2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# An Extrinsic Calibration Method between LiDAR and GNSS/INS for Autonomous Driving

Jiahao Pi, Guohang Yan†, Chengjie Wang, Xinyu Cai and Botian Shi

***Abstract*— Accurate and reliable sensor calibration is critical for fusing LiDAR and inertial measurements in autonomous driving. This paper proposes a novel three-stage extrinsic calibration method between LiDAR and GNSS/INS for autonomous driving. The first stage can quickly calibrate the extrinsic parameters between the sensors through point cloud surface features so that the extrinsic can be narrowed from a large initial error to a small error range in little time. The second stage can further calibrate the extrinsic parameters based on LiDAR-mapping space occupancy while removing motion distortion. In the final stage, the z-axis (the vertical direction relative to the ground plane) errors caused by the plane motion of the autonomous vehicle are corrected, and an accurate extrinsic parameter is finally obtained. Specifically, This method utilizes the planar features in the environment, making it possible to quickly carry out calibration. Experimental results on real-world datasets demonstrate the reliability and accuracy of our method. The codes are open-sourced on the Github website. The code link is https://github.com/OpenCalib/LiDAR2INS.**

## I. INTRODUCTION

Autonomous driving technology has attracted more and more attention with the continuous development of science and technology [1]. Precise and dependable location data is increasingly essential for enabling the sophisticated functionalities of self-driving cars. Therefore, most self-driving cars are equipped with the GNSS/INS (Global Navigation Satellite System / Inertial Navigation System) devices, such as NovAtel, Trimble, and other high-precision navigation devices. LiDAR (Light Detection and Ranging) is another crucial sensor in autonomous driving technology [2]. SLAM (Simultaneous Localization and Mapping) and object detection are the two most important applications of LiDAR in the field of autonomous driving [3]. The main role of SLAM is mapping and localization. In autonomous driving, the motion error of the laser frame caused by motion cannot be ignored. Usually, the motion distortion of LiDAR is removed with the help of GNSS/INS [4]. High-precision map construction and localization highly depend on LiDAR and GNSS/INS fusion. Accurate 3D-LiDAR and GNSS/INS extrinsic calibration are essential to determine their coordinate relationship and perform sensor fusion.

Lidar-align [5] is an open-source method for finding the extrinsic calibration between a 3D LiDAR and a 6-dof pose sensor, which also pointed out that accurate calibration results require highly non-planar motions. Due to the lack of motion excitation in the vertical direction in planar motion makes [5] and other similar technologies unsuitable for calibrating sensors mounted on cars. Another commonly used method such as [6], [7] calculate the relative pose of LiDAR and INS using hand-eye calibration, and the accuracy of calibration is limited by the result of odometry [8]–[11]. Due to the rapid development of LiDAR-IMU systems in recent years , many methods have also emerged to calibrate LiDAR-IMU [12]–[17]. These methods generally calibrate the intrinsic parameters of the IMU and the extrinsic parameters of the LiDAR-IMU for small robot platforms. Due to different calibration environments and application scenarios, these calibration methods do not work well when directly calibrating LiDAR to GNSS/INS in autonomous vehicles. Therefore, we propose a calibration method from LiDAR to GNSS/INS specifically for autonomous driving applications in this paper. Inspired by [18], we propose a rough calibration method based on the constraints imposed on the corresponding surface feature points between multiple frames. Subsequently, an optimization equation based on LiDAR-mapping space occupancy is used to further refine the rough calibration results. At the same time, the errors that cannot be evaluated in the z-axis caused by the plane motion of the vehicle are corrected through multiple fiducial points. This three-stage method gradually obtains an accurate ex-

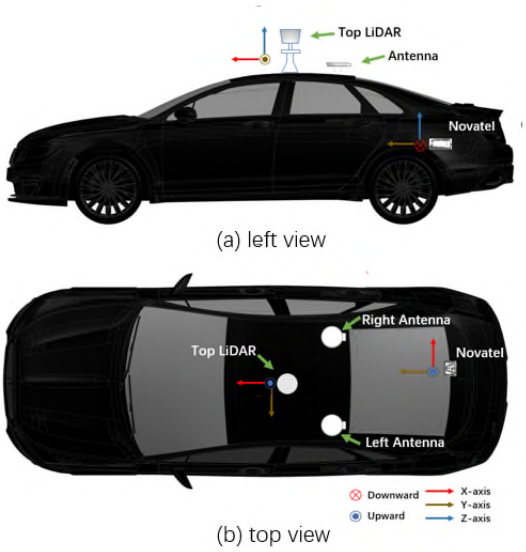


Fig. 1. (a) Left view of our experimental platform. (b) Top view of our experimental platform.

