Precision Increase for LiDAR-based Localisation using a predefined global Map

Martin Hesse[1], Andreas Nüchter[1]

[1] Informatics XVII – Robotics, Julius-Maximilians-Universität Würzburg, Germany
(martin.hesse, andreas.nuechter)@uni-wuerzburg.de

Keywords: Laser scanning, Localisation, Simultaneous Localisation and Mapping, Lidar Inertial Odometry.

Abstract

Localisation remains a crucial aspect of robotic design. It forms the basis of any kind of autonomous navigation for drones, cars and other specialized robots. This is usually achieved using a Simultaneous Localisation and Mapping (SLAM) algorithm, which uses an input sensor to localise the robot within a map that is created simultaneously. The input sensors are either cameras, which provide visual data, or Light Detection And Ranging (LiDAR) sensors, which automatically deliver a point cloud up to surveying quality. In recent years, LiDAR inertial odometry (LIO) algorithms, which combine measurements from a LiDAR sensor and inertial measurements from an IMU, have become more popular. These algorithms do not use a previously recorded map, but rather create their own map during runtime. This paper contributes an improvement to the precision by integrating a predefined 3D global point cloud map into the localisation algorithm. Over the course of multiple experiments in different testing scenarios, we have achieved a 71% reduction of the distance error for localisation, while there was no significant change regarding the orientation error. This makes the presented system a suitable localisation option for real-world robotic operations at construction sites.

1. Introduction

Robotics are used in an ever expanding number of fields and their localisation remains an important issue, that has to be solved to enable any kind of autonomous operations. The construction industry though is one, where the adoption of robotic automation is slow. (Yarovoi and Cho, 2024) Localisation is a particularly difficult problem, because many traditional localisation approaches fail. While in most scenarios Global Navigation Satellite Systems (GNSS) are used, there remain challenges, when this is either not precise enough or it is unavailable, such as in many interior spaces. This makes it infeasible for the construction industry, where robotic system have to work in interior spaces. Other common options, popular in warehouses, are localisation systems, that require a fixed setup, such as motion capture, which also are not an option in the dynamic environment of a construction site. In these situations a Simultaneous Localisation and Mapping (SLAM) algorithm is commonly used as a substitute, which relies only on sensors, that are mounted on the robot itself. These make it possible to have a localisation system, that works independently and does not require any additional set up at the deployment location. Typically, the sensors used in SLAM are either cameras, which produce RGB-images or in the case of stereo cameras potentially depth images, or Light Detection And Ranging (LiDAR) systems, which result in a point cloud encoding spatial information. The change in consecutive images or scans from these sensors is detected from which the movement of the robot is deduced. The individual images or scans are then merged, based on the robots relative position to form a map of the traversed area at runtime. Recently Visual Inertial Odometry (VIO) and LiDAR Inertial Odometry (LIO) algorithms have become more popular. These aim to provide a more accurate and higher frequency pose estimation, by combining lower frequency visual or LiDAR sensors with high frequency inertial measurements from an IMU. An important advantage for any of these algorithms is, that they work without any prior knowledge of the traversed area, but this lack of a previously defined global reference, leads to rising inaccuracies over longer time periods in particular for parts of the trajectory further from the starting point. While in many situations this is acceptable, as they are still coherent locally, we aim to support real-world robotic operations at constructions sites, where greater accuracy is necessary.

This paper introduces an addition to the FAST-LIO 2 (Xu et al., 2022) algorithm, which introduces a previously created point cloud map of the working environment of the robot. This is used to continuously adapt the pose estimation to remain in line with the global map. As construction sites more widely move to integrate Building Information Modelling (BIM) systems into the workflow, maps of the environment, that are used for our addition, often already exist. In cases were this is not the case 3D point clouds are quickly created using a terrestrial laser scanner with the typical accuracies. We aim to achieve localisation accuracies in a similar range, such that real-world robotic operations at construction sites become feasible.

