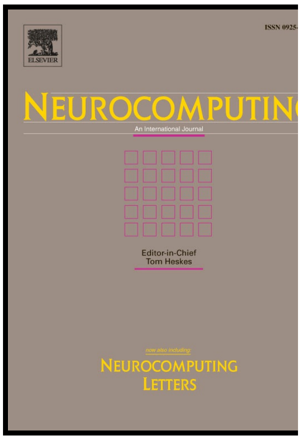

www.elsevier.com

A novel camera calibration technique based on differential evolution particle swarm optimization algorithm

Li Deng, Gen Lu, Yuying Shao, Minrui Fei, Huosheng Hu

PII: S0925-2312(15)01250-3
DOI: http://dx.doi.org/10.1016/j.neucom.2015.03.119
Reference: NEUCOM16010

To appear in: *Neurocomputing*

Received date: 22 October 2014
Revised date: 9 March 2015
Accepted date: 15 March 2015

Cite this article as: Li Deng, Gen Lu, Yuying Shao, Minrui Fei and Huosheng Hu, A novel camera calibration technique based on differential evolution particle swarm optimization algorithm, *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2015.03.119

A novel camera calibration technique based on differential evolution particle swarm optimization algorithm

Li Deng[a,c], Gen Lu[a,c], Yuying Shao[b], Minrui Fei[a,c,*], Huosheng Hu[d]

[a]*School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200072, PR China*
[b]*State Grid Shanghai Municipal Electric Power Company, Shanghai 200025, PR China*
[c]*Shanghai Key Laboratory of Power Station Automation Technology, Shanghai 200072, PR China*
[d]*Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, U.K.*

Abstract

Camera calibration is one of the fundamental issues in computer vision and aims at determining the intrinsic and exterior camera parameters by using image features and the corresponding 3D features. This paper proposes a relationship model for camera calibration in which the geometric parameter and the lens distortion effect of camera are taken into account in order to unify the world coordinate system (WCS), the camera coordinate system (CCS) and the image coordinate system (ICS). Differential evolution is combined with particle swarm optimization algorithm to calibrate the camera parameters effectively. Experimental results show that the proposed algorithm has a good optimization ability to avoid local optimum and can complete the visual identification tasks accurately.

Keywords: Camera calibration; Internal parameter; External parameter; Differential evolution; Particle swarm algorithm; Visual identification

[*]Corresponding author.
Email address: mrfei@staff.shu.edu.cn (M. Fei), dengli@shu.edu.cn (L. Deng).

1. Introduction

Camera calibration has been studied extensively in computer vision for a long time and many calibration methods have been proposed [1, 2, 3, 4]. The basic camera calibration methods can be divided into the traditional camera calibration and the self-calibration [5, 6, 7, 8]. The traditional calibration method has high calibration accuracy, but need specific calibrations reference substance. The self-calibration method does not rely on calibration reference substance, but the calibration results are relatively unstable.

In [9], the direct linear transformation (DLT) method was used for camera calibration, in which the camera imaging geometry linear model was established and the imaging model parameters can be directly obtained by the linear equation [9]. However the camera distortion was not considered. To avoid this kind of problem, a most common method of two-step calibration was proposed in [10]. Most parameters in the model were firstly obtained with the radial alignment constraint (RAC), and then the nonlinear search method was used to solve the distortion coefficient, effective focal length and a translation parameter. The method considers the radial lens distortion, assuming the main point of image lies at the image center. Coplanar points were used to solve both interior and exterior parameters.

Zhang [11] took into account the radial distortion and proposed a camera calibration method based on the plane pattern, which used the orthogonal condition of rotation matrix and nonlinear optimization in camera calibration. The method is simple and flexible, but needs multiple planar template images of different perspective as the calibration reference object and the distortion is considered deficiently.

Camera self-calibration method was firstly introduced by Faugeras and Maybank in 1992 [12], which did not need any special external calibration object and directly completed the calibration tasks by images. It has a great deal of flexibility and has been widely deployed. In [13], Ma proposed a linear method based on two groups of three orthogonal movements, which is the self-calibration method based on active vision. It used multiple images and the camera motion to determine the camera internal and external parameters. However, its computational cost is very high.

In recent years, some nature-inspired swarm intelligence optimization algorithms have been widely applied in visual calibration and detection, and achieved the remarkable effect and superiority [14, 15, 16, 17]. For instance, a camera calibration method based on the global search ability of particle swarm algorithm was presented in [18, 31, 32]. It can solve the camera parameters very well. But without giving full consideration to the distortion, the calibration precision remains to be future improved. Tian et al. [19] used the local convergence and stability of BP neural networks to improve the camera calibration accuracy and robustness. However, the limitations of these intelligent algorithms are easy to fall into the problems such as premature convergence or inefficiency, which is unable to find the optimal solution. To prevent these problems, it is necessary to combine many kinds of algorithms [20, 21]. Li et al. [22] combined genetic algorithm with particle swarm optimization for camera calibration, in which the crossover and mutation of genetic algorithm are adopted to avoid particle swarm prematurely into local optimum.

As we know, Particle Swarm Optimization (PSO) is a kind of global random search algorithm based on swarm intelligence [23]. It has less adjustable parameters and is easy to implement than the traditional intelligent algorithms such as genetic algorithm (GA), as well as the better ability of global optimization. Differential Evolution (DE) is a kind of global random search algorithm based on real parameter optimization problem [24]. It has a fast convergence speed, less adjustable parameters and good robustness. By combining both algorithms, a differential evolution particle swarm optimization (DEPSO) is presented in [25]. This paper proposes the use of DEPSO in camera calibration. It is more advanced than the existing single algorithm based on camera calibration techniques in [18, 19, 31, 32] that have the risk of premature convergence, or complex hybrid algorithm based on camera calibration techniques in [20, 21, 22] that have the weaker ability of optimization and are not easy to implement.

