remote sensing

MDPI

Article

Automatic Calibration between Multi-Lines LiDAR and Visible Light Camera Based on Edge Refinement and Virtual Mask Matching

Chengkai Chen [1,2,†], Jinhui Lan [1,2,†], Haoting Liu [1,2,*], Shuai Chen [1,2] and Xiaohan Wang [1,2]

[1] Shunde Innovation School, University of Science and Technology Beijing, Foshan 528300, China
[2] Beijing Engineering Research Center of Industrial Spectrum Imaging, School of Automation and Electrical Engineering, University of Science and Technology, Beijing 100083, China
* Correspondence: liuhaoting@ustb.edu.cn
† These authors contributed equally to this work.

Abstract: To assist in the implementation of a fine 3D terrain reconstruction of the scene in remote sensing applications, an automatic joint calibration method between light detection and ranging (LiDAR) and visible light camera based on edge points refinement and virtual mask matching is proposed in this paper. The proposed method is used to solve the problem of inaccurate edge estimation of LiDAR with different horizontal angle resolutions and low calibration efficiency. First, we design a novel calibration target, adding four hollow rectangles for fully automatic locating of the calibration target and increasing the number of corner points. Second, an edge refinement strategy based on background point clouds is proposed to estimate the target edge more accurately. Third, a two-step method of automatically matching between the calibration target in 3D point clouds and the 2D image is proposed. Through this method, i.e., locating firstly and then fine processing, corner points can be automatically obtained, which can greatly reduce the manual operation. Finally, a joint optimization equation is established to optimize the camera's intrinsic and extrinsic parameters of LiDAR and camera. According to our experiments, we prove the accuracy and robustness of the proposed method through projection and data consistency verifications. The accuracy can be improved by at least 15.0% when testing on the comparable traditional methods. The final results verify that our method is applicable to LiDAR with large horizontal angle resolutions.

Keywords: LiDAR; visible light camera; edge refinement; virtual mask; automatic calibration

check for updates

Citation: Chen, C.; Lan, J.; Liu, H.; Chen, S.; Wang, X. Automatic Calibration between Multi-Lines LiDAR and Visible Light Camera Based on Edge Refinement and Virtual Mask Matching. *Remote Sens.* **2022**, *14*, 6385. https://doi.org/10.3390/rs14246385

Academic Editor: Jorge Delgado García

Received: 26 September 2022
Accepted: 8 December 2022
Published: 17 December 2022

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

Copyright: © 2022 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

At present, traditional large-scale digital map technology has been well developed and applied, such as Google Maps, Baidu Maps, etc. However, the work related to fine terrain reconstruction in a small scale is relatively limited. In order to obtain the fine 3D topographic map in a small scale, the simplest way is to use unmanned aerial vehicles (UAVs) to reconstruct terrain using the tilt photography technology. Tilt photography technology uses pure 2D image analysis and modeling to restore the 3D structure from 2D information, and its calculation accuracy is directly affected by image quality and imaging environment. Recently, more and more research has been conducted on the fine 3D terrain reconstruction technology using UAVs equipped with light detection and ranging (LiDAR) and visible light cameras. The visible light camera can obtain high-resolution color information, but it is particularly vulnerable to the influence of external weather, illumination, and other factors; it also lacks the 3D information of the target. LiDAR can quickly obtain the 3D information of space objects, but it cannot get the texture, color, and other information of objects. Therefore, the LiDAR and visible light camera can achieve excellent complementary effects in performance. This will greatly improve the performance of existing UAVs 3D terrain reconstruction or low-altitude remote sensing works [1–7].

However, since the data obtained by the two sensors are based on their respective coordinate systems, and data fusion requires that the data collected by two sensors should be expressed in LiDAR or camera coordinate system. It is necessary to determine the transformation matrix between two coordinate systems through joint calibration, that is, the extrinsic parameters between LiDAR and the camera together with the intrinsic parameters of the camera.

Currently, according to the characteristics of different methods, we divide the calibration between the LiDAR and visible light camera into a target-based method and a target-less method. For details and representative works, refer to Figure 1. The target-based method is to find the 2D feature points in an image coordinate system and the 3D feature points in a LiDAR coordinate system with the help of a standard target to establish geometric constraints and solve the extrinsic parameters of sensors through perspective-n-point (PNP) or nonlinear optimization. According to the shape of calibration target, it can be divided into 1D objects, such as line-feature-based target [8–10], 2D objects, such as chessboard [11–13] or circular holes [14], and 3D objects [15,16], such as spherical targets [17]. Refs. [18,19] detected the corner points in images and point clouds by the intersection of edge fitting lines, respectively, and solved calibration parameters by a linear equation. The calibration target they used was a monochrome board. This is mainly due to the fact that two different colors have different effects on LiDAR ranging, which will affect the accuracy of plane fitting. Refs. [20–22] introduced nonlinear optimization to improve the calibration accuracy. Ref. [23] adopted the method of space joint calibration combining coarse measurement and fine adjustment. The innovation of [24] was to use a known plane to estimate the 3D corners. Refs. [24,25] used a LiDARTag and intensity information to locate target, which was similar to the method in [26].

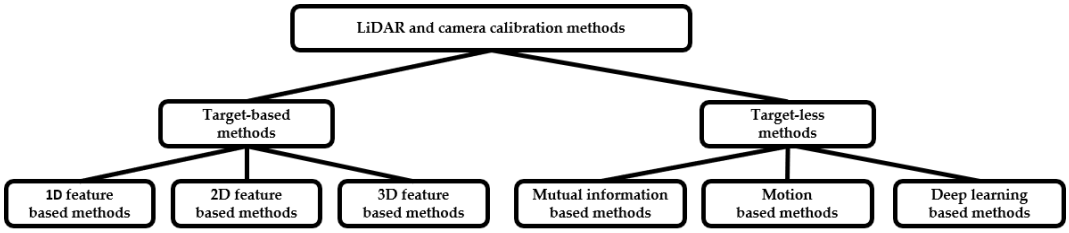


Figure 1. The representative methods of calibration between the LiDAR and visible light camera.

The target-less method can be divided into mutual-information-based approach [27,28], motion-based method [29,30], and learning-based technique [31–35]. The mutual-information-based approach can make full use of environmental information to complete online calibration. However, the disadvantage is that it needs to be carried out in the natural scene, and if the scene characteristics cannot meet the expectations, it will lead to considerable deviation of results. The motion-based method takes the calibration of LiDAR and camera as a hand–eye calibration problem, which can recover the right transformation by a series of rigid transformations. This method needs accurate motion information, so the static system is not applicable. Many learning networks use the convolutional neural network (CNN) to solve this task. These networks can calibrate the LiDAR–camera system without using calibration targets, matching information, and motion information, but the accuracy is dependent on the size of the training data set and the structure of the CNN. In the application, it requires high texture information of the environment and high computational power, and often needs to use a graphic processing unit to accelerate data processing.

The target-based method is still the mainstream because of its stability and low computational performance requirements. Therefore, sensors will undergo stable and high-precision target-based calibration methods to calibrate the extrinsic parameters before leaving the factory. The existing methods mainly have the following problems. First, the adaptability of LiDAR with different horizontal angle resolutions (θ) of calibration methods is not considered. For example, the method based on edge estimation [18,19] had a great

difference in the effectiveness of LiDAR with $\theta = 0.1^\circ$ and large $\theta = 0.4^\circ$. The second is that the camera intrinsic parameters, such as distortion coefficients, are not considered in the optimization process. The two-stage method, first calibrating the intrinsic parameters and then solving the extrinsic parameters, brings a lot of inconveniences to large-scale commercial applications in terms of efficiency. Third, for the locating of the calibration target, manual filtering is not conducive to the improvement of the automation level. The method of using intensity information to locate special marks will have different effects on LiDAR ranging due to two different color objects.

To solve the adaptability of edge fitting under different θs, we propose an approximation edge fitting technology based on background point clouds, which can refine the edge points under different θs by using a known geometric size of the calibration target. According to the geometric shape of the designed calibration board, a set of corner points extraction schemes of full-automatic locating and fine processing are designed, which can greatly reduce the manual operation and do not depend on the intensity information of LiDAR and special marks. Finally, camera intrinsic parameters are taken into account, and their initial values are sent to the optimization function, which avoids the inefficiency caused by the two-stage method when calibrating a large amount of equipment. The main contributions of this paper are as follows: We design a novel calibration target and propose a joint automatic calibration method based on edge refinement and virtual mask matching. An improved edge refinement scheme is introduced to refine the edge points. The maximum error of the edge estimation of the calibration board in this method does not exceed 5.0 mm, which greatly improves the accuracy of the corner points. The high-precision corner detection method makes the final calibration accuracy better than state-of-the-art techniques, especially in scenes with sparse point clouds, such as θ is 0.4°. An automatic location method, locating firstly and then fining processing, is proposed to get the corner points, which do not need any parameters set by users in the calibration process. In the experiments, we take multi-lines LiDAR as an example, such as 64 lines, and verify the accuracy and robustness of proposed method.

In the following sections, Section 2 is a brief introduction of the proposed method, Section 3 is the concrete introduction, Section 4 is the experiment and discussion, and Section 5 is the conclusion.

2. Proposed Calibration System and Computational Flow Chart

Figure 2a,b show the design of the calibration board. We need to place the calibration board about 30.0 cm in front of a flat wall to better obtain the points of the background wall for edge refinement, as shown in Figure 2c. In Section 3.1, we will use Euclidean distance to cluster all objects. In order to prevent the calibration board and the flat wall from being clustered into a category, the calibration board needs to keep a certain distance from the wall, e.g., 30.0 cm is considered in this paper. Actually, other suitable distances can also be used in practical applications. Holes dug in the calibration board are convenient for us to use geometric information for locating. Figure 2d is a diagrammatic sketch of this method, which involves the locating and matching of 3D feature points, the locating and refinement of 2D feature points, the solution of optimization equations, and the 3D–2D projection results. If we unify the data to the LiDAR coordinate system, we can get the point clouds with color, or unify the data to the camera coordinate system, and we can also get an image with sparse depth information. The symbols and function we will use in the following sections are given in Table 1.

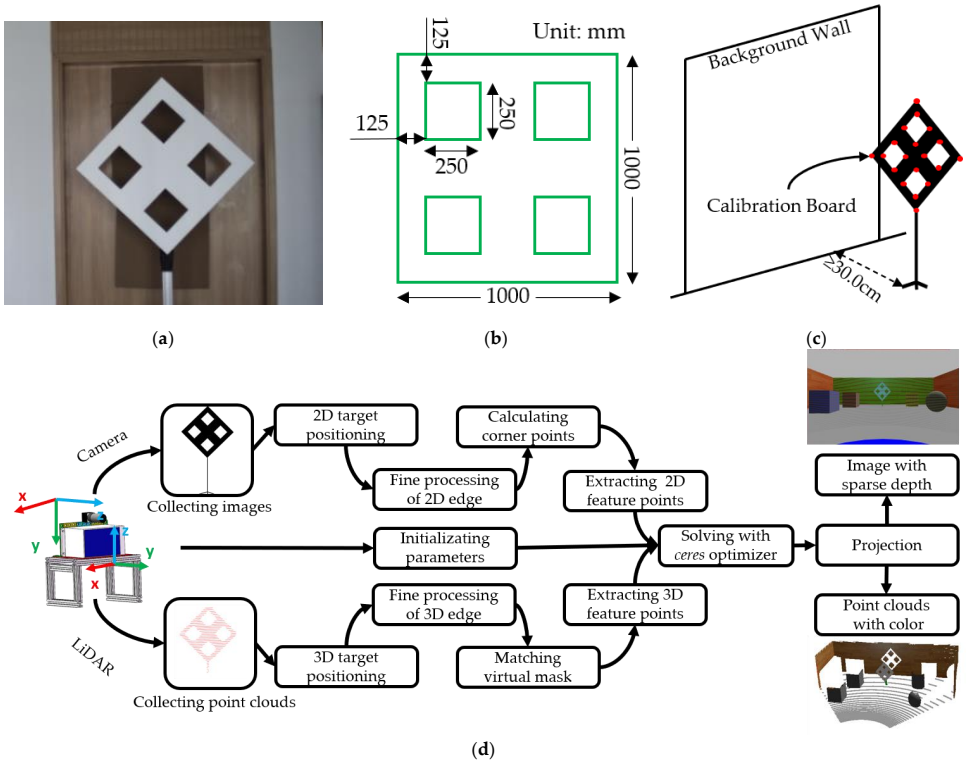


Figure 2. Calibration board and flow chart of proposed method, (**a**) is the calibration board, (**b**) is the geometric dimension drawing (unit: mm), (**c**) is the location of calibration board, red points represent the feature points we need, and (**d**) is the diagrammatic sketch.

Table 1. Symbols and functions definitions.

