An Adaptive Selection of Motion for Online Hand-Eye Calibration

Jing Zhang, Fanhuai Shi, and Yuncai Liu

Inst. Image Processing and Pattern Recognition,
Shanghai Jiao Tong University, Shanghai 200240, P.R. China
{zhjseraph, fhshi, whomliu}@sjtu.edu.cn

Abstract. As the robot makes unplanned movement, online hand-eye calibration determines the relative pose between the robot gripper/end-effector and the sensors mounted on it. With noisy measurements, hand-eye calibration is sensitive to small rotations in real applications. Moreover, degenerate cases such as pure translations have no effect in hand-eye calibration. This paper proposes an adaptive motion selection algorithm for online hand-eye calibration, which can adaptively set the thresholds of motion selection according to the characteristics of the unplanned motion sequence. It is achieved by using polynomial-regression to predict the relationship between RMS of calibration error and thresholds. Thus, this procedure leads to an adaptive method of motion selection. It can adapt itself to online hand-eye calibration in various applications. Experiments using simulated data are conducted and present good results. Experiments using real scenes also show that the method is promising.

1 Introduction

The goal of this paper is to improve the accuracy of online hand-eye calibration. We focus on the algorithm of adaptively setting the thresholds of "motion selection"[11], which is a method to exclude degenerate motions and small rotations during online hand-eye calibration. It is achieved by selecting the motions until the selected satisfy the given thresholds. Using this method, one can get "good" motions for online hand-eye calibration. The aim of this paper is to refine the method and give an adaptive algorithm to set the thresholds.

The calibration of robotic hand-eye relationship is a classical problem in robotics. Algebraically, the problem is known to lead to a linear homogeneous equation in the unknown pose matrix X , namely $A \cdot X = X \cdot B$ [1]-[7], where A is the rigid motion of the robot gripper and B is the corresponding camera motion. A, B and X are all homogeneous transformation 4×4 matrices. Because most hand-eye calibration methods are iterative, they are not fit for online computation. Andreff et al.[9][10] and Angeles et al.[8] first introduce the method of online implementation of hand-eye calibration. Whichever method is used, 2 motions with non-parallel rotation axes are necessary to determine the hand-eye transformation. The algebraically and geometrically analysis on hand-eye calibration can be seen in [2][3]. But when we make online hand-eye calibration,

S. Zhang and R. Jarvis (Eds.): AI 2005, LNAI 3809, pp. 520–529, 2005.

we cannot know the movements of the robot beforehand. And also, we cannot do motion plan in advance. So, there are maybe degenerate cases in the motion sequence, which can ruin the result. To solve this problem, Shi et al. [11] first introduced the concept of "motion selection", which try to select the "effective" motions for hand-eye calibration and has greatly decreased the error of online hand-eye calibration.

In [11], Shi et al. make motion selection according to the following observations in [2]:

> *Observation* 1: The RMS (root mean square) error of rotation from gripper to camera is inversely proportional to the sine of the angle α between the interstation rotation axes;
> *Observation* 2: The rotation and translation error are both inversely proportional to the interstation rotation angle β;
> *Observation* 4:The distance between the robot gripper coordinate centers at different stations d is also a critical factor in forming the error of translation.

If one motion fulfill the three constrain, the algorithm will regard it as an "effective" one and do calibration. Otherwise, the algorithm will combine it with its following motion until the synthesized movement satisfies the thresholds. Then they use those effective movements to do calibration. Using this method, not only can one avoid the degenerate cases, but also the small rotations to decrease the calibration error. However, the algorithm in [11] sets the thresholds of α, β, d by experience, which may not adapt to the requirements of different applications.

In this paper, we propose an adaptive algorithm to set the thresholds of motion selection in online hand-eye calibration. The remainder of this paper decomposes as follows. Section 2 describes the objective problem. Then, detailed algorithm of adaptive motion selection for online hand-eye calibration is presented in Section 3. Section 4 conducts some simulated and real experiments to validate the proposed algorithm.

2 Problem Formulation

In this section, we first give a simple description of the algorithm in Shi[11], which is the foundation of our method. Then we describe the problem we attempt to solve.

