2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)
September 12-14, 2018, Madrid, Spain

Line Feature Based Extrinsic Calibration of LiDAR and Camera

Jingjing Jiang[1], Peixin Xue[1], Shitao Chen[1], Ziyi Liu[1], Xuetao Zhang[1], Nanning Zheng[1]

***Abstract*—Reliable extrinsic calibration is a crucial first step for multi-sensor data fusion, which is the key part of the autonomous vehicle to perceive the environment carefully and effectively. In this paper, we propose an effective extrinsic calibration pipeline to establish the transformation between camera and LiDAR and update the decalibration online on an autonomous driving platform. We obtain rotation extrinsic parameters using parallel lines features in road scene, and infer translation extrinsic parameters by an online search approach based on selective edge alignment of point cloud and image. In order to evaluate our calibration system, it is first validated on KITTI benchmark and compared with the baseline algorithm. After that, the proposed method is tested on our own data. The results show that our method has a better rotation accuracy and demonstrate the necessity of error correction online.**

I. INTRODUCTION

Multi-sensor data fusion is the key to robust environmental perception of unmanned vehicle, while laser and camera are the most commonly used sensors in the perception stage, since laser can gain accurate 3D location information in any light condition, and camera can obtain rich texture information. Their complementary capabilities make their fusion significant for sensing algorithm.

In this paper, we propose a practical extrinsic calibration pipeline, shown in Fig.1. To deal with the extrinsic calibration problem of LiDAR and camera in autonomous driving system, we find that the rotational components have a larger impact on the quality of the final calibration results. So, we compute the analytical expression of rotation angles using any three parallel lines that can be detected both in camera and LiDAR, e.g. parallel lane lines in straight road, based on the triple-lines approach.

The proposed approach is implemented in three steps. First, analytical solutions of rotation extrinsic parameters are computed by the pose of the ground plane and the lines lay in the plane. Then, translation extrinsic parameters are inferred by a search algorithm based on edge alignment with the installation initial values. Finally, we have the online updating module monitoring the calibration error, once the reprojection error exceeds the setting threshold, which means the current extrinsic parameters are unable to map the positional relationship between camera and LiDAR, the search algorithm will run to correct the errors.

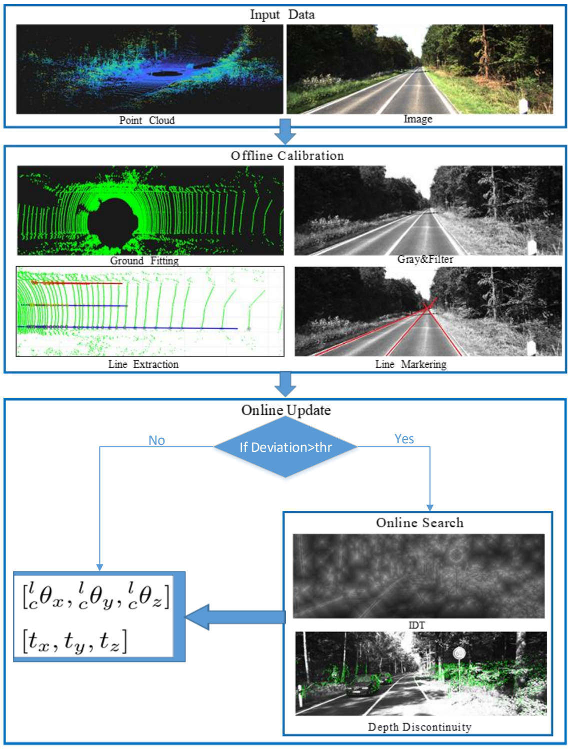

Fig. 1. The process of our calibration pipeline. We compute rotation angles and estimate translation parameters offline, after that we update decalibration in real time.

The main contributions of this paper contain three aspects. First, our calibration process without dedicated targets is simple. Moreover, we compute rotation extrinsic parameters using analytic solutions of parallel line constrains, which is more accurate and efficient in solving the error sensitive problem of rotation transformation. Last but not least, we monitor and update extrinsic parameters by online module for error tracking and compensation, which is important and efficiency for error correction online.

