RESEARCH ARTICLE OPEN ACCESS

Algorithm of Hand-Eye Calibration for Industrial Robot

Huang Wenji[1], Liu Ning[2]
[1](College of Information&Science, Jinan University, Guangzhou, China)
[2] (College of Information&Science, Jinan University, Guangzhou, China)

----------**********----------

Abstract:

For the eye-to-hand monocular vision system, based on the idea of "black box", which maps the image coordinates directly to the robot reference coordinates, the transform matrix about image coordinate and robot reference coordinate is directly obtained in the condition of fixed distance between the positioning plane and camera. The optimized transform matrix can be obtained as global calibration parameters for the eye-to-hand system, without the need for calculation of camera's internal and external parameters. As the transformation matrix does not consider the non-linear imaging of the lens, the direct use of this transform matrix of the two coordinates system will have a greater accuracy error. In order to reduce the positioning error of hand-eye vision system, this paper considers the obtainment of transform matrix as multiobjective optimization problem and applies the hybrid particle swarm algorithm to optimize the matrix parameters. The experimental results show that this method can effectively improve the hand-eye positioning accuracy and can be successfully applied to industrial production.

Keywords — **industrial robot, hand-eye vision, monocular vision, object position, particle swarm algorithm**

----------**********----------

I. INTRODUCTION

Hand-eye system is divided into eye-in-hand system and eye-to-hand system[1]. The traditional hand-eye positioning system needs to calibrate the inner parameters of the camera first, and then use the P3P space-point position and orientation principle to get the coordinate of the calibration plate relative to the camera[2]. Finally, get the hand-eye conversion relation with the projection matrix of the camera coordinates relative to the manipulator[3-4]. Conventional camera calibration methods[5-7] take enough images of the standard reference object to solve the camera parameters. This method is not suitable for applications that need to be re-calibrated frequently. The camera self-calibration method[8-9] does not require any auxiliary reference and is suitable for situations where the camera parameters are often changed, but it is not robust. The camera's active visual calibration method[10] requires a high-precision auxiliary platform, which requires high external conditions and high cost. The idea of "black box", in which the image coordinates are directly mapped to the robot reference coordinates, is applied to the eye-in-hand system in paper[11], paper[12] and paper[13]. The constant rotation matrix in [11] is greatly affected by the measurement error of the robot coordinate system information. Paper[12] requires rigorous installation of the camera and has a large positioning error. In paper[13], the calibration precision is improved compared with

paper[12], but it is sensitive to the distance from the camera to the calibration point. Paper[13] needs to improve the calibration accuracy by reducing the calibration height. Paper[14] applies the "black box" idea to monocular vision and proposes a fast detection algorithm for hole location in large field of view plane based on linear array CCD.

Based on the idea of "black box", according to the principle of camera linear model, this paper obtains the transform matrix which represents the camera ' s internal parameters and hand-eye relationship as a whole. The transform matrix makes the calibration of the hand-eye system easier. As the linear model does not consider the lens distortion and other non-linear factors, the accuracy of hand-eye calibration is low with the direct use of the transformation matrix. In order to improve the accuracy of calibration, this paper uses the hybrid particle swarm optimization (PSO) to optimize the transformation matrix. Experimental results indicate that the method has high precision.

II. Camera Imaging Model

In this paper, pinhole imaging model is used as the camera imaging model. The exact derivation of pinhole imaging model can be found in paper[15]. The transformation relationship between the world coordinate (X_w, Y_w, Z_w) and the image pixel coordinate (u, v) can be given as

$$Z_c\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & 0 & u_0 & 0 \\ 0 & fs_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$= \boldsymbol{M_1 M_2 X_w} = \boldsymbol{M X_w} \qquad (1)$$

Where f is the focal length of the camera, u_0 and v_0 are the coordinates of the primary point, s_x and s_y represent the number of pixels per unit length, ***M1*** is the camera's internal parameters, ***M2*** is the external parameters, ***M*** is a matrix of 3×4, known as the projection matrix of the image pixel coordinate system to the world coordinate system.

