Hand-Eye Calibration: 4-D Procrustes Analysis Approach

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

***Abstract*— We give a universal analytical solution to the hand-eye calibration problem $AX = XB$ with known matrices A and B and unknown variable X, all in the set of special Euclidean group SE(3). The developed method relies on the 4-D Procrustes analysis. A unit-octonion representation is proposed for the first time to solve such a Procrustes problem through which an optimal closed-form eigendecomposition solution is derived. By virtue of such a 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 and better description of uncertainty and consumes much less computation time.**

***Index Terms*— Hand-eye calibration, homogenous transformation, least squares, octonions, quaternions.**

I. Introduction

THE main hand-eye calibration problem studied in this paper is aimed to compute the unknown relative homogeneous transformation X between a robotic gripper and an 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 the 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 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 the representatives of rotation-logarithm-based ones, such as Tsai and Lenz [5], Shiu and Ahmad [6], Park and Martin [7], Horaud and Dornaika [8], and quaternion-based one from Chou and Kamel [9]. The simultaneous ones appear in the second group with the related representatives of as follows.

1) *Analytical Solutions*: Quaternion-based method by Lu and Chou [10], dual-quaternion-based one by Daniilidis [11], Sylvester-equation-based one by Andreff *et al*. [12], and dual-tensor-based one by Condurache and Burlacu [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], and pseudo-inverse-based one by Zhang [3] and Zhang *et al*. [17].

Each kind of algorithms has its own pros and cons. The separated ones cannot 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 a large quantity of time when using numerical iterations. Some algorithms will also suffer from their own ill-posed conditions in the presence of some extreme data sets [18]. What is more, the uncertainty description of X in a hand-eye problem $AX = XB$, being an important but difficult problem, has always troubled researchers until the first public general iterative solution by Nguyen and Pham [19]. An intuitive overview of these algorithms in the order of publication time can be found out in Table I.

Until 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 studies around it are still

Manuscript received April 24, 2019; revised June 18, 2019; accepted July 11, 2019. Date of publication August 6, 2019; date of current version May 12, 2020. This work was supported by the National Natural Science Foundation of China (Grant No. U1713211), the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, and No. 21202816, and Shenzhen Science, Technology and Innovation Commission (SZSTI) JCYJ20160428154842603, awarded to Prof. Ming Liu. The Associate Editor coordinating the review process was Peter Liu. *(Corresponding author: Ming Liu.)*

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

M. Wang is with the Department of Electronic and Computer Engineering, Western University, London, ON N6A 3K7, Canada (e-mail: mwang448@uwo.ca).

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

Digital Object Identifier 10.1109/TIM.2019.2930710

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
Horaud 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	Sylverster 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

very popular. There is still a remaining problem that no algorithm can simultaneously estimate $\boldsymbol{X}$ in $\boldsymbol{AX} = \boldsymbol{XB}$ while preserving highly accurate uncertainty descriptions and consuming extremely low computation time. These difficulties are rather practical since in the hand-eye problem $\boldsymbol{AX} = \boldsymbol{XB}$, the rotation and translation parts are tightly coupled with high nonlinearity, which motivates Nguyen and Pham 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-D Procrustes analysis tool for the 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 the data sets [23], [24]. Early studies on Procrustes analysis have been conducted since 1930s [25]–[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 so on. Motivated by these technological advances, this paper has the following contributions.

1) We show some analytical results to the 4-D 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 $\mathrm{SO}(n) := \{\boldsymbol{R} \in \mathbb{R}^{n\times n} | \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}$, where $\boldsymbol{I}$ denotes the identity matrix with proper size. The special Euclidean space is composed of a rotation matrix $\boldsymbol{R}$ and a translational vector $\boldsymbol{t}$, such that

$$\mathrm{SE}(n) := \left\{ \boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} | \boldsymbol{R} \in \mathrm{SO}(n), \boldsymbol{t} \in \mathbb{R}^n \right\} \tag{1}$$

with $\boldsymbol{0}$ denoting the zeros matrix with adequate dimensions. The Euclidean norm of a given squared matrix $\boldsymbol{X}$ will be defined with $\|\boldsymbol{X}\| = (\mathrm{tr}(\boldsymbol{X}^T\boldsymbol{X}))^{1/2}$, where tr denotes the matrix trace. The vectorization of an arbitrary matrix $\boldsymbol{X}$ is defined as $\mathrm{vec}(\boldsymbol{X})$, and $\otimes$ represents the kronecker product between two matrices. For a given arbitrary matrix $\boldsymbol{X}$, $\boldsymbol{X}^\dagger$ is called its Moore–Penrose generalized inverse. Any rotation $\boldsymbol{R}$ on SO(3) has its corresponding logarithm given by

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

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

$$[\boldsymbol{x}]_\times = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{3}$$

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

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

$$\begin{aligned} \mathcal{A} &= \{\boldsymbol{A}_i | \boldsymbol{A}_i \in \mathrm{SE}(3), i = 1, 2, \ldots, N\} \\ \mathcal{B} &= \{\boldsymbol{B}_i | \boldsymbol{B}_i \in \mathrm{SE}(3), i = 1, 2, \cdots, N\} \end{aligned} \tag{4}$$

consider the hand-eye calibration least square

$$\underset{\boldsymbol{X}\in\mathrm{SE}(3)}{\arg\min}\, \mathcal{J} = \sum_{i=1}^{N} \|\boldsymbol{A}_i\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}_i\|^2 \tag{5}$$

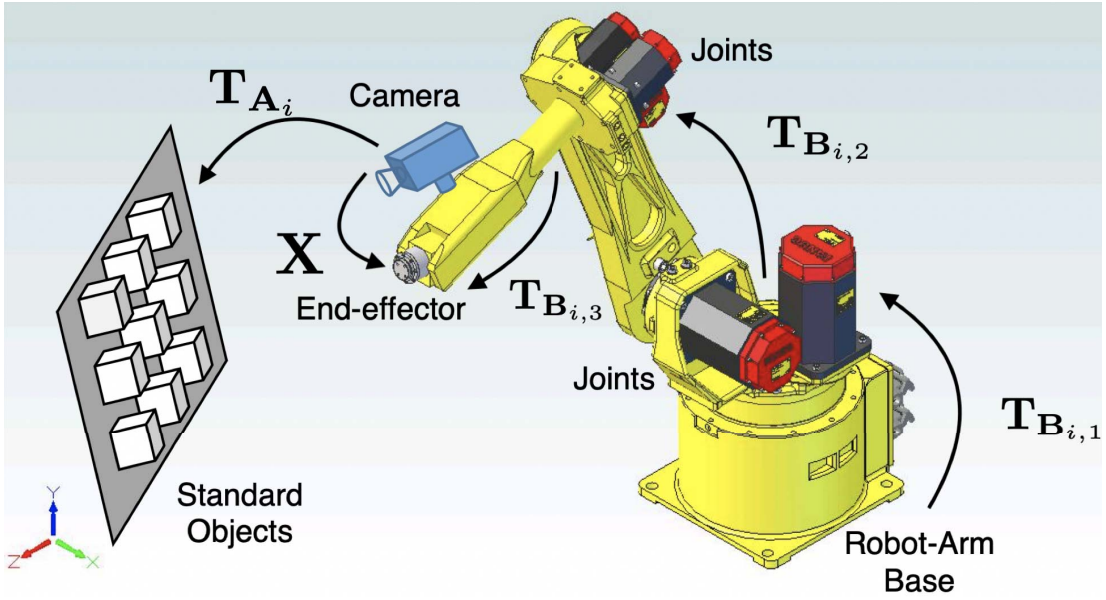


Fig. 1. Relationship between various homogeneous transformations for gripper–camera hand-eye calibration.

where $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ come to the reality using poses in two successive measurements (also see [11, Fig. 1])

$$\begin{aligned} \boldsymbol{A}_i &= \boldsymbol{T}_{\boldsymbol{A}_{i+1}} \boldsymbol{T}_{\boldsymbol{A}_i}^{-1} \\ \boldsymbol{B}_i &= \boldsymbol{T}_{\boldsymbol{B}_{i+1}}^{-1} \boldsymbol{T}_{\boldsymbol{B}_i} \end{aligned} \tag{6}$$

with $\boldsymbol{T}_{\boldsymbol{A}_i}$ being the ith camera pose with respect to the standard objects in a world frame and

$$\boldsymbol{T}_{\boldsymbol{B}_i} = \boldsymbol{T}_{\boldsymbol{B}_{i,3}} \boldsymbol{T}_{\boldsymbol{B}_{i,2}} \boldsymbol{T}_{\boldsymbol{B}_{i,1}} \tag{7}$$

are gripper poses with respect to the robotic base, in which $\boldsymbol{T}_{\boldsymbol{B}_{i,1}}$, $\boldsymbol{T}_{\boldsymbol{B}_{i,2}}$, and $\boldsymbol{T}_{\boldsymbol{B}_{i,3}}$ are the 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 $\boldsymbol{X}$ considering rotation and translation simultaneously and, moreover, derive the uncertainty description of $\boldsymbol{X}$.

