

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/332934242>

Hand-eye Calibration: 4D Procrustes Analysis Approach

Preprint · May 2019

CITATIONS

0

READS

168

4 authors, including:



Jin Wu

The Hong Kong University of Science and Technology

49 PUBLICATIONS 159 CITATIONS

[SEE PROFILE](#)



MiaoMiao Wang

The University of Western Ontario

19 PUBLICATIONS 89 CITATIONS

[SEE PROFILE](#)



Ming Liu

The Hong Kong University of Science and Technology

190 PUBLICATIONS 2,046 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Cloud Based Mobile Robot [View project](#)



Quadrotor aerial vehicle integrated navigation [View project](#)

Hand-eye Calibration: 4D Procrustes Analysis Approach

Jin Wu  , Member, IEEE, Yuxiang Sun  , Miaomiao Wang  , Student Member, IEEE and Ming Liu  , Senior Member, IEEE

Abstract—We give an universal analytical solution to the hand-eye calibration problem $AX = XB$ with known matrices A, B and unknown variable X , all in the set of special Euclidean group $SE(3)$. The developed method relies on the 4-dimensional Procrustes analysis. An unit-octonion representation is proposed for the first time to solve such Procrustes problem through which an optimal closed-form eigen-decomposition solution is derived. By virtue of such solution, the uncertainty description of X , being a sophisticated problem previously, can be solved in a simpler manner. The proposed approach is then verified using simulations and real-world experimentations on an industrial robotic arm. The results indicate that it owns better accuracy, better description of uncertainty and consumes much less computation time.

Index Terms—Hand-eye Calibration, Homogenous Transformation, Least Squares, Quaternions, Octonions

I. INTRODUCTION

THE main hand-eye calibration problem studied in this paper is aimed to compute the unknown relative homogeneous transformation X between robotic gripper and attached camera, whose poses are denoted as A and B respectively such that $AX = XB$. Hand-eye calibration can be solved via general solutions to the $AX = XB$ problems or through minimizing direct models established using reprojection errors [1]. However, the hand-eye problem $AX = XB$ is not restricted only to the manipulator-camera calibration. Rather, it has been applied to multiple sensor calibration problems including magnetic/inertial ones [2], camera/magnetic ones [3] and other general models [4]. That is to say, the solution of $AX = XB$ is more generalized and has broader applications than methods based on reprojection-error minimization. The early study of the hand-eye calibration problem can be dated back to 1980s when some researchers try to determine the

Manuscript received April 24, 2019; revised June 3, 2019 and June 18, 2019; accepted July 11, 2019. This research was supported by Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ20160401100022706, in part by National Natural Science Foundation of China under the grants of No. U1713211 and 41604025. The Associate Editor coordinating the review process was XXX. (Corresponding author: Ming Liu)

J. Wu, Y. Sun and M. Liu are with Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. (e-mail: jin_wu_uestc@hotmail.com; sun.yuxiang@outlook.com; eel-ium@ust.hk).

M. Wang is with Department of Electronic and Computer Engineering, Western University, London, Ontario, Canada. (e-mail: mwang448@uwo.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier XXX

gripper-camera transformation for accurate robotic perception and reconstruction. During the past over 30 years, there have been a large variety of algorithms solving the hand-eye problem $AX = XB$. Generally speaking, they can be categorized into two groups. The first group consists of those algorithms that calculate the rotation in the first step and then compute the translation part in the second step while in the second group, algorithms compute the rotation and translation simultaneously. There are quite a lot of methods belonging to the very first group that we call them as **separated** ones including representatives of rotation-logarithm based ones like Tsai et al. [5], Shiu et al. [6], Park et al. [7], Horaud et al. [8] and quaternion based one from Chou et al. [9]. The **simultaneous** ones appear in the second group with related representatives of

- 1) **Analytical solutions:** Quaternion-based method by Lu et al. [10], Dual-quaternion based one by Daniilidis [11], Sylvester-equation based one by Andreff et al. [12], Dual-tensor based one by Condurache et al. [13].
- 2) **Numerical solutions:** Gradient/Newton methods by Gwak et al. [14], Linear-matrix-inequality (LMI) based one by Heller et al. [15], Alternative-linear-programming based one by Zhao [16], pseudo-inverse based one by Zhang et al. [3], [17].

Each kind of algorithms have their own pros and cons. The separated ones can not produce good enough results with those cases when translation measurements are more accurate than rotation. The simultaneous ones can achieve better optimization performances but may consume large quantity of time when using numerical iterations. Some algorithms will also suffer from their own ill-posed conditions in the presence some extreme datasets [18]. What's more, the uncertainty description of the X in hand-eye problem $AX = XB$, being an important but difficult problem, has always troubled researchers until the first public general iterative solution by Nguyen et al. in 2018 [19]. An intuitive overview of these algorithms in the order of publication time can be found out in Table I.

Till now, hand-eye calibration has accelerated the development of robotics communities according to its various usages in sensor calibration and motion sensing [20], [21]. Although it has been quite a long time since the first proposal of hand-eye calibration, the researches around it are still very popular. There is still a remaining problem that no algorithm can simultaneously estimate the X in $AX = XB$ while

TABLE I
COMPARISONS BETWEEN RELATED METHODS

Methods	Type	Parameterization or Basic Tools	Computation Speed	Accuracy	Has Uncertainty Description?
Tsai et al. 1989 [5]	Separated, Analytical	Rotation Logarithms	High	Medium	No
Shiu et al. 1989 [6]	Separated, Analytical	Rotation Logarithms	Low	Low	No
Park et al. 1994 [7]	Separated, Analytical	Rotation Logarithms, SVD	Medium	Medium	No
Horraud 1995 [8]	Separated, Analytical	Rotation Logarithms, Eigen-decomposition	High	Medium	No
Chou et al. 1991 [9]	Separated, Analytical	Quaternion, SVD	High	Medium	No
Daniilidis 1999 [11]	Simultaneous, Analytical	Dual Quaternion, SVD	Medium	Medium	No
Andreff et al. 2001 [12]	Simultaneous, Analytical	Sylvester Equation, Kronecker Product	High	Medium	No
Lu et al. 2002 [10]	Simultaneous, Analytical	Quaternion, SVD	Low	Medium	No
Gwak et al. 2003 [14]	Simultaneous, Optimization	Gradient/Newton Method	Very Low	High	No
Heller et al. 2014 [15]	Simultaneous, Optimization	Quaternion, Dual Quaternion, LMI	Very Low	High	No
Condurache et al. 2016 [13]	Simultaneous, Analytical	Dual Tensor, SVD or QR Decomposition	Medium	Medium	No
Zhang et al. 2017 [17]	Simultaneous, Optimization	Dual Quaternion, Pseudo Inverse	Medium	Medium	No
Zhao 2018 [16]	Simultaneous, Optimization	Dual Quaternion, Alternative Linear Programming	Very Low	High	No
Nguyen et al. 2018 [19]	Separated, Optimization	Rotation Iteration	Very Low	High	Yes

preserving highly accurate uncertainty descriptions and consuming extremely low computation time. These difficulties are rather practical since in the hand-eye problem $\mathbf{AX} = \mathbf{XB}$, the rotation and translation parts are tightly coupled with high nonlinearity, which motivates Nguyen et al. to derive the first-order approximation of the error covariance propagation. It is also the presented nonlinearity that makes the numerical iterations much slower.

To overcome the current algorithmic shortcomings, in this paper, we study a new 4-dimensional (4D) Procrustes analysis tool for representation of homogeneous transformations. Understanding the manifolds has become a popular way for modern interior analysis of various data flows [22]. The geometric descriptions of these manifolds have always been vital, which, are usually addressed with the Procrustes analysis that extracts the rigid, affine or non-rigid geometric mappings between datasets [23], [24]. Early researches on Procrustes analysis have been conducted since 1930s [25], [26], [27] and later generalized solutions are applied to spacecraft attitude determination [28], [29], image registration [30], [31], laser scan matching using iterative closest points (ICP) [32], [33] and etc. Motivated by these technological advances, this paper has the following contributions:

- 1) We show some analytical results to the 4D Procrustes analysis in Section III and apply them to the solution of hand-eye calibration problem detailed in Section II.
- 2) Since all variables are directly propagated into final results, the solving process is quite simple and computationally efficient.
- 3) Also, as the proposed solution is in the form of the spectrum decomposition of a 4×4 matrix, the closed-form probabilistic information is given precisely and flexibly for the first time using some recent results in automatic control.

Finally, via simulations and real-world robotic experiments in Section IV, the proposed method is evaluated to own better potential accuracy, computational loads and uncertainty descriptions. Detailed comparisons are also shown to reveal the sensitivity of the proposed method subject to input noise and different parameter values.

II. PROBLEM FORMULATION

We start this section by first defining some important notations in this paper that are mostly inherited from [34]. The n -dimensional real Euclidean space is represented by \mathbb{R}^n which further generates the matrix space $\mathbb{R}^{m \times n}$ containing all real matrices with m rows and n columns. All n -dimensional rotation matrices belong to the special orthogonal group $\text{SO}(n) := \{\mathbf{R} \in \mathbb{R}^{n \times n} | \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$ where \mathbf{I} denotes the identity matrix with proper size. The special Euclidean space is composed of a rotation matrix \mathbf{R} and a translational vector \mathbf{t} such that

$$\text{SE}(n) := \left\{ \mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \mid \mathbf{R} \in \text{SO}(n), \mathbf{t} \in \mathbb{R}^n \right\} \quad (1)$$

with $\mathbf{0}$ denoting the zeros matrix with adequate dimensions. The Euclidean norm of a given squared matrix \mathbf{X} will be defined with $\|\mathbf{X}\| = \sqrt{\text{tr}(\mathbf{X}^T \mathbf{X})}$ where tr denotes the matrix trace. The vectorization of an arbitrary matrix \mathbf{X} is defined as $\text{vec}(\mathbf{X})$ and \otimes represents the kronecker product between two matrices. For a given arbitrary matrix \mathbf{X} , \mathbf{X}^\dagger is called its Moore-Penrose generalized inverse. Any rotation \mathbf{R} on the $\text{SO}(3)$ has its corresponding logarithm given by

$$\log(\mathbf{R}) = \frac{\phi}{2 \sin \phi} (\mathbf{R} - \mathbf{R}^T) \quad (2)$$

in which $1 + 2 \cos \phi = \text{tr}(\mathbf{R})$. Given a 3D vector $\mathbf{x} = (x_1, x_2, x_3)^T$, its associated skew-symmetric matrix is

$$[\mathbf{x}]_\times = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \quad (3)$$

satisfying $\mathbf{x} \times \mathbf{y} = [\mathbf{x}]_\times \mathbf{y} = -[\mathbf{y}]_\times \mathbf{x}$ where \mathbf{y} is also an arbitrary 3D vector. The inverse map from the skew-symmetric matrix to the 3D vector is denoted as $[\mathbf{x}]_\times^\wedge = \mathbf{x}$.

Now let us describe the main problem in this paper. Given two measurement sets

$$\begin{aligned} \mathcal{A} &= \{\mathbf{A}_i | \mathbf{A}_i \in \text{SE}(3), i = 1, 2, \dots, N\} \\ \mathcal{B} &= \{\mathbf{B}_i | \mathbf{B}_i \in \text{SE}(3), i = 1, 2, \dots, N\} \end{aligned} \quad (4)$$

consider the hand-eye calibration least square:

$$\arg \min_{\mathbf{X} \in \text{SE}(3)} \mathcal{J} = \sum_{i=1}^N \|\mathbf{A}_i \mathbf{X} - \mathbf{X} \mathbf{B}_i\|^2 \quad (5)$$

where A_i and B_i come to the reality using poses in two successive measurements (also see Fig. 1 in [11])

$$\begin{aligned} A_i &= T_{A_{i+1}} T_{A_i}^{-1} \\ B_i &= T_{B_{i+1}}^{-1} T_{B_i} \end{aligned} \quad (6)$$

with T_{A_i} being the i -th camera pose with respect to the standard objects in world frame and

$$T_{B_i} = T_{B_{i,3}} T_{B_{i,2}} T_{B_{i,1}} \quad (7)$$

are gripper poses with respect to the robotic base, in which $T_{B_{i,1}}, T_{B_{i,2}}, T_{B_{i,3}}$ are transformations between joints of robotic arms. The relationship between these homogeneous transformations can be found out in Fig. 1. The task for us in the remainder of this paper is to give a closed-form solution of X considering rotation and translation simultaneously and moreover, derive the uncertainty description of X .

Let us write A, B into

$$A = \begin{pmatrix} R_A & t_A \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} R_B & t_B \\ 0 & 1 \end{pmatrix} \quad (8)$$

Then one easily obtains

$$\begin{cases} R_A R_X = R_X R_B \\ R_A t_X + t_A = R_X t_B + t_X \end{cases} \quad (9)$$

The method by Park et al. [7] first computes R_X from the first equation of (9) and then solves t_X by inserting R_X into the second sub-equation. The Park's step for computing R_X is tantamount to the following optimization

$$\arg \min_{R_X \in SO(3)} \sum_{i=1}^N \|R_X a_i - b_i\|^2 \quad (10)$$

with

$$\begin{aligned} a_i &= [\log(R_{A_i})]^\wedge \\ b_i &= [\log(R_{B_i})]^\wedge \end{aligned} \quad (11)$$

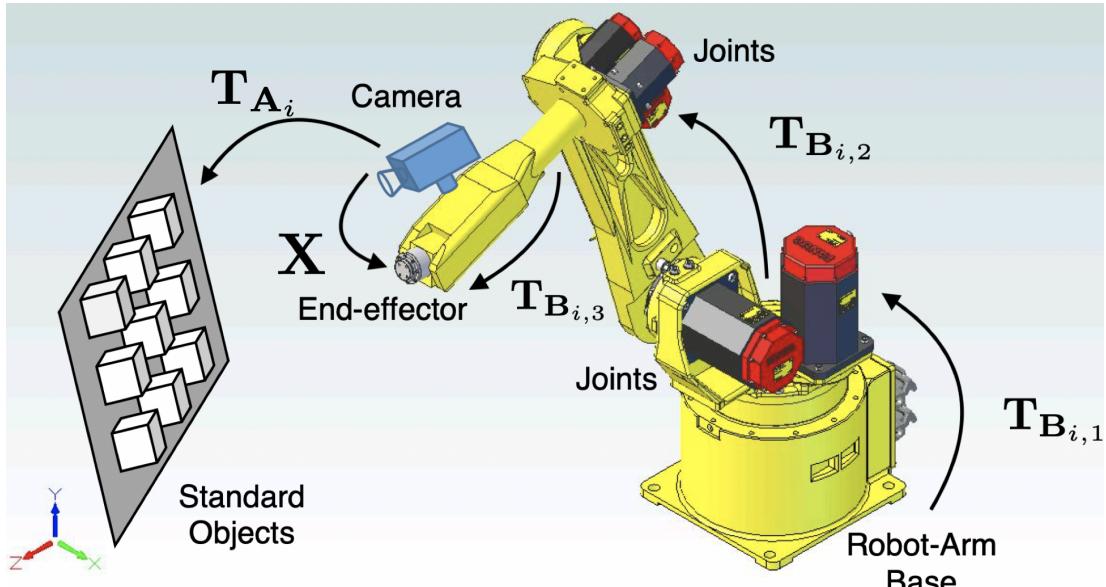


Fig. 1. The relationship between various homogeneous transformations for gripper-camera hand-eye calibration.