2. Related Work

Over the course of the last decades a lot of research in the direction of SLAM algorithms for localisation purposes has been done. Generally a larger focus was put on LiDAR based SLAM methods, as these tend achieve better results compared to visual SLAM, among others due their ability to immediately provide depth information. (Garigipati et al., 2022, Cadena et al., 2016, Yarovoi and Cho, 2024, Broggi et al., 2013) In recent years most SLAM algorithms, both visual and LiDAR based, use a multistep SLAM core, which first creates some kind of odometry data, based on scan-to-scan matching with the previous scan, before stitching together a map usually at a lower frequency. The LOAM (Zhang and Singh, 2014) and V-LOAM (Zhang and Singh, 2015) algorithms, were among the first to do so, for LiDAR and visual data respectively. Since then many adaptations of LOAM and other similar algorithms have been proposed, usually with a specific use-case in mind or leveraging specific kind of data. These include Loam Livox (Lin and Zhang, 2020), which is specialized for solid-state LiDAR sensors, which have become more popular due to their reduced cost, and LeGO-LOAM (Shan and Englot, 2018) and ART-SLAM (Frosi and Matteucci, 2022), both of which add a ground segmentation module to the SLAM pipeline.

While these systems do not require any additional input, other than from either the camera or LIDAR sensor, some of them do support it, from among others GNSS, an IMU or otherwise obtained odometry. Though recently many systems were introduced, that require inertial measurements from an IMU as an additional higher frequency input to smooth and better predict movement. Some of these include LIO-SAM (Shan et al., 2020), Wildcat(Ramezani et al., 2022), DLIO(Chen et al., 2023) and FAST-LIO 2 (Xu et al., 2022).

A common issue, that we see with many SLAM algorithms is drift. As there is no globally stable reference, that they use, the position estimation tends to drift away from a true representation of the space, which is less important, if you only require locally accurate data, but becomes a larger problem, if the system has to support globally stable operations. There has been some research to try to combat this, by either correcting for drift using post-processing (Keitaanniemi et al., 2023) or adding additional information into the scene like optical markers (Schischmanow et al., 2022). Another option is to introduce predefined map into the SLAM process. A popular data source for this is Open Street Map, like in (Floros et al., 2013, Frosi et al., 2023), which utilize the street grid and the 2D outline of buildings as additional information. (Parsley and Julier, 2010) discuss how prior information, like satellite imagery, that is of a different kind, than the sensor data of the robot, are integrated into the algorithm. While most of these approaches focus on outdoors, (Wang et al., 2019) uses schematic floor plans of buildings, from which features are extracted that are consistent in both the floor plan and the LiDAR data. In our work we use a point cloud as a previously generated map input, to increase the accuracy of the localisation system.

3. Approach

In this work, we introduce a system that uses a predefined global map alongside the FAST-LIO 2 algorithm (Xu et al., 2022) to improve the overall precision of the localisation system of a mobile robot. For this system both LiDAR and inertial data are used in the FAST-LIO 2 algorithm, while the global map is given in the form of a point cloud. The robot's pose is defined by the pose of the mounted LiDAR sensor, and the two terms are used interchangeably throughout. The system has been implemented in ROS 2 (Macenski et al., 2022) and the TF frames depicted in Fig. 1 represent the various pose estimations. There are three different branches, which all represent one estimation of the pose of the robot: first a ground truth, which is not actually calculated by or on the robot, but rather separately and is only used for verification; second the estimation of the FAST-LIO 2 algorithm, whose pose estimation is referred to as without correction; and third our estimation, which incorporates a global map and is referred to as with correction. Both of these are calculated at runtime of the robot on the onboard PC. The global map used by the system is given in the `map` frame, from where the three branches separate. The FAST-LIO 2 algorithm provides the transform between the fixed `camera_init` frame and the `body` frame, which represents the best guess of where the robot is based purely on FAST-LIO 2. To compare this both to the ground truth and our addition, this is linked to the `map` frame based on the initial position of the robot, as this is where the `camera_init` frame is fixed at, and the ground truth provides a transform at this pose to the global `map` frame. This transformation, as it depends on the ground truth, is not available during the runtime of the robot, which means it cannot be directly compared to the corrected pose, as they lack a connected global reference frame. Our addition to the system continuously performs an Iterative Closest Point (ICP) algorithm, in our current implementation the one from open3d (Zhou et al., 2018), matching the most recent point cloud, received from the LiDAR sensor, to the global map. As an initial transform for the ICP algorithm the current pose of the robot is used. This determines the transform from the `map` to the `body_prev` frame, which therefore always holds the pose of the latest matched LiDAR scan. As the ICP algorithm takes some time to calculate the transformation, the resulting pose is no longer accurate at the current timestamp cur, as it represents the pose at the timestamp k, at which the corresponding point cloud was taken by the LiDAR sensor. To overcome this offset and the time during which the calculation takes place, during all which, we also aim to provide an accurate pose estimation, we interpolate from the latest matched LiDAR scan at timestamp k to now with data from FAST-LIO 2. For this we temporarily save the transformation ${}^{\texttt{camera\_init}}_{\texttt{body}_k}T$, the result of FAST-LIO 2 at timestamp k and use it in the following

