Available online at: https://ijact.in

Date of Submission	30/03/2021
Date of Acceptance	30/04/2021
Date of Publication	02/06/2021
Page numbers	3973-3977 (5 Pages)

ISSN:2320-0790

COMPUSOFT

An International Journal of Advanced Computer Technology

REDUCING ERROR IN LIDAR-CAMERA CALIBRATION

Jihun Park[1]

[1]Department of Computer Engineering, Hongik University, 94 Wowsanro, Mapo, Seoul, S. Korea

Abstract: In this paper, we propose a method to reduce errors in the method of finding the relationship between the LiDAR and the camera using trihedron. The core concept of the proposed technique is to use a trihedron with three intersecting planes that can be commonly recognized by the camera and LiDAR, eliminating the calculation of the coordinate system photographed using the camera. The coordinate system setting operation is performed only for the LiDAR sense data. By using the method of this paper, it is possible to reduce the calculation of the coordinate system using data fitting by camera, so that the error decreases. The result is presented through the experimental results.

Keywords: 3D reconstruction; LiDAR camera calibration; coordinate transformation

I. Introduction and Related Works

Recently, many applications have arisen for autonomous driving using LiDAR. The camera shows good performance for object detection and classification, but it is poor in weather and changes in illumination. LiDAR and radar are more resistant to changes in weather and illumination than cameras, but object classification shows lower performance than cameras. Therefore, it is important to complement the weaknesses of each sensor by fusing two or more sensors. The camera has a lot of information about the scene, but the LiDAR is mainly provided with geometric information about a limited point.

In order to fuse two or more sensors, the different coordinate systems of each sensor must be transformed to one coordinate system, and then combined using various algorithms. If the fusion is done properly, the combined information from multiple sensors is used properly.

In order to take advantage of the characteristics that complement multiple sensors in various applications, a method of sensor fusing multiple sensors has been studied [1]. Among them, LiDAR and camera fusion is one of the most frequently considered settings in outdoor situations [2], followed by calculations to calculate the relational coordinate system of the two sensors. Unnikrishnan et al. [3] first dealt with LiDAR and camera calibration, and proposed an interactive solution based on manually marking those points. To reduce manual work, Naroditsky et al. [4] proposed an automatic calibration approach using reflectance measurements in LiDAR scanners. Geiger et al. [5] proposed an approach that utilizes multiple chessboards. Several recent studies have proposed a common calibration process for online scenarios to correct calibration errors based on objects viewed simultaneously by LiDAR and camera sensors. Bileschi [6] detected and aligned the projected depth and contours in the image. Pandey et al. [7] corrected the parameter based on the mutual information between LiDAR reflectivity and camera intensity.

The three-dimensional restoration of a trihedron is possible with the same ratio [8,9,10] or the actual size[11]. To scan a trihedron using an image, you must have multiple views of the trihedron. In this study, a trihedron is photographed using a camera combined with a LiDAR with an auto focus function, and the coordinate system is calculated using a vector calculated by calculating the three plane intersec-tion points from the photographed image and a three-dimensional restoration tech-nique.

For polyhedral three-dimensional restoration, the method used in this study is clas-sified as a multi-view passive method. Previous studies, such as Hartley's book [8] and Moons paper [10], are useful references for three-dimensional restoration. Many methods [12] are based on feature points. Many algorithms require initialization for three-dimensional restoration. All three-dimensional restoration methods [8,9,10,11,13] can be used for the purpose of three-dimensional polyhedral restoration, but the method must be modified to restore the actual size. Using a non-moving object, we first need to calculate the camera parameters. Based on the calculated camera parameters, the feature points of 3D restoration are calculated.

