2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Line-based Automatic Extrinsic Calibration of LiDAR and Camera

Xinyu Zhang[1], Shifan Zhu[1,*], Shichun Guo[1], Jun Li[1], and Huaping Liu[2]

***Abstract*— Reliable real-time extrinsic parameters of 3D Light Detection and Ranging (LiDAR) and camera are a key component of multi-modal perception systems. However, extrinsic transformation may drift gradually during operation, which can result in decreased accuracy of perception system. To solve this problem, we propose a line-based method that enables automatic online extrinsic calibration of LiDAR and camera in real-world scenes. Herein, the line feature is selected to constrain the extrinsic parameters for its ubiquity. Initially, the line features are extracted and filtered from point clouds and images. Afterwards, an adaptive optimization is utilized to provide accurate extrinsic parameters. We demonstrate that line features are robust geometric features that can be extracted from point clouds and images, thus contributing to the extrinsic calibration. To demonstrate the benefits of this method, we evaluate it on KITTI benchmark with ground truth value. The experiments verify the accuracy of the calibration approach. In online experiments on hundreds of frames, our approach automatically corrects miscalibration errors and achieves an accuracy of 0.2 degrees, which verifies its applicability in various scenarios. This work can provide basis for perception systems and further improve the performance of other algorithms that utilize these sensors.**

I. INTRODUCTION

Multiple LiDARs and cameras are widely applied on autonomous vehicles and robots in many scenarios, such as autonomous driving [1], [2], object classification [3], [4], segmentation [5], and SLAM [6], [7]. LiDARs can provide accurate three-dimensional geometric information but with sparse points. On the contrary, cameras are capable of offering a rich representation of the environment but with less accurate distance information. The extrinsic transformation matrix of these sensors can combine these two complementary sensors and maximize the effects. Therefore, accurate extrinsic parameters between them is crucial to provide accurate information for perception systems.

Traditional manual calibration methods require specially designed objects, such as checkboards [8] or manually selected points [9], which lead to a cumbersome calibration procedure. Furthermore, long-time operation and different loads can result in slight drifting and deviations to extrinsic parameters. Therefore, an automatic online correction is needed to adjust to this unforeseen sensor movement.

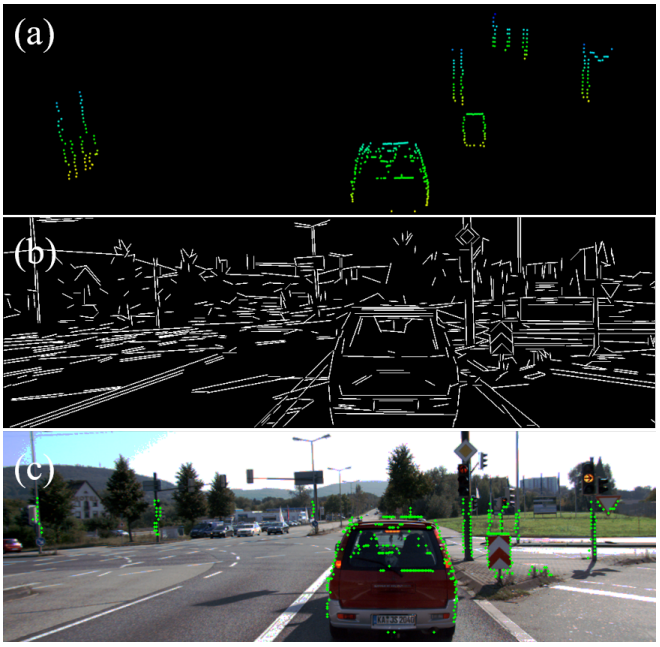

Fig. 1. Line features are extracted from a point cloud (a) and an image (b) of the same scene. By computing the correspondence between these 2D and 3D line features, the projected point cloud line features (c) are well aligned with the image line features.

