Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Extrinsic Calibration Between a Stereo System and a 3D LIDAR

Yanwu Zhai, Haibo Feng*, Jia He, Enbo Li, Songyuan Zhang and Yili Fu*

Abstract— **As the robot gradually develops from indoor to outdoor, multi-sensor fusion is increasingly being applied to the field of robot perception. Among them, LiDAR and binocular sensors are most widely used. Robots use the complementary and redundant data they provide to perform environmental perception and implement certain functions of the robot, such as SLAM, autonomous obstacle avoidance and so on. However, the data obtained by each sensor are relative to its own coordinate system. Data must be converted to a unified coordinate system before data fusion. Therefore, the calibration of the transformation among sensors is the first problem to be solved in data fusion. Moreover, the accuracy of calibration is directly related to the effect of data fusion. Therefore the calibration between sensors is very important and necessary. In this paper, we propose a novel method to find accurate rigid-body transformation for the extrinsic calibration of a 3D LiDAR and a stereo cameras, using two Aruco calibration boards. Also the validity of the algorithm is demonstrated by some experiments.**

I. Introduction

Recent years, the research hotspot of autonomous navigation for robots has gradually shifted from indoor to outdoor. As the outdoor environment is always complex and changeable, and only depending only one single sensor to obtain information cannot meet the requirements. Multi-sensor fusion technology is increasingly used in robots, AGV, driverless vehicles and other platforms [1]. Among them, binocular cameras and 3D LIDAR are the most widely used. Compared with laser radar, the camera can provide rich texture information and color information, which is often used in deep learning algorithms for image recognition and classification, extracting semantic information. However, visual information processing requires a lot of computation and is very sensitive to light. Moreover, the effective observation distance of the camera is positively correlated with its baseline. These shortcomings of binocular cameras [2] significantly limit its further application[2]. 3D LIDAR can directly provide the depth information of the environment, which greatly reduces the computational cost. But the resolution of LIDAR decreases sharply with the increase of detection distance. Moreover, the LIDAR can't get the texture information of the surrounding environment, which is not conducive to loop detection in SLAM [3]. Therefore, outdoor robots are usually equipped with a variety of sensors to reduce environmental perception errors. However, the data of each sensor is relative to their own coordinate system, so before data fusion, extrinsic calibration must be performed to transform their data into the same coordinate system. The quality of the calibration results will greatly affect the results of the subsequent fusion algorithm [4]. To this end, we propose an approach to calibrate the extrinsic of binocular camera and 3D LIDAR. We use the geometric constraints between the calibration plate and the ground to calculate the transformation relationship between the camera frame and LIDAR frame. Then, with this value as the initial value and the corresponding points between the two coordinate systems as constraints, the non-linear optimization is carried out to minimize these constraints. Finally, we get the exact optimization results. The rest of the paper is structured as follows. Some related work on extrinsic Calibrating Between a stereo system and a 3D LIDAR will be introduced in the section II, which is the fundamental work of our approach introduced in the following part. Section III presents details of our algorithm. In the section IV, we will design some experiments to verify our algorithm. Finally, in Section V, we will summarize our work and further elaborate directions for future work.

II. RELATED WORK

Recent years, several methods were proposed to calibrate multi-sensor System. These methods can be divided into two classes: the on-line and off-line calibration. On-line calibration is a type of real-time method that combines extrinsic calibration with data fusion. Nedevschi S [5] et al proposes an online calibration algorithm for binocular systems, which is mainly applied to far-distance, vision-base vehicles. This method can greatly improve the accuracy of vehicle work. Off-line calibration means that the calibration of external parameters must be completed before data fusion and no longer be calibrated in the process of data fusion. On-line calibration increases the computational cost of processors, and the corresponding features are not easy to find, and the results are inferior to off-line calibration. Therefore, offline calibration is generally used in engineering. In order to find the corresponding features between sensors more easily and accurately, calibration boards calibration boards is often used to determine matching points. Weimin Wang et al [6] proposed an Extrinsic Calibration algorithm based on a printed chessboard for 3D LIDAR and panoramic camera. The algorithm uses the reflected intensity of the lidar as an aid to find the corner points of the board and then matches the image obtained by the panoramic camera. Ankit Dhalle et al