II. RELATED WORK

A good fusion of point cloud and image heavily depends on the accurate extrinsic calibration for LiDAR and camera. In the past decade, many approaches for LiDAR-camera system calibration have been proposed. In general, these methods can be divided into two groups, off-line [1]–[6] and online [7]–[13], according whether they need manual targets or work in

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shannxi, P.R.China jjj52044, bestxin, shitaochen, ziyiliu@stu.xjtu.edu.cn; xuetaozh, nnzheng@xjtu.edu.cn

978-1-5386-3543-8/18/$31.00 ©2018 IEEE

any structured scenes. Off-line methods need manual targets or existence features in the scene for feature matching. In [1], multiple planar chessboards in different positions in the same frame are used as the targets. Some other approaches adopt special planar surfaces, such as [2], and [3] uses a triangular surface. There are also other types of targets, such as a sphere [4] and a cuboid [5]. In [6], they calibrate LiDAR-stereo camera pairs using a plane with 4 holes, without user intervention. Off-line calibration algorithms based on special targets can achieve higher precision for the matching features are easy to choose. However, these algorithms are separated from subsequent processing which make them hard to correct the calibration errors resulting from vibration, temperature and other accidents in real time.

Accordingly, many attempts have been made to calibrate LiDAR-camera system automatically, flexibly and without dedicated targets. For example, [7] and [8] estimate extrinsic parameters by maximizing the mutual information between the reflectivity of point cloud and the intensity of image. In [9], a online method that corresponding pixels edges in image and points edges of depth discontinuities in point cloud is proposed. An extension algorithm presents by [11], which aligns depth edges generated by 3D point cloud with intensity edges. The approach taken by [10], solves the 2D-3D registration problem using a minimum of one or two planar regions, which is visible in both camera and laser. In [12], a LiDAR-monocular visual odometry framework is used to track and compensate the temporal changes of extrinsic parameters. Besides, a deep convolutional neural network, RegNet [13] is presented to infer the 6 DOF(degree of freedom) extrinsic calibration between multimodal sensors. Online calibration is a process of recalibration and error compensation, without human intervention, which is convenient to operate, but it is difficult to achieve high precision.

Considering above advantages and disadvantages, in this paper, we propose an novel calibration pipeline. We compute the rotation angles accurately and search for the most matching translation values, which are used as the initial value of online updating module. Under the calibration system, the calibration error and the disturbance can be compensated and rectified in real time.

III. METHOD

To describe the transformation between LiDAR and Camera, coordinate systems are established as shown in Fig.2, which provide the spatial position relationships between them. ${}^{L}_{W}\Phi = ({}^{L}\theta_X, {}^{L}\theta_Y, {}^{L}\theta_Z)$ are the three rotation angles (Roll, Pitch, Yaw) of LiDAR with respect to the world system, and ${}^{C}_{W}\Phi = ({}^{C}\theta_X, {}^{C}\theta_Y, {}^{C}\theta_Z)$ are the poses of camera in the world system.

A. Pose Estimation of LiDAR

In LiDAR coordinate system, the MSAC (variant of the RANdom SAmple Consensus) algorithm [14] is applied to estimate the unit normal vector ${}^{l}\vec{\mathbf{n}} = (A, B, C)$ for the road plane model. As [15] presents, the pose of road plane can determine two rotation angles of the LiDAR − roll and pitch. Using the normal vector of ground, the rotation transformation can be described as:

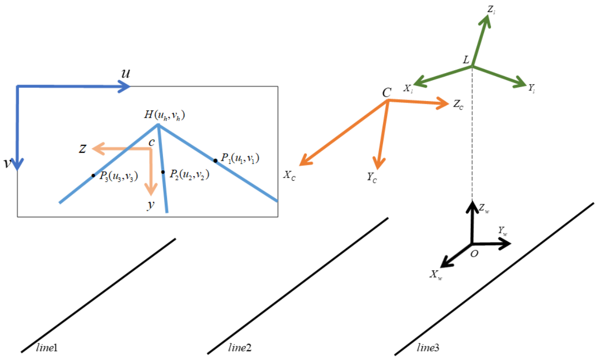

Fig. 2. The coordinate system (right), the three parallel lines lay in the image plane and their correlative points (left). Based on the assumption that the ground is flat and there are three mutually non-coincident parallel lines lay in it, the origin of world coordinate O_w is the projection of LIDAR on the ground plane, and X_w axis is defined as the parallel line of the three parallel lines.

