Camera calibration using two vanishing points

Nina Cholpankulova
Department of Computer Science
Friedrich-Alexander-University Erlangen-Nuremberg
Erlangen, Germany
nina.cholpankulova@fau.de

***Abstract*—** this work reproduces the approach[1], that utilizes two vanishing points produced by the planar calibration object. The goal is to determine the extrinsic parameters for six cameras, placed around the perimeter of a room. I concluded, that the results are not reliable for the current environment setup.

Keywords—Calibration, extrinsic parameters

I. INTRODUCTION

Camera calibration, referred to as extrinsic and intrinsic camera parameters estimation, is a fundamental step in attaining precision in measurements within computer vision applications[2].

Camera calibration, referred to as extrinsic and intrinsic camera parameters estimation, is a fundamental step in attaining precision in measurements within computer vision applications[2]. Establishing the relationship between the world and camera coordinate systems is crucial for accurate calibration. The world coordinate system represents the three-dimensional space in which objects and scenes exist, while the camera coordinate system is specific to the imaging device in which the points are defined relative to the center of the camera[2] The calibration process aims to establish a transformation between these coordinate systems, allowing us to relate real-world measurements to their corresponding image coordinates. In this work, I simultaneously calibrated six cameras placed in a confined room utilizing a 2D calibration object. In this project I derive the extrinsic parameters and locations of each camera, reproducing the existing methodology [1].

II. METHODOLOGY

I follow the methodology presented by Guillou et al. and Orghidan et al., which involves placing the calibration pattern on the room's floor, capturing images, and extracting two vanishing points of parallel lines of the pattern. The utilized approach from the paper[1] proposes formulas to calculate the rotation and translation of the cameras using the coordinates of the two vanishing points.

2.1 Focal length calculation

The focal length estimation uses a classical pinhole model. The camera center of projection O is projected onto the image plane as P, illustrated in Fig. 1[1]. An assumption made in the papers – P is located in the center of the image. $abcd$ is a perspectival projection onto the image plane of the rectangular ABCD, that contains two sets of parallel lines. Vanishing point F_u is defined as the intersection of lines ad and bc, F_v – is the intersection of lines ab and dc. Coordinates of vanishing points in the camera coordinate system are considered known. P_{UV} is the orthogonal projection of P on the line $(F_u\ F_v)$, see Fig. 2. Focal length is computed following the formulas [1, Eq. (5)]:

$$OP_{uv} = \sqrt{F_v P_{uv} * P_{uv} F_u} \tag{1}$$

$$f = OP = \sqrt{{OP_{uv}}^2 - {PP_{uv}}^2} \tag{2}$$

2.2 Rotation matrix calculation

The rotation matrix R determines the rotation between the world and camera coordinate system, which are called R_O and R_C respectively. The world reference system is centered at the corner A of the rectangle $ABCD$. Vanishing points are in the direction of the orthogonal axes of the system. Taking into account that all parallel lines meet at a vanishing point, a new coordinate system $R_{O'}$, centered at O, is constructed. The $R_{O'}$, is determined by these vectors [3, Eq. (5)]:

$$\begin{gathered} U' = OF_u \\ V' = OF_v \\ W' = U' \times V' \end{gathered} \tag{3}$$

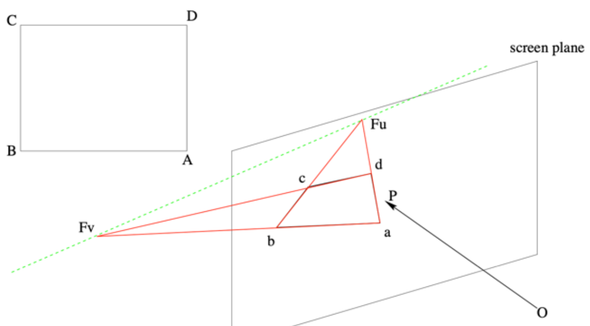


Fig. 1. Illustration of the projection.

