ROSCOE: Robot Scanning and Computing Equipment for Autonomous Terrestrial Mapping

Julian Raheema
Naval Information Warfare Center Pacific
University of California San Diego
La Jolla, CA, USA
0000-0001-9950-2921

Seth Farrell
Computer Science & Engineering
University of California San Diego
La Jolla, CA, USA
0009-0008-1497-0434

Michael Hess
Naval Information Warfare Center Pacific
San Diego, CA, USA
0009-0005-1518-488X

Raymond Provost
Naval Information Warfare Center Pacific
San Diego, USA
0009-0005-1757-5470

Mark Bilinski
Naval Information Warfare Center Pacific
San Diego, CA, USA
0000-0003-3635-2027

Henrik Christensen
Computer Science & Engineering
University of California San Diego
La Jolla, CA, USA
0000-0002-7465-7502

***Abstract*—Autonomous task-oriented robots are increasingly in demand across various domains; however, few existing systems address the challenge of autonomous high-resolution terrestrial scanning for construction and inspection purposes. This paper presents a task-oriented autonomy framework integrated with the Spot quadruped robot, enabling autonomous exploration, mapping, and deployment of a FARO terrestrial laser scanner. We introduce two novel algorithms for selecting optimal scanning positions: SCANSAFE (Scanpoint Navigator using Spatially-Aware Filtering and Evaluation), which prioritizes coverage of open space relative to prior scans, and PATHSAFE – Path-Aligned Trajectory Heuristic for Scanpoint Allocation with Filtering and Evaluation method, which places scan points along the robot's traveled path. These approaches are evaluated against two existing strategies: Next-Best-View Greedy (NBV-Greedy) and Frontier, as well as a manually guided baseline. Tested in multiple environments, the proposed algorithms successfully identified valid scanning points. On average, the SCANSAFE method generated 23.4% fewer scan points than NBV-Greedy, 44.4% fewer than Frontier, and 2.0% more than the manual baseline. The PATHSAFE method showed average reductions of 32.8% compared to NBV-Greedy, 51.6% compared to Frontier, and 10.4% compared to the manual approach. Both methods improved efficiency, reduced operational overhead, and increased safety in hazardous or constrained environments.**

***Index Terms*—Robots, Autonomous, Mapping, Terrestrial-LiDAR-scanning, Exploration**

I. Introduction

Terrestrial LiDAR scanning is widely used in the construction industry to create accurate 3D representations of physical environments. This technology, such as FARO scanners [1], works by projecting laser beams from a stationary device, often paired with cameras, to generate dense point clouds of the target area. Traditionally, this scanning process commonly referred to as dense scanning requires one person to select each scanning point, operate the FARO scanner mounted on a tripod, and independently reposition the equipment at every location. Once all desired scan locations are collected, the user employs FARO SCENE software to construct a high-resolution 3D map. Each scan had to have at least 40% overlap to be automatically registered and merged into a unified map. If the overlap was insufficient, the user must manually register the scans to complete the reconstruction. This process required the operator's full attention and became increasingly time-consuming with more than 30 scan positions. This manual approach is inefficient and can pose significant safety risks in hazardous environments, such as those involving explosives, radiation, or disaster zones. Therefore, a robot capable of autonomously navigating complex environments and performing these tasks is critical for effective automation.

Support for this work was provided by Naval Information Warfare Center Pacific RESTORE lab, the Naval Innovative Science and Engineering program, and the DoD SMART Scholarship for Service program.

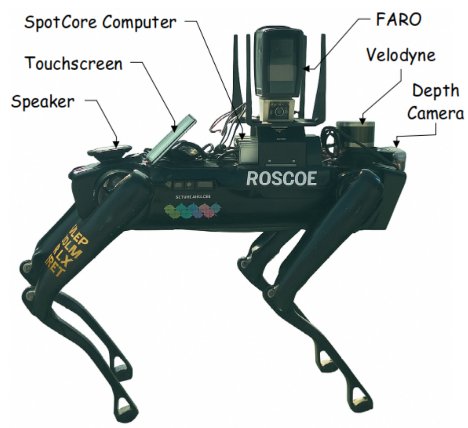


Fig. 1. SPOT equipped with autonomy and FARO scanner.

Mobile robots have become indispensable in a sort of domains, such as industrial inspection, search and rescue, scientific exploration, and 3D mapping of complex environments. Their ability to navigate cluttered, uneven, or hazardous spaces makes them well-suited for tasks where human access may be dangerous or impractical. In particular, quadruped robots like Boston Dynamics' Spot [2] offer robust mechanical stability, agile locomotion, and the payload capacity needed to carry advanced sensing equipment. These features make Spot an attractive choice for tasks requiring high mobility and detailed environmental sensing.

To address these limitations, we are investigating an autonomous system that integrates the Spot quadruped robot, the FARO Focus scanner, and the Robot Operating System (ROS) [3]. The system is not intended to replace human operators, but to serve as a robotic assistant that automates redundant tasks, or dangerous environment such as power-plants were human presence could be dangerous and not an option. Our goal is to enable the system to autonomously explore indoor environments, generate a 2D navigation map, and then automate the dense scanning process. We aim to **eliminate manual waypoint planning**, **decrease total scanning time**, and **boost productivity** by operating the system during off-hours, such as nights or weekends.