Symbols and Functions	Description
$p_i^L\left(x_i^L, y_i^L, z_i^L\right)$	a 3D point in LiDAR coordinate system
$p_i^C\left(x_i^C, y_i^C, z_i^C\right)$	a 3D point in camera coordinate system
$q_i(u_i, v_i)$	a 2D point in image coordinate system
P/Q	a 3D/2D points set
PC_i	a point cloud belonging to a same class of objects
V_i	a 3D space vector
$scale$	a scale factor about V_i
ss	the step size of $scale$
th	the expanded threshold of edges
$ring_i$	a scanning line of LiDAR
θ/θs	θ is the horizontal angle resolutions, θs is the plural form of θ
$R\left(r_x, r_y, r_z\right)$	a rotation matrix around the X, Y, and Z axes
$T\left(t_x, t_y, t_z\right)$	a translation matrix in the direction of X, Y, and Z axes
u_x/u_y	the coordinates of the image principal point
$f_x/\ f_y$	the scale factors in image x and y axes
$k_1,\ k_2$	the distortion coefficient of the image
Box	a bounding box in 2D plane
$Score(PC_i)$	a matching score between PC_i and the preset point cloud
$N(PC_i)$	the number of points in PC_i
$S(Box(PC_i))$	the area of Box enclosing PC_i
c	the number of cells for image segmentation
ε	the gray threshold of image
$E(q, q')$	the error function of q and q'
$HuberLoss(E(q, q'))$	the loss function of $E(q, q')$

3. Proposed Method

3.1. Automatic Locating Calibration Board

In this section, we describe how to automatically locate the point clouds of the calibration board (PC_{board}) and the background wall (PC_{wall}). First, to reduce the influence of ground points on 3D clustering, we use *patchwork* [36] to remove the ground. This is a robust algorithm for removing the ground, which can reduce the *ring* filtering parameters. The effect is illustrated in Figure 3a. We can cluster all objects according to Euclidean distance (Figure 3b), and each object after clustering is called PC_i. The normal vector V_{PC_i} of PC_i is obtained by plane fitting, shown in Figure 3c. Performing rotation transformation according to V_{PC_i} and $V_X(1,0,0)$ to transform the PC_i to the YOZ plane, which is called $PC\prime_i$. We take the geometric centroid of $PC\prime_i$ as the center and generate a box with the same size as our calibration board in the YOZ plane, as shown in the Figure 3d,e. We calculate the matching $Score$ between each $PC\prime_i$ and the preset box, and redefine $PC\prime_i$ as the set of PC_{in} and PC_{out}:

$$PC\prime_i = PC_{in} \cup PC_{out} \tag{1}$$

where PC_{in} is the set of red points in Figure 3d, PC_{out} is the set of black and blue points, and the $Score(PC\prime_i)$ is:

$$Score(PC'_i) = ScoreA \times ScoreB \times ScoreC \tag{2}$$

where $ScoreA = N(PC_{in})/N(PC_{out})$; $ScoreB$ is the ratio of the short side (w) to the long side (h) of the $Box(PC_{in})$, $ScoreB = w/h$; $ScoreC$ is the ratio of $S(Box(PC_{in}))$ to the actual calibration board square, $ScoreC$=$S(Box(PC_{in}))/S(board)$. Since our calibration board area is just 1.0 m^2, $ScoreC$ = $S(Box(PC_{in}))$.

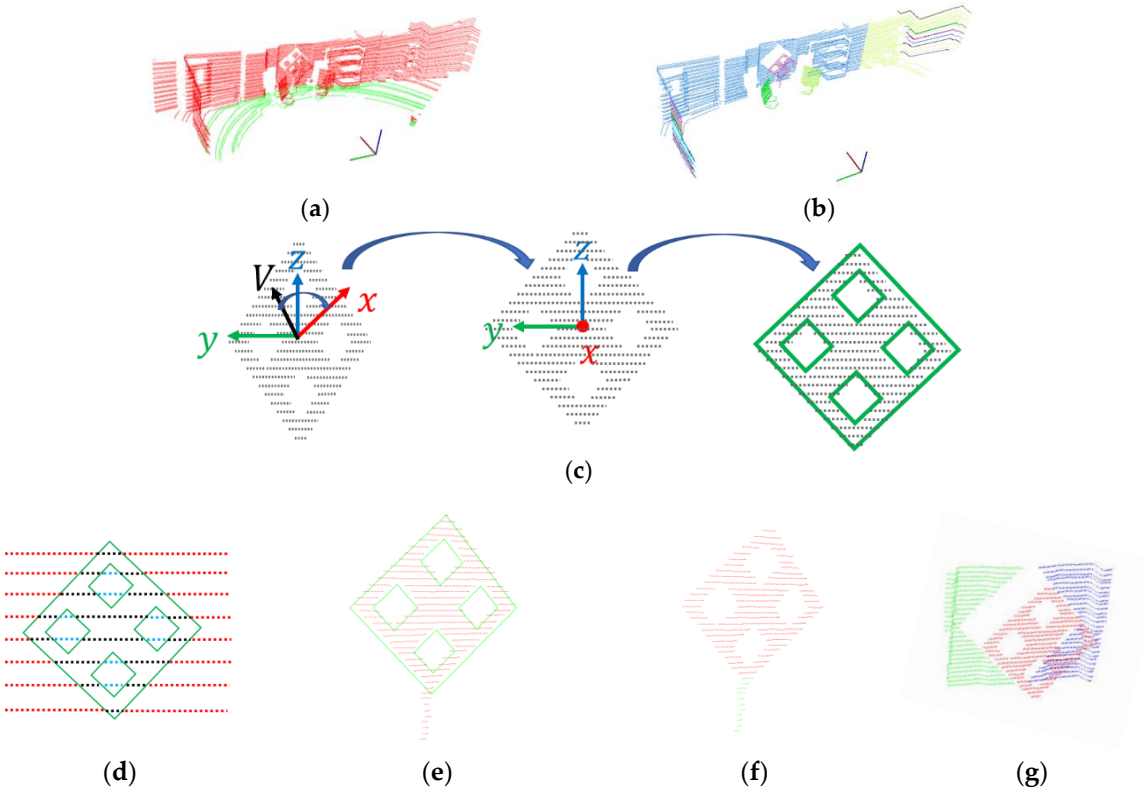


Figure 3. Automatic locating calibration object processing procedure. (**a**) is the point clouds with ground, (**b**) is the result of clustering, (**c**) is a schematic diagram of 3D to 2D conversion, (**d**) is an auxiliary diagram for locating, (**e**) is the point clouds with the highest matching score, (**f**) is the point clouds after removing the connecting rod, (**g**) is point clouds of calibration board (red points) and background point clouds (green and blue points).

Let us take *Score* as the evaluation function to screen PC_{i}. Three parts of *Score* can filter out the objects with a large or small number of points to ensure that the cluster where the calibration board is located can be filtered out with a high score. Figure 3e is the one with the highest *Score* among many point clouds. In Section 4.2, we will analyze the impact of *ScoreA*, *ScoreB*, and *ScoreC* on the automatic locating calibration board. It can be found that the connecting rod of the calibration board is also selected, shown in Figure 3e. The results of Figure 3c,d are ideal. In fact, due to the Euclidean clustering, the connecting rod of the calibration board will also be clustered into one category. We need to discard it for it should not participate in the calculation. Finally, we can delete the *ring* with a small number of point clouds by traversing each *ring*, as shown in the Figure 3f. The obtained red point cloud is PC_{board}. We select a 1.0 m $\times$ 1.0 m $\times$ 1.0 m box in the center of PC_{board} to ensure that the PC_{wall} (blue point clouds and green point clouds in Figure 3g) is framed together. Therefore, the calibration board is required to be placed within 1.0 m in front of the background wall. So far, we have acquired PC_{board} and PC_{wall} through automatic locating, and we can further refine the edge points by the method described in next section.

3.2. Approximation Edge Fitting

This section introduces an approximation edge fitting method, which can better fit the problems of inaccurate edge estimation caused by θ. As shown in Figure 4a, due to θ of LiDAR, two adjacent laser beams, one on the target and the other outside the target, cannot really scan the edge of the object. It can be seen from Figure 4b,c that although the laser spots of LiDAR have randomness within a certain range, overall, the blue laser spots with $\theta = 0.4^{\circ}$ are far from the real edge. Figure 4d–f show the sparsity of different θs The distance between the measured value and the real edge is closely related θ and the test distance, and the larger the θ, the worse the edge estimation of the target. Except the θ, vertical angle resolution, related to the number of rings of the LiDAR, may also be another factor. From Figure 4d–f, we can find that under the same number of lines of the LiDAR, the measurement results under different θs have different density levels of point clouds. Therefore, in the following sections, we take the LiDAR with 64 lines as an example (details are listed in Section 4.1) to research the calibration of different θs, while ensuring the same vertical angle resolution of the LiDAR, paying attention to the effect improvement in the case of different θs.

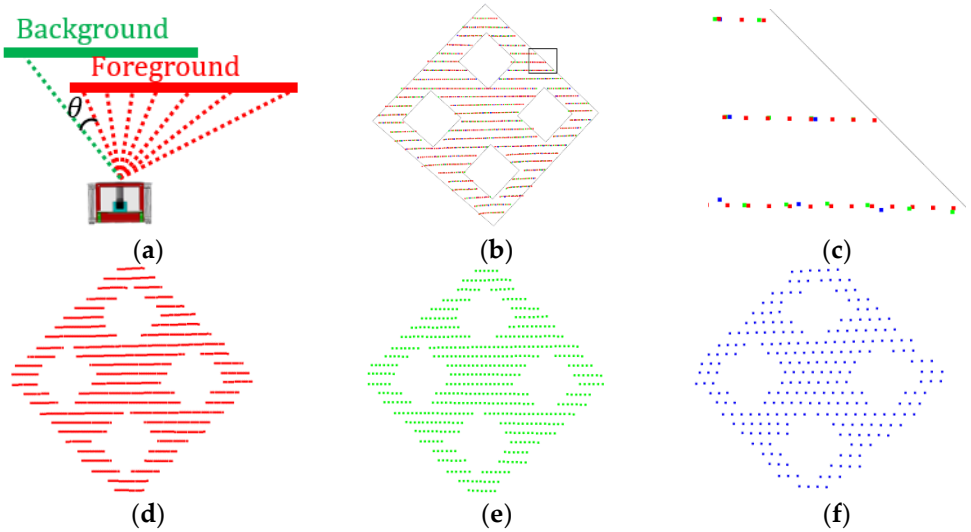


Figure 4. Point clouds distribution map with different θs, (**a**) is the top view of LiDAR, (**b**) is point clouds of three different θs and (**c**) is a partial diagram (black rectangle) of (**b**). Red point in (**d**) is $\theta = 0.1^{\circ}$, green point in (**e**) is $\theta = 0.2^{\circ}$, and blue point in (**f**) is $\theta = 0.4^{\circ}$.

As shown in the Figure 5a, for the partial schematic diagram of the calibration board scanned by LiDAR, we use $ring_i$ as an example, where the green background edge point is $p_i^b\left(x_i^b, y_i^b, z_i^b\right)$, the red foreground edge point on the board is $p_i^f\left(x_i^f, y_i^f, z_i^f\right)$, the direction vector (blue dotted line) between these two points is $V_i\left(\vec{x}_i, \vec{y}_i, \vec{z}_i\right)$, where $\vec{x}_i = x_i^b - x_i^f$, $\vec{y}_i = y_i^b - y_i^f$, $\vec{z}_i = z_i^b - z_i^f$. The set of background edge points is P^b, $p_i^b \in P^b$; the set of edge points of the foreground calibration object is P^f, $p_i^f \in P^f$.

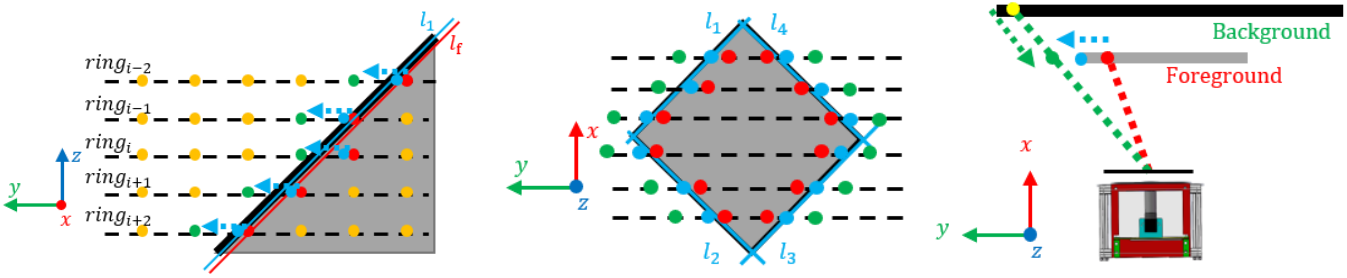


Figure 5. Illustration of edge points extraction. (**a**) is a partial view of (**b**), and (**c**) is a top view. Yellow points indicate the scanning points of LiDAR, red points indicate the edge points on the board, green points indicate the points projected to the board plane, and blue points indicate the improved points by the proposed method. The black dotted line is the scanning line of the LiDAR, the red line is fitted by the red points, and the blue line is fitted by the blue points.