There are two methods that are commonly used to perform a three-dimensional polyhedral restoration using feature points of an image. One method [8,9,10] calculates the F matrix by using the relationship between the feature points and decomposes the F matrix to find the position and direction of the projection camera. After that, the position and direction of the camera at the same ratio as the original three-dimensional space are found by solving a nonlinear equation. Another method [11] is to use known information such as a grid pattern, but we can modify this method to input the actual size. Regardless of which method is used, optimization should be done at the end of the calculation in order to calculate the correct camera internal parameters and position direction.

The camera calibration and trihedral 3D restoration method [13] used in this study is derived from Zhang's method [11]. Our method for trihedral 3D reconstruction optimizes the position and error of feature points on trihedral objects and environmental objects as well as camera parameters as variables. If you know the actual coordinate points using Zhang's transformed method, you can calculate the actual size of the reconstructed object. The trihedron used in this paper is similar to the ap-proach used by Gong et al. [14]. In this paper, the plane equation of 3 planes was calculated, and Gong et al. [14] calculated the edge of the trihedron.

II. Camera Calibration and Feature Point Calculation on a Trihedron

A. *Extended quaternion[15]*

The concept of extended quaternion was used to reduce the number of variables in using optimization techniques. Quaternions can only be used in rotation computation, and to compensate for this, an extended quaternion with a movement function was used. For better understanding, the equation in this paper is presented based on a matrix, but the actual implementation is an extended quaternion. The quaternion $q = w + pi + qj + rk$ is a 3D imaginary number used for rotation computation. w is scalar part of quaternion and is related to rotation angle while $pi + qj + rk$ is related to 3D rotation axis, while x, y and z represents translation values in extended quaternion.

B. *Notation used to transform the coordinate system[15]*

In order to represent the coordinate system transformation between two three-dimensional coordinate systems, this paper generally uses the Homogeneous transformation matrix. The notation ${}^{A}_{B}Tr$ is used to convert information expressed in the B coordinate system to the A coordinate system. All transform matrices expressed in this paper are actually coded using extended quaternions. The three-dimensional reference coordinate system is denoted by W, while the two-dimensional image coordinates are denoted by I.

C. *Algorithm*

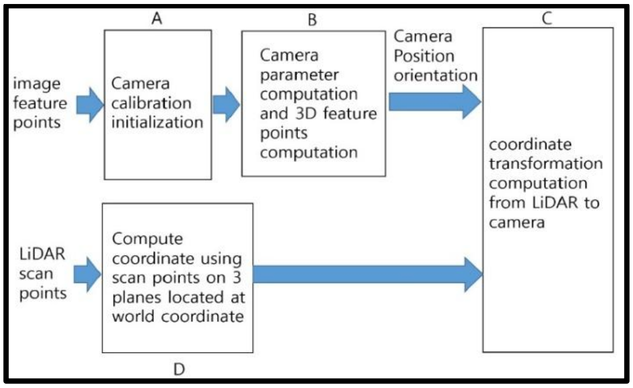

A
B
C
image feature points
Camera calibration initialization
Camera parameter computation and 3D feature points computation
Camera Position orientation
coordinate transformation computation from LiDAR to camera
LiDAR scan points
Compute coordinate using scan points on 3 planes located at world coordinate
D

Figure: 1 Overview of computation algorithm

Figure 1 shows the method for calculating the relationship between the LiDAR and the camera. Knowing the image feature points, it is possible to calculate the initial value of the 3D reconstruction using the method of obtaining Newton's roots in process A using images. Three-dimensional reconstruction is performed in process B using this value and camera feature points as inputs to process B. In addition, the trihedron's three plane equation and the intersection point and the trihedron's coordinate system are calculated in process D for the 3D points sensed by LiDAR. Since the coordinate system is located on world coordinate coinciding with three planes, we do not need to compute coordinates for images. Computed LiDAR sensed-coordinate in process D is the same as the world coordinate system. In C, the transformation relationship between the camera and the LiDAR can be found. In calculation process B and D, the camera and LiDAR can find the position of the feature point as a result of 3D reconstruction for the three planes. For these points, the equation of the three planes is found using RANSAC and plane data fitting.