III. The Principle and Derivation of Hand-Eye Coordinate Transformation

The formula of (1) can be expanded as

$$Z_c\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & 0 & u_0 & 0 \\ 0 & fs_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} fs_x & 0 & u_0 & 0 \\ 0 & fs_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} & t_x \\ c_{21} & c_{22} & c_{23} & t_y \\ c_{31} & c_{32} & c_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} \qquad (2)$$

The formula of (2) contains the following 3 equations:

$$\begin{cases} Z_c u = a_{11}x_r + a_{12}y_r + a_{13}z_r + a_{14} \\ Z_c v = a_{21}x_r + a_{22}y_r + a_{23}z_r + a_{24} \\ Z_c = a_{31}x_r + a_{32}y_r + a_{33}z_r + a_{34} \end{cases} \qquad (3)$$

To solve a_{ij}, simplify the formula of (3) as

$$\begin{cases} x_r a_{11} + y_r a_{12} + z_r a_{13} + a_{14} - u a_{31}x_r - u a_{32}y_r - u a_{33}z_r = u a_{34} \\ x_r a_{21} + y_r a_{22} + z_r a_{23} + a_{24} - v a_{31}x_r - v a_{32}y_r - v a_{33}z_r = v a_{34} \end{cases} \qquad (4)$$

For n pairs of image coordinates (u_i, v_i) and robot coordinates (x_{ri}, y_{ri}, z_{ri}), $2n$ linear equations about a_{ij} can be obtained by the formula of (4). The matrix form of the equations is as follows

$$\begin{bmatrix} x_{r1} & y_{r1} & z_{r1} & 1 & 0 & 0 & 0 & 0 & -u_1x_{r1} & -u_1y_{r1} & -u_1z_{r1} \\ 0 & 0 & 0 & 0 & x_{r1} & y_{r1} & z_{r1} & 1 & -v_1x_{r1} & -v_1y_{r1} & -v_1z_{r1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{rn} & y_{rn} & z_{rn} & 1 & 0 & 0 & 0 & 0 & -u_nx_{rn} & -u_ny_{rn} & -u_nz_{rn} \\ 0 & 0 & 0 & 0 & x_{rn} & y_{rn} & z_{rn} & 1 & -v_nx_{rn} & -v_ny_{rn} & -v_nZ_{rn} \end{bmatrix} \times \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} u_1a_{34} \\ v_1a_{34} \\ \cdots \\ \cdots \\ u_na_{34} \\ v_na_{34} \end{bmatrix} \quad (5)$$

The formula of （5）can be expressed as

$$\boldsymbol{Ba}=\boldsymbol{U} \quad (6)$$

where $\boldsymbol{B}$ is $2n\times 11$ matrix,it is the known constant about space point coordinates, $\boldsymbol{a}$ is 11-dimensional vector to be solved. $\boldsymbol{U}$ is 2n-dimensional vector,it is the known constant about image point coordinates. The vector $\boldsymbol{a}$ can be obtained by least squares method as

$$\boldsymbol{a}=(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{U}$$

The formula of (2) can be expressed as

$$Zc\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = A \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} \quad (7)$$

$b_{ij} = a_{ij} / Z_c$, the formula of (8) can be obtained as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = B \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} \quad (8)$$

Expand the formula of (8) as

$$\begin{cases} u = b_{11}x_r + b_{12}y_r + b_{13}z_r + b_{14} \\ v = b_{21}x_r + b_{22}y_r + b_{23}z_r + b_{24} \\ 1 = b_{31}x_r + b_{32}y_r + b_{33}z_r + b_{34} \end{cases} \quad (9)$$

$$\begin{cases} u - b_{14} = b_{11}x_r + b_{12}y_r + b_{13}z_r \\ v - b_{24} = b_{21}x_r + b_{22}y_r + b_{23}z_r \\ 1 - b_{34} = b_{31}x_r + b_{32}y_r + b_{33}z_r \end{cases} \quad (10)$$

The matrix form of the equations （10）is as follows

$$\begin{bmatrix} u - b_{14} \\ v - b_{24} \\ 1 - b_{34} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} \quad (11)$$

The formula of (11) can be expressed as

$$P = MR \quad (12)$$

$$R = M^{-1}P \quad (13)$$

where $\boldsymbol{P}$ and $\boldsymbol{R}$ are 3-dimensional vector, $\boldsymbol{M}$ is 3×3 matrix.

