Received 5 Oct 2021; | Revised 18 Feb 2022; | Accepted ???

DOI: xxx/xxxx

RESEARCH ARTICLE

On closed-form solutions to the 4D nearest rotation matrix problem

S. Sarabandi | F. Thomas*

Institut de Robòtica i Informàtica Industrial (CSIC-UPC), ETSEIB, Diagonal 647, Pavelló E, planta 1, 08028 Barcelona, Spain

Correspondence
*F. Thomas, Email: f.thomas@csic.es

Summary

In this paper, we address the problem of restoring the orthogonality of a numerically noisy 4D rotation matrix by finding its nearest (in Frobenius norm) correct rotation matrix. This problem can be straightforwardly solved using the Singular Value Decomposition (SVD). Nevertheless, to avoid numerical methods, we present two new closed-form methods. One relies on the direct minimization of the mentioned Frobenius norm, and the other on the passage to double quaternion representation. A comparison of these two methods with respect to the SVD reveals that the method based on a double quaternion representation is superior in all aspects.

KEYWORDS:
4D rotations, double quaternions, fourth-degree polynomials

Recently, it has been found that some problems arising in robotics, computer vision, and computer graphics —such as point-cloud registration[1] and hand-eye calibration problems[2]— can be formulated in terms of 4D rotation matrices. Unfortunately, experimental data leads to noisy 4D rotation matrices whose orthonormality must be restored.

Orthonormalizing a noisy rotation matrix, say $\mathbf{R}$, usually consists in finding the nearest proper orthonormal matrix in Frobenius norm, say $\hat{\mathbf{R}}$, to it. This can be formulated as the minimization of

$$\left\|\mathbf{R}-\hat{\mathbf{R}}\right\|_F^2 = \operatorname{Tr}\left[(\mathbf{R}-\hat{\mathbf{R}})^T(\mathbf{R}-\hat{\mathbf{R}})\right] = \operatorname{Tr}\left(\mathbf{R}^T\mathbf{R}-2\hat{\mathbf{R}}^T\mathbf{R}+\mathbf{I}\right), \quad (1)$$

which is equivalent to maximizing

$$\operatorname{Tr}\left(\hat{\mathbf{R}}^T\mathbf{R}\right), \quad (2)$$

subject to the orthogonality constraint of $\hat{\mathbf{R}}$. We can deal with this constraint by introducing a symmetric Lagrangian multiplier matrix $\mathbf{\Lambda}$[3] and looking for stationary values of

$$\epsilon(\hat{\mathbf{R}},\mathbf{\Lambda}) = \operatorname{Tr}\left(\hat{\mathbf{R}}^T\mathbf{R}\right) - \operatorname{Tr}\left[\mathbf{\Lambda}(\hat{\mathbf{R}}^T\hat{\mathbf{R}}-\mathbf{I})\right]. \quad (3)$$

Since it is possible to prove that[4,5]

$$\frac{d}{d\hat{\mathbf{R}}}\operatorname{Tr}\left(\hat{\mathbf{R}}^T\mathbf{R}\right) = \mathbf{R} \quad (4)$$

and

$$\frac{d}{d\hat{\mathbf{R}}}\operatorname{Tr}\left(\mathbf{\Lambda}\hat{\mathbf{R}}^T\hat{\mathbf{R}}\right) = \hat{\mathbf{R}}\left(\mathbf{\Lambda}+\mathbf{\Lambda}^T\right), \quad (5)$$

the differentiation of $\epsilon(\hat{\mathbf{R}},\mathbf{\Lambda})$ with respect to $\hat{\mathbf{R}}$ yields the condition

$$\mathbf{R}-\hat{\mathbf{R}}(\mathbf{\Lambda}+\mathbf{\Lambda}^T) = 0. \quad (6)$$

[1]This work was partially supported by the Spanish Government through project PID2020-117509GB-I00/AEI/10.13039/501100011033. This work does not have any conflicts of interest.

Then, since $\mathbf{\Lambda}^T = \mathbf{\Lambda}$, we have that

$$\mathbf{R} = 2\hat{\mathbf{R}}\mathbf{\Lambda}, \tag{7}$$

which is a useful decomposition of $\mathbf{R}$ into the product of an orthonormal and a symmetric matrix [6]. Now,

$$\mathbf{R}^T\mathbf{R} = 4\mathbf{\Lambda}\hat{\mathbf{R}}^T\hat{\mathbf{R}}\mathbf{\Lambda} = 4\mathbf{\Lambda}^2. \tag{8}$$

Hence,

$$2\mathbf{\Lambda} = (\mathbf{R}^T\mathbf{R})^{\frac{1}{2}}. \tag{9}$$

Finally,

$$\hat{\mathbf{R}} = \mathbf{R}(2\mathbf{\Lambda})^{-1} = \mathbf{R}(\mathbf{R}^T\mathbf{R})^{-\frac{1}{2}} \tag{10}$$

Therefore, the nearest 4D rotation matrix problem boils down to calculating the square root of $\mathbf{A} = \mathbf{R}\mathbf{R}^T$. As we will see later, this calculation can be avoided by computing the Singular Value Decomposition (SVD) of $\mathbf{R}$. Nevertheless, closed-form solutions are preferable because they involve a constant number of operations, and the contribution of each variable to the solution can be studied analytically. This is why we present in this paper two new closed-form methods as an alternative to the use of the SVD. They can be seen as generalizations of the diagonalization and the quaternion methods available for the 3D case [7].

