Fast Symbolic 3-D Registration Solution

Jin Wu, *Member, IEEE*, Ming Liu, *Senior Member, IEEE*, Zebo Zhou, and Rui Li, *Member, IEEE*

***Abstract*— 3-D registration has always been performed invoking singular value decomposition (SVD) or eigenvalue decomposition (EIG) in real engineering practices. However, these numerical algorithms suffer from uncertainty of convergence in many cases. A novel fast symbolic solution is proposed in this article by following our recent publication in this journal. The equivalence analysis shows that our previous solver can be converted to deal with the 3-D registration problem. Rather, the computation procedure is studied for further simplification of computing without complex-number support. Experimental results show that the proposed solver does not loose accuracy and robustness but improves the execution speed to a large extent by almost 50%–80%, on both a personal computer (PC) and an embedded processor.**

***Note to Practitioners*—3-D registration usually has a large computational burden in engineering tasks. The proposed symbolic solution can directly solve the eigenvalue and its associated eigenvector. A lot of computation resources can then be saved for better overall system performance. The deterministic behavior of the proposed solver also ensures long-endurance stability and can help an engineer better design thread timing logic.**

***Index Terms*— 3-D registration, fast computation speed, numerical algorithms, robotics, symbolic computation.**

I. Introduction

MOTION estimation from point correspondences is an important technique in robotics [1]–[3]. The point measurements can usually be acquired from laser scanner and camera for accurate relative attitude/position determination [4], [5]. The methodology behind is called the 3-D registration which figures out the rigid transformation consisting of rotation and translation [6]. Yet, this technology is employed for 3-D reconstruction of objects by means of multi-directional point-cloud snapshots, which extensively boosts the automation assembly [7]–[9]. Thanks to 3-D registration, the image stitching can be performed accurately for better sequence processing [10]. Moreover, the navigation performance of intelligent vehicles can be improved by existing registration techniques [11], [12].

The basic 3-D registration problem can usually be expressed as a least-square fitting one that takes the following form [13]:

$$\underset{\boldsymbol{C}\in\mathrm{SO}(3),\boldsymbol{T}\in\mathbb{R}^3}{\arg\min}\ \mathcal{L}=\sum_{i=1}^{n}a_i\|\boldsymbol{b}_i-\boldsymbol{C}\boldsymbol{r}_i-\boldsymbol{T}\|^2 \tag{1}$$

where $\boldsymbol{b}_i\in\{\mathcal{B}\}$ and $\boldsymbol{r}_i\in\{\mathcal{R}\}$ are the point measurement pair in the body and reference frames; a_i is the positive weight associated with the ith point pair. The target is to estimate the direction cosine matrix $\boldsymbol{C}$ in the special orthogonal group $\mathrm{SO}(3)$ such that $\boldsymbol{C}\boldsymbol{C}^T=\boldsymbol{I},\det(\boldsymbol{C})=+1$, with $\boldsymbol{T}$ in the real 3-D vector space $\mathbb{R}^3$, to minimize the sum $\mathcal{L}$. As there are both noise items inside $\boldsymbol{b}_i$ and $\boldsymbol{r}_i$, the problem is actually a total least square (TLS) [14], [15]. In real engineering applications, $\{\mathcal{B}\}$ and $\{\mathcal{R}\}$ do not always agree in the dimension. So, the problem (1) is usually dealt with using the iterative closest points (ICPs) for robust matching [16]. Apart from the ICP, i.e., only find-based approach, the local geometric features inside the point clouds and images can also help for more robust matching [17]. Local geometric features are more advanced understanding and aspects inside data sequences, and they can reflect those regular and visualizable geometric characteristics [18], [19], which intrinsically enhance the performance of 3-D registration in urban areas, e.g., in applications of the lidar odometry and mapping in real time (LOAM) [20], [21]). When there are rare local geometric features in the data, the ICP is still generally feasible for processing global registration. Some algorithms have been proposed in the last 30 years to solve $\boldsymbol{C}$ and $\boldsymbol{T}$ from (1) efficiently. The first famous solver was proposed by Arun *et al*. [22] who introduces singular value decomposition (SVD) for rotation estimation. However, when the problem contains a large noise density, only SVD cannot give robust estimation. Umeyama [23] improves Arun's method by changing the signs of the singular values. In fact, the only difficulty of the optimization (1) is that $\boldsymbol{C}$ is nonlinear. However, after parameterizing $\boldsymbol{C}$ with dual quaternion, the problem can also be solved [24]. A simpler approach is established by unit quaternions which converts the problem (1) into an eigenvalue decomposition (EIG) one [25]. In fair comparisons, the dual quaternion is the slowest, while the EIG is slightly slower than SVD [26]. But for both EIG and SVD, the numerical implementation requires many computation loads and space consumption of required libraries. This generates a difficulty for their usage on some critical platforms, e.g., field programmable gate arrays (FPGAs) and some low-configuration microcontroller units (MCUs) [27]. The current situation also sets an obstacle

Manuscript received May 29, 2018; revised January 3, 2019; accepted September 16, 2019. Date of publication October 8, 2019; date of current version April 7, 2020. This article was recommended for publication by Associate Editor D. Liu and Editor Y. Sun upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 41604025, in part by the General Research Fund of Research Grants Council, Hong Kong, under Grant 11210017, and in part by the Early Career Scheme Project of Research Grants Council, Hong Kong, under Grant 21202816. The work of M. Liu was supported by the Shenzhen Science, Technology and Innovation Commission (SZSTI) under Grant JCYJ20160401100022706. *(Corresponding author: Ming Liu.)*

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

Z. Zhou and R. Li are with the University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: klinsmann.zhou@gmail.com; hitlirui@gmail.com).

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/TASE.2019.2942324

for the mass production of specified low-power integrated circuit (IC).

Recently, we propose an algorithm for vector-observation attitude determination called the fast linear attitude estimator (FLAE) [28]. FLAE owns the much superior computation speed compared with previous representatives. Motivated by SVD, EIG, and FLAE, in this article, a novel symbolic method is proposed. Through tests, the algorithm is verified to have only 50%–80% execution time of recent fast SVD and EIG by C++ implementation on both personal computer (PC) and the MCU. In Section II, we present how to relate (1) with FLAE together. A simplified algorithm of eigenvalue is derived as well in this section. Experimental validations are presented in Section III, while Section IV consists of concluding remarks.

II. Main Results

The FLAE actually solves a more specific variant of (1) where $\boldsymbol{T}=0$ and $\|\boldsymbol{b}_i\|=\|\boldsymbol{r}_i\|=1$ [28], which can be further extended to some optimized results [29]. This is a preliminary for attitude determination from normalized vector observations in spacecraft motion measurement. In FLAE, the DCM is parameterized by the unit quaternion. The optimal quaternion is associated with the eigenvalue of $\boldsymbol{W}$ that is closest to 1 where $\boldsymbol{W}$ is given by

$$\begin{aligned}
\boldsymbol{W}_{1,1}&=H_{x1}+H_{y2}+H_{z3}\\
\boldsymbol{W}_{1,2}&=-H_{y3}+H_{z2}\\
\boldsymbol{W}_{1,3}&=-H_{z1}+H_{x3}\\
\boldsymbol{W}_{1,4}&=-H_{x2}+H_{y1}\\
\boldsymbol{W}_{2,1}&=-H_{y3}+H_{z2}\\
\boldsymbol{W}_{2,2}&=H_{x1}-H_{y2}-H_{z3}\\
\boldsymbol{W}_{2,3}&=H_{x2}+H_{y1}\\
\boldsymbol{W}_{2,4}&=H_{x3}+H_{z1}\\
\boldsymbol{W}_{3,1}&=-H_{z1}+H_{x3}\\
\boldsymbol{W}_{3,2}&=H_{x2}+H_{y1}\\
\boldsymbol{W}_{3,3}&=H_{y2}-H_{x1}-H_{z3}\\
\boldsymbol{W}_{3,4}&=H_{y3}+H_{z2}\\
\boldsymbol{W}_{4,1}&=-H_{x2}+H_{y1}\\
\boldsymbol{W}_{4,2}&=H_{x3}+H_{z1}\\
\boldsymbol{W}_{4,3}&=H_{y3}+H_{z2}\\
\boldsymbol{W}_{4,4}&=H_{z3}-H_{y2}-H_{x1}
\end{aligned}\tag{2}$$

in which $\boldsymbol{W}_{i,j}$ denotes the matrix entry of $\boldsymbol{W}$ in the ith row and jth column. The parameters inside are provided as follows:

$$\boldsymbol{H}=\begin{pmatrix}H_{x1}&H_{y1}&H_{z1}\\H_{x2}&H_{y2}&H_{z2}\\H_{x3}&H_{y3}&H_{z3}\end{pmatrix}=\sum_{i=1}^{n}a_i\boldsymbol{b}_i\boldsymbol{r}_i^T.\tag{3}$$

