Robust High-Precision LiDAR Localization in Construction Environments

Andrew Yarovoi[1] Pengyu Mo[2] Yong Kwon Cho[3]

[1]Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, USA
[3]School of Civil and Environmental Engineering, Georgia Institute of Technology, USA

ayarovoi3@gatech.edu, pmo30@gatech.edu yong.cho@ce.gatech.edu

Abstract -

Accurate localization plays a crucial role in the effective operation of autonomous robotics systems, especially in dynamic environments such as construction sites. Simultaneous Localization and Mapping (SLAM) utilizing LiDAR sensors has emerged as a popular solution due to its ability to function without external infrastructure. However, existing algorithms exhibit significant shortcomings. Despite current methods achieving high accuracy over long trajectories, they struggle with precision and reliability in complex indoor environments. This paper introduces a novel feature-based LiDAR SLAM system designed to address these limitations and enhance short-term precision and overall robustness. The proposed system is evaluated using both existing datasets and a physical robot platform, addressing the limitations of current implementations and showcasing improved performance in challenging real-world scenarios, particularly in construction environments.

Keywords -

Simultaneous Localization and Mapping; SLAM; LiDAR, Localization; Navigation

1 Introduction

Localization is a critical component of nearly all autonomous robotics systems. Accurately understanding the robot's pose relative to its environment is often crucial for navigation and manipulation tasks. While GPS can sometimes be sufficient for outdoor environments, more complex approaches are typically needed for indoor environments due to significant signal attenuation by the building. In warehousing and manufacturing, autonomous robots are conventionally localized using visual markers placed in the environment or external camera systems with fixed positions. These approaches have enabled increased robotic automation which has lowered costs, increased workforce productivity, and improved efficiency [1]. However, these approaches are often not feasible in construction due to construction sites continuously evolving throughout the building process. This makes setting up external localization systems difficult.

One popular solution is to use onboard sensors to perform simultaneous localization and mapping (SLAM), allowing autonomous systems to localize in previously unmapped environments. SLAM can provide high-accuracy and precise positional estimates in indoor environments, without the need for any external infrastructure. One popular onboard sensor for performing SLAM is LiDAR, which typically uses the time of flight data of a laser to measure distances to nearby obstacles. These distance measurements are converted to a point cloud, providing a 3D representation of the environment. LiDAR has numerous advantages over cameras such as direct measurement of depth and invariance to lighting conditions. This eliminates scale ambiguities and allows for robust measurements, even in harsh or dim lighting conditions such as those commonly present in night-shift work operations. Additionally, LiDARs are not impacted by textureless or highly repetitive textured environments, which are common in construction (e.g. unfinished/unfurnished rooms, brick walls).

Over the last couple of decades, various algorithms and frameworks have been proposed for performing LiDAR-based SLAM. One of the most popular and robust of the recent methods is LIO-SAM [2]. LIO-SAM uses a range-image generated from a point cloud to extract LOAM [3] features. It then uses frame-to-frame matching and frame-to-global-map matching, along with GTSAM [4] to generate a 3D point cloud of the environment and localize the LiDAR within it. Additionally, LIO-SAM tightly couples the LiDAR and IMU, allowing for point cloud deskewing and full utilization of the IMU data in the factor graph optimization. Unlike other more recent point-based methods such as ART-SLAM [5], LIO-SAM is feature-based and thus more computationally efficient. This allows it to run on smaller robots without a GPU. As such, it has become a popular option for mobile robots and is used extensively in the research community [6, 7, 8, 9].

While LIO-SAM has been shown effective on multiple datasets, it has several limitations that make it sub-optimal for real-world localization of a navigating autonomous sys-