The robot coordinates can be obtained from the image coordinates by the equation (13).

IV. HYBRID PARTICLE SWARM OPTIMIZATION FOR CALIBRATION PARAMETERS

The matrix $\boldsymbol{B}$ obtained in the formula of (8) expresses the hand-eye relationship of ideal camera model.However, due to mathematical model error,

image acquisition error and lens distortion error and other factors, the robot coordinates obtained directly from (13) have a large error compared with the actual coordinates. The formula of (13) can not be directly applied to industrial applications. The matrix element b_{ij} in (8) needs to be optimized to improve the accuracy of the hand-eye system. The optimization of b_{ij} is actually a multi-objective optimization problem. Particle swarm optimization (PSO) has better ability to solve multi-objective optimization problems. To enhance the global search capability, this paper combine PSO with genetic algorithm to optimize the calibration parameters b_{ij}.

A. Standard Particle Swarm Optimization

Particle Swarm Optimization is a group intelligent optimization algorithm[16]. Each particle position in the PSO algorithm represents a potential solution to the problem. The position and velocity of each particle are dynamically adjusted according to the movement experience of the other particles in the population and the population, so as to realize the optimization of each particle in the solvable space.

For a population with n particles, the dimension of each particle is D. In the k iteration, the particles ' velocity V_{id} and position X_{id} are adjusted by their individual extremes P_{id} and the population extremes P_{gd} in the following way[17]

$$V_{id}(k+1) = wV_{id}(k) + c_1 r_1 (P_{id}(k) - X_{id}(k)) + c_2 r_2 (P_{gd}(k) - X_{id}(k)) \quad (14)$$

$$X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \quad (15)$$

where w is inertia weight, k is the number of iterations, c_1 and c_2 are acceleration factors, r_1 and r_2 are random numbers.

B. Hybrid Particle Swarm Optimization

With the number of iterations increases, it is easy to converge to the local optimal solution for the standard particle swarm algorithm. This paper combines PSO with genetic algorithm. The crossover operator and the mutation operator are introduced to extend search space.

1) Individual Encoding

The individual encoding of particles is an integer encoding. Each particle represents the 12 elements in the transformation matrix $\boldsymbol{B}$. The individual encoding is expressed as [b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34].

2) Fitness Value

The particle fitness value represents the re-projection error between the robot coordinate and the actual robot coordinate calculated from the image coordinates. The fitness value in this paper is calculated by following formula

$$\text{Fitness(i)} = \left\| P_w - P_w' \right\|_2 \quad (16)$$

where P_w is the actual robot coordinate vector, P_w' is the robot coordinate vector calculated from the calibration parameters b_{ij}.

3) Cross Operation

particles are updated by intersecting individual extremes and population extremes. First select the first and last position of the intersection, and then the intermediate individual coding is crossed with the individual extremes and the population extremes. The new individuals were retained with the best individual strategy.

4) Mutation Operation

Mutation operation requires randomly generating particles by normal distribution. The new individuals were retained with the best individual strategy.

The steps of hybrid particle swarm algorithm is as follows:

- The population is initialized with a random distribution of normal distribution. The mean is matrix $\boldsymbol{B}$ element and the variance is $\sqrt{2}$.
- Calculate the fitness value of the particles according to (16).
- According to the particle fitness value, the optimal particle and the population optimal particle are updated.
- According to (14) and (15) to update the particle velocity and position.
- The individual and individual optimal particles are crossed to obtain new particles.
- The individual and population optimal particles are crossed to obtain new particles.
- Particle mutation operation.
- Check the end condition. If the condition is satisfied, the optimization is terminated.

