ORIGINAL ARTICLE

Automatic Miscalibration Detection and Correction of LiDAR and Camera Using Motion Cues

Pai Peng[1], Dawei Pi[2], Guodong Yin[1*] [ID], Yan Wang[3], Liwei Xu[1] and Jiwei Feng[1]

Abstract

This paper aims to develop an automatic miscalibration detection and correction framework to maintain accurate calibration of LiDAR and camera for autonomous vehicle after the sensor drift. First, a monitoring algorithm that can continuously detect the miscalibration in each frame is designed, leveraging the rotational motion each individual sensor observes. Then, as sensor drift occurs, the projection constraints between visual feature points and LiDAR 3-D points are used to compute the scaled camera motion, which is further utilized to align the drifted LiDAR scan with the camera image. Finally, the proposed method is sufficiently compared with two representative approaches in the online experiments with varying levels of random drift, then the method is further extended to the offline calibration experiment and is demonstrated by a comparison with two existing benchmark methods.

Keywords Autonomous vehicle, LiDAR and camera, Miscalibration detection and correction, Sensor drift

1 Introduction

Autonomous vehicle has drawn tremendous attention in the last decade from both industrial and academic due to its promise in boosting traffic efficiency, improving road safety and minimizing energy consumption [1, 2]. It is often equipped with LiDAR and camera to obtain reliable and comprehensive information about the surroundings. Road detection [3–5], object tracking [6–8], and trajectory prediction [9, 10] are exciting examples of LiDAR and camera fusion perceptions. In these paradigms, accurate extrinsic calibration is the prerequisite for optimally combining the measurements from the multi-sensor platform.

Extrinsic calibration is the process of identifying the rigid-body transformation between the coordinate system of the two sensors, which minimizes the alignment error between corresponding points [11]. Most of the calibration methods are based on recognizing and matching the common features from sensor data to determine the calibration parameters [12, 13]. Although they provide a good estimate of the calibration parameters, traditional methods [14, 15] require special fiducials and user intervention, it is widely seen as having been onerous and time-consuming.

Accordingly, many works attempt automatic calibrate LiDAR and camera utilizing features extracted from the natural scenes, such as planes [16, 17], lines [18–20], or edges [21–23] without any dedicated targets, and achieves well calibration performance. Nevertheless, most of these schemes are typically implemented offline, they cannot cope with the task of extrinsic calibration during vehicle driving.

To address these issues, Schneider et al. [24] employ sensor odometry to recursively estimate the extrinsic calibration online. One classic motion-based method

*Correspondence:
Guodong Yin
ygd@seu.edu.cn
[1] School of Mechanical Engineering, Southeast University, Nanjing 211102, China
[2] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210014, China
[3] School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore

proposed by Taylor and Nieto [25] extends the hand-eye calibration method and incorporates it into a probabilistic framework, which can be used to calibrate sensors during vehicle operation. However, these methods only focus on the online extrinsic calibration, and do not explore the monitoring of the possible relative pose changes of the sensor during vehicle driving. For instance, the relative pose of camera and LiDAR may change due to the vibration of vehicle body in the topologically variable driving scenario, significantly degrading further perception performance.

If such sensor drift cannot be detected and responded in time, it may put people and vehicle nearby under hazardous situations. Hence, an automated pipeline that detects the sensor drift and recalibrates the sensors online is required to maintain the continuous operation of autonomous vehicle. To the best of our knowledge, there are only a few works explored on such topic. Corte et al. [26] build a unified online pipeline to concurrently estimate the extrinsic parameters and observe the parameters evolution in long-term operation. However, the detection speed of this method is too slow to be used on a driving vehicle. Levinson's method [27] and later variant [28] first product the depth discontinuity of projected LiDAR points and edge intensity from the image, then derive a formula that yields the probability that the current calibration is accurate and further track the change in extrinsic parameter over time. But the detection and correction accuracy of this method is not satisfactory due to its inherent blurred projection edges, numerous invalid edges, and limited search space.

To address the research gap, i.e., to timely and accurately detect and correct the sensor drift during vehicle driving, an automatic LiDAR and camera miscalibration detection and correction framework using sensor motion cues is proposed. Unlike the above-mentioned works, the proposed method uses the deviations of the rotational motions from the two sensors to detect the relative pose change in the current frame. To cope with potential data noise, the sensor motions with their confidence level are considered between several historical frames and current frame, the number of historical frames and confidence is determined by the training data, achieving the best accuracy and timeliness in determining sensor drift. If the sensor drift in current frame is determined, it will be corrected immediately. The Perspective-n-Point (PnP) algorithm is used to establish the correspondences between 3-D LiDAR points and 2-D image points to estimate the scaled camera motion, which is further used to correct the extrinsic parameters. It also considers historical frame information to ensure the accuracy of the correction. Besides, unlike Levinson's method which tries to find the corresponding features from completely different modalities, the proposed method estimates each sensor motion in the same modality which has been widely validated and applied to ensure the accuracy of the algorithm.

Specifically, this paper mainly presents three contributions. First, an accurate and reliable miscalibration determination method is devised based on the motion deviation between LiDAR and camera. Second, an automatic correction algorithm is proposed to recalibrate the sensor pair using the scaled camera motions estimated by associating the 2-D feature points with corresponding 3-D points. Third, the proposed online miscalibration detection and correction method is further extended to the offline scenario.

2 Related Works
2.1 Offline Approaches
Researches [16] present similar approaches to such problem using geometric constraints associated with a trihedral object which is ubiquitous in human-made environments, such as two adjacent walls together with floor or ceiling. Aiming at the extrinsic calibration for multiple small FoV LiDARs and cameras, Liu et al. [17] propose an adaptive voxel map to shorten the calibration time based on the existence of natural plane features. Yu et al. [18] estimate the correspondences between vanishing points in the image and the structural lines in the point clouds to find camera pose in LiDAR coordinate system. Nevertheless, these methods are only applicable in structured environments.

To broaden the application range, Brown et al. [19] simultaneously consider both points and lines information from the image and point cloud, but it still requires a dense 3D point set. To solve these concerns, a robust and globally-optimal inlier set cardinality maximization approach is designed in Ref. [20], it can be used in circumstances where the structure is not obvious. Compared with point, line, or plane features mentioned above, edge features are more common in the environment. Kang et al. [21] and Taylor et al. [22] estimate calibration parameters by aligning the extracted edges from images and point cloud with the Gaussian mixture model and the gradient orientation measure, respectively. To achieve pixel-level accuracy of aligning natural edge features in the two sensors, Yuan et al. [23] specially developed an efficient and accurate LiDAR edge extraction method based on point cloud voxel cutting and plane fitting, leading to more accurate calibration parameters. Besides, Pandey et al. [29] maximize the mutual information between the reflectivity of point cloud and the corresponding intensity of the image pixels to obtain accurate calibration between LiDAR and camera, but this approach relies on specific sensor data and is susceptible

Peng *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:50

Page 3 of 12

to the uniformity of lighting. Ishikawa et al. [30] optimize the hand-eye LiDAR and camera calibration framework by integrating the scaled camera motion using the sensor fusion odometry, without the reflectance information. However, none of the above methods could cope with the online calibration situation.