† Corresponding author.
Jiahao Pi, Guohang Yan, Chengjie Wang, Xinyu Cai and Botian Shi are with Autonomous Driving Group, Shanghai AI Laboratory, China. {pijiahao, yanguohang, wangchengjie, caixinyu, shibotian}@pjlab.org.cn

trinsic parameter that overcomes the lack of height-direction excitation in planar motion.

The contributions of this work are listed as follows:

1) We introduce an algorithm for solving LiDAR and GNSS/INS extrinsic calibration initialization by directly minimizing the distance from the feature point to the plane.
2) An octree-based space occupancy refinement method is defined to further refine the extrinsic parameters for improving LiDAR's mapping quality through the pose-sensor. The calibration accuracy of the z-axis is improved through fiducial points matching.
3) Evaluating on real-world datasets, we quantitatively and qualitatively demonstrate our method's robustness and accuracy; meanwhile, the related codes have been open-sourced on GitHub.

## II. RELATED WORK

Researchers have proposed many methods to solve the multi-sensor calibration problem. Sensor calibration can be divided into two parts: intrinsic parameter calibration and extrinsic parameter calibration, and intrinsic parameters determine the internal mapping relationship of the sensor. For example, the IMU intrinsic parameters include the zero bias of gyroscope and accelerometer, scale factor, and installation error, which can be calibrated by method [19]–[23]. The extrinsic parameters determine the transformation relationship between the sensor and the external coordinate system, including 6 degrees of freedom parameters of rotation and translation. Lidar-align [5] proposed an extrinsic calibration method that minimizes the distance between each point of the stitching point cloud by GNSS/INS and its nearest neighbour. However, this method is time consuming and cannot solve the z-axis error caused by plane motion. Currently, most LiDAR to GNSS/INS calibration methods still refer to the calibration method of LiDAR-IMU. LiDAR-IMU extrinsic is usually calibrated together with IMU intrinsic calibration, e.g. methods [12]–[17].

LV et al. [14] proposed the first open-source LIDAR-IMU calibration toolbox based on continuous-time batch estimation. Subsequently, they extended this method and proposed OA-LICalib [17], which seeks to automatically select the most informative data segment for calibration, removing some data without sufficient motion or scene constraints, thus improving calibration accuracy and reducing the computational cost. Gentil et al. [16] used gauss equation regression to eliminate IMU motion distortion, and the optimization method based on the factor graph is used to calibrate LIDAR and IMU. Jiao et al. [24] obtained the initial value of calibration through hand-eye calibration [25]. Then, the appearance-based method is used to optimize the obtained parameters by minimizing the residual function composed of feature points to the plane. Mishra et al. [15] proposed an optimization scheme based on EKF to calibrate LIDAR and IMU. [26], [27] proposed a spatio-temporal calibration method using the continuous-time batch estimation framework for the camera-IMU calibration. Forster et al. [28] solved a non-linear batch estimation problem to determine the unknown extrinsic calibration parameter. However, the above calibration methods need good initial values. Park et al. [29] applied the calibration from coarse to fine, first estimating the closed-form solution and then batch optimizing the continuous-time trajectory to obtain more accurate results.

Due to the lack of motion excitation in the vertical direction during planar motion, existing calibration methods often struggle to correct for the z-axis errors of extrinsic parameters. To this end, we propose a calibration method from LiDAR to GNSS/INS specifically for autonomous driving applications. Our work represents a coarse to fine calibration for LIDAR and GNSS/INS. After getting a satisfactory result, our method also corrects the deviation of the z-axis through fiducial points matching. And our whole calibration procedure takes very little time.

## III. METHODOLOGY

This section introduces the details of our method, including rough calibration, calibration refinement, and z-axis correction. Fig. 2 shows the overview of the proposed method.

### A. Problem Formulation

As shown in Fig. 3, the vehicle collects LiDAR and GNSS/INS sequence data at the intersection by walking three figure-8-shape trajectories and keeps the vehicle speed between 10 km/h and 20 km/h. Then, the GNSS/INS pose data corresponding to the LiDAR timestamp is obtained through the data processing module (hardware triggers). With accurate extrinsic parameters, we can reconstruct the surrounding environment based on LiDAR and GNSS/INS data. Therefore, Our goal is to find a rigid transformation $\boldsymbol{T} = \{\boldsymbol{R}, \boldsymbol{t}\}$ from LiDAR to GNSS/INS so that the 3D reconstruction results obtained by splicing the multi-frame point clouds through the pose provided by GNSS/INS are more accurate. $\boldsymbol{R}$ is a 3D rotation, $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t}$ is a 3D translation, $\boldsymbol{t} \in R^3$. First, we process the LiDAR point cloud pose $T^L$ in the LiDAR coordinate system and convert it to obtain the pose $T^I$ of the point cloud in the GNSS/INS coordinate system by the extrinsic parameters $\boldsymbol{T_I^L}$. Then we get the pose of point cloud in the world coordinate system through GNSS/INS output pose $T_W^I$

$$T_i^W = (T_W^I)_i(\boldsymbol{T_I^L})^{-1}T_i^L = \{R_i^W, t_i^W\} \tag{1}$$

Then, recover the 3D reconstruction map $\boldsymbol{M}$ through pose $T^W$ and LiDAR point cloud sequence $\boldsymbol{P^L}$.