II. Research Background

In 2003, Philipp Althaus1 and Henrik I. Christensen [4] introduced a dynamical systems–based framework for coordinating robot behaviors in mobile navigation tasks. In 2005, Boston Dynamics' BigDog showcased advanced terrain navigation, marking a milestone in mobile robotics for rough environments [5]. Amazon's Kiva Robots, introduced in 2012, transformed warehouse logistics by automating inventory movement and management [6]. The Spot robot by Boston Dynamics, launched in 2015, excelled in versatile field tasks such as inspection and data collection [7].

The use of robots, especially quadrupeds, for industrial inspection has been a topic of great interest in recent years [8]–[11]. The specific task of terrestrial scanning, both tedious and repetitive, has been a prime target of robotic application and automation [12].

In the DARPA Subterranean (SubT) Challenge, the first-place winner, Team CERBERUS [13], developed a fully autonomous exploration system integrated onto the ANYmal quadruped robot [14]. The objective of the challenge was to deploy robots capable of autonomously navigating, mapping confined underground environments, and visually detecting artifacts within a 1-hour time limit. The coverage algorithm and planning system employed during the event relied on real-time sensor feedback to dynamically compute optimal waypoints for exploration and coverage. In post-event autonomous testing, the CERBERUS fleet achieved approximately 30% coverage of the final course. The resulting maps demonstrated high fidelity, with typical outlier rates between 10% and 14%, although one robot experienced higher error due to local misalignment. However, when using a FARO scanner for static mapping, real-time feedback is unavailable, making it challenging to determine optimal scanning locations for maximum coverage, often requiring human intervention.

Papers such as [15] utilize Building Information Models [16] in order to determine scan-position candidates. They do this by taking the BIM, converting it into a 2D Occupancy Grid and then defining a walkable path through the build which traverses through ideal scan positions. However, this process relies on having access to this building model, processing the points offline, and does not incorporate obstacles which are added to the building after the BIM is produced. This implies that the scanning positions determined to be optimal from the BIM may not be the actual best spots once obstacles are considered. This paper [17] requires the waypoints to be predetermined for the legged robot to navigate to the FARO scanning locations.

Our focus is on how to use a 2D online map to compute the best coverage and optimal scanning points for the FARO scanners without relying on FARO's feedback sensor. The core problem relates to the classical Art Gallery problem [18] and involves optimal viewpoint planning under mobility and sensor constraints. However, our formulation differs in that it considers a **mobile scanning robot** with limited sensor range and field of view, operating in a **3D environment** with occlusions and accessibility constraints. Rather than minimizing the number of stationary guards, our goal is to plan scan poses that maximize coverage while minimizing traversal cost and redundancy.

One of the algorithms implemented relies on active feedback from the sensor to compute the Next View Planner and Path Planning under Photogrammetric and Kinetic Constraints (NVP and PP-PKC) [19] to compute the optimal path planning for maximum coverages.

The Frontier exploration algorithm identifies the boundary between known (explored) and unknown (unexplored) areas in a map, directing the robot to these frontiers in order to expand coverage. Although it can function with only a 2D occupancy costmap, it relies on real-time sensor updates to remain effective, using it on a static map is limited. However, the method can become unsafe for equipment, as unknown areas may be near inaccessible zones or dangerously close to edges or obstacles, posing risks of collision or entrapment without proper validation or path planning [20].

DeepView is a method for selecting optimal scanning positions based on previously collected map data and visibility analysis. However, it relies on training data from similar environments to learn where the most useful scan points are likely to be. Its main limitation is that it may not perform well in unfamiliar or different spaces if it has not seen similar training examples [21]. The current state-of-the-art work relies on active sensor feedback and 3D data for view selection and mapping, which differs from our approach [22].

Our objective was to develop an autonomous robotic system capable of selecting strategic waypoints to achieve maximum coverage with minimal redundancy, without relying on real-time sensor feedback. To enable autonomous FARO scanning, the robot operates in two stages. First, it uses the Frontier exploration algorithm to map of the environment. Then, it leverages the map to compute optimal scanning locations for the FARO scanner.

III. Autonomous System Architecture

Our autonomy system for Boston Dynamics Spot robot is designed as a modular layered architecture. The following hardware components were used in this work:

- **Spot**: Quadrupedal mobility Robot
- **Velodyne VLP16 LiDAR:** A 16-channel LiDAR.
- **Intel RealSense Depth Camera (D455):** A depth camera utilized as part of the SLAM process.
- **FARO Scanner:** A precision laser scanner integrated as a Spot payload to capture high-fidelity 3D scans.
- **SpotCore Legacy computer:** Intel i5 NUC Lake-U Core with 16GB RAM as the primary computing resource.

Our system builds upon the existing system [23].