V. Experimental Results and Analysis

1) Hand-Eye System Calibration Steps

Calibration prototype system is shown in Fig.1. The industrial camera is mounted above the horizontal conveyor belt. Software interface is shown in Fig.2. The Calibration target is a long strip of metal with a solid circle. The calibration target is fixed at the end of the z-axis of the robot. Fix the robot z-axis height and rotation angle, the robot moves n(2n>11) times on the xy plane. Robot coordinates (x_{ri}, y_{ri}, z_{ri}) and image coordinates (u_i, v_i) can be obtained. Then the calibration parameters can be calculated by (13) and optimized by hybrid particle swarm optimization.

Fig.1 Calibration prototype system

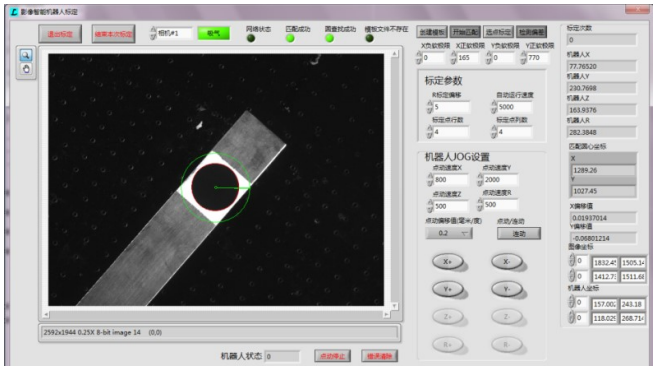

Fig.2 Calibration Software interface

2) Positioning Accuracy Comparison

The number of calibration points is 16 pairs. The parameters of particle swarm optimization are c1 = 1.4801, c2 = 1.4801, w = 0.7153, the number of population is 200 and the number of population evolution is 150. Schematic diagram of calibration target movement is shown in Fig.3. Choose 16 of them as the data points to get the calibration parameters and the remaining 9 points are used as check points to calculate the re-projection error ε . The point pairs data is shown in Table 1.

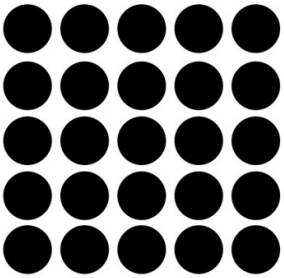

Fig.3 Schematic diagram of calibration target movement

TABLE1 POINT PAIRS DATA

Ordinal	Image coordinates of robot end	World coordinates of robot end	Ordinal	Image coordinates of robot end	World coordinates of robot end
0	(441.615，1362.01)	(11.7365,652.65,167.4166)	8	(718.591,649.72)	(26.1133,595.401,167.4166)
1	(667.076,1204.35)	(4.5699,632.65,167.4166)	9	(907.464,664.613)	(13.5498,587.968,167.4166)
2	(470.817,1119.82)	(20.7165,636.045,167.4166)	10	(1095.94,680.163)	(0.9865,580.535,167.4166)
3	(747.085,1024.5)	(7.5633,617.756,167.4166)	11	(347.302,344.585)	(63.2133,592.808,167.4166)
4	(337.659,892.748)	(39.2666,627.729,167.4166)	12	(560.333,391.44)	(47.6566,586.23,167.4166)
5	(663.593,861.039)	(20.1265,611.151,167.4166)	13	(773.913,438.103)	(32.1,579.652,167.4166)
6	(990.63,828.472)	(0.9866,594.573,167.4166)	14	(987.861,484.456)	(16.5433,573.074,167.4166)
7	(530.185,634.38)	(38.6766,602.835,167.4166)	15	(1201.54,531.46)	(0.9866,566.496,167.4166)