Due to θ, the red line l_f fitted by P^f is still a certain distance from the actual black edge line, so the quadrilateral fitted with these edge points is often smaller than the actual size, and with the increase of test distance or θ, the difference between the quadrilateral and the real size is larger (the verification is shown in Section 4.2). It is necessary to expand P^f to approach the actual edge. First, we should get the points on the calibration board (PC_{board}) and background wall (PC_{wall}), which have been achieved in Section 3.1. Then, the edge points on the background, i.e., yellow points in Figure 5c, should be projected to the calibration board plane according to the direction of the laser beam of LiDAR. We match the foreground point (red point in Figure 5b) and background point (green point in Figure 5b) of each scanning $ring_i$, which can be easily realized by the sorting algorithm. At this time, we can obtain the background edge point P_i^b and calibration board edge point P_i^f. We choose V_i, the direction of the blue arrow in Figure 5a,c as the outward expansion direction. The outward length of the expansion is the distance between two points multiplied by a *scale* coefficient, $scale \in (0 \sim 0.5)$, and the blue middle edge point $p_i^m\left(x_i^m, y_i^m, z_i^m\right)$ in Figure 5c can be obtained:

$$p_i^m = p_i^f + scale * V_i \tag{3}$$

where p_i^m must be located between p_i^f and p_i^b. By traversing all the *ring*s, we can obtain the remaining edge point set P^m after approximation. We use P^m as the new edge points to fit the edge line and obtain the blue edge line l_1, shown in Figure 5b. Similarly, the edge line l_2, l_3, and l_4 can also be computed.

As for the selection of *scale*, we use the actual size of the calibration board (Figure 2b) for reference. Taking $scale = 0.1$ as the initial value. The newly expanded edge points are used to fit a rectangle by random sample consensus (RANSAC). The termination condition of RANSAC fitting is the difference between the calculated rectangle size and the actual physical size. If this difference is less than the *th*, the outward expansion can be stopped, and the edge of the current outward expansion should be saved. If the difference is larger than *th*, it indicates that there is still a gap between the measured result and the actual size,

and it needs to be further expanded. That is, scale needs to continue to increase with a step size (*ss*):

$$scale = scale + ss \tag{4}$$

where *scale* is not recommended to exceed 0.5, because when *scale* is greater than 0.5, p_i^m is closer to the green point in Figure 5. If *scale* is less than 0.5, p_i^m is closer to the red point in Figure 5. We prefer the edge point to be the point on the calibration board rather than outside. The *scale* is expanded to 0.4 in *ss* of 0.01 in the later experiment.

In the subsequent experiments of this paper, $th = 0.005$ m is selected as the experimental data. The smaller the *th*, the higher the estimation of the edge point accuracy, but the longer the detection time. By repeating the above process, a rectangular edge point cloud with an error of no more than 5 mm from the actual physical size can be measured (we will verify this in Section 4.2).

3.3. 3D Virtual Mask Matching

This section introduces a scheme for obtaining 3D feature points of the calibration board by using virtual mask matching. The edge point clouds, PC_{edge} and PC_{board}, are combined into a new point clouds (PC_{calib}), $PC_{calib} = PC_{edge} \cup PC_{board}$. We generate a virtual mask PC_{mask} with $N(PC_{mask})$ random points in YOZ plane with the same size as our board, shown in Figure 6, set $N(PC_{mask}) = 10^4$. We use the iterative closest point (ICP) method to match the PC_{mask} with the PC_{calib}, and the matching result is a rigid transformation matrix H. The feature points $P(p_1^L, p_2^L, \ldots p_{20}^L)$ we need can be calculated by $P\prime * H$, where $P\prime(p_1'^L, p\prime_2^L, \ldots p\prime_{20}^L)$ are feature points set on the PC_{mask}.

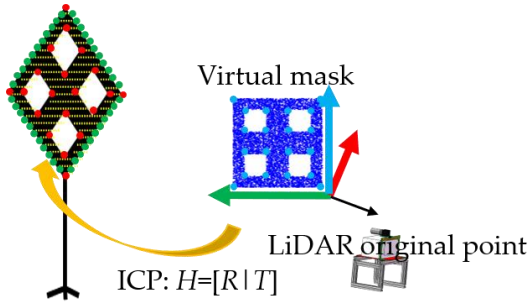


Figure 6. Schematic diagram of virtual mask matching. Red points represent the 3D feature points we need, green points represent the refined edge points, and blue points are generated for matching.

3.4. 2D Corner Points Detected in Image

We design an automatic matching method according to the characteristics of our calibration board, which can automatically find corners in images and finely process corners. It is mainly divided into two steps: corner points locating and corner points refinement. First, we use a region growing algorithm (RGA) to locate 2D feature points roughly. We divide the input image into $c \times c$ cells, and use the center point of each cell as the seed point of RGA, as shown in Figure 7a. In the next experiment, we choose $c = 10$, that is, the original image is divided into 10 equal parts horizontally and vertically, and 100 seed points are obtained. Since our calibration board is pure white, we only grow the seed points whose gray value is greater than ε, and set ε to 127, half of the maximum grayscale value, which can quickly filter out invalid seed points. Then, rectangle detection is carried out for the RGA results of each seed point, shown in Figure 7b. It is obvious that multiple rectangles are detected at the same location, shown in Figure 7c, which is not conducive to the selection of corner points. Finally, we perform the *k*-mean clustering method on all rectangles. The clustering index is the distance between the center points of each rectangular box. The *k*-value of clustering is five, which exactly corresponds to five rectangles to be detected. The clustering result is five clusters of rectangular boxes, and each cluster has at least one rectangle. The mean value of corner points of each cluster of

rectangular boxes is taken as the corner points for roughly locating position, as red points shown in Figure 7d.

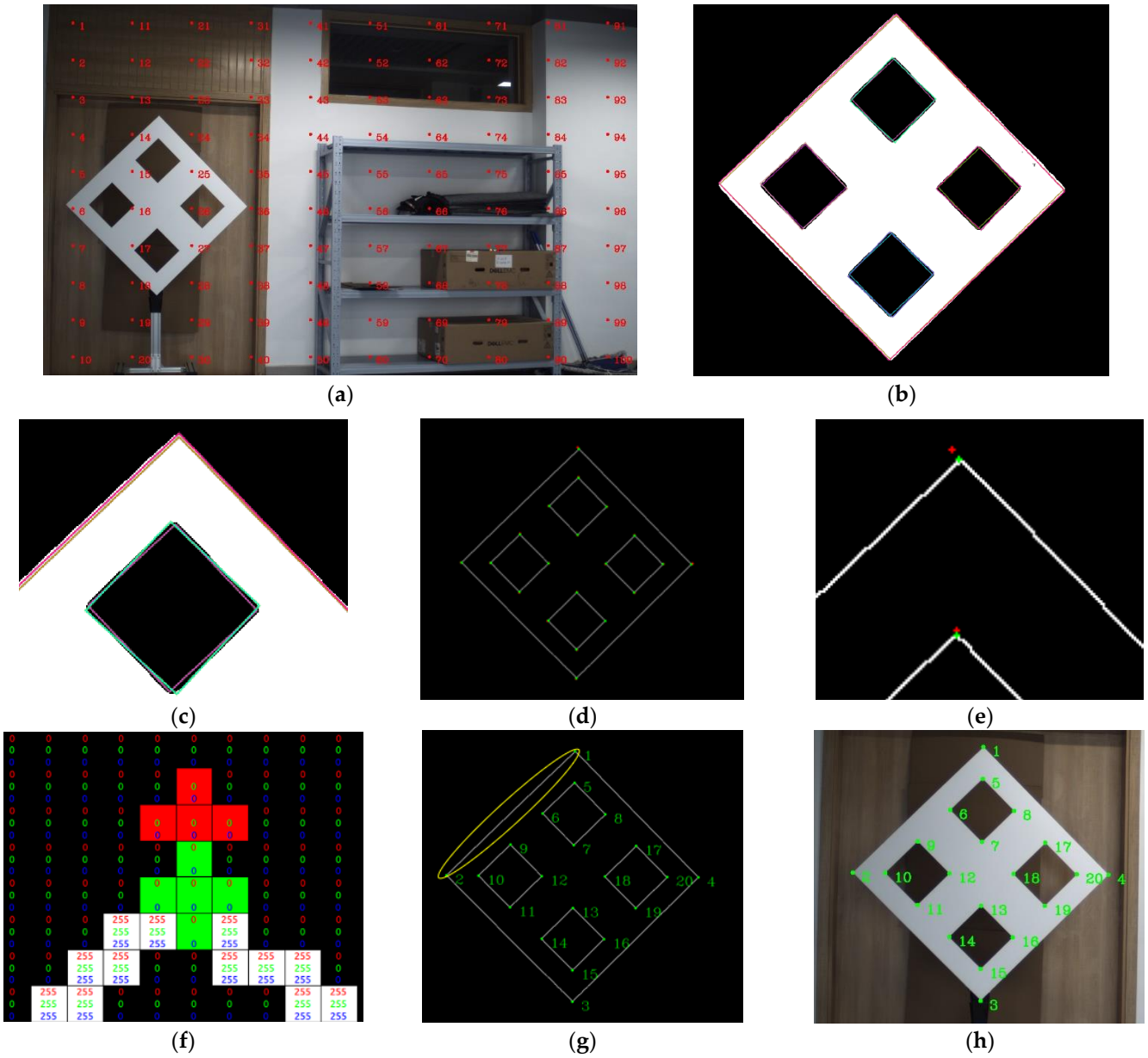


Figure 7. Automatic extraction process of 2D feature points. (**a**) shows the original image and seed points in each of cells, (**b**) presents the rectangle detected after the RGA by one seed point, (**c**) is the local graph of (**b**), it can be clearly seen that multiple rectangles are detected at the same location, green points in (**d**) illustrate refined corner points, red points illustrate roughly detected corner points, (**e**,**f**) are the partial diagram of (**d**), (**g**) is the detected and sorted corner points, the points in the yellow ellipse are used to fit $l\prime_{q_1}^{q_2}$, and (**h**) shows the final corner points colored in the original image.

After obtaining the corner points of roughly locating, it can be clearly found that the locating position is not accurate (red points in Figure 7d–f). This is because the rectangle detection has poor robustness and accuracy. It is necessary to further refine the corner points. Taking q_1 and q_2, the green points in Figure 7g as an example, we connect q_1 and q_2 to obtain a straight line $l_{q_1}^{q_2}$, count all the points close to $l_{q_1}^{q_2}$, which are circled by a yellow ellipse in Figure 7g. We use these points to refit the line and get the refined $l\prime_{q_1}^{q_2}$. We solve the connecting lines between the other points in the same way and estimate the refined corner points by the intersection of straight lines. As described in the green dots, it can be

found that the green points after the fine processing are more accurate than the red points detected before. In Figure 7g, for convenience of representation, we visualize the result in pixels. However, in fact, the coordinates of these corner points are sub-pixel. Finally, according to the order of feature points in Figure 7h, each refined corner point is sorted to facilitate pairing with 3D feature points.

3.5. Optimization Equation Modelling

In this section, an optimization equation is established to find the best intrinsic parameters of the camera and solve the extrinsic parameters of LiDAR and camera. The optimization process is intended to find the best LiDAR–camera extrinsic parameters and camera intrinsic parameters to minimize the re-projection error between the 3D–2D feature points. Through the methods of Sections 3.1–3.4, we can obtain 3D and 2D feature points after refinement processing. We use $P_i\left(p_{i,1}^L, p_{i,2}^L, \ldots p_{i,20}^L\right)$ and $Q_i(q_{i,1}, q_{i,2}, \ldots q_{i,20})$ to represent 3D/2D feature points set measured at the i th position, $i \in (1, n)$, n is the number of test positions. The projection process from 3D points to 2D points needs to go through a rigid transformation model, pinhole imaging model, and distortion model. The detailed formulas are as follows:

$$\begin{bmatrix} x^C \\ y^C \\ z^C \\ 1 \end{bmatrix} = [R^*|T^*] \begin{bmatrix} x^L \\ y^L \\ z^L \\ 1 \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} q^u \\ q^v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_x \\ 0 & f_y & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x\prime^C \\ y\prime^C \\ 1 \end{bmatrix} \tag{6}$$

where $x\prime^C = x^C/z^C, y\prime^C = y^C/z^C$, R^* represents the best rotation matrix, T^* represents the best translation matrix, and $K^* = \{f_x, f_y, u_x, v_y\}$ represents the best camera intrinsic, q^u and q^v are the coordinates of a 2D point in the image. Now we consider the distortion model of the camera, and we use $D = \{k_1, k_2\}$ to represent it. The final imaging of the camera is as follows:

$$r^2 = x\prime^C * x\prime^C + y\prime^C * y\prime^C \tag{7}$$

$$\begin{aligned} x''^C &= x\prime^C\left(1 + k_1 * r^2 + k_2 * r^4\right) \\ y''^C &= y\prime^C\left(1 + k_1 * r^2 + k_2 * r^4\right) \end{aligned} \tag{8}$$

$$\begin{bmatrix} q^u \\ q^v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_x \\ 0 & f_y & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x''^C \\ y''^C \\ 1 \end{bmatrix} \tag{9}$$

According to Equations (5)–(9), we establish the following optimization equations to solve R^*, T^*, K^*, and the best distortion parameter D^* :

$$[R^*, T^*, K^*, D^*] = \underset{R,T,K,D}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{20} HuberLoss\{E_{i,j}[q_{i,j},\ KD\left(Rp_{i,j}^L + T\right)]\} \tag{10}$$

where the form of $E_{i,j}(\cdot)$ is:

$$E_{i,j}(q, q') = \sqrt{(q^u - q\prime^u)^2 + (q^v - q\prime^v)^2} \tag{11}$$

and $HuberLoss(\cdot)$ is:

$$HuberLoss(\delta) = \begin{cases} \delta & \delta \le 1 \\ 2\sqrt{\delta} - 1 & \delta > 1 \end{cases} \tag{12}$$