The rest of this paper is organized as follows. Section 2 introduces the principle of camera calibration, including the establishment of the calibration model. Section 3 presents an improved camera calibration algorithm for solving the camera parameters. Some experimental results and analysis are presented in Section 4 to verify the feasibility and performance of the proposed algorithm. Finally, a brief conclusion and the future work are given in Section 5.

2. Camera calibration principle

2.1. Calibration parameters description

In computer vision, the imaging model is considered as the geometric objects in 3D space to the projection relationship of 2D image corresponding points. The corresponding geometric parameters are camera parameters and obtained by camera calibration [3, 8, 26]. The camera parameters can be divided into internal and external parameters.

2.1.1. Camera internal parameter

Internal parameter is the relationship between CCS and ICS, representing the camera optical and geometry features such as the main point (image centre), focal length, radial lens distortion, off-axis lens distortion and so on [27]. The mathematical model of camera internal parameter can be expressed in a 3×3 matrix.

$$A = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $f_x = f/d_x$ and $f_y = f/d_y$ are respectively the camera image plane scale factor of horizontal axis x and vertical axis y. f is the focal length of the camera. d_x and d_y are the physical distance of horizontal direction and vertical direction on the image plane. c_x and c_y are the offset of horizontal direction and vertical direction between the camera center and optical axis. s is the obliquity factor which is zero when the camera image plane coordinate axis is not orthogonal each other. The camera internal parameter calibration is mainly solving the four parameters (f_x, f_y, c_x, c_y).

2.1.2. Camera external parameter

External parameter denotes the location and direction of the camera in WCS, i.e. the relationship of unifying CCS and WCS [28]. It includes the rotation matrix R and translation matrix T as follows.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, T = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$$

where each direction vector is satisfied orthogonal constraint condition.

If α, β and θ denote the rotation angle of camera in the x, y, z axes and the direction of rotation is clockwise, then each axis transformation matrix is given below.

$$R_x\left(\alpha\right)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{bmatrix},\ R_y\left(\beta\right)=\begin{bmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{bmatrix},$$

$$R_z\left(\theta\right)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

The total rotation matrix is as follows:

$$\begin{aligned}&R=R_z\left(\theta\right)*R_y\left(\beta\right)*R_x\left(\alpha\right)\\ &=\begin{bmatrix}\cos\theta\cos\beta & \cos\theta\sin\beta\sin\alpha-\sin\theta\cos\alpha & \sin\theta\sin\alpha+\cos\theta\sin\beta\cos\alpha\\ \sin\theta\cos\beta & \cos\theta\cos\alpha+\sin\theta\sin\beta\sin\alpha & \sin\theta\sin\beta\cos\alpha-\cos\theta\sin\alpha\\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha\end{bmatrix}\end{aligned}$$

The translation matrix T is the offset between the two coordinate systems. The transformation from the point P_w in WCS to the point P_c in CCS is expressed as:

$$P_c=R\left(P_w-T\right)$$

By solving the external parameter, the unity of the camera coordinate system (CCS) and the world coordinate system (WCS) can be realized. Then, the internal parameters of camera can be combined to realize the unity between the coordinate systems, so as to realize the mapping from the 3D space to the 2D image and to achieve the goal of visual identification.

2.2. Camera imaging relationship

Camera imaging relationship can be determined by internal and external parameters of camera, and the transformation model is established from 3D space to 2D images. Assuming a 3D point of the world coordinate is $(x_w, y_w, z_w)^T$, a homogeneous coordinate of CCS is $(x_c, y_c, z_c)^T$, a 2D image pixel coordinate is $(u, v)^T$, the camera imaging relationship is expressed as follows.

2.2.1. The transformation between WCS and CCS

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_{R,T} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{1}$$

According to the transformation relation of rotation matrix and translation matrix, the coordinate transformation between the two coordinate systems can be realized.

2.2.2. The transformation between CCS and IPCS

The ideal perspective projection transformation under pinhole model is established as follows:

$$x = f \cdot x_c/z_c,\ y = f \cdot y_c/z_c$$

Using homogeneous coordinate and matrix are written as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{2}$$

where $(x, y)^T$ is the homogeneous coordinate of the image physical coordinate system (IPCS). The transformation between IPHCS (image physical coordinate system) and PCS (pixel coordinate system) is expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

where u_0 and v_0 are the intersection coordinate between the optical axis center and image plane. From (2) and (3), the transformation between CCS and IPCS can be obtained by:

$$
\begin{aligned}
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \frac{1}{z_c} \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \\
&= \frac{1}{z_c} \begin{bmatrix} f/d_x & 0 & u_0 & 0 \\ 0 & f/d_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}
\end{aligned} \tag{4}
$$

2.2.3. The transformation between WCS and IPCS

According to the transformation relation between the above coordinate systems, i.e. (1) - (4), the final camera imaging relationship is shown as:

$$
\begin{aligned}
z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} f/d_x & 0 & u_0 & 0 \\ 0 & f/d_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \\
&= M_A M_{R,T} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}
\end{aligned} \tag{5}
$$

where M_A is the camera internal parameter array, $M_{R,T}$ is the camera external parameter array. They represent the basic relationship between the 2D pixel coordinate and 3D world coordinate. According to the known world coordinate, using the matrix transformation relation can calculate the corresponding image pixel coordinate and realize the mapping from 3D space to 2D image.

2.3. Camera calibration model

The camera imaging model is the basis of camera calibration and determines the camera parameters. It is divided into either linear model or nonlinear model [1, 3, 8]. The linear model is based on the principle of pinhole imaging, and its geometric relationship is established between the image points and the corresponding object space points. In practical applications, the physical structure of camera leads to many kinds of distortion. The actual image point position may offset the ideal image point position, and damage the collinear relationship among the image points, the projection centre and the corresponding space points. Therefore, the nonlinear model needs to be established for distortion correction. Fig. 1 shows the imaging model.

