

Letter

# LiDAR-Camera Calibration Using Line Correspondences

Zixuan Bai <sup>†</sup>, Guang Jiang <sup>\*</sup> and Ailing Xu <sup>†</sup>

School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; zxbai\_1@stu.xidian.edu.cn (Z.B.); alxu@stu.xidian.edu.cn (A.X.)

\* Correspondence: gjiang@mail.xidian.edu.cn; Tel.: +86-1336-397-6876

† These authors contributed equally to this work.

Received: 14 September 2020; Accepted: 2 November 2020; Published: 5 November 2020



**Abstract:** In this paper, we introduce a novel approach to estimate the extrinsic parameters between a LiDAR and a camera. Our method is based on line correspondences between the LiDAR point clouds and camera images. We solve the rotation matrix with 3D–2D infinity point pairs extracted from parallel lines. Then, the translation vector can be solved based on the point-on-line constraint. Different from other target-based methods, this method can be performed simply without preparing specific calibration objects because parallel lines are commonly presented in the environment. We validate our algorithm on both simulated and real data. Error analysis shows that our method can perform well in terms of robustness and accuracy.

**Keywords:** extrinsic calibration; LiDAR-Camera; line correspondence; infinity point

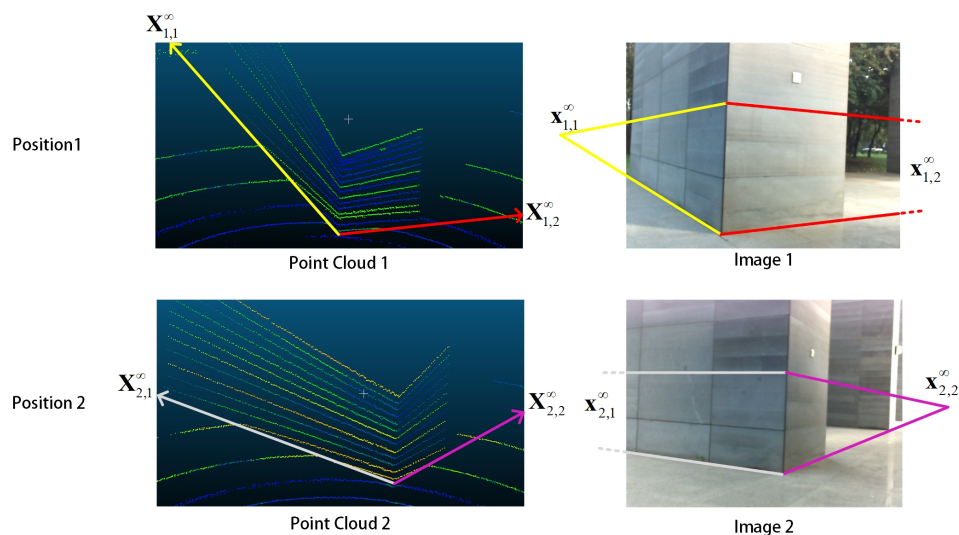
## 1. Introduction

Nowadays, with the popularity of unmanned vehicles, the navigation problems inherent in mobile robots are garnering even greater attention, among which the localization or calibration between different sensors is one of the basic problems. To fully utilize the information from sensors and make them complementary, the combination of 3D and 2D sensors is a good choice. Thus, the hardware devices of those systems are usually based on cameras and Light Detection and Ranging (LiDAR) devices. Comparing the two sensors, a camera is cheap and portable, and it can obtain color information about the scene, but it needs to correspond to feature points during calculation, which will be time consuming and sensitive to light. LiDAR can get 3D points directly and has an effective distance of up to 200 m. In addition, LiDAR is suitable for low-textured scenes and some scenes under varying light conditions. However, the data are sparse and lack texture information. When using a combination of cameras and LiDAR, it is necessary to obtain transformation parameters between coordinate systems of the two kinds of sensors. Once the transformation parameters, i.e., the rotation matrix and translation vector are obtained, the two coordinate systems are aligned, and the correspondence between 3D points and the 2D image is established. The 3D point cloud obtained by the LiDAR can be fused with the 2D image obtained by the camera.

The existing target-based methods require users to prepare specially designed calibration targets such as chessboard [1], circular pattern [2], orthogonal trihedron [3], etc., which limits the practicality of these methods. Target-less methods break through this limitation. These kinds of methods can be roughly divided into several categories according to work principles: odometry-based, neural network-based, and feature-based. The odometry-based methods[4,5] require many continuously inputted data, and the neural network-based methods[6,7] need even more data to train networks, and may lack clear geometric constraints. The feature-based methods usually use point or line features from scenes. Point feature is sensitive to noise, sometimes requiring user intervention to

establish 3D–2D point constraints [8]. Line feature is more stable, and 3D–2D line correspondence is usually required (known as the Perspective-n-Line problem) [9–11]. However, in an outdoor environment, this correspondence is usually hard to be established. Because LiDARs are generally placed horizontally, many detected 2D lines on the image cannot find their paired 3D counterparts due to the poor vertical resolution.

The main contribution of this paper is that we provide a novel line-based method to solve the extrinsic parameters between a LIDAR and a camera. Different from existing line-based methods, we take infinity points into consideration to utilize 2D lines, so that the proposed method can work in outdoor environments with artificial buildings, as shown in Figure 1. As long as there are enough parallel line features in a scene, it can be chosen as calibration environment. In addition, our method only requires a small number of data to achieve sufficient results. We transform the correspondence of parallel lines into the correspondence between 3D and 2D infinity points. By getting and aligning the direction vectors from the infinity points, the rotation matrix can be solved independently in the case that the camera intrinsic matrix is known. Then, we use a linear method based on point-on-line constraint to solve the translation vector.



**Figure 1.** General constraints of our method. The first row shows the point cloud (top-left) and image (top-right) captured at position 1, and the second row shows the case of position 2.  $X_{j,i}^{\infty}$  represents the 3D infinity points in the LiDAR coordinate system.  $x_{j,i}^{\infty}$  represents the corresponding 2D infinity points on the image plane. During the position change, the relative transformation of the two coordinate systems is fixed. In our method, an initial solution can be obtained from at least two positions.

## 2. Related Work

The external calibration between two sensors is always discussed. According to the different forms of data collected by these two devices, researchers have been looking for appropriate methods to obtain conversion parameters between the two coordinate systems. In some methods, the target is a chessboard, which is a plane object. Zhang et al. [1] proposed a method based on observing a moving chessboard. After getting points-on-plane constraints from images and 2D laser data, a direct solution was established to minimize the algebraic error, while they still needed several poses of planar pattern. Huang and Barth [12] first used chessboard to calibrate a multi-layer LiDAR and vision system. Vasconcelos et al. [13] formulated the problem as a standard P3P problem between the LiDAR and plane points by scanning the chessboard lines. Their method is more accurate than Zhang’s. Geiger et al. [14] arranged multiple chessboards in space to obtain enough constraint equations from a single shot. Zhou et al. [15] employed three line-to-plane correspondences, and then solved this problem with the algebraic structure of the polynomial system. Afterwards, they put forward their method based on the 3D line and plane correspondences and reduced the minimal number of chessboard poses [16].

However, the boundaries of the chessboard should be determined. Chai et al. [17] used ArUco marker, which is similar to the chessboard pattern, combined with a cube to solve the problem as a PnP problem. Surabhi Verma et al. [18] used 3D point and plane correspondences and genetic algorithm to solve the extrinsic parameters. An et al. [19] combined chessboard pattern with calibration objects to provide more point correspondences. However, those methods require the checkerboard pattern.

