

FULL PAPER

Open Access



# Dual-quaternion-based iterative algorithm of the three dimensional coordinate transformation

Huaien Zeng<sup>1,2,3\*</sup> , Zhihao Wang<sup>3</sup>, Junfeng Li<sup>4</sup>, Siyang Li<sup>3</sup>, Junjie Wang<sup>3</sup> and Xi Li<sup>3</sup>

## Abstract

Nowadays a unit quaternion is widely employed to represent the three-dimensional (3D) rotation matrix and then applied to the 3D similarity coordinate transformation. A unit dual quaternion can describe not only the 3D rotation matrix but also the translation vector meanwhile. Thus it is of great potentiality to the 3D coordinate transformation. The paper constructs the 3D similarity coordinate transformation model based on the unit dual quaternion in the sense of errors-in-variables (EIV). By means of linearization by Taylor's formula, Lagrangian extremum principle with constraints, and iterative numerical technique, the Dual Quaternion Algorithm (DQA) of 3D coordinate transformation in weighted total least squares (WTLS) is proposed. The algorithm is capable to not only compute the transformation parameters but also estimate the full precision information of computed parameters. Two numerical experiments involving an actual geodetic datum transformation case and a simulated case from surface fitting are demonstrated. The results indicate that DQA is not sensitive to the initial values of parameters, and obtains the consistent values of transformation parameters with the quaternion algorithm (QA), regardless of the size of the rotation angles and no matter whether the relative errors of coordinates (pseudo-observations) are small or large. Moreover, the DQA is advantageous to the QA. The key advantage is the improvement of estimated precisions of transformation parameters, i.e. the average decrease percent of standard deviations is 18.28%, and biggest decrease percent is 99.36% for the scaled quaternion and translations in the geodetic datum transformation case. Another advantage is the DQA implements the computation and precision estimation of traditional seven transformation parameters (which still are frequent used yet) from dual quaternion, and even could perform the computation and precision estimation of the scaled quaternion.

**Keywords** Quaternion, Dual quaternion, 3D similarity coordinate transformation, Weighted total least squares (WTLS), Dual quaternion algorithm (DQA)

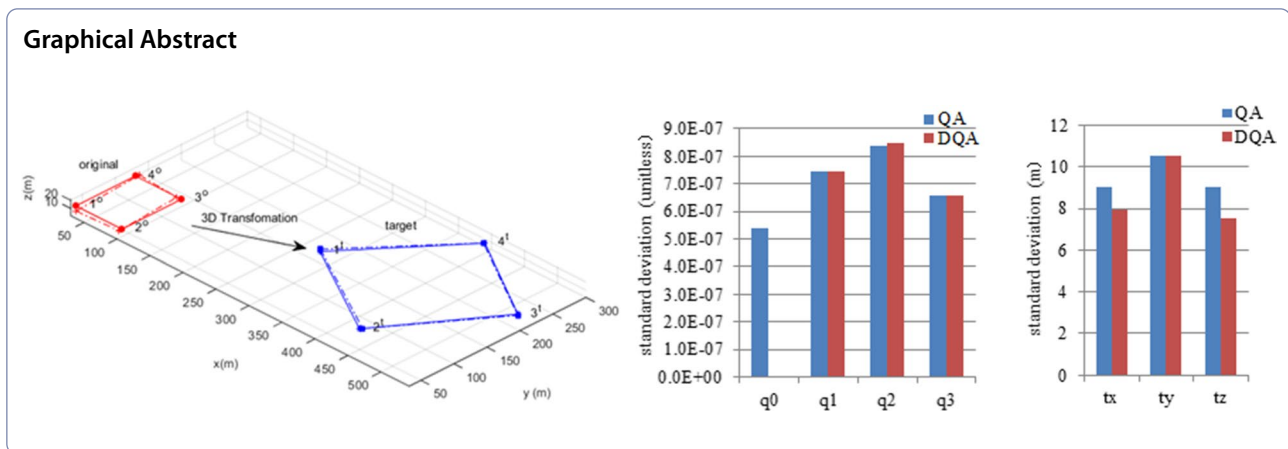
\*Correspondence:

Huaien Zeng  
zenghuaien\_2003@163.com

Full list of author information is available at the end of the article



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



## Introduction

Three dimensional (3D) coordinate transformation is a fundamental and common issue in geodesy, engineering surveying, photogrammetry, LIDAR, GIS, robotics and computer vision, etc. (cf. Arun et al. 1987; Besl and McKay 1992; Burša 1967; Chen et al. 2004; Crosilla and Beinat 2002; Fan et al. 2015; Gargula and Gawronek 2023; Ge and Wunderlich 2016; Horn 1987; Ioannidou and Pantazis 2020; Krarup 1985; Leick and Van Gelder 1975; Li et al. 2012; Odziemczyk 2020; Soycan and Soycan 2008; Teunissen 1988). Although it has different names in different fields, e.g. geodetic datum transformation in geodesy, absolute orientation in photogrammetry, point clouds registration in LIDAR, etc. its essence is to find the transformation parameters by a set of common points from one 3D rectangular coordinate system to another 3D rectangular coordinate system. The 3D similarity coordinate transformation is frequently used 3D coordinate transformation model. It has seven transformation parameters namely one scale factor, three rotation angles and three translations. And it is a nonlinear mathematical model. Traditionally the solution of 3D similarity coordinate transformation is implemented by the least squares (LS) estimation under the Gauss-Markov model, i.e., only the errors of coordinates in the target coordinate system are taken into account (Bektas 2022; Chen et al. 2004; Ioannidou and Pantazis 2020; Kurt 2018; Li et al. 2022; Uygur et al. 2020; Závoti and Kalmár 2016; Zeng et al. 2016, 2019;) Obviously the Gauss-Markov model is not in accord with the fact that all the coordinates in both the original coordinate system and the target coordinate system are contaminated by errors. Thus errors-in-variables (EIV) model, which considers all the errors in both the original and the target coordinate systems, is more reasonable than the Gauss-Markov model. The adjustment

method oriented towards the EIV model is total least squares (TLS) named by Golub and Van Loan (1980). There are a lot of literatures on TLS, e.g. Aydin et al. (2018), Fang (2015), Lv and Sui (2020), Ma et al. (2020), Mahboub (2016), Mercan et al. (2018), Mihajlović and Cvijetinović (2017), Qin et al. (2020), Schaffrin et al. (2012); Uygur et al. (2020); Zeng et al. (2020); Zeng et al. (2022a); Xu et al. (2012); Xu et al. (2023).

Besides the classical LS or TLS adjustment methods, a few analytical algorithms of 3D coordinate transformation in the LS or TLS sense have been put forward so far. The analytical algorithms include the Procrustes algorithms (e.g. Arun et al. 1987; Crosilla and Beinat 2002; Grafarend and Awange 2003; Felus and Burtch 2009; Chang 2015; Păun et al. 2017) utilizing singular value decomposition (SVD), unit quaternion based algorithms (Horn 1987; Shen et al. 2006; Chang et al. 2017) utilizing eigenvalue–eigenvector decomposition, the orthonormal matrix based algorithm (Horn et al. 1988; Zeng 2015) utilizing eigenvalue–eigenvector decomposition, dual quaternion based algorithm (Walker et al. 1991; Wang et al. 2014; Zeng et al. 2022b) utilizing eigenvalue–eigenvector decomposition, Gibbs vector based algorithm (Zeng and Yi 2010) utilizing the property of Rodrigues matrix. The analytical algorithms are able to obtain the theoretic solutions and fast because they employ the theoretic formulae to calculate the solutions without initial values of parameters, and iterations. The disadvantages of the analytical algorithms over numerical or iterative algorithms are the former does not offer the precision estimation of computed parameters, which should be as important as the computed parameters themselves. In addition, the analytical algorithms can only deal with point-wise or identity weight matrix of observations, thus are not available for general weight matrix of observations (Zeng et al. 2022a).

As far as the iterative algorithms are concerned, good initial values of parameters are needed before iterative computations in the most situations. Otherwise the iterative computation may fail. In some cases like registration of point clouds, good initial values are hard or impossible to obtain due to the any size of rotation angles (e.g. Zeng and Yi 2011). At times good initial values of parameters are not necessary or arbitrary initial values of parameters can be available if global optimization algorithms are employed (e.g. Xu 2003). On the one hand, the 3D similarity coordinate transformation is usually not implemented real-time. On the other hand, because the number of common points is not large in the most cases, the computational time of iterations usually is short. So the iterative algorithms are tested majorly with the precision of transformation parameters rather than the efficiency.

Traditional rotation angle (Euler angle) based rotation matrix representation has a lot of trigonometric functions (Bektas 2022; Zeng et al. 2018); as a result, the iterative computation needs very good initial values of parameters. If bad initial values are provided, more time are consumed even iteration diverges (Zeng and Yi 2011). Quaternion based rotation matrix representation involves no trigonometric functions but algebraic (polynomial) functions, which has efficient computational performance (Zeng and Yi 2011). Compared to quaternion, dual quaternion represents not only the rotation matrix but also the translations by polynomials (Walker et al. 1991; Jitka 2011; Zeng et al. 2018, 2022b). In other words, the rigid transformation including rotation and translation could be represented by a dual quaternion. In addition, it is worthy of note that the established transformation model based on a unit dual quaternion does not require centralization of coordinates to exclude the translations while traditional transformation models do (Grafarend and Awange 2003; Felus and Burtch 2009). Thus it is meaningful to explore the computational performance of dual quaternion in 3D coordinate transformation, including the sensitivity of the initial values of parameters or suitability of large or small rotation angle case (bad initial values of parameters usually is caused by large rotation angle), the precision of computed parameters and effect of errors in coordinates, etc.

The remainder of the paper is organized as follows. In next section, the basic concept of quaternion and dual quaternion are introduced in brief. And the paper derives the representation of 3D rotation matrix by a quaternion and the formula of translation vector by a unit dual quaternion from the viewpoint of geometry in detail. In the Sect. “Dual quaternion algorithm (DQA) of 3D coordinate transformation”, the 3D coordinate transformation in the errors-in-variables (EIV) model is established, and then the Dual Quaternion Algorithm (DQA) is presented

after the derivation of linearization of the mathematical model and WTLS solution based on the Lagrangian extremum principle with constraints. Additionally, the computation and precision estimation of traditional seven transformation parameters are fulfilled from dual quaternion considering that the seven transformation parameters are used widely in applications. Two numerical experiments including a geodetic datum transformation case and a simulated case oriented form surface fitting are demonstrated in the Sect. “Experiments and discussion”. Lastly conclusions are drawn in the last section, i.e. Sect. “Conclusions”.

### Concepts of quaternion and dual quaternion

This section firstly introduces the definition and basic properties of quaternion, and emphasizes on interpreting the meaning of 3D rotation of a point around an axis, and representing the 3D rotation by means of quaternion. Next the dual quaternion is introduced and the rigid motion involving 3D rotation about an axis and translation along the axis is expressed by a unit dual quaternion. Especially the relationship between the translation and the dual quaternion is derived in detail.

### Quaternion and 3D rotation matrix

Quaternion is a four-element vector invented by Hamilton in 1843, whose general form is

$$\mathbf{q} = iq_1 + jq_2 + kq_3 + q_4 \quad (1)$$

where  $q_1, q_2, q_3, q_4$  are all real numbers,  $i, j$  and  $k$  are imaginary units also named quaternion units with the following properties: ①  $i^2 = j^2 = k^2 = -1$ , ②  $ij = -ji = k$ , ③  $jk = -kj = i$ , ④  $ki = -ik = j$ . For  $i, j$  and  $k$  are imaginary,  $q_1, q_2, q_3$  are called imaginary parts, and  $q_4$  is called the real part. Quaternion  $\mathbf{q}$  is usually represented in the form of vector to separate the real and imaginary parts as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (2)$$

where  $\mathbf{q}$  is a vector used to express the imaginary parts, named vector part,  $q_4$  is also called the scalar part. Quaternion is called pure imaginary one when its scalar part is zero and vector part is not zero vector. Analogous to the definition of conjugate complex number, the definition of conjugate quaternion corresponding to  $\mathbf{q}$  is

$$\mathbf{q}^* = -iq_1 - jq_2 - kq_3 + q_4 \quad (3)$$

And the norm of quaternion  $\mathbf{q}$  is given as follows.

$$\|q\| = \sqrt{qq^*} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (4)$$

If  $\|q\| = 1$ ,  $q$  is called unit quaternion.