Research supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No.51521003).(corresponding author:Yili Fu and Haibo Feng)*

Yanwu Zhai, Haibo Feng, Jia He, Enbo Li, Songyuan Zhang and Yili Fu are with State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, Heilongjiang Province, 150001, China, fax: +86-0451-86414422; zslhit@126.com, haibo_feng@hotmail.com,1694081086@qq.com, li_en_bo@163.com, zhangsy@hit.edu.cn, zcbo96@163.com, meylfu@hit.edu.cn

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

[7] find the corresponding 3D points in the coordinate system of LIDAR and binocular vision with Aruco mark, and then obtain the 3D-3D point correspondences in LiDAR frame and binocular camera frame and get a closed form solution by using ICP algorithm. Zhaozheng Hu et al [8] extend the chessboard pattern and use it as the calibration target for extrinsic calibration. There are many other reasearch works, They usually use the calibration board to find the corresponding points of the two coordinate systems first, and then solve the transformation relationship between the two coordinate systems. Xiaojin Gong[9] et al takes advantage of a trihedron instead of using planar checkerboard patterns for calibration. This method is very convenient, because whether in indoor or outdoor unstructured environments, such trihedral targets are ubiquitous. [10] and [11] use line feature and circle feature matching in environment to calibrate. However, the LIDAR also has its inherent disadvantages. Compared with the binocular camera, its resolution is low and the measurement distribution is uneven, which makes it difficult to obtain high accuracy. To solve this problem, we propose an accurate and convenient method to calibrate binocular camera and 3D LIDAR, Our work is based on [7] and [9].

III. ALGORITHM

In this section, our algorithm for calibrating the Stereo System and 3D LIDAR is described in detail. There are three key steps to obtain the conversion relationship between LIDAR and binocular camera. The first step is to find the transformation relation according to geometric constraints as the initial value of optimization. The second step is to move the calibration board to find out several sets of corresponding points and optimize them by LM method. The third step is data fusion based on the results obtained.

A. Geometric constraint

Figure 1 shows the platform we need to calibrate.The platform consists of a MYNTEYE binocular camera and a VLP 3D laser. The binocular camera outputs 24 frames per second, always facing forward. The VLP-16 produces 300,000 points per second with a 360 degree field of view. Before starting the camera-LiDAR calibration process, we must calibrate the binocular camera and the LiDAR respectively. In the calibration process, two Aruco calibration boards are observed synchronously by them, and their placement is not parallel to the ground. We finally have to get the relative transformation between between the camera and the LIDAR by their constraint. This constraint can be assisted by the ARUCO calibration boards. In order to obtain accurate results, we keep the sensor system fixed during the experiment, and the calibration boards moves to obtain multiple configurations. As shown in Figure 1, it is our experimental platform. In such configurations, several reference frames should be defined: Camera frame: the binocular coordinate system coincides with its left camera. The frame is represented by $\{C_i\}$, in which $i = 1, 2 \cdots N$ indicates the i_{th} configuration. LIDAR frame: the coordinate system of the LIDAR is consistent with the frame of the factory default setting. The LIDAR frame is denoted by$\{L_i\}$, in which $i = 1, 2 \cdots N$ indicates thei_{th}configuration, as shown in Figure 1. World reference frame: The world coordinate system is a virtual coordinate system. We expand the two calibration boards to intersect with the ground, the origin of the world coordinate system coincides with the intersection of the three planes, the x-axis coincides with the intersection line of one of the calibration boards, and is denoted by $\{W_i\}$ in which $i = 1, 2 \cdots N$ indicates the i_{th} configuration. In the camera coordinate system, we fit the plane of two calibration plates and ground plane through the Aruco board. Each Aruco calibration board provides a lot of accurate corner information. Through these corners, we can fit a relatively accurate plane equation. The ground plane can be fitted by finding feature points, and only once in the experiment. These three planes are represented as $\left\{\mathbf{N}_j^c, d_j^c\right\}$, in which $j = 1, 2, 3$. The plane equation has been transformed into a standard form, so we have$\left\|\mathbf{N}_j^{c\tau}\right\| = 1$. At the same time, we use the to represent the$\mathbf{R}_{\text{urc}} = \left[\mathbf{r}_{wc}^1 \mathbf{r}_{uc}^2 \mathbf{r}_{uc}^3\right], \mathbf{t}_{vc} = \left[t_{wc}^1 t_{wc}^2 t_{wc}^3\right]^T$ transformation from the world coordinate system to the camera coordinate system, in which represent the column vector of the rotation matrix. The origin in the world coordinate system is on these three planes, we have:

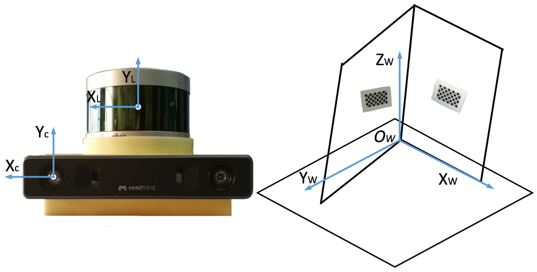

Fig. 1. Calibration Platform and Its Coordinate System Representation.

$$\mathbf{N}_j^{cT}\left(\mathbf{R}_{vc}\begin{bmatrix}0\\0\\0\end{bmatrix}+\mathbf{t}_{wc}\right)+d_j=0 \tag{1}$$

In which$j = 1, 2, 3$. From Equation 1, we can calculate the translation vectort_{wc} from the world coordinate system to the camera coordinate system.

$$\mathbf{t}_{wc} = \left[\mathbf{N}_1^c \mathbf{N}_2^c \mathbf{N}_3^c\right]^{-T}\left[d_1^c d_2^c d_3^c\right]^T \tag{2}$$

In addition, the points$[1\ 0\ 0]^T$in the world coordinates are on the plane 1 and plane 3, so in the camera coordinates. We have:

$$\mathbf{N}_j^{cT}\left(\mathbf{R}_{\text{vc}}\begin{bmatrix}1\\0\\0\end{bmatrix}+\mathbf{t}_{wc}\right)+d_j=0 \tag{3}$$

In which$j = 1, 3$, by substituting equation (1) into equation (3), we can get:

$$\begin{gathered}\mathbf{N}_1^{cT}\mathbf{r}_{wc}^1=0\\ \mathbf{N}_3^{cT}\mathbf{r}_{wc}^1=0\end{gathered} \tag{4}$$

At the same time, the distance from point$[0\ 0\ 1]^T$to plane 3 is 1. Converts this point to camera coordinate system, the distance from point to plane remains unchanged. We have:

$$\mathbf{N}_3^{cT}\left(\mathbf{R}_{wc}\begin{bmatrix}0\\0\\1\end{bmatrix}+\mathbf{t}_{wc}\right)+d_3^c=1 \tag{5}$$

By substituting equation (1) into equation (6), we can get:

$$\mathbf{N}_3^{cr}\mathbf{r}_{we}^3=1 \tag{6}$$

Comprehensive Equation 4, Equation 5 and Equation 7 we can get:

$$\begin{aligned}\mathbf{r}_{wc}^1&=\mathbf{N}_1^c\times\mathbf{N}_3^c\\ \mathbf{r}_{wc}^3&=\mathbf{N}_3^c\\ \mathbf{r}_{wc}^2&=\mathbf{r}_{uc}^3\times\mathbf{r}_{wc}^1\end{aligned} \tag{7}$$