When spatial information is taken into account, some methods based on special calibration objects are proposed. Li et al. [20] provided a right-angled triangular checkerboard as calibration object. By using the line features on the object, the parameters can be solved. Willis et al. [21] used a sequence of rectangular boxes to calibrate a 2D LiDAR and a camera. However, the settings for the devices are demanding. Kwak et al. [22] extracted line and point features which are located on the boundaries and centerline of a v-shaped target. Then, they obtained the extrinsic parameters by minimizing reprojection error. Naroditsky et al. [23] used line features of a black line on a white sheet of paper. Fremont et al. [2] designed a circular target. In the LiDAR coordinate system, they used 1D edge detection to determine the border of the target and fitted the circle center and plane normal, but the size of the target needs to be known. Gomez-Ojeda et al. [3] presented a method that relies on an orthogonal trihedron, which is based on the line-to-plane and point-to-plane constraints. Pusztai et al. [24] used boxes with known sizes to calibrate the extrinsic parameters between a LiDAR and camera. Dong et al. [25] presented a method based on plane-line constraints of a v-shaped target composed of two noncoplanar triangles with checkerboard inside. The extrinsic parameters can be determined from single observation. These methods have high requirements for customized artificial calibration objects, and this may make them hard to be popularly adopted.

Some methods explored calibration methods without using artificial targets. These methods usually start with basic geometric information in a natural scene. Forkuo and King provided a point-based method [26] and further improved it [27], but the feature points are obtained by corner detector, which is not suitable for depth sensors with low resolution. Scaramuzza et al. [8] provided a calibration method based on manually selecting corresponding points. However, too many manual inputs will cause the results to become unstable. Mirzaei et al. [28] presented a line to line method by extracting the straight line structure. This algorithm is used for calibrating the extrinsic parameters of a single camera with known 3D lines, but it gives inspiration to the follow-up methods. Moghadam et al. [9] used 3D–2D line segment correspondences and nonlinear least square optimization to establish the method. This method performs well in indoor scenes, but, in outdoor scenes, the number of reliable 3D lines may not be adequate because of the viewing angle, low resolution of depth sensors, etc. This may lead to situations where many detected 2D lines cannot find their corresponding 3D counterparts. Levinson et al. [29] presented a method based on analyzing the edges on images and 3D points. This method only considers boundaries without extracting other available geometric information, and 3D point features may not be stable. Tamas and Kato [30] designed a method based on aligning 3D and 2D regions. The regions in 2D and 3D are separated by different segmentation algorithms, which may lead to inaccurate alignments of segmented regions and affect result accuracy. Pandey et al. [31] used reflectivity of LiDAR points and gray-scale intensity value of image pixels to establish constraints. By maximizing Mutual Information (MI), the extrinsic parameters can be estimated. Xiao et al. [32] solved the calibration problem by analyzing the SURF descriptor error of the projection of laser points among different frames. This method needs to input the transformation relationship among a large amount of images in advance. Jiang et al. [33] provided an online calibration method using road lines. They assumed that there are three lines which can be detected by both the camera and LiDAR on the road. This method is more similar to the following odometry-based methods and is suitable for automatic driving platform.

There are also some works based on other aspects (e.g., odometry and network). Bileschi [34] designed an automatic method to associate video and LiDAR data on a moving vehicle, but the initial relative pose between the sensors is provided by an inertial measurement unit (IMU). Schneider et al. [35] presented a target-less method based on sensor odometry for calibration.

After this, they further gave an end-to-end deep neural method to calculate the extrinsic parameters [6]. Taylor and Nieto [4] presented an approach for calibrating the extrinsic parameters among cameras, LiDARs, and inertial sensors based on motion. Gallego et al. [36] provided a tracking method based on event camera in high speed application environments, but this method requires special devices and is used in special circumstances. Park et al. [5] aligned the odometry of the LiDAR and camera to obtain a rough estimation of extrinsic parameters, and then refined the results jointly with time lag estimation. These odometry-based methods require continuous input to estimate sensor trajectory, which demands many data. Cumulative errors are still a problem for odometry-based methods. However, they can work in targetless environments and are able to calibrate the extrinsic parameters continuously. With the development of neural networks, several novel methods appear. Schneider et al. [6] offered RegNet, which is the first convolutional neural network to estimate extrinsic parameters between sensors. Iyer et al. [7] presented a self-supervised deep network named CalibNet. Considering the Riemannian geometry, Yuan et al. [37] recently designed RGGNet to estimate the offsets from initial parameters. Neural network-based methods need more data to train the networks, and the performance is closely related to the training data.

### 3. Method

Throughout this paper the LiDAR coordinate system is regarded as the world coordinate system. The translation relationship of one point  $X$  in the world coordinate system to the image point  $x$  is

$$x = \mathbf{K}[\mathbf{R}|\mathbf{t}]X, \quad (1)$$

where  $\mathbf{K}$  is the intrinsic matrix of the camera. It can be easily calibrated by traditional methods, e.g., Zhang's method [38]. We aim to estimate the extrinsic parameters, i.e., rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{t}$ . To solve this problem, it is obvious that we need to find some features which can be detected in both the LiDAR point clouds and images. Considering robustness and commonality, line feature is an appropriate choice. In this paper, we choose the corners of common buildings to illustrate our method because they usually have sharp edges and available line textures, but this method can also be applied to any object with similar features. Some appropriate building corners are shown in Figure 2. We define each spin of LiDAR as a frame. We also define a frame and its corresponding image as one dataset. It is recommended to keep the devices fixed while collecting a dataset to avoid the distortion brought by movement.



Figure 2. Some building corners in common scenes.

#### 3.1. Solve Rotation Matrix with Infinity Point Pairs

To solve rotation matrices, direction vector pairs are usually required. Many target-based methods use chessboards as the calibration object because it is convenient to get normal vectors of board planes in both the camera and the LiDAR coordinate systems. However, the board plane is small, and the LiDAR points on it are noisy, which makes it difficult to get sufficient results from a small number of data. Considering there are enough parallel lines in common scenes, we can obtain the vector pairs through the 3D–2D infinity point pairs based on line feature.



One bunch of 3D parallel lines intersect at the same infinity point  $\mathbf{X}^\infty$ , which lies on the infinite plane  $\Pi^\infty$  in the space. Since 3D parallel lines are no longer parallel after perspective transformation, the intersection point of their projection lines is written as  $\mathbf{x}^\infty = [x_1, x_2, 1]^T$ , which is not at infinity [39].  $\mathbf{X}^\infty$  and  $\mathbf{x}^\infty$  make up a 3D–2D infinity point pair. Here, we use L-CNN [40] to detect the edges on an image. A RANSAC procedure is used to detect the infinity points of artificial buildings from images as in [41–43]. Then, three bunches of lines and their intersection points  $\mathbf{x}_i^\infty, i = 1, 2, 3$  on an image plane as shown in Figure 3 can be obtained. From Equation (2), we can get three 3D unit vectors  $\mathbf{c}_1, \mathbf{c}_2$  and  $\mathbf{c}_3$ .

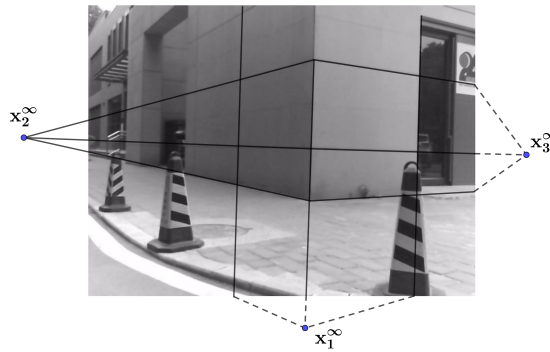


Figure 3. Corresponding points on an image plane.

$$\mathbf{c}_i = \text{norm}(\mathbf{K}^{-1}\mathbf{x}_i^\infty), \quad i = 1, 2, 3. \quad (2)$$