Let us write $\boldsymbol{A}$ and $\boldsymbol{B}$ into

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R}_A & \boldsymbol{t}_A \\ \boldsymbol{0} & 1 \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} \boldsymbol{R}_B & \boldsymbol{t}_B \\ \boldsymbol{0} & 1 \end{pmatrix}. \tag{8}$$

Then, one easily obtains

$$\begin{cases} \boldsymbol{R}_A \boldsymbol{R}_X = \boldsymbol{R}_X \boldsymbol{R}_B \\ \boldsymbol{R}_A \boldsymbol{t}_X + \boldsymbol{t}_A = \boldsymbol{R}_X \boldsymbol{t}_B + \boldsymbol{t}_X \end{cases}. \tag{9}$$

The method by Park and Martin [7] first computes $\boldsymbol{R}_X$ from the first equation of (9) and then solves $\boldsymbol{t}_X$ by inserting $\boldsymbol{R}_X$ into the second sub-equation. The Park's step for computing $\boldsymbol{R}_X$ is tantamount to the following optimization:

$$\underset{\boldsymbol{R}_X \in \mathrm{SO}(3)}{\arg\min} \sum_{i=1}^{N} \| \boldsymbol{R}_X \boldsymbol{a}_i - \boldsymbol{b}_i \|^2 \tag{10}$$

with

$$\begin{aligned} \boldsymbol{a}_i &= [\log(\boldsymbol{R}_{\boldsymbol{A}_i})]^{\wedge} \\ \boldsymbol{b}_i &= [\log(\boldsymbol{R}_{\boldsymbol{B}_i})]^{\wedge}. \end{aligned} \tag{11}$$

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

$$\underset{\boldsymbol{R}_X \in \mathrm{SO}(3), \boldsymbol{t}_X \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} \begin{pmatrix} \| \boldsymbol{R}_X \boldsymbol{a}_i - \boldsymbol{b}_i \|^2 \\ + \| \boldsymbol{R}_X \boldsymbol{t}_B + \boldsymbol{t}_X - \boldsymbol{R}_A \boldsymbol{t}_X - \boldsymbol{t}_A \|^2 \end{pmatrix} \tag{12}$$

is not always convex. Hence, the iterative numerical methods are proposed to achieve the globally optimal estimates of $\boldsymbol{R}_X$ and $\boldsymbol{t}_X$, including solvers in [3], [14], [16], and [17]. In Section III, we show a new analytical perspective for hand-eye calibration problem $\boldsymbol{AX} = \boldsymbol{XB}$ using the proposed 4-D Procrustes analysis.

III. 4-D Procrustes Analysis

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

A. Some New Analytical Results

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

$$\underset{\boldsymbol{R} \in \mathrm{SO}(4), \boldsymbol{t} \in \mathbb{R}^4}{\arg\min} \sum_{i=1}^{N} \| \boldsymbol{R} \boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i \|^2. \tag{13}$$

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

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

with

$$\begin{aligned} \boldsymbol{USV}^T &= \boldsymbol{H} \\ \boldsymbol{H} &= \sum_{i=1}^{N} (\boldsymbol{u}_i - \bar{\boldsymbol{u}})(\boldsymbol{v}_i - \bar{\boldsymbol{v}})^T \\ \bar{\boldsymbol{u}} &= \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{u}_i, \bar{\boldsymbol{v}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_i \end{aligned} \tag{15}$$

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

$$\boldsymbol{R} = \boldsymbol{R}_L(\boldsymbol{q}_L)\boldsymbol{R}_R(\boldsymbol{q}_R) \tag{16}$$

with

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

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

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

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

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

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

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

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

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

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

$$\boldsymbol{H} = \left(\boldsymbol{H}_1^T, \boldsymbol{H}_2^T, \boldsymbol{H}_3^T, \boldsymbol{H}_4^T\right)^T. \tag{23}$$

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

$$\boldsymbol{W}\boldsymbol{\sigma} = \lambda_{\boldsymbol{W},\max}\boldsymbol{\sigma}. \tag{24}$$

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

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{K} \\ \boldsymbol{K}^T & \boldsymbol{0} \end{pmatrix} \tag{25}$$

where $\boldsymbol{K}$ is shown in (19), as shown at the bottom of this page. This indicates that $\lambda_{\boldsymbol{W},\max}$ subjects to

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

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

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

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

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

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

$$\underset{\boldsymbol{R}\in\mathrm{SO}(4)}{\arg\min} \sum_{i=1}^{N} \|\boldsymbol{E}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{Z}_i\|^2. \tag{27}$$

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

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

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

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

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

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

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

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

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

$$\boldsymbol{G}_{jk,i} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{J}_{jk,i} \\ \boldsymbol{J}_{jk,i}^T & \boldsymbol{0} \end{pmatrix} \tag{31}$$

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

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

where

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

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

$$\boldsymbol{F}\boldsymbol{\sigma} = \lambda_{\boldsymbol{F},\min}\boldsymbol{\sigma}. \tag{34}$$

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

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

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

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

$$\underset{\boldsymbol{R}\in\mathrm{SO}(4)}{\arg\min} \sum_{i=1}^{N} \|\boldsymbol{E}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{Z}_i\|^2 \tag{36}$$

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

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

$$\begin{aligned}
&\underset{\boldsymbol{E}_{i,i+1}\in\mathrm{SO}(4)}{\arg\max}\ \mathrm{tr}\begin{bmatrix} w\left(\boldsymbol{E}_i\boldsymbol{E}_{i,i+1}^T - \boldsymbol{I}\right)^T\left(\boldsymbol{E}_i\boldsymbol{E}_{i,i+1}^T - \boldsymbol{I}\right) \\ +(1-w)\left(\boldsymbol{E}_{i,i+1}\boldsymbol{E}_{i+1}^T - \boldsymbol{I}\right)^T\left(\boldsymbol{E}_{i,i+1}\boldsymbol{E}_{i+1}^T - \boldsymbol{I}\right) \end{bmatrix} \\
&\quad \Rightarrow \underset{\boldsymbol{E}_{i,i+1}\in\mathrm{SO}(4)}{\arg\max}\ \mathrm{tr}\{\boldsymbol{E}_{i,i+1}[w\boldsymbol{E}_i + (1-w)\boldsymbol{E}_{i+1}]^T\}
\end{aligned} \tag{37}$$

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

$$w = \frac{\tau_{\boldsymbol{E},i+1} - \tau_{\boldsymbol{Z},i}}{\tau_{\boldsymbol{E},i+1} - \tau_{\boldsymbol{E},i}}. \tag{38}$$

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

B. Uncertainty Descriptions

In this section, we use $\hat{\boldsymbol{x}}$ for representing the true value of the noise-disturbed vector $\boldsymbol{x}$. The expectation is denoted using $\langle \cdots \rangle$ [29], [45]. All the errors in this section are assumed to be zero-mean, which can be found out in [29]. As all the solutions provided in Section III-A 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 $\boldsymbol{q}$ is

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

provided that $\boldsymbol{m} = \mathrm{vec}(\boldsymbol{M})$, $\lambda_{\max}$ being the maximum eigenvalue of the real symmetric matrix $\boldsymbol{M}$

$$\boldsymbol{M}\boldsymbol{q} = \lambda_{\max}\boldsymbol{q}. \tag{40}$$

The above-mentioned quaternion error is presented, given the assumption that $\delta\boldsymbol{q}$ is multiplicative, such that

$$\hat{\boldsymbol{q}} = \delta\boldsymbol{q} \odot \boldsymbol{q} \tag{41}$$

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:

$$\begin{aligned}
&\boldsymbol{\Sigma}_{\delta\boldsymbol{q}} \\
&= \langle \delta\boldsymbol{q}\delta\boldsymbol{q}^T \rangle \\
&= \langle [\boldsymbol{q}^T \otimes (\lambda_{\max}\boldsymbol{I} - \boldsymbol{M})^\dagger]\delta\boldsymbol{m}\delta\boldsymbol{m}^T [\boldsymbol{q}^T \otimes (\lambda_{\max}\boldsymbol{I} - \boldsymbol{M})^\dagger]^T \rangle \\
&= [\boldsymbol{q}^T \otimes (\lambda_{\max}\boldsymbol{I} - \boldsymbol{M})^\dagger]\boldsymbol{\Sigma}_{\delta\boldsymbol{m}} \\
&\quad \times [\boldsymbol{q}^T \otimes (\lambda_{\max}\boldsymbol{I} - \boldsymbol{M})^\dagger]^T
\end{aligned} \tag{42}$$

in which

$$\boldsymbol{\Sigma}_{\delta\boldsymbol{m}} = \frac{\partial\boldsymbol{m}}{\partial\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b}}\left(\frac{\partial\boldsymbol{m}}{\partial\boldsymbol{b}}\right)^T \tag{43}$$

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

$$\boldsymbol{F}_{11}\boldsymbol{q}_L = \lambda_{\boldsymbol{F},\min}\boldsymbol{q}_L = \lambda_{\boldsymbol{F}_{11},\min}\boldsymbol{q}_L \tag{44}$$

which yields that

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

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

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

in which $\boldsymbol{m}_L = \mathrm{vec}(\boldsymbol{F}_{11}), \boldsymbol{m}_R = \mathrm{vec}(\boldsymbol{F}_{22})$, and $\boldsymbol{\Sigma}_{\delta\boldsymbol{m}_L\delta\boldsymbol{m}_R}$ is given by

$$\boldsymbol{\Sigma}_{\delta\boldsymbol{m}_L\delta\boldsymbol{m}_R} = \frac{\partial\boldsymbol{m}_L}{\partial\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b}}\left(\frac{\partial\boldsymbol{m}_R}{\partial\boldsymbol{b}}\right)^T. \tag{47}$$

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

$$\boldsymbol{\Sigma}_{\boldsymbol{\sigma}} = \begin{pmatrix} \boldsymbol{\Sigma}_{\delta\boldsymbol{q}_L} & \boldsymbol{\Sigma}_{\delta\boldsymbol{q}_L\delta\boldsymbol{q}_R} \\ \boldsymbol{\Sigma}_{\delta\boldsymbol{q}_R\delta\boldsymbol{q}_L} & \boldsymbol{\Sigma}_{\delta\boldsymbol{q}_R} \end{pmatrix}. \tag{48}$$

