Targetless Extrinsic Calibration via Penetrating Lines for RGB-D Cameras with Limited Co-visibility

Jaeho Shin[1], Seungsang Yun[1], and Ayoung Kim[1]

[1]Seoul National University

Abstract. RGB-D cameras are crucial for robotic perception because they capture depth-added images. However, their limited field of view often necessitates multiple cameras to cover larger areas. In multi-camera setups, the goal is to maximize coverage with minimal overlap and as few cameras as possible, but this introduces challenges in extrinsic calibration. Existing calibration methods typically require specialized tools or rely on precise camera motion estimates. We propose a novel, line-based calibration method for RGB-D systems with low overlap, offering targetless, real-time performance by leveraging long-line features from the environment.

1 Introduction

Unlike RGB camera setups intended for shared FOV, systems deploying several RGB-D cameras are configured to minimize FOV overlap. This approach seeks to cover the surrounding area with the fewest possible cameras. Due to this design, it is difficult to use a calibration pattern that is concurrently visible inside the shared field of view when calibrating the system's extrinsic properties. We present a reliable but targetless extrinsic calibration methodology to overcome these obstacles, especially for low-overlap RGB-D camera systems. We concentrate on line features intersecting two cameras, which allows us to avoid the need for a large calibration target, which is often noticeable. As a result, even in situations where co-visibility is restricted, our algorithm operates dependably in real-time without the need for a pattern board, extra external devices, or inter-frame motion estimates.

2 Methodology

2.1 Merged Quadratic System

In order to take advantage of 3D lines in our approach, we first extract and match line features from two RGB images using a new deep matcher specifically made for line features [1]. This makes it possible to match lines between drastically dissimilar sceneries. The matching results are used to determine the related line's 3D coordinates using depth image analysis and RANSAC line fitting. The feature is classified as the *full 3D case* when the fraction of inliers in both images surpasses a predefined threshold. In this case, an equation system incorporates a constraint named "3D point on 3D line". Alternatively, the feature is categorized as *PnL case* if only the line associated with the source camera aligns accurately, in which case we apply a constraint as described in [2].

Full 3D Case A constraint that states that converted 3D points on the source line must lie on the 3D target line is used in the first case. The dual Plücker matrix property serves as the foundation for this representation. For a given 3D point $\mathbf{X}$, the following equation is satisfied:

$$\begin{aligned} \mathbf{L}_t^* \bar{\mathbf{X}}_{tj} &= \begin{pmatrix} -\mathbf{d}_t^\wedge & \mathbf{m}_t \\ -\mathbf{m}_t^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}\mathbf{X}_{sj} + \mathbf{t} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} -\mathbf{d}_t^\wedge \mathbf{R}\mathbf{X}_{sj} - \mathbf{d}_t^\wedge \mathbf{t} + \mathbf{m}_t \\ -\mathbf{m}_t^T \mathbf{R}\mathbf{X}_{sj} - \mathbf{m}_t^T \mathbf{t} \end{pmatrix} \\ &= \begin{pmatrix} -\mathbf{d}_t^\wedge \mathbf{R}\mathbf{X}_{sj} + \mathbf{m}_t \\ -\mathbf{m}_t^T \mathbf{R}\mathbf{X}_{sj} \end{pmatrix} + \begin{pmatrix} -\mathbf{d}_t^\wedge \mathbf{t} \\ -\mathbf{m}_t^T \mathbf{t} \end{pmatrix} \\ &= \mathbf{0}_{4\times 1}, \quad (j = 1, 2) \end{aligned} \tag{1}$$

where j represents each endpoint, respectively. Thus, two constraints are derived from a single line pair $\mathbf{L}_t \leftrightarrow \mathbf{L}_s$.

PnL Case We use the restriction, which is also used in [2], for the second instance. Given that the transformed 3D point $\hat{\mathbf{x}}_t$ is projected onto the 2D line $\mathbf{l}_t$ of the target image, $\mathbf{l}_t^T \hat{\mathbf{x}}_{tj} = 0$ is valid. Consequently, from a singular correspondence $\mathbf{l}_t \leftrightarrow \mathbf{L}_s$, we derive two constraints:

$$(\mathbf{K}_t^T \mathbf{l}_t)^T (\mathbf{R}\mathbf{X}_{sj} + \mathbf{t}) = 0. \quad (j = 1, 2) \tag{2}$$

System Solver and Solution Refinement In order to combine (1) and (2) into a single, cohesive system, we use the CGR parameterization [3] to represent the rotation matrix $\mathbf{R}$ as follows:

$$\mathbf{R} = \frac{\bar{\mathbf{R}}}{1 + \mathbf{s}^T \mathbf{s}}, \ \bar{\mathbf{R}} = (1 - \mathbf{s}^T \mathbf{s})\mathbf{I}_3 + 2[\mathbf{s}]_\times + 2\mathbf{s}^T \mathbf{s}, \tag{3}$$

where $\mathbf{s} = [s_1, s_2, s_3]$. We obtain the following readily combined equations by multiplying $(1 + \mathbf{s}^T \mathbf{s})$ on both sides of (1) and (2):

