A Direct Linear Solution With Jacobian Optimization to AX=XB for Hand-Eye Calibration[1]

MAO JIANFEI, TIAN QING, LIANG RONGHUA
Institute of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, 310032
China
(mjf, qingtian, rhliang) @zjut.edu.cn

Abstract: A linear closed-form solution followed by Jacobian optimization is proposed to solve AX=XB for hand-eye calibration. Our approach does not require A,B satisfying rigid transformation rather than the classic ones based on quaternion algebra or screw rule. We firstly give the detailed proof of the optimal orthonormal estimation for an arbitrary scale matrix. With the theorem, a linear closed solution based on singular value decomposition (SVD) and the rule of optimal rotation estimation, is presented, followed by the nonlinear optimization with the proposed Jacobian recursive formula. Detailed deduction and demonstration are given based on matrix theory. Since our approach is applicable for non-rigid transformation rather than the classic ones, our technique is more flexible. Plenty of computer simulation and real data implementation indicate that: (1) In computation of initial value, our technique has higher precision and more 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: Closed-form solution, Jacobian optimization, Hand-eye calibration, Orthonormal estimation, SVD

1 Introduction

Robots are in use throughout industry [1,2] and 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 [3,4]. Hand-eye calibration problem, 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 [5~8] were proposed in the past to solve for X as well as a nonlinear optimization method. Tsai [5] and Shiu [6] 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 [8] and are self-integrated in theory, but their deduction procedures are quite complicated. To overcome this deficiency, Zhuang *et al*. [7] used quaternions to solve the equation and simplified the problem. Chou and Kamel [9] presented the form-closed solution based on quaternions, subsequently, horaud and Dornaika [10] implement the optimization to their solution. Zhao [11] employed the screw theory with quaternions to solve the calibration problem.

However, the forementioned algorithms have a common limitation that they all require A, B satisfying rigid transformation, for they are based on quaternions or screw theory. For example, Tsai's algorithm [5] and Zhuang's algorithm [7] are well known for the precision and robustness, and cited in many textbooks and literatures, we call their algorithms generally as classic approaches in our paper. Although Tsai's algorithm is different from Zhuang's in theory, their solving formulas are approximately the same. In classic algorithms, also called two-stage algorithm, the first step is to solve the rotational axis of X linearly according to the rotational axes or quaternion vectors of A, B, and the next step is to apply the result in the first step to solve the translation vector linearly. This algorithm has a main deficiency that rotation estimation errors at first step can propagate to the translation computation directly. Zhuang [7] hence proposed

1 Supported by the National Natural Science Foundation of China under Grant No. 60703002; the Natural Science Foundation of Zhejiang Province of China under Grant No. Y1090335; key projects from Science and Technology Department of Zhejiang Province under Grant No. 2007C11022 and No. 2009C21008

one-stage algorithm to implement a non-linear optimization to alleviate the drawback. Not only is this algorithm less sensitive to noise, furthermore, this algorithm is easy to understand and use. With synthetical analysis, we think the classic algorithms have complete theory analysises and deductions based on quaternions or screw theory, and solve for the initial value linearly, However, the main disadvantages for the classic algorithms are two fold: (1) The required conditions are too rigorous, that is, to use 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) The classic algorithms doesn't consider the condition of 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 will cause the classic algorithms comparative big errors.

We propose a new 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 doesn't need to 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, 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 derivate the iterative Jacobian formula, and implement the non-linear optimization with Gauss-Newton method or Levenberg-Marquet method. Our technique has complete theoretical basis with rigorous demonstrations and deductions, in addition, the theoretical part, lemmas and formulas of the approach are a good reference for the other studies or applications. 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.

2.1 About Hand-Eye Calibration

In general hand-eye system, the sensor is mounted on one joint of robot. Though sometimes sensor is not mounted onto the robot, but fixed above the workbench unmovable, yet it can also be seen as hand-eye system, for this case can be seen as the sensor is mounted onto the robot base joint. Now we'll take this case as the following analysis, and the other cases can be analyzed similarly. The figure shows the case that robot take the planar move in front of the cameras to do the calibration.

Fig. 1. hand-eye calibration

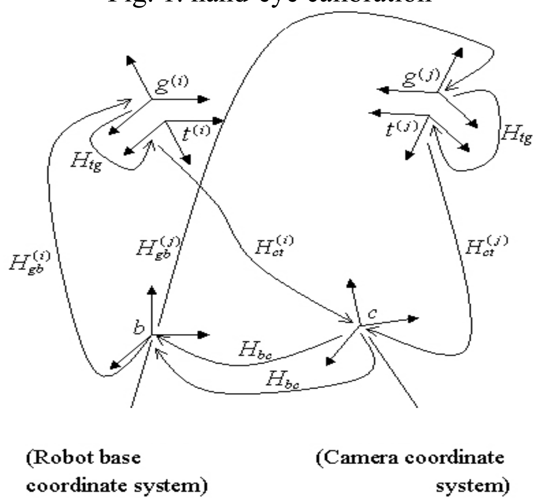


Fig. 2. various coordinate transformations

Here, $c\text{->}b\text{->}g^{(i)}\text{->}t^{(i)}\text{->}c$ and $c\text{->}b\text{->}g^{(j)}\text{->}t^{(j)}\text{->}c$ form a closed loop separately, then we'll have the following identity [12]:

$$[H_{gb}^{(i)}]^{-1} \bullet H_{gb}^{(j)} \bullet H_{tg} = H_{tg} \bullet [H_{tc}^{(i)}]^{-1} \bullet H_{tc}^{(j)} \quad (1)$$

(1) can be wrote as: $C^{ij} H_{tg} = H_{tg} D^{ij}$ (2)

similarly get the other equation:

$$E^{ij} H_{cb} = H_{cb} F^{ij} \quad (3)$$

where H_{tg} and H_{cb} are also called poses from pattern to robot gripper and from camera to robot base, respectively.