When setting up the devices, an initial guess of the camera optical axis and LiDAR orientation can be obtained, i.e., a coarse relative pose of the LiDAR and camera is known. In the LiDAR point cloud, the planes of a building corner can be separated by existing point cloud segmentation methods [44–46]. The three planes shown in Figure 4 can be extracted according to the known orientation. The RANSAC algorithm [47] is used to fit the extracted planes. Then, we can get their normal vectors  $\mathbf{n}_1, \mathbf{n}_2$  and  $\mathbf{n}_3$ :

$$\begin{aligned} A_1X + B_1Y + C_1Z + D_1 = 0, \quad \mathbf{n}_1 &= \begin{bmatrix} A_1 & B_1 & C_1 \end{bmatrix}^T \\ A_2X + B_2Y + C_2Z + D_2 = 0, \quad \mathbf{n}_2 &= \begin{bmatrix} A_2 & B_2 & C_2 \end{bmatrix}^T \\ A_3X + B_3Y + C_3Z + D_3 = 0, \quad \mathbf{n}_3 &= \begin{bmatrix} A_3 & B_3 & C_3 \end{bmatrix}^T \end{aligned} \quad (3)$$

$\mathbf{w}_1, \mathbf{w}_2$ , and  $\mathbf{w}_3$  are the normalized cross products of the plane normal vectors. They are the direction vectors of the 3D lines  $L_1, L_2$ , and  $L_3$ , as shown in Figure 4.

$$\begin{aligned} \mathbf{w}_1 &= \pm \text{norm}(\mathbf{n}_1 \times \mathbf{n}_2) \\ \mathbf{w}_2 &= \pm \text{norm}(\mathbf{n}_1 \times \mathbf{n}_3) \\ \mathbf{w}_3 &= \pm \text{norm}(\mathbf{n}_3 \times \mathbf{n}_2) \end{aligned} \quad (4)$$

Equation (5) shows the homogeneous form of the 3D infinity points in the LiDAR coordinate system. Notice that the direction (i.e., sign) of  $\mathbf{w}_i$  is still ambiguous, as shown in Equation (4). If  $\mathbf{c}_i$  is determined from an image, the direction it represents in the environment is roughly obtained. Then, its paired  $\mathbf{w}_i$  can be chosen by this condition, and the sign of  $\mathbf{w}_i$  can also be determined (consistent with the direction of  $\mathbf{c}_i$ ). Figure 5 shows the relationship between the two coordinate systems.

$$\mathbf{X}_1^\infty = \begin{bmatrix} \mathbf{w}_1 \\ 0 \end{bmatrix}, \mathbf{X}_2^\infty = \begin{bmatrix} \mathbf{w}_2 \\ 0 \end{bmatrix}, \mathbf{X}_3^\infty = \begin{bmatrix} \mathbf{w}_3 \\ 0 \end{bmatrix} \quad (5)$$

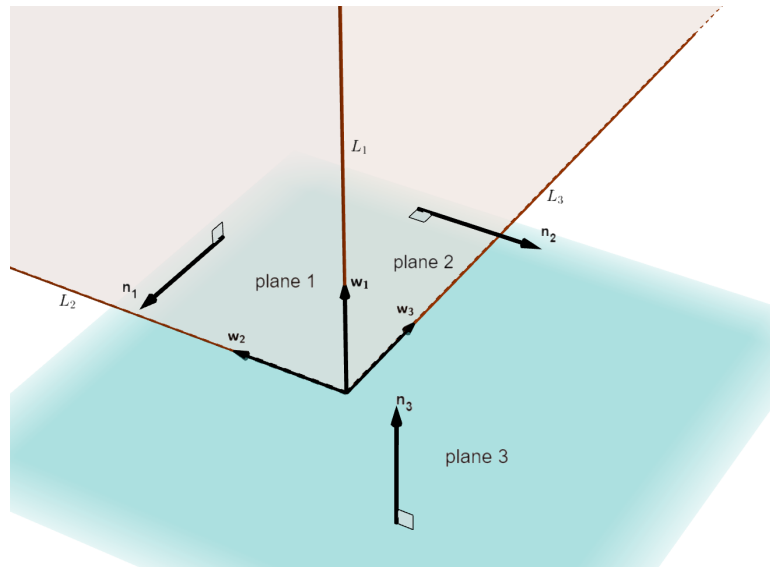


Figure 4. Geometric model in the LiDAR coordinate system.

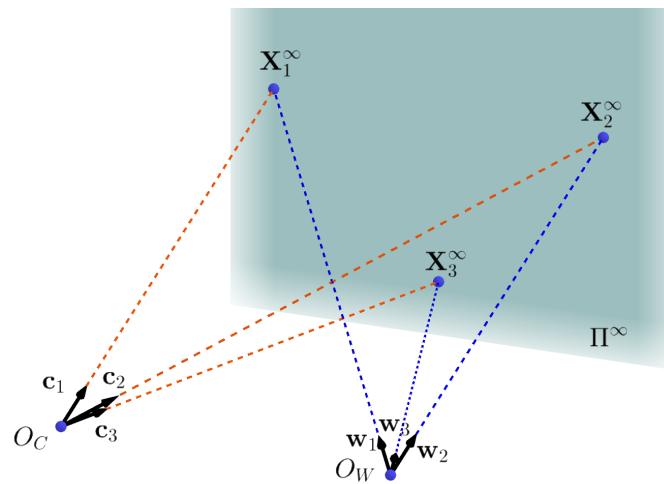


Figure 5. Geometric relationship between two coordinate systems.  $O_C$  is the origin of the camera coordinate system.  $O_W$  is the origin of LiDAR coordinate system.  $\Pi^\infty$  is the infinity plane in the space.  $X_1^\infty$ ,  $X_2^\infty$ , and  $X_3^\infty$  are 3D infinity points in the world coordinate system. Actually,  $c_i$  and  $w_i$  coincide in the space.

Then, the direction vector pairs made up of  $c_i$  and  $w_i$  are obtained. The rotation matrix can be solved in close form with at least two pairs of direction vectors [48,49]. Assume there is a unit direction vector  $w$  in the world coordinate system. The relationship between  $w$  and its paired vector  $c$  is  $c = R w$ . For the  $i$  th pair, we have  $c_i = R w_i$ . Let

$$A = \sum_{i=1}^n w_i c_i^T. \tag{6}$$

Applying singular value decomposition to  $A$ , we have  $A = U D V^T$  and  $R = V U^T$ . The minimum number of  $n$  to determine  $R$  is 2.

Furthermore, if we have three or more vector pairs,  $R$  can also be solved in a simpler way:

$$R = \begin{cases} C W^{-1} & n = 3 \\ (C W^T)(W W^T)^{-1} & n > 3 \end{cases} \tag{7}$$

where  $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n]$ ,  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$ . The rank of matrix  $\mathbf{W}$  must be greater or equal to 3 in this equation. Before computing  $\mathbf{R}$ , the 3D infinity points should be checked to ensure that at least three directions in the space are selected. The solved  $\mathbf{R}$  may not be orthogonal due to noise. To keep  $\mathbf{R}$  as an orthogonal matrix, let  $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$  be

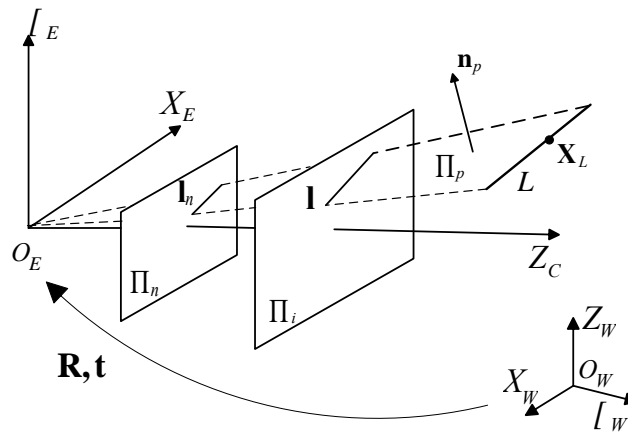
$$\begin{cases} \mathbf{r}_1 = \mathbf{r}_1 / \text{norm}(\mathbf{r}_1), \\ \mathbf{r}_2 = \mathbf{r}_2 / \text{norm}(\mathbf{r}_2), \\ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2. \end{cases} \quad (8)$$

### 3.2. Solve Translation Vector

The method presented above allows us to estimate rotation matrix  $\mathbf{R}$  without considering translation vector  $\mathbf{t}$ . Taking  $\mathbf{R}$  as a known factor, here we use a linear method to get  $\mathbf{t}$ .

