An Extrinsic Calibration Method of a 3D-LiDAR and a Pose Sensor for Autonomous Driving

Guohang Yan, Jiahao Pi, Chengjie Wang, Xinyu Cai and Yikang Li†

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 of a 3D-LiDAR and a pose sensor 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 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 natural characteristics of road scenes, making it independent and easy to apply in large-scale conditions. Experimental results on real-world data sets demonstrate the reliability and accuracy of our method. The codes are open-sourced on the Github website. To the best of our knowledge, this is the first open-source code specifically designed for autonomous driving to calibrate LiDAR and pose-sensor extrinsic parameters. 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. Accurate and reliable location information is becoming indispensable to realizing the various complex functions of self-driving cars. Therefore, most self-driving cars will be equipped with high-precision pose sensors, such as NovAtel, Trimble and other high-precision positioning devices. 3D-LiDAR (Light Detection and Ranging) is another crucial sensor in autonomous driving technology. SLAM (Simultaneous Localization and Mapping) and object detection are the two most important applications of LiDAR in the field of autonomous driving, and the main role of SLAM is mapping and localization. It should be noted that each laser point of LiDAR is generated at a different reference pose during the motion of the self-driving car, which is also the source of LiDAR motion distortion. In autonomous driving, the motion error of the laser frame caused by motion cannot be ignored. Usually, the motion distortion of the LiDAR is removed with the help of the pose sensor. High-precision map construction and localization highly depend on LiDAR and pose sensor fusion. Accurate 3D-LiDAR and pose sensor extrinsic calibration is essential to determine their coordinate relationship and perform sensor fusion.

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

Due to the rapid development of LiDAR-IMU systems in recent years [1]–[4], many methods have also emerged to calibrate LiDAR-IMU [5]–[10]. These methods generally calibrate the intrinsic parameters of the IMU and the extrinsic parameters of the LiDAR-IMU for small robot platforms. However, these calibration methods do not necessarily work well when directly used for calibrating LiDAR to pose sensors in autonomous vehicles. Autonomous vehicles are usually equipped with high-precision pose sensors and only need to calibrate extrinsic parameters. Currently, there is no calibration method from LiDAR to pose-sensor specifically for autonomous driving. Therefore, we propose a calibration method from LiDAR to pose sensors specifically for autonomous driving applications in this paper. Inspired by [11], 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. Through this three-stage method, an accurate extrinsic parameter that overcomes the dimension of plane motion is gradually obtained. This three-stage framework is targetless and can be calibrated using

Fig. 1. (a) Left view of our experimental platform. (b) Top view of our experimental platform.
some surface features in the environment. Although the first stage is called rough calibration, the calibration accuracy is also very high, especially in environments with more plane features. On the basis of the first stage, the second stage performs minor optimizations to further improve the calibration accuracy. At the same time, we use some fiducial points of world coordinates to correct the lack of constraint on the Z axis caused by the plane motion so that it is very accurate in the six degrees of freedom parameters for rotation and translation. Our method is less demanding in the environment setting and is a fully automatic calibration procedure, while the time required for calibration is relatively very small, so the proposed method can contribute to a more efficient and practical large-scale autonomous vehicle calibration and deployment.

The contributions of this work are listed as follows:

1) The proposed method is the first-known automatic and coarse-to-fine calibration method specifically designed for autonomous driving to calibrate LiDAR and pose-sensor extrinsic parameters.

2) We introduce an algorithm for solving LiDAR and pose-sensor extrinsic calibration initialization by directly minimizing the distance from the feature point to the plane.

3) 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.

4) Evaluated 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 [12]–[16]. 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-IMU extrinsic is usually calibrated together with IMU intrinsic calibration, e.g. methods [5]–[10].

LV et al. [7] proposed the first open-source LiDAR-IMU calibration toolbox based on continuous-time batch estimation. This method can correct the offset of the coordinate system in different positions, proving the method’s reliability. Subsequently, they extended this method and proposed OALiCAlib [10], which seeks to automatically select the most informative data segment for calibration, avoiding some data without sufficient motion or scene constraints, thus improving calibration accuracy and reducing the computational cost.

Gentil et al. [9] 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. [17] obtained the initial value of calibration through hand-eye calibration [18]. 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. [8] proposed an optimization scheme based on EKF to calibrate LiDAR and IMU. [19], [20] proposed a spatio-temporal calibration method using the continuous-time batch estimation framework for the camera-IMU calibration. Forster et al. [21] solved a non-linear batch estimation problem to determine the unknown extrinsic calibration parameter. The above calibration methods need to be based on good initial values. Park et al. [22] 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.

Most existing methods are used to calibrate the LiDAR-IMU extrinsic and the IMU intrinsic simultaneously. Due to the lack of Z-direction motion, the existing calibration methods are usually difficult to correct the offset of the z-axis. In autonomous driving, the pose-sensor can output the precise pose in the global coordinate system, and only the extrinsic parameters need to be calibrated without considering the intrinsic parameters. Our work represents a coarse to fine calibration for LIDAR and pose-sensor. After getting a satisfactory result, our method also corrects the deviation of the z-axis through fiducial points matching.

