Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Inertial measurement unit-camera calibration based on incomplete inertial sensor information*

Hong LIU, Yu-long ZHOU, Zhao-peng GU‡

(*Engineering Lab on Intelligent Perception for Internet of Things (ELIP), Shenzhen Graduate School, Peking University, Shenzhen 518055, China*)

E-mail: hongliu@pku.edu.cn; 1101213442@sz.pku.edu.cn; Guzp@pkusz.edu.cn

Received Feb. 7, 2014; Revision accepted Aug. 21, 2014; Crosschecked Oct. 15, 2014

Abstract: This paper is concerned with the problem of estimating the relative orientation between an inertial measurement unit (IMU) and a camera. Unlike most existing IMU-camera calibrations, the main challenge in this paper is that the information output from the IMU is incomplete. For example, only two tilt information can be read from the gravity sensor of a smart phone. Despite incomplete inertial information, there are strong restrictions between the IMU and camera coordinate systems. This paper addresses the incomplete information based IMU-camera calibration problem by exploiting the intrinsic restrictions among the coordinate transformations. First, the IMU transformation between two poses is formulated with the unknown IMU information. Then the defective IMU information is restored using the complementary visual information. Finally, the Levenberg-Marquardt (LM) algorithm is applied to estimate the optimal calibration result in noisy environments. Experiments on both synthetic and real data show the validity and robustness of our algorithm.

Key words: Calibration, Computer vision, Inertial sensor, Smart phone, Incomplete information
doi:10.1631/jzus.C1400038 **Document code:** A **CLC number:** TP391

1 Introduction

The fusion of inertial and visual data plays an important role in applications such as the inertial measurement unit (IMU) aided visual simultaneous localization and mapping (SLAM) (Nützi *et al.*, 2011; Gu and Dong, 2012). To achieve data fusion, the information output from the IMU or the camera should be translated into the camera or the IMU coordinate system. Therefore, it is important to calibrate the translation between the camera coordinate system and the IMU coordinate system. A lot of work has been done on IMU-camera calibration (Lobo and Dias, 2007; Mirzaei and Roumeliotis, 2008).

Mathematically, this calibration is essentially the same as the hand-eye calibration problem in robotics, but regarding the IMU as the hand. To calibrate the transformation $\boldsymbol{X}$, the robot arm is laid up at two positions with the camera looking at a calibration target (Fig. 1). The relative transformation $\boldsymbol{A}$ between the camera in the two poses can be calculated by the camera calibration methods (Zhang, 2000; Hartley and Zisserman, 2003). The relative transformation $\boldsymbol{B}$ of the hand can be easily computed through the data read from its controller. As a result, the IMU-camera calibration problem is equal to solving the well-known hand-eye calibration equation:

$$\boldsymbol{AX} = \boldsymbol{XB}. \tag{1}$$

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61340046, 60875050, and 60675025), the National High-Tech R&D Program (863) of China (No. 2006AA04Z247), the Science and Technology Innovation Commission of Shenzhen Municipality (Nos. JCYJ20120614152234873, CXC201104210010A, JCYJ20130331144631730, and JCYJ20130331144716089), and the Specialized Research Fund for the Doctoral Programme of Higher Education of China (No. 20130001110011)

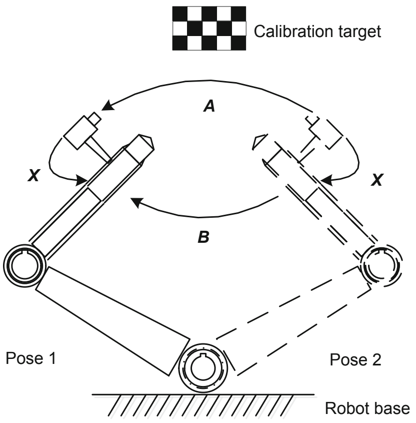


Fig. 1 Converting IMU-camera calibration to hand-eye calibration by mathematically solving $\boldsymbol{AX} = \boldsymbol{XB}$

Most early methods were proposed with linear solutions. The hand-eye calibration equation was first proposed by Shiu and Ahmad (1989) in the context of robot sensor calibration, and a closed-form solution and conditions for its uniqueness were also provided. Similarly, Tsai and Lenz (1989) put forward an efficient linear algorithm and stated that at least two rotations containing motions with nonparallel rotation axes are required. Chou and Kamel (1991) estimated the rotational part by introducing the quaternion representation and gave a linear solution based on singular value decomposition (SVD). Lu and Chou (1995) used the least-squares method by introducing the eight-space quaternion approach. Chen (1991) applied screw motion theory and analyzed the restrictions of hand-eye geometry. Zhao and Liu (2009) suggested a method in which the rotation and translation are calculated simultaneously using dual quaternions with an SVD approach.

Researchers increasingly concentrate on optimization methods due to random noises. Most of the work applied the L_2 nonlinear optimization. Fassi and Legnani (2005) presented an optimization strategy using a similar formulation to Frobenius norms as the cost function. Park and Martin (1994) simplified the parameters in optimization by importing the canonical coordinates. Horaud and Dornaika (1995) formulated the rotational parameters in quaternions and simultaneously solved the robot-world transformation. A new metric on rigid transformations in SE(3) for nonlinear optimization was proposed by Strobl and Hirzinger (2006) with the ability for automatic optimal weighting. Recently, global linear optimization has been applied without the need for initial values. Zhao (2011) proposed new calibration algorithms using convex optimization. Heller *et al.* (2012) presented a technique which recovers the rotation and translation simultaneously and guarantees that the solution will be globally optimal using L_∞ norm. Vicentini *et al.* (2011) obtained better results in terms of accuracy by importing manifolds.

One common point in existing IMU-camera calibration is that the information output from the IMU is complete. However, high-precision IMU may be expensive, and the cheap low-end IMU can output high-precision roll angle and pitch angle, while the yaw angle is exported with a relatively large error (Gu, 2011; Gu and Dong, 2012). In this circumstance, the yaw angle is neglected and this paper is concerned with IMU-camera calibration with only two angles output from the IMU; i.e., the IMU information $\boldsymbol{B}$ in Eq. (1) is partially known. As far as we know, no publication has dealt with this problem. This calibration may have significant applications besides the IMU-aided visual SLAM, such as the currently popular smart phone. A smart phone (Fig. 2) has a gravity sensor which can output only two tilt angles of the phone but has no idea when the phone is rotated around the axis perpendicular to the ground plane.