Therefore, our optimization problem can be transformed into the optimization of *f(Q,P)*. The initial value of the camera's intrinsic parameters is given in advance, which

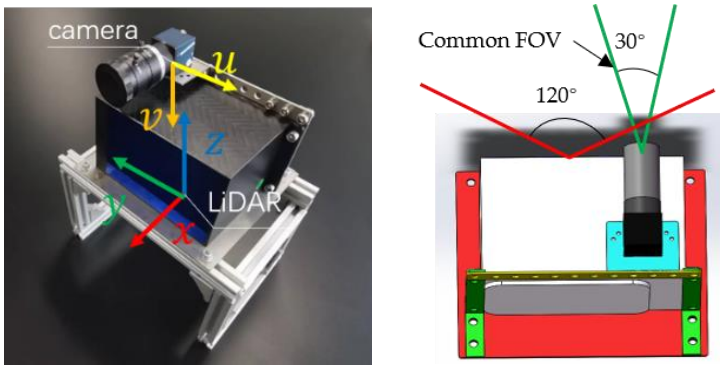
can come from the coarse calibration result of the camera, or the initial value given by the camera manufacturer. We use *ceres* optimizer (http://ceres-solver.org/, accessed on 1 December 2022) to optimize the intrinsic and extrinsic parameters. In the optimization, we set the maximum iteration as 500 and stopping tolerance as 10^{-10}.

$$f(Q,P) = \sum_{\substack{P_i \in P \\ Q_i \in Q \\ i=1}}^{n} \sum_{\substack{p_{i,j}^L \in P_i \\ q_{i,j} \in Q_i \\ j=1}}^{20} HuberLoss\{E_{i,j}[q_{i,j},\ KD\left(Rp_{i,j}^L + T\right)]\} \tag{13}$$

4. Experiments and Discussions

4.1. Proposed Experiment System

We setup the system with a LeiShen LiDAR with 64 lines and DaHeng camera with resolution of 1920 × 1200, shown in Figure 8a, and the field of view (FOV) of these two sensors is shown in Figure 8b. The size of the calibration target has been shown in Figure 2b. The maximum value of LiDAR's detection range is 100.0 m. The wavelength is 905 nm. The rotation speeds of our LiDAR can be set to 300.0 rpm (i.e., 5.0 Hz), 600.0 rpm (i.e., 10.0 Hz), and 1200.0 rpm (i.e., 20.0 Hz), and the corresponding θ are 0.1°, 0.2°, and 0.4°, respectively. We simulate the LiDAR with different θs by switching the rotation speed of the LiDAR motor. We mainly analyze the influence of θ on calibration, so other variables are fixed. In the controlled experiments of different methods in this section, only the θ is different.



(**a**) (**b**)

Figure 8. Setup of proposed method. (**a**) is the coordinate system definition, and (**b**) is the common FOV of two sensors.

4.2. Experiments

- Experiment 1: 3D feature points detected

This section verifies the accuracy of approximation edge points. We take the measurement results of the size of the calibration board as the evaluation index and analyze the evaluation index by the original edge points and the approximated edge points. Table 2 displays the measurement results under three θs when the test distance is 7.0, 10.0, and 12.0 m. Without any processing, due to the influence of θ, the length of the quadrilateral was lower than the actual value (1000.0 mm × 1000.0 mm), which also means that the selection of edge points was not accurate. The size of the bounding box (BB) will be much larger than the actual size or much smaller, which indicates the instability of initial edge points. After the approximation processing, when scale = 0.4, the measurement error reached within 5.0 mm. Compared with the results of BB, our method is more accurate for the estimation of edge points. The result can be seen that our approximation processing can make the measurement result reach the true size. In Figure 9, we can find that the edge estimation results at different θs are preferable to the initial values. In the following experiments, we

will apply this approximation refining strategy to [18,19,22]. The final calibration results are also greatly improved.

Table 2. Calibration board size measurement results.

Distance(m)	θ(°)	BB		Scale = 0.1		Scale = 0.2		Scale = 0.4	
		w (mm)	h (mm)	w (mm)	h (mm)	w (mm)	h (mm)	w (mm)	h (mm)
7.0	0.1	1012.34	996.22	984.11	981.82	997.90	991.14	998.51	997.95
	0.2	994.32	1004.66	979.91	987.12	993.41	997.94	997.20	996.22
	0.4	1003.72	993.731	974.41	988.20	994.90	997.01	997.81	999.17
10.0	0.1	1003.50	967.51	981.64	970.10	992.73	995.74	998.20	999.46
	0.2	984.19	1004.14	970.04	983.17	990.73	997.76	997.21	998.03
	0.4	992.16	1009.49	981.64	970.10	992.73	995.74	994.20	996.76
12.0	0.1	977.75	979.99	960.40	971.44	982.07	990.14	998.00	994.41
	0.2	978.03	979.76	954.70	988.00	991.43	996.01	996.24	996.90
	0.4	975.39	969.98	954.55	965.54	981.14	989.33	997.29	996.71

The w and h, respectively, represent the lengths of two sides of a rectangle fitted with the currently refined edge points; BB represents the 2D bounding box of this point clouds.

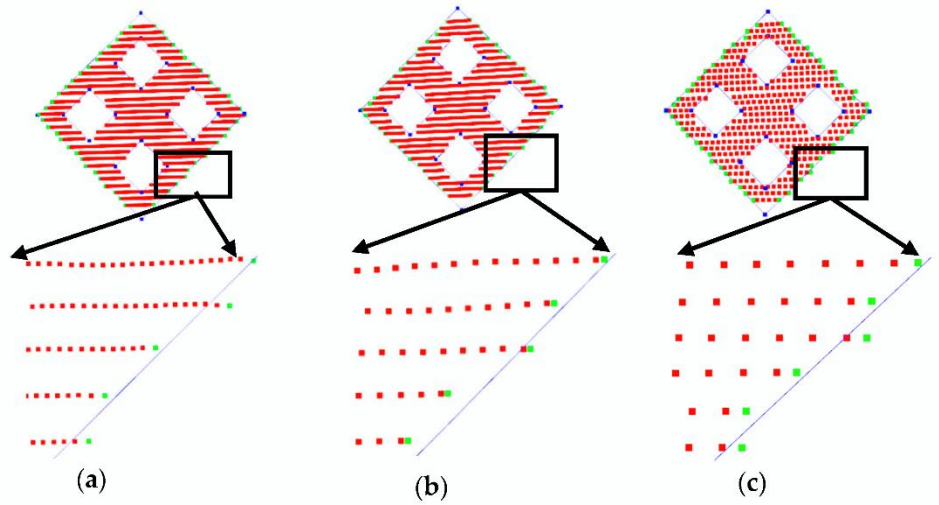


Figure 9. Refined edge point visualization results. Red points are the points scanned by LiDAR; green points are refined edge points. The blue lines are the boundary lines as large as the calibration object. (**a**) is the display of refined points when $\theta = 0.1°$, (**b**) is the result when $\theta = 0.2°$, and (**c**) is the result when $\theta = 0.4°$.

- Experiment 2: Automatic locating results

The experiment in this section is to verify the effect of automatic locating of the calibration board. After clustering in one frame data, nine objects, shown in Figure 10 and Table 3, can be obtained. It can be found that objects with many points, shown in Figure 10a,b, will lead to the reduction of overall final Score due to the reduction of *ScoreA*. For objects with a small number of point clouds, the final *Score* will reduce due to the decrease of *ScoreB* and *ScoreC*, shown in Figure 10d–i. Only our target board, whose *ScoreA*, *ScoreB*, and *ScoreC* all maintain the value close to 1.0, has the highest final *Score*. It can be said that our locating index is not affected by the number and shape of point clouds. For this reason, in Euclidean clustering, there is no need to specify the parameter of the minimum number of cluster point clouds. In this experiment, we set the minimum number of clusters as 30. Therefore, our method does not require the user to set additional parameters. In addition, the distance parameter of clustering needs to be specified additionally. The distance between LiDAR and the calibration board is often placed between 5.0–10.0 m. Within this distance range, the distance between adjacent points is not sensitive to the change of distance between the calibration board and the LiDAR origin point. We can set it

as 30.0 cm, which requires that our calibration board cannot be interfered by other objects within 30.0 cm around during the calibration process.

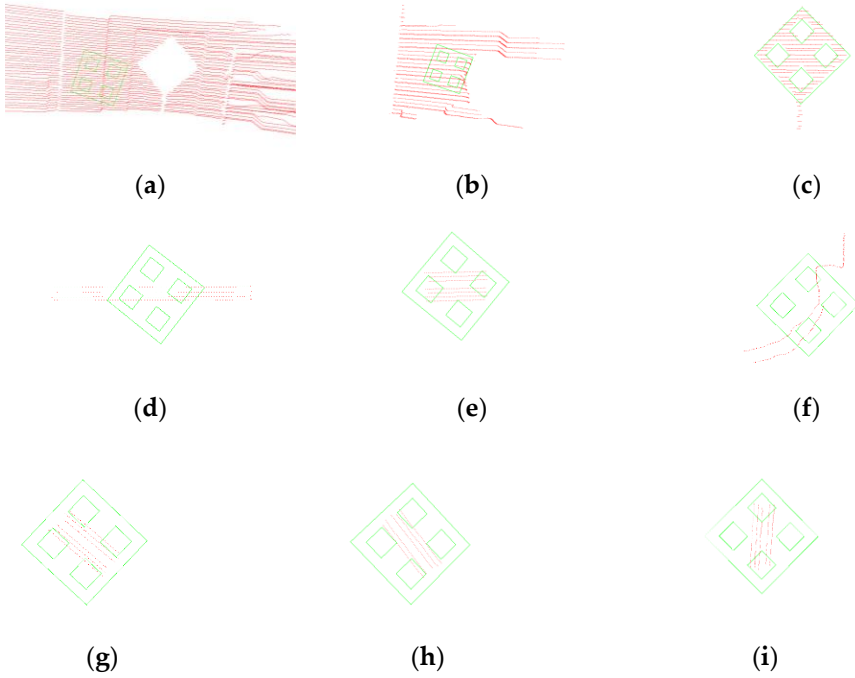


Figure 10. Results of Euclidean clustering when $\theta = 0.1°$. (**a**–**i**) are point clouds of different clusters, sorted in descending order according to the number of point clouds.

Table 3. Matching score of different objects.

θ(°)	Object	*N*	ScoreA	ScoreB	ScoreC	Score	IsTrue
0.1	Figure 10a	34,884	0.031361	**0.997999**	0.996798	0.031198	False
	Figure 10b	4180	0.136842	0.992114	**0.987158**	0.134019	False
	Figure 10c	1116	0.870968	0.994576	0.982246	**0.850865**	**True**
	Figure 10d	312	0.282051	0.134227	0.250382	0.009479	False
	Figure 10e	283	0.745583	0.461738	0.302227	0.104046	False
	Figure 10f	263	0.365019	0.279073	0.349486	0.035601	False
	Figure 10g	243	0.786008	0.400130	0.255705	0.080421	False
	Figure 10h	202	**0.876238**	0.338324	0.215452	0.063871	False
	Figure 10i	201	0.671642	0.323524	0.214348	0.046576	False
0.2	No. 1	9514	0.04225	**0.99512**	0.98902	0.04158	False
	No. 2	7917	0.04724	0.99461	**0.99401**	0.04670	False
	No. 3	2004	0.12974	0.99707	0.98869	0.12789	False
	No. 4	555	0.81441	0.98390	0.94908	**0.76051**	**True**
	No. 5	156	0.30769	0.17733	0.18889	0.01030	False
	No. 6	142	0.76056	0.46460	0.30043	0.10616	False
	No. 7	132	0.36363	0.29602	0.31777	0.03420	False
	No. 8	106	**0.94340**	0.39433	0.36821	0.13698	False
	No. 9	101	0.66336	0.34315	0.21464	0.04886	False
0.4	No. 10	3887	0.02727	0.96699	0.96381	0.02541	False
	No. 11	2276	0.06678	0.48287	0.80706	0.02602	False
	No. 12	1689	0.13025	0.99809	**0.99416**	0.12924	False
	No. 13	529	0.24952	**0.99838**	0.99115	0.24692	False
	No. 14	447	0.30648	0.98646	0.97205	0.29389	False
	No. 15	318	0.24842	0.52587	0.65521	0.08559	False
	No. 16	279	0.87813	0.97943	0.95880	**0.82464**	**True**
	No. 17	49	**0.91837**	0.72605	0.19707	0.13140	False
	No. 18	48	0.33333	0.32933	0.30258	0.03321	False

The Nos. 1–18 represent different objects after Euclidean clustering and *N* indicates the number of points belonging to one point cloud; the bold data indicate the optimal value in the current column; the item "IsTrue" indicates whether it is the calibration board.