Note that (10) is in fact a rigid 3D registration problem which can be solved instantly with singular value decomposition (SVD) or eigen-decomposition [29], [32]. However, the solution of Park et al. does not take the translation into account for R_X while the accuracy of R_X is actually affected by t_X . Therefore, there are some other methods trying to compute R_X and t_X simultaneously [11], [12]. While these methods fix the remaining problem of Park et al., they may not achieve global minimum as the optimization

$$\arg \min_{R_X \in SO(3), t_X \in \mathbb{R}^3} \sum_{i=1}^N \left(\|R_X a_i - b_i\|^2 + \|R_X t_B + t_X - R_A t_X - t_A\|^2 \right) \quad (12)$$

is not always convex. Hence, iterative numerical methods are proposed to achieve the globally optimal estimates of R_X and t_X , including solvers in [3], [14], [16], [17]. In the following section, we show a new analytical perspective for hand-eye calibration problem $AX = XB$ using the proposed 4D Procrustes analysis.

III. 4D PROCRUSTES ANALYSIS

The whole results in this section are proposed for the first time solving specific 4D Procrustes analysis problems. The developed approach is therefore named as the 4DPA method for simplicity in later sections.

A. Some New Analytical Results

Problem 1: Let $\{U\} = \{u_i \in \mathbb{R}^4\}, \{V\} = \{v_i \in \mathbb{R}^4\}$ where $i = 1, 2, \dots, N, N \geq 3$ be two point sets in which the correspondences are well matched such that u_i corresponds exactly to v_i . Find the 4D rotation R and translation vector t such that

$$\arg \min_{R \in SO(4), t \in \mathbb{R}^4} \sum_{i=1}^N \|Ru_i + t - v_i\|^2 \quad (13)$$

$$\mathbf{K} = \begin{pmatrix} H_{11} + H_{22} + H_{33} + H_{44} & H_{12} - H_{21} - H_{34} + H_{43} & H_{13} + H_{24} - H_{31} - H_{42} & H_{14} - H_{23} + H_{32} - H_{41} \\ H_{12} - H_{21} + H_{34} - H_{43} & H_{33} - H_{22} - H_{11} + H_{44} & H_{14} - H_{23} - H_{32} + H_{41} & -H_{13} - H_{24} - H_{31} - H_{42} \\ H_{13} - H_{24} - H_{31} + H_{42} & -H_{14} - H_{23} - H_{32} - H_{41} & H_{22} - H_{11} - H_{33} + H_{44} & H_{12} + H_{21} - H_{34} - H_{43} \\ H_{14} + H_{23} - H_{32} - H_{41} & H_{13} - H_{24} + H_{31} - H_{42} & -H_{12} - H_{21} - H_{34} - H_{43} & H_{22} - H_{11} + H_{33} - H_{44} \end{pmatrix} \quad (19)$$

Solution: **Problem 1** is actually a 4D registration problem that can be easily solved via the SVD:

$$\begin{aligned} \mathbf{R} &= \mathbf{U} \text{diag}[1, 1, 1, \det(\mathbf{U}\mathbf{V})] \mathbf{V}^T \\ \mathbf{t} &= \bar{\mathbf{v}} - \mathbf{R}\bar{\mathbf{u}} \end{aligned} \quad (14)$$

with

$$\begin{aligned} \mathbf{U}\mathbf{S}\mathbf{V}^T &= \mathbf{H} \\ \mathbf{H} &= \sum_{i=1}^N (\mathbf{u}_i - \bar{\mathbf{u}})(\mathbf{v}_i - \bar{\mathbf{v}})^T \\ \bar{\mathbf{u}} &= \frac{1}{N} \sum_{i=1}^N \mathbf{u}_i, \bar{\mathbf{v}} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i \end{aligned} \quad (15)$$

where $\mathbf{S} = \text{diag}(s_1, s_2, s_3, s_4)$ is the diagonal matrix containing all singular values of \mathbf{H} . However, SVD can not reflect the interior geometry of $\text{SO}(4)$ and such geometric information of special orthogonal groups will be very helpful for further proofs [35], [36]. The 4D rotation can be characterized with two unitary quaternions $\mathbf{q}_L = (a, b, c, d)^T$ and $\mathbf{q}_R = (p, q, r, s)^T$ by [37]

$$\mathbf{R} = \mathbf{R}_L(\mathbf{q}_L) \mathbf{R}_R(\mathbf{q}_R) \quad (16)$$

with

$$\begin{aligned} \mathbf{R}_L(\mathbf{q}_L) &= \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} \\ \mathbf{R}_R(\mathbf{q}_R) &= \begin{pmatrix} p & -q & -r & -s \\ q & p & s & -r \\ r & -s & p & q \\ s & r & -q & p \end{pmatrix} \end{aligned} \quad (17)$$

being the left and right matrices. Interestingly, such $\mathbf{R}_L(\mathbf{q}_L)$ and $\mathbf{R}_R(\mathbf{q}_R)$ are actually matrix expressions for quaternion products from left and right sides respectively. Rising from 3D spaces, the 4D rotation is much more sophisticated because the 4D cross product is not so unique as that in the 3D case [38]. Therefore, methods previously relying on the 3D skew symmetric matrices are no longer extendable for 4D registration. The parameterization of $\mathbf{R} \in \text{SO}(4)$ by (16) can also be unified with the unit octonion given by

$$\boldsymbol{\sigma} = \frac{1}{\sqrt{2}}(\mathbf{q}_L^T, \mathbf{q}_R^T)^T \in \mathbb{R}^8 \quad (18)$$

Our task here is to derive the closed-form solution of such a $\boldsymbol{\sigma}$ and therefore compute \mathbf{R} and \mathbf{t} . Using the analytical form in (16), we can rewrite the rotation matrix \mathbf{R} into $\mathbf{R} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ with $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4$ standing for the four columns of \mathbf{R} , respectively. For each column, the algebraic factorization can be performed via

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{P}_1(\boldsymbol{\sigma}) \boldsymbol{\sigma} \\ \mathbf{c}_2 &= \mathbf{P}_2(\boldsymbol{\sigma}) \boldsymbol{\sigma} \\ \mathbf{c}_3 &= \mathbf{P}_3(\boldsymbol{\sigma}) \boldsymbol{\sigma} \\ \mathbf{c}_4 &= \mathbf{P}_4(\boldsymbol{\sigma}) \boldsymbol{\sigma} \end{aligned} \quad (20)$$

where $\mathbf{P}_1(\boldsymbol{\sigma}), \mathbf{P}_2(\boldsymbol{\sigma}), \mathbf{P}_3(\boldsymbol{\sigma}), \mathbf{P}_4(\boldsymbol{\sigma}) \in \mathbb{R}^{4 \times 8}$ are given in the Appendix A. These matrices, however, are subjected to the following equalities

$$\begin{aligned} \mathbf{P}_1(\boldsymbol{\sigma}) \mathbf{P}_1^T(\boldsymbol{\sigma}) &= \mathbf{P}_2(\boldsymbol{\sigma}) \mathbf{P}_2^T(\boldsymbol{\sigma}) \\ &= \mathbf{P}_3(\boldsymbol{\sigma}) \mathbf{P}_3^T(\boldsymbol{\sigma}) = \mathbf{P}_4(\boldsymbol{\sigma}) \mathbf{P}_4^T(\boldsymbol{\sigma}) \\ &= \frac{1}{2} (a^2 + b^2 + c^2 + d^2 + p^2 + q^2 + r^2 + s^2) \mathbf{I} \\ &= \mathbf{I} \end{aligned} \quad (21)$$

Then following the step of [39], one can obtain that ideally

$$\mathbf{P}_1^T(\boldsymbol{\sigma}) \mathbf{H}_1 + \mathbf{P}_2^T(\boldsymbol{\sigma}) \mathbf{H}_2 + \mathbf{P}_3^T(\boldsymbol{\sigma}) \mathbf{H}_3 + \mathbf{P}_4^T(\boldsymbol{\sigma}) \mathbf{H}_4 = \boldsymbol{\sigma} \quad (22)$$

where $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4$ are four rows of the matrix \mathbf{H} , such that

$$\mathbf{H} = (\mathbf{H}_1^T, \mathbf{H}_2^T, \mathbf{H}_3^T, \mathbf{H}_4^T)^T \quad (23)$$

Evaluating the left part of (22), an eigenvalue problem is derived to [39]

$$\mathbf{W}\boldsymbol{\sigma} = \lambda_{\mathbf{W},\max}\boldsymbol{\sigma} \quad (24)$$

The optimal eigenvector $\boldsymbol{\sigma}$ is associated with the maximum eigenvalue $\lambda_{\mathbf{W},\max}$ of \mathbf{W} with \mathbf{W} being an 8×8 matrix in the form of

$$\mathbf{W} = \begin{pmatrix} \mathbf{0} & \mathbf{K} \\ \mathbf{K}^T & \mathbf{0} \end{pmatrix} \quad (25)$$

where \mathbf{K} is shown in (19). This indicates that $\lambda_{\mathbf{W},\max}$ subject to

$$\begin{aligned} &\det(\lambda_{\mathbf{W},\max} \mathbf{I} - \mathbf{W}) \\ &= \det(\lambda_{\mathbf{W},\max} \mathbf{I}) \det\left(\lambda_{\mathbf{W},\max} \mathbf{I} - \frac{1}{\lambda_{\mathbf{W},\max}} \mathbf{K}^T \mathbf{K}\right) \\ &= \det(\lambda_{\mathbf{W},\max}^2 \mathbf{I} - \mathbf{K}^T \mathbf{K}) \end{aligned} \quad (26)$$

where the details are shown in Appendix A. In other words, $\lambda_{\mathbf{W},\max}^2$ is the eigenvalue of the 4×4 matrix $\mathbf{K}^T \mathbf{K}$. As the symbolic solutions to generalized quartic equations have been detailed in [40], the computation of the eigenvalues of \mathbf{W} will be very simple. When $\boldsymbol{\sigma}$ is computed, it also gives the \mathbf{R} and thus will produce \mathbf{t} according to (14).

Sub-Problem 1: Given an improper rotation matrix $\tilde{\mathbf{R}}$ which is not strictly on $\text{SO}(4)$, find the optimal rotation $\mathbf{R} \in \text{SO}(4)$ to orthonormalize $\tilde{\mathbf{R}}$.

Solution: This is the orthonormalization problem and can be solved by replacing \mathbf{H} as \mathbf{R} in (15), indicated in [41], [42] and [43].

Problem 2: Let $\{\mathcal{E}\} = \{\mathbf{E}_i \in \text{SO}(4)\}$, $\{\mathcal{Z}\} = \{\mathbf{Z}_i \in \text{SO}(4)\}$ where $i = 1, 2, \dots, N$ be two matrix sets in which \mathbf{E}_i corresponds exactly to \mathbf{Z}_i . Find the 4D rotation \mathbf{R} such that

$$\arg \min_{\mathbf{R} \in \text{SO}(4)} \sum_{i=1}^N \|\mathbf{E}_i \mathbf{R} - \mathbf{R} \mathbf{Z}_i\|^2 \quad (27)$$

Solution: First we provide some properties on this problem. **Problem 2** is very different with (10) because for rotation on $\text{SO}(4)$, the exponential map indicates a 6×6 skew-symmetric matrix. Therefore the previous 3D registration method can not be extended to the 4D case. In the solution to **Problem 1**, we reveal some identities of unit octonion for representation of rotation on $\text{SO}(4)$. Note that this is very similar to the previous quaternion decomposition from rotation (QDR) that been used for solving $\mathbf{A}\mathbf{R} = \mathbf{B}\mathbf{R}$ where $\mathbf{A}, \mathbf{B}, \mathbf{R} \in \text{SO}(3)$ [44]. Then we are going to extend the QDR to the octonion decomposition from rotation (ODR) for the solution of **Problem 2**.

Like (20), $\mathbf{R} \in \text{SO}(4)$ can also be decomposed from rows such that

$$\begin{aligned} \mathbf{R} &= \left(\mathbf{r}_1^T, \mathbf{r}_2^T, \mathbf{r}_3^T, \mathbf{r}_4^T \right)^T \\ \mathbf{r}_1 &= \boldsymbol{\sigma}^T \mathbf{Q}_1(\boldsymbol{\sigma}) \\ \mathbf{r}_2 &= \boldsymbol{\sigma}^T \mathbf{Q}_2(\boldsymbol{\sigma}) \\ \mathbf{r}_3 &= \boldsymbol{\sigma}^T \mathbf{Q}_3(\boldsymbol{\sigma}) \\ \mathbf{r}_4 &= \boldsymbol{\sigma}^T \mathbf{Q}_4(\boldsymbol{\sigma}) \end{aligned} \quad (28)$$

where $\mathbf{Q}_1(\boldsymbol{\sigma}), \mathbf{Q}_2(\boldsymbol{\sigma}), \mathbf{Q}_3(\boldsymbol{\sigma}), \mathbf{Q}_4(\boldsymbol{\sigma}) \in \mathbb{R}^{4 \times 8}$ are shown in the Appendix A.

Invoking this ODR, we are able to transform $\mathbf{E}_i \mathbf{R} - \mathbf{R} \mathbf{Z}_i$ into

$$\mathbf{E}_i \mathbf{R} - \mathbf{R} \mathbf{Z}_i = (\mathbf{M}_{i,1} \boldsymbol{\sigma}, \mathbf{M}_{i,2} \boldsymbol{\sigma}, \mathbf{M}_{i,3} \boldsymbol{\sigma}, \mathbf{M}_{i,4} \boldsymbol{\sigma}) \quad (29)$$

where $i = 1, 2, \dots, N$ and

$$\begin{aligned} \mathbf{M}_{i,1} &= \begin{pmatrix} \boldsymbol{\sigma}^T \mathbf{G}_{11,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{12,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{13,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{14,i} \end{pmatrix}, \mathbf{M}_{i,2} = \begin{pmatrix} \boldsymbol{\sigma}^T \mathbf{G}_{21,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{22,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{23,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{24,i} \end{pmatrix}, \\ \mathbf{M}_{i,3} &= \begin{pmatrix} \boldsymbol{\sigma}^T \mathbf{G}_{31,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{32,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{33,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{34,i} \end{pmatrix}, \mathbf{M}_{i,4} = \begin{pmatrix} \boldsymbol{\sigma}^T \mathbf{G}_{41,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{42,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{43,i} \\ \boldsymbol{\sigma}^T \mathbf{G}_{44,i} \end{pmatrix} \end{aligned} \quad (30)$$

in which each $\mathbf{G}_{jk,i}, j, k = 1, 2, 3, 4$ takes the form

$$\mathbf{G}_{jk,i} = \begin{pmatrix} \mathbf{0} & \mathbf{J}_{jk,i} \\ \mathbf{J}_{jk,i}^T & \mathbf{0} \end{pmatrix} \quad (31)$$

with parameter matrices $\mathbf{J}_{jk,i} \in \mathbb{R}^{4 \times 8}$ given in Appendix A. Afterwards, the optimal octonion can be sought by

$$\begin{aligned} &\arg \min_{\mathbf{R} \in \text{SO}(4)} \sum_{i=1}^N \|\mathbf{E}_i \mathbf{R} - \mathbf{R} \mathbf{Z}_i\|^2 \\ &= \arg \min_{\boldsymbol{\sigma}^T \boldsymbol{\sigma} = 1} \sum_{i=1}^N \boldsymbol{\sigma}^T \left(\sum_{j=1}^4 \sum_{k=1}^4 \mathbf{G}_{jk,i}^2 \right) \boldsymbol{\sigma} \\ &= \arg \min_{\boldsymbol{\sigma}^T \boldsymbol{\sigma} = 1} \boldsymbol{\sigma}^T \mathbf{F} \boldsymbol{\sigma} \end{aligned} \quad (32)$$

where

$$\begin{aligned} \mathbf{F} &= \sum_{i=1}^N \sum_{j=1}^4 \sum_{k=1}^4 \mathbf{G}_{jk,i}^2 \\ &= \sum_{i=1}^N \sum_{j=1}^4 \sum_{k=1}^4 \begin{pmatrix} \mathbf{J}_{jk,i} \mathbf{J}_{jk,i}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{jk,i}^T \mathbf{J}_{jk,i} \end{pmatrix} \end{aligned} \quad (33)$$

(32) indicates that $\boldsymbol{\sigma}$ is the eigenvector belonging to the minimum eigenvalue of \mathbf{F} such that

$$\mathbf{F} \boldsymbol{\sigma} = \lambda_{\mathbf{F}, \min} \boldsymbol{\sigma} \quad (34)$$

Since $\mathbf{J}_{jk,i} \mathbf{J}_{jk,i}^T$ and $\mathbf{J}_{jk,i}^T \mathbf{J}_{jk,i}$ have quite the same spectrum distribution, (33) also implies that there are two different minimum eigenvalues for \mathbf{F} with their associated eigenvectors representing \mathbf{q}_L and \mathbf{q}_R respectively. That is to say, the \mathbf{q}_L and \mathbf{q}_R are eigenvectors of \mathbf{F}_{11} and \mathbf{F}_{22} such that

$$\begin{aligned} \mathbf{F}_{11} &= \sum_{i=1}^N \sum_{j=1}^4 \sum_{k=1}^4 \mathbf{J}_{jk,i} \mathbf{J}_{jk,i}^T \\ \mathbf{F}_{22} &= \sum_{i=1}^N \sum_{j=1}^4 \sum_{k=1}^4 \mathbf{J}_{jk,i}^T \mathbf{J}_{jk,i} \\ \mathbf{F} &= \begin{pmatrix} \mathbf{F}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{22} \end{pmatrix} \end{aligned} \quad (35)$$

associated with their minimum eigenvalues, respectively. Then inserting the computed \mathbf{q}_L and \mathbf{q}_R into (16) gives the optimal $\mathbf{R} \in \text{SO}(4)$ for **Problem 2**.