It is easy to prove the following properties utilizing the foregoing definitions.

$$p + q = \begin{pmatrix} p + q \\ p_4 + q_4 \end{pmatrix} \quad (5)$$

$$pq = \begin{pmatrix} p \times q + p_4q + q_4p \\ p_4q_4 - p \cdot q \end{pmatrix} \quad (6)$$

$$pqr = (pq)r = p(qr) \quad (7)$$

$$p(q + r) = pq + pr \quad (8)$$

$$(pq)^* = q^*p^* \quad (9)$$

$$q^{-1} = \frac{q^*}{\|q\|} \quad (10)$$

where  $p$ ,  $q$  and  $r$  are any quaternions,  $q^{-1}$  denotes the inverse of  $q$ . The symbols  $\cdot$  and  $\times$  stand for dot and cross product of vectors respectively as follows.

$$p \cdot q = p^T q \quad (11)$$

$$p \times q = C(p)q = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix} q \quad (12)$$

where subscript T denotes transpose computation, and  $C(p)$  is called a skew symmetric matrix. From the above properties of quaternion, it is seen apparently that the multiplication of quaternion meets associativity and distributivity but non-commutativity. The product  $pq$  can also be fulfilled by the product of a matrix and quaternion in the form of vector:

$$pq = Q(p)q = W(q)p \quad (13)$$

where

$$Q(p) = \begin{bmatrix} p_4I + C(p) & p \\ -p^T & p_4 \end{bmatrix} \quad (14)$$

$$W(q) = \begin{bmatrix} q_4I - C(q) & q \\ -q^T & q_4 \end{bmatrix} \quad (15)$$

Next the relationship of 3D rotation matrix and quaternion is investigated. See Fig. 1, assume any point P in the

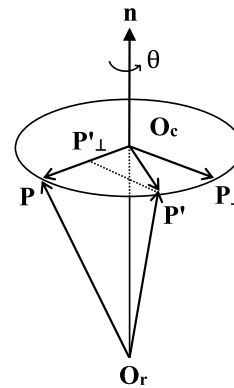


Fig. 1 3D rotation of a point about an axis

3D space rotates an angle of  $\theta$  (be positive in the direction of counterclockwise for right-hand coordinate system) around the rotation axis represented by a unit vector  $n$ , and the corresponding point after rotation is  $P'$ .  $O_r$  is the origin of 3D rectangular coordinate system,  $O_c$  is the foot of a perpendicular from pint P to the rotation axis,  $P_\perp$  is the point after 90° rotation of point P.  $P'_\perp$  is the foot of a perpendicular from pint  $P'$  to the line of  $O_c$  to P. The vectors  $\overrightarrow{O_rO_c}$ ,  $\overrightarrow{O_cP'_\perp}$ ,  $\overrightarrow{P'_\perp P'}$  are computed as follows.

$$\overrightarrow{O_rO_c} = n(n \cdot p) \quad (16)$$

$$\overrightarrow{O_cP'_\perp} = (p - n(n \cdot p)) \cos \theta \quad (17)$$

$$\overrightarrow{P'_\perp P'} = \overrightarrow{O_cP_\perp} \sin \theta = (n \times p) \sin \theta \quad (18)$$

So

$$\begin{aligned} p' &= \overrightarrow{O_rO_c} + \overrightarrow{O_cP'_\perp} + \overrightarrow{P'_\perp P'} \\ &= n(n \cdot p) + (p - n(n \cdot p)) \cos \theta + (n \times p) \sin \theta \\ &= p \cos \theta + (1 - \cos \theta)n(n \cdot p) + (n \times p) \sin \theta \\ &= p \cos \theta + (1 - \cos \theta)n(n^T p) + (n \times p) \sin \theta \\ &= p \cos \theta + (1 - \cos \theta)(nn^T)p + (n \times p) \sin \theta \\ &= (I_3 \cos \theta + (1 - \cos \theta)(nn^T) + \sin \theta C(n))p \\ &= Rp \end{aligned} \quad (19)$$

where  $I_3$  is a 3-by-3 identity matrix and  $R$  is the 3D rotation matrix. Thus the rotation matrix is expressed by the unit vector  $n$  and rotation angle  $\theta$ . Introducing a unit quaternion

$$\mathbf{r} = \begin{bmatrix} \mathbf{n} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} \quad (20)$$

The rotation matrix  $\mathbf{R}$  by Eq. (19) is rewritten as

$$\begin{aligned} \mathbf{R} &= \mathbf{I}_3 \left( \left( \cos \frac{\theta}{2} \right)^2 - \left( \sin \frac{\theta}{2} \right)^2 \right) + 2 \left( \sin \frac{\theta}{2} \right)^2 (\mathbf{nn}^T) + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} \mathbf{C}(\mathbf{n}) \\ &= \mathbf{I}_3 (r_4^2 - \mathbf{r}^T \mathbf{r}) + 2 (\mathbf{rr}^T + r_4 \mathbf{C}(\mathbf{r})) \\ &= \begin{bmatrix} r_4^2 + r_1^2 - r_2^2 - r_3^2 & 2(r_1 r_2 - r_3 r_4) & 2(r_1 r_3 + r_2 r_4) \\ 2(r_1 r_2 + r_3 r_4) & r_4^2 - r_1^2 + r_2^2 - r_3^2 & 2(r_2 r_3 - r_1 r_4) \\ 2(r_1 r_3 - r_2 r_4) & 2(r_2 r_3 + r_1 r_4) & r_4^2 - r_1^2 - r_2^2 + r_3^2 \end{bmatrix} \end{aligned} \quad (21)$$

Therefore the rotation matrix is directly represented by the unit quaternion. If constructing pure quaternions  $\mathbf{p}$  and  $\mathbf{p}'$  with the vectors  $\mathbf{p}$  and  $\mathbf{p}'$ , the rotation transformation is expressed with quaternions as (Walker et al. 1991; Zeng and Yi 2011)

$$\mathbf{p}' = \mathbf{r} \mathbf{p} \mathbf{r}^* = Q(\mathbf{r}) \mathbf{W}(\mathbf{r}^*) \mathbf{p} = Q(\mathbf{r}) \mathbf{W}(\mathbf{r})^T \mathbf{p} = \mathbf{W}(\mathbf{r})^T Q(\mathbf{r}) \mathbf{p} \quad (22)$$

and  $\mathbf{W}(\mathbf{r})^T Q(\mathbf{r})$  can be expanded as

$$\mathbf{W}(\mathbf{r})^T Q(\mathbf{r}) = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (23)$$

### Dual quaternion and rigid motion

Dual quaternion which was invented by William Kingdon Clifford in 1873 and originally named biquaternion (Clifford 1873), is described as

$$\begin{aligned} \bar{q} &= \mathbf{r} + \varepsilon \mathbf{s} \\ &= (r_1 + \varepsilon s_1) i + (r_2 + \varepsilon s_2) j + (r_3 + \varepsilon s_3) k + (r_4 + \varepsilon s_4), \\ &= q_{d1} i + q_{d2} j + q_{d3} k + q_{d4} \end{aligned} \quad (24)$$

where  $\mathbf{r}$  and  $\mathbf{s}$  are arbitrary quaternions, and  $\varepsilon$  is a basis element named dual unit.  $\varepsilon$  satisfies the following properties:  $\varepsilon^2 = 0$  and multiplication of  $\varepsilon$  and quaternion units meets commutativity, e.g.  $\varepsilon i = i \varepsilon$ .  $(q_{d1} \ q_{d2} \ q_{d3})^T$  is the vector (dual number vector) part,  $q_{d4}$  is the scalar (dual number) part. The product of dual quaternions  $\bar{p} = \mathbf{u} + \varepsilon \mathbf{v}$  and  $\bar{q}$  is easily deduced by the above properties of  $\varepsilon$  as

$$\bar{p} \bar{q} = \mathbf{ur} + \varepsilon (\mathbf{us} + \mathbf{vr}). \quad (25)$$

The conjugate of dual quaternion is defined utilizing the conjugate of quaternion:

$$\bar{q}^* = \mathbf{r}^* + \varepsilon \mathbf{s}^*. \quad (26)$$

And the norm of dual quaternion is a dual scalar (number) and is defined analogous to the definition of norm of quaternion as

$$\|\bar{q}\| = \sqrt{\bar{q}^* \bar{q}} = \sqrt{q_{d1}^2 + q_{d2}^2 + q_{d3}^2 + q_{d4}^2}. \quad (27)$$

If  $\|\bar{q}\| = 1$ ,  $\bar{q}$  is called a unit dual quaternion. Unit dual quaternion  $\bar{q}$  satisfies the following two conditions (Jitka 2011; Walker et al 1991).

$$\mathbf{r}^T \mathbf{r} = 1, \quad (28)$$

$$\mathbf{r}^T \mathbf{s} = 0. \quad (29)$$

In other words, for unit dual quaternion  $\bar{q}$ , the non-dual part i.e.  $\mathbf{r}$  is a unit quaternion, and non-dual part is perpendicular to the dual part i.e.  $\mathbf{s}$ .

The rigid motion involving rotation about an axis and translation along the axis can be expressed elegantly by a unit dual quaternion (Walker et al 1991; Jitka 2011; Zeng et al. 2018). The rigid motion is depicted in Fig. 2. The O-XYZ is a rectangular coordinate system, and  $\mathbf{p}$  is a vector starting from the origin O of coordinate system.

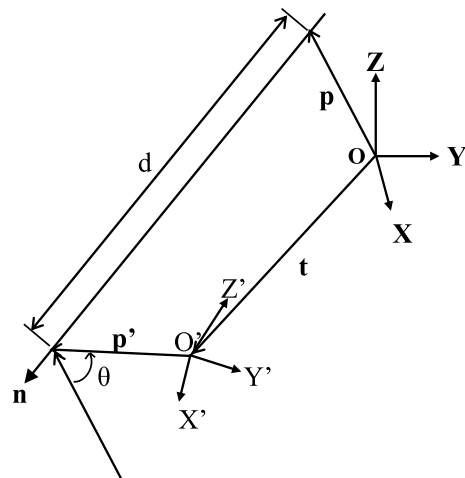


Fig. 2 Rigid motion

The vector  $\mathbf{p}$  is firstly translated by a distance  $d$  along the rotation axis denoted by the unit vector  $\mathbf{n}$ , and then rotated by an angle  $\theta$  about the rotation axis. The new rectangular coordinate system  $O'-X'Y'Z'$ , the new vector  $\mathbf{p}'$  after rotation, and the translation vector  $\mathbf{t}$  from the original system to the new one are denoted in the figure. Rewriting the unit dual quaternion  $\bar{\mathbf{q}}$  in Eq. (24) as (Walker et al 1991)

$$\bar{\mathbf{q}} = \begin{bmatrix} \bar{\mathbf{n}} \sin \frac{\bar{\theta}}{2} \\ \cos \frac{\bar{\theta}}{2} \end{bmatrix} \quad (30)$$

where  $\bar{\mathbf{n}}$  and  $\bar{\theta}$  are dual vector and dual angle represented as

$$\bar{\mathbf{n}} = \mathbf{n} + \varepsilon \mathbf{p} \times \mathbf{n} \quad (31)$$

$$\bar{\theta} = \theta + \varepsilon d \quad (32)$$

Inserting the above two equations into Eq. (30) and expanding the equation and comparing with Eq. (24), one obtains

$$\mathbf{r} = \begin{bmatrix} \mathbf{n} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix}, \quad (33)$$

$$\mathbf{s} = \begin{bmatrix} \frac{d}{2} \cos \frac{\theta}{2} \mathbf{n} + \sin \frac{\theta}{2} \mathbf{p} \times \mathbf{n} \\ -\frac{d}{2} \sin \frac{\theta}{2} \end{bmatrix} \quad (34)$$

where Eq. (33) is identical to Eq. (20). Thus the non-dual part of unit dual quaternion  $\bar{\mathbf{q}}$  i.e.  $\mathbf{r}$  is employed to express the rotation transformation as described early [see Eq. (21)], the left part of rigid body transformation i.e. translation transformation can be represented by  $\mathbf{r}$  and  $\mathbf{s}$  with derivation as follows.