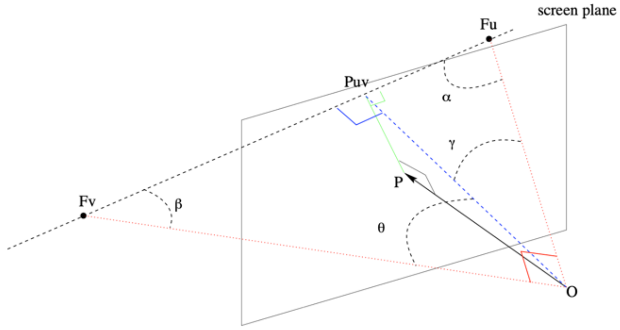


Fig. 2. Illustration of focal length calculation.

The new coordinate system is similar to the world reference system, hence the rotation between R_C and R_O is equal to the rotation between R_C and R'_O [1, Eq. (7)]:

$$u'_{/R_c} = \frac{\overrightarrow{OF_u}_{/R_c}}{\left\|\overrightarrow{OF_u}_{/R_c}\right\|} = \left(\frac{F_{ui}}{S_1}, \frac{F_{ui}}{S_1}, \frac{F_{ui}}{S_1}\right)^t$$

$$v'_{/R_c} = \frac{\overrightarrow{OF_v}_{/R_c}}{\left\|\overrightarrow{OF_v}_{/R_c}\right\|} = \left(\frac{F_{vi}}{S_2}, \frac{F_{vi}}{S_2}, \frac{F_{vi}}{S_2}\right)^t \quad (4)$$

$$w'_{/R_c} = u'_{/R_c} \times v'_{/R_c}$$

with $w'_{/R_c} = (w'_i, w'_j, w'_k)$, $S_1 = \left\|\overrightarrow{OF_u}_{/R_c}\right\|$, $S_2 = \left\|\overrightarrow{OF_v}_{/R_c}\right\|$ and f is the focal length.

The rotation matrix is determined following [1, Eq. (6)] as :

$$M_{o\to c} \times u_{/R_O} = u'_{/R_C}$$

$$M_{o\to c} \times v_{/R_O} = v'_{/R_C} \quad (5)$$

$$M_{o\to c} \times w_{/R_O} = w'_{/R_C}$$

Where $u_{/R_O}$, $v_{/R_O}$ and $w_{/R_O}$ are unit vectors, $M_{o\to c}$ – rotation matrix. Combining (4) and (5), authors deduce [1, Eq. (9)]:

$$M_{o\to c} = \begin{pmatrix} \frac{F_{ui}}{\sqrt{F_{ui}^2+F_{uj}^2+f^2}} & \frac{F_{vi}}{\sqrt{F_{vi}^2+F_{vj}^2+f^2}} & w'_i \\ \frac{F_{uj}}{\sqrt{F_{ui}^2+F_{uj}^2+f^2}} & \frac{F_{vj}}{\sqrt{F_{vi}^2+F_{vj}^2+f^2}} & w'_j \\ \frac{f}{\sqrt{F_{ui}^2+F_{uj}^2+f^2}} & \frac{f}{\sqrt{F_{vi}^2+F_{vj}^2+f^2}} & w'_k \end{pmatrix} (6)$$

2.3 Translation vector calculation

a is a perspective projection of A, ak is a vector parallel to the u axis of the world coordinate system, it lies on the ad line of the rectangular $abcd$. The length of the ak is l and is considered known.