$$\begin{gathered} \begin{pmatrix} -\mathbf{d}_t^\wedge \bar{\mathbf{R}}\mathbf{X}_{sj} + (1 + \mathbf{s}^T \mathbf{s})\mathbf{m}_t \\ -\mathbf{m}_t^T \bar{\mathbf{R}}\mathbf{X}_{sj} \end{pmatrix} + \begin{pmatrix} -\mathbf{d}_t^\wedge \boldsymbol{\tau} \\ -\mathbf{m}_t^T \boldsymbol{\tau} \end{pmatrix} = \mathbf{0}_{4\times 1}, \\ (\mathbf{K}_t^T \mathbf{l}_t)^T \bar{\mathbf{R}}\mathbf{X}_{sj} + (\mathbf{K}_t^T \mathbf{l}_t)\boldsymbol{\tau} = 0, \quad (j = 1, 2), \end{gathered} \tag{4}$$

where $\boldsymbol{\tau} = (1 + \mathbf{s}^T\mathbf{s})\mathbf{t}$. We stack the M and N measurements that are included in each example vertically to create the quadratic system that follows:

$$\mathbf{Ar} + \mathbf{B}\boldsymbol{\tau} = \mathbf{0}_{(8M+2N)\times 1}, \tag{5}$$

where $\mathbf{r} = [s_1^2, s_2^2, s_3^2, s_1s_2, s_1s_3, s_2s_3, s_1, s_2, s_3, 1]^T$. The rotation matrix and translation vector are then obtained by solving (5) using the method described in [2]. Based on the CGR parameter, the system is processed using the RE3Q3 solver [4]. The system's initial value is refined by optimizing a cost function that combines the line reprojection error and the 3D point-to-line error.

3 Experiment

Verification of Pose Variance Estimation Two RealSense D435i cameras are fixed to an aluminum framework in our experiment device. The source camera can be moved along the x-axis to determine translational changes with a ruler in place, and the target camera has a protractor to evaluate rotational movements. A precise algorithm should be able to identify and quantify the difference within a reasonable error margin, even though manual tweaks to the angle and spacing of the cameras may not produce the exact anticipated pose disparity.

We assessed poses in 30 distinct circumstances, starting from a base setup where both cameras faced forward and were separated by 20 cm. In these experiments, the angles ranged from 0°to 80°, rising in 20°increments, and the lengths ranged from 20 cm to 45 cm, increasing by 5 cm at each step. The precision of a 5 cm translational movement along the x-axis at a constant angle and a 20°angular adjustment at a predetermined distance were then examined. Keeping the alternative parameter constant at Xcm or $\theta°$, the error for the ith rotation or translation is calculated as follows:

$$\begin{aligned} \theta_{e_i}^{\mathrm{X}cm} &= \left\| (\boldsymbol{\theta}_{i+1}^{\mathrm{X}cm} - \boldsymbol{\theta}_{i}^{\mathrm{X}cm}) - (0, 20, 0)^T \right\|_2, \\ \mathrm{X}_{e_i}^{\theta^\circ} &= \left| \left\| \mathbf{X}_{i+1}^{\theta^\circ} - \mathbf{X}_{i}^{\theta^\circ} \right\|_2 - 5 \right|. \end{aligned} \tag{6}$$

Fixed Distance	Rotation Variation (°) 0→20	20→40	40→60	60→80	Fixed Angle	Distance Variation (cm) 20→25	25→30	30→35	35→40	40→45
45 cm	0.6398	0.0707	0.9675	1.2077	80°	0.1697	0.1137	1.3918	0.6780	1.3741
40 cm	0.7586	0.1131	0.3187	0.2768	60°	0.1584	1.0096	1.1799	1.3456	0.4132
35 cm	0.1203	0.4511	0.8060	0.1947	40°	0.3156	0.1831	0.7042	0.6292	0.5332
30 cm	0.3974	0.2406	0.2435	0.3669	20°	0.0120	0.0093	0.3852	0.1017	0.1975
25 cm	0.1790	0.1944	0.4898	0.1901	0°	0.2690	0.2260	0.1161	0.1105	0.0595
20 cm	0.4953	0.4800	0.4810	1.1286						

Table 1. Error Measured for Rotation (°) and Translation (cm) Variation

Results When the cameras are positioned at short distances and small angles, with a substantial field of view overlap, the results closely match the real changes in both parameters. Notably, we noted an angular error of 0.7586°and a translational error of up to 0.3156 cm in the first column with the least variance for each case. Nevertheless, the errors increased because there were fewer line features as the angle grew beyond 40°, and the FOV ceased to overlap. This is because the accuracy of the depth sensor data decreases when the angle between the line and the camera's central axis increases. However, in these difficult circumstances, where conventional calibration methods necessitating specific instruments break down, ours demonstrated variances as high as 1.2077°and 1.3918 cm.

Acknowledgements. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00241758).

References

1. R. Pautrat, I. Su´arez, Y. Yu, M. Pollefeys, and V. Larsson: Gluestick: Robust image matching by sticking points and lines together. arXiv preprint arXiv:2304.02008 (2023)
2. L. Zhou, D. Koppel, and M. Kaess: A complete, accurate and efficient solution for the perspective-n-line problem. IEEE Robot. and Automat. Lett., vol. 6, no. 2, 699–706 (2020)
3. F. M. Mirzaei and S. I. Roumeliotis: Globally optimal pose estimation from line correspondences. Proc. IEEE Intl. Conf. on Robot. and Automat. IEEE, 5581–5588 (2011)
4. L. Zhou, J. Ye, and M. Kaess: A stable algebraic camera pose estimation for minimal configurations of 2d/3d point and line correspondences. Asian Conference on Computer Vision (ACCV), 273–288 (2019)