This paper is structured as follows. In Section 1, it is shown how the nearest 4D rotation matrix problem can be solved using the SVD. In Sections 2 and 3, the two new closed-form methods are presented. Section 4 compares the performance of these two new methods with respect to the method based on the SVD. Finally, Section 5 summarizes the main results.

1 | STANDARD SVD METHOD

Let us assume that the SVD of $\mathbf{R}$ is given by

$$\mathbf{R} = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T, \tag{11}$$

and that $\hat{\mathbf{R}}$ is the orthogonal matrix that minimizes $\left\|\mathbf{R} - \hat{\mathbf{R}}\right\|_F^2$. Then, we have that

$$\left\|\mathbf{R} - \hat{\mathbf{R}}\right\|_F^2 = \left\|\mathbf{U}\mathbf{\Delta}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\hat{\mathbf{R}}\mathbf{V}\mathbf{V}^T\right\|_F^2 = \left\|\mathbf{\Delta} - \widetilde{\mathbf{R}}\right\|_F^2. \tag{12}$$

where $\widetilde{\mathbf{R}} = \mathbf{U}^T\hat{\mathbf{R}}V$ is another orthogonal matrix. Now, observe that minimizing (12) is equivalent to minimizing $\left\|\widetilde{\mathbf{R}}^T\mathbf{\Delta} - \mathbf{I}\right\|_F^2$, which in turn is equivalent to maximizing $\mathrm{Tr}(\mathbf{\Delta}\widetilde{\mathbf{R}})$, which is maximized for $\widetilde{\mathbf{R}} = \mathbf{I}$. Thus, if $\det(\mathbf{R}) = +1$, the optimal rotation matrix is

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T. \tag{13}$$

When working with very noisy systems, it could happen that $\det(\mathbf{R}) = -1$. In this case, it can be shown that the optimal rotation matrix is given by [8,9]

$$\hat{\mathbf{R}} = \mathbf{U}\ \mathrm{diag}\,(1, 1, 1, -1)\ \mathbf{V}^T. \tag{14}$$

2 | CLOSED-FORM SQUARE ROOT MATRIX METHOD

We have seen in the introduction how using Lagrange multipliers, it was proved that the orthogonal matrix $\hat{\mathbf{R}}$ that minimizes $\left\|\mathbf{R} - \hat{\mathbf{R}}\right\|_F^2$ is given by (10). Since $\mathbf{A} = \mathbf{R}^T\mathbf{R}$ is symmetric non-negative definitive, it has non-negative real eigenvalues. The square root of matrix $\mathbf{A}$ can thus be computed by applying Cayley-Hamilton theorem to the characteristic polynomial of $\mathbf{A}^{\frac{1}{2}}$. That is,

$$\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_1}\mathbf{I}\right)\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_2}\mathbf{I}\right)\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_3}\mathbf{I}\right)\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_4}\mathbf{I}\right) = \mathbf{A}^2 - a_3\mathbf{A}^{\frac{3}{2}} + a_2\mathbf{A} - a_1\mathbf{A}^{\frac{1}{2}} + a_0\mathbf{I} = 0, \tag{15}$$

where

$$\begin{aligned}
a_3 &= \sqrt{\lambda_1} + \sqrt{\lambda_2} + \sqrt{\lambda_3} + \sqrt{\lambda_4},\\
a_2 &= \sqrt{\lambda_1\lambda_2} + \sqrt{\lambda_1\lambda_3} + \sqrt{\lambda_1\lambda_4} + \sqrt{\lambda_2\lambda_3} + \sqrt{\lambda_2\lambda_4} + \sqrt{\lambda_3\lambda_4},\\
a_1 &= \sqrt{\lambda_1\lambda_2\lambda_3} + \sqrt{\lambda_1\lambda_2\lambda_4} + \sqrt{\lambda_1\lambda_3\lambda_4} + \sqrt{\lambda_2\lambda_3\lambda_4},\\
a_0 &= \sqrt{\lambda_1\lambda_2\lambda_3\lambda_4}.
\end{aligned} \tag{16}$$

