A Low-Cost Energy-Efficient Approach for Long-Term Autonomous Exploration

Adam Seewald[1], Marvin Chancán[1], Connor M. McCann[2], Seonghoon Noh[1], Omeed Fallahi[1], Hector Castillo[1], Ian Abraham[1], and Aaron M. Dollar[1]

***Abstract*—This paper tackles autonomous long-term exploration for robots with a lower sensory footprint. Requiring just an RGB-D camera and low-power computing hardware, the approach uses an experimental wheeled robot with rocker-bogie suspension. It operates in unknown and GPS-denied environments, and on indoor and outdoor challenging terrains. The exploration path uses a novel methodology that extends frontier- and sampling-based exploration with a path-following vector field and utilizes a state-of-the-art SLAM to derive the position. The approach allows the robot to explore its surroundings at lower update frequencies, utilizing cheaper hardware compared to the state-of-the-art, and it is generic in terms of adaptation to other mobile robots with similar sensing. The approach further consists of a methodology to interact with a remotely located human operator based on an inexpensive long-range and low-power communication technology from the internet-of-things domain (i.e., LoRa) and a customized communication protocol. Results show improved long-term performance in indoor and outdoor environments despite low-cost hardware over the baseline of existing approaches.**

***Index Terms*—Motion and Path Planning; Energy and Environment-Aware Autonomy.**

I. Introduction

THE promise of long-term autonomous robotic exploration is currently being held up in part by the expense of the required sensing, computing, and mechanical hardware. This cost is related to the computational intensity of most common navigation and communication approaches [1, 2], which significantly increases for challenging outdoor terrains. Addressing this challenge, we introduce novel techniques to reduce update frequencies and enhance the communication capabilities of existing approaches. By loosening the required update frequencies and communication requirements, our methods enable the use of lower-performing and lower-cost hardware while still retaining good autonomous performance.

Recent efforts in this direction include low-cost robots for exploration but lack terrain adaptability [3] and capabilities often required to navigate outdoors in the real world [4, 5]. Furthermore, in areas that are challenging to traverse, state-of-the-art approaches rely on humans for supervision and high-level decision-making [6–8]. As a result, robots often operate close to humans or require expensive network equipment, such as a mesh of communication devices [9, 10], or existing network infrastructure [11, 12], thereby restricting autonomous exploration to indoor settings only [13–16].

Fig. 1: A robot needs to explore its surroundings with a lower sensory footprint–the picture illustrates an experimental robotic platform that carries an RGB-D camera and low-power computing hardware to derive an exploratory coverage path for long-term, and on challenging terrains.

Our methodology exploits LoRa–an inexpensive long-range and low-power communication technology [17] 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.

For visual sensing, our approach maintains a low sensory

[1]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;

[2]C. M. C. is with the School of Engineering and Applied Sciences, Harvard University, MA, USA, but the work was performed while affiliated with Yale.

footprint with low-cost components compared to state-of-the-art–an RGB depth (RGB-D) camera to sense the environment–and operates in a variety of terrains. Most approaches tackling autonomous exploration use expensive equipment such as LiDARs [6–10, 18–20] and laser range finders [21, 22] instead. Even though approaches that utilize cheaper sensors, such as RGB-D cameras [23, 24], RGB cameras [19], and sonars [3, 4], are studied, they often operate along other more expensive equipment [19] or indoors only [23], and have limited autonomy [24] or obstacle avoidance features [3, 4]. Recent approaches minimize exploration costs nonetheless by, e.g., exploiting sensing capabilities of commercial smartphones [3, 4] or using use case-specific aspects [8] but are generally unable to operate in a wide variety of challenging environments for long-term [15, 23]. Software-wise, recent efforts into autonomous exploration require expensive prior learning [25] or run on multiple robots [6, 7, 9], whereas approaches with little computing resources are scarce [3, 5, 20, 23]. More traditional approaches such as those based on frontiers [7, 20, 21], graphs [6, 9, 19], grids [8, 16], and random trees are also studied, but mixed approaches are preferred [22–26] to maximize performance and resources [2, 23]. Similarly, our methodology is based on a mixed approach. A frontier- and sampling-based method that exploits the scarcity of resources while still running with comparable autonomy and obstacle avoidance features to its more expensive counterparts.