- **SLAM** We integrate Real-Time Appearance-Based Mapping (RTAB-Map) [24] for Simultaneous Localization and Mapping (SLAM), combining LiDAR and RGB-D data for robust localization. Key features include loop closure detection and graph-based optimization.
- **Exploration** We use frontier exploration [20] with RTAB-Map to update the occupancy grid and expand 2D map coverage. The system detects unexplored regions, generates waypoints using heuristics, and sends them to MoveBase for navigation. This 2D map serves as input to our coverage algorithms for selecting FARO scan points.
- **Planning and Control** Navigation uses MoveBase's global and local planners. While Spot handles low-level locomotion, our system issues high-level Cartesian path goals. A custom interface allows users to set exploration bounds, trigger FARO scans at waypoints, or switch to teleoperation when needed.

With this autonomy stack, the system autonomously explores, maps, and executes 3D scanning tasks. We implemented two new coverage algorithms, **SCANSAFE** and **PATHSAFE**, and evaluated them alongside **NBV**, **Frontier**, and **Human-Manual** methods to assess scan point selection performance in diverse environments.

IV. Methods

A. SCANSAFE Method

The **SCANSAFE** The method estimates the required number of scan points (SP). n, is upper limit of 30 the maximum number of SPs that can be executed in real-world deployments due to battery constraints. The algorithm gets the 2D costmap from RTAB-Map's SLAM module and distributes the SP uniformly across the free-space regions. Each candidate point must satisfy two constraints:

- Maintain a minimum distance from obstacles to ensure scanner visibility and safety.
- Maintain a minimum distance from other waypoints to reduce redundancy and maximize coverage.

Cells in the occupancy grid encode distances to the nearest obstacles, enabling efficient placement of evenly spaced, non-overlapping scan points.

Algorithm 1 SCANSAFE Waypoint Generation

Require: Grid-based map M, $n = 30$
Require: Minimum distance from obstacles d_{obs}, minimum distance between waypoints d_{wp}
Ensure: List of waypoint coordinates SP
Convert M to binary: 1 for free space, 0 for obstacles
Compute distance-to-obstacle map D from M
Filter out cells in D where distance $< d_{\text{obs}}$
Initialize waypoint list: $SP \leftarrow [\,]$
Rank valid cells by decreasing distance to obstacles
for each valid cell (x_i, y_i) in ranked order **do**
    Transform (x_i, y_i) to world frame $(x_{\text{world}}, y_{\text{world}})$
    if distance from $(x_{\text{world}}, y_{\text{world}})$ to all points in $SP \geq d_{\text{wp}}$ **then**
        Add $(x_{\text{world}}, y_{\text{world}})$ to SP
        if $|SP| = n$ **then**
            break
        end if
    end if
end for
return SP

B. PATHSAFE Method

Our PATHSAFE approach automatically places waypoints along the robot's traveled path. While the robot explores, whenever it moves x meters, a waypoint is placed if it is within y meters away from existing waypoints. This simpler heuristic:

- Avoids the need for a full post-processing distribution algorithm.
- Ensures a consistent spacing of scan points in continuous or corridor-like environments. * Minimizes the risk of large unscanned regions by methodically covering the path the robot has physically traversed.

Algorithm 2 PATHSAFE Waypoint Generation

Require: Traveled path $P = \{p_1, p_2, \ldots, p_n\}$, $n = 30$
Require: Step distance threshold x, min SP spacing y
Ensure: List of waypoint coordinates SP
Initialize waypoint list: $SP \leftarrow [\,]$
Set *last_wp_position* $\leftarrow p_1$
Add p_1 to SP
for each point p_i in P **do**
    if distance(p_i, *last_wp_position*) $\geq x$ **then**
        if distance(p_i, all $SP \in SP$) $\geq y$ **then**
            if $|SP| = n$ **then**
                break
            end if
            Add p_i to SP
            Set *last_wp_position* $\leftarrow p_i$
        end if
    end if
end for
return SP

(a) Basement **(b)** Office Space **(c)** Forest

Fig. 2. Three testing environments (a) an indoor wide space, (b) indoor narrow hallway, and (c) an outdoor forest

C. NBV-Greedy Method

The NBV-Greedy [25] starts by converting a grayscale map into a binary representation. It iteratively evaluates candidate points, ensuring they are spaced apart by a minimum distance and maximize coverage within a defined circular radius. For each valid point, it checks how much new area it would cover and updates the list of selected waypoints if the candidate contributes to additional coverage. The process stops once the desired number of waypoints is selected or all candidates are exhausted, and the final set of coordinates is returned.