Current automatic calibration works utilize mutual information [10] or artificially designed targets [11] to calibrate the extrinsic parameters. The use of specific targets and intensity information limits the calibration process to laboratory settings and specific sensors. Some other feature-based calibration method [12] utilizes edge features to compute the extrinsic parameters. However, these features are not well corresponded to each other under some scenarios. In this paper, we aim to release this limitation and simplify the process required to calibrate the extrinsic parameters of LiDAR and camera. For this purpose, we utilize the robust and wide-spread line features to compute the transformation matrix between these sensors automatically with no hand-labeling. As input, our method requires a single image and previous several point clouds as well as the initial extrinsic parameters. The line features used in this work are generally spread in outdoor settings, such as trees, street lamps, cars, etc. By applying these line features, the proposed method can automatically adjust according to drifting and changes

This work was supported by the National High Technology Research and Development Program of China under Grant No. 2018YFE0204300, and the Beijing Science and Technology Plan Project (Z191100007419008), and the Guoqiang Research Institute Project (2019GQG1010), and the National Natural Science Foundation of China under Grant No. U1964203.

[1]X. Zhang, S. Zhu, S. Guo, J. Li are with the State Key Laboratory of Automotive Safety and Energy, and the School of Vehicle and Mobility, Tsinghua University, Beijing, 100084 China. (e-mail: xyzhang@tsinghua.edu.cn; shifzhu@gmail.com; shichunguo@gmail.com; lijun19580326@126.com).

[2]H. Liu is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 China. (e-mail: hpliu@mail.tsinghua.edu.cn).

*Author to whom correspondence should be addressed. (e-mail: shifzhu@gmail.com).

of extrinsic transformation matrix. The main contributions of this paper are as follows:

Firstly, we introduce a novel extrinsic calibration method that can automatically estimate the six degree of freedom (6-DOF) parameters online. The proposed method utilizes generic line features to reduce the drifted errors between sensors without requiring manually selected points and special targets, therefore it is able to be applied to any given scenarios.

Secondly, we present a point cloud line extraction approach that can obtain line features from point clouds, where a point cloud processing approach is utilized to filter out noise data and extract line features accurately.

Thirdly, we introduce an adaptive optimization method and a result confidence estimation method to enable optimization towards the correct direction and computing the calibration results efficiently.

The rest of this article is organized as follows: Section II summarizes related works. Section III introduces the methods proposed in this article. Section IV evaluates the accuracy of the proposed methods based on the KITTI dataset. Section V summarizes our research and prospects for future work.

II. RELATED WORKS

Based on whether the calibration process needs targets, the extrinsic calibration methods of LiDAR and camera can be broadly divided into two categories: target-based methods and target-less methods.

A. Target-based methods

The calibration problem between LiDAR and camera was firstly solved by Q. Zhang et al. [13], they utilized a checkboard to calibrate the extrinsic parameters of a two-dimensional LiDAR and a camera. This approach minimizes the re-projection error by using plane constraints and non-linear optimization method. R. Unnikrishnan et al. [14] published a MATLAB tool kit for calibrating LiDAR and camera through manually selected points. D. Scaramuzza et al. [15] described a calibration method for drone by selecting several corresponding points from two sensors. S. Kato et al. [9] proposed an open approach to calibrate LiDAR and camera through selected points. The main limitation of above approaches is that they need to manually assign corresponding points between point clouds and images, which is laborious especially when calibrating frequently. It can also result in man-made errors since the selected points may not correspond to each other precisely.

To overcome above limitations, some researchers developed several ways to calibrate the extrinsic parameters more intelligently. They employed special artificial calibration targets, such as triangular boards [16], polygonal boards [17], circular boards [18], [19] or spheres [20], to replace the process of manually selecting corresponding points. A. Kassir et al. [21] developed a corner detection method so that they can extract the chessboard to align with the point cloud. E.-s. Kim et al. [22] also utilized multiple plane features to calibrate the LiDAR and camera. Without having to choose corresponding points, these methods still share the limitation of specially designed targets and man-made target errors. Therefore, these methods are not feasible in real-world settings where there are no artificial targets placed in the field-of-view of the sensors.

B. Target-less methods

Some methodologies have been developed to solve the limitations of target-based methods by utilizing real-world features. [23], [24], [25] reported targetless extrinsic calibration methods based on maximization of mutual information obtained between surface intensities measured by two sensors. These methods do release the limitation of requiring specific targets. However, the essential intensity information may be not available and the variation of intensity can result in fluctuation of the transform matrix between LiDAR and camera.