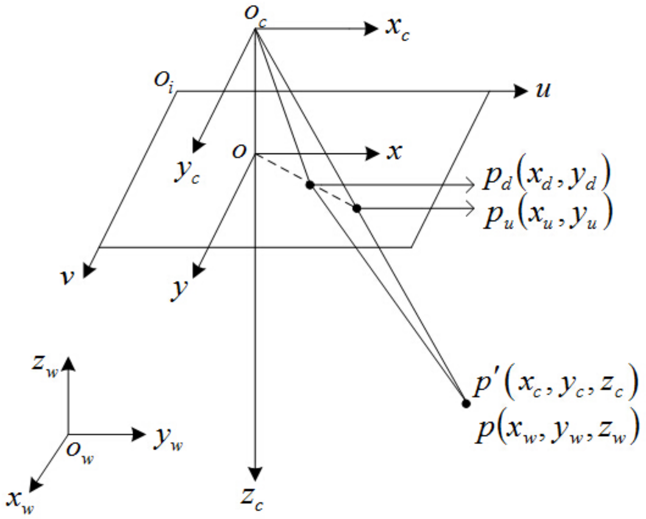

Figure 1: Camera calibration imaging model

(1) IPCS (image pixel coordinate system-o_iuv): the origin o_i is located in the upper left corner of the CCD image plane. The coordinate axis u and v are respectively parallel to the image plane pixel rows and columns. (u, v) denotes the pixel coordinate of object point P in the image coordinate system.

(2) IPHCS (image physical coordinate system-oxy): the origin o is located in the intersection of the camera optical axis and the CCD image plane. The coordinate axis x and y are respectively parallel to the axis u and v. $p_u(x_u, y_u)$ is the image coordinate of object point P in the ideal pinhole model. $p_d(x_d, y_d)$ denotes the actual image coordinate in the pinhole model considered the lens distortion.

(3) CCS (camera coordinate system-$o_cx_cy_cz_c$): the origin o_c is located in the optical center of camera. The coordinate axis x_c and y_c are respectively parallel to the axis x and y. The axis z_c is the camera main optical axis. o_co is the camera effective focal length f. (x_c, y_c, z_c) denotes the coordinate of object point P in the camera coordinate system.

(4) WCS (world coordinate system-$o_w x_w y_w z_w$): it is a reference coordinate system in an appropriate choice of the space environment. (x_w, y_w, z_w) denotes the coordinate of object point P in the world coordinate system.

2.3.1. Solving camera parameters

The camera parameters can be determined according to the camera imaging model. In order to simplify the calculation, it is assume that the template plane is on $z_w = 0$ in WCS. The i^{th} column of the rotation matrix R is denoted as r_i. From (5), the points of the template plane are expressed as follows:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_A M_{R,T} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{6}$$

To define the homography H, the mapping between the point on template plane and its corresponding image point is $H = M_A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$. So the transformation can be written below:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{7}$$

Namely:

$$\begin{bmatrix} x_w & y_w & 1 & 0 & 0 & 0 & -ux_w & -uy_w & -u \\ 0 & 0 & 0 & x_w & y_w & 1 & -vx_w & -vy_w & -v \end{bmatrix} h = 0 \tag{8}$$

Given an image of the template plane, a homography can be denoted as $H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$. It can be expressed as:

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda M_A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{9}$$

where λ is the scale factor. Because r_1 and r_2 are orthogonal vectors, thus $r_1^T r_1 = r_2^T r_2 = 1$ and $r_1^T r_2 = 0$. Then the constraints are given by:

$$h_1^T {M_A}^{-T} {M_A}^{-1} h_2 = 0 \tag{10}$$

$$h_1^T {M_A}^{-T} {M_A}^{-1} h_1 = h_2^T {M_A}^{-T} {M_A}^{-1} h_2 \tag{11}$$

These are the two basic constraints for solving the camera intrinsic parameter. Assuming $B = {M_A}^{-T}{M_A}^{-1}$, it can be written as:

$$B = {M_A}^{-T}{M_A}^{-1} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & -\frac{c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & -\frac{c_y}{f_y^2} \\ -\frac{c_x}{f_x^2} & -\frac{c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix} \tag{12}$$

It can be seen that matrix B is symmetric, which can be defined as a six-dimensional vector $b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T$. The i^{th} column of the matrix H is denoted as $h_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T$. From (12), the two constraints can be expressed as $h_i^T B h_j$, namely:

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}^T \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \tag{13}$$

According to the constraints (10) and (11), the two homogeneous equations about vector b can be expressed as follows:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{14}$$

If n images of the template plane are observed, the linear equation can be given from (14).

$$Vb = 0 \tag{15}$$

where V is $2n \times 6$ matrix. If $n \geq 2$, the analysis shows that the above equations have the linear solution. Then the camera intrinsic parameter can be obtained from the closed-form solution of matrix B.

$$\begin{cases} f_x = \sqrt{\lambda / B_{11}} \\ f_y = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ c_x = -B_{13} f_x^2 / \lambda \\ c_y = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \end{cases} \tag{16}$$

According to the intrinsic parameter and homography H, the camera extrinsic parameter can be obtained by:

$$\begin{cases} r_1 = \lambda M_A^{-1} h_1 \\ r_2 = \lambda M_A^{-1} h_2 \\ r_3 = r_1 \times r_2 \\ t = \lambda M_A^{-1} h_3 \end{cases} \tag{17}$$

where scale factor λ is determined by orthogonal conditions, $\lambda = 1/\left\| M_A^{-1} h_1 \right\| = 1/\left\| M_A^{-1} h_2 \right\|$. It is important to note that the solved matrix $R = \left[\begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \right]$ does not in general satisfy the properties of a rotation matrix. So the singular value decomposition (SVD) method is used to obtain the best rotation matrix here.