In the Lidar coordinate system, we can use the 3D points falling on the calibration board and ground to fit the three plates. Similarly, the ground equation only needs to be fitted once, and they are represented as$\left\{\mathbf{N}_j^l,d_j^l\right\}$, in which$j=1,2,3$. The plane equation has been transformed into a standard form, so we have$\left\|\mathbf{N}_j^{|T|}\right\|=1$. At the same time, we use the$\mathbf{R}_{wl}=\left[\mathbf{r}_{wl}^1\mathbf{r}_{wl}^2\mathbf{r}_{wl}^3\right]$,$\mathbf{t}_{wl}=\left[t_{wl}^1t_{wl}^2t_{wl}^3\right]^T$to represent the transformation from the world coordinate system to the LIDAR frame, in which$\mathbf{r}_{wl}^i$ represent thei_{th} column vector of the rotation matrix. By using the same method, we can get:

$$\begin{aligned}\mathbf{r}_{wl}^1&=\mathbf{N}_1^l\times\mathbf{N}_3^l\\ \mathbf{r}_{wl}^3&=\mathbf{N}_3^l\\ \mathbf{r}_{wl}^2&=\mathbf{r}_{\mathbf{r}l}^3\times\mathbf{r}_{wl}^1\end{aligned} \tag{8}$$

$$\mathbf{t}_{wl}=\begin{bmatrix}\mathbf{N}_1' & \mathbf{N}_2' & \mathbf{N}_3'\end{bmatrix}^{-T}\begin{bmatrix}d_1' & d_2' & d_3'\end{bmatrix}^T \tag{9}$$

So far, we obtained the transformation from the world coordinate system to the camera coordinate system.

$$\mathbf{T}_{wc}=\begin{bmatrix}\mathbf{R}_{wc} & \mathbf{t}_{wc}\\ \mathbf{0} & 1\end{bmatrix} \tag{10}$$

And the transformation from the world coordinate system to the LIDAR coordinate system.

$$\mathbf{T}_{wl}=\begin{bmatrix}\mathbf{R}_{wl} & \mathbf{t}_{wl}\\ \mathbf{0} & 1\end{bmatrix} \tag{11}$$

Now, in order to obtain the transformation between the lidar coordinate system and the binocular visual coordinate system, we chain the transforms, $\mathbf{T}_{wc}$ and $\mathbf{T}_{wl}$.

$$\mathbf{T}_d=\mathbf{T}_{w_w}^{\mathbf{l}}\mathbf{r}_w \tag{12}$$

Through the above geometric constraints, we can get the transformation relationship between camera coordinate system and radar coordinate system, but due to the existence of noise, the plane fitting will have errors. So, we can only get one raw data. However, we can take this result as the initial value of the next optimization, which can greatly reduce the number of iterations and avoid falling into local optimum.

B. Feature point constraints

In order to achieve accurate results, we can use corresponding 3D points in the camera frame and the LIDAR frame to restrict the transformation between the camera and LIDAR. For velodyne LiDAR, its low resolution does not provide a dense point cloud, and the farther away from the LIDAR, the more sparse the cloud. Therefore, we can't get texture information. However, we can use the 3D points falling on the calibration board to fit the boundary and calculate the intersection points of the boundary. RanSaC is used to fit lines on the points from the LiDAR. In each configuration, two calibration boards can provide 8 3D corners. By fixing the sensor and moving the calibration board, multiple sets of three-dimensional points can be obtained. For camera, it can provide rich texture and color information. It's relatively easy to find the corresponding corner. We can directly find the boundary line of the calibration plate by Hough transform, and further calculate the intersection point. These intersections are the corresponding ones we are looking for. We can also find corners through Aruco calibration board. The position relationship between Aruco and the four corners of the calibration board is known, and the position of Aruco in the camera coordinate system is easy to determine. Once we get the two sets of point correspondences, we can use ICP algorithm to calculate the transformation relationship between camera frame and LIDAR frame$\mathbf{T}_{cl}$. The ICP algorithm obtains the transformation between two coordinate systems by minimizing the distance deviation of the corresponding 3D points, which can be expressed by Equation (13).

$$\underset{R\in SO(3),t\in R^3}{\arg\min}\sum_{i=1}^{n}\sum_{j=1}^{8}\left\|(\mathbf{R}P_{ij}+\mathbf{t})-\mathbf{Q}_{ij}\right\|^2 \tag{13}$$

In which point $\mathbf{P}_{ij}$ represented the j_{th} corresponding point in the i_{th} configuration in the camera coordinate system. And point$\mathbf{Q}_{ij}$represented thej_{th} corresponding point in thei_{th} configuration in the LIDAR coordinate system. $\mathbf{R}$and$\mathbf{t}$ represents rotation matrix and translation vector from camera coordinate system to LIDAR coordinate system. We merge all points and get:

$$\min_T\frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{T}\mathbf{P}_i-\mathbf{Q}_i\right\|^2 \tag{14}$$

In which$N=8n$,$\mathbf{T}=\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix}$, $\mathbf{P}_i$and$\mathbf{Q}_i$ are homogeneous coordinate.