2.2 Online Approaches
Schneider et al. [24] use the observed motion difference in sensor odometry to estimate the extrinsic parameter with the unscented kalman filter, however, the filter is sensitive to the initial state vectors and the covariance matrices, the optimization of which might get stuck in local minimum. Taylor and Nieto [25] extend the motion-based calibration in the form of hand-eye calibration, which examines the individual sensor motion and the uncertainty of its reading, an appearance-based refinement approach is further used to increase its accuracy. This method provides comparable quality to most laborious techniques and is suitable for end users. Based on the dual quaternions representation, Horn et al. [31] provide online capable global and local strategies that are used for non-planar and planar motion-based calibration. However, these methods lack the mechanism of detecting the sensor drift during vehicle driving, if an autonomous vehicle can not detect the sensor drift in time, it will bring potential safety hazards to the following driving.

To address this issue, Corte et al. [26] devise a unified motion-based calibration method to simultaneously estimate the kinematic parameters and extrinsic parameters, while continuously observe the parameters evolution. But the detection speed reported in their experiments reaches 30 seconds, which is unacceptable for a driving vehicle. Levinson et al. [27] report a real-time miscalibration detection and automatic tracking algorithm, it operates on the principle that depth discontinuities in the LiDAR data tend to project onto the edges in the image when the calibration is accurate. Based on this principle and the statistical test, Levinson's method can detect whether the current sensor calibration is accurate in each frame and track the sensor drift over time. However, the background edge tends to intersect with the foreground edge due to the parallax between two sensors, resulting in edge blur in the projected depth image. Moreover, a large number of invalid edge features in the camera image are not related to the depth discontinuities, which may lead the optimization in the wrong direction. To narrow the search space of the extrinsic parameters for robust convergence, later variant work by Xu et al. [28] utilize the monocular depth estimation together with the edge information to penalize the large deviation caused by invalid edges, but the depth estimated by the monocular visual odometry is not accurate due to its scale ambiguity.

Deep learning based techniques are also leveraged to monitor and correct calibration errors online. Schneider et al. [32] train five neural network models with different degrees of sensor drift to iteratively refine the calibration output, but it needs to be retrained each time the sensor intrinsics vary since it overlooks the output space geometry nature. For this, Iyer et al. [33] train CalibNet with geometric supervision by reducing the dense photometric error and dense point cloud distance error between the misaligned and ground truth depth maps. On this basis, Mharolkar et al. [34] propose RGBDTCalibNet in which low level features and high level features are combined to establish correspondences in the scene. But, the above methods emphasize minimizing the calibration error while ignoring the tolerance within the error bounds. To address this issue, Yuan et al. [35] use a deep generative model to learn an implicit tolerance model, it can estimate a wide range of miscalibration using a single frame image and point cloud as the inputs. However, the deep learning based frameworks need a large number of training data which their performance heavily relies on, this means the calibration may be unsatisfying when the environment or sensor configuration changes.

3 Overview of the Approach
When the sensors drift unexpectedly during operation of the autonomous vehicle, the calibration parameters change correspondingly and the original extrinsic parameters are not applicable anymore, deteriorating the proper continuous task of the vehicle. To maintain accurate calibration, detecting the sensor drift with high reliability is firstly required, then an automated approach is expected to update the extrinsic parameters online in response to the sensor pose changing. Based on this insight, as depicted in Figure 1, the approach can be divided into the following stages.

1) Miscalibration detection: A time-domain series of corresponding camera images and LiDAR scans is first retrieved, and determine the calibration state by comparing the camera motions estimated from the image cues and point cloud cues respectively.

2) Automatic correction: If sensor drift is determined, the scaled camera motion is estimated using the projection constraints between visual feature points and LiDAR 3-D points and correct the sensor drift with point cloud transformation and registration.

4 Miscalibration Detection
First, camera is synchronized with LiDAR by searching the nearest neighbor sensor data on the time domain; then, the individual camera motions between the current frame and the last *n* frames are estimated from the two modalities; finally, considering the position error of
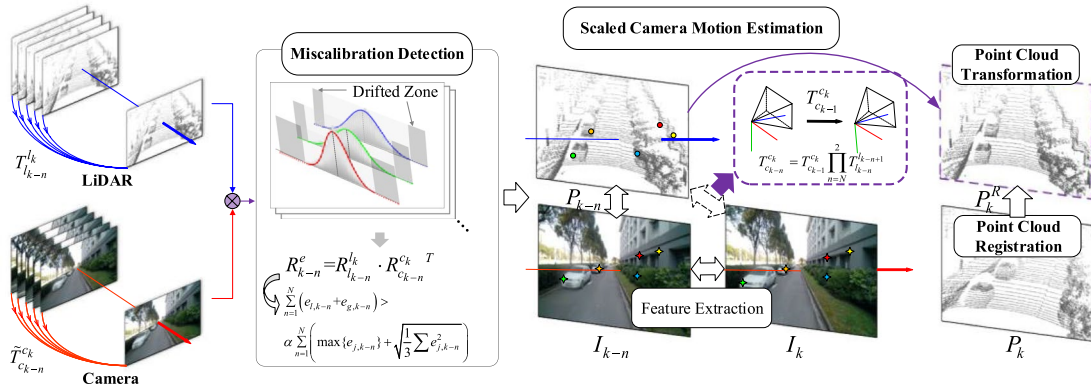
Figure 1 Overview of the proposed framework

LiDAR points at a closer distance is dominated by translation while at further distances is dominated by rotation [36], and most LiDAR points have a long-range measurement in the outdoor scene, the rotational drift in miscalibration detection is concerned only, the camera rotational motions estimated from the camera cues and LiDAR cues are compared to determine whether the current frame drifts.

4.1 Camera Motion Estimation from Visual Cues
For each new incoming camera image I_k, oriented fast and rotated brief features $u_{k,g}^f$, $g = 1, 2, ..., N_I$ are extracted and matched with the previous images I_{k-n}, $n = 1, 2, ..., N$ in a local window using the lowe's algorithm [37], the essential matrix $E_{c_{k-n}}^{c_k}$ are determined by the eight-point algorithm [38]. Then the essential matrix $E_{c_{k-n}}^{c_k}$ is decomposed into the transformation matrix $\tilde{T}_{c_{k-n}}^{c_k} = [R_{c_{k-n}}^{c_k} | \tilde{t}_{c_{k-n}}^{c_k}]$. It should be noted that the recovered translation $\tilde{t}_{c_{k-n}}^{c_k}$ is the unit norm due to the inherent scale ambiguity in monocular camera.

