A NOVEL GEOMETRIC APPROACH FOR CAMERA CALIBRATION

Zhe Zhang *Kin Hong Wong*

Department of Computer Science & Engineering
The Chinese University of Hong Kong

ABSTRACT

In this paper, we propose a novel geometric approach for solving camera calibration efficiently. Our camera calibration procedure is implemented on planar objects with rectangles on them. Images of the planar object are taken from different views where corner features are extracted. The geometric relationship between the rectangle corners is utilized to reduce the number of extrinsic unknown parameters. This process can further reduce the complexity of the optimization space greatly and thus enable our approach to be more accurate and more efficient than other geometric approaches. Experimental results on synthetic images and real images witness the advantage of our approach over state-of-the-art approaches.

Index Terms— Camera calibration, geometric approach

1. INTRODUCTION

Due to the crucial significance of camera calibration, it has been thoroughly studied and lots of approaches have been proposed during last decades. A comprehensive classification and detailed description have been conducted in [1]. Generally, camera calibrating methods can be divided into two groups, algebraic methods and geometric methods in terms of whether the energy function to be minimized is based on algebraic distances or geometric distances [2]. The main difference is that geometric methods have a reasonable geometric interpretation while algebraic ones are not statistically or geometrically meaningful.

Representative algebraic methods are DLT [3], normalized DLT [4]. Other well known approaches are the model using vanishing points [5] and the model proposed by Zhang [6]. The advantage of algebraic methods is that they have analytical solutions while the disadvantage of them is that they cannot solve lens distortions. Thus algebraic approaches are always used as initialization of geometric methods which always rely on non-linear iteration methods to achieve their optimal goals. Such two-step method was first introduced in [7], where Maximum Likelihood Estimation was applied to minimize the geometric error between corners of planar quadrangles and their corresponding image points. Other approaches such as [8] based on properties of circles can also perform well.

The common point about nearly all the geometric methods is that they all need to compute the rotational angles during projection which involves the computation of differentiation on trigonometric functions. As we all know, this computation is so expensive that it limits the application of such methods in resource-limited systems. Another limitation of these models is that they introduce a large number of extrinsic parameters and make the optimization space extremely complex.

In this paper, we propose a geometric approach which doesn't rely on the differentiation computation of trigonometric functions. Furthermore, we only have 3 extrinsic parameters in one image but not 6 extrinsic parameters in existing geometric approaches. Thus, the unknown parameters in our model is approximately half of existing methods such that under same circumstances, our approach is much less complex than others and can be optimized more efficiently. The paper is organized as follows. Section 2 covers the camera model and traditional representations. Section 3 describes our model and corresponding solutions. The detailed implementation of our approach is presented in section 4. Section 5 provides our experimental results on both synthetic images and real images.

2. CAMERA MODEL AND CLASSICAL REPRESENTATIONS

The pinhole camera assumes that all the rays from objects are mapped onto images through the camera lens center. Assume a point in 3D world is denoted by $w = (X, Y, Z)$ and the corresponding image coordinate is denoted by $q = (x, y)$. If represented by homogeneous coordinates, they become $w = (X, Y, Z, 1)$ and $q = (x, y, 1)$. Then the transformation between w and q is denoted by

$$\gamma q = Pw, \tag{1}$$

where γ is an scale factor and $P \in R^{3\times 4}$ is the product of intrinsic matrix K_{int} and extrinsic matrix K_{ext}. The intrinsic matrix is given by

$$K_{int} = \begin{bmatrix} \alpha & s & c_1 \\ 0 & \beta & c_2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where (c_1, c_2) denote the coordinates of the camera center, α and β denote focal distances along horizontal and vertical axes respectively, and s denotes the skew parameter. The extrinsic matrix K_{ext} is composed of 6 parameters, including 3 rotational parameters and 3 translation parameters. Then the goal of camera calibration is to estimate all the unknowns in the intrinsic matrix K_{int}.

The classic geometric approach estimates the above unknowns by minimizing the following energy function.

$$\sum_{j=1}^{M}\sum_{k=1}^{N}\|\gamma q_{\hat{j,k}} - Pw\|^2, \tag{3}$$

where $q_{\hat{j,k}}$ denotes the measurement in the images, M denotes the number of images, N denotes the number of features in one image, γ, P and w are the same as above. For M images, the total number of unknown parameters is $5 + 6M$. Thus the above energy function is optimized in the space $\mathcal{S} \in R^{5+6M}$. Furthermore, the optimization process includes the computation of the differentiation on trigonometric functions.