We use upper-case boldface letters for matrices, e.g. X , and lower-case boldface letters for 3-D vectors, e.g. x. The angle between two vectors is denoted by $\angle(x, y)$. The $\|\cdot\|$ means the Frobenius norm of a vector or a matrix. Rigid transformation is represented with a 4×4 homogeneous matrix X , which is often referred to as the couple (R, t) .At the i-th measurement, the camera pose with respect to reference object is denoted by homogeneous matrix P_i, and the recorded gripper pose relative to robot base is homogeneous matrix Q_i.

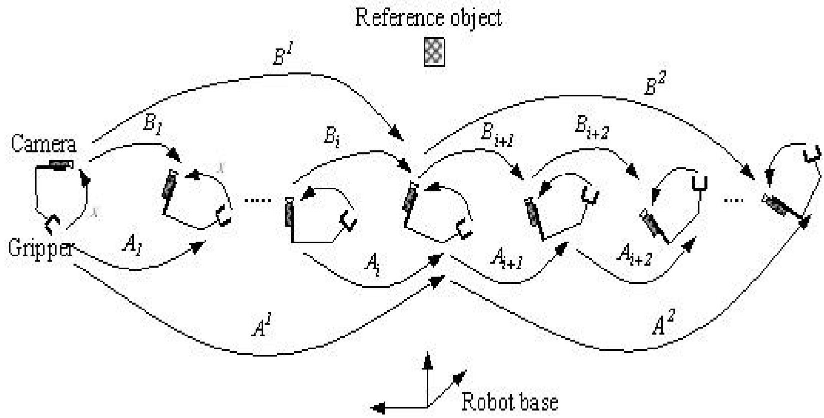

Fig. 1. Algorithm of motion selection for online hand-eye calibration

The usual way to describe the hand-eye calibration is by means of homogeneous transformation matrices. We denote the transformation from gripper to camera by $X = (R_x, t_x)$,the i-th motion matrix of the gripper by $A_i = (R_{a,i}, t_{a,i})$,and the i-th motion matrix of the camera by $B_i = (R_{b,i}, t_{b,i})$. As a rotation matrix R can be expressed as a rotation around a rotation axis k by an angle , the relations between θ, k and R are given by Rodrigues theorem[13]. Moreover,R_a and R_b have the same angle of rotation [1]. We can rewrite R_a and R_b as $Rot(k_a, \theta)$ and $Rot(k_b, \theta)$ respectively. Motion selection is to sequentially find the pairs of consecutive motions(A_i, B_i) and (A_{i+1}, B_{i+1}) for hand-eye computation from sampled motion series[11].

In Ref. [11], the following golden rules are used for motion selection:

Rule 1: Try to make $\angle(k_{a,i}, k_{a,i+1})$ (which is equal to $\angle(k_{b,i}, k_{b,i+1})$ [2]) large, the minimal threshold is set to be α_0 ;
Rule 2: Try to make θ_i large, the minimal threshold is β_i ;
Rule 3: Try to make $\|t_{a,i}\|$ small, the maximal threshold is d_0 .

The i-th sampled hand-eye pose and motion are denoted by (P_i, Q_i) and (A_i, B_i) respectively in this section.(A', B') and (A'', B'') are selected motion pairs for calibration (see Fig. 1). For A' and A'', the rotation axis, rotation angle and translation are denoted by (k'_a, θ'_a, t'_a) , $(k''_a, \theta''_a, t''_a)$ respectively.

At the beginning of the calibration process, they need to estimate (A', B'). The (A', B') is first recovered from (P_1, Q_1) and (P_2, Q_2). If $\theta' \geq \beta_0$ and $\|t'_a\| \leq d_0$, they claim that the (A', B') has been found. Or else, they continue to compute (A', B') from (P_1, Q_1) and(P_3, Q_3) and judge the value θ' and $\|t'_a\|$ in the same way as before. Repeat this procedure until θ' and $\|t'_a\|$ fulfill the given conditions. Here, they assume that the first (A', B') is estimated from (P_1, Q_1) and(P_i, Q_i). After (A', B') has been found, another motion pair (A'', B'') can be sought starting from (P_i, Q_i) and (P_{i+1}, Q_{i+1}) in the similar way as that of

(A', B'), but the constrained conditions are changed to be $\theta' \geq \beta_0$ and $\|t'_a\| \leq d_0$ and $\angle(k'_a, k''_a) \leq \alpha_0$. When both motion pairs are found, they can make one calibration using the method of Andreff [10].

