

Extrinsic Calibration of the 2D Laser Rangefinder and Binocular Cameras System Base on Virtual Points

CHANG Wenkai¹, LI En¹, YANG Guodong¹, LIANG Zize¹, WU Yi², LIE Jianping²

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190
E-mail: {changwenkai2013,en.li,guodong.yang,zize.liang}@ia.ac.cn

2. State grid general avlation company limited, 150001, China
E-mail: {okstill, liejp}@163.com

Abstract: In this paper a novel method is proposed to calibrate the extrinsic parameters of the laser rangefinder (LRF) and binocular cameras system. Currently, many calibration methods based on 2D laser rangefinder and signal camera have been investigated by extracting observable feature points in images and point clouds matching to calculate the relative translation and rotation between the coordinate frames. Superior to the monocular vision, stereo vision can obtain more information of scene, such as 3D planes. In this paper, virtual points and lines are deduced by analyzing the intersected 3D planes and they are used as inputs to an optimization problem which requires jointly estimating the relative translation and rotation between left camera and LRF. Moreover, two parameters for LRF geometric model can also be estimated. The proposed method only need range data of LRF and scan once of scene. It is suitable for various types of LRF. Experimental results demonstrate the validity of the proposed method.

Key Words: Calibration, 2D Laser Rangefinder, Binocular Cameras System

1 Introduction

In recent years, more and more robots are equipped with 2D LRF and cameras [1] [2]. It is mainly divided into two categories follow the installation of LRF and cameras.

The first one, relative pose between camera and LRF is constant. They are mounted on the robot or moved together. The other one, the relative pose is variable. 2D LRF scan around an external axis with the fixed cameras is a typical installation for the latter. Nevertheless, both installations are required extrinsic calibration to reconstruct the environment with both color and 3D information.

Calibration methods for the former prefer to use geometric constraint, such as, point-to-line and line-to-plane. Wasielewski and Strauss [3] proposed a method based on V-shaped pattern. They calculate the intersection point of the LRF scan plane with the V-shaped target, and then minimizes the distance from the projected point to the line extracted in image. Kiho Kwak [4] consider the sparsity of the points from LRF and calculate the virtual intersection point and virtual end-points to improve performance of method that base on V-shaped pattern. Different from the

above methods, Zhang [5] and Vasconcelos [6] make use of a planar checkboard pattern. They estimate checkboard pose in image and combine point-to-plane to complete the calibration. But the initial value affect the result greatly. In general, the methods above all need a large number of observations to get good results.

Calibration methods for the latter one prefer to use special checkboard to get as much feature points as possible in one scanning. Yan and Fei [7] make holes array on a large checkboard and then minimizes the distance between the projected points of holes centers and the virtual centers calculated in image. Jiyoung [8] [9] makes use of a novel 2.5D checkboard with more holes in random distribution. Thus, each holes gets a unparalleled position ID. It improves the robustness of the matching. But it also need many

checkboards or many observations to achieve good performance. Peyman and Michael [10] propose a method exploits natural linear features in the scene. They extract 3D lines from the point cloud and 2D line segments from image. And the data sets are used as inputs to an optimization problem. It only scan once of scene, but they need more objects with regular shape (e.g. cuboid) to get a better performance.

In this paper, we use planar checkboards as markers of target planes. Compared with the methods based on natural linear, it is easier to control the position and number of feature points. In general, it is a contradiction to get more feature points or scan scene more times. We take advantage of stereo vision to create enough feature points in one scanning.

The rest of this paper is organized as follows: Section 2 : convert LRF data from polar coordinates to Cartesian coordinates. Section 3 : check and calculate 3D information of target planes by binocular cameras and extract planes from segmented point cloud and recognize the matching one. Section 4 : calculate virtual feature points and optimization. Section 5 : the experimental results. Section 6 is summary.