2.3.2. Distortion correction

Note that the lens distortion of a camera is not considered in the above calibration process. Since the used camera lens produce large geometric distortion in the region far away from the center of an image [11, 29, 30], the radial distortion and tangential distortion are dealt with. As for radial distortion, the center of the imaging plane distortion is 0. When moving to the edge, the distortion becomes more serious. The several terms of Taylor series expansion around the location $r = 0$ can be used to quantitative description. For a general camera, we usually use the first two coefficients, i.e. k_1 and k_2. As for a big distortion camera such as fisheye lens, the third radial distortion item k_3 can be used. The three distortion coefficients are adopted here to correct the distortion as follows:

$$\begin{cases} x_u = x_d \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \\ y_u = y_d \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \end{cases} \tag{18}$$

The tangential distortion is due to the lens manufacturing defects, i.e. the lens in un-parallel with the image plane, and is commonly described by two parameters p_1 and p_2 as follows:

$$\begin{cases} x_u = x_d + \left[2 p_1 y_d + p_2 \left(r^2 + 2x_d^2\right)\right] \\ y_u = y_d + \left[p_1 \left(r^2 + 2y_d^2\right) + 2 p_2 x_d\right] \end{cases} \tag{19}$$

where $r^2 = x_d^2 + y_d^2$.

The inference of the two kinds of distortion above is considered in the process of calibration. After transforming (18) and (19), the distortionless calibration results can be obtained by:

$$\begin{aligned}\begin{bmatrix} x_u \\ y_u \end{bmatrix} &= \left[1 + k_1\left(x_d^2 + y_d^2\right) + k_2(x_d^2 + y_d^2)^2 + k_3(x_d^2 + y_d^2)^3\right]\begin{bmatrix} x_d \\ y_d \end{bmatrix} \\ &+ \begin{bmatrix} 2p_1 x_d y_d + p_2\left(3x_d^2 + y_d^2\right) \\ p_1\left(x_d^2 + 3y_d^2\right) + 2p_2 x_d y_d \end{bmatrix}\end{aligned} \tag{20}$$

Here the nonlinear model introduced distortion is used to calibrate camera. The camera parameters are redefined and evaluated, which can be obtained by a large number of equations. In the practical application, distortion correction can effectively reduce the influence brought of the distortion and ensure the accuracy of calibration results, which has very important research value.

3. Algorithm design and application

3.1. Particle swarm optimization

PSO algorithm is based on the evolution of population and has been successfully applied in many optimization problems [16, 18, 31, 32]. It is initialized to a group of random particles and updated themselves by tracking two ”extremum” in each iteration. Assuming that the information of particle i is a D-dimensional vector, its location is represented by $X_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$, its speed is represented as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$. The renewal equations of speed and position can be expressed as follows:

$$v_{id}^{k+1} = wv_{id}^k + c_1 rand()\left(pbest - x_{id}^k\right) + c_2 rand()\left(gbest - x_{id}^k\right) \tag{21}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{22}$$

where w is the inertia weight, which is affecting the global and local searching ability of the algorithm. c_1 and c_2 are the accelerated factor (or the learning factor), which respectively adjust the maximum step of the global and individual best particle in the flying direction. If they are too small, the particles may be far away from the object region, otherwise, may lead to fly over the object region. The appropriate c_1 and c_2 can accelerate the convergence and not easy to fall into local optimum, usually $c_1 = c_2 = 2$. $rand()$ is a random number between [0, 1]. *pbest* is the individual extreme value point location of the particle i. *gbest* is the global extreme value point location of the whole population.

3.2. Differential evolution particle swarm optimization

Differential evolution (DE) is a kind of global optimization algorithm based on real number coding and the evolution of population, which shows the great advantages in many optimization problems and has been successfully applied in many fields [33, 34]. On the other hands, particle swarm optimization has some limitations, such as premature and slow convergence, population diversity decreased too fast with algebraic increasing and may not converge the global optimal solution and so on. Therefore, we propose a new algorithm here, namely differential evolution particle swarm optimization (DEPSO).

In our algorithm, the mutation and crossover operation of differential evolution are introduced at each iteration into the PSO, which can maintain the diversity of population particles and select the optimal particle at each iteration to the next iteration. It can improve the convergence of the algorithm and prevent the particles into premature convergence [25, 33, 35].

Assuming the population size is N and each individual has a D-dimensional vector, the target vector and the test vector are respectively represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$ and $V_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$, $i = 1, 2, \cdots, N$. The initial population is $S = \{X_1, X_2, \cdots, X_N\}$. The each target vector of G^{th} generation individual is represented as $X_{i,G}$. Mutation operator is usually used as follows:

$$v_{i,G} = x_{best,G} + F\left(x_{r2,G} - x_{r3,G}\right) \tag{23}$$

$$v_{i,G} = x_{r1,G} + F\left(x_{r2,G} - x_{r3,G}\right) \tag{24}$$

$$v_{i,G} = x_{i,G} + F_1\left(x_{best,G} - x_{i,G}\right) + F_2\left(x_{r1,G} - x_{r2,G}\right) \tag{25}$$

$$v_{i,G} = x_{best,G} + F\left(x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G}\right) \tag{26}$$

$$v_{i,G} = x_{r1,G} + F\left(x_{r2,G} + x_{r3,G} - x_{r4,G} - x_{r5,G}\right) \tag{27}$$

where $r1, r2, r3, r4, r5$ are random unequal integers between [1, N]. F is the mutation control parameter, which controls the magnification of difference and general value in [0, 2]. It affects the convergence speed of DE.

In this paper, Eq. (24) is used as an operating mode of population particle mutation. The mutation mechanism is introduced into the particle swarm iteration computation, which generates mutation after each iteration and prevents particle swarm premature into a local optimum. In order to increase the diversity of population, the crossover operation is introduced. The test vector V_i and the target vector X_i are permeated to achieve the purpose of improving the population global search ability. Assuming the cross vector is $U_i = (u_{i1}, u_{i2}, \cdots, u_{iD})^T$, and the crossover operator is:

$$u_{j,i,G} = \begin{cases} v_{j,i,G},\ (rand_j \leq CR)\, or\, (j = j_{rand}) \\ x_{j,i,G},\ else \end{cases} \tag{28}$$

where $j = 1, 2, \cdots, N$, $j_{rand} \in [1, N]$. CR is a cross control parameter, i.e. a general value in [0, 1]. The big the value is, the better the crossover probability and the diversity of population.