Now, if we multiply (15) by $\mathbf{A}^{\frac{1}{2}}$ and we substitute $\mathbf{A}^{\frac{3}{2}}$ from (15) in the result, we obtain

$$\mathbf{A}^{\frac{1}{2}} = \frac{1}{-a_0 + \frac{a_1 a_2}{a_3} - \frac{a_1^2}{a_3^2}} \cdot \left[\frac{1}{a_3}\mathbf{R}^3 + \left(-a_3 - \frac{a_1}{a_3^2} + \frac{2a_2}{a_3} \right)\mathbf{R}^2 + \left(\frac{a_2^2}{a_3} + \frac{a_0}{a_3} - \frac{a_1 a_2}{a_3^2} - a_1 \right)\mathbf{R} + \left(-\frac{a_0 a_1}{a_3^2} + \frac{a_0 a_2}{a_3} \right)\mathbf{I} \right]. \quad (17)$$

Therefore, we have to find the roots of the characteristic polynomial of $\mathbf{A}$ to calculate the square root of $\mathbf{A}$. That is, the roots of

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \lambda^4 + p\lambda^3 + q\lambda^2 + r\lambda + s, \quad (18)$$

where

$$\begin{aligned} p &= -\mathrm{Tr}(\mathbf{A}), \\ q &= \frac{1}{2}\left[p^2 - \mathrm{Tr}(\mathbf{A}^2)\right], \\ r &= \frac{1}{6}p^3 - \frac{1}{2}\left[p\mathrm{Tr}(\mathbf{A}^2)\right] - \frac{1}{3}\mathrm{Tr}(\mathbf{A}^3), \\ s &= \det(\mathbf{A}). \end{aligned}$$

These coefficients can also be computed using the recursive Faddeev–LeVerrier method [10], but in our implementation we have directly use the above explicit formulas.

Many closed-form methods have been proposed for solving quartic equations like the one in (18), but they have been designed with aims of elegance, generality or simplicity rather than error minimization or computational cost. All methods require the solution of a subsidiary cubic equation (known as "cubic resolvent"). In [11], five different resolvents are examined from the computational point of view to conclude that the one presented in [12] is numerically the most stable and less subject to round-off errors. The later methods presented in [13] and [14] provide further improvements. Nevertheless, since in our case the problem is well-conditioned because the four roots of (18) are known to be real and positive, we have adopted a simple variation on Ferrari's method. First, we define

$$\begin{aligned} a &= \frac{-q^2}{3} - 4s + pr, \\ b &= -sp^2 + \frac{pqr}{3} - r^2 + \frac{8qs}{3} - \frac{2q^3}{27}. \end{aligned}$$

From which, we obtain

$$y = \sqrt[3]{-\frac{b}{2} + \sqrt{\left(\frac{b}{2}\right)^2 + \left(\frac{a}{3}\right)^3}} + \sqrt[3]{-\frac{b}{2} - \sqrt{\left(\frac{b}{2}\right)^2 + \left(\frac{a}{3}\right)^3}} + \frac{q}{3} \quad (19)$$

and

$$D = \begin{cases} \sqrt{\frac{3}{4}p^2 - 2q + 2\sqrt{y^2 - 4s}}, & \text{if } C = 0, \\ \sqrt{\frac{3}{4}p^2 - C^2 - 2q + \frac{1}{4C}(4pq - 8r - p^3)}, & \text{otherwise.} \end{cases} \quad (20)$$

$$E = \begin{cases} \sqrt{\frac{3}{4}p^2 - 2q - 2\sqrt{y^2 - 4s}}, & \text{if } C = 0, \\ \sqrt{\frac{3}{4}p^2 - C^2 - 2q - \frac{1}{4C}(4pq - 8r - p^3)}, & \text{otherwise.} \end{cases} \quad (21)$$

where

$$C = \sqrt{\frac{1}{4}p^2 - q + y}. \quad (22)$$

Subsequently, we have that the sought roots in decreasing order are:

$$\lambda_1 = -\frac{p}{4} + \frac{1}{2}(C + D), \quad (23)$$

$$\lambda_2 = -\frac{p}{4} + \frac{1}{2}(C - D), \quad (24)$$

$$\lambda_3 = -\frac{p}{4} - \frac{1}{2}(C - E), \quad (25)$$

$$\lambda_4 = -\frac{p}{4} - \frac{1}{2}(C + E). \quad (26)$$

Then, $\mathbf{A}^{\frac{1}{2}}$ can be calculated by substituting the above values in (16), and substituting the result in (17). Finally, $\hat{\mathbf{R}}$ is obtained using (10).