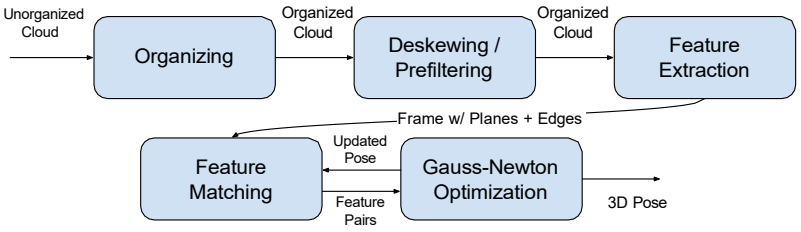


Figure 1. An overview of the proposed algorithm

tem. Despite achieving high long-term trajectory accuracy, LIO-SAM often provides poor short-term precision. This results in large short-term errors in the pose estimate that can lead to navigational issues and poor map quality. These issues are particularly evident when working with robotic systems that exhibit high vibrations (such as quadruped robots) or systems utilizing cheaper, lower-quality IMUs. Additionally, the algorithm lacks robustness in complex environments where limited visibility of the ground planes and LIO-SAM's lack of consideration for surface normals leads to weak constraints on the roll, pitch, and z-height. This can lead to complete loss of tracking in many confined indoor environments, such as stairways and small rooms. These limitations make current implementations unreliable in common construction environments, endangering the robot and the people around it.

To address these issues, we present a novel feature-based LiDAR-inertial SLAM system that significantly improves short-term precision and overall robustness. The proposed system is validated on existing datasets and a physical robotics platform.

2 Methodology

Figure 1 shows an overview of the different modules in the proposed algorithm. Unlike existing approaches, the proposed algorithm first organizes the cloud into a CxN array (organized cloud), where C is the number of channels in the LiDAR, N is the number of samples taken per revolution of the LiDAR, and each element in the array is a point storing its x, y, z, and intensity values. This preprocessing step speeds up computations of later steps in the pipeline, enabling real-time performance. After organizing, the cloud is deskewed using IMU data to undo any distortions caused by rotations of the LiDAR sensor over the capture interval of the point cloud. This is necessary as LiDARs typically output their data as scans (one full rotation of the laser array) which contains points captured at different times over the scan period. Deskewing removes any distortions caused by rotations of the sensor over the scan period. The deskewed point cloud then undergoes feature extraction where planar points (Figure 2) and edge points (Figure 3) are extracted from the cloud using novel proprietary feature extractors. Planar points are further clustered into individual planes. Unlike LIO-SAM which labels points as planar based on the local roughness of the range image, we consider the full 3D positions of each point and its neighbors to more reliably identify planar surfaces and remove non-planar points. Additionally, our approach efficiently estimates the normals of the points, providing us with richer features and additional information during feature matching and pose optimization. The individual planes, each containing a point cloud of their constituent points, along with an edge cloud comprising all the edge points, are subsequently assembled into a Frame object. This Frame is initialized with an initial pose derived from the last predicted pose and the IMU's orientation estimate.

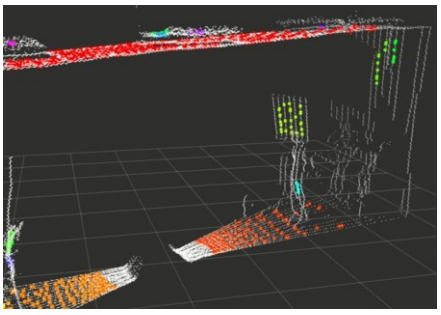


Figure 2. Example of extracted plane points

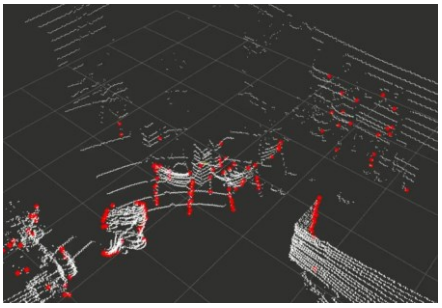


Figure 3. Example of extracted edge points

An iterative process is then used to match the features and optimize the pose of the current frame with respect to the key frames. At each iteration, a 3-stage matching approach is used to first match the Frame to its nearest key frames, then match the Frame's planes to the selected key frames' planes, and finally match planar and edge points in the Frame to the planar and edge points stored in the matched planes and key frames' edge clouds. The feature matching step results in a set of planar and edge correspondences that are used to calculate multiple error metrics. These metrics are minimized with respect to the Frame's pose using Newton's method. The Frame's estimated pose is then updated and the iterations are repeated until we converge on a pose or have repeated a set number of iterations. The final pose of the LiDAR is returned as the final output of the system. If needed, the frame is added to the key frames to iteratively grow the map.