The candidate individual $U_{i,G}$ is evaluated by fitness function in order to decide whether the new generation individual should be selected with the following selecting operator:

$$x_{i,G+1} = \begin{cases} u_{i,G},\ f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G},\ else \end{cases} \tag{29}$$

The optimal individual is chosen at each iteration to achieve optimization purpose.

3.3. Algorithm for camera calibration

According to the analysis of the camera model, the proposed DEPSO algorithm is applied to obtain camera parameters during camera calibration. The 2D coordinates of the obtained images are compared with the actual measured image coordinates in order to verify the feasibility and superiority of the algorithm.

3.3.1. Objective function

The camera internal and external parameters and distortion coefficients are to be optimized in this research. Assuming we have n plane template images each of which has m calibration points. Each point is equal in size and located in the same noise environment. Objective function is established as follows:

$$f_{obj} = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \hat{p}_{ij} - p\left(M_A, k_1, k_2, k_3, p_1, p_2, R_i, T_i, P_j\right) \right\| \tag{30}$$

where $\hat{p}_{ij}$ is the pixel coordinate of the j^{th} calibration point in the i^{th} image. R_i and T_i are the rotation and translation matrices corresponding to the i^{th} image. Note that $p\left(M_A, k_1, k_2, k_3, p_1, p_2, R_i, T_i, P_j\right)$ is obtained by using (5), which is the pixel coordinate of the j^{th} world coordinate point P_j in the i^{th} image. The objective function is optimized by the proposed algorithm, and then the optimal stable solution of camera parameters can be obtained.

3.3.2. Algorithm application

The application process of differential evolution particle swarm optimization algorithm is given as follows:

Step 1 Population initialization: it randomly generates the position and speed of N particles within the allowed scope, and sets the upper and lower limit of particle velocity. In order to avoid the unstable search ability of the proposed algorithm, a kind of dynamic adjustment strategy is adopted to adjust the algorithm parameters (w, F, CR). Setting the upper and lower limit of parameters, the maximum iteration number $\lambda_{\max}$ and the current iteration number λ, then the dynamic adjustment strategy can be written as:

$$\delta\left(w, F, CR\right) = \delta_{\max} - \frac{\delta_{\max} - \delta_{\min}}{\lambda_{\max}}\lambda \tag{31}$$

Step 2 Fitness function selection: the objective function is the distance of the obtained pixels and the actual pixels, which is used as the fitness evaluation standard and calculated to get the individual extremum and global extremum of the initialization population. Fitness function is expressed as:

$$fitness = \min \sum_{i=1}^{m} \sqrt{\left(u_i - x_i\right)^2 + \left(v_i - y_i\right)^2} \tag{32}$$

Step 3 Renewing population: according to (21) and (22), the speed and position of each particle are renewed.

Step 4 Selective renewal: using the selection strategy of differential evolution, the fitness of the renewed particle is compared with the fitness of particle before renewing, and choosing higher fitness particle to update location.

Step 5 Crossover operation: increasing diversity of the population and ensuring that the excellent individuals have a high fitness.

Step 6 Mutation operation: according to comparing fitness value, the low fitness individuals are generated mutation with greater probability, which is beneficial to produce excellent schema and guarantee the existence of superior individuals. Thus a new generation of excellent population is forming.

Step 7 Renewing the population extreme: according to the fitness value of a new generation population, renewing the individual extreme and global extreme of population.

Step 8 Determining whether the termination condition is satisfied: if it reaches the maximum number of iteration, then the end of the loop and output results, otherwise go to Step 3 to continue iteration.

4. Experiments

4.1. Experimental preparation

In order to verify the application performance of the proposed algorithm, a company visual identification project is chosen as an example. The calibrating camera for experiment is the ARTCAM-150PIII CCD camera of ARTRAY Company, Japanese Seiko lens TAMRON 53513, the effective pixel is 1360×1024. To ensure the fairness of comparison, the experiment is carried out on Windows XP system platform, 2.67GHz, 2GB RAM and the Matlab. Unified setting particle population size N is 30, the maximum number of iteration is 1000 and 100 times continuous optimization.

The calibration image used for experiment is the classic black and white chessboard with a size of 8×10, namely 80 corners. The unequal row-column dimension is adopted, which can always determine the direction of chessboard corner detection and ensure the corner correct arrangement. The checkerboard is fixed on a flat plate, which is taken as calibration template. A number of different perspective calibration plate images are collected. As shown in Fig.2.

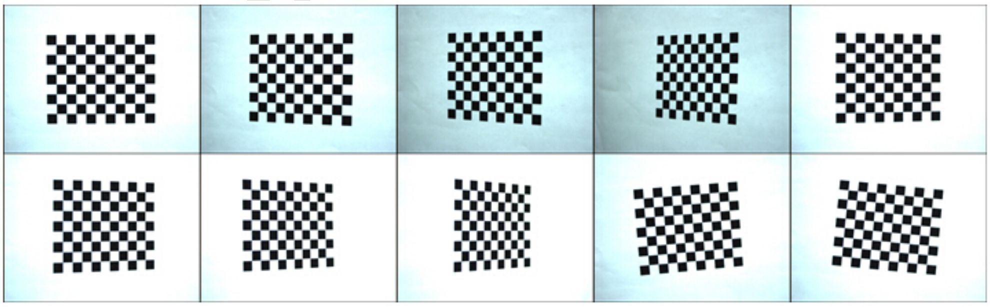

Figure 2: Calibration plate images

The 2D coordinate of image corresponding to each corner is detected by the corner detection algorithm. The all corners are marked in RGB and shown in Fig. 3.

