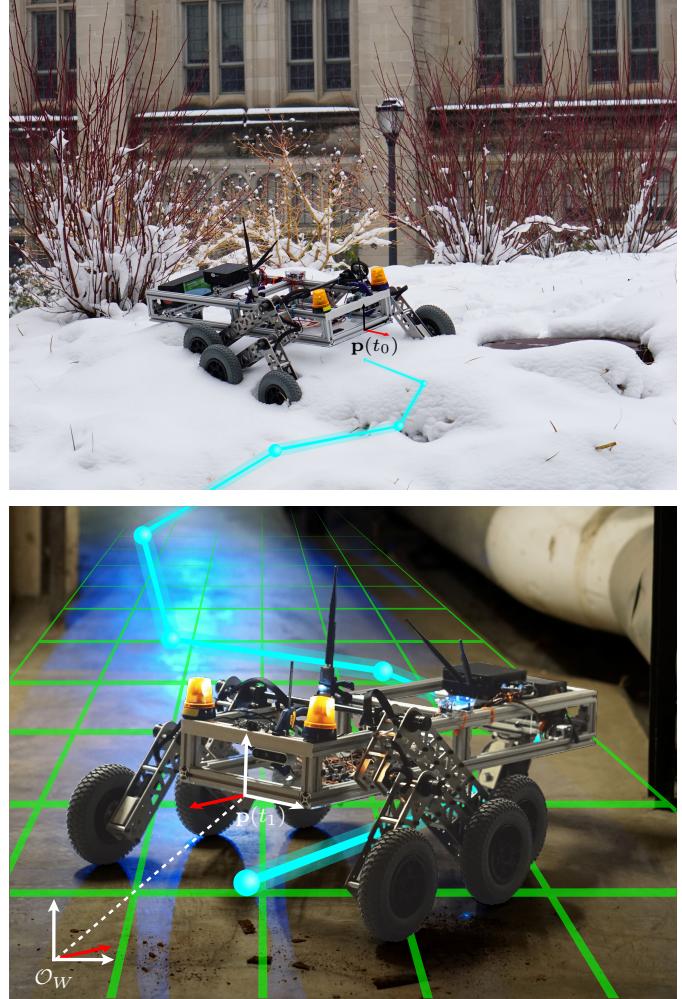


Fig. 1:

infrastructure [25]–[27], thereby restricting autonomous exploration to indoor settings only [12], [28]–[31]. Conversely, our methodology exploits LoRa—an inexpensive long-range and low-power communication technology [32] from the internet-of-things domain—with a customized communication protocol for human intervention in, e.g., the eventuality of the robot being unable to move with the local sensory information.

Starting from the cost advantages of LoRa communication, we develop here a low-cost approach consisting of an exploration framework based on the open-source robot operating system (ROS) middleware [33] and an experimental robotic

platform–RB5, a mobile robot with rocker-bogie suspension–capable of exploring autonomously dynamic indoor and outdoor environments. Similar platforms in the literature comprise two degrees of freedom suspension with pivots [24], [34], [35] (see Figure 1) and provide rough terrain static adaptability [36]. They are cheaper than, e.g., legged robots in terms of cost per unit and operation, as they are able to overcome obstacles without costly computations for gait adaptation and planning [17]. Although specific to RB5 in the letter, the approach is generic in terms of portability to other mobile robots with cost and computational constraints. Hardware-wise, the approach maintains a lower sensory footprint with low-cost components, whereas software-wise, it integrates multiple modules into the exploration framework. Being able to operate in both unknown and GPS-denied environments, the approach derives the robot’s position using a state-of-the-art simultaneous localization and mapping (SLAM) algorithm [37] and the trajectory with a novel methodology that extends exploration literature with a path following vector field [38] from the aerial robotics domain [39]–[41]. This allows the robot to explore its surroundings at lower frequencies and in real-time, utilizing cheaper computing hardware compared to state-of-the-art approaches [8], [9], [15], [19].

The remainder of the letter is structured as follows. In Section V, data show the exploration performance and obstacle avoidance features over the baseline of existing autonomous exploration system-of-systems with indoor and outdoor “in the field” experiments. Sec. IV describes the approach from the software and hardware standpoints. Sec. II summarizes and compares existing literature, Sec. III formalizes the problem of autonomous exploration, and Sec. VI drafts conclusions and future directions.

## II. RELATED WORK

Literature that tackles autonomous exploration is broad and diverse [14], [15], [21], yet, most approaches use expensive sensing equipment such as LiDARs [1]–[3], [7], [9], [18], [19], [42] and laser range finders [4], [20]. Even though approaches that utilize cheaper sensors, such as RGB depth (RGB-D) cameras [3], [4], [7]–[9], [12], [16], [31], [43], RGB cameras [17], [19], [23], and sonars [17], [23], are studied, they often operate along other more expensive equipment [3], [4], [7], [9], [19] or indoors only [12], [16], [23], [31], [43], and have limited autonomy [16], [31], [43] or obstacle avoidance features [17], [23]. Recent approaches minimize exploration costs nonetheless [8], [12], [16], [17], [23] by, e.g., exploiting sensing capabilities of commercial smartphones [17], [23] or using case-specific aspects [8] but are generally unable to operate in a wide variety of challenging environments. It is such a state of practice that has prompted us to propose a low-cost approach to autonomous exploration. Similar to recent literature, it can operate on a large scale but with a lower sensory footprint—an RGB-D camera to sense the environment—and in a variety of terrains both indoors and outdoors.

