Robotica (2007) volume 25, pp. 529–536. © 2007 Cambridge University Press
doi:10.1017/S0263574707003426 Printed in the United Kingdom

Adaptive motion selection for online hand–eye calibration

Jing Zhang*‡, Fanhuai Shi† and Yuncai Liu‡

†*School of Materials Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, P.R. China.*
‡*Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China.*

(Received in Final Form: January 26, 2007. First published online: March 9, 2007)

SUMMARY
While a robot moves, online hand–eye calibration to determine the relative pose between the robot gripper/end-effector and the sensors mounted on it is very important in a vision-guided robot system. During online hand–eye calibration, it is impossible to perform motion planning to avoid degenerate motions and small rotations, which may lead to unreliable calibration results. This paper proposes an adaptive motion selection algorithm for online hand–eye calibration, featured by dynamic threshold determination for motion selection and getting reliable hand–eye calibration results. Simulation and real experiments demonstrate the effectiveness of our method.

KEYWORDS: Adaptive; Motion selection; Online hand-eye calibration; Polynomial regression.

1. Introduction

The problem of computing the relative three-dimensional (3-D) position and orientation between a robot gripper and rigidly attached camera arises in hand–eye calibration.[2] In this process, accuracy is an essential factor, and the goal of this paper is to improve the accuracy of online hand–eye calibration. The method we proposed here is an extension of "motion selection" algorithm in ref. [14], which intends to exclude degenerate motions and small rotations during the calibration. Shi *et al*.'s method[14] is achieved by combining motions until the accumulated motion satisfies the pre-given thresholds, which are determined by experience. In this paper, we focus our mind on refining the method and proposing an adaptive algorithm to set the thresholds for "motion selection". Using this method, we can get "good" motions for online hand–eye calibration and improve the system accuracy.

The calibration of robotic hand–eye relationship is a classical problem in robotics.[1–17] Algebraically, the problem can be defined as a linear homogeneous equation in the unknown pose matrix X, namely, $AX = XB$,[1–10,21] where A is the rigid motion of the robot gripper, and B is the corresponding camera motion. A, B and X are all 4×4 homogeneous transformation matrices.

Although hand–eye calibration has been studied for many years, most hand–eye calibration methods are iterative and time consuming, which are not suitable for online computation.[1–10,15–17] Online hand–eye calibration is a useful and necessary technology in many applications. For example, in hand–eye robotic system for total knee replacement,[20] the camera may be bumped accidentally during the surgery process, causing the changes of the hand–eye relationship. Using online calibration, one can judge the change and correct it quickly.

Andreff *et al*.[12,13] and Angeles *et al*.[11] firstly introduced the method of online implementation of hand–eye calibration. It has been shown in ref. [11] that we can finally obtain a solution based on recursive linear least squares. In refs. [12],[13] Andreff *et al*. derived a linear formulation of the problem. This makes an algebraic analysis possible to extend this formulation into an online hand–eye calibration, getting rid of the calibration object required by the conventional approaches. Boctor *et al*.[22] presented a method performing accurate online hand–eye partial calibration for ultrasound probe. The key enabler to this method is "Real-time Tracker", which is used to recover the motion of robot gripper.

Whichever method one chooses, two motions with non-parallel rotation axes must be used to determine the hand–eye transformation. The algebraic and geometrical analysis on hand–eye calibration can be seen in refs. [2],[3]. In practical online hand–eye calibration, one cannot know the movements of a robot beforehand, which precludes the motion planning for hand–eye calibration in advance. Small rotations and degenerate motions used in calibration can ruin the result. However, there are no effective techniques proposed to solve this problem in refs. [11–13].

Shi *et al*.[14] firstly introduced the concept of "motion selection", trying to select the "effective" motions in online hand–eye calibration to reduce the risk of bad results. They make "motion selection" by defining the "golden rules" according to the *observations* of ref. [2]. If a motion fulfills these rules, the algorithm will regard the motion as "effective," and it will be accepted for further computation. Otherwise, the algorithm will combine it with its consecutive motions to form a new motion until the accumulated motion satisfies the pre-defined thresholds. The selected effective motions are then used for calibration. Using this method, one can avoid small rotations and degenerate motions and, hence, can greatly decrease the calibration error. However, the algorithm in ref. [14] sets the thresholds by experience, which may not be flexible for the requirements of different applications.