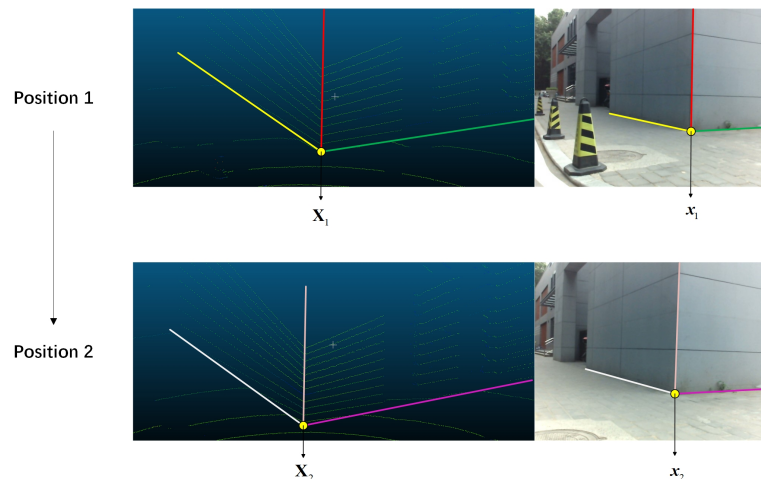
Assume that there is a 3D point  $\mathbf{X}_L$  located on the line  $L$  in the LiDAR coordinate system as shown in Figure 6.  $\mathbf{l}_n : a_n x + b_n y + c_n = 0$  and  $\mathbf{l} : ax + by + c = 0$  are the projections of  $L$  on the normalized image plane  $\Pi_n$  and image plane  $\Pi_i$ , respectively.  $\mathbf{n}_p = [a_n, b_n, c_n]^T$  is the normal vector of the interpretation plane  $\Pi_p$ . We can obtain  $\mathbf{n}_p$  easily from  $\mathbf{l}$  through the known intrinsic matrix of the camera. For each pair of corresponding lines, we have one equation for  $\mathbf{t}$  [50]:

$$\mathbf{n}_p^T [\mathbf{R} | \mathbf{t}] \mathbf{X}_L = 0. \quad (9)$$



**Figure 6.** Geometry of camera projection model.  $\Pi_i$  is the image plane,  $\Pi_n$  is the normalized image plane, and  $\Pi_p$  is the interpretation plane.

With the fitted planes in Equation (3), the 3D wall intersection lines  $L_1$ ,  $L_2$ , and  $L_3$  are easy to obtain, as is the 3D point  $\mathbf{X}_L$ , which lies on the 3D line. We choose  $L_i$  in the LiDAR coordinate system and  $\mathbf{l}_i$  on the image as corresponding line pairs. In general case, if three different sets of corresponding line pairs are known, the translation vector  $\mathbf{t}$  can be solved. However, in our scene,  $\mathbf{l}_1$ ,  $\mathbf{l}_2$ , and  $\mathbf{l}_3$  intersect at the same point on the image plane. This leads to a case that the three equations based on line constraint are not independent [51]. Thus, the equation factor matrix cannot be full rank, and this makes it difficult to solve  $\mathbf{t}$  from a single dataset. To avoid this, we choose to move the LiDAR and camera and use at least two datasets to compute  $\mathbf{t}$ . An example is shown in Figure 7, and then we can choose any three disjoint lines from these data to solve the problem.



**Figure 7.** Two datasets from different positions. The lines marked with the same color correspond.

### 3.3. Optimization

To make the results more accurate, the minimum of  $\mathbf{R}$  and  $\mathbf{t}$  under some constraints needs to be found. In this part, we construct cost function and then minimize the reprojection error to optimize  $\mathbf{R}$  and  $\mathbf{t}$ . We use the intersection of lines to establish the constraint. Assume that we have collected  $n$  ( $n \geq 2$ ) datasets. For each set, we can get the intersection points  $\mathbf{x}$  and  $\mathbf{X}$  of the 2D and 3D lines, as shown in Figure 7. The cost function is:

$$\min \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_i\|. \quad (10)$$

We solve it by nonlinear optimization methods, such as the Levenberg–Marquard (LM) algorithm [52]. For the initial solutions with very low accuracy, we regard them as outliers and reject them before optimizing. The filter procedure is based on the RANSAC algorithm; a distance threshold for the reprojection error is set to distinguish the initial solutions. In this way, we can solve  $\mathbf{R}$  and  $\mathbf{t}$  and remove the influence from noise as much as possible. The complete process for the algorithm is described in Algorithm 1.

---

#### Algorithm 1:

---

**Input:**  $n$  datasets ( $n \geq 2$ ) and camera intrinsic matrix  $\mathbf{K}$ .

**Output:**  $\mathbf{R}$ ,  $\mathbf{t}$ .

1. Process all point cloud data to obtain the plane equation, unit vector  $\mathbf{w}$ , intersection point  $\mathbf{X}$  of the 3D lines and point  $\mathbf{X}_L$  on the 3D plane intersection line.
2. Process all images to obtain the plane intersection line  $\mathbf{l}$ , unit vector  $\mathbf{c}$  and the intersection point  $\mathbf{x}$ .

**repeat**

3. Select two datasets randomly to form a sub-scene  $S_1$ .
4. Solve initial  $\mathbf{R}$  and  $\mathbf{t}$  by  $S_1 = \{ \langle \mathbf{c}, \mathbf{w} \rangle^i, \langle \mathbf{X}_L, \mathbf{l} \rangle^i | i \in [1, 2] \}$
- for**  $j = 1$  to  $n$  **do**
  5.  $d = \|\mathbf{x}_j - \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_j\|$ .
  6. **if**  $d < \text{Distance Threshold}$  **then**
    - └ Add  $i$ th dataset to  $S_2$ .  $S_2 = \{ \langle \mathbf{X}, \mathbf{x} \rangle^j | j \in [1, 2, \dots, M_k] \}$

**until**  $M_k > \text{Number Threshold}$ ;

7.  $S_2$  is the inlier set. It is used to optimize  $\mathbf{R}$  and  $\mathbf{t}$  based on Equation (10) to obtain the final results.
-

Similar to Fremont's work [2], we estimate the precision of the calibration solution by the Student's  $t$ -distribution. The covariance matrix  $\mathbf{C}_{ov}$  of the estimated parameters is defined as follows:

$$\mathbf{C}_{ov} = \epsilon^2 \cdot (\mathbf{J}^T \mathbf{J})^{-1} \quad (11)$$

where  $\epsilon^2$  is an unbiased estimate of the variance and  $\mathbf{J}$  is the Jacobian matrix of the last LM algorithm iteration. Then, the width of confidence interval is given by:

$$\delta C_i = t_{value} \cdot \sqrt{\mathbf{C}_{ov}(i, i)} \quad (12)$$

where  $\sqrt{\mathbf{C}_{ov}(i, i)}$  is the standard deviation of the  $i$ th parameter.  $t_{value}$  is determined by the degrees of freedom of the Student's  $t$ -distribution and confidence (e.g., 95%).