From Fig. 2, the following equation is obtained.

$$\mathbf{t} = \mathbf{p} + d\mathbf{n} - \mathbf{p}' = \mathbf{p} + d\mathbf{n} - \mathbf{R}\mathbf{p}. \quad (35)$$

Substituting the  $\mathbf{R}\mathbf{p}$  in Eq. (19) into the above equation, and considering

$$\mathbf{n}\mathbf{n}^T = \mathbf{C}(\mathbf{n})\mathbf{C}(\mathbf{n}) + \mathbf{I}_3, \quad (36)$$

One gets

$$\begin{aligned} \mathbf{t} &= \mathbf{p} + d\mathbf{n} - \mathbf{p}' \\ &= \mathbf{p} - (\mathbf{I}_3 \cos \theta + (1 - \cos \theta)(\mathbf{C}(\mathbf{n})\mathbf{C}(\mathbf{n}) + \mathbf{I}_3) + \sin \theta \mathbf{C}(\mathbf{n}))\mathbf{p} + d\mathbf{n} \\ &= \mathbf{p} - (\mathbf{I}_3 + (1 - \cos \theta)\mathbf{C}(\mathbf{n})\mathbf{C}(\mathbf{n}) + \sin \theta \mathbf{C}(\mathbf{n}))\mathbf{p} + d\mathbf{n} \\ &= - (1 - \cos \theta)\mathbf{C}(\mathbf{n})\mathbf{C}(\mathbf{n})\mathbf{p} - \sin \theta \mathbf{C}(\mathbf{n})\mathbf{p} + d\mathbf{n} \\ &= (1 - \cos \theta)\mathbf{n} \times (\mathbf{p} \times \mathbf{n}) + \sin \theta (\mathbf{p} \times \mathbf{n}) + d\mathbf{n}. \end{aligned} \quad (37)$$

By the properties of quaternions, one can compute a new quaternion  $\mathbf{t}$  as

$$\begin{aligned} \mathbf{t} &= \mathbf{s}\mathbf{r}^{-1} \\ &= \mathbf{s}\mathbf{r}^* \\ &= \begin{bmatrix} -\left(\frac{d}{2} \cos \frac{\theta}{2} \mathbf{n} + \sin \frac{\theta}{2} \mathbf{p} \times \mathbf{n}\right) \times \mathbf{n} \sin \frac{\theta}{2} + \frac{d}{2} \sin \frac{\theta}{2} \mathbf{n} \sin \frac{\theta}{2} + \cos \frac{\theta}{2} \left(\frac{d}{2} \cos \frac{\theta}{2} \mathbf{n} + \sin \frac{\theta}{2} \mathbf{p} \times \mathbf{n}\right) \\ -\frac{d}{2} \sin \frac{\theta}{2} \cos \frac{\theta}{2} + \left(\frac{d}{2} \cos \frac{\theta}{2} \mathbf{n} + \sin \frac{\theta}{2} \mathbf{p} \times \mathbf{n}\right) \cdot (\mathbf{n} \sin \frac{\theta}{2}) \end{bmatrix} \\ &= \begin{bmatrix} \left(\sin \frac{\theta}{2}\right)^2 \mathbf{n} \times (\mathbf{p} \times \mathbf{n}) + \frac{d}{2} \left(\sin \frac{\theta}{2}\right)^2 \mathbf{n} + \frac{d}{2} \left(\cos \frac{\theta}{2}\right)^2 \mathbf{n} + \sin \frac{\theta}{2} \cos \frac{\theta}{2} \mathbf{p} \times \mathbf{n} \\ -\frac{d}{2} \sin \frac{\theta}{2} \cos \frac{\theta}{2} + \left(\frac{d}{2} \cos \frac{\theta}{2} \mathbf{n}\right) \cdot (\mathbf{n} \sin \frac{\theta}{2}) \end{bmatrix} \\ &= \begin{bmatrix} \left(\sin \frac{\theta}{2}\right)^2 \mathbf{n} \times (\mathbf{p} \times \mathbf{n}) + \frac{d}{2} \mathbf{n} + \sin \frac{\theta}{2} \cos \frac{\theta}{2} \mathbf{p} \times \mathbf{n} \\ -\frac{d}{2} \sin \frac{\theta}{2} \cos \frac{\theta}{2} + \frac{d}{2} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \mathbf{n} \cdot \mathbf{n} \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} (1 - \cos \theta)\mathbf{n} \times (\mathbf{p} \times \mathbf{n}) + d\mathbf{n} + \sin \theta \mathbf{p} \times \mathbf{n} \\ 0 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix} \end{aligned} \quad (38)$$

Therefore, actually  $\mathbf{t}$  is a pure imaginary quaternion constructed by half of translation vector  $\mathbf{t}$ , which can be represented by

$$\mathbf{t} = \mathbf{s}\mathbf{r}^* = W(\mathbf{r}^*)\mathbf{s} = W(\mathbf{r})^T\mathbf{s}. \tag{39}$$

### Dual Quaternion Algorithm (DQA) of 3D coordinate transformation

#### Mathematical model and its linearization

The similarity 3D coordinate transformation model is usually expressed as follows.

$$\mathbf{p}_i^t = \lambda\mathbf{R}\mathbf{p}_i^o + \mathbf{t}, \tag{40}$$

Subject to

$$\mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = +1, \tag{41}$$

where  $\mathbf{p}_i^t = [x_i^t \ y_i^t \ z_i^t]^T$  and  $\mathbf{p}_i^o = [x_i^o \ y_i^o \ z_i^o]^T$  are the 3D coordinate vectors of control point, in which superscript  $t$  or  $o$  denotes target coordinate or original coordinate respectively, and subscript  $i = 1, 2, \dots, n$  ( $n \geq 3$ ) denotes the number of control point.  $\det$  represents determinant computation of matrix.  $\det(\mathbf{R}) = +1$  is imposed necessarily to ensure  $\mathbf{R}$  stands for a rotation matrix rather than a reflection one when  $\det(\mathbf{R}) = -1$ .  $\mathbf{t} = [t_x \ t_y \ t_z]^T$  is the translation vector.  $\lambda$  is the scale factor which is tightly close to 1 for most situations. Traditionally  $\mathbf{R}$  is repressed by three rotation angles  $\theta_x, \theta_y, \theta_z$  about the X, Y, Z axes respectively as (Zeng et al. 2019)

$$\mathbf{R} = \begin{bmatrix} \cos\theta_z \cos\theta_y & \sin\theta_z \cos\theta_x + \cos\theta_z \sin\theta_y \sin\theta_x & \sin\theta_z \sin\theta_x - \cos\theta_z \sin\theta_y \cos\theta_x \\ -\sin\theta_z \cos\theta_y & \cos\theta_z \cos\theta_x - \sin\theta_z \sin\theta_y \sin\theta_x & \cos\theta_z \sin\theta_x + \sin\theta_z \sin\theta_y \cos\theta_x \\ \sin\theta_y & -\cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x \end{bmatrix} \tag{42}$$

Obviously the computation of  $\mathbf{R}$  in Eq. (42) involves lots of trigonometric functions thus the computation burden is larger than that of  $\mathbf{R}$  in Eq. (20). If  $\mathbf{R}$  is known or computed, the rotation angles and the quaternion can be computed as elaborated in Zeng et al. (2019).

Introducing two pure imaginary quaternions with coordinate vectors as

$$\mathbf{p}_i^o = \begin{bmatrix} \mathbf{p}_i^o \\ 0 \end{bmatrix}, \mathbf{p}_i^t = \begin{bmatrix} \mathbf{p}_i^t \\ 0 \end{bmatrix}, \tag{43}$$

the similarity 3D coordinate transformation in the errors-in-variables (EIV) model is re-expressed by dual quaternion as

$$\mathbf{p}_i^t - \mathbf{e}_i^t = \lambda\mathbf{W}(\mathbf{r})^T Q(\mathbf{r})(\mathbf{p}_i^o - \mathbf{e}_i^o) + 2W(\mathbf{r})^T\mathbf{s}, \tag{44}$$

subject to Eqs. (28, 29). In Eq. (44),  $\mathbf{e}_i^t$  and  $\mathbf{e}_i^o$  are the error quaternion corresponding to  $\mathbf{p}_i^t$  and  $\mathbf{p}_i^o$  respectively.

The computation of transformation parameters needs to utilize linearization and iterative procedure. Linearizing Eq. (44) by Taylor's formula, one gets

$$\begin{aligned} \mathbf{p}_i^t - \mathbf{e}_i^t &= \lambda^j \mathbf{W}(\mathbf{r}^j)^T Q(\mathbf{r}^j)(\mathbf{p}_i^o - \mathbf{e}_i^{o,j}) \\ &\quad + 2\mathbf{W}(\mathbf{r}^j)^T \mathbf{s}^j - \lambda^j \mathbf{W}(\mathbf{r}^j)^T Q(\mathbf{r}^j) d\mathbf{e}_i^{o,j} \\ &\quad + \mathbf{W}(\mathbf{r}^j)^T Q(\mathbf{r}^j)(\mathbf{p}_i^o - \mathbf{e}_i^{o,j}) d\lambda + \frac{\partial f}{\partial r_1} dr_1 \\ &\quad + \frac{\partial f}{\partial r_2} dr_2 + \frac{\partial f}{\partial r_3} dr_3 + \frac{\partial f}{\partial r_4} dr_4 + 2\mathbf{W}(\mathbf{r}^j)^T d\mathbf{s} \end{aligned} \tag{45}$$

where superscript  $j$  denotes the  $j$ -th iterative time, and

$$\begin{aligned} \frac{\partial f}{\partial r_i} &= \lambda^j \left( \frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_i} Q(\mathbf{r}^j) + \mathbf{W}(\mathbf{r}^j)^T \frac{\partial Q(\mathbf{r})}{\partial r_i} \right) \\ &(\mathbf{p}_i^o - \mathbf{e}_i^{o,j}) + 2 \frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_i} \mathbf{s}^j, \quad i = 1, 2, 3, 4 \end{aligned} \tag{46}$$

$$\frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_1} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_2} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{47}$$

$$\frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_3} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \frac{\partial \mathbf{W}(\mathbf{r})^T}{\partial r_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{48}$$

$$\frac{\partial Q(\mathbf{r})}{\partial r_1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \frac{\partial Q(\mathbf{r})}{\partial r_2} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \tag{49}$$

$$\frac{\partial Q(\mathbf{r})}{\partial r_3} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \frac{\partial Q(\mathbf{r})}{\partial r_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{50}$$

Considering  $\mathbf{e}_i^o = \mathbf{e}_i^{o,j} + d\mathbf{e}_i^{o,j}$ , Eq. (45) can be rewritten as

$$\begin{aligned} \mathbf{p}_i^t - \mathbf{e}_i^t &= \lambda^j \mathbf{W}(\mathbf{r}^j)^T Q(\mathbf{r}^j)(\mathbf{p}_i^o - \mathbf{e}_i^o) + 2\mathbf{W}(\mathbf{r}^j)^T \mathbf{s}^j \\ &\quad + \mathbf{W}(\mathbf{r}^j)^T Q(\mathbf{r}^j)(\mathbf{p}_i^o - \mathbf{e}_i^{o,j}) d\lambda + \frac{\partial f}{\partial r_1} dr_1 + \frac{\partial f}{\partial r_2} dr_2 + \frac{\partial f}{\partial r_3} dr_3 + \frac{\partial f}{\partial r_4} dr_4 + 2\mathbf{W}(\mathbf{r}^j)^T d\mathbf{s} \\ &= \mathbf{B}_i^j \mathbf{p}_i^o + \mathbf{C}_i^j + \mathbf{A}_i^{1j} dx_1 + \mathbf{A}_i^{2j} dx_2 - \mathbf{B}_i^j \mathbf{e}_i^o \end{aligned} \tag{51}$$

where

$$dx_1 = [d\lambda \ dr_1 \ dr_2 \ dr_3 \ dr_4]^T, \tag{52}$$

$$dx_2 = ds = [ds_1 \ ds_2 \ ds_3 \ ds_4]^T, \tag{53}$$

$$A_i^{1j} = \left[ \mathbf{W}(r^j)^T Q(r^j) (\mathbf{p}_i^o - \mathbf{e}_i^{o,j}) \ \frac{\partial f}{\partial r_1} \ \frac{\partial f}{\partial r_2} \ \frac{\partial f}{\partial r_3} \ \frac{\partial f}{\partial r_4} \right], \tag{54}$$

$$A_i^{2j} = 2\mathbf{W}(r^j)^T, \tag{55}$$

$$B_i^j = \lambda^j \mathbf{W}(r^j)^T Q(r^j), \tag{56}$$

$$C_i^j = 2\mathbf{W}(r^j)^T s^j. \tag{57}$$

For  $n$  ( $n \geq 3$ ) control points, one has

$$\mathbf{p}^t - \mathbf{e}^t = \mathbf{B}^j \mathbf{p}^o + \mathbf{C}^j + A^{1j} dx_1 + A^{2j} dx_2 - \mathbf{B}^j \mathbf{e}^o \tag{58}$$

where

$$\mathbf{p}^t = \begin{bmatrix} \mathbf{p}_1^t \\ \vdots \\ \mathbf{p}_n^t \end{bmatrix}, \mathbf{e}^t = \begin{bmatrix} e_1^t \\ \vdots \\ e_n^t \end{bmatrix}, \mathbf{p}^o = \begin{bmatrix} \mathbf{p}_1^o \\ \vdots \\ \mathbf{p}_n^o \end{bmatrix}, \mathbf{e}^o = \begin{bmatrix} e_1^o \\ \vdots \\ e_n^o \end{bmatrix} \tag{59}$$

$$A^{1j} = \begin{bmatrix} A_1^{1j} \\ A_2^{1j} \\ \vdots \\ A_n^{1j} \end{bmatrix}, A^{2j} = \begin{bmatrix} A_1^{2j} \\ A_2^{2j} \\ \vdots \\ A_n^{2j} \end{bmatrix}, \tag{60}$$

$$C^j = \begin{bmatrix} C_1^j \\ C_2^j \\ \vdots \\ C_n^j \end{bmatrix}, B^j = \begin{bmatrix} B_1^j & & & \\ & B_2^j & & \\ & & \ddots & \\ & & & B_n^j \end{bmatrix}.$$