Algorithm 3 NBV-Greedy Waypoint Selection

Require: $map, res, ox, oy, n, d_{min}, th_{obs}, r_{cov}$
$H, W \leftarrow$ shape of map
$free \leftarrow (map == 254)$
$dist \leftarrow$ distance_transform($free$)
$safe \leftarrow dist \geq (th_{obs}/res)$
$V \leftarrow$ indices where $safe$
$cov \leftarrow$ zero matrix same size as $free$
$SP \leftarrow [\ ]$
Generate circular mask M of radius r_{cov}
for all $(y, x) \in V$ **do**
  $SPx \leftarrow ox + x \cdot res$
  $SPy \leftarrow oy + (H - y) \cdot res$
  if $SP \neq \emptyset$ **then**
    $D \leftarrow$ distance(SP, (wx, wy))
    if $\exists d \in D < d_{min}$ **then continue**
    end if
  end if
  Define ROI $[x_{min}, x_{max}], [y_{min}, y_{max}]$ around (x, y)
  $crop \leftarrow M$ within ROI bounds
  $sub \leftarrow$ zeros like $free$; insert $crop$ into ROI
  $gain \leftarrow$ sum of $(sub \wedge free \wedge \neg cov)$
  if $gain > 0$ **then**
    Append (wx, wy) to SP
    $cov \leftarrow cov \vee (sub \wedge free)$
  end if
  if $|SP| \geq n$ **then break**
  end if
end for
return SP

D. Frontier Method

It begins by creating binary masks for free and unknown cells, then uses morphological dilation to find free cells adjacent to unknown regions, marking these as frontier points. These frontier areas were grouped into connected regions, and the centroid of each region was converted from image to world coordinates. Finally, it returns the total number of selected frontier scan points.

Algorithm 4 Frontier Waypoint Selection

Require: FrontierWaypoints$map, res, ox, oy, n, d_{min}$
$H, W \leftarrow$ shape of map
$F \leftarrow 254$, $U \leftarrow 205$
$free \leftarrow (map == F)$
$unk \leftarrow (map == U)$
$kern \leftarrow 3 \times 3$ ones
$dil\_unk \leftarrow$ dilate(unk, $kern$)
$frontier \leftarrow free \wedge (dil\_unk > 0)$
Label $frontier \rightarrow regions$
$C \leftarrow [\ ]$
for all $r \in regions$ **do**
  $(y, x) \leftarrow r.centroid$
  $SPx \leftarrow ox + x \cdot res$
  $SPy \leftarrow oy + (H - y) \cdot res$
  Append (wx, wy) to C
end for
$SP \leftarrow [\ ]$
for all $p \in C$ **do**
  if SP empty **or** $dist(p, S) \geq d_{min}$ **then**
    Append p to SP
  end if
  if $|SP| \geq n$ **then break**
  end if
end for
return SP

E. Manual Method

This method depends on an operator to manually select scan points, leveraging human intuition to identify areas of particular interest, such as regions with complex geometry (e.g., dense machinery, doors, stairs), potential structural concerns, or key vantage points offering wide or overlapping fields of view. While this approach can be more time-consuming, it often results in optimal coverage for specialized or safety-critical applications where expert judgment is essential.

V. Experimental and Test Environments

Here we investigate five coverage approaches, each differing in how the waypoints are selected. The scanner maintains the same configuration (resulting in $\sim$3.5 minute scans), ensuring consistent data collection. We used FARO SCENE [26] software to reconstruct the 3D data from the point clouds and generate a dense 2D map. Table I compares the number of scan points generated by each algorithm across the two test environments. In the Basement environment, algorithms were configured to be placed at least 1.0 meter away from obstacles, with a minimum spacing of 3.0 meters between them. For the office environment, the original settings were too strict for the narrow spaces, so we adjusted the configuration to allow waypoints as close as 0.8 meters from obstacles, with a minimum spacing of 7.0 meters. With these settings in place, we then performed our tests using the other algorithms.

We selected three different environments to test our robot autonomous navigation and scanning. However, The third environment is an outdoor forest-like area (Fig. 2c) with numerous eucalyptus trees and limited structural diversity. Due to unresolved autonomy challenges in mapping and navigation, it was excluded from the main evaluation, with key findings summarized in the Section V-A section. The first environment is the basement of a campus building from UCSD (as shown in Fig. 2a), a low traffic area with the only obstacles being tables and chairs, making it an ideal test site; the narrowest section is approximately 10 feet wide and extended up to 100 feet, total of 4,176 ft^2. The second environment is an office area in the Computer Science and Engineering department at UCSD, referred to as office environment (as shown in Fig. 2b), which also had minimal foot traffic but featured a more complex layout; its narrowest section measured approximately 5.5 feet in width and stretched up to 185 feet total or 7,982. ft^2. For each run, we first operate the robot in mapping mode to generate a 2D map using SLAM. Once mapping is complete and the scanning points are confirmed to be safe for navigation, we switch to scanning mode, allowing the robot to navigate to each scanning location and collect data. **Note on comparative evaluation:** While our method was compared with the NBV-Greedy and Frontier algorithms, detailed quantitative analysis was limited in Sections V and VI. Preliminary tests showed both methods generated excessive and unsafe scan points, resulting in long scanning times and significant data overlap. As shown in Figs. 3e and 4e, NBV-Greedy produced up to 45.24% more waypoints than SCANSAFE and PATHSAFE, many near obstacles and unsafe for deployment. Similarly, Figs. 3f and 4f show the Frontier algorithm generated up to 71.95% more waypoints, often in unnavigable areas. Hence, both algorithms were excluded from physical robot tests. Although Frontier could aid exploration by collecting frontier points, it risks placing them in non-navigable regions.