We use the exponential mapping (at the identity) exp: se(3) → SE(3) to associate the Lie algebra with the rotation matrix, and the transformation relationship is like the Equation (15):

$$\mathbf{T}=\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix}=\exp\left(\xi^\wedge\right)=\begin{bmatrix}\exp\left(\phi^\wedge\right) & \mathbf{J}\rho\\ \mathbf{0} & 1\end{bmatrix} \tag{15}$$

In which$\exp\left(\phi^\wedge\right)$ can be obtained by Rodrigues formula:

$$\begin{aligned}\exp\left(\phi^\wedge\right)&=\exp\left(\theta\mathbf{a}^\wedge\right)\\&=\cos\theta\mathbf{I}+(1-\cos\theta)\mathbf{a}\mathbf{a}^T+\sin\theta\mathbf{a}^{\mathbf{l}}\end{aligned} \tag{16}$$

And $\mathbf{J}$ can be obtained by formula (17):

$$\mathbf{J}=\frac{\sin\theta}{\theta}\mathbf{I}+\left(1-\frac{\sin\theta}{\theta}\right)\mathbf{a}\mathbf{a}^{T}+\frac{1-\cos\theta}{\theta}\mathbf{a}^{\wedge} \tag{17}$$

So the optimization equation can be expressed by Lie algebra:

$$\min_{\xi}\frac{1}{2}\sum_{i=1}^{N}\left\|\exp\left(\xi^{\wedge}\right)\mathbf{P}_{i}-\mathbf{Q}_{i}\right\|^{2} \tag{18}$$

Equation (18) is a nonlinear equation for transform parameters and is also a least squares problem. This equation can obtain an optimal solution by nonlinear optimization. In this experiment, we use the ceres for optimization, and the initial value is the result we calculated in the first step. The Jacobian matrix of the function can be expressed as:

$$\mathbf{J}_{ac}=\frac{\partial\mathbf{e}}{\partial\delta\xi}=\left[\begin{array}{cc}\mathbf{I} & -\left(\mathbf{R}\mathbf{P}_{i}+\mathbf{t}\right)^{\wedge}\\ \mathbf{0}^{T} & 0^{T}\end{array}\right] \tag{19}$$

Then, we use the Levenberg–Marquardt method for iterative solution. We first give the system an initial value $\mathbf{T}_0$ and then iteratively updates the parameters over the equation (20) until the system converges:

$$\mathbf{T}_{t+1}=\mathbf{T}_{t}+\Delta\mathbf{T}_{t} \tag{20}$$

Where$\Delta\mathbf{T}$is obtained by solving the following equation:

$$\left(\mathbf{J}^{T}\mathbf{J}+\lambda\,\mathrm{diag}\left(\mathbf{J}^{T}\mathbf{J}\right)\right)\Delta\mathbf{T}_{t}=\mathbf{J}^{T}\left(-f\left(\mathbf{T}_{t}\right)\right) \tag{21}$$

Then the initial values$\mathbf{T}_0$ can be given by the geometric constraints of the first step.

IV. EXPERIMENT RESULT

In our experiments, we are using the MYNTEYE stereo camera, whose coordinate system coincides with that of its left camera. Some intrinsic parameters of the camera are as follows: frame rate is 20 Hz, pixel size is $d_x = d_y = 6\mu m$, image resolution is 752*480, base line is 120 mm, focal length is 2.1 mm. MYNTEYE-SDK provides simple API to record information of image, so it is convenient to record some data sets to support the experiments. As for LIDAR, we choose VLP-16 velodyne, which is a 16-line LIDAR. It's measurement range is 100 meters, and has 360 degrees horizontal field of view and 30 degrees vertical field of view. It can generate 300000 3D points per second. We designed a pedestal to fix the camera and the LIDAR. The LIDAR is fixed to the top of the pedestal and can scan a 360-degree field of vision around it. The camera is fixed in front of the pedestal, and only the local field of vision in front can be observed. In the experiment, two calibration boards should be kept in the common field of view of LIDAR and binocular camera.