3. OUR MODEL AND THE CORRESPONDING SOLUTION

3.1. Basic model

Our model requires at least one square or one rectangle in each object plane while the size of squares or the rectangles is arbitrary. Figure 1 shows the projection from one object plane to the image plane. O_c denotes the camera center. The image plane is denoted by p_c while the object plane is represented by p_w. Without loss of generality, we assume the shape in the plane is a square. Four corners of the square are denoted by p_1, p_2, p_3, p_4 respectively in the anticlockwise direction. Additionally, p_1', p_2', p_3', p_4' denote the corresponding image points of p_1, p_2, p_3, p_4.

The intrinsic parameters in our model are the same as classic models, including $(\alpha, \beta, s, c_1, c_2)$. Instead of the rotational parameter or the translation parameters, the extrinsic parameters in our model are three distances from the camera center to any three corners of the object square. Here let's assume the three corners are p_1, p_2, p_3 in Figure 1 and the corresponding distances are denoted by l_1, l_2, l_3 respectively. The four vectors $\overrightarrow{u_1}, \overrightarrow{u_2}, \overrightarrow{u_3}, \overrightarrow{u_4}$ represents the directions from the camera center to the four image corners (or object corners) which can be easily computed from intrinsic parameters and corner image coordinates.

The geometrical properties of squares give us 6 constraints on the distances and vectors;

$$\begin{gathered}\|l_1\overrightarrow{u_1} - l_2\overrightarrow{u_2}\| = D \quad \|l_2\overrightarrow{u_2} - l_3\overrightarrow{u_3}\| = D \\ \|l_3\overrightarrow{u_3} - l_4\overrightarrow{u_4}\| = D \quad \|l_4\overrightarrow{u_4} - l_1\overrightarrow{u_1}\| = D \\ \|l_1\overrightarrow{u_1} - l_3\overrightarrow{u_3}\| = \sqrt{2}D \quad \|l_2\overrightarrow{u_2} - l_4\overrightarrow{u_4}\| = \sqrt{2}D\end{gathered} \tag{4}$$

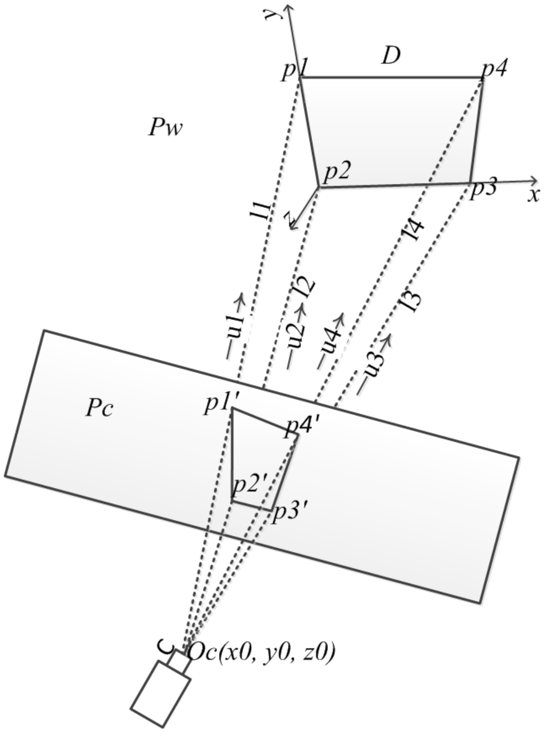

Fig. 1. Projection From 3D World to Image Plane

where D denotes the edge length of the object square. However, the exact value of D is unimportant because it only affects the extrinsic parameters in our model but has little influence on our intrinsic parameters. l_4 is not regarded as unknown parameter since it can be represented by l_1, l_2, l_3 and D which is illustrated in the following part.

