International Journal of Advances in Electronics Engineering– IJAEE
Volume 6 : Issue 1 [ISSN : 2278-215X]
Publication Date: 18 April, 2016

Combined Numerical And PSO-based Camera Calibration Focusing On Lens Distortion

Vahid Rashtchi, Mohammad Hemmati

***Abstract*— Camera calibration is one of the most important steps, recognized as a bottleneck, in 3D scanning devices that a very small error in the camera parameters can lead to imprecise 3D models. In the last two decades, different approaches have been taken and improved the precision of the calibration parameters; but still the desire for better precisions and more stability exists.**

In this work, we have used an easy-to-implement and cheap checkerboard as the calibration object. The initial data for calibration is obtained with only three images captured from the checkerboard. To optimize the non-linear camera equations, both numerical and Particle Swarm Optimization (PSO) algorithms together have been applied in order to increase the precision and reduce the sensitivity of the algorithm to noise and errors. Capability of combined numerical and PSO-based algorithm for camera calibration especially lens distortion is demonstrated using a criterion called re-projection error. This work is distinguished from the former works in the application of combined numerical and PSO-based optimizations together in camera calibration focusing on lens distortion. This paper shows that solving camera calibration equations using numerical and PSO algorithms in an integrated manner can lead to acceptable results.

Keywords*— *Camera Parameters; intrinsic and Extrinsic Parameters; PSO; Re-projection Error; Lens Distortion; Checkerboard

I. Introduction

3D computer vision systems are mainly used to build computer 3D models of real objects in the environment or totally extract metric information from 2D images. [1] In all computer vision systems at least a camera exists to capture the environment. The aim of camera calibration is to obtain the mapping relation, described by some parameters, which maps 3D world into camera's 2D plane. [2]. Calibration parameters include internal camera geometric and optical characteristics (intrinsic parameters) and the 3D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters).[3]

Camera calibration is a necessary and very important step in 3D vision systems especially 3D scanners. In most applications such as part inspection, assembly, robots, etc. the final accuracy of the computer model or metric data is

Vahid Rashtchi
Electrical Engineering Department, University Of Zanjan
Zanjan, Iran

Mohammad Hemmati
Electrical Engineering Department, University Of Zanjan
Zanjan, Iran

the final accuracy of the computer model or metric data is highly influenced by the accuracy and precision of this step.[1],[3] So, for many years of study and practice, researchers have sought ways to develop new techniques or improve the existing ones to get better precisions; these techniques fall into the following three main categories:

A. *object-based calibration*

In this technique, a 3D object with good precision is used to calibrate the camera; the object usually consists of two or three orthogonal planes to give enough constraints on equations. [1] The main advantage of this method is its efficiency and high-precision; on the other hand, manufacturing a precise object is expensive and hard. For many years of study, researchers have worked on different 3D objects to maximize the precision and ease of calibration procedure ([5], [6] to cite a few).

B. *Self-Calibration*

This method, does not use any calibration objects; just moving the camera in a static scene, can provide enough constraints on equations and make the calibration possible. This technique is very simple and user-friendly, but it suffers un-reliability and lack of precision. [1]

C. *Enhanced Object-based Calibration*

Because high-precision 3D objects are expensive and hard to manufacture, a new method using 2D objects, especially a checkerboard, was suggested by Zhang [1]. In this technique, a 2D planar object is captured in a few positions having the camera static. This method revolutionized the vision systems from expensive ones to relative cheap and handy ones. Currently, this method is the most common one. During years this method is also improved and many novelties have been offered in [6], [7], [8], etc.

The discussed techniques are categorized according to the required hardware or "data collection" for camera calibration. On the other hand, many works associated with algorithm development and solving calibration equations (After data collection) have been done in order to increase the precision and speed while simplifying the procedure.

Increasing the desire to use heuristic or nature-inspired optimization algorithms such as genetic [9], Particle Swarm Optimization (PSO) [10], etc. and their unique characteristics have tempted researchers into the application of these algorithms into the camera calibration. Genetic Algorithm is used in [11], [12] to calibrate the intrinsic and extrinsic parameters of a camera and has had acceptable accuracy and stability. PSO algorithm has also been used for this purpose in [2], and they have found PSO algorithm applicable to camera calibration; the stability and sensitivity

International Journal of Advances in Electronics Engineering– IJAEE
Volume 6 : Issue 1 [ISSN : 2278-215X]
Publication Date: 18 April, 2016

to noise are also discussed. But they have not applied PSO into solving lens distortion equations.

