Computationally Efficient Robust Algorithm for Generalized Sensor Calibration Problem $\boldsymbol{AR = RB}$

Jin Wu, *Member, IEEE,* Ming Liu, *Senior Member, IEEE* and Yuhua Qi

***Abstract*—This paper involves solving the generalized sensor calibration problem $\boldsymbol{AR = RB}$ with rotations $\boldsymbol{A, B, R} \in SO(3)$ where $\boldsymbol{A, B}$ are known and $\boldsymbol{R}$ is to be figured out. We introduce a methodology called the quaternion decomposition from rotation (QDR) to obtain a robust solution to this problem. The method is applicable to all those cases where $\boldsymbol{A, B}$ are noise-free, noisy or even not rigid. The non-iterative framework of the eigen-decomposition of 4×4 matrices is derived to give very computationally efficient analytical quaternion result. By numerical examples and experimental robotic results, the effectiveness of the proposed method has been verified. The proposed solution is evaluated to own at least the same accuracy and robustness of currently best algorithm, using real-world experiments for the camera/magnetometer sensor calibration on a quadrotor. While it takes much faster computation speed than all existing representatives to our knowledge.**

***Index Terms*—Sensor Calibration, Eigenvalue Problem, Robust Solver, Quaternion Estimation, Non-iterative Algorithm**

I. Introduction

MULTIPLE sensor calibration is an emerging task in instrumentation and measurement for accurate intelligent sensing [1], [2], [3]. In particular, for robotics applications, there are numerous needs for calibration between low-cost sensors including cameras, inertial measurement units (IMUs), magnetometers, optical distance sensors and etc. [4], [5]. Rotation $\boldsymbol{R}$ on the special orthogonal group $SO(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$, as two main factors of homogeneous pose transformation $\boldsymbol{X}$ in the special Euclidean space $SE(3)$, are significant aspects considered in current calibration methods. The pose estimation methods are very helpful for robotic perception [6], [7] and the problem $\boldsymbol{AX = XB}$ is also usually referred to as the generalized hand-eye calibration in which $\boldsymbol{A, B} \in SE(3)$ are known transformations and $\boldsymbol{X} \in SE(3)$ is unknown [8], [9], [10]. This paper solves an important sub-problem of the hand-eye calibration i.e. $\boldsymbol{AR = RB}$ where $\boldsymbol{A, B} \in SO(3)$ are known and $\boldsymbol{R} \in SO(3)$ is to be figured out. Let us first introduce the basic principle of the hand-eye calibration. In the tasks of visual servoing, practitioners always need to know the exact transformation between the gripper and attached camera so that precision localization and control can be implemented [11], [12]. Suppose that using machinery theory we can obtain the transformations between the robotic-arm joints. Then the transformation from the robotic base along to the gripper (see Fig. 1) can be determined by $\boldsymbol{T}_{\boldsymbol{B}_i} = \boldsymbol{T}_{\boldsymbol{B}_{i,3}} \boldsymbol{T}_{\boldsymbol{B}_{i,2}} \boldsymbol{T}_{\boldsymbol{B}_{i,1}}$. Meanwhile, using existing results of the computer vision like the Perspective-n-Points (PnP) [13], [14], we can also obtain the transformation $\boldsymbol{T}_{\boldsymbol{A}_i}$ between the camera and standard objects (like a chessboard or a pre-known 3D model) in the world frame.

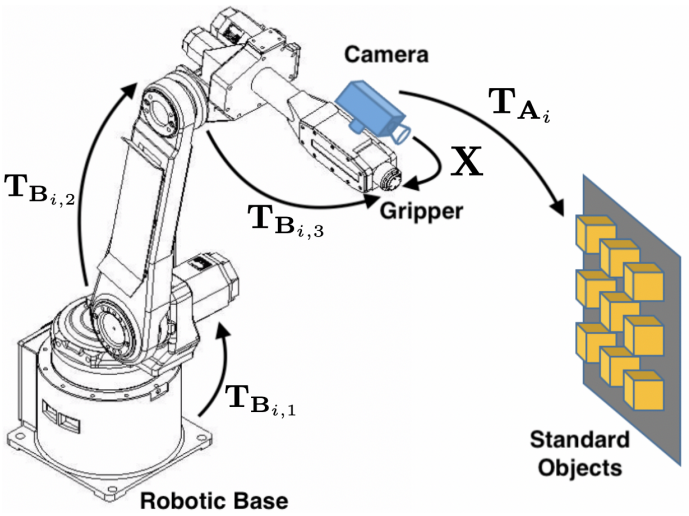


Fig. 1. Various transformations and their relationships.

Then with successive measurements of $\boldsymbol{T}_{\boldsymbol{A}_i}$ and $\boldsymbol{T}_{\boldsymbol{B}_i}$, we are able to construct the following pairs [9]

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

so that the $\boldsymbol{AX = XB}$ equation is established. If we expand the $\boldsymbol{AX = XB}$ with $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}$, then we have

$$\begin{cases} \boldsymbol{R_A R} = \boldsymbol{R R_B} \\ \boldsymbol{R_A t} + \boldsymbol{t_A} = \boldsymbol{R t_B} + \boldsymbol{t} \end{cases} \tag{2}$$

where $\boldsymbol{R_A}$ and $\boldsymbol{t_A}$ constitute $\boldsymbol{A}$, $\boldsymbol{R_B}$ and $\boldsymbol{t_B}$ form the transformation $\boldsymbol{B}$. We can see that one sub-problem of the

Manuscript received X X, 201X; revised X X, 201X; accepted X X, 201X. Date of publication X X, 201X; date of current version X X, 201X. This research was supported by Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ20160401100022706, in part by General Research Fund of Research Grants Council Hong Kong, 11210017, in part by Early Career Scheme Project of Research Grants Council Hong Kong, 21202816. The associate editor coordinating the review of this paper and approving it for publication was XXX. (*Corresponding author: Ming Liu*).

J. Wu and M. Liu are with Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. e-mail: jin_wu_uestc@hotmail.com; eelium@ust.hk.

Y. Qi is with School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China. e-mail: fatmoonqyp@126.com.

Digital Object Identifier XXX

generalized hand-eye calibration $\boldsymbol{AX}=\boldsymbol{XB}$ takes the form of $\boldsymbol{AR}=\boldsymbol{RB}$ with $\boldsymbol{A},\boldsymbol{B},\boldsymbol{R}\in SO(3)$ in which $\boldsymbol{R}$ is the unknown rotation matrix.

Efficient solutions to $\boldsymbol{AX}=\boldsymbol{XB}$ have been extensively studied where the sub-problem $\boldsymbol{AR}=\boldsymbol{RB}$ is addressed as the most important process of solving $\boldsymbol{AX}=\boldsymbol{XB}$ [15]. For example, Shiu et al. use the rotation logarithm to minimize an analytical error representation [16]. To enhance the robustness of the method by Shiu et al., Park et al. study the least-square registration method using rotation logarithm [17]. Using quaternions, Chou et al. also solve the problem $\boldsymbol{AR}=\boldsymbol{RB}$ [18]. The above algorithms also have their own disadvantages. Some of these methods can not always give orthonormal rotation results and may require further orthonormalization using singular value decomposition (SVD). Besides, many of them suffer from weak robustness facing noisy measurements since the assumption that $\boldsymbol{A},\boldsymbol{B}$ strictly belong to $SO(3)$ may be too ideal to adapt real measurement procedures.

Moreover, the problem $\boldsymbol{AR}=\boldsymbol{RB}$ is important and not limited only to the robotic-arm based hand-eye calibration. For instance, recently, a new camera/inertial/magnetic calibration scheme is designed by Zhang [19]. It also converts the calibration problem into solving $\boldsymbol{AR}=\boldsymbol{RB}$. With iterative algorithm comprising SVD, it has been verified to be accurate and much more computationally efficient than previous non-convex optimization methods. Based on this contribution, the $\boldsymbol{AX}=\boldsymbol{XB}$ problem is also solved in a brand-new manner [10]. In a new study of the motion capture system [20], the problem $\boldsymbol{AR}=\boldsymbol{RB}$ is also proposed to iteratively solve the internal sensor calibration issues. It is also introduced for obtaining the extrinsic parameters of RGB-D cameras, omnidirectional far infrared camera system [21], [22] and other optical systems [23], [24], [25]. A recent research by Modenini also shows the feasibility of $\boldsymbol{AR}=\boldsymbol{RB}$ in aerospace attitude determination [26]. As $\boldsymbol{AR}=\boldsymbol{RB}$ is important, till now, there are still many other problems highly related to its solution e.g. more generalized hand-eye calibration problems $\boldsymbol{AX}=\boldsymbol{YB}$ [27], [28] and $\boldsymbol{AXB}=\boldsymbol{YCZ}$ [29], [30] where $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}\in SE(3)$. Therefore we can see that the studied problem has quite wide applications and deserve detailed investigation.

