A Flexible Solution to AX=XB for Robot Hand-Eye Calibration

Jianfei Mao, Xianping Huang, Li Jiang
Computer Science Institute
Zhejiang University of Technology
China
mjf@zjut.edu.cn

Abstract: A direct linear closed-form solution followed by Jacobian optimization is proposed to solve AX=XB for hand-eye calibration. The approach does not require A,B satisfying rigid transformation rather than the classic ones based on quaternion algebra or screw rule, so the technique is more flexible. Rigid deduction and demonstration are given based on matrix theory. Computer simulation and real data implementation indicate that: (1) In computation of initial value, our technique has good precision and robustness. (2) As more equations are added, initial value will converge to final value gradually, which shows it credible to regard initial value as final solution when many equations are supplied.

Key-Words: - Hand-eye Calibration; Jacobian Optimization; Linear Closed-form Solution

1 Introduction
Hand-eye calibration is the basis of robot off-line programming. The calibrated hand-eye helps to robot realizing precise location and metric reconstruction, which is important particularly for precision manufacturing industry. Hand-eye calibration problem[1,2], which is to determine the transformation from camera to robot coordinate system, virtually yields a homogeneous matrix equation of the form \( AX = XB \). Several closed-form solutions [2~7] were proposed in the past to solve for \( X \) as well as a nonlinear optimization method. Tsai [3] and Shiu [4] presented the linear algorithms based on the screw theory respectively, here, Tsai analyzed the error elaborately in detail, and proposed very practical calibration scheme to improve the calibration accuracy and robustness. Their algorithms are described at the geometric insight [7] and are self-integrated in theory, but their deduction procedures are quite complicated. To overcome this deficiency, Zhuang et al. [5] used quaternions to solve the equation and simplified the problem. Chou and Kamel [8] presented the form-closed solution based on quaternions, subsequently, horaud and Dornaika [9] implement the optimization to their solution. Konstantinos [10] and Schmidt [11] used dual quaternions in the hand-eye calibration and described the solution by singular value decomposition. Zhao [12] employed the screw theory with quaternions to solve the calibration problem.

The forementioned algorithms are based on quaternions or screw theory, it requires that A, B are rigid transformations. In fact the given A or B might be the approximate rigid transformations, it needs rigid transformation estimation on A, B, and this will produce cumulative errors. (2) They don’t consider the rotational deviations between A, B. In fact though A,B can satisfy the rigid transformations, the rotational angles of A and B are not equal generally, and small deviations between the rotational angles may cause comparative big errors.

We propose a direct linear method to solve the rotational part of A, B through SVD, which doesn’t need to decompose A, B for the screw vectors or rotational axes, therefore, our approach needn’t consider whether A, B satisfy the rigid transformation or their rotational angles are equal, so it can widely satisfy general situation. As the solution by this linear method mayn’t satisfy rotational transformation, we estimate the optimal rotational matrix by maximum likelihood method, and then we employ the same approach as the class ones to solve the translation part linearly. To alleviate the cumulative errors, we establish the objective function of optimization, and derive the iterative Jacobian formula, and implement the non-linear optimization with Gauss-Newton method or Levenberg-Marquet method [13]. Our technique has complete theoretical basis with rigorous demonstrations and deductions. Both Simulations and real data implementation have been done to compare our and classic approaches.

2 Closed-Form Solution
In this section we will mainly elaborate the linear method based on SVD, the estimation of rotational matrix and its derivation procedure.

Firstly, we know \( AX=XB \) can be represented as:
With SVD, because \( A \) is a real matrix.

Lemma 2 The identical norm least square solution for \( AX=0 \) is the identical eigenvector corresponding to the minimal eigenvalue of \( A^*A \).

The above lemmas are easy to prove or see article [14] for the detail.

Lemma 3 There always exists real eigenvector corresponding to real eigenvalue for arbitrary real square matrix.

Lemma 4 As a supplement of Lemma 2, if \( A \) is a real matrix, it can be decomposed by SVD in real field, that is to say both of unitary matrices \( U,V \) got by SVD can belong to real field.

Lemma 5 By SVD of nonzero real matrix \( Q: Q=USV^* \), the orthonormal matrix most approximated to a scale matrix \( \lambda Q \) is exclusively determined by \( R=UY^* \).

See article [14] for the detailed proof of Lemmas 5,6.