C. Solving $\boldsymbol{AX} = \boldsymbol{XB}$ From SO(4) Perspective

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

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} \underset{\mathcal{F}_1^{-1}}{\overset{\mathcal{F}_1}{\longleftrightarrow}} \boldsymbol{R}_{\boldsymbol{T},\mathrm{SO}(4)} = \begin{pmatrix} \boldsymbol{R} & \varepsilon\boldsymbol{t} \\ \varepsilon\boldsymbol{t}^T\boldsymbol{R} & 1 \end{pmatrix} \tag{49}$$

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 $\boldsymbol{R}_{\boldsymbol{T},\mathrm{SO}(4)}$

$$\boldsymbol{R}_{\boldsymbol{T},\mathrm{SO}(4)} \approx \begin{pmatrix} \boldsymbol{R} & \frac{1}{d}\boldsymbol{t} \\ \frac{1}{d}\boldsymbol{t}^T\boldsymbol{R} & 1 \end{pmatrix}. \tag{50}$$

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

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} \underset{\mathcal{F}_2^{-1}}{\overset{\mathcal{F}_2}{\longleftrightarrow}} \boldsymbol{R}_{\boldsymbol{T},\mathrm{SO}(4)} = \begin{pmatrix} \boldsymbol{R} & \varepsilon\boldsymbol{t} \\ -\varepsilon\boldsymbol{t}^T\boldsymbol{R} & 1 \end{pmatrix}. \tag{51}$$

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 $\boldsymbol{AX} = \boldsymbol{XB}$ can be shifted to

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

which can be instantly solved via the solution to Problem 2. With asynchronously sampled measurements, the problem can be refined with solution to Sub-Problem 2. While the uncertainty descriptions of $\boldsymbol{\sigma}$ related to $\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}$ are shown in Section III-B, we would reveal what the covariances of the rotation and translation look like. Suppose that, now, we have obtained the covariance of $\boldsymbol{\sigma}$ in (48) for $\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}$. The covariances between the columns of the rotation matrix and the cross-covariances between the columns of rotation and translation are considered. Recalling (20), one finds out that the covariance between the ith column and jth column of $\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}$ is calculated by

$$\begin{aligned}&\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_i\delta\boldsymbol{c}_j}\\ &= \langle[\boldsymbol{P}_i(\delta\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}][\boldsymbol{P}_j(\delta\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_j(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}]^T\rangle\\ &= \left\langle\begin{array}{c}\boldsymbol{P}_i(\delta\boldsymbol{\sigma})\boldsymbol{\sigma}\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\delta\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\boldsymbol{\sigma})\\ +\boldsymbol{P}_i(\delta\boldsymbol{\sigma})\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\delta\boldsymbol{\sigma})\end{array}\right\rangle\\ &= \left\langle\begin{array}{c}\boldsymbol{Y}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{Y}_j^T(\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\boldsymbol{\sigma})\\ +\boldsymbol{Y}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{P}_j^T(\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\boldsymbol{Y}_j^T(\boldsymbol{\sigma})\end{array}\right\rangle\\ &= \boldsymbol{Y}_i(\boldsymbol{\sigma})\boldsymbol{\Sigma}_{\boldsymbol{\sigma}}\boldsymbol{Y}_j^T(\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\boldsymbol{\Sigma}_{\boldsymbol{\sigma}}\boldsymbol{P}_j^T(\boldsymbol{\sigma})\\ &\quad+\boldsymbol{Y}_i(\boldsymbol{\sigma})\boldsymbol{\Sigma}_{\boldsymbol{\sigma}}\boldsymbol{P}_j^T(\boldsymbol{\sigma})+\boldsymbol{P}_i(\boldsymbol{\sigma})\boldsymbol{\Sigma}_{\boldsymbol{\sigma}}\boldsymbol{Y}_j^T(\boldsymbol{\sigma})\end{aligned} \tag{53}$$

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

$$\boldsymbol{Y}_i(\boldsymbol{\sigma}) = \boldsymbol{P}_i(\boldsymbol{\sigma}). \tag{54}$$

Therefore, (53) can be interpreted as

$$\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_i\delta\boldsymbol{c}_j} = 4\boldsymbol{P}_i(\boldsymbol{\sigma})\boldsymbol{\Sigma}_{\boldsymbol{\sigma}}\boldsymbol{P}_j^T(\boldsymbol{\sigma}). \tag{55}$$

In particular, the rotation–translation cross-covariances will be described by taking the first three rows and three columns of covariance matrices of $d\cdot\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_1\delta\boldsymbol{c}_4}$, $d\cdot\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_2\delta\boldsymbol{c}_4}$, $d\cdot\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_3\delta\boldsymbol{c}_4}$, respectively. More specifically, if we need to obtain the covariance of $\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}$, one arrives at

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

where

$$\begin{aligned}&\delta\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}\\ &= \begin{bmatrix}\delta\boldsymbol{P}_1(\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_1(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}, \delta\boldsymbol{P}_2(\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_2(\boldsymbol{\sigma})\delta\boldsymbol{\sigma},\\ \delta\boldsymbol{P}_3(\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_3(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}, \delta\boldsymbol{P}_4(\boldsymbol{\sigma})\boldsymbol{\sigma}+\boldsymbol{P}_4(\boldsymbol{\sigma})\delta\boldsymbol{\sigma}\end{bmatrix}\\ &= \begin{Bmatrix}[\boldsymbol{Y}_1(\boldsymbol{\sigma})+\boldsymbol{P}_1(\boldsymbol{\sigma})]\delta\boldsymbol{\sigma}, [\boldsymbol{Y}_2(\boldsymbol{\sigma})+\boldsymbol{P}_2(\boldsymbol{\sigma})]\delta\boldsymbol{\sigma},\\ [\boldsymbol{Y}_3(\boldsymbol{\sigma})+\boldsymbol{P}_3(\boldsymbol{\sigma})]\delta\boldsymbol{\sigma}, [\boldsymbol{Y}_4(\boldsymbol{\sigma})+\boldsymbol{P}_4(\boldsymbol{\sigma})]\delta\boldsymbol{\sigma}\end{Bmatrix}.\end{aligned} \tag{57}$$

The covariance of $\boldsymbol{R}_{\boldsymbol{X}}$ then equals to

$$\boldsymbol{\Sigma}_{\boldsymbol{R}_{\boldsymbol{X}}} = \boldsymbol{\Sigma}_{\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}}(1{:}3, 1{:}3) \tag{58}$$

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

$$\boldsymbol{\Sigma}_{\boldsymbol{t}_{\boldsymbol{X}}} = d^2\boldsymbol{\Sigma}_{\delta\boldsymbol{c}_4\delta\boldsymbol{c}_4}(1{:}3, 1{:}3). \tag{59}$$

D. Discussion

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

1) It can simultaneously solve rotation and translation in $\boldsymbol{X}$ for hand-eye calibration problem $\boldsymbol{AX} = \boldsymbol{XB}$ and thus own comparable accuracy and robustness with previous representatives.
2) All the items from $\boldsymbol{A}$ and $\boldsymbol{B}$ are directly propagated to the final forms of eigendecomposition without any pre-processing techniques, e.g., quaternion conversion from rotation, rotation logarithm remaining in the previous studies.
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 $\boldsymbol{X}$ 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 the 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 $d = 1\times10^5 \sim 1\times10^6 \approx 2^{16} \sim 2^{20}$ will be redundant enough, guaranteeing the accuracy of at least 2^{20-32} m $= 2^{-12}$ m $= 2.44\times10^{-04}$ m, which is enough for most optical systems with measurement ranges of 10 m. How to choose the most appropriate d value 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 the links in the Acknowledgment section for some MATLAB codes.

IV. Experimental Results

A. Experiments on a Robotic Arm

The first category of experiments is conducted for a gripper–camera hand-eye calibration shown in Fig. 1.

Algorithm 1 Proposed 4DPA Method for Hand-Eye Calibration

Parameter: Empirical value of d.

Require:

1) Get N measurements of $\boldsymbol{A}_i, \boldsymbol{B}_i$ 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 SE(3) − SO(4) mapping.

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

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

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

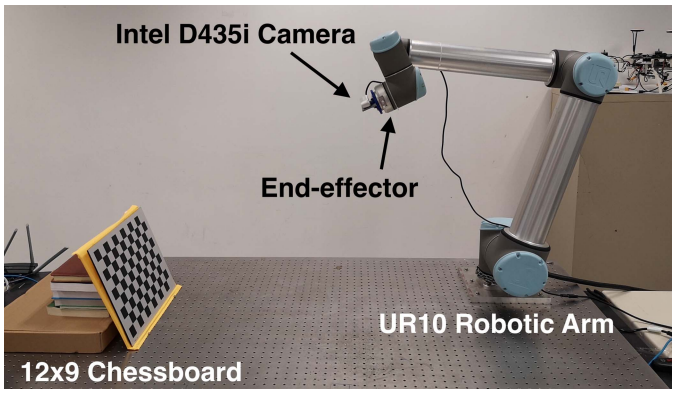


Fig. 2. Gripper–camera hand-eye calibration experiment.

The data set is generated using a UR10 robotic arm and an Intel Realsense D435i camera attached firmly to the end-effector (gripper) of the robotic arm (see Fig. 2).