#### 4. Experiments

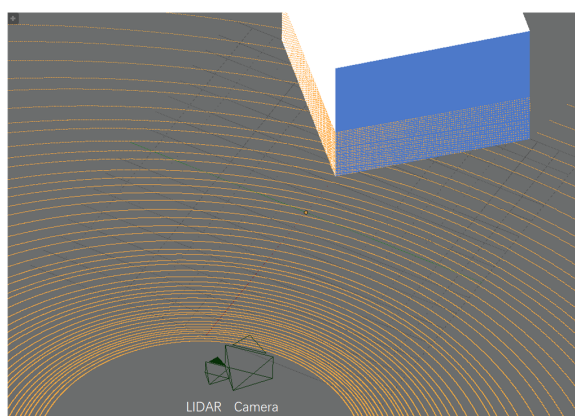
We conducted two experiments to verify our method. The first one was set up from simulated data to prove the veracity of our method and evaluate its robustness to noise. The second one used real outdoor data collected by a Leishen C16 LiDAR Scanner and a stereo camera system. We regard the transformation between the two cameras as ground truth. By comparing it with our results, we can quantify the accuracy of our method in real environments.

##### 4.1. Simulated Data

We used Blensor [53] to create simulated data. It is a sensor simulation package with different kinds of devices. In this experiment, we set Velodyne HDL-64E as the LiDAR, which works at 24 Hz with an angle resolution of  $0.17^\circ$ . The resolution of the virtual camera was set to  $1920 \times 1080$  pixels, and its focal length was 30 mm. The intrinsic parameters are shown in Table 1. We established a scene and set up a virtual LiDAR-Camera pair to collect data, as shown in Figure 8.

**Table 1.** Intrinsic parameters of virtual camera.

$f_x$	$f_y$	$c_x$	$c_y$
1800	1800	960	540



**Figure 8.** Simulation scene in Blensor. The yellow lines are LiDAR scan lines.

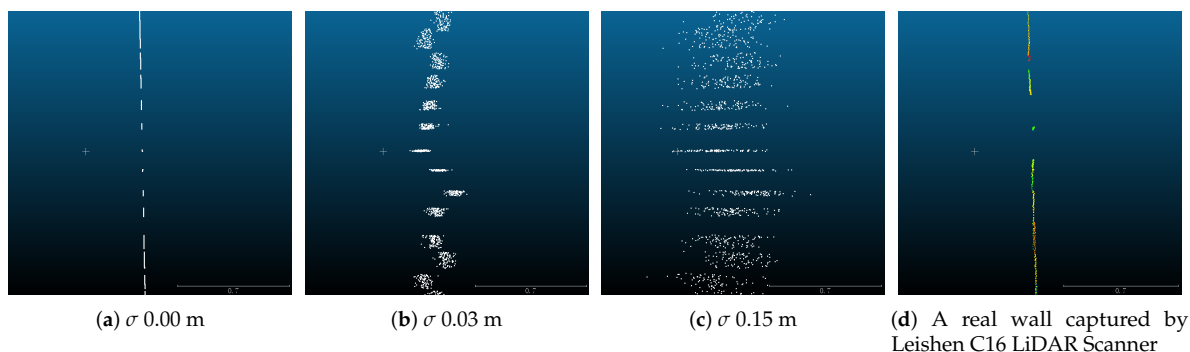
The translation vector between the virtual LiDAR and camera was set to  $\hat{\mathbf{t}} = [-1, 0, 0]^T$  in meters; the rotation matrix was  $\hat{\mathbf{R}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$ ; and Gaussian noise was added to the LiDAR point cloud to verify our method and test the robustness. The standard deviation  $\sigma$  was set to vary from 0.00 to



0.15 m. Figure 9 shows the effects from noise to point clouds. For each level of noise, we collected 10 datasets from different poses by moving the LiDAR-Camera pair. We first randomly chose 2 datasets to compute an initial solution and used the other 8 sets to optimize. We then repeated this procedure 100 times. The average of the 100 results was regarded as the extrinsic transformation under this noise level. Then, we calculated the error of rotation and translation for all noise levels:

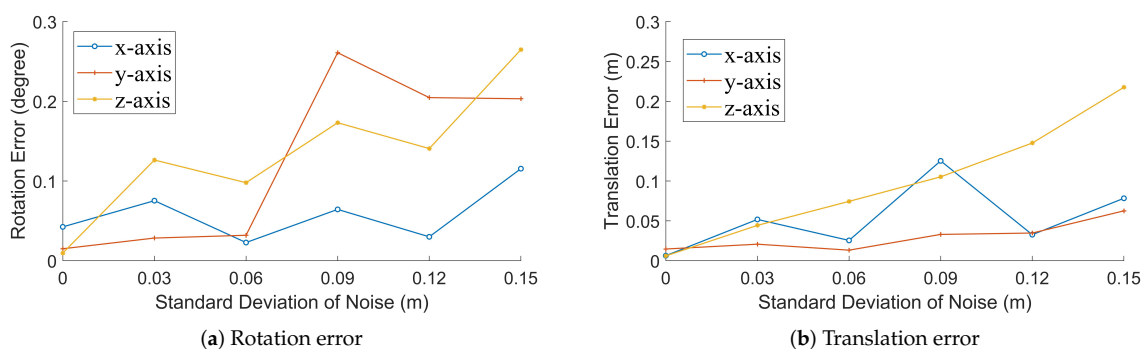
$$\begin{aligned} \mathbf{e}_R &= |\boldsymbol{\theta}_R - \boldsymbol{\theta}_{\hat{R}}| \\ \mathbf{e}_t &= |\mathbf{t} - \hat{\mathbf{t}}| \end{aligned} \quad (13)$$

where  $[\mathbf{R}|\mathbf{t}]$  is the estimated transformation and  $[\hat{\mathbf{R}}|\hat{\mathbf{t}}]$  is the ground truth.  $\boldsymbol{\theta} = [\theta_{x-axis}, \theta_{y-axis}, \theta_{z-axis}]^T$  is the Euler angle form of the rotation matrix.



**Figure 9.** Side view of a wall to show noise effects on noise on point clouds. The measurement scale of the four maps is the same. The ruler at the right bottom of the images is measured in meters. The scan points on planes becomes quite noisy when  $\sigma$  becomes closer to 0.15 m.

Figure 10 shows the rotation and translation errors in our method. With the increase in noise, errors also increase. However, the rotation error of a single axis does not exceed  $0.3^\circ$ , and the translation error is still lower than 30 cm when  $\sigma = 0.15$  m. This shows that our method can provide a stable and accurate solution.



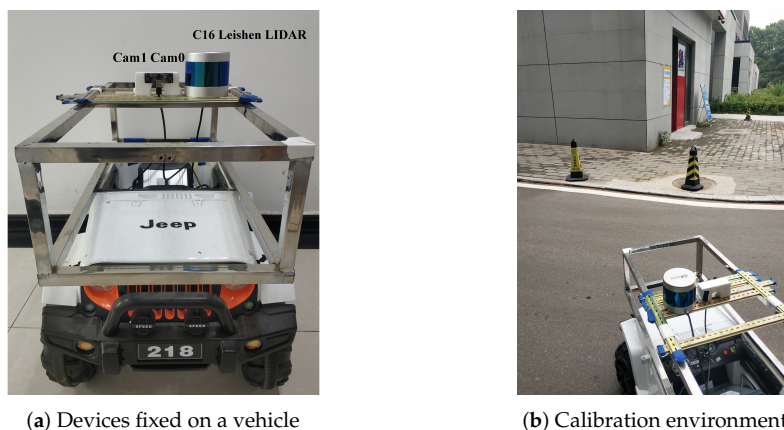
**Figure 10.** Errors for simulated data with different noise level.

#### 4.2. Real Data

In this experiment, we utilized a stereo camera system and a Leishen C16 LiDAR Scanner to capture point clouds. Generally, our method does not require a second camera. To evaluate our method quantitatively, we utilized a stereo camera system with pre-calibrated extrinsic parameters between the two cameras. The proposed method can calibrate the extrinsic parameters between a LiDAR and a single camera. When using a stereo camera system, we can calibrate two pairs of extrinsic parameters between the LiDAR and two cameras separately, and then we can estimate the extrinsic

parameters between the two cameras from them. We regarded the pre-calibrated parameters as ground truth. By comparing the estimated results and the ground truth, we could analyze the accuracy of our method. A comparison with Pandey’s method [31] is also given.