To determine the quality of a calibration method, one often focuses on the accuracy and computation time. However, in real-time processing of recent calibration procedures [10], [20], one can hardly efficiently implement the algorithms due to: 1) They requires storage of large stacked historical matrices; 2) They rely on matrix manipulations including pseudo inverse and SVD which are time-costly. The current shortcomings of existing algorithms on robustness and computational load provide the ground for the challenges of the studied problem. The homogeneous form of $\boldsymbol{AR}=\boldsymbol{RB}$ gives us a motivation to reformulate it into an eigenvalue problem which is very similar to our previous solver of the Wahba's problem for spacecraft attitude determination [31]. As such, this paper proposes a new rotation factorization technique with which the $\boldsymbol{AR}=\boldsymbol{RB}$ problem is properly solved with at least the same accuracy and robustness compared with representatives. It is mentioned that all the developed steps are analytical, iteration-free and easy-to-implement.

The remainder of this paper is organized as follows: Section II contains our proposed theory for efficient solution to $\boldsymbol{AR}=\boldsymbol{RB}$ problem. Section III consists of theoretical comparisons showing why the proposed method owns computational and storage superiority in real-time embedded processing. Simulation and experiment analysis are shown in Section IV while concluding remarks are drawn in Section V.

II. Proposed Solution

A. Problem Formulation

Given two rotation measurements $\boldsymbol{A}_i,\boldsymbol{B}_i\in SO(3)$ in sequences $\{\mathcal{A}\},\{\mathcal{B}\}$ respectively with index of $i=1,2,\cdots,N$, the $\boldsymbol{AR}=\boldsymbol{RB}$ problem can be characterized with the following optimization of error metric

$$\mathcal{L}=\underset{\boldsymbol{R}\in SO(3)}{\arg\min}\sum_{i=1}^{N}\|\boldsymbol{A}_i\boldsymbol{R}-\boldsymbol{R}\boldsymbol{B}_i\|^2 \tag{3}$$

where $\|\boldsymbol{R}\|=\sqrt{tr\left(\boldsymbol{R}^T\boldsymbol{R}\right)}$. In this paper, the constraint is a bit loose i.e. $\boldsymbol{A}_i,\boldsymbol{B}_i$ do not have to be rigorously in $SO(3)$ according to measurement errors and word-length limitations of number storage and transmission. It has also been reported recently that $\boldsymbol{A}_i,\boldsymbol{B}_i$ here may be symmetric in aerospace applications [26]. Therefore the studied problem (3) is actually an another type of the generalized Procrustes problem [32], [33].

B. Quaternion Decomposition from Rotation

Parameterizing attitude with quaternion is not only simple but is free of gimbal lock of Euler angles mathematically [34]. Let us present the quaternion decomposition from rotation (QDR). An arbitrary rotation matrix $\boldsymbol{R}$ can be decomposed with unit quaternion $\boldsymbol{q}=(q_0,q_1,q_2,q_3)^T$ in the columns by $\boldsymbol{R}=(\boldsymbol{C}_1,\boldsymbol{C}_2,\boldsymbol{C}_3)$ where [31]

$$\boldsymbol{C}_1=\begin{pmatrix}q_0 & q_1 & -q_2 & -q_3\\ -q_3 & q_2 & q_1 & -q_0\\ q_2 & q_3 & q_0 & q_1\end{pmatrix}\boldsymbol{q}=\boldsymbol{P}_1\boldsymbol{q} \tag{4a}$$

$$\boldsymbol{C}_2=\begin{pmatrix}q_3 & q_2 & q_1 & q_0\\ q_0 & -q_1 & q_2 & -q_3\\ -q_1 & -q_0 & q_3 & q_2\end{pmatrix}\boldsymbol{q}=\boldsymbol{P}_2\boldsymbol{q} \tag{4b}$$

$$\boldsymbol{C}_3=\begin{pmatrix}-q_2 & q_3 & -q_0 & q_1\\ q_1 & q_0 & q_3 & q_2\\ q_0 & -q_1 & -q_2 & q_3\end{pmatrix}\boldsymbol{q}=\boldsymbol{P}_3\boldsymbol{q} \tag{4c}$$

$\boldsymbol{R}$ can also be decomposed in rows by $\boldsymbol{R}=\begin{pmatrix}\boldsymbol{q}^T\boldsymbol{Q}_1\\ \boldsymbol{q}^T\boldsymbol{Q}_2\\ \boldsymbol{q}^T\boldsymbol{Q}_3\end{pmatrix}$ where

$$\begin{aligned}\boldsymbol{Q}_1^T&=\begin{pmatrix}q_0 & q_1 & -q_2 & -q_3\\ q_3 & q_2 & q_1 & q_0\\ -q_2 & q_3 & -q_0 & q_1\end{pmatrix}\\ \boldsymbol{Q}_2^T&=\begin{pmatrix}-q_3 & q_2 & q_1 & -q_0\\ q_0 & -q_1 & q_2 & -q_3\\ q_1 & q_0 & q_3 & q_2\end{pmatrix}\\ \boldsymbol{Q}_3^T&=\begin{pmatrix}q_2 & q_3 & q_0 & q_1\\ -q_1 & -q_0 & q_3 & q_2\\ q_0 & -q_1 & -q_2 & q_3\end{pmatrix}\end{aligned} \tag{5}$$

The proposed QDR can be instantly obtained for those rotation matrices in $SO(3)$. For non-rigid matrices, extracting quaternion is equivalent to the orthonormalization of rotation. QDR in columns has already been introduced for solution of Wahba's problem [31]. While Wahba's solution is also invoked for matrix orthonormalization previously with same accuracy of SVD, the proposed QDR can actually be treated as an implicit method for orthonormalization, which significantly ensures its robustness in applications [35]. In following contents, we show how to use this useful QDR results to reformulate the error metric of the original problem.

C. Quaternion Solution

Defining $\boldsymbol{A}_i, \boldsymbol{B}_i$ as

$$\begin{aligned}\boldsymbol{A}_i &= (\boldsymbol{a}_{i,1}, \boldsymbol{a}_{i,2}, \boldsymbol{a}_{i,3})^T = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \\ \boldsymbol{B}_i &= (\boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}, \boldsymbol{b}_{i,3}) = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}\end{aligned} \tag{6}$$

respectively, we are able to write out

$$\begin{aligned}&\boldsymbol{A}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{B}_i \\ &= \begin{pmatrix} \boldsymbol{a}_{i,1}^T \\ \boldsymbol{a}_{i,2}^T \\ \boldsymbol{a}_{i,3}^T \end{pmatrix} (\boldsymbol{P}_1\boldsymbol{q}, \boldsymbol{P}_2\boldsymbol{q}, \boldsymbol{P}_3\boldsymbol{q}) - \begin{pmatrix} \boldsymbol{q}^T\boldsymbol{Q}_1 \\ \boldsymbol{q}^T\boldsymbol{Q}_2 \\ \boldsymbol{q}^T\boldsymbol{Q}_3 \end{pmatrix} (\boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}, \boldsymbol{b}_{i,3}) \\ &= \begin{bmatrix} \boldsymbol{a}_{i,1}^T\boldsymbol{P}_1\boldsymbol{q} & \boldsymbol{a}_{i,1}^T\boldsymbol{P}_2\boldsymbol{q} & \boldsymbol{a}_{i,1}^T\boldsymbol{P}_3\boldsymbol{q} \\ \boldsymbol{a}_{i,2}^T\boldsymbol{P}_1\boldsymbol{q} & \boldsymbol{a}_{i,2}^T\boldsymbol{P}_2\boldsymbol{q} & \boldsymbol{a}_{i,2}^T\boldsymbol{P}_3\boldsymbol{q} \\ \boldsymbol{a}_{i,3}^T\boldsymbol{P}_1\boldsymbol{q} & \boldsymbol{a}_{i,3}^T\boldsymbol{P}_2\boldsymbol{q} & \boldsymbol{a}_{i,3}^T\boldsymbol{P}_3\boldsymbol{q} \end{bmatrix} - \\ &\qquad \begin{bmatrix} \boldsymbol{q}^T\boldsymbol{Q}_1\boldsymbol{b}_{i,1} & \boldsymbol{q}^T\boldsymbol{Q}_1\boldsymbol{b}_{i,2} & \boldsymbol{q}^T\boldsymbol{Q}_1\boldsymbol{b}_{i,3} \\ \boldsymbol{q}^T\boldsymbol{Q}_2\boldsymbol{b}_{i,1} & \boldsymbol{q}^T\boldsymbol{Q}_2\boldsymbol{b}_{i,2} & \boldsymbol{q}^T\boldsymbol{Q}_2\boldsymbol{b}_{i,3} \\ \boldsymbol{q}^T\boldsymbol{Q}_3\boldsymbol{b}_{i,1} & \boldsymbol{q}^T\boldsymbol{Q}_3\boldsymbol{b}_{i,2} & \boldsymbol{q}^T\boldsymbol{Q}_3\boldsymbol{b}_{i,3} \end{bmatrix}\end{aligned} \tag{7}$$