Software-wise, recent efforts into autonomous exploration often require expensive prior learning [11] or run on multiple

agents [2], [3], [7], whereas approaches with little computing resources are scarce [16], [17], [24], [42]. More traditional approaches such as these based on frontiers [4], [7], [42], graphs [2], [3], [19], grids [8], [30], and random trees [31] are also studied, but mixed approaches are to be preferred [1], [10], [11], [16], [20], [44] to maximize performance and resources [15], [16]. Similarly, we propose a mixed approach. A frontier-based method that exploits the scarcity of resources while still running in real-time with comparable autonomy and obstacle avoidance features to its more expensive counterparts [3], [4], [7], [19].

## III. PROBLEM FORMULATION

The problem considered in this work to showcase the approach for large-scale exploration is that of exploring a bounded volume  $\mathcal{Q} \subseteq \mathbb{R}^3$  with respect to an inertial navigation frame  $\mathcal{O}_W$ . If the notation  $[n]$  denotes a set with positive naturals up to  $n$  and  $[n]_{>0}$  with strictly positive naturals, we are interested in collision-free trajectories that explore  $\mathcal{Q}$  and avoid  $i \in [n]_{>0}, n \in \mathbb{N}_{\geq 0}$  obstacles  $\mathcal{Q}^{O_i} \subset \mathbb{R}^3$ . We can approximate the space that delimits  $\mathcal{Q}$  and  $\mathcal{Q}^{O_i}$  for each  $i$  with a set of vertices within which the two sets are contained.

**Problem (Exploration).** Consider sets of vertices  $V := \{\mathbf{v}_1, \mathbf{v}_2, \dots\}$ ,  $O_i := \{\mathbf{o}_{i,1}, \mathbf{o}_{i,2}, \dots\}$  with  $i \in [n], \mathbf{v}_j, \mathbf{o}_{i,k} \in \mathbb{R}^2$ ,  $\forall j \in [|v|], k \in [|o_i|]$  a point w.r.t.  $\mathcal{O}_W$ . Let  $V$  enclose  $\mathcal{Q}$ ,  $O_i \subset \mathcal{Q}^{O_i}$  per each  $i$ . The *exploration problem* is the problem of finding the coverage that visits each point  $\mathbf{p} \in \mathcal{Q} \cap \mathcal{Q}^{O_1} \cap \mathcal{Q}^{O_2} \cap \dots \cap \mathcal{Q}^{O_n} := \mathcal{Q}^V$ .

Here the notation  $|\cdot|$  denotes the cardinality and  $\mathbb{R}, \mathbb{Z}$  are reals and integers. Bold notation is used for vectors.

Let  $\phi$  be a path function, i.e., a function the robot tracks as it explores its surroundings in  $\mathcal{Q}^V$ , avoiding the obstacles  $\mathcal{Q}^{O_i}$ .

**Definition III.1** (Path function).  $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$  is a two-dimensional continuous and differentiable *path function* of the  $x, y$  components of  $\mathbf{p}$ .

**Definition III.2** (Coverage). Given a tuple with a path function and its time component,  $\langle \phi, t \rangle$ , the *coverage* is the collection of multiple tuples.

The large-scale exploration framework (see Sec. IV-A) derives  $\phi$  at each sampling step and adds it to the global “coverage stack”. The process ends once  $\mathcal{Q}^V$  is covered.

## IV. APPROACH

The approach section details the implementation and design choice in terms of both the software implementation for autonomous large-scale exploration and the low-cost hardware design in respectively Sec. IV-A and IV-B.

### A. Autonomous large-scale exploration

There is a large body of work for robot exploration [15], [19], [21], [22], [45]. While the majority exploits the concept of frontiers [44], i.e., boundaries between known and unknown

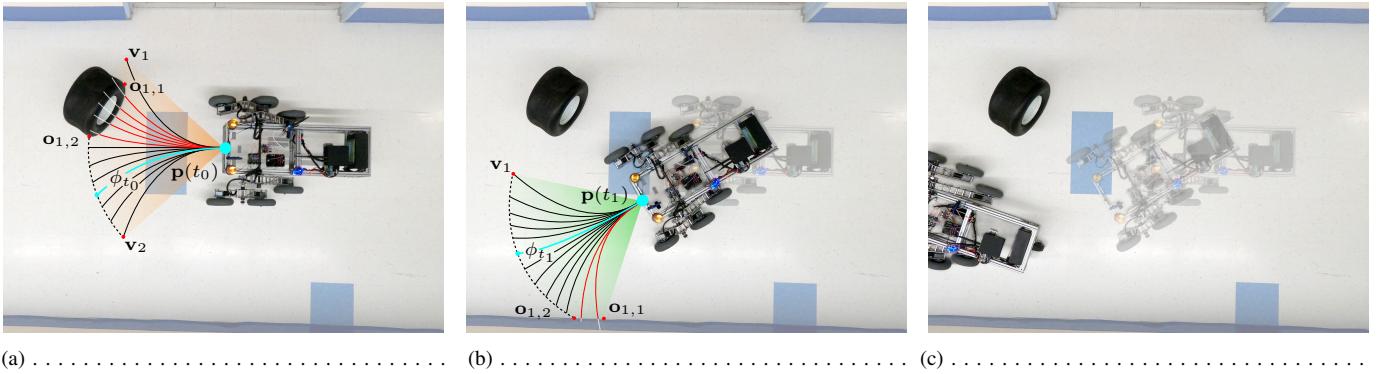


Fig. 2: . . . . .

space, mixed approaches are emerging [15], [43], [46]. Especially useful in the presence of diverse sensing modalities, e.g., involving raw sensory data, topologies, semantics, etc., they have multiple advantages for real-world environments [15], [42]. We propose a mixed approach for our large-scale exploration framework, combining frontier- and sampling-based methods, similar to some recent approaches [43], [44], [19].