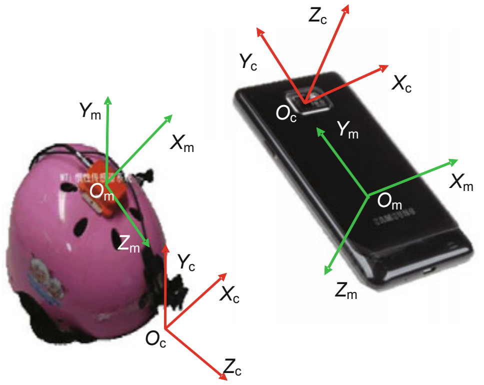


Fig. 2 IMU-aided visual SLAM system (left) and a smart phone with a gravity sensor (right). The coordinate frames with subscripts 'c' and 'm' represent the camera frame and sensor frame, respectively

2 Problem statement

2.1 Original IMU-camera calibration

A 3D rigid transformation $\boldsymbol{X}$ in SE(3) can be expressed by a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$, written as a 4 × 4 matrix:

$$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0}^{\mathrm{T}} & 1\end{bmatrix}. \tag{2}$$

In the IMU-camera system shown in Fig. 3 where the system is measured in two different poses, the relative transformation $\boldsymbol{X}$ between the camera frame and the robot IMU frame is constant since the camera and the IMU are fixed together. Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ denote the transformations from the calibration frame to the camera frame in poses 1 and 2, respectively. Then transformation $\boldsymbol{A}$ between the camera frames in the two poses can be obtained as $\boldsymbol{A}=\boldsymbol{A}_1\boldsymbol{A}_2^{-1}$. Let $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ be the transformations from the world frame to the IMU frames. Then the corresponding transformation of the two poses is $\boldsymbol{B}=\boldsymbol{B}_1\boldsymbol{B}_2^{-1}$. Similarly, $\boldsymbol{A}$ and $\boldsymbol{B}$ can be expressed as follows:

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{R}_A & \boldsymbol{t}_A\\ \boldsymbol{0}^{\mathrm{T}} & 1\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}\boldsymbol{R}_B & \boldsymbol{t}_B\\ \boldsymbol{0}^{\mathrm{T}} & 1\end{bmatrix}, \tag{3}$$

where $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ are rotation matrices, and $\boldsymbol{t}_A$ and $\boldsymbol{t}_B$ are translation vectors. Therefore, the calibration Eq. (1) can be expanded as

$$\boldsymbol{R}_A\boldsymbol{R}=\boldsymbol{R}\boldsymbol{R}_B, \tag{4}$$

$$\boldsymbol{R}_A\boldsymbol{t}+\boldsymbol{t}_A=\boldsymbol{R}\boldsymbol{t}_B+\boldsymbol{t}. \tag{5}$$

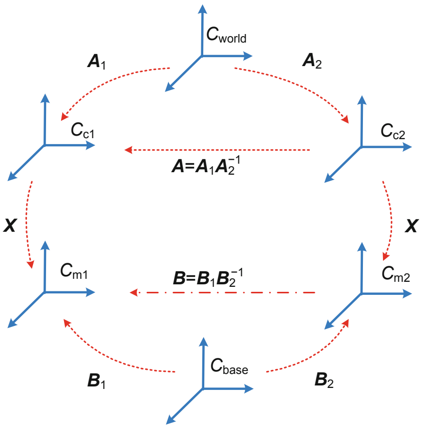


Fig. 3 Coordinate transformations among camera coordinate frames in two poses, IMU coordinate frames in the two poses, and their reference frames

2.2 Formulation of $\boldsymbol{R}_B$

Unlike traditional IMU-camera calibration where $\boldsymbol{B}_i$ $(i=1,\ 2)$ can be easily calculated from the data output from the IMU, here only two tilt angles for the rotation part $\boldsymbol{R}_{B_i}$ of $\boldsymbol{B}_i$ can be obtained. The rotation matrix $\boldsymbol{R}_{B_i}$ can be decomposed into three rotations around three coordinate axes by Euler angle representation (Bajd *et al.*, 2013):

$$\boldsymbol{R}_{B_i}=\mathrm{Rot}(z,\gamma_i)\mathrm{Rot}(y,\beta_i)\mathrm{Rot}(x,\alpha_i), \tag{6}$$

where $\mathrm{Rot}(v,\theta)$ denotes rotation θ around the v axis. Here, as is the case with most inertial devices, it just follows the Z-Y-X form without loss of generality.

Assume the roll angle γ_i and pitch angle β_i can be gained from the auxiliary sensor while the yaw angle α_i remains unknown. Let

$$\begin{cases}\boldsymbol{O}_i=\mathrm{roll}(\gamma_i)=\begin{bmatrix}\cos\gamma_i & -\sin\gamma_i & 0\\ \sin\gamma_i & \cos\gamma_i & 0\\ 0 & 0 & 1\end{bmatrix},\\ \boldsymbol{P}_i=\mathrm{pitch}(\beta_i)=\begin{bmatrix}\cos\beta_i & 0 & \sin\beta_i\\ 0 & 1 & 0\\ -\sin\beta_i & 0 & \cos\beta_i\end{bmatrix},\\ \boldsymbol{Y}_i=\mathrm{yaw}(\alpha_i)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha_i & -\sin\alpha_i\\ 0 & \sin\alpha_i & \cos\alpha_i\end{bmatrix}.\end{cases} \tag{7}$$

Therefore, $\boldsymbol{R}_{B_i}=\boldsymbol{O}_i\boldsymbol{P}_iY_i$ can be decomposed by the Euler angle representations. A result $\boldsymbol{R}_B$ is

$$\boldsymbol{R}_B=\boldsymbol{R}_{B_1}\boldsymbol{R}_{B_2}^{-1}=\boldsymbol{O}_1\boldsymbol{P}_1\boldsymbol{Y}_1(\boldsymbol{O}_2\boldsymbol{P}_2\boldsymbol{Y}_2)^{-1}. \tag{8}$$