2 Laser Data Transition

Figure 1 shows the configuration of our platform and the coordinate frames are shown in Figure 2. Without loss ofgenerality, let left camera coordinate O_c be the binocular

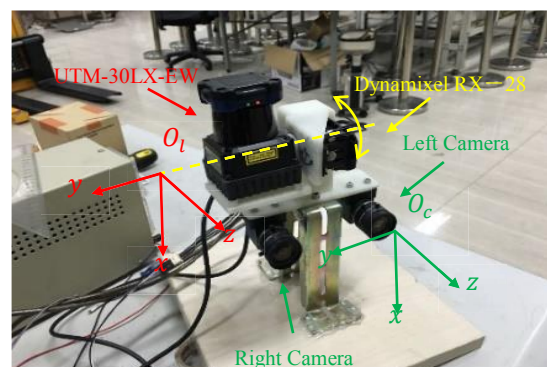


Fig. 1. The Installation position of LRF and binocular cameras

* This work was supported in part by The National Natural Science Foundation (NNSF) of China 61403374 and The National Science and Technology Support Program of China (2015BAK06B01)

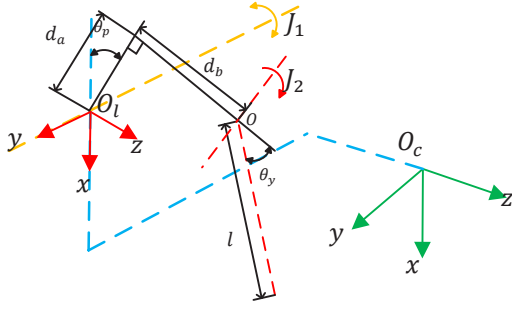


Fig. 2. The coordinate frames of platform

cameras coordinate. y axis of LRF coordinate O_l coincides with joint J_1 . The servo motor periodically feedbacks the pitching angle α_p of LRF. There is a high-speed motor in LRF rotating around J_2 . And then we can periodically obtain the range l and yaw angle α_y . In addition, we add two parameters d_a and d_b to response the real position of launch point.

The relative translation matrix between O_l and O_c is T_{lc} and the rotation matrix is R_{lc} . The 3D laser scanning data B^L in the Cartesian coordinate system can be calculated as

$$B^L = [x^l \ y^l \ z^l]^T \quad (1)$$

$$\begin{bmatrix} x^l \\ y^l \\ z^l \end{bmatrix} = \begin{bmatrix} \cos \theta_y \cos \theta_p & \cos \theta_p & \sin \theta_p \\ \cos \theta_y \sin \theta_p & \sin \theta_p & -\cos \theta_p \\ \sin \theta_y & 0 & 0 \end{bmatrix} \begin{bmatrix} l \\ d_b \\ d_a \end{bmatrix} \quad (2)$$

Considering the efficiency of the calibration, the stop-scan-go fashion is ignored. In actually, LRF and motor independently run and feedback data. Each data set has two additional timestamps and the data set is expressed as

$$S_{t_{begin}^{t_{end}}}^L = [\mathbf{L} \ \boldsymbol{\alpha}] \quad (3)$$

where the t_{begin} and t_{end} is the begin and end moment of scanning. \mathbf{L} and $\boldsymbol{\alpha}$ are the range and yaw angle sequences in the data set. The details shown as

$$\mathbf{L} = [l_1 \ \dots \ l_i \ \dots \ l_N]^T \quad (4)$$

$$\boldsymbol{\alpha} = [\theta_{y1} \ \dots \ \theta_{yi} \ \dots \ \theta_{yN}]^T \quad (5)$$

Since the feedback cycle of servo motor is much shorter than the LRF. And the rotational speed is a constant in most of time. Wherefore, we can calculate the real pitching angle at t_{begin} and t_{end} by the linear relationship between time and pitching angle. This condition is expressed with θ_p^{start} as an example as

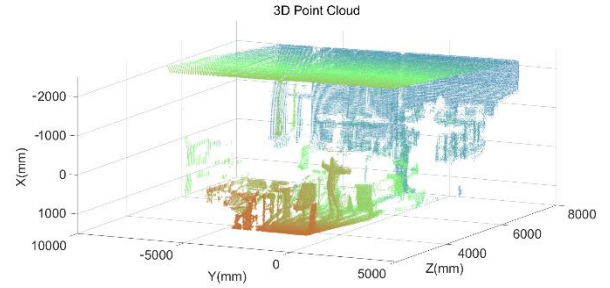
$$\theta_p^{start} = \begin{cases} \frac{(t_{begin}-t_1)}{(t_2-t_1)}([\theta_p^{t_2}]_{i+1} - [\theta_p^{t_1}]_i) + [\theta_p^{t_1}]_i; t_{begin} \geq t_1 \\ \frac{(t_{begin}-t_2)}{(t_1-t_2)}([\theta_p^{t_1}]_i - [\theta_p^{t_2}]_{i-1}) + [\theta_p^{t_1}]_{i-1}; t_{begin} < t_1 \end{cases} \quad (6)$$