The characteristic polynomial of $\boldsymbol{W}$ takes the form of

$$\lambda^4+\tau_1\lambda^2+\tau_2\lambda+\tau_3=0\tag{4}$$

where

$$\begin{aligned}
\tau_1&=-2\begin{pmatrix}H_{x1}^2+H_{x2}^2+H_{x3}^2+H_{y1}^2\\+H_{y2}^2+H_{y3}^2+H_{z1}^2+H_{z2}^2+H_{z3}^2\end{pmatrix}\\
\tau_2&=8(H_{x3}H_{y2}H_{z1}-H_{x2}H_{y3}H_{z1}-H_{x3}H_{y1}H_{z2}\\
&\quad+H_{x1}H_{y3}H_{z2}+H_{x2}H_{y1}H_{z3}-H_{x1}H_{y2}H_{z3})\\
\tau_3&=\det(\boldsymbol{W}).
\end{aligned}\tag{5}$$

For the problem (1), the quaternion solution is produced by the optimal eigenvector of the following matrix [1], [16]:

$$\boldsymbol{G}=\begin{bmatrix}tr(\boldsymbol{D})&\boldsymbol{z}^T\\\boldsymbol{z}&\boldsymbol{D}+\boldsymbol{D}^T-tr(\boldsymbol{D})\boldsymbol{I}\end{bmatrix}\tag{6}$$

in which

$$\begin{aligned}
\boldsymbol{D}&=\sum_{i=1}^{n}a_i(\boldsymbol{b}_i-\bar{\boldsymbol{b}})(\boldsymbol{r}_i-\bar{\boldsymbol{r}})^T\\
\boldsymbol{z}&=\sum_{i=1}^{n}a_i(\boldsymbol{b}_i-\bar{\boldsymbol{b}})\times(\boldsymbol{r}_i-\bar{\boldsymbol{r}})\\
\bar{\boldsymbol{b}}&=\sum_{i=1}^{n}a_i\boldsymbol{b}_i,\bar{\boldsymbol{r}}=\sum_{i=1}^{n}a_i\boldsymbol{r}_i.
\end{aligned}\tag{7}$$

It is obvious that $\boldsymbol{D}$ has the same structure with $\boldsymbol{H}$. Then, if $\boldsymbol{D}=\boldsymbol{H}$, we would like to prove that $\boldsymbol{G}=\boldsymbol{W}$. It can be directly obtained that

$$\begin{aligned}
&\boldsymbol{D}+\boldsymbol{D}^T-tr(\boldsymbol{D})\boldsymbol{I}\\
&=\begin{pmatrix}H_{x1}-H_{y2}-H_{z3}&H_{x2}+H_{y1}&H_{x3}+H_{z1}\\H_{x2}+H_{y1}&H_{y2}-H_{x1}-H_{z3}&H_{y3}+H_{z2}\\H_{x3}+H_{z1}&H_{y3}+H_{z2}&H_{z3}-H_{y2}-H_{x1}\end{pmatrix}\\
&\mathrm{tr}(\boldsymbol{D})\\
&=H_{x1}+H_{y2}+H_{z3}.
\end{aligned}\tag{8}$$

For $\boldsymbol{z}$, it has another form according to the skew-symmetric matrix of cross product, such that

$$\begin{aligned}
\boldsymbol{z}&=(\boldsymbol{D}_{2,3}-\boldsymbol{D}_{3,2},\boldsymbol{D}_{3,1}-\boldsymbol{D}_{1,3},\boldsymbol{D}_{1,2}-\boldsymbol{D}_{2,1})^T\\
&=(-H_{y3}+H_{z2},-H_{z1}+H_{x3},-H_{x2}+H_{y1})^T.
\end{aligned}\tag{9}$$

Inserting these results into (6), one can observe that $\boldsymbol{W}=\boldsymbol{G}$. So, the characteristic polynomial of $\boldsymbol{W}$ can also be used for eigenvalue solving of $\boldsymbol{G}$.

The FLAE gives the following symbolic roots of (4):

$$\begin{aligned}
\lambda_1&=\frac{1}{2\sqrt{6}}\left(T_2-\sqrt{-T_2^2-12\tau_1-\frac{12\sqrt{6}\tau_2}{T_2}}\right)\\
\lambda_2&=\frac{1}{2\sqrt{6}}\left(T_2+\sqrt{-T_2^2-12\tau_1-\frac{12\sqrt{6}\tau_2}{T_2}}\right)\\
\lambda_3&=-\frac{1}{2\sqrt{6}}\left(T_2+\sqrt{-T_2^2-12\tau_1+\frac{12\sqrt{6}\tau_2}{T_2}}\right)\\
\lambda_4&=-\frac{1}{2\sqrt{6}}\left(T_2-\sqrt{-T_2^2-12\tau_1+\frac{12\sqrt{6}\tau_2}{T_2}}\right)
\end{aligned}\tag{10}$$

in which

$$\begin{aligned} T_0 &= 2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3 \\ T_1 &= \left(T_0 + \sqrt{-4\left(\tau_1^2 + 12\tau_3\right)^3 + T_0^2}\right)^{\frac{1}{3}} \\ T_2 &= \sqrt{-4\tau_1 + \frac{2^{\frac{4}{3}}\left(\tau_1^2 + 12\tau_3\right)}{T_1} + 2^{\frac{2}{3}}T_1}. \end{aligned} \tag{11}$$

Let us first determine the signs of τ_1, τ_2, and τ_3. $\boldsymbol{W}$ is real symmetric, and the eigenvalues are two positive and two negative values. This gives $\tau_3 = \det(\boldsymbol{W}) = \lambda_1\lambda_2\lambda_3\lambda_4 > 0$. τ_1 is obviously negative, and τ_2 is indefinite. In this way, T_0 is definitely real number. Let us write T_1 and T_2 into

$$\begin{aligned} T_1 &= \alpha_{T_1} + \beta_{T_1}\boldsymbol{i} \\ T_2 &= \alpha_{T_2} + \beta_{T_2}\boldsymbol{i} \end{aligned} \tag{12}$$

where $\boldsymbol{i}$ denotes the unit imaginary number, while $\alpha_{T_1}, \beta_{T_1}, \alpha_{T_2}, \beta_{T_2} \in \mathbb{R}$. Obviously, T_1 meets

$$\begin{aligned} T_1^3 &= \alpha_{T_1}^3 - 3\alpha_{T_1}\beta_{T_1}^2 + \left(3\alpha_{T_1}^2\beta_{T_1} - \beta_{T_1}^3\right)\boldsymbol{i} \\ &= T_0 + \sqrt{-4(\tau_1^2 + 12\tau_3)^3 + T_0^2}. \end{aligned} \tag{13}$$

Likewise, we have

$$\begin{aligned} T_2^2 &= \alpha_{T_2}^2 - \beta_{T_2}^2 + 2\alpha_{T_2}\beta_{T_2}\boldsymbol{i} \\ &= -4\tau_1 + \frac{2^{\frac{4}{3}}\left(\tau_1^2 + 12\tau_3\right)}{T_1} + 2^{\frac{2}{3}}T_1 \\ &= -4\tau_1 + \frac{2^{\frac{4}{3}}\left(\tau_1^2 + 12\tau_3\right)}{\alpha_{T_1} + \beta_{T_1}\boldsymbol{i}} + 2^{\frac{2}{3}}\left(\alpha_{T_1} + \beta_{T_1}\boldsymbol{i}\right) \\ &= -4\tau_1 + \left[\sqrt[3]{4} + \frac{2\sqrt[3]{2}\left(\tau_1^2 + 12\tau_3\right)}{\alpha_{T_1}^2 + \beta_{T_1}^2}\right]\alpha_{T_1} \\ &\quad + \left[\sqrt[3]{4} - \frac{2\sqrt[3]{2}\left(\tau_1^2 + 12\tau_3\right)}{\alpha_{T_1}^2 + \beta_{T_1}^2}\right]\beta_{T_1}\boldsymbol{i}. \end{aligned} \tag{14}$$