Some researchers developed motion-based methods to compute extrinsic parameters of sensors. In [26], [27], [28], they presented automatic targetless calibration methods based on hand-eye calibration, which only require three-dimensional point clouds and camera images to compute motion information. Besides that, reflectance information and overlapping field of view are unnecessary. However, these methods can only provide a rough calibration result since the motion estimation errors can affect the accuracy of calibration parameters.

P. Moghadam et al. [29] introduced a line-based calibration method, where planar surfaces are estimated to extract plane intersection lines. However, their approach is limited to environments containing plane intersection features. J. Levinson et al. [30] introduced the approach where the alignment of the edge correspondences between LiDAR points and image pixels can be employed for calibration. To that end, they assumed that LiDAR points with large depth discontinuity will project onto edge features more often than other features. They utilized discontinuities to extract edge points and formulated a cost function by summing up each corresponding pixel. Finally, the cost was optimized using the grid searching algorithm. However, as demonstrated in [31], this method may result in biases in scenes with many clustered edge features, such as the scenes with many trees. And they considered each beam independently so they only got vertical edge features, which cannot constrain all direction errors. Inspired by their work, we utilized neat line features and an adaptive optimization method to search for the optimal direction and obtain the extrinsic parameters, which enable us to find out a more accurate result with higher efficiency.

III. METHODOLOGY

In this paper, line features are selected to accurately calibrate the extrinsic parameters of a LiDAR-camera system. The main benefit of these features is that they are ubiquitous both in outdoor and indoor scenarios. Besides that, they have good correspondence in point clouds and images. Figure 2 offers an overview of the proposed approach. The proposed method utilizes three steps. First, a series of preprocessing

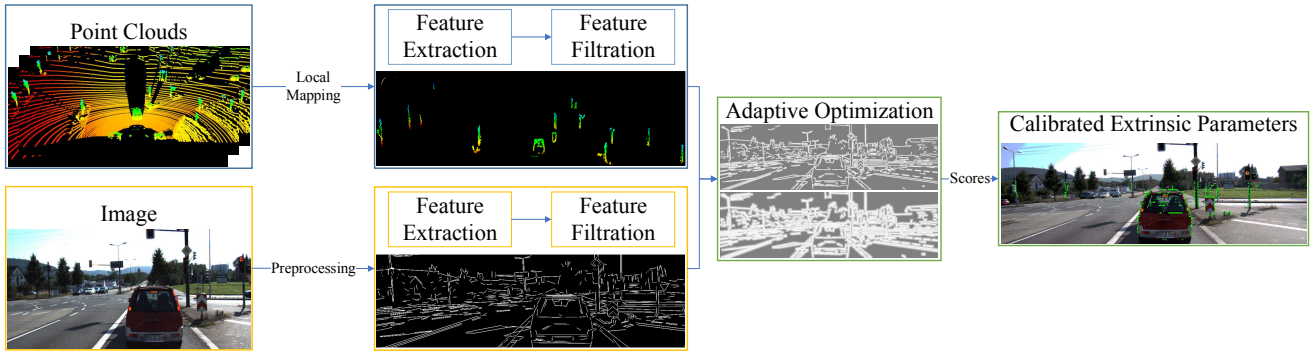

Fig. 2. The architecture of the proposed calibration system.

methods are employed to images and point clouds for further feature extraction. Then, the line features are extracted from images and point clouds and refined by applying feature filtration. Finally, the point cloud line features are projected onto pixel frames by adding small perturbation to initial extrinsic parameters, during which the score of each perturbation will be computed and optimized. Details are presented in the following sections.

