

Quantitative 3D Map Accuracy Evaluation Hardware and Algorithm for LiDAR(-Inertial) SLAM

Sanghyun Hahn^{1†}, Seunghun Oh^{2†}, Minwoo Jung², Ayoung Kim², and Sangwoo Jung^{2*}

¹Department of Mechanical and Aerospace Engineering, Seoul National University,
Seoul, 08826, Korea (steve0221@snu.ac.kr)

²Department of Mechanical Engineering, Seoul National University,
Seoul, 08826, Korea (alvin0808, moonshot, ayoungk, dan0130@snu.ac.kr) * Corresponding author

Abstract: Accuracy evaluation of a 3D pointcloud map is crucial for the development of autonomous driving systems. In this work, we propose a user-independent software/hardware system that can quantitatively evaluate the accuracy of a 3D pointcloud map acquired from LiDAR(-Inertial) SLAM. We introduce a LiDAR target that functions robustly in the outdoor environment, while remaining observable by LiDAR. We also propose a software algorithm that automatically extracts representative points and calculates the accuracy of the 3D pointcloud map by leveraging GPS position data. This methodology overcomes the limitations of the manual selection method, that its result varies between users. Furthermore, two different error metrics, relative and absolute errors, are introduced to analyze the accuracy from different perspectives. Our implementations are available at: https://github.com/SangwooJung98/3D_Map_Evaluation

Keywords: SLAM, LiDAR SLAM, 3D Map Accuracy, Relative Error, Absolute Error

1. INTRODUCTION AND RELATED WORKS

Light Detection and Ranging (LiDAR) is one of the representative sensors that provide 3D points containing accurate information around it. Due to its robustness of light conditions and convenient pointcloud data generation, LiDAR is widely exploited in simultaneous localization and mapping (SLAM) research [1–5]. With the development of LiDAR SLAM, the importance of an accurate 3D pointcloud map is rising as it can be exploited in various fields such as autonomous car driving [6, 7], high-resolution map [8], and long-term map management [9].

Following the increase of 3D pointcloud map utilization, evaluating the accuracy of each map has also become important. The accuracy of a 3D pointcloud map generated by LiDAR SLAM is generally approached indirectly by the accuracy of odometry, which is another output of LiDAR SLAM. [10] Traditionally, the direct calculation of the accuracy of a 3D pointcloud map is performed by placing a small object at a specific location and selecting a single point that represents the object from the 3D map by human hand. However, due to the characteristic of LiDAR that the pointcloud density decreases proportional to the distance from the sensor, selection of the representative point may vary depending on the user.

To overcome the problems mentioned above, we propose a target base 3D pointcloud map accuracy measurement algorithm. Research on targets for LiDAR [11–13] has been done while their design purpose was mostly the extrinsic calibration between LiDAR and cameras. Unlike existing LiDAR targets, the proposed target is robust in outdoor environments while remaining detectable by LiDAR. Furthermore, by exploiting K-means clustering [14], random sample consensus (RANSAC) [15], and sin-

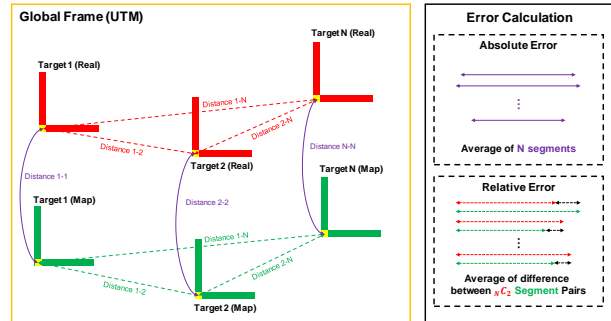


Fig. 1. An overview of the absolute and relative error metrics. Absolute error is derived from the distance between N pairs of corresponding target pose and ground truth pose. Relative error is derived from the distance between every two target poses.

gular value decomposition to the target pointcloud, the target position can be extracted consistently without depending on the user. With the calculated target pose, we propose two different error metrics (relative and absolute errors) for measuring the accuracy of 3D pointcloud maps. An overview of the two error metrics is illustrated in Fig. 1, while the major contributions of the work are as follows:

- We introduce the hardware design of the LiDAR target, which is robust in the outdoor environment, and a software algorithm that calculates the position of each LiDAR target included in the 3D pointcloud map.
- We suggest two different error metrics (relative and absolute error) that can be exploited for analyzing the accuracy of the 3D pointcloud map from different perspectives.
- We release our evaluation algorithm and hardware design to the public.

