2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

AUTONOMOUS AS-BUILT DOCUMENTATION WITH A QUADRUPED ROBOT AND LIDAR-SLAM

Johannes Schillberg, Christoph Blut, and Jörg Blankenbach
RWTH Aachen University, Aachen, Germany

Abstract

Automating as-built documentation enhances efficiency, safety, and quality in construction projects. This study explores a quadruped robot equipped with a LiDAR sensor for autonomous navigation and mapping. We evaluate three open-source LiDAR-SLAM algorithms in diverse indoor environments: a long hallway, a cellar, and a stairwell. Results reveal the strengths and limitations of each algorithm, providing critical insights into their suitability for as-built documentation in complex scenarios. By addressing challenges in dynamic and constrained settings, our findings contribute to advancing robotic solutions for autonomous as-built documentation in varied architectural environments.

Introduction

As-built documentation plays a critical role throughout the lifecycle of structures, offering a foundation for informed decision-making, efficient operations, and long-term sustainability. Yet, its importance is often underappreciated or overlooked, leading to fragmented knowledge, costly inefficiencies, and avoidable environmental impacts. From the early design stages through construction, occupancy, and eventual decommissioning, comprehensive documentation serves as a living repository of data (Blut and Blankenbach, 2022). This includes architectural designs, material specifications, system integrations, and maintenance records. Accurate and up-to-date documentation enables stakeholders – architects, engineers, facility managers, and researchers – to collaborate effectively, mitigate risks, and adapt to evolving requirements. Moreover, as structures grow increasingly complex with integrated smart systems, the value of well-maintained documentation becomes even more pronounced. Specifically, construction progress monitoring is a crucial part of as-built documentation, since it gives real-time insights into the execution phase of a structure's lifecycle. Through on-site observations and building information modeling (BIM) (Kaden et al., 2024), progress monitoring ensures alignment between planned designs and actual outcomes. This process not only reduces delays and cost overruns but also enhances transparency and accountability and captures the "as-built" reality, which is indispensable for future renovations, repairs, and energy performance assessments (Blut et al., 2024). However, as-built documentation is cost- and labor-intensive to achieve accurate and meaningful results (Kopsida et al., 2015). By equipping modern robots with Light Detection and Ranging (LiDAR) sensors and combining them with advanced Simultaneous Localization and Mapping (SLAM) algorithms, mapping structures as rich 3D point clouds can fully be automated (Kim et al., 2018). The point clouds can in turn be used to derive BIM models (Blut et al., 2023). Robotic platforms can address one of the most significant challenges in construction progress monitoring: consistently collecting accurate, up-to-date, and comprehensive site data. Opting for a quadruped robot with superior maneuverability on uneven terrain further extends the capability of autonomous documentation to scenarios where traditional wheeled robots would find maneuvering impossible, such as navigating stairwells or overcoming ground-level obstacles. However, selecting the appropriate SLAM algorithm for this task significantly influences the robot's accuracy of navigation on the one side and the quality of the resulting point cloud on the other side. In this paper, we show a possible hardware setup and compare three open-source SLAM algorithms across different environments in terms of accuracy and resulting point cloud quality. We begin with a review of related research, followed by a detailed explanation of the methods and conducted experiments. Finally, we present and discuss the results.

Related work

With construction progress monitoring being a critical component of traditional building processes, the automation of this task has become an increasingly important area of research. As construction projects grow in complexity, the demand for accurate, real-time data on project status has driven efforts to integrate advanced technologies into monitoring systems. One promising direction in this research is the use of quadruped robots, which offer unique advantages in navigating complex and uneven construction sites (Halder et al., 2023). Quadruped robots equipped with LiDAR sensors can autonomously