We implement the proposed method in the following experiments. Firstly, two calibration plate planes and ground planes are fitted in camera coordinate system and Lidar coordinate system respectively. According to the method of section III, the transformation relationship between the two coordinate systems is calculated and used as the initial value of the later optimization. Then the relative position of the calibration plate is constantly changed to obtain multiple configurations. The experiment is shown in Fig. 2. Finally, Ceres is used to optimize and get accurate results.

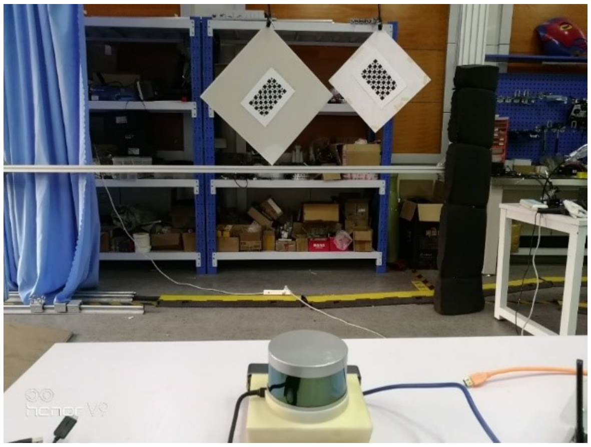

Fig. 2. Experimental platform.

A. Experiments on solving geometric constraints

In the camera coordinate system, we use Aruco to extract points on the plane, and use the extracted points to fit the plane, as shown in Figure 3. In the process of plane fitting, RanSaC is used to remove outliers, which can reduce the error of Ceres optimization. Finally, we standardize the coefficients of the fitted plane equation. In the Lidar coordinate system, we manually select points on the plane and use the PCL library to remove outliers, as shown in Figure 4. In the process of plane fitting, RanSaC is used to remove outliers again, which can reduce the error of Ceres optimization. Finally, we standardize the coefficients of the fitted plane equation. In the process of plane fitting, plane 1 is the plane where the large calibration plate is located, plane 2 is the plane where the small calibration plate is located, and plane 3 is the ground.

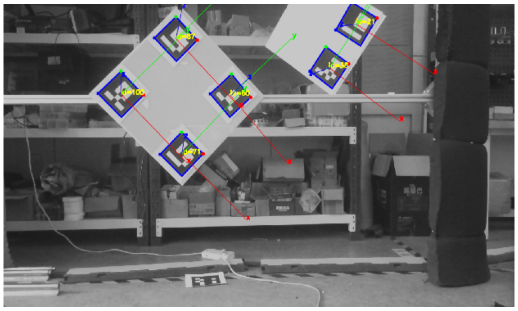

Fig. 3. Plane Point Recognition in Camera Coordinate System.

Based on the results of plane fitting and the geometric constraint algorithm in Section III, we can roughly calculate the transformation relationship between camera coordinate system and Lidar coordinate system as the initial value of

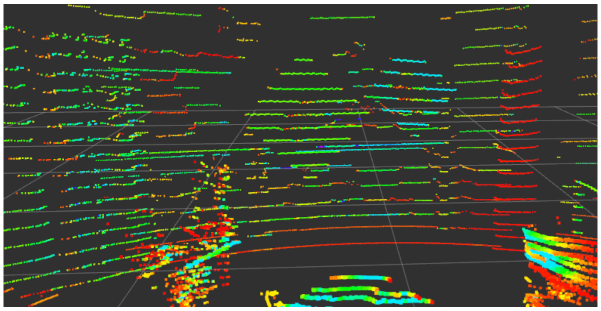

Fig. 4. Plane Point Recognition in Lidar Coordinate System.