In the next calibration, they take the last motion pair (A'', B'') as the new motion pair (A', B'), and then continue to seek for new (A'', B'') from the successive sampled series and make a new hand-eye calibration in the same way as before.

The above is the motion selection procedure in [11]. But α_0, β_0 and d_0 in this algorithm are set by experience, which is not robust to different situations, as the movements of robot gripper vary with applications. So, in order to improve the robustness and accuracy of the method of online hand-eye calibration by using motion selection, we should set the threshold adaptively to different applications. For this aim, we propose an adaptive motion selection algorithm to do online hand-eye calibration.

3 Adaptive Selection of Motion for Online Hand-Eye Calibration

3.1 Main Algorithm

In our algorithm, we use $sin(\alpha)$ instead of α as the threshold of sine of the angle between the interstation rotation axes, which is derived from *observation 1* in [2]. At the beginning of the process, we just set the average value of $sin(\alpha), \beta, d$ of the first N movements as the initial thresholds.

$$(sin(\alpha_0), \beta_0, d_0) = \frac{1}{N}(\sum_{n=1}^{N} sin(\alpha_n), \sum_{n=1}^{N} \beta_n, \sum_{n=1}^{N} d_n) \tag{1}$$

One can set N according to the application. Such as, if you can predict that the motion number will be large, N may be set larger than 5. Then the initial threshold may reflect the characteristics of the motion sequence better. Otherwise, N may be set smaller than 5 to fulfill the requirement of threshold prediction. But as our algorithm is adaptive according to local characteristic, the influence of the value of N is not remarkable.Then, we use the initial thresholds to do motion selection and calibration for five times. In each calibration, we will compute the RMS of rotation and translation and modifying the thresholds by multiplying $sin(\alpha_0), \beta_0$ a parameter larger than 1 and multiplying d_0 a parameter less than 1. During this process, if we cannot do one calibration for more than certain number motions, such as five in our algorithm, we will simply reduce $sin(\alpha_0), \beta_0$ and increase d_0 by multiplying parameters.

After we have done five calibrations, we begin to adaptively set the thresholds of $(sin(\alpha), \beta, d)$ using polynomial-regression method. In the subsequent calibration, we take the last motion pair (A'', B'') as the new motion pair (A', B') , and then continue to seek for new (A'', B'') from the successive sampled series using the new thresholds. And make a new hand-eye calibration in the same way as before.

The corresponding algorithm is as follows:

Main Algorithm

1. $N \leftarrow 5$,Compute the initial thresholds of $sin(\alpha_0), \beta_0, d_0$, using Eq(1);
2. $i \leftarrow N+1$, $start \leftarrow i$, $interval \leftarrow 5$, $calibNo \leftarrow 0$;
3. $A' = Q_5^{-1}Q_i$, $B' = P_5^{-1}P_i$;
4. Compute θ' and t'_a from A';
5. if $\theta \geq \beta_0$ and $\|t'_a\| \leq d_0$, then go to 8;
6. if $i - start \leq interval$, then $i \leftarrow i+1$, go to 3; (Sample one more motion)
7. if $\theta' < \beta_0$, then $\beta_0 \leftarrow \beta_0 \times 0.8$, if $\|t'_a\| > d_0$, then $d_0 \leftarrow d_0 \times 1.2$, $i \leftarrow i+1, start \leftarrow i$,goto 3;
8. $j \leftarrow i+1, start \leftarrow j$ (Begin to search forA'');
9. $A' = Q_i^{-1}Q_j$, $B' = P_i^{-1}P_j$;
10. Compute $\angle(k'_a, k''_a), \theta''$ and $t''_a$from A' and A'';
11. if $sin(\angle(k'_a, k''_a)) \geq sin(\alpha_0)$ and $\theta' \geq \beta_0$ and $\|t''_a\| \leq d_0$, then go to 14;
12. if $j - start \leq interval$, then $j \leftarrow j+1$, go to 9; (Sample one more motion)
13. if $sin(\angle(k'_a, k''_a)) \leq sin(\alpha_0)$,then $sin(\alpha_0) \leftarrow sin(\alpha_0) \times 0.8$
    if $\theta' \leq \beta_0$, then $\beta_0 \leftarrow \beta_0 \times 0.8$
    if $\|t''_a\| \geq d_0$,then$d_0 \leftarrow d_0 \times 1.2$
    $j \leftarrow j+1, start \leftarrow j$, go to 9;
14. Make one hand-eye calibration using the method in Andreff[10], compute RMS of rotation and translation, save the value of $(sin(\alpha_0), \beta_0, d_0)$,$calibNo \leftarrow caliNo + 1$;
15. if $calibNo \geq 5$, then use *AlgorithmI,II,III* respectively to predict the next set of thresholds;Else $sin(\alpha_0) \leftarrow sin(\alpha_0) \times 1.2$,$\beta_0 \leftarrow \beta_0 \times 1.2$,$d_0 \leftarrow d_0 \times 0.8$;
16. $A' \leftarrow A'', B' \leftarrow B''$;
17. $i \leftarrow j, j \leftarrow j+1$, go to 9 for next calibration.