So, to solve for H_{cb} or H_{tg} is equavilent to solve $AX=XB$ for X.

Though A,B,X might not satisfy rigid transformation, they are all homogeneous matrices, and can be denoted as:

$$A = \begin{pmatrix} R_a & T_a \\ 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} R_b & T_b \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} R_x & T_x \\ 0 & 1 \end{pmatrix} \quad (4)$$

Then $AX=XB$ can be represented as:

$$R_a R_x = R_x R_b \quad (5)$$
$$(R_a - I) T_x = R_x T_b - T_a \quad (6)$$

To solve $AX=XB$ is equivalent to solve (5)(6).

2.2 About Hand-Eye Calibration

General linear algorithms take $R_a R_x$ as rotational matrix and assume their rotational angles equal. On the contrary, our algorithm can be widely applicable for general cases, it doesn't require $R_a R_x$ satisfying rotational transformation.

Lemma 1: Equation (5) can represented as:

$$(\boldsymbol{R_a} \otimes \boldsymbol{I_3} - \boldsymbol{I_3} \otimes \boldsymbol{R_b}^*)\ \mathbf{vec}(\boldsymbol{R_x}) = 0 \quad \mathbf{(7)}$$

Proof: Use kronecker product to Expand (5):

$$(5) \Leftrightarrow R_a R_x I_3 - I_3 R_x R_b = 0 \Leftrightarrow (7)$$

Where **vec** is an operator, which stretches a matrix as row's direction, then transpose of it, $\otimes$ is kronecker product operator. The general form of (7) is the equation AX=0, and the solution of AX=0 is equivalent to the identical norm least square solution to the equation.

Lemma 2: The identical norm least square solution for AX=0 is the identical eigenvector corresponding to the minimal eigenvalue of $A*A$.

Proof: The identical norm least square solution is:

$$\left\{X \mid \min\left(\|AX\|_2\right) : \|X\|_2 = 1\right\} \quad (8)$$

Use Lagrange's formula to transform the above constrained problem into the unconstrained:

$$f(X) = X^*\left(A^*A\right)X + \lambda\left(X^*X - 1\right) \quad (9)$$

when $\frac{\partial f}{\partial X} = 2\left(A^*A\right)X + 2\lambda X = 0$, there exists extremum for $f(X)$, here, $-\lambda$ is the eigenvalue of A^*A and X is the eigenvector of A^*A.

Decompose A with SVD: $A=USV^*$, $A^*A=VSSV^*$, where U,V are all unitary matrices. Let $\boldsymbol{V}=[v_1, v_2, \ldots]$, **diag**($\boldsymbol{SS}$)$=[s_1, s_2, \ldots]$, where **diag** is a set of the diagonal elements of a matrix.

So: $-\lambda \in \{s_1, s_2, \ldots\}$, $X \in \{v_1, v_2, \ldots\}$

For A^*A is positive or semi-positive definite, its eigenvalue and eigenvector satisfy: $A^*Av_i = v_i s_i$

When get extremum, X is a identical eigenvector of A^*A, let $X=v_i$ and substitute into (8):

$$\left\{v_i \mid \min\left[(v_i^* A^* A v_i)^{\frac{1}{2}},\ i = 1, 2\cdots\right]\right\} =$$
$$\left\{v_i \mid \min\left[(v_i^* v_i s_i)^{\frac{1}{2}},\ i = 1, 2\cdots\right]\right\} = \left\{v_i \mid \min\left[\sqrt{s_i},\ i = 1, 2\cdots\right]\right\}$$

Let $\sqrt{s_k}$ represent the minimal element of the set $\{\sqrt{s_1}, \sqrt{s_2}, \ldots\}$, then $\sqrt{s_k}$ is the minimal singular value of A, yet is the square root of the minmal eigenvalue of $A*A$, and the corresponding eigenvector v_k can minimize $\|AX\|$.

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

Proof: Let A be an arbitrary real square matrix, λ_i be a real eigenvalue of A and u_i be the corresponding eigenvector. Since A, λ_i are all in real field, then

$$(A-\lambda_i)u_i = 0 \Leftrightarrow (A-\lambda_i)\ \mathrm{Re}(u_i) = 0 \text{ and } (A-\lambda_i)\ \mathrm{Im}(u_i) = 0 \quad (10)$$

where **Re, Im** are operators separately exacting the real part and imaginary part from a complex matrix, and the extracted matrices are all real matrices, yet the same size as the complex matrix.

The above equations indicate that the basis of the kernel of $A-\lambda_i$ in complex field is equivalent to the basis of kernel in real field. Then the eigenvector of $A-\lambda_i$ in complex field can be completely represented by the eigenvector in real field, and also corresponding to the same eigenvalue λ_i.

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 $\boldsymbol{U},\boldsymbol{V}$ got by SVD can belong to real field.

Proof: Decompose A by SVD:

$$A=USV^* \Rightarrow (AV)^*(AV)=SS=D \Rightarrow (A^*A)V = VD$$

Where D is a diagonal matrix and the diagonal element set are $\{d_1, d_2, \ldots\}$, U is composed of vectors $\{u_1, u_2, \ldots\}$, and V is composed of eigenvectors $\{v_1, v_2, \ldots\}$ of A^*A. Since A is a real matrix, A^*A is real semi-positive definite or real positive definite, then it is diagonalizable and all its eigenvalues $\{d_1, d_2, \ldots\}$ are always greater than and equal to zero, and from Lemma 3, $\{v_1, v_2, \ldots\}$ can all belong to real field, so V is a real matrix.