generate detailed point clouds of the construction site, which provide valuable insights into the current state of the construction project, enabling more effective management and decision-making. The core component that enables point cloud creation is SLAM. There are some other studies that have evaluated autonomous mapping with SLAM. For example, Merkle et al. (2023) utilized a wheeled robot equipped with two Velodyne LiDAR sensors to compare the performance of three SLAM algorithms: KISS-ICP (Vizzo et al., 2023), SC-LIO-SAM (Shan et al., 2020, Kim & Kim, 2018), and MA-LIO (Jung et al., 2023). The experiments were conducted by tracking the robot's trajectory with a total station while mapping sections of a federal road bridge. Multiple trajectories were recorded beneath the bridge, covering diverse topologies including a flat section, a hilly area, and a pump track featuring multiple slopes, elevations, and sharp curves. The results demonstrated that the MA-LIO algorithm outperformed the others in terms of trajectory accuracy, achieving the lowest RMSE on four out of six test tracks, with errors ranging between 5 and 7 cm. On the other hand, SC-LIO-SAM exhibited superior performance in point cloud accuracy, achieving a lower mean cloud distance in four out of six tracks, with values ranging from 4 to 12 cm. Another example is Yarovoi and Cho (2024), who utilized Exp04 - 06 from the Hilti Challenge Dataset (Zhang et al., 2023) to evaluate several SLAM algorithms: LeGO-LOAM (Shan & Englot, 2018), LIO-SAM, and ART-SLAM (Frosi & Matteucci, 2022). This dataset was captured in a live construction environment featuring bare concrete walls, narrow corridors, on multiple floors using a handheld device equipped with five cameras, an Inertial Measurement Unit (IMU), and a LiDAR sensor. The sequences reflect varying walking speeds and motion patterns, at changing elevations, providing diverse and challenging conditions for evaluation. Their results demonstrate the superior performance of the LIO-SAM algorithm. In terms of trajectory accuracy, LIO-SAM exhibited the lowest RMSE across all tracks, with values ranging between 9 and 31 cm. Additionally, it achieved the lowest average nearest neighbor distance when compared to a generated reference point cloud, with distances ranging between 6 and 9 cm. Further LiDAR SLAM solutions and comparative studies are described in the comprehensive survey by Zhang et al. (2024). However, studies exploring the use of quadruped robots equipped with LiDAR sensors remain scarce (Lee et al., 2023). This work helps close this gap by comparing three different LiDAR SLAM algorithms in three different indoor environments.

Methods and Experiments

Hardware Setup

We equipped a Boston Dynamics Spot robot with an Ouster OS1-128 LiDAR and a Raspberry Pi 4 (Figure 1). Both are powered by the robot's internal battery. The Raspberry Pi functions as an interface between the LiDAR sensor and the robot, enabling efficient communication and data processing. All sensor data collected during the experiment was stored on the Raspberry Pi's internal memory, ensuring streamlined data management and retrieval. The Ouster OS1-128 is a mid-range, high-resolution imaging LiDAR sensor with a 360° horizontal and a 45° vertical field of view (FoV). It features a 10Hz rotation rate and offers a precision of ±1 cm for scans ranging between 1 to 20 meters, and ±2 cm for scans ranging between 20 to 50 meters. The sensor was operated with a vertical resolution of 128 channels and a horizontal resolution of 1024. Additionally, the Ouster OS1-128 is equipped with an internal 6-axis IMU, capable of a 100Hz sampling rate.

To accurately determine the robot's ground truth position, a 360-degree prism was placed directly above the LiDAR. This setup enabled precise and continuous tracking with millimeter accuracy using a Leica TS30 total station. The reference point clouds were gathered using a RIEGL VZ-400i and a RIEGL VZ-600i terrestrial laser scanner, providing dense and precise reference data with millimeter accuracy.

Figure 1: Boston Dynamics Spot robot with Ouster OS1-128 LiDAR.

Environments

The tests were conducted in three different locations: a straight path in a hallway, closed-loop route in a cellar, and a stairwell connecting two floors with an intermediate landing. The environments were selected to showcase a diverse range of challenges associated with SLAM-based mapping in indoor settings, including scenarios where traditional wheeled robots would reach their limits. The selected routes cover use cases in which a robot either revisits a location during mapping or does not. In a closed-loop scenario, the SLAM algorithm can detect and correct errors that inherently accumulate over time by recognizing previously seen places, and refine the map and the robot's pose estimate, resulting in a more accurate and consistent representation of the environment. This is referred to as loop closure. Though, in many real-world mapping scenarios the robot never returns to a previously

visited location, for example in corridors. Here, the benefits of loop closure cannot be exploited.

The hallway environment (Figure 2) spans about 70 × 3 meters, with a trajectory length of 44 meters. This space is highly repetitive, featuring flat wooden surfaces occasionally interrupted by glass surfaces from display cases embedded in the walls. The trajectory runs along one side of the hallway instead of the center.

The cellar environment (Figure 3) is characterized by numerous obstacles and a rectangular closed-loop trajectory. The robot often navigates close to objects and glass surfaces due to the many shelves present. The area measures approximately 35 × 15 meters, with the trajectory covering a length of 49 meters.

The stairwell environment (Figure 4) covers an area of approximately 15 × 8 meters, with a trajectory length of 32 meters. It is notable for its large window fronts and wide stairs with an intermediate landing. The trajectory moves from one floor down through the landing to the lower floor and then ascends again.

Figure 2: Hallway test environment and trajectory.

Figure 3: Cellar test environment and trajectory.