In this paper, we propose an algorithm to determine thresholds adaptively for motion selection in online hand–eye calibration. The remainder of this paper is decomposed as follows. Section 2 describes the objective problem. Section 3 gives the detailed algorithm of adaptive motion selection

* Corresponding author. E-mail: zhjseraph@sjtu.edu.cn

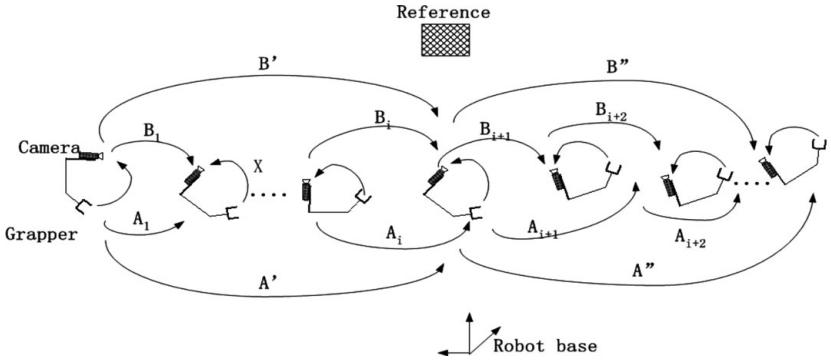

Fig. 1. Algorithm of motion selection for online hand–eye calibration.

for online hand–eye calibration. Section 4 conducts some simulated and real experiments to validate the proposed algorithm.

2. Problem Formulation

In this section, we first describe the algorithm of Shi, [14] which is the foundation of our method. Then we will discuss the problem we attempt to solve.

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

Usually, hand–eye calibration is described by a homogeneous transformation matrix. 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 ith motion matrix of the camera by $B_i = (R_{b,i}, t_{b,i})$. The motion of the gripper is computed directly from the joint-angle readings by a simple composition of $A'_i = Q_i^{-1} Q_{i+1}$. With the known intrinsic camera parameters, the camera poses P_i and P_{i+1} relative to the reference object are estimated. Then the camera motion can also be determined from $B'_i = P_i^{-1} P_{i+1}$. When dealing with an unknown scene (such as the scene without special calibration object), we can use a structure from motion algorithm[12,13] to estimate the camera motion directly. Thus, the well-known hand–eye equation of $A_i X = X B_i$ can be established.[1,2]

Since a rotation matrix R can be expressed as a rotation of angle θ around a rotation axis k, the relations between θ, k and R can be determined by Rodrigues theorem.[19] Moreover, R_a and R_b have the same angle of rotation,[1] and we can rewrite R_a and R_b as $Rot(k_a, \theta)$ and $Rot(k_b, \theta)$, respectively. Due to the fact that degenerated motions or small rotations will introduce large errors in hand–eye calibration, motion selection is essential to find "good" pairs of consecutive motions(A_i, B_i) and (A_{i+1}, B_{i+1}) for hand–eye computation from sampled motion series.[14]

Shi *et al.*[14] made motion selections with the following observations in ref. [2].

Observation 1: The RMS (root mean square) error of rotation from gripper to camera is inversely proportional to the sine of the angle between the interstation rotation axes (denoted by $\sin(\alpha)$).

Observation 2: The rotation and translation error are both inversely proportional to the interstation rotation angle (denoted by β).

Observation 4: The distances between the robot gripper coordinate centers at different stations (denoted by d) are also a critical factor in forming the error of translation.

According to the above three observations, 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, which is no less than α_0.

Rule 2: Try to make θ_i large, which is no less than β_0.

Rule 3: Try to make $\|t_{a,i}\|$ small, which is no bigger than d_0.

Let us denote the ith sample of hand–eye pose and motion by (P_i, Q_i) and (A_i, B_i), respectively, in the following paper, and α_0, β_0, d_0 are thresholds determined by experience. Also (A', B') and (A'', B'') are selected motion pairs for the calibration (see Fig. 1). In motions A' and A'', the rotation axis, rotation angle and translation are denoted by (k'_a, θ'_a, t'_a) and $(k''_a, \theta''_a, t''_a)$, respectively.