2.3 Problem with incomplete IMU information

As stated before, the rotational part $\boldsymbol{R}$ of $\boldsymbol{X}$ can be calibrated while the translation vector $\boldsymbol{t}$ is omitted. A similar omission of $\boldsymbol{t}$ was presented by Seo *et al.* (2009) where the rotational part is solved optimally, but all the translations are assumed to be zeros. To sum up, the goal of this study is to calibrate a new IMU-camera system with incomplete IMU information, and the problem can be simplified and started as follows:

Problem: Given $\boldsymbol{R}_A$ from the camera and four Euler angles β_1,β_2,γ_1, and γ_2 from the sensor with the following equation established:

$$\boldsymbol{R}_A\boldsymbol{R}=\boldsymbol{R}\boldsymbol{O}_1\boldsymbol{P}_1\boldsymbol{Y}_1(\boldsymbol{O}_2\boldsymbol{P}_2\boldsymbol{Y}_2)^{-1}, \tag{9}$$

calculate the relative orientation transform $\boldsymbol{R}$ between the camera frame and sensor frame.

3 Theoretical solution

To deal with the newly formulated IMU-camera calibration problem, we will analyze the formula. Although $\boldsymbol{R}_B$ is partially known, the equation is well structured since all the matrices are 3D orthogonal matrices. The problem can be solved in three steps: (1) Restore $\boldsymbol{R}_B$ by applying the matrix similarity; (2) Calculate $\boldsymbol{R}$ with a linear solution; (3) Implement Levenberg-Marquardt (LM) optimization to obtain the optimal calibration result $\boldsymbol{R}$.

3.1 Recovery of $\boldsymbol{R}_B$

There is only one parameter unknown in $\boldsymbol{R}_B$ as formulated in the previous section. Through the constraint of the orthogonal matrix, it is expected that the unknown parameter can be solved. Make a little change to Eq. (4) giving $\boldsymbol{R}_A = \boldsymbol{R}\boldsymbol{R}_B\boldsymbol{R}^{\mathrm{T}}$. Therefore, $\boldsymbol{R}_A$ is similar to $\boldsymbol{R}_B$ since $\boldsymbol{R}$ is orthogonal, written as

$$\boldsymbol{R}_A \sim \boldsymbol{R}_B. \tag{10}$$

Reform $\boldsymbol{R}_B$ in a more appropriate way on the basis of Eq. (8):

$$\begin{aligned}\boldsymbol{R}_B &= \boldsymbol{O}_1\boldsymbol{P}_1\boldsymbol{Y}_1(\boldsymbol{O}_2\boldsymbol{P}_2\boldsymbol{Y}_2)^{-1}\\ &= \boldsymbol{O}_1\boldsymbol{P}_1(\boldsymbol{Y}_1\boldsymbol{Y}_2^{\mathrm{T}})\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{O}_2^{\mathrm{T}}\boldsymbol{O}_1\boldsymbol{P}_1(\boldsymbol{O}_1\boldsymbol{P}_1)^{\mathrm{T}}\\ &= \boldsymbol{O}_1\boldsymbol{P}_1\widetilde{\boldsymbol{R}_B}(\boldsymbol{O}_1\boldsymbol{P}_1)^{\mathrm{T}}.\end{aligned} \tag{11}$$

Note that all the matrices in Eq. (11) are orthogonal and that the inverse of an orthogonal matrix is equal to the transpose of this matrix. Let

$$\begin{cases}\boldsymbol{Y} = \boldsymbol{Y}_1\boldsymbol{Y}_2^{\mathrm{T}},\\ \boldsymbol{T} = \boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{O}_2^{\mathrm{T}}\boldsymbol{O}_1\boldsymbol{P}_1,\\ \widetilde{\boldsymbol{R}_B} = (\boldsymbol{Y}_1\boldsymbol{Y}_2^{\mathrm{T}})\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{O}_2^{\mathrm{T}}\boldsymbol{O}_1\boldsymbol{P}_1.\end{cases} \tag{12}$$

Here, $\boldsymbol{T}$ can be directly calculated since $\boldsymbol{O}_i$ and $\boldsymbol{P}_i$ $(i=1,2)$ are known while $\boldsymbol{Y}$ can be written as

$$\begin{aligned}\boldsymbol{Y} &= \boldsymbol{Y}_1\boldsymbol{Y}_2^{\mathrm{T}}\\ &= \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\alpha_1-\alpha_2) & -\sin(\alpha_1-\alpha_2)\\ 0 & \sin(\alpha_1-\alpha_2) & \cos(\alpha_1-\alpha_2)\end{bmatrix}.\end{aligned} \tag{13}$$

Assuming $\alpha=\alpha_1-\alpha_2$, $\boldsymbol{R}_B$ and $\widetilde{\boldsymbol{R}_B}$ are formulated with only one parameter α to be determined. $\widetilde{\boldsymbol{R}_B}$ can be expressed in the following form:

$$\widetilde{\boldsymbol{R}_B} = \begin{bmatrix}\boldsymbol{t}_1\\ \cos\alpha\cdot\boldsymbol{t}_2 - \sin\alpha\cdot\boldsymbol{t}_3\\ \sin\alpha\cdot\boldsymbol{t}_2 + \cos\alpha\cdot\boldsymbol{t}_3\end{bmatrix}, \tag{14}$$

where $\boldsymbol{t}_i$ represents the ith row vector of $\boldsymbol{T}$. Since $\boldsymbol{O}_1\boldsymbol{P}_1$ is orthogonal in Eq. (11), we obtain

$$\boldsymbol{R}_B \sim \widetilde{\boldsymbol{R}_B}. \tag{15}$$

As the similarity relation is an equivalence relation, combining Eq. (15) with Eq. (10), we obtain

$$\boldsymbol{R}_A \sim \widetilde{\boldsymbol{R}_B}. \tag{16}$$

The trace of a square matrix is defined as the sum of the diagonal elements of the matrix. Note that the trace of a matrix is similarity invariant, that is, matrices have the same trace if they are similar. Therefore,

$$\begin{aligned}\mathrm{tr}(\boldsymbol{R}_A) &= \mathrm{tr}(\widetilde{\boldsymbol{R}_B})\\ &= T_{11} + (T_{22}+T_{33})\cos\alpha\\ &\quad + (T_{23}-T_{32})\sin\alpha,\end{aligned} \tag{17}$$

where T_{ij} is the (i,j)th entry of $\boldsymbol{T}$.