The framework evaluates local frontiers at each step, samples the environment, and determines feasible candidate path functions  $\phi$  that intersect  $\mathcal{Q}^V$  (see Definition III.1). The next  $\phi$  is selected so that the frontier is the largest, but other costs are possible (see Sec. VI). The framework then derives a path-following vector field that points to  $\phi$  at any point and guides the robot utilizing the gradient descent algorithm. This allows the robot to, e.g., follow the covering path for longer and in real-time compared to approaches that utilize frontiers only, decreasing computational and cost requirements (see Sec. V).

To derive the path-following vector field, let the gradient of  $\phi$  be defined

$$\nabla\phi := \begin{bmatrix} \partial\phi(\mathbf{p})/\mathbf{p}_x \\ \partial\phi(\mathbf{p})/\mathbf{p}_y \end{bmatrix}, \quad (1)$$

where  $\partial\phi/\mathbf{p}$  is the differential, and  $\mathbf{p}_x, \mathbf{p}_y$  are the  $x$  and  $y$  components of  $\mathbf{p}$ . It points in the direction where  $\phi$  maximally locally increases. To assign the direction to each point, we use the construct of vector fields, which is common in other motion planning literature [38], [40], [45]

$$\Phi(t, \phi) := \bigcup_{\mathbf{p}(t) \in \mathcal{Q}} \nabla\phi(\mathbf{p}(t)). \quad (2)$$

We modify the vector field in Equation (2) to point to the contour of the path function  $\phi$  rather than its local maxima

$$\Delta\phi(\mathbf{p}(t)) := E\nabla\phi(\mathbf{p}(t)) - k_e\phi(\mathbf{p}(t))\nabla\phi(\mathbf{p}(t)), \quad (3)$$

where  $E\nabla\phi$  points perpendicularly to the gradient and  $\phi\nabla\phi$  to  $\phi$  at  $k_e \in \mathbb{R}_{>0}$  rate [40].  $E$  is the following direction, i.e.,

$$E = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (4)$$

is counterclockwise and  $-E$  clockwise directions [41].

Let thus the path-following equivalent of Eq. (2) be

$$\Phi_\phi(t, \phi) := \bigcup_{\mathbf{p}(t) \in \mathcal{Q}} \Delta\phi(\mathbf{p}(t)). \quad (5)$$

---

**Algorithm 1** Derivation of the exploration coverage  $\langle\phi, t\rangle$ 


---

```

1: for all  $t \in \mathcal{T}$  do
2:   if  $\mathcal{P} \cap \mathcal{Q} = \{\emptyset\}$  then return  $\langle\phi, t\rangle$ 
3:    $\mathcal{Q}_t^V := O_{1,t}, O_{2,t}, \dots, O_{n,t}, V_t \leftarrow$  sensor readings
4:   if  $\mathcal{Q}_t^V \neq \mathcal{Q}_{t-1}^V$  then
5:      $\{\phi_{1,t}, \phi_{2,t}, \dots\} \leftarrow \phi$ s in Def. III.1, inters.  $\mathcal{Q}^V \cap \Psi(\mathcal{Q}_t^V)$ 
6:     if  $\phi_t := \{\phi_{1,t}, \phi_{2,t}, \dots\} = \{\emptyset\}$  then the robot is stuck
7:     else
8:        $\phi_t \leftarrow \arg \max_\phi l(\phi_t, t, \mathcal{Q}_t^V)$  in Eq. (7)
9:        $\langle\phi, t\rangle \leftarrow \langle\phi, t\rangle \cup \langle\phi_t, t\rangle$  in Def. III.2
10:       $\mathcal{P} \leftarrow \mathcal{P} \cup \Psi(\mathcal{Q}_t^V)$ 
11:    end if
12:   end if
13:    $\varphi(t, \mathbf{p}(t)) \leftarrow \varphi(t-1, \mathbf{p}(t-1)) + \theta \Delta\phi(\mathbf{p}(t))$  in Eq. (3)
14: end for

```

---

The path-following vector field is summarized in the pseudo-code in Algorithm 1, with the gradient descent in Line 13. The vector  $\varphi \in \mathbb{R}^2$  points the robot in the direction of the path function  $\phi$  with a scalar step size  $\theta \in \mathbb{R}_{>0}$ . The algorithm runs at the highest frequency  $\mathcal{T} := \{t_0, t_0+h, \dots\}$  with a time-step  $h \in \mathbb{R}_{>0}$ . Practically, there might be different  $hs$  at different times (see Sec. V). In Line 2, the algorithm evaluates if the bounded volume  $\mathcal{Q}$  is covered utilizing the covered volume  $\mathcal{P} \subseteq \mathbb{R}^3$ . The latter is updated in Line 5, where the function  $\Psi : \mathbb{R}^{2n} \times \mathbb{R}^2 \rightarrow \mathbb{R}^{3n} \times \mathbb{R}^3$  maps the vertices to the volume. The vertices of  $\mathcal{Q}_t^V$  in Line 3 are derived from sensor readings, assuming the presence of a low-cost depth camera. The framework read the camera's point cloud, clustering the obstacles  $O_{1,t}, O_{2,t}, \dots$  by checking if the distance between consecutive points in space is within a given threshold  $\varepsilon \in \mathbb{R}_{>0}$  and deriving their vertices. The vertices of the space at time instant  $t$ ,  $V_t$  are simply the limits of the sensor's field of view.