4.2 Camera Motion Estimation from Point Cloud
The LiDAR point cloud is firstly preprocessed, including the ground points and distant points removing, and the voxel grid down-sampling, to enhance the robustness and speed of point cloud registration. Then the point cloud is transformed into the camera coordinate system using the original extrinsic parameters according to:

$$p_c = R_l^c \cdot p_l + t_l^c, \qquad (1)$$

where $p_c \in \mathbb{R}^3$ and $p_l \in \mathbb{R}^3$ denote a 3-D point in the camera coordinate frame and the LiDAR coordinate frame, respectively. $R_l^c \in SO(3)$, is the rotation matrix and can be parametrized by the Euler angle (yaw, pitch, and roll) $[R_\psi, R_\theta, R_\phi]^T$, $t_l^c = [t_x, t_y, t_z]^T$ is the translation vector.

For each new point cloud $P_k = \{p_{k,i} \in \mathbb{R}^3 | i = 1, 2, ..., N_L\}$ with N_L number of 3-D points, it is registered with the prior point cloud P_{k-n}, $n = 1, 2, ..., N$ to find the sensor motion $T_{l_{k-n}}^{l_k}$ from the previous scans to the current. The normal distribution transform (NDT) algorithm is deployed for point cloud registration, due to its shown speed and accuracy [39]. NDT subdivides the point cloud P_k into a set of voxels and represents the spatial information of the voxels by the normal distributions M_k:

$$\mu_{k,i} = \frac{1}{N_V} \sum_{i=1}^{N_V} p_{k,i}, \qquad (2)$$

$$\Sigma_{k,i} = \frac{1}{N_V - 1} \sum_{i=1}^{N_V} \left(p_{k,i} - \mu_{k,i} \right) \left(p_{k,i} - \mu_{k,i} \right)^T, \qquad (3)$$

where N_V is the number of the voxel points, $\mu_{k,i}$ is the mean vector, and $\Sigma_{k,i}$ is the covariance matrix of the cell.

Given the reference point cloud distribution M_k and the preceding point cloud P_{k-n}, the transformation matrix $T_{l_{k-n}}^{l_k} = [R_{l_{k-n}}^{l_k} | t_{l_{k-n}}^{l_k}]$ can be found by optimizing the following objective function:

$$f\left(P_{k-n}, R_{l_{k-n}}^{l_k}, t_{l_{k-n}}^{l_k} \right) = \sum_{i=1}^{|P_{k-n}|} -d_1 \exp\left(-\frac{d_2}{2} \hat{p}_{k,i}^T \Sigma_{k,i}^{-1} \hat{p}_{k,i} \right), \qquad (4)$$

$$\hat{p}_{k,i} = R_{l_{k-n}}^{l_k} p_{k-n,i} + t_{l_{k-n}}^{l_k} - \mu_{k,i}, \qquad (5)$$

where d_1 and d_2 are positive regularizing factors.

4.3 Calibration State Determination
Once a set of sensor motions are estimated, the calibration state can be determined by comparing the corresponding rotational components $R_{c_{k-n}}^{c_k}$ and $R_{l_{k-n}}^{l_k}$. Ideally,

if there is no relative offset between the two sensors, there exists:

$$R_{l_{k-n}}^{l_k} R_{c_{k-n}}^{c_k}{}^{\mathrm{T}} = E, \qquad (6)$$

where E is an identity matrix. However, in real implementations, the motion estimations are always corrupted by measurement noises, synchronization offsets, and motion estimation errors, which means the error R_{k-n}^e between $R_{c_{k-n}}^{c_k}$ and $R_{l_{k-n}}^{l_k}$ is unavoidable even when no sensor drift occurred.

$$R_{k-n}^e = R_{l_{k-n}}^{l_k} R_{c_{k-n}}^{c_k}{}^{\mathrm{T}}. \qquad (7)$$

The error R_{k-n}^e threatens the determination of calibration state, to distinctly show the error in each rotation axis, the rotation error R_{k-n}^e is converted into the form of Euler angle:

$$\begin{aligned} e_{\psi,k-n} &= \mathrm{atan2}(r_{21}, r_{11}), \\ e_{\theta,k-n} &= \mathrm{atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right), \\ e_{\phi,k-n} &= \mathrm{atan2}(r_{32}, r_{33}), \end{aligned} \qquad (8)$$

where r_{ij} is the elements in the rotation matrix.

The typical frequency distribution histograms of the Euler angle errors from the real-world data are indicated in Figure 2. These training data including the camera image and LiDAR points are collected by the autonomous vehicle. A Gaussian can be fitted to each of the three distributions,

$$f(e_{j,k-n}) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{(e_{j,k-n} - \mu_j)^2}{2\sigma_j^2}\right), j = \psi, \phi, \theta, \qquad (9)$$
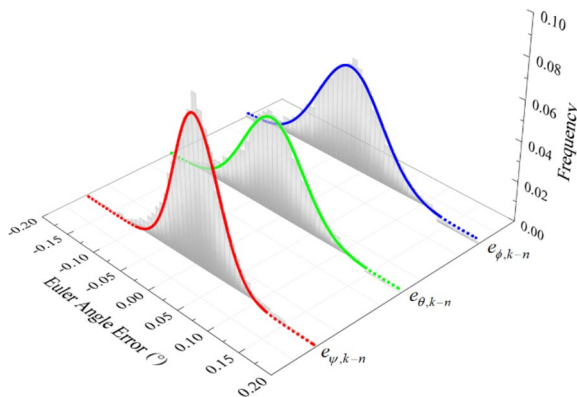

Figure 2 Typical frequency distribution histograms of the Euler angle errors from the real-world statistical data and fitted Gaussian distributions probability density curve (Solid line represents the area of $[\mu_j - 3\sigma_j, \mu_j + 3\sigma_j]$, $j = \psi, \phi, \theta$)

where μ_j is the mean, σ_j is the standard variance.

Therefore, to avoid error detections, a certain amount of error margin t_e is considered in the implementation. If the Euler angle errors are all within the margin t_e, it is believed that no sensor drift occurred in the current frame.

If any Euler angle error $e_{j,k-n}$ exceeds the margin t_e, the current frame may have drifted, the confidence $C_{j,k-n}$ of drifting occurred on the corresponding axis can be described by

$$C_{j,k-n} = \int_{-e_{j,k-n}}^{e_{j,k-n}} f(e_{j,k-n})\mathrm{d}x, j = \psi, \phi, \theta. \qquad (10)$$

Here, to maintain the detection sensitivity in each rotation direction, the maximum Euler angle error is taken into consideration:

$$e_{l,k-n} = \max\{C_{j,k-n} e_{j,k-n}\}. \qquad (11)$$

And there is also a tradeoff between the window size and error detection, a larger window could be needed to reliably detect errors, with the downside that the previous frames farther from the current frame contribute less to the detection, since the noise of motion estimation between two frames increases. Taking the above factors, the miscalibration can be determined by

$$\sum_{n=1}^{N}\left(e_{l,k-n} + e_{g,k-n}\right) > \alpha \sum_{n=1}^{N}\left(\max\{e_{j,k-n}\} + \sqrt{\frac{1}{3}\sum e_{j,k-n}^2}\right), \qquad (12)$$

where α is the global drift coefficient.