Ultimately, our approach differs from LIO-SAM in three main ways. First, we organize the cloud and use the full 3D positions of the points during feature extraction, instead of using the range image. The organization of the point cloud allows us to compute features in a comparable amount of time to LIO-SAM while utilizing more of the point cloud's information. Second, we do not use LOAM features, and instead use novel feature extractors that extract more reliable planes and edges. Our edge detector explicitly handles edges caused by occlusions and our plane detector efficiently estimates normals, providing more information for pose optimization. Our use of different feature extractors also enables us to utilize 3-stage feature matching which exploits the additional information captured by the features to improve matching accuracy. Lastly, our method uses key framing to efficiently represent the map, enabling frame-to-global-map matching for every frame. This results in a highly memory-efficient map representation and limits the short-term drift, improving the short-term precision of the localization estimate.

An additional difference in the current implementation is that we do not use a factor graph to optimize the global map or fuse IMU orientation estimates into our pose graph. This is a limitation of our current approach as it can lead to larger long-term drift for very long sequences. However, we plan to address this issue in future work by incorporating our design with GTSAM [4].

3 Experiments

To evaluate the performance of the proposed approach against a baseline, the algorithm is quantitatively compared to LIO-SAM on an existing construction site dataset, as well as qualitatively evaluated on physical robotics hardware. LIO-SAM provides a good baseline as it requires the same sensors (a 360-LiDAR and an IMU), is also feature-based, and requires similar computational resources. For numerical analysis, we primarily evaluate the accuracy of the localization estimates, as localization is the primary use case of real-time SLAM algorithms for most systems. Additionally, localization can be evaluated more directly than map quality and better localization directly leads to improved mapping.

3.1 Hilti Dataset

To evaluate the absolute positional accuracy, both algorithms were tested on some of the additional sequences provided by the Hilti SLAM Challenge 2022 dataset [10]. This dataset provides ROS bag recordings of LiDAR and IMU sequences collected from various construction and indoor environments using a Hesai Pandar XT-32 LiDAR. Vitally, the dataset also provides millimeter-accurate synchronized ground truth poses collected with a motion-capture system. This allows direct comparison of the predicted and ground truth trajectories.

To enable both algorithms to process the data, the point clouds were preprocessed to convert them into a Velodyne point format (does not change any data, but changes point format). Additionally, the IMU data was preprocessed as both algorithms expect the IMU to provide fused orientation estimates. These were generated using the open-source imu_filter_madgwick package [11]. To provide a wide range of difficulties and environments, 5 sequences from the dataset were tested. These are Exp04, Exp05, Exp06, Exp14, and Exp18.

Table 1. Dataset statistics for Hilti sequences calculated based on the ground truth poses

Dataset	Max Pitch / Roll (°)	Mean Ang Vel (°/s)	Mean Lin Vel (m/s)	Max Ang Vel (°/s)	Max Lin Vel (m/s)
Exp04	10.968	19.970	0.606	142.194	1.886
Exp05	17.324	17.110	0.561	146.696	1.616
Exp06	58.033	38.971	0.617	263.522	2.408

The first three scans were captured on three floors of a real-world indoor construction site with progressively more aggressive motions. The three datasets all include variations in Z-height, open and confined spaces, and tilting in both pitch and roll. Table 1 provides some statistics generated from the ground truth poses provided by the datasets. Exp06 in particular offers very fast motions and aggressive rotations. Both Exp04 and Exp05 were designated as easy difficulty by the dataset creators, while Exp06 was designated as medium difficulty due to the fast motions.

Captured in more demanding indoor settings, Exp14 and Exp18 both present challenging sequences with geometric ambiguity and confined spaces. Notably, Exp14

showcases a rectangular staircase, while Exp18 incorporates a spiral staircase at the beginning and end of the sequence, resulting in their classification as medium and hard difficulty, respectively.

Table 2. Results from the Hilti Sequences, N/A indicates an algorithm lost tracking and could not recover

Seq	Algorithm	Trans RMSE (m)	Trans SD (m)	Rot RMSE (°)	Rot SD (°)
Exp04	LIO-SAM	0.1670	0.0879	1.486	0.355
	Ours	**0.1147**	**0.0566**	**0.966**	**0.493**
Exp05	LIO-SAM	**0.0945**	**0.0450**	**0.873**	**0.336**
	Ours	0.1124	0.0564	1.818	1.055
Exp06	LIO-SAM	**0.3599**	**0.2274**	**2.258**	**1.053**
	Ours	0.4825	0.2578	4.098	1.707
Exp14	LIO-SAM	N/A	N/A	N/A	N/A
	Ours	**0.7966**	**0.5292**	**3.985**	**1.559**
Exp18	LIO-SAM	N/A	N/A	N/A	N/A
	Ours	**0.7979**	**0.5713**	**10.433**	**3.854**