The remaining lines compute the feasible path functions  $\{\phi_{1,t}, \phi_{2,t}, \dots\}$  by intersecting the space  $\Psi(\mathcal{Q}_t^V)$  with possible candidate trajectories that have their final points laying at the edges of  $\mathcal{Q}_t^V$ , i.e., splines of the form

$$a(x - \mathbf{p}_x)^3 + b(x - \mathbf{p}_x)^2 + c(x - \mathbf{p}_x) + d - y = 0, \quad (6)$$

where  $a, b, c \in \mathbb{R}$  are the coefficients of the spline. The best trajectory is then derived via the cost  $l$  in Line 8, utilizing the intersection of the largest frontier. For instance, if there are no

obstacles, Eq. (6–7) are such that  $\phi$  is a line parallel to the direction of the robot. Formally

$$l := \{ \| \mathbf{p}_1 - \mathbf{p}_2 \| \mid \exists \mathbf{p}_1, \mathbf{p}_2 \in \Psi(Q_t^V) \text{ s.t. } \mathbf{p}_1 \neq \mathbf{p}_2, \phi(\mathbf{p}_1 - \mathbf{p}_2) \approx 0 \}, \quad (7)$$

where the condition  $\phi(\mathbf{p}_1 - \mathbf{p}_2)$  is evaluated on a given  $\varepsilon \in \mathbb{R}_{>0}$ , i.e.,  $|\phi(\mathbf{p}_1 - \mathbf{p}_2)| < \varepsilon$ .

Using the algorithm, the framework provides a way to explore space  $\mathcal{Q}$  and avoid obstacles  $\mathcal{Q}^{O_i}$ . There are configurations at which there are no feasible trajectories nonetheless, e.g., if  $\{\phi_{1,t}, \phi_{2,t}, \dots\} = \{\emptyset\}$  in Line 6. In this scenario, the framework allows a human to intervene via standard wireless and LoRa communication technology. The robot can then be teleoperated on long distances—studies from the internet-of-things domain [32], [47] report a range of up to five kilometers in an urban setting—and with a relatively inexpensive hardware equipment (two LoRa bundles). The framework we propose utilizes a web interface to parse human commands into our custom communication protocol which utilizes the LoRa physical layer’s payload to transfer  $\varphi$ ’s  $x$  and  $y$  components.

The algorithm is illustrated in Fig. 2. At each iteration, the robot samples the environment and derives a set of possible candidate trajectories  $\{\phi_{1,t}, \phi_{2,t}, \dots\}$ . If there is no obstacle ahead, the optimal trajectory per iteration  $\phi_t$  is a line parallel to the robot’s direction of travel (see Fig. 2c). If there are obstacles, the framework selects the trajectory via the cost  $l$ ,  $\phi_t$ , which goes through the middle of the largest frontier (see Fig. 2a and 2b for respective obstacles “wheel” and “wall”).

To derive a map of the environment and to keep the track of the robot within it in Line 13, the framework uses a state-of-the-art visual SLAM algorithm from the literature [37]. The robot’s location is also used to determine whether the exploration is complete in Line 2, showing that the algorithm is effective in exploring unknown environments with a lower sensory footprint (see Sec. V). Furthermore, earlier iteration of the work exploited a different SLAM algorithm from the visual SLAM community [48], showing that some of the framework components are interchangeable.

The framework is distributed under the popular open-source CC BY-NC-SA license<sup>1</sup>. It is composed of three distinct components. A “ground robot” ROS2 [33] package implements the communication with a base station using either the IEEE 802.11 wireless communication or long-range LoRa protocols. The package further implements the serial communication with the microcontroller implemented in Arduino and the vertices detection (see Algorithm 1). A “ground navigation” ROS package collects point clouds from an RGB-D sensor (an Intel (R) RealSense (TM) Depth Camera [49] D435) and other data from the SLAM algorithm [37] and ports them into ROS2. Finally, a “base server” implements the necessary functionality for remote human intervention. Both “ground robot” and “ground navigation” are implemented in C++ in ROS2 and ROS respectively, whereas “base station” is in PHP and JavaScript.

### B. Low-cost hardware design

The RB5 mobile robot adopts a rocker-bogie suspension system [50] found on NASA’s rovers including Sojourner

and Curiosity. On either side of the robot, an upside-down V-shaped linkage called the rocker pivots about an axis on the robot frame. The rocker has a wheel at one end and a smaller V-shaped linkage on the other arm. The smaller linkage, called the bogie, can pivot about an axis on the rocker and has two wheels at its tips. The articulated nature of the rocker-bogie suspension allows the mobile robot to adapt to uneven terrains [24], [35], [36] as the rocker and bogie pivot to maintain wheel contact [35]. Each of the six wheels in the rocker-bogie suspension is actuated by a DC gear motor, whereas the rotational degrees of freedom in the rocker-bogie suspensions are passive. Since the wheels are all parallel and cannot rotate out of the plane, the robot uses the same actuation strategy as that of a differential drive vehicle to move straight and make turns by controlling the left and right sets of wheels in the same and opposite directions. Given that RB5 has multiple wheels on each side, its ability to make turns is reduced compared to a differential drive vehicle. Due to its extended body length, RB5 incorporates a caster wheel in the back to support the rear end of the frame.