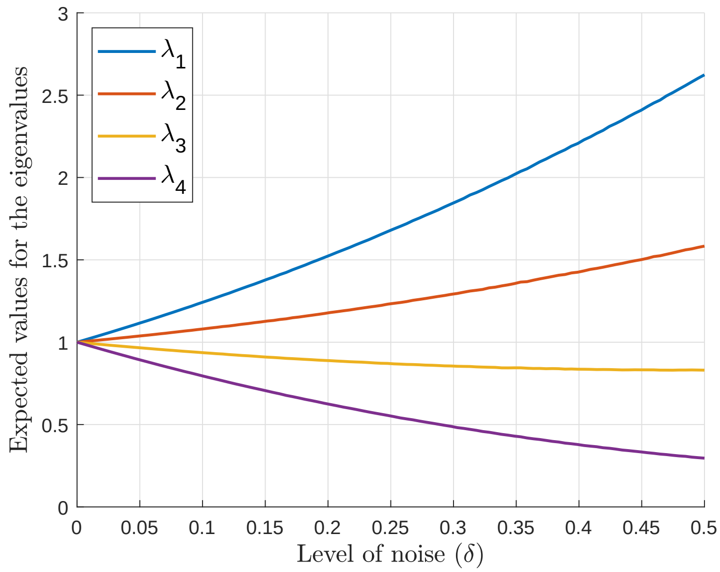

FIGURE 1 Expected values for the eigenvalues of $\mathbf{A} = \mathbf{R}\mathbf{R}^T$ as a function of the added noise

To see the effect of noise on the eigenvalues of $\mathbf{A}$, we can compute the eigenvalues of a noiseless rotation matrix whose elements are contaminated with zero-mean additive noise normally distributed with standard deviation equal to δ. If this operation, is repeated 10^4 times for values of δ ranging from 0 to 0.5 and the average of the obtained eigenvalues is computed, we obtain the plot shown in Fig. 1 .

Clearly, the four eigenvalues of $\mathbf{A}$ are equal to one for noiseless rotation matrices. Then, for low levels of noise, we have that $a_3 \approx 4$, $a_2 \approx 6$, $a_1 \approx 4$, and $a_0 \approx 1$. As a consequence,

$$\hat{\mathbf{R}} \approx \mathbf{R}\left(\frac{1}{16}\mathbf{R}^3 - \frac{5}{16}\mathbf{R}^2 + \frac{15}{16}\mathbf{R} + \frac{5}{16}\mathbf{I}\right)^{-1}, \tag{27}$$

which can be used to calculate the 4D nearest rotation matrix for low levels of noise. The three-dimensional counterpart to this formula can be found in[7]. Since this asymptotic expansion is valid for levels of noise tending to zero, $\mathbf{R}\mathbf{R}^T$ tends to the identity, and so tends $(\mathbf{R}\mathbf{R}^T)^{\frac{1}{2}}$. Therefore, the required inversion is well-conditioned.

3 | CLOSED-FORM DOUBLE QUATERNION METHOD

While a rotation in 3D is defined by an axis of rotation and angle rotated about it, a rotation in four dimensions is defined by two orthogonal planes and two angles of rotation (see[15,16] for details on the geometric interpretation of rotations in four dimensions). A 4D rotation can actually be seen as the commutative composition of two rotations in a pair of orthogonal two-dimensional subspaces. It is thus said that every 4D rotation matrix can be decomposed into the commutative product of a right- and a left-isoclinic rotation matrix as[17]:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{pmatrix} = \begin{pmatrix} l_0 & -l_3 & l_2 & -l_1 \\ l_3 & l_0 & -l_1 & -l_2 \\ -l_2 & l_1 & l_0 & -l_3 \\ l_1 & l_2 & l_3 & l_0 \end{pmatrix} \begin{pmatrix} r_0 & -r_3 & r_2 & r_1 \\ r_3 & r_0 & -r_1 & r_2 \\ -r_2 & r_1 & r_0 & r_3 \\ -r_1 & -r_2 & -r_3 & r_0 \end{pmatrix}, \tag{28}$$

where $\mathbf{l} = (l_0, l_1, l_2, l_3)^T$ and $\mathbf{r} = (r_0, r_1, r_2, r_3)^T$ can be interpreted as unit commutative quaternions in two different bases[18]. Therefore, every 4D rotation can be represented by a double quaternion $\left(\mathbf{l}^T, \mathbf{r}^T\right)$. The reader is referred to[19,20] for a formulation of this factorization in the language of geometric algebra, where it is shown how it boils down to factorizing a four-dimensional bivector into the sum of two simple orthogonal bivectors.