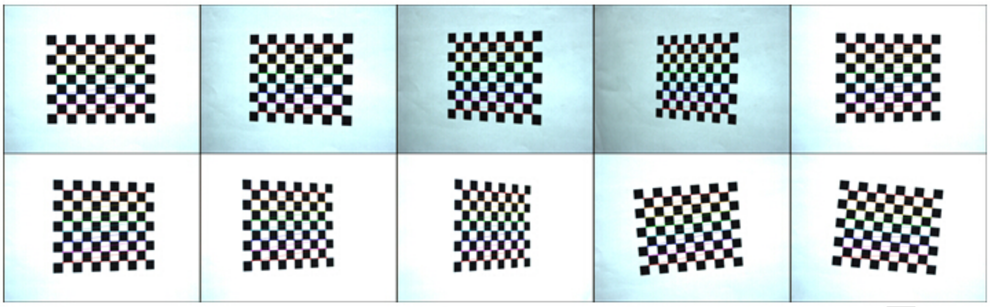

Figure 3: Corner detection results

As can be seen in Fig. 3, all corners are detected in each calibration plate image at the same direction. There are not missing any corners and able to ensure the accuracy of the detection results. Corner detection results provide coordinate information for the camera calibration.

4.2. Experimental results and analysis

To decide the geometric parameters during camera calibration, more than two perspective images are needed. A large number of experiments show that the camera internal parameter (f_x, f_y, c_x, c_y) can convergence and the relative error is small when there are more than three different visual angle images. Fig. 4 shows the calibration errors under different numbers of images.

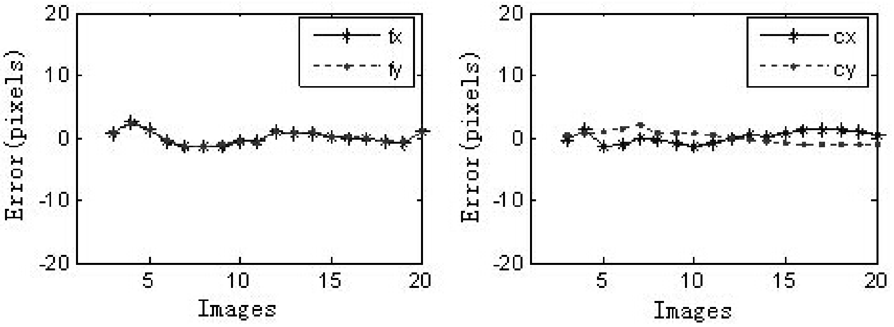

Figure 4: Calibration error under different number of images

In order to ensure the convergence and accuracy of camera parameters, 10 calibration board images at different perspectives are used in this paper, as shown in Fig. 2. The corners are considered as calibration points and the proposed algorithm is used for camera calibration. In view of the different visual angles, the camera external parameters are uncertain. Each perspective image is corresponding to one group of external parameters, which is described by a rotation matrix and translation matrix. Table 1 shows the 10 groups of external parameters obtained by the algorithm.

Table 1: External parameters under different perspective

Groups	External Parameters					
	Rotation Matrix (R)			Translation Matrix (T)		
1	0.005746	0.999937	-0.009672			
	0.999486	-0.006048	-0.031489	-23.724825	2.844623	371.641571
	-0.031546	-0.009486	-0.999457			
2	-0.010711	0.933501	-0.358414			
	0.998677	-0.008045	-0.050797	-20.125275	2.419088	380.607178
	-0.050303	-0.358484	-0.932180			
3	-0.005160	0.850631	-0.525737			
	0.999330	-0.014657	-0.033524	-20.182161	1.308327	385.429962
	-0.036222	-0.525558	-0.849986			
4	-0.020839	0.747789	-0.663609			
	0.997849	-0.025711	-0.060308	-12.392050	3.052024	404.491943
	-0.062160	-0.663439	-0.745644			
5	-0.002098	0.977248	0.212091			
	0.999059	0.011234	-0.041879	-21.713768	1.552888	367.576874
	-0.043309	0.211803	-0.976352			
6	-0.001609	0.930233	0.366967			
	0.998232	0.023301	-0.054689	-18.073584	0.321233	355.749512
	-0.059424	0.366230	-0.928625			
7	0.011253	0.852777	0.522154			
	0.996395	0.034351	-0.077575	-24.403740	0.074992	363.162506
	-0.084091	0.521144	-0.849316			
8	-0.010385	0.732618	0.680561			
	0.997830	0.051847	-0.040587	-14.340678	0.076405	359.779785
	-0.065020	0.678663	-0.731566			
9	0.078206	0.992904	0.089580			
	0.996935	-0.077695	-0.009191	-26.565165	6.383250	382.194092
	-0.002166	0.090024	-0.995937			
10	-0.096142	0.995348	-0.006307			
	0.990626	0.095064	-0.098101	-20.909475	0.394634	388.014008
	-0.097045	-0.015679	-0.995156			

Because of the uncertain camera external parameters, the camera internal parameters and distortion coefficients are calibrated. In order to verify the validity of the algorithm, our method is compared with the planar pattern calibration method by Zhang [11]. Table 2 presents a comparison of the calibration results from different methods.

Table 2: Camera parameter calibration results

Parameters	DEPSO	DE	PSO	Zhang
f_x (pixels)	3580.468994	3585.466215	3588.167245	3588.147217
f_y (pixels)	3620.209961	3624.256194	3627.341156	3628.451172
c_x (pixels)	597.373413	600.623577	600.551365	602.444153
c_y (pixels)	249.571625	251.239451	251.276542	252.145920
k_1 (pixels)	1.246226	1.843260	2.377163	2.549327
k_2 (pixels)	-49.397354	-52.673249	-55.214301	-56.367922
k_3 (pixels)	689.814819	694.130256	698.334672	-
p_1 (pixels)	-0.047936	-0.066324	-0.070329	-
p_2 (pixels)	0.003382	0.004263	0.004430	-

As shown in Table 2, the simulation results are close to the calibration result from Zhang's method and the relative difference is small. It should be noticed that Zhang's calibration method has only considered the first two term coefficients of radial distortion, without fully considering the influence of other distortion. However, the radial and tangential distortions are adequately considered in our method to further reduce the effects of camera lens distortion and obtain the more accurate camera calibration parameters.

According to the model transformation relation, the calibrated camera parameters and the 3D space coordinates (x_w, y_w, z_w) are used to obtain the corresponding 2D image coordinates (u, v), and then compared with the actual image coordinates $(\tilde{u}, \tilde{v})$ obtained by the image processing, so that the validity and accuracy of the calibration method results can be verified. Experiments are based on the visual identification project and 10 groups of the actual measured data are randomly selected. Table 3 presents the verification results.