Sub-Problem 2: Let $\{\mathcal{E}\} = \{\mathbf{E}_i \in \text{SO}(4)\}, \{\mathcal{Z}\} = \{\mathbf{Z}_i \in \text{SO}(4)\}$ where $i = 1, 2, \dots, N$, be two sequential matrix sets in which \mathbf{E}_i does not exactly correspond to \mathbf{Z}_i . Find the 4D rotation \mathbf{R}

$$\arg \min_{\mathbf{R} \in \text{SO}(4)} \sum_{i=1}^N \|\mathbf{E}_i \mathbf{R} - \mathbf{R} \mathbf{Z}_i\|^2 \quad (36)$$

provided that $\{\mathcal{E}\}$ and $\{\mathcal{Z}\}$ are sampled asynchronously.

Solution: In this problem, $\{\mathcal{E}\}$ and $\{\mathcal{Z}\}$ are asynchronously sampled measurements with different timestamps. First we need to interpolate the rotations for smooth consensus. Suppose that we have two successive homogeneous transformations $\mathbf{E}_i, \mathbf{E}_{i+1}$ with timestamps of $\tau_{\mathbf{E},i}, \tau_{\mathbf{E},i+1}$ respectively. There exists a measurement of \mathbf{Z}_i with timestamp of $\tau_{\mathbf{Z},i} \in [\tau_{\mathbf{E},i}, \tau_{\mathbf{E},i+1}]$. Then the linear interpolation $\mathbf{E}_{i,i+1}$ on $\text{SO}(4)$ can be found out by

$$\begin{aligned} &\arg \min_{\mathbf{E}_{i,i+1} \in \text{SO}(4)} \text{tr} \left[\begin{pmatrix} w(\mathbf{E}_i \mathbf{E}_{i+1}^T - \mathbf{I})^T (\mathbf{E}_i \mathbf{E}_{i+1}^T - \mathbf{I}) + (1-w)(\mathbf{E}_{i+1} \mathbf{E}_{i+1}^T - \mathbf{I})^T (\mathbf{E}_{i+1} \mathbf{E}_{i+1}^T - \mathbf{I}) \end{pmatrix} \right] \\ &\Rightarrow \arg \min_{\mathbf{E}_{i,i+1} \in \text{SO}(4)} \text{tr} \left\{ \mathbf{E}_{i,i+1} [w \mathbf{E}_i + (1-w) \mathbf{E}_{i+1}]^T \right\} \end{aligned} \quad (37)$$

where w is the timestamp weight between \mathbf{E}_i and \mathbf{E}_{i+1} such that

$$w = \frac{\tau_{\mathbf{E},i+1} - \tau_{\mathbf{Z},i}}{\tau_{\mathbf{E},i+1} - \tau_{\mathbf{E},i}} \quad (38)$$

Then the interpolation can be solved using solution to the **Problem 1** by letting $\mathbf{H} = [w \mathbf{E}_i + (1-w) \mathbf{E}_{i+1}]^T$. After the interpolation, a new interpolated set $\{\tilde{\mathcal{E}}\}$ can be established that well corresponds to $\{\mathcal{Z}\}$ and \mathbf{R} can be solved via the solution to **Problem 2**.

B. Uncertainty Descriptions

In this sub-section, we use $\hat{\mathbf{x}}$ for representing the true value of the noise-disturbed vector \mathbf{x} . The expectation is denoted using $\langle \cdots \rangle$ [29], [45]. All the errors in this sub-section are assumed to be zero-mean which can be found out in [29]. As all the solutions provided in the last sub-section are all in the spectrum-decomposition form, the errors of the derived quaternions can be given by the perturbation theory [46]. In a recent error analysis for the attitude determination from vector observations by Chang et al. [47], the first-order error of the estimated deterministic quaternion \mathbf{q} is

$$\delta\mathbf{q} = [\mathbf{q}^T \otimes (\lambda_{\max} \mathbf{I} - \mathbf{M})^\dagger] \delta\mathbf{m} \quad (39)$$

provided that $\mathbf{m} = \text{vec}(\mathbf{M})$, λ_{\max} being the maximum eigenvalue of the real symmetric matrix \mathbf{M}

$$\mathbf{M}\mathbf{q} = \lambda_{\max}\mathbf{q} \quad (40)$$

The above quaternion error is presented given the assumption that $\delta\mathbf{q}$ is multiplicative such that

$$\hat{\mathbf{q}} = \delta\mathbf{q} \odot \mathbf{q} \quad (41)$$

where \odot denotes the quaternion product. The following contents will discuss the covariance expressions for this type of quaternion error.

Using (39), we have the following quaternion error covariance

$$\begin{aligned} \Sigma_{\delta\mathbf{q}} &= \langle \delta\mathbf{q}\delta\mathbf{q}^T \rangle \\ &= \left\langle [\mathbf{q}^T \otimes (\lambda_{\max} \mathbf{I} - \mathbf{M})^\dagger] \delta\mathbf{m} \delta\mathbf{m}^T [\mathbf{q}^T \otimes (\lambda_{\max} \mathbf{I} - \mathbf{M})^\dagger] \right\rangle^T \\ &= [\mathbf{q}^T \otimes (\lambda_{\max} \mathbf{I} - \mathbf{M})^\dagger] \Sigma_{\delta\mathbf{m}} [\mathbf{q}^T \otimes (\lambda_{\max} \mathbf{I} - \mathbf{M})^\dagger]^T \end{aligned} \quad (42)$$

in which

$$\Sigma_{\delta\mathbf{m}} = \frac{\partial\mathbf{m}}{\partial\mathbf{b}} \Sigma_{\mathbf{b}} \left(\frac{\partial\mathbf{m}}{\partial\mathbf{b}} \right)^T \quad (43)$$

where \mathbf{b} denotes all input variables contributed to the final form of \mathbf{M} . Let us take the solution to **Problem 2** for example. For \mathbf{q}_L , we have

$$\mathbf{F}_{11}\mathbf{q}_L = \lambda_{\mathbf{F},\min}\mathbf{q}_L = \lambda_{\mathbf{F}_{11},\min}\mathbf{q}_L \quad (44)$$

which yields that

$$\begin{aligned} \mathbf{M} &= -\mathbf{F}_{11}, \lambda_{\max} = -\lambda_{\mathbf{F}_{11},\min} \\ \mathbf{m} &= -\text{vec}(\mathbf{F}_{11}) \\ \frac{\partial\mathbf{m}}{\partial\mathbf{b}} &= \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^4 \sum_{k=1}^4 \frac{\partial \text{vec}(\mathbf{J}_{jk,i} \mathbf{J}_{jk,i}^T)}{\partial\mathbf{b}} \quad (45) \\ \mathbf{b} &= [\text{vec}(\mathbf{E}_i)^T, \text{vec}(\mathbf{Z}_i)^T]^T \end{aligned}$$

where we assume that \mathbf{b} for every pair of $\{\mathbf{E}_i, \mathbf{Z}_i\}$ have the same probabilistic distribution. The computation of $\frac{\partial \text{vec}(\mathbf{J}_{jk,i} \mathbf{J}_{jk,i}^T)}{\partial\mathbf{b}}$ can be intuitively conducted using analytical forms of matrices in the Appendix A and this part of work is left for the audience of this paper. The covariance of \mathbf{q}_R can therefore be computed by replacing \mathbf{F}_{11} with \mathbf{F}_{22} in (45).

The cross-covariance between \mathbf{q}_L and \mathbf{q}_R can also be given as follows

$$\begin{aligned} \Sigma_{\delta\mathbf{q}_L \delta\mathbf{q}_R} &= \langle \delta\mathbf{q}_L \delta\mathbf{q}_R^T \rangle \\ &= \left\langle [\mathbf{q}_L^T \otimes (\mathbf{F}_{11} - \lambda_{\mathbf{F}_{11},\min} \mathbf{I})^\dagger] \delta\mathbf{m}_L \right. \\ &\quad \left. \delta\mathbf{m}_R^T [\mathbf{q}_R^T \otimes (\mathbf{F}_{22}, \lambda_{\mathbf{F}_{22},\min} \mathbf{I})^\dagger] \right\rangle^T \\ &= [\mathbf{q}_L^T \otimes (\mathbf{F}_{11} - \lambda_{\mathbf{F}_{11},\min} \mathbf{I})^\dagger] \Sigma_{\delta\mathbf{m}_L \delta\mathbf{m}_R} \\ &\quad [\mathbf{q}_R^T \otimes (\mathbf{F}_{22}, \lambda_{\mathbf{F}_{22},\min} \mathbf{I})^\dagger]^T \end{aligned} \quad (46)$$

in which $\mathbf{m}_L = \text{vec}(\mathbf{F}_{11})$, $\mathbf{m}_R = \text{vec}(\mathbf{F}_{22})$ and $\Sigma_{\delta\mathbf{m}_L \delta\mathbf{m}_R}$ is given by

$$\Sigma_{\delta\mathbf{m}_L \delta\mathbf{m}_R} = \frac{\partial\mathbf{m}_L}{\partial\mathbf{b}} \Sigma_{\mathbf{b}} \left(\frac{\partial\mathbf{m}_R}{\partial\mathbf{b}} \right)^T \quad (47)$$

Eventually, the covariance of the octonion $\boldsymbol{\sigma}$ will be

$$\Sigma_{\boldsymbol{\sigma}} = \begin{pmatrix} \Sigma_{\delta\mathbf{q}_L} & \Sigma_{\delta\mathbf{q}_L \delta\mathbf{q}_R} \\ \Sigma_{\delta\mathbf{q}_R \delta\mathbf{q}_L} & \Sigma_{\delta\mathbf{q}_R} \end{pmatrix} \quad (48)$$

C. Solving $\mathbf{AX} = \mathbf{XB}$ from SO(4) Perspective

The SO(4) parameterization of SE(3) is presented by Thomas in [37] that

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \xleftarrow[\mathcal{F}_1^{-1}]{\mathcal{F}_1} \mathbf{R}_{\mathbf{T},\text{SO}(4)} = \begin{pmatrix} \mathbf{R} & \varepsilon\mathbf{t} \\ \varepsilon\mathbf{t}^T \mathbf{R} & 1 \end{pmatrix} \quad (49)$$

in which ε denotes the dual unit that enables $\varepsilon^2 = 0$. The right part of (49) is on SO(4) and a practical method for approaching the corresponding homogeneous transformation is that we can choose very tiny numbers for $\varepsilon = 1/d$ where $d \gg 1$ is a positive scaling factor, to generate real-number approximation of $\mathbf{R}_{\mathbf{T},\text{SO}(4)}$:

$$\mathbf{R}_{\mathbf{T},\text{SO}(4)} \approx \begin{pmatrix} \mathbf{R} & \frac{1}{d}\mathbf{t} \\ \frac{1}{d}\mathbf{t}^T \mathbf{R} & 1 \end{pmatrix} \quad (50)$$

It is also noted that the mapping in (49) is not unique. For instance, the following mapping also holds for $\mathbf{R}_{\mathbf{T},\text{SO}(4)}^T \mathbf{R}_{\mathbf{T},\text{SO}(4)} = \mathbf{R}_{\mathbf{T},\text{SO}(4)} \mathbf{R}_{\mathbf{T},\text{SO}(4)}^T = \mathbf{I}$ when $d \gg 1$:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \xleftarrow[\mathcal{F}_2^{-1}]{\mathcal{F}_2} \mathbf{R}_{\mathbf{T},\text{SO}(4)} = \begin{pmatrix} \mathbf{R} & \varepsilon\mathbf{t} \\ -\varepsilon\mathbf{t}^T \mathbf{R} & 1 \end{pmatrix} \quad (51)$$