the next optimization:

$$\mathbf{R}' = \begin{bmatrix} 1.2406 & 0.0435 & 0.0626 \\ 0.1613 & 0.9256 & 0.2847 \\ 0.7224 & 0.1535 & 1.1404 \end{bmatrix} \tag{22}$$

$$\mathbf{t}' = \begin{bmatrix} -62.2323 \\ 62.4248 \\ -45.3417 \end{bmatrix} \tag{23}$$

B. Experiments of 3D-3D points optimization

The results obtained by geometric constraints are more sensitive to noise. The results are not very accurate. Therefore, other constraints must be introduced to get a better result. 3D-3D Point constraints are the most effective and easiest to find. So we choose the corresponding 3D points to optimize. The most prominent feature on the board is the corner. Corner points can be easily identified on pictures, and more mature corner detection algorithms have been integrated in opencv. But for Lidar, we must fit the edge line of the calibration board. During the experiment, each edge of the boards cannot be parallel to the ground, due to the horizontal nature of the LiDAR's scan lines one can obtain the vertical edges, but not necessarily the horizontal ones. In this way, we can use the RANSAC method to fit the four edges of each calibration board, their intersection is calculated in 3D. Collecting data over multiple experiments, Ceres is used for further optimization. Eventually we get an exact result:

$$\mathbf{R} = \begin{bmatrix} 1.0031 & 0.0012 & 0.0021 \\ 0.0018 & 0.9925 & 0.0032 \\ 0.0053 & 0.0038 & 1.0017 \end{bmatrix} \tag{24}$$

$$\mathbf{t} = \begin{bmatrix} -67.2354 \\ 60.4032 \\ -51.6324 \end{bmatrix} \tag{25}$$

After obtaining the results, we transform the 3D points in LiDAR coordinate system into the image plane, comparing with the corresponding feature points in the image plane. In this process, RanSaC is used to remove outliers. In order to test the effect of optimization, we back-projected 3D points to the image plane and observed the error of the pixels. Before optimization, using only the extrinsic parameters obtained by the geometric constraints, we get an average back projection error of about 10 pixels. After optimization, the mean value of back projection erroris is less than 3 pixels, which proves that the results obtained by our method are quite accurate.

C. Experiments of Data fusion

Data fusion is the most effective method to verify the calibration results. Poor calibration results can greatly affect the effect of data fusion, such as ghosting in point cloud fusion. Therefore, data fusion can give us an intuitive feeling about the quality of the calibration results.

If we have external parameters of binocular vision, that is, the transformation matrix of the left and right cameras, we can fuse the data of the left and right images to obtain point cloud data and reconstruct the 3D environment. We first run our algorithm, with left camera of the binocular camera and LiDAR, and obtain a transformation matrix that transforms all points in the LiDAR frame to the left camera frame:

$$\mathbf{T}_{lcl} = \begin{bmatrix} \mathbf{R}_{lcl} & \mathbf{t}_{lcl} \\ \mathbf{0} & 1 \end{bmatrix} \tag{26}$$

Then we run the algorithm again, with right camera of the binocular camera and LiDAR, and obtain a transformation matrix that transforms all points in the LiDAR frame to the right camera frame:

$$\mathbf{T}_{lcr} = \begin{bmatrix} \mathbf{R}_{lcr} & \mathbf{t}_{lcr} \\ \mathbf{0} & 1 \end{bmatrix} \tag{27}$$

Now, in order to get a transform that transforms all points in right camera of the binocular camera to the left camera, we chain the transforms, $\mathbf{T}_{lcl}$ and $\mathbf{T}_{lcr}$:

$$\mathbf{T}_{crcl} = \mathbf{T}_{lcl}\mathbf{T}_{lcr}^{-1} \tag{28}$$

Equation (28), finds the transform from right camera of the binocular camera to the left camera, we can fuse left and right images of binocular cameras to get dense point clouds using this transform. Figure 7 shows the effect of fusion. As can be seen from the Fig.5, although some point clouds do not have data (possibly due to the camera itself), there is no hallucinations, which proved that the results obtained are quite accurate.