Our work falls into the "Enhanced Object-based Calibration" category; because we have used a printed checkerboard as the calibration object and with only three images the data for the calibration has been ready. Then we have applied both numerical methods and PSO algorithm into calibration equations. Especially, PSO has been used only for lens distortion finding algorithm; this work shows that precision of the algorithm is improved. The main difference of our work with [2] is the combined method used for camera calibration and PSO-based solution to lens distortion, while the mentioned works use PSO to solve all the camera equations; which this affects the precision and speed inversely.

This paper is configured in seven sections as the following:

In section II, the famous pin-hole model of a camera and related equations are introduced and the calibration parameters are extracted. Section III, explains the calibration procedure and numerical solution into camera equations. In section IV, PSO algorithm and its notations are introduced and the application of it into lens-distortion equations is explained in section V. Section VI contains the simulation results while section VII gives the details on real data. Finally, in section VIII, a conclusion is made based on observations and qualitative criteria.

II. Camera Model

First, we need to model a camera to formulate its behavior; a very famous model for camera is the pin-hole model which it is used to map a point in 3D coordinates into 2D image plane, and vice versa. This model with an extension on lens distortion is completely explained here with details on notations according to Zhang [1].

In a calibration setup, we have 2 coordinates: 3D world coordinates and camera coordinates as shown in Figure. 1. A point in 3D world coordinates and camera coordinates is denoted by $\mathrm{p_w} = [\mathrm{X_w}, \mathrm{Y_w}, \mathrm{Z_w}]^{\mathrm{T}}$ and $\mathrm{p_c} = [\mathrm{x_c}, \mathrm{y_c}, \mathrm{z_c}]^{\mathrm{T}}$; it is obvious that the relation between two different coordinates is determined by a 3×3 rotation matrix describing the rotation of the camera coordinates around x, y, and z axes and a translation vector:

$$p_c = R.p_w + T \qquad (1)$$

in which R and T are the rotation and translation matrices, respectively. Because the mentioned matrices depend on the position of the camera and reference world coordinates, they are called extrinsic matrices containing extrinsic parameters. The projection of $\mathrm{p_c}$ on camera plane gives the 2D point $\mathrm{p_U} = [u, v, 1]^T$; $\mathrm{p_U}$ is determined by applying the intrinsic matrix A to $\mathrm{p_c}$:

$$p_U = \mathrm{s.A.}(R.p_w + T)\,; \quad \mathrm{A} = \begin{bmatrix} \alpha & \gamma & \mathrm{u_0} \\ 0 & \beta & \mathrm{v_0} \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \qquad (2)$$

where s is the arbitrary scale factor, $(\mathrm{u_0}, \mathrm{v_0})$ is the coordinates of the principal point; α and β are the scale factors for image axes u and v, respectively and γ is the parameter describing the skew of the two image axes.

In practice, we have $\mathrm{p_w}$ from the dimensions of the checkerboard and p_{U} from each point in the image, respectively.

Till here, it is assumed that the camera lens is distortion-less; but in practice a lens exhibits two kinds of distortion: radial and tangential [14]. Because the tangential distortion is relatively small and the radial one is the dominant [1], tangential distortion is ignored and only first two terms of

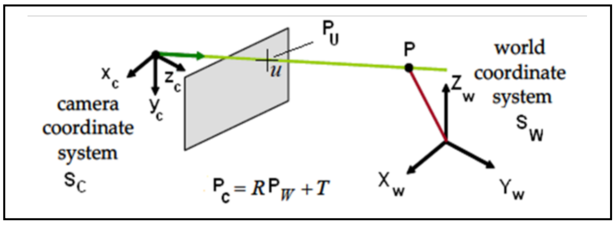

Figure 1. world coordinates vs. camera coordinates.(from [14])

radial distortion is discussed here.

Let (u, v) be the ideal (non-observable distortion-free) pixel image coordinates, and (u', v') the corresponding real observed image coordinates. The ideal points are the projection of the model points according to the pinhole model. Similarly, (x, y) and $(\mathrm{x_c}, \mathrm{y_c})$ are the ideal (distortion-free) and real (distorted) normalized image coordinates. We have [16]:

$$\mathrm{x_c}' = \mathrm{x_c} + \mathrm{x_c}[\mathrm{K_1}(\mathrm{x_c}^2 + \mathrm{y_c}^2) + \mathrm{K_2}(\mathrm{x_c}^2 + \mathrm{y_c}^2)^2] \qquad (3)$$

$$\mathrm{y_c}' = \mathrm{y_c} + \mathrm{y_c}[\mathrm{K_1}(\mathrm{x_c}^2 + \mathrm{y_c}^2) + \mathrm{K_2}(\mathrm{x_c}^2 + \mathrm{y_c}^2)^2] \qquad (4)$$