3.2 The Algorithm to Adaptively Set the Thresholds

From the observations in [2] we can see that $(sin(\alpha_0), \beta_0, d_0)$ affect the accuracy of hand-eye calibration. So we use polynomial-regression [12] to simulate the relationship between $(sin(\alpha_0), \beta_0, d_0)$ and RMS of rotation and translation. In the following paragraph, we will represent RMS of rotation error as $rmsR$ and RMS of translation error as $rmsT$. Because the polynomial-regression needs at least four sets of data to insure the reliability, we do five calibrations at first. We normalized the five sets of $rmsR$ and $rmsT$. They are divided by the maximum value of them respectively. The mathematical model to evaluate the error and thresholds is given by polynomial regression model [12] as depicted by the following equation:

$$y = b_0 + b_1 \times x + b_2 \times x^2 + b_3 \times x^3 \tag{2}$$

where x is $rmsR$ or $rmsT$, y is threshold, b_0, b_1, b_2, b_3 is constants to be determined.

The anterior four sets of normalized $rmsR$ and $rmsT$ and four sets of $(sin(\alpha_0), \beta_0, d_0)$ are used to compute the constants of polynomial-regression.

After we have got three cubic curves of $(sin(\alpha_0), \beta_0, d_0)$ respectively, we use the fifth set of normalized $rmsR$ and $rmsT$ to predict the new value of $(sin(\alpha_0), \beta_0, d_0)$. Using the new thresholds, we will do a next calibration in the same way as before.

To fulfill the requirement of least-square,b_1, b_2, b_3 must satisfy the following equations:

$$\begin{cases} L_{11} \times b_1 + L_{12} \times b_2 + L_{13} \times b_3 = L_{10} \\ L_{21} \times b_1 + L_{22} \times b_2 + L_{23} \times b_3 = L_{20} \\ L_{31} \times b_1 + L_{32} \times b_2 + L_{33} \times b_3 = L_{30} \end{cases} \tag{3}$$

where
$L_{11} = \sum(x - \bar{x}_1)^2, L_{12} = L_{21} = \sum(x - \bar{x}_1)(x^2 - \bar{x}_2), L_{10} = \sum(x - \bar{x})(y - \bar{y}),$
$L_{22} = \sum(x^2 - \bar{x}_2)^2, L_{13} = L_{31} = \sum(x - \bar{x}_1)(x^3 - \bar{x}_3), L_{20} = \sum(x^2 - \bar{x}^2)(y - \bar{y}),$
$L_{33} = \sum(x^3 - \bar{x}_3)^2, L_{23} = L_{32} = \sum(x^2 - \bar{x}_2)(x^3 - \bar{x}_3), L_{30} = \sum(x^3 - \bar{x}_3)(y - \bar{y}),$
$\bar{x}_1 = \frac{1}{4}\sum x, \bar{x}_2 = \frac{1}{4}\sum x^2, \bar{x}_3 = \frac{1}{4}\sum x^3, \bar{y} = \frac{1}{4}\sum y,$
$b_0 = \bar{y} - b_1 \times \bar{x}_1 - b_2 \times \bar{x}_2 - b_3 \times \bar{x}_3$

The corresponding algorithm is as follows. In these algorithms, $alfaMin$, $betaMin$ are the minimum value that $sin(\alpha_0), \beta_0$ should satisfy respectively, $dMax$ is the maximum value that d_0 should satisfy. One can estimate them using *Observation 1,2,4* in [2] according to the requirement of application.