D. *Analysis of scenes taken by camera and calculation of camera calibration*

The 3D reconstruction method using a camera in this study is a variation of Zhang's method [11]. The parameter optimization technique for camera calibration is similar to Zhang's method. This part can be changed with any 3D restoration algorithm [8,9,10,11].

Three-dimensional reconstruction is generally metric. In this study, actual distance information was added to the wall for actual restoration. If the object to be tracked has three-dimensional feature points, let's put the k-th feature point as M_k. These points are projected onto every scene image. Of course, the location of this point changes every scene, and its location is unknown right away. Assume that there are two shots of the scene and that the body part with these feature points is not moving. When a M_k point is projected onto an image, the projected point is assumed to be m_{ik}, and when projected onto an image, the projected point is assumed to be m_{jk}. Using an image and an image, assume that a ray gun was fired from the camera at that image pixel.

Since the rigid body did not move, the ray fired from each image must meet at the 3D feature point M_k. The ray emitted from the image i is called $\overrightarrow{ray(m_{ik})}$, and the ray vector launched from image j is called $\overrightarrow{ray(m_{jk})}$. When two rays intersect, the three-dimensional point is calculated as M_k. If the rays do not meet due to camera distortion, etc., the midpoint of the line making the shortest distance between the two rays is regarded as the intersection point $\widetilde{M_k}$.

Project the calculated intersection $\widetilde{M_k}$ back onto the image again. When projected onto an image i, it should be m_{ik}, and when projected onto an image j, it should be m_{jk}. By calculating the difference between the calculated $\widetilde{M_k}$ projected and the actual image feature points, the error is minimized by modifying variables such as camera internal and external parameters to reduce the gap. The scene value to be determined is found through an optimization process.

$$w_1 \sum_{i=1}^{n} \sum_{l=1}^{v} \left\| m_{il} - \widehat{m} \ (A, Ri, \overrightarrow{ti}, Ml) \right\|^2 + w_2 \sum_{k=1}^{p} \sum_{i=1}^{n-1} \sum_{j=\mathrm{i}+1}^{n} \left\| m_{ik} - \widehat{m} \ (A, Ri, \overrightarrow{ti}, raymik, Rj, tj, ray(mjk)) \right\|_2 \qquad (1)$$

Here, $\widehat{m}$ is camera projection function, A is the matrix representing the internal parameters of the camera, Ri is the rotation matrix of the camera that took the i-th image, $\overrightarrow{ti}$ is the position vector of the camera that took the i-th image. $\widehat{m} \ (A, Ri, \overrightarrow{ti}, Ml)$ is a pixel point as a result of projecting a three-dimensional point (Ml) on the i-th image using a camera with parameters of A, Ri, and $\overrightarrow{ti}$. From these calculation results, this study obtains the camera's internal and external parameters as well as 3D feature points. In calculation process C, the camera can find the position of the feature point as a result of 3D restoration for the three planes. For these points, the equation of the three planes is found using RANSAC and plane data fitting.

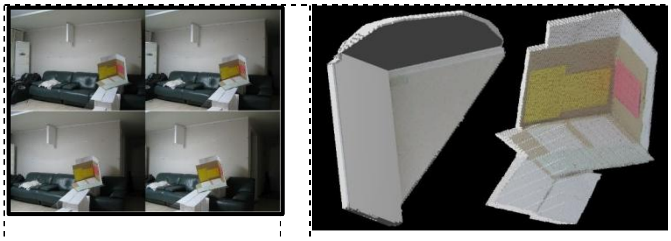
The calculation formula for the intersection point[15], $\vec{p}$, of the intersecting three planes is:

$$\vec{p} = (\, d_1(\overrightarrow{n_2} \times \overrightarrow{n_3}\,) + d_2(\overrightarrow{n_3} \times \overrightarrow{n_1}\,) + d_3(\overrightarrow{n_1} \times \overrightarrow{n_2}\,)\,)/\,\overrightarrow{n_1} \cdot (n_2 \times \overrightarrow{n_3}) \qquad (2)$$

where $\overrightarrow{n_1} \cdot \vec{x} = d_1$, $\overrightarrow{n_2} \cdot \vec{x} = d_2$, $\overrightarrow{n_3} \cdot \vec{x} = d_3$ are three intersecting planes of a trihedron. There is a straight line between the two intersecting planes, and the direction vector of this straight line can be known. Finally, the intersection of the three planes and three direction vectors can be obtained. The direction vector of a straight line that occurs when two planes meet each other is $\overrightarrow{n_1} \times \overrightarrow{n_2}$, $\overrightarrow{n_2} \times \overrightarrow{n_3}$, $\overrightarrow{n_3} \times \overrightarrow{n_1}$.

Here, a vector is placed in one of x y z and the coordinate system is determined using the remaining vectors. This process is performed twice for points found by camera feature point calculation and points found by LiDAR sensing. Let us call LiDAR coordinate as L, camera coordinate as C, and trihedron coordinate as B. When ${}^{L}_{B}Tr$ is a transformation matrix that observes a three-plane coordinate system from a LiDAR perspective, and ${}^{C}_{B}Tr$ is a transformation matrix that observes a three-plane coordinate system from a camera perspective, ${}^{C}_{L}Tr = {}^{C}_{B}Tr \ {}^{L}_{B}Tr^{\,-1}$.

E. RANSAC and LiDAR-Camera coordinate system calculation

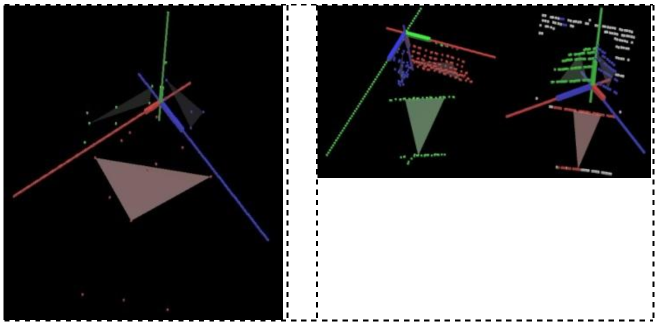


(a) Input images

(b) Two 3D reconstructed result of 3-plane objects using images alone

Figure: 2 input images and two 3D reconstructed results using camera only

There are two 3-plane objects. Left trihedron is set as a world coordinate eliminating the need of computing data fitting for camera feature points. Right trihedron computation is similar to the method presented in the paper [15]. Three points from RANSAC (Random sample consensus) are used as the initial value of the planar calculation using data fitting. When three planes are obtained by data fitting, the intersection of 3 planes and the intersection vector between 2 planes are calculated. This operation is performed once on 3D points of feature points calculated using a camera, and points sensed using LiDAR are also performed. Figure 2(a) is input images, and Figure 2(b) is resulting 3D reconstructed VRML file using image 3D reconstruction of the camera. Our 3D volume carving cannot handle a concave trihedron properly. Feature points were extracted from the input image, and 3D restoration was performed using several input images. As a result, it is possible to calculate the internal parameters of the camera, the camera position and direction, and the positions of each feature point. Among the feature points, only points that make up three planes corresponding to each plane were extracted using the RANSAC technique. After extraction, data fitting was performed on the plane to calculate the precise plane. Figure 3(a) and Figure 3(b) is the result of calculating the plane for the points measured by camera and LiDAR, respectively. Points constituting three planes were extracted using the RANSAC technique. After extraction, data fitting was performed on the plane to calculate the precise plane.



(a) RANSAC for camera feature points of right 3-plane object/trihedron.

(b) RANSAC for VLP32 LiDAR sensed points for both left and right 3-plane object/trihedron.