- Experiment 3: Re-projection error analyses

This section is the evaluation of calibration accuracy. We compare the proposed method with other comparable methods. For each method, we select at least 100 pairs of feature points for a re-projection error test. Figure 11a–r are the error distribution diagrams of three methods. We can show the error distribution of different methods more clearly in the form of error circles. Figure 12a–o show the experimental results of different frame numbers. Table 4 is the qualitative and quantitative analysis comparing the results.

After testing a large number of feature points, our proposed method has a maximum error of no more than 2.0 pixels at each angle resolution, shown in Figure 11a–c. The method of [18] used half of the average distance between adjacent scanning points, as a new position of an edge point was greatly affected by θ. It can also show up from the error circles of Figure 11d–f, which is not suitable for using in LiDAR with large θ. It can be observed in the re-projection error analysis that the projection error is about five times of ours in Table 4. On the one hand, it is related to the poor estimation of edge points, and on the other hand, it is related to the linear equation solution without considering the camera distortion. We apply the approximation edge point fitting method to solve corner points and solve the calibration parameters with the same linear equation. Finally, we can find that the re-projection error is significantly reduced, shown in Table 4. Ref. [22] used a two-stage calibration method, and the deviation of camera intrinsic parameter calibration affected the final calibration effect. In addition, the feature point pairs of [22] are manually selected, which is extremely inconvenient for large-scale applications. Ref. [19] obtained the point clouds at different positions by means of distance statistical filtering, which to a certain extent solved the error caused by LiDAR in ranging to the calibration system. However, the amount of data processed is too heavy (300 frames at least). Compared with the method that we only use one frame of data, our method is obviously better. Similarly, we use our edge refinement method to replace the steps of extracting corner points in [19,22], and conduct experiments on the re-projection error of corner points. It can be found in Table 4 that the accuracy had been improved. Ref. [22] ignored the influence of θ on edge estimation, whilst Ref. [19] only considered the accuracy of edge estimation from one side. Our method considers four sides of the calibration board at the same time, and takes the actual size of the four sides as the reference index for fine processing, which has a higher accuracy. Through the comparison experiment with [18,19,22], we can find that improving the accuracy of corner points can effectively improve the accuracy of calibration. The edge refinement method proposed in this paper is helpful to improve the accuracy of corner points. The work of [20] applied two chessboards and one auxiliary calibration object to calibrate the extrinsic parameters of the camera and LiDAR system, which was not convenient. Ref. [24] had high accuracy, and the 3D feature points obtained were the same as the actual physical size. However, when applied to LiDAR with large θ, it is not as stable as our method. This is because when θ is large, the distance between the edge point of the calibration board and the real edge is often large, which will affect the matching accuracy. Finally, we make a qualitative evaluation of various methods in Table 4. We can clearly find from Figure 11a–c that, using our method, the re-projection error will not change significantly with the increase of θ, and the overall error remains within 2.0 pixels (red circles). In general, our method is superior to other methods in stability and automation level.

Figure 12a–o analyze the re-projection results under different numbers of frames. It is obvious that other methods need at least 3–5 frames of data to converge, and with the increase of θ, the error of final convergence will also increase. The proposed method has a low re-projection error in the first frame, and with the increase of the number of frames, the error change is not significant. By increasing the position of the calibration board, more data can be obtained. We use 2–10 positions to carry out the calibration experiment, shown in Table 5. It can be clearly found that with the increase of data volume, the average value and variance of re-projection error are reduced, which also shows the robustness of the proposed method. Our calibration board can add 20 more pairs of feature points without

adding a position, which is larger than the data volume of other methods, and the results are more convincing.

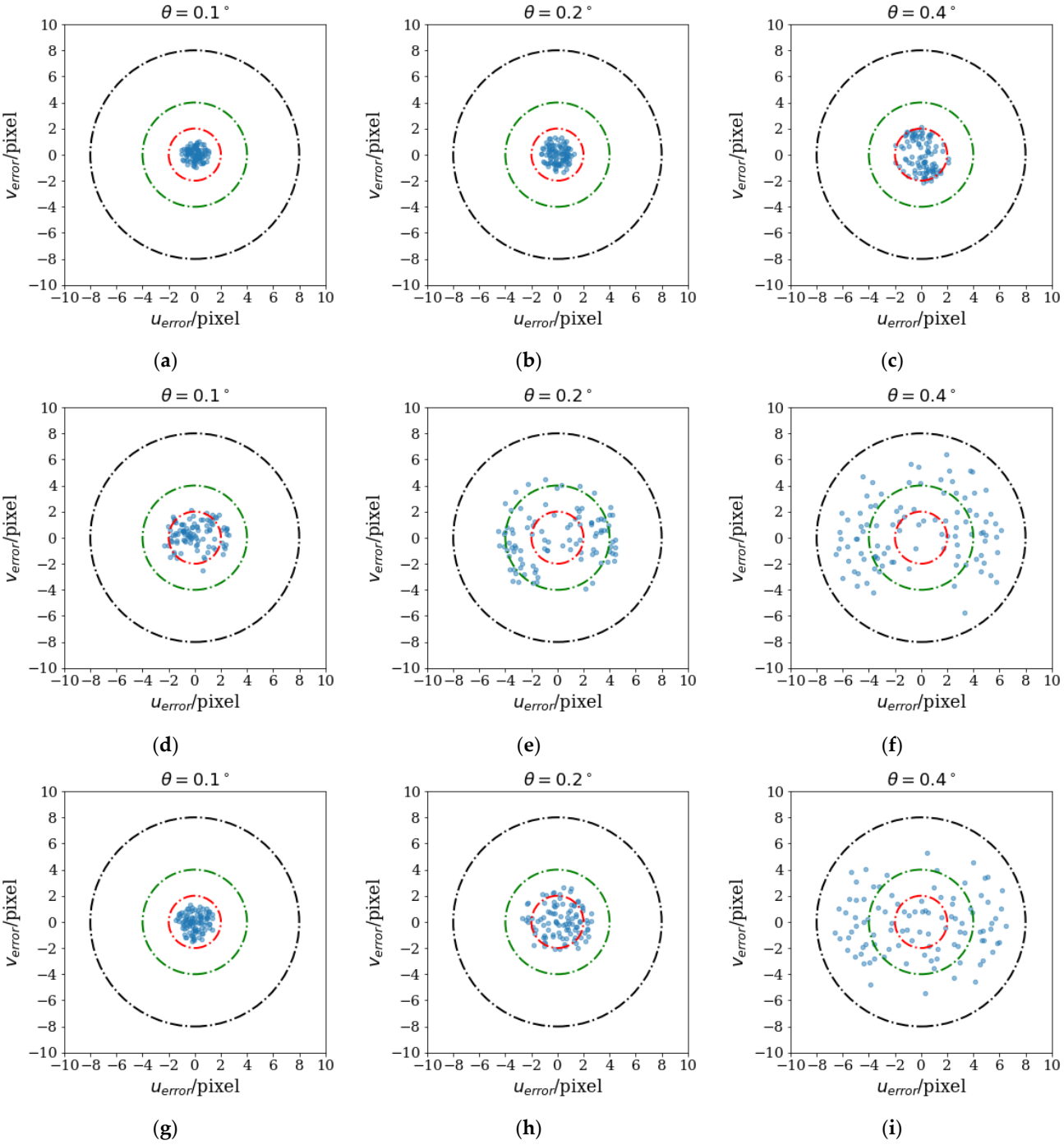


Figure 11. *Cont.*

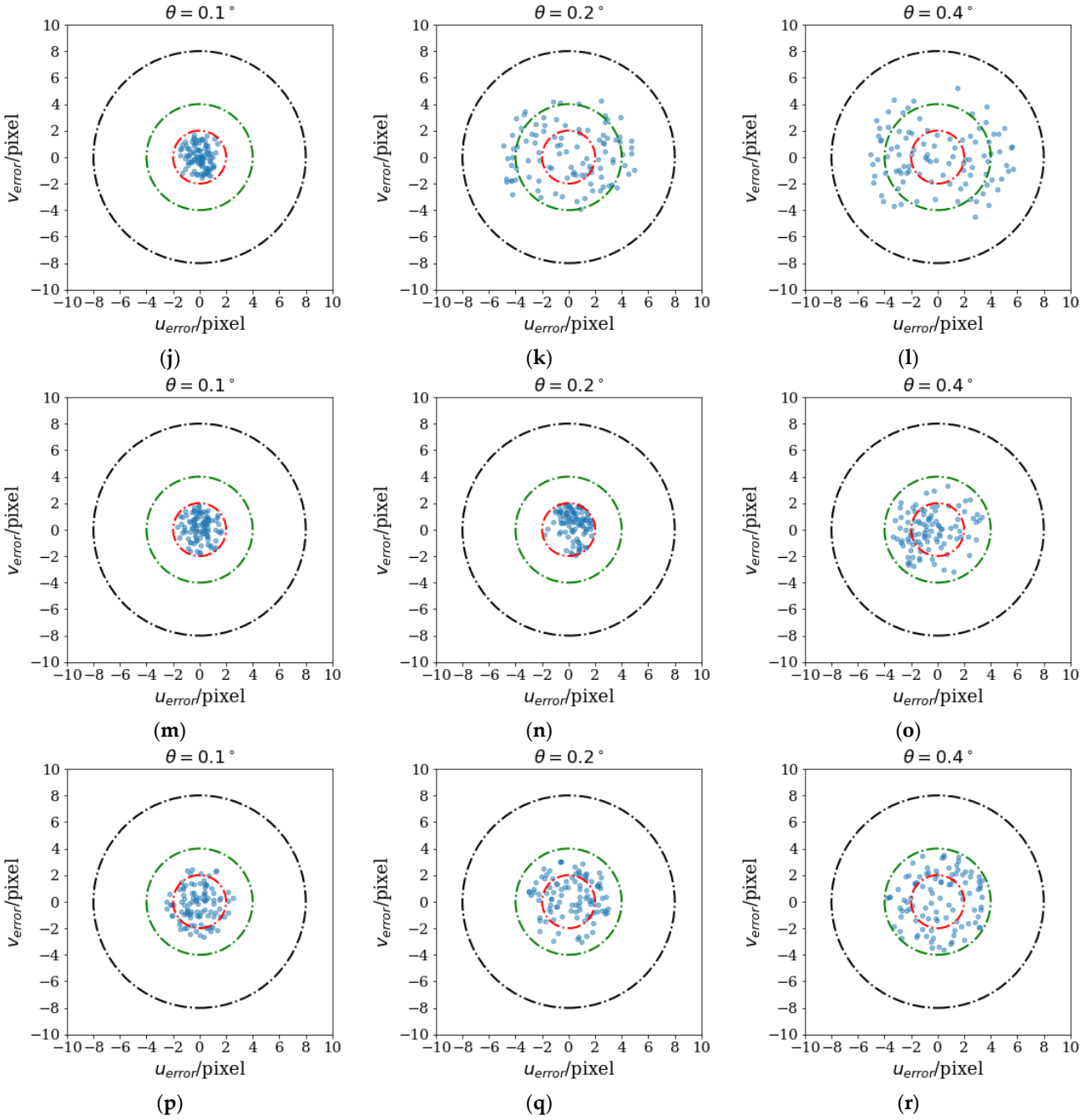


Figure 11. Error circle of different methods, the u_{error} *and* v_{error} are projection errors in the horizontal and vertical directions, respectively. (**a**–**c**) are proposed methods, respectively, corresponding to θ = 0.1, 0.2, and 0.4; we select one frame data for each position, (**d**–**f**) are methods in [18], respectively, corresponding to θ = 0.1, 0.2, and 0.4; (**g**–**i**) are the methods in [22], respectively, corresponding to θ = 0.1, 0.2, and 0.4, (**j**–**l**) are the methods in [19], respectively, corresponding to θ = 0.1, 0.2, and 0.4, (**m**–**o**) are the methods in [20], respectively, corresponding to θ = 0.1, 0.2, and 0.4, and (**p**–**r**) are the methods in [24], respectively, corresponding to θ = 0.1, 0.2, and 0.4.