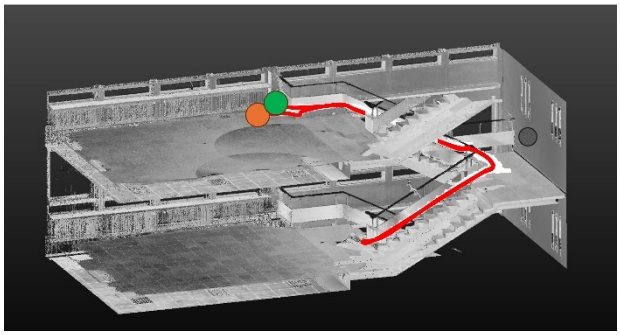

Figure 4: Stairwell test environment and trajectory.

Software and Algorithms

To facilitate efficient data collection, we developed a custom middleware and installed it on the Raspberry Pi, ensuring seamless integration with the robot's control system via the supplied tablet. This middleware utilizes the Boston Dynamics Spot API, allowing for coordination between the LiDAR sensor and the robot's control system, thereby streamlining the data acquisition process.

To compare the trajectories generated by the SLAM algorithms with those captured by the total station in terms of Absolute Pose Error (APE) and Relative Pose Error (RPE), the widely used python package evo (Grupp, 2024) was employed. It also provides an implementation of the Umeyama algorithm (Umeyama, 1991) for aligning trajectories across different reference frames. For point cloud comparison, the free software CloudCompare was used, assisting with alignment and calculating cloud-to-cloud distances. For the SLAM algorithms, we selected three distinct approaches: KISS-ICP, FAST-LIO2 (Xu et al., 2021), and LIO-SAM-6AXIS (Hu, 2023). KISS-ICP serves as a simple, IMU-free baseline. LIO-SAM and its variants were included in both Merkle et al. (2023) and Yarovoi & Cho (2024), where they performed well in generating accurate point clouds. To broaden the methodological spectrum, we added FAST-LIO2, which combines LiDAR and IMU data using a computationally efficient filter-based approach. Together, these algorithms capture a range of current SLAM strategies for indoor autonomous mapping. For an overview see Table 1.

Table 1: The SLAM algorithms and their features.

Algorithm	IMU	Feature Extraction	Method	Loop Closure
KISS-ICP	No	No	ICP	No
FAST-LIO2	Yes	No	Filter	No
LIO-SAM	Yes	Yes	Graph	Yes

KISS-ICP

KISS-ICP (Keep It Small and Simple - Iterative Closest Point) is primarily an odometry technique that leverages the Iterative Closest Point (ICP) algorithm, with an adaptive threshold for data association. It is designed to be lightweight and efficient, operating independently of IMUs, making it compatible with a wide variety of range-sensing technologies. We left most parameters as default, only the voxel_size was set to 0.18 and the deskew parameter was set to True.

FAST-LIO2

FAST-LIO2 builds upon its predecessor FAST-LIO, a LiDAR-inertial Odometry (LIO) framework that fuses LiDAR feature points with IMU data using a tightly coupled iterated extended Kalman filter. FAST-LIO2 introduces two key innovations: removing the feature extraction module from the original FAST-LIO and directly inserting the raw points into the internal map, which is maintained as an incremental k-d tree (ikd-Tree). These enhancements allow FAST-LIO2 to increase accuracy and reduce computational load compared to traditional LIO approaches. We left most parameters as default, except filter_size_surf and filter_size_map, with both set to 0.2.

LIO-SAM-6AXIS

LIO-SAM-6AXIS is an open-source modification of the LIO-SAM algorithm, enabling the use of 6-axis IMUs, such as the internal IMU of the Ouster OS1-128 LiDAR used in our work, whereas the original algorithm supported only 9-axis IMUs. LIO-SAM is a tightly coupled LiDAR inertial odometry framework designed for highly accurate trajectory estimation and map-building. It achieves this by maintaining and optimizing a factor graph that fuses LiDAR odometry, IMU pre-integration, loop closure, and optionally GPS. In this work, the test ouster128.yaml configuration file was used with only the sensor-related parameters modified. Additionally, the odometrySurfLeafSize and mappingSurfLeafSize parameters were set to 0.2, and the globalMapLeafSize parameter was set to 0.05.

Given the confined nature of indoor environments and the need for detailed mapping, parameter values across all SLAM algorithms were adjusted empirically to improve robustness and increase point cloud density. This helped maintain tracking stability and capture finer structural features.

Data Collection

At the test sites, the Raspberry Pi with the connected LiDAR mounted on Spot was powered on and it automatically logged into the robot's hotspot. Also, Spot's controller and a notebook were logged into the hotspot. Like this, we were able to access the Raspberry Pi via an SSH connection from a laptop, for example to control the LiDAR.