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 an inertial measurement unit (IMU). In this section, the transformation of the end-effector $\boldsymbol{T}_{\boldsymbol{B}_i}$ 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:

$$\boldsymbol{l}_{\text{cam},j} = \begin{pmatrix} l_{\text{cam},1,j} \\ l_{\text{cam},2,j} \end{pmatrix}$$
$$\begin{pmatrix} l_{\text{cam},1,j} \\ l_{\text{cam},2,j} \\ 1 \end{pmatrix} = \boldsymbol{O}\left(\frac{L_{\text{cam},1,j}}{L_{\text{cam},3,j}}, \frac{L_{\text{cam},2,j}}{L_{\text{cam},3,j}}, 1\right)^T \tag{60}$$

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

$$\boldsymbol{L}_{\text{chess},j} = \boldsymbol{T}_{\boldsymbol{A}_i} \boldsymbol{L}_{\text{cam},j}. \tag{61}$$

By minimizing the projection errors from (61), $\boldsymbol{T}_{\boldsymbol{A}_i}$ 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 the extraction of the corner points of the chessboard [50]. We use several data sets captured from our platform to produce comparisons with representatives, including classical ones of Tsai and Lenz [5], Park and Martin [7], Chou and Kamel [9], Daniilidis [11], and Andreff *et al.* [12] and recent ones of Heller *et al.* [15], Zhao [16], and Zhang *et al.* [17]. The error of the hand-eye calibration is defined as follows:

$$\text{Error} = \frac{1}{N}\sqrt{\sum_{i=1}^{N} \|\boldsymbol{A}_i \boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}_i\|^2} \tag{62}$$

where $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are detailed in (6). All the timing statistics, computation and visualization are carried out on a MacBook Pro 2017 with i7-3.5-GHz CPU along with the MATLAB r2018a software. All the algorithms are implemented using the least coding resources. We employ `YALMIP` to solve the 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 above-described mechanisms, we form the series of $\{\mathcal{A}\}, \{\mathcal{B}\}$. We select $d = 10^4$ as a scaling factor for evaluation in this section, as the translational components are all within $[-2, 2]$ m, and in such a 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 eight 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 Tables 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 Tables II and III 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 $\boldsymbol{X}$. While with

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}	4.0506×10^{-02}	6.8255×10^{-03}	1.0190×10^{-02}	6.8254×10^{-03}	3.5959×10^{-02}	$\mathbf{6.7082 \times 10^{-03}}$	3.8308×10^{-02}	7.2650×10^{-03}	5.7292×10^{-03}	6.7857×10^{-03}	$\mathbf{4.4819 \times 10^{-03}}$
2 (253)	1.0243×10^{-02}	4.2952×10^{-02}	5.8650×10^{-03}	1.1760×10^{-02}	5.8650×10^{-03}	3.9486×10^{-02}	$\mathbf{5.7704 \times 10^{-03}}$	4.1492×10^{-02}	5.8754×10^{-03}	6.8972×10^{-03}	5.8290×10^{-03}	$\mathbf{4.8991 \times 10^{-03}}$
3 (298)	8.0653×10^{-03}	4.9823×10^{-02}	5.0514×10^{-03}	1.3105×10^{-02}	5.0517×10^{-03}	4.6559×10^{-02}	$\mathbf{4.9084 \times 10^{-03}}$	4.8626×10^{-02}	5.4648×10^{-03}	7.2326×10^{-03}	4.9803×10^{-03}	$\mathbf{5.5470 \times 10^{-03}}$
4 (342)	6.8136×10^{-03}	5.3854×10^{-02}	4.7192×10^{-03}	1.4585×10^{-02}	4.7254×10^{-03}	5.6970×10^{-02}	4.0379×10^{-03}	5.3721×10^{-02}	4.8374×10^{-03}	6.7880×10^{-03}	$\mathbf{4.0207 \times 10^{-03}}$	$\mathbf{5.5404 \times 10^{-03}}$
5 (392)	5.5242×10^{-03}	6.3039×10^{-02}	3.4047×10^{-03}	1.7253×10^{-02}	3.4012×10^{-03}	6.0697×10^{-02}	3.3850×10^{-03}	6.4505×10^{-02}	4.0410×10^{-03}	8.9159×10^{-03}	$\mathbf{3.1678 \times 10^{-03}}$	$\mathbf{7.3957 \times 10^{-03}}$
6 (433)	4.8072×10^{-03}	6.7117×10^{-02}	2.9723×10^{-03}	1.8354×10^{-02}	2.9694×10^{-03}	6.9967×10^{-02}	2.8957×10^{-03}	6.5703×10^{-02}	3.6410×10^{-03}	9.8837×10^{-03}	$\mathbf{2.7890 \times 10^{-03}}$	$\mathbf{7.8954 \times 10^{-03}}$
7 (470)	4.3853×10^{-03}	7.6869×10^{-02}	2.7302×10^{-03}	1.9827×10^{-02}	2.7270×10^{-03}	1.9827×10^{-02}	2.6640×10^{-03}	7.5553×10^{-02}	3.3262×10^{-03}	1.0373×10^{-02}	$\mathbf{2.5397 \times 10^{-03}}$	$\mathbf{8.1632 \times 10^{-03}}$
8 (500)	4.0938×10^{-03}	8.5545×10^{-02}	2.5855×10^{-03}	2.1506×10^{-02}	2.5807×10^{-03}	7.5722×10^{-02}	2.4610×10^{-03}	7.7225×10^{-02}	3.0083×10^{-03}	1.2008×10^{-02}	$\mathbf{2.3137 \times 10^{-03}}$	$\mathbf{8.5382 \times 10^{-03}}$

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

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 $\boldsymbol{X}$ 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 1–3 using the method of Daniilidis. This indicates that with a few samples, the accuracy of the proposed 4DPA is lower than that of the method of Daniilidis but is still close. However, a 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 4-D 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 eigendecomposition as the solution that is regarded as a robust tool for engineering problems. Our method can reduce the estimation error to about 3.06%–94.01% of original stats compared with the classical algorithms and 0.39%–86.11% compared with the recent numerical ones. The proposed method is also free of pre-processing techniques such as 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%–70.68% of original stats compared with the classical analytical algorithms and 1.45%–28.58% compared with the recent numerical ones. A synchronized sequence of the camera–chessboard poses and end-effector poses is made open-source (see the links in the Acknowledgment section). Every researcher can freely download this data set 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. Error Sensitivity to the Noises and Different Parameters of d

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

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{A},i} &= \hat{\boldsymbol{R}}_{\boldsymbol{A},i} + \text{Error}_{\boldsymbol{R}_{\boldsymbol{X}}} \mathcal{R}_1 \\ \boldsymbol{R}_{\boldsymbol{B},i} &= \hat{\boldsymbol{R}}_{\boldsymbol{B},i} + \text{Error}_{\boldsymbol{R}_{\boldsymbol{X}}} \mathcal{R}_2 \end{aligned} \tag{63}$$

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

$$\begin{aligned} \boldsymbol{t}_{\boldsymbol{A},i} &= \hat{\boldsymbol{t}}_{\boldsymbol{A},i} + \text{Error}_{\boldsymbol{t}_{\boldsymbol{X}}} \mathcal{T}_1 \\ \boldsymbol{t}_{\boldsymbol{B},i} &= \hat{\boldsymbol{t}}_{\boldsymbol{B},i} + \text{Error}_{\boldsymbol{t}_{\boldsymbol{X}}} \mathcal{T}_2 \end{aligned} \tag{64}$$

with noise scale of $\text{Error}_{\boldsymbol{t}_{\boldsymbol{X}}}$ and $\mathcal{T}_1$ and $\mathcal{T}_2$ noise vectors subject to normal distribution also with a covariance of $\boldsymbol{I}$. 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 Section IV-A to this part by adding three more ones of the proposed method with different d values of $d = 10^3$, $d = 10^5$, and $d = 10^6$. Then, we can see both the comparisons with the representatives and observe the influence of the positive scaling factor d. Several simulations are conducted where we generate data sets of $\mathcal{A}, \mathcal{B}$ with $N = 1000$, and the obtained results are averaged for ten times. We independently evaluate the effect of $\text{Error}_{\boldsymbol{R}_{\boldsymbol{X}}}$ and $\text{Error}_{\boldsymbol{t}_{\boldsymbol{X}}}$ imposed on Error. The relationship between $\text{Error}_{\boldsymbol{R}_{\boldsymbol{X}}}$ and Error is shown in Fig. 3, while the relationship between $\text{Error}_{\boldsymbol{t}_{\boldsymbol{X}}}$ and Error is shown 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 $\boldsymbol{X}$ value all arise to a large extent. One can see in the magnified plot of Fig. 3 that

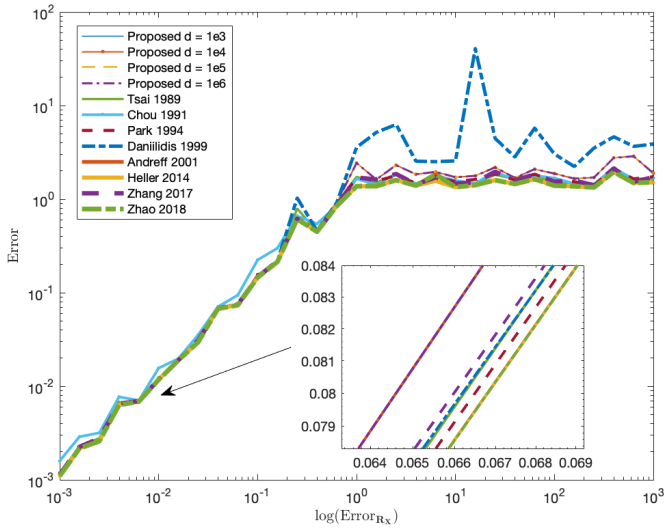


Fig. 3. Sensitivity of errors subject to input rotation noises.