$$\boldsymbol{M} = \sum_{i=1}^{N}(R_i^W \boldsymbol{P_i^L} + t_i^W) \tag{2}$$

Where $\boldsymbol{M}$ is the global 3D point-cloud map (stitching through $N$-frame LiDAR point cloud) in the world coordinate system, we aim to find an extrinsic parameter $\boldsymbol{T_I^L}$ to make the reconstructed map $\boldsymbol{M}$ have the best quality.

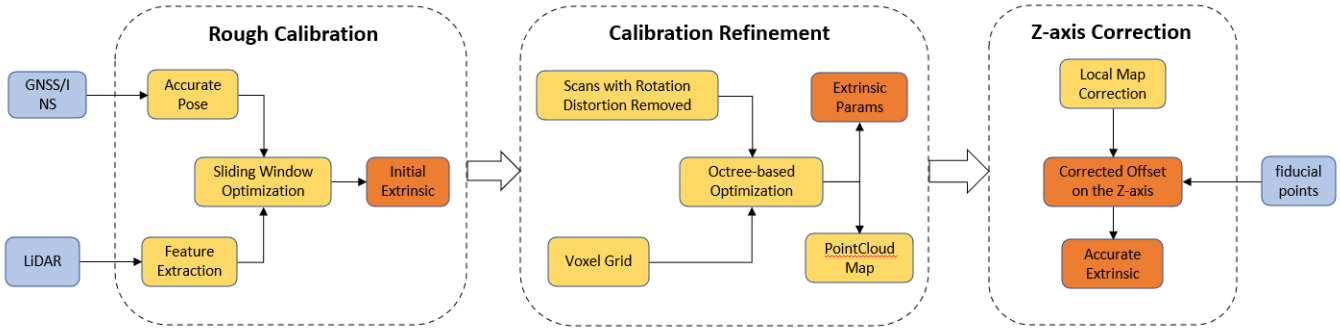


Fig. 2. The pipeline of the proposed LIDAR and pose-sensor extrinsic calibration method includes three parts: rough calibration, calibration refinement and Z-axis correction.

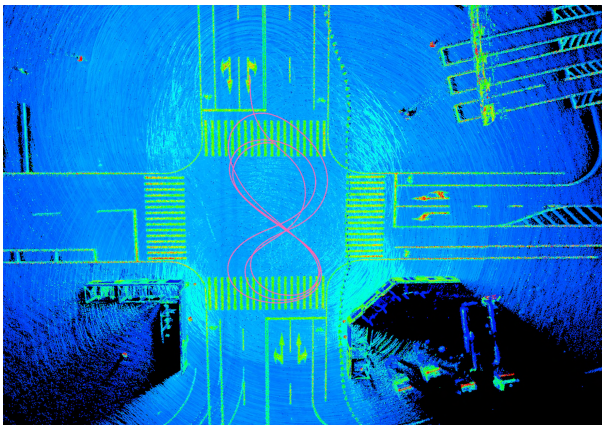

Fig. 3. The calibration data collection method is that the self-driving car circles three "8" characters around the intersection, and the trajectory in the figure is the driving trajectory when the vehicle collects data.

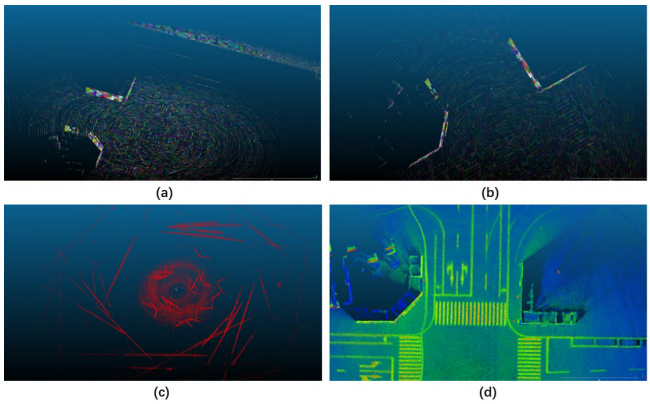


Fig. 4. (a) and (b) are the feature maps of the extracted point cloud surface. (c) The feature map point cloud is extracted in the initial state. (d) Point cloud stitching results after rough calibration.