Accordingly, to determine the optimal hyperparameter, a statistical test is performed under an extensive series of coefficients and window sizes with the collected data, shown in Figure 3. As expected, the accuracy improves with the benefit of added frames, however, the difference in the data feature between the front frame and current frame increases with the window size, engendering a larger motion estimation error and further degrading the signal-to-noise ratio in miscalibration determination. Consequently, the determination accuracy under window sizes 9 and 10 decrease slightly. Instead, it is noted that a 7-frame window is significantly more robust, and the maximum determination accuracy 99.63% appears under coefficient 0.945. Hence, the above two hyperparameters are selected to accurately and robustly detect the current sensor drift.

5 Automatic Correction

When the sensor drift at the current position is detected, the automatic recalibration strategy is then implemented. The scaled camera motions $T_{c_{k-n}}^{c_k}$ are first retrieved through associating the 2-D feature points $u_{k,g}^f$ from the
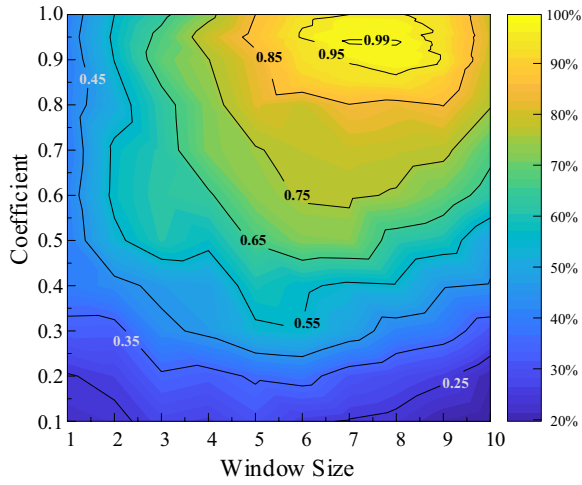
Figure 3 Determination accuracy under different coefficients and window sizes

Figure 4 Diagram of associating feature points $u_{k,g}^f$ in image I_k with the points $p_{k-n,i}$ in LiDAR scan P_{k-n}

image I_k and the corresponding 3-D points $p_{k-n,i}$ from prior point cloud P_{k-n}; then the camera motions estimated from point cloud $T_{l_{k-n}}^{l_{k-n+1}}$ is further employed to get the motion $\overline{T}_{c_{k-1}}^{c_k}$ between the current and previous frames. Through transforming the prior un-drifted LiDAR scan P_{k-1} to the current position and registered with the drifted point cloud P_k, the extrinsic parameters of the current frame $T_{l_k}^{c_k}$ are finally corrected by multiplying the registration matrix $T_{c_k}^R$.

5.1 Association of Point Cloud with Feature Points

As depicted in Figure 4, association of depth measurements $p_{k-n,i}$ with the visual feature points $u_{k,g}^f$ is executed by first projecting the prior un-drifted LiDAR scan P_{k-n} to the corresponding image I_{k-n} using

$$u_{k-n,i}^p = \pi(Kp_{k-n,i}),\tag{13}$$

where $u_{k-n,i}^p$ is the projection of $p_{k-n,i}$ in the pixel coordinate system, $\pi(\cdot)$ is the projection function.

The projected point cloud is aligned well with the image since no sensor drift happened in the previous frames. After that, the feature points $u_{k-n,g}^f$ extracted in the camera motion estimation are associated with depth projections $u_{k-n,i}^p$ using the nearest neighbor search if a valid range value is found within a small local neighborhood t_a:

$$\left\| u_{k-n,g}^f - u_{k-n,i}^p \right\|_{L_2} < t_a.\tag{14}$$

In the implementation, t_a is generally 2 pixels. And the projected point $u_{k-n,i}^p$ whose depth is closer than its neighbors are paired up with the feature point $u_{k-n,g}^f$, on
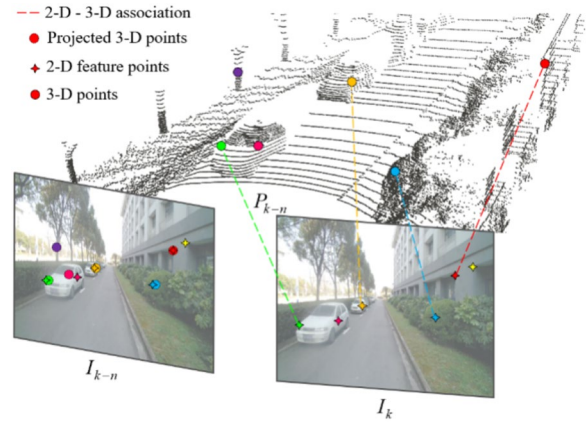
account of the farther 3-D range points are less likely to coincide with the feature points due to the occlusion and parallax; besides, the points which actually fall in the image are considered only.

Then, by the knowledge of the matched feature points between the prior image I_{k-n} and current image I_k, the 3-D-to-2-D correspondence between the point $p_{k-n,i}$ in the previous LiDAR scan and the feature point $u_{k,g}^f$ in the current image can be accordingly constructed. Then the camera motion $T_{c_{k-n}}^{c_k}$ can be estimated with the perspective-from-n-points (PnP) algorithm embedded in the RANSAC scheme to robustly discard the outliers:

$$T_{c_{k-n}}^{c_k} = \arg\min \frac{1}{2} \sum \left\| u_{k,g}^f - \pi(KT_{c_{k-n}}^{c_k} p_{k-n,i}) \right\|_2^2.\tag{15}$$

By minimizing the reprojection error of all inliers, this bundle adjustment (BA) problem is solved using Levenberg–Marquart (LM) optimization method, the Ceres Solver C++ Library is employed to perform the nonlinear optimization.