The robot frame’s dimensions are 914 by 330 millimeters, and the robot’s bounding box dimensions are 991 by 762 mm. The frame consists of one inch aluminum extrusions and acrylic sheets, and the rocker and bogie linkages are assembled from aluminum sheets and standoffs. The pivots of the bogie and rocker sit at 240 and 330 mm from the ground respectively, providing a clearance of approximately 190 mm beneath the robot frame. The two wheels on each bogie linkage are coplanar, but the wheel on the corresponding rocker linkage is closer to the medial plane of the robot. Motor control is performed by a Teensy (R) 4.0 microcontroller board sending PWM commands to six DRV8871 motor driver boards. An onboard 24 volts LiFePO<sub>4</sub> battery provides power for the logic boards and motor drives.

## V. FIELD EXPERIMENTS

Field experiments involving RB5 in autonomous large-scale exploration are conducted in a variety of environments, including indoors structured, unstructured underground, and outdoors. In each, the microcontroller executes a finite set of motion primitives via velocity control. These primitives are transmitted serially to the microcontroller from RB5’s onboard computing hardware, an NVIDIA (R) Jetson NX (TM) embedded board, which implements the autonomous large-scale exploration framework. The computing hardware mounts peripherals for sensing the environment and for communication. The former group consists of a low-cost upward-facing RGB camera for detections (see Sec. VI) and the RGB-D sensor (see Sec. IV-A), whereas the latter of a LoRa wireless network bundle with the RN2903 module of the Intel (R) AX200 network card for standard wireless communication via 802.11 protocol when, e.g., RB5 is in reach of an available wireless network. All the components of the framework (detailed at the end of Sec. IV-A) run in real-time onboard RB5, and some additional processing, such as the derivation of the 3D reconstructions in the supplementary material, is carried on an external device connected to RB5’s computing hardware via ROS network.

<sup>1</sup>[github.com/adamseew/ytcg\\_ground-based](https://github.com/adamseew/ytcg_ground-based)

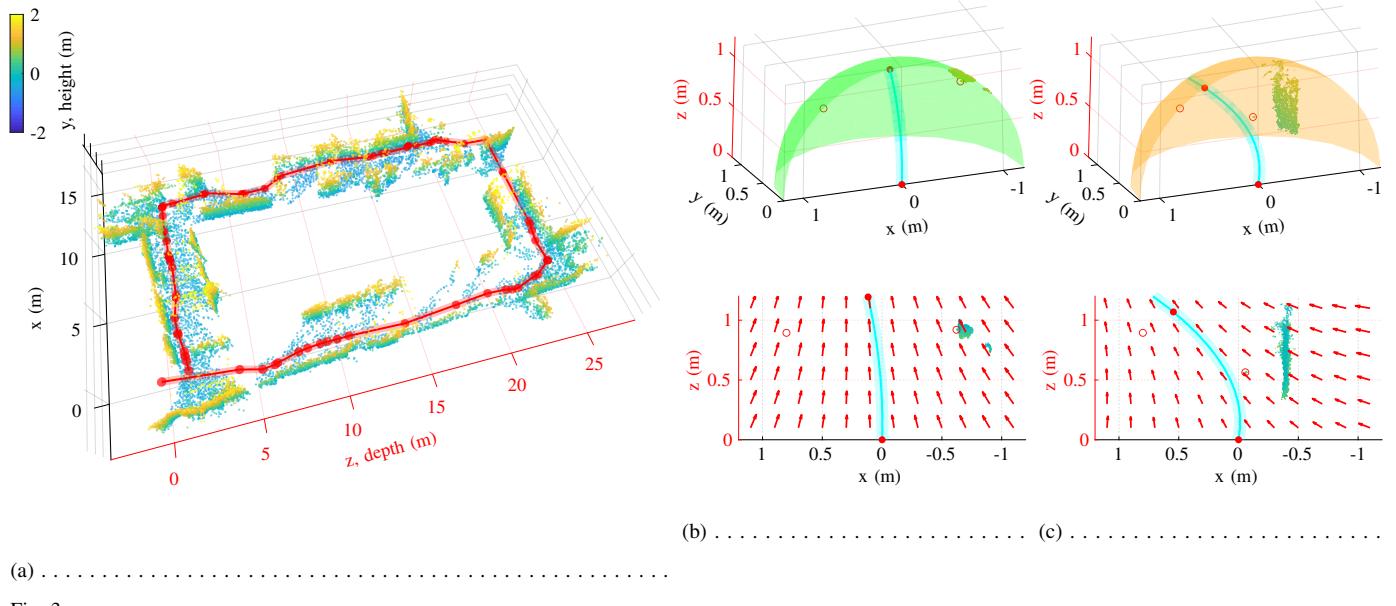
Fig. 3 shows experimental results for a structured indoor environment, a university hall, located on the second floor of a multistorey building. The hall is composed of four connected corridors for a total approximate length of eighty meters in a closed circuit, i.e., the initial and final points coincide. The resulting point cloud is shown in Fig. 3a, where the color scheme in the top-left indicates the different heights of points in the point cloud. The Low-cost components mitigation of the large-scale exploration framework in Sec. IV-A is to be observed in the figure, where between, e.g., fifteen and twenty meters on the z-axis and zero and five meters at the x-axis, there are significantly fewer points in the point cloud than in other parts of the figure. The algorithm here keeps track (see Line 13) of the path function  $\phi_t$  (see Line 8) in the event of, e.g., the computing hardware being busy while executing other tasks such as communication. While specific to the computing hardware onboard RB5, the occurrence and the unpredictable nature of the execution is common in the literature, especially if involving heterogeneous elements, i.e., CPU, GPU, and microcontrollers [51].