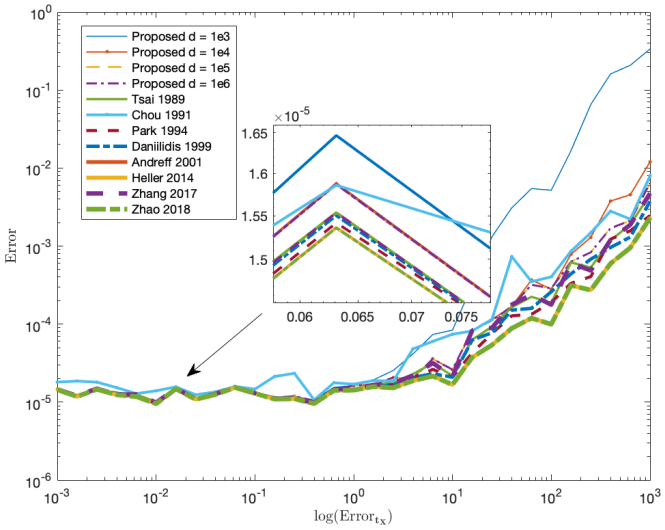


Fig. 4. Sensitivity of errors subject to input translation noises.

the optimization methods achieve the best accuracy, but the proposed method can also obtain the 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 a larger d value, 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 Section IV-A, 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.

C. Simulations on Uncertainty Descriptions

The uncertainty description of the hand-eye calibration problem $\boldsymbol{AX} = \boldsymbol{XB}$ is studied by Nguyen and Pham for the first time iteratively in [19]. It works very well with both synthetic and real-world data. However, it still has its drawbacks as follows.

1) The covariance of the rotation $\boldsymbol{R_X}$ is independently estimated from $\boldsymbol{R_A R_X} = \boldsymbol{R_X R_B}$, but, in fact, the accuracy of $\boldsymbol{R_X}$ is also affected by $\boldsymbol{t_A}$ and $\boldsymbol{t_B}$.
2) The covariances of $\boldsymbol{R_X}$ and $\boldsymbol{t_X}$ 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 $\boldsymbol{t_A}$ and $\boldsymbol{t_B}$ and, if possible, do not require iterations. The proposed SO(4) method in this paper, however, simultaneously estimates $\boldsymbol{R_X}$ and $\boldsymbol{t_X}$ together and can also generate the analytical probabilistic information within several deterministic steps considering the tightly coupled relationship inside $\boldsymbol{AX} = \boldsymbol{XB}$. Let us define $\boldsymbol{\xi}_{\boldsymbol{R_X},x}$, $\boldsymbol{\xi}_{\boldsymbol{R_X},y}$, and $\boldsymbol{\xi}_{\boldsymbol{R_X},z}$ as errors in rotation $\boldsymbol{R_X}$ around the x-, y-, and z-axes, while $\boldsymbol{\xi}_{\boldsymbol{t_X},x}$, $\boldsymbol{\xi}_{\boldsymbol{t_X},y}$, and $\boldsymbol{\xi}_{\boldsymbol{t_X},z}$ being errors in translation $\boldsymbol{t_X}$ about the x-, y-, and

$$
\begin{aligned}
&\delta \boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)} \delta \boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}^{T} \\
&\quad = \begin{pmatrix} \delta \boldsymbol{R_X} & \delta \boldsymbol{t_X}/d \\ -\delta \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{R_X}/d - \boldsymbol{t}_{\boldsymbol{X}}^{T} \delta \boldsymbol{R_X}/d & 0 \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} & -\boldsymbol{R}_{\boldsymbol{X}}^{T} \delta \boldsymbol{t_X}/d - \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} \boldsymbol{t_X}/d \\ \delta \boldsymbol{t}_{\boldsymbol{X}}^{T}/d & 0 \end{pmatrix} \\
&\quad = \begin{bmatrix} \delta \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} + \dfrac{1}{d^2} \delta \boldsymbol{t_X} \delta \boldsymbol{t}_{\boldsymbol{X}}^{T} & -\dfrac{1}{d} \left(\delta \boldsymbol{R_X} \boldsymbol{R}_{\boldsymbol{X}}^{T} \delta \boldsymbol{t_X} + \delta \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} \boldsymbol{t_X} \right) \\ -\dfrac{1}{d} \left(\delta \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} + \boldsymbol{t}_{\boldsymbol{X}}^{T} \delta \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} \right) & \dfrac{1}{d^2} \left(\boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{R_X} \boldsymbol{R}_{\boldsymbol{X}}^{T} \delta \boldsymbol{t_X} + \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} \boldsymbol{t_X} + \boldsymbol{t}_{\boldsymbol{X}}^{T} \delta \boldsymbol{R_X} \boldsymbol{R}_{\boldsymbol{X}}^{T} \delta \boldsymbol{t_X} + \boldsymbol{t}_{\boldsymbol{X}}^{T} \delta \boldsymbol{R_X} \delta \boldsymbol{R}_{\boldsymbol{X}}^{T} \boldsymbol{t_X} \right) \end{bmatrix} \\
\boldsymbol{\Sigma}_{\boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}} &= \left\langle \delta \boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)} \delta \boldsymbol{R}_{\boldsymbol{X},\mathrm{SO}(4)}^{T} \right\rangle \qquad (65) \\
&= \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} + \dfrac{1}{d^2} \boldsymbol{\Sigma}_{\boldsymbol{t_X}} & -\dfrac{1}{d} \left(\langle \delta \boldsymbol{t_X} \times \delta \boldsymbol{\theta_X} \rangle + \boldsymbol{\Sigma}_{\boldsymbol{R_X}} \boldsymbol{t_X} \right) \\ -\dfrac{1}{d} \left(\langle \delta \boldsymbol{t_X} \times \delta \boldsymbol{\theta_X} \rangle + \boldsymbol{\Sigma}_{\boldsymbol{R_X}} \boldsymbol{t_X} \right)^{T} & \dfrac{1}{d^2} \left(\boldsymbol{t}_{\boldsymbol{X}}^{T} \langle \delta \boldsymbol{t_X} \times \delta \boldsymbol{\theta_X} \rangle + \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} \boldsymbol{t_X} \right) \end{bmatrix} \\
&\approx \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} + \dfrac{1}{d^2} \boldsymbol{\Sigma}_{\boldsymbol{t_X}} & -\dfrac{1}{d} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} \boldsymbol{t_X} \\ -\dfrac{1}{d} \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} & \dfrac{1}{d^2} \boldsymbol{t}_{\boldsymbol{X}}^{T} \boldsymbol{\Sigma}_{\boldsymbol{R_X}} \boldsymbol{t_X} \end{bmatrix} \qquad (66)
\end{aligned}
$$

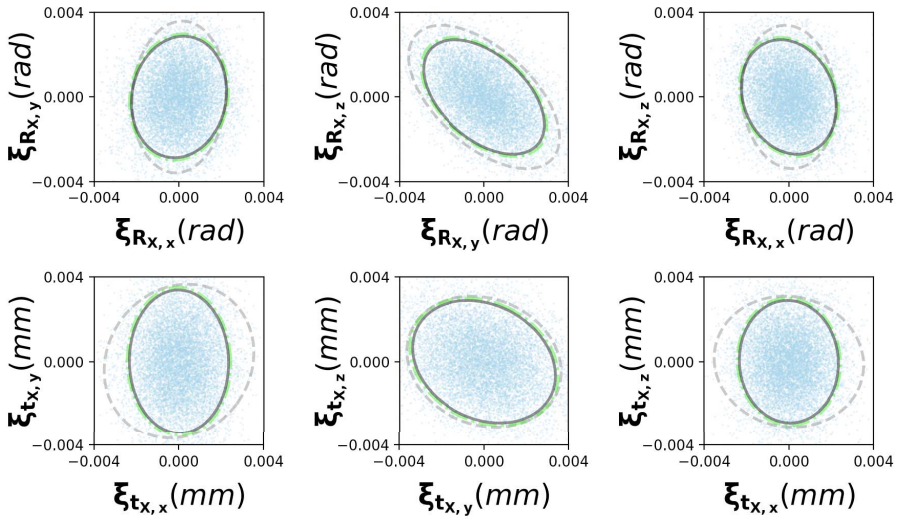


Fig. 5. 2-D covariance projections of the solutions to hand-eye calibration using the proposed method and that of Nguyen and Pham. The gray dashed lines indicate the mean bounds of the simulated statistics. The green dashed lines are from the solution of Nguyen and Pham, while the black solid ones are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

z-axes, respectively. Given the covariances $\boldsymbol{\Sigma}_{\boldsymbol{R}_X}$ and $\boldsymbol{\Sigma}_{\boldsymbol{t}_X}$, the covariance of the equivalent SO(4) transformation can be computed by (66) where we have

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

and $\delta \boldsymbol{R}_{X,\mathrm{SO}(4)} \delta \boldsymbol{R}_{X,\mathrm{SO}(4)}^T$ is simplified from (65) to (66), as shown at the bottom of the previous page, according to [51]

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

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

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{R}_{A_i}} &= \boldsymbol{\Sigma}_{\boldsymbol{R}_A}, \quad \boldsymbol{\Sigma}_{\boldsymbol{t}_{A_i}} = \boldsymbol{\Sigma}_{\boldsymbol{t}_A} \\ \boldsymbol{\Sigma}_{\boldsymbol{R}_{B_i}} &= \boldsymbol{\Sigma}_{\boldsymbol{R}_B}, \quad \boldsymbol{\Sigma}_{\boldsymbol{t}_{B_i}} = \boldsymbol{\Sigma}_{\boldsymbol{t}_B}. \end{aligned} \quad (69)$$