where θ_p^t is the pitching angle at moment t . $[\theta_p^t]_i$ is the sequence of θ_p^t and $[\theta_p^{t_1}]_i$ means the angle in sequence with timestamp t_1 closest to t_{begin} . θ_p^{end} is calculated in the same way. After this, we can calculate the real pitching angle θ_{pi} for each element in \mathbf{L} or $\boldsymbol{\alpha}$. The solution is given as

$$\theta_{pi} = \frac{(\theta_p^{end} - \theta_p^{start})}{N-1} * (i-1) + \theta_p^{start} \quad (7)$$

It is noteworthy that, d_a and d_b are unknown before the calibration. We set them zeros as the initial values of optimization. The point cloud P_{Lidar} is expressed as in Figure 3(a). This paper focus on the calibration between

LRF and binocular cameras. Therefore, the cameras are default calibrated by Matlab toolbox.



(a) 3D point cloud from LRF



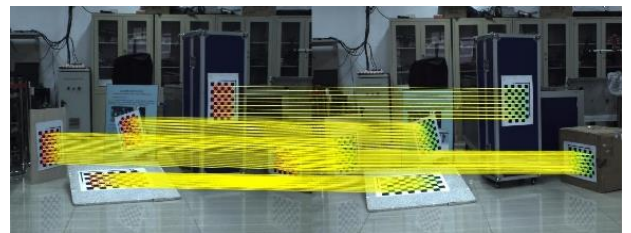
(b) The original photo of point cloud and position of platform
Fig. 3. Convert the LRF data from polar coordinates to Cartesian coordinate

3 Extract Planes

As we know, three non-parallel planes intersect at a point. A plurality of planes with different poses can produce many intersection points with permutations and combinations. In this paper, it is the key to obtain the feature points. Considering the fact that it is unreliable and imprecise to extract planes from stereo vision system directly, we mark the target planes by checkboards. Plane extraction in point cloud is widely used in many robotics applications [11][12]. In this proposed method, accuracy is the major consideration.

3.1 Extract planes by binocular cameras

The detection of checkboard is mature and precise, we use the functions in Matlab toolbox to detect planes that marked checkboards. However, the original functions only support the scenes with one checkboard. We ignore the checkboards have been detected by excluding the detected areas in image and repeat several times with the same scenes. After that, corners on all checkboards are detected completely and they are arranged in order. Matching the first corner between left and right images is more reliable than all of them at once. Iterative Closest Point (ICP) is used to improve the robustness of matching. The matching result is shown in Figure 4.



(a) left image (b) right image
Fig. 4. Extract checkboards in images and match the corners

Extracted planes from corner array is shown as

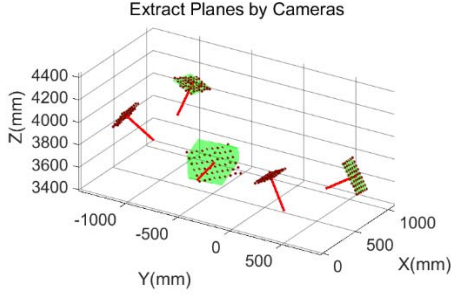
$$A_i^c P_i^c = \mathbf{1} \quad (8)$$

$$P_i^c = (A_i^{cT} A_i^c)^{-1} A_i^{cT} \quad (9)$$

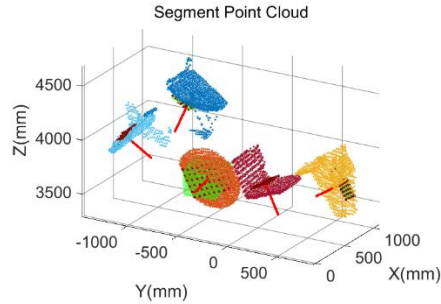
where A_i^c is the corners array on i-th checkboard in cameras coordinate system. P_i^c is the calculated plane with normal vector towards Z-axis negative direction. We normalized the normal vector as