By the detailed analysis on *Observation 1* in [2], we note that $sin(\alpha_0)$ affecting the RMS of rotation error. So we use $rmsR$ to predict $sin(\alpha_0)$.

Algorithm I

1. Using the four normalized $rmsR$ as x , anterior four $sin(\alpha_0)$ as y to compute the constants b_0, b_1, b_2, b_3 of the cubic curve of $sin(\alpha_0)$, that is

$$\begin{bmatrix} 1 \; rmsR_1 \; rmsR_1^2 \; rmsR_1^3 \\ 1 \; rmsR_2 \; rmsR_2^2 \; rmsR_2^3 \\ 1 \; rmsR_3 \; rmsR_3^2 \; rmsR_3^3 \\ 1 \; rmsR_4 \; rmsR_4^2 \; rmsR_4^3 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} sin(\alpha_0)_1 \\ sin(\alpha_0)_2 \\ sin(\alpha_0)_3 \\ sin(\alpha_0)_4 \end{bmatrix} \tag{4}$$

2. Using the fifth normalized $rmsR$ as x_{new} to compute the new value of $sin(\alpha_0)$, that is

$$sin(\alpha_0)_{new} = b_0 + b_1 \times rmsR_5 + b_2 \times rmsR_5^2 + b_3 \times rmsR_5^3 \tag{5}$$

3. $if sin(\alpha_0)_{new} \leq 0, then\ sin(\alpha_0) = abs(sin(\alpha_0)_{new})$
$if sin(\alpha_0)_{new} \leq alfaMin, then\ sin(\alpha_0) = alfaMin$
$if sin(\alpha_0)_{new} \geq 1, then\ sin(\alpha_0) = 0.9$

By the detailed analysis on *Observation 2* in [2], we note that $beta_0$ affecting both rotation and translation error. So we use $rmsR$ and $rmsT$ to predict $beta_0$.

Algorithm II

1. Using the four normalized $rmsR$ as x , anterior four β_0 as y to compute the constants b_0, b_1, b_2, b_3 of the cubic curve of β_0 ;
2. Using the fifth normalized$rmsR$ as x_{new} to compute the new value $\beta_1 = y_{new}$;
3. Using the four normalized $rmsT$ as x', anterior four β_0 as y' to compute the constants b'_0, b'_1, b'_2, b'_3 of the second cubic curve;
4. Using the fifth normalized $rmsT$ as x'_{new} to compute the new value $\beta_2 = y'_{new}$;
5. $while \quad abs(\beta_1) > 360, \beta_1 = abs(\beta_1) - 360;$
   $if\, \beta_1 > 180,\, then\, \beta_1 = 360 - \beta_1,$
   $while \quad abs(\beta_2) > 360,\, \beta_2 = abs(\beta_2) - 360;$
   $if\, \beta_2 > 180,\, then\, \beta_2 = 360 - \beta_2;$
6. $\beta_0 = (3 \times \beta_1 + \beta_2)/4, if\, \beta_0 < betaMin, then \beta_0 = betaMin$(for the reason that RMS of rotation has a more important effect on the value of $beta_0$than translation).

By the detailed analysis on Observation 4 in [2], we note that d_0 affecting the RMS of translation error. So we use rmsT to predict.

Algorithm III

1. Using the four normalized $rmsT$ as x, anterior four d_0 as y to compute the constants b_0, b_1, b_2, b_3 of the cubic curve of d_0;
2. Using the fifth normalized $rmsT$ as x_{new},to compute the new value of d_{0new};
3. $if \ d_{0new} < 0, d_0 = 10; if \ d_{0new} > dMax, d_0 = 150$;

One can see that all three algorithms are linear, so the new method don't need extra computational time.

4 Experiments

In this section, experiments on synthetic data and real scenes are carried out to validate our algorithm.As Shi [11] did comparison to traditional hand-eye online calibration, like in Andreff[9][10], we just do experiments to compare the performance between our method and the method in Shi[11].