The LiDAR works at 10 Hz with an angle resolution of  $0.18^\circ$ . The two cameras have a resolution of  $640 \times 480$ , and the relative pose between the LiDAR and stereo camera system is fixed. Figure 11a shows the devices, while Figure 11b shows the scene. The stereo camera system was pre-calibrated through Zhang’s method, and the intrinsic parameters are shown in Table 2. In the mean time, the extrinsic parameters were also determined, as shown in Table 3.



**Figure 11.** (a) Stereo camera system and C16 Leishen LiDAR Scanner. The two cameras provide ground truth parameters. (b) The environment of calibration.

**Table 2.** Intrinsic parameters of stereo cameras.

	$f_x$	$f_y$	$c_x$	$c_y$
Cam0	759.377	759.791	352.516	237.499
Cam1	764.215	764.137	318.168	257.122

**Table 3.** Calibrated extrinsic parameters from Cam0 to Cam1.

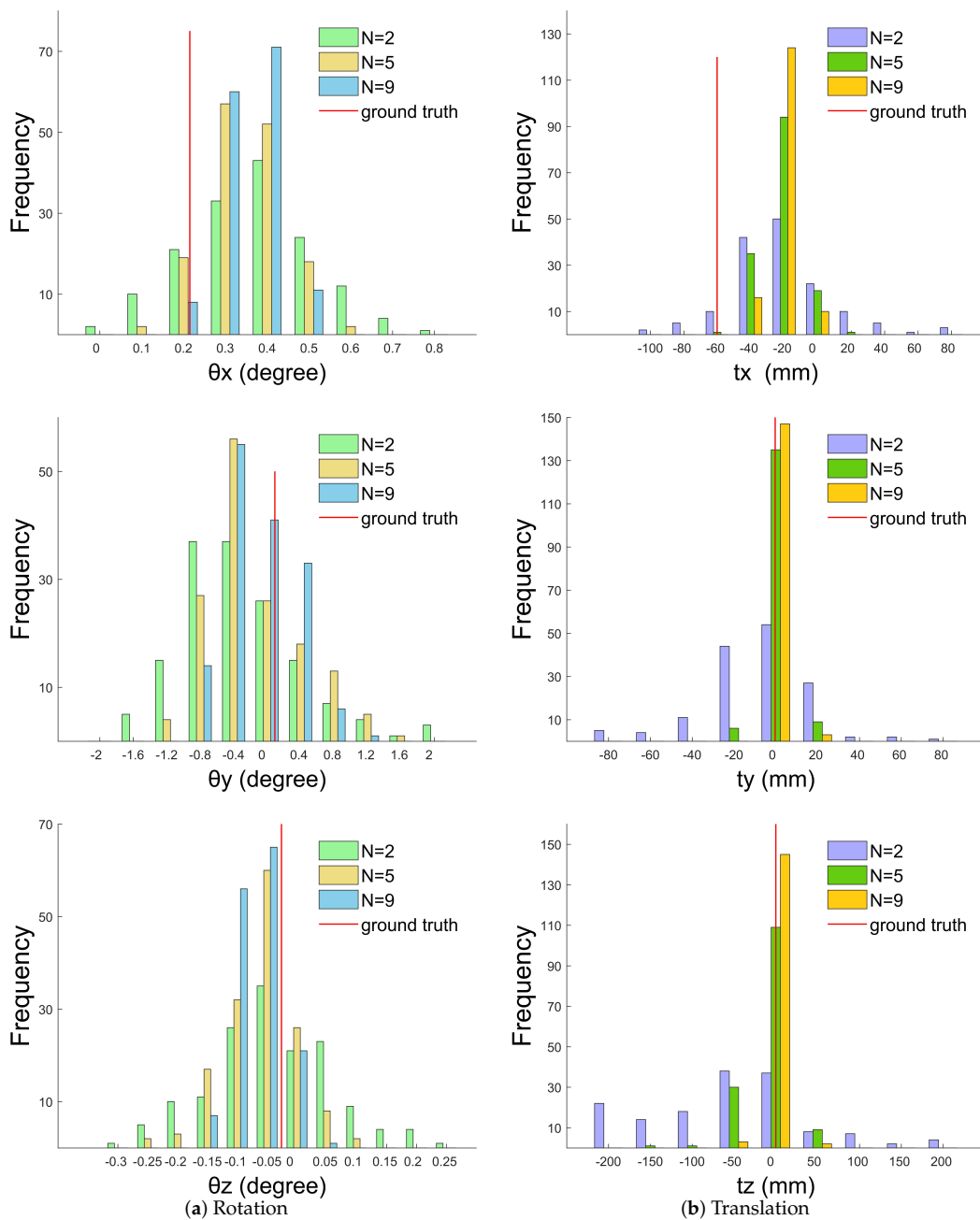
$\theta_x$	$\theta_y$	$\theta_z$
$0.215^\circ$	$0.092^\circ$	$-0.026^\circ$
$t_x$	$t_y$	$t_z$
$-60.088$ mm	$-0.300$ mm	$0.217$ mm

In the calibration scene, we placed the vehicle in front of a building corner and moved it in any direction 25 times. Each pose provided dataset. From any 2 of them, we could get initial extrinsic parameters  $R_{Lto0}$  and  $\mathbf{t}_{Lto0}$  ( $R_{Lto1}$  and  $\mathbf{t}_{Lto1}$ ) from LiDAR to Cam0 (Cam1). Then, we could get the extrinsic parameters  $R_{0to1}$  and  $\mathbf{t}_{0to1}$  from Cam0 to Cam1 through a simple transformation:

$$R_{0to1} = R_{Lto1}R_{Lto0}^{-1} \quad (14)$$

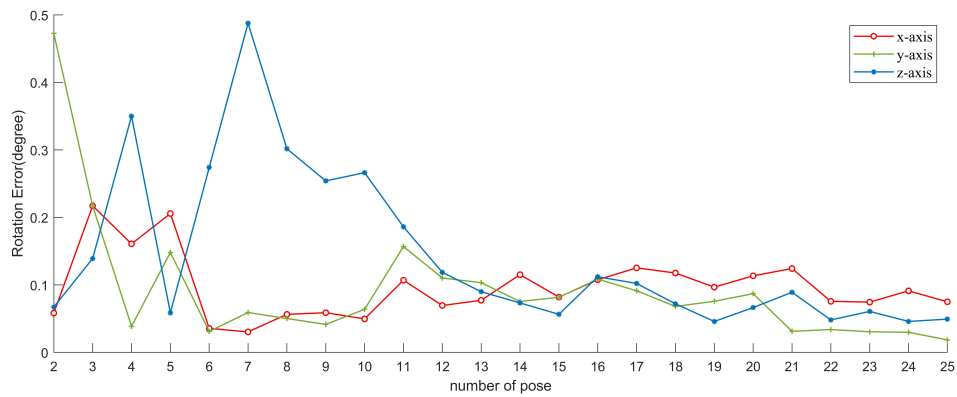
$$\mathbf{t}_{0to1} = \mathbf{t}_{Lto1} - R_{0to1}\mathbf{t}_{Lto0} \quad (15)$$

We randomly chose  $N(N > 2)$  sets from among the 25 sets 150 times so that 150 initial extrinsic solutions from Cam0 to Cam1 were created. Let  $N = 2, 5, 9$ ; the distribution of initial extrinsic solutions is shown in Figure 12. It is clear that, with the increase of  $N$ , the data distribution gradually improved and tended to be stable. The red line in the graph represents ground truth. After comparing the initial solutions with the real value, it is clear that optimization is still required to refine the results.