TABLE I: Comparison of scanning algorithms tested across two environments

Algorithm	Basement-Env SP	office-Env SP
SCANSAFE	26	23
PATHSAFE	22	21
NBV-Greedy	36	28
Frontier	51	38
Manual	21	27

A. *Outoor-Forest*

We tested the system in a forest-like environment with limited visual features, which caused RTAB-Map to crash due to poor mapping and localization. Although the robot operated for 5–10 minutes and initial waypoints were generated, mapping failures prevented further autonomous navigation or scan generation. These failures reinforce the need for robust feature-based localization methods, particularly in low-texture or natural environments.

B. *SCANSAFE Results*

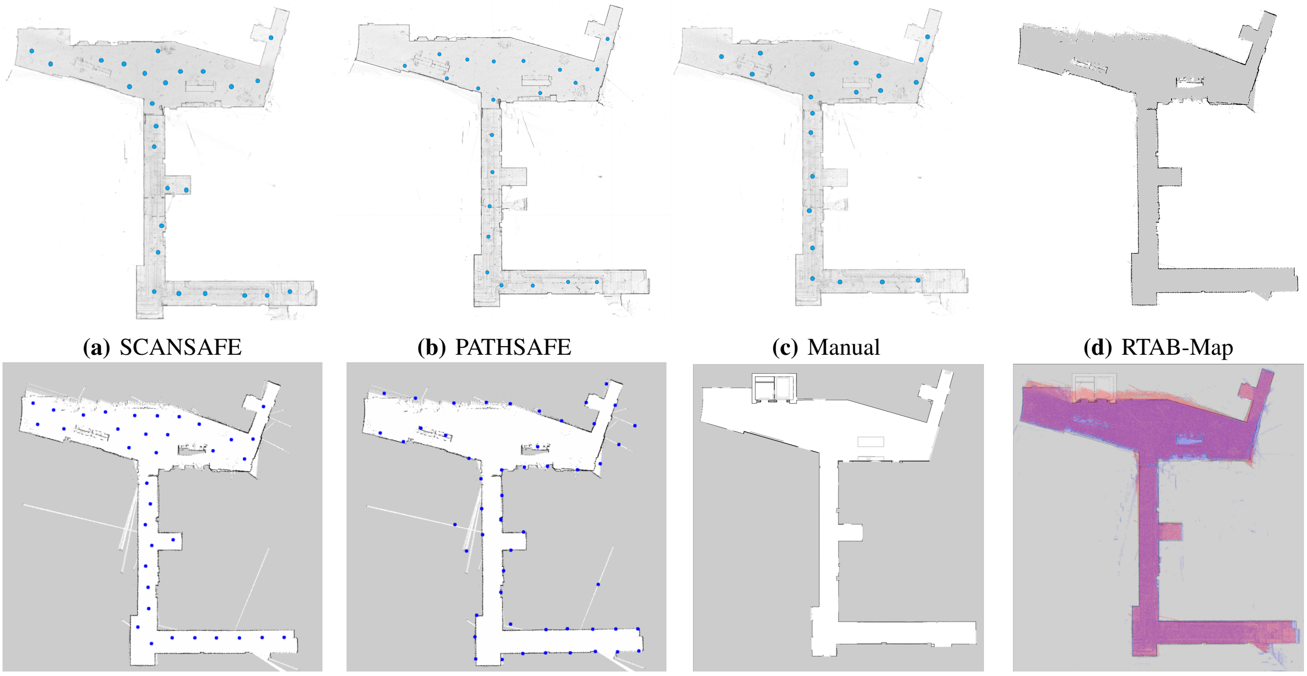
1) Basement Environment: The algorithm generated 27 waypoints, and the robot navigated to each waypoint to initiate scanning. During the scanning process, we had to intervene twice, once to replace the robot's battery, and second, when a person left a door open, the robot tried to enter the new room while navigating to the next waypoint. After scanning, we constructed the 3D data with minimal operator intervention to align the point clouds (as shown in Fig. 3a).

2) Office Environment: In the second environment, the algorithm generated 22 waypoints, which we found were placed in ideal locations such as intersections between hallways. The results from this environment (as shown in Fig. 4a).

There were several manual interventions due to camera failures, three battery swaps, and one case in which the robot got stuck under a table which was positioned slightly above the sensors. Additionally, one of the main room's walls is composed almost entirely of glass, which introduced substantial noise into the mapping process due to the LiDAR returns. In this complex and large environment, the scanning waypoints were much too sparse resulting in an excessively long post-processing phase – algorithmic alignment created multiple small clusters requiring significant manual stitching to fully align. This could be fixed by increasing the number of scanning points allowed, which we had capped at a maximum of 30 for all environments.

C. *PATHSAFE Results*

1) Basement Environment: The algorithm generated 22 waypoints, and the robot navigated to each waypoint to perform scanning. During the process, we intervened once to replace the robot's battery. After completing the scans, we processed the data, and the operator efficiently constructed the 3D data without needing extra effort to align the point clouds (as shown in Fig. 3b).