The two algorithms were evaluated on all five sequences using a Ryzen 5600H processor, processing the data in real-time. The predicted poses were captured and stored in a text file, and then synchronized with the ground truth poses using timestamps. For evaluation, translation errors were calculated using the Euclidean distance between the estimated and ground truth trajectories. The rotation errors were calculated as the smallest angle between the estimated and ground truth orientations. The root-mean-square (RMSE) and the standard deviation (SD) of the translational and rotational errors are reported to evaluate the accuracy and consistency of the estimated trajectories in Table 2.

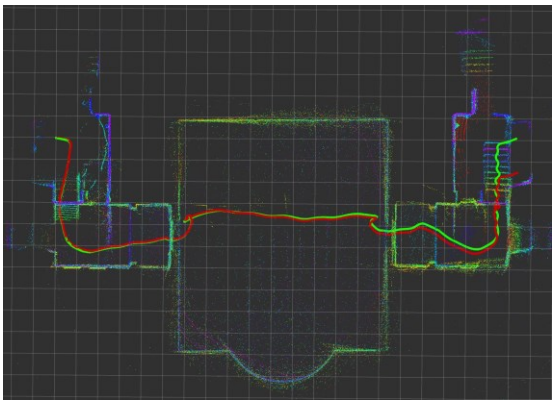


Figure 4. Predicted trajectory (red) versus ground truth (green) for Exp14 using our approach

As shown in Table 2, our approach achieved better results in 3 of the 5 sequences. Importantly, Exp14 and Exp18 show that our approach is more robust than LIO-SAM by completing the sequence. LIO-SAM on the other hand lost tracking in the first 5 seconds of both sequences due to starting in confined spaces and having to navigate stairways. Additionally, most of the errors in Exp14 occurred in the last few seconds due to the cloud being highly geometrically ambiguous as a result of the translational symmetry of the stairway (Figure 4). Exp18 also highlights the generalization capability of our approach to various built environments. While the other sequences primarily feature standard built environments, Exp18 was captured in a gallery with curved walls, ornate columns, and tight walkways. Despite there being few truly planar surfaces, our plane extractor was able to identify locally planar surfaces and outperformed LIO-SAM, completing the sequence.

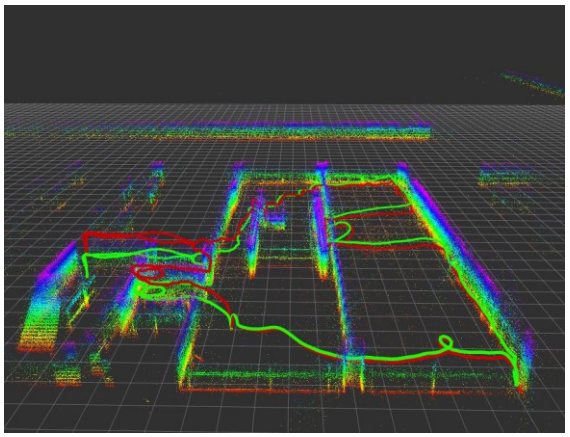
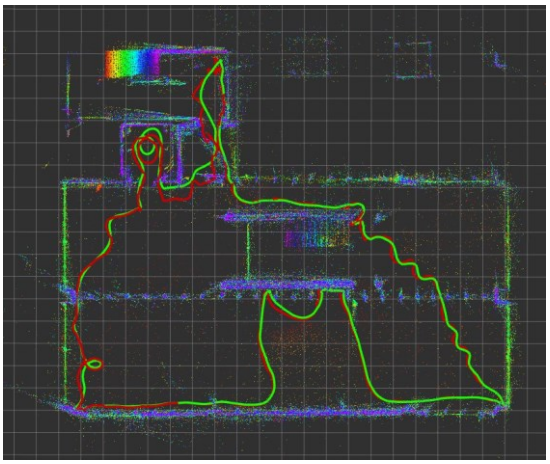
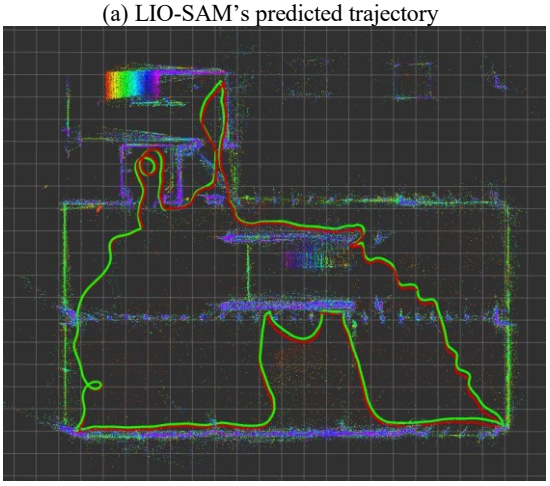