[†]The authors contributed equally to this paper.

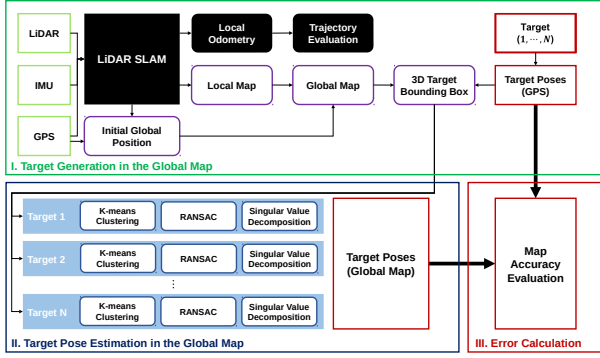


Fig. 2. Pipeline of the algorithm. The target generation (green) process can be performed with any LiDAR based SLAM algorithms. Using the target pointclouds and GPS poses, target poses on the map are estimated (blue). Map accuracy is evaluated using the GPS target pose and estimated target pose (red).

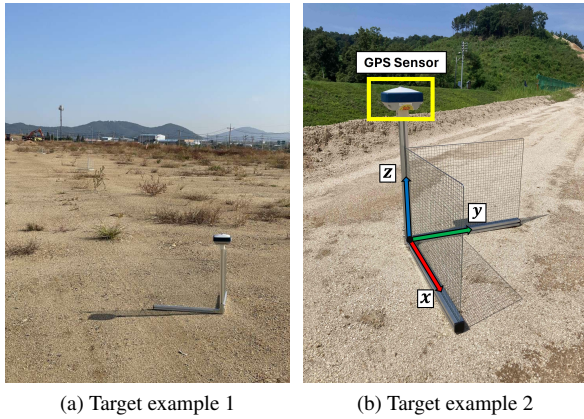


Fig. 3. (a) and (b) shows the target arrangement example. Due to the holes in the target, it is robust to wind in outdoor environments while remaining detectable by LiDAR.

2. HARDWARE DESIGN AND TARGET POSE ESTIMATION

Due to the random wind, dust, and other unpredictable factors of the outdoor environment, the target should be **1) robust to wind**, **2) GPS attachable**, and **3) easy to build**. Using the 30×30 aluminum extrusion and cross-shaped meat grills, we built an outdoor robust LiDAR target, as shown in Fig. 3. The length of each edge is 0.6m, and the ground truth target pose can be acquired using the GPS sensor (Sokkia GRX3) attached to the top of it.

Fig. 2 illustrates the overall pipeline of the full algorithm. As an output of LiDAR SLAM framework, a global 3D pointcloud map is acquired. By cropping a sphere with a 5m radius, centered on the GPS target pose obtained from the GPS sensor attached to the target, a pointcloud that includes the target can be cropped from the full 3D map. This process is defined as **loose cropping**, as the resulting pointcloud includes both the target and surrounding ground points. Alternatively, the loose

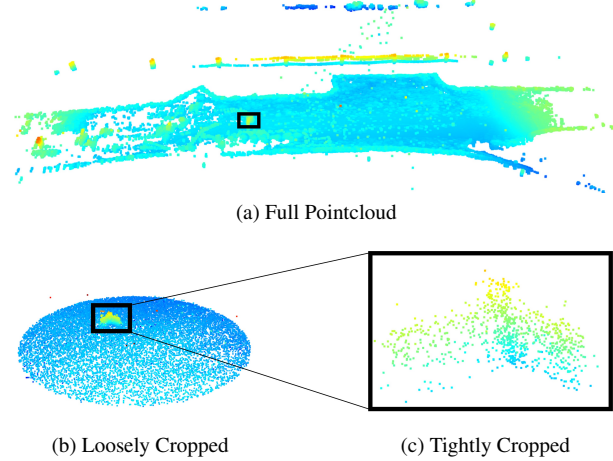


Fig. 4. Visualization of the cropping process. (b) is cropped automatically from (a) based on the GPS target pose. Ground points and outliers are manually removed from (b) to obtain (c). All black boxes indicate the same area that contains a single target.