Invoking the fact that dot product of two vectors is commutative, such that

$$\begin{aligned}\boldsymbol{q}^T\boldsymbol{Q}_1\boldsymbol{b}_{i,1} &= \boldsymbol{b}_{i,1}^T\boldsymbol{Q}_1^T\boldsymbol{q} \\ \boldsymbol{q}^T\boldsymbol{Q}_2\boldsymbol{b}_{i,1} &= \boldsymbol{b}_{i,1}^T\boldsymbol{Q}_2^T\boldsymbol{q} \\ \boldsymbol{q}^T\boldsymbol{Q}_3\boldsymbol{b}_{i,1} &= \boldsymbol{b}_{i,1}^T\boldsymbol{Q}_3^T\boldsymbol{q}\end{aligned} \tag{8}$$

we obtain $\boldsymbol{A}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{B}_i = (\boldsymbol{M}_{i,1}\boldsymbol{q}, \boldsymbol{M}_{i,2}\boldsymbol{q}, \boldsymbol{M}_{i,3}\boldsymbol{q})$ where

$$\begin{aligned}\boldsymbol{M}_{i,1} &= \begin{pmatrix} \boldsymbol{q}^T\boldsymbol{K}_{11,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{12,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{13,i} \end{pmatrix}, \boldsymbol{M}_{i,2} = \begin{pmatrix} \boldsymbol{q}^T\boldsymbol{K}_{21,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{22,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{23,i} \end{pmatrix}, \\ \boldsymbol{M}_{i,3} &= \begin{pmatrix} \boldsymbol{q}^T\boldsymbol{K}_{31,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{32,i} \\ \boldsymbol{q}^T\boldsymbol{K}_{33,i} \end{pmatrix}\end{aligned} \tag{9}$$

with parameter matrices given in (10)

$$\boldsymbol{K}_{11,i} = \begin{pmatrix} a_{11}-b_{11} & 0 & a_{13}+b_{31} & -a_{12}-b_{21} \\ 0 & a_{11}-b_{11} & a_{12}-b_{21} & a_{13}-b_{31} \\ a_{13}+b_{31} & a_{12}-b_{21} & b_{11}-a_{11} & 0 \\ -a_{12}-b_{21} & a_{13}-b_{31} & 0 & b_{11}-a_{11} \end{pmatrix}$$

$$\boldsymbol{K}_{12,i} = \begin{pmatrix} a_{21}-b_{21} & -b_{31} & a_{23} & b_{11}-a_{22} \\ -b_{31} & a_{21}+b_{21} & a_{22}-b_{11} & a_{23} \\ a_{23} & a_{22}-b_{11} & -a_{21}-b_{21} & -b_{31} \\ b_{11}-a_{22} & a_{23} & -b_{31} & b_{21}-a_{21} \end{pmatrix} \tag{10a}$$

$$\boldsymbol{K}_{13,i} = \begin{pmatrix} a_{31}-b_{31} & b_{21} & a_{33}-b_{11} & -a_{32} \\ b_{21} & a_{31}+b_{31} & a_{32} & a_{33}-b_{11} \\ a_{33}-b_{11} & a_{32} & b_{31}-a_{31} & -b_{21} \\ -a_{32} & a_{33}-b_{11} & -b_{21} & -a_{31}-b_{31} \end{pmatrix}$$

$$\boldsymbol{K}_{21,i} = \begin{pmatrix} a_{12}-b_{12} & -a_{13} & b_{32} & a_{11}-b_{22} \\ -a_{13} & -a_{12}-b_{12} & a_{11}-b_{22} & -b_{32} \\ b_{32} & a_{11}-b_{22} & a_{12}+b_{12} & a_{13} \\ a_{11}-b_{22} & -a_{32} & a_{13} & b_{12}-a_{12} \end{pmatrix} \tag{10b}$$

$$\boldsymbol{K}_{22,i} = \begin{pmatrix} a_{22}-b_{22} & -a_{23}-b_{32} & 0 & a_{21}+b_{12} \\ -a_{23}-b_{32} & b_{22}-a_{22} & a_{21}-b_{12} & 0 \\ 0 & a_{21}-b_{12} & a_{22}-b_{22} & a_{23}-b_{32} \\ a_{21}+b_{12} & 0 & a_{23}-b_{32} & b_{22}-a_{22} \end{pmatrix}$$

$$\boldsymbol{K}_{23,i} = \begin{pmatrix} a_{32}-b_{32} & b_{22}-a_{33} & -b_{12} & a_{31} \\ a_{22}-b_{33} & b_{32}-a_{32} & a_{31} & -b_{12} \\ -b_{12} & a_{31} & a_{32}+b_{32} & a_{33}-b_{22} \\ a_{31} & -b_{12} & a_{33}-b_{22} & -a_{32}-b_{32} \end{pmatrix} \tag{10c}$$

$$\boldsymbol{K}_{31,i} = \begin{pmatrix} a_{13}-b_{13} & b_{12} & b_{33}-a_{11} & b_{23} \\ a_{12} & -a_{13}-b_{13} & -b_{23} & a_{11}-b_{33} \\ b_{33}-a_{11} & -b_{23} & b_{13}-a_{13} & a_{12} \\ -b_{23} & a_{11}-b_{33} & a_{12} & a_{13}+b_{13} \end{pmatrix}$$

$$\boldsymbol{K}_{32,i} = \begin{pmatrix} a_{23}-b_{23} & a_{22}-b_{33} & -a_{21} & b_{13} \\ a_{22}-b_{33} & b_{23}-a_{23} & -b_{13} & a_{21} \\ -a_{21} & -b_{13} & -a_{23}-b_{23} & a_{22}-b_{33} \\ b_{13} & a_{21} & a_{22}-b_{33} & a_{23}+b_{23} \end{pmatrix}$$

$$\boldsymbol{K}_{33,i} = \begin{pmatrix} a_{33}-b_{33} & a_{32}+b_{23} & -a_{31}-b_{13} & 0 \\ a_{32}+b_{23} & b_{33}-a_{33} & 0 & a_{31}-b_{13} \\ -a_{31}-b_{13} & 0 & b_{33}-a_{33} & a_{32}-b_{23} \\ 0 & a_{31}-b_{13} & a_{32}-b_{23} & a_{33}-b_{33} \end{pmatrix} \tag{10d}$$

Let us define

$$\begin{aligned}\boldsymbol{G}_i &= \boldsymbol{A}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{B}_i \\ &= \begin{pmatrix} \boldsymbol{q}^T\boldsymbol{K}_{11,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{21,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{31,i}\boldsymbol{q} \\ \boldsymbol{q}^T\boldsymbol{K}_{12,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{22,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{32,i}\boldsymbol{q} \\ \boldsymbol{q}^T\boldsymbol{K}_{13,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{23,i}\boldsymbol{q} & \boldsymbol{q}^T\boldsymbol{K}_{33,i}\boldsymbol{q} \end{pmatrix}\end{aligned} \tag{11}$$

Then (3) is converted to

$$\mathcal{L} = \underset{\boldsymbol{q}^T\boldsymbol{q}=1}{\arg\min} \sum_{i=1}^{N} \|\boldsymbol{G}_i\|^2 \tag{12}$$

In particular, the sub-problem is treated by

$$\underset{\boldsymbol{q}^T\boldsymbol{q}=1}{\arg\min} \|\boldsymbol{G}_i\|^2 = \underset{\boldsymbol{q}^T\boldsymbol{q}=1}{\arg\min} \boldsymbol{q}^T \begin{pmatrix} \boldsymbol{K}_{11,i}^2 + \boldsymbol{K}_{12,i}^2 + \boldsymbol{K}_{13,i}^2 + \\ \boldsymbol{K}_{21,i}^2 + \boldsymbol{K}_{22,i}^2 + \boldsymbol{K}_{23,i}^2 + \\ \boldsymbol{K}_{31,i}^2 + \boldsymbol{K}_{32,i}^2 + \boldsymbol{K}_{33,i}^2 \end{pmatrix} \boldsymbol{q} \tag{13}$$

which indicates that using Lagrange multiplier, the quaternion is the eigenvector such that

$$\begin{aligned}\boldsymbol{K}_i &= \sum_{j=1}^{3}\sum_{k=1}^{3} \boldsymbol{K}_{jk,i}^2 \\ \boldsymbol{K}_i\boldsymbol{q} &= \lambda_{\boldsymbol{K}_i,\min}\boldsymbol{q}\end{aligned} \tag{14}$$

i.e. it corresponds to the minimum eigenvalue $\lambda_{\boldsymbol{K}_i,\min}$ of $\boldsymbol{K}_i$. Then it is obvious that the globally optimal solution to (13) equals to the eigenvector satisfying $\boldsymbol{K}\boldsymbol{q} = \lambda_{\boldsymbol{K},\min}\boldsymbol{q}$ with

$$\boldsymbol{K} = \sum_{i=1}^{N}\sum_{j=1}^{3}\sum_{k=1}^{3} \boldsymbol{K}_{jk,i}^2 \tag{15}$$