4.1 Simulated Data

The motivation of the simulated experiments is to test the performance of the new method for random motions.

The simulation is conducted as follows: we establish a consecutive random motion series with 500 hand stations Q_i. We add uniformly distributed random noise with relative amplitude of 0.1% on the rotation matrix and of 1% on the translation vector. We assume a hand-eye setup and compute the camera pose P_i , to which we also add uniformly distributed random noise as before.

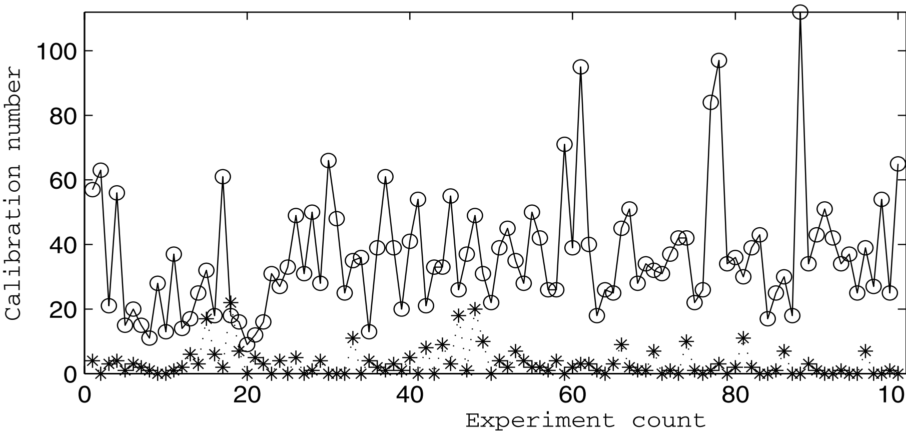

Fig. 2. Calibration number in 100 experiment, where the solid with label "O" denote our method and the dotted with label "*" denote the method in Ref.[11]

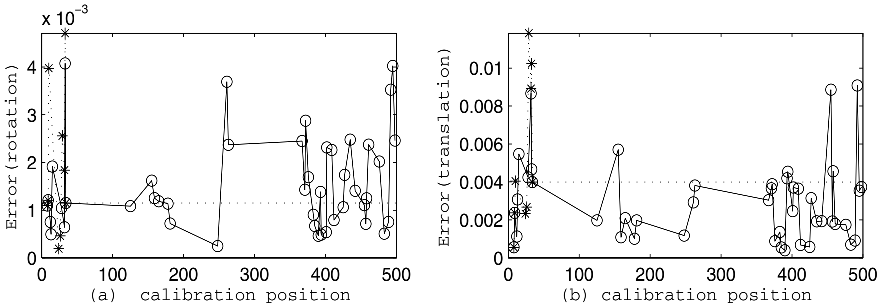

Fig. 3. RMS of the errors in the rotation matrix and the RMS of the relative errors $\|t - \hat{t}\|/\|t\|$ in the translation, where the solid with label "O" denote our method and the dotted with label "*" denote the method in Ref.[11]

We did 100 experiments and recorded the calibration number each time. To compare the performance, we make additional experiments by using the method of Shi [11] and use the same random consecutive motion series each time. We also record the 100 result of calibration number. The thresholds in the method of Shi [11] are set by the average value of the first five motions. Fig. 2 shows the results of the test. It can be seen that by using our method we can do much more calibrations than that of the method in Ref. [11]. Using our method, the average number of calibration is 36.4300. Using the method in Ref. [11], it is only 3.12 and it cannot do calibration at all in 29 instances of 100 tests. We also give a result of RMS of the errors in the rotation matrix and the RMS of the relative errors $\|t - \hat{t}\|/\|t\|$ in the translation of one of the 100 experiments. The result is shown in Fig.3. One can see that the accuracy of our method is about equal to the method in Ref[11].Using our method, it can do 45 calibrations, but using the method in Ref[11],the times is only 9.

4.2 Real Scenes

We also demonstrate the foregoing algorithm on a real setup composed of an infrared marker and a pair of CCD cameras (Watec-902B (CCIR)), which are attached to the end-effector of a 6-DOF robot (MOTOMAN CYR-UPJ3-B00). After the stereo rig is precisely calibrated, we mount an infrared filter on each camera. Thus, we get an infrared navigation system with stereoscopic vision. Without loss of generality, we compute the hand-eye transformation between the left camera and the gripper.