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = {}^{L}_{W}\mathbf{R}_y {}^{L}_{W}\mathbf{R}_x \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{1}$$

where, rotation matrix, ${}^{L}_{W}\mathbf{R}_y {}^{L}_{W}\mathbf{R}_x$ presented in (1).

$${}^{L}_{W}\mathbf{R}_{y\,x} = {}^{L}_{W}\mathbf{R}_y {}^{L}_{W}\mathbf{R}_x = \begin{bmatrix} {}^{l}c_y & {}^{l}s_x {}^{l}s_y & {}^{l}c_x {}^{l}s_y \\ 0 & {}^{l}c_x & -{}^{l}s_x \\ -{}^{l}s_y & {}^{l}c_y {}^{l}s_x & {}^{l}c_y {}^{l}c_x \end{bmatrix} \tag{2}$$

where ${}^{l}c_y, {}^{l}s_y, {}^{l}c_x, {}^{l}s_x$ are $\cos{}^{L}\theta_Y, \sin{}^{L}\theta_Y, \cos{}^{L}\theta_X$ and $\sin{}^{L}\theta_X$ respectively.

Relating (1) and (2), the two rotation parameters are obtained by (3).

$$\begin{cases} {}^{L}\theta_X = -\arcsin(B) \\ {}^{L}\theta_Y = \arcsin(\frac{A}{\cos{}^{L}\theta_X}) \end{cases} \tag{3}$$

Since 3D plane has only two rotational degrees of freedom, it is impossible to determine the yaw angle just by the road plane. Consequently, the parallel lines lay in the ground are presented to estimate the yaw angle of LiDAR. A method based on OTSU thresholding for LiDAR [16] is used to extract road curbs or the parallel lines. Fitting the lines used least squares method, it is easy to obtain the direction vector of the line ${}^{l}\vec{\mathbf{a}} = (i, j, k)$. And the yaw is defined as the angle between the projection of ${}^{l}\vec{\mathbf{a}}$ on the $O_w - X_wY_w$ and X_w axis:

$${}^{L}\theta_Z = \arccos(\frac{i}{\sqrt{i^2 + j^2}}) \tag{4}$$

B. Pose Estimation of Camera

To calibrate the pose angles of camera with respect to the world reference, as depicted in [17], an analytical expression can be obtained using the constraints between camera external parameters with vanishing point and slope, which generated by the mutually non-coincident parallel lines in image plane.

The transformation of a point $P_w(X_w, Y_w, Z_w)$ in the world frame into $P_c(X_c, Y_c, Z_c)$ of the camera frame can be presented as:

$$\begin{aligned} &{P_w}^T = {}_W^C\mathbf{R} \cdot {P_c}^T + t^T \\ &\Rightarrow {P_c}^T = {}_W^C\mathbf{R}^{-1} \cdot {P_w}^T - {}_W^C\mathbf{R}^{-1} \cdot t^T \\ &\Rightarrow {P_c}^T = {}_W^C\mathbf{R}^T \cdot {P_w}^T - {}_W^C\mathbf{R}^T \cdot t^T \end{aligned} \tag{5}$$

where, $t(l, d, h)$ is the expression of the origin of camera in the world frame, and the rotation matrix is defined as (6).

$$\begin{aligned} {}_W^C\mathbf{R} &= \begin{bmatrix} {}^c c_y^c c_z & {}^c c_z^c s_x^c s_y +^c c_x^c s_z & {}^c s_x^c s_z -^c c_x^c c_z^c s_y \\ -{}^c c_y^c s_z & {}^c c_x^c c_z -^c s_x^c s_y^c s_z & {}^c c_x^c s_y^c s_z +^c c_z^c s_x \\ {}^c s_y & -{}^c c_y^c s_x & {}^c c_x^c c_y \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \end{aligned} \tag{6}$$

Where ${}^c c_z$, ${}^c s_z$, ${}^c c_y$, ${}^c s_y$, ${}^c c_x$, ${}^c s_x$ are $\cos {}^C\theta_Z$, $\sin {}^C\theta_Z$, $\cos {}^C\theta_Y$, $\sin {}^C\theta_Y$, $\cos {}^C\theta_X$ and $\sin {}^C\theta_X$ respectively.