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 [27] and the exploration trajectory with a novel methodology that extends exploration literature with a path-following vector field from the aerial robotics domain [28–30]. This allows the robot to explore its surroundings for longer and at lower update frequencies, utilizing cheaper computing hardware compared to the state-of-the-art (see Fig. 3).

Utilizing these components along with the open-source robot operating system (ROS) middleware, we demonstrate a low-cost exploration approach using an experimental robotic platform–RB5 in Figure 1, a wheeled mobile robot with rocker-bogie suspension–capable of exploring autonomously dynamic indoor and outdoor environments. Comparable platforms in the literature comprise two degrees of freedom suspension with pivots [5, 31] and provide rough terrain static adaptability. They are cheaper than, e.g., legged robots in terms of the cost of sensors and operation, as they are able to overcome obstacles without costly computations for gait adaptation and planning [3]. Although we implement out approach on the RB5 rover, the approach is generic in terms of portability to other mobile robots with cost and computational constraints.

The main **contributions** of this paper are *(a)* an approach for *long-term autonomous exploration* that exploits *energy–efficient design* and *low-power communication* and *(b)* an extension of existing approaches to *mobile robots with fewer and cheaper sensors*. We demonstrate the exploration performance and obstacle avoidance performance of our approach over the baseline of existing autonomous exploration system-of-systems with a set of indoor and outdoor "in the field" experiments (Section IV). The remainder of the paper is structured as follows. Sec. III describes the approach from the software and hardware standpoints. Sec. II formalizes the problem of autonomous exploration and Sec. V drafts conclusions and future directions.

II. Problem Description

The problem considered in this work is that of exploring an unknown bounded space, i.e., visiting every point in within. The robot is free to move except for some possible obstacles. We define in this section the problem with mathematical rigor to later derive a technique to solve it at lower update frequencies and thus for longer than the state-of-the-art. Formally, the problem 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 of positive naturals up to $n \in \mathbb{N}_{>0}$ and $[n]_{>0}$ of strictly positive naturals, we are interested in collision-free trajectories that explore $\mathcal{Q}$ and avoid n obstacles $\mathcal{Q}^{O_i} \subset \mathbb{R}^3, i \in [n]_{>0}$. 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 $i \in [n]_{>0}, \mathbf{v}_j, \mathbf{o}_{i,k} \in \mathbb{R}^2$, $\forall j \in [|V|]$, $k \in [|O_i|]$ a point w.r.t. $\mathcal{O}_W$. Let V enclose $\mathcal{Q}$, O_i $\mathcal{Q}^{O_i}$ per each i. The *exploration problem* is the problem of finding the coverage that visits every 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 ϕ be a path function, i.e., a function the robot tracks as it explores its surroundings in $\mathcal{Q}^V$, avoiding obstacles $\mathcal{Q}^{O_i}$.

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

Definition II.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.