Now, we conduct the same simulation as that provided in the Python open-source codes of Nguyen and Pham [19] (https://github.com/dinhhuy2109/python-cope/examples/test_axxb_covariance.py). The input covariances are

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{R}_A} &= 10^{-10}\boldsymbol{I}, \quad \boldsymbol{\Sigma}_{\boldsymbol{t}_A} = 10^{-10}\boldsymbol{I} \\ \boldsymbol{\Sigma}_{\boldsymbol{R}_B} &= \begin{pmatrix} 4.15625 & -2.88693 & -0.60653 \\ -2.88693 & 32.0952 & -0.14482 \\ -0.60653 & -0.14482 & 1.43937 \end{pmatrix} \times 10^{-5} \\ \boldsymbol{\Sigma}_{\boldsymbol{t}_B} &= \begin{pmatrix} 19.52937 & 2.12627 & -1.06675 \\ 2.12627 & 4.44314426 & 0.38679 \\ -1.06675 & 0.38679 & 2.13070 \end{pmatrix} \times 10^{-5}. \end{aligned} \quad (70)$$

The simulation is carried out for 10 000 times, generating the randomly perturbed $\{\mathcal{A}\}$, $\{\mathcal{B}\}$, and in each set, there are 60 measurements. $\boldsymbol{\Sigma}_{\boldsymbol{b}}$ is computed according to the simulated statistics for $\{\mathcal{A}\}$, $\{\mathcal{B}\}$. The statistical covariance bounds of the estimated $\boldsymbol{R}_X$ and $\boldsymbol{t}_X$ values are then logged. Using the method by Nguyen and Pham and our proposed method, the 2-D covariance projections are shown in Fig. 5. One can see that both methods can estimate the covariance correctly, while our method achieves very slightly smaller covariances bounds. This reflects that our proposed method has reached the accuracy of Nguyen and Pham 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 and Pham. While the 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 a short computational period, which is beneficial to those applications with high demands on a real-time system with rigorous scheduling logics and timing.

D. Extension to the Extrinsic Calibration Between 3-D Laser Scanner and a Fisheye Camera

In this section, the developed 4DPA method in this paper is employed to solve the extrinsic calibration problem between a 3-D laser scanner and a fisheye camera, mounted rigidly on an experimental platform shown in Fig. 6. This platform contains a high-end 2-D 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 a resolution of 848 × 800 and a frame rate of 30 frames/s, along with an onboard factory-calibrated IMU. The spin mechanism and the feedback of an internal encoder of the servo guarantee the seamless stitching of successive laser scans that produce highly accurate 3-D scene reconstructions.

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**

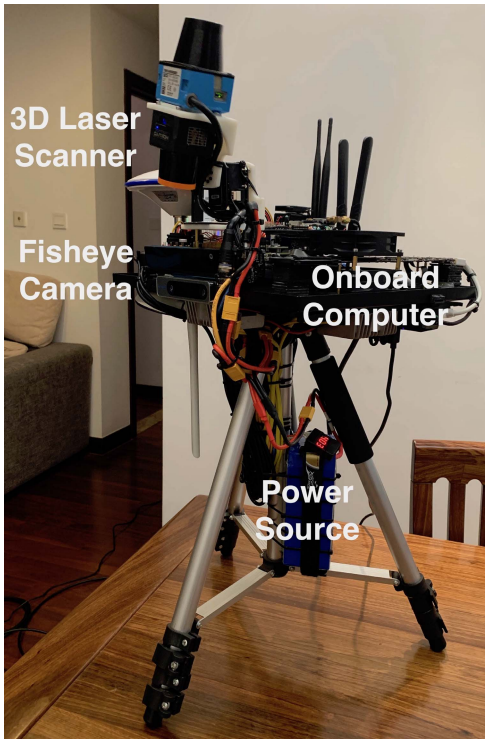


Fig. 6. Experimental platform equipped with a 3-D laser scanner and a fisheye camera, along with other processing devices.

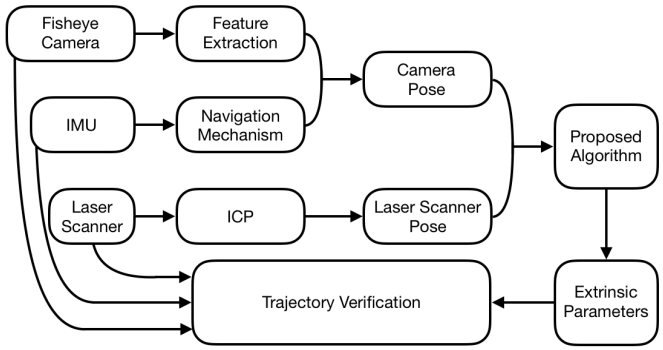


Fig. 7. Signal flowchart in the extrinsic calibration between the 3-D laser scanner and the fisheye camera using the proposed algorithm.

The single sensor or a combination of a laser scanner and a camera will be of great importance for scene measurement and reconstruction [52]–[54]. However, due to inevitable installation misalignments, the extrinsic calibration between the laser scanner and the camera should be performed for reliable measurement accuracy. Several algorithms have been proposed to deal with the calibration issues inside these sensors recently [55]–[57]. These methods in fact require some standard objects such as 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 in Fig. 7.

For the developed system, we can gather three sources of data, i.e., images from the fisheye camera, the inertial readings of angular rate and acceleration, and the 3-D laser scans. At the first stage, the fisheye camera and the 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 $\boldsymbol{T}_{\boldsymbol{A}_i}$ with an index of i, using the method in [59]. The pose of the 3-D laser scanner, denoted as $\boldsymbol{T}_{\boldsymbol{B}_i}$ with index i, is computed via the 3-D ICP [33], as shown in Fig. 8. As the camera and the laser scanner poses have the output frequencies of 200 and 1 Hz, respectively, the synchronization between them is conducted by continuous linear quaternion interpolation that we developed recently [43]. Then, using the properly synced $\boldsymbol{T}_{\boldsymbol{A}_i}$ and $\boldsymbol{T}_{\boldsymbol{B}_i}$, 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 the translation between the laser scanner and the fisheye camera, are calculated.

Then, these parameters are applied to the developed platform for 3-D trajectory verification using the Visual-lidar Odometry and Mapping (V-LOAM), method [60]. We put the system into measurement mode and then start moving it from origin to origin. Then, when computing the trajectories with the 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, a 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-, and 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 of d, as described in the previous sections. Therefore, a study on such an influence is conducted using the data for experiment 3 (see Table V).

We tune the d value from 1×10^3 to 1×10^7. The errors indicate that the chosen value $d = 1 \times 10^5$ in this 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 cannot deal with the calibration accurately. The reason is that

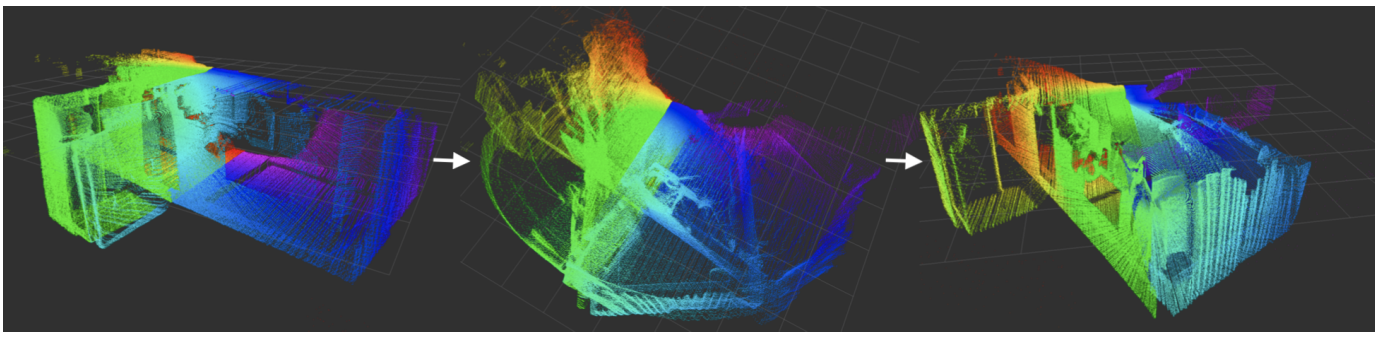


Fig. 8. Reconstructed scenes using the presented 3-D laser scanner. They are later used for the pose estimation of the laser scanner frame with ICP.

TABLE V
VERIFIED 3-D 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
$\mathbf{1 \times 10^5}$	**1.463**	**0.605**	**1.825**
1×10^6	1.462	0.600	1.799
1×10^7	1.462	0.599	1.798

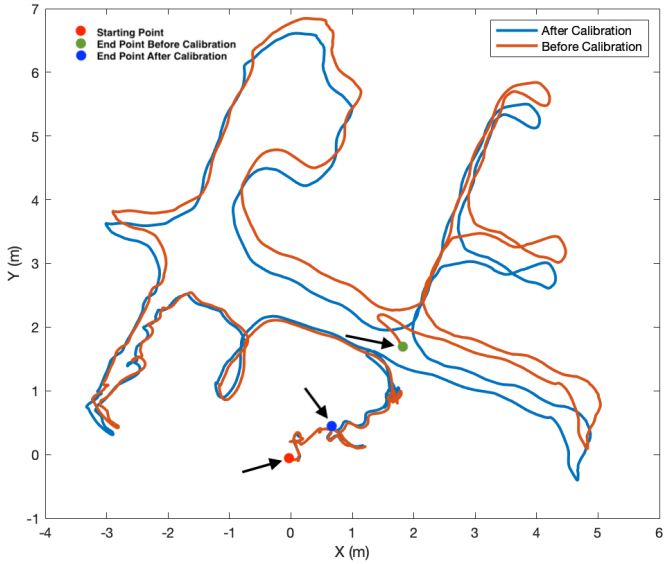


Fig. 9. Projected XY trajectories before and after the extrinsic calibration between the 3-D laser scanner and the fisheye camera.