It is easy to prove that the following inequalities are always true with the restriction of orthogonality of $\boldsymbol{T}$:

$$\begin{cases}T_{22}+T_{33}\neq 0,\\ T_{23}-T_{32}\neq 0.\end{cases} \tag{18}$$

Therefore, Eq. (17) can be reduced to the following form:

$$\boldsymbol{v}_1\cdot(\cos\alpha,\ \sin\alpha) = c. \tag{19}$$

Here, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2 = (\cos\alpha,\ \sin\alpha)$ are both unit vectors and their inner product is c. As shown in Fig. 4, there will be two solutions of α if $-1 < c < 1$ while there is a unique solution of α if $c=\pm 1$.

As a result, one or two solutions of α will be obtained by solving Eq. (17). Putting α into Eq. (11),

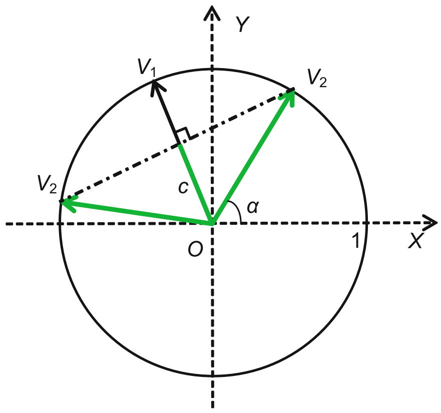


Fig. 4 Geometric interpretation of solving α. Generally two solutions will be obtained

$\boldsymbol{R}_B$ can be computed with one or two candidate values between which only one solution is correct. The final determination of the unique $\boldsymbol{R}_B$ will be processed in the following subsection. Note that the yaw angles α_1 and α_2 cannot be recovered but their difference α can be recovered, which conforms with the physical phenomenon that a single yaw angle is undetermined but their relation in accordance with the relative camera frame is thus restricted. The whole recovery of $\boldsymbol{R}_B$ is summarized in Algorithm 1.

Algorithm 1 Recovery of $\boldsymbol{R}_B$

```
Input: $\boldsymbol{R}_A$ and four Euler angles $\beta_1, \beta_2, \gamma_1, \gamma_2$
Output: $\boldsymbol{R}_B$
Begin
  Calculate $\boldsymbol{O}_i$ and $\boldsymbol{P}_i$, $i$=1, 2
  Calculate $\boldsymbol{T}$
  Formulate $\widetilde{\boldsymbol{R}_B} = \text{yaw}(\alpha)\boldsymbol{T}$
  Compute tr($\boldsymbol{R}_A$) and tr($\widetilde{\boldsymbol{R}_B}$)
  Solve $\alpha$ from Eq. (17)
  Recover $\boldsymbol{R}_B = \boldsymbol{O}_1\boldsymbol{P}_1\text{yaw}(\alpha)\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{O}_2^{\mathrm{T}}$
End
```

3.2 Calculation of $\boldsymbol{R}$

Assume $\boldsymbol{k}_A$ and $\boldsymbol{k}_B$ are unit rotation vectors of $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$, respectively, while θ is the common rotation angle of both $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$. $\boldsymbol{k}_A$ and $\boldsymbol{k}_B$ are also the eigenvectors of $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ corresponding to eigenvalue 1, since every orthogonal matrix has three eigenvalues, one of which is 1 and the other two are a pair of conjugate complex numbers. Ma and Zhang (1998) showed that the relationship between $\boldsymbol{k}_A$ and $\boldsymbol{k}_B$ is $\boldsymbol{k}_A = \boldsymbol{R}\boldsymbol{k}_B$. Given two different such equivalents

$$\begin{cases} \boldsymbol{k}_A^1 = \boldsymbol{R}\boldsymbol{k}_B^1, \\ \boldsymbol{k}_A^2 = \boldsymbol{R}\boldsymbol{k}_B^2, \end{cases} \tag{20}$$

we obtain

$$(\boldsymbol{k}_A^1, \boldsymbol{k}_A^2, \boldsymbol{k}_A^1 \times \boldsymbol{k}_A^2) = \boldsymbol{R}(\boldsymbol{k}_B^1, \boldsymbol{k}_B^2, \boldsymbol{k}_B^1 \times \boldsymbol{k}_B^2). \tag{21}$$

As long as $\boldsymbol{k}_B^1$ is not parallel to $\boldsymbol{k}_B^2$, $\boldsymbol{R}$ can be calculated from Eq. (21):

$$\boldsymbol{R} = (\boldsymbol{k}_A^1, \boldsymbol{k}_A^2, \boldsymbol{k}_A^1 \times \boldsymbol{k}_A^2) \cdot (\boldsymbol{k}_B^1, \boldsymbol{k}_B^2, \boldsymbol{k}_B^1 \times \boldsymbol{k}_B^2)^{-1}. \tag{22}$$

As stated in Section 3.1, there may be two candidate values for $\boldsymbol{R}_B$. Therefore, there are two candidates for each of $\boldsymbol{k}_B^1$ and $\boldsymbol{k}_B^2$. As a result, four candidates of $\boldsymbol{R}$ may be obtained with only one correct solution. To obtain the correct $\boldsymbol{R}$, an additional pair of Eq. (20) is needed and another four candidates of $\boldsymbol{R}$ may be obtained. Comparing the two pairs of solutions of $\boldsymbol{R}$, the correct one appears (because $\boldsymbol{R}$ is fixed, the same $\boldsymbol{R}$ in both candidate groups is the correct one). Then the correct version of the corresponding $\boldsymbol{R}_B$ can be obtained with a backtrace.

3.3 Optimization method

The theoretical solution to the IMU-camera calibration with incomplete IMU information has been obtained in previous sections. However, the Euler angles β_i and γ_i output from the sensor contain measurement noise. Also, $\boldsymbol{R}_A$ computed from two images of the camera is not accurate due to the computational error and image noise. This section deals with the optimal solution of $\boldsymbol{R}$ in noisy environments. In practice, a set of m measurements $\{\beta_1^i, \beta_2^i, \gamma_1^i, \gamma_2^i, \boldsymbol{R}_A^i\}$, will be provided, where $i = 1, 2, \cdots, m$.