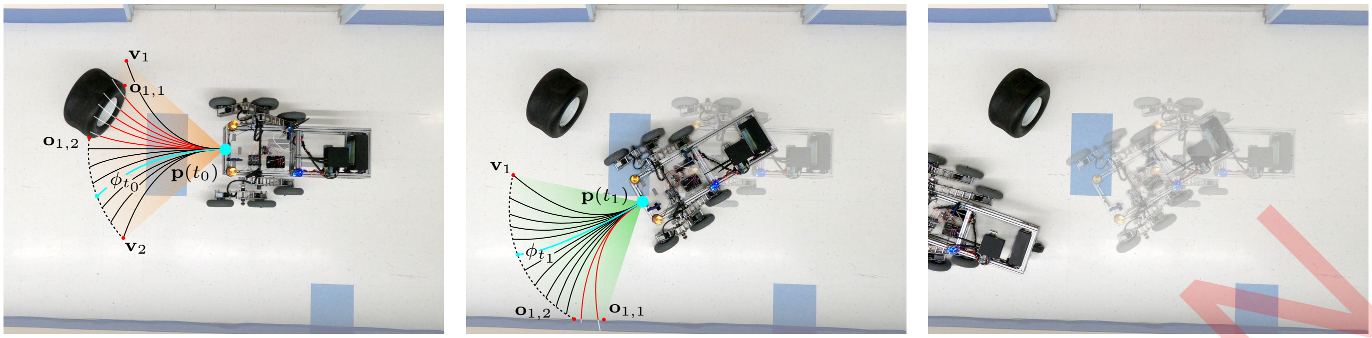
The exploration approach (see Sec. III-A) derives ϕ at each time step and adds it to the global "coverage stack". The process ends once $\mathcal{Q}^V$ is covered.

III. Approach

In this section, we detail the implementation and design choices in terms of the software for autonomous long-term exploration and the low-cost hardware in respectively Sec. III-A and III-B.

A. Autonomous exploration

While the majority of work on robot exploration exploits the concept of frontiers, i.e., boundaries between known and unknown space [2, 19], mixed approaches are emerging [2, 24, 32]. Especially useful in the presence of diverse



(a) Initial detection of an obstacle "wheel" with ϕ_{t_0} selected so that it avoids the obstacle.

(b) The robot continues to track ϕ_{t_0} up to the next iteration. Here it finds a new trajectory ϕ_{t_1}.

(c) The process continues up to when the entire space is explored.

Fig. 2: The autonomous long-term exploration approach consists of the robot sampling the environment and searching for obstacles and unexplored areas. The approach clusters the two groups into vertices sets and builds candidate path functions. From these, it selects the optimal trajectory w.r.t. a given cost and iterates the operation at each step. In between the iterations, it tracks the trajectory, saving computational and sensing resources.

sensing modalities, e.g., involving raw sensory data, topologies, semantics, etc., they have multiple advantages for real-world environments [2, 20]. We propose a mixed approach for our exploration, combining frontier- and sampling-based methods, similar to some recent approaches [24, 25].

Our approach evaluates local frontiers at each step, samples the environment, and determines feasible candidate path functions ϕ that intersect $\mathcal{Q}^V$ (see Definition II.1). The next ϕ is selected so that the frontier is the largest, but other costs are possible (see Sec. V). The approach then derives a path-following vector field that points to ϕ 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. IV).

To derive the path-following vector field, let the gradient of ϕ be defined

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

where $\partial\phi/\mathbf{p}$ is the partial differential, and $\mathbf{p}_x$, $\mathbf{p}_y$ are the x and y components of $\mathbf{p}$. It points in the direction where ϕ 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 [29, 33]

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

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

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

where $E\nabla\phi$ points perpendicularly to the gradient and $\phi\nabla\phi$ to ϕ at $k_e \in \mathbb{R}_{>0}$ rate [29]. E is the following direction, i.e.,

$$E = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{4}$$

is counterclockwise and $-E$ clockwise directions [30].

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

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

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

1: **for all** $t \in \mathcal{T}$ **do**
2:   **if** $\mathcal{P} \cap \mathcal{Q} = \{\varnothing\}$ **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. II.1, inters. $\mathcal{Q}^V \cap \Psi(\mathcal{Q}_t^V)$
6:     **if** $\phi_t := \{\phi_{1,t}, \phi_{2,t}, \dots\} = \{\varnothing\}$ **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. II.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 ϕ 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. IV). In Line 2, the algorithm evaluates if the bounded volume $\mathcal{Q}$ is covered utilizing $\mathcal{P} \subseteq \mathbb{R}^3$ updated in Line 5, where the function $\Psi : \mathbb{R}^{2n} \times \mathbb{R}^2 \to \mathbb{R}^{3n} \times \mathbb{R}^3$ maps the vertices to the volume. The vertices of the local free space $\mathcal{Q}_t^V$ in Line 3 are derived from sensor readings, assuming the presence of an RGB-D camera. The approach 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 $\varepsilon \in \mathbb{R}_{>0}$. The vertices of the free 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 local free 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