It can be checked that (28) can be rewritten as[21,22]

$$\underbrace{\frac{1}{4}\begin{pmatrix} r_{11}+r_{22}+r_{33}+r_{44} & -r_{41}+r_{32}-r_{23}+r_{14} & -r_{31}-r_{42}+r_{13}+r_{24} & r_{21}-r_{12}-r_{43}+r_{34} \\ r_{41}+r_{32}-r_{23}-r_{14} & r_{11}-r_{22}-r_{33}+r_{44} & r_{21}+r_{12}+r_{43}+r_{34} & r_{31}-r_{42}+r_{13}-r_{24} \\ -r_{31}+r_{42}+r_{13}-r_{24} & r_{21}+r_{12}-r_{43}-r_{34} & -r_{11}+r_{22}-r_{33}+r_{44} & r_{41}+r_{32}+r_{23}+r_{14} \\ r_{21}-r_{12}+r_{43}-r_{34} & r_{31}+r_{42}+r_{13}+r_{24} & -r_{41}+r_{32}+r_{23}-r_{14} & -r_{11}-r_{22}+r_{33}+r_{44} \end{pmatrix}}_{\stackrel{\text{def}}{=}\,\mathbf{H}} = \underbrace{\begin{pmatrix} l_0r_0 & l_0r_1 & l_0r_2 & l_0r_3 \\ l_1r_0 & l_1r_1 & l_1r_2 & l_1r_3 \\ l_2r_0 & l_2r_1 & l_2r_2 & l_2r_3 \\ l_3r_0 & l_3r_1 & l_3r_2 & l_3r_3 \end{pmatrix}}_{=\,\mathbf{l}\,\mathbf{r}^T} \tag{29}$$

and that the maximization of (2) can be rewritten as the maximization of

$$\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R}) = \mathrm{Tr}\left[\begin{pmatrix} \hat{r}_0 & -\hat{r}_3 & \hat{r}_2 & \hat{r}_1 \\ \hat{r}_3 & \hat{r}_0 & -\hat{r}_1 & \hat{r}_2 \\ -\hat{r}_2 & \hat{r}_1 & \hat{r}_0 & \hat{r}_3 \\ -\hat{r}_1 & -\hat{r}_2 & -\hat{r}_3 & \hat{r}_0 \end{pmatrix}^T \begin{pmatrix} \hat{l}_0 & -\hat{l}_3 & \hat{l}_2 & -\hat{l}_1 \\ \hat{l}_3 & \hat{l}_0 & -\hat{l}_1 & -\hat{l}_2 \\ -\hat{l}_2 & \hat{l}_1 & \hat{l}_0 & -\hat{l}_3 \\ \hat{l}_1 & \hat{l}_2 & \hat{l}_3 & \hat{l}_0 \end{pmatrix}^T \begin{pmatrix} l_0 & -l_3 & l_2 & -l_1 \\ l_3 & l_0 & -l_1 & -l_2 \\ -l_2 & l_1 & l_0 & -l_3 \\ l_1 & l_2 & l_3 & l_0 \end{pmatrix} \begin{pmatrix} r_0 & -r_3 & r_2 & r_1 \\ r_3 & r_0 & -r_1 & r_2 \\ -r_2 & r_1 & r_0 & r_3 \\ -r_1 & -r_2 & -r_3 & r_0 \end{pmatrix}\right], \tag{30}$$

which, using the commutativity of left- and right-isoclinic rotations, can be rewritten as

$$\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R}) = 4\,(\hat{\mathbf{l}}^T\,\mathbf{l}\,)(\mathbf{r}^T\,\hat{\mathbf{r}}) \tag{31}$$

$$= 4\,\hat{\mathbf{l}}^T\,\mathbf{H}\,\hat{\mathbf{r}} \tag{32}$$

$$= 4\,\hat{\mathbf{r}}^T\,\mathbf{H}^T\,\hat{\mathbf{l}}. \tag{33}$$

If (32) is left-multiplied by $\mathbf{l}$ and the result divided by $\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R})$, we obtain

$$\hat{\mathbf{l}} = \frac{4}{\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R})}\,\mathbf{H}\,\hat{\mathbf{r}}. \tag{34}$$

Likewise, if (33) is left-multiplied by $\mathbf{r}$, we obtain

$$\hat{\mathbf{r}} = \frac{4}{\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R})}\,\mathbf{H}^T\,\hat{\mathbf{l}}. \tag{35}$$

Substituting (35) in (34), left-multiplying by $\mathbf{l}^T$, and rearranging terms, we have that

$$\left[\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R})\right]^2 = 16\,\hat{\mathbf{l}}^T\,\mathbf{H}\,\mathbf{H}^T\,\hat{\mathbf{l}}. \tag{36}$$

Likewise, substituting (34) in (35), left-multiplying by $\mathbf{r}^T$, and rearranging terms, we obtain

$$\left[\mathrm{Tr}(\hat{\mathbf{R}}^T\mathbf{R})\right]^2 = 16\,\hat{\mathbf{r}}^T\,\mathbf{H}^T\,\mathbf{H}\,\hat{\mathbf{r}}. \tag{37}$$