For each measurement, $\boldsymbol{R}_B^i$ can be expressed with one parameter free, $\boldsymbol{R}_B^i(\alpha^i)$, according to Eqs. (11) and (14). So, the residual matrix can be expressed as

$$\boldsymbol{Z}^i(\boldsymbol{R}, \alpha^i) = \boldsymbol{R}_A^i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{R}_B^i(\alpha^i). \tag{23}$$

The rotation matrix $\boldsymbol{R}$ can be expressed in the form of a quaternion $\boldsymbol{q}$ (Bajd *et al.*, 2013):

$$\boldsymbol{q} = (\lambda_0, \lambda_1, \lambda_2, \lambda_3). \tag{24}$$

The relationship between the rotation matrix $\boldsymbol{R}$ and the quaternion $\boldsymbol{q}$ is as follows:

$$\boldsymbol{R}=\begin{bmatrix} 1-2(\lambda_2^2+\lambda_3^2) & 2(\lambda_1\lambda_2-\lambda_0\lambda_3) & 2(\lambda_1\lambda_3+\lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2+\lambda_0\lambda_3) & 1-2(\lambda_1^2+\lambda_3^2) & 2(\lambda_2\lambda_3-\lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3-\lambda_0\lambda_2) & 2(\lambda_2\lambda_3+\lambda_0\lambda_1) & 1-2(\lambda_1^2+\lambda_2^2) \end{bmatrix}. \tag{25}$$

Writing $\boldsymbol{Z}^i$ in its column order as a 9D column vector $\boldsymbol{z}^i(\boldsymbol{R}, \alpha^i)$, and letting the argument

$$\boldsymbol{x} = (\boldsymbol{q}, \alpha^1, \alpha^2, \cdots, \alpha^m), \tag{26}$$

we obtain the cost function:

$$E(\boldsymbol{x}) = \sum_{i=1}^{m}(\boldsymbol{z}^i(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{z}^i(\boldsymbol{x}). \tag{27}$$

Considering the quaternion restriction, the optimizing model is

$$\begin{aligned} &\min \sum_{i=1}^{m} (\boldsymbol{z}^i(\boldsymbol{x}))^{\mathrm{T}} \boldsymbol{z}^i(\boldsymbol{x}) \\ &\text{s.t. } \|\boldsymbol{q}\|_2 = 1. \end{aligned} \tag{28}$$

The optimal solution of $\boldsymbol{R}$ can be obtained by minimizing the nonlinear cost function (27). Nonlinear optimization can achieve good results; however, a good initial value is needed. Note that a set of $\boldsymbol{R}^i$ have already been obtained and they can be used as the initial value. The classic method for solving nonlinear least squares problems is the Gauss-Newton method which has the advantage of being rapidly convergent in the neighborhood of a solution. A modification of this algorithm is the Levenberg-Marquardt (LM) algorithm (Marquardt, 1963), which is more robust than the Gauss-Newton procedure. Applying the LM algorithm, the optimal solution $\boldsymbol{R}$ is obtained. The whole IMU-camera calibration using incomplete IMU information is summarized in Fig. 5.

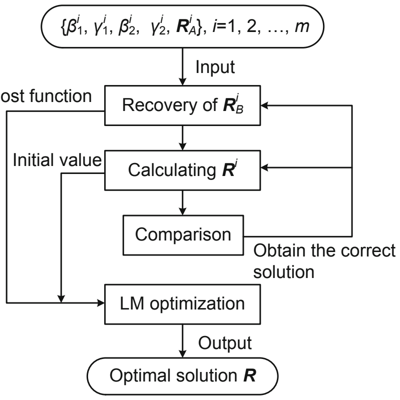

$\{\beta_1^i, \gamma_1^i, \beta_2^i, \gamma_2^i, \boldsymbol{R}_A^i\}, i=1, 2, \dots, m$
Input
Cost function
Recovery of $\boldsymbol{R}_B^i$
Initial value
Calculating $\boldsymbol{R}^i$
Comparison
Obtain the correct solution
LM optimization
Output
Optimal solution $\boldsymbol{R}$

Fig. 5 Flow chat of the whole method for solving IMU-camera calibration with incomplete IMU information

4 Simulations

To verify the performance of IMU-camera calibration with incomplete IMU information, a simulation with synthetic data is given. An overview of the simulation is summarized as follows:

1. Generate real data to obtain the ground truth.

2. Add Gaussian noise to the real data.

3. Solve the IMU-camera calibration with different noise variances and different numbers of measurements using the proposed method.

4. Compare the solutions in different conditions to the ground truth.

The ground truth means the exact value of $\boldsymbol{R}, \boldsymbol{R}_A$, and $\boldsymbol{R}_B$.

4.1 Preliminaries for simulations

For each measurement, six real Euler angles of the sensor (i.e., α_1, β_1, γ_1, α_2, β_2, and γ_2) are generated. To produce the real rotation matrix $\boldsymbol{R}$, three Euler angles α, β, and γ are also generated to represent $\boldsymbol{R}$, which is the same as α_i, β_i, and γ_i for representing $\boldsymbol{R}_{B_i}$. Now the ground truth of $\boldsymbol{R}_A$ is obtained as

$$\boldsymbol{R}_A = \boldsymbol{R}\boldsymbol{O}_1\boldsymbol{P}_1\boldsymbol{Y}_1(\boldsymbol{O}_2\boldsymbol{P}_2\boldsymbol{Y}_2)^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}. \tag{29}$$

Note the inputs are $\beta_1, \gamma_1, \beta_2, \gamma_2$, and $\boldsymbol{R}_A$. We add Gaussian noise to these data. In reality, the Euler angle output from an IMU has an error at about 0.5° (0.008 rad). As a consequence, the simulation is carried out in different Gaussian noise environments with standard derivation (SD) $\sigma = 0.002$, 0.005, 0.01, and 0.02. For $\boldsymbol{R}_A$, the Gaussian noise with a similar standard derivation is appended to each element of $\boldsymbol{R}_A$.