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

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. Formally

$$
\begin{aligned}
l := \{\|\mathbf{p}_1 - \mathbf{p}_2\| \mid \exists\, \mathbf{p}_1, \mathbf{p}_2 \in \Psi(\mathcal{Q}_t^V), i \in [|\phi_t|] \\
\text{s.t. } \mathbf{p}_1 \neq \mathbf{p}_2, \phi_{i,t}(\mathbf{p}_1 - \mathbf{p}_2) \trianglelefteq 0\},
\end{aligned} \tag{7}
$$

where the operator $\trianglelefteq$ evaluates $\phi(\mathbf{p}_1 - \mathbf{p}_2)$ on a given $\varepsilon \in \mathbb{R}_{>0}$, i.e., $|\phi_{i,t}(\mathbf{p}_1 - \mathbf{p}_2)| \leq \varepsilon$ and in such a way that the middle path functions of the largest subset of the contiguous path functions are selected preferably, e.g., if the largest subset is $\{\phi_{1,t}, \phi_{2,t}, \dots, \phi_{5,t}\}$, $\phi_{3,t}$ is selected. In this way, if there are no obstacles, Eq. (6–7) are such that ϕ is a line parallel to the direction of the robot.

Using the algorithm, the approach 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 $\{\phi_{1,t}, \phi_{2,t}, \dots\} = \{\varnothing\}$ in Line 6. In this scenario, the platform 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 [17] report a range of up to five kilometers in an urban setting–and with relatively inexpensive hardware equipment (two LoRa bundles). The approach 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 φ'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 path functions $\{\phi_{1,t}, \phi_{2,t}, \dots\}$. If there is no obstacle ahead, the optimal function per iteration ϕ_t is a line parallel to the robot's direction of travel (see Fig. 2c). If there are obstacles, the approach selects the trajectory via the cost l, ϕ_t, which goes through the middle of the largest frontier (see Fig. 2a and 2b for 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 approach uses a state-of-the-art visual SLAM algorithm from the literature [27]. 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. IV). Furthermore, an earlier iteration of the work exploited a different SLAM algorithm from the visual SLAM community [34], showing that some of the approach components are interchangeable.

The platform is distributed under the popular open-source CC BY-NC-SA license[1]. It is composed of three distinct components. A "ground robot" ROS2 package implements the communication with a base station using either the IEEE 802.11 wireless communication or long-range LoRa technology. The package further implements 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 camera (an Intel (R) RealSense (TM) Depth Camera D435) and other data from the SLAM algorithm [27] 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.

[1] github.com/adamseew/ytcg_ground-based

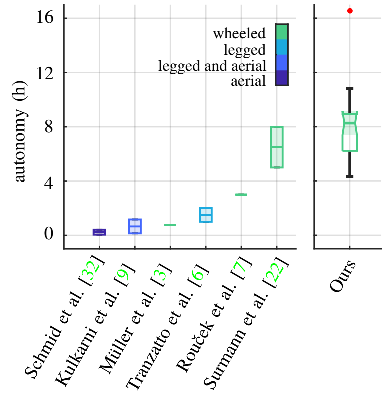


Fig. 3: Autonomy is reported in hours between the time the battery is fully charged to discharged for ours against other approaches tackling autonomous exploration. Usually, approaches that use aerial [9, 32] and wheeled [3, 7, 22] robots report respectively lowest and highest autonomy, whereas those that use legged [6, 9] robots are between the two groups. Here [3] is an outlier as it uses a small wheeled robot. We have conducted multiple trials under varying grounds and velocities. The minimum and maximum are approximately four hours and twenty minutes and sixteen hours and a half when the robot respectively moves at full speed and is not moving (red outlier). The first quartile is six hours and ten minutes, the third is nine hours. The median is then eight hours and twenty minutes when the average velocity is two-thirds of the maximum.