$$V_i^c = [a_i \ b_i \ c_i \ d_i]^T \quad (10)$$

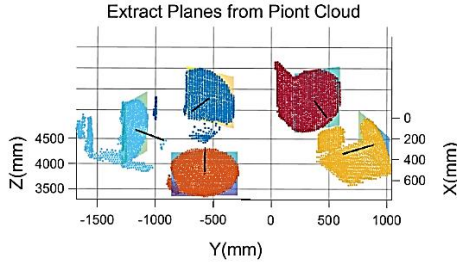
and $C_i^c = [e_i \ f_i \ g_i]^T$ is the center of plane. $C_i^v = [e_i' \ f_i' \ g_i']^T$ is the vertical intersection of C_i^c and P_i^c . The planes and normal vectors are expressed in Figure 5(a).



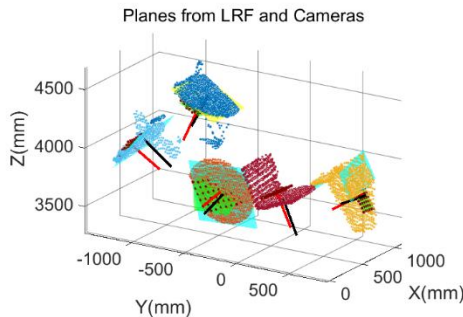
(a) Extract planes from 3D corners in binocular cameras system



(b) Segment point set near the planes



(c) Extract planes from point sets



(d) The difference of planes between LRF and cameras

Fig. 5. Extract planes from binocular cameras system and LRF

3.2 Extract planes in point cloud

Different from the usual plane extraction in point cloud, the planes in point cloud are near the corresponding ones in cameras system. It is more suitable to extract planes from

local areas. Therefore, We segment some points near the known planes in cylinder space and this condition is expressed as

$$B_i^{LC} = P_{Lidar} \{B_j^l \mid P d_i^j < d_{max}, \quad C d_i^j < r_{max}\} \quad (11)$$

$$P d_i^j = \|[B_j^{lT} \ 1] V_i^c\|_1 \quad (12)$$

$$C d_i^j = \|B_j^l - C_i^v\|_2 - P d_i^j \quad (13)$$

where B_i^{LC} is the point set to P_i^c . The parameters r_{max} and $2d_{max}$ are the radius and height of the cylinder. $B_j^l = [x_j^l \ y_j^l \ z_j^l]^T$ is the j-th point in P_{Lidar} . $P d_i^j$ is the distance from B_j^l to P_i^c and $C d_i^j$ is the distance from B_j^l to the line that via C_i^v with the normal vector is V_i^c . The segmented points are expressed in Figure 5(b).

The above operation segment point cloud roughly. The point sets apparently contain a number of planes and many noise points. We extract the plane match to P_i^c in three steps. Firstly, we apply the Hough transform to extract main planes. Secondly, compared the normal vector to identify the right plane. At last, the right planes are refined by least squares.

B_k^{LC} is the element of randomly selected points from B_i^{LC} . The points that do not meet the requirements should be eliminated before the Hough Transform and the eliminate rules and new point set S_i are shown as

$$S_i = \{B_{k-1}^{LC}, B_k^{LC}, B_{k+1}^{LC} \mid k = 2, \dots, N-1\} \quad (14)$$

Where $B_{k-1}^{LC} \neq B_k^{LC} \neq B_{k+1}^{LC}$. Then, we translate S_i into parameter space and the result is K_i . In most cases, K_i has some extreme points that do not conducive to find the density maximum points. Therefore, we eliminate them before next operation and the new data set is donated as K_i' . As shown in Figure 6, K_i' is a typical model to find the density maximum points by mean-shift. Since the density distribution vary widely and the traditional mean-shift may fall into the local optimal solution. In this proposed method, variable step length mean-shift is applied to obtain the real density maximum point in global space. This condition is expressed as

$$D_{n+1}^{DM} = D_n^{DM} + \frac{\sum_{k=1}^{N'} (D_k^{DM} - D_n^{DM})}{N'} \quad (15)$$

$$step_n = step_{max} - (step_{max} - step_{min}) \log_n n \quad (16)$$

Where D_n^{DM} and $step_n$ are the density maximum point and step length in n-th step. N' is the total number of K_i' . $step_{max}$ and $step_{min}$ are maximum and minimum values of step length respectively. And n' is the total number of iterations.