Then further formulas are true:

$$\overrightarrow{AP}_{/R_O} = l \cdot u;\ \overrightarrow{AP}_{/R_C} = M_{o\to c} \cdot \overrightarrow{AP}_{/R_O} K''$$

K'' is the intersection of the AD line, that passes through a and OK. Looking at the similar triangular OAK'' and OAK, authors get [1, Eq. (11-14)]:

$$\frac{\left\|\overrightarrow{A'K''}_{/R_C}\right\|}{\left\|\overrightarrow{AK}_{/R_C}\right\|} = \frac{\left\|\overrightarrow{OA'}_{/R_C}\right\|}{\left\|\overrightarrow{OA}_{/R_C}\right\|} \quad (7)$$

or

$$\left\|\overrightarrow{OA}_{/R_C}\right\| = \cdot \frac{\left\|\overrightarrow{OA'}_{/R_C}\right\| \cdot \left\|\overrightarrow{AK}_{/R_C}\right\|}{\left\|\overrightarrow{A'K''}_{/R_C}\right\|} \quad (8)$$

$$\overrightarrow{OA}_{/R_C} = \left\|\overrightarrow{OA}_{/R_C}\right\| \cdot \frac{\overrightarrow{OA'}_{/R_C}}{\left\|\overrightarrow{OA'}_{/R_C}\right\|} \quad (9)$$

The translation vector is:

$$T_{o\to c/R_O} = M_{o\to c} \cdot \overrightarrow{OA}_{/R_C} \quad (10)$$

III. IMPLEMENTATION

33.1 Experiment setup

As a calibration object, I use eight by five checkerboard pattern, that is printed on the 841×1189 mm size paper with 125×125 mm size of each checker. The pattern is positioned on the floor in three distinct locations. That is done to guarantee optimal visibility and high-quality images captured by all cameras. A point on the floor is selected as the start of the global coordinate system. Locations of the patterns are known relative to that point. Six cameras, that are located around the perimeter of a room, capture the pattern. Twenty images are extracted from each camera's recordings.

Corners of the pattern are detected for the obtained images, as well as two vanishing points using the two sets of parallel lines of the planar pattern, Fig. 3 illustrates that. To follow the methodology presented in the paper [1], those points had to be set in a certain way. Point A is always set to the closest corner of the pattern on the image. It is the start of the local coordinate system for the camera. Points B, C, D are assigned in a clockwise manner to the rest of the corners. F_v – vanishing point detected from parallel lines AB and DC, F_u - vanishing point detected from parallel lines AD and BC.

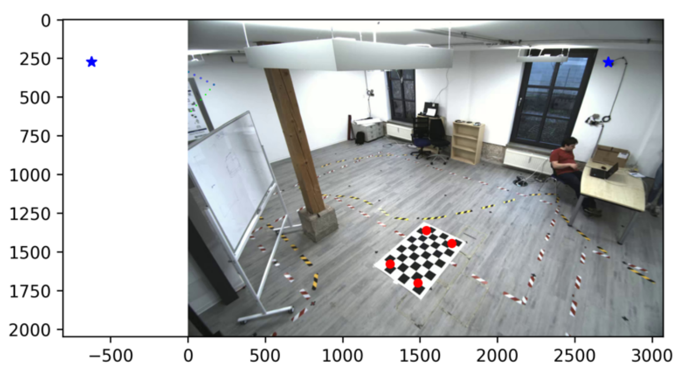


Fig. 3. Detection of the pattern corners and vanishing points.

Extrinsic parameters in relation to the local coordinate system are derived from the dataset comprising 20 images. That is done independently for each camera using formulas (6) and (10). Average parameters over 20 frames are computed to filter noise.

3.2 Translation vector

Computing translation vector, I had to implement some modifications.

a) Length l of ak is defined as :

- $\frac{2}{4} \cdot ad$ for the first, fourth and sixth cameras.
- $\frac{4}{7} \cdot ad$ for the second, third and fifth cameras.

The difference occurs depending, on which side of the rectangular is taken as ad. It could be either the long or short side of the rectangular defined by eight or five pattern corners respectively.

b) I don't transfer the mentioned AD line to pass through the a point of the image plane. Instead, to find K'' I project vector ak onto the OF_u vector in the camera coordinate system.

3.3 Local to global transformation

To obtain the global camera rotations and camera positions I transition from a local coordinate system to a global one. It is done by multiplying the local camera position by two transformation matrices, rotation and translation.