B. Low-cost rover design

The RB5 experimental robotic platform in this paper adopts a rocker-bogie suspension system [35] 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 [5] as the rocker and bogie pivot to maintain wheel contact.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.

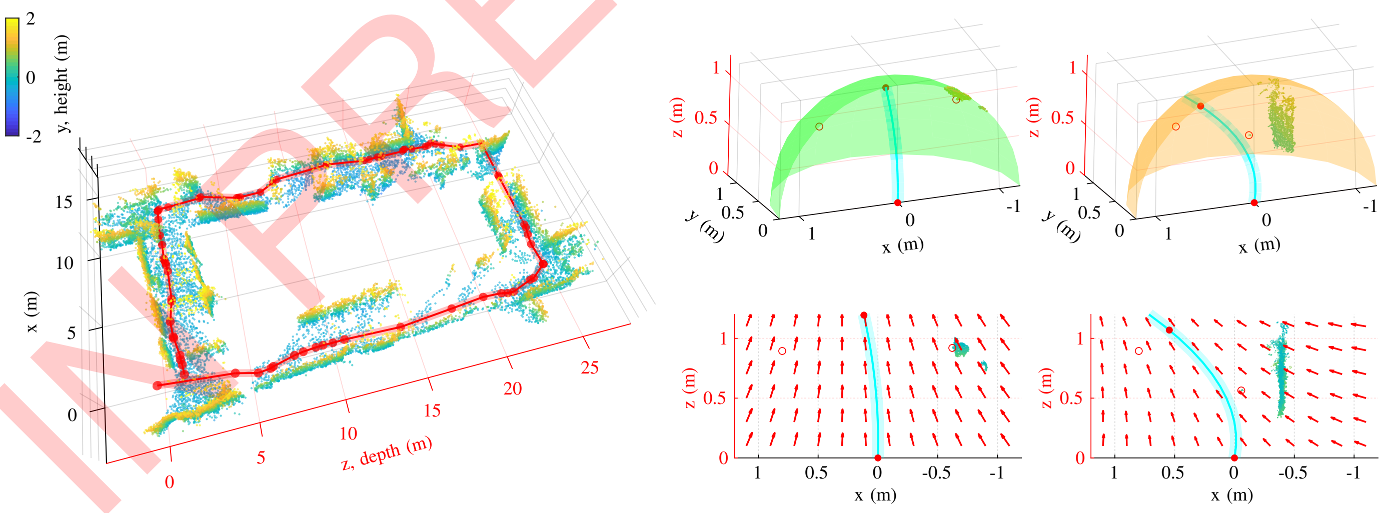
The robot frame's dimensions are 914 by 330 millimeters, and the robot's bounding box dimensions are 991 by 762 mm. The frame consists of one inch aluminum extrusions and acrylic sheets, and the rocker and bogie linkages are assembled from aluminum sheets and standoffs. The pivots of the bogie and rocker 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 $LiFePO_4$ battery provides power for the microcontroller, motor drives, and computing hardware.

IV. Field Experiments

In order to demonstrate the effectiveness of our approach, we conduct a set of field experiments in autonomous long-term exploration involving our RB5 experimental robotic platform 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 our autonomous exploration. 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. V) and the RGB-D camera (see Sec. III-A). The latter consists of a LoRa wireless network bundle with the RN2903 module and an 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 (detailed at the end of Sec. III-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.

Fig. 3 compares our hardware approach to others. "Autonomy" is reported in hours between the time the battery is fully charged to discharged–the time when the robot can actively explore its surroundings–and is compared to representative approaches in the literature tackling autonomous exploration [3, 6, 7, 9, 22, 32]. Our approach is similar to the most efficient currently in the list [22] but ours requires only an RGB-D camera instead of an expensive laser range finder. Others require a multitude of LiDARs and RGB-D cameras [6, 7, 9], and/or do not operate for long-term [3, 32].