$$ {}^{\texttt{body\_prev}}_{\texttt{body\_guess}}T = {}^{\texttt{body}_k}_{\texttt{body}_{cur}}T = {}^{\texttt{camera\_init}}_{\texttt{body}_k}T^{-1}\; {}^{\texttt{camera\_init}}_{\texttt{body}_{cur}}T \quad (1) $$

to calculate the transformation from the `body_guess` to the `body_prev` frame. The `body_guess` frame now holds the pose of the robot based on the ICP result, interpolated to the current time using FAST-LIO 2.

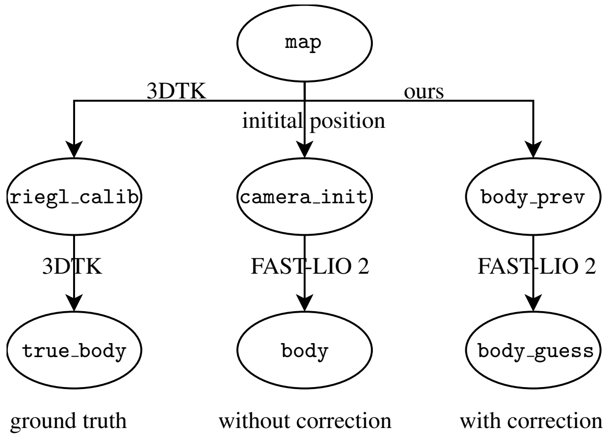


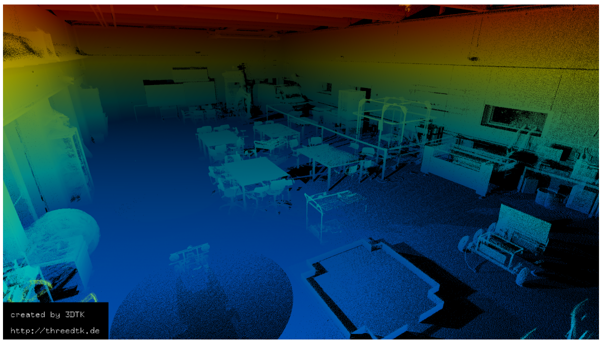
Figure 1. An overview of the used TF frames in ros. The annotation next to an arrow denotes what is responsible for that transformation.

4. Experiments

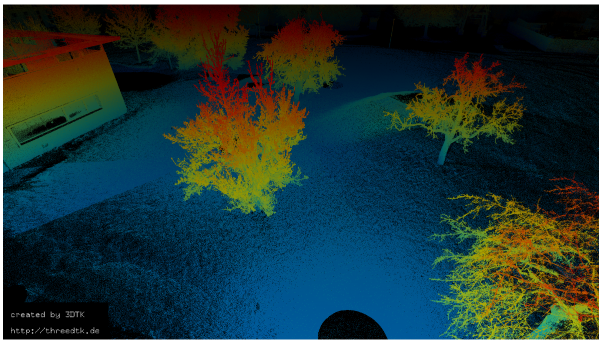
To verify the system accuracy we preform several experiments in multiple scenarios, which are summarized in Tab. 1. The test runs take place either in an inside or outside environment, of which Fig. 2 depicts a point cloud. Both of these are a combination of multiple high definition laser scans. One of the resulting point clouds is used as the global map for the correction step, depending on the location of the test run. As the LiDAR sensor of the system we use a Livox Mid-360, which we attach to a mobile volksbot robot (Nüchter et al., 2013), which Fig. 3 shows. In most test runs the LiDAR sensor is mounted