(a) SCANSAFE **(b)** PATHSAFE **(c)** Manual **(d)** RTAB-Map

(e) NBV-Greedy Sim **(f)** Frontier Sim **(g)** Ground Truth **(h)** Overlay

Fig. 3. Waypoints scan position for the Basement environment generated by: (a) SCANSAFE (26), (b) PATHSAFE (22), and (c) Manual (21). (d) RTAB-Map 2D map generated for robot navigation (e) NBV-Greedy (36) in simulation, (f) Frontier (51) in simulation, (g) Ground Truth, and (h) Overlay FARO and RTAB-Map over the Ground Truth.

TABLE II: Combined Summary of Experiment and Algorithm Testing Results

Env	Alg	SP	PP-Time (hr)	Coverage (%)		Acc (%)		ToC	Safety Stop
				RTAB-Map	FARO-Scanner	RTAB-Map	FARO-Scanner		
Basement	SCANSAFE	26	2.5	89.97	91.61	91.05	96.71	2h 04m (2 intv.)	1
	PATHSAFE	22	2.0	90.00	87.89	86.94	93.73	1h 44m (1 intv.)	0
	Manual	21	1.5	89.78	91.84	89.38	96.62	1h 41m (1 intv.)	0
office	SCANSAFE	23	3.3	94.83	90.86	67.82	83.83	2h 25m (7 intv.)	4
	PATHSAFE	21	3.2	93.66	89.42	67.84	81.35	2h 05m (7 intv.)	3
	Manual	27	3.5	87.93	83.70	70.40	72.20	2h 12m (2 intv.)	0

2) Office Environment: the algorithm generated 21 waypoints (as shown in the Fig. 4b). The robot encountered difficulties navigating the narrow sections due to the graph-based SLAM algorithm's limitations in recognizing features and achieving loop closure. This required about 10 manual interventions and three battery swaps.

Similar to the SCANSAFE algorithm, the operator spent a couple of hours realigning the point clouds, with the scanning data forming seven puzzle-like pieces.

D. Human-Manual Results

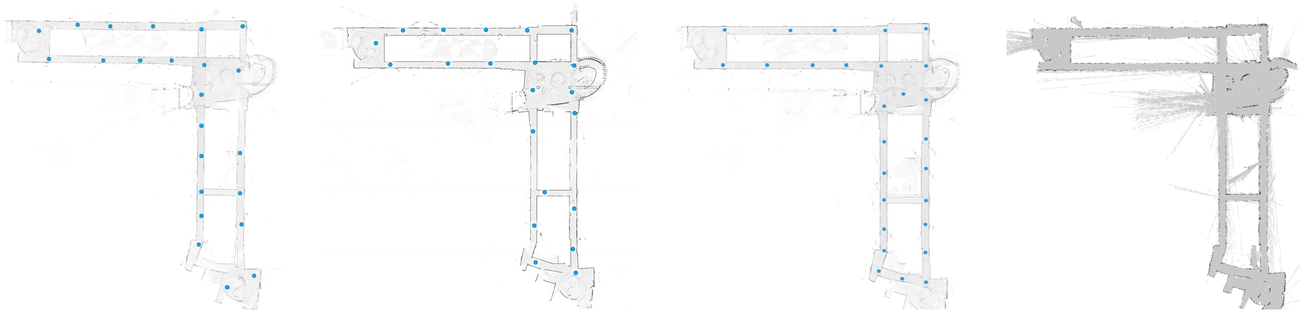
1) Basement Environment: The A human expert manually identified suitable scanning locations, generating 21 waypoints. The robot was then navigated to each waypoint, where scans were manually initiated. After completing the scans, the data was processed, and the FARO SCENE software automatically aligned the collected scans without requiring additional input from the operator (as shown in Fig. 3c).

2) Office Environment: We manually navigated the robot to each of the total 27 waypoints and initiated the scans. However, the scanning data required at least an hour of manual effort to reconstruct the point clouds, resulting in four puzzle-like pieces.

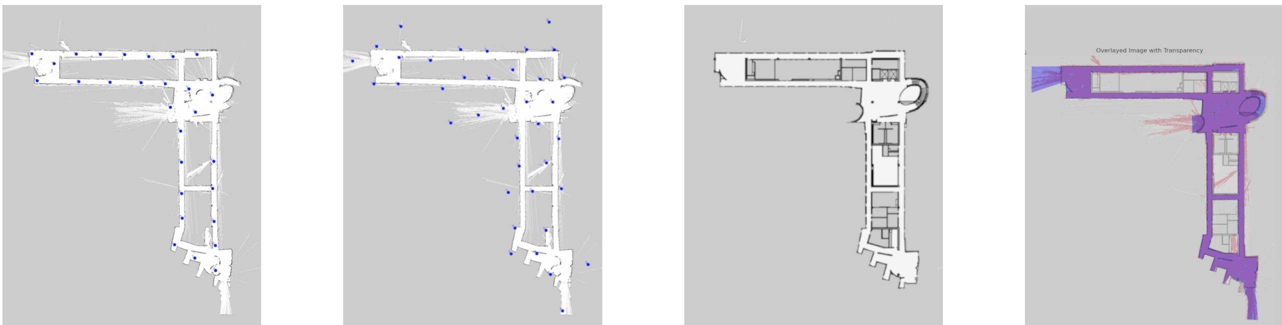
VI. Evaluation

We used the key metrics and parameters shown in Table II for our evaluation, described as follows:

- **Env:** The type and layout of the environment in which the experiment was carried out.
- **Alg:** The specific method used to determine the scanning points.
- **SP:** The total number of waypoints generated by the algorithm or manual for the robot to navigate and scan.
- **PP-Time, hours:** The post-processing time that the operator needs to process the scan data using FARO SCENE software after all the scan points have been collected.
- **Coverage:** The ratio of overlapping white pixels between the ground truth and sensor image to the total white pixels in the ground truth.