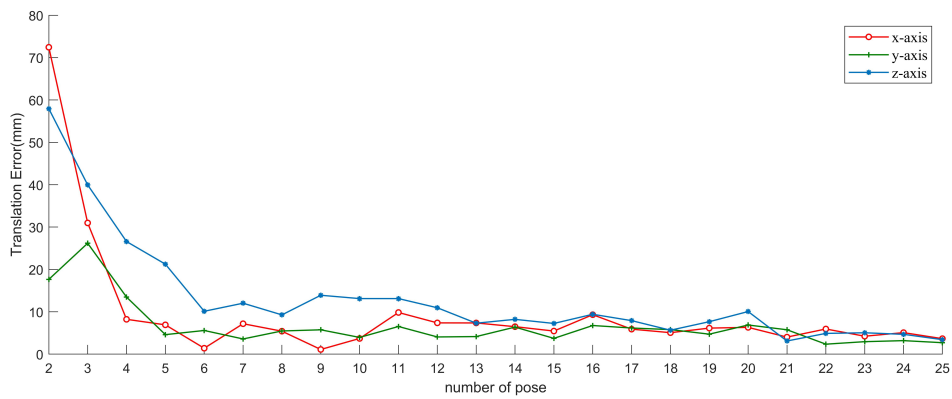


**Figure 12.** Distribution of initial extrinsic parameter solutions.  $N$  represents the number of poses we use to get an initial solution.

The rotation and translation errors after optimization are shown in Figure 13. In the beginning, the errors are big and unstable. When more poses are used to optimize, the total error for the three axes decreases. After about 10 poses, the results become stable, and the translation error gradually decreases to below 1 cm. The result solved from 25 poses is shown in Table 4; the confidence interval is calculated with confidence 95%.



(a) Rotation error



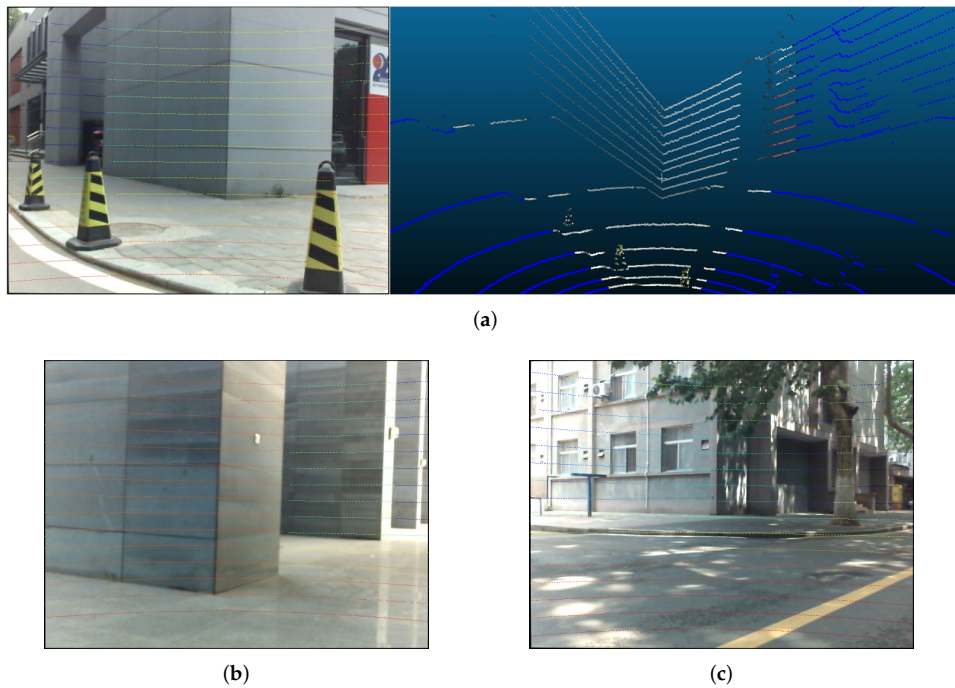
(b) Translation error

Figure 13. Errors for real data with respect to the number of poses we used.

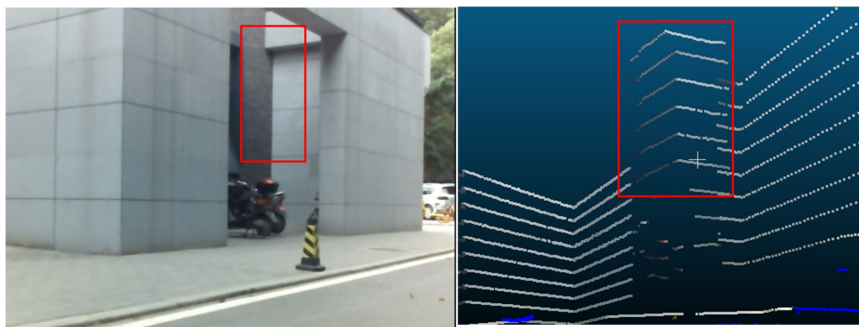
Table 4. Calibration result obtained with 25 poses.

	LiDAR to Cam0	Confidence	LiDAR to Cam1	Confidence
Translation (m)				
$t_x$	-0.1041	$\pm 0.0078$	-0.1622	$\pm 0.0077$
$t_y$	-0.0324	$\pm 0.0093$	-0.0297	$\pm 0.0087$
$t_z$	-0.0211	$\pm 0.0154$	-0.0177	$\pm 0.0145$
Rotation (axis-angle)				
$r_x$	1.5549	$\pm 0.0019$	1.5600	$\pm 0.0018$
$r_y$	-0.0292	$\pm 0.0032$	-0.0294	$\pm 0.0034$
$r_z$	0.0495	$\pm 0.0037$	0.0473	$\pm 0.0038$

The visualized results are shown in Figure 14. The colors of the projected points in Figure 14a,b are determined according to distance. The misalignments of some points (such as the blue and cyan ones on the traffic cone in Figure 14a) are caused by occlusion, because the points are observed by the LiDAR but not by the camera. In Figure 15, we can also intuitively observe the accuracy of the colorized results. The pose between the LiDAR and camera is well estimated.



**Figure 14.** Visualized results: (a) a calibration scene, where the left part shows the point projection on the image captured by Cam0 and the right part shows the corresponding point cloud with extracted color information from the image; and (b,c) the projection result of other scenes using the same extrinsic parameters as (a).



**Figure 15.** Detail of a colored point cloud: the color of the wall changes correctly at the edge. The left and right part shows the image and the colored point cloud, respectively.

To further show the performance, we compared Pandey's method using the same 25 datasets. By calibrating the extrinsic parameters from the LiDAR to Cam0 and Cam1, respectively, the relative pose between Cam0 and Cam1 was estimated. The rotation and translation errors of Pandey's method and ours are shown in Table 5. Our proposed method performs better due to the use of more geometric constraints of artificial buildings.

**Table 5.** Rotation and translation error (25 poses).

	$t_x$	$t_y$	$t_z$	$r_x-axis$	$r_y-axis$	$r_z-axis$
Pandey [31]	0.0188 m	0.0305 m	0.0073 m	0.4476°	0.8375°	0.4839°
proposed	0.0020 m	0.0028 m	0.0032 m	0.0771°	0.0138°	0.0529°

## 5. Conclusions

In this paper, we present a LiDAR-Camera extrinsic calibration method without preparing specific calibration object. We start with obtaining the 3D infinity points from the point cloud. Because there



are sufficient parallel lines in the scene, we can obtain their corresponding 2D infinity points on the image. By obtaining the direction vectors from the 2D infinity points and aligning them to the 3D ones, we can solve the rotation matrix. Next,  $\mathbf{t}$  can also be solved by point-on-line constraint linearly. Experiments show that our algorithm can calibrate the extrinsic parameters between camera and LiDAR with accuracy in common outdoor scenes. Meanwhile, the algorithm can also be applied to any scene with similar parallel line features.