Seen from (10), we directly observe that $\boldsymbol{K}_{jk,i}$ is symmetric which leads to the symmetry-preserving of $\boldsymbol{K}_i$ and $\boldsymbol{K}$. As $\boldsymbol{K}$ is in the form of a real symmetric 4×4 matrix, all its eigenvalues are real and can be computed with the following characteristic polynomial

$$\det\left(\boldsymbol{K} - \lambda \boldsymbol{I}\right) = 0 \Rightarrow \lambda^4 + \tau_1\lambda^3 + \tau_2\lambda^2 + \tau_3\lambda + \tau_4 = 0 \tag{16}$$

with $\boldsymbol{I}$ denoting the identity matrix and this formulation can be transformed into

$$\lambda^4 + \upsilon_1\lambda^2 + \upsilon_2\lambda + \upsilon_3 = 0 \tag{17}$$

with [36]

$$\begin{aligned}
\upsilon_1 &= \tau_2 - \frac{3}{8}\tau_1^2 \\
\upsilon_2 &= \tau_3 - \frac{\tau_1\tau_2}{2} + \frac{\tau_1^3}{8} \\
\upsilon_3 &= \tau_4 - \frac{\tau_1\tau_3}{4} + \frac{\tau_1^2\tau_2}{16} - \frac{3\tau_1^4}{256}
\end{aligned} \tag{18}$$

Note that for an arbitrary real symmetric $\boldsymbol{K}$ of entries

$$\boldsymbol{K} = \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{12} & K_{22} & K_{23} & K_{24} \\ K_{13} & K_{23} & K_{33} & K_{34} \\ K_{14} & K_{24} & K_{34} & K_{44} \end{pmatrix} \tag{19}$$

, we are able to compute [31]

$$\begin{aligned}
\tau_1 &= -tr\left(\boldsymbol{K}\right) \\
\tau_2 &= -K_{12}^2 - K_{13}^2 - K_{14}^2 - K_{23}^2 - K_{24}^2 - K_{34}^2 + \\
&\quad K_{11}K_{22} + K_{11}K_{33} + K_{22}K_{33} + \\
&\quad K_{11}K_{44} + K_{22}K_{44} + K_{33}K_{44} \\
\tau_3 &= K_{33}K_{12}^2 + K_{44}K_{12}^2 - 2K_{13}K_{23}K_{12} - \\
&\quad 2K_{14}K_{24}K_{12} + K_{11}K_{23}^2 + K_{11}K_{24}^2 + \\
&\quad K_{11}K_{34}^2 + K_{22}K_{34}^2 + K_{13}^2K_{22} + \\
&\quad K_{14}^2K_{22} + K_{14}^2K_{33} + K_{24}^2K_{33} - \\
&\quad K_{11}K_{22}K_{33} - 2K_{13}K_{14}K_{34} - \\
&\quad 2K_{23}K_{24}K_{34} + K_{13}^2K_{44} + K_{23}^2K_{44} - \\
&\quad K_{11}K_{22}K_{44} - K_{11}K_{33}K_{44} - K_{22}K_{33}K_{44} \\
\tau_4 &= \det\left(\boldsymbol{K}\right)
\end{aligned} \tag{20}$$

Then the eigenvalues of $\boldsymbol{K}$ are calculated by our previous symbolic computation results [31]:

$$\begin{aligned}
\lambda_{\boldsymbol{K},1} &= \tfrac{1}{2\sqrt{6}}\left(T_2 - \sqrt{-T_2^2 - 12\upsilon_1 - \tfrac{12\sqrt{6}\upsilon_2}{T_2}}\right) \\
\lambda_{\boldsymbol{K},2} &= \tfrac{1}{2\sqrt{6}}\left(T_2 + \sqrt{-T_2^2 - 12\upsilon_1 - \tfrac{12\sqrt{6}\upsilon_2}{T_2}}\right) \\
\lambda_{\boldsymbol{K},3} &= -\tfrac{1}{2\sqrt{6}}\left(T_2 + \sqrt{-T_2^2 - 12\upsilon_1 + \tfrac{12\sqrt{6}\upsilon_2}{T_2}}\right) \\
\lambda_{\boldsymbol{K},4} &= -\tfrac{1}{2\sqrt{6}}\left(T_2 - \sqrt{-T_2^2 - 12\upsilon_1 + \tfrac{12\sqrt{6}\upsilon_2}{T_2}}\right)
\end{aligned} \tag{21}$$

in which

$$\begin{aligned}
T_0 &= 2\upsilon_1^3 + 27\upsilon_2^2 - 72\upsilon_1\upsilon_3 \\
T_1 &= \left(T_0 + \sqrt{-4\left(\upsilon_1^2 + 12\upsilon_3\right)^3 + T_0^2}\right)^{\frac{1}{3}} \\
T_2 &= \sqrt{-4\upsilon_1 + \frac{2^{\frac{4}{3}}\left(\upsilon_1^2 + 12\upsilon_3\right)}{T_1} + 2^{\frac{2}{3}}T_1}
\end{aligned} \tag{22}$$

By finding out the minimum eigenvalue $\lambda_{\boldsymbol{K},\min}$, the optimal quaternion is solved by taking the null space of $\boldsymbol{K} - \lambda_{\boldsymbol{K},\min}\boldsymbol{I}$. The eigen-decomposition is a robust algorithm because any matrix has eigenvalues and corresponding eigenvectors. It is also fast since no iterative steps are required. The computational procedures are shown in Algorithm 1:

Algorithm 1 The proposed algorithm for solving the generalized sensor calibration problem $\boldsymbol{AR} = \boldsymbol{RB}$.

Require: Input matrices $\{\boldsymbol{A}_i, \boldsymbol{B}_i | i = 1, 2, \cdots, N\}$
Step 1: Compute matrices according to $\boldsymbol{A}_i, \boldsymbol{B}_i$ by (10).
Step 2: Compute $\boldsymbol{K}$ using (15).
Step 3: Determine the coefficients by (20) and (18). Then get the eigenvalues of $\boldsymbol{K}$ with (21).
Step 4: Find the minimum eigenvalue of $\boldsymbol{K}$ and compute the corresponding eigenvector. Normalizing this eigenvector obtains the quaternion solution for the proposed method.

III. Theoretical Comparisons

In this section, representatives are compared for theoretical explanation of the fast computation speed and low memory storage of the proposed method. The method of [26] is a single-point one that can not deal with the generalized problem of (3) thus we do not compare it with the proposed method here. In conventional works of [16] and [17] for $\boldsymbol{AR} = \boldsymbol{RB}$ problem, the matrix logarithms $\log(\boldsymbol{A}), \log(\boldsymbol{B})$ are required for computation. In particular, in [17], the problem $\boldsymbol{AR} = \boldsymbol{RB}$ is solved by

$$\begin{aligned}
&\arg\min_{\boldsymbol{R}\in SO(3)} \sum_{i=1}^{N} \left\|\boldsymbol{A}_i\boldsymbol{R} - \boldsymbol{R}\boldsymbol{B}_i\right\|^2 \\
&= \arg\min_{\boldsymbol{R}\in SO(3)} \sum_{i=1}^{N} \left\|\boldsymbol{x}_i - \boldsymbol{R}\boldsymbol{y}_i\right\|^2
\end{aligned} \tag{23}$$

in which $\boldsymbol{x}_i = [\log(\boldsymbol{A}_i)]^\wedge, \boldsymbol{y}_i = [\log(\boldsymbol{B}_i)]^\wedge$ where, for one matrix $\boldsymbol{A}$ rigorously in $SO(3)$, the matrix logarithm can be computed analytically via [17]

$$\log(\boldsymbol{A}) = \frac{\phi}{2\sin\phi}\left(\boldsymbol{A} - \boldsymbol{A}^T\right) \tag{24}$$

with $1 + 2\cos\phi = tr(\boldsymbol{A})$. And $\wedge$ denotes the operation to extract the 3D vector from a 3×3 skew-symmetric matrix. However, when $\boldsymbol{A}_i, \boldsymbol{B}_i$ are not in $SO(3)$, the matrix logarithm computation process is much more computationally expensive and can not be achieved with (24). Moreover, such matrix logarithm is not robust since complex-number results of $\boldsymbol{x}_i, \boldsymbol{y}_i$ may take place. An alternative is to conduct the double-side orthonormalization of $\boldsymbol{A}_i, \boldsymbol{B}_i$ simultaneously using SVD [38], [39]. But the remaining problem is evident that such double-side SVD is neither optimal nor fast. The proposed method, however, deals with $\boldsymbol{A}_i, \boldsymbol{B}_i$ in a joint decomposition manner i.e. QDR. As QDR is analytical and simple, methods in [16] and [17] are fundamentally slower than the proposed method.

For other methods like dual-quaternion method [9] and Zhang's method [19], the SVD operation is heavily invoked in computation. While the 3×3 SVD has been long ago evaluated

as at least twice slower than the 4×4 eigen-decomposition [40], [41]. Therefore the proposed method will definitely be faster than SVD methods.