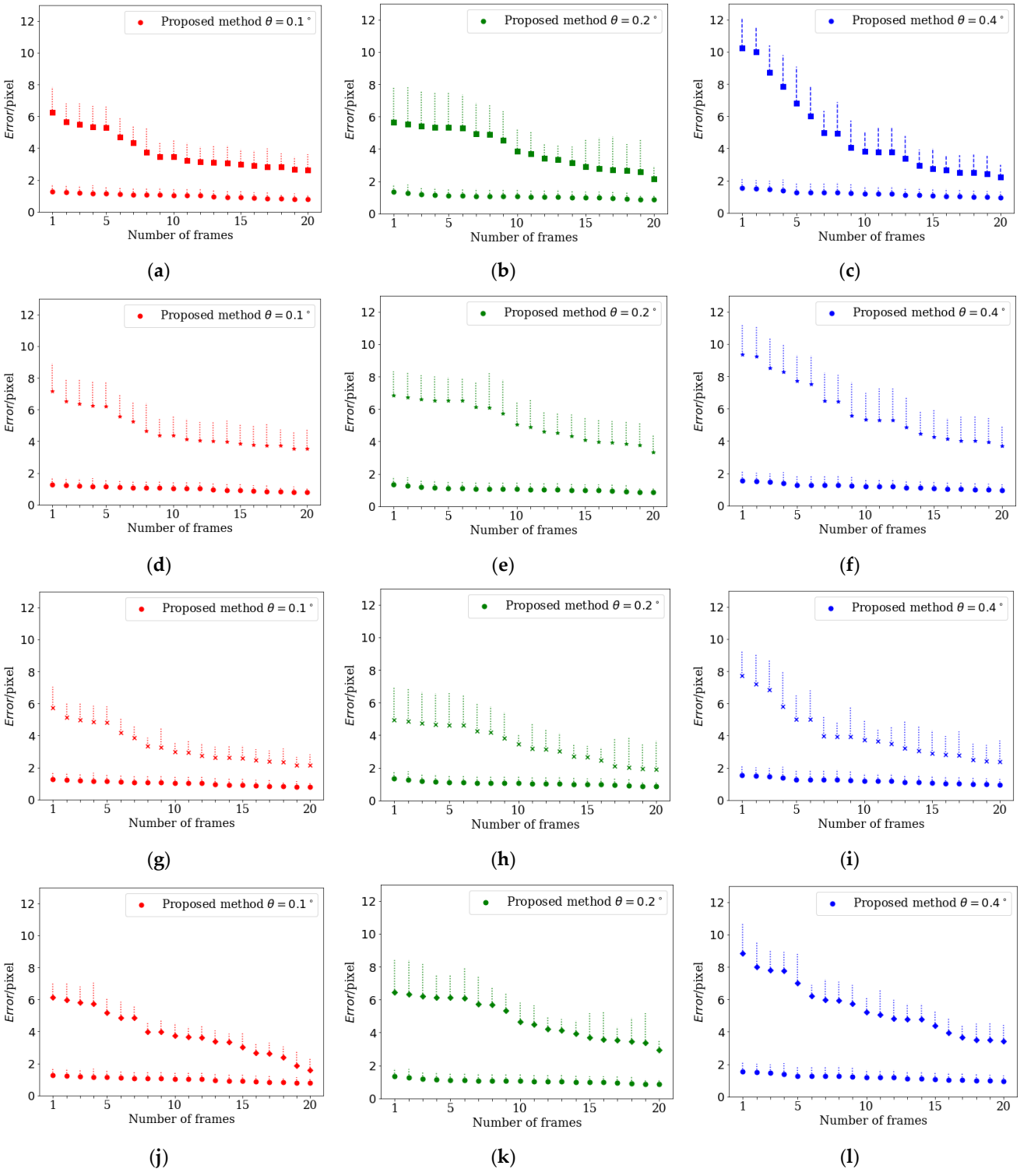


Figure 12. *Cont.*

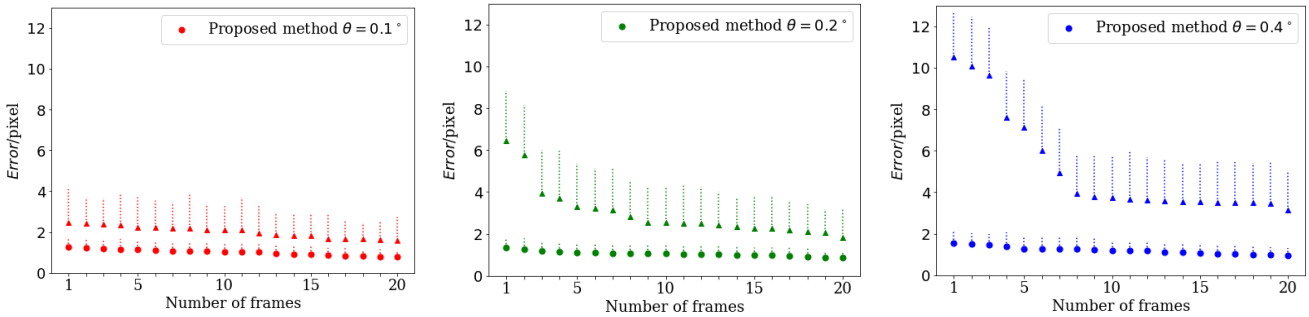


Figure 12. Re-projection error of different numbers of frames. The symbols and vertical lines represent the mean and 3σ range of re-projection errors calculated by different methods. (**a**–**c**) are the results compared with [18], (**d**–**f**) are compared with [22], (**g**–**i**) are compared with [19], (**j**–**l**) are compared with [20], (**m**–**o**) are compared with [24].

Table 4. Calibration results of different methods.

Re-Projection Result								
Method	**θ (°)**	**Error (pixel)**	**Improved (pixel)**	**Gain (%)**	**Intrinsic Parameters**	**Extrinsic Parameters**	**Stability**	**Automation Level**
Proposed method	0.1	0.9350	-	-	√	√	High	High
	0.2	1.0942	-	-				
	0.4	1.1992	-	-				
[18]	0.1	2.7422	2.2021	+19.7	√	√	Low	Low
	0.2	3.2414	2.4074	+25.7				
	0.4	5.9693	3.0073	+49.6				
[22]	0.1	2.1751	1.0416	+52.1	×	√	Middle	Middle
	0.2	2.9751	2.0465	+31.2				
	0.4	4.8978	3.3172	+32.3				
[19]	0.1	2.2684	1.6406	+27.7	√	√	Middle	Low
	0.2	2.6452	1.9234	+27.3				
	0.4	4.4114	3.7310	+15.4				
[20]	0.1	1.3317	-	-	×	√	Middle	Low
	0.2	1.3982	-	-				
	0.4	3.1492	-	-				
[24]	0.1	0.7433	-	-	×	√	Middle	High
	0.2	2.4157	-	-				
	0.4	3.5512	-	-				

Table 5. Mean and standard deviation of re-projection errors for 2–10 groups of images and point clouds.

θ(°)	**NP**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.1	Mean	1.535	1.469	1.426	1.301	1.283	1.274	1.272	1.249	1.21
	Std	0.696	0.703	0.549	0.658	0.565	0.529	0.547	0.492	0.476
0.2	Mean	2.109	1.945	1.637	1.470	1.457	1.408	1.324	1.303	1.256
	Std	1.007	0.682	0.740	0.623	0.697	0.644	0.737	0.624	0.552
0.4	Mean	2.301	1.858	1.750	1.745	1.620	1.453	1.388	1.311	1.28
	Std	1.372	0.720	0.712	0.630	0.720	0.802	0.790	0.755	0.586

NP represents number of test positions; Mean is the average error; and Std means the standard deviation.

- Experiment 4: Parameter consistency check

The calibration of LiDAR and camera is a no ground true problem, so the measurement means of calibration effect is not only the re-projection error, but also the data consistency. We collect data from a total of 100 positions, randomly select data from 10 positions for

calibration experiments, randomly select 100 times, and record the intrinsic parameters $\{f_x, f_y, u_x, v_y, k_1, k_2\}$ of camera and extrinsic parameters $\{R\{r_x, r_y, r_z\},\ T\{t_x, t_y, t_z\}\}$. The mean and Std are shown in the following Table 6; the change of intrinsic parameters is less than 1.0%, the change of rotation matrix is about 0.1°, and the change of translation matrix is about 4.0 mm, which indicates that the calibration parameters of our method are consistent.

Table 6. Mean and Std of optimization parameters.

θ(°)		f_x	f_y	u_x(pixel)	v_y(pixel)	k_1	k_2	r_x(°)	r_y(°)	r_z(°)	t_x(cm)	t_y(cm)	t_z(cm)
0.1	Mean	2825.75	2817.57	969.026	597.901	−0.220	0.187	89.767	−89.738	−0.321	−6.0651	9.6231	−1.4969
	Std	1.947	2.382	6.471	2.873	0.001	0.010	0.042	0.032	0.103	0.5986	0.3348	0.1995
0.2	Mean	2826.23	2818.55	970.405	595.841	−0.230	0.161	90.001	−89.775	−0.582	−6.0065	10.0176	−1.5216
	Std	2.827	3.463	5.824	3.604	0.009	0.000	0.069	0.035	0.111	0.4551	0.3172	0.4093
0.4	Mean	2826.85	2818.18	966.150	602.771	−0.236	0.201	90.106	−89.690	−0.624	−5.9884	8.4497	−1.7355
	Std	3.146	3.955	9.0291	5.042	0.007	0.001	0.044	0.042	0.171	0.4098	0.4856	0.4395

- Experiment 5: Qualitative analysis of calibration accuracy

The following experiment is a qualitative measurement of calibration effect. In order to verify the detection effect of UAVs under low-altitude remote sensing, we select different reference objects 3.0–100.0 m away from the LiDAR for qualitative evaluation. We project the point clouds onto the image plane and color them according to the distance. Figure 13 is the projection of the indoor environment and Figures 14–16 are the projection of the outdoor environment. The farthest projection distance in Figures 14–16 is about 100.0 m, which is enough to explain that our method has not offset in this distance. The buildings and vehicles can also show that the proposed method can add accurate depth information for nearby objects. The 3D reconstruction effect of Figure 17 also shows that the proposed calibration method is helpful to the advanced application of the LiDAR–camera system.

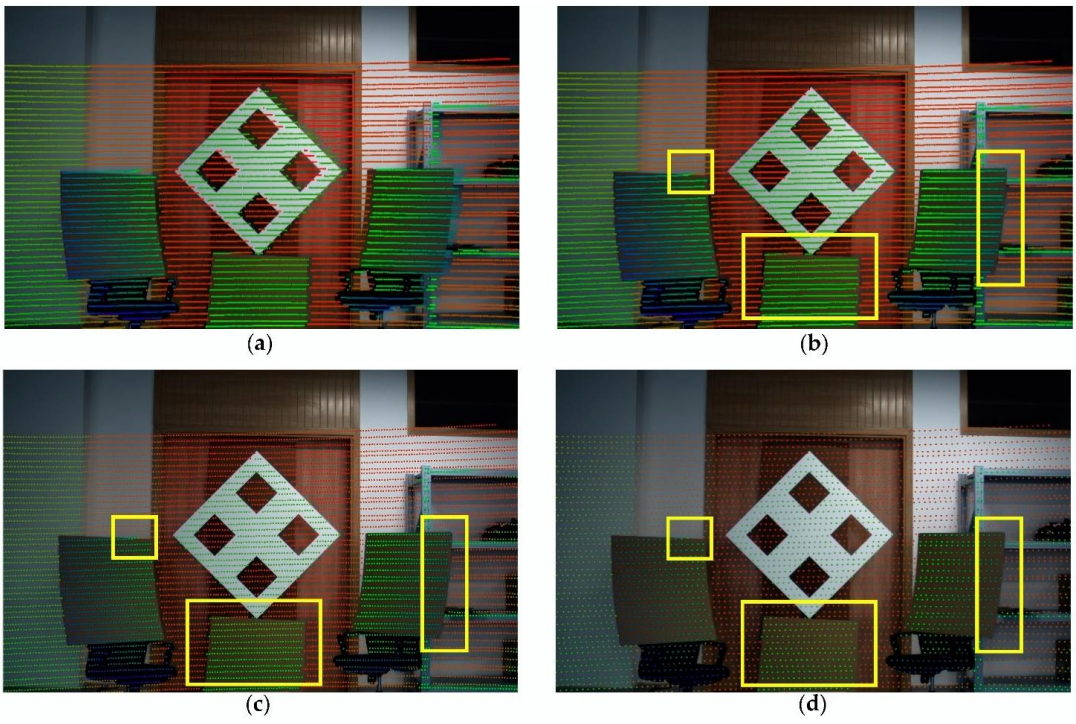


Figure 13. Results of indoor projection effect. (**a**) is projection in initial parameters, that is, calibration is not performed, (**b**–**d**) are the projection of θ which is equal to 0.1, 0.2, and 0.4, respectively. The contents in yellow rectangles can provide detailed reference.

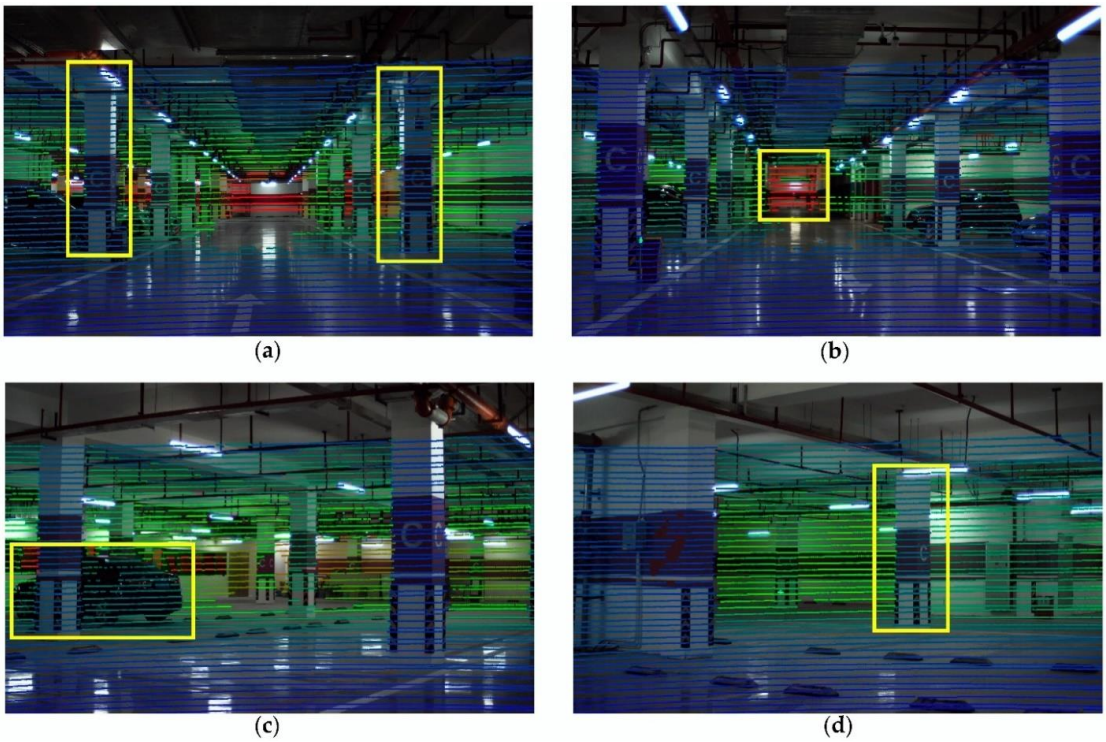


Figure 14. Results of outdoor scene test, $\theta = 0.1°$. The contents in yellow rectangles in (**a**–**d**) are projections at different distances.