Lemma 7 If both \( A,B \) are rotational, it is necessary to solve for rotational part of \( X \) by at least two consistent equations.

If both \( AB \) are rotational, Zhuang and Shui have proved the lemma in detail. Moreover, since (4) is linear, obviously, it is robust to use multi-equations to solve. However, \( AB \) might not be rotational, through lemma 1 in book[15], the sufficient and necessary condition that there exists solution is that: some eigenvalue of \( A \) is equal to some eigenvalue of \( B \).

2.2 Linear Solution to the translation part

By Lemma of Zhuang, if \( R_3 \) is rotational, it requires at least two consistent equations as (2) to solve for \( T_x \). The identical norm least square solution of \( X \) to multi-equations will enhance the robustness of solution. Obviously, QR or SVD, which are all good linear algorithms, can be used to solve (2) for \( T_x \).

2.3 Brief Summary

Lemmas 3~4 are the basis of the other lemmas, and of our algorithms, as they ensure the algorithms are closed in real field. The scheme for the solution of initial value is given as follows. Note that multi-equations will ensure the solution robust.

(1) According to lemmas 1~2, solve (3) to get a particular real solution and fold it into a square matrix \( B \).

(2) According to Lemma 5~6, the optimal rotational estimation most approximated to \( \lambda Q \) is: \( R=UY^* \).

(3) Solve (2) for the translation part with QR or SVD.

3 Jacobian Optimization

The solution steps in section 2 have a deficiency that the rotation errors will propagate to the solution for the translation part, and are amplified by \( \|T_x\| \) approximately. It is necessary to implement an optimization to solve the translation and rotation simultaneously to alleviate the cumulative errors. According with the aim, the optimal formulas also don’t require \( A,B \) satisfying rigid transformation or equivalent rotational angle.

3.1 Objective Function of Optimization

Obviously, the objective function is

\[
L(X) = \min_{i} \|A(i)X - XB(i)\|_2
\]

which is equivalent to the following:

\[
L(X) = \min_{i} \left\| \left[ (R_y(i) \otimes I_3 - I_3 \otimes R_y(i)) \text{vec}(R_y) \right] + \right\|_2
\]

Let
\[ F(i) = \left( R_y(i) \otimes I_3 - I_3 \otimes R_y(i) \right) \text{vec}(R_y) \]  
\[ G(i) = \left( R_y(i) - I_3 \right) T_x - R_y T_y(i) + T_x \]  
Since \( X \) must be rigid transformation, \( X \) consists of the rotation \( c = [c_1, c_2, c_3]^T \) and the translation \( t = [t_1, t_2, t_3]^T \). Then the final optimal function is

\[ L(c, t) = \min \sum_i \left[ \|F(i)\|^2 + \|G(i)\|^2 \right] \]  

### 3.2 Jacobian Formula

Through some deductions, the Jacobian formula for the objective function (10) is:

\[ J(i) = \begin{bmatrix} \frac{dF(i)}{dc} & \frac{dF(i)}{dt} \\ \frac{dG(i)}{dc} & \frac{dG(i)}{dt} \end{bmatrix} = \begin{bmatrix} \frac{dF(i)}{dvec(x)} & \frac{dvec(x)}{dc} \\ \frac{dG(i)}{dvec(x)} & \frac{dvec(x)}{dt} \end{bmatrix} \]  

where the initial translation \( t_0 \) is the solution of (2), and the initial rotation \( c_0 \) is RPY-angle corresponding to the initial rotation. If the derivation of RPY-angle is ill-conditioned badly, though singularly, By Rodrigues formula, yet you can obtain a more robust RPY-angle.

### 3.3 Optimization

Multi-equations can be represented as follows:

\[ H = \begin{bmatrix} F(1) \\ G(1) \\ F(2) \\ G(2) \\ \vdots \end{bmatrix}, \quad J = \begin{bmatrix} J(1) \\ J(2) \\ \cdots \end{bmatrix}, \quad \chi = \begin{bmatrix} c \\ t \end{bmatrix} \]  

Then the iterative formula for optimization is

\[ J\Delta \chi = -H \]  
\[ \chi^{(n+1)} = \chi^{(n)} + \Delta \chi \]  

where \( \Delta \chi \) can be solved by the quickly algorithm QR or more precise but slowly SVD. Here you can also take the more robust but slower Levenberg-Marquet as the iterative formula:

\[ (J^TJ + \mu I) \Delta \chi = -J^TH \]  

### 3.4 Brief Summary

The above optimization also doesn’t require \( A, B \) satisfying the rigid transformation. Now we conclude the solution steps as follows:

1. **Solution for initial value**
   - Implement the scheme in section 2.3, then solve for the RPY-angles of the initial rotation and take the solved RPY-angles and the translation as the initial value.
2. **Optimization for final value**
   1. Compute Jacobian formula (11) for \( J \);
   2. Compute (8–9) for \( F(i), G(i) \) and acquire \( H \);
   3. Use (13)(14) or (15)(14) to implement the iterative optimization.