Introducing

$$E = \mathbf{p}^t - \mathbf{B}^j \mathbf{p}^o - \mathbf{C}^j, \tag{61}$$

Equation (58) can be rewritten as

$$\mathbf{e}^t = E - A^{1j} dx_1 - A^{2j} dx_2 + \mathbf{B}^j \mathbf{e}^o \tag{62}$$

Linearizing the constraints of unit dual quaternion i.e. Equations (28) and (29), one obtains

$$C_{x1} dx_1 + C_{x2} dx_2 = W_x \tag{63}$$

where

$$C_{x1} = \begin{bmatrix} 0 & 2r_1 & 2r_2 & 2r_3 & 2r_4 \\ 0 & s_1 & s_2 & s_3 & s_4 \end{bmatrix}, \tag{64}$$

$$C_{x2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ r_1 & r_2 & r_3 & r_4 \end{bmatrix}, \tag{65}$$

$$W_x = \begin{bmatrix} 1 - r_1^2 - r_2^2 - r_3^2 - r_4^2 \\ -r_1 s_1 - r_2 s_2 - r_3 s_3 - r_4 s_4 \end{bmatrix}. \tag{66}$$

#### Derivation of formulae based on the dual quaternion

The solution of 3D similarity transformation in EIV model is based on the principle of weighted total least squares (WTLS) i.e.

$$\mathbf{e}^{tT} \mathbf{W}^t \mathbf{e}^t + \mathbf{e}^{oT} \mathbf{W}^o \mathbf{e}^o = \min \tag{67}$$

where  $\mathbf{W}^t$  and  $\mathbf{W}^o$  are the weight matrices of observations in the target system and in the original system respectively. Considering the constraints in Eq. (63), a Lagrangian extremum problem with constraints is constructed as follows.

---


$$\min_{\mathbf{e}^o, \mathbf{e}^t, dx_1, dx_2, \mathbf{k}_1, \mathbf{k}_2} \left\{ l = \mathbf{e}^{tT} \mathbf{W}^t \mathbf{e}^t + \mathbf{e}^{oT} \mathbf{W}^o \mathbf{e}^o + 2\mathbf{k}_1^T (\mathbf{e}^t - E + A^{1j} dx_1 + A^{2j} dx_2 - \mathbf{B}^j \mathbf{e}^o) \right. \tag{68}$$

$$\left. + 2\mathbf{k}_2^T (C_{x1} dx_1 + C_{x2} dx_2 - W_x) \right\}$$


---



According to the Lagrangian extremum principle, the minimum exists if and only if the following conditions are met.

$$\frac{\delta l}{\delta \mathbf{e}^o} = 2\mathbf{e}^{oT} \mathbf{W}^o - 2\mathbf{k}_1^T \mathbf{B}^j = 0 \tag{69}$$

$$\frac{\delta l}{\delta \mathbf{e}^t} = 2\mathbf{e}^{tT} \mathbf{W}^t + 2\mathbf{k}_1^T = 0 \tag{70}$$

$$\frac{\delta l}{\delta dx_1} = 2\mathbf{k}_1^T \mathbf{A}^{1j} + 2\mathbf{k}_2^T \mathbf{C}_{x1} = 0 \tag{71}$$

$$\frac{\delta l}{\delta dx_2} = 2\mathbf{k}_1^T \mathbf{A}^{2j} + 2\mathbf{k}_2^T \mathbf{C}_{x2} = 0 \tag{72}$$

$$\frac{\delta l}{\delta \mathbf{k}_1} = 2(\mathbf{e}^t - \mathbf{E} + \mathbf{A}^{1j} dx_1 + \mathbf{A}^{2j} dx_2 - \mathbf{B}^j \mathbf{e}^o) = 0 \tag{73}$$

$$\frac{\delta l}{\delta \mathbf{k}_2} = 2(\mathbf{C}_{x1} dx_1 + \mathbf{C}_{x2} dx_2 - \mathbf{W}_x) = 0 \tag{74}$$

From Eq. (69), one gets

$$\mathbf{e}^o = \mathbf{W}^{o-1} \mathbf{B}^{jT} \mathbf{k}_1 \tag{75}$$

From Eq. (70), one gets

$$\mathbf{e}^t = \mathbf{W}^{t-1} \mathbf{k}_1 \tag{76}$$

Substituting Eqs. (75) and (76) into (73), and making arrangement, one gets

$$\mathbf{k}_1 = \mathbf{W}^{ot} (-\mathbf{E} + \mathbf{A}^{1j} dx_1 + \mathbf{A}^{2j} dx_2), \tag{77}$$

where

$$\mathbf{W}^{ot} = (\mathbf{W}^{t-1} + \mathbf{B}^j \mathbf{W}^{o-1} \mathbf{B}^{jT})^{-1}. \tag{78}$$

Substituting Eqs. (77) into (71) and introducing

$$\mathbf{N}_{11} = \mathbf{A}^{1jT} \mathbf{W}^{ot} \mathbf{A}^{1j}, \tag{79}$$

$$\mathbf{N}_{12} = \mathbf{A}^{1jT} \mathbf{W}^{ot} \mathbf{A}^{2j}. \tag{80}$$

one gets

$$dx_1 = -\mathbf{N}_{11}^{-1} \mathbf{N}_{12} dx_2 + \mathbf{N}_{11}^{-1} \mathbf{A}^{1jT} \mathbf{W}^{ot} \mathbf{E} - \mathbf{N}_{11}^{-1} \mathbf{C}_{x1}^T \mathbf{k}_2. \tag{81}$$

Substituting Eqs. (81) into (72) and introducing

$$\mathbf{N}_{22} = \mathbf{A}^{2jT} \mathbf{W}^{ot} \mathbf{A}^{2j}, \tag{82}$$

$$\mathbf{N}_{21} = \mathbf{A}^{2jT} \mathbf{W}^{ot} \mathbf{A}^{1j}, \tag{83}$$

$$\mathbf{N}_{33} = \mathbf{N}_{22} - \mathbf{N}_{21} \mathbf{N}_{11}^{-1} \mathbf{N}_{12}, \tag{84}$$

$$\mathbf{D}_{11} = \mathbf{N}_{33}^{-1} (-\mathbf{C}_{x2}^T + \mathbf{N}_{21} \mathbf{N}_{11}^{-1} \mathbf{C}_{x1}^T) \tag{85}$$

$$\mathbf{D}_{10} = \mathbf{N}_{33}^{-1} (\mathbf{N}_{21} \mathbf{N}_{11}^{-1} \mathbf{A}^{1jT} - \mathbf{A}^{2jT}) \tag{86}$$

one gets

$$dx_2 = \mathbf{D}_{11} \mathbf{k}_2 - \mathbf{D}_{10} \mathbf{W}^{ot} \mathbf{E}. \tag{87}$$

Substituting Eqs. (87) into (81) and introducing

$$\mathbf{D}_{21} = -\mathbf{N}_{11}^{-1} \mathbf{N}_{12} \mathbf{D}_{11} - \mathbf{N}_{11}^{-1} \mathbf{C}_{x1}^T \tag{88}$$

$$\mathbf{D}_{20} = -\mathbf{N}_{11}^{-1} (\mathbf{N}_{12} \mathbf{D}_{10} + \mathbf{A}^{1jT}) \tag{89}$$

one gets

$$dx_1 = \mathbf{D}_{21} \mathbf{k}_2 - \mathbf{D}_{20} \mathbf{W}^{ot} \mathbf{E}. \tag{90}$$

Substituting Eqs. (90) and (87) into (74), one obtains the solution of  $\mathbf{k}_2$  as

$$\mathbf{k}_2 = (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} (\mathbf{C}_{x1} \mathbf{D}_{20} + \mathbf{C}_{x2} \mathbf{D}_{10}) \mathbf{W}^{ot} \mathbf{E} + (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} \mathbf{W}_x \tag{91}$$

Substituting Eqs. (91) into (90) and (87), one obtains

$$dx_1 = \mathbf{F}_1 \mathbf{W}^{ot} \mathbf{E} + \mathbf{D}_{21} (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} \mathbf{W}_x \tag{92}$$

$$dx_2 = \mathbf{F}_2 \mathbf{W}^{ot} \mathbf{E} + \mathbf{D}_{11} (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} \mathbf{W}_x \tag{93}$$

where

$$\mathbf{F}_1 = \mathbf{D}_{21} (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} (\mathbf{C}_{x1} \mathbf{D}_{20} + \mathbf{C}_{x2} \mathbf{D}_{10}) - \mathbf{D}_{20}, \tag{94}$$

$$\mathbf{F}_2 = \mathbf{D}_{11} (\mathbf{C}_{x1} \mathbf{D}_{21} + \mathbf{C}_{x2} \mathbf{D}_{11})^{-1} (\mathbf{C}_{x1} \mathbf{D}_{20} + \mathbf{C}_{x2} \mathbf{D}_{10}) - \mathbf{D}_{10}. \tag{95}$$

So far the derivations of formulae have been accomplished. Then one can employ the classic Gauss–Newton iterative method to seek the solution of transformation parameters. The stop condition of iteration is

$$abs(\sigma_j^2 - \sigma_{j-1}^2) < \tau \tag{96}$$

where

$$\sigma^2 = \frac{\mathbf{e}^{tT} \mathbf{W}^t \mathbf{e}^t + \mathbf{e}^{oT} \mathbf{W}^o \mathbf{e}^o}{3n - 7} \tag{97}$$

Subscript  $j$  and  $j-1$  denote the present iterative times and the previous iterative times. And  $\tau$  is a threshold ( $1.0 \times 10^{-14}$  is set in the paper).

After the iterative computation converges, the variance factor of unit weight is estimated as

$$\sigma = \pm \sqrt{\frac{\mathbf{e}^{tT} \mathbf{W}^t \mathbf{e}^t + \mathbf{e}^{oT} \mathbf{W}^o \mathbf{e}}{3n - 7}} \quad (98)$$

By Eqs. (92) and (93), the estimated variance or covariance matrices of transformation parameters  $x_1$  and  $x_2$  is derived as

$$\mathbf{D}_{x_1} = \sigma^2 \mathbf{F}_1 \mathbf{W}^{ot} \mathbf{F}_1^T, \quad (99)$$

$$\mathbf{D}_{x_2} = \sigma^2 \mathbf{F}_2 \mathbf{W}^{ot} \mathbf{F}_2^T. \quad (100)$$

$$\mathbf{D}_{x_1 x_2} = \sigma^2 \mathbf{F}_1 \mathbf{W}^{ot} \mathbf{F}_2^T \quad (101)$$

$$\mathbf{D}_{x_2 x_1} = \sigma^2 \mathbf{F}_2 \mathbf{W}^{ot} \mathbf{F}_1^T \quad (102)$$

Introducing the vector of all transformation parameters

$$\mathbf{x}_{1,2} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (103)$$

Thus the variance matrix of  $\mathbf{x}_{1,2}$  is

$$\mathbf{D}_{\mathbf{x}_{1,2}} = \begin{bmatrix} \mathbf{D}_{x_1} & \mathbf{D}_{x_1 x_2} \\ \mathbf{D}_{x_2 x_1} & \mathbf{D}_{x_2} \end{bmatrix} \quad (104)$$

### Computation and precision estimation of seven parameters from dual quaternion

Traditionally the similarity 3D coordinate transformation model i.e. Equation (41) includes seven parameters i.e. one scale, three rotation angles and three translations. Thus it is necessary to compute the seven parameters and estimate their precisions. Firstly one needs to construct the function of the seven parameters with dual quaternion-based transformation parameter i.e.  $\mathbf{x}_{1,2}$ . And then compute the variance or covariance matrices of the seven parameters according to the variance or covariance matrices propagation principle.