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 pose-sensor sequence data at the intersection by walking three figure-8-shape trajectories. Then, the pose-sensor pose data corresponding to the LiDAR timestamp is obtained through the data processing module. With accurate extrinsic parameters, we can reconstruct the surrounding environment based on LiDAR and pose-sensor data. Therefore, Our goal is to find a rigid transformation \( T = \{ R, t \} \) from LiDAR to the pose-sensor so that the 3D reconstruction results obtained by splicing the multi-frame point clouds through the pose provided by the pose-sensor are more accurate. \( R \) is a 3D rotation, \( R \in SO(3) \) and \( t \) is a 3D translation, \( t \in \mathbb{R}^3 \). First, we process the pose-sensor output pose \( T_{\text{pose-sensor}} \) in the world coordinate system and convert it to obtain the pose \( T_{\text{lidar}} \) of the point cloud by the extrinsic parameters \( T \).

\[
T_{\text{lidar}}^i = T \cdot T_{\text{pose-sensor}}^i = \{ R_{\text{lidar}}^i, t_{\text{lidar}}^i \} \tag{1}
\]
Then, recover the 3D reconstruction map $M$ through LiDAR pose $T_{\text{lidar}}$ and LiDAR point cloud sequence $P_{\text{lidar}}$.

$$M = \sum_{i=1}^{N} (R_{\text{lidar}}^i \cdot P_{\text{lidar}}^i + t_{\text{lidar}}^i)$$  \hspace{1cm} (2)$$

Where $M$ is the global 3D point-cloud map in the world coordinate system, we aim to find an extrinsic parameter $T$ to make the reconstructed map $M$ has the best quality.

**B. Rough Calibration**

The first step is rough calibration. Rough calibration aims to quickly reduce the extrinsic parameter from any initial value to a small error range. However, the initial extrinsic parameters will not be particularly outrageous through the sensors’ coordinate axis alignment and displacement measurement, so there is no selection of the particular outrageous initial extrinsic parameters in our test. 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 time-consuming, we perform feature extraction on the point cloud. Similar to method [11], 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 the pose-sensor 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 the pose-sensor is denoted as $T_{I1},...,T_{In}$.

We denote the extrinsic parameter from LiDAR to the pose-sensor as $T$. 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 center point $X$ and normal vector $N$ of the current plane are recorded through the plane fitting. Then the point cloud coordinate transformation casts the point cloud on the first frame. The formula for projecting the point cloud in the LiDAR coordinate system of frame $n$ to the pose-sensor coordinate system of frame 1 is as follow:

$$x_{I1} = T_{I1}^{-1}T_{In}T x_{Ln}$$  \hspace{1cm} (3)$$

where $x_{Ln}$ represents the n-th frame point cloud in the LiDAR coordinate system. After being converted to the same coordinate system, the next step is the data association. Similarly, the points on the plane are found through point cloud feature extraction, and then the point cloud in frame $n$ with the plane corresponding to frame 1 and the nearest neighbors of all plane points in frame 1 and frame $n$ are jointly solved for the optimization problem:

$$T_{I1}^L = \arg\min\{\sum_{k=1}^{N} ||\gamma_{\text{plane}}^k||^2\}$$  \hspace{1cm} (4)$$

$$||\gamma_{\text{plane}}^k||^2 = \sum_{p=1}^{M} (x_p - X)N$$  \hspace{1cm} (5)$$
where \(||r_k^{A}||^2\) are the residual errors about point to plane. By solving the above least squares problem, we can get an initial solution of the calibration. It should be noted that do not perform de-distortion processing in the rough calibration, because the speed of the vehicle to collect the calibration data is very slow. In our test, the results of de-distortion and no de-distortion are not much different. Therefore, to increase the speed of rough calibration, the de-distortion process is not performed.

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 time-consuming calibration, the calibration can be completed very quickly by extracting features and optimizing, thus saving a lot of time-consuming calibration, 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 excellent initial value of extrinsic parameters. To further enhance the effect of mapping, we use octree-based optimization similar to [24] to divide the three-dimensional space into a voxel grid. Compared with [24], we use multi-frame point cloud construction, and the point cloud density is relatively dense, so the effect is better. The goal of our optimization here is 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 pose-sensor system of the first frame through the pose-sensor pose 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.

\[
T = \arg \min_T \{\text{occupancy}(T, x_p)\} \tag{6}
\]

where \(x_p\) represents the point cloud after stitching by \(T\).

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, so space and time costs are very high. 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, and the calibration is very fast. 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.](image)

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 pose-sensor in the z-axis direction will be terrible. To solve this problem, we propose to use fiducial points to optimize the calibration of the z-axis. Because the pose-sensor is output by the fusion of GPS and IMU, it outputs a global Pose. The fiducial points are the three-dimensional point of the accurate world coordinates measured by the positioning device. In our calibration venue, we can randomly select three or more points on the ground as the fiducial points for precise measurements. Optimize the error of the Z-axis through the fiducial points to further reduce the mapping error in height to a smaller value.