The convenience of such mapping from SE(3) to SO(4) is that some nonlinear equations on SE(3) can be turned to linear ones on SO(4). Choosing a scaling factor d makes an approximation of homogeneous transformation on SO(4). Then the conventional hand-eye calibration problem $\mathbf{AX} = \mathbf{XB}$ can be shifted to

$$\begin{aligned} \arg \min_{\mathbf{X} \in \text{SE}(3)} \mathcal{J} &= \sum_{i=1}^N \|\mathbf{A}_i \mathbf{X} - \mathbf{X} \mathbf{B}_i\|^2 \\ \Rightarrow \arg \min_{\mathbf{R}_{\mathbf{X},\text{SO}(4)} \in \text{SO}(4)} \mathcal{J} &= \\ &\sum_{i=1}^N \|\mathbf{R}_{\mathbf{A}_i, \text{SO}(4)} \mathbf{R}_{\mathbf{X}, \text{SO}(4)} - \mathbf{R}_{\mathbf{X}, \text{SO}(4)} \mathbf{R}_{\mathbf{B}_i, \text{SO}(4)}\|^2 \end{aligned} \quad (52)$$

which can be instantly solved via the solution to **Problem 2**. With asynchronously sampled measurements, the problem can be refined with solution to the **Sub-Problem 2**. While the

uncertainty descriptions of σ related to $\mathbf{R}_{X,SO(4)}$ are shown in last sub-section, we would reveal what the covariances of the rotation and translation look like. Suppose that now we have obtained the covariance of σ in (48) for $\mathbf{R}_{X,SO(4)}$. The covariances between columns of the rotation matrix and the cross-covariances between columns of rotation and translation are considered. Recalling (20), one finds out that the covariance between i -th column and j -th column of $\mathbf{R}_{X,SO(4)}$ is calculated by

$$\begin{aligned} \Sigma_{\delta c_i \delta c_j} &= \left\langle [\mathbf{P}_i(\delta\sigma)\sigma + \mathbf{P}_i(\sigma)\delta\sigma] [\mathbf{P}_j(\delta\sigma)\sigma + \mathbf{P}_j(\sigma)\delta\sigma]^T \right\rangle \\ &= \left\langle \mathbf{P}_i(\delta\sigma)\sigma\sigma^T \mathbf{P}_j^T(\delta\sigma) + \mathbf{P}_i(\sigma)\delta\sigma\delta\sigma^T \mathbf{P}_j^T(\sigma) + \right. \\ &\quad \left. \mathbf{P}_i(\delta\sigma)\sigma\delta\sigma^T \mathbf{P}_j^T(\sigma) + \mathbf{P}_i(\sigma)\delta\sigma\sigma^T \mathbf{P}_j^T(\delta\sigma) \right\rangle \\ &= \left\langle \mathbf{Y}_i(\sigma)\delta\sigma\delta\sigma^T \mathbf{Y}_j^T(\sigma) + \mathbf{P}_i(\sigma)\delta\sigma\delta\sigma^T \mathbf{P}_j^T(\sigma) + \right. \\ &\quad \left. \mathbf{Y}_i(\sigma)\delta\sigma\delta\sigma^T \mathbf{P}_j^T(\sigma) + \mathbf{P}_i(\sigma)\delta\sigma\delta\sigma^T \mathbf{Y}_j^T(\sigma) \right\rangle \\ &= \mathbf{Y}_i(\sigma)\Sigma_{\sigma}\mathbf{Y}_j^T(\sigma) + \mathbf{P}_i(\sigma)\Sigma_{\sigma}\mathbf{P}_j^T(\sigma) + \\ &\quad \mathbf{Y}_i(\sigma)\Sigma_{\sigma}\mathbf{P}_j^T(\sigma) + \mathbf{P}_i(\sigma)\Sigma_{\sigma}\mathbf{Y}_j^T(\sigma) \end{aligned} \quad (53)$$

where $\mathbf{P}_i(\delta\sigma)\sigma = \mathbf{Y}_i(\sigma)\delta\sigma$ and $\mathbf{Y}_i(\sigma)$ is a linear mapping of σ which can be evaluated by symbolic computations. For the current 4D Procrustes analysis, interestingly, we have

$$\mathbf{Y}_i(\sigma) = \mathbf{P}_i(\sigma) \quad (54)$$

Therefore (53) can be interpreted as

$$\Sigma_{\delta c_i \delta c_j} = 4\mathbf{P}_i(\sigma)\Sigma_{\sigma}\mathbf{P}_j^T(\sigma) \quad (55)$$

In particular, the rotation-translation cross-covariances will be described by taking the first 3 rows and 3 columns of covariance matrices of $d \cdot \Sigma_{\delta c_1 \delta c_4}$, $d \cdot \Sigma_{\delta c_2 \delta c_4}$, $d \cdot \Sigma_{\delta c_3 \delta c_4}$ respectively. More specifically, if we need to obtain the covariance of $\mathbf{R}_{X,SO(4)}$, one arrives at

$$\begin{aligned} \Sigma_{\mathbf{R}_{X,SO(4)}} &= \left\langle \delta\mathbf{R}_{X,SO(4)}\delta\mathbf{R}_{X,SO(4)}^T \right\rangle \\ &= \sum_{i=1}^4 [\mathbf{Y}_i(\sigma) + \mathbf{P}_i(\sigma)]\Sigma_{\sigma}[\mathbf{Y}_i(\sigma) + \mathbf{P}_i(\sigma)]^T \\ &= 4 \sum_{i=1}^4 \mathbf{P}_i(\sigma)\Sigma_{\sigma}\mathbf{P}_i^T(\sigma) \end{aligned} \quad (56)$$

where

$$\begin{aligned} \delta\mathbf{R}_{X,SO(4)} &= \begin{bmatrix} \delta\mathbf{P}_1(\sigma)\sigma + \mathbf{P}_1(\sigma)\delta\sigma, \delta\mathbf{P}_2(\sigma)\sigma + \mathbf{P}_2(\sigma)\delta\sigma, \\ \delta\mathbf{P}_3(\sigma)\sigma + \mathbf{P}_3(\sigma)\delta\sigma, \delta\mathbf{P}_4(\sigma)\sigma + \mathbf{P}_4(\sigma)\delta\sigma \end{bmatrix} \\ &= \left\{ [\mathbf{Y}_1(\sigma) + \mathbf{P}_1(\sigma)]\delta\sigma, [\mathbf{Y}_2(\sigma) + \mathbf{P}_2(\sigma)]\delta\sigma, \right. \\ &\quad \left. [\mathbf{Y}_3(\sigma) + \mathbf{P}_3(\sigma)]\delta\sigma, [\mathbf{Y}_4(\sigma) + \mathbf{P}_4(\sigma)]\delta\sigma \right\} \end{aligned} \quad (57)$$

The covariance of \mathbf{R}_X then equals to

$$\Sigma_{\mathbf{R}_X} = \Sigma_{\mathbf{R}_{X,SO(4)}}(1:3, 1:3) \quad (58)$$

where $(1:3, 1:3)$ denotes the block containing first 3 rows and columns. Finally, the covariance of \mathbf{t}_X is given by

$$\Sigma_{\mathbf{t}_X} = d^2 \Sigma_{\delta c_4 \delta c_4}(1:3, 1:3) \quad (59)$$

D. Discussion

The presented SO(4) algorithm for hand-eye calibration has the following advantages:

- 1) It can simultaneously solve rotation and translation in \mathbf{X} for hand-eye calibration problem $\mathbf{AX} = \mathbf{XB}$ and thus own comparable accuracy and robustness with previous representatives.
- 2) All the items from \mathbf{A} and \mathbf{B} are directly propagated to the final forms of eigen-decomposition without any preprocessing techniques e.g. quaternion conversion from rotation, rotation logarithm remaining in previous literatures.
- 3) According to the direct propagation of variables to the final result, the computation speed is extremely fast.
- 4) The uncertainty descriptions can be obtained easily with the given analytical results.

However, the proposed method also owns its drawback, that is, the accuracy of the final computed \mathbf{X} is actually affected by the scaling factor d . Here one can find out that d is actually a factor that scales the translation part to a small vector. However, this does not mean that larger d will lead to better performance, since very large d may reduce the significant digits of a fixed word-length floating point number. Therefore, d can be empirically determined according to the scale of translation vector and required accuracy of floating-number processing. For instance, for a 32-bit computer, one single-precision floating point number requires 4 bytes for storage, then a $d = 1 \times 10^5 \sim 1 \times 10^6 \approx 2^{16} \sim 2^{20}$ will be redundant enough guaranteeing the accuracy of at least $2^{20-32} \text{ m} = 2^{-12} \text{ m} = 2.44 \times 10^{-4} \text{ m}$, which is enough for most optical systems with measurement ranges of 10 m. How to choose the most appropriate d dynamically and optimally will be a difficult but significant task in later works. The algorithmic procedures of the proposed method are described in Algorithm 1 for intuitive implementation. Engineers can also turn to https://github.com/zarathustr/hand_eye_SO4 for some MATLAB codes.

Algorithm 1 The Proposed 4DPA Method for Hand-eye Calibration

Parameter: Empirical value of d .

Require:

- 1) Get N measurements of $\mathbf{A}_i, \mathbf{B}_i$ in $\{\mathcal{A}\}$ and $\{\mathcal{B}\}$ respectively. If they are not synchronously measured, get the most appropriate interpolated sets using solution to **Sub-Problem 2**.
- 2) Select a scaling factor d empirically for $\text{SE}(3) - \text{SO}(4)$ mapping.

Step 1: Convert measurements in $\{\mathcal{A}\}$ and $\{\mathcal{B}\}$ to rotations on $\text{SO}(4)$ via (51).

Step 2: Solve the hand-eye calibration problem $\mathbf{AX} = \mathbf{XB}$ via the solution to **Problem 2**. Remap the calculated $\text{SO}(4)$ solution to $\text{SE}(3)$ using (51).

Step 3: Obtain the covariance of the octonion σ related to \mathbf{X} . Compute the rotation-rotation and rotation-translation cross-covariances via (55).

IV. EXPERIMENTAL RESULTS

A. Experiments on A Robotic Arm

The first category of experiments are conducted for an gripper-camera hand-eye calibration depicted in Fig. 1. The dataset is generated using an UR10 robotic arm and an Intel Realsense D435i camera attached firmly to the end-effector (gripper) of the robotic arm (see Fig. 2).

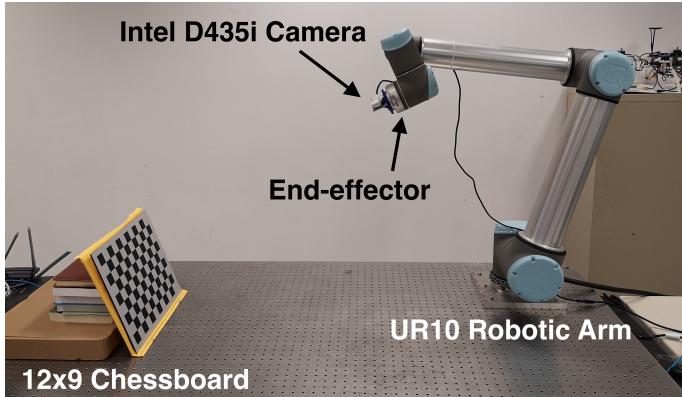


Fig. 2. The gripper-camera hand-eye calibration experiment.

The UR10 robotic arm can give accurate outputs of homogeneous transformations of various joints relative to its base. The D435i camera contains color, depth and fisheye sub-cameras along with and inertial measurement unit. In this sub-section the transformation of the end-effector T_{B_i} of the robotic arm is computed using those transformations from all joints via (7). We only pick up the color images from D435i to obtain the transformation of the camera with respect to the 12×9 chessboard. Note that in Fig. 1, the standard objects can be arbitrary ones with certain pre-known models e.g. point cloud model and computer-aided-design (CAD) model. The D435i is factory-calibrated for its intrinsic parameters and we construct

the following projection model for the utilized camera

$$\begin{aligned} \mathbf{l}_{cam,j} &= \begin{pmatrix} l_{cam,1,j} \\ l_{cam,2,j} \end{pmatrix}, \\ \begin{pmatrix} l_{cam,1,j} \\ l_{cam,2,j} \\ 1 \end{pmatrix} &= \mathbf{O} \left(\frac{L_{cam,1,j}}{L_{cam,3,j}}, \frac{L_{cam,2,j}}{L_{cam,3,j}}, 1 \right)^T \end{aligned} \quad (60)$$

where $\mathbf{l}_{cam,j}$ denotes the j -th measured feature points (corner) of the chessboard in the camera imaging frame; \mathbf{O} is the matrix accounting for the intrinsic parameters of the camera; $\mathbf{L}_{cam,j} = (L_{cam,1,j}, L_{cam,2,j}, L_{cam,3,j})^T$ is the projected j -th feature point in the camera ego-motion frame. To obtain the i -th pose between the camera and chessboard, we can relate the standard point coordinates of the chessboard $\mathbf{L}_{chess,j}$ for $j = 1, 2, \dots$ in the world frame from a certain model with that in the camera frame by

$$\mathbf{L}_{chess,j} = \mathbf{T}_{A_i} \mathbf{L}_{cam,j} \quad (61)$$

By minimizing the projection errors from (61), \mathbf{T}_{A_i} will be obtained with nonlinear optimization techniques e.g. the Perspective-n-Point algorithm [48], [49]. In our experiment, the scale-invariant feature transform (SIFT) is invoked for extraction of corner points of the chessboard [50]. We use several datasets captured from our platform to produce comparisons with representatives including classical ones of Tsai et al. [5], Chou et al. [9], Park et al. [7], Daniilidis [11], Andreff et al. [12] and recent ones of Heller et al. [15], Zhang et al. [17], Zhao [16]. The error of the hand-eye calibration is defined as follows

$$\text{Error} = \frac{1}{N} \sqrt{\sum_{i=1}^N \|\mathbf{A}_i \mathbf{X} - \mathbf{X} \mathbf{B}_i\|^2} \quad (62)$$

where \mathbf{A}_i and \mathbf{B}_i are detailed in (6). All the timing statistics, computation and visualization are carried out on a MacBook Pro 2017 with i7-3.5Ghz CPU along with the MATLAB r2018a software. All the algorithms are implemented using the least coding resources. We employ YALMIP to solve the

TABLE II
COMPARISONS WITH CLASSICAL METHODS FOR GRIPPER-CAMERA HAND-EYE CALIBRATION.

Cases	Tsai 1989 [5]		Chou 1991 [9]		Park 1994 [7]		Daniilidis 1999 [11]		Andreff 2001 [12]		Proposed 4DPA 2019	
	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)
1 (224)	1.2046×10^{-02}	4.0506 $\times 10^{-02}$	6.8255×10^{-03}	1.0190 $\times 10^{-02}$	6.8254×10^{-03}	3.5959 $\times 10^{-02}$	6.7082×10^{-03}	3.8308 $\times 10^{-02}$	7.2650×10^{-03}	5.7292×10^{-03}	6.7857×10^{-03}	4.4819×10^{-03}
2 (253)	1.0243×10^{-02}	4.2952 $\times 10^{-02}$	5.8650×10^{-03}	1.1760 $\times 10^{-02}$	5.8650×10^{-03}	3.9486 $\times 10^{-02}$	5.7704×10^{-03}	4.1492 $\times 10^{-02}$	5.8754×10^{-03}	6.8972×10^{-03}	5.8290×10^{-03}	4.8991×10^{-03}
3 (298)	8.0653×10^{-03}	4.9823 $\times 10^{-02}$	5.0514×10^{-03}	1.3105 $\times 10^{-02}$	5.0517×10^{-03}	4.6559 $\times 10^{-02}$	4.9084×10^{-03}	4.8626 $\times 10^{-02}$	5.4648×10^{-03}	7.2326×10^{-03}	4.9803×10^{-03}	5.5470×10^{-03}
4 (342)	6.8136×10^{-03}	5.3854 $\times 10^{-02}$	4.7192×10^{-03}	1.4585 $\times 10^{-02}$	4.7254×10^{-03}	5.6970 $\times 10^{-02}$	4.0379×10^{-03}	5.3721×10^{-02}	4.8374×10^{-03}	6.7880×10^{-03}	4.0207×10^{-03}	5.5404×10^{-03}
5 (392)	5.5242×10^{-03}	6.3039 $\times 10^{-02}$	3.4047×10^{-03}	1.7253 $\times 10^{-02}$	3.4012×10^{-03}	6.0697 $\times 10^{-02}$	3.3850×10^{-03}	6.4505×10^{-02}	4.0410×10^{-03}	8.9159×10^{-03}	3.1678×10^{-03}	7.3957×10^{-03}
6 (433)	4.8072×10^{-03}	6.7117 $\times 10^{-02}$	2.9723×10^{-03}	1.8354 $\times 10^{-02}$	2.9694×10^{-03}	6.9967 $\times 10^{-02}$	2.8957×10^{-02}	6.5703×10^{-02}	8.8837×10^{-03}	7.8954×10^{-03}	2.7890×10^{-03}	7.8954×10^{-03}
7 (470)	4.3853×10^{-03}	7.6869 $\times 10^{-02}$	2.7302×10^{-03}	1.9827 $\times 10^{-02}$	2.7270×10^{-03}	1.9827 $\times 10^{-02}$	2.6640×10^{-03}	7.5553×10^{-02}	3.3262×10^{-03}	1.0373×10^{-02}	2.5397×10^{-03}	8.1632×10^{-03}
8 (500)	4.0938×10^{-03}	8.5545 $\times 10^{-02}$	2.5855×10^{-03}	2.1506 $\times 10^{-02}$	2.5807×10^{-03}	7.5722 $\times 10^{-02}$	2.4610×10^{-03}	7.7225×10^{-02}	3.0083×10^{-03}	1.2008×10^{-02}	2.3137×10^{-03}	8.5382×10^{-03}

TABLE III
COMPARISONS WITH RECENT METHODS FOR GRIPPER-CAMERA HAND-EYE CALIBRATION.