Using (2) assuming γ=0 we have:

$$\mathrm{u}' = \mathrm{u} + (\mathrm{u} - \mathrm{u_0})[\mathrm{K_1}(\mathrm{x_c}^2 + \mathrm{y_c}^2) + \mathrm{K_2}(\mathrm{x_c}^2 + \mathrm{y_c}^2)^2] \qquad (5)$$

$$\mathrm{v}' = \mathrm{v} + (\mathrm{v} - \mathrm{v_0})[\mathrm{K_1}(\mathrm{x_c}^2 + \mathrm{y_c}^2) + \mathrm{K_2}(\mathrm{x_c}^2 + \mathrm{y_c}^2)^2] \qquad (6)$$

Equations (5) and (6) explain the relation that describes the lens distortion.

It is clear that, in camera calibration we have five intrinsic parameters (matrix A), six extrinsic parameters (three for rotation and three for translation) and finally in our model two lens distortion parameters (K1, K2). In camera calibration we aim to find the mentioned parameters.

III. Camera Calibration procedure and solutions to the equations

The procedure for camera calibration in our method is as the following:

1. **Data Collection:** Capture at least three images from the checkerboard in different orientations; Find the edges as the feature points.
2. **Finding homography:** Assuming $\mathrm{Z_w} = 0$ for each checkerboard, Calculate (optimize) the homography (H_i) between the checkerboard and the image I which H_i satisfies the following equation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{ij} = H_{i3\times 3}\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}_{ij} ; H_i=\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (7)$$

which j is the feature point's number in image i. It must be noted that H_i is defined up to a scale factor. Numerical solution to find H_i is introduced in Appendix A of [16].

3. **Finding A:** By putting equations (2) and (7) equal, we have:

$$H_i A^{-T} = s(R|T)_i \quad (8)$$

Using the orthonormality characteristic of R, we can have two constraints on A using each H_i:

$$h_{1i}{}^T A^{-T} . A^{-1} h_{2i} = 0 \quad (9)$$

$$h_{1i}{}^T A^{-T} . A^{-1} h_{1i} = h_{2i}{}^T A^{-T} . A^{-1} h_{2i} \quad (10)$$

where h_1, h_2, h_3 are the columns 1 to 3 of H_i, respectively. There a closed- form numerical solution for obtaining A which is explained in Appendix B of [15]. It is interesting to know that, there must be at least 3 images to find all the intrinsic parameters.[1], [15].

4. **Finding Extrinsic parameters:** After optimizing A, R and T for each image are calculated with the following equations:

$$r_1 = sA^{-1}h_1 \quad (11)$$

$$r_2 = sA^{-1}h_2 \quad (12)$$

$$r_3 = r_1 \times r_2 \quad (13)$$

$$T = sA^{-1}h_3 \quad (14)$$

which r_1, r_2, r_3 are the columns 1 to 3 of rotation matrix R and s, the scale factor ,is given by:

$$s = \frac{1}{\|A^{-1}h_1\|} = \frac{1}{\|A^{-1}h_2\|} \quad (15)$$

5. **Finding Distortion Parameters:** to simplify the equations (5) and (6), they are integrated into one matrix; we have:

$$\begin{bmatrix} (u-u_0)(x_c{}^2+y_c{}^2) & (u-u_0)(x_c{}^2+y_c{}^2)^2 \\ (v-v_0)(x_c{}^2+y_c{}^2) & (v-v_0)(x_c{}^2+y_c{}^2)^2 \end{bmatrix}\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} u'-u \\ v'-v \end{bmatrix} \quad (16)$$

Now, we have to find the best K_1 and K_2 which satisfies the equation (16). If we write the above equation as **D**k=**d**, the numerical solution to it becomes:

$$k = (D^T D)^{-1} D^T d \quad (17)$$

6. **Un-distorting the image:** using the distortion coefficients K_1 and K_2 and equations (5) and (6) with a small change, new un-distorted images are extracted.

7. **Repeat:** The steps 2 to 6 are repeated as many times as required. It is important to note that in step 2, captured images will be substituted by the un-distorted images.