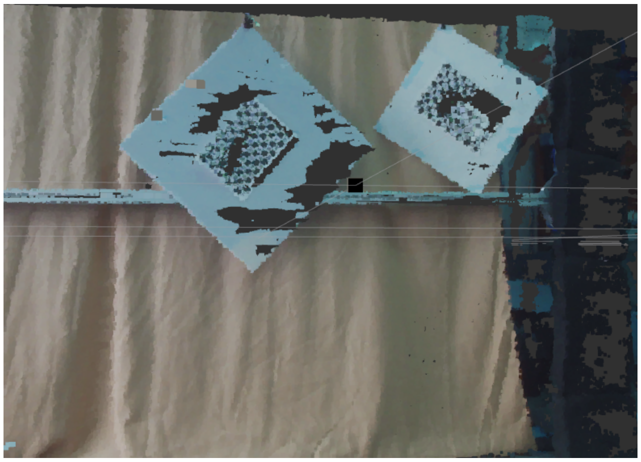

Fig. 5. Dense Point Cloud obtained from Binocular Cameras.

V. Conclusions and Future Work

In this paper, we proposed a novel pipeline to calibrate the extrinsic of the LIDAR-camera systems using Aruco boards. Then we obtained the transformation between the LIDAR frame and the camera frame through experiments. The binocular vision fusion is used to visually verify the calibration results. Unlike other methods, we provide a good initial value for 3D-3D point optimization using geometric constraints between the calibration board and the ground, which reduces the number of iterations in the optimization process and avoids falling into local optimum. At the same time, we do many experiments to collect enough data to ensure the accuracy of the optimization results. Finally, the fusion experiments of left and right images verify the accuracy of our algorithm. In addition, this algorithm can be applied to any other camera and LIDAR calibration experiments, not just the calibration of monocular cameras or binocular cameras and LIDAR. In the future, Sensor timestamp alignment will be scheduled. For multi-sensor systems, not only space calibration, but also time calibration will have a great impact on future applications.

ACKNOWLEDGMENT

The authors would like to thank the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No.51521003), Heilongjiang Postdoctoral Scientific Research Foundation (AUGA4120001817), the self-managed project of the State Key Laboratory of Robotics and System in Harbin Institute of Technology (SKLRS201801A01, SKLRS201801A02).

References

[1] Liu, Zhi Xiang , et al. ”Mobile robot positioning method based on multi-sensor information fusion laser SLAM.” Cluster Computing (2018).

[2] Mur-Artal, Raul , and J. D. Tardos . ”ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras.” (2016).

[3] Zhang, Ji , and S. Singh . ”Low-drift and real-time lidar odometry and mapping.” Autonomous Robots 41.2(2017):401-416.

[4] Li, You , Y. Ruichek , and C. Cappelle . ”Optimal Extrinsic Calibration Between a Stereoscopic System and a LIDAR.” IEEE Transactions on Instrumentation&Measurement 62.8(2013):2258-2269.

[5] Forster, Christian , et al. ”On-Manifold Preintegration for Real-Time Visual-Inertial Odometry.” IEEE Transactions on Robotics PP.99(2016):1-21.

[6] Weimin, Wang , S. Ken , and K. Nobuo . ”Reflectance Intensity Assisted Automatic and Accurate Extrinsic Calibration of 3D LiDAR and Panoramic Camera Using a Printed Chessboard.” Remote Sensing 9.8(2017):851-.

[7] Dhall, Ankit , et al. ”LiDAR-Camera Calibration using 3D-3D Point correspondences.” (2017).

[8] Hu, Zhaozheng , et al. ”Registration of image and 3D LIDAR data from extrinsic calibration.” 2015 International Conference on Transportation Information and Safety (ICTIS) IEEE, 2015.

[9] Gong, Xiaojin , Y. Lin , and J. Liu . ”3D LIDAR-Camera Extrinsic Calibration Using an Arbitrary Trihedron.” Sensors 13.2(2013):1902-1918.

[10] Li, Ganhua , et al. ”An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features.” Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on IEEE, 2007.

[11] Rodriguez, F , Fremont, and Bonnifait. ”Extrinsic calibration between a multi-layer lidar and a camera.” IEEE International Conference on Multisensor Fusion&Integration for Intelligent Systems IEEE, 2008.