Figure: 3 Computing three planes using RANSAC

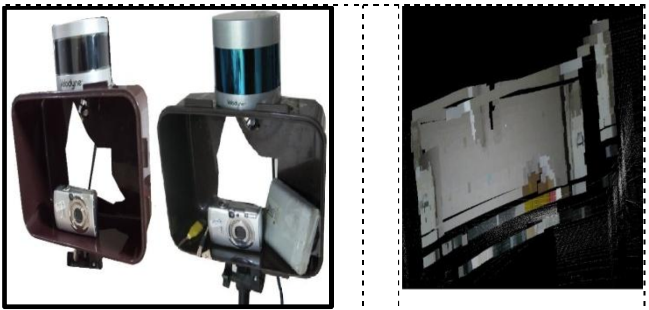
III. EXPERIMENTAL RESULTS

By setting the coordinate system for the trihedron object, the transformation matrix between the camera and the LiDAR was calculated. The distance information of the LiDAR is assumed to be perfect, and the position and direction of the camera used to capture the four input images can be found through 3D reconstruction calculation. The camera has lens distortion, so the distance information computed using camera images is not precise. The result of calculating the transformation matrix between the LiDAR and the camera using the four camera positions and directions, (a), (b), (c), (d), respectively calculated is in Table 1. Camera distortion has a small error at around the center of the lens, but the distortion increases as it goes out. Table 1 VLP16 (c) information has the smallest error. Figure 4(a) shows scenes where VLP16 LiDAR and IXY 810 camera/ VLP32 LiDAR and IXY 810 cameras are combined, and Figure 4(b) shows the result of three-dimensional restoration using the result of Table 1 VLP32 left trihedron (a) for the input image of Figure 2(a). The correct displacement values measured in terms of camera coordinate is (-0.7, 18.84, -1.78) in cm. According to the result, VLP32 using three planes as a world coordinate have smallest errors in general.

Table 1: Computed ${}^{C}_{L}\boldsymbol{Tr}$ is extended quaternion values using four input images

Images	***w***	***p***	***q***	***r***	***x (cm)***	***y (cm)***	***z (cm)***	***Displacement error (cm)***
VLP 16 (a) [15]	0.67613	-0.73477	-0.00965	-0.05357	-12.27	69.34	-12.42	52.89
VLP 16 (b) [15]	0.71350	-0.69692	-0.03675	-0.06218	-29.55	32.63	-17.89	35.81
VLP 16 (c) [15]	0.72590	-0.68749	-0.01402	-0.01498	-0.41	22.10	-4.13	4.03
VLP 16 (d) [15]	0.69266	-0.72059	-0.03054	0.006400	0.07	53.06	-3.07	34.25
VLP 32 left trihedron (a)	0.73519	-0.67709	-0.02521	-0.02014	0.74	14.73	-18.53	17.31
VLP 32 left trihedron (b)	0.73826	-0.67346	-0.03111	-0.02106	-2.28	14.27	-17.14	16.10
VLP 32 left trihedron (c)	0.73406	-0.67802	-0.02820	-0.02565	-3.11	19.84	-13.42	11.93
VLP 32 left trihedron (d)	0.73110	-0.68086	-0.01966	-0.03921	-7.20	22.61	-3.04	7.62
VLP 32 right trihedron (a)	0.73367	-0.67896	-0.02746	-0.00048	10.41	10.47	-21.50	24.13
VLP 32 right trihedron (b)	0.74267	-0.66661	-0.06230	0.01430	-5.38	1.89	-19.30	24.82
VLP 32 right	0.72820	-0.67	-0.09	-0.031	-41.	18.41	-19.	44.80

trihedron (c)		860	065	68	77		67	
VLP 32 right trihedron (d)	0.77882	-0.62447	-0.05442	-0.02256	-15.97	-25.09	-5.85	46.69