Figure 5. Angled view of the predicted trajectory for Exp06 using our approach

Our algorithm also achieved better performance on Exp04. Upon initial inspection, it seems that LIO-SAM exhibited slightly better performance in Exp05 and Exp06. However, as shown in Figure 5 and Figure 6, our approach produced a smoother and more locally accurate trajectory, despite experiencing some additional drift in roll and pitch that resulted in comparable but slightly higher overall RMSE. This additional drift is due to how our algorithm integrates the IMU data into our pose estimates. Currently, our algorithm only utilizes the IMU data for initial pose estimation and point cloud deskewing. However, IMUs also generate attitude measurements, which provide absolute constraints on the roll and pitch of the system. Since our system does not fuse the IMU orientation estimates with our final predicted pose, our algorithm is more susceptible to long-term drift in the roll and pitch axes for longer sequences with aggressive motions. Therefore, even though our system provided better short-term estimates for Exp 05 and Exp06, the full trajectory errors were slightly higher. Future work will focus on reducing these errors by reintegrating the IMU data into the LiDAR pose estimate follow-



(a) LIO-SAM's predicted trajectory



(b) Our predicted trajectory

Figure 6. The predicted (red) and ground truth (green) trajectories for Exp06 (cloud generated from ground truth for reference)

ing Gauss-Newton optimization. Nevertheless, even with our current implementation, the preference for a smoother and locally accurate trajectory for navigation purposes may outweigh the marginal gain in long-term positional accuracy offered by LIO-SAM. A similar story was observed in Exp05.

3.2 Quadruped Robot Dataset

While the Hilti SLAM Challenge dataset provides real-world sequences collected via a high-quality hand-held system, it may not be fully representative of the types of trajectories followed and the vibrations produced by many robotics systems. To evaluate the algorithm on a robotics platform, our algorithm and LIO-SAM were evaluated on a dataset collected in an indoor environment using a teleoperated Unitree Go1 quadruped robot equipped with a Velodyne VLP-16 LiDAR and a YostLabs 3-Space Micro USB IMU (Figure 7). Compared to the Hilti Dataset, the data produced by this system is much noisier due to lower IMU precision, high vibrations due to the robot's walking gait, and a lower precision LiDAR. Additionally, the LiDAR has a very limited field-of-view (30°) and lower resolution (only 16 channels), providing an additional challenge to the SLAM system. Both algorithms took less than 50 ms to process each frame.

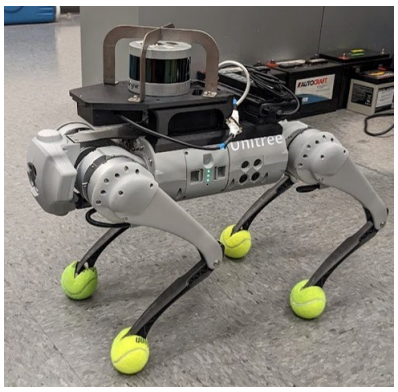
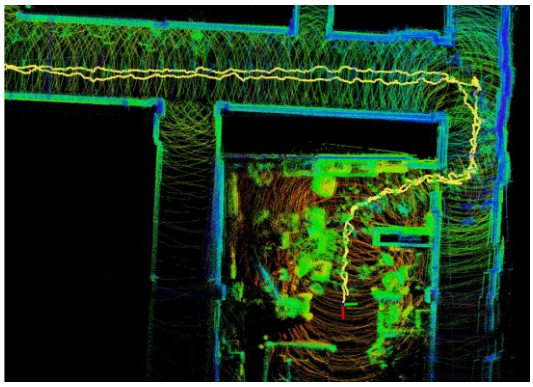