These equations lead to the system of

$$\begin{cases} \alpha_{T_1}^3 - 3\alpha_{T_1}\beta_{T_1}^2 = T_0 \\ \left(3\alpha_{T_1}^2\beta_{T_1} - \beta_{T_1}^3\right)^2 = 4\left(\tau_1^2 + 12\tau_3\right)^3 - T_0^2 \\ \alpha_{T_2}^2 - \beta_{T_2}^2 = -4\tau_1 + \left[\sqrt[3]{4} + \dfrac{2\sqrt[3]{2}\left(\tau_1^2 + 12\tau_3\right)}{\alpha_{T_1}^2 + \beta_{T_1}^2}\right]\alpha_{T_1} \\ 2\alpha_{T_2}\beta_{T_2} = \left[\sqrt[3]{4} - \dfrac{2\sqrt[3]{2}\left(\tau_1^2 + 12\tau_3\right)}{\alpha_{T_1}^2 + \beta_{T_1}^2}\right]\beta_{T_1}. \end{cases} \tag{15}$$

From the first two subequations, one can easily arrive at

$$\begin{aligned} &\left(\alpha_{T_1}^3 - 3\alpha_{T_1}\beta_{T_1}^2\right)^2 + \left(3\alpha_{T_1}^2\beta_{T_1} - \beta_{T_1}^3\right)^2 = 4\left(\tau_1^2 + 12\tau_3\right)^3 \\ &\quad \Rightarrow \alpha_{T_1}^6 + 3\alpha_{T_1}^4\beta_{T_1}^2 + 3\alpha_{T_1}^2\beta_{T_1}^4 + \beta_{T_1}^6 = 4\left(\tau_1^2 + 12\tau_3\right)^3 \\ &\quad \Rightarrow \alpha_{T_1}^2 + \beta_{T_1}^2 = \sqrt[3]{4}\left(\tau_1^2 + 12\tau_3\right). \end{aligned} \tag{16}$$

Inserting (16) into (15), we have

$$\begin{cases} \alpha_{T_2}^2 - \beta_{T_2}^2 = -4\tau_1 + 2\sqrt[3]{4}\alpha_{T_1} \\ 2\alpha_{T_2}\beta_{T_2} = 0. \end{cases} \tag{17}$$

This indicates that $\alpha_{T_2} = 0$ or $\beta_{T_2} = 0$. If $\alpha_{T_2} = 0$, then T_2 is a pure imaginary number leading to the eigenvalues of complex numbers, which is not true for the real symmetric matrix. Therefore, we have $\beta_{T_2} = 0$, i.e., T_2 is a pure positive real number with no imaginary part. Using this finding, the maximum eigenvalue is immediately λ_2. The components of T_1 can be computed using

$$\begin{aligned} T_1^3 &= T_0 + \sqrt{-4\left(\tau_1^2 + 12\tau_3\right)^3 + T_0^2} \\ &= T_0 + \sqrt{4\left(\tau_1^2 + 12\tau_3\right)^3 - T_0^2}\boldsymbol{i} \\ &= 2\left(\tau_1^2 + 12\tau_3\right)^{\frac{3}{2}} e^{\boldsymbol{i}\arctan\frac{\sqrt{4(\tau_1^2+12\tau_3)^3 - T_0^2}}{T_0}}. \end{aligned} \tag{18}$$

After the maximum eigenvalue is computed, the elementary row operations are needed to calculate the associated eigenvector from $(\boldsymbol{G} - \lambda_{\max}\boldsymbol{I})\boldsymbol{q} = \boldsymbol{0}$. Given an arbitrary real symmetric matrix as [28]

$$\boldsymbol{G} - \lambda_{\max}\boldsymbol{I} = \begin{pmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{12} & G_{22} & G_{23} & G_{24} \\ G_{13} & G_{23} & G_{33} & G_{34} \\ G_{14} & G_{24} & G_{34} & G_{44} \end{pmatrix}. \tag{19}$$

The optimal quaternion $\boldsymbol{q}$ from row operations can be categorized as follows:

$$\begin{aligned} q_0 &= G_{14}G_{23}^2 - G_{13}G_{24}G_{23} - G_{12}G_{34}G_{23} \\ &\quad - G_{14}G_{22}G_{33} + G_{12}G_{24}G_{33} + G_{13}G_{22}G_{34} \\ q_1 &= G_{24}G_{13}^2 - G_{12}G_{34}G_{13} - G_{13}G_{14}G_{23} \\ &\quad + G_{12}G_{14}G_{33} - G_{11}G_{24}G_{33} + G_{11}G_{23}G_{34} \\ q_2 &= G_{34}G_{12}^2 - G_{14}G_{23}G_{12} - G_{13}G_{24}G_{12} \\ &\quad + G_{13}G_{14}G_{22} + G_{11}G_{23}G_{24} - G_{11}G_{22}G_{34} \\ q_3 &= -G_{33}G_{12}^2 + 2G_{13}G_{23}G_{12} - G_{11}G_{23}^2 - G_{13}^2G_{22} \\ &\quad + G_{11}G_{22}G_{33} \end{aligned} \tag{20}$$

where $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T$. The estimated attitude quaternion is then $\hat{\boldsymbol{q}} = \boldsymbol{q}/\|\boldsymbol{q}\|$.

A. Numerical Robustness

Here, the numerical robustness of the proposed method is referred to the behavior when the two largest eigenvalues almost coincide. In such extreme case, the measurements from $\{\mathcal{B}\}$ and $\{\mathcal{R}\}$ are basically collinear [30]. This makes $\boldsymbol{D}$ almost a rank-deficient matrix. Then, we immediately have

$$\tau_2 = -8\det(\boldsymbol{D}) \approx 0 \tag{21}$$

and also

$$\begin{cases} \tau_1^2 - 4\tau_3 = 0 \\ \tau_3 = \det(\boldsymbol{W}) = \lambda_1\lambda_2\lambda_3\lambda_4 = \lambda_{\max}^4. \end{cases} \tag{22}$$

Inserting $\tau_1^2 = 4\tau_3$ into (18), it is quite straightforward for one to obtain

$$\begin{aligned} &\sqrt{4(\tau_1^2 + 12\tau_3)^3 - T_0^2} \approx 0 \\ &\Rightarrow \begin{cases} \theta \approx 0 \\ \alpha_{T_1} \approx -2\sqrt[3]{2}\tau_1 \end{cases} \Rightarrow T_2 \approx 0. \end{aligned} \tag{23}$$

Note that here, both T_2 and τ_2 approach 0 and there is an indefinite limit in the eigenvalue, i.e., $\lim_{\tau_2\to 0}(\tau_2/T_2)$. Repeating the L'Hospital rule, we can eventually arrive at (24), as shown at the bottom of this page, where d is the differentiation operator. Therefore, the limiting maximum eigenvalue is

$$\lambda_{\max} \approx \sqrt{-\frac{\tau_1}{2}}. \tag{25}$$

This indicates that in extreme cases, the eigenvalue is still not singular which always leads to meaningful quaternion solutions. However, for iterative algorithms, such as Gauss–Newton iteration, the solving process can hardly stop according to word length of floating numbers [30]. This shows that the proposed method may be more practical in real engineering implementation. The final computation procedure is summarized in Algorithm 1.

Algorithm 1 FS3R Algorithm

Require: Point correspondences $\{\mathcal{B}\}$ and $\{\mathcal{R}\}$ with the same dimension of n, provided that the weights $\{a_i, i = 1, 2, 3, \cdots\}$ exist. If no weights, each weight is equalized to $\frac{1}{a}$. The numerical tolerance threshold for detecting extreme case is defined as ξ which is normally a very tiny positive number.

Step 1: Calculate mean points $\bar{\boldsymbol{b}} = \sum_{i=1}^{n} a_i \boldsymbol{b}_i, \bar{\boldsymbol{r}} = \sum_{i=1}^{n} a_i \boldsymbol{r}_i$.

Step 2: Compute $\boldsymbol{H}$ matrix using form $\boldsymbol{H} = \sum_{i=1}^{n} a_i(\boldsymbol{b}_i - \bar{\boldsymbol{b}})(\boldsymbol{r}_i - \bar{\boldsymbol{r}})^T$ and then compute $\boldsymbol{W}$ using (2).

Step 3: Compute coefficients of characteristic polynomial from (5).