(a) Left, VLP16 and IXY 810 combined/right, VLP32 and IXY 810 combined, lidars have same dimension

(b) 3D reconstruction using Lidar scan points

Fig. 1. 3D reconstruction result

IV. Conclusion

In this paper, three-dimensional reconstruction is used to find the transformation matrix between the camera and the LiDAR sensor. Three-dimensional reconstruction is usually reconstructed in metric. In this study, actual distance information was added to the wall for actual reconstruction. LiDar has a limited resolution, so trihedron's edge calculation may be inaccurate. Since the number of points for computing a plane is much larger and elaborate calculations are not required to compute a plane, plane calculations using data fitting can be relatively accurate using multiple points of trihedron. Using higher resolution LiDAR may not affect calibration accuracy. But removing extra coordinate computation using data fitting, by setting a world coordinate on three intersecting planes, helps in increasing accuracy according to the experimental result.

V. Acknowledgement

This work was supported by 2021 Hongik University Research Fund.

VI. References

[1] Chen, X. Z., Kundu, K., Zhu, Y., Fidle, S., Urtasun, R. and Ma, H. 2018. 3D object proposals using stereo imagery for accurate object class detection, IEEE Trans. Pattern Anal. Machine Intelligence, vol. 40, no. 5, pp. 1259-1272.

[2] Chavez-Garcia , R. O. and Aycard, O. 2016. Multiple sensor fusion and classification for moving object detection and tracking, IEEE Trans. Intelligent Transportation System, vol. 17, no. 2, pp. 525-534.

[3] Unnikrishnan R. and Hebert, M. 2005. Fast extrinsic calibration of a laser rangefinder to a camera, Robotics Institute, Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-05-09.

[4] Naroditsky, O., Patterson, A. and Daniilidis, K. 2011. Automatic align-ment of a camera with a line scan LiDAR system, in Proc. IEEE Int. Conf. Robot. Automat., pp. 3429-3434.

[5] Geiger, A., Moosmann, F., Car, O. and Schuster, B. 2012.Automatic camera and range sensor calibration using a single shot, in Proc. IEEE Int. Conf. Robot. Automat., pp. 3936-3943.

[6] Bileschi, S. 2009. Fully automatic calibration of LiDAR and video streams from a vehicle, in Proc. IEEE 12th Int. Conf. Computer Vision Workshops (ICCV), pp. 1457-1464.

[7] Pandey, G., McBride, J. R., Savarese, S. and Eustice, R. M. 2015. Automatic extrinsic calibration of vision and LiDAR by maximizing mutual information, J. Field Robot., vol. 32, no. 5, pp. 696-722.

[8] Hartley, R. and Zisserman, 2004. A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.

[9] Pollefeys, M., Koch, R. and Gool, L. V. 1999. Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters, International Journal of Computer Vision, Vol. 32, No. 1, pp. 7-25.

[10] Moons, T., van Gool, L., and Vergauwen, M. 2008. *Foundation and Trends in Computer Graphics and Vision* **4**(4), 287–404.

[11] Zhang, Z. 2000. A flexible new technique for camera calibration, IEEE Trans. Pattern Analysis and Machine Intelligence **22**(11), 1330–1334 November.

[12] Lowe, D. G. 2004. Distinctive image features from scale-invariant key points, International Journal of Computer Vision **60**(2), 91–110 Nov.

[13] Park, J. and Park, S. 2013. Improvement on Zhang's camera calibration, *Applied Mechanics and Materials* **479-480**, 170–173.

[14] Gong, X., Lin, Y. and Liu, J. 2013. 3D LiDAR -Camera Extrinsic Calibration Using an Arbitrary Trihedron; Sensors 13(2):1902-18.

[15] Park, J. 2020. LiDAR Camera Calibration Using 3D Reconstruction of a Trihedron (in Korean), Journal of Next-generation Convergence Technology Association, Vol.4, No.6, pp. 623-629.