TABLE2 SIMULATION RESULTS ON PAPER[19](UNIT：MILLIMETER)

Ordinal	Image coordinates of robot end	Actual world coordinates of robot end	Calculated world coordinates of robot end	Reprojection error ε
0	(329.150,1440.610)	(15.3199,662.6500)	(15.2997,662.0002)	(-0.0202,-0.6498)
1	(554.166,1283.440)	(8.1532,642.6500)	(8.1482,642.4887)	(-0.0050,-0.1623)
2	(780.161,1125.060)	(0.9866,622.6500)	(0.9891,622.8573)	(0.0025,0.2073)
3	(333.206,1167.010)	(27.2932,645.1890)	(27.2746,645.7261)	(-0.0186,-0.4629)
4	(608.893,1072.190)	(14.1398,626.9000)	(14.1434,626.9283)	(0.0036,0.0283)
5	(885.365,976.862)	(0.9866,608.6110)	(0.9854,609.0649)	(-0.0012,0.4539)
6	(500.655,876.818)	(29.6966,619.4400)	(29.6966,619.3789)	(0.0000,-0.0611)
7	(827.171,844.685)	(10.5566,602.8620)	(10.5606,603.3101)	(0.0040,0.4481)
8	(342.178,618.781)	(51.2399,610.2680)	(51.2163,610.0765)	(-0.0236,-0.1915)

TABLE3 UNOPTIMIZED CALIBRATION PARAMETER SIMULATION RESULTS(UNIT：MILLIMETER)

Ordinal	Image coordinates of robot end	Actual world coordinates of robot end	Calculated world coordinates of robot end	Reprojection error $\mathcal{E}$
0	(329.150,1440.610)	(15.3199,662.6500,167.4166)	(15.5594,662.3742,167.4166)	(0.2395,-0.2758,0)
1	(554.166,1283.440)	(8.1532,642.6500,167.4166)	(8.2296,642.5941,167.4166)	(0.0764,-0.0559,0)
2	(780.161,1125.060)	(0.9866,622.6500,167.4166)	(0.8913,622.6950,167.4166)	(-0.0953,0.0450,0)
3	(333.206,1167.010)	(27.2932,645.1890,167.4166)	(27.4709,645.0160,167.4166)	(0.1777,-0.1730,0)
4	(608.893,1072.190)	(14.1398,626.9000,167.4166)	(14.1429,626.9190,167.4166)	(0.0031,0.0190,0)
5	(885.365,976.862)	(0.9866,608.6110,167.4166)	(0.7876,608.7556,167.4166)	(-0.1990,0.1446,0)
6	(500.655,876.818)	(29.6966,619.4400,167.4166)	(29.7217,619.4190,167.4166)	(0.0251,-0.0210,0)
7	(827.171,844.685)	(10.5566,602.8620,167.4166)	(10.3705,603.0194,167.4166)	(-0.1861,0.1574,0)
8	(342.178,618.781)	(51.2399,610.2680,167.4166)	(51.2851,610.1970,167.4166)	(0.0452,-0.0710,0)

TABLE4 SIMULATION RESULTS ON PROPOSED METHOD(UNIT：MILLIMETER)

Ordinal	Image coordinates of robot end	Actual world coordinates of robot end	Calculated world coordinates of robot end	Reprojection error $\mathcal{E}$
0	(329.150,1440.610)	(15.3199,662.6500,167.4166)	(15.3502,662.6016,167.4166)	(0.0303,-0.0484,0)
1	(554.166,1283.440)	(8.1532,642.6500,167.4166)	(8.1841,642.6694,167.4166)	(0.0309,0.0194,0)
2	(780.161,1125.060)	(0.9866,622.6500,167.4166)	(1.0103,622.6173,167.4166)	(0.0237,-0.0327,0)
3	(333.206,1167.010)	(27.2932,645.1890,167.4166)	(27.3015,645.1882,167.4166)	(0.0083,-0.0008,0)
4	(608.893,1072.190)	(14.1398,626.9000,167.4166)	(14.1608,626.9238,167.4166)	(0.0210,0.0238,0)
5	(885.365,976.862)	(0.9866,608.6110,167.4166)	(0.9933,608.5923,167.4166)	(0.0007,-0.0187,0)
6	(500.655,876.818)	(29.6966,619.4400,167.4166)	(29.6975,619.4445,167.4166)	(0.009,-0.0045,0)
7	(827.171,844.685)	(10.5566,602.8620,167.4166)	(10.5572,602.8620,167.4166)	(0.0006,-0.0000,0)
8	(342.178,618.781)	(51.2399,610.2680,167.4166)	(51.1956,610.2583,167.4166)	(-0.0443,-0.0097,0)