**Author Contributions:** Conceptualization, Z.B. and G.J.; Data curation, Z.B. and A.X.; Formal analysis, A.X.; Methodology, Z.B. and G.J.; Project administration, G.J.; Resources, A.X.; Software, Z.B. and A.X.; Supervision, G.J.; Validation, Z.B. and A.X.; Visualization, Z.B. and A.X.; Writing—original draft, Z.B. and A.X.; and Writing—review and editing, Z.B. and G.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Zhang, Q.; Pless, R. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2301–2306.
- Fremont, V.; Rodriguez Florez, S.A.; Bonnifait, P. Circular targets for 3d alignment of video and lidar sensors. *Adv. Robot.* **2012**, *26*, 2087–2113.
- Gomez-Ojeda, R.; Briaies, J.; Fernandez-Moral, E.; Gonzalez-Jimenez, J. Extrinsic calibration of a 2D laser-rangefinder and a camera based on scene corners. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3611–3616.
- Taylor, Z.; Nieto, J. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Trans. Robot.* **2016**, *32*, 1215–1229.
- Park, C.; Moghadam, P.; Kim, S.; Sridharan, S.; Fookes, C. Spatiotemporal camera-LiDAR calibration: A targetless and structureless approach. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1556–1563.
- Schneider, N.; Piewak, F.; Stiller, C.; Franke, U. RegNet: Multimodal sensor registration using deep neural networks. In Proceedings of the 2017 IEEE intelligent vehicles symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1803–1810.
- Iyer, G.; Ram, R.K.; Murthy, J.K.; Krishna, K.M. CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1110–1117.
- Scaramuzza, D.; Harati, A.; Siegwart, R. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 4164–4169.
- Moghadam, P.; Bosse, M.; Zlot, R. Line-based extrinsic calibration of range and image sensors. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3685–3691.
- Xu, C.; Zhang, L.; Cheng, L.; Koch, R. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1209–1222. [[PubMed](#)]
- Wang, P.; Xu, G.; Cheng, Y. A novel algebraic solution to the perspective-three-line pose problem. *Comput. Vis. Image Underst.* **2020**, *191*, 102711.
- Huang, L.; Barth, M. A novel multi-planar LIDAR and computer vision calibration procedure using 2D patterns for automated navigation. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 117–122.
- Vasconcelos, F.; Barreto, J.P.; Nunes, U. A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2097–2107.
- Geiger, A.; Moosmann, F.; Car, Ö.; Schuster, B. Automatic camera and range sensor calibration using a single shot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 3936–3943.

15. Zhou, L. A new minimal solution for the extrinsic calibration of a 2D LIDAR and a camera using three plane-line correspondences. *IEEE Sens. J.* **2013**, *14*, 442–454.
16. Zhou, L.; Li, Z.; Kaess, M. Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5562–5569.
17. Chai, Z.; Sun, Y.; Xiong, Z. A Novel Method for LiDAR Camera Calibration by Plane Fitting. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 286–291.
18. Verma, S.; Berrio, J.S.; Worrall, S.; Nebot, E. Automatic extrinsic calibration between a camera and a 3D Lidar using 3D point and plane correspondences. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 9–12 July 2019; pp. 3906–3912.
19. An, P.; Ma, T.; Yu, K.; Fang, B.; Zhang, J.; Fu, W.; Ma, J. Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences. *Opt. Express* **2020**, *28*, 2122–2141.
20. Li, G.; Liu, Y.; Dong, L.; Cai, X.; Zhou, D. An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3854–3859.
21. Willis, A.R.; Zapata, M.J.; Conrad, J.M. A linear method for calibrating LIDAR-and-camera systems. In Proceedings of the 2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, London, UK, 21–23 September 2009; pp. 1–3.
22. Kwak, K.; Huber, D.F.; Badino, H.; Kanade, T. Extrinsic calibration of a single line scanning lidar and a camera. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, London, UK, 21–23 September 2011; pp. 3283–3289.
23. Naroditsky, O.; Patterson, A.; Daniilidis, K. Automatic alignment of a camera with a line scan lidar system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3429–3434.
24. Pusztai, Z.; Hajder, L. Accurate calibration of LiDAR-camera systems using ordinary boxes. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 394–402.
25. Dong, W.; Isler, V. A novel method for the extrinsic calibration of a 2D laser rangefinder and a camera. *IEEE Sens. J.* **2018**, *18*, 4200–4211.
26. Forkuo, E.; King, B. Registration of Photogrammetric Imagery and Laser Scanner Point Clouds. In Proceedings of the Mountains of data, peak decisions, 2004 ASPRS Annual Conference, Denver, CO, USA, 23–28 May 2004, p. 58.
27. Forkuo, E.K.; King, B. Automatic fusion of photogrammetric imagery and laser scanner point clouds. *Int. Arch. Photogramm. Remote Sens.* **2004**, *35*, 921–926.
28. Mirzaei, F.M.; Roumeliotis, S.I. Globally optimal pose estimation from line correspondences. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5581–5588.
29. Levinson, J.; Thrun, S. Automatic Online Calibration of Cameras and Lasers. In Proceedings of the 2013 MIT Press Robotics: Science and Systems, Berlin, Germany, 24–28 June 2013; Volume 2.
30. Tamas, L.; Kato, Z. Targetless calibration of a lidar-perspective camera pair. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 1–8 December 2013; pp. 668–675.
31. Pandey, G.; McBride, J.R.; Savarese, S.; Eustice, R.M. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. *J. Field Robot.* **2015**, *32*, 696–722.
32. Xiao, Z.; Li, H.; Zhou, D.; Dai, Y.; Dai, B. Accurate extrinsic calibration between monocular camera and sparse 3D lidar points without markers. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 424–429.
33. Jiang, J.; Xue, P.; Chen, S.; Liu, Z.; Zhang, X.; Zheng, N. Line feature based extrinsic calibration of LiDAR and camera. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Los Angeles, CA, USA, 11–14 June 2018; pp. 1–6.

34. Bileschi, S. Fully automatic calibration of lidar and video streams from a vehicle. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1457–1464.
35. Schneider, S.; Luettel, T.; Wuensche, H.J. Odometry-based online extrinsic sensor calibration. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1287–1292.
36. Gallego, G.; Lund, J.E.; Mueggler, E.; Rebecq, H.; Delbruck, T.; Scaramuzza, D. Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2402–2412.
37. Yuan, K.; Guo, Z.; Wang, Z.J. RGGNet: Tolerance Aware LiDAR-Camera Online Calibration with Geometric Deep Learning and Generative Model. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6956–6963.
38. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334.
39. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
40. Zhou, Y.; Qi, H.; Ma, Y. End-to-end wireframe parsing. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 962–971.
41. Rother, C. A new approach to vanishing point detection in architectural environments. *Image Vis. Comput.* **2002**, *20*, 647–655.
42. Tardif, J.P. Non-iterative approach for fast and accurate vanishing point detection. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 1250–1257.
43. Zhai, M.; Workman, S.; Jacobs, N. Detecting vanishing points using global image context in a non-manhattan world. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5657–5665.
44. Nurunnabi, A.; Belton, D.; West, G. Robust segmentation in laser scanning 3D point cloud data. In Proceedings of the 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA), Fremantle, WA, Australia, 3–5 December 2012; pp. 1–8.
45. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100.
46. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds. *Remote Sens.* **2016**, *8*, 5.
47. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
48. Zhang, C.; Zhang, Z. Calibration between depth and color sensors for commodity depth cameras. In *Computer Vision and Machine Learning with RGB-D Sensors*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 47–64.
49. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 698–700.
50. Zhang, X.; Zhang, Z.; Li, Y.; Zhu, X.; Yu, Q.; Ou, J. Robust camera pose estimation from unknown or known line correspondences. *Appl. Opt.* **2012**, *51*, 936–948.
51. Dhome, M.; Richetin, M. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 1265–1278.
52. Brown, K.M.; Dennis, J. Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation. *Numer. Math.* **1971**, *18*, 289–297.
53. Blender Sensor Simulation. Available online: <https://www.blensor.org/> (accessed on 26 May 2020).

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).