Fig. 3b–3c shows the detail of the algorithm in the experiment in terms of obstacle detection and avoidance. Here RB5 detects an obstacle, a door with a surrounding wall, as it cruises through the hall at approximately fifteen and zero on the respective z- and x-axis. Fig. 3b shows the initial detection of the obstacle on top. The vertices  $V, O_i$  are the empty red circles and represent the field of view on the left and the edge of the obstacle on the right. On the bottom is the path-following vector field from Eq. (3) in red and the path function  $\phi_t$  in cyan. Fig. 3c shows the following time step as RB5 comes closer, and the robot has to perform a sharper maneuver to avoid the obstacle.

Fig. 4 shows an unstructured environment of a hall connecting to an underground tunnel on the respective left and right sides of Fig. 4a for approximately one hundred meters. Conversely to the experiment in Fig. 3, the experiment showcases an open circuit, in the sense that the exploration is considered concluded when a specific frontier from the initial frontier is encountered. Fig. 4b–4c shows the obstacle detection, similar to Fig. 3b–3c, for a wheel placed close to the left edge of the first length of the figure wide approximately 0.42 meters. The generated path with the robot avoiding the obstacle is to be observed between fifteen and twenty meters on the z-axis where in Fig. 4.

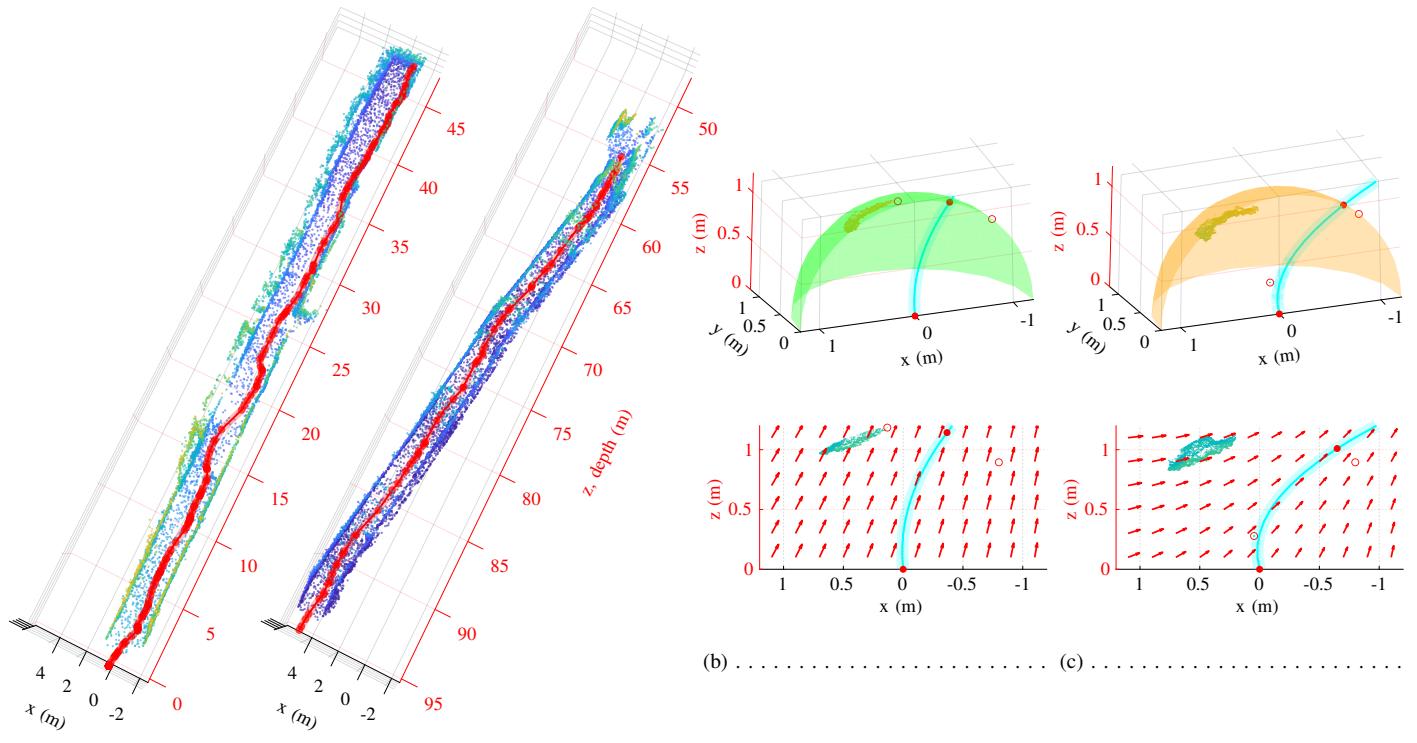
The turning direction  $E$  in Eq. (3) is positive for left turns (see Fig. 3b) and negative for right turns (see Fig. 4c–4b). The turning rate  $k_e$  is derived empirically similar to other literature [39], [40], and is 0.05, 0.1, and 0.4 depending on the turning maneuver, i.e., it is 0.05 when  $\phi_t$  is a line (or close to it), 0.4 when a sharp curve in respectively Fig. 3b and Fig. 4c. The points in the point cloud are adjusted for height and length and filtered for visualization purposes, i.e., we have reported one point every two hundred and fifty, five hundred, etc., in Fig. 3a and 4a.