Test Run #	Length [m]	# GT-Scan Positions n	Location	Sensor Orientation
1	21.09	3	inside	level
2	22.93	3	inside	angled
3	49.85	5	inside	angled
4	36.97	4	inside	angled
5	85.89	5	outside	angled
6	82.20	5	outside	angled
7	68.80	5	inside	level

Table 1. Overview of the different configurations of the individual experiment test runs.



(a) Inside



(b) Outside

Figure 2. A high definition laser scan of the testing environment.

at an angle, as shown in Fig. 3, which aims to improve the performance as suggested in (Yarovoi and Cho, 2024), whereas in some it is mounted level, with the dome of the sensor pointing upwards. The inertial measurements are collected from the IMU integrated into the Mid-360. Each test run consists of an arbitrary path of varying length driven with the robot. While not exact, the end point of each driven trajectory is at a similar position as the start point.

To determine the accuracy of the system we use up to five high definition laser scans from a terrestrial Riegl VZ-400 laser scanner per test run, which are taken at the beginning and end of each test run and at a few points in between, at all of which the robot is also stationary. The positions of the robot at each of the scans (GT-Scan Position 1 through n) are used as the ground truth of the system. These scans are then manually evaluated after each test run for comparison. To facilitate this, the robot is equipped with two calibration spheres of known diameter, the position of which relative to the LiDAR sensor is also known. By selecting the points of the calibration spheres and the plane, on which the LiDAR sensor is mounted, in each of the high definition laser scans using 3DTK (Nüchter and Lingemann, 2011), depicted in Fig. 4, we determine the exact pose of the robot relative to the pose of the Riegl laser scanner at each of

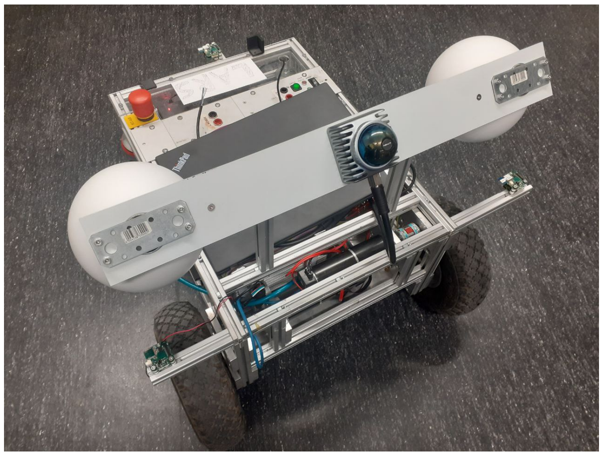


Figure 3. The robotic test platform used to perform the experiment runs, including the LiDAR sensor and the two calibration spheres mounted on it.

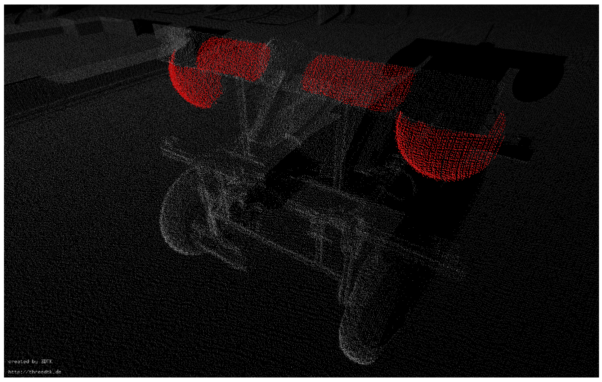


Figure 4. The robotic test platform in a calibration scan, with the calibration spheres and the mounting plate of the LiDAR sensor selected.