Relating (5) and (6), the presentation of line $X_w = s, Y_w = a, Z_w = 0, s \in R$ in the world frame is (7),

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} sr_{11} + ar_{21} - lr_{11} - dr_{21} - hr_{31} \\ sr_{12} + ar_{22} - lr_{12} - dr_{22} - hr_{32} \\ sr_{13} + ar_{23} - lr_{13} - dr_{23} - hr_{33} \end{bmatrix} \tag{7}$$

As shown in the Fig.2, the pixel coordinate (u, v) and image plane coordinate (z, y) satisfy (8).

$$\begin{cases} z = (u - u_c)du \\ y = (v - v_c)dv \end{cases} \tag{8}$$

where (u_c, v_c) is the principal point, du, dv are scale coefficients along u and v axis. With pinhole imaging model, the image coordinate can also be presented as below.

$$\begin{cases} z = -f_u d_u \frac{Z_C}{X_C} \\ y = f_v d_v \frac{Y_C}{X_C} \end{cases} \tag{9}$$

where, f_u, f_v are the focal length along the u and v coordinate directions.

After obtaining the camera intrinsic parameters, the distance a_1, a_2, a_3 of the three lines with regard to X_w axis, the vanishing point $H(u_h, v_h)$ and any three points $P_1(u_1, v_1)$, $P_2(u_2, v_2)$, $P_3(u_3, v_3)$ except point H, the slopes of three lines can be defined as $r_n = -\frac{f_u}{f_v} \cdot \frac{u_h - u_n}{v_h - v_n}$ using pixel coordinate, as well as $k_n = \frac{d_z/d_s}{d_y/d_s}, n = 1, 2, 3$ using camera pose parameters. Accordingly, mathematical derivation of extrinsic analytical expressions is described as follows:

$$\begin{aligned} &k_n = -\frac{f_v d_v}{f_u d_u} \frac{a_n(r_{13}r_{21} - r_{11}r_{23}) - f^*)}{a_n(r_{12}r_{21} - r_{11}r_{22}) - f^{**})} \\ &\Rightarrow \frac{k_1 - k_2}{k_1 - k_3} = \frac{(a_1 - a_2)\tan {}^C\theta_X - (k_1 a_1 - k_2 a_2)}{(a_1 - a_3)\tan {}^C\theta_X - (k_1 a_1 - k_3 a_3)} \\ &\Rightarrow {}^C\theta_X = \\ &atan\frac{(r_1 - r_3)(r_1 a_1 - r_2 a_2) - (r_1 - r_2)(r_1 a_1 - r_3 a_3)}{(r_1 - r_3)(a_1 - a_2) - (r_1 - r_2)(a_1 - a_3)} \end{aligned} \tag{10}$$

where, f^* and f^{**} are the constant expressions about d and h, and they can be eliminated by calculating the slope ratio.

Then the other two rotation angles can be deduced from the expression of the vanishing point where these three lines intersect in image plane as (11).

$$\begin{cases} y_h = \lim_{s\to\infty} -f_u d_u \cdot \frac{Y_c}{X_c} = -f_u d_u \frac{r_{12}}{r_{11}} \\ z_h = \lim_{s\to\infty} f_v d_v \cdot \frac{Z_c}{X_c} = f_v d_v \frac{r_{13}}{r_{11}} \end{cases} \tag{11}$$

Relating (6) and (11), the rest angles are (12).

$$\begin{cases} {}^C\theta_Y = \arctan \frac{y_h}{f_u d_u} \sin {}^C\theta_X - \frac{z_h}{f_v d_v} \cos {}^C\theta_X \\ {}^C\theta_Z = (\frac{y_h}{f_u d_u} - \sin {}^C\theta_Y \tan {}^C\theta_X) \arctan \frac{\cos {}^C\theta_Y}{\cos {}^C\theta_X} \end{cases} \tag{12}$$

C. Extrinsic Calibration of LiDAR and Camera

In contrast to rotation, translation is measurable, so we optimize the initial translation value (installation value) with a search algorithm based on the correlation of intensity edge of image and the depth discontinuity edge of point cloud and obtain the fine translations.