## VI. CONCLUSION AND FUTURE DIRECTIONS



(a) . . . . .

Fig. 3: . . . . .



(a) . . . . .

Fig. 4: . . . . .

## REFERENCES

- [1] S. Kohlbrecher, J. Meyer, T. Gruber *et al.*, “Hector open source modules for autonomous mapping and navigation with rescue robots,” in *RoboCup 2013: Robot World Cup XVII*. Springer, pp. 624–631. 1, 2
- [2] M. Kulkarni, M. Dharmadhikari, M. Tranzatto *et al.*, “Autonomous teamed exploration of subterranean environments using legged and aerial robots,” in *International Conference on Robotics and Automation (ICRA’22)*. IEEE, 2022, pp. 3306–3313. 1, 2
- [3] M. Tranzatto, F. Mascarich, L. Bernreiter *et al.*, “CERBERUS: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA Subterranean Challenge,” *Field Robotics*, vol. 2, pp. 274–324, 2022. 1, 2
- [4] H. Kim, H. Kim, S. Lee *et al.*, “Autonomous exploration in a cluttered environment for a mobile robot with 2d-map segmentation and object detection,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6343–6350, 2022. 1, 2
- [5] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 22, 2019. 1
- [6] T. Miki, J. Lee, J. Hwangbo *et al.*, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Science Robotics*, vol. 7, no. 62, p. 14, 2022. 1
- [7] T. Rouček, M. Pecka, P. Čížek *et al.*, “DARPA subterranean challenge: Multi-robotic exploration of underground environments,” in *Modelling and Simulation for Autonomous Systems*. Springer, 2020, pp. 274–290. 1, 2
- [8] W. Tabib, K. Goel, J. Yao *et al.*, “Autonomous cave surveying with an aerial robot,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1016–1032, 2022. 1, 2
- [9] K. Ebadi, Y. Chang, M. Palieri *et al.*, “LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments,” in *International Conference on Robotics and Automation (ICRA’20)*. IEEE, 2020, pp. 80–86. 1, 2
- [10] Y. Mei, Y.-H. Lu, C. Lee *et al.*, “Energy-efficient mobile robot exploration,” in *International Conference on Robotics and Automation (ICRA’06)*. IEEE, 2006, pp. 505–511. 1, 2
- [11] R. Shrestha, F.-P. Tian, W. Feng *et al.*, “Learned map prediction for enhanced mobile robot exploration,” in *International Conference on Robotics and Automation (ICRA’19)*. IEEE, 2019, pp. 1197–1204. 1, 2
- [12] A. Eldemiry, Y. Zou, Y. Li *et al.*, “Autonomous exploration of unknown indoor environments for high-quality mapping using feature-based RGB-D SLAM,” *Sensors*, vol. 22, no. 14, p. 16, 2022. 1, 2
- [13] M. B. Alatise and G. P. Hancke, “A review on challenges of autonomous mobile robot and sensor fusion methods,” *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020. 1
- [14] I. Lluvia, E. Lazcano, and A. Ansustegui, “Active mapping and robot exploration: A survey,” *Sensors*, vol. 21, no. 7, 2021. 1, 2
- [15] J. A. Placed, J. Strader, H. Carrillo *et al.*, “A survey on active simultaneous localization and mapping: State of the art and new frontiers,” *IEEE Transactions on Robotics*, p. 20, 2023, doi.org/10.48550/arXiv.2207.00254. 1, 2, 3
- [16] A. Bircher, M. Kamel, K. Alexis *et al.*, “Receding horizon “next-best-view” planner for 3D exploration,” in *International Conference on Robotics and Automation (ICRA’16)*. IEEE, 2016, pp. 1462–1468. 1, 2
- [17] M. Müller and V. Koltun, “OpenBot: Turning smartphones into robots,” in *International Conference on Robotics and Automation (ICRA’21)*. IEEE, 2021, pp. 9305–9311. 1, 2
- [18] D. Tardioli, L. Riazuelo, D. Sicignano *et al.*, “Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments,” *Journal of Field Robotics*, vol. 36, no. 6, pp. 1074–1101, 2019. 1, 2
- [19] T. Dang, F. Mascarich, S. Khattak *et al.*, “Graph-based path planning for autonomous robotic exploration in subterranean environments,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3105–3112. 1, 2, 3
- [20] H. Surmann, A. Nüchter, and J. Hertzberg, “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments,” *Robotics and Autonomous Systems*, vol. 45, no. 3, pp. 181–198, 2003. 1, 2
- [21] M. Juliá, A. Gil, and O. Reinoso, “A comparison of path planning strategies for autonomous exploration and mapping of unknown environments,” *Autonomous Robots*, vol. 33, pp. 427–444, 2012. 1, 2
- [22] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. ‘Towards New Computational Principles for Robotics and Automation’*, 1997, pp. 146–151. 1, 2
- [23] B. Zhou, Z. Wu, and X. Liu, “Smartphone-based robot indoor localization using inertial sensors, encoder and map matching,” in *International Conference on Automation, Control and Robots (ICACR’21)*. IEEE, 2021, pp. 145–149. 1, 2
- [24] S. M. F. Faisal, T. Rahman, and M. A. Kabir, “A low-cost rough terrain explorer robot fabrication using rocker bogie mechanism,” in *International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2’21)*, 2021, pp. 1–4. 1, 2, 4
- [25] K. Ismail, R. Liu, J. Zheng *et al.*, “Mobile robot localization based on low-cost LTE and odometry in GPS-denied outdoor environment,” in *International Conference on Robotics and Biomimetics (ROBIO’19)*. IEEE, 2019, pp. 2338–2343. 1
- [26] “ROS-based unmanned mobile robot platform for agriculture,” *Applied Sciences*, vol. 12, no. 9, p. 13, 2022. 1
- [27] F. Voigtlander, A. Ramadan, J. Eichinger *et al.*, “5G for robotics: Ultra-low latency control of distributed robotic systems,” in *International Symposium on Computer Science and Intelligent Controls (ISCSIC’17)*. IEEE, 2017, pp. 69–72. 1
- [28] C. Delgado, L. Zanzi, X. Li *et al.*, “OROS: Orchestrating ROS-driven collaborative connected robots in mission-critical operations,” in *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM’22)*. IEEE, 2022, pp. 147–156. 1
- [29] C. Cadena, L. Carlone, H. Carrillo *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016. 1
- [30] M. Corah, C. O’Meadhra, K. Goel *et al.*, “Communication-efficient planning and mapping for multi-robot exploration in large environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019. 1, 2
- [31] C. Papachristos, S. Khattak, and K. Alexis, “Uncertainty-aware receding horizon exploration and mapping using aerial robots,” in *International Conference on Robotics and Automation (ICRA’17)*. IEEE, 2017, pp. 4568–4575. 1, 2
- [32] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao, “A survey on LoRa networking: Research problems, current solutions, and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2020. 1, 4
- [33] M. Quigley, K. Conley, B. Gerkey *et al.*, “ROS: An open-source robot operating system,” in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009, p. 5. 1, 4
- [34] T. P. Setterfield and A. Ellery, “Terrain response estimation using an instrumented rocker-bogie mobility system,” *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 172–188, 2013. 2
- [35] M. Mann and Z. Shiller, “Dynamic stability of a rocker bogie vehicle: Longitudinal motion,” in *International Conference on Robotics and Automation (ICRA’05)*, 2005, pp. 861–866. 2, 4
- [36] D. Kim, H. Hong, H. S. Kim *et al.*, “Optimal design and kinetic analysis of a stair-climbing mobile robot with rocker-bogie mechanism,” *Mechanism and Machine Theory*, vol. 50, pp. 90–108, 2012. 2, 4
- [37] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. 2, 4
- [38] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia *et al.*, “Vector fields for robot navigation along time-varying curves in  $n$ -dimensions,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010. 2, 3
- [39] A. Seewald, H. García de Marina, H. S. Midtiby *et al.*, “Energy-aware planning-scheduling for autonomous aerial robots,” in *International Conference on Intelligent Robots and Systems (IROS’22)*. IEEE, 2022, pp. 2946–2953. 2, 5
- [40] H. García de Marina, Y. A. Kapitanyuk, M. Bronz *et al.*, “Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows,” in *International Conference on Robotics and Automation (ICRA’17)*. IEEE, 2017, pp. 5740–5745. 2, 3, 5
- [41] A. Seewald, “Energy-aware coverage planning and scheduling for autonomous aerial robots,” Ph.D. thesis, Syddansk Universitet, 2021, doi.org/10.21996/7ka6-r457. 2, 3
- [42] A. Batinovic, T. Petrovic, A. Ivanovic *et al.*, “A multi-resolution frontier-based planner for autonomous 3d exploration,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4528–4535, 2021. 2, 3