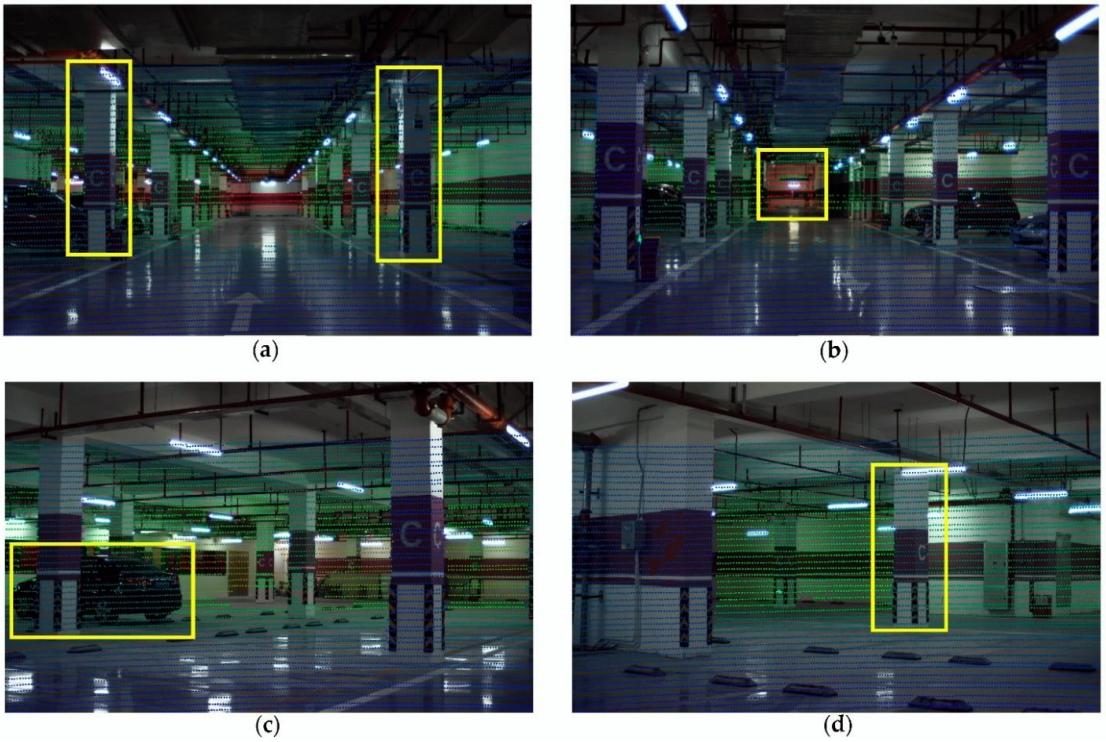


Figure 15. Results of outdoor scene test, $\theta = 0.2°$. The contents in yellow rectangles in (**a**–**d**) are projections at different distances.

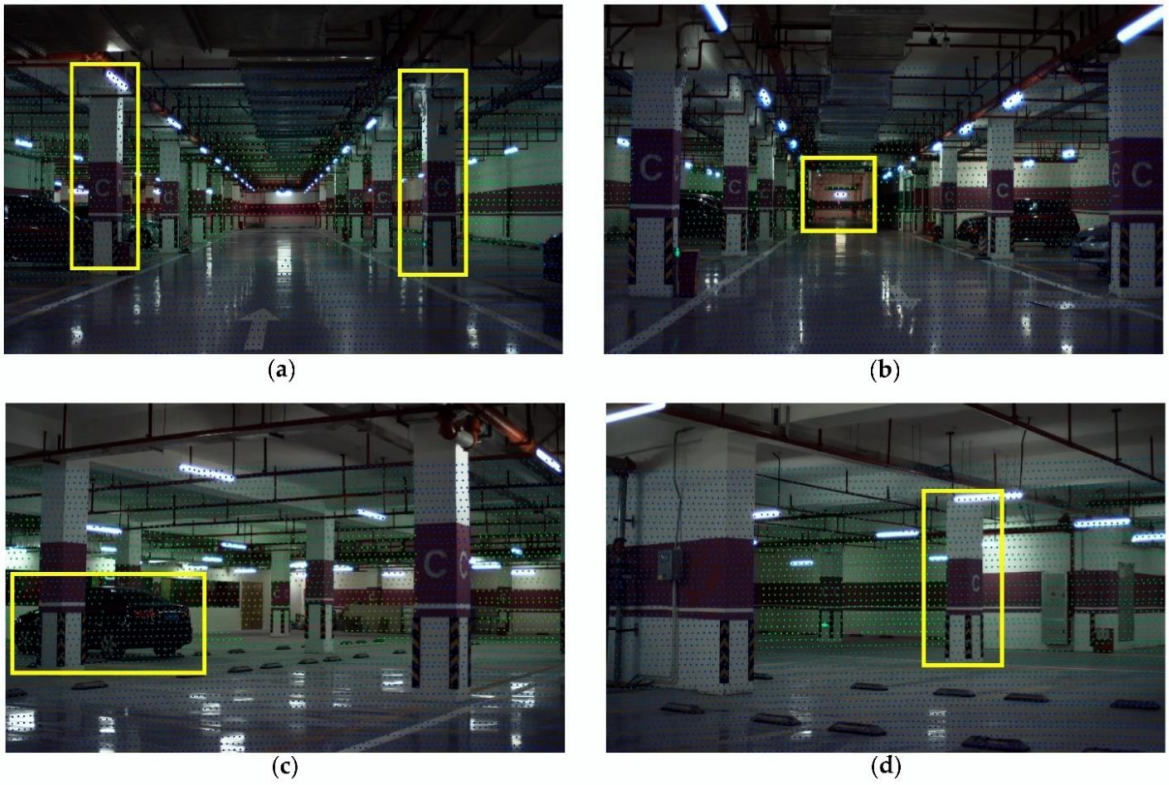


Figure 16. Results of outdoor scene test, $\theta = 0.4°$. The contents in yellow rectangles in (**a**–**d**) are projections at different distances.

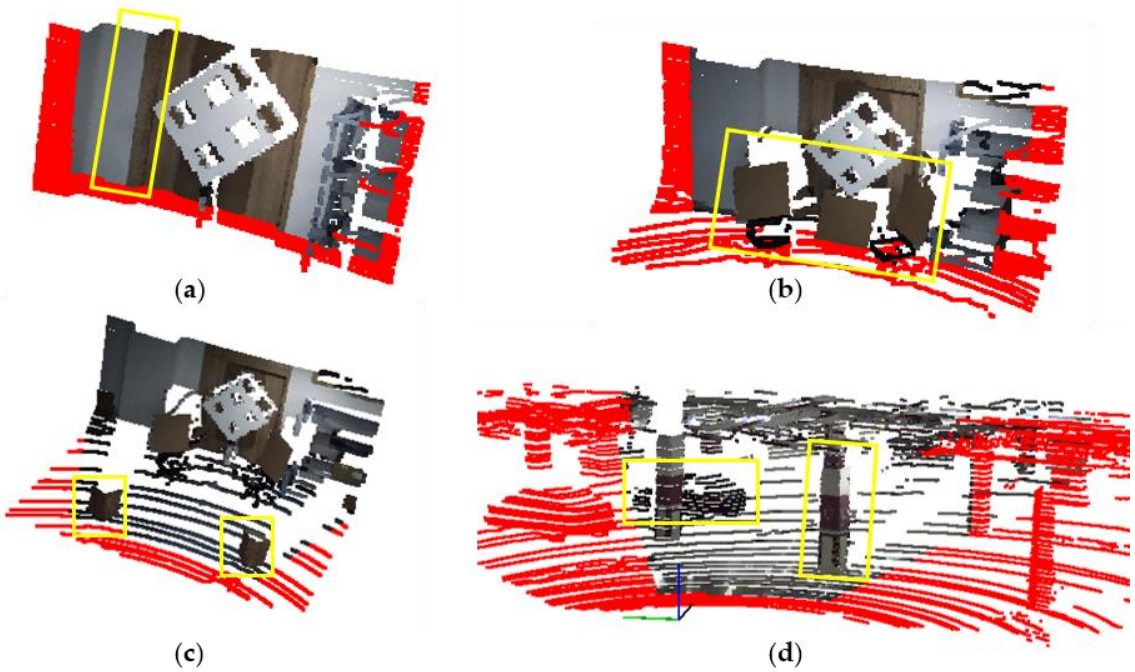


Figure 17. Results of 3D reconstruction effect. The test distances of (**a**–**c**) are 3.0 m, 5.0 m, and 7.0 m, respectively, (**d**) is the test result of outdoor scene. The red point is the point in the non-common FOV of the LiDAR and camera. The contents in yellow rectangles can provide detailed reference.

4.3. Discussions

Low-altitude UAVs have been used in many remote sensing fields [37] in recent years such as urban management, crop monitoring, and industrial patrol inspection, etc. It is

essential to apply multi-sensor fusion technology to low-altitude UAVs. Calibration is the basic requirement for multi-sensor platforms where data need to be represented in a common coordinate system for the purpose of analysis and information fusion [38]. The target-based calibration method has become the first choice for the calibration methods because of its stability and high accuracy. This paper designs a special calibration target and its corresponding calibration algorithm, which can achieve high precision and great calibration efficiency. While considering the calibration accuracy, we also consider the implementation efficiency of this method. The fully automatic data processing flow and joint optimization scheme make this method suitable for large-scale commercial applications.

At present, there are few methods to consider the effect of LiDAR with different θs on calibration. Experiment 3 in this paper shows that LiDAR with different θs has significant differences in calibration. The method of fine edge processing based on background point clouds proposed in this method is to solve the problem of insufficient calibration accuracy caused by large θ. After using the proposed method, the calibration accuracy and robustness have been improved to some extent, as analyzed in Table 4, and the demand for data volume has also been decreased, as shown in Figure 12 and Table 5. In addition, the method in this paper focuses on large-scale application, so every step of the design is to minimize the artificial setting of parameters, so as to improve the implementation efficiency of calibration. The average processing time of this method for one frame of data, including point cloud and image, is 7.5 s. It has higher time complexity than the traditional method, which is one aspect that can be optimized in the future. However, we are more focused on the improvement of calibration accuracy and automation level. Higher accuracy and less human participation are our final goals. In terms of optimizing functions, we need to give the initial extrinsic parameters in advance, which can be easily given by PnP [39,40].

Here, we discuss the advantages of the calibration board we designed. We have designed a special hollow monochromatic plane plate, which has the following advantages: (1) the monochromatic flat plate avoids the interference of different colors when the LiDAR is ranging, which is conducive to plane fitting and plane normal vector extraction. (2) The hollow calibration object is convenient for automatically locating its position in the point clouds by using geometric information. As for these two advantages, we conduct the following comparative experiments. We paste a small piece of checkerboard on the calibration object and find that the depth discontinuity with other positions appears at the place where the LiDAR scans, shown in Figure 18. This phenomenon is related to the intensity information performance of LiDAR itself. When we use the LiDAR to detect the calibration object, the chessboard in black and white areas will make the ranging depth uneven [18,19], while the monochromatic calibration board can reflect a flat plane. Our method does not depend on the intensity information [41] or additional sensors [42], and only uses geometric information to locate the calibration board; thus, it can be used with any types of LiDAR device and has better universality.

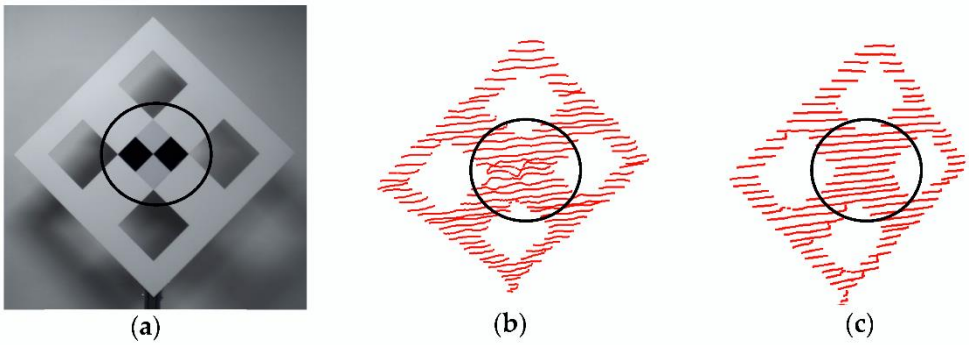


Figure 18. Influence of different colors on LiDAR ranging. (**a**) is the calibration board with one chessboard pattern, (**b**) is the affected ranging map, (**c**) is the ranging map with no chessboard pattern.