Table 1. Results of the real experiment

	stations	Times of calibration
Our method	35	7
Method in Ref[11]	35	1

In the real test, the aim is to prove the adaptability of the proposed algorithm to translation and compare the performance of our method and the method of Shi [11]. The robot is fixed on a workbench and the moving cameras observe the static infrared mark. We move the gripper to 35 locations. Each time, the gripper is moved closer to the mark than the previous one. For every time instant, gripper pose Q_i can be read from robot controller and the pose of reference object P_i relative to the camera can be solved by binocular vision. As the camera pose can be computed, we adopt Algorithm Main in the real experiment. We perform the hand-eye calibration using the similar methods as in the synthetic experiments. The different point is the computation of RMS of rotation error and translation error. In each calibration, we let $A_iX - XB_i = \begin{bmatrix} \Delta R_{3\times3} & \Delta T_{3\times1} \\ 1 & 0 \end{bmatrix}$ where the 3×3 matrix $\Delta R_{3\times3}$ take the responsibility of rotation error and the 3×1 vector $\Delta T_{3\times1}$ take the responsibility of translation error. Although they are not the real error of rotation and translation, the change of them can reflect the change of real couple and so take the responsibility.

In this situation, as our method is adaptive, the thresholds can be changed according to the characteristic of the motion. Although the translation become larger and larger, it still can do calibration. But the thresholds of the method of Shi [11] cannot be changed. So it only did few calibrations in this situation.

The experiment results of times of calibration are shown in Table 1. One can see that our method makes more calibrations than the method in Ref. [11].

5 Conclusion

In this paper, we propose an algorithm of adaptive motion selection for online hand-eye calibration, which can not only avoid the degenerate cases in hand-eye calibration, but also increase the calibration number by adaptively modify the thresholds according to the characteristics of motion sequence. So it is robust to different applications. Experimental results from simulated data and real setup show that the method can greatly increase the performance of online hand-eye calibration.

References

1. Y. C. Shiu and S. Ahmad, Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB, IEEE Trans. Robot. Automat., vol. 5, pp. 16-29, Feb. 1989.
2. R. Y. Tsai and R. K. Lenz, A new technique for fully autonomous and efficient 3d robotics hand/eye calibration, IEEE Trans. Robot. Automat., vol. 5, pp. 345-358, 1989.
3. H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. in Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Maui, Hawaii, USA, pp. 145-151, June 1991.
4. C. Wang. Extrinsic calibration of a robot sensor mounted on a robot. IEEE Trans. Robot. Automat., 8(2):161-175, Apr. 1992.
5. Hanqi Zhuang and Yui Cheung Shiu, A Noise-Tolerant Algorithm for Robotic Hand-Eye Calibration With or Without Sensor Orientation Measurement. IEEE Trans. on System, Man and Cybernetics, 23(4):1168-1175,1993.
6. R. Horaud and F. Dornaika,Hand-eye calibration, Int. J. Robot. Res., 14(3):195-210, 1995.
7. K. Daniilidis.Hand-eye calibration using dual quaternions. Int. J. Robot. Res., 18(3):286-298, 1999.
8. J. Angeles, G. Soucy and F. P. Ferrie,The online solution of the hand-eye problem, IEEE Trans. Robot. Automat., vol. 16, pp. 720-731, Dec. 2000.
9. N. Andreff, R. Horaud and B. Espiau,On-line hand-eye calibration, in Proc. Int. Conf. on 3-D Digital Imaging and Modeling, pp. 430 - 436, Oct. 1999.
10. N. Andreff, R. Horaud, and B. Espiau,Robot hand-eye calibration using structure-from-motion, Int. J. Robot. Res., 20(3):228-248, 2001.
11. F.H. Shi, J.H.Wang and Y.C.Liu,An Approach to Improve Online Hand-Eye Calibration. In Proc. of IbPRIA 2005, LNCS 3522, pp.647-655, 2005.
12. Regression Analysis, Numeral Statistic group, Chinese Academy of ScienceBeijing-Science Press1974.
13. S. Ma and Z. Zhang, Computer Vision, 2nd edition. Beijing : Science Press, 1998. ch.6.