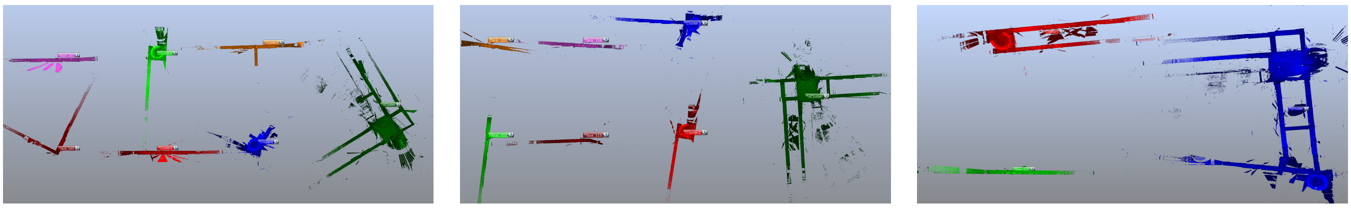


(a) SCANSAFE **(b)** PATHSAFE **(c)** Manual **(d)** RTAB-Map



(e) NBV-Greedy **(f)** Frontier **(g)** Ground Truth **(h)** Overlay

Fig. 4. Scan results for office building. (a) SCANSAFE (23), (b) PATHSAFE (21), (c) Manual (27), (d) RTAB-Map online generated 2D map for robot navigation and autonomous waypoint, (e) NBV-Greedy (28), (f) Frontier (38), (g) Ground Truth, (h) Overlay FARO and RTAB-Map over the Ground Truth.



(a) SCANSAFE **(b)** PATHSAFE **(c)** Manual

Fig. 5. SCANSAFE, PATHSAFE, and Manual methods applied on the office 3rd Floor map, with scans processed in FARO SCENE for automatic reconstruction. Results highlight reconstruction challenges due to poor synchronization or insufficient 40% overlap.

- **Acc:** The ratio of overlapping white pixels to the total white pixels in the sensor image.
- **ToC:** Time of Completion to complete the scanning process, including robot navigation and user interaction.
- **Safety Stop:** The total number of waypoints the robot navigates to where the FARO scanner captures data.

The primary finding (as shown in Table II) is that the SCANSAFE and PATHSAFE algorithms perform effectively in open and less constrained environments. The PATHSAFE method tends to require fewer scan points, reducing operational time and effort. However, in more complex and narrow areas, such as the office building, waypoint selection proved challenging. The scans could not be successfully reconstructed algorithmically using FARO SCENE software (as shown in Fig. 5). All methods struggled to identify optimal scan positions in such environments, which hindered FARO SCENE's ability to perform accurate 2D registration. Based on expert feedback, scans must be placed approximately 3 feet apart in narrow spaces with few distinguishing features. Nonetheless, the SCANSAFE method was more likely to place scan points near intersections, contributing to better coverage. Overall, the SCANSAFE algorithm achieves a balance between coverage and redundancy, while the PATHSAFE method is more straight forward. However,it can lead to more overlap. The manual approach, although subjective, often focuses on areas of higher interest. Shadow issues were also observed when the scanner was used on the robot similar to [17]. Additionally, the navigation map generated in seconds by the SLAM algorithm proved useful and could help operators pre-plan scanning points for better result, a direction we plan to explore in future work.

A. Discussion: Limitations and Ethical Considerations

The algorithms are sensitive to poor mapping, causing navigation failures in environments like forests. Battery life and sensor occlusion (e.g., glass walls) also impact reliability. Ethical concerns include deployment in active construction sites, where human presence poses safety challenges. The system is designed to assist, not replace, human operators, and should be used with oversight. Data privacy and environmental impact warrant further consideration. Moreover, such robotic systems could prove valuable in hazardous or hard-to-reach areas, such as disaster zones or structurally unstable environments. In these scenarios, autonomous operation can reduce risk to human workers while still enabling critical inspection and data collection.

VII. Conclusion

We presented a task-oriented autonomy system, tested in two environments and compared against existing baselines and a manual approach. The system integrates a SLAM and navigation stack based on ROS with a FARO high-resolution scanner payload. It automates both exploration and scanning, offering two distinct methods for scan point selection: SCANSAFE, which maximizes coverage, and PATHSAFE, which is quicker to implement and more intuitive. The manual approach leverages expert insight. In all methods, the operator is not required to carry the scanning equipment — the robot handles transportation. The system generates a 2D map of the environment, which serves as input to the navigation system for determining optimal scanning waypoints. Our findings emphasize the importance of aligning the coverage strategy with specific operational goals such as resolution, time constraints, and redundancy. By evaluating these methods through real-world experiments, we provide a reference for future deployments of quadruped-based scanning systems, supporting broader use in automated workflows particularly in environments hazardous to humans.