Step 4: Compute $T_0 = 2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3$ and then compute

$$\theta = \arctan \frac{\sqrt{4(\tau_1^2 + 12\tau_3)^3 - T_0^2}}{T_0}$$

T_1 by

$$\alpha_{T_1} = \sqrt[3]{2}\sqrt{\tau_1^2 + 12\tau_3}\cos\frac{\theta}{3}$$

$$\beta_{T_1} = \sqrt[3]{2}\sqrt{\tau_1^2 + 12\tau_3}\sin\frac{\theta}{3}$$

Step 5: Compute $T_2 = |\alpha_{T_2}| = \sqrt{-4\tau_1 + 2\sqrt[3]{4}\alpha_{T_1}}$. If $|\tau_2| > \xi, |T_2| > \xi$, then compute the maximum eigenvalue $\lambda_{\max} = \frac{1}{2\sqrt{6}}\left(T_2 + \sqrt{-T_2^2 - 12\tau_1 - \frac{12\sqrt{6}\tau_2}{T_2}}\right)$. Else, compute eigenvalue according to (25).

Step 6: Compute required elements in (19) and then calculate the normalized unit quaternion according to (20).

Step 7: Reconstruct the rotation from quaternion as $\boldsymbol{C}$. The translation is computed by $\boldsymbol{T} = \bar{\boldsymbol{b}} - \boldsymbol{C}\bar{\boldsymbol{r}}$.

III. Experiments and Comparisons

In this section, several experiments are conducted to present the comparisons of the proposed fast symbolic 3-D registration (FS3R) algorithm with representatives. Note that recently, some similar analytical methods have been proposed. For instance, Yang and Zhou developed an analytical method for root solving of quartic equation [31]. In addition, a novel analytical SVD method is proposed recently by us to conduct factorization of 3×3 matrix [32]. These methods are faster than representative numerical ones. Therefore, we mainly compare them with the proposed FS3R on the accuracy, robustness, and computation speed. The algorithms are first implemented using MATLAB for the validation of accuracy and robustness. They are then translated into C++ programming language for rigorous execution time performance test on both the PC and ARM processors.

A. Accuracy and Robustness Performance

In this section, the statistics are collected using the MATLAB r2016a software on a MacBook Pro 2017 with the CPU clock speed of i7 4-core 3.5 GHz. Here, simulated samples with different dimensions and noise density are generated by means of

$$\boldsymbol{b}_i = \boldsymbol{C}\boldsymbol{r}_i + \boldsymbol{T} + \boldsymbol{\eta}_i, \quad i = 1, 2, \ldots, n \tag{26}$$

where $\boldsymbol{\eta}_i$ denotes the noise item subject to normal distribution with zero mean and covariance of $\boldsymbol{\Sigma}_{\boldsymbol{\eta}_i}$. By designing the experiments in Table I, we evaluate the accuracy and robustness performance of various algorithms. The first cases employ the same rotation and translation while they differ mainly in $\mathrm{rank}(\boldsymbol{D})$ column. When $\mathrm{rank}(\boldsymbol{D}) < 3$, the case is defined to be extreme and some methods will fail to converge. Cases $4 \sim 6$ consist of comparisons with different vector numbers. In cases $7 \sim 9$, we mainly describe the effect of the noise density. The evaluated results are shown in Tables II–IV for rotation, translation, and loss function value $\mathcal{L}$ in (1), respectively. The rotation matrix is first estimated and then converted to the Euler angles, i.e., roll φ, pitch ϑ, and yaw ψ through "$X - Y - Z$" rotation sequence. The NaN value stands for the "Not a Number" one, which is usually caused by indefinite divisions like $(0/0)$ and (∞/∞). Here, the "SVD" and "EIG" are implemented using MATLAB

$$\lim_{\tau_2\to 0}\frac{\tau_2}{T_2} = \lim_{\tau_2\to 0}\frac{d\tau_2}{d\sqrt{\begin{array}{l}-4\tau_1 + \dfrac{2^{\frac{4}{3}}\left(\tau_1^2+12\tau_3\right)}{\left(\begin{array}{l}2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3\\ +\sqrt{-4\left(\tau_1^2+12\tau_3\right)^3 + \left(2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3\right)^2}\end{array}\right)^{\frac{1}{3}}}\\ +2^{\frac{2}{3}}\left(\begin{array}{l}2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3\\ +\sqrt{-4\left(\tau_1^2+12\tau_3\right)^3 + \left(2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3\right)^2}\end{array}\right)^{\frac{1}{3}}\end{array}}} = 0 \tag{24}$$

TABLE I
STUDIED CASES FOR COMPARISONS

Case	Euler Angles φ, ϑ, ψ	Translation $\boldsymbol{T}$	Noise Covariance $\boldsymbol{\Sigma}_{\boldsymbol{\eta}_i}$	Vector Number n	$\mathrm{rank}(\boldsymbol{D})$
1	$\left(-\frac{\pi}{6}, \frac{4\pi}{11}, -\frac{5\pi}{7}\right) = (-0.52359878, 1.1423973, -2.2439948)$	$(100, -50, 80)^T$	$diag\,(0.0, 0.0, 0.0)$	100	3
2	$\left(-\frac{\pi}{6}, \frac{4\pi}{11}, -\frac{5\pi}{7}\right) = (-0.52359878, 1.1423973, -2.2439948)$	$(100, -50, 80)^T$	$diag\,(0.0, 0.0, 0.0)$	100	2
3	$\left(-\frac{\pi}{6}, \frac{4\pi}{11}, -\frac{5\pi}{7}\right) = (-0.52359878, 1.1423973, -2.2439948)$	$(100, -50, 80)^T$	$diag\,(0.0, 0.0, 0.0)$	100	1
4	$\left(\frac{4\pi}{7}, \frac{\pi}{2}, -\frac{9\pi}{20}\right) = (1.7951958, 1.5707963, -1.4137167)$	$(-60, 70, 40)^T$	$diag\,(10, 10, 10)$	100	3
5	$\left(\frac{4\pi}{7}, \frac{\pi}{2}, -\frac{9\pi}{20}\right) = (1.7951958, 1.5707963, -1.4137167)$	$(-60, 70, 40)^T$	$diag\,(10, 10, 10)$	1000	3
6	$\left(\frac{4\pi}{7}, \frac{\pi}{2}, -\frac{9\pi}{20}\right) = (1.7951958, 1.5707963, -1.4137167)$	$(-60, 70, 40)^T$	$diag\,(10, 10, 10)$	10000	3
7	$\left(\frac{5\pi}{9}, -\frac{7\pi}{10}, \frac{4\pi}{13}\right) = (-1.3962634, -0.9424778, -2.1749488)$	$(80, -20, -160)^T$	$diag\,(0.1, 10, 1000)$	1000	3
8	$\left(\frac{5\pi}{9}, -\frac{7\pi}{10}, \frac{4\pi}{13}\right) = (-1.3962634, -0.9424778, -2.1749488)$	$(80, -20, -160)^T$	$diag\,(1000, 10, 0.1)$	1000	3
9	$\left(\frac{5\pi}{9}, -\frac{7\pi}{10}, \frac{4\pi}{13}\right) = (-1.3962634, -0.9424778, -2.1749488)$	$(80, -20, -160)^T$	$diag\,(0.1, 0, 1, 0.1)$	1000	3

TABLE II
ESTIMATED EULER ANGLES φ, ϑ, AND ψ

Case	SVD	EIG	EIG Analytical	SVD Analytical	Proposed FS3R
1	$(-0.5235, 1.1424, -2.2439)$	$(-0.5235, 1.1424, -2.2439)$	$(-0.5235, 1.1424, -2.2439)$	$(-0.5235, 1.1424, -2.2439)$	$(\mathbf{-0.5235, 1.1424, -2.2439})$
2	$(-2.8874, 0.6156, -2.3558)$	$(1.3088, 0.6156, -2.3558)$	$(1.3088, 0.6156, -2.3558)$	(NaN, NaN, NaN)	$(\mathbf{1.3088, 0.6156, -2.3558})$
3	$(-0.04696, -0.04481, -0.04696)$	$(-2.0344, 0.7297, -2.0344)$	$(-2.0344, 0.7297, -2.0344)$	$(0.4621, -1.571 + 10.693\boldsymbol{i}, 2.356)$	$(\mathbf{-2.0344, 0.7297, -2.0344})$
4	$(0.3614, 1.258, -0.5719)$	$(0.3614, 1.258, -0.5719)$	$(0.3614, 1.258, -0.5719)$	$(0.3614, 1.258, -0.5719)$	$(\mathbf{0.3614, 1.258, -0.5719})$
5	$(0.6665, 1.4052, -0.4474)$	$(0.6665, 1.4052, -0.4474)$	$(0.6665, 1.4052, -0.4474)$	$(0.6665, 1.4052, -0.4474)$	$(\mathbf{0.6665, 1.4052, -0.4474})$
6	$(-0.02599, 1.4942, 0.4477)$	$(-0.02599, 1.4942, 0.4477)$	$(-0.02599, 1.4942, 0.4477)$	$(-0.02599, 1.4942, 0.4477)$	$(\mathbf{-0.02599, 1.4942, 0.4477})$
7	$(2.7465, 0.5139, 2.7899)$	$(2.7465, 0.5139, 2.7899)$	$(2.7465, 0.5139, 2.7899)$	$(2.7465, 0.5139, 2.7899)$	$(\mathbf{2.7465, 0.5139, 2.7899})$
8	$(0.2577, -0.3486, 0.2181)$	$(0.2577, -0.3486, 0.2181)$	$(0.2577, -0.3486, 0.2181)$	$(0.2577, -0.3486, 0.2181)$	$(\mathbf{0.2577, -0.3486, 0.2181})$
9	$(-1.4018, -0.9443, -2.1804)$	$(-1.4018, -0.9443, -2.1804)$	$(-1.4018, -0.9443, -2.1804)$	$(-1.4018, -0.9443, -2.1804)$	$(\mathbf{-1.4018, -0.9443, -2.1804})$