I set the rotation matrix for each camera independently, conditional to which corner of the pattern is used as a start of the local coordinate system. A translation matrix is a matrix that is determined based on the distance between the start of the local coordinate system and the global one.

IV. RESULTS

I retrieved Euler angles from the extrinsic matrices listed in Table I, while Table II provides information regarding the camera coordinate systems. The Euler angles, camera coordinates and respective standard deviations, presented in tables, are rounded to two decimal places for clarity in presentation.

Validation data is acquired using the five-point calibration method described in Batra et al.[4]. The method was reproduced in the same environment, the camera setup remained the same for the validation data collection [5-6]. A comparison of the results is presented in Table III.

TABLE I.

Camera	***Average Euler angles, degree***	***Standard deviations of Euler angles***
cam_1	[-135.65, 0.0, 45.37]	[0.05, 0.0, 0.09]
cam_2	[-121.45, 0.0, -20.31]	[0.1, 0.0, 0.16]
cam_3	[-131.98, 0.0, -53.61]	[0.02, 0.0, 0.02]
cam_4	[-122.32, 0.0, -159.67]	[0.01, 0.0, 0.02]
cam_5	[-133.77, 0.0, 136.91]	[0.01, 0.0, 0.02]
cam_6	[-132.81, 0.0, 44.09]	[0.02, 0.0, 0.09]

TABLE II.

Camera	***Average camera coordinates, mm***	***Standard deviations of camera coordinates***
cam_1	[4548.5, -619.31, 2850.92]	[6.95, 3.02, 4.06]
cam_2	[-1465.88, -913.49, 2630.48]	[4.31, 17.82, 5.11]
cam_3	[-754.38, 4573.22, 2804.0]	[1.38, 0.88, 0.45]
cam_4	[-1452.98, 12967.16, 2699.78]	[2.8, 7.83, 3.4]
cam_5	[4370.54, 12469.75, 2634.88]	[0.94, 1.8, 1.48]
cam_6	[5299.96, 4530.53, 2599.72]	[9.69, 2.08, 3.21]

Visual comparison of the 2D camera positions, detected using two methods, is presented in Fig.4. We overlayed a virtual calibration pattern on the real images, to qualitatively assess the estimated Euler angles. That is done in the Blender application. The results are presented in the Appendix.

TABLE III.

Camera	View angle and position errors with respect to the reference data	
	Euler angles, degree	*Camera position coordinates, mm*
cam_1	[5.8, 2.35, 0.11]	[160.81, 226.42, 11.03]
cam_2	[2.07, 0.59, 9.15]	[1793.74, 1157.58, 1023.12]
cam_3	[14.98, 2.8, 6.54]	[1477.16, 672.07, 113.99]
cam_4	[1.29, 2.21, 3.33]	[114.2, 964.7, 78.96]
cam_5	[11.96, 0.36, 11.92]	[1098.05, 1017.81, 198.72]
cam_6	[4.27, 0.36, 1.7]	[441.56, 121.73, 502.84]

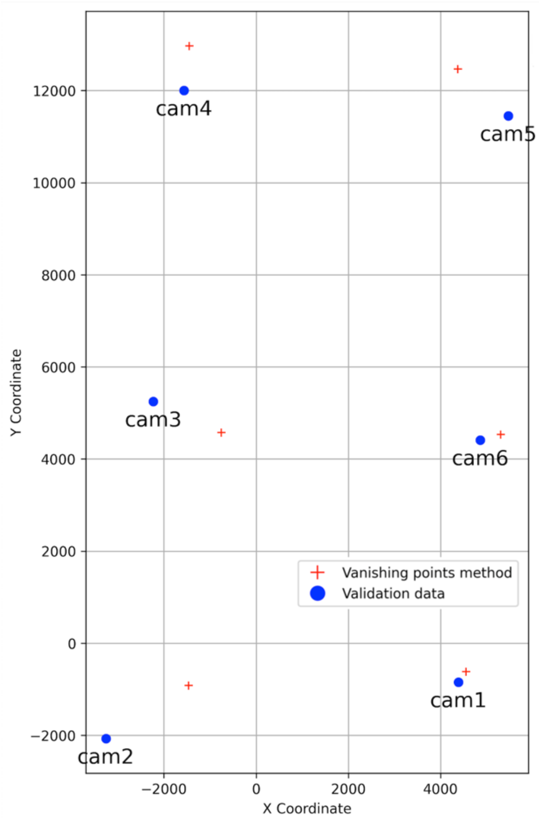


Fig. 4. 2D representation of the camera positions.