Since time synchronization is crucial for the evaluation, and there was no direct way to sync the Raspberry Pi/LiDAR and the total station, we set the time from an online time server based on Global Navigation Satellite System (GNSS). The total station's time was then set via GNSS connection. This process provides rough time synchronization, which can be refined during data processing to align positions accurately. On the laptop a Network Time Protocol (NTP) time server was set up, so that the Raspberry Pi's time could be set.

Using Spot's controller and Autowalk feature, we defined a trajectory for the robot to follow during data collection. To do this, fiducials were placed in the environment, one of which is visible in Figure 1. While planning the robot's trajectory, actions to control the data collection -such as starting, stopping, or pausing- can be integrated via the previously launched middleware. This setup allows the robot to autonomously capture onsite data, minimizing the risk of unintended objects being recorded in the point cloud. A key limitation of the total station is its need for a direct line of sight, making indoor use challenging. To address this, we established a reference point network within a local coordinate frame, enabling accurate trajectory recording. During total station relocation, data capture was paused and resumed at predefined points to prevent disruptions of the SLAM algorithm. The collected data then were later merged in post-processing.

Once everything is set up, we can start tracking the prism mounted above the LiDAR sensor using the total station and begin the data collection process.

Data Processing

The LiDAR and IMU data were stored in a PCAP file along with the start timestamp of the recording. The positions captured by the total station, along with their timestamps, were saved in a CSV file. For the SLAM algorithms, however, the data needed to be formatted as Robot Operating System (ROS) messages. For that the PCAP files were converted to ROS bag files using Pcap-to-Bag (Korda, 2024). These bag files hold timestamped messages of different types containing the LiDAR and the IMU data. In cases where data acquisition included pauses, two bag files were generated. Those files were then merged, and the timestamps for the message types in the second bag file were adjusted as shown in equation 1:

$$t_{bag2} = t_{bag2} - t_{bag2}^{start} + t_{bag1}^{start} + t_{bag1}^{duration} + \overline{\Delta t}_{bag1} \quad (1)$$

This process effectively combines the paused data to simulate a continuous trajectory, allowing SLAM algorithms to utilize the data as if it were captured in a single, uninterrupted session. Subsequently, SLAM algorithms can be used to generate both the trajectory and the point clouds.

For the evaluation, a precise time synchronization is crucial. Initially, coarse synchronization is achieved by matching the beginning of the total station trajectory with the start timestamp of the data collection. For finer synchronization, the total station timestamps are shifted within a ±10-second window around the coarse synchronization, with a step size of 0.01 seconds. Each time shift is evaluated using Umeyama alignment of the trajectories, considering only points from the first 1.5 meters to minimize drift effects. The RMS APE of the aligned trajectories is computed, and the time shift with the smallest value is selected. In cases where data acquisition included pauses, time synchronization is more complex. Suitable segments from the total station trajectory were extracted. The boundaries of these segments are determined by the closest timestamp using the following equation 2, 3, 4 and 5:

$$t_1^{start} \approx t_{bag1}^{start} + t_{offset} \quad (2)$$

$$t_2^{start} \approx t_{bag2}^{start} + t_{offset} \quad (3)$$

$$t_1^{end} \approx t_{bag1}^{start} + t_{offset} + t_{bag1}^{duration} \quad (4)$$

$$t_2^{end} \approx t_{bag2}^{start} + t_{offset} + t_{bag2}^{duration} \quad (5)$$

Here, t_{offset} is initially set to zero. The timestamps in the first segment are roughly synchronized, and the timestamps in the second segment are adjusted as shown in equation 6:

$$t_2 = t_2 - t_{bag2}^{start} + t_{bag1}^{start} + t_{bag1}^{duration} + \overline{\Delta t}_{bag1} \quad (6)$$

The two segments are then combined to create a continuous trajectory. This preliminary synchronization allows us to refine t_{offset} using the fine synchronization method described earlier. With the updated t_{offset}, the total station data can then be segmented again but this time with higher accuracy. The adjusted segments are then merged to form a finely synchronized total station trajectory.