The advantages of the method described in this paper are as follows: (1) The refined edge points enable the LiDAR with large horizontal angle resolution (θ = 0.4) to complete

calibration with high accuracy; (2) The two-step corner points detection method designed by us can realize fully automatic data processing, greatly reduce manual participation, and does not need parameters to be manually set in the calibration process; (3) Our method does not require a high amount of calibration data. For 5–10 positions, one frame of data per position can complete the calibration with high accuracy. This is also related to the results of our 2D and 3D feature point refinement processing because the feature points with high accuracy can reduce the amount of data involved in optimization. The experimental scheme in this paper also has some shortcomings. For example, our research is based on LiDAR with 64 lines, and has not been verified on the LiDAR with fewer lines (32 lines or less) [43,44], and the vertical angle resolution needs to be further considered, which requires further research and improvements. In future work, we will apply the results of joint calibration of LiDAR and visible light camera to the application of low-altitude UAVs remote sensing, ground object recognition, depth information completion, and 3D terrain reconstruction, etc. Our application scenario is to use UAV with a visible light camera and LiDAR to conduct high-precision 3D mapping in a small range. The joint calibration of these sensors is the pre-work of this application. In this paper, this calibration method is completed in an indoor environment, as shown in Figure 2a. As for the verification of an outdoor environment, in order to show the projection effect at different distances, we chose the scene with very rich features, the nearest distance is 3.0 m, and the farthest distance is 100.0 m, as shown in Figures 14–16. The final projection effect shows that the projection of our method at different distances is accurate. As for the limitations of our method on the application level, the efficiency of the algorithm is lower than that of a pure image algorithm, which is also related to the increase of 3D data. In addition, we pay more attention to the ground object detection within 100.0 m of low-altitude, and do not consider the objects with a distance greater than 100.0 m, which is related to the inherent properties of our LiDAR.

5. Conclusions

In this paper, we propose a novel, fully automatic joint calibration method of LiDAR and visible light camera. Compared with the existing methods, we propose an approximate edge fitting technology, which has obvious improvements based on the existing methods and has definite accuracy improvement. Our method does not need to use the intensity information of LiDAR, which has better universality for the method itself and can also be applied when the LiDAR's horizontal resolution is sparse, or the intensity information is poor. For the automatic process, we propose a strategy of first locating and then refinement, which can fully realize 2D and 3D corner points detection. In the calibration process, the user does not need manual intervention, which greatly increases the efficiency of calibration. In the aspect of effect display, we project the 3D point clouds onto the image plane and show the calibration effect in different test distances. The 2D image with depth information can be used for advanced visual applications. We also color the 3D point cloud according to the calibrated parameters, which is instructive for 3D reconstruction.

Author Contributions: Conceptualization, C.C., J.L., H.L., S.C. and X.W.; data curation, C.C.; formal analysis, C.C., J.L., H.L. and X.W.; funding acquisition, J.L. and H.L.; investigation, J.L., H.L., S.C. and X.W.; methodology, C.C., H.L. and X.W.; project administration, J.L., H.L. and S.C.; resources, J.L., H.L. and S.C.; software, C.C., H.L., S.C. and X.W.; supervision, H.L.; validation, C.C., H.L. and S.C.; visualization, C.C., H.L. and X.W.; writing—original draft, C.C. and H.L.; writing—review and editing, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific and Technological Innovation Foundation of Foshan, USTB under Grant BK20AF007, the National Natural Science Foundation of China under Grant 61975011, the Fund of State Key Laboratory of Intense Pulsed Radiation Simulation and Effect under Grant SKLIPR2024, and the Fundamental Research Fund for the China Central Universities of USTB under Grant FRF-BD-19-002A.

Data Availability Statement: Not applicable.

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

References

1. Goian, A.; Ashour, R.; Ahmad, U.; Taha, T.; Almoosa, N.; Seneviratne, L. Victim Localization in USAR Scenario Exploiting Multi-Layer Mapping Structure. *Remote Sens.* **2019**, *11*, 2704. [CrossRef]
2. Hong, Z.; Zhong, H.; Pan, H.; Liu, J.; Zhou, R.; Zhang, Y.; Han, Y.; Wang, J.; Yang, S.; Zhong, C. Classification of Building Damage Using a Novel Convolutional Neural Network Based on Post-Disaster Aerial Images. *Sensors* **2022**, *22*, 5920. [CrossRef] [PubMed]
3. Raman, M.; Carlos, E.; Sankaran, S. Optimization and Evaluation of Sensor Angles for Precise Assessment of Architectural Traits in Peach Trees. *Sensors* **2022**, *22*, 4619. [CrossRef] [PubMed]
4. Zhu, W.; Sun, Z.; Peng, J.; Huang, Y.; Li, J.; Zhang, J.; Yang, B.; Liao, X. Estimating Maize Above-Ground Biomass Using 3D Point Clouds of Multi-Source Unmanned Aerial Vehicle Data at Multi-Spatial Scales. *Remote Sens.* **2019**, *11*, 2678. [CrossRef]
5. Wang, D.; Xing, S.; He, Y.; Yu, J.; Xu, Q.; Li, P. Evaluation of a new Lightweight UAV-Borne Topo-Bathymetric LiDAR for Shallow Water Bathymetry and Object Detection. *Sensors* **2022**, *22*, 1379. [CrossRef]
6. Chen, S.; Nian, Y.; He, Z.; Che, M. Measuring the Tree Height of Picea Crassifolia in Alpine Mountain Forests in Northwest China Based on UAV-LiDAR. *Forests* **2022**, *13*, 1163. [CrossRef]
7. Song, J.; Qian, J.; Li, Y.; Liu, Z.; Chen, Y.; Chen, J. Automatic Extraction of Power Lines from Aerial Images of Unmanned Aerial Vehicles. *Sensors* **2022**, *22*, 6431. [CrossRef]
8. Zhang, Z. Camera Calibration with One-dimensional Objects. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2004**, *26*, 892–899. [CrossRef]
9. Wu, F.; Hu, Z.; Zhu, H. Camera Calibration with Moving One-dimensional Objects. *Pattern Recognit.* **2005**, *38*, 755–765. [CrossRef]
10. Bai, Z.; Jiang, G.; Xu, A. LiDAR-Camera Calibration Using Line Correspondences. *Sensors* **2020**, *20*, 6319. [CrossRef]
11. 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 (IEEE ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 3936–3943.
12. Cai, H.; Pang, W.; Chen, X.; Wang, Y.; Liang, H. A Novel Calibration Board and Experiments for 3D LiDAR and Camera Calibration. *Sensors* **2020**, *20*, 1130. [CrossRef]
13. 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 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5562–5569.
14. Guindel, C.; Beltrán, J.; Martin, D.; Garcia, F. Automatic Extrinsic Calibration for Lidar-stereo Vehicle Sensor Setups. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
15. Gong, X.; Lin, Y.; Liu, J. 3D LIDAR-Camera Extrinsic Calibration Using an Arbitrary Trihedron. *Sensors* **2013**, *13*, 1902–1918. [CrossRef] [PubMed]
16. Pusztai, Z.; Hajder, L. Accurate Calibration of Lidar-camera Systems Using Ordinary Boxes. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 394–402.
17. Kümmerle, J.; Kühner, T. Unified Intrinsic and Extrinsic Camera and LiDAR Calibration under Uncertainties. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 6028–6034.
18. Park, Y.; Yun, S.; Won, C.S.; Cho, K.; Um, K.; Sim, S. Calibration between Color Camera and 3D LIDAR Instruments with a Polygonal Planar Board. *Sensors* **2014**, *14*, 5333–5353. [CrossRef]
19. Xu, X.; Zhang, L.; Yang, J.; Liu, C.; Xiong, Y.; Luo, M.; Tan, Z.; Liu, B. LiDAR–camera Calibration Method Based on Ranging Statistical Characteristics and Improved RANSAC Algorithm. *Robot. Auton. Syst.* **2021**, *141*, 103776–103789. [CrossRef]
20. 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. [CrossRef]
21. Ye, Q.; Shu, L.; Zhang, W. Extrinsic Calibration of a Monocular Camera and a Single Line Scanning LiDAR. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 1047–1054.
22. Liao, Q.; Chen, Z.; Liu, Y.; Wang, Z.; Liu, M. Extrinsic Calibration of Lidar and Camera with Polygon. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 200–205.
23. Yao, Y.; Huang, X.; Lv, J. A Space Joint Calibration method for LiDAR and Camera on Self-driving Car and Its Experimental Verification. In Proceedings of the 6th International Symposium on Computer and Information Processing Technology (ISCIPT), Changsha, China, 11–13 June 2021; pp. 388–394.
24. Huang, J.; Grizzle, J. Improvements to Target-Based 3D LiDAR to Camera Calibration. *IEEE Access* **2020**, *8*, 134101–134110. [CrossRef]
25. Huang, J.; Wang, S.; Ghaffari, M.; Grizzle, J. LiDARTag: A Real-Time Fiducial Tag System for Point Clouds. In Proceedings of the IEEE Robot and Automation Letters, 31 March 2021; pp. 4875–4882.
26. Wang, W.; Sakurada, K.; Kawaguchi, N. Reflectance Intensity Assisted Automatic and Accurate Extrinsic Calibration of 3D LiDAR and Panoramic Camera Using a Printed Chessboard. *Remote Sens.* **2017**, *9*, 851. [CrossRef]

27. Pandey, G.; McBride, J.R.; Savarese, S.; Eustice, R.M. Automatic Extrinsic Calibration of Vision and LiDAR by Maximizing Mutual Information. *J. Field Robot.* **2014**, *32*, 696–722. [CrossRef]
28. Taylor, Z.; Nieto, J. A Mutual Information Approach to Automatic Calibration of Camera and LiDAR in Natural Environments. In Proceedings of the Australasian Conference on Robotics and Automation (ACRA), Victoria University of Wellington, Wellington, New Zealand, 3 December 2012; pp. 3–5.
29. Taylor, Z.; Nieto, J. Motion-based Calibration of Multimodal Sensor Extrinsics and Timing Offset Estimation. *IEEE Trans. Robot.* **2016**, *32*, 1215–1229. [CrossRef]
30. Taylor, Z.; Nieto, J. Motion-based Calibration of Multimodal Sensor Arrays. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4843–4850.
31. 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.
32. Iyer, G.; Ram, R.; Murthy, J.; Krishna, K. 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.
33. Yuan, K.; Guo, Z.; Wang, Z. RGGNet: Tolerance Aware LiDAR-Camera Online Calibration with Geometric Deep Learning and Generative Model. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6956–6963. [CrossRef]
34. Lv, X.; Wang, B.; Dou, Z.; Ye, D.; Wang, S. LCCNet: LiDAR and Camera Self-Calibration using Cost Volume Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 2888–2895.
35. Rotter, P.; Klemiato, M.; Skruch, P. Automatic Calibration of a LiDAR–Camera System Based on Instance Segmentation. *Remote Sens.* **2022**, *14*, 2531. [CrossRef]
36. Lim, H.; Myung, H. Patchwork: Concentric Zone-based Region-wise Ground Segmentation with Ground Likelihood Estimation Using a 3D LiDAR Sensor. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6458–6465. [CrossRef]
37. Giyenko, A.; Cho, Y. Intelligent UAV in Smart Cities Using IoT. In Proceedings of the 16th international Conference on Control,Automation and Systems (ICCAS'16), Gyeongju, Korea, 16–19 October 2016; pp. 207–210.
38. Unnikrishnan, R.; Hebert, M. *Fast Extrinsic Calibration of a Laser Rangefinder to a Camera*; Robotics Institute: Pittsburgh, PA, USA, Technical Report, CMU-RI-TR-05-09; 2005.
39. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [CrossRef]
40. Kneip, L.; Li, H.; Seo, Y. UPnP: An optimal O(n) Solution to The Absolute Pose Problem with Universal Applicability. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 127–142.
41. Grammatikopoulos, L.; Papanagnou, A.; Venianakis, A.; Kalisperakis, I.; Stentoumis, C. An Effective Camera-to-LiDAR Spatiotemporal Calibration Based on a Simple Calibration Target. *Sensors* **2022**, *22*, 5576. [CrossRef]
42. Nnez, P.; Jr, P.D.; Rocha, R.; Dias, J. Data Fusion Calibration for a 3D Laser Range Finder and a Camera Using Inertial Data. In Proceedings of the European Conference on Mobile Robots (ECMR), Dubrovnik, Croatia, 23–25 September 2009; pp. 31–36.
43. Kim, E.; Park, S. Extrinsic Calibration between Camera and LiDAR Sensors by Matching Multiple 3D Planes. *Sensors* **2020**, *20*, 52. [CrossRef] [PubMed]
44. Chai, Z.; Sun, Y.; Xiang, 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.