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 $\boldsymbol{AX} = \boldsymbol{XB}$ by exploiting a new generalized method on SO(4). The investigated 4-D Procrustes analysis provides us with very useful closed-form results for hand-eye calibration. With such a 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 the real-world data sets. The proposed uncertainty descriptions for the 4×4 matrices are also universal to other similar problems, such as spacecraft attitude determination [29] and 3-D registration [32]. We also notice that the Procrustes analysis on SO(n) may be of benefit to solve the generalized hand-eye problem $\boldsymbol{AX} = \boldsymbol{XB}$ in which 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 $\boldsymbol{c}_1 = \boldsymbol{P}_1(\boldsymbol{\sigma})\boldsymbol{\sigma}$ as an example, one can explicitly write out

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

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

$$\boldsymbol{P}_1(\boldsymbol{\sigma}) = \frac{1}{\sqrt{2}}(\times)$$

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

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

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

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

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

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

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

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

$$\boldsymbol{J}_{11,i} = \begin{pmatrix} e_{11}-z_{11} & e_{12}+z_{21} & e_{13}+z_{31} & e_{14}+z_{41} \\ e_{12}+z_{21} & z_{11}-e_{11} & e_{14}-z_{41} & z_{31}-e_{13} \\ e_{13}+z_{31} & z_{41}-e_{14} & z_{11}-e_{11} & e_{12}-z_{21} \\ e_{14}+z_{41} & e_{13}-z_{31} & z_{21}-e_{12} & z_{11}-e_{11} \end{pmatrix}$$

$$\boldsymbol{J}_{12,i} = \begin{pmatrix} e_{21}-z_{21} & e_{22}-z_{11} & e_{23}+z_{41} & e_{24}-z_{31} \\ e_{22}-z_{11} & z_{21}-e_{21} & e_{24}+z_{31} & z_{41}-e_{23} \\ e_{23}-z_{41} & z_{31}-e_{24} & -e_{21}-z_{21} & e_{22}-z_{11} \\ e_{24}+z_{31} & e_{23}+z_{41} & z_{11}-e_{22} & -e_{21}-z_{21} \end{pmatrix}$$

$$\boldsymbol{J}_{13,i} = \begin{pmatrix} e_{31}-z_{31} & e_{32}-z_{41} & e_{33}-z_{11} & e_{34}+z_{21} \\ e_{32}+z_{41} & -e_{31}-z_{31} & e_{34}+z_{21} & z_{11}-e_{33} \\ e_{33}-z_{11} & z_{21}-e_{34} & z_{31}-e_{31} & e_{32}+z_{41} \\ e_{34}-z_{21} & e_{33}-z_{11} & z_{41}-e_{32} & -e_{31}-z_{31} \end{pmatrix}$$

$$\boldsymbol{J}_{14,i} = \begin{pmatrix} e_{41}-z_{41} & e_{42}+z_{31} & e_{43}-z_{21} & e_{44}-z_{11} \\ e_{42}-z_{31} & -e_{41}-z_{41} & e_{44}-z_{11} & z_{21}-e_{43} \\ e_{43}+z_{21} & z_{11}-e_{44} & -e_{41}-z_{41} & e_{42}+z_{31} \\ e_{44}-z_{11} & e_{43}+z_{21} & z_{31}-e_{42} & z_{41}-e_{41} \end{pmatrix}$$

$$\boldsymbol{J}_{21,i} = \begin{pmatrix} e_{12}-z_{12} & z_{22}-e_{11} & e_{14}+z_{32} & z_{42}-e_{13} \\ z_{22}-e_{11} & z_{12}-e_{12} & -e_{13}-z_{42} & z_{32}-e_{14} \\ z_{32}-e_{14} & z_{42}-e_{13} & e_{12}+z_{12} & e_{11}-z_{22} \\ e_{13}+z_{42} & -e_{14}-z_{32} & z_{22}-e_{11} & e_{12}+z_{12} \end{pmatrix}$$

$$\boldsymbol{J}_{22,i} = \begin{pmatrix} e_{22}-z_{22} & -e_{21}-z_{12} & e_{24}+z_{42} & -e_{23}-z_{32} \\ -e_{21}-z_{12} & z_{22}-e_{22} & z_{32}-e_{23} & z_{42}-e_{24} \\ -e_{24}-z_{42} & z_{32}-e_{23} & e_{22}-z_{22} & e_{21}-z_{12} \\ e_{23}+z_{32} & z_{42}-e_{24} & z_{12}-e_{21} & e_{22}-z_{22} \end{pmatrix}$$

$$\boldsymbol{J}_{23,i} = \begin{pmatrix} e_{32}-z_{32} & -e_{31}-z_{42} & e_{34}-z_{12} & z_{22}-e_{33} \\ z_{42}-e_{31} & -e_{32}-z_{32} & z_{22}-e_{33} & z_{12}-e_{34} \\ -e_{34}-z_{12} & z_{22}-e_{33} & e_{32}+z_{32} & e_{31}+z_{42} \\ e_{33}-z_{22} & -e_{34}-z_{12} & z_{42}-e_{31} & e_{32}-z_{32} \end{pmatrix}$$

$$\boldsymbol{J}_{24,i} = \begin{pmatrix} e_{42}-z_{42} & z_{32}-e_{41} & e_{44}-z_{22} & -e_{43}-z_{12} \\ -e_{41}-z_{32} & -e_{42}-z_{42} & -e_{43}-z_{12} & z_{22}-e_{44} \\ z_{22}-e_{44} & z_{12}-e_{43} & e_{42}-z_{42} & e_{41}+z_{32} \\ e_{43}-z_{12} & z_{22}-e_{44} & z_{32}-e_{41} & e_{42}+z_{42} \end{pmatrix}$$

$$\boldsymbol{J}_{31,i} = \begin{pmatrix} e_{13}-z_{13} & z_{23}-e_{14} & z_{33}-e_{11} & e_{12}+z_{43} \\ e_{14}+z_{23} & e_{13}+z_{13} & -e_{12}-z_{43} & z_{33}-e_{11} \\ z_{33}-e_{11} & z_{43}-e_{12} & z_{13}-e_{13} & -e_{14}-z_{23} \\ z_{43}-e_{12} & e_{11}-z_{33} & z_{23}-e_{14} & e_{13}+z_{13} \end{pmatrix}$$

$$\boldsymbol{J}_{32,i} = \begin{pmatrix} e_{23}-z_{23} & -e_{24}-z_{13} & z_{43}-e_{21} & e_{22}-z_{33} \\ e_{24}-z_{13} & e_{23}+z_{23} & z_{33}-e_{22} & z_{43}-e_{21} \\ -e_{21}-z_{43} & z_{33}-e_{22} & -e_{23}-z_{23} & -e_{24}-z_{13} \\ z_{33}-e_{22} & e_{21}+z_{43} & z_{13}-e_{24} & e_{23}-z_{23} \end{pmatrix}$$

$$\boldsymbol{J}_{33,i} = \begin{pmatrix} e_{33}-z_{33} & -e_{34}-z_{43} & -e_{31}-z_{13} & e_{32}+z_{23} \\ e_{34}+z_{43} & e_{33}-z_{33} & z_{23}-e_{32} & z_{13}-e_{31} \\ -e_{31}-z_{13} & z_{23}-e_{32} & z_{33}-e_{33} & z_{43}-e_{34} \\ -e_{32}-z_{23} & e_{31}-z_{13} & z_{43}-e_{34} & e_{33}-z_{33} \end{pmatrix}$$

$$\boldsymbol{J}_{34,i} = \begin{pmatrix} e_{43}-z_{43} & z_{33}-e_{44} & -e_{41}-z_{23} & e_{42}-z_{13} \\ e_{44}-z_{33} & e_{43}-z_{43} & -e_{42}-z_{13} & z_{23}-e_{41} \\ z_{23}-e_{41} & z_{13}-e_{42} & -e_{43}-z_{43} & z_{33}-e_{44} \\ -e_{42}-z_{13} & e_{41}+z_{23} & z_{33}-e_{44} & e_{43}+z_{43} \end{pmatrix}$$

$$\boldsymbol{J}_{41,i} = \begin{pmatrix} e_{14}-z_{14} & e_{13}+z_{24} & z_{34}-e_{12} & z_{44}-e_{11} \\ z_{24}-e_{13} & e_{14}+z_{14} & e_{11}-z_{44} & z_{34}-e_{12} \\ e_{12}+z_{34} & z_{44}-e_{11} & e_{14}+z_{14} & -e_{13}-z_{24} \\ z_{44}-e_{11} & -e_{12}-z_{34} & z_{24}-e_{13} & z_{14}-e_{14} \end{pmatrix}$$

$$\boldsymbol{J}_{42,i} = \begin{pmatrix} e_{24}-z_{24} & e_{23}-z_{14} & z_{44}-e_{22} & -e_{21}-z_{34} \\ -e_{23}-z_{14} & e_{24}+z_{24} & e_{21}+z_{34} & z_{44}-e_{22} \\ e_{22}-z_{44} & z_{34}-e_{21} & e_{24}-z_{24} & -e_{23}-z_{14} \\ z_{34}-e_{21} & z_{44}-e_{22} & z_{14}-e_{23} & -e_{24}-z_{24} \end{pmatrix}$$

$$\boldsymbol{J}_{43,i} = \begin{pmatrix} e_{34}-z_{34} & e_{33}-z_{44} & -e_{32}-z_{14} & z_{24}-e_{31} \\ z_{44}-e_{33} & e_{34}-z_{34} & e_{31}+z_{24} & z_{14}-e_{32} \\ e_{32}-z_{14} & z_{24}-e_{31} & e_{34}+z_{34} & z_{44}-e_{33} \\ -e_{31}-z_{24} & -e_{32}-z_{14} & z_{44}-e_{33} & -e_{34}-z_{34} \end{pmatrix}$$

$$\boldsymbol{J}_{44,i} = \begin{pmatrix} e_{44}-z_{44} & e_{43}+z_{34} & -e_{42}-z_{24} & -e_{41}-z_{14} \\ -e_{43}-z_{34} & e_{44}-z_{44} & e_{41}-z_{14} & z_{24}-e_{42} \\ e_{42}+z_{24} & z_{14}-e_{41} & e_{44}-z_{44} & z_{34}-e_{43} \\ -e_{41}-z_{14} & z_{24}-e_{42} & z_{34}-e_{43} & z_{44}-e_{44} \end{pmatrix}$$

where $e_{jk}, z_{jk}, j, k = 1, 2, 3, 4$ are the matrix entries of $\boldsymbol{E}_i$ and $\boldsymbol{Z}_i$, 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

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

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

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

Inserting the above-mentioned result into

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

gives (26).