To evaluate the performance of the recovery of $\boldsymbol{R}_B$, the calculated $\widehat{\alpha}$ will be compared to the real value $\alpha_1 - \alpha_2$:

$$e_\alpha = |\alpha_1 - \alpha_2 - \widehat{\alpha}|. \tag{30}$$

Here, the generated α_1 and α_2 are between $-\pi/2$ and $\pi/2$. The calculated $\widehat{\alpha}$ is between $-\pi$ and π.

According to Euler's rotation theorem (Bajd *et al.*, 2013), $\boldsymbol{R}$ can be expressed as a single rotation θ about some axis $\boldsymbol{n} = (n_1, n_2, n_3)$, written as

$$\boldsymbol{r} = (n_1, n_2, n_3, \theta). \tag{31}$$

Here, the rotation angle θ is in its radian measure. The calibration result $\widehat{\boldsymbol{R}}$ will be compared to the ground truth $\boldsymbol{R}$ using their rotation vector representations $\widehat{\boldsymbol{r}}$ and $\boldsymbol{r}$, respectively:

$$e_{\boldsymbol{R}} = \|\widehat{\boldsymbol{r}} - \boldsymbol{r}\|_2. \tag{32}$$

The rotation vector representation is used to measure the rotation matrix because this is geometrically more meaningful.

4.2 Test on accurate data

Before showing the results in noisy environments, the calibration using accurate synthetic data is evaluated. The three Euler angles of the real $\boldsymbol{R}$ are assigned as $\alpha = \pi/4$, $\beta = \pi/4$, and $\gamma = \pi/6$.

The calibration does not contain the optimization step since there is no noise added to the input data. Calculation with different measurements shows that the error is of the order of magnitude of about 10^{-15}. The error is not zero due to the truncation error and computational error such as computation of the eigenvector. Such an error level is considered acceptable, which thus verifies the correctness of the proposed method.

4.3 Test in noisy environments

Considering Gaussian noise, multiple measurements are necessary for an optimal solution. In the linear method (i.e., the algebraic method), a pair of two measurements is required to calculate $\boldsymbol{R}$.

First, 100 measurements were generated with Gaussian noise. The 100 measurements were divided into 50 pairs, and a solution $\boldsymbol{R}^i$ $(i=1, 2, \ldots, 50)$ was calculated for each pair by the algebraic method. The simulations were repeated four times in four Gaussian noise environments with SD σ = 0.002, 0.005, 0.01, and 0.02, respectively. In these simulations, the ground truth of $\boldsymbol{R}$ was assigned with three Euler angles $\alpha = \pi/4$, $\beta = \pi/4$, and $\gamma = \pi/6$.

The error of the calculated $\boldsymbol{R}^i$ for every single pair is exhibited in Fig. 6. In the process of the calculation, 100 values of α were also recovered for each measurement. We are also interested in the error of the recovered α. For comparison, the mean of the two e_α is regarded as the error of α for each pair since there are two measurements in each pair. Therefore, the two errors for a single pair are obtained and the error curves of the 50 pairs are drawn in Fig. 6.

From Fig. 6, it can be observed that most errors

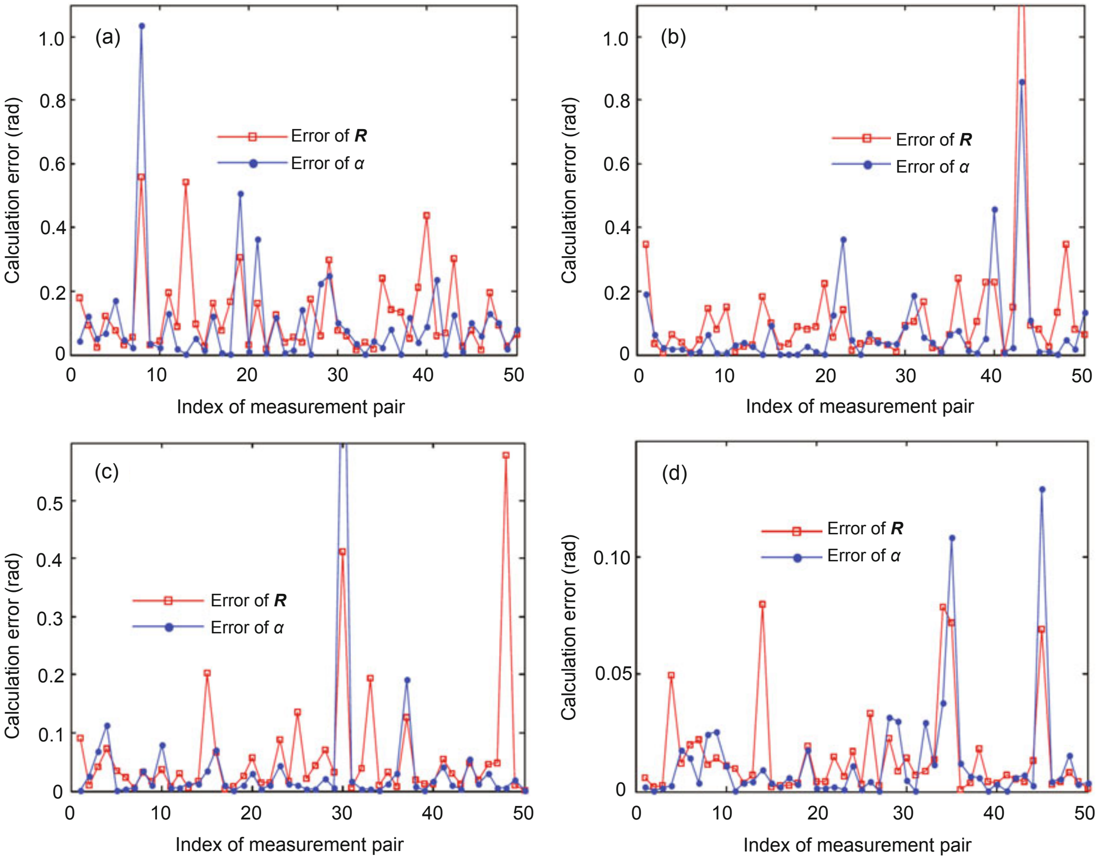


Fig. 6 Calculation errors of α and R in different Gaussian noise environments with σ = 0.002 (a), 0.005 (b), 0.01 (c), and 0.02 (d)

of $\boldsymbol{R}$ are within the limit of 20σ whereas a few errors are relatively large, which is not surprising due to the strong noise there. By carefully analyzing the two curves in each plot, it is found that the two curves generally follow the same trends. Inconsistencies of the two curves may result from the error of $\boldsymbol{R}_A$ or other computational errors.