- [43] A. Dai, S. Papatheodorou, N. Funk *et al.*, “Fast frontier-based information-driven autonomous exploration with an MAV,” in *International Conference on Robotics and Automation (ICRA’20)*. IEEE, 2020, pp. 9570–9576. [2](#), [3](#)
- [44] W. Qiao, Z. Fang, and B. Si, “A sampling-based multi-tree fusion algorithm for frontier detection,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 14, 2019. [2](#), [3](#)
- [45] S. M. LaValle, *Planning algorithms*. Cambridge Univ. Press, 2006. [2](#), [3](#)
- [46] L. Schmid, M. Pantic, R. Khanna *et al.*, “An efficient sampling-based method for online informative path planning in unknown environments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020. [3](#)
- [47] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. [4](#)
- [48] C. Campos, R. Elvira, J. J. G. Rodríguez *et al.*, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. [4](#)
- [49] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen *et al.*, “Intel RealSense stereoscopic depth cameras,” in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW’17)*. IEEE, 2017, pp. 1267–1276. [4](#)
- [50] D. B. Bickler, “Articulated suspension system,” 1989, U.S. Patent № 4840394. [4](#)
- [51] A. Seewald, U. P. Schultz, E. Ebeid *et al.*, “Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems,” *International Journal of Parallel Programming*, vol. 49, no. 2, pp. 136–157, 2021. [5](#)