5.2 Correction Matrix Calculation

Then the estimated camera motions from LiDAR cues $T_{l_{k-n}}^{l_{k-n+1}}, n = 2, 3, ..., N$ between the adjacent scans in the window are utilized to establish further constraints,

$$T_{c_{k-n}}^{c_k} = T_{c_{k-1}}^{c_k} \prod_{n=N}^{2} T_{l_{k-n}}^{l_{k-n+1}}.\tag{16}$$

Therefore, a series of motion matrices ${}^n T_{c_{k-1}}^{c_k}$ is obtained, $n = 2, 3, ..., N$ between the current frame and the previous frame, and the final motion $\overline{T}_{c_{k-1}}^{c_k} = [\overline{R}_{c_{k-1}}^{c_k} | \overline{t}_{c_{k-1}}^{c_k}]$ is obtained by

Peng *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:50

Page 7 of 12

$$q_{c_{k-1}}^{c_k} = \frac{1}{N-1} \sum_{n=2}^{N} {}^n q_{c_{k-1}}^{c_k}, \tag{17}$$

$$\overline{R}_{c_{k-1}}^{c_k} = \overline{q}_{c_{k-1}}^{c_k} = q_{c_{k-1}}^{c_k} / ||q_{c_{k-1}}^{c_k}||, \tag{18}$$

$$\overline{t}_{c_{k-1}}^{c_k} = \frac{1}{N-1} \sum_{n=2}^{N} {}^n t_{c_{k-1}}^{c_k}, \tag{19}$$

where ${}^n q_{c_{k-1}}^{c_k}$ is the quaternion corresponding to the rotation matrix ${}^n R_{c_{k-1}}^{c_k}$. Then the prior un-drifted LiDAR scan P_{k-1} is transformed to the current position as the reference point cloud P_k^R using the motion matrix $\overline{T}_{c_{k-1}}^{c_k}$. After that, the current drifted point cloud P_k is registered with the reference scan P_k^R to find the correction matrix $T_{c_k}^R$. Finally, the extrinsic parameter $T_{l_k}^{c_k}$ between the LiDAR and camera is corrected by

$$T_{l_k}^{c_k} = T_{c_k}^R T_{l_k}^{c_k}. \tag{20}$$

6 Experiments and Discussion

In this section, to evaluate the accuracy and robustness of the proposed calibration technique, the experiment results are described in detail with a real-world dataset collected from the built autonomous driving test platform, and compare it to other state-of-the-art calibration methods.

6.1 Experiment Setup

The sensor setup is mounted on the autonomous driving platform, as shown in Figure 5, and it consists of a 40 beams LiDAR and a high-definition camera with a resolution of 1280×720. The initial extrinsic parameters between the two sensors are firstly obtained by the method in Ref. [40] for its conspicuous accuracy, and further finetuned manually by the experts.

To verify the effectiveness of the method, the test data are collected in a road environment different from that of determining the hyperparameters. In the experiment, a random offset is applied to the LiDAR scans to emulate sensor drift, this way provides us with the ground truth in the sensor extrinsic parameters.

6.2 Online Miscalibration Detection and Correction

To examine the method performance, an extensive series of experiments are performed with the collected data. Random synthetic offsets with different sizes ((0.2°–2.0°) and (0.5–3 cm)) and frequencies (1–10 Hz) are applied to all six calibration parameters to more realistically

Figure 5 The autonomous vehicle carrying the sensor setup

emulate the sensor drift under the margin of 10° in rotation and 20 cm in translation which represents quite a high size of drift for the sensors on autonomous vehicle.

The proposed approach is compared with two representative techniques including Levinson's method [27] and deep learning based method RGGNet [35] which are implemented as follows: for Levinson's method, the grid search space is expanded from 1 to 4 according to the synthetic offsets for reliable results, and the Gaussian distribution corresponding to the objective function is fitted. For RGGNet, the model and weight parameters provided in the original framework is used to conduct the experiments. Considering the huge amount of data required for the network, transfer learning technique is employed to train the annotated 4000 pairs of training data which has applied random offsets as the above setting.

Figure 6 reports the calibration results of the above methods with the same deviations which include no drift, small drift, and large drift stages. The first 50 frames are used to inspect whether the methods incorrectly classify a normal calibration, it is noted that the proposed method always discerns the correct calibration while Levinson's method makes some mistakes. Two issues combine to give these results, the first is the edge features between different modalities are not easy to align, and often lead the optimization in the wrong direction, while the proposed method uses the same modality to estimate the sensor motions for miscalibration detection which eliminates the systemic errors induced by the feature alignment; the second is that Levinson' method relies on a statistical test and does not consider the changes on each axis, its inherent probability-based detecting mechanism will inevitably lead to the wrong classification. As for RGGNet, it runs calibration network every frame for lacking miscalibration detection mechanism, and the
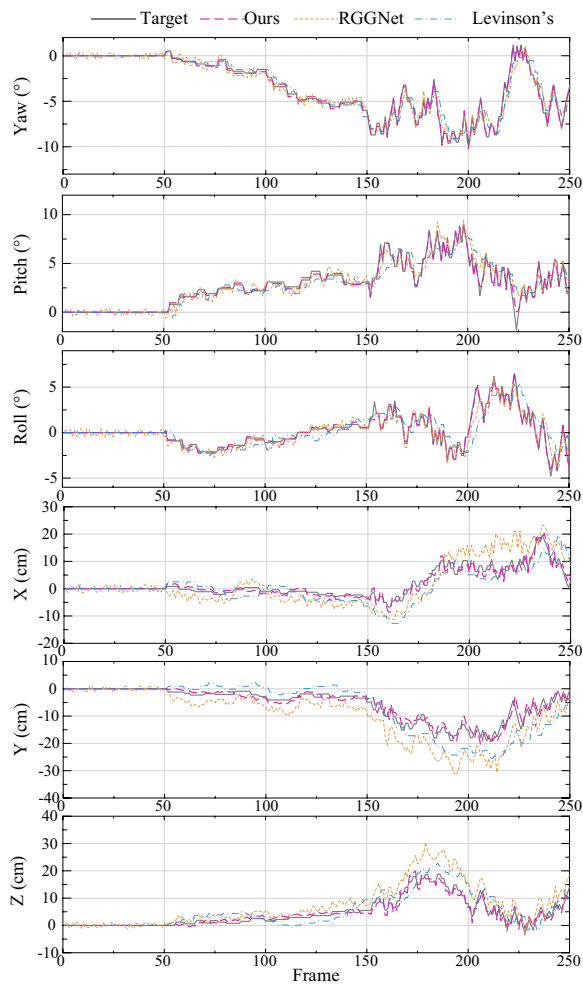
Figure 6 Evaluation results of the online drift detection and correction

It can be seen that yaw is the most precise on the whole among the three rotation directions, as the high horizontal angular resolution of LiDAR makes no matter the point cloud registration, the edge extraction, or the semantic extraction work in a more natural way. This also makes the proposed method perform well in two translation directions X and Z on the horizontal plane. On the other side, the relatively low vertical resolution degrades the calibration accuracy of the above methods in pitch and Y direction, for example, the proposed method fails to calibrate the sharp drift in pitch around frame 224, but it still quickly maintains the continuous and accurate correction of the following drift frames thanks to multi-frame based correction mechanism. For roll, X, and Z direction, drift induces smaller deviations in the point cloud projection map according to the projection principle, which explains the difficulty of accurate calibration of the two compared methods since their inputs are both the point cloud projection. The proposed method performs well and is not affected at all for that it can accurately estimate the changes in these directions based on point cloud registration.