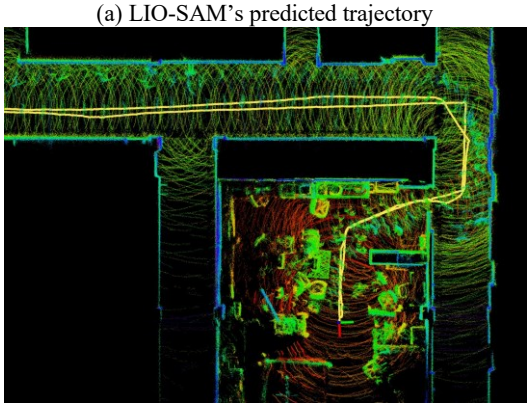
Figure 7. Lab robot used to collect dataset

Figure 8 shows a close-up of the resultant trajectory and generated map. While the overall trajectories produced by both algorithms were similar, the difference in short-term precision becomes apparent. During the test, LIO-SAM suffered from poor short-term precision in the predicted pose. This is likely due to their strategy of performing frame-to-global map matching at a slower rate than frame-to-frame matching, as well as vibrations picked up by the IMU. Our approach did not suffer from these issues and produced a much smoother and more accurate trajectory. Additionally, our 3-stage matching approach can use normal information to distinguish between the two sides of a wall, accurately recovering the thicknesses of the walls. Meanwhile, LIO-SAM uses only proximity during matching, meaning both sides of the wall typically merge into a single plane, making it difficult to recover the geometry of the building. This has significant implications for building information modeling (BIM) as wall thicknesses are often of great interest for accurately modeling the building geometry.

As shown in Figure 9, the higher precision in predicted poses also leads to improved map quality and sharper point clouds. The difference is most evident when comparing the thin legs of the chairs, and the guard rail by the stairs. Improved map quality is of significant importance for tasks such as object detection, where sharper maps translate to lower noise in the object point clouds, capturing finer details of the object and improving the odds of accurate classification or segmentation. Additionally, the reduction in

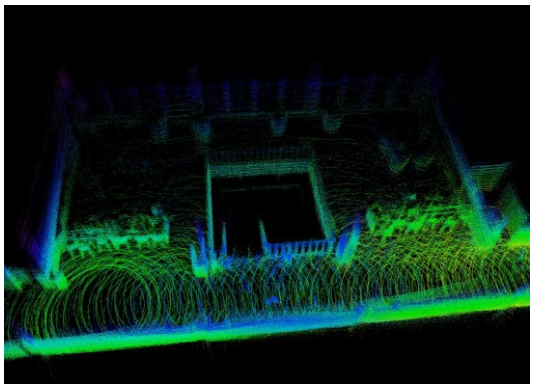


(a) LIO-SAM's predicted trajectory

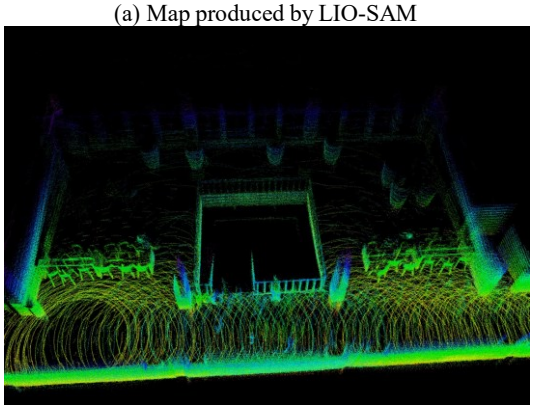


(b) Our predicted trajectory

Figure 8. The predicted trajectory (yellow) and resultant map for the quadruped robot dataset (from same viewing angle)



(a) Map produced by LIO-SAM



(b) Map produced by our approach

Figure 9. Example maps recovered by LIO-SAM and our approach using the quadruped robot

noise in the object point clouds can improve the precision of measurements between object features, enabling more accurate object modeling.

Local stability in the estimated pose is also crucial for autonomous navigation. Poor short-term positional drift can cause the robot to temporarily assume it is too close or inside an obstacle, causing it to react unpredictably or sporadically. This can lead to jerky or dangerous recovery behaviors where the robot attempts to continuously re-plan its trajectory around obstacles. Our algorithm generates smoother, more accurate local trajectories than LIO-SAM, which can lead to less time spent re-planning and more consistent robot behavior.