TABLE III
ESTIMATED TRANSLATION $\boldsymbol{T}$

Case	SVD	EIG	EIG Analytical	SVD Analytical	Proposed FS3R
1	$(100.015, -50.0834, 79.9858)^T$	$(100.015, -50.0834, 79.9858)^T$	$(100.015, -50.0834, 79.9858)^T$	$(100.015, -50.0834, 79.9858)^T$	$(\mathbf{100.015, -50.0834, 79.9858})^{\mathbf{T}}$
2	$(99.8532, -49.9929, -49.9929)^T$	$(99.8532, -49.9929, -49.9929)^T$	$(99.8532, -49.9929, -49.9929)^T$	$(NaN, NaN, NaN)^T$	$(\mathbf{99.8532, -49.9929, -49.9929})^{\mathbf{T}}$
3	$(100.0, 100.0, 100.0)^T$	$(100.0, 100.0, 100.0)^T$	$(100.0, 100.0, 100.0)^T$	$(100.354, 100.354, 100.354)^T$	$(\mathbf{100.0, 100.0, 100.0})^{\mathbf{T}}$
4	$(-59.3406, 69.5444, 39.2757)^T$	$(-59.3406, 69.5444, 39.2757)^T$	$(-59.3406, 69.5444, 39.2757)^T$	$(-59.3406, 69.5444, 39.2757)^T$	$(\mathbf{-59.3406, 69.5444, 39.2757})^{\mathbf{T}}$
5	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(\mathbf{-59.8461, 69.6513, 40.1395})^{\mathbf{T}}$
6	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(-59.8461, 69.6513, 40.1395)^T$	$(\mathbf{-59.8461, 69.6513, 40.1395})^{\mathbf{T}}$
7	$(79.9458, -19.9293, 141.043)^T$	$(79.9458, -19.9293, 141.043)^T$	$(79.9458, -19.9293, 141.043)^T$	$(79.9458, -19.9293, 141.043)^T$	$(\mathbf{79.9458, -19.9293, 141.043})^{\mathbf{T}}$
8	$(91.9475, -20.049, 160.038)^T$	$(91.9475, -20.049, 160.038)^T$	$(91.9475, -20.049, 160.038)^T$	$(91.9475, -20.049, 160.038)^T$	$(\mathbf{91.9475, -20.049, 160.038})^{\mathbf{T}}$
9	$(79.9251, -20.0097, 159.997)^T$	$(79.9251, -20.0097, 159.997)^T$	$(79.9251, -20.0097, 159.997)^T$	$(79.9251, -20.0097, 159.997)^T$	$(\mathbf{79.9251, -20.0097, 159.997})^{\mathbf{T}}$

TABLE IV
LOSS FUNCTION VALUE $\mathcal{L}$ IN (1)

Case	SVD	EIG	EIG Analytical	SVD Analytical	Proposed FS3R
1	0.01440	0.01440	0.01440	0.01440	**0.01440**
2	0.24102	0.24102	0.24102	NaN	**0.24102**
3	0.00222	0.00222	0.00222	0.00489	**0.00222**
4	284.54905	284.54905	284.54905	284.54905	**284.54905**
5	302.03084	302.03084	302.03084	302.03084	**302.03084**
6	298.76512	298.76512	298.76512	298.76512	**298.76512**
7	966940.84856	966940.84856	966940.84856	966940.84856	**966940.84856**
8	977583.31035	977583.31035	977583.31035	977583.31035	**977583.31035**
9	0.03313	0.03313	0.03313	0.03313	**0.03313**

internal functions, while "EIG Analytical" is from [31] and "SVD Analytical" refers to [32].

From the computed results, one can immediately observe from cases 1–3 that the robustness of the proposed FS3R maintains the same level with "SVD," "EIG," and "EIG Analytical." While in all these statistics, "SVD Analytical" is the most weak one due to its low immunity to matrix rank deficiency. In the computation procedure, some steps break according to numerical problems and thus generate NaN values. Such a disadvantage is deadly, because once this happens in an embedded computation system, without proper detection, the system is very likely to crash, since these digits are meaningless. The cases $4 \sim 9$ describe the general accuracy of various algorithms. From cases 4 to 6, the number of vectors increases. Then, from Table III, we can see that the estimated result becomes more accurate as the vector number increases. In cases $7 \sim 9$, it is noticed that, according to the Cannikin law, the final estimation results are significantly influenced by the worst measurement axis, and all the algorithms produce the same behaviors in such

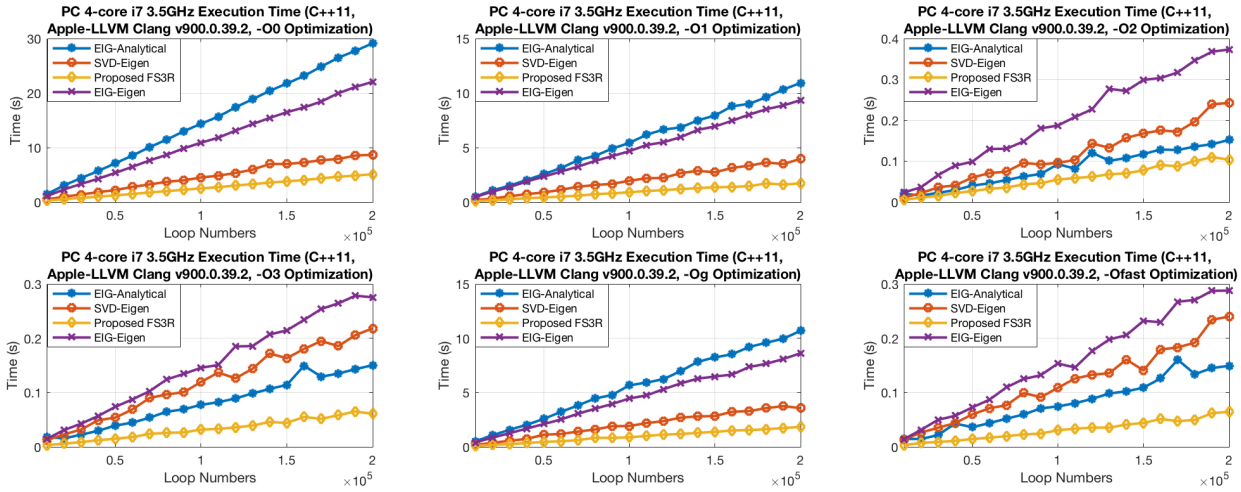


Fig. 1. Computation time comparisons on the PC.