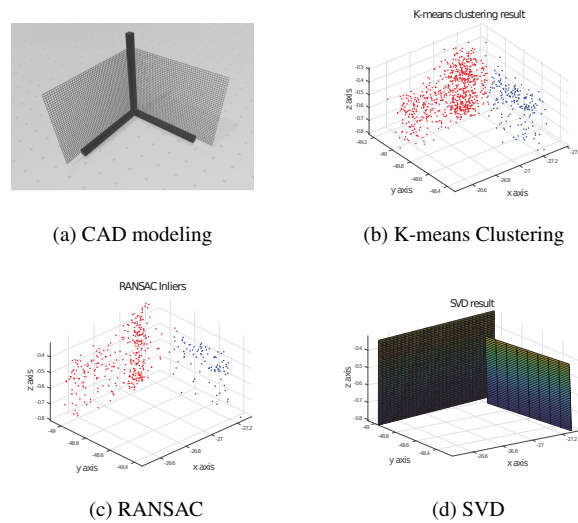


Fig. 5. Example of target plane estimation in step by step. The red and blue points in (b) and (c) indicate the identified target planes respectively.

cropping process can be achieved by cropping the target from the full 3D map manually without GPS. Following loose cropping, the ground points of the loosely cropped pointcloud can be removed manually. This process is defined as **tight cropping**, as the resulting pointcloud mostly consists of the target itself with only a small number of outliers. Although the tight cropping procedure is done manually, ground removal from the loosely cropped pointcloud is much easier than selecting a single point that can represent the target from the full 3D map. As a result, the tight cropping result is more consistent among different users compared to the previous methods. The cropping process is displayed in Fig. 4.

After extracting the tightly cropped pointcloud from the 3D pointcloud map, the calculation of its pose is

Table 1. Relative error experiment results. The relative error(E_{rel}) and standard deviation(σ_{rel}) of each sequence is written in **bold**. t1 & t2 indicates the error in the distance between target 1 and target 2.

	E_{rel} (m)	σ_{rel} (m)	t1 & t2	t1 & t3	t1 & t4	t1 & t5	t2 & t3	t2 & t4	t2 & t5	t3 & t4	t3 & t5	t4 & t5
Sequence 1	0.0754	0.0413	0.0197	0.0826	0.0701	0.1271	0.1021	0.0902	0.1470	0.0140	0.0440	0.0569
Sequence 2	0.0647	0.0352	0.0133	0.0697	0.0546	0.1136	0.0836	0.0680	0.1269	0.0141	0.0445	0.0589
Sequence 3	0.0499	0.0315	0.0257	0.0748	0.0903	0.0932	0.0488	0.0650	0.0680	0.0132	0.0167	0.0030
Sequence 4	0.0532	0.0290	0.0056	0.0455	0.0708	0.0912	0.0521	0.0765	0.0971	0.0265	0.0461	0.0206
Sequence 5	0.0633	0.0431	0.0095	0.0796	0.0988	0.0990	0.0900	0.1089	0.1096	0.0187	0.0179	0.0008

Table 2. Absolute error experiment results. The absolute error(E_{abs}) and standard deviation(σ_{abs}) of each sequence is written in **bold**.