IV. Particle Swarm Optimization (PSO)

Particle Swarm Optimization, known as PSO, is a very famous and popular population-based optimization algorithm widely used in most of numerical optimization problems. The particles or members of the swarm fly through a multidimensional search space looking for a potential solution. Each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors. For a N-dimensional search space (solving an equation with N variables) the position of the i^{th} particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$. The best previous position (which gives the best fitness value) of the ith particle is recorded and represented as $P_{ibest} = (p_{i1}, p_{i2}, \dots, p_{iN})$. The best one among all the particles in the population is represented as $P_{gbest} = (p_{g1}, p_{g2}, \dots, p_{gN})$. The velocity of each particle is represented as $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})$.

In each iteration, the *P* vectors of the particle with best fitness in the local neighborhood, designated *g*, and the *P* vectors of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. So a fitness function to find the best particle is necessary. The two basic equations which govern the working of PSO are that of velocity vector and position vector given by:

$$V_{i+1} = \omega \times V_i + C_1 \times rnd1_{N\times 1} \times (P_{ibest} - X_i) + C_2 \times rnd1_{N\times 1} \times (P_{gbest} - X_i) \quad (18)$$

$$X_{i+1} = X_i + V_{i+1} \quad (19)$$

where ω is the inertia weight and depending on the current iteration; as iteration increases ω becomes smaller to let the algorithm search the values in the near neighborhood of the best answer. C_1 and C_2 are PSO coefficients that normally both of them are equal 2. rnd1 and rnd2 are random vectors produced in each iteration. The number of iteration depends on how much deviation is acceptable for fitness function; on the other hand, in some application, time of execution limits number of iteration.

V. Application Of PSO Into Lens Distortion Equations

In this work, lens distortion equation (16), is a good candidate for PSO algorithm in order to increase the precision of intrinsic parameters. Because of K_1 and K_2, N becomes 2. Number of particles is set to 20. $C_1 = C_2 = 2$. In a heuristic manner, ω is obtained as the following for iteration "iter":

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \times iter/iteration_{max} \quad (20)$$

ω_{max} and ω_{min} are assigned heuristically and $iteration_{max}$ is the number of allowed iterations for the algorithm. According to (16), minimization fitness function is calculated by:

$$fit = \sum_{i=1}^{Nimages} \sum_{j=1}^{Npoints} (Err_u(i,j)^2 + Err_v(i,j)^2) \quad (21)$$

Where

$$\mathrm{Err}_u(i,j) = (u'(i,j) - [u(i,j) + (u(i,j) - u_0)[K_1(x_c(i,j)^2 + y_c(i,j)^2) + K_2(x_c(i,j)^2 + y_c(i,j)^2)^2]]) \quad (22)$$

$$\mathrm{Err}_v(i,j) = (v'(i,j) - [v(i,j) + (v(i,j) - v_0)[K_1(x_c(i,j)^2 + y_c(i,j)^2) + K_2(x_c(i,j)^2 + y_c(i,j)^2)^2]]) \quad (23)$$

VI. Simulation Results And Analysis

Using some calibration data given by Zhang [15] a simulation has been done. First three images each including 256 feature points is generated using the parameters in the 2nd column of table. 1. Both numerical methods and PSO-based lens distortion are executed and the results for intrinsic matrix A are provided in Table. I. It is significant to note that in this simulation, CCD noise, Image processing error, and tangential lens-distortion have not been inserted in the generated data.

TABLE I. SIMULATION DATA RESULTS FOR MATRIX A AND DISTORTION COEFFICIENTS (FOR 100 ITERATIONS OF THE CALIBRATION STEPS)

Par	*Image Generation Parameters*	*Numerical Optimization Result*		*PSO-based Optimization Results*	
		Value	*~Err %*	*Value*	*Err %*
α	830.8	830.8	0	830.8	0
γ	0	0	0	0	0
β	830.69	830.69	0	830.69	0
u_0	305.77	305.72	0.02	305.77	0
v_0	206.42	206.46	0.02	206.42	0
K1	-0.229	-0.22711	0.83	-0.229	0
K2	0.196	0.18633	6.31	0.196	0

The simulation results show that PSO algorithm has the enough capability to be as accurate as numerical methods in this application. Also, when lens distortion equation is solved with PSO and other steps with numerical methods, we have better results for the principal point(u_0, v_0). As it is clear in the last two rows of Table I, the accuracy of the PSO-based technique for finding lens distortion parameters is better.