- **Edge of Image**

To extract image edge generated by the geometric boundary of the target as much as possible, we calculate the vertical and horizontal edge respectively using directional gradient operators, like the Sobel gradient operator. Note that the parallel lines are also considered as features for alignment, since they are obvious in the distance, which can constrain the distant alignment. Then, isolated points of these edges are filtered and the Inverse Distance Translation (IDT) [9] is applied to smooth the edge.

- **Edge of Point Cloud**

Besides these points of parallel lines, the outlier points in fitting ground plane are chosen as candidates. Afterwards, we look for points that are closer than at least one of their two neighbors and exploit (13) to define their intensity values in outlier points. Finally, we further filter these found points by setting a intensity threshold to remove points with small depth discontinuities.

$$X_i = \max(P_{i-1}^r - P_i^r, P_i^r - P_{i+1}^r, 0)^\gamma \tag{13}$$

where P_i^r is depth of the i-th point, $\gamma \in (0, 1)$ is a weight to stretch the contrast of these low depth points.

After that, the objective function is represented by the weighted product of the edge of point cloud and the edge image after IDT, as presented as (14).

$$F_T = \sum_{i=1}^{N} \lambda_i E(u_i, v_i) \cdot \omega_i D(u_i, v_i) \tag{14}$$

where λ_i is the weight associated with the direction of edge points, and we use the empirical value 1.0 for vertical edge and 0.7 for the horizontal since vertical edges are likely to be generated by geometric boundary; D is the depth map of 3D points, whose pixel value is (u, v) when projected to the image plane with intensity X; ω_i is the weight of distance between target and the sensor and it is defined by (15).

$$\omega_i = \frac{D_{roi}}{\sqrt{X_{L_i}^2 + Y_{L_i}^2}} \tag{15}$$

where, D_{roi} is the set range for the ROI of point cloud.

We use $\Psi = [t_X, t_Y, t_Z, \theta_X, \theta_Y, \theta_Z]$ to represent extrinsic parameters that LiDAR is relative to camera. The overall search method is concluded in **Algorithm 1**. In this stage, the steps of rotation angles are set to 0 for better to search translation parameters, which are initialized by factory default or measurement values.

Algorithm 1 Extrinsic Calibration of LiDAR and Camera

```
Input:
    initial parameters $\Psi_0$, intrinsic matrix $K_{in}$,
    depth discontinuity $THR_d$, search step $\Delta_\Psi$,
    reprojection $THR_r$, PC set $\mathbf{P}$, RGB image $\mathbf{I}$
Output:
    fine extrinsic parameters $\Psi$
$\mathbf{I}_g \leftarrow rgb2gray(\mathbf{I})$
$E_x \leftarrow Sobel_x(\mathbf{I}_g)$, $E_y \leftarrow Sobel_y(\mathbf{I}_g)$
$E \leftarrow IDT(E_{x/y})$
$X_i \leftarrow \mathbf{P}$
$X \leftarrow X_i > THRES_d$
compute $\omega_i$
$\Psi_{max} \leftarrow \Psi_0$; Initial $\Psi_{optm} \leftarrow 0$
while $\Psi_{max} \neq \Psi_{optm}$ do
    $\Psi_{optm} \leftarrow \Psi_{max}$
    $\Psi_n \leftarrow$Give $\Psi_{optm}$ a perturbation
    for $k = 1$ to $n$ do
        Transform $\Psi_k$ to extrinsic matrix $K_e$
        $D \leftarrow$ Project $X$ on image plane by $K_{in}$ $K_e$
        $F_{Tk} \leftarrow \sum E \cdot D$
    end for
    Find optimal $\Psi_{max}$ whose $F_T$ is max
end while
update $\Psi$
```

D. Online Update for Decalibration

Once the autonomous driving system goes online, external forces, such as mechanical vibrations, may make the accurate parameters shift with the movement of driverless car. As these calibration results of different periods shown in Fig.3, fine parameters of offline calibration dosen't represent the real transformation between LiDAR and camera after some bumps and jolts. Therefore, the calibration system has to correct such decalibration in real time, so we use the average projection error of point cloud to determine whether there is need to execute online search algorithm.