At the beginning of the calibration, we need to estimate (A', B'), which is first recovered from (P_1, Q_1) and (P_2, Q_2). If $\theta' \geq \beta_0$ and $\|t'_a\| \leq d_0$, we can claim that (A', B') has been found. Otherwise, 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, it is assumed 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 found in the similar way, starting from (P_i, Q_i) and (P_{i+1}, Q_{i+1}). But the constrained conditions here are changed to $\theta' \geq \beta_0$, $\|t'_a\| \leq d_0$ and $\angle(k'_a, k''_a) \geq \alpha_0$. When both motion pairs are found, we can make a calibration using the method of Andreff.[12]

In the next calibration, we take the motion pair (A'', B'') of the last process as the new motion pair (A', B'), and then continue to seek for new (A'', B'') from the consecutive series and make a new hand–eye calibration repeatedly.

The corresponding algorithm of *Motion selection algorithm* by Shi *et al.*[14] is as follows:

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

1) $i \leftarrow 2$;
2) $A' = Q_1^{-1} Q_i$, $B' = P_1^{-1} P_i$;
3) Compute θ' and t'_a from A';
4) If $\theta' \geq \beta_0$ and $\|t'_a\| \leq d_0$, then go to 6;
5) $i \leftarrow i + 1$, go to 2; (sample one more motion);
6) $j \leftarrow i + 1$ (begin to search for A'');
7) $A'' = Q_i^{-1} Q_j$, $B'' = P_i^{-1} P_j$;
8) Compute $\angle(k'_a, k''_a)$, θ'' and t''_a from A' and A'';
9) If $\angle(k'_a, k''_a) \geq \alpha_0$, $\theta'' \geq \beta_0$ and $\|t''_a\| \leq d_0$, then go to 11;
10) $j \leftarrow i + 1$, go to 7 (sample one more motion);
11) Make one hand–eye calibration using the method in Andreff *et al.*;[12]
12) $A' \leftarrow A''$, $B' \leftarrow B''$;
13) $i \leftarrow j$, $j \leftarrow j + 1$, go to 7 for next calibration.

In the above motion selections, α_0, β_0 and d_0 are set by experience, which are not robust in most cases. In order to improve the robustness and accuracy of the motion selection method in online hand–eye calibration, we will discuss the adaptive algorithm of motion selection in the next section.

3. Adaptive Selection of Motion for Online Hand–Eye Calibration

3.1. Adaptive selection algorithm

We initially set the average value of $\sin(\alpha)$, β, d of the first N motions as the initial thresholds

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

The number N can be chosen based on the real applications, i.e. if the robot will take large number motions, we can choose a large N; otherwise, a small N will be used. Then, we do motion selection and hand–eye calibration four times. In each calibration, we compute the RMS error in rotation and translation and modify the thresholds by multiplying $\sin(\alpha_0)$, β_0 and d_0 with a parameter, respectively. During this process, if the calibration can not be successfully carried out after processing "*interval*" motions (in our application, we set $interval = 5$) , which means that these thresholds are hard to be satisfied, then we will simply reduce $\sin(\alpha_0)$, β_0 and increase d_0 by multiplying parameters.

After doing four calibrations, we begin to adaptively set the thresholds of $(\sin(\alpha), \beta, d)$ using polynomial-regression method. In the subsequent calibration, we take the motion pair (A'', B'') of the last process as the new motion pair (A', B'), and then continue to seek for new (A'', B'') from the consecutive motion series with new thresholds.

The corresponding algorithm is as follows.

Main Algorithm

1) Compute the initial thresholds of $\sin(\alpha_0)$, β_0, d_0 , using Eq(1);
2) $i \leftarrow N + 1$, $calibNo \leftarrow 0$;
3) $A' = Q_N^{-1} Q_i$, $B' = P_N^{-1} P_i$;
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 reduce β_0 by multiplying a parameter less than 1,
   If $\|t'_a\| > d_0$, then increase d_0 by multiplying a parameter larger than 1, $i \leftarrow i + 1$, $start \leftarrow i$, goto 3;
8) $j \leftarrow i + 1$, $start \leftarrow j$ (begin to search for A'');
9) $A'' = Q_i^{-1} Q_j$, $B'' = P_i^{-1} P_j$;
10) Compute $\angle(k'_a, k''_a)$, θ'' and t''_a from A' and A'';
11) If $\sin(\angle(k'_a, k''_a)) \geq \sin(\alpha_0)$, $\theta'' \geq \beta_0$ 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 reduce $\sin(\alpha_0)$ by multiplying a parameter less than 1,
   if $\theta' \leq \beta_0$, then reduce β_0 by multiplying a parameter less than 1,
   if $\|t''_a\| \geq d_0$, then increase d_0 by multiplying a parameter larger than 1, $j \leftarrow j + 1$, $start \leftarrow j$, go to 9;
14) Make one hand–eye calibration using the method in Andreff *et al.*,[12] $calibNo \leftarrow caliNo + 1$, compute RMS of the errors in the rotation matrix and the RMS of the relative errors $\|t - \hat{t}\|/\|t\|$ in the translation (which are customary error metrics in the literature),[2,6,7] record the value of $(\sin(\alpha_0), \beta_0, d_0)$;
15) if $calibNo \geq 4$, then use *Algorithms I, II,* and *III*, respectively, to predict the next set of thresholds;
16) $A' \leftarrow A''$, $B' \leftarrow B''$;
17) $i \leftarrow j$, $j \leftarrow j + 1$, go to 9 for next calibration.