3.2. Analysis on our model

Assume the Euclidean coordinate center is located at point p_2, with positive x axis along the direction of $\overrightarrow{p_2p_3}$ and positive y axis along the direction of $\overrightarrow{p_2p_1}$. The z axis is perpendicular to the $xy-$plane. Under such coordinate system, the coordinates of camera center is denoted by (x_0, y_0, z_0). Then we can have the following equations:

$$\begin{aligned} l_2^2 &= x_0^2 + y_0^2 + z_0^2 \\ I_1^2 &= x_0^2 + (y_0 - D)^2 + z_0^2 \\ l_3^2 &= (x_0 - D)^2 + y_0^2 + z_0^2 \\ l_4^2 &= (x_0 - D)^2 + (y_0 - D)^2 + z_0^2 \end{aligned} \tag{5}$$

From the above, we can infer that $l_4^2 = l_1^2 + l_3^2 - l_2^2$.

For M images, the total number of unknown variables to be optimized in our model is $5 + 3M$ which is approximately half of the unknown variables in classical geometric approaches. Moreover, since there are not rotational parameters in our model, numerical calculation of trigonometric function is unnecessary. Thus, our model can reduce the computation complexity greatly.

3.3. Extended model

In practice, planes like checkerboard with many regular square printed on one plane are utilized in order to increase the robustness. Here we extend our geometric model from one square to regular square group. Assume a corner in object planes are denoted by $p_{i,j}$, where (i, j) denotes the horizontal and vertical coordinates of corner $p_{i,j}$ respectively on condition that the center of object plane coordinate is the corner p_{11}. Then the corresponding image point is represented by $p'_{i,j}$. In this way, the energy function can be formulated as follows:

$$E_1 = \|l_{i+1,j}\overrightarrow{u_{i+1,j}} - l_{i,j}\overrightarrow{u_{i,j}}\|^2 - D^2$$

$$E_2 = \|l_{i,j+1}\overrightarrow{u_{i,j+1}} - l_{i,j}\overrightarrow{u_{i,j}}\|^2 - D^2$$

$$E_3 = \|l_{i+1,j+1}\overrightarrow{u_{i+1,j+1}} - l_{i,j}\overrightarrow{u_{i,j}}\|^2 - 2D^2$$

$$E_4 = \|l_{i,j+1}\overrightarrow{u_{i,j+1}} - l_{i+1,j}\overrightarrow{u_{i+1,j}}\|^2 - 2D^2 \qquad (6)$$

where l_{ij} can be represented by l_{11}, l_{12}, l_{22} and D. u_{ij} denotes the direction vector from camera center to p'_{ij} in images. The Gaussian-Newton method is utilized in our approach to find the optimal solution.

4. EXPERIMENTAL RESULTS AND ANALYSIS

All our experiments are tested in MATLAB R2010b on a PC with 3.1GHz Intel Core i5 CPU and 4GB memory. Both synthetic images and real images are used in the comparison.

4.1. Synthetic images free from noise

The checkerboard used in our simulation has 10×8 squares. The edge length of each square is $5cm$. The parameters of simulated camera are as follows: $\alpha = \beta = 500, s = 0.5, c_1 = 405, c_2 = 295$. Image resolution is 800×600. The rotation transformation is represented by 3 angles around x, y and z axes respectively. The value of three angles can be $-30^o, 15^o$ or 45^o. Translation transformation is also performed. Sample synthesized images are shown in Figure 2.

In our experiment, we relied on TOOLBOX released by Bouguet in [9] to extract corners of squares in the checkerboard patterns. Details of our results are shown in Table 1. Observed from the results, our method outperforms the traditional one in both average value and standard deviation. The average running time per trial of our method is 10.675 seconds, while the average running time per trial of traditional method is 21.31 seconds which verifies our judgment that our method is faster than the classical method.