After obtaining the precision calibration results, the last step is to correct the z-direction translation of the calibration results. As we mentioned above, due to the lack of movement of the vehicle in the z-axis direction, the results obtained are likely to have errors in the translation in the z-direction, and

![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.](image)
where $X$ as follows: the local map for least square optimization to obtain the final coordinates are obtained by the 3D coordinate measuring device. Then the nearest neighbor of each datum point is found on the local map to the global coordinate system to build a local map.

We take $K$ reference points of the whole map and project the map to the 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. Otherwise, some errors in the $z$-axis will cause misaligning in height.

**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. Through qualitative and quantitative evaluation, the results show that the proposed method has high accuracy and robustness.

**A. Experiment Settings**

We conducted experiments on real driverless platforms, Fig. 1 shows our realistic experiment setup. Top is Hesai Pandar64 LiDAR. Pose-sensor 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 $\Delta \text{roll}$, $\Delta \text{pitch}$, and $\Delta \text{yaw}$, 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. 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 as shown in Fig. 10 and Fig. 11.

Moreover, as our method requires less time consuming and is fully automatic, we conclude that our method is highly efficient and user-friendly to be deployed on a broad scale of autonomous vehicles.

**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, in order to evaluate the convergence properties of the rough calibration, we gave the initial values of the extrinsic parameters that are huge differences. As shown in Fig. 10 and Fig. 11 the convergence and stability of the rough calibration are relatively good, and Fig. 12 shows the consistency under different calibration scenarios is also rather good. 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 the manual tuning tools in [25]. First, we got an extrinsic parameter through our calibration algorithm, and then manual tuning was performed based on this extrinsic parameter. After some verification by hand tuning and downstream algorithms, the result can be taken as the ground truth. The MAE is shown in Table I. We made the error 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.

```plaintext
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. Through qualitative and quantitative evaluation, the results show that the proposed method has high accuracy and robustness.

A. Experiment Settings

We conducted experiments on real driverless platforms, Fig. 1 shows our realistic experiment setup. Top is Hesai Pandar64 LiDAR. Pose-sensor 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 $\Delta \text{roll}$, $\Delta \text{pitch}$, and $\Delta \text{yaw}$, 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. 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 as shown in Fig. 10 and Fig. 11. Moreover, as our method requires less time consuming and is fully automatic, we conclude that our method is highly efficient and user-friendly to be deployed on a broad scale of autonomous vehicles.

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, in order to evaluate the convergence properties of the rough calibration, we gave the initial values of the extrinsic parameters that are huge differences. As shown in Fig. 10 and Fig. 11 the convergence and stability of the rough calibration are relatively good, and Fig. 12 shows the consistency under different calibration scenarios is also rather good. 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 the manual tuning tools in [25]. First, we got an extrinsic parameter through our calibration algorithm, and then manual tuning was performed based on this extrinsic parameter. After some verification by hand tuning and downstream algorithms, the result can be taken as the ground truth. The MAE is shown in Table I. We made the error 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 0.01 m.

D. Comparison Experiments

We mainly compared it with [23], an open-source extrinsic calibration method between a 3D lidar and a 6-dof pose sensor. Table II and Table II show the comparison of calibration time and calibration accuracy. [23] has lower calibration accuracy when the initial value error of the extrinsic parameter is large. When the initial value is better, the accuracy is also good, 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

![Fig. 9](image1)
![Fig. 10](image2)

V. CONCLUSIONS

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

![Fig. 11](image3)
![Fig. 12](image4)

| Method         | $\Delta t_x$(m) | $\Delta t_y$(m) | $\Delta$roll(°) | $\Delta$pitch(°) | $\Delta$yaw(°) |
|----------------|-----------------|-----------------|-----------------|-----------------|----------------|
| Calib          | 0.01705         | 0.01541         | 0.03742         | 0.10124         | 0.13361        |
| Rough          | 0.00902         | 0.00716         | 0.01561         | 0.04878         | 0.06104        |
| Refined        | 0.21749         | 0.20326         | 1.49031         | 1.45612         | 0.11949        |
| Method [23]    | 0.21749         | 0.20326         | 1.49031         | 1.45612         | 0.11949        |
REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] S. Mishra, G. Panedi, 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.

[9] 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. 138803–138816, 2021.

[10] 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.

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

[12] 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.

[13] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalib: 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.

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

[15] 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.

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

[17] 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.

[18] R. Horand and F. Dormaika, “Hand-eye calibration,” *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] P. Wei, G. Yan, Y. Li, K. Fang, W. Liu, X. Cai, and J. Yang, “Croon: Automatic multi-lidar calibration and refinement method in road scene,” *arXiv preprint arXiv:2203.03182*, 2022.

[25] G. Yan, Z. Liu, C. Wang, C. Shi, P. Wei, X. Cai, T. Ma, Z. Liu, Z. Zhong, Y. Liu, et al., “Openecalib: A multi-sensor calibration toolbox for autonomous driving.” *Software Impacts*, vol. 14, p. 100393, 2022.