### *B. Rough Calibration*

The first step is rough calibration. Rough calibration aims to quickly reduce the extrinsic parameter from an initial value with a large error to a small error range. In our experiments, our rough calibration can reduce the extrinsic error in angle and translation from over 20° and 0.5m compared to the ground truth to within 0.2° and 0.03m in less than 20s. In order to reduce the running time, we perform feature extraction on the point cloud. Similar to method [18], we extract the plane features in the point cloud features through adaptive voxelization. The next step is to project the point cloud into the same coordinate system combined with the pose information of GNSS/INS and then optimize it in a sliding window. The specific method of optimization is to assume that there is a sliding window of $n$ frames, in which the pose of GNSS/INS is denoted as $T_1^I, .., T_n^I$. In the sliding window, we first extract the point cloud's feature through each point cloud's curvature. The points with small curvature are considered to be in the plane, and then the centroid $\mathbf{X}$ and normal vector $\mathbf{N}$ of the current plane are recorded through the plane fitting. Then, all the point clouds in different frames are projected into the world coordinate system which refers to the GNSS/INS coordinate system of the first frame. The formula for projecting the point cloud in the LiDAR coordinate system of frame $n$ to GNSS/INS coordinate system of frame 1 is as follow:

$$x_1^I = (T_1^I)^{-1}(T_n^I)\boldsymbol{T_L^I}x_n^L \tag{3}$$

where $x_n^L$ represents the n-th frame point cloud in the LiDAR coordinate system. After being converted to the world coordinate system, the next step is the data association. Then, the nearest neighbors of all plane points in frame 1 and frame $n$ are jointly solved for the optimization problem. The extrinsic $\boldsymbol{T_L^I}$ from LiDAR to GNSS/INS is:

$$\boldsymbol{T_L^I} = argmin\{\sum_{k=1}^{N} ||\gamma_{plane}^k||^2\} \tag{4}$$

$$||\gamma_{plane}^k||^2 = \sum_{p=1}^{M}(\mathbf{N}^T(x_p - \mathbf{X}))^2 \tag{5}$$

where $||\gamma_{plane}^k||^2$ are the residual errors about point to plane. $M$ is the total number of points in the plane.

Rough calibration extracts point cloud features and then optimizes the extrinsic parameters according to the features. Fig. 4 shows the surface features extracted by the rough calibration process and the feature point cloud used by the

rough calibration to quickly and completely calibrate from the initial state. The calibration speed of this process is very fast. We collected multiple calibration sets of data for three different scenes in our experiment, and each data contains 1000+ frame point clouds, the average time of rough calibration is shown in Table I. The calibration accuracy is shown in Table II. In terms of running time, the calibration can be completed very quickly by extracting features and optimizing, thus saving a lot of running time, which is conducive to large-scale calibration. In terms of calibration quality, if the requirements are not particularly harsh, the rough calibration result can be used as the final calibration result.

### C. Calibration Refinement

Through the previous step, we obtained an accurate enough initial value of extrinsic parameters. To further enhance the effect of mapping, we use octree-based optimization to divide the three-dimensional space into a voxel grid. We use multi-frame point clouds for splicing and construction, and the point cloud density is relatively dense. In order to improve the calibration speed, we down-sampled the spliced point cloud to a certain extent. The goal of our

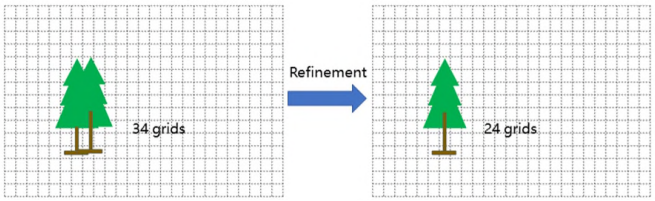


Fig. 5. Schematic diagram based on space occupancy optimization. For the point cloud bifurcation caused by inaccurate extrinsic parameters, the number of grids occupied in the space is reduced after refinement.

optimization here is the space occupancy rate. The fewer the number of grids occupied by the space is, the better the mapping effect and the more accurate the corresponding extrinsic parameters are. First, we use the result of rough calibration to remove the point cloud motion distortion through the uniform speed model. Then, the initial calibration results are converted to the world coordinate system as Eq.(3). After the point cloud is transformed into the same coordinate system, the space is divided into a voxel grid. If the calibration result is accurate, the space voxels occupied by all point clouds in the same coordinate system are the smallest, as shown in Fig. 5.

$$\boldsymbol{T_L^{I,new}} = \arg\min_{T} \{occupancy(\boldsymbol{T_L^I}, x_p)\} \tag{6}$$

where $x_p$ represents the point cloud after stitching by $\boldsymbol{T_L^I}$. We follow the grid search method described in [30] to find an optimal calibration parameter so that the point cloud occupies the least number of grids.