A. Problems Formulation and Assumption

The extrinsic calibration problem of the LiDAR and camera lies in determining the correct transform matrix between them. In this paper, we define the problem as finding the rotation angle vector $\theta = (\theta_x, \theta_y, \theta_z)$ and translation vector $t = (t_x, t_y, t_z)$. For the point clouds and images, we notate the point cloud as P_t and image as I_{ij}^t, representing point i and pixel ij value at frame t. We optimize the 6-DOF parameters by projecting each point onto pixel frame and computing the score of current parameters by adding grayscale value of each pixel. The cost score can be computed by projecting LiDAR points onto images and the objective function can be defined as:

$$S_t = \sum_{t=n-w}^{n} I_{ij}^t \left[\alpha \sum_{p_t}^{p_t \in F_h^t} T_t p_t + (1-\alpha) \sum_{p_t}^{p_t \in F_v^t} T_t p_t \right] / w \tag{1}$$

where each LiDAR point p_t iterates over horizontal features F_h and vertical features F_v respectively. The coefficient α assigns differents weight to horizontal and vertical line features. In this paper, α is assigned as 0.65 to enhance the constraint of horizontal errors. In addition, w is the size of the sliding window. Score of frame t is computed by considering the previous w frames.

The basic assumption of this work is that the line features from point clouds are highly responsive to line features from images when the calibration parameters are correct. Besides that, we also assume that the intrinsic parameters of camera and LiDAR are already calibrated and the LiDAR data and camera images are captured simultaneously.

B. Image processing

In image processing, the RGB images are initially converted into grayscale images, after which the line features are extracted by a line detection algorithm [32]. Afterwards, a distance transform model is applied to these gray images. An example of an original image, line features, and edge features is shown in Figure 3. The white edges in Figure 3 (b) and the white lines in Figure 3 (c) represent clustered edge features and line features respectively. As is shown in Figure 3 (b), the clustered edge features are more disordered after applying the distance transform model. On the contrary, line features in Figure 3 (c) are more well-organized, generating smaller grayscale changes. It can allow for a larger searching step size, thus preventing the optimization process from getting into a local solution.

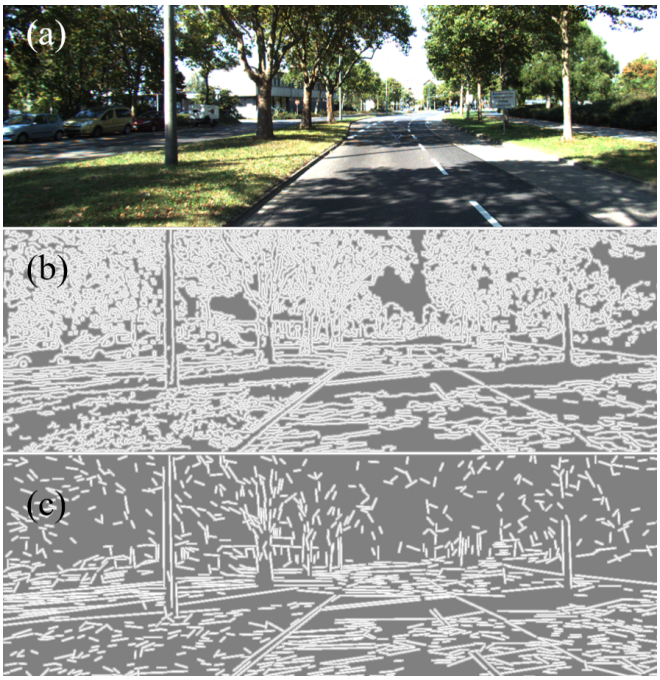

Fig. 3. A comparison of edge features and line features. (a) shows the original scenario. (b) shows the edge features and (c) shows the line features. The white pixels represent extracted features and the grayscale changes represent the distance to edge or line features. The whiter the pixels are, the closer to the center of these line features.

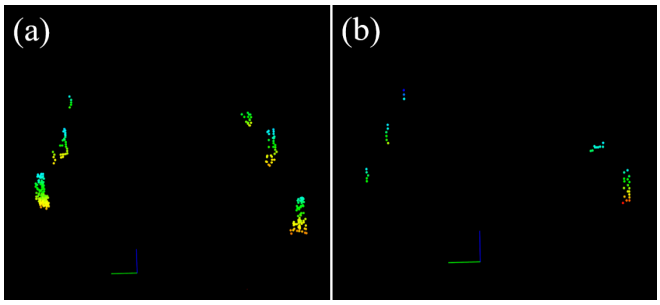

Fig. 4. A comparison of line features from a three-in-one point cloud (a) and a single point cloud (b).