With SVD, $u_j = \sqrt{d_j}\,(AV)^{(j)}$. because d_j>0 and A,V belong real field, u_j also belongs to real field. Since Hilbert space is complete for real field, so the orthonormal system of Hilbert space, from the above U is sure to be found in real field. Then we must get real U,V from SVD of a real matrix.

Lemma 5: By SVD of nonzero matrix $\boldsymbol{Q}$: $\boldsymbol{Q}=\boldsymbol{USV}$*, then the orthonormal matrix most approximated to a scale matrix $\boldsymbol{\lambda Q}$, is determined by $\boldsymbol{R}=\pm\boldsymbol{UV}$*, and here, $\lambda = \pm\frac{\mathrm{tr}(S)}{\mathrm{tr}(QQ^*)}$, and operators +,- between the two expressions are one-to-one correspondence.

Proof: the optimal estimation should satisfy:

$$R=\{R : \min\|\lambda Q - R\|_F\} \quad (11)$$

$$\|\lambda B - R\|_F = \mathrm{tr}[(\lambda Q - R)^*(\lambda Q - R)]$$
$$= 3 + \mathrm{tr}(QQ^*)\left[\lambda - \frac{\mathrm{tr}(Q^*R)}{\mathrm{tr}(QQ^*)}\right]^2 - \frac{\mathrm{tr}^2(Q^*R)}{\mathrm{tr}(QQ^*)} \geq 0 \quad (12)$$

Since $\mathrm{tr}(Q^*Q)$>0 and (11)$\geq$0, to minimize (11), the following must be satisfied.

$$\lambda = \frac{\mathrm{tr}(Q^*R)}{\mathrm{tr}(QQ^*)} \text{ and } \mathrm{Max}\frac{\mathrm{tr}^2(Q^*R)}{\mathrm{tr}(QQ^*)} \quad (13)$$

Then:

$$(11)=\{R: \max\left[\mathrm{tr}^2(Q^*R)\right]\}=\{R : \max[\mathrm{tr}^2(ZS)]\} \quad (14)$$

where $Z=U^*RV$ and Z is an orthonormal matrix. Let Z_i be the ith column of Z, Z_{ii} be the diagonal element in the ith column. since $\| Z_i \|=1$, $|Z_{ii}|\leq 1$.

$$\mathrm{tr}^2(ZS)=\left(\sum_i^n Z_{ii}S_{ii}\right)^2 \leq \left(\sum_i^n |Z_{ii}|S_{ii}\right)^2 \leq \left(\pm\sum_i^n S_{ii}\right)^2 \quad (15)$$

To maximize the above, Z_{ii} must be ± 1, thereby, $Z=\pm I$, then$\pm I=Z=U^*RV \Rightarrow R=\pm UV^*$, and substitute R into (13), we will acquire:

$$\lambda=\pm\frac{\mathrm{tr}(S)}{\mathrm{tr}(QQ^*)}.$$

With lemma 5 we can acquire the orthonormal matrix, but a 3×3 orthonormal matrix may not be a rotational matrix, the following will elaborate the estimation of a rotational matrix based on lemma 5.

Definition 1: For any odd real square matrix, define:

$$|A|=\begin{cases} A & \det(A)>=0 \\ -A & \det(A)<0 \end{cases} \quad (16)$$

From the above, det($|A|$) is always greater than or equal to zero.

Lemma 6: By SVD of nonzero real matrix $Q_{3\times3}$: $Q=USV^*$, then the rotational matrix most approximated to a scale matrix λQ, is exclusively determined by $R=|UV^*|$.

Proof: As a rotational matrix in right hand coordinate system, R must be orthonormal and satisfy: $r_3=r_1\times r_2$, where r_i are the *ith* column vector of R, thereby, $\det(R) = r_3\cdot(r_1\times r_2) = 1$.Then That R is orthonormal and $\det(R)=1$ are equivalent to that R is rotational. Once $R\in\{\pm UV^*\}$ and $\det(R)=1$, R must be the optimal rotational matrix. Now we'll prove '$R\in\{\pm UV^*\}$ and $\det(R)=1 \Leftrightarrow R=|UV^*|$'.

Since $B_{3\times3}$ is a real matrix, from lemma 4, both U,V can belong to real field, then $\det(UV^*)$ is real. Then:

$I=(UV^*)(UV^*)^* \Rightarrow 1=\det^2[(UV^*)] \Rightarrow \det(UV^*)=\pm 1$ (17)

For both U, V are 3×3 square matrix, $-\det(UV^*)=\det(-UV^*)$:

$$(17) \Leftrightarrow \det(\pm UV^*)=1 \Leftrightarrow \det(|UV^*|)=1 \quad (18)$$

From (18), only $R=|UV^*|$, then $\det(R)=1$, thereby R is rotational, and $R\in\{\pm UV^*\}$ or $\{\pm|UV^*|\}$, for $\{\pm UV^*\}=\{\pm|UV^*|\}$. Synthetically, the optimal rotational matrix R is exclusive.

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 A,B are rotational, Zhuang and Shui have proved the lemma in detail. Moreover, since (10) is linear, obviously, it is robust to use multi-equations to solve. However, A,B might not be rotational, through lemma 1 in book[13], the sufficient and necessary condition that there exists solution is that: some eigenvalue of A is equal to some eigenvalue of B.

2.3 Linear Solution to the translation part

By Lemma of Zhuang, if R_a is rotational, it requires at least two consistent equations as (6) to solve for T_x, and multi-equations will enhance the robustness of solution. Obviously, QR or SVD, which are all good linear algorithms, can be used to solve (6) for T_x.

2.4 Brief Summary

Lemma 3~4 are the basis of the other lemmas, and of our algorithms, as they ensure the algorithms are closed in real field. Since in calibration A,B are from various coordianate transformations, A,B are physically in real field. The other lemmas provide the necessary theoretical support for estimation of the optimal rotational matrix, yet provide the detailed resolving apporoch, and the resolving steps in computer are given as follows. Note that multi-equations will ensure the solution robust.