Cases	Heller 2014 [15]		Zhang 2017 [17]		Zhao 2018 [16]		Proposed 4DPA 2019	
	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)
1 (224)	7.9803×10^{-03}	1.4586 $\times 10^{-02}$	7.6125×10^{-03}	8.8201 $\times 10^{-03}$	7.1485×10^{-03}	8.0861 $\times 10^{-02}$	6.7857×10^{-03}	4.4819×10^{-03}
2 (253)	6.7894×10^{-03}	1.5661 $\times 10^{-02}$	6.7908×10^{-03}	9.7096 $\times 10^{-03}$	6.1710×10^{-03}	1.0787 $\times 10^{-01}$	5.8290×10^{-03}	4.8991×10^{-03}
3 (298)	5.6508×10^{-03}	1.8173 $\times 10^{-02}$	6.0203×10^{-03}	1.2561 $\times 10^{-02}$	5.3021×10^{-03}	1.4893 $\times 10^{-01}$	4.9803×10^{-03}	5.5470×10^{-03}
4 (342)	4.7192×10^{-03}	2.2755 $\times 10^{-02}$	5.2743×10^{-03}	1.5332 $\times 10^{-02}$	4.4003×10^{-03}	2.2291 $\times 10^{-01}$	4.0207×10^{-03}	5.5404×10^{-03}
5 (392)	3.8706×10^{-03}	2.5399 $\times 10^{-02}$	4.3269×10^{-03}	1.8800 $\times 10^{-02}$	3.6041×10^{-03}	3.0347 $\times 10^{-01}$	3.1678×10^{-03}	7.3957×10^{-03}
6 (433)	3.3504×10^{-03}	2.6373 $\times 10^{-02}$	3.9732×10^{-03}	2.1613 $\times 10^{-02}$	3.1658×10^{-03}	3.8768 $\times 10^{-01}$	2.7890×10^{-03}	7.8954×10^{-03}
7 (470)	3.0547×10^{-03}	2.7973 $\times 10^{-02}$	3.3633×10^{-03}	2.5498 $\times 10^{-02}$	2.8912×10^{-03}	5.0153 $\times 10^{-01}$	2.5397×10^{-03}	8.1632×10^{-03}
8 (500)	2.8862×10^{-03}	2.9879 $\times 10^{-02}$	3.0215×10^{-03}	3.2041 $\times 10^{-02}$	2.6871×10^{-03}	5.8727 $\times 10^{-01}$	2.3137×10^{-03}	8.5382×10^{-03}

LMI dqhec optimization in the method of Heller et al. [15]. For Zhao's method [16], we invoke the fmincon function in MATLAB for numerical solution.

The robotic arm is rigidly installed on the testing table and is operated smoothly and periodically to capture the images of the chessboard from various directions. Using the mechanisms described above we form the series of $\{\mathcal{A}\}, \{\mathcal{B}\}$. We select $d = 10^4$ as scaling factor for evaluation in this sub-section as the translational components are all within $[-2, 2]$ m and in such range the camera has the empirical positioning accuracy of about $0.05 \sim 0.2$ m. We choose the \mathcal{F}_2 mapping in (51) for conversion from SE(3) to SO(4) since in real applications it obtains much more accurate hand-eye calibration results than the \mathcal{F}_1 mapping (49) presented in [37]. The scalar thresholds for the other numerical methods are all set to 1×10^{-15} to guarantee the accuracy. We conduct 8 groups of the experiments using the experimental platform. The errors and computation timespans are processed 100 times for averaged performance evaluation, which are provided in Table II and III. The least errors are marked using the green color and the best ones for computation time are tagged in blue. The statistics of the proposed method are marked bold in the tables for emphasis. The digits after the case serial numbers indicate the sample counts for the studied case.

One can see that with growing sample counts, all the methods obtain more accurate estimates of \mathbf{X} . While with larger quantities of measurements, the processing speeds for the algorithms become slower. However, among all methods, despite they are analytical or iterative, the proposed SO(4) method almost always gives the most accurate results within the least computation time. The reason is that the proposed 4DPA computes the rotation and translation in \mathbf{X} simultaneously and well optimizes the loss function \mathcal{J} of the hand-eye calibration problem. The proposed algorithm can obtain better results than almost all other analytical and numerical ones, except for the cases 1 \sim 3 using the method of Daniilidis. This indicates that with few samples, the accuracy of the proposed 4DPA is lower than the method of Daniilidis but is still close. However, few samples indicate relative low confidence in calibration accuracy and for cases with higher quantities of measurements the proposed 4DPA method is always the best. This shows that the designed 4D Procrustes analysis for mapping from SE(3) to SO(4) is more efficient than other tools e.g. the mappings based on dual quaternion [11] and Kronecker product [12]. Furthermore, our method uses the eigen-decomposition as the solution that is regarded as a robust tool for engineering problems. Our method can reduce the estimation error to about 3.06% \sim 94.01% of original stats compared with classical algorithms and 0.39% \sim 86.11% compared with recent numerical ones. The proposed method is also free of pre-processing techniques like quaternion conversion in other algorithms. All the matrix operations are simple and intuitive which makes the computation very fast. Our method can lower the computation time to about 9.98% \sim 70.68% of original stats compared with classical analytical algorithms and 1.45% \sim 28.58% compared with recent numerical ones. A synchronized sequence of camera-chessboard poses and end-effector poses is made open-source by us and is available on

https://github.com/zarathustra/hand_eye_data. Every researcher can freely download this dataset and evaluate the accuracy and computational efficiency. The advantages of the developed method on both precision and computation time will lead to very effective implementations of hand-eye calibration for industrial applications in the future.

B. The Error Sensitivity to the Noises and Different Parameters of d

In this sub-section, we study the sensitivity of the proposed method subject to input measurement noises. We define the noise corrupted models of rotations as

$$\begin{aligned} \mathbf{R}_{\mathbf{A},i} &= \hat{\mathbf{R}}_{\mathbf{A},i} + \text{Error}_{\mathbf{R}_X} \mathcal{R}_1 \\ \mathbf{R}_{\mathbf{B},i} &= \hat{\mathbf{R}}_{\mathbf{B},i} + \text{Error}_{\mathbf{R}_X} \mathcal{R}_2 \end{aligned} \quad (63)$$

where $\mathcal{R}_1, \mathcal{R}_2$ are random matrices whose columns subject to Gaussian distribution with covariances of \mathbf{I} and $\text{Error}_{\mathbf{R}_X}$ is a scalar accounting for the rotation error. Likewise, the noise models of translations can be given by

$$\begin{aligned} \mathbf{t}_{\mathbf{A},i} &= \hat{\mathbf{t}}_{\mathbf{A},i} + \text{Error}_{\mathbf{t}_X} \mathcal{T}_1 \\ \mathbf{t}_{\mathbf{B},i} &= \hat{\mathbf{t}}_{\mathbf{B},i} + \text{Error}_{\mathbf{t}_X} \mathcal{T}_2 \end{aligned} \quad (64)$$

with noise scale of $\text{Error}_{\mathbf{t}_X}$ and $\mathcal{T}_1, \mathcal{T}_2$ noise vectors subject to normal distribution also with covariance of \mathbf{I} . The perturbed rotations are orthonormalized after the addition of the noises. Here, the Gaussian distribution is selected by following the tradition in [19] since this assumption of distribution covers most cases that we may encounter in the real-world applications.

We take all the compared representatives from last sub-section to this part by adding three more ones of the proposed method with different d of $d = 10^3, d = 10^5$ and $d = 10^6$. The we can both see the comparisons with the representatives and observe the influence of the positive scaling factor d . Several simulations are conducted where we generate datasets of \mathcal{A}, \mathcal{B} with $N = 1000$ and the obtained results are averaged for 10 times. We independently evaluate the effect of $\text{Error}_{\mathbf{R}_X}$ and $\text{Error}_{\mathbf{t}_X}$ imposed on Error. The relationship between $\text{Error}_{\mathbf{R}_X}$ and Error is depicted in Fig. 3 while that between $\text{Error}_{\mathbf{t}_X}$ and Error is presented in Fig. 4. These relationships are demonstrated in the form of the log plot. We can see that with increasing errors in rotation and translation, the errors in the computed \mathbf{X} all arise to a large extent. One can see in the magnified plot of the Fig. 3 that the optimization methods achieves the best accuracy but the proposed method can also obtain comparable estimates. It is shown that with various values of d , the performances of the proposed method differ quite a lot. Fig. 4 indicates that with $d = 10^3$, the evaluated errors on translation are the worst among all compared ones. However, with larger d , this situation has been significantly improved, generating the magnified image in Fig. 4 that when $d = 10^5$ and $d = 10^6$, the errors of the proposed method are quite close to the least ones. As in last sub-section we have tested that the proposed method has the fastest execution speed, it is shown that for the studied cases the developed method can be regarded as a balancing one between the accuracy and computation speed.

$$\begin{aligned}
& \delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)} \delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)}^T \\
&= \begin{pmatrix} \delta \mathbf{R}_{\mathbf{X}} & \delta \mathbf{t}_{\mathbf{X}}/d \\ -\delta \mathbf{t}_{\mathbf{X}}^T \mathbf{R}_{\mathbf{X}}/d - \mathbf{t}_{\mathbf{X}}^T \delta \mathbf{R}_{\mathbf{X}}/d & 0 \end{pmatrix} \begin{pmatrix} \delta \mathbf{R}_{\mathbf{X}}^T & -\mathbf{R}_{\mathbf{X}}^T \delta \mathbf{t}_{\mathbf{X}}/d - \delta \mathbf{R}_{\mathbf{X}}^T \mathbf{t}_{\mathbf{X}}/d \\ \delta \mathbf{t}_{\mathbf{X}}^T/d & 0 \end{pmatrix} \\
&= \begin{bmatrix} \delta \mathbf{R}_{\mathbf{X}} \delta \mathbf{R}_{\mathbf{X}}^T + \frac{1}{d^2} \delta \mathbf{t}_{\mathbf{X}} \delta \mathbf{t}_{\mathbf{X}}^T & -\frac{1}{d} (\delta \mathbf{R}_{\mathbf{X}} \mathbf{R}_{\mathbf{X}}^T \delta \mathbf{t}_{\mathbf{X}} + \delta \mathbf{R}_{\mathbf{X}} \delta \mathbf{R}_{\mathbf{X}}^T \mathbf{t}_{\mathbf{X}}) \\ -\frac{1}{d} (\delta \mathbf{t}_{\mathbf{X}}^T \mathbf{R}_{\mathbf{X}} \delta \mathbf{R}_{\mathbf{X}}^T + \mathbf{t}_{\mathbf{X}}^T \delta \mathbf{R}_{\mathbf{X}} \delta \mathbf{R}_{\mathbf{X}}^T) & \frac{1}{d^2} (\mathbf{t}_{\mathbf{X}}^T \mathbf{R}_{\mathbf{X}} \mathbf{R}_{\mathbf{X}}^T \delta \mathbf{t}_{\mathbf{X}} + \mathbf{t}_{\mathbf{X}}^T \delta \mathbf{R}_{\mathbf{X}} \mathbf{R}_{\mathbf{X}}^T \delta \mathbf{t}_{\mathbf{X}} + \mathbf{t}_{\mathbf{X}}^T \delta \mathbf{R}_{\mathbf{X}} \delta \mathbf{R}_{\mathbf{X}}^T \mathbf{t}_{\mathbf{X}}) \end{bmatrix} \quad (65)
\end{aligned}$$

$$\begin{aligned}
\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}, \text{SO}(4)} &= \langle \delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)} \delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)}^T \rangle \\
&= \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} + \frac{1}{d^2} \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{X}}} & -\frac{1}{d} (\langle \delta \mathbf{t}_{\mathbf{X}} \times \delta \boldsymbol{\theta}_{\mathbf{X}} \rangle + \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} \mathbf{t}_{\mathbf{X}}) \\ -\frac{1}{d} (\langle \delta \mathbf{t}_{\mathbf{X}} \times \delta \boldsymbol{\theta}_{\mathbf{X}} \rangle + \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} \mathbf{t}_{\mathbf{X}})^T & \frac{1}{d^2} (\mathbf{t}_{\mathbf{X}}^T \langle \delta \mathbf{t}_{\mathbf{X}} \times \delta \boldsymbol{\theta}_{\mathbf{X}} \rangle + \mathbf{t}_{\mathbf{X}}^T \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} \mathbf{t}_{\mathbf{X}}) \end{bmatrix} \\
&\approx \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} + \frac{1}{d^2} \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{X}}} & -\frac{1}{d} \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} \mathbf{t}_{\mathbf{X}} \\ -\frac{1}{d} \mathbf{t}_{\mathbf{X}}^T \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} & \frac{1}{d^2} \mathbf{t}_{\mathbf{X}}^T \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}} \mathbf{t}_{\mathbf{X}} \end{bmatrix} \quad (66)
\end{aligned}$$

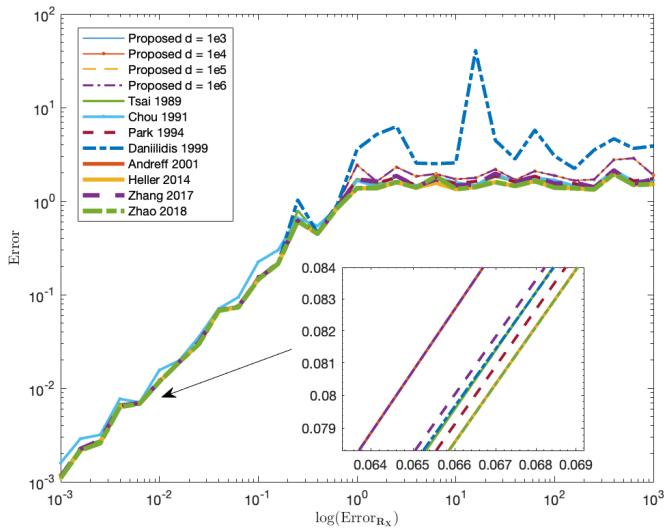


Fig. 3. The sensitivity of errors subject to input rotation noises.

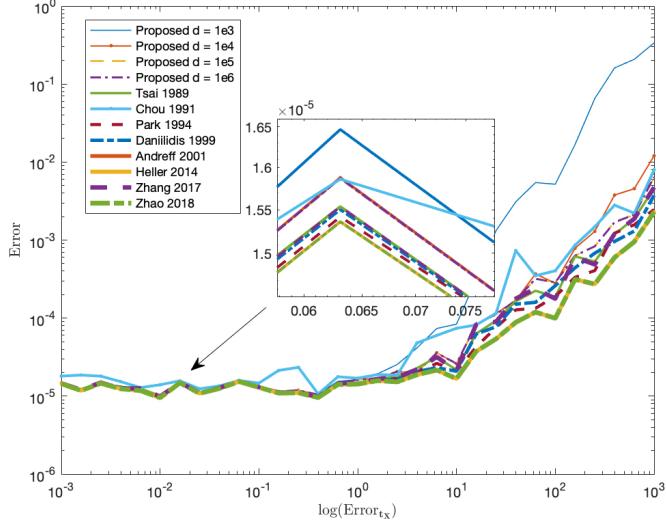


Fig. 4. The sensitivity of errors subject to input translation noises.