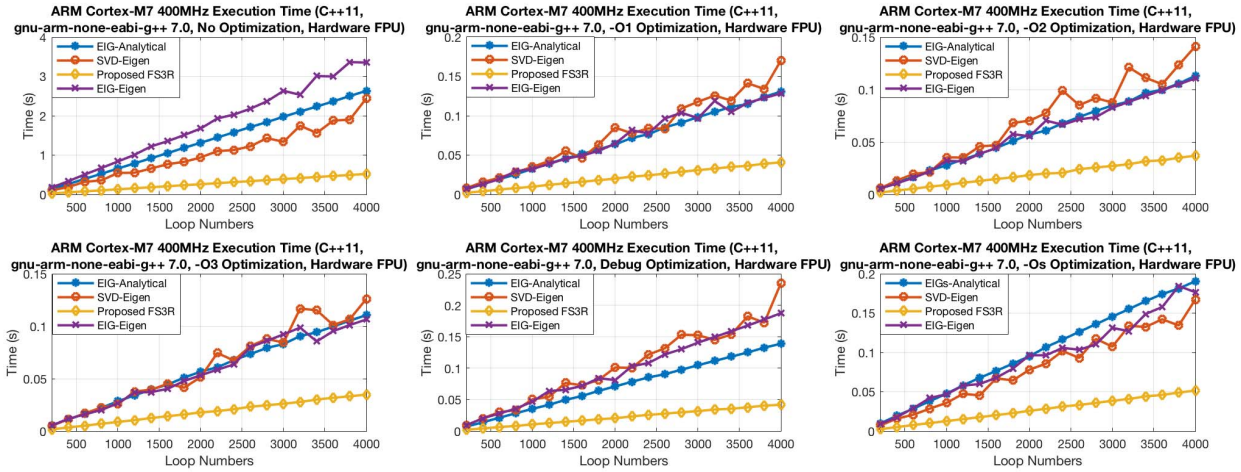


Fig. 2. Computation time comparisons on the ARM-embedded processor.

cases. Therefore, till now, we can draw the conclusion that the proposed FS3R owns the same accuracy and robustness with SVD and EIG.

B. Computation Time

The main superiority of the proposed FS3R is that it owns very simple symbolic computation procedure. It is the main reason that it execute very fast in engineering practice. In this section, we rewrite the algorithms "SVD," "EIG," "EIG Analytical," and FS3R using the C++ programming language. They are tested not only on the PC but also on the ARM-embedded processor as well. The eigenmatrix computation library is used for matrix manipulations and factorizations. The C++11 programming standard is utilized here ensuring feasible eigen implementation.

For different engineering uses, the developer may choose quite different optimization levels for code generation. Commonly, for high-security productions, the optimization level is relatively low, since many optimization options may result in a fatal problem in program execution. Hence, we especially evaluate all the algorithms under various optimization levels. The PC is an x64-based laptop with 4-core i7 3.5-GHz CPU, and the ARM processor is single-chip Cortex-M7 STM32H743VIT6 with the clock speed of 400 MHz and external FPU for fast double/float number computation. For the PC test, each algorithm is run for 10 000 times for averaging execution time. On the ARM processor, as we only have a small RAM area of 1 MByte, each algorithm is averaged every 200 cycles. The computation time performances are depicted in Figs. 1 and 2.

All these algorithms behave with the linear time complexity of $O(n)$, but it is obvious that numerical algorithms using eigenmatrix have evident time variance. The main factor is that the stop conditions of such algorithms are usually uncontrollable. While for analytical or symbolic methods, the computation time is quite deterministic. In all the tests, the proposed FS3R shows definite superiority. The time consumption in general takes 54.43%–87.12% of existing ones, which is a very large advantage that no previous algorithm has reached. The simple procedure of the FS3R saves implementation and compiling time and also decreases the program space. The insurance of the FS3R's accuracy, robustness, and its extremely low computation time makes it a booster in related applications.

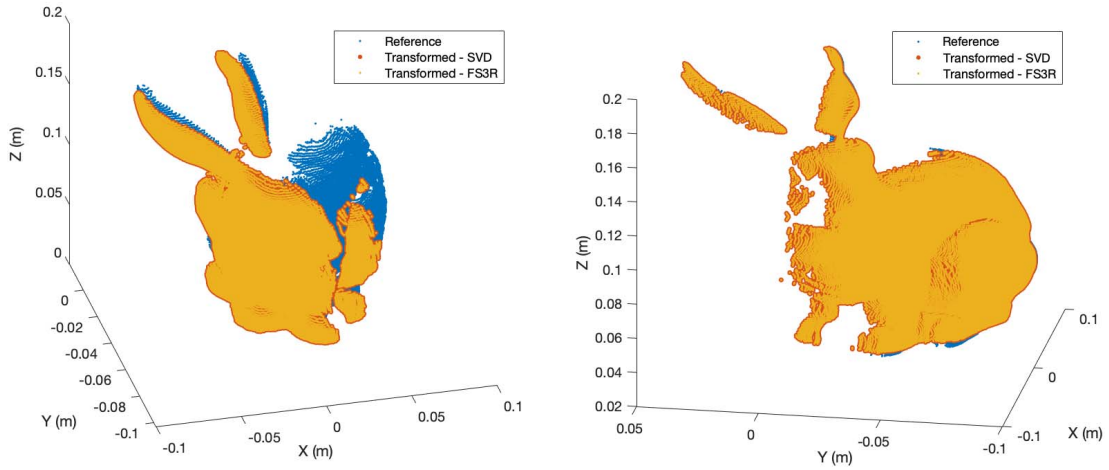


Fig. 3. Registration results using the "bunny" data set from Stanford University [33]. Left: registration from `bun270.ply` to `bun315.ply`. Right: registration from `bun000.ply` to `bun045.ply`.

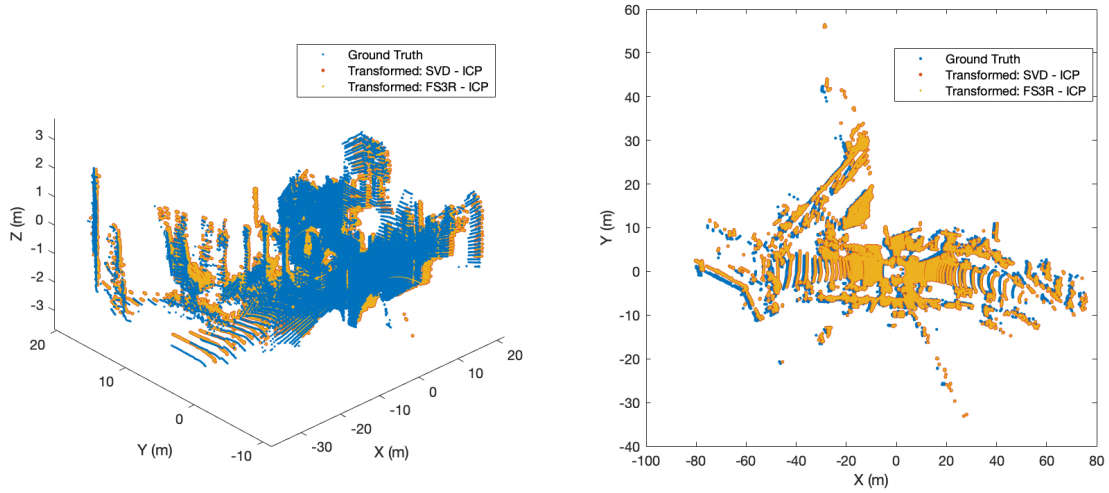


Fig. 4. Transformations using the KITTI data set. Left: 3-D results. Right: 2-D evaluation performance.

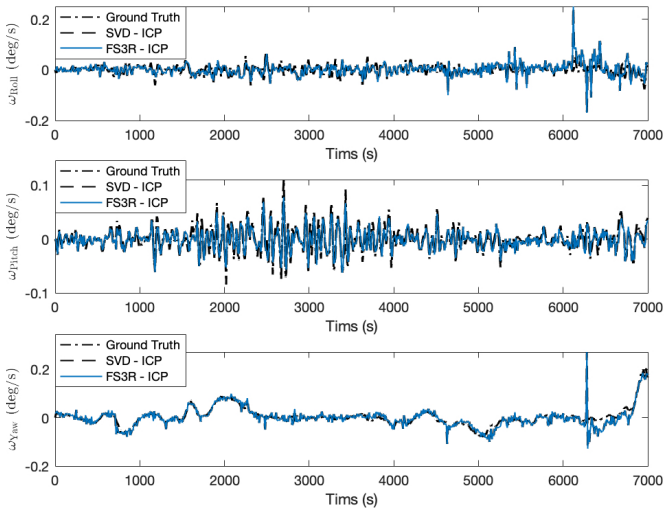


Fig. 5. Estimated angular rates from KITTI data set.