Metrics for Evaluation

We first compared the trajectories to the ground truth trajectory captured by the total station. To achieve this, the time-synchronized SLAM trajectories were aligned with the total station trajectory using the Umeyama algorithm. Only points within the first 1.5 meters of the trajectory were used for alignment to identify potential temporal or spatial drift. Following this alignment, the APE and RPE were calculated. While APE provides a measure of global performance, RPE allows for a comparison of the SLAM algorithms' local performance. The RPE was specifically calculated for consecutive frames with a distance greater than 30 cm to minimize noise and to mitigate the impact of larger RPE values caused by brief tracking interruptions of the total station. Subsequently, the point clouds generated by the SLAM algorithms were compared with reference point clouds previously created using the Riegl VZ-400i and Riegl VZ-600i terrestrial laser scanners. This comparison involved first aligning the clouds by selecting equivalent point pairs and then applying an ICP algorithm for fine alignment. After alignment, the cloud-to-cloud distances were calculated using the nearest neighbor distance method, with a maximum distance of 25 cm in the hallway and stairwell environments, and 50 cm in the cellar environment. Although a point-to-plane distance approach with local models was also possible, the high density of the reference point clouds meant this would only produce minor differences. The resulting distances reflect errors from SLAM positioning, the accuracy of the LiDAR, and any discrepancies or gaps in the reference point clouds.

Results

Table 2: Results in comparison to the ground truth trajectory.

Environment	Algorithm	RMS	
		APE (cm)	RPE (cm)
Hallway	KISS-ICP	**5.88**	5.87
	FAST-LIO2	7.37	**5.05**
	LIO-SAM	7.32	6.50
Cellar	KISS-ICP	26.16	5.53
	FAST-LIO2	**9.15**	**5.16**
	LIO-SAM	35.32	7.37
Stairwell	KISS-ICP	8.32	3.90
	FAST-LIO2	**7.35**	**3.24**
	LIO-SAM	10.91	4.90

Table 3: Results in comparison to the ground truth point cloud.

Environment	Algorithm	mean (cm)	points
Hallway	KISS-ICP	8.80	728.640
	FAST-LIO2	**3.70**	6.807.842
	LIO-SAM	5.10	269.745
Cellar	KISS-ICP	16.4	1.210.300
	FAST-LIO2	**5.60**	8.456.519
	LIO-SAM	11.7	356.900
Stairwell	KISS-ICP	8.60	396.095
	FAST-LIO2	**4.00**	12.094.288
	LIO-SAM	8.30	289.627

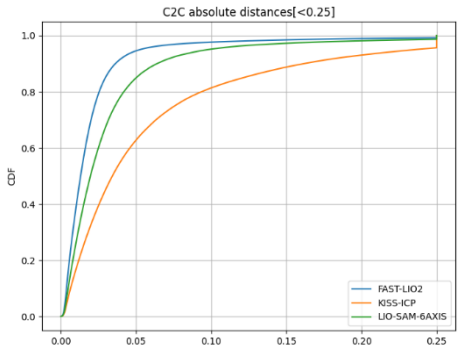

Figure 5: Cumulative Distribution Function for the different environments and algorithms (Hallway).

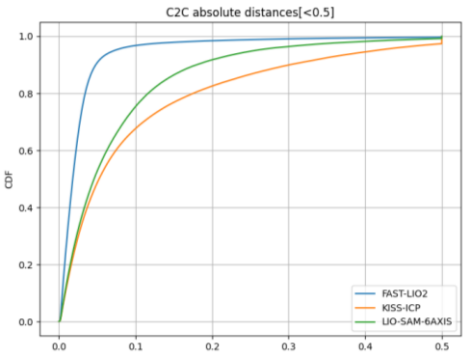

Figure 6: Cumulative Distribution Function for the different environments and algorithms (Cellar).

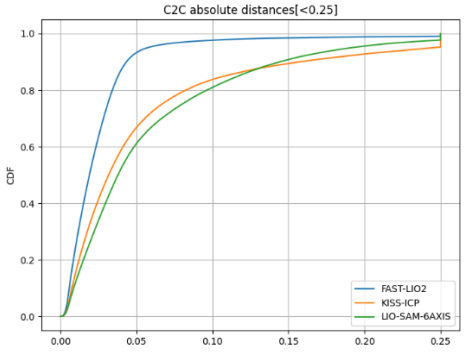

Figure 7: Cumulative Distribution Function for the different environments and algorithms (Stairwell).

The RMS APE (Figure 8 - 10) and RPE are presented in Table II. Table III provides the mean cloud-to-cloud distance and the number of points within each point cloud. Furthermore, the cumulative distribution function (CDF) of the point clouds as a function of the cloud-to-cloud distance in the respective environments is depicted in Figures 5, 6 and 7. In the hallway environment, all SLAM algorithms demonstrated good performance, with RMS APE values ranging between 6 and 7 cm and mean cloud distances between 3 and 8 cm. KISS-ICP achieved the lowest RMS APE at 5.88 cm, while FAST-LIO2 had the smallest mean cloud-to-cloud distance at 3.7 cm. Generally, the APE increases with distance for all SLAM algorithms. It is noticeable that FAST-LIO2 and KISS-ICP exhibit slightly poorer point cloud coverage at the start of the trajectory, particularly on the closer side of the hallway. This may be due to the minimum LiDAR range, which could mean these parts of the hallway were only covered later in the trajectory.