To more intuitively show the performance of the above methods in miscalibration detection and correction, the experimental outcomes are numerically summarized in Table 1. Precision and recall denote the percentage of correctly classified frames in all detected frames and all drifted frames, respectively. RGGNet is not considered for lacking a classification mechanism. Compared with Levinson's method, the proposed approach gives a much more accurate classification, the precision and recall are boosted from 84.97% and 88.02% to 94.67% and 95.8%, respectively. The remaining metrics are the mean absolute errors between the estimated calibration parameters and the ground truth, the proposed method outshines the other two counterparts in all directions. It is worth noting that Levinson's method is often slightly biased to sensor overlap and this may cause local optima, moreover, the limited search space is unable to cope with large drift. RGGNet calibrates with only the current frame, and its limited generalization ability does not guarantee an accurate calibration without the benefits of historical frames.

limited amount of training data does impede the generalization capability of the network, resulting in the oscillation about the right value.

The next 200 frames are leveraged to evaluate the calibration correcting performance under random small drift and large drift in all six dimensions at once. It is evident that the proposed approach outperforms Levinson's method and RGGNet in both rotation and translation.

Table 1 Performance comparison of miscalibration detection and correction

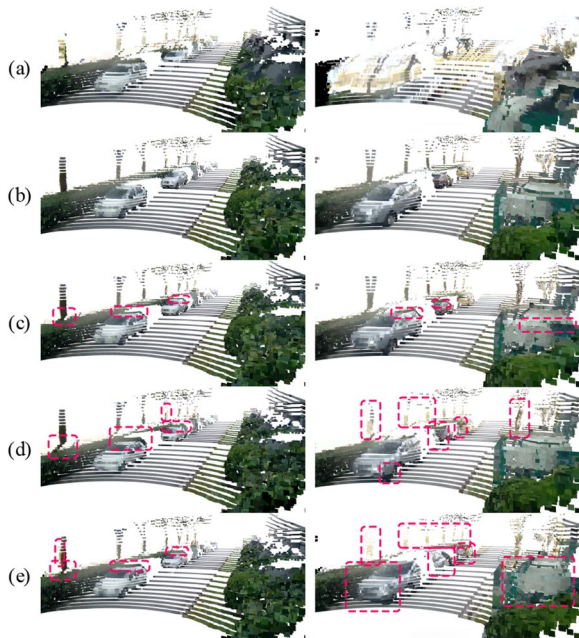| Methods	Precision (%)	Recall (%)	Yaw (°)	Pitch (°)	Roll (°)	X (cm)	Y (cm)	Z (cm)
Lev	84.97	88.02	0.495	0.652	0.742	7.20	7.69	6.47
RGGNet	-	-	0.517	0.541	0.428	8.50	9.87	8.58
Ours	**94.67**	**95.81**	**0.157**	**0.189**	**0.172**	**2.45**	**3.62**	**2.46**

Figure 7 Textured point clouds using miscalibration correction parameters of each method: (**a**) Drifted, (**b**) Ground truth, (**c**) The proposed method, (**d**) Levinson's method, and (**e**) RGGNet method (The left and right columns show the typical correction results of small drifts and large drifts, respectively, the red rectangles indicate the misalignment)

Figure 8 Determination accuracy of calibration state under different number of reference frames

A qualitative example of miscalibration correction is depicted in Figure 7, the camera images are projected onto the LiDAR scans giving each of the points an associated color. The typical miscalibration correcting results under small drift and large drift conditions are shown in the left and right column, respectively, the red rectangle areas indicate the misalignment. There exists some slight misalignment in the results of the proposed method when the depth changes dramatically, such as the distant green belt projected onto the vehicle top. However, from the global perspective, it can be discerned that the proposed method delivers radically higher alignment accuracy and robustness, compared to Levinson's method and RGGNet method.

6.3 Offline Miscalibration Detection and Correction
Besides the online calibration scenario, the proposed method can be extended to offline miscalibration detection and correction task, for example, when an autonomous vehicle returns to the parking spot after a period of operation, the sensors may have drifted and therefore an in-field calibration is needed.

To address this problem, some adjustments are made to the proposed method. First, several well-calibrated scan-image pairs with different poses at the parking spot are previou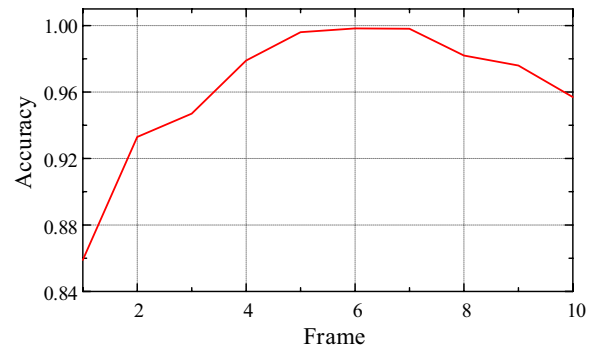sly sampled as the reference data, their function is equivalent to that of the previous frames in Section 4. Then, only one current scan-image pair from the spot is required to determine the calibration state according to Eq. (12). Finally, according to Section 5, the transformation matrices from the reference frame to the current frame are estimated and the final correction matrix is further obtained. Compared with the online task, the reference data pairs of the offline task have a stronger spatial association, which is conducive to the determination of calibration state, in addition, more data may bring a higher probability of outliers. Hence, the number of reference data pairs should be reconsidered. The determination accuracy under different number of reference frames is tested, the experiment is repeated 100 times for each number with the mean reported (see Figure 8). It can be seen that the best accuracy 0.9983 reaches under 6 frames.

To assess the accuracy and robustness of the proposed method in the offline scenario, edge-based method GMM [21] and intensity based method IM [29] are employed to compare with the proposed one, these two in-field calibration approaches are implemented as below: for GMM, images from monocular camera are used instead of omnidirectional camera, and replace the spherical camera model with pinhole model, the thresholds for image and point cloud edge scores are 0.15 and 0.12, respectively. For IM, 20 scan-image pairs at multiple nearby locations of the parking spot, as in the original manuscript, are used to generate a better estimate of the joint and marginal probability distributions.

In the experiments, the LiDAR pose is manually adjusted within the order of 10° for rotation parameters and 20 cm for translation parameters to emulate the sensor drift, then use the target-based calibration method [40] to find extrinsic parameters with minimal projection error, the ground truth of each LiDAR pose

Peng *et al. Chinese Journal of Mechanical Engineering* (2024) 37:50

Page 10 of 12

is subsequently obtained by manually finetuning these parameters. 100 calibration trials with random drift are performed, the results are presented in Figure 9 and Table 2.