VII. Experimental Results And Analysis

To test the algorithm with real data which all environmental effects such as CCD noise, image processing error and higher-order distortions exist, three images are captured with a Logitech® C920 camera shown in Figure. 2. In order to check the method's accuracy, the points are re-projected using the calibrated parameters; then for each point sum of abstract errors between image points and re-projected points are calculated; finally, their average is reported as a criterion for checking the accuracy of calibrated parameters called re-projection error. The results on accuracy of the methods are shown in Table. 2. It is inferred from the results that in the worst case of PSO, it is as accurate as numerical method.

Figure 2. Captured Images from the checkerboard by logitech® C920

TABLE II. EXPERIMENTAL RESULTS

Numerical Optimization Re-projection Error (pixels)	*PSO-based Optimization Re-projection Error (pixels-worst case)*
1.48	1.38

VIII. Conclusion

In this work, we applied PSO algorithm only into lens distortion equation and the other optimization problems in the calibration procedure were solved by numerical methods (combined method) to have better speed and precision. Both results of simulation and experimental results demonstrated that PSO algorithm can be a good and reliable choice to solve the distortion equation; because the intrinsic parameters in PSO-based method have better precisions.

As a future work, one can apply the PSO algorithm into other equations of camera calibration in different combinations of numerical and PSO optimizations and find the best combination.

References

[1] Z.Zhang, "A Flexible New Technique for Camera Calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, November 2000.

[2] X. Song, B. Yang, Z. Feng, T. Xu, D. Zhu and Y. Jiang, "Camera Calibration Based on Particle Swarm Optimization", 2nd International Congress on Image and Signal Processing (CISP), China, 17-19 October, 2009.

[3] R.Y. 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, vol. 3, no. 4, pp. 323-344, August, 1987.

[4] X. Hui-yuan, X. You, Z. Zhi-jian, "Accurate extrinsic calibration method of a line structured-light sensor based on a standard ball", IET Image Processing- Special Issue on Imaging Systems and Techniques, January, 2010.

[5] J. S. Kim, I. S. Kweon, "Camera calibration based on arbitrary parallelograms", Computer Vision and Image Understanding, vol. 113,pp. 1-10, June, 2008.

[6] Z. Song and R. Chung, "2008-Use of LCD Panel for Calibrating-Structured-Light Based Range Sensing System", IEEE Transactions On Instrumentation And Measurement, vol. 57, Issue. 11, pp. 2623 – 2630, June, 2008.

[7] A. Ben-Hamadou, C. Soussen, C. Daul, W. Blondel, D. Wolf, "2013-Flexible calibration of structured-light systems projecting point patterns", Computer Vision and Image Understanding, vol. 117, pp. 1468–1481, June, 2013.

[8] F. Bergamasco, L. Cosmo, A. Albarelli and A. Torsello, "Camera Calibration from Coplanar Circles", 22nd International Conference on Pattern Recognition, Stockholm, Sweden, August 24-28, 2014.

[9] J. H. Holland, Adaption in Natural And Artificial Systems", University Of Michigan Press, Ann Arbor, MI, 1975.

[10] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.

[11] Q. Ji and Y. Zhang, "Camera Calibration with Genetic Algorithms", IEEE Transactions on Systems, Man, and Cybernetics, vol. 31, no. 2, March, 2001.

[12] S. Hati, S. Sengupta, "Robust Camera Parameter Estimation Using Genetic Algorithm", Pattern Recognition Letters, no.22, pp. 289-298, 2001.

[13] D. Lanman, and G. Taubin, "Build Your Own 3D Scanner: 3D Photography for Beginners" , ACM SIGGRAPH2009, August, 2009.

[14] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no.10 , pp.965–980, October, 1992.

[15] Z. Zhang, "A Flexible New Technique for Camera Calibration," Updated Technical Report, Microsoft Research, Microsoft Corporation, December 2009.

About Authors:

Vahid Rashtchi was born in 1967 in Zanjan, Iran. He received his B.Sc. in electrical engineering from Tabriz University in 1991 and his M.Sc. and Ph.D. in electrical engineering from Amirkabir University of Technology in 1993 and 2001, respectively.
He is currently working as an associate professor at University Of Zanjan, working on power electronics and applications of artificial intelligent systems since 2001. Up to now, he has had several outstanding positions such as "Head of the Engineering Faculty of University Of Zanjan" and "Research Vice Chancellor of University Of Zanjan".

Mohammad Hemmati received his B.Sc degree from University Of Zanjan, Zanjan, Iran in Electrical and Electronics Engineering, in 2014. He is currently doing research on implementation of 3D scanners and camera calibration techniques as the Master student in Electrical Engineering department at University Of Zanjan.
His research interests include computer vision, FPGA-based digital design and fault-tolerant systems.