C. Simulations on Uncertainty Descriptions

The uncertainty description of the hand-eye calibration problem $\mathbf{AX} = \mathbf{XB}$ is studied by Nguyen et al. for the first

time iteratively in [19]. It works very well with both synthetic and real-world data. However, it still has its drawbacks:

- 1) The covariance of the rotation $\mathbf{R}_{\mathbf{X}}$ is independently estimated from $\mathbf{R}_A \mathbf{R}_{\mathbf{X}} = \mathbf{R}_{\mathbf{X}} \mathbf{R}_B$ but in fact the accuracy of $\mathbf{R}_{\mathbf{X}}$ is also affected by \mathbf{t}_A and \mathbf{t}_B .
- 2) The covariances of $\mathbf{R}_{\mathbf{X}}$ and $\mathbf{t}_{\mathbf{X}}$ are required to be computed iteratively while how many iterations would be sufficient to provide accurate enough results is still an unsolved problem.

Hence the covariance should also be decided by \mathbf{t}_A and \mathbf{t}_B and, if possible, do not require iterations. The proposed SO(4) method in this paper, however, simultaneously estimates the $\mathbf{R}_{\mathbf{X}}$ and $\mathbf{t}_{\mathbf{X}}$ together and can also generate the analytical probabilistic information within several deterministic steps considering the tightly coupled relationship inside $\mathbf{AX} = \mathbf{XB}$. Let us define the $\xi_{\mathbf{R}_{\mathbf{X}},x}$, $\xi_{\mathbf{R}_{\mathbf{X}},y}$ and $\xi_{\mathbf{R}_{\mathbf{X}},z}$ as errors in rotation $\mathbf{R}_{\mathbf{X}}$ around X, Y, Z axes while $\xi_{\mathbf{t}_{\mathbf{X}},x}$, $\xi_{\mathbf{t}_{\mathbf{X}},y}$ and $\xi_{\mathbf{t}_{\mathbf{X}},z}$ being errors in translation $\mathbf{t}_{\mathbf{X}}$ about the X, Y, Z axes, respectively. Given the covariances $\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{X}}}$, $\mathbf{\Sigma}_{\mathbf{t}_{\mathbf{X}}}$, the covariance of the equivalent SO(4) transformation can be computed by (66) where we have

$$\delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)} = \begin{pmatrix} \delta \mathbf{R}_{\mathbf{X}} & \delta \mathbf{t}_{\mathbf{X}}/d \\ -\delta \mathbf{t}_{\mathbf{X}}^T \mathbf{R}_{\mathbf{X}}/d - \mathbf{t}_{\mathbf{X}}^T \delta \mathbf{R}_{\mathbf{X}}/d & 0 \end{pmatrix} \quad (67)$$

and $\delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)} \delta \mathbf{R}_{\mathbf{X}, \text{SO}(4)}^T$ is simplified from (65) to (66) according to [51]

$$\begin{aligned}
\dot{\mathbf{R}}_{\mathbf{X}} &= -[\boldsymbol{\omega}]_{\times} \mathbf{R}_{\mathbf{X}} \\
\delta \mathbf{R}_{\mathbf{X}} &= -[\delta \boldsymbol{\theta}_{\mathbf{X}}]_{\times} \mathbf{R}_{\mathbf{X}} \\
\delta \mathbf{R}_{\mathbf{X}} \mathbf{R}_{\mathbf{X}}^T &= -[\delta \boldsymbol{\theta}_{\mathbf{X}}]_{\times}
\end{aligned} \quad (68)$$

in which $\boldsymbol{\omega}$ is the angular velocity vector and $\boldsymbol{\theta}_{\mathbf{X}}$ denotes the small angle rotation of $\mathbf{R}_{\mathbf{X}}$. Therefore, with $\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{A},i}}$, $\mathbf{\Sigma}_{\mathbf{t}_{\mathbf{A},i}}$, $\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{B},i}}$, $\mathbf{\Sigma}_{\mathbf{t}_{\mathbf{B},i}}$, we can compute $\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{A},i, \text{SO}(4)}}$ and $\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{B},i, \text{SO}(4)}}$. In this paper, we consider that the system errors in each measurement step are identical so we have

$$\begin{aligned}
\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{A},i}} &= \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{A}}}, \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{A},i}} = \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{A}}} \\
\mathbf{\Sigma}_{\mathbf{R}_{\mathbf{B},i}} &= \mathbf{\Sigma}_{\mathbf{R}_{\mathbf{B}}}, \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{B},i}} = \mathbf{\Sigma}_{\mathbf{t}_{\mathbf{B}}}
\end{aligned} \quad (69)$$

Now we conduct the same simulation as that provided in the Python open-source codes of Nguyen et al. [19] (<https://github.com/dinhhuynh2109/python-cope>,

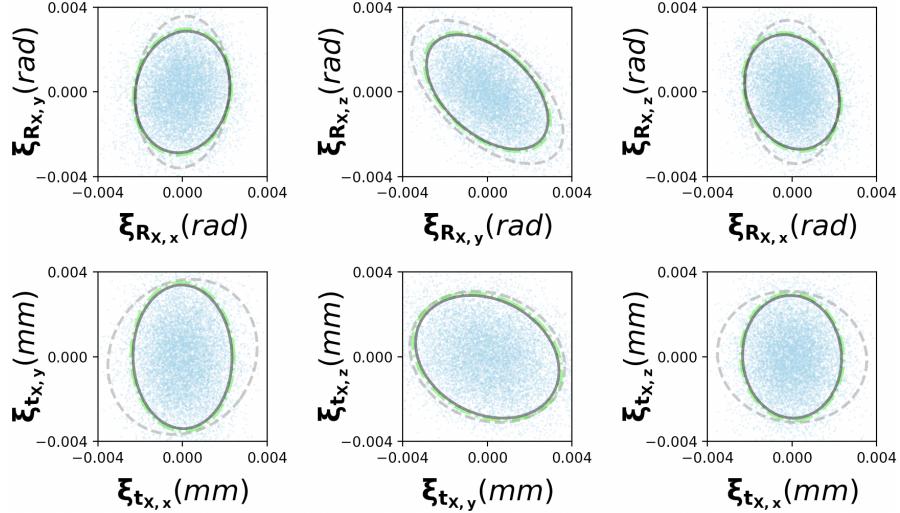


Fig. 5. The 2D covariance projections of the solutions to hand-eye calibration using the proposed method and that of Nguyen et al. The dashed grey lines indicate the mean bounds of the simulated statistics. The green dashed lines are from the solution of Nguyen et al. while the solid black ones are from our proposed algorithm. The discrete points in blue reflect the simulated samples.

`./examples/test_axxb_covariance.py`). The input covariances are

$$\begin{aligned} \Sigma_{R_A} &= 10^{-10} \mathbf{I}, \Sigma_{t_A} = 10^{-10} \mathbf{I} \\ \Sigma_{R_B} &= \begin{pmatrix} 4.15625 & -2.88693 & -0.60653 \\ -2.88693 & 32.0952 & -0.14482 \\ -0.60653 & -0.14482 & 1.43937 \end{pmatrix} \times 10^{-5} \\ \Sigma_{t_B} &= \begin{pmatrix} 19.52937 & 2.12627 & -1.06675 \\ 2.12627 & 4.44314426 & 0.38679 \\ -1.06675 & 0.38679 & 2.13070 \end{pmatrix} \times 10^{-5} \end{aligned} \quad (70)$$

The simulation is carried out for 10000 times, generating the randomly perturbed $\{\mathcal{A}\}$, $\{\mathcal{B}\}$ and in each set there are 60 measurements. Σ_b is computed according to the simulated statistics for $\{\mathcal{A}\}$, $\{\mathcal{B}\}$. The statistical covariance bounds of the estimated \mathbf{R}_X and \mathbf{t}_X are then logged. Using the method by Nguyen et al. and our proposed method, the 2D covariance projections are plotted in Fig. 5. One can see that the both methods can estimate the covariance correctly while our method achieves very slightly smaller covariance bounds. This reflects that our proposed method has reached the accuracy of Nguyen et al. for uncertainty descriptions. What needs to be pointed out is that the proposed method is fully analytical rather than the iterative solution in the method of Nguyen et al. While analytical methods are always much faster than iterative ones, this simulation has indirectly reflected that the proposed method can both correctly estimate the transformation and determine the precision covariance information within short computational period, which is beneficial to those applications with high demands on real-time system with rigorous scheduling logics and timing.

D. Extension to The Extrinsic Calibration between 3D Laser Scanner and A Fisheye Camera

In this sub-section, the developed 4DPA method in this paper is employed to solve the extrinsic calibration problem between a 3D laser scanner and a fisheye camera, mounted rigidly on an experimental platform shown in Fig. 7. This

platform contains a high-end 2D laser scanner of Hokuyo UST 10-lx spinned by a powerful Dynamixel MX-28T servo controlled through the serial ports by the onboard Nvidia TX1 computer with graphics processing unit (GPU). It also consists of an Intel Realsense T265 fisheye camera with resolution of 848x800 and frame rate of 30fps, along with an onboard factory-calibrated inertial measurement unit (IMU). The spin mechanism and the feedback of internal encoder of the servo guarantee the seamless stitching of successive laser scans that produce highly accurate 3D scene reconstructions.

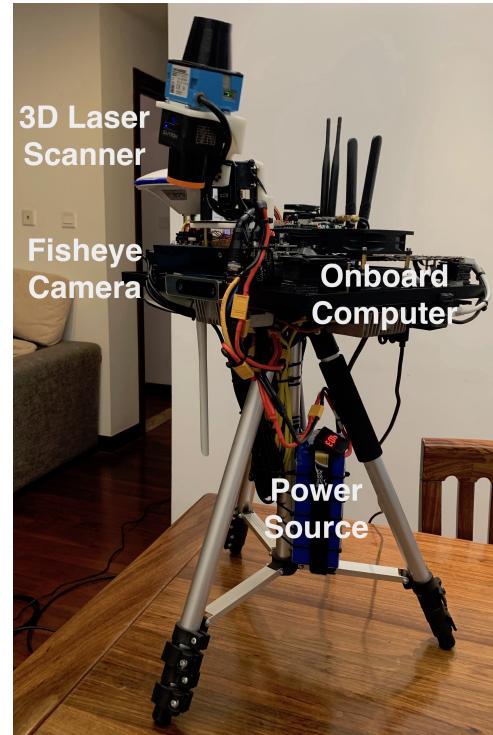


Fig. 7. The experimental platform equipped with a 3D laser scanner and a fisheye camera, along with other processing devices.

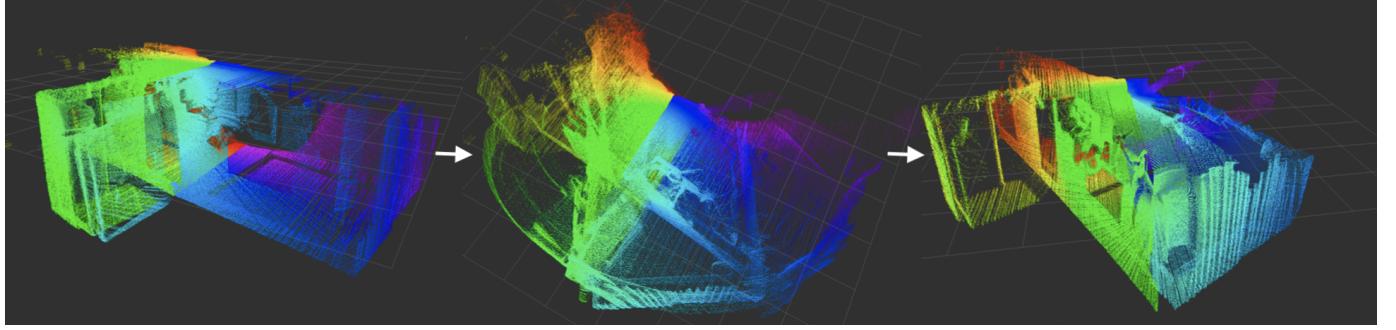


Fig. 6. The reconstructed scenes using the presented 3D laser scanner. They are later used for pose estimation of the laser scanner frame with ICP.

The single sensor or combination of laser scanner and camera will be of great importance for scene measurement and reconstruction [52], [53], [54]. However, due to inevitable installation misalignments, the extrinsic calibration between the laser scanner and camera should be performed for reliable measurement accuracy. Several algorithms have been proposed to deal with the calibration issues inside these sensors recently [55], [56], [57]. These methods in fact require some standard objects like large chessboards to obtain satisfactory results. We here extend our method to solving this extrinsic calibration, without needs of any other standard reference systems. The sensor signal flowchart can be seen from Fig. 8.

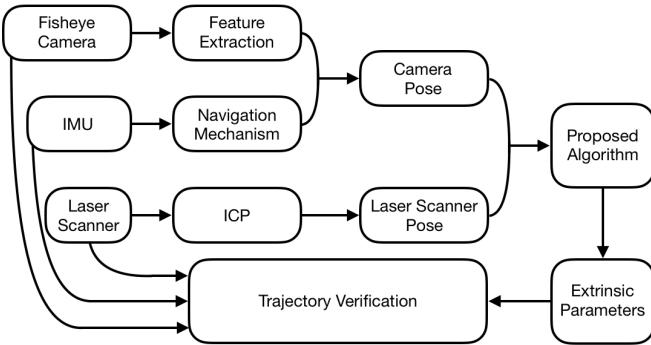


Fig. 8. The signal flowchart in the extrinsic calibration between the 3D laser scanner and fisheye camera using the proposed algorithm.

For the developed system, we can gather three sources of data i.e. images from the fisheye camera, inertial readings of angular rate and acceleration, and 3D laser scans. At the first stage, the fisheye camera and IMU measurements are processed via feature extraction [50] and navigation integral mechanisms [58], respectively. Then they are integrated together for the camera pose, denoted as T_{A_i} with index of i , using the method in [59]. The pose of the 3D laser scanner, denoted as T_{B_i} with index i , is computed via the 3D ICP [33],

as indicated in Fig. 6. As the camera and laser-scanner poses have the output frequencies of 200Hz and 1Hz respectively, the synchronization between them is conducted by continuous linear quaternion interpolation that we developed recently [43]. Then using the properly synced T_{A_i} and T_{B_i} , we are able to form the proposed hand-eye calibration principle with entry point equation in (6). With procedures shown in Algorithm 1 where d is set to $d = 10^5$ empirically, the extrinsic parameters i.e. the rotation and translation between the laser scanner and fisheye camera, are calculated.

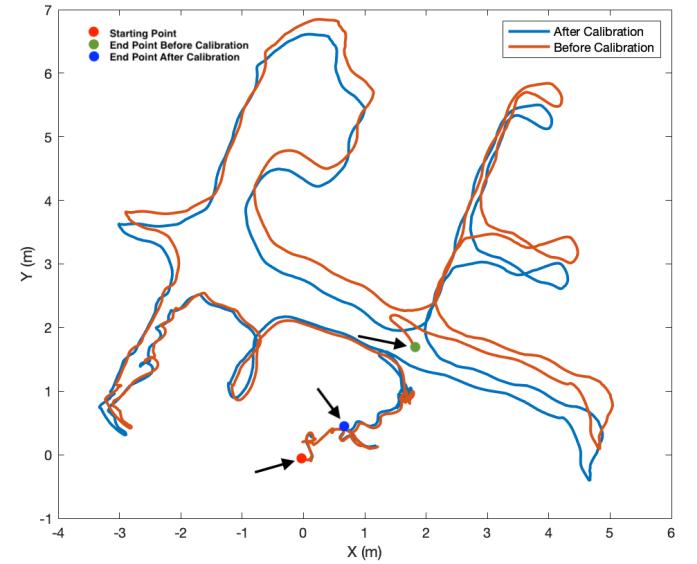


Fig. 9. The projected XY trajectories before and after the extrinsic calibration between 3D laser scanner and fisheye camera.