The convergence curve of the hybrid particle swarm optimization algorithm is shown in Fig.4. From the curve of the trend of change can be seen, when the number of iterations is less than 80, the fitness decreases significantly with the increase of the number of iterations. When the number of iterations is around 150, the fitness value decreases to 0.1, and then the fitness value tends to be stable. Therefore, this paper takes 150 as the hybrid particle swarm optimization algorithm of population evolution.

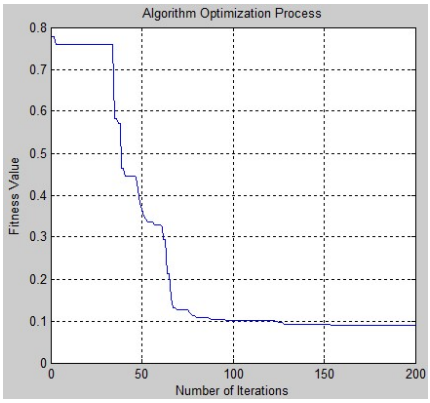

Fig.4 Hybrid particle swarm optimization convergence curve

The test points data are simulated by paper [19], unoptimized calibration parameter algorithm and proposed method respectively. The simulation results are shown in Table 2, Table 3 and Table 4.

Table 2 shows the simulation results of paper [19]. The range of re-projection error is $[-0.0236,0.0040]$ in the x-direction and $[-0.6498,0.4539]$ in the y-direction.

Table 3 shows the simulation results of unoptimized calibration parameter algorithm. The range of re-projection error is [-0.1990,0.2395] in the x-direction and [-0.2758,0.1574] in the y-direction.

Table 4 shows the simulation results of proposed method. It can be seen that the error between the actual robot coordinates and the calculated robot coordinates does not exceed 0.0898. The range of re-projection error is [-0.0443,0.0309] in the x-direction and [-0.0484,0.0238] in the y-direction.

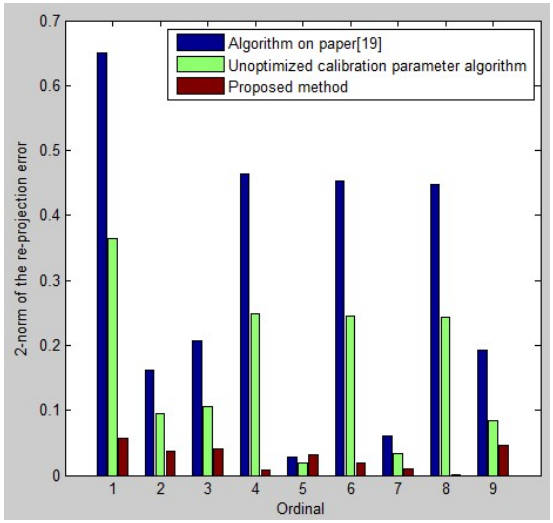

Fig.5 2-norm of the re-projection error on three methods

Take 2-norm of the re-projection error to represent the precision error. The errors of the three methods are shown in Fig.5. It shows that the proposed method has high accuracy and stability compared with the other two methods. By comparing the unoptimized calibration parameter algorithm and proposed method, it can be seen that the hybrid particle swarm optimization algorithm can effectively optimize the calibration parameters and reduce the calibration error.