3.2. Adaptive selection of the thresholds

The purpose of our method is to predict the thresholds of motion selection, which are suitable for the current motion sequence and improve the calibration accuracy. That is, if an application needs certain accuracy, we should set suitable thresholds in selecting motions to get the required accuracy. (In the following paragraphs, we aggregated all errors into RMS error. Therefore, when we say "error in rotation", we mean RMS error.) We represent the required rotation accuracy as $rmsRMax$ and the required translation accuracy as $rmsTMax$. From the observations in ref. [2], we can see that there are non-linear relationship between $(\sin(\alpha_0), \beta_0, d_0)$ and the error in rotation and translation in hand–eye calibration. Cubic polynomial-regression[18] is suitable for predicting suitable thresholds. In the following paragraphs, we will represent the rotation error by $rmsR$ and the translation error by $rmsT$. Because at least four sets of independent motions are required to solve the unknowns, we do four calibrations at the beginning of the process. We denote the average value of four $rmsR$ and four $rmsT$ as $avgRrms$ and $avgTrms$, respectively. The mathematical model of evaluating the errors and thresholds is given by the polynomial regression model,[18] as depicted by the following equation:

$$y = b_0 + b_1 \times x + b_2 \times x^2 + b_3 \times x^3 \quad (2)$$

where x is $rmsR$ or $rmsT$, y is the threshold, and $b_0, b_1, b_2$and b_3 are the parameters to be determined. We

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

use four sets of $rmsR$ and $rmsT$ as independent variables, and four sets of $(\sin(\alpha_0), \beta_0, d_0)$ as dependent variables to simulate the models of cubic polynomial-regression. After getting three cubic curves of $(\sin(\alpha_0), \beta_0, d_0)$, we compare $avgRrms$ with $rmsRMax$ and $avgTrms$ with $rmsTMax$, respectively, and then use the smaller couple to predict the new value of $(\sin(\alpha_0), \beta_0, d_0)$. A smaller $avgRrms$ or $avgTrms$ implies that the former selection of thresholds is suitable for the application and, therefore, satisfies the required accuracy. Otherwise, we should use $rmsRMax$ and $rmsTMax$ to modify the thresholds. Using the new thresholds, a new calibration should be done.

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

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

where

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

The corresponding algorithm is as follows. In these algorithms, $alphaMin$, $betaMin$ are the minimum values that $\sin(\alpha_0)$, β_0 should satisfy, $dMax$ is the maximum value that d_0 should satisfy. We can estimate them using *Observations 1, 2* and *4* in ref. [2] according to the accuracy requirement of application.

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

Algorithm I

1) Set the four normalised *rmsR* to form x , four normalised $\sin(\alpha_0)$ to form y, and then compute the parameters $b_0, b_1, b_2$and b_3 of the cubic curve of $\sin(\alpha_0)$, that is

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

2) If $avgRrms \le rmsRMax$, use $avgRrms$ as x_{new} to compute the new value of $\sin(\alpha_0)$, namely

$$\sin(\alpha_0)_{\text{new}} = b_0 + b_1 \times avgRrms + b_2 \times avgRrms^2 + b_3 \times avgRrms^3; \quad (5)$$

Otherwise, use $rmsRMax$ as x_{new} to compute the new value of $\sin(\alpha_0)$;

3) If $\sin(\alpha_0) < alphaMin$, then $\sin(\alpha_0) = alphaMin$.