From the aspect of memory consumption, Zhang's method [19] requires a storage of all historical data of $\boldsymbol{A}_i, \boldsymbol{B}_i, i = 1, 2, \cdots, N$:

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_N \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \vdots \\ \boldsymbol{B}_N \end{pmatrix} \tag{25}$$

to compute their Moore-Penrose pseudo inverse $\boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger$. That is to say, the embedded computer needs to reserve large memory area for such dynamic storage. However, according to the datasheets of recently released representative micro-controllers e.g. STMicroelectronics STM32H7x3, Microchip SAME70/SAMV71 and etc. [42], [43], we may find out that typical area of the internal RAM is restricted within 1MB. While an external RAM chip will consume huge loads of controller pins, more circuit designing areas and higher prices. Therefore in applications, Zhang's method can hardly be implemented on popular embedded platforms according to its large storage requirement. The same problems also occurs in representatives e.g. [8] where large stacked historical data matrices are required. Recalling the computation procedures of the proposed method in Section II, it is obvious that the computation of $\boldsymbol{G}$ relies only on the $\boldsymbol{G}_i$ for each step. As $\boldsymbol{G}_i$ is additive over time, we do not have to save any historical data. This point well overcomes the shortcoming of microcontrollers with valuable RAM spaces. With fully analytical computation procedures shown in Section II, the microcontroller may handle the calibration process with high sensor sampling frequencies using the proposed method.

IV. Experimental Results

A. Accuracy and Robustness

Park et al. have proved that ideal condition for $\boldsymbol{AR} = \boldsymbol{RB}$ problem holds when $\log(\boldsymbol{A}) = \log(\boldsymbol{B})$ [17]. Therefore the noise sources are based on two aspects: 1) For those cases where $\log(\boldsymbol{A}) \neq \log(\boldsymbol{B})$; 2) For those cases where $\boldsymbol{A}, \boldsymbol{B}$ are arbitrary and not in $SO(3)$. In this sub-section, we first present accuracy comparisons on ideal cases. Then noisy cases are simulated to show the robustness performances of various algorithms. We pick up Zhang's iterative method [19] and other classical but popular methods from Shiu et al. 1989 [16], Park et al. 1994 [17], Daniilidis 1999 [9], Andreff et al. 2001 [8] and Schmidt et al. 2008 [37]. The nonorthogonality of the raw data is defined as

$$\epsilon = \sum_{i=1}^{N} \left\| \boldsymbol{A}_i \boldsymbol{A}_i^T - \boldsymbol{I} \right\|^2 + \left\| \boldsymbol{B}_i \boldsymbol{B}_i^T - \boldsymbol{I} \right\|^2 \tag{26}$$

while the rotation accuracy is characterized in the same manner by $\gamma = \sum_{i=1}^{N} \left\| \boldsymbol{R} \boldsymbol{R}_{true}^T - \boldsymbol{I} \right\|^2$ where $\boldsymbol{R}_{true}$ is the simulated true value of rotation. For Zhang's method, the terminating relative accuracy for norm of matrix difference is set to 1×10^{-15}. We simulate 9 cases with different configurations of $\boldsymbol{A}, \boldsymbol{B}$. The test cases are chosen based on the following criteria:

1) Cases $1 \sim 3$ contain exactly orthonormalized $\boldsymbol{A}, \boldsymbol{B}$ and the matrix logarithms holds such that $\log(\boldsymbol{A}) = \log(\boldsymbol{B})$.
2) For Cases $4 \sim 6$, the ideal orthonormality is not broken. However, $\boldsymbol{A}, \boldsymbol{B}$ are simulated with $\log(\boldsymbol{A}) \neq \log(\boldsymbol{B})$.
3) Last cases $7 \sim 9$ will not guarantee ideal orthonormality and logarithm equality between $\boldsymbol{A}, \boldsymbol{B}$ at all.

For each case, there are 2000 samples simulated. Each case is tested for 100 times for mean evaluation performance. The

TABLE I
Rotation Accuracy γ

Case	Nonorthogonality ϵ	$\log(\boldsymbol{A}) = \log(\boldsymbol{B})$?	Shiu 1989 [16]	Park 1994 [17]	Zhang 2016 [19]	**Proposed Method**
1	2.2276×10^{-31}	Yes	5.87263×10^{-16}	4.75239×10^{-15}	1.00278×10^{-15}	$\mathbf{1.60262 \times 10^{-15}}$
2	2.8058×10^{-31}	Yes	3.88729×10^{-15}	5.49049×10^{-16}	4.09010×10^{-16}	$\mathbf{1.35700 \times 10^{-15}}$
3	3.2996×10^{-31}	Yes	8.31880×10^{-16}	7.95811×10^{-16}	4.13413×10^{-16}	$\mathbf{2.24437 \times 10^{-15}}$
4	2.2276×10^{-31}	No	2.06726×10^{-02}	2.02990×10^{-02}	1.64420×10^{-02}	$\mathbf{8.66107 \times 10^{-03}}$
5	2.8058×10^{-31}	No	1.60767×10^{-01}	5.93726×10^{-02}	2.36142×10^{-02}	$\mathbf{1.03893 \times 10^{-02}}$
6	3.2996×10^{-31}	No	1.11383×10^{-01}	5.08396×10^{-02}	1.48497×10^{-02}	$\mathbf{9.02140 \times 10^{-03}}$
7	0.0012032	No	1.61657×10^{-02}	3.51532×10^{-02}	2.13694×10^{-03}	$\mathbf{8.39891 \times 10^{-04}}$
8	0.11949	No	2.62925×10^{-01}	2.02546×10^{-01}	1.41941×10^{-02}	$\mathbf{1.31290 \times 10^{-02}}$
9	0.48775	No	2.50554×10^{-01}	1.21573×10^{-01}	1.65432×10^{-02}	$\mathbf{1.40847 \times 10^{-02}}$

Case	Nonorthogonality ϵ	$\log(\boldsymbol{A}) = \log(\boldsymbol{B})$?	Daniilidis 1999 [9]	Andreff 2001 [8]	Schmidt 2008 [37]	**Proposed Method**
1	2.2276×10^{-31}	Yes	1.82311×10^{-15}	5.77959×10^{-16}	3.20836×10^{-15}	$\mathbf{1.60262 \times 10^{-15}}$
2	2.8058×10^{-31}	Yes	1.19643×10^{-15}	1.36752×10^{-15}	7.87045×10^{-16}	$\mathbf{1.35700 \times 10^{-15}}$
3	3.2996×10^{-31}	Yes	9.53364×10^{-16}	6.74939×10^{-16}	6.66064×10^{-16}	$\mathbf{2.24437 \times 10^{-15}}$
4	2.2276×10^{-31}	No	9.52718×10^{-03}	7.89654×10^{-02}	6.92886×10^{-03}	$\mathbf{8.66107 \times 10^{-03}}$
5	2.8058×10^{-31}	No	1.14283×10^{-02}	1.43016×10^{-01}	8.31146×10^{-03}	$\mathbf{1.03893 \times 10^{-02}}$
6	3.2996×10^{-31}	No	9.92354×10^{-03}	2.75096×10^{-01}	7.21712×10^{-03}	$\mathbf{9.02140 \times 10^{-03}}$
7	0.0012032	No	9.23880×10^{-04}	3.26839×10^{-03}	6.71912×10^{-04}	$\mathbf{8.39891 \times 10^{-04}}$
8	0.11949	No	1.66419×10^{-02}	1.89236×10^{-02}	1.21032×10^{-02}	$\mathbf{1.31290 \times 10^{-02}}$
9	0.48775	No	1.54932×10^{-02}	9.30965×10^{-01}	1.12678×10^{-02}	$\mathbf{1.40847 \times 10^{-02}}$