C. LiDAR processing

In the LiDAR processing, the principle is to utilize distance discontinuity to get more boundary line features. To achieve this goal, a local mapping method is applied to combine three frames of point cloud into one, which can present more points in one frame. Specifically, a Normal Distribution Transform (NDT) method is utilized to compute the transformation matrix between current and previous two frames. A comparison between boundary line points extracted in a single frame and a three-in-one frame is shown in Figure 4. Figure 4 (a) shows a more intensive point cloud by transforming three frames point cloud $P_{t-2:t}$ to one frame P_t, which can reveal more points compared with the other one in Figure 4 (b). This can improve the extraction performance especially when a low-beam LiDAR is applied.

Afterwards, the denser point cloud is transformed into image form, with each pixel storing the distance information of corresponding LiDAR point. By comparing the distance between current point and adjacent points, more accurate line features can be extracted by eliminating outliers that are too far away from neighbors. It should be noticed that different from [30] which considered each beam independently, we take the distance information between multiple beams into consideration. It allows the proposed method to extract horizontal features, thus minimizing both horizontal and vertical errors using the line features. Horizontal line features F^h and vertical line F^v features are stored in two different point clouds respectively. In this setting, the plane intersection lines, a rarely appeared feature, are neglected, which is beneficial to enhance the computation efficiency.

D. Features Filtration

The extracted line features from point clouds can be disordered after the previous section. Therefore, two filtration methods are employed to eliminate outliers. Since the point clouds are already transformed into image form, a convolution kernel is designed to filter out points that are far away from all eight adjacent points beyond a certain threshold. The line features before and after filtration can be seen in Figure 5. This filtration method can remove all the outliers as well as the points that correspond to the ground. As a result, remained features can be identified as line features.

After the first filtration, a point cloud cluster algorithm is applied to remove line features that have few adjacent points. The above two filtration steps can provide more well organized point cloud line features, which guarantee a better optimization result in the subsequent steps.

A more organized line features from images can be obtained after filtering out line features that are shorter than 8 pixels.

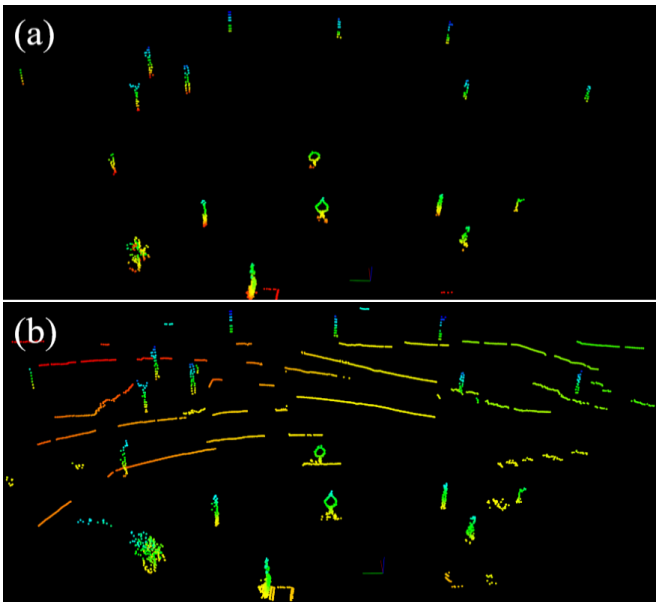

Fig. 5. A comparison of line features after (a) and before (b) feature filtration.