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

Algorithm II

1) Set the four normalised $rmsR$ to form x , four normalised β_0 to form y, and compute the parameters b_0, b_1, b_2, b_3 of the cubic curve of β_0;
2) Use the smaller one of $avgRrms$ and $rmsRMax$ as x_{new} to compute the new value $\beta_1 = y_{\text{new}}$;
3) Use the four normalised $rmsT$ as x', four β_0 as y' to compute the parameters b'_0, b'_1, b'_2, b'_3 of the second cubic curve;
4) Use the smaller one of $avgTrms$ and $rmsTMax$ as x'_{new} to compute the new value $\beta_2 = y'_{\text{new}}$;
5) $\beta_0 = (3 \times \beta_1 + \beta_2)/4$, if $\beta_0 < betaMin$, then $\beta_0 = betaMin$ (according to *Observation 2* in ref. [2], error of the rotation is more important than that of the translation in affecting the value of $beta_0$).

By the detailed analysis on *Observation 4* in ref. [2], we note that d_0 affects the translation error. Therefore, we use $rmsT$ to predict d_0.

Algorithm III

1) Set the four normalised $rmsT$ to be x, four normalised d_0 to be y to compute the parameters b_0, b_1, b_2, b_3 of the cubic curve of d_0;
2) Use the smaller one of $avgTrms$ and $rmsTMax$ to be x_{new} and compute the new value of $d_{0\text{new}}$;
3) If $d_{0\text{new}} > dMax$, then $d_{0\text{new}} = dMax$.

One can see that all the three algorithms are analytical, therefore, the new methods needs no extra computational time.

4. Experiments

In this section, experiments using synthetic data and real scenes are carried out to validate the accuracy, adaptability and real-time quality of our new method.

4.1. Simulated data

4.1.1. Accuracy test. The experiment is carried out to test the performance of the new method with respect to the computing error. To make a comparison, we also solve $AX = XB$ without any motion selection, as the method given by Andreff *et al.*[12,12'] It is an applicable method to do online hand–eye calibration now. In the following paragraphs, we denote the proposed method by "new method" and the direct approach by "conventional method".

The simulation is conducted as follows: we establish a random consecutive motion series with 200 hand positions Q_i. Then 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.

With 20 motion series, we perform calibrations for each one using the new and the conventional method, respectively.

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

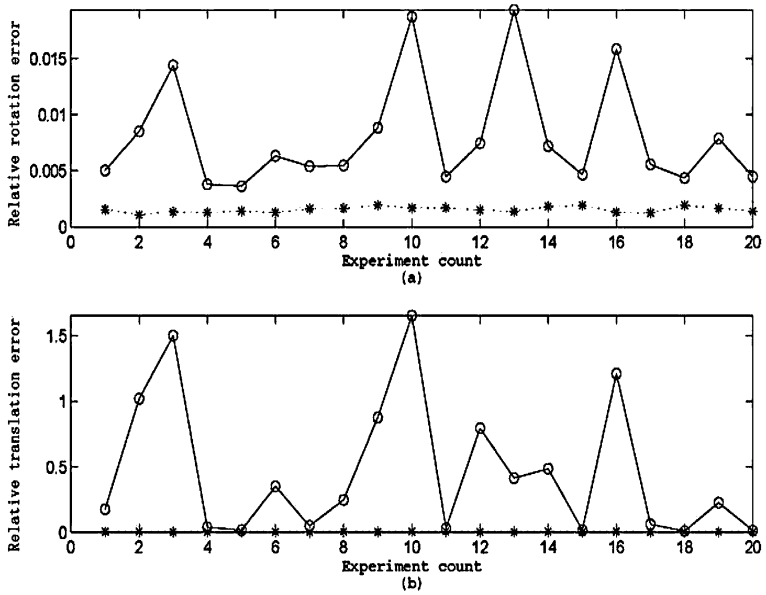

Fig. 2. (a) Average errors in the rotation matrix and (b) average relative errors $\|t - \hat{t}\|/\|t\|$ in the translation, where the *solid curve* with label "O" denotes the conventional method while the *dotted curve* with label "*" denotes the new method.