VI. CONCLUSION AND PROSPECT

In this paper, a calibration algorithm of eye-to-hand monocular vision system is proposed for industrial application. The validity and accuracy of proposed method are verified by experiments. Compared with other methods, the proposed method does not need to calculated the camera's internal and external parameters. The calibration target is easy to obtain and the calibration process is simple. The proposed method has been applied to the industrial field and has a good practicality. The next step is to apply the idea of hybrid particle swarm optimization to the calibration parameters in binocular vision, so that the proposed method can be further extended.

REFERENCES

[1] Hager G D,Hutchinson S,Corke P I. A tutorial on visual servo control [J].IEEE Transaction on Robotics and Automation, 1996, 12(5): 651-670.

[2] Cui J, Huo J, Yang M, et al. Research on the rigid body posture measurement using monocular vision by coplanar feature points[J]. Optik-International Journal for Light and Electron Optics, 2015, 126(24): 5423-5429.

[3] Xiaoping Hu, Fuyong Zuo, Ke Xie. Research on hand—eye calibration method for micro-assembly robot[J]. Chinese Journal of Scientific Instrument, 2012, 33(7):1521-1525.

[4] Xiaoping Hu, Fuyong Zuo, Ke Xie. Research on Hand-eye Calibration Method of Assembly Robot Based on P3P Principle[J]. Mechanical Science and Technology for Aerospace Engineering, 2013, 22(5):762-765.

[5] Yuting Lu. Research on Robot Calibration Based on Vision[D]. Guangzhou: South China University of Technology, 2013.

[6] Kan Huang. Research and application on the algorithms of camera calibration on computer stereo vision[D]. Shenyang: Shenyang University of Technology, 2007.

[7] Tsai R Y. An efficient and accurate camera calibration technique for 3D machine vision[C]//Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986.

[8] Song Xu, Xiuxia Sun, Xi Liu, et al. Geometry Method of Camera Self-Calibration Based on a Rectangle[J]. Acta Optica Sinica, 2014, 34(11):1115002-7 - 1115002-11.

[9] Elamsy T, Habed A, Boufama B. Self-calibration of stationary non-rotating zooming cameras[J]. Image and Vision Computing, 2014, 32(3): 212-226.

[10] Guangwen J, Zhichao C, Sihua F. High-accurate camera calibration technique based on controllable rotation[J]. Acta Optica Sinica, 2010, 30(5): 1308-1314.

[11] Chungshan Xiong, Xinhan Huang, Min Wang. Algorithm for hand-eye stereo vision and implementation[J]. Robot, 2001, 23(2):113-117.

[12] Shiqiang Yang,Weiping Fu, Hongtao Wang. Research on calibration for hand-eye stereo vision[J]. Engineering and Applications, 2007, 43(33):196-199.

[13] Haixia Wang, Xiangru Li, Xuecheng Su, et al. Applied positioning method for hand-eye vision system[J]. Computer Engineering and Applications, 2007,43(24):235-238.

[14] Shi Y, Sun C, Wang P, et al. High-speed measurement algorithm for the position of holes in a large plane[J]. Optics and Lasers in Engineering, 2012, 50(12): 1828-1835.

[15] Deng L, Lu G, Shao Y, et al. A novel camera calibration technique based on differential evolution particle swarm optimization algorithm[J]. Neurocomputing, 2016, 174: 456-465.

[16] Sun Y, Zhang L, Gu X. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems[J]. Neurocomputing, 2012, 98: 76-89.

[17] Mahi M, Baykan Ö K, Kodaz H. A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem[J]. Applied Soft Computing, 2015, 30: 484-490.

[18] Ling Chen. Research on time optimal TSP based on Hybrid PSO-GA[D]. Hefei: Hefei University of Technology,2015.

[19] Hongjuan Yin. Mobile robot vision location research based on improved camera calibration[D]. Qinhuangdao: Yanshan University, 2012.