(1) According to lemma 1~2, solve (7) 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 λQ is: R=|UV*|
(3) Solve (6) for the translation part with QR or SVD.

3 Jacobian Optimization

In this section we will establish the objective model of optimization and elaborate the derivation of the Jacobian formula, finally, the executive steps of the algorithm will be given.

The above steps in section 2 has the same deficiency as the classic approaches, for the rotation errors will propogate to the following solution for the translation part, and they are amplified by $\|T_B\|$ approximately via estimation. It is necessary to implement an optimization solving the translation and rotation simultaneously to alleviate this cumulative errors. According with the aim of our technique, the following objective function and Jacobian optimal formula also don't require A,B satisfying rigid transformation or equivalent rotational angle.

3.1 Objective Function of Optimization

Obviously, the objective function is:

$$L(X)=\min\sum_i \|A(i)X-XB(i)\|_F \quad (19)$$

Where i represents the sequence number of equations as $AX=XB$ *and* which is equivalent to the combination of (10) and (9):

$$L(X)=\min\sum_i\left[\begin{array}{l}\left\|\left(R_a(i)\otimes I_3-I_3\otimes R_b^*(i)\right)\mathrm{vec}(R_x)\right\|^2+\\ \left\|\left(R_a(i)-I_3\right)T_x-R_xT_b(i)+T_a(i)\right\|^2\end{array}\right] \quad (20)$$

Since rigid, X consists of the rotation and translation, of which the rotation can be determined enough by three angles such as RPY-angles. Therefor, six parameters is enough to determine X, which can be represented by the rotation $\mathbf{c}=[c_1, c_2, c_3]^{\mathrm{T}}$ and the translation $\mathbf{t}=[t_1, t_2, t_3]^{\mathrm{T}}$ separately. Let:

$$F(i)=\left(R_a(i)\otimes I_3-I_3\otimes R_b^*(i)\right)\mathrm{vec}(R_x) \quad (21)$$

$$G(i)=\left(R_a(i)-I_3\right)T_x-R_xT_b(i)+T_a(i) \quad (22)$$

$F(i)$,$G(i)$ are vectors of 9×1 and 3×1 separately, then the final function of optimization is:

$$L(\mathbf{c}\,,\mathbf{t}\,)=\min\sum_i\left[\|F(i)\|^2+\|G(i)\|^2\right] \quad (23)$$

3.2 Jacobian Formula

The Jacobian formula for the objective function (23) is:

$$\boldsymbol{J(i)}=\begin{bmatrix}\dfrac{dF(i)}{d\mathbf{c}^{\mathrm{T}}} & \dfrac{dF(i)}{d\mathbf{t}^{\mathrm{T}}}\\ \dfrac{dG(i)}{d\mathbf{c}^{\mathrm{T}}} & \dfrac{dG(i)}{d\mathbf{t}^{\mathrm{T}}}\end{bmatrix}=\begin{bmatrix}\dfrac{dF(i)}{d\mathrm{vec(x)}^{\mathrm{T}}}\dfrac{d\mathrm{vec(x)}}{d\mathbf{c}^{\mathrm{T}}} & 0_{9\times3}\\ \dfrac{dG(i)}{d\mathrm{vec(x)}^{\mathrm{T}}}\dfrac{d\mathrm{vec(x)}}{d\mathbf{c}^{\mathrm{T}}} & \dfrac{dG(i)}{d\mathbf{t}^{\mathrm{T}}}\end{bmatrix} \quad (24)$$

where the initial translation $\mathbf{t_0}$ is the solution of (6) clearly, and the initial rotation $\mathbf{c_0}$ 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 such as (21~22) can be represented as follows:

$$H=\begin{bmatrix}F(1)\\G(1)\\F(2)\\G(2)\\\cdots\end{bmatrix},\quad J=\begin{bmatrix}J(1)\\J(2)\\\cdots\end{bmatrix}\quad \chi=\begin{bmatrix}\mathbf{c}\\\mathbf{t}\end{bmatrix} \quad (25)$$

then the iterative formula for optimization:

$$J\vartriangle\chi=\text{-}H \quad (26)$$

$$\chi^{(n+1)}=\chi^{(n)}+\Delta\chi \quad (27)$$

where $\Delta\chi$ can be solved by the quickly algorithm QR or more precise but slowly SVD. According to (24), you can also take the more robust but slower Levenberg-Marquet as the iterative formula:

$$\left(J^{\mathrm{T}}J+\mu I\right)\vartriangle\chi=\text{-}J^{\mathrm{T}}H \quad (28)$$

3.4 Brief Summary

The above algorithm for optimization also doesn't require A,B satisfying the rigid transformation, and can solve the translation part and rotational part of X simultaneously, so it alleviates the cumulative errors. Now we'll conclude the steps as follows:

(1) Solution for initial value

As described in section 2.4, the initial rotation and translation are determined in turn. Then solve the initial rotation for the RPY-angles and take the solved RPY-angles and the translation as the initial value.

(2) Optimization

1. Solve Jacobian formula (20) for $\boldsymbol{J}$;
2. Solve (21~22) for $F(i)$,$G(i)$ and obtain H;
3. Use (26)(27) or (28)(27) to implement the iterative optimization.

4 Simulation

To compare our approach with classic algorithm, we assume A,B to be rigid. We firstly generate X and A randomly, then B according to formula $B=X^{-1}AX$, then we add white noise to A,B to generate all noised A,B, which are the data of the equations in simulation really.

The simulation principals: 1. Simulation data should cover all the area of all the probably real data; 2. Simulation data should be distributed uniformly in whole data space. According to the principals, we randomly generate 100 X in the supposed solution space, and 25 pairs of A,B for each X , as add 21 levels of uniformly distributed noises to A,B, here, the noise levels are [0,0.05,0.1,…,1] in turn.