TABLE II
COMPUTATION TIME

Case	Nonorthogonality ϵ	$\log(\boldsymbol{A}) = \log(\boldsymbol{B})$?	Shiu 1989 [16]	Park 1994 [17]	Zhang 2016 [19]	**Proposed Method**
1	2.2276×10^{-31}	Yes	3.15998×10^{-01} sec	3.71002×10^{-01} sec	2.15746×10^{-01} sec	$\mathbf{3.93071 \times 10^{-02}}$ sec
2	2.8058×10^{-31}	Yes	3.04786×10^{-01} sec	3.76098×10^{-01} sec	2.33885×10^{-01} sec	$\mathbf{3.98027 \times 10^{-02}}$ sec
3	3.2996×10^{-31}	Yes	3.44262×10^{-01} sec	4.31380×10^{-01} sec	2.66047×10^{-01} sec	$\mathbf{3.95336 \times 10^{-02}}$ sec
4	2.2276×10^{-31}	No	3.31409×10^{-01} sec	4.45042×10^{-01} sec	2.41945×10^{-01} sec	$\mathbf{4.06673 \times 10^{-02}}$ sec
5	2.8058×10^{-31}	No	2.75584×10^{-01} sec	3.93046×10^{-01} sec	2.73172×10^{-01} sec	$\mathbf{4.07170 \times 10^{-02}}$ sec
6	3.2996×10^{-31}	No	3.03051×10^{-01} sec	4.81821×10^{-01} sec	2.41919×10^{-01} sec	$\mathbf{4.01627 \times 10^{-02}}$ sec
7	0.0012032	No	6.96741×10^{-01} sec	6.84522×10^{-01} sec	2.39788×10^{-01} sec	$\mathbf{4.08159 \times 10^{-02}}$ sec
8	0.11949	No	6.45124×10^{-01} sec	6.93940×10^{-01} sec	2.34114×10^{-01} sec	$\mathbf{4.08490 \times 10^{-02}}$ sec
9	0.48775	No	6.35042×10^{-01} sec	6.46904×10^{-01} sec	2.88826×10^{-01} sec	$\mathbf{4.06804 \times 10^{-02}}$ sec
Case	Nonorthogonality ϵ	$\log(\boldsymbol{A}) = \log(\boldsymbol{B})$?	Daniilidis 1999 [9]	Andreff 2001 [8]	Schmidt 2008 [37]	**Proposed Method**
1	2.2276×10^{-31}	Yes	1.39045×10^{-01} sec	8.26899×10^{-02} sec	3.49526×10^{-01} sec	$\mathbf{3.93071 \times 10^{-02}}$ sec
2	2.8058×10^{-31}	Yes	1.51638×10^{-01} sec	9.11281×10^{-02} sec	3.51825×10^{-01} sec	$\mathbf{3.98027 \times 10^{-02}}$ sec
3	2.2996×10^{-31}	Yes	1.35494×10^{-01} sec	8.24217×10^{-02} sec	3.94642×10^{-01} sec	$\mathbf{3.95336 \times 10^{-02}}$ sec
4	2.2276×10^{-31}	No	1.37405×10^{-01} sec	1.00737×10^{-01} sec	4.53282×10^{-01} sec	$\mathbf{4.06673 \times 10^{-02}}$ sec
5	2.8058×10^{-31}	No	1.48113×10^{-01} sec	1.06168×10^{-01} sec	3.56268×10^{-01} sec	$\mathbf{4.07170 \times 10^{-02}}$ sec
6	3.2996×10^{-31}	No	1.38741×10^{-01} sec	8.37352×10^{-02} sec	3.46696×10^{-01} sec	$\mathbf{4.01627 \times 10^{-02}}$ sec
7	0.0012032	No	1.74023×10^{-01} sec	1.04875×10^{-01} sec	4.36616×10^{-01} sec	$\mathbf{4.08159 \times 10^{-02}}$ sec
8	0.11949	No	1.92827×10^{-01} sec	9.99334×10^{-02} sec	5.00230×10^{-01} sec	$\mathbf{4.08490 \times 10^{-02}}$ sec
9	0.48775	No	1.86904×10^{-01} sec	8.49712×10^{-02} sec	4.29743×10^{-01} sec	$\mathbf{4.06804 \times 10^{-02}}$ sec

rotation accuracy results are shown in Table I where our results are marked bold and the best ones in each case are marked using red. The blue-marked values indicate the best ones but they are not the best from pure aspect of solving $\boldsymbol{AR} = \boldsymbol{RB}$.

The three kinds of cases are designed to fully verify the robustness of various algorithms dealing with quite different input matrices. The first three cases reflect very ideal configurations and for studied four algorithm, they behave quite similarly with very accurate determination. When the equality of $\log(\boldsymbol{A}) = \log(\boldsymbol{B})$ no longer holds, from cases $4 \sim 6$, we can see that all the algorithms are sensitive to this kind of noise but the proposed method and Zhang's method [19] have the best performance. While for those cases that contain both kinds of noises (see cases $7 \sim 9$), the estimation errors rise with increasing magnitude of measurement nonorthogonalities. However, in such condition, the proposed method still holds its superiority on accuracy and robustness. The reason is that the proposed QDR method does not depend on the type of input matrix and is thus more robust facing non-rigid matrices. We can also see that the Schmidt's vector quantization method is more accurate than the proposed method. [16], [17], [9], [8] have been detailed in the Section III. [37] reports a novel data selection method based on vector quantization for better accuracy, robustness and success rate of hand-eye calibration. The vector quantization method pays its most attention in data selection and relies on traditional solvers of hand-eye calibration problem. Therefore, it owns better accuracy but will consume more computation time in vector quantization. That is to say, combining the same data selection scheme in [37] with the proposed method will also give more accurate results. Here, repeat representatives are executed for Monte-Carlo performance evaluation with different cases of $\boldsymbol{A}$ and $\boldsymbol{B}$ (see Fig. 2, 3, 4). Each evaluation is conducted for 10000 times for average results. Fig. 2 contains 10 examples where bias-free data is generated with Monte-Carlo simulations to ensure nonorthogonality ϵ being uniformly selected from 1×10^{-31} to 5×10^{-30}. Fig. 3 presents data evaluation results using data where $\log(\boldsymbol{A}) \neq \log(\boldsymbol{B})$ and also $\epsilon \in [1 \times 10^{-31}, 5 \times 10^{-30}]$ uniformly. While in Fig. 4, we simulate 10 examples with ϵ ranging from 5×10^{-3} to 5×10^{-1} uniformly. The three figures correspond to the three kinds of cases in Table I, III respectively. We use the $-20\log(\gamma)$ to enlarge and visualize the difference. That is to say the method with larger error will have shorter bars in the figures. We could see that in almost all examples the proposed method behaves the best. Such statistical verification well coincides with aforementioned accuracy of the proposed algorithm. The current evaluations have completely validated the framework and numerical robustness of the proposed algorithm. The computational efficiency information is provided in the following sub-section.

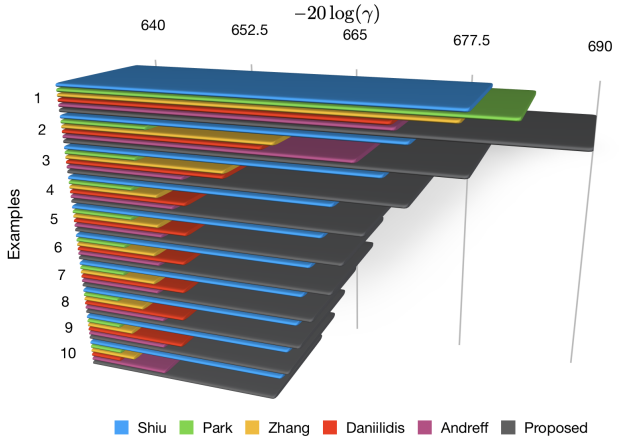


Fig. 2. Performance results using the 10 examples simulated using bias-free data.

B. Computational Efficiency

To evaluate the time complexities of various algorithms, we re-simulate the same cases with that described in last sub-section. All the algorithms are executed for 1000 times for

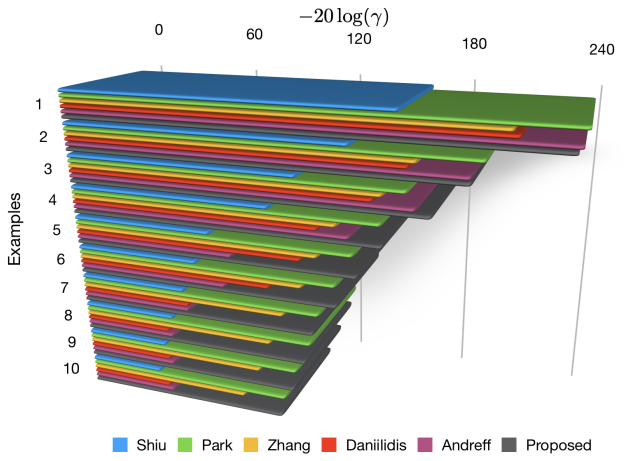


Fig. 3. Performance results using the 10 examples simulated with data where $\log(\boldsymbol{A}) \neq \log(\boldsymbol{B})$.