To qualify the results, we compute average error in each case. (In the case of one motion series, we get the estimated rotation matrix $\widehat{R}$ and the translation vector $\widehat{t}$ in each calibration and compute the errors in the rotation matrix and the the relative errors $\|t - \hat{t}\|/\|t\|$ in the translation vector. Then sum the errors in the rotation and translation, respectively. The average error is computed through dividing the sum by the calibration number). Figure 2 shows the simulation results. One can see that the adaptive motion selection approach behaves much better than the conventional method in the existence of noisy measurements.

4.1.2. Adaptability test. To compare the adaptivity performances, we also carry out the experiment with "fixed thresholds method".[14] In this experiment, we establish the consecutive random motion series with 500 hand positions Q_i. Other conditions are the same as the first simulated experiment. With 100 such motion series, we calibrate each using the new method and the fixed thresholds method, and record the calibration number. The thresholds used in the fixed thresholds method are set by the average value of the first five motions. Figure 3 shows the results. It can be seen that with the new method, we can successfully do more calibrations than using the fixed thresholds method: with the new method, the average calibration number is 36.4300, whereas the fixed thresholds method makes only 3.12 calibrations. This test result demonstrates that the new method is much more adaptive in various applications.

4.1.3. Real-time test. To make further comparison, we make an additional experiment using optimization approach, called "optimal method" here. It first selects five "good" motions using the same algorithm as the new method, then performs the calibration with these "good" motions by optimization approach (lease-squares method). In this experiment, we establish the consecutive random motion series of 100 hand positions Q_i. Other conditions are the same as the first simulated experiment. We record the average error in the rotation and translation, respectively. We also record operation time of the calibration. From the result shown in

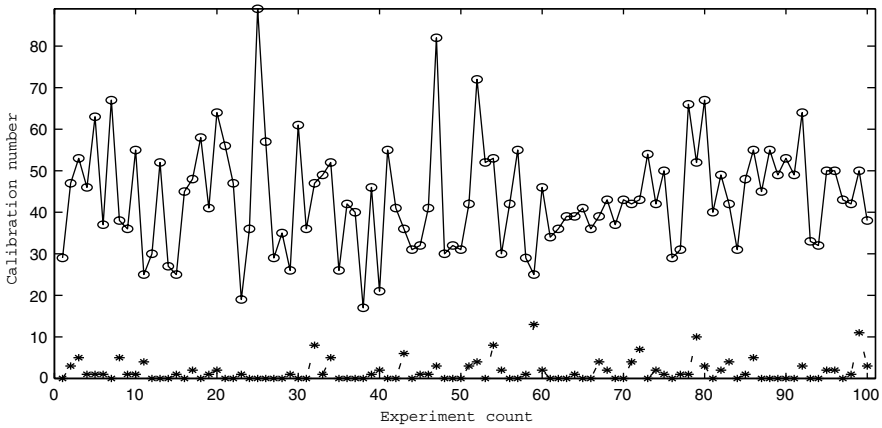

Fig. 3. Calibration number in 100 experiments, where the *solid curve* with label "O" denotes the new method while the *dotted curve* with label "*" denotes the fixed threshold method.

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

Table I. Results of the real-time test.

	Average time for once calibration (s)	Average error of rotation	Average error of translation
New method	0.0020	0.0013442	0.0039
Optimal method	6.7114	0.0012907	0.00067138

Fig. 4. A physical setup. Two CCD cameras are rigidly mounted on the end-effector of an Motorman Robot for performing hand–eye calibration.

Table I, one can see that the accuracy of optimal method is not much better than the new method. But the optimal method operates much longer, which may not satisfy the requirement of online calibration. Therefore, we do not use optimization in our method.

4.2. *Real scenes*

We demonstrate the foregoing algorithm by a motorman robot (MOTOMAN CYR-UPJ3-B00) system with Charge-Coupled Device (CCD) stereo cameras [Watec-902B (CCIR)], as shown in Fig. 4. We use this setup in our research on Robot Assisted Surgical System for Total Knee Replacement.[20] The robot operation mode is vertically articulated. Its repetitive positioning accuracy is 0.03 mm. The robot is fixed on a workbench, and the cameras are mounted on the end-effector of the robot, observing the static infrared marks. 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. For each time instant, the gripper pose Q_i can be read from robot controller and the pose of the reference object P_i related to the camera can be solved by binocular vision.