Various Error Statistics

$$\mathrm{avg}(F)=\frac{1}{mn}\sum_{i=1}^{mn}\|F(i)\| \quad (29)$$

$$\mathrm{avg}(G)=\frac{1}{mn}\sum_{i=1}^{mn}\|G(i)\| \quad (30)$$

$$\mathrm{std}(F)=\frac{1}{mn-1}\sum_{i=1}^{mn}\left(\|F(i)\|-\mathrm{avg}(F)\right)^2 \quad (31)$$

$$\mathrm{std}(G)=\frac{1}{mn-1}\sum_{i=1}^{mn}\left(\|G(i)\|-\mathrm{avg}(G)\right)^2 \quad (32)$$

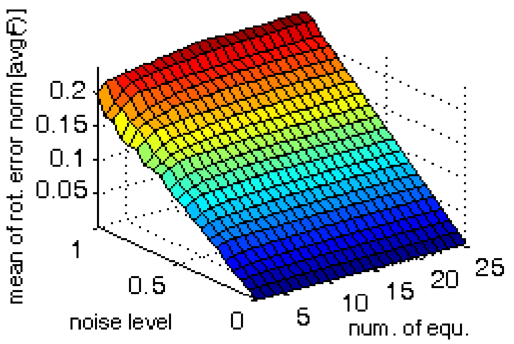
$$\|H\|_{\mathrm{m}}=\left[\frac{1}{mn}\sum_i^{mn}\left(\|F(i)\|^2+\|G(i)\|^2\right)\right]^{\frac{1}{2}} \quad (32)$$

where m is the number of X, n is the number of equations for one X, so there are m×n pairs of || $F(i)$ ||,|| $G(i)$ || for (21~22), corresponding to the above five statistics separately named mean of rotation error norm, mean of translation error norm, std of rotation error norm, std of translation error norm, and mean residual norm in turn, where std is the abbreviation of standard derivation.

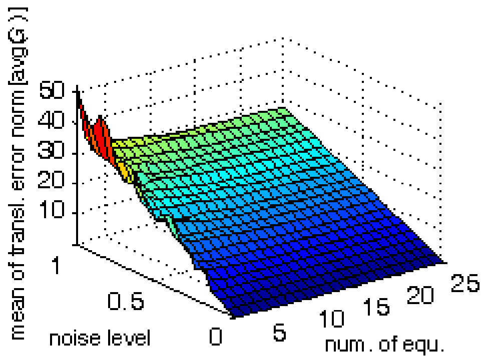
We have implemented plenty of simulations to test our approach, and also compared the technique with the classic technique, and we get the similar results as follows.

4.1 Simulation for Initial Value

(1) Simulation error with the proposed algorithm on various number of equations



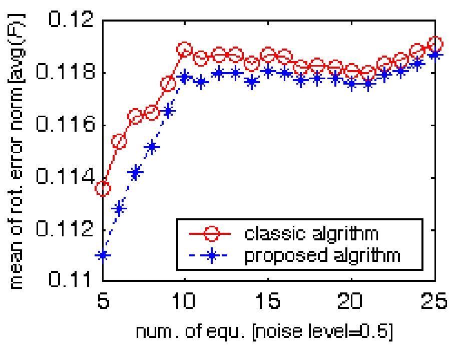
(a) analysis of Rotation error



(b) analysis of Translation error

Fig. 3. The error in relation to the noise level and number of equations

(2) Error between the proposed and the classic on various number of equations



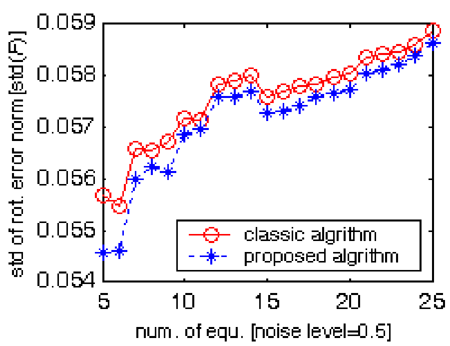
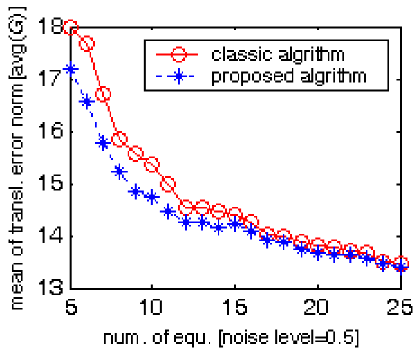


Fig. 4 (a) Rotation error in comparison with the classic algorithm on noise level 0.5



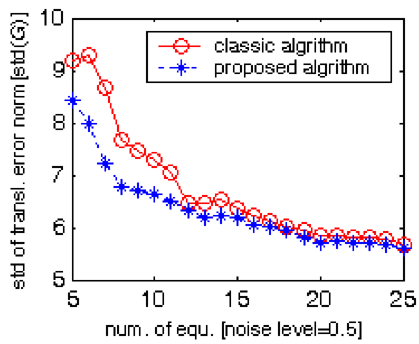
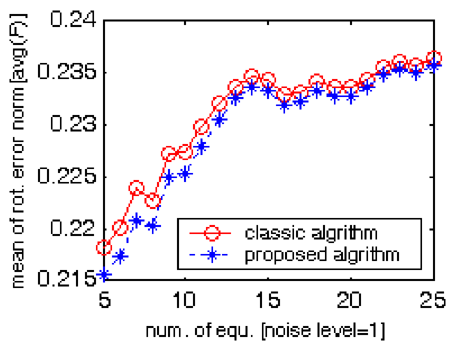


Fig. 4 (b) Translation error in comparison with the classic algorithm on noise level 0.5



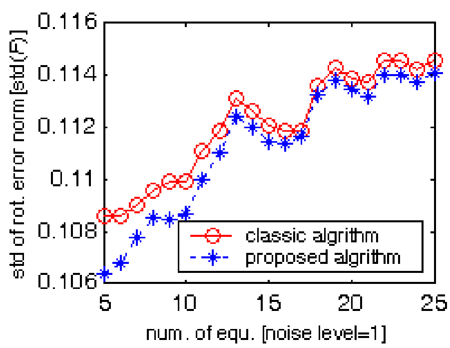


Fig. 4 (c) Rotation error in comparison with the classic algorithm on noise level 1

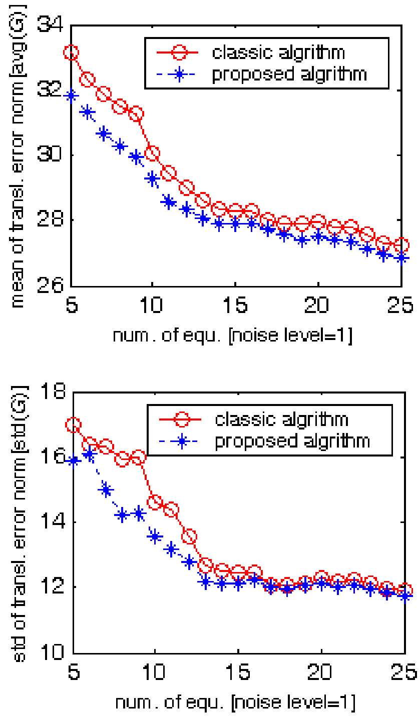


Fig. 4 (d) Translation error in comparison with the classic algorithm on noise level 1

4.2 Simulation for Final Optimal Value

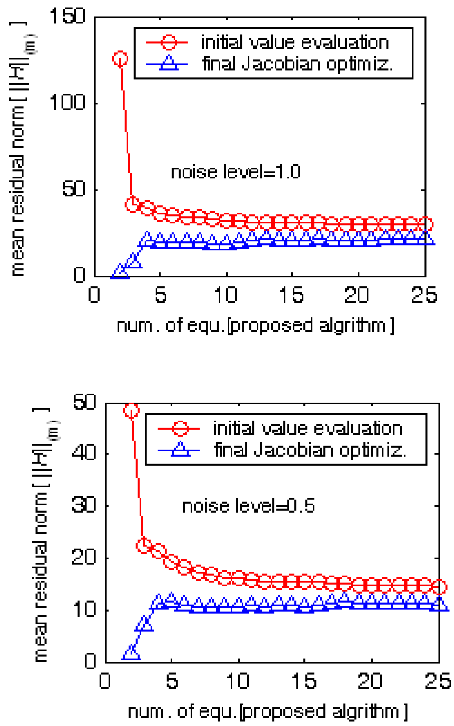


Fig. 5. The residual norm comparison for the proposed algorithm

4.3 Analysis of Simulation

We have implemented plenty of simulations, and obtained similar results as above. We find:

(1) When the number of equations is few, both of approaches have comparatively big error, but the error of the classic approach is still much bigger than of ours.

(2) Figure 3 shows a plane bevel when the number of equations is greater than 3, which indicates that the error is proportional to the noise level in the condition. It seems very interested, and the reason will be elaborated in another paper.

(3) Figure 3 also indicates that when the number of equations is few the solution is not robust and is sensitive to noise, however, as the number increases, the calibration result will converge stability quickly, even when the noise level is much high, and it shows our approach very robust.

(4) Simulation Comparison shows that the average error of our approach, as well as the error oscillation, is generally smaller than of the classic one, which shows our approach is more robust, as can be seen from the comparison of std(F),std(G) in above figures.

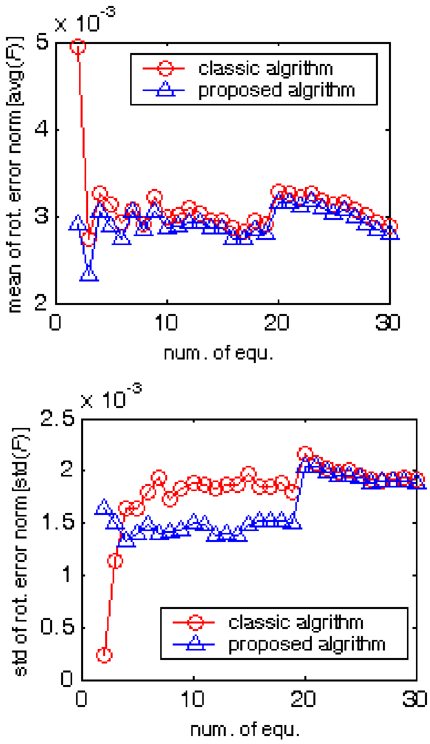
5 Real Data Implementation

The real experimental environment is showed in figure 1. From the theoretical analysis, three images from different orientations are enough to do the calibration, but more images will ensure the robustness of the computation. For the calibration of the camera, Yet there are some fast linear methods [14], but we would rather use the planar-pattern-calibration [15] to acquire more precision, moreover, we can determine the rigid transformation H_{tc} from the planar pattern to the camera, then we can use robot joint angle to calculate the H_{gb}, then with H_{tc} and H_{gb}, use formula (1) to calculate C,D, similarly, calculate E,F, then use our technique to solve (2)(3) for H_{tg} and H_{cb} respectively.