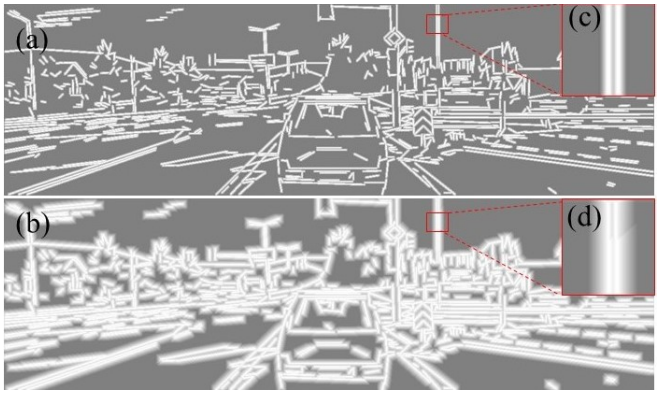

Fig. 6. Two different grayscale changes. (a) shows a larger grayscale change utlized in precise searching. (b) shows a smaller grayscale change utilized in rough searching. Zoomed-in view (c) and (d) demonstrate the comparison of different grayscale changes more clearly.

E. Adaptive Optimization

In the optimization process, the proposed approach takes both computation accuracy and efficiency into consideration. Before optimization, the proposed approach project extracted LiDAR line features onto the image, and the proportion of the LiDAR points that projected onto the gray area will be computed.

For computation accuracy, as is shown in Figure 6, two searching steps are adopted to find the solution accurately.

First of all, to prevent searching from trapping in a local solution, a rough searching with wider image lines, smaller grayscale changes, and relatively larger step size is employed, enabling to discover the areas that may contain the best solution quickly. Afterwards, thinner image line features with larger grayscale changes, along with a smaller step size, are applied for a more precise calibration result. The switch between these two-step sizes and grayscale changes will happen when the proportion of the LiDAR points projected to the gray area exceeds a certain threshold.

For computation efficiency, an adaptive optimization method is proposed to enable the optimization towards the correct direction. In [30], they did calculations for 729 different values to compute the function score, which is an inefficient way as some steps are redundant. In this paper, a searching method is applied to optimize the cost function. It will compare the current score with adjacent 728 scores. In this process, if the searching program finds parameters that have a higher score, it will stop current searching process and begin a new searching process at the position providing a higher score. Besides that, this searching process will stop when reaching the set iteration count or finding the best score, thus being able to increase the computation efficiency. In addition, a sliding window is used to set the frames that the optimization progress should consider. In this paper, three frames are utilized to prevent the optimization searching from a wrong direction or falling into a local optimal solution. Therefore, the final optimized extrinsic parameters should exceed other parameters in all frames in the sliding window.

In conclusion, the two settings work together to get a robust and precise calibration result. The termination strategy enables a faster calibration. This process can be seen at Algorithm 1.

Algorithm 1 Optimization process

Require: Image line features I_t, Horizontal F_h^t and vertical F_v^t line features at frame t, Initial extrainsic matrix T_t, last frame gray rate $gray\_rate$

Ensure: Calibrated extrainsic matrix;

Initialization: $score$, $max_{score} \leftarrow 0$.
if $gray\_rate > \gamma$ **then**
    $step\_size = \alpha_1$
else
    $step\_size = \alpha_2$
end if
Add disturbance based on step_size
for each $LiDAR\ point$ in F_h^t **do**
    $gray\_value = weight * T_t * p_t$
    $score+ = gray\_value$
end for
for each $LiDAR\ point$ in F_v^t **do**
    $gray\_value = (1 - weight) * T_t * p_t$
    $score+ = gray\_value$
end for
if $score > max\_score$ **then**
    $max\_score = score$
    Update current_parameters
end if
$gray\_rate = score/255/points\_num$
return current_parameters

IV. EXPERIMENT RESULTS

In order to validate the proposed approach, experimental tests were conducted on KITTI dataset [33], from which we used a Velodyne HDL-64E LiDAR and a high-resolution color camera. The scanning frequency of the LiDAR is 10 Hz. The data used in the experiment is synced and rectified. The ground truth extrinsic parameters can be obtained from calibration files.

We carried out two experiments on different scenarios in the KITTI dataset. Experiment I gave the result in Figure 7 (a) and (b), and experiment II gave the result in Figure 7 (c) and (d). In these two experiments, we initially added a one-degree rotation bias on X, Y, and Z axes and a 0.05-meter transformation bias to the ground truth parameters. Then, a 0.5-degree rotation bias was added every 10 frames. It should be pointed out that whether the one-degree rotation bias is positive or negative is randomized. During the experiment, we compared the calibration error with the ground truth. In addition, we tested the ability to detect miscalibration and the speed of correcting bias.