The search algorithm of online update module is the same as **Algorithm 1**. In this case, we use 6 frames of point cloud and the corresponding images collected in real time as the matching patterns. The current extrinsic parameters serve as initial value. In addition, we find that the translation along vertical direction t_Z, the pitch angle θ_Y and the yaw angle θ_Z are more susceptible to external forces than the rest extrinsic parameters. So, we first search these three values that are more sensitive to external forces while fixing the other parameters. After that, we set a smaller search step to fine-tune the 6 DOF of extrinsic parameters for better calibration quality.

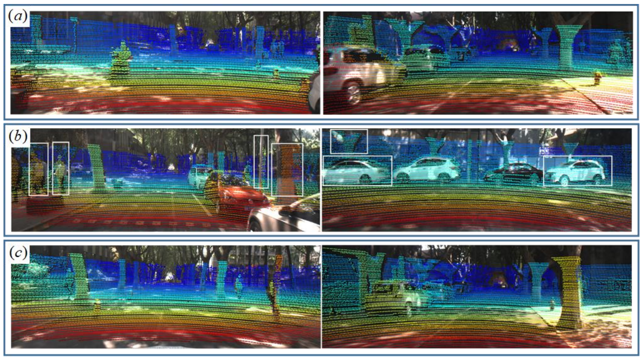

Fig. 3. Validation of calibration results. (a) results of offline calibration, (b) after the unmanned car tested a week, extrinsic parameters changed and needed to be updated. (c) after running the online search algorithm, the decalibration was corrected.

IV. Experiment

A. Our Dataset and Validation Method

To evaluate the performance of the proposed calibration algorithm, KITTI road datasets [18] is used to validate the feasibility of our calibration method. After that, we test and apply the algorithm using our own data collected by our autonomous driving system as Fig.4 (a), whose sensors (LiDAR and camera) and installation position are similar to KITTI's [18]. But, the extrinsic calibration result is difficult to quantitatively evaluate since the ground truth of extrinsic parameters is unable to measure. So, the calibration quality is usually verified manually with a visual inspection. We project 3D point (X_L, Y_L, Z_L) on image plane by (16), and obtain the point cloud projection image colored with depth or reflectivity.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K_{in} K_e \begin{bmatrix} Z_L \\ Y_L \\ X_L \\ 1 \end{bmatrix} = K_{in} \begin{bmatrix} {}_W^C\mathbf{R}^{-1} {}_W^L\mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} Z_L \\ Y_L \\ X_L \\ 1 \end{bmatrix} \tag{16}$$

Getting the pixel fusion result of the point cloud projection image and the RGB image, we can inspect the edge alignment, in which these object and sign edges are obvious. Specifically, if the target can be seen both in camera and LiDAR, and its edge of point cloud can coincide with the edge of image, then the accuracy of the calibration results can be proved to be high. On the contrary, if there is a misplacement, the calibration results are possibly wrong and the extrinsic parameters will not be available when the misplacement error is larger than a certain range. Fig.4 shows the comparison of edge alignment with a high and bad calibration quality.

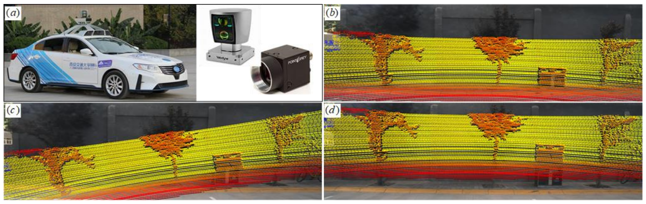

Fig. 4. (a)our test platform; (b) high quality calibration result; (b) (c) the bad one with tilt or misplacement.