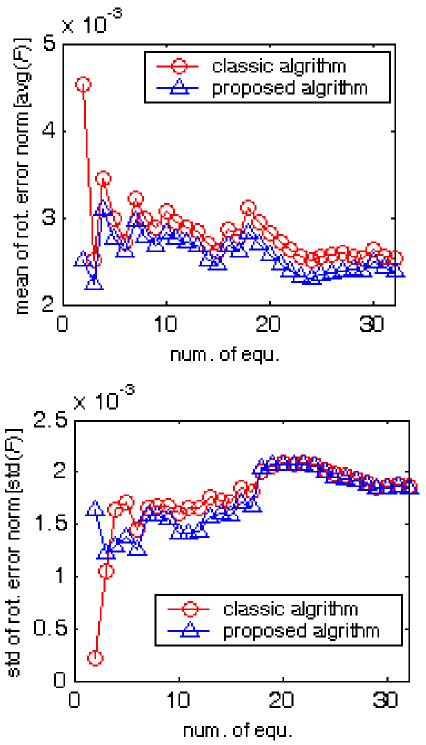
We have implement many real experiments, and obtained similar results, too. The following is the result of one real experiment. In the experiment, we use binocular cameras to do calibration. For left camera, number of the available images is 31, for right camera, is 33, so there are 30 equations for left camera, 32 equations for right camera. Now we'll use these 30 equations and 33 equations respectively to solve for the poses from pattern to gripper (H_{tg}), theoretically, H_{tg} obtained through two groups of equations respectively should be equal, but for various errors, the practical results will deviate from each other. In the following section we'll show the convergence degree of these two H_{tg}, as it also reveals the precise degree of our calibration technique.

5.1 Real Experimental Results

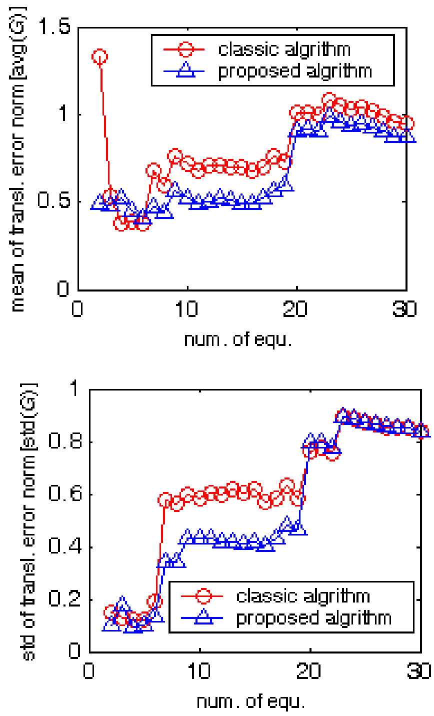
(1) Solution to the initial poses from pattern to gripper (H_{tg}) via two cameras respectively



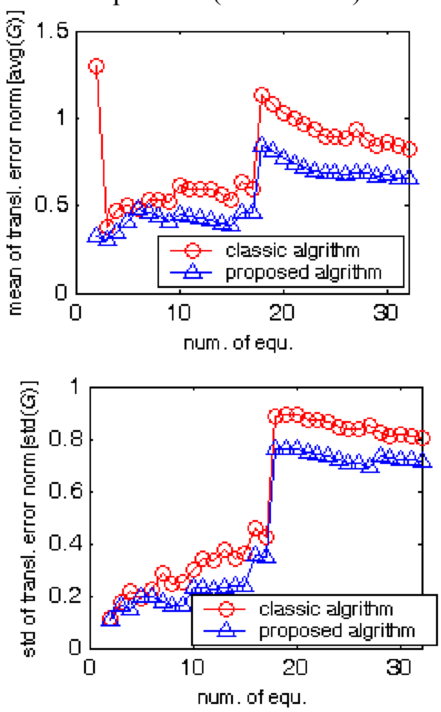
(a) Comparison of rotation error vs. the number of equations (left camera)



(b) Comparison of rotation error vs. the number of equations (right camera)



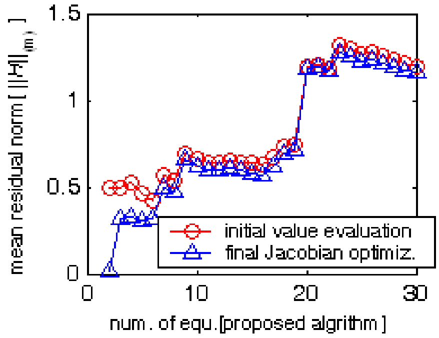
(c) Comparison of translation error vs. the number of equations (left camera)



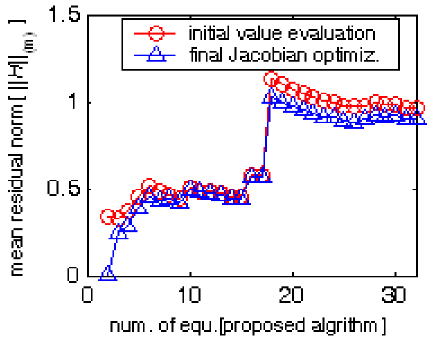
(d) Comparison of translation error vs. the number of equations (right camera)

Fig. 6. Comparison of errors in initial value computation between two algorithms (right camera)

(2) Solution of final optimal pose from pattern to gripper (H_{tg}) via two cameras respectively



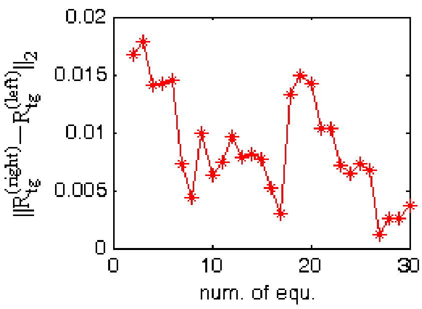
(a) The residual norm comparison (left camera)



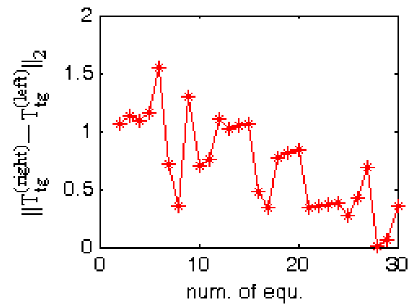
(b) The residual norm comparison (right camera)

Fig. 7. The residual norm comparison between initial value and final value vs. the number of equations

(3) The convergence degree of two H_{tg} solved via two cameras respectively for our algorithm



(a) Comparison of rotation



(b) Comparison of translation

Fig. 8. The comparison of the final optimal H_{tg} solved via two cameras respectively

Then the final H_{tg} that the pose from pattern to gripper can be calculated as:

$$H_{tg} = \frac{H_{tg}^{(\text{right})} + H_{tg}^{(\text{left})}}{2}$$

5.2 Analysis of Real Experiment

(1) As that 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 6.