In the cellar environment, notable differences between the SLAM algorithms become evident. The RMS APE ranges from 9 to 35 cm, and the mean cloud-to-cloud distance varies between 5 and 16 cm. FAST-LIO exhibited the smallest values, with an RMS APE of 9.15 cm and a mean cloud-to-cloud distance of 5.6 cm. All algorithms struggled with positioning, especially after the first curve. The APE indicates that the error propagated further, though it decreased upon reaching the starting position for FAST-LIO2 and LIO-SAM-6AXIS. This is likely due to the loop closure in LIO-SAM-6AXIS and the size of the internal map in FAST-LIO2, which contribute to similar effects. Examining the point clouds from the cellar environment, FAST-LIO2 shows excellent coverage, with larger distance areas primarily resulting from reflections or discrepancies in the reference cloud. In contrast, KISS-ICP and LIO-SAM-6AXIS exhibit significant deviations in the last third of the trajectory, with KISS-ICP showing double surfaces in the starting area.

In the stairwell environment, the RMS APE values for the SLAM algorithms range from 7 to 11 cm, and the mean cloud distances range from 4 to 9 cm. Again, FAST-LIO2 achieved the best values, with an RMS APE of 7.35 cm and a mean cloud-to-cloud distance of 4 cm. Both the APE and RPE show a noticeable increase in error when the robot enters the stairs, partly due to the SLAM algorithm but also because prism mounted on top of the sensor may shift sideways due to the robot's tilt on the stairs, causing errors of about 3-5 cm. This could explain abrupt errors in RPE at the stair edges. For FAST-LIO2, the APE mainly shows errors on the first stair steps, whereas KISS-ICP and LIO-SAM-6AXIS exhibit significant errors in the lower floor area. The resulting point clouds reveal that FAST-LIO2 provides good coverage of the environment, with slight shearing on the upper floor. In contrast, KISS-ICP and LIO-SAM-6AXIS show considerable shearing in the lower floor area, while the upper floor is better represented, with less distortion. Looking at Table III, we can see that the resulting point clouds have very different densities, with LIO-SAM-6AXIS having the lowest density, FAST-LIO2 having the highest, and KISS-ICP falling in between. When examining the point clouds, it is evident that for FAST-LIO2 even smaller surfaces are captured with precision, and there is minimal noise.

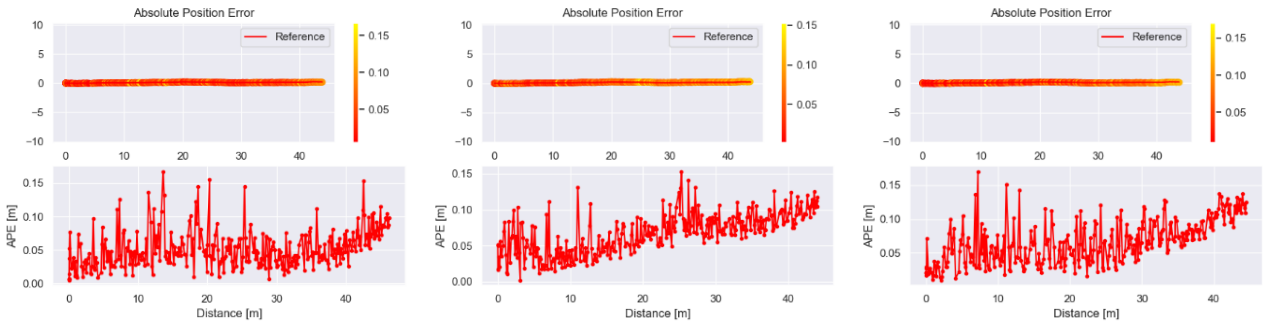

Figure 8: APE in the evaluation environment hallway.