To evaluate the accuracy of proposed approach, we computed the absolute value of deviation of calibrated results from ground truth in two experiments. The quantitative analysis of calibration deviation can be seen in Figure 7. In Figure 7 (b), the proposed method was able to simultaneously correct roll with a mean average error of 0.288 degrees, pitch with a mean average error of 0.268 degrees, and yaw with a mean average of 0.108 degrees each. If the confidence below 0.7 was ignored, the error of roll, pitch, and yaw will decrease to 0.217, 0.209, 0.082 degrees. In Figure 7 (d), the mean error of roll, pitch, and yaw are 0.217, 0.228, and 0.079 degrees. If the confidence below 0.7 was ignored, the error of roll, pitch, and yaw will decrease to 0.159, 0.164, and 0.070 degrees. Without counting the frames with manual error, the maximum error of roll, pitch, and yaw are always within 0.5 degrees. The calibration results of yaw are the most precise since the LiDAR has a high horizontal resolution. Even though the LiDAR's resolution in vertical direction is much lower and the 3D features in that direction are less frequently presented, our approach can still achieve high accuracy due to the adaptive optimization algorithm and higher weight in this direction. Overall, an average rotational error of 0.12 degrees across all dimensions is achieved, which is lower than most offline calibration techniques.

Additionally, we also tested the speed of correcting bias. Figure 7 (a) and (c) represent the confidence of current calibration result. The extrinsic parameters were worse after artificially applying a bias, showing a larger angle deviation. In Figure 7 (a) and (b), the black arrows 1, 2, 3, and 4 pointed at the frames of adding biases. For arrows 1 and 3 (30th

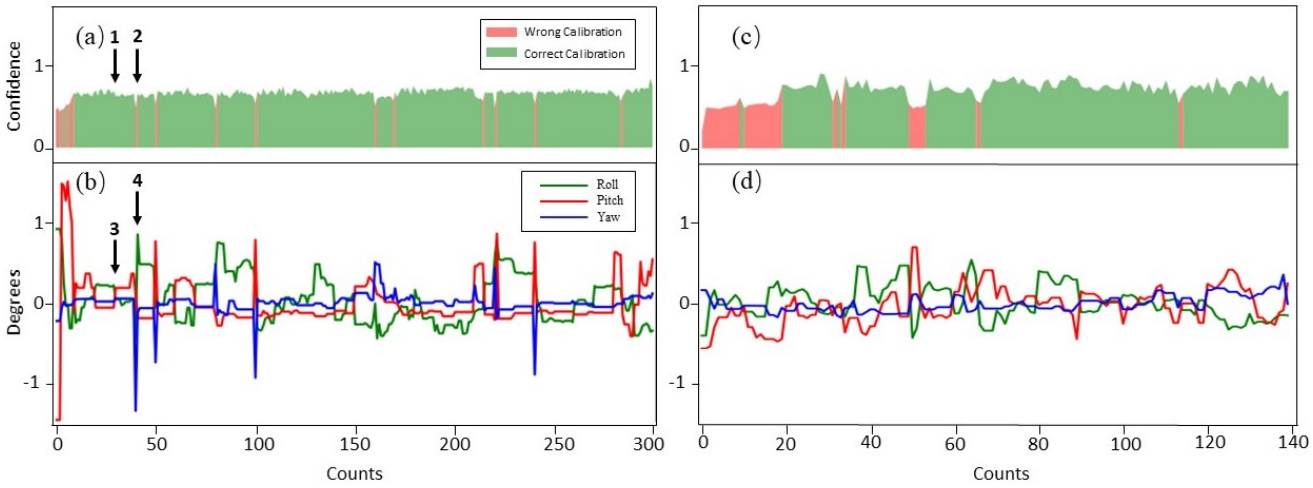

Fig. 7. Simultaneously detecting and correcting roll, pitch, and yaw deviations in real time. (a) shows the confidence of calibration results in experiment I; (b) shows the calibration results of our method in experiment I; (c) shows the confidence of calibration results in experiment II; (d) shows the calibration results of our method in experiment II. The green, red, and blue lines in (b) and (d) show the biases of roll, pitch, and yaw compared to ground truth. The red sections in (a) and (c) represent a low confidence of current parameters. On the contrary, the green sections in (a) and (c) shows a high confidence of current calibration result.