We perform the hand–eye calibration using similar methods as in the synthetic experiments. However, we cannot have the ground truth of calibrations. Therefore, to predict thresholds we compute

$$A_i X - X B_i = \begin{bmatrix} \Delta R_{3\times 3} & \Delta T_{3\times 1} \\ 0 & 1 \end{bmatrix}.$$

In $Algorithms I$, II and III, the RMS of $\Delta R_{3\times 3}$ stands for $rmsR$, and the RMS of $\Delta T_{3\times 1}$ stands for $rmsT$.

To make further comparison, we also calibrate the system using Andreff's "offline method"[12] with all the frontal motion pairs in computation, i.e. at the ith interstation, we use i motion pairs to compute the result.

4.2.1. Accuracy test. We randomly move the gripper to get 50 locations with different rotation or/and translation controlled by the program. To qualify the results, we take the errors of all calibrations according to the pre-computed result (this error is not used for threshold prediction, but only for accuracy tests). The pre-computed result was obtained offline using optimization and motion planning according to ref. [2]. Thus, it is an accurate result reflecting the real value of hand–eye relationship.

The results of this experiment are shown in Table II and Fig. 5. From Table II, we see that, the average error of the new method and the offline method are similar, both much lower than that of the conventional method. From Fig. 5, we can see that the new method and the offline method are much more stable than the conventional one.

4.2.2. Adaptability test. In this experiment, we try to show that the adaptability of the proposed algorithm is better than the "fixed thresholds method" and the "offline method". We move the gripper to get 35 locations, with each of which, the gripper is made closer to the mark than the previous location.

In our method, the threshold d_0 can be changed adaptively based on the characteristic of the motion sequence. Although the translation becomes larger and larger, it can still do calibration accurately. In comparison, however, the thresholds of the fixed thresholds method cannot be changed. So, it did only few calibrations in this situation.

Numbers and accuracy of the calibration are shown in Table III. One can see that this test validates our prediction because the new method makes more calibrations than the fixed thresholds method and the accuracy of the new method is superior to the offline method. That means, when the motion sequence is not suitable to do hand–eye calibration, the new method can get better result than the fixed thresholds method and the offline method.

Table II. Results of the first real experiment.

	Times of calibration	Average error in rotation	Average error in translation
New method	9	0.068417	0.18171
Conventional method	48	0.1747	0.9654
Offline method	48	0.0591	0.1382

Table III. Results of second real experiment.

	Calibration numbers	Average error in rotation	Average error in translation
New method	7	0.3387	1.1197
Fixed thresholds method	1	0.4233	1.1954
Offline method	35	0.64503	1.6946

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

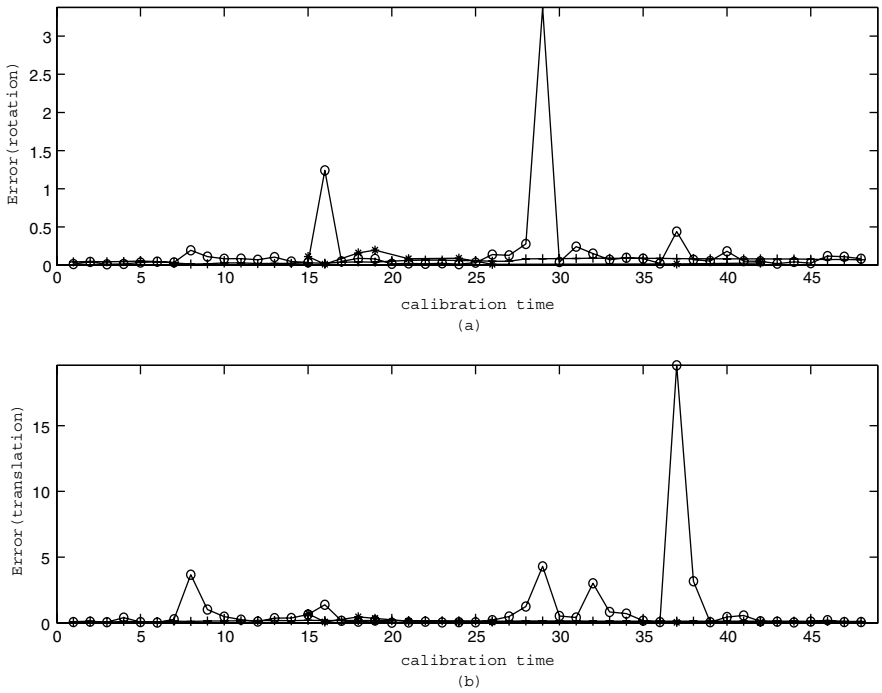

Fig. 5. (a) The errors in the rotation matrix in each calibration and (b) the relative errors in the translation, where the *solid curve* with label "O" denotes the conventional method, label "*" denotes the new method and label "+" denotes the offline method.