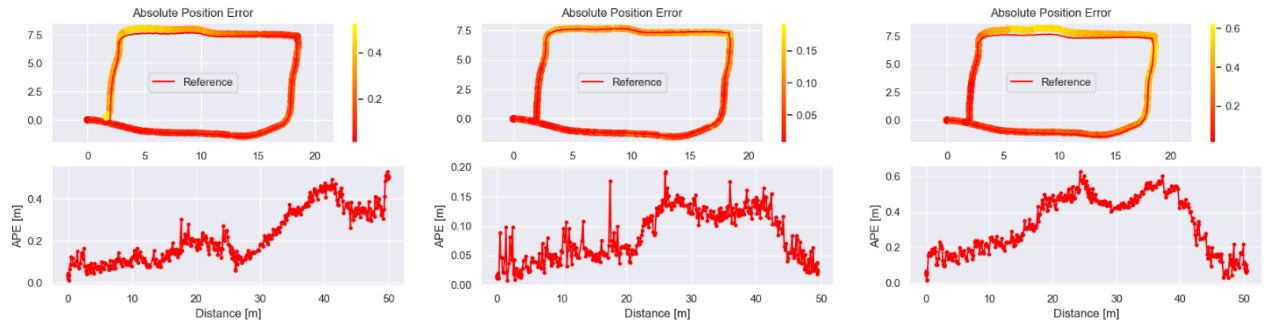

Figure 9: APE in the evaluation environment cellar.

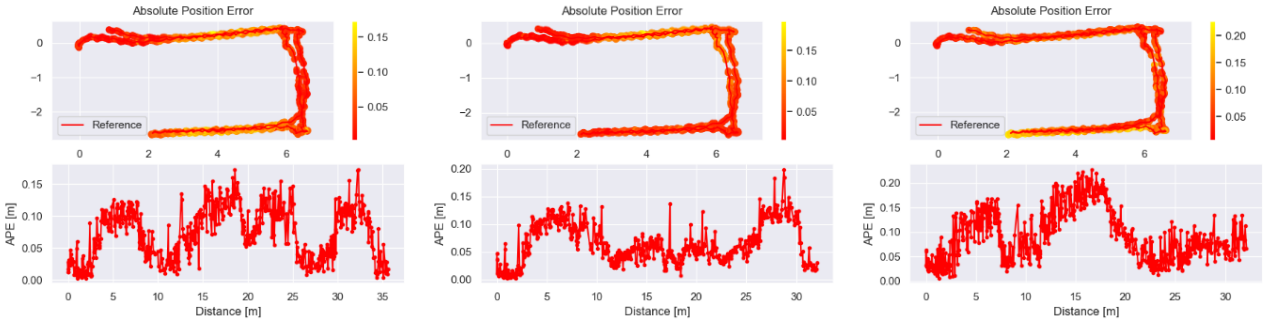

Figure 10: APE in the evaluation environment stairwell.

Discussion and Conclusions

We compared three popular SLAM algorithms in terms of their mapping capabilities using an Ouster OS1-128 LiDAR mounted on a Boston Dynamics Spot robot. Three unique indoor scenarios were evaluated, and the results highlighted the good performance of the FAST-LIO2 algorithm for our specific tasks.

For the evaluation the APE and RPE were calculated using synchronized timestamps, making the errors less susceptible to the temporal resolution of the total station and the differences between SLAM algorithms compared to a distance-based approach. By using the proposed method to finely synchronize timestamps and artificially combine sensor and total station data in cases where data collection was interrupted, we successfully established a way to compare SLAM algorithms in environments like indoor settings where a direct line-of-sight for the entire trajectory is not possible. While directly synchronizing the timestamps of the total station and the LiDAR would be the preferred approach, the proposed method offers a simple and effective alternative when direct synchronization is not feasible. This method eliminates the need for additional hardware or preparation when comparing different SLAM trajectories. Additionally, we introduced a method for generating ground truth trajectory data using a total station without direct line of sight over the whole trajectory.

For comparing the point clouds, we first aligned the generated point clouds with the reference point cloud by selecting equivalent point pairs and then finely aligning the point clouds using an ICP algorithm. While this approach is valid for evaluating the overall quality of the generated point clouds from a global perspective, an alternative method would involve moving the reference point clouds into the total station's reference frame. This could be done by placing markers in the environment, measuring them with the total station, and then referencing them in the reference point cloud. The generated point clouds could then be aligned using the same transformation from the trajectory alignment. This approach would allow for the examination of the generated point clouds for potential temporal or spatial drift.

The results show the superior performance of FAST-LIO2 compared to KISS-ICP and LIO-SAM-6AXIS, achieving the lowest RMS APE in 2 out of 3 environments and the lowest mean cloud-to-cloud distance in all environments. Additionally, the resulting point clouds exhibited high density with minimal noise. FAST-LIO2 also managed to achieve minimal drift when revisiting previously observed positions, likely due to its large local map, even in the absence of an explicit loop closure module.