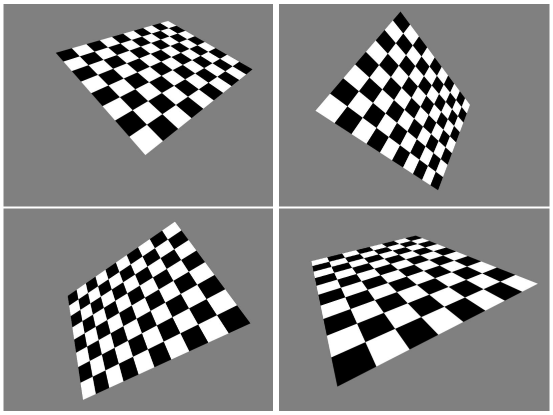

Fig. 2. Sample Synthetic Images without noise

Table 1. Results on synthetic images without noise

Number		(α, β)	s	(c_1, c_2)
3 imgs	tra(mean)	(500.32, 500.41)	0.530	(403.94, 293.98)
	our(mean)	(499.95, 500.24)	0.590	(404.01, 293.79)
	tra(σ)	(0.94, 0.91)	0.380	(0.73, 0.67)
	our(σ)	(1.01, 1.18)	0.705	(0.53, 1.07)
4 imgs	tra(mean)	(500.30, 500.31)	0.440	(404.27, 293.89)
	our(mean)	(500.05, 500.23)	0.485	(403.96, 294.21)
	tra(σ)	(0.66, 0.51)	0.410	(0.70, 0.35)
	our(σ)	(0.56, 0.60)	0.255	(0.39, 0.34)
5 imgs	tra(mean)	(500.17, 500.29)	0.475	(404.03, 293.99)
	our(mean)	(500.30, 500.33)	0.535	(403.86, 294.23)
	tra(σ)	(0.69, 0.78)	0.215	(0.52, 0.34)
	our(σ)	(0.65, 0.83)	0.310	(0.55, 0.42)
6 imgs	tra(mean)	(500.30, 500.28)	0.520	(404.12, 294.12)
	our(mean)	(499.99, 500.22)	0.630	(403.82, 294.30)
	tra(σ)	(0.63, 0.62)	0.265	(0.63, 0.35)
	our(σ)	(0.54, 0.61)	0.165	(0.25, 0.23)
10 imgs	tra(mean)	(500.30, 500.31)	0.595	(404.28, 293.97)
	our(mean)	(500.02, 500.20)	0.560	(403.85, 294.15)
	tra(σ)	(0.40, 0.33)	0.200	(0.45, 0.20)
	our(σ)	(0.32, 0.34)	0.075	(0.12, 0.15)

4.2. Synthetic images with noise added

In the second test, white Gaussian noise is added to our synthetic images. The white Gaussian noise in this part has the following property: mean = 0, variance = 0.1. Sample images are shown in Figure 3. Detailed results are shown in Table 2.

By observing the result table, we can find that our method is still able to have a satisfactory result even when the noise level is so high. In this experiment, the average running time of our method is about 5.48 seconds while that of classical method is about 9.41 seconds.

4.3. Real Images

The resolution of the images used in our experiment is 1296×972. Our checkerboard consists of 9×7 squares whose edges have a length of 35 mm each. The checkerboard was printed and pasted on a wood-board. Fifteen pictures of the checkerboard was taken from different perspectives. Four sample images are shown in Figure 4.