Table 3: Validity verification results

x_w(mm)	y_w(mm)	z_w(mm)	$\tilde{u}$(pixels)	u(pixels)	$\lvert\tilde{u} - u\rvert$(pixels)	$\tilde{v}$(pixels)	v(pixels)	$\lvert\tilde{v} - v\rvert$(pixels)
-10.5	126	420	563.546832	563.472900	0.073932	484.466282	484.567932	0.101650
-31.5	147	420	368.147888	368.124481	0.023407	276.855286	276.652496	0.202790
10.5	147	420	773.985962	774.065552	0.079590	274.455292	274.467499	0.012207
-17.5	133	420	504.371429	504.474915	0.103486	412.283691	412.303558	0.019867
17.5	140	420	842.426575	842.565552	0.138977	342.306091	342.106873	0.199218
-3.5	119	420	640.172485	640.254761	0.082276	547.503052	547.561218	0.058166
-31.5	112	420	370.515656	370.341034	0.174622	616.466858	616.472778	0.005920
24.5	98	420	912.918518	912.706079	0.212439	748.386597	748.385681	0.000916
10.5	133	420	774.656921	774.763550	0.106629	410.976685	410.816132	0.160553
3.5	105	420	708.007996	708.240516	0.232520	682.816284	682.811279	0.005005

As can be seen in Table 3, the derived image coordinates are close to the actual image coordinates, which shows the good performance of the proposed calibration method. The object average (Ave) and standard deviation (Std) of all images pixel coordinates are calculated and compared with the other calibration results. The arbitrary 10 groups of solutions from one image, all 80 groups of solutions from any one image and all 800 groups of solutions from the whole images are respectively selected to compare and verify the reliability and stability of the proposed calibration algorithm, as shown in Table 4.

Table 4: Reliability verification and comparison

Methods	10 groups		80 groups		800 groups	
	Ave	Std	Ave	Std	Ave	Std
DEPSO (pixels)	0.167186	0.156672	0.192335	0.165329	0.177186	0.162637
DE (pixels)	0.261793	0.235602	0.280495	0.262143	0.265371	0.253308
PSO (pixels)	0.305274	0.282761	0.345165	0.323514	0.331645	0.308450
Zhang (pixels)	0.327475	0.289034	0.407481	0.345590	0.334262	0.317475

As can be seen in Table 4, the object average and standard deviation in the results by the proposed algorithms are less than ones produced by the other methods. It is clear that differential evolution combined with the particle swarm optimization can effectively avoid local optimum. The proposed DEPSO algorithm has a better optimization effect and achieved accurate calibration results and stability. In the experiment of visual identification, the LED board is used as a 3D space object. The board is a 16×16 dot matrix and the distance between any two LED lights is 7mm. One point on the LED board is randomly selected to calibrate the corresponding 2D image coordinates, which is marked in the image. It can verify the practical application of the proposed calibration method. Fig. 5 presents visual identification results.

As shown in Fig. 5, the 3D space coordinate corresponding to the actual 2D image coordinate is shown in Table 3. The calibration result error is very small and the visual identification is very accurate. It is clear that using the proposed DEPSO is suitable for camera calibration, and can achieve good reliability and high accuracy. It has a good effect in practical applications.

5. Conclusions

Camera calibration plays very important role in the computer vision application. This paper investigates the novel method that can be used to

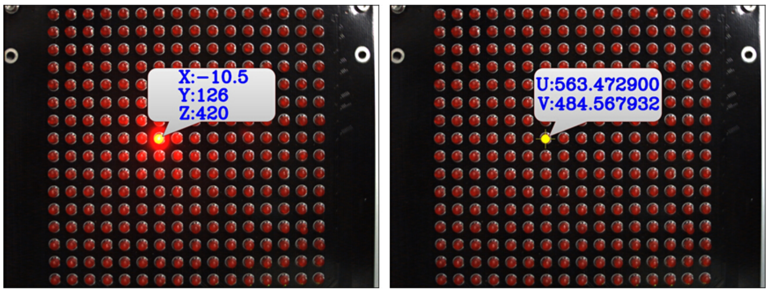

Figure 5: Visual identification results

calibrate camera accurately. Based on actual project and the consideration of the camera distortion effect, the nonlinear relationship model is established for camera calibration and the differential evolution particle swarm optimization is used to calibrate camera parameters. The simulation results show that the algorithm is simple and effective, as well as good optimization ability. It is outperformed the traditional methods in terms of accuracy and robustness in practical applications.

Currently, many methods have been proposed for camera calibration, including the traditional calibration methods and the self-calibration methods. Both of them have limitations, and further improvement is needed. Our future research will be focused on the development of intelligent learning algorithms to solve the camera calibration problems in the diversified real-world applications.

Acknowledgments

This work is supported by the Key Project of Science and Technology Commission of Shanghai Municipality under Grant No. 14JC1402200 and the National Key Scientific Instrument and Equipment Development Project under Grant No. 2012YQ15008703. This work was also supported by the fundamental research project of Shanghai Municipal Science and Technology Commission under grant 12JC1404201, Shanghai College Young Teachers' Training Plan (No. B37010913003) and the Project of Partial Discharge Fault Source Location Visualization Research (No. D11010913033) which was cooperated with State Grid Shanghai Municipal Electric Power Company. The authors would like to thank editors and anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

[1] Y. Xu, D.X. Guo, T.X. Zheng, A.Y. Cheng, Research on camera calibration methods of the machine vision, in: Second International Conference on Mechanic Automation and Control Engineering, 2011, pp. 5150-5153.

[2] J. Li, N.M. Allinson, A comprehensive review of current local features for computer vision, Neurocomputing, 71(10-12)(2008) 1771-1787.

[3] L.M. Song, W.F. Wu, J.R. Guo, X.H. Li, Survey on camera calibration technique, in: Proceedings of fifth International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 2, 2013, pp. 389-392.