At the end of the calibration refinement, we get a more accurate calibration result with a better mapping effect. The octree-based calibration refinement needs to block and traverse the space, which cost much time. Thanks to the excellent performance of rough calibration, we only optimize within a small extrinsic parameter range in the refine calibration step, so the overall consumption is very low, ensuring fast calibration speed. Table I shows the average time of refinement calibration. The result of the calibration refinement is visually similar to the rough calibration in Fig. 6, the performance at this stage is evaluated by quantitative evaluation in Table II.

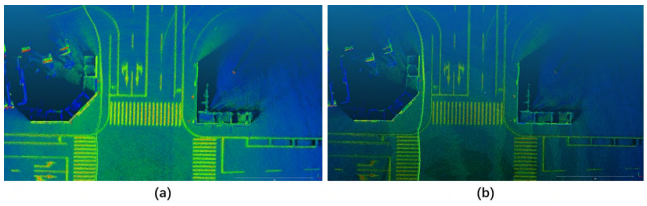


Fig. 6. The calibration results are further improved by calibration refinement. (a) is the result of rough calibration, (b) is the result of refinement, it can be seen that the wall and lane line become thinner after refinement.

TABLE I
CALIBRATION RUNNING TIME OF THREE DIFFERENT SCENES

| | Scene1 | Scene2 | Scene3 |
|---|---|---|---|
| Our Rough Calibration | 15.85 s | 15.80 s | 13.95 s |
| Our Refinement Calibration | 8.64 s | 8.24 s | 6.75 s |
| Our Z-axis Correction | 0.26 s | 0.22 s | 0.17 s |
| Lidar-align [5] | 179.18 s | 190.96 s | 199.79 s |
| hand-eye [6] | 76.04 s | 101.43 s | 167.82 s |

### D. Z-axis Correction

Because the ground is flat enough for most calibration scenarios, and the excitation in the Z-axis direction is not sufficient in this case, the extrinsic parameter calibration effect of LiDAR and GNSS/INS in the z-axis direction will be less accurate. To solve this problem, we propose to use fiducial points to optimize the calibration of the z-axis. The fiducial points are the three-dimensional points of the accurate world coordinates measured by the more precise positioning device and multiple measurements and processing. The calibration accuracy of the z-axis depends on the accuracy of the fiducial points. It is worth mentioning that each calibration site only needs to measure the fiducial points once in advance, and any subsequent calibration of vehicles can be performed at this calibration site without re-measurement of the fiducial points.

Finally, we correct the error on the z-axis. We take $K$ fiducial points of the whole map and project the map to the global coordinate system to build a local map. Then the nearest neighbor of each fiducial point is found on the local map for least square optimization to obtain the final corrected offset on the Z axis. Moreover, this process takes almost no time shown in Table I. The optimization equation is as follows:

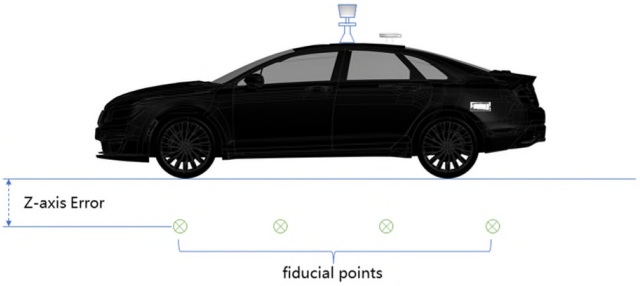


Fig. 7. The calibration error of the z-axis is further corrected by the fiducial points set in advance on the ground. The reference points in world coordinates are obtained by the 3D coordinate measuring device.

$$Z_{fix} = \arg\min_{Z} \sum_{i=1}^{K} \{||X_{mi} - X_{bi}||^2\} \tag{7}$$

Where $X_{bi}$ represents the i-th fiducial point and $X_{mi}$ represents the nearest neighbor point of the i-th fiducial point in the local map. Fig. 7 shows the height error between the ground point on the local map and the fiducial point in the actual global coordinate system due to the z-axis error. Fig. 8 shows the projection results of the map with and without Z-axis correction onto the image. If the Z-axis is corrected, it can be perfectly matched. This is to show an example of the impact of an inaccurate Z-axis. In many cases, minor errors in the Z-axis will not affect the automatic driving function because the car is walking on the ground plane. Otherwise, some errors in the z-axis will cause misaligning in height. The performance at this stage is evaluated by quantitative evaluation in Table II.