Fig. 4 shows experimental results for a structured indoor environment, a university hall, located on the second floor of a multistore 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. 4a, where the color scheme in the top-left indicates the different heights of points in the point cloud. One negative of the low-cost approach is a reduced density of the point cloud. This is to be observed in the figure, where between, e.g., fifteen and twenty meters on the z-axis and zero and five meters on 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 ϕ_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 [36]. Fig. 4b–4c shows a 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. 4b 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

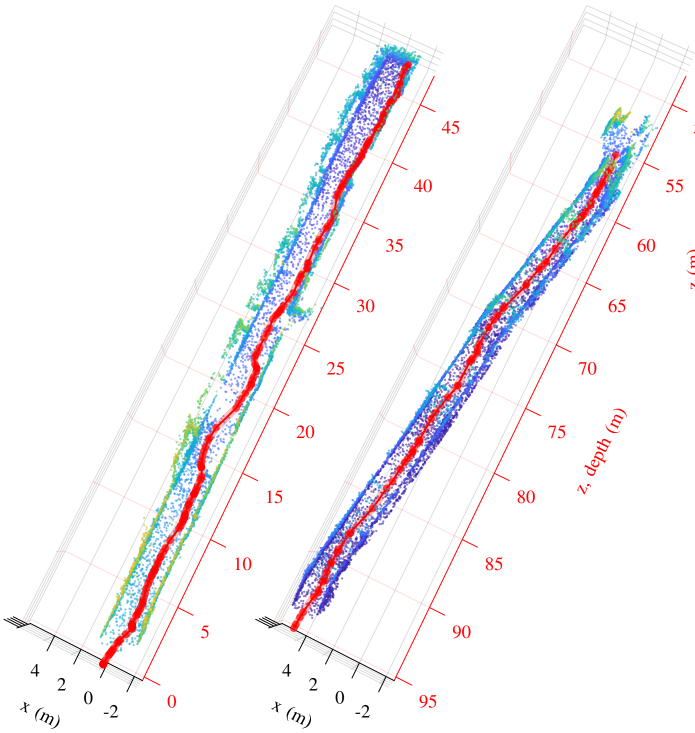


(a) Point cloud view of a structured indoor environment with visible countors of the exploration space. Points are colored for different heights.

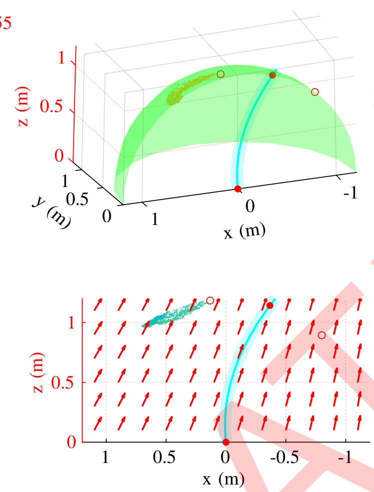
(b) The first detection of an obstacle "door". A path function is selected to avoid the obstacle.

(c) The new path function is selected at the next time step as the obstacle occurrence is observed closer.

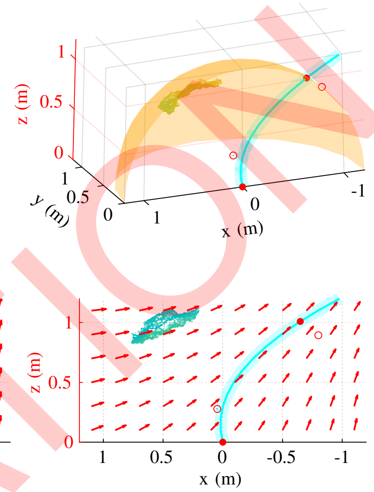
Fig. 4: Experimental results are reported for a structured indoor environment, a university hall composed of four connected corridors for a total length of approx. eighty meters. The view includes the point cloud in Fig. 4a and the detail of the algorithm for obstacle avoidance and detection at successive time steps in Fig. 4b and 4c. The points in the point cloud are filtered to report one point every two hundred and fifty. The colors of the spheres in Fig. 4b–4c indicate the proximity of an obstacle (orange indicates close proximity) and arrows the path-following vector field in Eq. (3). Robot's actual trajectory is in red and red dots indicate SLAM's registration points.