TABLE IV.

Camera	*View direction error with respect to the reference data, degree*	*Magnitude, mm*
cam_1	32.82	277.72
cam_2	17.06	2134.83
cam_3	44.44	1622.86
cam_4	60.02	971.43
cam_5	36.18	1497.21
cam_6	36.69	458.03

Table IV provides us with some extra data comparison. The distances between camera locations are computed. Additionally, I retrieve camera view direction vectors from validation data rotation and rotation, obtained using the vanishing point method. The angle between these two vectors is computed for each camera.

V. Conclusion

Based on the comparison of retrieved extrinsic parameters of the cameras using the vanishing points method and the more established five-point method, I can conclude about the quality of the received data.

The detected Euler angles seem to be retrieved with good precision. The average discrepancy from validation angles in all of the directions is approximately four degrees, which is a tolerable result. The Euler rotations around X and Z axis show the largest difference for the third and fifth cameras. The reason for this result is unknown.

The results of the camera coordinates of the cameras don’t have satisfying quality. The location varies up to 1,5 meters, which leaves us with very poor precision. The reproduced method in our environment is not valid to make reliable estimations. I can assume the errors come up due to assumptions made in the original method I am reproducing[1], such as neglecting camera distortion and choosing the principal point to be located in the image center.

Improvement in the results can be expected by utilizing a 3D calibration object instead of the planar checkerboard. In the algorithm I was following[1], the third basic vector of the world coordinate system is deduced as a vector product of the first two, whereas this action would be eliminated with the third vanishing point.

References

[1] E. Guillou, D. Meneveaux, E. Maisel, K. Bouatouch, “Using vanishing points for camera calibration and coarse 3D reconstruction from a single image,” The Visual Computer, 16(7), 2000, pp. 396–410.

[2] R. Szeliski, Computer Vision: Algorithms and Applications. Cham: Springer, 2023.

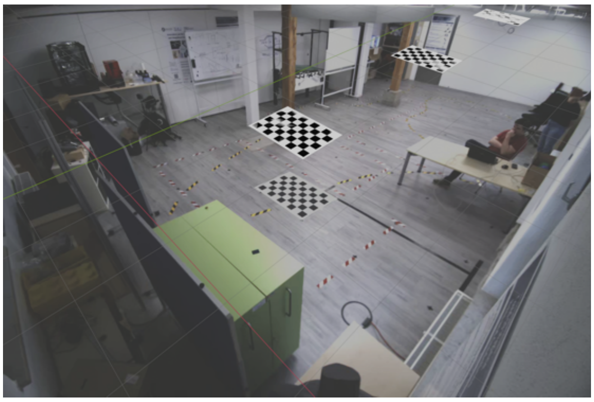
[3] R. Orghidan, J. Salvi, M. Gordan and B. Orza, “Camera calibration using two or three vanishing points,” 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, Poland, 2012, pp. 123-130.

[4] D. Batra, B. Nabbe, and M. Hebert, “An alternative formulation for five point relative pose problem,” 2007 IEEE Workshop on Motion and Video Computing (WMVC’07), 2007

[5] K. Madsen, H. B. Nielsen, and O. Tingleff, “Methods for Non-Linear Least Squares Problems (2nd ed), Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[6] Ethan Eade. Gauss-newton / levenberg-marquardt optimization, 2013.Implemented in Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

APPENDIX


cam_1


cam_2


cam_3


cam_4


cam_5


cam_6