B. Validation On KITTI Benchmark

We choose a image-point cloud pair which has three parallel lane lines to compute the rotation extrinsic parameters, and 10 pairs whose intensity edge and depth discontinuity are easy to be extracted for searching translation extrinsic parameters, in ROAD-KITTI data set. We firstly get points P_1, P_2, P_3, P_h by manually marking and compute a_1, a_2, a_3 using the actual 3D information from point cloud. And then, we gain the intrinsic parameters of camera with checkerboard image provided by KITTI calibration data sets.

After finishing the preparatory work, we compute the poses of LiDAR and camera in the world coordinate system and use search algorithm to tune these translation parameters slightly, which are initialized by KITTI's baseline values. Next, putting everything together, we can calculate the 2D pixel coordinates of the 3D points and get the validation image of calibration results based on pixel fusion.

We compare the rotation angles calibrated by our proposed algorithm with KITTI's baseline, Table I shows that the rotation angle errors are less than 0.5 degree and the yaw angle has the largest error. For better comparison, the verification image is shown in Fig.5, which demonstrates that our calibration results have a better effect of aligning point cloud with pixel in image boundary. It indicates that the method we proposed is even more accurate than KITTI' baseline algorithm, and it also verifies the feasibility of our calibration system and the accuracy of our rotation calibration method.

TABLE I
ROTATION ANGLES COMPARISON

Rotation Angle	θ_X	θ_Y	θ_Z
KITTI /Ground Truth	-0.682	89.401	-88.713
Our Test	-0.678	89.179	-88.30

C. Experiment Results of Our Data

In this section, we test our rotation extrinsic calibration algorithm on a more general scene, without three parallel lane lines, just using the curb lines and the median separation line of the left and right path. The image-point cloud pair we use to calibrate rotation angles shows in Fig.6.

Then, we use the measurements to initialize the translation extrinsic parameters and infer them using the search algorithm. To improve calibration accuracy, we set different steps in a gradient descent way and repeat the search process. The initial values and the optimum results of translation show in Table II. Combing translation values with rotation parameters, the visualization of final calibration results on our own data is shown as Fig.7 and more alignment details of distant objects are shown in Fig.8.

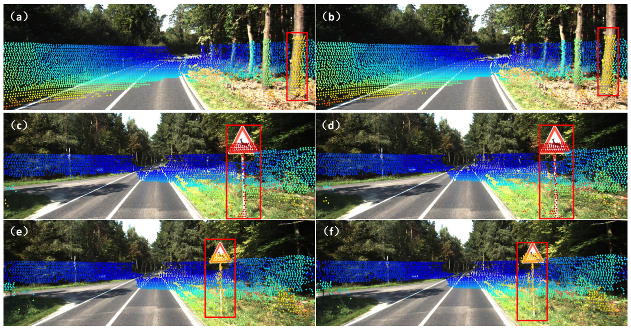

Fig. 5. The validation images. (a)(c)(e) the validation results of the ground truth of KITTI [18], (b)(d)(f) results of the proposed method.

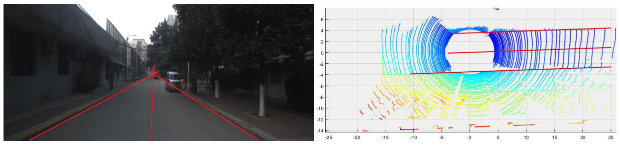

Fig. 6. The image-point cloud pair we use to calibrate rotation extrinsic parameters and the three parallel lines extracted both in LiDAR and camera

TABLE II
TRANSLATION CALIBRATION RESULTS

Translation /mm	t_X	t_Y	t_Z
Initialization (Measurement)	-110	-500	50
Calibration (Search Algorithm)	-134.0	-499.4	32.8

Finally, we apply our calibration system to our autonomous driving platform to monitor and correct the decalibration errors, which avoids the tedious and repeated offline calibration work.