(a) Point cloud view of an unstructured indoor environment (left) and an underground tunnel (right) with visible contours of the exploration space. The color scale is the same as in Fig. 4a.



(b) The obstacle "wheel" is detected in the path of the robot, which selects a path function that avoids the obstacle.



(c) The path function is refined at the next time step, similar to Fig. 4c, as the obstacle appears closer.

Fig. 5: Experimental results are reported for an unstructured indoor environment and an underground tunnel for a total length of approx. one hundred meters. The view of the point cloud in Fig. 5a is filtered to report one point every five hundred. The detail of the algorithm for successive time steps is shown in Fig 5b–5c, similar to Fig. 4.

right. On the bottom is the path-following vector field from Eq. (3) in red and the path function ϕ_t in cyan. Fig. 4c shows the following time step as RB5 comes closer, and the robot has to perform a sharper maneuver to avoid the obstacle.

Fig. 5 shows an unstructured environment of a hall connecting to an underground tunnel on the respective left and right sides of Fig. 5a for approximately one hundred meters. Conversely to the experiment in Fig. 4, this 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. 5b–5c shows the obstacle detection, similar to Fig. 4b–4c, for a "wheel" placed close to the left edge of the first length of the figure wide approximately 0.42 meters. The trajectory the robot avoiding the obstacle is to be observed in Fig. 5a between fifteen and twenty meters on the z-axis.

The turning direction E in Eq. (3) is positive for left turns (see Fig. 4b) and negative for right turns (see Fig. 5c–5b). The turning rate k_e is derived empirically similar to other literature [28, 29], and is 0.05, 0.1, and 0.4 depending on the turning maneuver, i.e., it is 0.05 when ϕ_t is a line (or close to it), 0.4 when a sharp curve in respectively Fig. 4a and 5b. 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, every five hundred, etc., in Fig. 4a and 5a.

V. Conclusion and Future Directions

This paper consists of an approach for low-cost long-term autonomous exploration in both indoor and outdoor challenging environments. While comparable with other approaches tackling autonomous exploration, our approach extends the state-of-the-art further to operate in the presence of fewer sensory and computing requirements. Requiring only an RGB-D camera, all the exploration is computed in real-time on low-power computing hardware that is cheaper compared to the existing literature operating in similar settings.

The exploration is based on a novel mixed approach– a frontier- and sampling-based method from the literature extended with a path-following vector field from the aerial robotics domain–which allows the robot to operate at lower update frequencies. Human intervention, if required, is implemented via a novel methodology based on the LoRa low-power long-range communication technology from the internet-of-things domain. The position is from a state-of-the-art SLAM algorithm. Requiring only two low-cost LoRa bundles for communication, the approach enables operations on long distances with a custom communication protocol with no significant impact on costs and resources conversely to existing methodologies based on a mesh of devices.

The results show comparable data to existing literature but improved performance for long-term both indoor and outdoor experiments in a variety of settings. To enable further savings, we are currently extending the approach to account for energy requirements and to guarantee the completeness of the exploration cover via, e.g., exploiting the ergodicity of the exploration path with respect to an information density map. We are also extending the functionality to account for different cost functions in Eq. (7) and detections with the upward-facing RGB camera–unused in the current setup–to, e.g., detect objects similar to other literature.

ACKNOWLEDGMENTS

We thank Paedyn Gomes, Victoria Ereskina, and Beau Birdsall for their contribution to the mechanical design of the experimental robotic platform used in this paper.