It can be discerned that the proposed method converges to more consistent results compared with GMM and MI, except for the roll direction. Nevertheless, the proposed method completely outshines its counterparts in mean absolute error metric, as summarized in Table 2, indicating the accuracy and robustness of the proposed method. Similar to Levinson's method, GMM has difficulty in estimating the vertical drift (*Y* direction) owing to the sparse angle resolution of LiDAR and its limited field of view. In addition, a reasonably tuning of weight, displacement, and standard deviation for GMM can be invariably heuristic and experiential. MI runs on the establishment of correspondence between the surface reflectivity from LiDAR and the gray-scale intensity reported by camera, which is looser than the edge constraints. Besides, the shadows of occluding objects in the image corrupt the correlation of intensity and reflectivity, producing a weak input for MI algorithm. These induce

the greater fluctuation and more undesired outliers of the MI outcomes. As listed in Table 2, the offline results of the proposed method show increased performance compared to the online results. This is mainly ascribed to the stronger spatial association of the scan-image pairs, and there are no asynchronous time stamps and motion distortion problems engendered by data acquisitions on a moving vehicle.

To give a qualitative comparison, the point cloud textured by each representative calibration parameter of these methods is illustrated in Figure 10. In this instance, the result of the proposed method is visually comparable with the textured point cloud generated from the ground truth, while the other two methods show slightly larger uncertainty in the translation, for example, the wall is obviously offset to the right. These findings prove the validity and progressiveness of the proposed proposed method regarding its accuracy and robustness.

7 Conclusions

This paper presents an automatic miscalibration detection and correction framework for LiDAR and camera to confront the calibration error induced by sensor drift during vehicle operation.

Its effectiveness is first demonstrated in the online miscalibration and correction experiments, the quantitative and qualitative comparison results with two representative techniques, Levinson's method and RGGNet, verify the accuracy and robustness of the proposed scheme with varying levels of random drift.
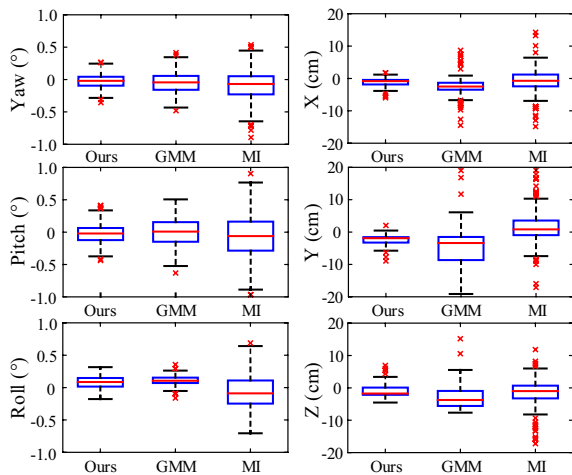

Figure 9 Box plot of calibration error (The boxes indicate the 25th and 75th percentiles, and the red lines in the boxes represent median values. The whiskers represent the most distinct values within 1.5 times box height from the boxes. Red "×" symbols indicate errors out of that range)
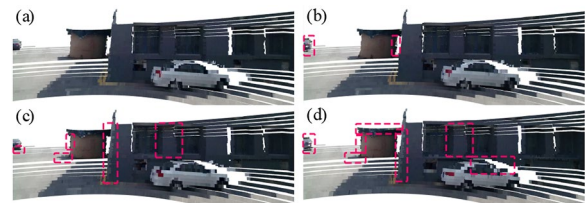

Figure 10 Qualitative comparison of the offline calibration task: (**a**) Ground truth, (**b**) The proposed method, (**c**) GMM, and (**d**) MI (The red rectangles indicate the misalignment)

Table 2 Mean absolute error of offline calibration

Methods	Yaw (°)	Pitch (°)	Roll (°)	X (cm)	Y (cm)	Z (cm)
GMM	0.132	0.168	0.114	3.48	5.44	3.47
IM	0.201	0.277	0.219	3.40	6.24	5.65
Ours	**0.090**	**0.121**	**0.102**	**2.31**	**2.62**	**2.09**

The proposed method is also extended to the offline task, a comparative study with two existing in-field calibration approaches, GMM and MI, is also performed. The results further validate the consistency and precision of the proposed method.

A direction to improve this work is to boost the feature extraction speed without sacrificing accuracy for the registration algorithms of point cloud and image to realize the real-time operation of the proposed method.

Availability of data and materials
The datasets supporting the conclusions of this article are included within the article.

Declarations

Competing interests
The authors declare no competing financial interests.