Fig. 8. The calibration results on several scenarios. Green points represent the projected LiDAR line points.

frame), the proposed method corrected the bias immediately without showing the decline of confidence, while for the black arrows 2 and 4 (40th frame), the red section turned green within two frames (Figure 7 (a)) which means the bias was corrected within two frames. The same situation apperaed in experiment II (Figure 7 (c) and (d)). The overall calibration results in more scenarios on KITTI dataset can be seen in Figure 8, which demonstrate that the proposed method is applicable to different scenarios.

V. CONCLUSIONS

With the growing importance of sensors used in the multi-modal perception system in autonomous vehicles, the precise extrinsic parameters between these sensors are essential to provide a high-precision perception system. However, long-term operation under different loads can cause a slight change and drift from the extrinsic transformation matrix, which compromises perception accuracy. Therefore, it is crucial to automatically correct the miscalibration during operation.

In this paper, we have proposed an online approach that can automatically calibrate the extrinsic parameters of LiDAR and camera. Different from previous automated methods, this new calibration approach requires no markers to be placed in the scene. And we demonstrate that the line features from point clouds and images are robust features to correct calibration biases. The artificially added bias can be corrected within one or two frames, which is faster than other methods. In addition, we illustrate that the degree of confidence for current calibration result can be computed and further utilized to improve the computation efficiency and accuracy.

For future work, we would like to evaluate the accuracy of extracted line features to reduce the calibration bias in scenes with few features. And the Monte Carlo method can be utilized to provide the initial parameters.

ACKNOWLEDGMENT

It is sponsored by Tsinghua University-Didi Joint Research Center for Future Mobility.

REFERENCES

[1] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[2] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[6] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[7] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.

[8] V. Fremont, P. Bonnifait, *et al.*, "Extrinsic calibration between a multi-layer lidar and a camera," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2008, pp. 214–219.

[9] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[10] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information." in *AAAI*. Citeseer, 2012.

[11] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3936–3943.

[12] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2862–2866.

[13] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[14] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.

[15] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 4164–4169.

[16] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 696–701.

[17] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[18] Z. Deng, L. Xiong, D. Yin, and F. Shan, "Joint calibration of dual lidars and camera using a circular chessboard," SAE Technical Paper, Tech. Rep., 2020.

[19] V. Fremont, S. A. Rodriguez F, and P. Bonnifait, "Circular targets for 3d alignment of video and lidar sensors," *Advanced Robotics*, vol. 26, no. 18, pp. 2087–2113, 2012.

[20] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple lidars and cameras on autonomous vehicles," *Robotics and Autonomous Systems*, vol. 83, pp. 326–337, 2016.

[21] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," in *Proceedings of the 2010 Australasian Conference on Robotics & Automation:*. ARAA, 2010, pp. 1–10.

[22] E.-s. Kim and S.-Y. Park, "Extrinsic calibration between camera and lidar sensors by matching multiple 3d planes," *Sensors*, vol. 20, no. 1, p. 52, 2020.

[23] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Australian Conference on Robotics and Automation*, 2012, pp. 3–5.

[24] ——, "Automatic calibration of lidar and camera images using normalized mutual information," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[25] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[26] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4843–4850.

[27] ——, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.

[28] R. Ishikawa, T. Oishi, and K. Ikeuchi, "Lidar and camera calibration using motions estimated by sensor fusion odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7342–7349.

[29] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3685–3691.

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

[31] T. Li, J. Fang, Y. Zhong, D. Wang, and J. Xue, "Online high-accurate calibration of rgb+ 3d-lidar for autonomous driving," in *International Conference on Image and Graphics*. Springer, 2017, pp. 254–263.

[32] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.