Now, observe that, according to (10), $\hat{\mathbf{R}}^T\mathbf{R} = (\mathbf{R}^T\mathbf{R})^{\frac{1}{2}} = \mathbf{A}^{\frac{1}{2}}$ is a symmetric non-negative definite matrix. As a consequence, since the trace of a matrix is equal to the sum of its eigenvalues, the trace of $\hat{\mathbf{R}}^T\mathbf{R}$ is non-negative. Then, the maximization of (2) can be decomposed into the following two independent optimization problems

$$\max(\hat{\mathbf{l}}^T\,\mathbf{H}\mathbf{H}^T\,\hat{\mathbf{l}}) \text{ subject to } \hat{\mathbf{l}}^T\,\hat{\mathbf{l}} = 1 \tag{38}$$

and

$$\max(\hat{\mathbf{r}}^T\,\mathbf{H}^T\mathbf{H}\,\hat{\mathbf{r}}) \text{ subject to } \hat{\mathbf{r}}^T\,\hat{\mathbf{r}} = 1. \tag{39}$$

This result generalizes the method for the three dimensional case recently presented in[23].

Using Lagrange multipliers, the value of $\hat{\mathbf{l}}$ that solves the optimization problem in (38) is obtained by solving the equation

$$\frac{\partial\,(\mathbf{l}^T\,\mathbf{H}\mathbf{H}^T\,\hat{\mathbf{l}})}{\partial\,\hat{\mathbf{l}}} = \lambda\frac{\partial\,(\hat{\mathbf{l}}^T\hat{\mathbf{l}})}{\partial\,\hat{\mathbf{l}}}. \tag{40}$$

That is,

$$\mathbf{H}\mathbf{H}^T\,\hat{\mathbf{l}} = \lambda\,\hat{\mathbf{l}}. \tag{41}$$

Thus, we have four solution candidates for $\hat{\mathbf{l}}$: the four eigenvectors of $\mathbf{H}\mathbf{H}^T$. Nevertheless, the one that maximizes the quadratic form is clearly the one corresponding to the largest eigenvalue. Since the columns of the co-factor matrix of $\mathbf{H}\mathbf{H}^T - \lambda_{\text{max}}\mathbf{I}$ are proportional to the sought eigenvector, we can take any of these four columns as solution. Nevertheless, from the numerical point of view, it is better to take the one with the largest modulus. Likewise, it is possible to solve the optimization problem in (39) to obtain $\hat{\mathbf{r}}$. Substituting $\hat{\mathbf{l}}$ and $\hat{\mathbf{r}}$ in (28), $\hat{\mathbf{R}}$ is finally obtained.

To end up this section, a final consideration must be added. Since $(\hat{\mathbf{l}}, \hat{\mathbf{r}})$ and $(-\hat{\mathbf{l}}, -\hat{\mathbf{r}})$ represent the same rotation, it might happen that we obtain $(-\hat{\mathbf{l}}, \hat{\mathbf{r}})$ or $(\hat{\mathbf{l}}, -\hat{\mathbf{r}})$ because we have obtained $\hat{\mathbf{l}}$ and $\hat{\mathbf{r}}$ separately without taking into account their sign consistency. Nevertheless, observe that this problem can be easily fixed by changing the sign of $\hat{\mathbf{R}}$ if $\det(\hat{\mathbf{R}}) = -1$.

4 | PERFORMANCE ANALYSIS

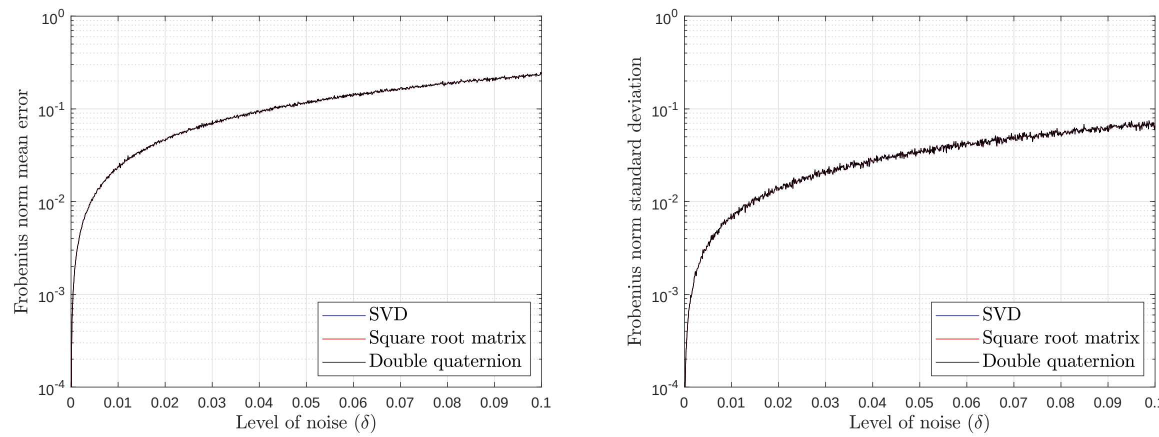

FIGURE 2 Mean (left) and standard deviation (right) of the Frobenius norm of the difference between the original noiseless rotation matrix and the estimated matrix, obtained using the three analyzed methods, as a function of the variance of the added noise. The three curves overlap.