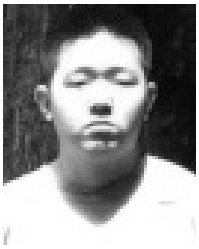
ACKNOWLEDGMENT

The authors would like to thank Dr. Z. Zhang from the University of Leeds for his detailed explanation of the implemented codes in [17]. They would like to thank Dr. H. Nguyen from Nanyang Technological University, Singapore, for the discussion with him on his useful codes of the uncertainty descriptions of hand-eye calibration [19]. They would also like to thank Dr. D. Modenini from the University of Bologna and Prof. D. Condurache from the 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, Apr. 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*, vol. 142, pp. 1–9, Aug. 2019. doi: 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, Jun. 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, Jun. 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, Jun. 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, Feb. 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, Oct. 1994.
[8] R. Horaud and F. Dornaika, "Hand-eye Calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, Jun. 1995.
[9] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 240–254, Jun. 1991.
[10] Y.-C. Lu and J. Chou, "Eight-space quaternion approach for robotic hand-eye calibration," in *Proc. IEEE ICSMC*, vol. 4, Oct. 2002, pp. 3316–3321.
[11] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, Mar. 1999.
[12] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *Int. J. Robot. Res.*, 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. Mach. Theory*, vol. 104, pp. 382–404, Oct. 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, Feb. 2003.
[15] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *Proc. IEEE ICRA*, May 2014, pp. 3157–3164.
[16] Z. Zhao, "Simultaneous robot-world and hand-eye calibration by the alternative linear programming," *Pattern Recogn. Lett.*, to be published. doi: 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, Oct. 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, Dec. 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, Jan. 2014.
[21] A. Alamri, M. Eid, R. Iglesias, S. Shirmohammadi, and A. E. Saddik, "Haptic virtual rehabilitation exercises for poststroke diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 9, pp. 1876–1884, Sep. 2007.
[22] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler principal component analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 498–518, 2013.
[23] A. Bartoli, D. Pizarro, and M. Loog, "Stratified generalized Procrustes analysis," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 227–253, Jan. 2013.
[24] L. Igual, X. Perez-Sala, S. Escalera, C. Angulo, and F. De La Torre, "Continuous generalized procrustes analysis," *Pattern Recognit.*, vol. 47, no. 2, pp. 659–671, Feb. 2014.
[25] C. I. Mosier, "Determining a simple structure when loadings for certain tests are known," *Psychometrika*, vol. 4, no. 2, pp. 149–162, Jun. 1939.
[26] B. F. Green, "The orthogonal approximation of an oblique structure in factor analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, Dec. 1952.
[27] M. W. Browne, "On oblique procrustes rotation," *Psychometrika*, vol. 32, no. 2, pp. 125–132, Jun. 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," *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, Mar. 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, May 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, Sep. 1987.
[33] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 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*, vol. 64, no. 8, pp. 3399–3406, Aug. 2019. doi: 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, Jan. 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*, to be published. doi: 10.1109/TAC.2018.2885241.
[37] F. Thomas, "Approaching dual quaternions from matrix algebra," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1037–1048, Oct. 2014.
[38] W. S. Massey, "Cross products of vectors in higher dimensional euclidean spaces," *Amer. 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. Automat. Sci. Eng.*, vol. 15, no. 1, pp. 307–319, Jan. 2018.
[40] J. Wu, M. Liu, Z. Zhou, and R. Li, "Fast symbolic 3D registration solution," 2019, *arxiv:1805.08703*.[Online]. Available: https://arxiv.org/abs/1805.08703
[41] I. Y. Bar-Itzhack, "New method for extracting the quaternion from a rotation matrix," *J. Guid., Control, Dyn.*, vol. 23, no. 6, pp. 1085–1087, 2000.
[42] A. H. J. de Ruiter and J. R. Forbes, "On the solution of Wahba's problem on SO(n)," *J. Astron. Sci.*, vol. 60, no. 1, pp. 1–31, Mar. 2013.
[43] J. Wu, "Optimal continuous unit quaternions from rotation matrices," *AIAA J. Guid. Control Dyn.*, vol. 42, no. 4, pp. 919–922, 2018.
[44] J. Wu, M. Liu, and Y. Qi, "Computationally efficient robust algorithm for generalized sensor calibration problem AR=RB," *IEEE Sensors J.*, to be published. doi: 10.1109/JSEN.2019.2924668.
[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, Jan. 2017.

[46] R. L. Dailey, "Eigenvector derivatives with repeated eigenvalues," *AIAA J.*, vol. 27, no. 4, pp. 486–491, Apr. 1989.
[47] G. Chang, T. Xu, and Q. Wang, "Error analysis of Davenport's q-method," *Automatica*, vol. 75, pp. 217–220, Jan. 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, Aug. 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, Mar. 2018.
[50] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, 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, Jun. 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, Oct. 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, Nov. 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, Feb. 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, Feb. 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, Apr. 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.*, to be published. doi: 10.1109/TIM.2018.2879705.
[58] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 110–132, Jan. 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, Aug. 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, Jan. 2017.



Jin Wu (S'15–M'17) was born in Zhenjiang, China, in May 1994. He received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China.

He has been a Research Assistant with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, since 2018. He has coauthored over 30 technical papers in representative journals and conference proceedings of the IEEE, AIAA, and IET. His current research interests include robot navigation, multi-sensor fusion, automatic control, and mechatronics.

Mr. Wu received the Outstanding Reviewer Award for the *Asian Journal of Control*. One of his papers published in the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING was selected as the ESI Highly Cited Paper by the ISI Web of Science from 2017 to 2018.

Yuxiang 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.

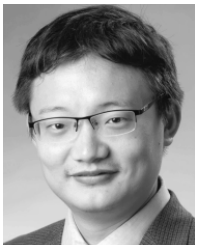
He is currently a Research Associate with the Department of Electronic and Computer Engineering, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include mobile robots, autonomous vehicles, deep learning, simultaneous localization and mapping (SLAM) and navigation, and motion detection.

Dr. Sun was a recipient of the Best Student Paper Finalist Award at the IEEE International Conference on Robotics and Biomimetics (ROBIO) 2015.

Miaomiao Wang (S'15) received the B.Sc. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2013, and the M.Sc. degree in control engineering from Lakehead University, Thunder Bay, ON, Canada, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Western University, London, ON, Canada.

He is also a Research Assistant with the Department of Electrical and Computer Engineering, Western University. He has been an author of technical papers in flagship journals and conference proceedings, including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and the IEEE Conference on Decision and Control (CDC). His current research interests include multiagent systems and geometric estimation and control.

Mr. Wang received the prestigious Ontario Graduate Scholarship (OGS) at Western University in 2018.



Ming Liu (S'12–M'13–SM'18) 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, under the supervision of Prof. R. Siegwart.

During his master's study at Tongji University, he stayed one year at Erlangen-Nürnberg University, Erlangen, Germany, and the Fraunhofer Institute IISB, Erlangen, 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 the Shanghai Swing Automation Co., Ltd., Shanghai. He is coordinating and involved in NSF Projects and National 863-Hi-TechPlan Projects in China. He has published many popular papers in top robotics journals, including the IEEE TRANSACTIONS ON ROBOTICS, the *International Journal of Robotics Research*, and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. His current 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), two awards from the International Aerial Robot Competition (IARC'14) as a Team Member, the Best Student Paper Award as a first author of the IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2012), the Best Paper Award in Information of the IEEE International Conference on Information and Automation (ICIA 2013) as a first author, the Best Paper Award Finalist as a coauthor, the Best RoboCup Paper Award of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), the Best Conference Paper Award of the IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent System (CYBER) 2015, the Best Student Paper Finalist of the IEEE International Conference on Real-time Computing and Robotics (RCAR 2015), the Best Student Paper Finalist of ROBIO 2015, the Best Student Paper Award of the IEEE-ICAR 2017, the Best Paper in Automation Award of the 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 the IEEE RCAR 2016 and the Program Chair of the International Robotics Conference in Foshan 2017. He was the Conference Chair of the International Conference on Computer Vision System (ICVS) 2017. He is currently an Associate Editor of the IEEE ROBOTICS AND AUTOMATION LETTERS.