REFERENCES

[1] I. Lluvia, E. Lazkano, and A. Ansuategi, "Active mapping and robot exploration: A survey," *Sensors*, vol. 21, no. 7, 2021. 1

[2] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov *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

[3] M. Müller and V. Koltun, "OpenBot: Turning smartphones into robots," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9305–9311. 1, 2, 4, 5

[4] B. Zhou, Z. Wu, and X. Liu, "Smartphone-based robot indoor localization using inertial sensors, encoder and map matching," in *Int. Conf. on Automation, Control and Robots (ICACR)*. IEEE, 2021, pp. 145–149. 1, 2

[5] S. M. F. Faisal, T. Rahman, and M. A. Kabir, "A low-cost rough terrain explorer robot fabrication using rocker bogie mechanism," in *Int. Conf. on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, 2021, pp. 1–4. 1, 2, 4

[6] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho *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, 5

[7] T. Rouček, M. Pecka, P. Čížek, T. Petříček *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, 4, 5

[8] W. Tabib, K. Goel, J. Yao, C. Boirum *et al.*, "Autonomous cave surveying with an aerial robot," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1016–1032, 2022. 1, 2

[9] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann *et al.*, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3306–3313. 1, 2, 4, 5

[10] K. Ebadi, Y. Chang, M. Palieri, A. Stephens *et al.*, "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 80–86. 1, 2

[11] K. Ismail, R. Liu, J. Zheng, C. Yuen *et al.*, "Mobile robot localization based on low-cost LTE and odometry in GPS-denied outdoor environment," in *Int. Conf. on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 2338–2343. 1

[12] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz *et al.*, "5G for robotics: Ultra-low latency control of distributed robotic systems," in *Int. Symposium on Computer Science and Intelligent Controls (ISCSIC)*. IEEE, 2017, pp. 69–72. 1

[13] C. Delgado, L. Zanzi, X. Li, and X. Costa-Pérez, "OROS: Orchestrating ROS-driven collaborative connected robots in mission-critical operations," in *Int. Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2022, pp. 147–156. 1

[14] C. Cadena, L. Carlone, H. Carrillo, Y. Latif *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

[15] A. Eldemiry, Y. Zou, Y. Li, C.-Y. Wen *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

[16] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "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. 2

[17] 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

[18] D. Tardioli, L. Riazuelo, D. Sicignano, C. Rizzo *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. 2

[19] T. Dang, F. Mascarich, S. Khattak, C. Papachristos *et al.*, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3105–3112. 2

[20] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric *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

[21] H. Kim, H. Kim, S. Lee, and H. Lee, "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. 2

[22] 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. 2, 4, 5

[23] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova *et al.*, "Receding horizon "next-best-view" planner for 3D exploration," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1462–1468. 2

[24] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas *et al.*, "Fast frontier-based information-driven autonomous exploration with an MAV," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9570–9576. 2, 3

[25] R. Shrestha, F.-P. Tian, W. Feng, P. Tan *et al.*, "Learned map prediction for enhanced mobile robot exploration," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1197–1204. 2, 3

[26] W. Qiao, Z. Fang, and B. Si, "A sampling-based multi-tree fusion algorithm for frontier detection," *Int. Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 14, 2019. 2

[27] 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

[28] A. Seewald, H. García de Marina, H. S. Midtiby, and U. P. Schultz, "Energy-aware planning-scheduling for autonomous aerial robots," in *Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2946–2953. 2, 6

[29] H. García de Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger *et al.*, "Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5740–5745. 2, 3, 6

[30] 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

[31] 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

[32] L. Schmid, M. Pantic, R. Khanna, L. Ott *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. 2, 4, 5

[33] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra *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. 3

[34] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel *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

[35] D. B. Bickler, "Articulated suspension system," 1989, U.S. Patent № 4840394. 4

[36] A. Seewald, U. P. Schultz, E. Ebeid, and H. S. Midtiby, "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems," *Int. Journal of Parallel Programming*, vol. 49, no. 2, pp. 136–157, 2021. 5