[4] J. Sun, H.B. Gu, Research of linear camera calibration based on planar pattern, World Academy of Science, Engineering and Technology, 60(2011) 627-631.

[5] R.V. Carlos, S.S. Antonio-Jose, Optimal conditions for camera calibration using a planar template, in: Proceedings of eighteenth IEEE International Conference on Image Processing, 2011, pp. 853-856.

[6] X.F. Yang, Y.M. Huang, F. Gao, A simple camera calibration method based on sub-pixel corner extraction of the chessboard image, Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems, vol.3, 2010, pp. 688-692.

[7] J.H. Park, S.H. Park, Improvement on Zhang's camera calibration, Applied Science and Precision Engineering Innovation, 479-480(2014) 170-173.

[8] Q. Wang, L. Fu, Z.Z. Liu, Review on camera calibration, in: Chinese Control and Decision Conference, 2010, pp. 3354-3358.

[9] Y.I. Abdel-Aziz, H.M. Karara, Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry, in: Proceeding of the Symposium on Close-Range Photogrammetry, 1971, pp. 1-18.

[10] R.Y. Tsai, An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1986, pp. 364-374.

[11] Z.Y. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11)(2000) 1330-1334.

[12] O.D. Faugeras, Q.T. Luong, S.J. Maybank, Camera self-calibration: theory and experiments, in: Proceedings of Second European Conference on Computer Vision, Lecture Notes in Computer Science, Springer-Verlag, 588(1992) 321-334.

[13] S.D. Ma, A self-calibration technique for active vision systems, IEEE Transactions on Robotics and Automation, 12(1)(1996) 114-120.

[14] J.W. Ma, B.X. Li, A method of camera calibration by iterative optimization algorithm, in: Proceedings of International Conference on Advanced Mechatronic Systems, 2013, pp. 302-305.

[15] M. Merras, N. El Akkad, A. Saaidi, A. Gadhi Nazih, K. Satori, A new method of camera self-calibration with varying intrinsic parameters using an improved genetic algorithm, in: eighth International Conference on Intelligent Systems: Theories and Applications, 2013, pp. 1-8.

[16] T.F. Chen, X. Wu, Z. Ma, Y. Wang, D.F. Wu, Non-metric lens distortion correction using modified particle swarm optimization, International Journal of Modelling, Identification and Control, 21(3)(2014) 330-337.

[17] R. Kouskouridasa, A. Gasteratosa, C. Emmanouilidis, Efficient representation and feature extraction for neural network-based 3D object pose estimation, Neurocomputing, 120(2013) 90-100.

[18] K. Deep, M. Arya, M. Thakur, B. Raman, Stereo camera calibration using particle swarm optimization, Applied Artificial Intelligence, 27(7)(2013) 618-634.

[19] Z. Tian, J.L. Xiong, Q. Zhang, Camera calibration with neural networks, Applied Mechanics and Mechanical Engineering, 29-32(2010) 2762-2767.

[20] Q.S. Li, L.C. Lin, Z.T. Jiang, A Camera Self-calibration Method Based on Hybrid Optimization Algorithm, in: Second International Symposium on Electronic Commerce and Security, 2009, pp. 60-64.

[21] D.Y. Ge, X.F. Yao, M.Q. Yu, Camera calibration by hopfield neural network and simulated annealing algorithm, ICIC Express Letters Office, 4(4)(2012) 1257-1262.

[22] J. Li, Y.M. Yang, G.P. Fu, Camera self-calibration method based on GA-PSO algorithm, in: IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 149-152.

[23] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the fourth IEEE International Conference on Neural Networks, Piscataway: IEEE Service Center, 1995, pp. 1942-1948.

[24] R. Storn, K. Price, Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11(4)(1997) 341-359.

[25] Z.F. Hao, G.H. Guo, H. Huang, A Particle Swarm Optimization Algorithm with Differential Evolution, in: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007, pp. 1031-1035.

[26] F.Q. Zhou, Y. Cui, B. Peng, Y.X. Wang, A novel optimization method of camera parameters used for vision measurement, Optics and Laser Technology, 44(6)(2012) 1840-1849.

[27] P. Swapna, N. Krouglicof, R. Gosine, A novel technique for estimating intrinsic camera parameters in geometric camera calibration, in: Twenty third Canadian Conference on Electrical and Computer Engineering, 2010, pp. 1-7.

[28] Y.X. Zhao, X.P. Lou, N.G. Lv, Camera calibration external parameters amendments in vision measuring, in: Proceedings of SPIE - The International Society for Optical Engineering, Advanced Sensor Systems and Applications, vol. 8561, 2012.

[29] J.G. Fryer, D.C. Brown, Lens distortion for close-range photogrammetry, Photogrammetric Engineering and Remote Sensing, 52(1)(1986) 51-58.

[30] J.H. Wang, F.H. Shi, J. Zhang, Y.C. Liu, A new calibration model of camera lens distortion, Pattern Recognition, 41(2)(2008) 607-615.

[31] J.D. Zhang, J.G. Lu, H.L. Li, M. Xie, Particle swarm optimization algorithm for non-linear camera calibration, International Journal of Innovative Computing and Applications, 4(2)(2012) 92-99.

[32] X.N. Song, B. Yang, Z.Q. Feng, T. Xu, D.L. Zhu, Y. Jiang, Camera calibration based on particle swarm optimization, in: Proceedings of the Second International Congress on Image and Signal Processing, 2009, pp. 1-5.

[33] G.Y. Li, M.G. Liu, The summary of differential evolution algorithm and its improvements, in: Third International Conference on Advanced Computer Theory and Engineering, vol.3, 2010, pp. 153-156.

[34] D.L. Fraga, L. Gerardo, Self-calibration from planes using differential evolution, Lecture Notes in Computer Science, 5856(2009) 724-731.

[35] S.Z. Wan, S.W. Xiong, J.L. Kou, Y. Liu, Differential evolution with improved mutation strategy, Lecture Notes in Computer Science, 6728(1)(2011) 431-438.