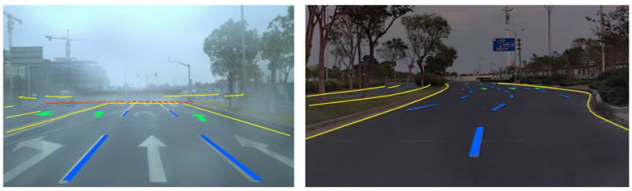


Fig. 8. (a) is the projection of the map on the image without z-axis correction, and the lane line projection of the map and the lane lines in the image are not aligned. (b) is the projection of the map on the image with z-axis correction.

## IV. EXPERIMENTS

To evaluate the performance of our method, experiments are conducted on three sets of data. As shown in Fig. 9, we selected three calibration scenarios on the road and recorded ten group calibration data for each scenario. The data collection method is shown in Fig. 3 and keeps the vehicle speed between 10 km/h and 20 km/h.

### A. Experiment Settings

We conducted experiments on real driverless platforms, Fig. 1 shows our realistic experiment setup. Top is Hesai Pandar64 LiDAR. GNSS/INS deveice is Novatel PwrPak7 placed in the trunk with two antennas on the roof. To evaluate the performance of our method with respect to the reference calibration, we separately measure the error for translation and rotation. We also calculate the mean absolute error (MAE) for the three components of translation, namely $\Delta t_x$, $\Delta t_y$, and $\Delta t_z$, as well as the MAE for the three Euler angles $\Delta roll$, $\Delta pitch$, and $\Delta yaw$, which follow the $ZYX$ representation. Our method is implemented in C++ on a desktop computer with an Intel Core i7-8700 CPU and a Nvidia 1660 GPU.

### B. Qualitative Results

In order to better visualize the performance of our method, we spliced the point clouds according to the calibration results. Fig. 9 shows the results before and after calibration. Because the rough calibration results are already excellent, the final and rough calibration results are very similar and hard to see the difference in visualization. In addition, to evaluate the convergence properties of the rough calibration, we gave the initial values of the extrinsic parameters randomly generated in angle and translation from 20° and 0.5m compared to the ground truth. The whole procedure only takes around 30 s to run in our experiment, and If only the rough calibration process is required, it only takes less than 10 s to converge. It is necessary to mention that, to perform rough calibration more robustly, the number of rounds we actually perform calibration is more than shown in Fig. 10 and Fig. 11, so the final running time is about 15s, shown in Table I. Fig. 8 shows a comparison of visualizations with and without Z-axis correction, and Z-axis correction reduces the height error with the High-precision map. These experimental results fully demonstrate the robustness and adaptability of our algorithm.

### C. Quantitative Results

To quantitatively evaluate the error of our calibration method, we obtained a relatively accurate calibration ground truth through multiple measurements and the manual tuning tools. The calibration parameter was verified by various methods, compared with the measured values, and can be taken as the ground truth. The MAE is shown in Table II and our calibration has a higher accuracy. It should be noted that the three-stage results of our algorithm shown in Table II are in order instead of their individual effects. We also made the distribution statistics on the calibration results of thirty calibration data and obtained the residual vector whose statistical information (e.g., mean, variance) reveals the rough and refine optimization quantitatively. The results are shown in Fig. 12. It can be seen from the figure that the fluctuation of the calibration extrinsic is smaller for rough and fine calibration, and calibration consistency is also better. Meanwhile, the calibration refinement further improves the calibration accuracy, especially rotation. On the quantitative evaluation of the z-axis, the average error of the z-axis after z-axis correction is kept within 0.015 m.

### D. Comparison Experiments

We mainly compared it with Lidar-align [5] and hand-eye [6]. Table I and Table II show the calibration running time

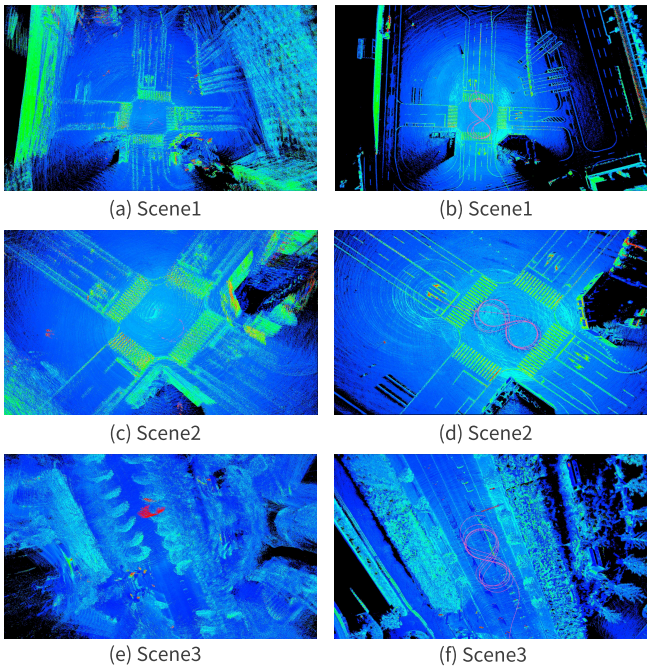


Fig. 9. The left column in the figure is the point cloud stitching result under the initial inaccurate extrinsic parameters, and the right column is the point cloud stitching result after calibration.