(2) Comparisons indicates the mean errors of our approach are generally smaller than of the classic one, and the same case the error oscillation. It reveals our approach is more robust.

(3) Figure 7 shows the general tendency that the initial value gradually approaches the final value as the number of equations increases, so it is available directly taking the initial value as the final result for the not very precise application.

(4) The comparison also indicates it is necessary to implement optimization when the equations are few.

(5) Figure 8 shows the calibration results (H_{tg}) for the two cameras converges gradually as the number of equations increases, which reveals more equations help to increasing the precision and robustness of calibration.

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 decompose matrices A,B to get the screw vectors, therefore, 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 situation of non-rigidness of A,B.

(2) After solution of (7) with SVD, the following step is to estimate the optimal rotational matrix, we firstly give proof of the optimal orthonormal estimaion of any rank matrix, then elaborate the proof of the optimal rotational matrix. Moreover, we also give the proof of that all computations in our approach is closed in real field, which ensures our technique robust in theory.

(3) Our approach is based on rigorous deduction and integral theory, furthermore, the theoretical part, derivation and formula of the approach are

good references for the other study and application.

Simulation and real data implementation show that:

(1) For computation of the initial value, the mean errors of our approach are smaller than of classic one, and the error oscillation of our approach also, which indicate our approach is more robust and precise.
(2) Optimization will improve the solution, specially 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 for not very precise application.

Our approach is more precise and robust, the reason is that: the classic approaches require the rotational angles of A,B equal in solution of initial value, and doesn't consider the deviation between the angles, which cause its sensibility to deviation, for deviation always exists, it leads the classic approaches not as robust as our approach. On the contrary, our approach considers various deviations of A,B, furthermore, it fits for the situation of non-rigidness of A,B. In summary our approach is robust, precise and flexible.

References:

[1] Marius-Constantin Popescu, Ilie Borcosi Onisifor Olaru, Luminita Popescu, Florin Grofu. The Simulation Hybrid Fuzzy Control of SCARA Robot. WSEAS Transactions on Systems and Control, Vol.3, No.2, 2008, pp.105-114

[2] Joo Han Kim, Se Hyun Rhyu, In Soung Jung, Jung Moo Seo. An investigation on development of precision actuator for small robot. The Proceedings of the 9th WSEAS Int. Conf. on Robotics, Control and Manufacturing (ROCOM '09), Hangzhou, China, 2009, pp.62-66.

[3] Bach H. Dinh, Matthew W. Dunnigan, Donald S. Reay. A Practical Approach for Position Control of a Robotic Manipulator Using a Radial Basis Function Network and a Simple Vision System, WSEAS Transactions on Systems and Control. Vol.3, No.4, 2008, 289-298.

[4] Catrina Chivu, Catalin Chivu, Catalin Georgescu. Design and Fuzzy Control of Hand Prosthesis or Anthropomorphic Robotic Hand. WSEAS Transactions on Systems and Control, Vol.3, No.5, 2008, 333-342.

[5] R. Tsai, R. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Transactions on Robotics and Automation, Vol.5, No.3, 1989, pp.345-358.

[6] Y. C. Shiu, S. Ahmad. Calibration of wrist-mounted robotics sensors by solving homogeneous transform equations of the form AX=XB. IEEE Transactions on Robotics and Automation, Vol.5, No.1, 1989, pp.16-27.

[7] Hanqi Zhuang, Zvi S Roth. Simultaneous Calibration of a Robot and a Hand-Mounted Camera[J].IEEE Transactions on Robotics and Automations, Vol.11, No.5, 1995, pp.649-660

[8] Irene Fassi, Giovanni Legnani. Hand to Sensor Calibration: A Geometrical Interpretation of the Matrix Equation AX=XB. Journal on Robotics Systems, Vol.22, No.9, 2005,pp.497-506

[9] J. Chou, M. Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. The International Journal of Robotics Research, Vol.10, No.3, 1991, pp.240-254

[10] R. Horaud, F. Dornaika. Hand-eye calibration. The International Journal of Robotics Research, Vol.14, No.13, 1995, pp.195-210

[11] Zijian Zhao, Yuncai Liu. Hand-Eye Calibration Based on Screw Motions. The 18th International Conference on Pattern Recognition (ICPR'06), 2006, pp.1022-1026.

[12] Nanning Zheng. Computer vision and Pattern Recognition. Beijing, China: National Defense Industry Press, pp.188, 1998

[13] Yunpeng Chen. Matrix Theory. Xi'an, China: Northwest Polytechnical Univ. Press, 2001, pp.284.

[14] Barranco Gutierrez Alejandro Israel, Jose De Jesus Medel Juarez. Digital Camera Calibration Analysis Using Perspective Projection Matrix. The Proceedings of the 8th WSEAS Int. Conf. on Signal Processing, Robotics and Automation(ISPRA'09), Cambridge, UK, 2009, pp.321-325.

[15] Zhenyou Zhang, A flexible new technique for camera calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.22, No.11, 2000, 1330-1334.