Comparison results between our model and the classical

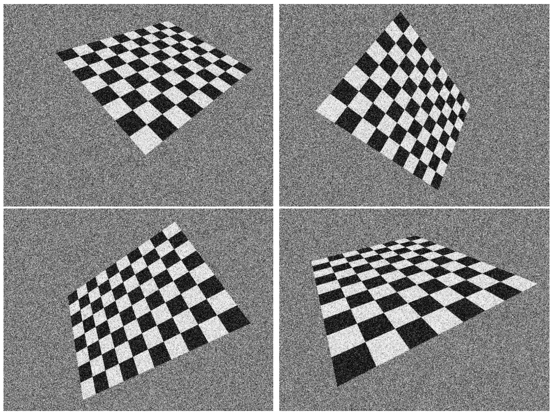

Fig. 3. Sample Synthetic Images with noise added

Table 2. Results on synthetic images with noise added

Number		(α, β)	s	(c_1, c_2)
3 imgs	tra(mean)	(500.75, 500.26)	1.015	(402.77, 295.32)
	our(mean)	(501.24, 501.42)	0.545	(404.28, 294.68)
	$tra(\sigma)$	(4.50, 3.60)	2.235	(3.56, 4.21)
	$our(\sigma)$	(4.95, 5.03)	1.805	(2.47, 3.48)
4 imgs	tra(mean)	(499.52, 499.44)	0.555	(403.72, 293.73)
	our(mean)	(499.93, 501.09)	0.355	(403.75, 294.61)
	$tra(\sigma)$	(3.01, 2.70)	1.360	(2.21, 1.98)
	$our(\sigma)$	(3.29, 3.08)	1.095	(1.41, 1.65)
5 imgs	tra(mean)	(498.55, 498.37)	0.280	(404.12, 294.25)
	our(mean)	(500.11, 500.48)	0.245	(403.52, 294.63)
	$tra(\sigma)$	(2.87, 2.54)	0.690	(1.98, 1.28)
	$our(\sigma)$	(2.96, 2.44)	0.970	(0.99, 1.24)
6 imgs	tra(mean)	(498.84, 499.11)	0.775	(404.17, 293.66)
	our(mean)	(500.31, 500.47)	0.410	(403.78, 294.30)
	$tra(\sigma)$	(2.66, 2.47)	0.815	(1.30, 0.96)
	$our(\sigma)$	(2.65, 2.40)	0.795	(1.11, 1.19)
10 imgs	tra(mean)	(498.30, 498.77)	0.835	(403.65, 293.24)
	our(mean)	(500.46, 500.88)	0.360	(403.81, 294.42)
	$tra(\sigma)$	(2.19, 1.97)	0.765	(1.20, 0.87)
	$our(\sigma)$	(2.01, 1.69)	0.405	(0.70, 0.62)

one are shown in Figure 5. The left part of Figure 5 shows the difference of computed focal distance while the right part shows the difference of computed camera center. The average running time of our method is 13.95 seconds, meanwhile the average running time of the traditional method is 17.67 seconds.

5. CONCLUSION

In this paper, we have proposed a novel geometric method for camera calibration. Three or more images of a planar pattern from arbitrary view are required in our method. Our approach involves no trigonometric function, hence it's more efficient. Observed from the above experimental results and analysis, we can conclude that our method is more accurate and more efficient than the existing geometric methods. Lens distortion is not handled here, it can be a future direction.

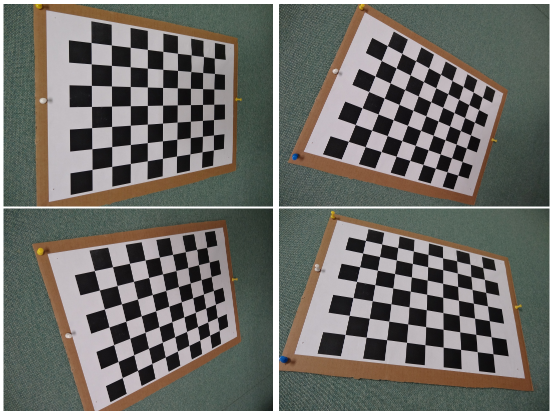

Fig. 4. Sample Real Images

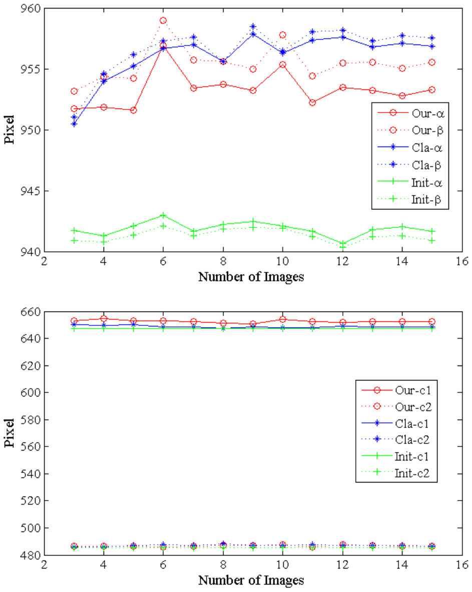

Fig. 5. Results in real images

6. REFERENCES

[1] X. Armangu, J. Salvi, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognition*, pp. 1617–1635, 2000.

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

[3] Ivan E. Sutherland, "Sketch pad a man-machine graphical communication system," in *Proceedings of the SHARE Design Automation Workshop*, 1964, pp. 6329–6346.

[4] R.I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 580–593, 1997.

[5] B. Caprile and V. Torre, “Using vanishing points for camera calibration,” *International Journal of Computer Vision*, pp. 127–140, 1990.

[6] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 666–673.

[7] R. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *IEEE Journal of Robotics and Automation*, pp. 323 –344, 1987.

[8] D. Douxchamps and K. Chihara, “High-accuracy and robust localization of large control markers for geometric camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 376 –383, 2009.

[9] J. Y. Bouguet, “Camera calibration toolbox for matlab,” 2003, `http://www.vision.caltech.edu/bouguetj/calib_doc/`.