5. Conclusion

In this paper, we proposed an algorithm of adaptive motion selection for online hand–eye calibration. There are nonlinear relationships between the motion parameters and the calibration accuracy. We used cubic polynomial-regression to predict suitable thresholds for motion selection. Using this method, we can not only avoid the degenerate case in hand–eye calibration, but also increase the calibration number by adaptively modifying the thresholds. Experimental results from simulated data and the real setup show that the method greatly increased the quality of the performance of online hand–eye calibration. It is assumed that the proposed method is very promising in the vision-guided robot systems.

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. Autom.* **5**, 16–29 (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. Autom.* **5**, 345–358 (1989).
3. H. Chen "A screw motion approach to uniqueness analysis of head-eye geometry". *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii (Jun. 1991). Pp. 145–151.
4. C. Wang, "Extrinsic calibration of a robot sensor mounted on a robot," *IEEE Trans. Robot. Autom.* **8**(2), 161–175 (Apr. 1992).
5. H. Q. Zhuang and Y. C. Shiu, "A noise-tolerant algorithm for robotic hand–eye calibration with or without sensor orientation measurement," *IEEE Trans. Syst., Man Cybern.* **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. Y. Motai and A. Kosaka, "SmartView: Hand–eye robotic calibration for active viewpoint generation and object grasping," *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2183–2190 (2001).
9. A. Muis and O. Kouhei, "An iterative approach in pose measurement through hand–eye calibration," *Proceedings of the IEEE International Conference on Control Application*, Istanbul, Turkey. (2003), pp. 983–988.
10. H. Malm and A. Heyden, "Simplified intrinsic camera calibration and hand–eye calibration for robot vision," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada (2003), pp. 1037–1043.
11. J. Angeles, G. Soucy and F. P. Ferrie, "The online solution of the hand–eye problem," *IEEE Trans. Robot. Autom.* **16**, 720–731 (Dec. 2000).
12. N. Andreff, R. Horaud and B. Espiau, "On-line hand–eye calibration," *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, 430 – 436 (Oct. 1999).
13. N. Andreff, R. Horaud and B. Espiau, "Robot hand–eye calibration using structure-from-motion," *Int. J. Robot. Res.* **20**(3), 228–248 (2001).
14. F. H. Shi, J. H.Wang and Y. C. Liu, "An Approach to improve online hand–eye calibration," *Proc. IbPRIA 2005, LNCS 3522*, 647–655 (2005).
15. H. Q. Zhuang and A. Melchinger, "Calibration of a Hand/eye matrix and a connection matrix using relative pose measurements," *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico (1997), pp. 2888–2893.
16. T. Heikkila and T. Matsushita, "Flexible hand–eye calibration for multi-camera systems," *IEEE/RSJ Int.Conf. Intell. Robots Syst.* 2292–2297 (2000).
17. R. L. Hirsh and G. N. DeSouza, "An interative approach to the hand–eye and base-borld calibratino problem," *Proceedings of the IEEE International Conference on Robotics and Automation*. Seoul, Korea (2001), pp. 2171–2176.

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press

18. Numeral Statistic group, *Regression Analysis*, (Chinese Academy of Science, Science Press, Beijing, China, 1974).
19. S. Ma and Z. Zhang, *Computer Vision, 2nd ed*. (Beijing, China: Science Press, 1998. ch. 6).
20. F. H. Shi, J. Zhang, Y. C. Liu and Z. J. Zhao. "A hand–eye robotic model for total knee replacement surgery," *Proceedings of the 8th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'2005)* (2005), pp. 122–130.
21. I. Fassi and G. Legnani, "Hand to sensor calibration: a geometrical interpretation of the matrix equation $AX = XB$", *J. Robot. Syst.*, **22**(9), 497–506 (2005).
22. E. Boctor, I. Iordachita, G. Fichtinger and G. D. Hager, " Real-time quality control of tracked ultrasound," Proceedings of the MICCAI Conference (2005), pp. 621–630.

https://doi.org/10.1017/S0263574707003426 Published online by Cambridge University Press