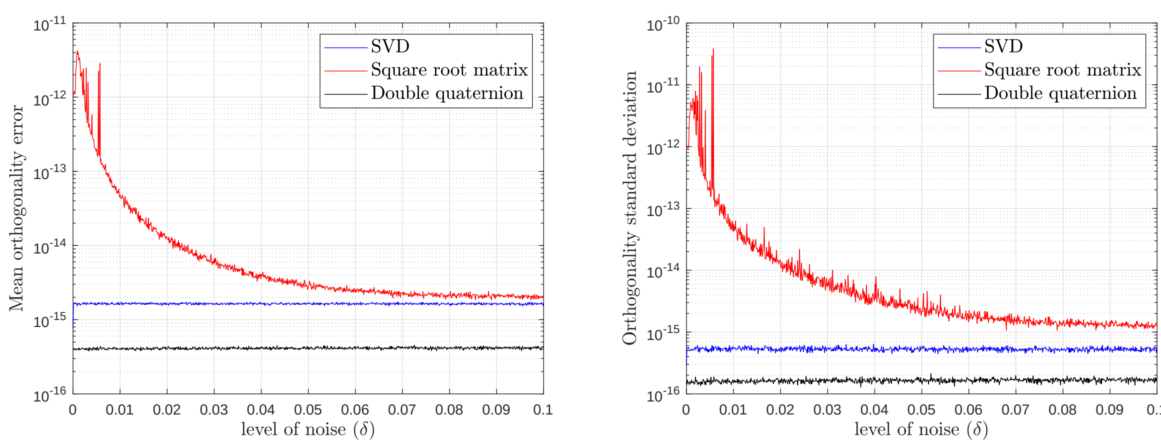

FIGURE 3 Mean (left) and standard deviation (right) of the orthogonality error of $\hat{\mathbf{R}}$ for the three analyzed methods as a function of the variance of the added noise.

In order to compare the two presented closed-form methods with respect to the SVD built-in MATLAB® function, the following procedure have been implemented in MATLAB® (on a PC with a CoreTMi7 processor running at 3.70 GHz and 16 GB of RAM) using double-precision arithmetic:

1. Generate two sets of 200 quaternions, $\mathbf{q}_l$ and $\mathbf{q}_r$, using the algorithm for generating uniformly distributed points on $\mathbb{S}^4$ described in[24].
2. Convert these quaternions to 4D rotation matrices using equation (28) whose elements are then contaminated with additive uncorrelated normally distributed noise with standard deviation equal to δ.
3. Compute the nearest rotation matrices for these 200 noisy rotation matrices using the SVD and the two closed-form methods.
4. Compute the mean and the standard deviation of the Frobenius norm of the difference between the original noiseless matrices and the obtained matrices for each method.
5. Compute the average and the standard deviation of the orthogonality error of the obtained matrices computed as the Frobenius norm of $\hat{\mathbf{R}}\hat{\mathbf{R}}^T - \mathbf{I}$.

If this procedure is repeated for 1000 values of δ uniformly distributed between 0 and 0.1, the plots in Fig. 2 and Fig. 3 are obtained. Fig. 2 shows the mean and the standard deviation of the Frobenius norm of the difference between the noisy rotation matrices and the corresponding original noiseless rotation matrices obtained using the three compared methods. There are no practical differences between the three methods in terms of this error. The situation changes when analyzing the orthogonality error. Fig. 3 shows the mean and the standard deviation of the orthogonality error for the estimated rotation matrices. Although for all practical applications this error is negligible for the three methods, the double quaternion method is clearly the best and the square root matrix method leads to higher orthogonality errors for low levels of noise. This is because the adopted method for resolving the quartic equation exhibits higher numerical inaccuracies when the three roots are close to each other, but they asymptotically decrease as the roots move far apart.

Using MATLAB®, and in terms of ratios, the average computational times are 1 : 1.72 : 1.88 for the SVD, the square root matrix and the double quaternion methods, respectively. Nevertheless, we cannot compare, in terms of time performance, an interpreted program with a compiled program. While Matlab's SVD calls the corresponding function in LAPACK —an optimized and compiled library written in Fortran 90— our Matlab implementation is interpreted. For a fair comparison, we have translated our code in MATLAB® to C++ and compared their performances with the SVD implementation of the celebrated EIGEN library[25]. In this case, the average computational ratios are 1 : 0.23 : 0.33.

The implemented programs, written in both MATLAB® and C++, needed to reproduce the above results, can be downloaded from http://www.iri.upc.edu/people/thomas/Soft/Nearest4DRotationMatrix.zip.