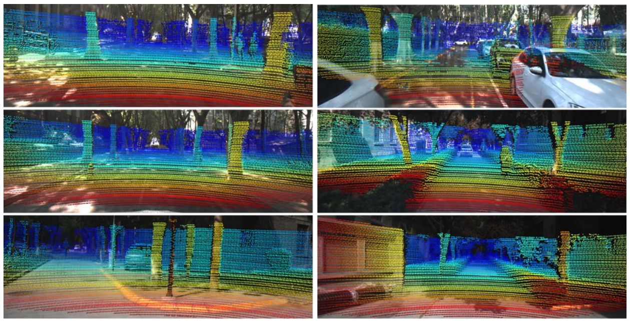

Fig. 7. The pixel fusion results based on extrinsic calibration on our test datasets. It shows that edges of trees, cars, buildings along the road and pedestrians in front of the car are alignment.

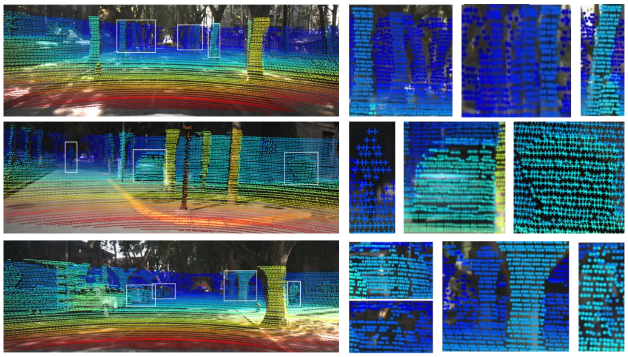

Fig. 8. The alignment details of distant objects. It shows that edges of trees, cars and pedestrians in the distance are alignment.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel online updating extrinsic calibration pipeline to solve the rigid transformation problem that aligns 3D point cloud with 2D pixel, and we evidently verify its effectiveness in real-time calibration application. First of all, we obtain the rotation angles that 3D LiDAR coordinate is relative to camera coordinate at the aid of ground plane and lines in the road. Secondary, we extract the edges both in image and LIDAR point set, and adopt the online search method based on edges to acquire a more accurate translation. Finally, we add a online updating module to monitor and correct decalibration. The calibration results show a better performance than the KITTI's baseline on rotation extrinsic parameters and prove the necessity of updating extrinsic parameters online. Moreover, our method have a lower dependency on the calibration scenes compared with others. In our future work, the feature of road mark will be added to compute the analytical expression of translation rather than search potential optimal value, making the calibration more precise and robust. Besides, we will improve the search algorithm to avoid the convergence due to local optimal values.

ACKNOWLEDGMENT

This research was partially supported by the National Natural Science Foundation of China (No. 61773312, 61790563), the Programme of Introducing Talents of Discipline to University (No. B13043) and the National Key R&D Program of China(No.2017YFC 0803905).

REFERENCES

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

[2] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3d lidar camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 452–467, 2012.

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

[4] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple lidars and cameras on autonomous vehicles," *Robotics & Autonomous Systems*, vol. 83, no. C, pp. 326–337, 2016.

[5] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Joint Workshop on Multi-Sensor Fusion for Dynamic Scene Understanding at ICCV*, 2017.

[6] C. Guindel, J. Beltrn, D. Martn, and F. Garca, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," 2017.

[7] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Acra*, 2012.

[8] 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 *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012, pp. 2053–2059.

[9] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems*, 2013.

[10] L. Tamas and Z. Kato, "Targetless calibration of a lidar - perspective camera pair," in *IEEE International Conference on Computer Vision Workshops*, 2014, pp. 668–675.

[11] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[12] H. J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular lidar-camera systems," in *International Conference on Pattern Recognition*, 2017, pp. 2848–2853.

[13] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "Regnet: Multimodal sensor registration using deep neural networks," in *Intelligent Vehicles Symposium*, 2017, pp. 1803–1810.

[14] P. H. S. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision & Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[15] C. H. Rodriguez-Garavito, A. Ponz, F. Garcia, D. Martin, A. D. L. Escalera, and J. M. Armingol, "Automatic laser and camera extrinsic calibration for data fusion using road plane," in *International Conference on Information Fusion*, 2014, pp. 1–6.

[16] A. Hata and D. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," in *IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 584–589.

[17] Q. Li, N. Zheng, and X. Zhang, "Triple-lines approach to calibration of a camera mounted in an intelligent vehicle," in *Defense and Security*, 2004.

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