the GT-scan locations, representing the transformation from the `true_body` to the `riegl_calib` frame. Using an ICP algorithm from 3DTK we match the calibration scans into the global map and therefore determine the transform from the `riegl_calib` to the `map` frame. The pose of the robot at the first of these GT-scans is also used to define the initial pose of the robot and therefore the `camera_init` frame, as mentioned in Section 3. We now compare both the result of just the FAST-LIO 2 algorithm and our addition to this ground truth, to determine their respective accuracy, at each GT-scan position. As we define the initial position, and therefore also the `camera_init` frame, using the first calibration scan, the position and orientation error of the FAST-LIO 2 algorithm at the first GT-scan location is by definition 0.0 cm or °; these data points are excluded in all further analysis.

The results of our experiments are listed in Tab. 2, in which we tabularize both the position and orientation error of both FAST-LIO 2 and our addition at each GT-scan location. Fig. 5 shows the results graphically.

5. Discussion

Overall we see a significant improvement when using our addition to continuously correct the FAST-LIO 2 estimation, by

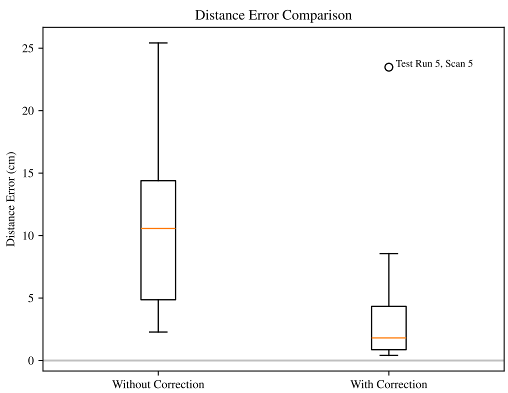
Test Run #	Length [m]	Distance Error at GT-Scan Position # [cm]									
		1		2		3		4		5	
		FL2	ours	FL2	ours	FL2	ours	FL2	ours	FL2	ours
1	21.09	0.00	0.53	15.05	0.41	3.89	1.15	-	-	-	-
2	22.93	0.00	0.95	11.97	0.49	2.28	0.59	-	-	-	-
3	49.85	0.00	0.69	11.63	1.50	7.80	0.72	10.57	1.86	4.09	1.42
4	36.97	0.00	0.92	8.54	0.85	10.03	4.75	5.05	0.81	-	-
5	85.89	0.00	5.15	18.47	2.21	8.95	1.96	23.56	2.55	4.66	23.48
6	82.20	0.00	4.87	13.74	2.89	18.21	3.09	25.42	4.88	4.68	4.98
7	68.80	0.00	1.41	19.13	8.15	11.93	3.02	11.19	8.56	3.33	1.76

(a) Distance error

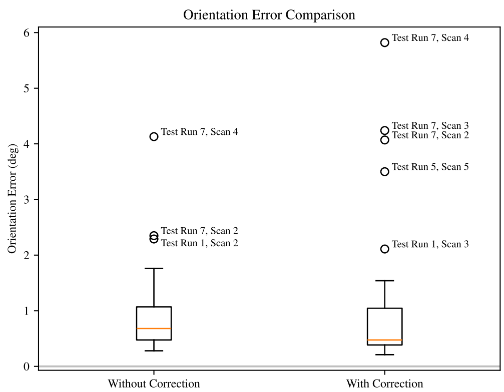
Test Run #	Length [m]	Orientation Error at GT-Scan Position # [deg]									
		1		2		3		4		5	
		FL2	ours	FL2	ours	FL2	ours	FL2	ours	FL2	ours
1	21.09	0.00	1.07	2.29	1.40	0.91	2.11	-	-	-	-
2	22.93	0.00	0.21	0.51	0.41	0.29	0.26	-	-	-	-
3	49.85	0.00	0.25	0.38	0.75	0.28	0.24	0.67	0.97	0.38	0.40
4	36.97	0.00	0.33	1.11	0.45	0.68	0.47	0.56	0.52	-	-
5	85.59	0.00	0.70	0.54	0.38	0.95	0.52	1.03	0.37	0.44	3.50
6	82.20	0.00	0.47	0.83	0.40	0.60	0.37	0.87	0.47	0.37	0.48
7	68.80	0.00	1.54	4.13	4.07	1.76	5.82	2.35	4.24	1.64	0.80

(b) Orientation error

Table 2. The error of FAST-LIO 2 compared to our addition at various ground truth calibration scan positions for the test runs. The error of FAST-LIO 2 at scan position 1 is 0.0 cm or ° by definition, due to our evaluation. A dash (-) indicates, that the given test run did not have four or five calibration scans.