Then these parameters are applied to the developed platform for 3D trajectory verification using the V-LOAM method [60]. We put the system into measurement mode and then start moving it from origin to origin. Then, when computing the

TABLE IV
TRAJECTORY ERRORS BEFORE AND AFTER THE EXTRINSIC CALIBRATION USING THE PROPOSED METHOD

Experiment	Before: X(m)	After: X(m)	Before: Y(m)	After: Y(m)	Before: Z(m)	After: Z(m)
1	1.997	0.725	1.803	0.476	0.828	0.379
2	2.278	1.080	1.722	0.997	1.322	0.763
3	1.463	0.605	1.825	0.583	1.199	0.691

trajectories with uncalibrated and calibrated data, we can find out that the trajectory after the calibration has much less odometric errors (see Fig. 9). In later periods, the similar experiment is repeated twice. The detailed statistics of the trajectory errors are presented in Table IV containing results on each of the X, Y, Z axes. One can see that the errors have been significantly reduced after calibration, which indicates the effectiveness of the proposed calibration scheme in real scene measurement application. Also, the results of the proposed 4DPA method for hand-eye calibration will be affected by the value d , as described in previous sections. Therefore, a study on such influence is conducted using the data for experiment 3 (see Table V).

TABLE V

THE VERIFIED 3D TRAJECTORY ERRORS AFTER HAND-EYE CALIBRATION WITH DIFFERENT VALUES OF d (EXPERIMENT 3)

d	X(m)	Y(m)	Z(m)
1×10^3	2.068	1.275	2.342
1×10^4	1.472	0.637	1.944
1×10^5	1.463	0.605	1.825
1×10^6	1.462	0.600	1.799
1×10^7	1.462	0.599	1.798

We tune the d from 1×10^3 to 1×10^7 . The errors indicate that the chosen value $d = 1 \times 10^5$ in this sub-section results in sufficiently accurate estimates. And with larger values of d , the error bounds almost reach their limits. While for those small values of d , we can see that they can not deal with the calibration accurately. The reason is that the approximation in (51) requires large d for more precise computation (but not too large, see Section III.D). The optimal dynamic selection of the parameter d will be the next task for us in the near future.

V. CONCLUSION

This paper studies the classical hand-eye calibration problem $\mathbf{AX} = \mathbf{XB}$ by exploiting a new generalized method on $\text{SO}(4)$. The investigated 4D Procrustes analysis provides us with very useful closed-form results for hand-eye calibration. With such framework, the uncertainty descriptions of the obtained transformations can be easily computed. It is verified that the proposed method can achieve better accuracy and much less computation time than representatives in real-world datasets. The proposed uncertainty descriptions for the 4×4 matrices are also universal to other similar problems like spacecraft attitude determination [29] and 3D registration [32]. We also notice that the Procrustes analysis on $\text{SO}(n)$ may be of benefit to solve the generalized hand-eye problem $\mathbf{AX} = \mathbf{XB}$ in which $\text{SE}(n)$ and this is going to be discussed in the next task for us in further works.

APPENDIX A SOME CLOSED-FORM RESULTS

A. Analytical Forms of Some Fundamental Matrices

Taking $\mathbf{c}_1 = \mathbf{P}_1(\boldsymbol{\sigma})\boldsymbol{\sigma}$ as an example, one can explicitly write out

$$\mathbf{c}_1 = \begin{pmatrix} ap - bq - cr - ds \\ aq + bp + cs - dr \\ ar + cp - bs + dq \\ as + br - cq + dp \end{pmatrix}$$

One would be very easy to verify that $\mathbf{c}_1 = \mathbf{P}_1(\boldsymbol{\sigma})\boldsymbol{\sigma}$. Then the similar factorization can be established for $\mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4$ and $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ respectively, generating the following results:

$$\begin{aligned} \mathbf{P}_1(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} p & -q & -r & -s & a & -b & -c & -d \\ q & p & s & -r & b & a & -d & c \\ r & -s & p & q & c & d & a & -b \\ s & r & -q & p & d & -c & b & a \end{pmatrix} \\ \mathbf{P}_2(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} -q & -p & s & -r & -b & -a & -d & c \\ p & -q & r & s & a & -b & c & d \\ -s & -r & -q & p & d & -c & -b & -a \\ r & -s & -p & -q & -c & -d & a & -b \end{pmatrix} \\ \mathbf{P}_3(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} -r & -s & -p & q & -c & d & -a & -b \\ s & -r & -q & -p & -d & -c & -b & a \\ p & q & -r & s & a & b & -c & d \\ -q & p & -s & -r & b & -a & -d & -c \end{pmatrix} \\ \mathbf{P}_4(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} -s & r & -q & -p & -d & -c & b & -a \\ -r & -s & p & -q & c & -d & -a & -b \\ q & -p & -s & -r & -b & a & -d & -c \\ p & q & r & -s & a & b & c & -d \end{pmatrix} \\ \mathbf{Q}_1(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} p & -q & -r & -s & a & -b & -c & -d \\ -q & -p & s & -r & -b & -a & -d & c \\ -r & -s & -p & q & -c & d & -a & -b \\ -s & r & -q & -p & -d & -c & b & -a \end{pmatrix} \\ \mathbf{Q}_2(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} q & p & s & -r & b & a & -d & c \\ p & -q & r & s & a & -b & c & d \\ s & -r & -q & -p & -d & -c & -b & a \\ -r & -s & p & -q & c & -d & -a & -b \end{pmatrix} \\ \mathbf{Q}_3(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} r & -s & p & q & c & d & a & -b \\ -s & -r & -q & p & d & -c & -b & -a \\ p & q & -r & s & a & b & -c & d \\ q & -p & -s & -r & -b & a & -d & -c \end{pmatrix} \\ \mathbf{Q}_4(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{2}} \begin{pmatrix} s & r & -q & p & d & -c & b & a \\ r & -s & -p & -q & -c & -d & a & -b \\ -q & p & -s & -r & b & -a & -d & -c \\ p & q & r & -s & a & b & c & -d \end{pmatrix} \end{aligned}$$

The results of $\mathbf{J}_{jk,i}$ can then be computed using symbolic computation tools e.g. MATLAB and Mathematica:

$$\begin{aligned} \mathbf{J}_{11,i} &= \begin{pmatrix} e_{11} - z_{11} & e_{12} + z_{21} & e_{13} + z_{31} & e_{14} + z_{41} \\ e_{12} + z_{21} & z_{11} - e_{11} & e_{14} - z_{41} & z_{31} - e_{13} \\ e_{13} + z_{31} & z_{41} - e_{14} & z_{11} - e_{11} & e_{12} - z_{21} \\ e_{14} + z_{41} & e_{13} - z_{31} & z_{21} - e_{12} & z_{11} - e_{11} \end{pmatrix} \\ \mathbf{J}_{12,i} &= \begin{pmatrix} e_{21} - z_{21} & e_{22} - z_{11} & e_{23} + z_{41} & e_{24} - z_{31} \\ e_{22} - z_{11} & z_{21} - e_{21} & e_{24} + z_{31} & z_{41} - e_{23} \\ e_{23} - z_{41} & z_{31} - e_{24} & -e_{21} - z_{21} & e_{22} - z_{11} \\ e_{24} + z_{31} & e_{23} + z_{41} & z_{11} - e_{22} & -e_{21} - z_{21} \end{pmatrix} \\ \mathbf{J}_{13,i} &= \begin{pmatrix} e_{31} - z_{31} & e_{32} - z_{41} & e_{33} - z_{11} & e_{34} + z_{21} \\ e_{32} + z_{41} & -e_{31} - z_{31} & e_{34} + z_{21} & z_{11} - e_{33} \\ e_{33} - z_{11} & z_{21} - e_{34} & z_{31} - e_{31} & e_{32} + z_{41} \\ e_{34} - z_{21} & e_{33} - z_{11} & z_{41} - e_{32} & -e_{31} - z_{31} \end{pmatrix} \\ \mathbf{J}_{14,i} &= \begin{pmatrix} e_{41} - z_{41} & e_{42} + z_{31} & e_{43} - z_{21} & e_{44} - z_{11} \\ e_{42} - z_{31} & -e_{41} - z_{41} & e_{44} - z_{11} & z_{21} - e_{43} \\ e_{43} + z_{21} & z_{11} - e_{44} & -e_{41} - z_{41} & e_{42} + z_{31} \\ e_{44} - z_{11} & e_{43} + z_{21} & z_{31} - e_{42} & z_{41} - e_{41} \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
\mathbf{J}_{21,i} &= \\
& \begin{pmatrix} e_{12} - z_{12} & z_{22} - e_{11} & e_{14} + z_{32} & z_{42} - e_{13} \\ z_{22} - e_{11} & z_{12} - e_{12} & -e_{13} - z_{42} & z_{32} - e_{14} \\ z_{32} - e_{14} & z_{42} - e_{13} & e_{12} + z_{12} & e_{11} - z_{22} \\ e_{13} + z_{42} & -e_{14} - z_{32} & z_{22} - e_{11} & e_{12} + z_{12} \end{pmatrix} \\
\mathbf{J}_{22,i} &= \\
& \begin{pmatrix} e_{22} - z_{22} & -e_{21} - z_{12} & e_{24} + z_{42} & -e_{23} - z_{32} \\ -e_{21} - z_{12} & z_{22} - e_{22} & z_{32} - e_{23} & z_{42} - e_{24} \\ -e_{24} - z_{42} & z_{32} - e_{23} & e_{22} - z_{22} & e_{21} - z_{12} \\ e_{23} + z_{32} & z_{42} - e_{24} & z_{12} - e_{21} & e_{22} - z_{22} \end{pmatrix} \\
\mathbf{J}_{23,i} &= \\
& \begin{pmatrix} e_{32} - z_{32} & -e_{31} - z_{42} & e_{34} - z_{12} & z_{22} - e_{33} \\ z_{42} - e_{31} & -e_{32} - z_{32} & z_{22} - e_{33} & z_{12} - e_{34} \\ -e_{34} - z_{12} & z_{22} - e_{33} & e_{32} + z_{32} & e_{31} + z_{42} \\ e_{33} - z_{22} & -e_{34} - z_{12} & z_{42} - e_{31} & e_{32} - z_{32} \end{pmatrix} \\
\mathbf{J}_{24,i} &= \\
& \begin{pmatrix} e_{42} - z_{42} & z_{32} - e_{41} & e_{44} - z_{22} & -e_{43} - z_{12} \\ -e_{41} - z_{32} & -e_{42} - z_{42} & -e_{43} - z_{12} & z_{22} - e_{44} \\ z_{22} - e_{44} & z_{12} - e_{43} & e_{42} - z_{42} & e_{41} + z_{32} \\ e_{43} - z_{12} & z_{22} - e_{44} & z_{32} - e_{41} & e_{42} + z_{42} \end{pmatrix} \\
\mathbf{J}_{31,i} &= \\
& \begin{pmatrix} e_{13} - z_{13} & z_{23} - e_{14} & z_{33} - e_{11} & e_{12} + z_{43} \\ e_{14} + z_{23} & e_{13} + z_{13} & -e_{12} - z_{43} & z_{33} - e_{11} \\ z_{33} - e_{11} & z_{43} - e_{12} & z_{13} - e_{13} & -e_{14} - z_{23} \\ z_{43} - e_{12} & e_{11} - z_{33} & z_{23} - e_{14} & e_{13} + z_{13} \end{pmatrix} \\
\mathbf{J}_{32,i} &= \\
& \begin{pmatrix} e_{23} - z_{23} & -e_{24} - z_{13} & z_{43} - e_{21} & e_{22} - z_{33} \\ e_{24} - z_{13} & e_{23} + z_{23} & z_{33} - e_{22} & z_{43} - e_{21} \\ -e_{21} - z_{43} & z_{33} - e_{22} & -e_{23} - z_{23} & -e_{24} - z_{13} \\ z_{33} - e_{22} & e_{21} + z_{43} & z_{13} - e_{24} & e_{23} - z_{23} \end{pmatrix} \\
\mathbf{J}_{33,i} &= \\
& \begin{pmatrix} e_{33} - z_{33} & -e_{34} - z_{43} & -e_{31} - z_{13} & e_{32} + z_{23} \\ e_{34} + z_{43} & e_{33} - z_{33} & z_{23} - e_{32} & z_{13} - e_{31} \\ -e_{31} - z_{13} & z_{23} - e_{32} & z_{33} - e_{33} & z_{43} - e_{34} \\ -e_{32} - z_{23} & e_{31} - z_{13} & z_{43} - e_{34} & e_{33} - z_{33} \end{pmatrix} \\
\mathbf{J}_{34,i} &= \\
& \begin{pmatrix} e_{43} - z_{43} & z_{33} - e_{44} & -e_{41} - z_{23} & e_{42} - z_{13} \\ e_{44} - z_{33} & e_{43} - z_{43} & -e_{42} - z_{13} & z_{23} - e_{41} \\ z_{23} - e_{41} & z_{13} - e_{42} & -e_{43} - z_{43} & z_{33} - e_{44} \\ -e_{42} - z_{13} & e_{41} + z_{23} & z_{33} - e_{44} & e_{43} + z_{43} \end{pmatrix} \\
\mathbf{J}_{41,i} &= \\
& \begin{pmatrix} e_{14} - z_{14} & e_{13} + z_{24} & z_{34} - e_{12} & z_{44} - e_{11} \\ z_{24} - e_{13} & e_{14} + z_{14} & e_{11} - z_{44} & z_{34} - e_{12} \\ e_{12} + z_{34} & z_{44} - e_{11} & e_{14} + z_{14} & -e_{13} - z_{24} \\ z_{44} - e_{11} & -e_{12} - z_{34} & z_{24} - e_{13} & z_{14} - e_{14} \end{pmatrix} \\
\mathbf{J}_{42,i} &= \\
& \begin{pmatrix} e_{24} - z_{24} & e_{23} - z_{14} & z_{44} - e_{22} & -e_{21} - z_{34} \\ -e_{23} - z_{14} & e_{24} + z_{24} & e_{21} + z_{34} & z_{44} - e_{22} \\ e_{22} - z_{44} & z_{34} - e_{21} & e_{24} - z_{24} & -e_{23} - z_{14} \\ z_{34} - e_{21} & z_{44} - e_{22} & z_{14} - e_{23} & -e_{24} - z_{24} \end{pmatrix} \\
\mathbf{J}_{43,i} &= \\
& \begin{pmatrix} e_{34} - z_{34} & e_{33} - z_{44} & -e_{32} - z_{14} & z_{24} - e_{31} \\ z_{44} - e_{33} & e_{34} - z_{34} & e_{31} + z_{24} & z_{14} - e_{32} \\ e_{32} - z_{14} & z_{24} - e_{31} & e_{34} + z_{34} & z_{44} - e_{33} \\ -e_{31} - z_{24} & -e_{32} - z_{14} & z_{44} - e_{33} & -e_{34} - z_{34} \end{pmatrix} \\
\mathbf{J}_{44,i} &= \\
& \begin{pmatrix} e_{44} - z_{44} & e_{43} + z_{34} & -e_{42} - z_{24} & -e_{41} - z_{14} \\ -e_{43} - z_{34} & e_{44} - z_{44} & e_{41} - z_{14} & z_{24} - e_{42} \\ e_{42} + z_{24} & z_{14} - e_{41} & e_{44} - z_{44} & z_{34} - e_{43} \\ -e_{41} - z_{14} & z_{24} - e_{42} & z_{34} - e_{43} & z_{44} - e_{44} \end{pmatrix}
\end{aligned}$$

where $e_{jk}, z_{jk}, j, k = 1, 2, 3, 4$ are matrix entries of \mathbf{E}_i and \mathbf{Z}_i respectively. Note that these computation procedures can also be found out at https://github.com/zarathustr/hand_eye_SO4.