	E_{abs} (m)	σ_{abs} (m)	target1	target2	target3	target4	target5
Sequence 1	0.0513	0.0204	0.0523	0.0720	0.0337	0.0236	0.0752
Sequence 2	0.0420	0.0211	0.0450	0.0584	0.0282	0.0099	0.0688
Sequence 3	0.0398	0.0098	0.0589	0.0316	0.0361	0.0336	0.0385
Sequence 4	0.0370	0.0130	0.0403	0.0467	0.0140	0.0327	0.0511
Sequence 5	0.0516	0.0110	0.0571	0.0643	0.0313	0.0513	0.0538

achieved through the following three steps. First, **K-means clustering** with $K = 2$ is applied to the tightly cropped pointcloud, dividing it into two groups of points, P_1, P_2 . Each cluster represents one plate of the target. Next, **RANSAC** is applied to the pointcloud of each plate, P_1, P_2 , in order to remove the outliers. The fitting function of RANSAC is a plane function based on random 3 points, and the inlier threshold is 0.03m. After RANSAC, inlier pointclouds that represent each plate, P_1^R, P_2^R remain. Finally, **SVD** is applied to each inlier pointcloud, P_1^R, P_2^R , calculating the optimal plane function, P_1^{op}, P_2^{op} . An example of data processing at each step is displayed in Fig. 5.

Based on the strategy, P_1^{op}, P_2^{op} should be perpendicular while each plane itself is perpendicular to the ground. Only plane functions that satisfy the perpendicular conditions within a threshold of 1° proceed to the target pose calculation, while those that do not meet the threshold are redirected back to the K-means clustering process.

From the functions of P_1^{op}, P_2^{op} , the intersection line l is calculated as a cross of the normal vectors of P_1^{op} and P_2^{op} . If the target is perpendicular to the ground, the estimated target pose is ideally defined by the intersection point between l and the ground plane. However, inevitable errors exist in the angle between intersection line l and the ground plane. Furthermore, the target is not exactly located at $z = 0$; the target has its own height along the z axis. Therefore, the point on l corresponding to the the average height of P_1^{op} and P_2^{op} is utilized as the estimated target pose. To enhance the robustness of our algorithm, we calculated 100 sample poses for each target, averaging these to obtain the final estimated target pose.



Fig. 6. Satellite view example of the experimental environment and target positions.

3. RELATIVE AND ABSOLUTE ERROR METRIC

Based on the estimated target pose and GPS-based ground truth pose, we propose two different error metrics that represent the accuracy of a 3D pointcloud map in various aspects. Before calculating the accuracy of the map, the estimated target pose and ground truth pose has to be aligned since the frame of the acquired 3D map may not match the frame of the ground truth pose. We exploited a 2D image registration method based on the argmin function to minimize the sum of distances between the corresponding estimated target pose and GPS target pose. This is achieved by translating and rotating the map frame optimally. The 2D transformation matrix R and 2D translation vector t is obtained by the following Eq. (1):

$$\operatorname{argmin}_{R,t} \sum_{i=1}^N \|(Rx_i + t) - \hat{x}_i\|_2 \quad (1)$$

where N is the number of targets, x and \hat{x} denote the estimated target pose and GPS target pose respectively.

After registering the estimated target pose to GPS target pose, map accuracy is calculated based on two differ-

ent metrics: **Relative error** and **Absolute error**.

Relative error is calculated by averaging the difference of distance between the estimated target positions with the GPS-based distance between the same combination of targets. For the relative error calculation of n targets, distance error is calculated for all $\binom{n}{2} = \frac{n(n-1)}{2}$ combinations of targets and the average is defined as the relative error. In this study, 5 targets were exploited and the relative error was calculated by averaging 10 different combinations. Since relative error is calculated based on the distance error between the targets, it focuses on evaluating the general accuracy of the map.

Absolute error is the average of the distances between the estimated target position and its corresponding GPS-based target position. As absolute error is calculated based on the target position error itself, the absolute error metric focuses more on the map accuracy of the local area where the targets are positioned. Furthermore, high standard deviation of the absolute errors indicate that the accuracy of the 3D map varies across different locations of the map.

4. EXPERIMENTS AND RESULTS

We tested the proposed algorithm based on a real-world dataset acquired from a highway construction site. The 3D pointcloud map was generated by the LiDAR SLAM algorithm LIO-SAM[1], while Velodyne VLP-32C LiDAR, MicroStrain 3DM-GX5-25 9 DoF IMU, and NovAtel CPT7 GPS were exploited as the hardware. The experiment environment and hardware position are displayed in Fig. 6.