Utilizing Eqs. (42) and (21), the rotation angles  $\theta_x, \theta_y, \theta_z$  are computed by

$$\theta_x = -\tan^{-1} \left( \frac{2(r_2 r_3 + r_1 r_4)}{r_4^2 - r_1^2 - r_2^2 + r_3^2} \right) \quad (105)$$

$$\theta_y = \sin^{-1}(2(r_1 r_3 - r_2 r_4)) \quad (106)$$

$$\theta_z = -\tan^{-1} \left( \frac{2(r_1 r_2 + r_3 r_4)}{r_4^2 + r_1^2 - r_2^2 - r_3^2} \right) \quad (107)$$

### Introducing

$$s_x = \sin \theta_x, \quad s_y = \sin \theta_y, \quad s_z = \sin \theta_z, \quad c_x = \cos \theta_x, \quad c_y = \cos \theta_y, \quad c_z = \cos \theta_z \quad (108)$$

one obtains the partial derivatives of  $\theta_x$  with respect to  $r_1$  by Eq. (105).

$$\begin{aligned} \frac{\partial \theta_x}{\partial r_1} &= -\frac{1}{1 + \left( \frac{2(r_2 r_3 + r_1 r_4)}{r_4^2 - r_1^2 - r_2^2 + r_3^2} \right)^2} \\ &= -\frac{2r_4(r_4^2 - r_1^2 - r_2^2 + r_3^2) + 2r_1 \cdot 2(r_2 r_3 + r_1 r_4)}{(r_4^2 - r_1^2 - r_2^2 + r_3^2)^2} \\ &= -\frac{2r_4(r_4^2 - r_1^2 - r_2^2 + r_3^2) + 2r_1 \cdot 2(r_2 r_3 + r_1 r_4)}{(r_4^2 - r_1^2 - r_2^2 + r_3^2)^2 + (2(r_2 r_3 + r_1 r_4))^2} \quad (109) \\ &= -\frac{2r_4(c_x c_y) + 2r_1(-s_x c_y)}{(c_x c_y)^2 + (-s_x c_y)^2} \\ &= -\frac{2(r_4 c_x - r_1 s_x)}{c_y} \end{aligned}$$

where the dots  $\cdot$  denote computation of multiplication. Similarly one obtains

$$\frac{\partial \theta_x}{\partial r_2} = -\frac{2(r_3 c_x - r_2 s_x)}{c_y} \quad (110)$$

$$\frac{\partial \theta_x}{\partial r_3} = -\frac{2(r_2 c_x + r_3 s_x)}{c_y} \quad (111)$$

$$\frac{\partial \theta_x}{\partial r_4} = -\frac{2(r_1 c_x + r_4 s_x)}{c_y} \quad (112)$$

Further one can get the following partial derivatives of  $\theta_y, \theta_z$  with respect to  $r_1, r_2, r_3, r_4$  by Eqs. (106) and (107) respectively.

$$\frac{\partial \theta_y}{\partial r_1} = \frac{2r_3}{c_y} \quad (113)$$

$$\frac{\partial \theta_y}{\partial r_2} = \frac{-2r_4}{c_y} \quad (114)$$

$$\frac{\partial \theta_y}{\partial r_3} = \frac{2r_1}{c_y} \quad (115)$$

$$\frac{\partial \theta_y}{\partial r_4} = \frac{-2r_2}{c_y} \quad (116)$$

$$\frac{\partial \theta_z}{\partial r_1} = -\frac{2(r_2 c_z + r_1 s_z)}{c_y} \quad (117)$$

**Table 1** The dual quaternion algorithm of the WTLS 3D coordinate transformation

*Input and initialization:* input coordinates of control points, construct pure imaginary quaternions  $\mathbf{p}_i^t, \mathbf{p}_i^o$ , and vectors  $\mathbf{p}^t, \mathbf{p}^o$ . Input weight matrix  $\mathbf{W}^t$  and  $\mathbf{W}^o$ , set the initial value of  $x_1$  as  $[1 \ 0 \ 0 \ 0 \ 0]^T$ ,  $x_2$  as  $[0 \ 0 \ 0 \ 0]^T$ , and the initial value of  $\mathbf{e}^o$  as  $\underbrace{[0 \ 0 \ \dots \ 0]}_{4n}^T$

*Iterative computation and precision estimation:* Step 1. Calculate  $\mathbf{A}^{1j}, \mathbf{A}^{2j}, \mathbf{B}^j, \mathbf{C}^j$  by Eqs. (55, 56, 57 and 60),  $\mathbf{E}$  by Eq. (61),  $\mathbf{C}_{x1} \mathbf{C}_{x2} \mathbf{W}_x$  by Eqs. (64, 65, 66),  $\mathbf{W}^{ot}$  by Eq. (78),  $\mathbf{N}_{11}, \mathbf{N}_{12}$  by Eqs. (79, 80),  $\mathbf{N}_{22}, \mathbf{N}_{21}, \mathbf{N}_{33}, \mathbf{D}_{11}, \mathbf{D}_{10}$  by Eqs. (82, 83, 84, 85, 86),  $\mathbf{D}_{21}, \mathbf{D}_{20}$  by Eqs. (88, 89)

Step 2. Compute  $\mathbf{k}_2$  by Eq. (91),  $dx_1, dx_2$  by Eqs. (92, 93),  $\mathbf{k}_1$  by Eq. (77)  $\mathbf{e}^o, \mathbf{e}^t$  by Eqs. (75, 76). If the stop condition i.e. Equation (96) of iteration is satisfied ( $\tau$  is set  $1.0 \times 10^{-14}$  in the paper), turn to Step 3, otherwise turn to Step 1 with the new value  $x_1 = x_1 + dx_1, x_2 = x_2 + dx_2$

Step 3. Compute  $\mathbf{R}$  by Eq. (21) and rotation angles  $\theta_x, \theta_y, \theta_z$  by Eqs. (105, 106, 107). Compute  $\mathbf{t}$  and  $\mathbf{t}$  by Eqs. (39) and (38)

Step 4. Estimate the variance factor of unit weight  $\sigma$  by Eq. (98), estimate the covariance matrix of  $x_{1,2}$  by Eq. (104) and the covariance matrix of  $y$  by Eq. (128)

*Output:* Output  $x_{1,2}, \mathbf{R}$  (if needed),  $y, \mathbf{e}^o, \mathbf{e}^t$ . Output the variance factor of unit weight  $\sigma$  and the covariance matrices of  $\mathbf{D}_{x_{1,2}}$  and  $\mathbf{D}_y$

$$\frac{\partial \theta_z}{\partial r_2} = -\frac{2(r_1 c_z - r_2 s_z)}{c_y} \tag{118}$$

$$\frac{\partial \theta_z}{\partial r_3} = -\frac{2(r_4 c_z - r_3 s_z)}{c_y} \tag{119}$$

$$\frac{\partial \theta_z}{\partial r_4} = -\frac{2(r_3 c_z + r_4 s_z)}{c_y} \tag{120}$$

Next, the translations can be computed by Eqs. (38) and (39) as

$$\begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix} = 2\mathbf{W}(\mathbf{r})^T \mathbf{s} = \begin{bmatrix} r_4 s_1 - r_3 s_2 - r_2 s_3 - r_1 s_4 \\ r_3 s_1 + r_4 s_2 - r_1 s_3 - r_2 s_4 \\ -r_2 s_1 + r_1 s_2 + r_4 s_3 - r_3 s_4 \\ r_1 s_1 + r_2 s_2 + r_3 s_3 + r_4 s_4 \end{bmatrix} \tag{121}$$

It is easy to deduce the following partial derivatives as

$$\begin{aligned} \frac{\partial t_x}{\partial r_1} &= -s_4, \quad \frac{\partial t_x}{\partial r_2} = -s_3, \quad \frac{\partial t_x}{\partial r_3} = -s_2, \quad \frac{\partial t_x}{\partial r_4} \\ &= s_1, \quad \frac{\partial t_x}{\partial s_1} = r_4, \quad \frac{\partial t_x}{\partial s_2} = -r_3, \quad \frac{\partial t_x}{\partial s_3} = -r_2, \quad \frac{\partial t_x}{\partial s_4} = -r_1 \end{aligned} \tag{122}$$

$$\begin{aligned} \frac{\partial t_y}{\partial r_1} &= -s_3, \quad \frac{\partial t_y}{\partial r_2} = -s_4, \quad \frac{\partial t_y}{\partial r_3} = s_1, \quad \frac{\partial t_y}{\partial r_4} = s_2, \quad \frac{\partial t_y}{\partial s_1} \\ &= r_3, \quad \frac{\partial t_y}{\partial s_2} = r_4, \quad \frac{\partial t_y}{\partial s_3} = -r_1, \quad \frac{\partial t_y}{\partial s_4} = -r_2 \end{aligned} \tag{123}$$

$$\begin{aligned} \frac{\partial t_z}{\partial r_1} &= s_2, \quad \frac{\partial t_z}{\partial r_2} = -s_1, \quad \frac{\partial t_z}{\partial r_3} = -s_4, \quad \frac{\partial t_z}{\partial r_4} = s_3, \quad \frac{\partial t_z}{\partial s_1} \\ &= -r_2, \quad \frac{\partial t_z}{\partial s_2} = r_1, \quad \frac{\partial t_z}{\partial s_3} = r_4, \quad \frac{\partial t_z}{\partial s_4} = -r_3 \end{aligned} \tag{124}$$

Introducing

$$y = [\lambda \ \theta_x \ \theta_y \ \theta_z \ t_x \ t_y \ t_z]^T \tag{125}$$

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \theta_x}{\partial r_1} & \frac{\partial \theta_x}{\partial r_2} & \frac{\partial \theta_x}{\partial r_3} & \frac{\partial \theta_x}{\partial r_4} & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \theta_y}{\partial r_1} & \frac{\partial \theta_y}{\partial r_2} & \frac{\partial \theta_y}{\partial r_3} & \frac{\partial \theta_y}{\partial r_4} & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial r_1}{\partial r_1} & \frac{\partial r_2}{\partial r_2} & \frac{\partial r_3}{\partial r_3} & \frac{\partial r_4}{\partial r_4} & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial t_x}{\partial r_1} & \frac{\partial t_x}{\partial r_2} & \frac{\partial t_x}{\partial r_3} & \frac{\partial t_x}{\partial r_4} & \frac{\partial t_x}{\partial s_1} & \frac{\partial t_x}{\partial s_2} & \frac{\partial t_x}{\partial s_3} & \frac{\partial t_x}{\partial s_4} \\ 0 & \frac{\partial t_y}{\partial r_1} & \frac{\partial t_y}{\partial r_2} & \frac{\partial t_y}{\partial r_3} & \frac{\partial t_y}{\partial r_4} & \frac{\partial t_y}{\partial s_1} & \frac{\partial t_y}{\partial s_2} & \frac{\partial t_y}{\partial s_3} & \frac{\partial t_y}{\partial s_4} \\ 0 & \frac{\partial t_z}{\partial r_1} & \frac{\partial t_z}{\partial r_2} & \frac{\partial t_z}{\partial r_3} & \frac{\partial t_z}{\partial r_4} & \frac{\partial t_z}{\partial s_1} & \frac{\partial t_z}{\partial s_2} & \frac{\partial t_z}{\partial s_3} & \frac{\partial t_z}{\partial s_4} \end{bmatrix} \tag{126}$$

one has

**Table 2** 3D coordinates in the original system

Station Name	$x^o(m)$	$y^o(m)$	$z^o(m)$	$\sigma^{o2}(m^2)$
Solitude	4,157,222.543	664,789.307	4,774,952.099	0.14330
Buoch Zeil	4,149,043.336	688,836.443	4,778,632.188	0.15510
Hohenneuffen	4,172,803.511	690,340.078	4,758,129.701	0.15030
Kuehlenberg	4,177,148.376	642,997.635	4,760,764.800	0.14000
Ex Mergelaec	4,137,012.190	671,808.029	4,791,128.215	0.14590
Ex Hof Asperg	4,146,292.729	666,952.887	4,783,859.856	0.14690
Ex Kaisersbach	4,138,759.902	702,670.738	4,785,552.196	0.12200