B. Matrix Determinant Property

Given an arbitrary square matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$$

If \mathbf{D} is invertible, then the determinant of \mathbf{M} is

$$\det(\mathbf{M}) = \det(\mathbf{D}) \det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})$$

Inserting the above result into

$$\det(\lambda_{\mathbf{W},\max} \mathbf{I} - \mathbf{W}) = \det \left[\begin{pmatrix} \lambda_{\mathbf{W},\max} \mathbf{I} & -\mathbf{K} \\ -\mathbf{K}^T & \lambda_{\mathbf{W},\max} \mathbf{I} \end{pmatrix} \right]$$

gives (26).

ACKNOWLEDGMENT

This research was supported by Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ20160401100022706, in part by National Natural Science Foundation of China under the grants of No. U1713211 and 41604025. The authors thank to Dr. Zhiqiang Zhang from University of Leeds for his detailed explanation of the implemented codes in [17]. The authors are grateful to Dr. Huy Nguyen from Nanyang Technological University, Singapore, for the discussion with him on his useful codes of the uncertainty descriptions of hand-eye calibration [19]. The authors also would like to thank Dr. Dario Modenini from University of Bologna and Prof. Daniel Condurache from Gheorghe Asachi Technical University of Iasi for constructive communications. The codes and data of this paper have been archived on https://github.com/zarathustr/hand_eye_SO4 and https://github.com/zarathustr/hand_eye_data.

REFERENCES

- [1] K. Koide and E. Menegatti, "General Hand-Eye Calibration Based on Reprojection Error Minimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1021–1028, 2019.
- [2] Y.-T. Liu, Y.-A. Zhang, and M. Zeng, "Sensor to Segment Calibration for Magnetic and Inertial Sensor Based Motion Capture Systems," *Measurement*, 2019. DOI: [10.1016/j.measurement.2019.03.048](https://doi.org/10.1016/j.measurement.2019.03.048)
- [3] Z. Q. Zhang, "Cameras and Inertial/Magnetic Sensor Units Alignment Calibration," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 6, pp. 1495–1502, 2016.
- [4] D. Modenini, "Attitude Determination from Ellipsoid Observations: A Modified Orthogonal Procrustes Problem," *AIAA J. Guid. Control Dyn.*, vol. 41, no. 10, pp. 2320–2325, 2018.
- [5] R. Y. Tsai and R. K. Lenz, "A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, 1989.
- [6] Y. C. Shiu and S. Ahmad, "Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, 1989.
- [7] F. C. Park and B. J. Martin, "Robot Sensor Calibration: Solving $AX = XB$ on the Euclidean Group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, 1994.
- [8] R. Horaud and F. Dornaika, "Hand-eye Calibration," *Int. J. Robot. Research*, vol. 14, no. 3, pp. 195–210, 1995.

[9] J. C. Chou and M. Kamel, "Finding the Position and Orientation of a Sensor on A Robot Manipulator Using Quaternions," *Int. J. Robot. Research*, vol. 10, no. 3, pp. 240–254, 1991.

[10] Y.-C. Lu and J. Chou, "Eight-space Quaternion Approach for Robotic Hand-eye Calibration," *IEEE ICSMC 2002*, pp. 3316–3321, 2002.

[11] K. Daniilidis, "Hand-eye Calibration Using Dual Quaternions," *Int. J. Robot. Research*, vol. 18, no. 3, pp. 286–298, 1999.

[12] N. Andreff, R. Horaud, and B. Espiau, "Robot Hand-eye Calibration Using Structure-from-Motion," *Int. J. Robot. Research*, vol. 20, no. 3, pp. 228–248, 2001.

[13] D. Condurache and A. Burlacu, "Orthogonal Dual Tensor Method for Solving the $AX = XB$ Sensor Calibration Problem," *Mech. Machine Theory*, vol. 104, pp. 382–404, 2016.

[14] S. Gwak, J. Kim, and F. C. Park, "Numerical Optimization on the Euclidean Group with Applications to Camera Calibration," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 65–74, 2003.

[15] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and Robot-World Calibration by Global Polynomial Optimization," *IEEE ICRA 2014*, pp. 3157–3164, 2014.

[16] Z. Zhao, "Simultaneous Robot-World and Hand-eye Calibration by the Alternative Linear Programming," *Pattern Recogn. Lett.*, 2018. DOI: [10.1016/j.patrec.2018.08.023](https://doi.org/10.1016/j.patrec.2018.08.023)

[17] Z. Zhang, L. Zhang, and G. Z. Yang, "A Computationally Efficient Method for Hand-eye Calibration," *Int. J. Comput. Assist. Radio. Surg.*, vol. 12, no. 10, pp. 1775–1787, 2017.

[18] H. Song, Z. Du, W. Wang, and L. Sun, "Singularity Analysis for the Existing Closed-Form Solutions of the Hand-eye Calibration," *IEEE Access*, vol. 6, pp. 75407–75421, 2018.

[19] H. Nguyen and Q.-C. Pham, "On the Covariance of X in $AX = XB$," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1651–1658, 2018.

[20] Y. Tian, W. R. Hamel, and J. Tan, "Accurate Human Navigation Using Wearable Monocular Visual and Inertial Sensors," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 1, pp. 203–213, 2014.

[21] A. Alamri, M. Eid, R. Iglesias, S. Shirmohammadi, and A. E. Saddik, "Haptic Virtual Rehabilitation Exercises for Post- stroke Diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 9, pp. 1–10, 2007.

[22] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler Principal Component Analysis," *Int. J. Comput. Vision*, vol. 101, no. 3, pp. 498–518, 2013.

[23] A. Bartoli, D. Pizarro, and M. Loog, "Stratified Generalized Procrustes Analysis," *Int. J. Comput. Vision*, vol. 101, no. 2, pp. 227–253, 2013.

[24] L. Igual, X. Perez-Sala, S. Escalera, C. Angulo, and F. De La Torre, "Continuous Generalized Procrustes analysis," *Pattern Recogn.*, vol. 47, no. 2, pp. 659–671, 2014.

[25] C. I. Mosier, "Determining A Simple Structure When Loadings for Certain Tests Are Known," *Psychometrika*, vol. 4, no. 2, pp. 149–162, 1939.

[26] B. F. Green, "The Orthogonal Approximation of An Oblique Structure in Factor Analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.

[27] M. W. Browne, "On Oblique Procrustes Rotation," *Psychometrika*, vol. 32, no. 2, pp. 125–132, 1967.

[28] G. Wahba, "A Least Squares Estimate of Satellite Attitude," *SIAM Rev.*, vol. 7, no. 3, p. 409, 1965.

[29] M. D. Shuster and S. D. Oh, "Three-axis Attitude Determination from Vector Observations," *AIAA J. Guid. Control Dyn.*, vol. 4, no. 1, pp. 70–77, 1981.

[30] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and Articulated Point Registration with Expectation Conditional Maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 587–602, 2011.

[31] N. Duta, A. K. Jain, and M. P. Dubuisson-Jolly, "Automatic Construction of 2D Shape Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 5, pp. 433–446, 2001.

[32] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[33] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[34] M. Wang and A. Tayebi, "Hybrid Pose and Velocity-bias Estimation on $SE(3)$ Using Inertial and Landmark Measurements," *IEEE Trans. Autom. Control*, 2018. DOI: [10.1109/TAC.2018.2879766](https://doi.org/10.1109/TAC.2018.2879766)

[35] J. Markdahl and X. Hu, "Exact Solutions to A Class of Feedback Systems on $SO(n)$," *Automatica*, vol. 63, pp. 138–147, 2016.

[36] J. D. Biggs and H. Henninger, "Motion Planning on a Class of 6-D Lie Groups via A Covering Map," *IEEE Trans. Autom. Control*, 2018. DOI: [10.1109/TAC.2018.2885241](https://doi.org/10.1109/TAC.2018.2885241)

[37] F. Thomas, "Approaching Dual Quaternions from Matrix Algebra," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1037–1048, 2014.

[38] W. S. Massey, "Cross Products of Vectors in Higher Dimensional Euclidean Spaces," *America. Math. Monthly*, vol. 90, no. 10, pp. 697–701, 1983.

[39] J. Wu, Z. Zhou, B. Gao, R. Li, Y. Cheng, and H. Fourati, "Fast Linear Quaternion Attitude Estimator Using Vector Observations," *IEEE Trans. Auto. Sci. Eng.*, vol. 15, no. 1, pp. 307–319, 2018.

[40] J. Wu, M. Liu, Z. Zhou, and R. Li, "Fast Symbolic 3D Registration Solution," *IEEE Trans. Auto. Sci. Eng.*, 2019. arxiv: [1805.08703](https://arxiv.org/abs/1805.08703)

[41] I. Y. Bar-Itzhack, "New Method for Extracting the Quaternion from a Rotation Matrix," *AIAA J. Guid. Control Dyn.*, vol. 23, no. 6, pp. 1085–1087, 2000.

[42] A. H. J. D. Ruiter and J. Richard, "On the Solution of Wahba's Problem on $SO(n)$," *J. Astronautical Sci.*, no. December, pp. 734–763, 2014.

[43] J. Wu, "Optimal Continuous Unit Quaternions from Rotation Matrices," *AIAA J. Guid. Control Dyn.*, vol. 42, no. 4, pp. 919–922, 2019.

[44] J. Wu, M. Liu, and Y. Qi, "Computationally Efficient Robust Algorithm for Generalized Sensor Calibration Problem $AR = RB$," *IEEE Sensors J.*, 2019. DOI: [10.1109/RG.2.2.17632.74240](https://doi.org/10.1109/RG.2.2.17632.74240)

[45] P. Lourenço, B. J. Guerreiro, P. Batista, P. Oliveira, and C. Silvestre, "Uncertainty Characterization of The Orthogonal Procrustes Problem with Arbitrary Covariance Matrices," *Pattern Recogn.*, vol. 61, pp. 210–220, 2017.

[46] R. L. Dailey, "Eigenvector Derivatives with Repeated Eigenvalues," *AIAA J.*, vol. 27, no. 4, pp. 486–491, 1989.

[47] G. Chang, T. Xu, and Q. Wang, "Error Analysis of Davenport's q-method," *Automatica*, vol. 75, pp. 217–220, 2017.

[48] X.-S. Gao, X.-R. Hou, J. Tang, and H.-f. Cheng, "Complete Solution Classification for the Perspective-Three-Point Problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, 2003.

[49] T. Hamel and C. Samson, "Riccati Observers for the Nonstationary PnP Problem," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 726–741, 2018.

[50] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[51] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[52] P. Payeur and C. Chen, "Registration of Range Measurements with Compact Surface Representation," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 5, pp. 1627–1634, 2003.

[53] L. Wei, C. Cappelle, and Y. Ruichek, "Camera/Laser/GPS Fusion Method for Vehicle Positioning under Extended NIS-Based Sensor Validation," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 11, pp. 3110–3122, 2013.

[54] A. Wan, J. Xu, D. Miao, and K. Chen, "An Accurate Point-Based Rigid Registration Method for Laser Tracker Relocation," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 2, pp. 254–262, 2017.

[55] Y. Zhuang, N. Jiang, H. Hu, and F. Yan, "3-D-Laser-Based Scene Measurement and Place Recognition for Mobile Robots in Dynamic Indoor Environments," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 438–450, 2013.

[56] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic Calibration of 2-D Laser Rangefinder and Camera From Single Shot Based on Minimal Solution," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 915–929, 2016.

[57] S. Xie, D. Yang, K. Jiang, and Y. Zhong, "Pixels and 3-D Points Alignment Method for the Fusion of Camera and LiDAR Data," *IEEE Trans. Instrum. Meas.*, 2018. DOI: [10.1109/TIM.2018.2879705](https://doi.org/10.1109/TIM.2018.2879705)

[58] Yuanxin Wu, Xiaoping Hu, Dewen Hu, Tao Li, and Junxiang Lian, "Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions," *IEEE Trans. Aerosp. Elec. Syst.*, vol. 41, no. 1, pp. 110–132, 2005.

[59] N. Enayati, E. D. Momi, and G. Ferrigno, "A Quaternion-Based Unscented Kalman Filter for Robust Optical/Inertial Motion Tracking in Computer-Assisted Surgery," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 8, pp. 2291–2301, 2015.

[60] J. Zhang, M. Kaess, and S. Singh, "A Real-time Method for Depth Enhanced Visual Odometry," *Auto. Robots*, vol. 41, no. 1, pp. 31–43, 2017.



Jin Wu was born in May, 1994 in Zhenjiang, China. He received the B.S. Degree from University of Electronic Science and Technology of China, Chengdu, China. He has been a research assistant in Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology since 2018. His research interests include robot navigation, multi-sensor fusion, automatic control and mechatronics. He is a co-author of over 30 technical papers in representative journals and conference proceedings of IEEE, AIAA, IET and etc. Mr. Jin Wu received the outstanding reviewer award for ASIAN JOURNAL OF CONTROL. One of his papers published in IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING was selected as the ESI Highly Cited Paper by ISI Web of Science during 2017 to 2018. He is a member of IEEE.



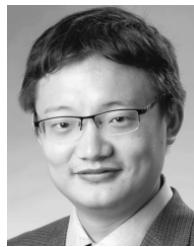
Xuxiang Sun received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017. He is currently a Research Associate with the Robotics Institute, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include mobile robots, autonomous vehicles, deep learning, SLAM and navigation, motion detection, and so on.

Dr. Sun is a recipient of the Best Student Paper Finalist Award at the IEEE ROBIO 2015.



estimation and control.

He has been an author of technical papers in flagship journals and conference proceedings including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE CDC and etc.



Ming Liu received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zürich, Zürich, Switzerland, in 2013, supervised by Prof. Roland Siegwart. During his master's study with Tongji University, he stayed one year with the Erlangen-Nürnberg University and Fraunhofer Institute IISB, Erlangen, Germany, as a Master Visiting Scholar.

He is currently with the Electronic and Computer Engineering, Computer Science and Engineering Department, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong, as an Assistant Professor. He is also a founding member of Shanghai Swing Automation Ltd., Co. He is coordinating and involved in NSF Projects and National 863-Hi-TechPlan Projects in China. His research interests include dynamic environment modeling, deep-learning for robotics, 3-D mapping, machine learning, and visual control.

Dr. Liu was a recipient of the European Micro Aerial Vehicle Competition (EMAV'09) (second place) and two awards from International Aerial Robot Competition (IARC'14) as a Team Member, the Best Student Paper Award as first author for MFI 2012 (IEEE International Conference on Multisensor Fusion and Information Integration), the Best Paper Award in Information for IEEE International Conference on Information and Automation (ICIA 2013) as first author, the Best Paper Award Finalists as co-author, the Best RoboCup Paper Award for IROS 2013 (IEEE/RSJ International Conference on Intelligent Robots and Systems), the Best Conference Paper Award for IEEE-CYBER 2015, the Best Student Paper Finalist for RCAR 2015 (IEEE International conference on Real-time Computing and Robotics), the Best Student Paper Finalist for ROBIO 2015, the Best Student Paper Award for IEEE-ICAR 2017, the Best Paper in Automation Award for IEEE-ICIA 2017, twice the innovation contest Chunhui Cup Winning Award in 2012 and 2013, and the Wu Wenjun AI Award in 2016. He was the Program Chair of IEEERCAR 2016 and the Program Chair of International Robotics Conference in Foshan 2017. He was the Conference Chair of ICVS 2017. He has published many popular papers in top robotics journals including IEEE TRANSACTIONS ON ROBOTICS, INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH and IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. Dr. Liu is currently an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS. He is a Senior Member of IEEE.