(a) Distance error



(b) Orientation error

Figure 5. The error of FAST-LIO 2 (without correction) compared to our addition (with correction).

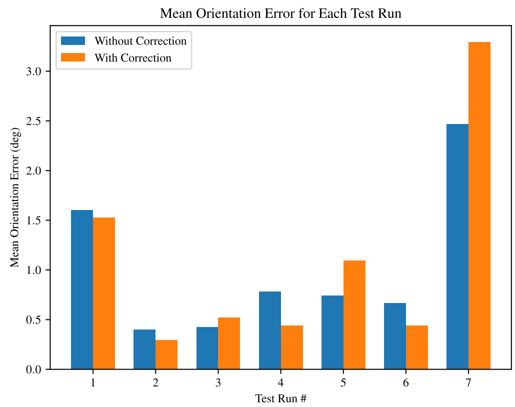


Figure 6. Mean orientation error for each test run.

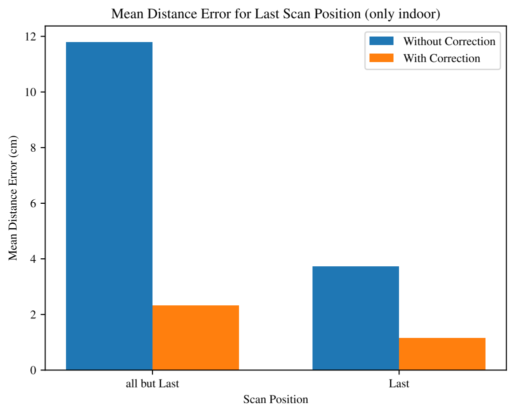


Figure 7. Mean distance error at the last GT-scan position compared to all others.

matching it into a global predefined map. With regard to the orientation we notice no significant difference between either system, with the mean error over all test runs barley worsening from $1.02°$ to $1.13°$, when adding the global map. This decrease in precision is mainly driven by Test Runs 1 and 7 and one particularly bad data point, at GT-scan 5 in Test Run 5, which is the worst performing data point in the distance error evaluation. Why the matching into the global map performed much poorer at Test Runs 1 and 7 is explained in the following, why it is much poorer at the remaining point is unclear, especially as it seems to be a singular outlier in particular with regard to the distance error. We explain the overall similar performance level for the orientation error, mainly by three things: first the overall error is fairly small anyway, so only a marginal improvement could be made; second the robot barley rolled or pitched along the driven trajectory; third we assume orientation is much more stable as translational movement, as with the ground and the ceiling in the inside test runs there is always a large relatively flat plane in the FOV of the LiDAR scanner, which is easy to track. Analysing the orientation error on a run by run basis, depicted in Fig. 6, we notice a larger difference between the accuracy in Test Runs 1 and 7 compared to the others. In these test runs the LiDAR sensor was mounted level with the robot and not at an angle (cf. Tab. 1), which due to the limited vertical FOV of $59°$ of the sensor, results in much smaller and further away parts of the ceiling and ground being detected. This further supports the notion, that the ground is an important reason for stable orientation estimations.

For the distance error there is a significant improvement from a mean error of 11.05 cm when only using the FAST-LIO 2 algorithm down to 3.22 cm, with the correction. When comparing the error at the last GT-scan position compared to at all others (Fig. 7) we notice, that especially the FAST-LIO 2 algorithm performs significantly better at the last position. As mentioned in Section 3 the last GT-scan is always made with the robot at a position similar to the initial one. This shows the fact that, without the correction the system tends to drift the further the robot is from the initial position, highlighting the need for a globally stable addition.