**Table 3** 3D coordinates in the target system

Station name	$x^t(m)$	$y^t(m)$	$z^t(m)$	$\sigma^{t2}(m^2)$
Solitude	4,157,870.237	664,818.678	4,775,416.524	0.01030
Buoch Zeil	4,149,691.049	688,865.785	4,779,096.588	0.00380
Hohenneuffen	4,173,451.354	690,369.375	4,758,594.075	0.00060
Kuehlenberg	4,177,796.064	643,026.700	4,761,228.899	0.01140
Ex Mergelaec	4,137,659.549	671,837.337	4,791,592.531	0.00680
Ex Hof Asperg	4,146,940.228	666,982.151	4,784,324.099	0.00002
Ex Kaisersbach	4,139,407.506	702,700.227	4,786,016.645	0.00410

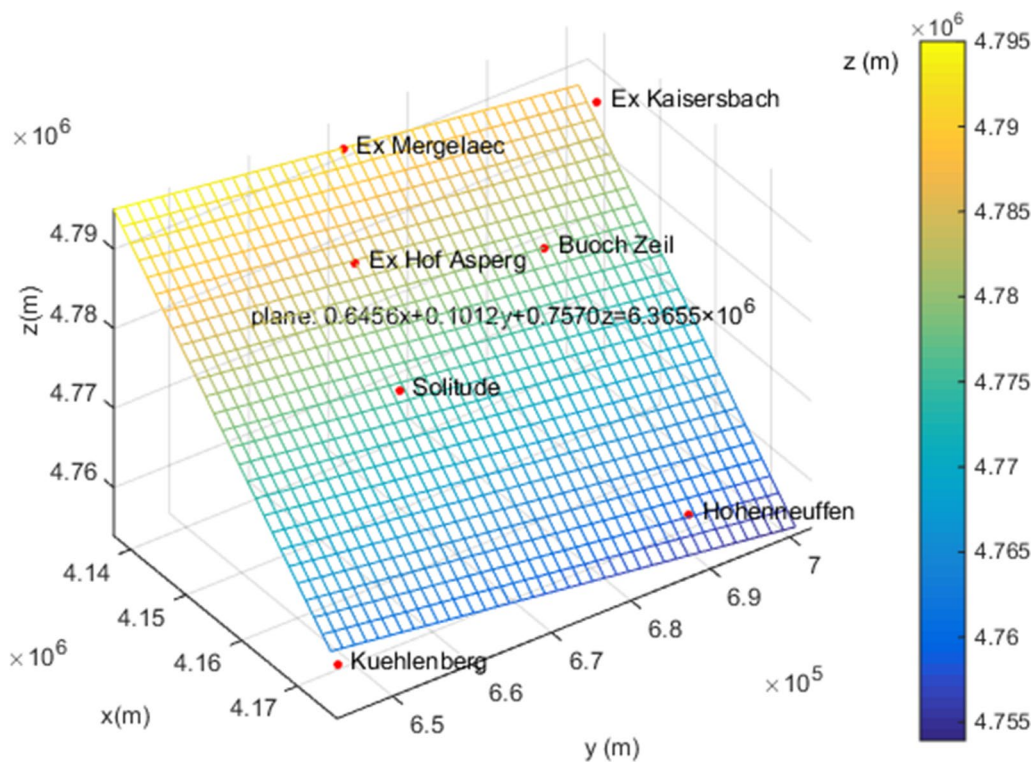


Fig. 3 Distribution of the stations

Table 4 Computed scale and dual quaternion and their precisions by DQA in the geodetic case

Scale	Quaternion <i>r</i>		Quaternion <i>s</i>		
$\lambda$	1.00000561108964±0.000000690171	$r_1$	0.00000241852729±0.00000074327410	$s_1$	320.92010787499300±3.95932109025866
		$r_2$	-0.00000217217855±0.00000084931086	$s_2$	34.23769229231280±5.25258027604197
		$r_3$	-0.00000238984738±0.00000065898365	$s_3$	208.10698182051300±3.75202102184052
		$r_4$	0.9999999999186±0.00000000000392	$s_4$	-0.00020443973190±0.00021876004598

Table 5 Computed seven transformation parameters and their precisions by DQA in the geodetic case

Scale	
$\lambda$	1.00000561108964±0.00000000690171
Rotation angles (°)	
$\theta_x$	-0.99771626707544±0.30662312364179
$\theta_y$	0.89608559290677±0.35036577554091
$\theta_z$	0.98588498193093±0.27185075393370
Translation (m)	
$t_x$	641.83948±7.91832
$t_y$	68.47284±10.50456
$t_z$	416.21552±7.50345

$$dy = Jdx_{1,2} \tag{127}$$

According to the variance or covariance matrices propagation principle, one gets

$$D_y = JD_{x_{1,2}}J^T \tag{128}$$

**Dual quaternion algorithm**

The proposed algorithm namely the Dual Quaternion Algorithm is summarized in Table 1.

**Table 6** Computed scale, rotation matrix and variance component by DQA and QA in the geodetic case

	DQA			QA		
Scale						
$\lambda$	1.00000561108964 ± 0.00000000690171			1.00000561108951 (no precision provided)		
Rotation matrix						
<b>R</b>	0.999999999979	0.000004779684	- 0.000004344369	0.999999999979	0.000004779684	- 0.000004344369
	- 0.000004779705	0.999999999977	- 0.000004837044	- 0.000004779705	0.999999999977	- 0.000004837044
	0.000004344346	0.000004837065	0.999999999979	0.000004344345	0.000004837065	0.999999999979
Variance component (m)						
$\sigma^2$	0.039043823461			0.039043823416		

**Table 7** Computed scaled quaternion, translations by DQA and QA, and decrease percent of standard deviations from QA to DQA

	DQA	QA	Decrease percent (%)
Scaled quaternion			
$q_0$	1.00000280553274 ± 0.00000000345052	1.00000280553268 ± 0.00000054146075	99.36
$q_1$	0.00000241853408 ± 0.00000074327618	0.00000241853407 ± 0.00000074326600	0.00
$q_2$	- 0.00000217218465 ± 0.00000084931325	- 0.00000217218465 ± 0.00000084028102	- 1.07
$q_3$	- 0.00000238985409 ± 0.00000065898550	- 0.00000238985409 ± 0.00000065903067	0.01
Translation (m)			
$t_x$	641.83948 ± 7.91832	641.83948 ± 9.03275	12.34
$t_y$	68.47284 ± 10.50456	68.47284 ± 10.53177	0.26
$t_z$	416.21552 ± 7.50345	416.21552 ± 9.04950	17.08

**Table 8** Predicted errors of coordinates in the original and target systems by DQA and QA in the geodetic case

Station name	Errors in target system (m)			Errors in original system (m)		
	$x^t$	$y^t$	$z^t$	$x^o$	$y^o$	$z^o$
Solitude	0.0064	0.0091	0.0094	- 0.0885	- 0.1261	- 0.1313
Buoch Zeil	0.0015	- 0.0012	0.0003	- 0.0593	0.0489	- 0.0140
Hohenneuffen	- 0.0002	- 0.0004	0.0000	0.0386	0.0887	0.0071
Kuehlenberg	0.0015	- 0.0017	- 0.0065	- 0.0181	0.0203	0.0803
Ex Mergelaec	- 0.0040	0.0006	- 0.0002	0.0860	- 0.0138	0.0049
Ex Hof Asperg	0.0000	0.0000	0.0000	0.0105	- 0.0069	0.0542
Ex Kaisersbach	- 0.0009	0.0001	0.0001	0.0257	- 0.0035	- 0.0022

**Table 9** Transformation residuals by DQA and QA in the geodetic case

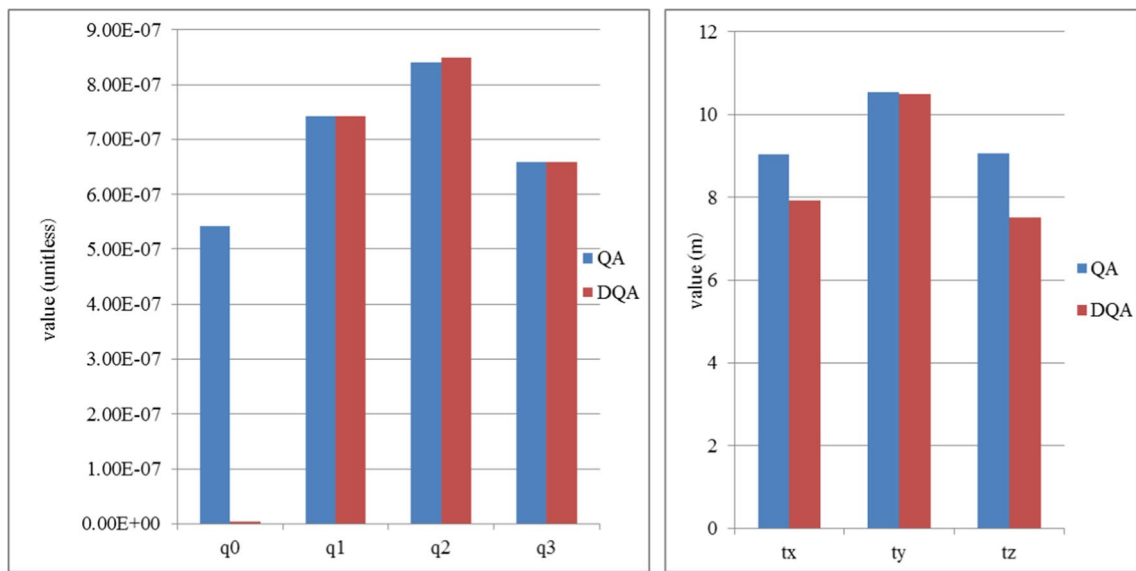
Station name	$x(m)$	$y(m)$	$z(m)$
Solitude	0.0948	0.1352	0.1407
Buoch Zeil	0.0608	- 0.0501	0.0143
Hohenneuffen	- 0.0388	- 0.0891	- 0.0072
Kuehlenberg	0.0195	- 0.0219	- 0.0868
Ex Mergelaec	- 0.09	0.0144	- 0.0052
Ex Hof Asperg	- 0.0105	0.0069	- 0.0542
Ex Kaisersbach	- 0.0266	0.0036	0.0022

**Experiments and discussion**

Two numerical cases including actual geodetic datum transformation and a simulated case are demonstrated to validate the presented algorithm. The former case involves small rotation angles while the latter one involves large rotation angles. The results and analyses are as follows.

**Actual geodetic datum transformation case**

Mostly geodetic datum transformation involves very small rotation angles (not bigger than 1°). This case



**Fig. 4** Comparison of estimated standard deviations by DQA and QA

data is chosen from Grafarend and Awange (2003), which is discussed in Mercan et al. (2018), Zeng et al. (2022b), etc. The original (local) and target (global) 3D coordinates and their variances of seven control points are listed in Tables 2 and 3. The distribution of seven control points in the original coordinate system is illustrated in Fig. 3. They are almost located in a plane that is represented by the grids in the figure (the biggest distance from the control point to the plane is 68.29 m, corresponding to station Solitude, which is quite small compared to about 50 km × 50 km of the control point area. The biggest distance between the control points is 75,159.625 m). And the fitted plane equation is  $0.6456x + 0.1012y + 0.7570z = 6.3655 \times 10^6$ . Supposed that the components of  $p_i^o$  and  $p_i^t$  are isotropic and not correlative, the covariance matrices  $D^o$  and  $D^t$  of  $p^o$  and  $p^t$  are both diagonal as

$$D^o = \text{diag}(\sigma_1^{o2}, \sigma_1^{o2}, \sigma_1^{o2}, \sigma_1^{o2}, \dots, \sigma_n^{o2}, \sigma_n^{o2}, \sigma_n^{o2}, \sigma_n^{o2}) \tag{129}$$

$$D^t = \text{diag}(\sigma_1^{t2}, \sigma_1^{t2}, \sigma_1^{t2}, \sigma_1^{t2}, \dots, \sigma_n^{t2}, \sigma_n^{t2}, \sigma_n^{t2}, \sigma_n^{t2}) \tag{130}$$

where n is the number of point, i.e. 7 in this case. The weight matrices  $W^o$  and  $W^t$  are constructed as

$$W^o = (D^o)^{-1} \tag{131}$$

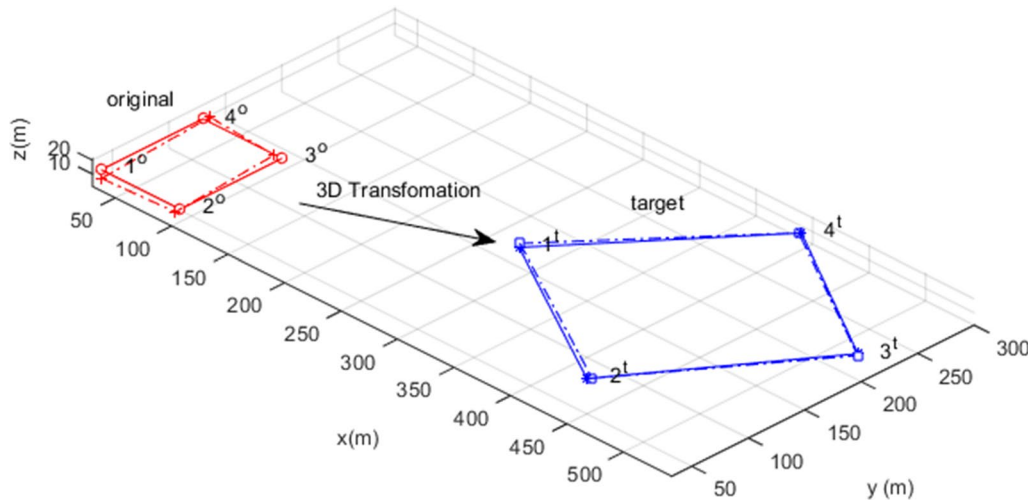
$$W^t = (D^t)^{-1} \tag{132}$$

The proposed algorithm namely DQA is employed to calculate the transformation parameters. The estimated scale and unit dual quaternion by DQA are listed in Table 4. The computed traditional seven transformation parameters and their precisions by DQA are listed in Table 5. In order to compare with the quaternion algorithm (QA) in Mercan et al. (2018), the paper computes the scaled quaternion and their precisions from the estimated scale and dual quaternion by DQA as follows.