In summary, we demonstrated that by equipping a Boston Dynamics Spot robot with a Raspberry Pi and an Ouster OS1-128 LiDAR sensor we can automate geometric as-built documentation with high quality results in a relatively cost-effective manner. By combining this data with open-source LiDAR SLAM algorithms, it is possible to create detailed maps of environments, which can for example be used to derive BIM models using Scan-to-BIM methods for as-built/as-is documentation. An extensive review of possible Scan-to-BIM methods is provided by Abreu et. al. (2023). Also, it is conceivable that such a setup could be employed on modern construction sites to automatically capture the current state at regular intervals. Further evaluations could focus on the performance of these SLAM algorithms in more diverse and complex outdoor settings and could investigate the integration of semantic information into the point clouds to enable more comprehensive as-built documentation, including object recognition and classification.

References

Abreu, N., Pinto, A., Matos, A. and Pires, M. 2023. Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. ISPRS International Journal of Geo-Information. 12(7), p.260. Available from: https://doi.org/10.3390/ijgi12070260

Blut, C., Becker, R., Kinnen, T., Schluetter, D., Emunds, C., Frisch, J., Heidermann, D., Wenthe, M., Rettig, T., Baranski, M., van Treeck, C. and Blankenbach, J. 2024. Optimizing Building Energy Systems through BIM-enabled georeferenced Digital Twins. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XLVIII-4-W11-2024, pp.1–8.

Blut, C. and Blankenbach, J. 2022. *DigiPark - Digitization in parking garage maintenance*. Research Publishing: Singapore.

Blut, C., Kinnen, T., Schellong, F., Heidermann, D., Bleimann-Gather, G. and Blankenbach, J. 2023. X-Reality for intuitive BIM-based as-built documentation.

Frosi, M., & Matteucci, M. (2022). Art-slam: Accurate real-time 6dof lidar slam. IEEE Robotics and Automation Letters, 7(2), 2692-2699.

Grupp, M. 2024. MichaelGrupp/evo. [Accessed 29 November 2024]. Available from: https://github.com/MichaelGrupp/evo.

Halder, S., Afsari, K., Chiou, E., Patrick, R. and Hamed, K.A. 2023. Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study. *Journal of Building Engineering*. 65, p.105814.

Hu (2023) LIO_SAM_6AXIS [Source code]. Available at: https://github.com/JokerJohn/LIO_SAM_6AXIS (Accessed: 26 March 2025).

Jung, M., Jung, S., & Kim, A. (2023). Asynchronous multiple lidar-inertial odometry using point-wise inter-lidar uncertainty propagation. IEEE Robotics and Automation Letters, 8(7), 4211-4218.

Kaden, R., Clemen, C., Seuß, R., Blankenbach, J., Eichhorn, A., Donaubauer, A., Gruber, U. and Fritsch, M. 2024. Leitfaden Geodäsie und BIM.

Kim, P., Chen, J. and Cho, Y.K. 2018. SLAM-driven robotic mapping and registration of 3D point clouds. *Automation in Construction*. 89, pp.38–48.

Kim, G., & Kim, A. (2018). Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4802-4809). IEEE.

Kopsida, M., Brilakis, I. and Vela, P.A. 2015. A Review of Automated Construction Progress Monitoring and Inspection Methods *In*: *Proc. of the 32nd CIB W78 Conference*. Eindhoven, Netherlands.

Korda, K. 2024. Krishtof-Korda/pcap-to-bag. [Accessed 29 November 2024]. Available from: https://github.com/Krishtof-Korda/pcap-to-bag.

Lee, D., Jung, M., Yang, W. and Kim, A. 2023. LiDAR Odometry Survey: Recent Advancements and Remaining Challenges.

Merkle, D. and Reiterer, A. 2023. Automated Method for SLAM Evaluation in GNSS-Denied Areas. *Remote Sensing*. 15(21), p.5141.

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

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C. and Rus, D. 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping.

Umeyama, S. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 13(4), pp.376–380.

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J. and Stachniss, C. 2023. KISS-ICP: In Defense of Point-to-Point ICP - Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)*. [Online]. [Accessed 29 November 2024]. Available from: https://www.ipb.uni-bonn.de/wp-content/papercite-data/pdf/vizzo2023ral.pdf.

Xu, W., Cai, Y., He, D., Lin, J. and Zhang, F. 2021. FAST-LIO2: Fast Direct LiDAR-inertial Odometry.

Yarovoi, A. and Cho, Y.K. 2024. Review of simultaneous localization and mapping (SLAM) for construction robotics applications. *Automation in Construction*. 162, p.105344.

Zhang, L., Helmberger, M., Fu, L.F.T., Wisth, D., Camurri, M., Scaramuzza, D. and Fallon, M. 2023. Hilti-Oxford Dataset: A Millimetre-Accurate Benchmark for Simultaneous Localization and Mapping.

Zhang, Y., Shi, P., & Li, J. (2024). 3D LiDAR SLAM: A survey. The Photogrammetric Record, 39(186), 457-517.