6. Conclusion

Overall this work introduces an addition to the existing FAST-LIO 2 LiDAR localisation algorithm, that enhances the result significantly, by using a predefined global map, in which the laser scans are continuously located. This is noticed especially regarding the distance error, whose mean is decreased from 11.05 cm when only using the FAST-LIO 2 algorithm down to 3.22 cm. Thus, we add a globally stable localisation to the FAST-LIO 2 algorithm. The achieved accuracy for indoor localisation is acceptable to enable the use of this system for usage on construction sites. Needless to say a lot of work remains to be done. Better understanding of the accuracy of the system in different real-world scenarios with varying complications has to be achieved. These complications may include more obscured viewing angles, dust or other moving objects, which makes localisation more difficult, but are unavoidable in a real-world construction site setting. Also, the long term stability of the system has to be better understood, to support advanced autonomous operations. Finally, there remain individual data points in the experiments, at which the system performed notably worse, which are not yet fully explained.

References

Broggi, A., Grisleri, P., Zani, P., 2013. Sensors technologies for intelligent vehicles perception systems: A comparison between vision and 3d-lidar. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 887–892.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J. J., 2016. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309-1332.

Chen, K., Nemiroff, R., Lopez, B. T., 2023. Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3983–3989.

Floros, G., van der Zander, B., Leibe, B., 2013. Openstreetslam: Global vehicle localization using openstreetmaps. *2013 IEEE International Conference on Robotics and Automation*, 1054–1059.

Frosi, M., Gobbi, V., Matteucci, M., 2023. OSM-SLAM: Aiding SLAM with OpenStreetMaps priors. *Frontiers in Robotics and AI*, Volume 10 - 2023.

Frosi, M., Matteucci, M., 2022. ART-SLAM: Accurate Real-Time 6DoF LiDAR SLAM. *IEEE Robotics and Automation Letters*, 7(2), 2692-2699.

Garigipati, B., Strokina, N., Ghabcheloo, R., 2022. Evaluation and comparison of eight popular lidar and visual slam algorithms. *2022 25th International Conference on Information Fusion (FUSION)*, IEEE, 1–8.

Keitaanniemi, A., Rönnholm, P., Kukko, A., Vaaja, M. T., 2023. Drift analysis and sectional post-processing of indoor simultaneous localization and mapping (SLAM)-based laser scanning data. *Automation in Construction*, 147, 104700.

Lin, J., Zhang, F., 2020. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3126–3131.

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W., 2022. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), eabm6074.

Nüchter, A., Elseberg, J., Borrmann, D., 2013. Irma3D – An Intelligent Robot for Mapping Applications. *Proceedings of the 3rd IFAC Symposium on Telematics Applications (TA ’13)*, 119–124.

Nüchter, A., Lingemann, K., 2011. 3DTK – The 3D Toolkit. https://github.com/JMUWRobotics/3DTK. Accessed: 12.11.2025.

Parsley, M. P., Julier, S. J., 2010. Towards the exploitation of prior information in SLAM. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2991-2996.

Ramezani, M., Khosoussi, K., Catt, G., Moghadam, P., Williams, J., Borges, P., Pauling, F., Kottege, N., 2022. Wildcat: Online continuous-time 3d lidar-inertial slam.

Schischmanow, A., Dahlke, D., Baumbach, D., Ernst, I., Linkiewicz, M., 2022. Seamless Navigation, 3D Reconstruction, Thermographic and Semantic Mapping for Building Inspection. *Sensors*, 22(13).

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

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Daniela, R., 2020. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 5135–5142.

Wang, X., Marcotte, R. J., Olson, E., 2019. Glfp: Global localization from a floor plan. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1627–1632.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics*, 38(4), 2053-2073.

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

Zhang, J., Singh, S., 2014. LOAM : Lidar Odometry and Mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, 109-111.

Zhang, J., Singh, S., 2015. Visual-lidar odometry and mapping: low-drift, robust, and fast. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2174–2181.

Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIX-B1-2026-161-2026