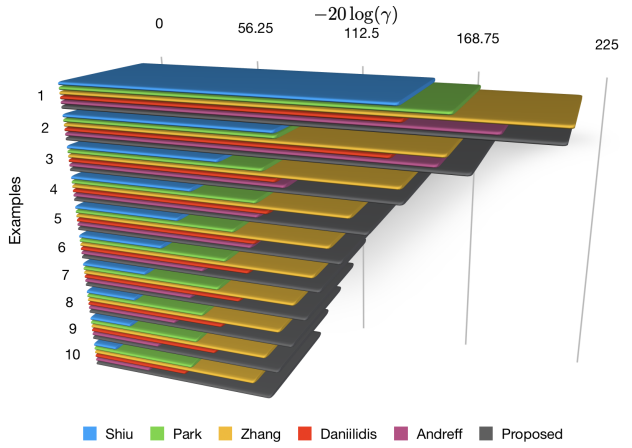


Fig. 4. Performance results using the 10 examples simulated with data where $\log(\boldsymbol{A}) \neq \log(\boldsymbol{B})$ and nonorthogonality varies uniformly from 5×10^{-3} to 5×10^{-1}.

total computation time. They are implemented together using the C++ language and Eigen matrix library on a MacBook Pro 2017 with CPU of i7 3.5GHz. The results are shown in Table II. The architecture of the proposed method is completely closed-form so the execution speed is extremely fast. We can also see that Zhang's method suffers from large computation time variances. The reason is that this algorithm is iterative and significantly depends on the stop criterion given by accuracy or maximum iteration number. Yet the SVD required by Zhang's method is also iterative in general and thus needs uncertain recursions in practice. An indirect reason that the Zhang's method is slower may be that it heavily consumes RAM spaces and thus will need more time in memory scheduling of onboard system. The overall decrease in computation time of the proposed method is about 70.588% to 77.273% which supports the claim of fast computation speed of the proposed method. For a sensor sampling system with frequency of 60fps, running for 5 minutes will generate 18000 samples. Traditional methods will require 2.5 sec $\sim$ 12.5 sec to accomplish. Using the proposed algorithm, such time span can be shortened to merely 0.5 sec $\sim$ 0.8 sec. In other words, the computation resources will be maximumly saved and thus improve the productivity of the whole robotic system.

C. Experimental Validation

The simulation results have shown that the proposed method can at least reach the same accuracy and robustness that Zhang's method [19] owns. In this sub-section, we design a hardware where the MT9V034 CMOS camera, MPU6000 inertial sensor comprising 3-axis gyroscope/accelerometer and HMC5983 digital magnetic compass are integrated. The microcontroller is an STM32H743 with RAM of 1MB and clock speed of 400MHz. The camera/inertial/magnetic sensor samplings are synchronized on 30Hz with on-chip data ready (DRDY) pins. The mounted sensors are rotated in various directions to conduct the calibration process. A sensor log of 5-minute length is generated in this way for final calibration.

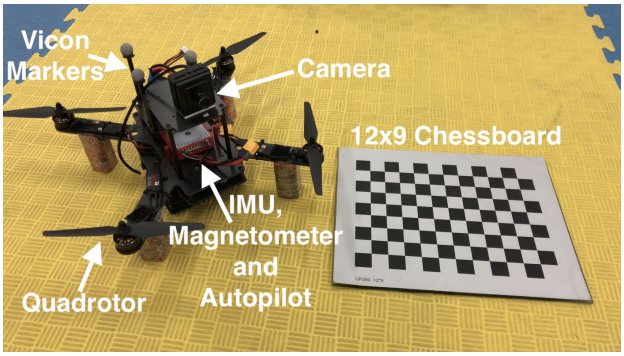


Fig. 5. The designed camera/inertial/magnetic calibration system.

Fig. 6. Hovering quadrotor's attitude/velocity/position information is captured by the Vicon ground-truth system.

The sensor array is installed on a quadrotor platform for the validation of sensor calibration precision. As we know that the camera/inertial/magnetic sensor array is adequate for a monocular visual inertial navigation system (VINS, [44]), after applying the obtained calibration parameters to the designed hardware, we use the calibrated sensor outputs to compute indoor velocities that are later compared with ground-truth information from a high-accuracy Vicon system (see Fig. 6). Note that here the intrinsic parameters of camera is

TABLE III
RMS VELOCITY ACCURACY OF VARIOUS METHODS

Flights	Zhang's method [19] ($\mathcal{M}=10, \varrho=10^{-5}$)	Zhang's method [19] ($\mathcal{M}=50, \varrho=10^{-10}$)	Zhang's method [19] ($\mathcal{M}=100, \varrho=10^{-15}$)	Proposed Method
1: Fast Dynamic move	v_x : 0.409 m/s v_y : 0.527 m/s v_z : 0.221 m/s	v_x : 0.361 m/s v_y : 0.482 m/s v_z : 0.153 m/s	v_x : 0.236 m/s v_y : 0.302 m/s v_z : 0.109 m/s	v_x : 0.258 m/s v_y : 0.293 m/s v_z : 0.098 m/s
2: Rotating move	v_x : 0.501 m/s v_y : 0.426 m/s v_z : 0.379 m/s	v_x : 0.419 m/s v_y : 0.294 m/s v_z : 0.288 m/s	v_x : 0.373 m/s v_y : 0.251 m/s v_z : 0.220 m/s	v_x : 0.321 m/s v_y : 0.267 m/s v_z : 0.192 m/s
3: Slow move	v_x : 0.298 m/s v_y : 0.180 m/s v_z : 0.133 m/s	v_x : 0.207 m/s v_y : 0.152 m/s v_z : 0.107 m/s	v_x : 0.124 m/s v_y : 0.117 m/s v_z : 0.095 m/s	v_x : 0.116 m/s v_y : 0.120 m/s v_z : 0.093 m/s

obtained with [45] and the inertial axes alignment is achieved via offline method in [46] and [47]. The feature extraction is accomplished using the scale-invariant feature transform (SIFT) descriptor [48]. An Nvidia TX1 computation board is employed to conduct image processing and optimization of VINS. Velocity (denoted as $\boldsymbol{v} = (v_x, v_y, v_z)^T$ in the local frame) in visual inertial navigation system are computed directly from rotation estimation and inertial pre-integration [49] so it can directly reflect the accuracy of performed calibration methods. Zhang's method [19] is employed for comparison in this sub-section. Here we denote the maximum iteration of Zhang's method as $\mathcal{M}$ and the relative numerical accuracy being

$$\varrho = \|\boldsymbol{R}_k - \boldsymbol{R}_{k-1}\| \tag{27}$$

where k represents the iteration index for $k = 1, 2, 3, \cdots$. We choose several combinations of $\mathcal{M}$ and ϱ to evaluate the algorithms. In accordance with the performance shown in last sub-section, the proposed method generally owns the same accuracy with Zhang's method that has large $\mathcal{M}$ and low ϱ. The velocity determination results from two algorithms (see Fig. 7 where $\mathcal{M} = 100, \varrho = 10^{-15}$) are generally the same. The multi-flight accuracy can be seen from root-mean-squared (RMS) statistics in Table III. It is shown that the accuracy of the proposed method and Zhang's method [19] ($\mathcal{M} = 100, \varrho = 10^{-15}$) is basically equivalent. The results indicate the real efficacy of the proposed method for robotic calibration and navigation. However, for some realtime applications, the users may not have enough computational resources for many iterations. And in those cases, the accuracy performances are worse. We also gather the run-time stats of the calibration system using the proposed method and Zhang's method [19]. The results can be seen in Table IV. The actual run-time performances of two algorithms basically coincide with the theoretical comparison in Section III. Therefore, the proposed method can replace Zhang's method [19] for future solving of $\boldsymbol{AR} = \boldsymbol{RB}$ problem for much high computational efficiency in practice.

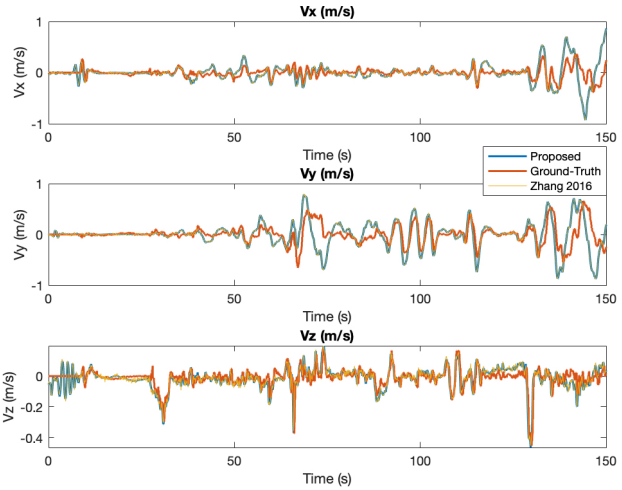


Fig. 7. The computed velocity using the calibrated data from the visual/inertial/magnetic navigation system. For Zhang's method, the parameters are $\mathcal{M} = 100, \varrho = 10^{-15}$.

TABLE IV
RUN-TIME STATS (PEAK)

Algorithms	CPU Load	RAM Occupation
Proposed Method	32.876%	12.201%
Zhang's method [19] ($\mathcal{M}=100, \varrho=10^{-15}$)	96.534%	88.935%
Zhang's method [19] ($\mathcal{M}=50, \varrho=10^{-100}$)	68.279%	82.533%
Zhang's method [19] ($\mathcal{M}=10, \varrho=10^{-5}$)	42.711%	77.607%

V. CONCLUSION

We present a simple algorithm for solving the sensor calibration problem $\boldsymbol{AR} = \boldsymbol{RB}$. Analytical quaternion results are given according to the derived eigenvalue problem for 4×4 matrices. It directly propagates the elements from input matrices without conducting pre-processings such as quaternion conversion or orthonormalization. The proposed method is verified to own the best balanced accuracy, robustness and computation time, compared with representatives. We hope that the proposed method will be of benefit to industrial applications for sensor calibration based on $\boldsymbol{AR} = \boldsymbol{RB}$ problem in the future. The open-source code of this paper is available on `https://github.com/zarathustr/ARRB`.