Then, a plane is extracted from B_i^{LC} . Some other planes are extracted in the same way after excluding the points near the detected plane. The right planes are shown in Figure 5(c).

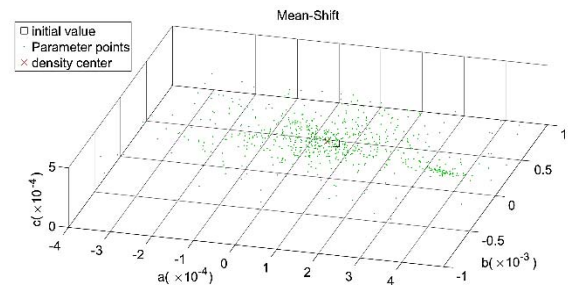


Fig. 6. Find density maximum point by the variable step length mean-shift in Hough Transform

The match plane is easily recognized by comparing the normal vector with the extracted planes from cameras system. Then, fewer points are segmented by a smaller cylinder near the extracted planes to refine planes \mathbf{P}_i^L . The result is shown in Figure 5(d).

3.3 Produce Virtual Points

As described in III, the intersection points of planes are important to this method. However, a scenes with a limited number of checkboards produces insufficient points to complete the calibration. Moreover, some extreme points should be excluded (e.g. intersection points by almost parallel planes). The intersections are calculated as

$$\mathbf{v}^C = \{f_1(\mathbf{P}_{i_1}^c, \mathbf{P}_{i_2}^c, \mathbf{P}_{i_3}^c) | (i_1, i_2, i_3) \in \text{combination 3 from } m\} \quad (17)$$

$$\mathbf{v}^L = \{f_2(\mathbf{P}_{i_1}^L, \mathbf{P}_{i_2}^L, \mathbf{P}_{i_3}^L) | (i_1, i_2, i_3) \in \text{combination 3 from } m\} \quad (18)$$

Where \mathbf{v}^C and \mathbf{v}^L are intersection points in binocular cameras system and LRF system. f_1 and f_2 mean to calculate the intersection points and abandon the extreme points. The two systems measure the same scene in the same unit, we can get some extra match points by equidistantly sampling on the intersection lines. \mathbf{V}^C and \mathbf{V}^L are the final match point sets. As shown in Figure 7, We use five checkboards to mark the planes and all of them are extracted both in LRF system and binocular cameras system. Total 48 virtual points are calculated to be the input of the optimization problem.

Given the match point sets \mathbf{V}^C and \mathbf{V}^L extracted from same scene of the stereo vision and LRF system, the aim is to find the relative pose and the additional parameters that minimizes the matching errors of between the two sets. It can be formulated as a maximum likelihood estimation (MLE) processes for above problem, whose solutions are obtained by solving the weighted nonlinear least squares problem expressed as

$$\hat{\theta}_p^r = \arctan(x\mathbf{V}_{ii}^L / z\mathbf{V}_{ii}^L) \quad (19)$$

$$\nabla z = d_a \sin \hat{\theta}_p^r + d_b \cos \hat{\theta}_p^r \quad (20)$$

$$\nabla x = -d_a \cos \hat{\theta}_p^r + d_b \sin \hat{\theta}_p^r \quad (21)$$

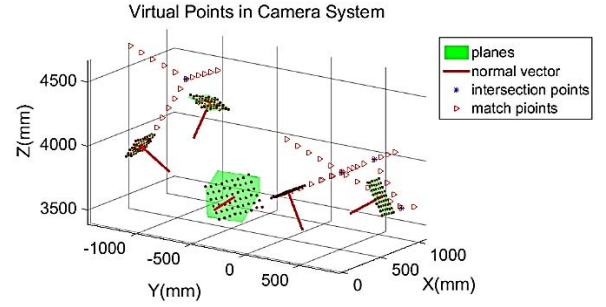
$$\widehat{\mathbf{V}}_{ii}^L = [(\mathbf{V}_{ii}^L + [\nabla x \ 0 \ \nabla z]^T) \mathbf{R}_{lc} + \mathbf{T}_{lc}] \mathbf{M} \quad (22)$$

$$\underset{\mathbf{R}_{lc}, \mathbf{T}_{lc}, d_a, d_b}{\text{argmin}} \sum_{ii=1}^{m'} \|\mathbf{V}_{ii}^C - \widehat{\mathbf{V}}_{ii}^L(\mathbf{R}_{lc}, \mathbf{T}_{lc}, d_a, d_b)\|^2 \quad (23)$$