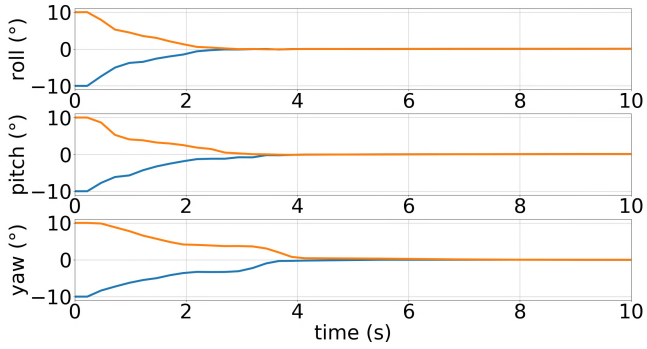


Fig. 10. In the rough calibration process, the three angles converge to a fixed value with time, and the vertical axis is the error between the current optimization result and the ground truth.

and accuracy comparison using the same calibration initial value on thirty calibration data. At the same time, we found [5] has lower calibration accuracy when the initial value error of the extrinsic parameter is large and this method does not correct the z-axis calibration accuracy. In contrast, our method still maintains high accuracy even with poor initial values. Comparative experiments show that our method has advantages in both running time and calibration accuracy.

## V. CONCLUSIONS

This paper proposes a three-stage LiDAR to GNSS/INS extrinsic calibration method that maintains high performance in both running time and accuracy. Our method is specifically designed for autonomous driving and can be applied to the rapid and large-scale calibration of autonomous vehicles.

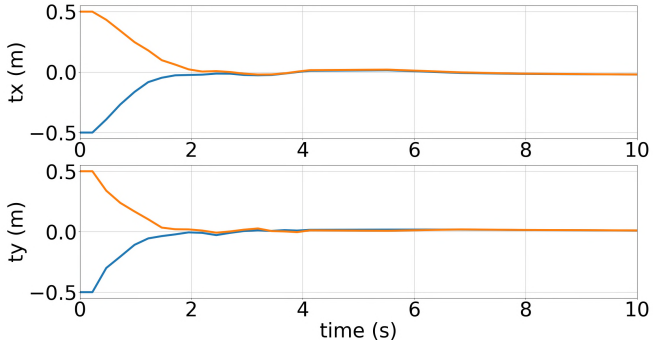


Fig. 11. In the rough calibration process, the two translation converge to a fixed value with time, and the vertical axis is the error between the current optimization result and the ground truth.

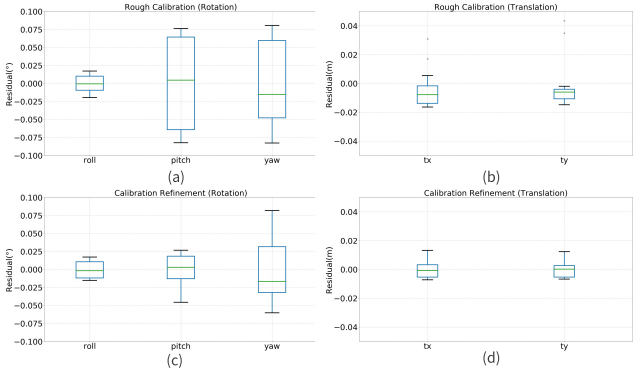


Fig. 12. (a) and (b) is the consistency evaluation for the rotation and translation of rough calibration. (c) and (d) is the consistency evaluation for the rotation and translation of calibration refinement.

This method is optimized by multi-plane features, and the performance may be reduced in the case of relatively few plane features scene. In the future, we will improve our method to adapt to extreme environments with fewer features.

## VI. ACKNOWLEDGEMENT

We thank the anonymous reviewers for their valuable comments. The research was supported by Shanghai Artificial Intelligence Laboratory, the National Key R&D Program of China (Grant No. 2022ZD0160104) and the Science and Technology Commission of Shanghai Municipality (Grant No. 22DZ1100102).