The calibration result is very rough despite some relatively accurate results. In practice, it is unclear which solution is more accurate than the others. As a result, LM optimization is used to estimate the optimal calibration solution based on the above work. The number of measurements m was assigned from 1 to 50 in the cost function; as a consequence, the LM algorithm was called 200 times in four Gaussian noise environments. The estimation errors are plotted in Fig. 7. Each error curve expresses the estimation error over the number of measurements in the cost function in a Gaussian noise environment.

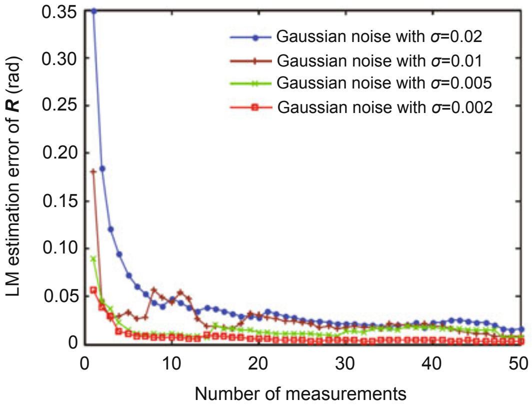


Fig. 7 A comparison of the LM estimation errors using different Gaussian noise with $\sigma = 0.002$, 0.005, 0.01, and 0.02

Fig. 7 also shows the influence of the standard derivation σ on the estimation error. It is seen that the error curves converge as m increases and the converged error value is almost the same as σ. As a result, the calibration result $\boldsymbol{R}$ using optimization is much more accurate than the calculated $\boldsymbol{R}$ before optimization, demonstrating that the calibration result is robust to random noise. On the whole, by employing additional measurements, estimation errors decrease gradually but this trend slows down significantly when $m > 10$.

To better tell how the errors of the estimates grow with the Gaussian noise, we carried out simulations in 19 Gaussian noise environments with standard derivation ranging from 0.002 to 0.02 using both the theoretical method and LM optimization. The results are shown in Fig. 8. The figure plots the average error against the Gaussian noise standard deviation. The average error of the theoretical method is about $5\sigma - 10\sigma$ and the error of LM optimization is about σ.

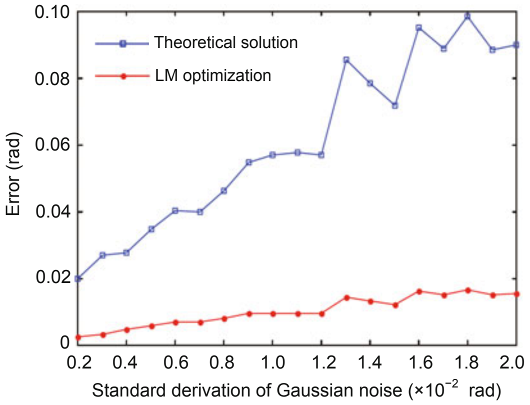


Fig. 8 A comparison of the theoretical solution and LM optimization with different Gaussian noises ($m = 50$ under the theoretical method and $m = 10$ under LM optimization)

5 Experiments

The IMU-camera calibration was also applied in a real system where an IMU was attached to a camera (Fig. 9). The system can achieve IMU-aided visual tasks such as the SLAM system. A fixed chessboard with 7×8 squares was used as the calibration target. In every pose, an image containing the chessboard taken by the camera and the pitch angle and roll angle output from the IMU were obtained. Combining every two data in two different poses, a measurement was obtained. All the images containing the fixed chessboard were used to calculate the rotation transformation from the chessboard frame to the camera frame. Currently, the GML C++ calibration toolbox (http://graphics.cs.msu.ru/en/node/909) (Fig. 10) was used here to compute camera external parameters. Ten measurements were used to estimate the calibration result and the calibrated rotation transformation from the camera frame to the IMU frame was

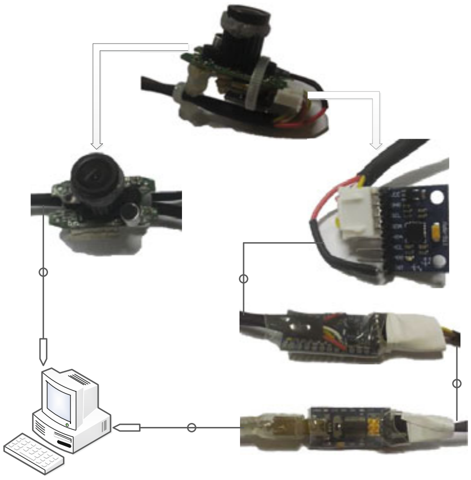
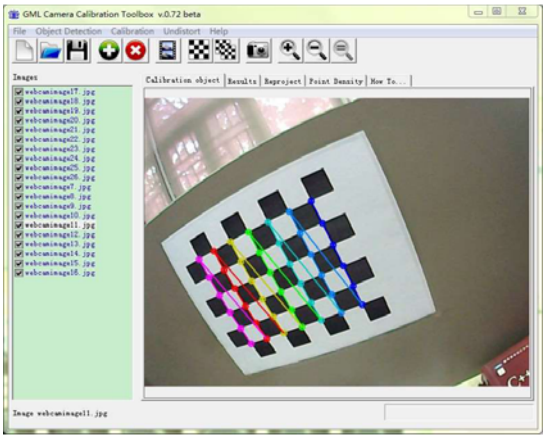


Fig. 9 A real IMU-camera system to be calibrated. The IMU is fixed on the camera which is connected to a computer. The IMU consists of an MPU-6050 IMU, an Arduino Mini microcontroller, and a USB serial converter in order, and then is connected to the computer



☑ webcamimage17.jpg
☑ webcamimage18.jpg
☑ webcamimage19.jpg
☑ webcamimage20.jpg
☑ webcamimage21.jpg
☑ webcamimage22.jpg
☑ webcamimage23.jpg
☑ webcamimage24.jpg
☑ webcamimage25.jpg
☑ webcamimage26.jpg
☑ webcamimage7.jpg
☑ webcamimage8.jpg
☑ webcamimage9.jpg
☑ webcamimage10.jpg
☑ webcamimage11.jpg
☑ webcamimage12.jpg
☑ webcamimage13.jpg
☑ webcamimage14.jpg
☑ webcamimage15.jpg
☑ webcamimage16.jpg