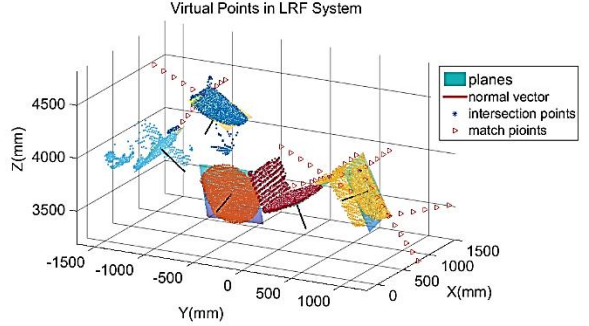
where $x\mathbf{V}_{ii}^L$ and $z\mathbf{V}_{ii}^L$ are the elements of vector \mathbf{V}_{ii}^L . ∇z and ∇x are change of $x\mathbf{V}_{ii}^L$ and $z\mathbf{V}_{ii}^L$. m' is total number of match points in \mathbf{V}^C or \mathbf{V}^L . Without loss of generality, we take the LRF pose as the reference coordinates and $\widehat{\mathbf{V}}_{ii}^L$ means the projection in binocular cameras coordinate system. \mathbf{M} is the intrinsic parameters matrix of left camera.

4 Experiment

The LRF on this platform is UTM-30LX-EW produced by HOKUYO, the accuracy is ± 30 millimeter at 10 meter and the resolution is 0.25° . The servo motor is Dynamixel RX28, the effective resolution is 0.29° . The cameras are MER-500-7UM/UC and the resolution is 2592×1944 with the lens focal length is 8 millimeter. The finally calibration result is shown in Table I.



(a) Calculate the virtual points in binocular cameras coordinate system



(b) Calculate the virtual points in LRF coordinate system
Fig. 7. Calculate the virtual points in binocular cameras and LRF coordinate system

Table 1: Calibration Result

Symbol	Calculated	Measured
\mathbf{R}_{lc}^x	3.64°	--
\mathbf{R}_{lc}^y	2.43°	--
\mathbf{R}_{lc}^z	-5.02°	--
\mathbf{T}_{lc}^x	90.52 mm	100 mm *
\mathbf{T}_{lc}^y	140.41 mm	120 mm *
\mathbf{T}_{lc}^z	9.25 mm	20 mm *
d_a	4.01 mm	20 mm *
d_b	3.29 mm	10 mm *

* means it is hard to get the true value and measured by hands. --means we can not obtain the measurements.

The calculated result is consistent to the measured. Two RGB-D images are shown in Figure 8 to verify the performance of the proposed approach. Scanning range of LRF is larger than the field of view and the points outside the image are set blue. The vast majority of points get the right color. As described in IV, most of the match points on the right side of scene. Therefore, the details of the result on the right side is better than the left side. Similar to the closed area in stereo vision system, the cameras are under the LRF, some points of a chair get color from ground.