5 | CONCLUSION

We have presented two closed-form methods for solving the nearest 4D rotation matrix problem in Frobenius norm which generalize previous methods available for the 3D case. Their efficiency and accuracy have been demonstrated on simulated data. The obtained results indicate that their computational costs are better than that of the SVD method, the one of choice till now. The reason is simple: the SVD method relies on an iterative algorithm that halts when the improvement in an iteration is below a certain threshold. The presented closed-form methods directly deliver the optimum up to numerical inaccuracies.

ADDENDUM

After the publication of this paper, Prof. Andrew J. Hanson contacted me with an important number of relevant comments, the most important referring to the fact that an alternative derivation to the 4D nearest rotation matrix solution presented in Section 3 can be found in Section 4.2 of the supporting information material of reference [23]. As a consequence, the closed-form double quaternion method presented in this paper should be viewed as an alternate derivation of the existing previous result that should be credited to Prof. Hanson.

F. Thomas

References

1. Sarabandi S, Thomas F. Approximating displacements in R^3 by rotations in R^4 and its application to pointcloud registration. *IEEE Transactions on Robotics* 2022; to appear.

2. Wu J, Sun Y, Wang M, Liu M. Hand-eye calibration: 4D Procrustes analysis approach. *IEEE Transactions on Instrumentation and Measurement* 2020; 69(6): 2966–2981.

3. Morawiec A. *Orientations and Rotations*. Springer . 2004.

4. Horn B. *Robot Vision*. New York: McGraw-Hill . 1986.

5. Petersen KB, Pedersen MS. *The Matrix Cookbook*. available at www2.imm.dtu.dk/pubdb/pubs/3274-full . 2012.

6. Horn BKP, Hilden HM, Negahdaripour S. Closed form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society A* 1988; 5(7): 1127–1135.

7. Sarabandi S, Shabani A, Porta J, Thomas F. On closed-form formulas for the 3D nearest rotation matrix problem. *IEEE Transactions on Robotics* 2020; 36(4): 1333–1339.

8. Eggert DW, Lorusso A, Fisher RB. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications* 1997; 9(5): 272–290.

9. Roberts RG, Repperger DW. On the absolute orientation problem in computer vision. *2009 IEEE International Conference on Systems, Man and Cybernetics* 2009: 4142–4145.

10. Hou S. Classroom note: a simple proof of the Leverrier–Faddeev characteristic polynomial algorithm. *SIAM Review* 1998; 40(3): 706–709.

11. Herbison-Evans D. Solving quartics and cubics for graphics. In: Paeth A., ed. *Graphics Gems V*Chesnut Hill: Academic Press. 1995 (pp. 3–15).

12. S.Neumark . *Solution of Cubic and Quartic Equations*. Oxford: Pergamon Press . 1965.

13. Shmakov SL. A universal method of solving quartic equations. *International Journal of Pure and Applied Mathematics* 2011; 71(2).

14. Tiruneh AT. An alternative approach to solving cubic and quartic polynomial equations. *Journal of Advances in Mathematics and Computer Science* 2021; 36(2): 57–72.

15. Lounesto P. *Clifford Algebras and Spinors*. Cambridge University Press . 2001.

16. Weiner J, Wilkens G. Quaternions and rotations in E^4. *The American Mathematical Monthly* 2005; 112(1): 69–76.

17. Cayley A. Recherches ultérieures sur les déterminants gauches. *The Collected Mathematical Papers of Arthur Cayley, article 137* 1891: 202–215.

18. Thomas F. Approaching dual quaternions from matrix algebra. *IEEE Transactions on Robotics* 2014; 30(5): 1037–1048.

19. Perez-Gracia A, Thomas F. On Cayley's Factorization of 4D Rotations and Applications. *Advances in Applied Clifford Algebras* 2017; 27(1): 523–538.

20. Lasenby J. A Geometric Algebra Approach to Orthogonal Transformations and their use in Signal and Image Processing. *8th Conference on Applied Geometric Algebras in Computer Science and Engineering* 2021.

21. van Elfrinkhof L. Eene eigenschap van de orthogonale substitutie van de vierde orde. *Handelingen van het zesde Nederlandsch Natuur- en Geneeskundig Congres (Acts of the sixth Dutch nature and medical congress)* 1897: 237–240.

22. Rosen N. Note on the general Lorentz transformation. *Journal of Mathematics and Physics* 1930; 9: 181–187.

23. Hanson A. The quaternion-based spatial-coordinate and orientation-frame alignment problems. *Acta Crystallographica Section A: Foundations and Advances* 2020; 76(4): 432–457.

24. Marsaglia G. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics* 1972; 43(2): 645–646.

25. Guennebaud G. Eigen: A C++ linear algebra library. *Eurographics/CGLibs* 2013.

How to cite this article: S. Sarabandi, and F. Thomas (2022), On closed-form solutions to the 4D nearest rotation matrix problem, *Mathematical Methods in the Applied Sciences*, *2022;00:1–8*.