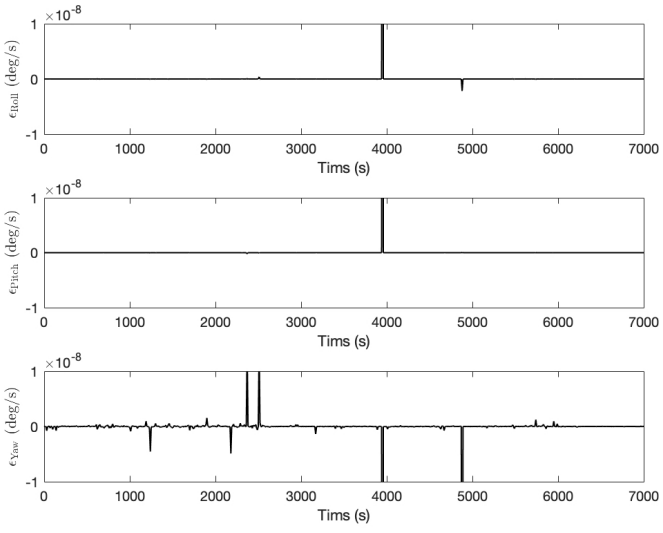


Fig. 6. Angular-rate error difference between SVD and the proposed FS3R.

C. Evaluation With Open Data Sets

In this section, the proposed FS3R is introduced for open-data set evaluation. We pick up two categories of data sets, i.e., the "bunny" data set from Stanford University [33] and the KITTI data set from the Karlsruhe Institute of Technology and the Toyota Technological Institute at Chicago [34]. Both the data sets are widely compared in the existing literature [35], [36]. The "bunny" data set contains multiple-view point-cloud

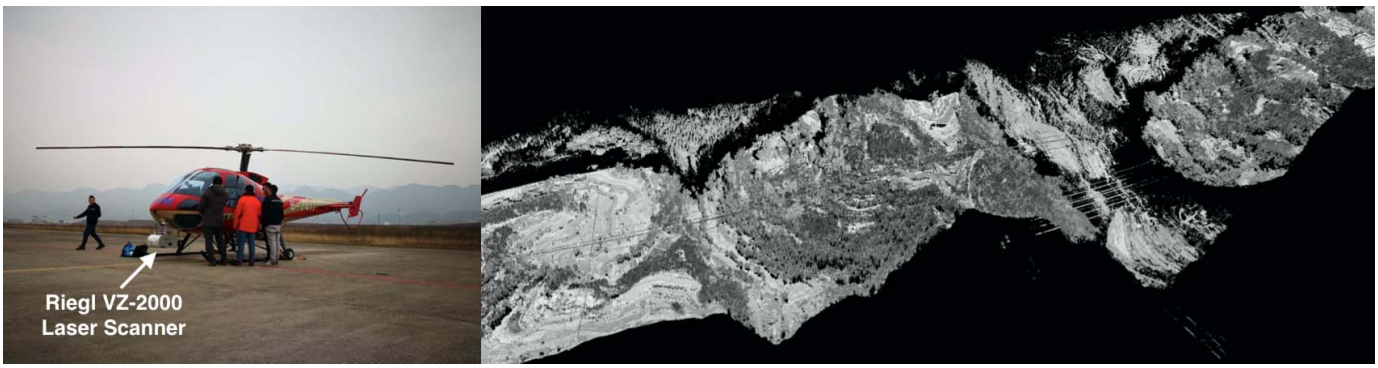


Fig. 7. Left: point-cloud capture system on the helicopter. Right: matching results for power-line inspection using the combination of the proposed FS3R and `libpointmatcher` from ETHZ ASL Lab.

scans of a decorated rabbit model. We use two pairs of correspondences in the "bunny" data set to conduct 3-D reconstruction using ICP algorithms comprising SVD and proposed FS3R, respectively (see Fig. 3). The matching part is implemented using the k-d tree. After 30 iterations, SVD and FS3R converge to $\mathcal{L}_{\mathrm{SVD}} = 0.003092129178551$ and $\mathcal{L}_{\mathrm{FS3R}} = 0.003092129178549$, respectively. The difference is so tiny that can be ignored in the reconstruction of the model. What needs to be pointed out here is that FS3R only takes 0.56738 s in computation, which is much faster than that 1.13026 s from SVD.

For the KITTI data set, the source data folder `2011_09_26_drive_0014_sync` is selected. The KITTI data set has a high-precision ground truth system supported by Velodyne 3-D laser scanners, high-end inertial measurement units (IMUs), global positioning system (GPS) receivers, and high-resolution color stereo image captures. With point-cloud measurements from the Velodyne laser scanners, the transformation sequence is restored using SVD and the proposed FS3R along with the ICP (see Fig. 4). The Euler angles φ, ϑ, and ψ are converted from rotation matrices from the transformation sequence. With this Euler-angle sequence, the angular rates in three directions, roll, pitch, and yaw, i.e., ω_{Roll}, ω_{Pitch}, and ω_{Yaw} are reconstructed using the tagged timestamps (see Fig. 5).

The estimated angular rates from SVD and FS3R are generally the same. Let us define the "Axis" difference by

$$\epsilon_{\mathrm{Axis}} = |\omega_{\mathrm{Axis,SVD}} - \omega_{\mathrm{Axis,True}}| - |\omega_{\mathrm{Axis,FS3R}} - \omega_{\mathrm{Axis,True}}| \tag{27}$$

where $\mathrm{Axis} \in \{\mathrm{Roll}, \mathrm{Pitch}, \mathrm{Yaw}\}$, and $\omega_{\mathrm{Axis,True}}$ denotes the true angular rate in the direction of Axis obtained from the ground-truth data. We can observe from Fig. 6 that the difference between SVD and proposed FS3R has been enlarged. Here, $\epsilon_{\mathrm{Axis}} > 0$ reflects better performance of FS3R and vice versa. In Fig. 6, there are more positive peaks than negative ones, which indicates that here FS3R is slightly more accurate than SVD. Note that the scale of such an error is in fact enlarged by a mass matching process inside the ICP. Here, the root mean-squared (rms) statistics are summarized in Table V.

TABLE V
RMS RESULTS FOR ANGULAR RATE ESTIMATION FROM KITTI DATA SET

	SVD	Proposed FS3R
ϵ_{Roll}	2.0979×10^{-08} deg/s	1.9883×10^{-08} deg/s
$\epsilon_{\mathrm{Pitch}}$	1.9432×10^{-08} deg/s	1.8075×10^{-08} deg/s
ϵ_{Yaw}	3.1302×10^{-08} deg/s	2.9963×10^{-08} deg/s

The shown errors from FS3R are slightly smaller than that from SVD. FS3R can analytically compute the eigenvalue without iterations and thus will not be influenced by numerical thresholds. Therefore, the ICP from FS3R may be more applicable on low-configuration platforms. However, we need to point out that here the error scales in Table V can almost be ignored as they have nearly reached the level of nano deg/s, which is better than most of expensive fiber-optic gyroscopes. Such requirements may seldom occur in real engineering practices. Thus, here the SVD and proposed FS3R can be regarded to own identical accuracy and robustness.

D. Application Notes

The FS3R, since its invention, has been applied to some time-consuming tasks, e.g., point-cloud registration and video stitching. In a recent test, a huge point cloud containing 7 889 456 points is captured using the Riegl VZ-2000 3-D laser scanner on a real-world helicopter (see Fig. 7) for power-line inspection. The original registration method is motivated by the `libpointmatcher` from the ETHZ ASL Lab [26] in which the ICP is completed using the SVD. By replacing SVD with the proposed FS3R, the matching time has been decreased from 1 to 0.65 h. The general results of the point matching are also shown in Fig. 7. The SVD and FS3R in real engineering use have been verified to own the same accuracy and robustness. The huge amount of point correspondences ensures the precision estimates of rigid-body transformation. This shows the potential applicability of FS3R in industrial processing. We also have made all the codes in C++ and MATLAB open source, and the audience can verify its effectiveness (see the footnote of the first page).

IV. Conclusion

Our recent algorithm FLAE is revisited, which is later related to the 3-D registration problem. Some proofs are presented to show the equivalence. The previous solution to the quartic equation is then simplified getting rid of complex numbers for easier implementation. The numerical robustness of the proposed method is also investigated showing its immunity to degenerated matrices. The proposed algorithm is systematically evaluated with other representatives. The results indicate that it maintains the accuracy and robustness but consumes much less computation time. Real applications, including large-point-cloud registration, have shown its superiority in engineering processing. However, it is still noticed that the current method highly relies on the floating-point operations. Unless we have reached the limit of the EIG numerical algorithm, we still have an expectation to develop the next-generation algorithm in which the floating number is no longer needed, which would be of great benefit for parallel-computing platforms, such as FPGA and GPU, for accelerated performance.