4 Conclusion

In this study, we present a novel, computationally lightweight, LiDAR SLAM system for real-time localization and mapping. The proposed system was designed as a stand-alone C++ package to be used with 360 LiDAR scanners. The system was evaluated on a real construction dataset as well as on a quadruped robot. The system achieved state-of-the-art performance on the majority of the datasets and demonstrated improvements in the pose estimate's robustness and precision.

While the proposed system provides clear advantages for the localization of autonomous systems, it does have some limitations. The current approach only uses the IMU for initial pose estimation and deskewing the cloud. This leads to eventual drift in roll and pitch as the IMU data is not utilized during pose estimation. Future work will integrate the final pose estimates with a factor graph to enable more flexible integration of additional sensors. This will also allow us to incorporate attitude or preintegrated IMU factors, improving the pitch and roll estimation over long trajectories.

5 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2222723. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Amandeep Dhaliwal. The Rise of Automation and Robotics in Warehouse Management. pages 63–72. 12 2020. ISBN 9781003032410. doi:10.1201/9781003032410-5.

[2] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 10 2020. ISBN 978-1-7281-6212-6. doi:10.1109/IROS45743.2020.9341176. URL https://ieeexplore.ieee.org/document/9341176/.

[3] Ji Zhang and Sanjiv Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems X*, volume 2, pages 1–9. Robotics: Science and Systems Foundation, 7 2014. ISBN 9780992374709. doi:10.15607/RSS.2014.X.007. URL http://www.roboticsproceedings.org/rss10/p07.pdf.

[4] Frank Dellaert and GTSAM Contributors. borglab/gtsam, 5 2022. URL https://github.com/borglab/gtsam).

[5] Matteo Frosi and Matteo Matteucci. ART-SLAM: Accurate Real-Time 6DoF LiDAR SLAM. *IEEE Robotics and Automation Letters*, 7 (2):2692–2699, 4 2022. ISSN 2377-3766. doi:10.1109/LRA.2022.3144795. URL https://ieeexplore.ieee.org/document/9691876/.

[6] Anton Koval, Christoforos Kanellakis, and George Nikolakopoulos. Evaluation of Lidar-based 3D SLAM algorithms in SubT environment. *IFAC-PapersOnLine*, 55(38):126–131, 2022. ISSN 24058963. doi:10.1016/j.ifacol.2023.01.144. URL https://linkinghub.elsevier.com/retrieve/pii/S2405896323001519.

[7] Juhyeon Kim, Jeehoon Kim, Sunwoong Paik, and Hyoungkwan Kim. Point cloud registration considering safety nets during scaffold installation using sensor fusion and deep learning. *Automation in Construction*, 159:105277, 3 2024. ISSN 09265805. doi:10.1016/j.autcon.2024.105277. URL https://linkinghub.elsevier.com/retrieve/pii/S092658052400013X.

[8] Yaoqiang Pan, Kewei Hu, Hao Cao, Hanwen Kang, and Xing Wang. A novel perception and semantic mapping method for robot autonomy in orchards. *Computers and Electronics in Agriculture*, 219:108769, 4 2024. ISSN 01681699. doi:10.1016/j.compag.2024.108769. URL https://linkinghub.elsevier.com/retrieve/pii/S0168169924001601.

[9] Qipeng Li, Yuan Zhuang, and Jianzhu Huai. Multi-sensor fusion for robust localization with moving object segmentation in complex dynamic 3D scenes. *International Journal of Applied Earth Observation and Geoinformation*, 124:103507, 11 2023. ISSN 15698432. doi:10.1016/j.jag.2023.103507. URL https://linkinghub.elsevier.com/retrieve/pii/S156984322300331X.

[10] Lintong Zhang, Michael Helmberger, Lanke Frank Tarimo Fu, David Wisth, Marco Camurri, Davide Scaramuzza, and Maurice Fallon. Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping. *IEEE Robotics and Automation Letters*, 8(1):408–415, 1 2023. ISSN 2377-3766. doi:10.1109/LRA.2022.3226077. URL https://ieeexplore.ieee.org/document/9968057/.

[11] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7. IEEE, 6 2011. ISBN 978-1-4244-9862-8. doi:10.1109/ICORR.2011.5975346. URL http://ieeexplore.ieee.org/document/5975346/.