ACKNOWLEDGMENT

This research was supported by Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ20160401100022706, in part by General Research Fund of Research Grants Council Hong Kong, 11210017, in part by Early Career Scheme Project of Research Grants Council Hong Kong, 21202816. We would like to thank Prof. Zebo Zhou from University of Electronic Science and

Technology of China for his constructive comments to the proposed theory.

REFERENCES

[1] Y. Wu, D. Zou, P. Liu, and W. Yu, "Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering," *IEEE Trans. Control Syst. Tech.*, vol. 26, no. 2, pp. 716–723, 2018.

[2] J. Wu, Z. Zhou, J. Chen, H. Fourati, and R. Li, "Fast Complementary Filter for Attitude Estimation Using Low-Cost MARG Sensors," *IEEE Sensors J.*, vol. 16, no. 18, pp. 6997–7007, 2016.

[3] J. Wu, "Real-time magnetometer disturbance estimation via online nonlinear programming," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4405–4411, 2019.

[4] F. Zheng and Y.-H. Liu, "SE(2)-Constrained Visual Inertial Fusion for Ground Vehicles," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9699–9707, 2018.

[5] Z. Zhou, "Optimal batch distributed asynchronous multi-sensor fusion with feedback," *IEEE Trans. Aerosp. Elec. Syst.*, vol. 55, no. 1, pp. 46–56, 2019.

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

[7] D. Condurache and A. Burlacu, "Orthogonal dual tensor method for solving the AX = XB sensor calibration problem," *Mechanism Mach. Theory*, vol. 104, pp. 382–404, 2016.

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

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

[10] Z. Zhang, L. Zhang, and G. Z. Yang, "A computationally efficient method for hand–eye calibration," *Int. J. Comput. Assist. Radio. Surgery*, vol. 12, no. 10, pp. 1775–1787, 2017.

[11] H. Wang, Y. H. Liu, and D. Zhou, "Dynamic visual tracking for manipulators using an uncalibrated fixed camera," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 610–617, 2007.

[12] H. Tang, Y. Liu, and H. Wang, "Constraint Gaussian Filter With Virtual Measurement for On-Line Camera-Odometry Calibration," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 630–644, 2018.

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

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

[15] R. Y. Tsai and R. K. Lenz, "A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, 1989.

[16] Y. C. Shiu and S. Ahmad, "Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, 1989.

[17] F. C. Park and B. J. Martin, "Robot Sensor Calibration: Solving AX = XB on the Euclidean Group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, 1994.

[18] J. C. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Research*, vol. 10, no. 3, pp. 240–254, 1991.

[19] Z. Q. Zhang, "Cameras and Inertial/Magnetic Sensor Units Alignment Calibration," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 6, pp. 1495–1502, 2016.

[20] 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, no. 8, pp. 1–9, 2019.

[21] G. Chen, G. Cui, Z. Jin, F. Wu, and X. Chen, "Accurate Intrinsic and Extrinsic Calibration of RGB-D Cameras with GP-Based Depth Correction," *IEEE Sensors J.*, vol. 19, no. 7, pp. 2685–2694, 2019.

[22] D. L. Stone, G. Shah, and Y. Motai, "Direct spherical calibration of omnidirectional far infrared camera system," *IEEE Sensors J.*, vol. 19, no. 13, pp. 5285–5298, 2019.

[23] L. Yu, Y. Yan, X. Yu, and Y. Xia, "Design and Realization of Forceps with 3-D Force Sensing Capability for Robot-Assisted Surgical System," *IEEE Sensors J.*, vol. 18, no. 21, pp. 8924–8932, 2018.

[24] L. Yang, E. Li, T. Long, J. Fan, and Z. Liang, "A Novel 3-D Path Extraction Method for Arc Welding Robot Based on Stereo Structured Light Sensor," *IEEE Sensors J.*, vol. 19, no. 2, pp. 763–773, 2019.

[25] J. Fan, F. Jing, L. Yang, L. Teng, and M. Tan, "A Precise Initial Weld Point Guiding Method of Micro-Gap Weld Based on Structured Light Vision Sensor," *IEEE Sensors J.*, vol. 19, no. 1, pp. 322–331, 2019.

[26] D. Modenini, "Attitude determination from ellipsoid observations: A modified orthogonal procrustes problem," *AIAA J. Guid. Control Dyn.*, vol. 41, no. 10, pp. 2320–2325, 2018.

[27] M. Liu, "Robotic online path planning on point cloud," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1217–1228, 2016.

[28] F. Dornaika and R. Horaud, "Simultaneous Robot-World and Hand-Eye Calibration," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 617–622, 1998.

[29] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q. Meng, "Simultaneous Hand-Eye, Tool-Flange, and Robot-Robot Calibration for Comanipulation by Solving the AXB = YCZ Problem," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 413–428, 2016.

[30] Q. Ma, Z. Goh, S. Ruan, and G. S. Chirikjian, "Probabilistic approaches to the AXB= YCZ calibration problem in multi-robot systems," *Auto. Robots*, vol. 42, no. 7, pp. 1497–1520, 2018.

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

[32] A. Mooijaart and J. J. F. Commandeur, "A general solution of the weighted orthonormal procrustes problem," *Psychometrika*, vol. 55, no. 4, pp. 657–663, 1990.

[33] F. L. Markley, "Unit Quaternion from Rotation Matrix," *AIAA J. Guid. Control Dyn.*, vol. 31, no. 2, pp. 440–442, 2008.

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

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

[36] D. A. Cox, *Galois Theory*. John Wiley & Sons, 2011.

[37] J. Schmidt and H. Niemann, "Data selection for hand-eye calibration: A vector quantization approach," *Int. J. Robot. Research*, vol. 27, no. 9, pp. 1027–1053, 2008.

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

[39] J. Wu, "Optimal continuous quaternions from rotation matrices," *AIAA J. Guid. Control Dyn.*, 2018.

[40] M. D. Shuster and S. D. Oh, "Three-axis attitude determination from vector observations," *AIAA J. Guid. Control Dyn.*, vol. 4, no. 1, pp. 70–77, 1981.

[41] F. L. Markley and D. Mortari, "Quaternion Attitude Estimation Using Vector Observations," *J. Astronautical Sci.*, vol. 48, no. 2/3, pp. 359–380, 2000.

[42] STMicroelectronics, "STM32H743VIT6 Datasheet," 2018.

[43] Microchip Inc., "SAM E MCUs."

[44] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[45] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

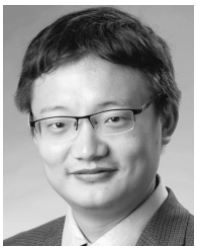
[46] Y. Wu and S. Luo, "On misalignment between magnetometer and inertial sensors," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6288–6297, 2016.

[47] Y. Wu and W. Shi, "On Calibration of Three-Axis Magnetometer," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6424–6431, 2015.

[48] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE ICCV 1999*, vol. 2, 1999, pp. 1150–1157.

[49] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual - Inertial odometry," *IEEE Trans. Robot.*, vol. PP, no. 99, pp. 1–21, 2016.

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



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

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

Dr. Liu was a recipient of the European Micro Aerial Vehicle Competition (EMAV09) (second place) and two awards from International Aerial Robot Competition (IARC14) as a Team Member, the Best Student Paper Award as first author for MFI 2012 (IEEE International Conference on Multisensor Fusion and Information Integration), the Best Paper Award in Information for IEEE International Conference on Information and Automation (ICIA 2013) as first author, the Best Paper Award Finalists as co-author, the Best RoboCup Paper Award for IROS 2013 (IEEE/RSJ International Conference on Intelligent Robots and Systems), the Best Conference Paper Award for IEEE-CYBER 2015, the Best Student Paper Finalist for RCAR 2015 (IEEE International conference on Real-time Computing and Robotics), the Best Student Paper Finalist for ROBIO 2015, the Best Student Paper Award for IEEE-ICAR 2017, the Best Paper in Automation Award for IEEE-ICIA 2017, twice the innoviation contest Chunhui Cup Winning Award in 2012 and 2013, and the Wu Wenjun AI Award in 2016. He was the Program Chair of IEEERCAR 2016 and the Program Chair of International Robotics Conference in Foshan 2017. He was the Conference Chair of ICVS 2017. He has published many popular papers in top journals including IEEE TRANSACTIONS ON ROBOTICS, INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH. Dr. Liu is currently an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS. He is a Senior Member of IEEE.

Yuhua Qi received his B.S. degree from Beijing Institute of Technology, Beijing China, in 2014. He is currently a Ph.D. candidate at Beijing Institute of Technology.

His research interests include cooperative control and autonomous UAV systems.