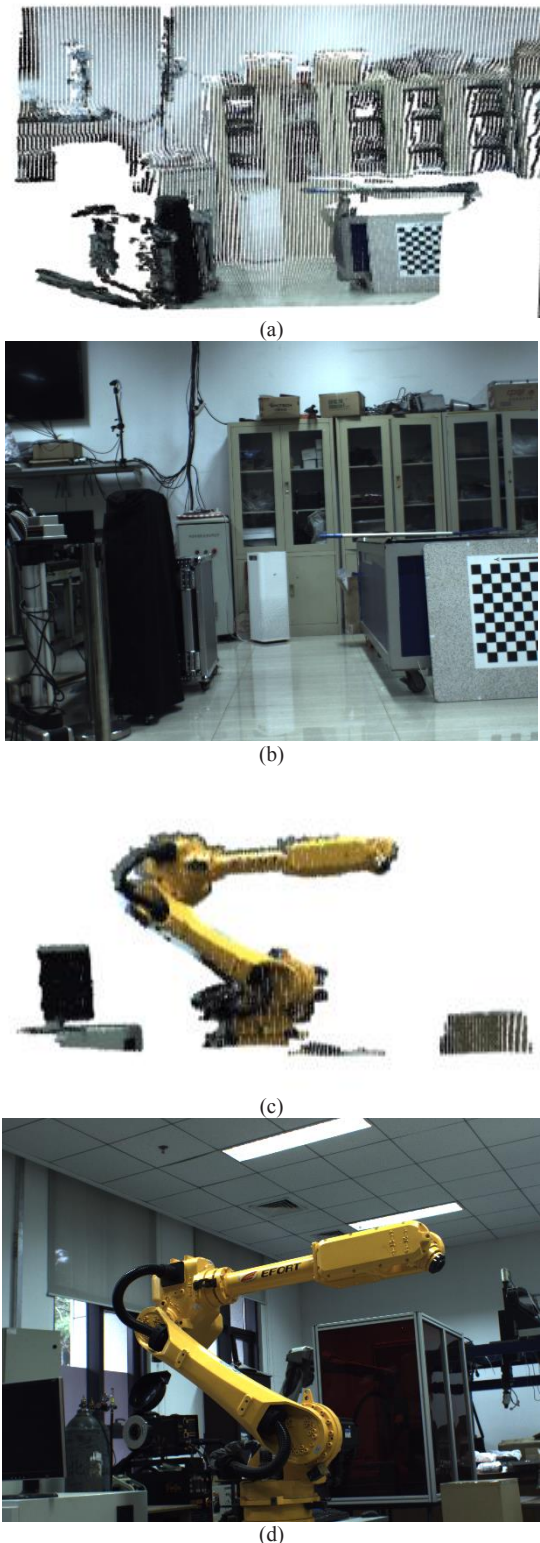


Fig. 8. (a)(c)Point cloud with color and (b)(d) the original images

5 Conclusion

In this paper, we proposed a novel method for extrinsic calibration of the LRF and binocular cameras. Compared with other methods, it only need to scan the scene once to complete the calibration and acquires good result. The method only need some simple checkboards to mark the target planes. The performance of calibration has a great influence by the distribution of match points.

References

- [1] C. Premebida and U. J. C. Nunes, "Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 371–384, 2013.
- [2] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte, "Classification and semantic mapping of urban environments," *The international journal of robotics research*, vol. 30, no. 1, pp. 5–32, 2011.
- [3] S. Wasielewski and O. Strauss, "Calibration of a multi-sensor system laser rangefinder/camera," in *Intelligent Vehicles' 95 Symposium*, Proceedings of the, pp. 472–477, IEEE, 1995.
- [4] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3283–3289, IEEE, 2011.
- [5] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 666–673, IEEE, 1999.
- [6] F. Vasconcelos, J. P. Barreto, and U. Nunes, "A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2097–2107, 2012.
- [7] Yan Zhuang, Fei Yan, Huosheng Hu, "Automatic Extrinsic Self-Calibration for Fusing Data From Monocular Vision and 3-D Laser Scanner," *IEEE Trans. Instrumentation and Measurement*, vol. 63, no. 7, pp. 1874–1876, July 2014.
- [8] Jung, Jiyoung, Jeong Yekeun, Jaesik Park and more authors, "A novel 2.5D pattern for extrinsic calibration of tof and camera fusion system," *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3290 – 3296, IEEE, 2011.
- [9] Jiyoung Jung, Joon-Young Lee, Yekeun Jeong and more authors, "Time-of-Flight Sensor Calibration for a Color and Depth Camera Pair," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1501–1513, July 2015.
- [10] Peyman Moghadam, Michael Bosse and Robert Zlot, "Line-based Extrinsic Calibration of Range and Image Sensors," *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3685–3691, IEEE, 2013.
- [11] Okada K, Kagami S, Inaba M and more authors, "Plane segment finder: algorithm, implementation and applications," *Robotics and Automation, (ICRA) 2001 IEEE International Conference on*, pp. 2120–2125, IEEE, 2001.
- [12] Chen Feng, Yuichi Taguchi and Vineet R. Kamat, "Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering," *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6218–6225, IEEE, 2014.