References

[1] FARO Technologies, Inc., "Faro 3d scanner," https://www.faro.com, n.d., accessed: 2025-01-03.

[2] B. D. Inc., "Spot SDK — Spot 4.0.1 documentation." [Online]. Available: https://dev.bostondynamics.com/

[3] W. Garage, "Robot operating system." [Online]. Available: http://wiki.ros.org/

[4] P. Althaus and H. I. Christensen, "Behavior coordination in structured environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, 2003, pp. 1070–1075. [Online]. Available: https://faculty.cc.gatech.edu/~hic/hic-papers/ar03-beh-comp.pdf

[5] R. Playter, M. Buehler, and M. Raibert, "Bigdog," in *Proc. SPIE 6230, Unmanned Systems Technology VIII*, Orlando (Kissimmee), Florida, May 2006.

[6] Amazon Robotics, "Warehouse automation with kiva robots," https://www.aboutamazon.com/news/operations/how-robots-help-sort-packages-at-amazon, 2012, online resource.

[7] M. Staniaszek, T. Flatscher, J. Rowell, H. Niu, W. Liu, Y. You, R. Skilton, M. Fallon, and N. Hawes, "Autoinspect: Towards long-term autonomous industrial inspection," 2024. [Online]. Available: https://arxiv.org/abs/2404.12785

[8] Y. Jang, W. Seol, K. Lee, K. Kim, and S. Kim, "Development of quadruped robot for inspection of underground pipelines in nuclear power plants," *Electronics Letters*, vol. 58, 12 2021.

[9] D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, 10 2018.

[10] S. Halder, K. Afsari, J. Serdakowski, S. DeVito, M. Ensafi, and W. Thabet, "Real-time and remote construction progress monitoring with a quadruped robot using augmented reality," *Buildings*, vol. 12, p. 2027, 11 2022.

[11] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, "Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study," *Journal of Building Engineering*, vol. 65, p. 105814, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710222018204

[12] M. Zagorski and R. Miltchev, "Facilitation of the 3d scanning process of industrial sites using a self-moving autonomous robotic system," in *Environment. Technology. Resources. Proceedings of the 15th International Scientific and Practical Conference*, vol. 3, 06 2024.

[13] N. Hudson, S. Khattak, H. M. L. Dang *et al.*, "Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned," *Field Robotics*, vol. 2, pp. 595–636, 2022.

[14] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal – a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[15] S. Park, S. Yoon, S. Ju, and J. Heo, "Bim-based scan planning for scanning with a quadruped walking robot," *Automation in Construction*, vol. 152, p. 104911, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580523001711

[16] I. Skrzypczak, G. Oleniacz, A. Lesniak, K. Zima, M. Mrówczyńska, and J. Kazak, "Scan-to-bim method in construction: assessment of the 3d buildings model accuracy in terms inventory measurements," *Building Research & Information*, vol. 50, pp. 1–22, 01 2022.

[17] E. Wetzel, J. Liu, T. Leathem, and A. Sattineni, "The use of boston dynamics spot in support of lidar scanning on active construction sites," in *Proceedings of the 39th International Symposium on Automation and Robotics in Construction (ISARC)*, July 2022.

[18] J. O'Rourke, "Art gallery theorems and algorithms," in *Theoretical Computer Science*. Oxford University Press, 1987, original problem posed by Victor Klee in 1973.

[19] C. Fan, H. Wang, Z. Cao, X. Chen, and L. Xu, "Path planning of autonomous 3-d scanning and reconstruction for robotic multi-model perception system," *Machines*, vol. 11, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2075-1702/11/1/26

[20] B. Yamauchi, "A frontier-based approach for autonomous exploration," *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151, 1997. [Online]. Available: https://api.semanticscholar.org/CorpusID:206561621

[21] L. Li, L. Wan, and X. Zhang, "Boundary exploration of next best view policy in 3d robotic scanning," 2025. [Online]. Available: https://arxiv.org/abs/2412.10444

[22] W. Jiang, B. Lei, and K. Daniilidis, "Fisherrf: Active view selection and mapping with radiance fields using fisher information," in *European Conference on Computer Vision (ECCV)*, vol. 13409. Springer, 2024, pp. 422–440. [Online]. Available: https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/02130.pdf

[23] R. J. Y, M. R. Provost, H. R. C, M. Bilinski, and H. I. Christensen, "Autonomous exploration and mapping payload integrated on a quadruped robot," in *Intl. Symp. on Robotics Research (ISRR)*, Long Beach, CA, 2024.

[24] 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, pp. 416–446, 3 2019.

[25] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1462–1468, 2016.

[26] FARO Technologies, Inc., "Scene: 3d visualization and processing software," https://www.faro.com/products/software/scene/, n.d., accessed: 2025-01-03.