With consistent target and environment settings, we acquired 5 sequences and calculated the relative and absolute errors for each sequence. The experiment results are displayed in Table. 1 and Table. 2.

The experiment results show that relative error and absolute error have similar tendencies following the sequences. Sequences with smaller relative error show small absolute error, and vice versa. This indicates that the general map error of the 3D map is highly related to the local map error, suggesting that one error metric includes information of the other error metric indirectly.

Furthermore, as presented in Table. 1, there is a tendency for the relative error to increase as the distance between targets increases. Due to this effect, relative error may show higher values for larger maps. On the other hand, absolute error compares the position of a single target based on GPS target pose and estimated target pose, providing map-size-independent accuracy information.

For absolute error, the results displayed in Table. 2 indicate that the error from targets 2, 3, and 4 tends to be lower than those from targets 1 and 5. This is due to the robot's trajectory which makes an ellipsoid around the targets, visiting targets 1 and 5 only once while other targets are visited twice.

It is challenging to keep the factors such as the distance from the robot to each target and the number of visits

per target constant across different sequences. In order to remove these effects and measure the accuracy of the 3D pointcloud map robustly, the relative error metric may be exploited.

5. CONCLUSION

In this work, we propose hardware and a software algorithm that can be exploited to measure the accuracy of a 3D pointcloud map while maintaining the robustness in outdoor environments and LiDAR sparsity. Furthermore, we introduce two different error metrics, Relative error and Absolute error, that represent general map accuracy and local map accuracy. Through the real-world experiment, we demonstrated the relationship between two error metrics and the robustness of our hardware and algorithm. This approach offers insights into the local and global quality of the 3D pointcloud map, enhancing evaluation standards for LiDAR(-Inertial) SLAM algorithms. To overcome the current limitations of this research, we plan to develop a fully automated software that operates without any manual intervention.

REFERENCES

- [1] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2020, pp. 5135–5142.
- [2] A. Reinke, M. Palieri, B. Morrell, Y. Chang, K. Ebadi, L. Carlone, and A.-A. Agha-Mohammadi, "Locus 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping," *IEEE Robot. and Automat. Lett.*, vol. 7, no. 4, pp. 9043–9050, 2022.
- [3] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fastlio2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot. and Automat.*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [4] M. Jung, S. Jung, and A. Kim, "Asynchronous multiple lidar-inertial odometry using point-wise inter-lidar uncertainty propagation," *IEEE Robot. and Automat. Lett.*, 2023.
- [5] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Trans. Robot. and Automat.*, vol. 38, no. 1, pp. 351–371, 2021.
- [6] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2014, pp. 176–183.
- [7] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.
- [8] P. Mandikal and V. B. Radhakrishnan, "Dense 3d

point cloud reconstruction using a deep pyramid network,” in *Proc. IEEE Winter Conf. on Appl. of field Vision*, 2019, pp. 1052–1060.

- [9] M. Labbé and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *J. of Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019.
- [10] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2012, pp. 3354–3361.
- [11] J.-K. Huang, S. Wang, M. Ghaffari, and J. W. Grizzle, “Lidartag: A real-time fiducial tag system for point clouds,” *IEEE Robot. and Automat. Lett.*, vol. 6, no. 3, pp. 4875–4882, 2021.
- [12] Y. Xie, L. Deng, T. Sun, Y. Fu, J. Li, X. Cui, H. Yin, S. Deng, J. Xiao, and B. Chen, “A4lidartag: Depth-based fiducial marker for extrinsic calibration of solid-state lidar and camera,” *IEEE Robot. and Automat. Lett.*, vol. 7, no. 3, pp. 6487–6494, 2022.
- [13] J. Zhang, Y. Liu, M. Wen, Y. Yue, H. Zhang, and D. Wang, “L 2 v 2 t 2 calib: Automatic and unified extrinsic calibration toolbox for different 3d lidar, visual camera and thermal camera,” in *Proc. IEEE Intell. Vehicle Symposium*, 2023, pp. 1–7.
- [14] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, 1967, pp. 281–297.
- [15] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, p. 381–395, 1981.