Acknowledgment

The authors would like to thank Dr. F. L. Markley from the NASA Goddard Space Flight Center and Prof. Y. Wu from Shanghai Jiao Tong University for their constructive comments to this article. They would like to thank Dr. Y. Yang from the Nuclear Research Centre, U.S. Navy, for providing his codes from [31]. They would also like to genuinely thank Chongqing Fengmai Innovation Inc., Chongqing, China, for the help and support through the experimental devices, e.g., laser scanner and helicopter. The source code of this article has been uploaded to https://github.com/zarathustr/FS3R (C++), https://github.com/zarathustr/FS3R-MATLAB (MATLAB), and https://github.com/zarathustr/FS3R-CrossWorks (Embedded).

References

[1] F. Aghili and C.-Y. Su, "Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Mar. 2016.

[2] L. Wang, M. Liu, and M. Q.-H. Meng, "Real-time multisensor data retrieval for cloud robotic systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 507–518, Apr. 2015.

[3] C. Ye, S. Hong, and A. Tamjidi, "6-DOF pose estimation of a robotic navigation aid by tracking visual and geometric features," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1169–1180, Oct. 2015.

[4] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[5] C. Ye and J. Borenstein, "A novel filter for terrain mapping with laser rangefinders," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 913–921, Oct. 2004.

[6] T. L. Faber and E. M. Stokely, "Orientation of 3-D structures in medical images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-10, no. 5, pp. 626–633, Sep. 1988.

[7] S. Ying, J. Peng, S. Du, and H. Qiao, "A scale stretch method based on ICP for 3D data registration," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 3, pp. 559–565, Jul. 2009.

[8] Z. Yang and S. Shen, "Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, Jan. 2016.

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

[10] Y. He, B. Liang, J. Yang, S. Li, and J. He, "An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features," *Sensors*, vol. 17, no. 8, p. 1862, 2017.

[11] Z. Zhou, Y. Li, J. Liu, and G. Li, "Equality constrained robust measurement fusion for adaptive Kalman-filter-based heterogeneous multisensor navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2146–2157, Oct. 2013.

[12] Z. Zhou, Y. Li, J. Zhang, and C. Rizos, "Integrated navigation system for a low-cost quadrotor aerial vehicle in the presence of rotor influences," *J. Surv. Eng.*, vol. 143, no. 1, 2017, Art. no. 05016006.

[13] K. Kanatani, "Analysis of 3-D rotation fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 543–549, May 1994.

[14] G. Chang, "Total least-squares formulation of Wahba's problem," *Electron. Lett.*, vol. 51, no. 17, pp. 1334–1335, 2015.

[15] 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. 734–763, 2014.

[16] 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.

[17] V. Govindu and C. Shekhar, "Alignment using distributions of local geometric properties," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1031–1043, Oct. 1999.

[18] O. Choi and I. S. Kweon, "Robust feature point matching by preserving local geometric consistency," *Comput. Vis. Image Understand.*, vol. 113, no. 6, pp. 726–742, Jun. 2009.

[19] C. F. Olson, "A general method for geometric feature matching and model extraction," *Int. J. Comput. Vis.*, vol. 45, no. 1, pp. 39–54, 2001.

[20] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *Auton. Robots*, vol. 41, no. 1, pp. 31–43, Jan. 2017.

[21] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[22] 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.

[23] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.

[24] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP, Image Understand.*, vol. 54, no. 3, pp. 358–367, Nov. 1991.

[25] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.

[26] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[27] H. Guo, H. Chen, F. Xu, F. Wang, and G. Lu, "Implementation of EKF for vehicle velocities estimation on FPGA," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3823–3835, Sep. 2013.

[28] J. Wu, Z. Zhou, B. Gao, R. Li, Y. Cheng, and H. Fourati, "Fast linear quaternion attitude estimator using vector observations," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 307–319, Jan. 2018.

[29] J. Wu, Z. Zhou, H. Fourati, R. Li, and M. Liu, "Generalized linear quaternion complementary filter for attitude estimation from multisensor observations: An optimization approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1330–1343, Jul. 2019.

[30] Y. Cheng and M. D. Shuster, "Improvement to the implementation of the QUEST algorithm," *J. Guid., Control, Dyn.*, vol. 37, no. 1, pp. 301–305, 2014.

[31] Y. Yang and Z. Zhou, "An analytic solution to Wahba's problem," *Aerosp. Sci. Technol.*, vol. 30, no. 1, pp. 46–49, 2013.

[32] Z. Liu, W. Liu, X. Gong, and J. Wu, "Optimal attitude determination from vector sensors using fast analytical singular value decomposition," *J. Sensors*, vol. 2018, Jun. 2018, Art. no. 6308530.

[33] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 313–324.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. IEEE CVPR*, Jun. 2012, pp. 3354–3361.

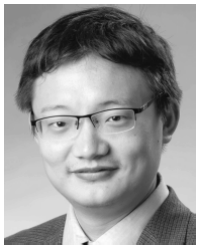
[35] T.-H. Kwok, "DNSS: Dual-normal-space sampling for 3-D ICP registration," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 241–252, Jan. 2018.

[36] H. Wang, R. Jiang, H. Zhang, and S. S. Ge, "Heading reference-assisted pose estimation for ground vehicles," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 448–558, Jan. 2018.

Jin Wu (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.

Since 2018, he has been a Research Assistant with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. He has been in the UAV industry since 2012 and has launched two companies ever since. He has co-authored over 30 technical articles in representative journals and conference proceedings of the IEEE, the American Institute of Aeronautics and Astronautics, the Institution of Engineering and Technology, etc. His 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 ISI Web of Science during 2017 to 2018.



Ming Liu (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–Nuremberg University, Erlangen, Germany, and the Fraunhofer Institute IISB, Erlangen, as a Master's Visiting Scholar.

He is currently an Assistant Professor with the Electronic and Computer Engineering Department and the Computer Science and Engineering Department, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong. He is also a Founding Member of Shanghai Swing Automation Ltd., Co., China. He is coordinating and involved in the National Science Foundation Projects and National 863-Hi-TechPlan Projects in China. He has published many popular articles 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 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 second place prize of the European Micro Aerial Vehicle Competition (EMAV'09), two awards from the International Aerial Robot Competition (IARC'14) as a Team Member, the Best Student Paper Award as the first author at the IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2012), the Best Paper Award in Information at the IEEE International Conference on Information and Automation (ICIA 2013) as the first author, the Best Paper Award Finalists as a coauthor, the Best RoboCup Paper Award at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), the Best Conference Paper Award at the IEEE-CYBER 2015, the Best Student Paper Finalist at the IEEE International Conference on Real-Time Computing and Robotics (RCAR 2015), the Best Student Paper Finalist at ROBIO 2015, the Best Student Paper Award at the IEEE-ICAR 2017, the Best Paper in Automation Award at 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 IEEERCAR 2016 and the Program Chair of the International Robotics Conference in Foshan 2017. He was the Conference Chair of ICVS 2017. He is currently an Associate Editor of the IEEE ROBOTICS AND AUTOMATION LETTERS.

Zebo Zhou was born in Yongchuan, Chongqing, China, in 1982. He received the B.Sc. and M.Sc. degrees from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2004 and 2006, respectively, and the Ph.D. degree from the College of Surveying and Geoinformatics, Tongji University, Shanghai, China, in 2009.

He was a Visiting Fellow with the Surveying and Geospatial Engineering Group, School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW, Australia, in 2009 and 2015, respectively. He is currently an Associate Professor with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China. He has been in charge of projects of the National Natural Science Foundation of China and has taken part in the National 863 High-tech Founding of China. His research interests include GNSS navigation and positioning, GNSS/INS integrated navigation, and multi-sensor fusion.

Dr. Zhou served as a Guest Editor for several special issues published in the *International Journal of Distributed Sensor Networks* and the *Asian Journal of Control*. He has been presenting related works at the Annual Conference of the Institute of Navigation (ION) and the Annual Chinese Satellite Navigation Conference (CSNC) for several times. He has received the best paper awards in these conferences.

Rui Li (M'12) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2008.

She worked as a Visiting Research Associate with the Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, and the Department of Mathematics and Statistics, Curtin University of Technology, Bentley, WA, Australia. From September 2011 to September 2012, she was a Visiting Scholar with the Department of Electrical Engineering, University of California at Riverside, Riverside, CA, USA. She joined the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2008, where she is currently an Associate Professor with the School of Automation. Her research interests include optimization theory and optimal control, nonlinear control, multi-agent systems, and aircraft control.