$$q = [q_1 \ q_2 \ q_3 \ q_0]^T = \sqrt{\lambda}r \tag{133}$$

**Table 10** Coordinates and point-wise weights of control points

Point no.	Weight	Original coordinates (m)			Target coordinates (m)		
		$x^o$	$y^o$	$z^o$	$x^t$	$y^t$	$z^t$
1	1	30	40	10	290	150	15
2	2	100	40	10	420	80	2
3	2.5	100	130	10	540	200	20
4	4	30	130	10	390	300	5



**Fig. 5** Distribution of the control points in the original and target coordinate systems

**Table 11** Computed scale and dual quaternion and their precisions by DQA in the simulated case

Scale	Quaternion $r$		Quaternion $s$		
$\lambda$	$2.13618931887411 \pm 0.15248995183090$	$r_1$	$0.01015942751985 \pm 0.04893072388863$	$s_1$	$75.09345366954858 \pm 11.96977789113642$
		$r_2$	$-0.02255774253599 \pm 0.05308425209055$	$s_2$	$80.96103957803537 \pm 12.02106203728454$
		$r_3$	$-0.29771767907456 \pm 0.03411742353052$	$s_3$	$-14.21810455226187 \pm 19.72177547831338$
		$r_4$	$0.95433333686433 \pm 0.01071519188167$	$s_4$	$-3.32126017108111 \pm 7.23696213343644$

**Table 12** Computed seven transformation parameters and their precisions by DQA in the simulated case

Scale	
$\lambda$	$2.13618931887411 \pm 0.15248995183090$
Rotation angles ( $^\circ$ )	
$\theta_x$	$-1.88222617859100 \pm 5.88105385300878$
$\theta_y$	$2.12076778302949 \pm 5.82194309812054$
$\theta_z$	$34.68692971526144 \pm 4.09850995531577$
Translation (m)	
$t_x$	$192.24438 \pm 20.2709$
$t_y$	$109.95340 \pm 20.1299$
$t_z$	$-24.08230 \pm 29.0657$

where the definition of quaternion in the form of vector here is different from Mercan et al. (2018). Further, one obtains

$$dq = \begin{bmatrix} \frac{r_1}{2\sqrt{\lambda}} & \sqrt{\lambda} & 0 & 0 & 0 \\ \frac{r_2}{2\sqrt{\lambda}} & 0 & \sqrt{\lambda} & 0 & 0 \\ \frac{r_3}{2\sqrt{\lambda}} & 0 & 0 & \sqrt{\lambda} & 0 \\ \frac{r_4}{2\sqrt{\lambda}} & 0 & 0 & 0 & \sqrt{\lambda} \end{bmatrix} \begin{bmatrix} d\lambda \\ dr_1 \\ dr_2 \\ dr_3 \\ dr_4 \end{bmatrix} = J_q dx_1 \tag{134}$$

According to the variance or covariance matrices propagation principle, one gets

$$D_q = J_q D_{x_1} J_q^T \tag{135}$$

The computed results by QA and DQA are listed in Tables 6, 7, 8 and 9. And the result by QA is directly from Mercan et al. (2018). From Table 6, it is seen that the computed scale, rotation matrix and variance component are identical regardless of the rounding errors. However the QA does not provide the precision of scale while DQA does. Table 7 shows that the computed scaled quaternion and translations are consistent if the rounding errors are ignored, but the estimated precisions are different greatly. As far as  $q_0$  is concerned, the estimated standard deviation is improved from  $10^{-7}$  by QA to  $10^{-9}$  by DQA. In other words, the decrease percent of standard deviations is 99.36%. For  $q_1, q_2, q_3$ , the estimated precisions by two algorithms are slightly different. As far as the translations are concerned, the estimated standard deviations decrease 12.34%, 0.26%, 17.08% for  $t_x, t_y, t_z$  respectively from QA to DQA. The average decrease percent is 18.28% for the scaled quaternion and translations (seven parameters). The comparison of estimated

**Table 13** Computed scale, rotation matrix and variance component by DQA and QA in the simulated case

	DQA			QA		
Scale						
$\lambda$	2.13618931887411 ± 0.15248995183090			2.13618931887411 (no precision provided)		
Rotation matrix						
<b>R</b>	0.821710663636	0.567785464729	- 0.049104493777	0.821710663636	0.567785464729	- 0.049104493777
	- 0.568702159730	0.822521939198	- 0.005959283225	- 0.568702159730	0.822521939198	- 0.005959283225
	0.037005929049	0.032822638237	0.998775868568	0.037005929049	0.032822638237	0.998775868568
Variance component (m)						
$\sigma^2$	116.012049766184			116.012049766184		

**Table 14** Computed scaled quaternion, translations by DQA and QA in the simulated case

	DQA	QA
Scaled quaternion		
$q_0$	1.39482577632278 ± 0.05218939548330	1.39482577632278 ± 0.05218939548330
$q_1$	0.01484872300902 ± 0.07151768293004	0.01484872300902 ± 0.07151768293004
$q_2$	- 0.03296973869553 ± 0.07759531835570	- 0.03296973869553 ± 0.07759531835570
$q_3$	- 0.43513547813872 ± 0.05222766986151	- 0.43513547813872 ± 0.05222766986151
Translation (m)		
$t_x$	192.24438 ± 20.2709	192.24437 ± 20.2709
$t_y$	109.95340 ± 20.1299	109.95340 ± 20.1299
$t_z$	- 24.08230 ± 29.0657	- 24.08230 ± 29.0657

**Table 15** Predicted errors of coordinates in the original and target systems by DQA in the simulated case

Point no.	Errors in target system (m)			Errors in original system (m)		
	$x^t$	$y^t$	$z^t$	$x^o$	$y^o$	$z^o$
1	- 0.4262	1.1391	2.2595	1.9534	- 1.6429	- 4.8511
2	0.8548	3.8425	- 1.0719	3.2523	- 7.7132	2.4255
3	2.8032	- 3.0124	1.0293	- 8.6615	1.8208	- 1.9404
4	- 2.0729	- 0.3233	- 0.6723	3.2989	3.1293	1.2128

**Table 16** Covariance matrix  $D_{x_{1,2}}$  obtained by DQA

0.0233	0.0000	0.0000	0.0000	0.0000	- 1.0498	- 0.9073	- 0.1395	- 0.0538
0.0000	0.0024	- 0.0003	0.0000	0.0000	0.0096	0.0270	- 0.6107	- 0.3483
0.0000	- 0.0003	0.0028	0.0000	0.0001	- 0.0376	- 0.0007	0.8637	0.0582
0.0000	0.0000	0.0000	0.0012	0.0004	0.3023	- 0.3265	- 0.0146	0.0018
0.0000	0.0000	0.0001	0.0004	0.0001	0.0933	- 0.1021	0.0224	0.0056
- 1.0498	0.0096	- 0.0376	0.3023	0.0933	143.2756	- 43.8112	- 6.7913	2.5779
- 0.9073	0.0270	- 0.0007	- 0.3265	- 0.1021	- 43.8112	144.5059	- 0.0372	- 3.4099
- 0.1395	- 0.6107	0.8637	- 0.0146	0.0224	- 6.7913	- 0.0372	388.9484	96.0516
- 0.0538	- 0.3483	0.0582	0.0018	0.0056	2.5779	- 3.4099	96.0516	52.3736



**Table 17** Covariance matrix  $D_y$  obtained by DQA

0.0233	0.0000	0.0000	0.0000	-2.5365	-1.1062	-0.3641
0.0000	0.0105	-0.0016	-0.0002	-0.2630	-0.2182	2.4952
0.0000	-0.0016	0.0103	0.0003	0.2950	-0.1054	-1.7349
0.0000	-0.0002	0.0003	0.0051	-0.5107	1.1934	-0.0682
-2.5365	-0.2630	0.2950	-0.5107	410.9082	0.8242	-57.9322
-1.1062	-0.2182	-0.1054	1.1934	0.8242	405.2118	-12.6089
-0.3641	2.4952	-1.7349	-0.0682	-57.9322	-12.6089	844.8156

**Table 18** Transformation residuals by DQA and QA in the simulated case

Point no.	x(m)	y(m)	z(m)
1	-2.3712	6.3371	12.5704
2	4.7557	21.3770	-5.9632
3	15.5950	-16.7587	5.7264
4	-11.5319	-1.7986	-3.7400

standard deviations by DQA and QA is depicted in Fig. 4. In a word, the estimated precisions are improved significantly comparing DQA to QA.

From Tables 8 and 9, it is seen that the predicted errors of coordinates in the original and target systems by DQA and QA are the same; the transformation residuals are identical too for DQA and QA. Thus the DQA and QA are correct.

**Simulated case**

The adopted data in this case is originally from Felus and Burtch (2009). The data is synthesized and rounded based on a surface fitting experiment. In the experiment, the surface is surveyed in two different coordinate systems. And four control points on the surface are identified and their coordinates in the original and target coordinate systems are listed in Table 10. The area of control points are 70 m×90 m in the original coordinate system. It is assumed that all points are not dependent on each other and the coordinate components of each point are isotropic and not correlated. And the point-wise weights are given in Table 10 too. The distribution of control points are depicted in Fig. 5. The superscript o or t denotes original coordinate system or target coordinate system respectively in the figure. The red circles represent the positions of control point regardless of the errors in the original system. And red solid lines are drawn between them. The blue asterisks represent the positions of control point regardless of the errors in the target system. And blue solid lines are drawn between them. It is easily seen that the rotation angle around the z axis is big, while

rotation angles around the x and y axes are small and the scale is much greater than 1.

The DQA and QA are utilized to estimate the transformation parameters. The calculated scale and unit dual quaternion by DQA are listed in Table 11. The estimated traditional seven transformation parameters and their precisions by DQA are listed in Table 12. The obtained results by QA and their counterparts by DQA are given in Tables 13, 14. Predicted errors of coordinates and estimated covariance matrix  $D_{x_{1,2}}$ ,  $D_y$  by DQA are listed in Tables 15, 16, 17 respectively. The transformation residuals by DQA and QA are listed in Table 18. The adjusted coordinates of points are depicted in Fig. 5. The red crosses represent the positions of control point after adjustment in the original system. And red dash-dotted lines are drawn between them. The blue squares represent the positions of control point after adjustment in the target system. And blue dash-dotted lines are drawn between them.

Table 12 shows that the standard deviations of seven transformation parameters are relatively big, especially those of the rotation angle which are worthy of attention are around 5°. It is seen that from Table 13, the computed scale, rotation matrix and variance component by DQA and QA are identical, however QA does not offer the standard deviation of scale while DQA does. Table 14 shows that the computed scaled quaternion, translations by DQA and QA are consistent regardless of the rounding error. Table 15 or Fig. 5 shows the predicted errors of coordinates are nearly 10 m, which are relatively big. The reason is that the original and target coordinates errors of control points are relatively big from the viewpoint of variance component in Table 13. It is shown that the transformation residuals by DQA and QA are same from Table 18.