Image webcamimage11.jpg

Fig. 10 GML C++ calibration toolbox for calibration of the camera external parameters

$$\widehat{\boldsymbol{R}}=\begin{bmatrix} -0.1478 & 0.9836 & -0.1029 \\ -0.9890 & -0.1464 & 0.0205 \\ 0.0051 & 0.1048 & 0.9945 \end{bmatrix}.$$

The ground truth of $\boldsymbol{R}$ was not known. However, when fixing the IMU to the camera we try hard to let $\boldsymbol{R}$ be approximately

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, it can be inferred that the calibration result for the IMU-camera pair is basically correct. This demonstrates the validity of IMU-camera calibration with incomplete IMU information in a real system.

6 Conclusions

This paper introduces a new IMU-camera system to be calibrated which is widely applied in vision-inertial systems. Unlike the traditional IMU-camera calibration, the motion information $\boldsymbol{B}$ is incomplete. To solve the new calibration problem, we firstly recover the rotational part $\boldsymbol{R}_B$ by utilizing the property of similar matrices. Then the problem is converted to the traditional calibration problem, and the LM algorithm is applied to estimate the optimal calibration result. Experiments on both synthetic data and real data confirm the validity of the IMU-camera calibration with incomplete IMU information. Also, the experimental results using an optimization method show that the calibration is robust to noise. This paper shows that the calibration can be achieved even without enough information, and the result can be accurate by using an optimization algorithm. Future work could include reducing the running time of the algorithm and use of different optimization methods.

References

Bajd, T., Mihelj, M., Munih, M., 2013. Introduction to Robotics. Springer, the Netherlands. [doi:10.1007/978-94-007-6101-8]

Chen, H.H., 1991. A screw motion approach to uniqueness analysis of head-eye geometry. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.145-151. [doi:10.1109/CVPR.1991.139677]

Chow, J.C.K., Kamel, M., 1991. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *Int. J. Robot. Res.*, **10**(3):240-254. [doi:10.1177/027836499101000305]

Fassi, I., Legnani, G., 2005. Hand to sensor calibration: a geometrical interpretation of the matrix equation AX=XB. *J. Robot. Syst.*, **22**(9):497-506. [doi:10.1002/rob.20082]

Gu, Z.P., 2011. A Study on Monocular Simultaneous Localization and Mapping. PhD Thesis, Chinese Academy of Sciences, Beijing, China (in Chinese).

Gu, Z.P., Dong, Q.L., 2012. Monocular SLAM based on partial IMU information. *J. Comput. Aid. Des. Comput. Graph.*, **24**(2):155-160 (in Chinese).

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision. Cambridge University Press, NY, USA.

Heller, J., Havlena, M., Pajdla, T., 2012. A branch-and-bound algorithm for globally optimal hand-eye calibration. Proc. IEEE Conf. on Computer Vision and

Pattern Recognition, p.1608-1615. [doi:10.1109/CVPR.2012.6247853]

Horaud, R., Dornaika, F., 1995. Hand-eye calibration. *Int. J. Robot. Res.*, **14**(3):195-210. [doi:10.1177/027836499501400301]

Lobo, J., Dias, J., 2007. Relative pose calibration between visual and inertial sensors. *Int. J. Robot. Res.*, **26**(6):561-575. [doi:10.1177/0278364907079276]

Lu, Y., Chou, J.C.K., 1995. Eight-space quaternion approach for robotic hand-eye calibration. Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, p.3316-3321. [doi:10.1109/ICSMC.1995.538297]

Ma, S., Zhang, Z., 1998. Computer Vision. Science Press, Beijing, China (in Chinese).

Marquardt, D.W., 1963. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.*, **11**(2):431-441. [doi:10.1137/0111030]

Mirzaei, F.M., Roumeliotis, S.I., 2008. A Kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. *IEEE Trans. Robot.*, **24**(5):1143-1156. [doi:10.1109/TRO.2008.2004486]

Nützi, G., Weiss, S., Scaramuzza, D., *et al.*, 2011. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *J. Intell. Robot. Syst.*, **61**(1-4):287-299. [doi:10.1007/s10846-010-9490-z]

Park, F.C., Martin, B.J., 1994. Robot sensor calibration: solving AX=XB on the Euclidean group. *IEEE Trans. Robot. Autom.*, **10**(5):717-721. [doi:10.1109/70.326576]

Seo, Y., Choi, Y.J., Lee, S.W., 2009. A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system. Proc. 12th Int. Conf. on Computer Vision, p.1173-1178. [doi:10.1109/ICCV.2009.5459343]

Shiu, Y.C., Ahmad, S., 1989. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. *IEEE Trans. Robot. Autom.*, **5**(1):16-29. [doi:10.1109/70.88014]

Strobl, K.H., Hirzinger, G., 2006. Optimal hand-eye calibration. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, p.4647-4653. [doi:10.1109/IROS.2006.282250]

Tsai, R.Y., Lenz, R.K., 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans Robot. Autom.*, **5**(3):345-358. [doi:10.1109/70.34770]

Vicentini, F., Pedrocchi, N., Malosio, M., *et al.*, 2011. High-accuracy hand-eye calibration from motion on manifolds. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, p.3327-3334. [doi:10.1109/IROS.2011.6094587]

Wang, Y., Ma, X., Wang, Y., *et al.*, 2012. Location optimization of multiple distribution centers under fuzzy environment. *J. Zhejiang Univ.-Sci. A (Appl. Phys. & Eng.)*, **13**(10):782-798. [doi:10.1631/jzus.A1200137]

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**(11):1330-1334. [doi:10.1109/34.888718]

Zhao, Z., 2011. Hand-eye calibration using convex optimization. Proc. IEEE Int. Conf. on Robotics and Automation, p.2947-2952. [doi:10.1109/ICRA.2011.5979569]

Zhao, Z., Liu, Y., 2009. A hand-eye calibration algorithm based on screw motions. *Robotica*, **27**(2):217-223. [doi:10.1017/S0263574708004608]