References
[1]   D Cao, X Wang, L Li, et al. Future directions of intelligent vehicles: Potentials, possibilities, and perspectives. *IEEE Transactions on Intelligent Vehicles*, 2022, 7(1): 7-10.
[2]   W Chu, Q Wuniri, X Du, et al. Cloud control system architectures, technologies and applications on intelligent and connected vehicles: A review. *Chinese Journal of Mechanical Engineering*, 2021, 34: 139.
[3]   Y Luo, W Chu and D Cao. Key technologies in connected autonomous electrified vehicles. *Chinese Journal of Mechanical Engineering*, 2021, 34: 144.
[4]   Q Meng, H Guo, X Zhao, et al. Loop-closure detection with a multiresolution point cloud histogram mode in lidar odometry and mapping for intelligent vehicles. *IEEE/ASME Transactions on Mechatronics*, 2021, 26(3): 1307-1317.
[5]   Y Wang, J Hu, F Wang, et al. Tire road friction coefficient estimation: Review and research perspectives. *Chinese Journal of Mechanical Engineering*, 2022, 35: 6.
[6]   R Chavez-Garcia, O Aycard, Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 17(2): 525-534.
[7]   S Li, K Shu, C Chen, et al. Planning and decision-making for connected autonomous vehicles at road intersections: A review. *Chinese Journal of Mechanical Engineering*, 2021, 34: 133.
[8]   P Peng, K Geng, G Yin, et al. Adaptive multi-modal fusion instance segmentation for caevs in complex conditions: dataset, framework and verifications. *Chinese Journal of Mechanical Engineering*, 2021, 34: 81.
[9]   Y Li, X Lu, J Wang, et al. Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning. *IEEE Transactions on Intelligent Vehicles*, 2020, 5(3): 461-474.
[10]   H Dong, W Zhuang, G Yin, et al. Energy-optimal braking control using a double-layer scheme for trajectory planning and tracking of connected electric vehicles. *Chinese Journal of Mechanical Engineering*, 2021, 34: 83.
[11]   J Domhof, J Kooij, D Gavrila, et al. A joint extrinsic calibration tool for radar, camera and lidar. *IEEE Transactions on Intelligent Vehicles*, 2021, 6(3): 571-582.
[12]   Y Zhang, H Zhang, G Wang, et al. Bundle adjustment for monocular visual odometry based on detections of traffic signs. *IEEE Transactions on Vehicular Technology*, 2019, 69(1): 151-162.
[13]   B Jia, J Chen, K Zhang, et al. Sequential monocular road detection by fusing appearance and geometric information. *IEEE/ASME Transactions on Mechatronics*, 2019, 24(2): 633-643.
[14]   H Song, W Choi, H Kim. Robust vision-based relative-localization approach using an RGB-depth camera and LiDAR sensor fusion. *IEEE Transactions on Industrial Electronics*, 2016, 63(6): 3725-3736.
[15]   S Xie, D Yang, K Jiang, et al. Pixels and 3-D points alignment method for the fusion of camera and LiDAR data. *IEEE Transactions on Instrumentation and Measurement*, 2018, 68(10): 3661-3676.
[16]   Z Hu, Y Li, N Li, et al. Extrinsic calibration of 2-D laser rangefinder and camera from single shot based on minimal solution. *IEEE Transactions on Instrumentation and Measurement*, 2016, 65(4): 915-929.
[17]   X Liu, C Yuan, F Zhang, Targetless extrinsic calibration of multiple small FoV LiDARs and cameras using adaptive voxelization. *IEEE Transactions on Instrumentation and Measurement*, 2022, 71: 1-12.
[18]   H Yu, W Zhen, W Yang, et al. Line-based 2-D–3-D registration and camera localization in structured environments. *IEEE Transactions on Instrumentation and Measurement*, 2020, 69(11): 8962-8972.
[19]   M Brown, D Windridge, J Guillemaut. A family of globally optimal branch-and-bound algorithms for 2D–3D correspondence-free registration. *Pattern Recognition*, 2019, 93: 36-54.
[20]   D Campbell, L Petersson, L Kneip, et al. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 42(2): 328-342.
[21]   J Kang, N Doh. Automatic targetless camera–lidar calibration by aligning edge with gaussian mixture model. *Journal of Field Robotics*, 2020, 37(1): 158-179.
[22]   Z Taylor, J Nieto, D Johnson. Multi-modal sensor calibration using a gradient orientation measure. *Journal of Field Robotics*, 2015, 32(5): 675-695.
[23]   C Yuan, X Liu, X Hong, et al. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters*, 2021, 6(4): 7517-7524.
[24]   S Schneider, T Luettel, H Wuensche, Odometry-based online extrinsic sensor calibration. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013: 1287-1292.
[25]   Z. Taylor, J Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 2016, 32(5): 1215-1229.
[26]   B Corte, H Andreasson, T Stoyanov, et al. Unified motion-based calibration of mobile multi-sensor platforms with time delay estimation. *IEEE Robotics and Automation Letters*, 2019, 4(2): 902-909.
[27]   J Levinson, S Thrun. Automatic Online Calibration of Cameras and Lasers. *Robotics: science and systems*. Berlin, Germany, 2013: 24–28.
[28]   H Xu, G Lan, S Wu, et al. Online intelligent calibration of cameras and lidars for autonomous driving systems. *IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019: 3913-3920.
[29]   G Pandey, J McBride, S Savarese, et al. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. *Journal of Field Robotics*, 2015, 32(5): 696-722.
[30]   R Ishikawa, T Oishi, K Ikeuchi. Lidar and camera calibration using motions estimated by sensor fusion odometry. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018: 7342-7349.
[31]   M Horn, T Wodtko, M Buchholz, et al. Online extrinsic calibration based on per-sensor ego-motion using dual quaternions. *IEEE Robotics and Automation Letters*, 2021, 6(2): 982-989.
[32]   N Schneider, F Piewak, C Stiller, et al. RegNet: Multimodal sensor registration using deep neural networks. *IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, USA, June 11-14, 2017: 1803–1810.

[33] G Iyer, R Ram, J Murthy, et al. CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018: 1110–1117.

[34] S Mharolkar, J Zhang, G Peng, et al. RGBDTCalibNet: End-to-end online extrinsic calibration between a 3D LiDAR, an RGB camera and a thermal camera. *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China, 2022: 3577-3582.

[35] K Yuan, Z Guo, Z Wang. RGGNet: Tolerance aware LiDAR-camera online calibration with geometric deep learning and generative model. *IEEE Robotics and Automation Letters*, 2020, 5(4): 6956-6963.

[36] C Yuan, W Xu, X Liu, et al. Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry. *IEEE Robotics and Automation Letters*, 2022, 7(3): 8518-8525.

[37] T Hassner, S Filosof, V Mayzels, et al. SIFTing through scales. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 39(7): 1431-1443.

[38] D Barath, L Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 2018, 27(11): 5328-5337.

[39] W Zong, M Li, Y Zhou, et al. A fast and accurate planar-feature-based global scan registration method. *IEEE Sensors Journal*, 2019, 19(24): 12333-12345.

[40] A Geiger, F Moosmann, Ö Car, et al. Automatic camera and range sensor calibration using a single shot. *IEEE International Conference on Robotics and Automation*, St Paul, USA, 2012: 3936-3943.

Pai Peng born in 1993, is currently a Ph.D. candidate at *School of Mechanical Engineering, Southeast University, China*. He received his M.S. degree from *Nanjing University of Science and Technology, China*, in 2019. His research interests include autonomous vehicles and environment perception.

Dawei Pi born in 1983, is currently a professor and a Ph.D. candidate supervisor at *School of Mechanical Engineering, Nanjing University of Science and Technology, China*. He received the Ph.D. degree in vehicle engineering from *Southeast University, China*, in 2010. From 2016 to 2017, he was a visiting scholar at *Department of Mechanical and Aerospace Engineering, Ohio State University, USA*. His main research interests include vehicle system dynamics, intelligent control for electric vehicles and intelligent fault-tolerant control of electro-mechanical systems.

Guodong Yin born in 1976, is currently a professor and a Ph.D. candidate supervisor at *School of Mechanical Engineering, Southeast University, China*. He received the Ph.D. degree in vehicle engineering from *Southeast University, China*, in 2007. From 2011 to 2012, he was a visiting scholar at *Department of Mechanical and Aerospace Engineering, Ohio State University, USA*. He was the recipient of the National Science Fund for Distinguished Young Scholars. His current research interests include vehicle dynamics and control, connected and automated vehicle.

Yan Wang born in 1992, received the Ph.D. degree in mechanical engineering from *Southeast University, China*, in 2022. He is currently a research fellow with the *School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore*. His research interests include vehicle state estimation and automotive active safety control.

Liwei Xu born in 1986, is currently a research associate at *School of Mechanical Engineering, Southeast University, China*. He received the Ph.D. degree on mechanical engineering from *Southeast University, China*, in 2019. His current research interests include vehicle dynamics and control, autonomous vehicles, and connected vehicles.

Jiwei Feng born in 1992, is currently a Ph.D. candidate at *School of Mechanical Engineering, Southeast University, China*. His current research interests include dynamic coordination control of four-wheel steering electric vehicle, and man-machine collaborative control.