To sum up, DQA and QA obtain the consistent results. However it is worthy of notice that the precisions of adjusted parameters are not high for this case with relatively big errors in the coordinates.

## Conclusions

A unit quaternion is widely used to represent the 3D rotation matrix, and the paper presents its own derivation from the geometric meaning of 3D rotation of a point around an axis. Further, a unit dual quaternion is employed to represent not only the 3D rotation matrix but also the translation vector, and the paper gives the derivation of representation of the translation vector by the unit dual quaternion. Based on the unit dual quaternion, the 3D similarity coordinate transformation in the errors-in-variables (EIV) model is formulated. By means of linearization by Taylor's formula and Lagrangian extremum principle with constraints, the Dual Quaternion Algorithm (DQA) is proposed. The algorithm is able to output not only the computed parameters but also the full precision information of computed parameters. In addition, the traditional seven parameters and their estimated precisions are output.

The two numerical experiments including a real-world geodetic datum transformation case and a simulated case from surface fitting show that the presented algorithm i.e. DQA is not sensitive to the initial values of transformation, in other words the arbitrary initial value set in this paper is feasible regardless of the size of rotation angles. DQA obtains the consistent results with the quaternion algorithm i.e. QA from Mercan et al. (2018), no matter how big the rotation angles are, and whether the relative errors of coordinates (pseudo-observations) are large or small. On the other hand, the DQA has some advantages over the QA. The key one is the improvement of estimated precisions of transformation parameters, i.e. the average decrease percent of standard deviations is 18.28%, and biggest decrease percent is 99.36% for the scaled quaternion and translations in the geodetic datum transformation case. Another advantage is the DQA fulfills the computation and precision estimation of traditional seven transformation parameters (which still are widely applied yet) from dual quaternion, and even could perform the computation and precision estimation of the scaled quaternion.

## Acknowledgements

The first author thanks his research team for valuable suggestions.

## Author contributions

HZ was the coordinator of the research group, and responsible for the derivation of relevant theories in the paper; HZ conceived the original ideas; ZW, SL, and JW participated in deriving formulae; HZ, JL and XL implemented case study; HZ and XL worked on preparing the original draft; HZ and JL reviewed and edited the manuscript; HZ supervised the research work. All the authors read and approved the final manuscript.

## Funding

The study is supported jointly by National Natural Science Foundation of China (Grant No. 42074005), the Open Foundation of Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University (Grant No. 2023KSD11), and the Open Foundation

of National Field Observation and Research Station of Landslides in the Three Gorges Reservoir Area of Yangtze River, China Three Gorges University (Grant No. 2018KTL14).

## Availability of data and materials

The case data were collected from Grafarend and Awange (2003), and Felus and Burtch (2009).

## Declarations

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University, Yichang 443002, China. <sup>2</sup>National Field Observation and Research Station of Landslides in the Three Gorges Reservoir Area of Yangtze River, China Three Gorges University, Yichang 443002, China. <sup>3</sup>College of Civil Engineering & Architecture, China Three Gorges University, Yichang 443002, China. <sup>4</sup>Hubei Jiefan Technology Corporation, Yichang 443000, China.

Received: 24 October 2023 Accepted: 18 January 2024

Published online: 29 January 2024

## References

- Arun KS, Huang TS, Blostein SD (1987) Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal Mach Intell* 9:698–700
- Aydin C, Mercan H, Uygur SO (2018) Increasing numerical efficiency of iterative solution for total least-squares in datum transformations. *Stud Geophys Geod* 62:223–242
- Bektas S (2022) A new algorithm for 3D similarity transformation with dual quaternion. *Arab J Geosci* 15(14):1–9. <https://doi.org/10.1007/s12517-022-10457-z>
- Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 14(2):239–256
- Burša M (1967) On the possibility of determining the rotation elements of geodetic reference systems on the basis of satellite observations. *Stud Geophys Geod* 11(4):390–396
- Chang G (2015) On least-squares solution to 3D similarity transformation problem under Gauss-Helmert model. *J Geod* 89:573–576
- Chang G, Xu T, Wang Q, Liu M (2017) Analytical solution to and error analysis of the quaternion based similarity transformation considering measurement errors in both frames. *Measurement* 110:1–10. <https://doi.org/10.1016/j.measurement.2017.06.013>
- Chen Y, Shen YZ, Liu DJ (2004) A simplified model of three dimensional-datum transformation adapted to big rotation angle. *Geomat Inf Sci Wuhan Univ* 29:1101–1104
- Clifford WK (1873) Preliminary sketch of biquaternions. *Proc Lond Math Soc* 4:381–395
- Crosilla F, Beinat A (2002) Use of generalised Procrustes analysis for the photogrammetric block adjustment by independent models. *ISPRS J Photogramm Remote Sens* 56(3):195–209
- Fan L, Smethurst JA, Atkinson PM, Powrie W (2015) Error in target-based georeferencing and registration in terrestrial laser scanning. *Comput Geosci* 83:54–64
- Fang X (2015) Weighted total least-squares with constraints: a universal formula for geodetic symmetrical transformations. *J Geod* 89:459–469
- Felus YA, Burtch RC (2009) On symmetrical three-dimensional datum conversion. *GPS Solut* 13:65–74
- Gargula T, Gawronek P (2023) The Helmert transformation: a proposal for the problem of post-transformation corrections. *Adv Geod Geoinf*. <https://doi.org/10.24425/agg.2022.141920>
- Ge X, Wunderlich T (2016) Surface-based matching of 3D point clouds with variable coordinates in source and target system. *ISPRS J Photogramm Remote Sens* 111:1–12. <https://doi.org/10.1016/j.isprsjprs.2015.11.001>
- Golub GH, Van Loan CF (1980) An analysis of the total least squares problem. *SIAM J Numer Anal* 17(6):883–893

- Grafarend EW, Awange JL (2003) Nonlinear analysis of the three-dimensional datum transformation [conformal group C7(3)]. *J Geod* 77:66–76
- Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am Ser A* 4:629–642
- Horn BKP, Hilden HM, Negahdaripour S (1988) Closed-form solution of absolute orientation using orthonormal matrices. *J Opt Soc Am Ser A* 5:1127–1135
- Ioannidou S, Pantazis G (2020) Helmert transformation problem: from Euler angles method to quaternion algebra. *ISPRS Int J Geo Inf* 9:494. <https://doi.org/10.3390/ijgi9090494>
- Jitka P (2011) Application of dual quaternions algorithm for geodetic datum transformation. *Aplimat-J Appl Math* 4(2):225–236
- Krarup T (1985) Contribution to the Geometry of the Helmert transformation. Geodetic Institute, Denmark
- Kurt O (2018) An integrated solution for reducing ill-conditioning and testing the results in non-linear 3D similarity transformations. *Inverse Probl Sci Eng* 26(5):708–727
- Leick A, Van Gelder BHW (1975) On similarity transformations and geodetic network distortions based on Doppler satellite observations. Report No. 235, Dep. of Geodetic Sci., The Ohio State University, Columbus
- Li B, Shen Y, Li W (2012) The seamless model for three-dimensional datum transformation. *Sci China* 55(12):2099–2108
- Li RB, Yuan XP, Gan S, Bi R, Guo Y, Gao S (2022) A point cloud registration method based on dual quaternion description with point-linear feature constraints. *Int J Remote Sens* 43:2538–2558
- Lv Z, Sui L (2020) The BAB algorithm for computing the total least trimmed squares estimator. *J Geod* 94:110. <https://doi.org/10.1007/s00190-020-01427-y>
- Ma YQ, Liu SC, Li QZ (2020) An advanced multiple outlier detection algorithm for 3D similarity datum transformation. *Measurement* 163:107945. <https://doi.org/10.1016/j.measurement.2020.107945>
- Mahboub V (2016) A weighted least-squares solution to a 3-D symmetrical similarity transformation without linearization. *Stud Geophys Geod* 60:195–209
- Mercan H, Akyilmaz O, Aydin C (2018) Solution of the weighted symmetric similarity transformations based on quaternions. *J Geod* 92:1113–1130
- Mihajlović D, Cvijetinović Ž (2017) Weighted coordinate transformation formulated by standard least-squares theory. *Surv Rev* 49(356):328–345. <https://doi.org/10.1080/00396265.2016.1173329>
- Odziemczyk W (2020) Application of simulated annealing algorithm for 3D coordinate transformation problem solution. *Open Geosci* 12:491–502
- Păun C, Oniga E, Dragomir P (2017) Three-dimensional transformation of coordinate systems using nonlinear analysis—procrustes algorithm. *Int J Eng Sci Res Technol* 6(2):355–363
- Qin Y, Fang X, Zeng W, Wang B (2020) General total least squares theory for geodetic coordinate transformations. *Appl Sci* 10:2598. <https://doi.org/10.3390/app10072598>
- Schaffrin B, Neitzel F, Uzun S, Mahboub V (2012) Modifying cadzow's algorithm to generate the optimal TLS-solution for the structured EIV-model of a similarity transformation. *J Geod Sci* 2:98–106
- Shen YZ, Chen Y, Zheng DH (2006) A quaternion-based geodetic datum transformation algorithm. *J Geod* 80:233–239
- Soycan M, Soycan A (2008) Transformation of 3D GPS Cartesian coordinates to ED50 using polynomial fitting by robust re-weighting technique. *Surv Rev* 40(308):142–155
- Teunissen PJG (1988) The non-linear 2D symmetric Helmert transformation: an exact non-linear least-squares solution. *Bull Géodésique* 62:1–16
- Uygur SO, Aydin C, Akyilmaz O (2020) Retrieval of Euler rotation angles from 3D similarity transformation based on quaternions. *J Spat Sci*. <https://doi.org/10.1080/14498596.2020.1776170>
- Walker MW, Shao L, Volz RA (1991) Estimating 3-D location parameters using dual number quaternions. *CVGIP Image Underst* 54:358–367
- Wang YB, Wang YJ, Wu K, Yang HC, Zhang H (2014) A dual quaternion-based, closed-form pairwise registration algorithm for point clouds. *ISPRS J Photogramm Remote Sens* 94:63–69
- Xu PL (2003) A hybrid global optimization method: the multi-dimensional case. *J Comput Appl Math* 155:423–446
- Xu PL, Liu JN, Shi C (2012) Total least squares adjustment in partial errors-in-variables models: algorithm and statistical analysis. *J Geod* 86:661–675. <https://doi.org/10.1007/s00190-012-0552-9>
- Xu PL, Liu J, Shi Y (2023) Almost unbiased weighted least squares location estimation. *J Geod* 97:68. <https://doi.org/10.1007/s00190-023-01742-0>
- Závoti J, Kalmár J (2016) A comparison of different solutions of the Bursa-Wolf model and of the 3D, 7-parameter datum transformation. *Acta Geod Geophys* 51:245–256
- Zeng HE (2015) Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix. *Earth Planet Space* 67:105
- Zeng HE, Yi QL (2010) A new analytical solution of nonlinear geodetic datum transformation. In: Proceedings of the 18th International Conference on Geoinformatics
- Zeng HE, Yi QL (2011) Quaternion-based iterative solution of three-dimensional coordinate transformation problem. *J Comput* 6(7):1361–1368
- Zeng HE, Yi QL, Wu Y (2016) Iterative approach of 3D datum transformation with a non-isotropic weight. *Acta Geod Geophys* 51:557–570
- Zeng HE, Fang X, Chang G, Yang R (2018) A dual quaternion algorithm of the Helmert transformation problem. *Earth Planet Space* 70:26. <https://doi.org/10.1186/s40623-018-0792-x>
- Zeng HE, Chang G, He H, Tu Y, Sun S, Wu Y (2019) Iterative solution of Helmert transformation based on a unit dual quaternion. *Acta Geod Geophys* 54:123–141
- Zeng HE, Chang G, He H, Li K (2020) WTLS iterative algorithm of 3D similarity coordinate transformation based on Gibbs vectors. *Earth Planet Space* 72:53. <https://doi.org/10.1186/s40623-020-01179-1>
- Zeng HE, He HW, Chen LG, Chang GB, He HQ (2022a) Extended WTLS iterative algorithm of 3D similarity transformation based on Gibbs vector. *Acta Geod Geophys* 57:43–61
- Zeng HE, Wang JJ, Wang ZH, Li SY, He HQ, Chang GB, Yang RH (2022) Analytical dual quaternion algorithm of the weighted three-dimensional coordinate transformation. *Earth Planet Space* 74:170. <https://doi.org/10.1186/s40623-022-01731-1>

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.