TABLE II
MAE FOR TRANSLATION AND ROTATION OF OUR METHOD

| Calib | $\Delta t_x$(m) | $\Delta t_y$(m) | $\Delta t_z$(m) | $\Delta roll$(°) | $\Delta pitch$(°) | $\Delta yaw$(°) |
|---|---|---|---|---|---|---|
| Rough | 0.01705 | 0.01541 | 0.35101 | 0.03742 | 0.10124 | 0.13361 |
| Refined | 0.00902 | 0.00716 | 0.23717 | 0.01561 | 0.04878 | 0.06104 |
| Z-axis Correction | 0.00726 | 0.00793 | 0.01169 | 0.00311 | 0.03426 | 0.07468 |
| Lidar-align [5] | 0.21749 | 0.20326 | 0.32463 | 1.49031 | 1.45612 | 0.11949 |
| hand-eye [6] | 0.15609 | 0.05542 | 0.5749 | 0.0406 | 0.11013 | 1.23493 |

## REFERENCES


[1] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2018.

[2] S. U. H. Syed, "Lidar sensor in autonomous vehicles," 03 2022.

[3] D. Fernandes, T. Afonso, P. Girão, D. Gonzalez, A. Silva, R. Névoa, P. Novais, J. Monteiro, and P. Melo-Pinto, "Real-time 3d object detection and slam fusion in a low-cost lidar test vehicle setup," *Sensors*, vol. 21, no. 24, p. 8381, 2021.

[4] H. Tang, X. Niu, T. Zhang, L. Wang, and J. Liu, "Le-vins: A robust solid-state-lidar-enhanced visual-inertial navigation system for low-speed robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[5] A. M. Zachary Taylor, "A simple method for finding the extrinsic calibration between a 3d lidar and a 6-dof pose sensor," [EB/OL], https://github.com/ethz-asl/lidar_align Accessed April 4, 2019.

[6] Y.Li, "Calculate the relative pose of lidar and ins using hand-eye calibration," [EB/OL], https://github.com/liyang-whu/lidar_rtk_calibration Accessed March 27, 2023.

[7] M. Horn, T. Wodtko, M. Buchholz, and K. Dietmayer, "Online extrinsic calibration based on per-sensor ego-motion using dual quaternions," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 982–989, 2021.

[8] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[9] T. Shan and B. Englot, "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)*. IEEE, 2018, pp. 4758–4765.

[10] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5112–5119.

[11] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.

[12] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "3d lidar-imu calibration based on upsampled preintegrated measurements for motion distortion correction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2149–2155.

[13] J. Lyu, J. Xu, X. Zuo, and Y. Liu, "An efficient lidar-imu calibration method based on continuous-time trajectory," in *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications. Macau, China*, 2019.

[14] J. Lv, J. Xu, K. Hu, Y. Liu, and X. Zuo, "Targetless calibration of lidar-imu system based on continuous-time batch estimation. in 2020 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9968–9975.

[15] S. Mishra, G. Pandey, and S. Saripalli, "Target-free extrinsic calibration of a 3d-lidar and an imu," in *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2021, pp. 1–7.

[16] S. Li, L. Wang, J. Li, B. Tian, L. Chen, and G. Li, "3d lidar/imu calibration based on continuous-time trajectory estimation in structured environments," *IEEE Access*, vol. 9, pp. 138 803–138 816, 2021.

[17] J. Lv, X. Zuo, K. Hu, J. Xu, G. Huang, and Y. Liu, "Observability-aware intrinsic and extrinsic calibration of lidar-imu systems," *IEEE Transactions on Robotics*, 2022.

[18] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[19] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 409–416.

[20] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4304–4311.

[21] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online imu intrinsic calibration: Is it necessary?" in *Robotics: Science and Systems*, 2020.

[22] W. Liu and Y. Li, "Error modeling and extrinsic–intrinsic calibration for lidar-imu system based on cone-cylinder features," *Robotics and Autonomous Systems*, vol. 114, pp. 124–133, 2019.

[23] W. Liu, Z. Li, R. Malekian, M. A. Sotelo, Z. Ma, and W. Li, "A novel multifeature based on-site calibration method for lidar-imu system," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9851–9861, 2019.

[24] J. Jiao, Y. Yu, Q. Liao, H. Ye, R. Fan, and M. Liu, "Automatic calibration of multiple 3d lidars in urban environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 15–20.

[25] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[26] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

[27] J. Rehder, P. Beardsley, R. Siegwart, and P. Furgale, "Spatio-temporal laser to visual/inertial calibration with applications to hand-held, large scale scanning," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 459–465.

[28] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *IEEE Transactions on Robotics*, pp. 1–18, 2015.

[29] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-lidar calibration: A targetless and structureless approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1556–1563, 2020.

[30] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, vol. 2, 2013, p. 7.