### 4 Simulation

We have implemented plenty of simulations to test our approach. Simulation conditions are like the conditions in article [14]. For simulation on solution of initial value has been noted in article [14] by Ronghua Liang, et al., Here we will demonstrate the simulation on solution of final optimal value.

![Figure 1: The residual norm comparison vs. num. of equ.](image)

**4.1 Analysis of Simulation**

1. As more equations are added, the initial value converges to the final value gradually. It indicates that as there are enough equations, it is credible to regard the initial value as the final value.
2. When few equations, the solution is not robust and is sensitive to noise, however, as more equations add, the calibration result will converge stability quickly, even when the noise level is much high.

### 5 Real Data Implementation

The real experimental environment is shown in figure 2. From the theoretical analysis, three images from different orientations are enough to do the calibration, but more images may ensure the robustness of the computation. In real experiments we use the two-stage methods to calibrate the robot vision quickly, that is, to let the robot arm grasp a grid template to motion several times as star-shaped track in front of the surveillant camera to complete calibration. Firstly, calibrate the camera with the planar-pattern-calibration [15] and determine the rigid transformation \( H_{bt} \) from the planar pattern to the camera, and use robot joint angles to get the \( H_{gb} \).

With the solved \( H_{bt} \) and \( H_{gb} \), we can acquire multi-equations of that form \( AX=XB \) as Ronghua Liang, et al. have done in [14], then we can solve the
equations for the initial value and the final optimal value by our approach.

We have implemented many real experiments with our algorithm and the classic Zhuang’s algorithm, and obtained similar results. The following figures are the results of one real experiment. In the experiment, the robot hand-eye system captured one image as the robot moved along the star-shaped track to one pose and the number of available images is up to 31, so there are 30 equations for the calibration. Now we’ll use the 30 equations to solve for the poses from pattern to gripper ($H_{tg}$).

### 5.1 Real Experimental Results

1. Solve for the initial pose $H_{tg}$ with two algorithms, respectively.

   ![Figure 2: Shot in robot hand-eye calibration](image)

   ![Figure 3: A planar-pattern image for the calibration](image)

2. Solve for final optimal pose $H_{tg}$ with our algorithm

   ![Figure 4: Comparison of errors in initial value between two algorithms](image)

   ![Figure 5: The residual norm comparison vs. num. of equ.](image)

### 5.2 Analysis of Real Experiment

1. As in simulation, the calibration results converge stability as the equations or calibration images increase, more than 3 equations or 4 images are necessary, otherwise the errors will be big, as seen from figure 4.

2. Comparisons indicate the mean errors of our approach are generally a little smaller than of the classic one, and the same case are the standard derivations.

3. Figure 5 shows the general tendency that the initial value gradually approaches the final value as the number of equations increases, so it is available to directly take the initial value as the final result when there are enough equations.

### 6 Conclusions

The issue of robot hand-eye calibration is to solve $AX=XB$. We present our approach. Compared with the classic approach, it has the following characteristics:

1. In solution of the initial value, our approach is based on directly linear method, and doesn’t need to concern on the problem whether $A, B$ satisfy the rigid transformation, in addition, in the following Jacobian optimization, it needn’t concern on the problem, too. That is to say our approach is fit for the case of non-rigidity of $A, B$.
Whole computations in our approach are closed in real field, which ensures our technique robust in theory. Simulations and real data implementations indicate that:

1. For computation of the initial value, the mean errors of our approach are generally a little smaller than of classic one, so are the standard derivations, which means our approach is more robust and precise.

2. Optimization will improve the solution, especially when only a few equations, optimization is prerequisite.

3. With the equations increasing, the initial value converges to the final value gradually, and the computation is stable, when a number of equations, it is credible to regard the initial value as the final value when there are enough equations.

References:


