

FULL PAPER

Open Access



Analytical dual quaternion algorithm of the weighted three-dimensional coordinate transformation

Huaien Zeng^{1,2,3*} , Junjie Wang³, Zhihao Wang³, Siyang Li³, Haiqing He⁴, Guobin Chang⁵ and Ronghua Yang⁶

Abstract

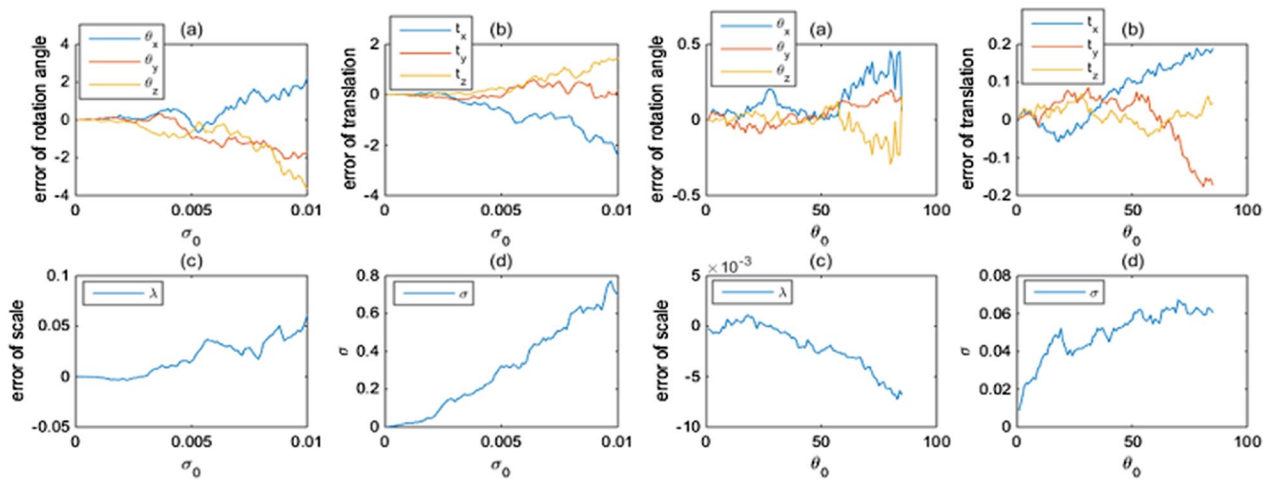
Considering that a unit dual quaternion can describe elegantly the rigid transformation including rotation and translation, the point-wise weighted 3D coordinate transformation using a unit dual quaternion is formulated. The constructed transformation model by a unit dual quaternion does not need differential process to eliminate the three translation parameters, while traditional models do. Based on the Lagrangian extremum law, the analytical dual quaternion algorithm (ADQA) of the point-wise weighted 3D coordinate transformation is proved existed and derived in detail. Four numerical cases, including geodetic datum transformation, the registration of LIDAR point clouds, and two simulated cases, are studied. This study shows that ADQA is valid as well as the modified procrustes algorithm (MPA) and the orthonormal matrix algorithm (OMA). ADQA is suitable for the 3D coordinate transformation with point-wise weight and no matter rotation angles are small or big. In addition, the results also indicate that if the distribution of common points degrades from 3D or 2D space to 1D space, the solvable correct transformation parameters decrease. In other words, all common points should not be located on a line. From the perspective of improving the transformation accuracy, high accurate control points (with small errors in the coordinates) should be chosen, and it is preferred to decrease the rotation angles as much as possible.

Keywords: Weighted three-dimensional coordinate transformation, Dual quaternion, Analytical dual quaternion algorithm (ADQA), Modified procrustes algorithm (MPA), Orthonormal Matrix Algorithm (OMA), Transformation accuracy

*Correspondence: zenghuaien_2003@163.com

¹ Key Laboratory of Geological Hazards on Three Gorges Reservoir Area, Ministry of Education, China Three Gorges University, Yichang 443002, China
Full list of author information is available at the end of the article

Graphical Abstract



Introduction

Three-dimensional coordinate transformation is widely used in geodesy, engineering surveying, terrestrial laser scanning, LIDAR, photogrammetry, robotics and machine vision, etc. (cf. Arun et al. 1987; Besl and McKay 1992; Chang 2016; Chen et al. 2004; Crosilla and Beinat 2002; Fan et al. 2015; Fang 2015; Felus and Burtch 2009; Horn 1987; Horn et al. 1988; Ioannidou and Pantazis 2020; Li et al. 2022; Odziemczyk 2020; Ruffhead 2021; Qin et al. 2020; Závoti and Kalmár 2016). The three-dimensional coordinate transformation usually adopts seven-parameter similarity transformation model. And the key process of three dimensional coordinate transformation is to compute the transformation parameters utilizing at least three common points with coordinates both in the original system and in the target system.

So far, a large number of algorithms of three-dimensional coordinate transformation have been presented (cf., e.g., Aydin et al. 2018; Grafarend and Awange 2003; Kanatani and Niitsuma 2012; Kurt 2018; Ligas and Prochniewicz 2019; Mahboub 2016; Marx 2017; Mercan et al. 2018; Mihajlović and Cvijetinović 2017; Păun et al. 2017; Shen et al. 2006; Uygur et al. 2020; Walker et al. 1991; Wang et al. 2014; Zeng 2015; Zeng and Yi 2010, 2011; Zeng et al. 2016, 2018, 2019). They can be classified into iterative algorithm or analytical algorithm. As far as the iterative algorithms are concerned, there are lots of transformation models. The transformation model differs as the representation of rotation matrix differs. The rotation matrix may be represented by rotation angles (Wang et al. 2018; Yang et al. 2022; Zeng and Yi 2011), direction cosine matrix (Chen et al. 2004; Wang et al. 2018), unit

quaternion (Mercan et al. 2018; Uygur et al. 2020; Zeng and Yi 2011), dual quaternion (Zeng et al. 2018, 2019), and Rodrigues matrix or Gibbs vector (Kurt, 2018; Yang et al. 2022; Závoti and Kalmár 2016; Zeng et al. 2018). Iterative algorithms usually need initial values of transformation parameters, linearization based on Taylor formula, and iterative computation (e.g., Zeng and Yi 2011; Kanatani and Niitsuma 2012). If bad initial values of parameters are provided, the algorithms probably diverge or spend lots of iterative computations (e.g., Zeng and Yi 2011). In some situations, e.g., registration of terrestrial laser scanning point clouds, it is not easy to obtain a good initial values due to the arbitrary size of rotation angles. Iterative algorithms can provide the accuracy estimation of transformation parameters and computed coordinates of non-control points, while the analytical algorithms cannot (Zeng et al. 2020, 2022).

In contrast to iterative, analytical algorithms are limited. Analytical algorithms do not need initial values of transformation parameters, linearization, or iterative computations. They use exact analytical formulae to compute transformation parameters, so are straightforward and fast. At present, analytical algorithms involve Procrustes algorithms (Arun et al. 1987; Crosilla and Beinat 2002; Grafarend and Awange 2003; Păun et al. 2017), Gibbs vector-based algorithm (Zeng and Yi 2010), unit quaternion-based algorithms (Horn 1987; Shen et al. 2006), dual quaternion-based algorithm (Walker et al. 1991; Wang et al. 2014), and orthonormal matrix-based algorithm (Horn et al. 1988; Zeng 2015). However, only the Procrustes algorithms and orthonormal matrix-based algorithm can deal with point-wise weight of observation

(Grafarend and Awange 2003; Zeng 2015). As mentioned above, analytical algorithms cannot implement accuracy estimation of transformation parameters or the computed coordinates of non-control points.

Regarding that a unit dual quaternion can represent elegantly the rigid transformation including rotation and translation, the constructed transformation model utilizing a unit dual quaternion does not need differential process to eliminate the three translation parameters, while traditional models do. Zeng et al. (2018) thought wrongly that the analytical dual quaternion solution of weighted three-dimensional coordinate transformation did not exist and then presented an iterative algorithm. This paper will prove that the analytical solution exists and present an analytical algorithm. Moreover, the presented algorithm should be suitable for point-wise weight and no matter how big rotation angles are. This paper will investigate the stability, i.e., the ability of obtaining the correct parameters of the algorithm when the distribution of common points deteriorates from 3D or 2D space to 1D space, and the reliability of the algorithm in terms of accuracy of transformed parameter under different noises and different rotation angles. This paper will propose some suggestions in improving the transformation accuracy.

The rest of the paper is organized as follows. In the next section, the point-wise weighted 3D coordinate transformation is formulated, and then the related works on the analytical algorithms of the coordinate transformation are reviewed. After introducing the basic concepts of dual quaternion, the Analytical Dual Quaternion Algorithm is derived and presented in “Analytical dual quaternion algorithm” section. Four numerical experiments, including two real world cases (geodetic datum transformation and LIDAR point cloud registration) and two simulative

the orthonormal matrix-based algorithm based on Zeng (2015), respectively.

Formulation of weighted 3D coordinate transformation

Assume there are a set of control points, i.e., known points whose coordinates in both the original system and target coordinate system are known. The essential work to fulfill 3D coordinate transformation is to compute the transformation parameters with the set of control points. The similarity 3D coordinate transformation model is as follows:

$$\mathbf{p}_i^t = \lambda \mathbf{R} \mathbf{p}_i^o + \mathbf{t}, \tag{1}$$

subject to

$$\mathbf{R}^T \mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = +1, \tag{2}$$

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 target 3D coordinate vector and original 3D coordinate vector of control point i , respectively (superscripts t and o denote target coordinate and original coordinate; subscript $i = 1, 2, \dots, n$ denotes the number of control point). \mathbf{I}_3 is a 3-by-3 identity matrix, superscript T represents matrix transpose, and \det denotes determinant computation of matrix. $\mathbf{t} = [t_x \ t_y \ t_z]^T$ is a vector of three translation parameters. λ is a scale factor from original coordinate system to target coordinate system, that is very close to 1 for most cases, like geodetic datum transformation or LiDAR point cloud registration. \mathbf{R} stands for a 3-by-3 rotation matrix, which is an orthonormal matrix with the property in Eq. (2). Conventionally \mathbf{R} is represented by three rotation angles about the three coordinate axes. Assume \mathbf{R} is produced by rotating angles θ_x, θ_y , and θ_z counterclockwise about the X, Y, and Z axes, respectively (θ_x firstly, θ_y secondly, and θ_z lastly). Thus, \mathbf{R} is expressed by rotation angles as

$$\mathbf{R} = \begin{bmatrix} \cos \theta_z \cos \theta_y & \sin \theta_z \cos \theta_x + \cos \theta_z \sin \theta_y \sin \theta_x & \sin \theta_z \sin \theta_x - \cos \theta_z \sin \theta_y \cos \theta_x \\ -\sin \theta_z \cos \theta_y & \cos \theta_z \cos \theta_x - \sin \theta_z \sin \theta_y \sin \theta_x & \cos \theta_z \sin \theta_x + \sin \theta_z \sin \theta_y \cos \theta_x \\ \sin \theta_y & -\cos \theta_y \sin \theta_x & \cos \theta_y \cos \theta_x \end{bmatrix}. \tag{3}$$

cases, are studied and discussed in “Experiments and discussions” section. Lastly conclusions are drawn in the last section, i.e., “Conclusions”.

Formulation of the question and related works

This section firstly introduces the formulation of the point-wise weighted 3D coordinate transformation and then reviews related works about the analytical algorithms of the point-wise weighted 3D coordinate transformation, namely, the Procrustes algorithm based on Grafarend and Awange (2003) and its modification, and

There are seven transformation parameters in the similarity 3D coordinate transformation model, i.e., Equation (1) including one scale factor, three translations, and three rotation angles. Thus, at least three control points ($n \geq 3$) should be given to compute the transformation parameters. In practice, different control points probably have different accuracies; therefore, it is more rational to take into account point-wise weight α_i ($i = 1, 2, \dots, n$) than to consider all points an identity weight, cf Grafarend and Awange (2003), Závoti and Kalmár (2016), Zeng (2015), Zeng et al. 2020, etc.

Taking into account the transformation error, Eq. (1) can be re-expressed as

$$\mathbf{p}_i^t = \lambda \mathbf{R} \mathbf{p}_i^o + \mathbf{t} + \mathbf{e}_i, \tag{4}$$

where \mathbf{e}_i is the transformation error vector of control point i . Then the transformation parameter solution in the least squares sense is essentially to find the argument of Lagrangian minimum problem

$$\min_{\mathbf{R}, \mathbf{t}, \lambda} \left\{ l = \sum_{i=1}^n \alpha_i \mathbf{e}_i^T \mathbf{e}_i \right\}, \tag{5}$$

subject to Eq. (2).

Procrustes algorithm and its modification

Introducing coordinate matrices $\mathbf{P}^o = [\mathbf{p}_1^o \ \mathbf{p}_2^o \ \cdots \ \mathbf{p}_n^o]^T$, $\mathbf{P}^t = [\mathbf{p}_1^t \ \mathbf{p}_2^t \ \cdots \ \mathbf{p}_n^t]^T$, and coordinate transformation error matrix $\mathbf{E} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_n]^T$, the transformation model in Eq. (4) is rewritten as

$$\mathbf{P}^t = \lambda \mathbf{P}^o \mathbf{R}^T + \mathbf{1}_n \mathbf{t}^T + \mathbf{E}, \tag{6}$$

where $\mathbf{1}_n = \underbrace{[1 \ 1 \ \cdots \ 1]}_n^T$. The Lagrangian minimum problem in Eq. (5) can be rewritten as matrix form as

$$\min_{\mathbf{R}, \mathbf{t}, \lambda} \left\{ l = \text{tr}(\mathbf{E}^T \Gamma \mathbf{E}) \right\}, \tag{7}$$

where $\Gamma = \text{Diag}(\alpha_1, \alpha_2, \dots, \alpha_n)$ is a diagonal weight matrix and tr denotes trace computation of matrix. The Lagrangian minimum exists if and only if the partial differentials with respect to λ , \mathbf{t} , \mathbf{R} as follows are equal to zero:

$$\frac{\partial l}{\partial \mathbf{t}} = 0, \tag{8}$$

$$\frac{\partial l}{\partial \lambda} = 0, \tag{9}$$

$$\frac{\partial l}{\partial \mathbf{R}} = 0. \tag{10}$$

Grafarend and Awange (2003) derived the solution of transformation step by step. Firstly by Eq. (8), the solution of \mathbf{t} is obtained as

$$\mathbf{t} = \left(\mathbf{1}_n^T \Gamma \mathbf{1}_n \right)^{-1} \left(\mathbf{P}^{tT} - \lambda \mathbf{R} \mathbf{P}^{oT} \right) \Gamma \mathbf{1}_n. \tag{11}$$

Secondly substituting the solution of \mathbf{t} to Eq. (9), λ is obtained as

$$\lambda = \frac{\text{tr}(\mathbf{P}^{tT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o \mathbf{R}^T)}{\text{tr}(\mathbf{P}^{oT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o)}, \tag{12}$$

where $\tilde{\mathbf{C}} = \mathbf{I}_n - (\mathbf{1}_n^T \Gamma \mathbf{1}_n)^{-1} \mathbf{1}_n \mathbf{1}_n^T \Gamma$, \mathbf{I}_n is a n -by- n identity matrix. Finally substituting the solutions of \mathbf{t} and λ to Eq. (10), \mathbf{R} is obtained using the singular value decomposition as

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

where $\mathbf{P}^{tT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ is a singular value decomposition of $\mathbf{P}^{tT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o$. \mathbf{U} , \mathbf{V} , and $\mathbf{\Sigma}$ are left orthonormal matrix, right orthonormal matrix, and diagonal matrix of singular values, respectively. Eq. (13) ensures that $\mathbf{R}^T \mathbf{R} = \mathbf{I}_3$ is satisfied. However, it is worthy of notice that $\det(\mathbf{R}) = -1$ rather than $\det(\mathbf{R}) = +1$ usually occurs when the common points are located in a rigid or approximate plane. In other words, for the case that $\det(\mathbf{R}) = -1$ the computed \mathbf{R} is a reflection instead of rotation. In order to obtain the correct rotation matrix, Eq. (13), i.e., Equation (22) in Grafarend and Awange (2003), should be modified as

$$\mathbf{R} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U} \mathbf{V}^T) \end{bmatrix} \mathbf{V}^T. \tag{14}$$

After \mathbf{R} , λ , and \mathbf{t} are computed, \mathbf{E} is compute by Eq. (6), and the variance factor of unit weight σ is computed as follows:

$$\sigma = \pm \sqrt{\frac{\text{tr}(\mathbf{E}^T \Gamma \mathbf{E})}{3n - 7}}. \tag{15}$$

If the rotation angles are needed, they are computed by the formulae as follows:

$$\theta_x = -\tan^{-1} \frac{\mathbf{R}_{32}}{\mathbf{R}_{33}}, \theta_y = \sin^{-1}(\mathbf{R}_{31}), \theta_z = -\tan^{-1} \frac{\mathbf{R}_{21}}{\mathbf{R}_{11}}, \tag{16}$$

where \mathbf{R}_{ij} is the element of \mathbf{R} in the i th row and j th column.

Orthonormal matrix algorithm

The orthonormal matrix algorithm constructs the Lagrangian minimum problem with the constraint $\mathbf{R}^T \mathbf{R} = \mathbf{I}_3$ as follows (Zeng 2015):

$$\min_{\mathbf{R}, \mathbf{t}, \lambda, \Lambda} \left\{ l = \text{tr}(\mathbf{E}^T \Gamma \mathbf{E}) + \text{tr}(\Lambda (\mathbf{R}^T \mathbf{R} - \mathbf{I}_3)) \right\}, \tag{17}$$

where Λ is a symmetric Lagrangian multiplier matrix. The Lagrangian minimum exists if and only if the partial differentials with respect to λ , \mathbf{t} , \mathbf{R} , and Λ as follows are equal to zero.

$$\frac{\partial l}{\partial \mathbf{t}} = 0, \tag{18}$$

$$\frac{\partial l}{\partial \lambda} = 0, \tag{19}$$

$$\frac{\partial l}{\partial \mathbf{R}} = 0, \tag{20}$$

$$\frac{\partial l}{\partial \Lambda} = 0. \tag{21}$$

Zeng (2015) derived the solution of transformation step by step. By Eq. (18), the solution of \mathbf{t} is obtained as

$$\mathbf{t} = \left(\mathbf{1}_n^T \Gamma \mathbf{1}_n \right)^{-1} \left(\mathbf{P}^{tT} - \lambda \mathbf{R} \mathbf{P}^{oT} \right) \Gamma \mathbf{1}_n, \tag{22}$$

which is identical to Eq. (11). Substituting the solution of \mathbf{t} to Eq. (19), λ is obtained as

$$\lambda = \frac{\text{tr}(\mathbf{P}^{tT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o \mathbf{R}^T)}{\text{tr}(\mathbf{P}^{oT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o)}, \tag{23}$$

which is identical to Eq. (12). By Eqs. (20) and (21), the solution of \mathbf{R} is obtained as

$$\mathbf{R} = \mathbf{D}(\mathbf{D}^T \mathbf{D})^{-\frac{1}{2}}, \tag{24}$$

where $\mathbf{D} = \mathbf{P}^{tT} \tilde{\mathbf{C}}^T \Gamma \tilde{\mathbf{C}} \mathbf{P}^o$. Since $\mathbf{D}^T \mathbf{D}$ is symmetric, non-negative definite, it has non-negative real eigenvalues. Therefore, the inverse of the square root of $\mathbf{D}^T \mathbf{D}$ can be calculated by eigenvalue–eigenvector decomposition.

$$(\mathbf{D}^T \mathbf{D})^{-\frac{1}{2}} = \frac{v_1 v_1^T}{\sqrt{d_1}} + \frac{v_2 v_2^T}{\sqrt{d_2}} + \frac{v_3 v_3^T}{\sqrt{d_3}}, \tag{25}$$

where v_i and d_i , $i = 1, 2, 3$, are the eigenvectors and corresponding eigenvalues of the matrix $\mathbf{D}^T \mathbf{D}$. Thus, Eq. (24) can be re-expressed as

$$\mathbf{R} = \mathbf{D} \left(\frac{v_1 v_1^T}{\sqrt{d_1}} + \frac{v_2 v_2^T}{\sqrt{d_2}} + \frac{v_3 v_3^T}{\sqrt{d_3}} \right). \tag{26}$$

One should notice that if one or two of d_i for $i = 1, 2, 3$ equals to 0, the construction of Eq. (25) makes no sense. Suppose that the eigenvalues of the matrix $\mathbf{D}^T \mathbf{D}$ meet the condition as follows:

$$d_1 \leq d_2 \leq d_3, \tag{27}$$

Zeng (2015) presented the formulae to compute rotation matrix as follows for the cases that the common points are distributed in a rigid or approximate plane, i.e., d_1 equals 0 or approaches very close to 0.

$$\mathbf{D}_1 = \left(\frac{v_2 v_2^T}{\sqrt{d_2}} + \frac{v_3 v_3^T}{\sqrt{d_3}} \right), \tag{28}$$

$$\mathbf{D}_2 = \left(\mathbf{D} \mathbf{D}_1 (\mathbf{D} \mathbf{D}_1)^T - \mathbf{I}_3 \right) v_1 v_1^T, \tag{29}$$

$$\mathbf{R} = \mathbf{D} \mathbf{D}_1 \pm \frac{\mathbf{D}_2}{\sqrt{|\text{tr}(\mathbf{D}_2)|}}, \tag{30}$$

where the sign \pm is chosen so that $\det(\mathbf{R}) = +1$ is satisfied.

Zeng (2015) presented the formulae to compute rotation matrix as follows for the cases that the common points are distributed in a rigid or approximate line, i.e., $d_2 = d_1 = 0$, $d_3 > 0$ or $d_3 > d_2 > d_1 = 0$.

$$\mathbf{R} = \frac{v_3 v_3^T}{\sqrt{d_3}}. \tag{31}$$

After \mathbf{R} , λ , and \mathbf{t} are computed, the variance factor of unit weight σ is computed as same as in Eq. (15). The rotation angles can be computed from \mathbf{R} by Eq. (16). However, for this cases that the common points are distributed in a rigid or approximate line, the solvable rotation angles are 0–2 (Zeng 2015).

Analytical dual quaternion algorithm

Basic concept of dual quaternion

Quaternions r and s are written as follows (Walker et al. 1991):

$$r = r_1 i + r_2 j + r_3 k + r_4, \tag{32}$$

$$s = s_1 i + s_2 j + s_3 k + s_4, \tag{33}$$

where $r_1, r_2, r_3, r_4, s_1, s_2, s_3$, and s_4 are real numbers; i, j , and k are basic quaternion units. And then a dual quaternion is expressed as follows:

$$\begin{aligned} \bar{q} &= r + \epsilon s \\ &= (r_1 + \epsilon s_1) i + (r_2 + \epsilon s_2) j + (r_3 + \epsilon s_3) k + (r_4 + \epsilon s_4), \\ &= q_{d1} i + q_{d2} j + q_{d3} k + q_{d4} \end{aligned} \tag{34}$$

where ε is a dual unit with the property $\varepsilon^2 = 0$ and ε commutes with quaternion units. $(q_{d1} \ q_{d2} \ q_{d3})^T$ is the vector part; q_{d4} is the scalar part.

The conjugate of \bar{q} is defined as

$$\bar{q}^* = r^* + \varepsilon s^*, \tag{35}$$

where r^* and s^* are conjugates of r and s as follows:

$$r^* = -ir_1 - jr_2 - kr_3 + r_4, \tag{36}$$

$$s^* = -s_1i - s_2j - s_3k + s_4. \tag{37}$$

$\bar{p} = u + \varepsilon v$ denotes a dual quaternion, and then the product of \bar{q} and \bar{p} is defined as

$$\bar{p}\bar{q} = ur + \varepsilon(us + vr). \tag{38}$$

The norm of a dual quaternion is defined as

$$\|\bar{q}\| = \sqrt{\bar{q}^* \bar{q}} = \sqrt{q_{d1}^2 + q_{d2}^2 + q_{d3}^2 + q_{d4}^2}. \tag{39}$$

If $\|\bar{q}\| = 1$, \bar{q} is named a unit dual quaternion. At this time, these conditions as follows exist:

$$r^T r = 1, \tag{40}$$

$$r^T s = 0. \tag{41}$$

The rigid transformation, including rotation and translation, can be represented by unit dual quaternion (Walker et al. 1991). The rotation matrix is represented by the unit quaternion r (e.g., Walker et al. 1991; Zeng and Yi 2011).

$$\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{W}(r)^T Q(r), \tag{42}$$

where

$$Q(r) = \begin{bmatrix} r_4 \mathbf{I} + \mathbf{C}(r) & r \\ -r^T & r_4 \end{bmatrix}, \tag{43}$$

$$\mathbf{W}(r) = \begin{bmatrix} r_4 \mathbf{I} - \mathbf{C}(r) & r \\ -r^T & r_4 \end{bmatrix}, \tag{44}$$

$$r = [r_1 \ r_2 \ r_3]^T, \tag{45}$$

$$\mathbf{C}(r) = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}. \tag{46}$$

Expanding Eq. (42), one obtains

$$\mathbf{R} = (r_4^2 - r^T r) \mathbf{I} + 2(r r^T + r_4 \mathbf{C}(r)). \tag{47}$$

Assume t is the pure imaginary quaternion composed of translation \mathbf{t} , i.e.,

$$t = \frac{1}{2} \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix}, \tag{48}$$

according to Walker et al. (1991),

$$t = W(r)^T s. \tag{49}$$

Derivation and implementation of analytical dual quaternion algorithm

Introducing coordinate vector quaternions

$$p_i^o = \begin{bmatrix} p_i^o \\ 0 \end{bmatrix}, \quad p_i^t = \begin{bmatrix} p_i^t \\ 0 \end{bmatrix}, \tag{50}$$

the similarity 3D coordinate transformation model considering the error, i.e., Equation (4), is rewritten by dual quaternion as

$$p_i^t = \lambda \mathbf{W}(r)^T Q(r) p_i^o + 2W(r)^T s + e_i, \tag{51}$$

subject to Eqs. (40) and (41). In Eq. (51), $e_i = [e_i^T 0]^T$ is the transformation error quaternion. A Lagrangian extremum problem without constrains is constructed as

$$\min_{r,s,\lambda,\beta_1,\beta_2} \left\{ \tilde{e} = \sum_{i=1}^n \alpha_i e_i^T e_i + \beta_1 (r^T r - 1) + \beta_2 s^T r \right\}, \tag{52}$$

where β_1 and β_2 are two Lagrangian multipliers. Replacing the expression of e_i from Eq. (51) into error function \tilde{e} and expanding the error function \tilde{e} , according to Zeng et al. (2018), one gets

$$\begin{aligned} \tilde{e} = & a + b\lambda^2 + 4cs^T s - 2\lambda r^T \mathbf{A}r \\ & - 4s^T \mathbf{B}r + 4\lambda s^T \mathbf{C}r + \beta_1 (r^T r - 1) \\ & + \beta_2 s^T r, \end{aligned} \tag{53}$$

where

$$a = \sum_{i=1}^n \alpha_i p_i^{tT} p_i^t, \tag{54}$$

$$b = \sum_{i=1}^n \alpha_i \mathbf{p}_i^{oT} \mathbf{p}_i^o, \tag{55}$$

$$c = \sum_{i=1}^n \alpha_i, \tag{56}$$

$$\mathbf{A} = \sum_{i=1}^n \alpha_i W(\mathbf{p}_i^o)^T Q(\mathbf{p}_i^t), \tag{57}$$

$$\mathbf{B} = \sum_{i=1}^n \alpha_i Q(\mathbf{p}_i^t), \tag{58}$$

$$\mathbf{C} = \sum_{i=1}^n \alpha_i W(\mathbf{p}_i^o), \tag{59}$$

The Lagrangian minimum in Eq. (52) exists if and only if the following conditions are satisfied:

$$\frac{\delta \tilde{e}}{\delta \mathbf{r}} = -4\lambda \mathbf{r}^T \mathbf{A} - 4\mathbf{s}^T \mathbf{B} + 4\lambda \mathbf{s}^T \mathbf{C} + 2\beta_1 \mathbf{r}^T + \beta_2 \mathbf{s}^T = 0, \tag{60}$$

$$\frac{\delta \tilde{e}}{\delta \mathbf{s}} = 8c\mathbf{s}^T - 4\mathbf{r}^T \mathbf{B}^T + 4\lambda \mathbf{r}^T \mathbf{C}^T + \beta_2 \mathbf{r}^T = 0, \tag{61}$$

$$\frac{\delta \tilde{e}}{\delta \lambda} = 2b\lambda - 2\mathbf{r}^T \mathbf{A} \mathbf{r} + 4\mathbf{s}^T \mathbf{C} \mathbf{r} = 0, \tag{62}$$

$$\frac{\delta \tilde{e}}{\delta \beta_1} = \mathbf{r}^T \mathbf{r} - 1 = 0, \tag{63}$$

$$\frac{\delta \tilde{e}}{\delta \beta_2} = \mathbf{s}^T \mathbf{r} = 0. \tag{64}$$

The derivation of Eq. (60) makes use of the property that \mathbf{A} is a symmetric matrix. The property is proved in Zeng et al. (2018).

For skew symmetric matrixes \mathbf{B} and \mathbf{C} , the following property is proved in Zeng et al. (2018).

$$\mathbf{r}^T \mathbf{B} \mathbf{r} = 0, \quad \mathbf{r}^T \mathbf{C} \mathbf{r} = 0. \tag{65}$$

Right multiplying Eq. (61) by \mathbf{r} , and considering Eq. (63), Eq. (64) and Eq. (65), one obtains

$$\beta_2 = 0. \tag{66}$$

By Eq. (61), one obtains

$$\mathbf{s} = \frac{1}{2c}(\mathbf{B} - \lambda \mathbf{C})\mathbf{r}. \tag{67}$$

Substituting Eq. (67) into Eq. (62) and making arrangements, one gets

$$\lambda = \frac{\mathbf{r}^T \mathbf{A} \mathbf{r} - c^{-1} \mathbf{r}^T \mathbf{B}^T \mathbf{C} \mathbf{r}}{b - c^{-1} \mathbf{r}^T \mathbf{C}^T \mathbf{C} \mathbf{r}}. \tag{68}$$

Substituting Eq. (67) into Eq. (60) and considering Eq. (66), and making arrangements one gets

$$\left[2\lambda \mathbf{A} + \frac{1}{c}(\mathbf{B} - \lambda \mathbf{C})^T (\mathbf{B} - \lambda \mathbf{C}) \right] \mathbf{r} = \beta_1 \mathbf{r}. \tag{69}$$

Introducing

$$\mathbf{F} = 2\lambda \mathbf{A} + \frac{1}{c}(\mathbf{B} - \lambda \mathbf{C})^T (\mathbf{B} - \lambda \mathbf{C}), \tag{70}$$

Equation (69) is re-expressed as

$$\mathbf{F} \mathbf{r} = \beta_1 \mathbf{r}. \tag{71}$$

It is obviously seen that β_1 and \mathbf{r} are the eigenvalue and eigenvector of \mathbf{F} . Because \mathbf{A} is symmetric and real, \mathbf{F} is symmetric and real. Thus, \mathbf{F} has four real eigenvalues and four orthogonal real eigenvectors. The next work is to find the solution of \mathbf{r} and β_1 from the eigenvectors and eigenvalues. The expression of error function e was derived in Zeng et al. (2018) as

$$e = a + b\lambda^2 - \beta_1, \tag{72}$$

It is shown that e gets its minimum if β_1 is the largest eigenvalue of \mathbf{F} . So solution of \mathbf{r} is the eigenvector of \mathbf{F} corresponding to its largest eigenvalue.

This paper will prove that the analytical solution exists and present an analytical algorithm as follows.

Expanding the formula of \mathbf{F} in Eq. (70), one gets

$$\mathbf{F} = 2\lambda \mathbf{A} + \frac{1}{c} \left(\mathbf{B}^T \mathbf{B} - \lambda \mathbf{B}^T \mathbf{C} - \lambda \mathbf{C}^T \mathbf{B} + \lambda^2 \mathbf{C}^T \mathbf{C} \right). \tag{73}$$

$\mathbf{B}^T \mathbf{C}$ is a symmetric matrix, referring to the Appendix for the proof. $\mathbf{B}^T \mathbf{B}$ and $\mathbf{C}^T \mathbf{C}$ are constant matrixes as follows:

$$\mathbf{B}^T \mathbf{B} = \left[\left(\sum_{i=1}^n \alpha_i x_i^t \right)^2 + \left(\sum_{i=1}^n \alpha_i y_i^t \right)^2 + \left(\sum_{i=1}^n \alpha_i z_i^t \right)^2 \right] \mathbf{I}_4, \tag{74}$$

$$\mathbf{C}^T \mathbf{C} = \left[\left(\sum_{i=1}^n \alpha_i x_i^o \right)^2 + \left(\sum_{i=1}^n \alpha_i y_i^o \right)^2 + \left(\sum_{i=1}^n \alpha_i z_i^o \right)^2 \right] \mathbf{I}_4. \tag{75}$$

Further, Eq. (70) is rewritten as

$$\mathbf{F} = 2\lambda \left(\mathbf{A} - \frac{1}{c} \mathbf{B}^T \mathbf{C} \right) + \frac{1}{c} \left(\mathbf{B}^T \mathbf{B} + \lambda^2 \mathbf{C}^T \mathbf{C} \right). \quad (76)$$

Suppose that the eigenvalue and eigenvector decomposition of \mathbf{F} is written as

$$\mathbf{F}\mu = \kappa\mu, \quad (77)$$

where κ, μ are the eigenvalue and eigenvector decomposition of \mathbf{F} . Substituting Eq. (76) into Eq. (77), one gets

$$2\lambda \left(\mathbf{A} - \frac{1}{c} \mathbf{B}^T \mathbf{C} \right) \mu + \frac{1}{c} \left(\mathbf{B}^T \mathbf{B} + \lambda^2 \mathbf{C}^T \mathbf{C} \right) \mu = \kappa\mu. \quad (78)$$

Substituting Eqs. (74) and (75) into Eq. (78) and making arrangements, one obtains

$$\left(\mathbf{A} - \frac{1}{c} \mathbf{B}^T \mathbf{C} \right) \mu = \frac{1}{2\lambda} \left[\kappa - \frac{1}{c} \left(\tau^t + \lambda^2 \tau^o \right) \right] \mu, \quad (79)$$

where

$$\tau^t = \left[\left(\sum_{i=1}^n \alpha_i x_i^t \right)^2 + \left(\sum_{i=1}^n \alpha_i y_i^t \right)^2 + \left(\sum_{i=1}^n \alpha_i z_i^t \right)^2 \right], \quad (80)$$

$$\tau^o = \left[\left(\sum_{i=1}^n \alpha_i x_i^o \right)^2 + \left(\sum_{i=1}^n \alpha_i y_i^o \right)^2 + \left(\sum_{i=1}^n \alpha_i z_i^o \right)^2 \right]. \quad (81)$$

Introducing

$$\mathbf{G} = \left(\mathbf{A} - \frac{1}{c} \mathbf{B}^T \mathbf{C} \right), \quad (82)$$

$$\gamma = \frac{1}{2\lambda} \left[\kappa - \frac{1}{c} \left(\tau^t + \lambda^2 \tau^o \right) \right], \quad (83)$$

Equation (76) is rewritten as

$$\mathbf{G}\mu = \gamma\mu. \quad (84)$$

Therefore, it is obviously seen that γ and μ are the eigenvalue and eigenvector of \mathbf{G} . \mathbf{G} is symmetric and real as \mathbf{F} ; thus, \mathbf{G} also has four couples of real eigenvalues and orthogonal real eigenvectors as \mathbf{F} . Further, μ is the eigenvector of both \mathbf{G} and \mathbf{F} , and γ and κ are related linearly. When κ increases, γ increases. Although it is not possible to obtain the largest eigenvalue based on eigenvalue and eigenvector decomposition of \mathbf{F} directly because of the ambiguity of λ in \mathbf{F} , it is feasible to compute the largest eigenvalue and corresponding eigenvector based on eigenvalue and eigenvector decomposition of \mathbf{G} . Then the largest eigenvalue of \mathbf{F} is computed from Eq. (83) as

$$\beta_1 = 2\lambda\gamma_{\max} + \frac{1}{c} \left(\tau^t + \lambda^2 \tau^o \right), \quad (85)$$

where γ_{\max} is the largest eigenvalue of \mathbf{G} . And corresponding eigenvector of \mathbf{F} , i.e., \mathbf{r} , is the eigenvector corresponding to largest eigenvalue of \mathbf{G} .

At last, after $\mathbf{r}, \mathbf{s}, \lambda,$ and \mathbf{t} are computed, the standard deviation of unit weight is estimated by

$$\sigma = \pm \sqrt{\frac{\sum_{i=1}^n \alpha_i \mathbf{e}_i^T \mathbf{e}_i}{3n - 7}}, \quad (86)$$

or

$$\sigma = \pm \sqrt{\frac{a + b\lambda^2 - \beta_1}{3n - 7}}. \quad (87)$$

Table 1 The analytical dual quaternion algorithm of the weighted 3D coordinate transformation

Input: input 3D coordinates of control points and construct 3D coordinate vector quaternion by Eq. (50), and input the point-wise weight α_i

Analytical computation:

Step 1. Compute $a, b, c, \mathbf{A}, \mathbf{B}, \mathbf{C}$ by Eq. (54) to Eq. (59)

Step 2. Compute \mathbf{G} by Eq. (82), and perform the eigenvalue–eigenvector decomposition of \mathbf{G} . γ_{\max} is set to the largest eigenvalue, and \mathbf{r} is set to the eigenvector corresponding to the largest eigenvalue

Step 3. Compute λ by Eq. (68)

Step 4. Compute \mathbf{s} by Eq. (67)

Step 5. Compute \mathbf{R} by Eq. (47); furthermore, if rotation angles $\theta_x, \theta_y, \theta_z$ are needed, compute them by Eq. (16)

Step 6. Compute \mathbf{t} by Eq. (49) and then translation \mathbf{t} by Eq. (48)

Step 7. Compute \mathbf{e}_i by Eq. (51) and then compute σ by Eq. (86). Or compute β_1 by Eq. (85) and then compute σ by Eq. (87)

Output: output $\mathbf{r}, \mathbf{s}, \lambda, \mathbf{t}, \mathbf{R}, (\theta_x, \theta_y, \theta_z$ if needed), σ

Table 2 3D coordinates of control points

| Station Name | Original system (local system) (m) | | | Target system (WGS-84 system) (m) | | |
|----------------|------------------------------------|-------------|---------------|-----------------------------------|-------------|---------------|
| | x^o | y^o | z^o | x^t | y^t | z^t |
| Solitude | 4,157,222.543 | 664,789.307 | 4,774,952.099 | 4,157,870.237 | 664,818.678 | 4,775,416.524 |
| Buoch Zeil | 4,149,043.336 | 688,836.443 | 4,778,632.188 | 4,149,691.049 | 688,865.785 | 4,779,096.588 |
| Hohenneuffen | 4,172,803.511 | 690,340.078 | 4,758,129.701 | 4,173,451.354 | 690,369.375 | 4,758,594.075 |
| Kuehlenberg | 4,177,148.376 | 642,997.635 | 4,760,764.800 | 4,177,796.064 | 643,026.700 | 4,761,228.899 |
| Ex Mergelaec | 4,137,012.190 | 671,808.029 | 4,791,128.215 | 4,137,659.549 | 671,837.337 | 4,791,592.531 |
| Ex Hof Asperg | 4,146,292.729 | 666,952.887 | 4,783,859.856 | 4,146,940.228 | 666,982.151 | 4,784,324.099 |
| Ex Kaisersbach | 4,138,759.902 | 702,670.738 | 4,785,552.196 | 4,139,407.506 | 702,700.227 | 4,786,016.645 |

Table 3 Point-wise weights

| Station Name | Solitude | Buoch Zeil | Hohenneuffen | Kuehlenberg | Ex Mergelaec | Ex Hof Asperg | Ex Kaisersbach |
|--------------|----------|------------|--------------|-------------|--------------|---------------|----------------|
| Weight | 2.170137 | 2.097755 | 2.208968 | 2.201671 | 2.182928 | 2.268808 | 2.643404 |

Table 4 Computed dual quaternion by ADQA

| Quaternion | r | s |
|------------|------------------|------------------|
| Value | 0.000002418528 | 320.920158312595 |
| | − 0.000002172181 | 34.237708673610 |
| | − 0.000002389849 | 208.107012357002 |
| | 0.999999999992 | − 0.000204439773 |

The analytical dual quaternion algorithm is summarized in Table 1.

Experiments and discussion

Four numerical cases, including geodetic datum transformation, the registration of LIDAR point clouds, and two simulated cases, are studied. The former two cases are actual and chosen to investigate whether the presented algorithm is valid to transformation with point-wise weight and with small or big rotation angles. The latter two cases, i.e., simulated cases, are designed to test the performance of the presented algorithm on the aspects of stability with different distributions of control points (case I) and reliability with different noises and with different rotation angles (case II).

Geodetic datum transformation case

Usually the rotation angles in geodetic datum transformation are small (less than 1°). The data in this case are from Grafarend and Awange (2003), and the rotation angles are all less than 1". The 3D coordinates of control points in both original system and target system

Table 5 Calculated transformation parameters by ADQA, MPA, and OMA

| Rotation matrix | | | |
|---------------------------------------|----------------|--------------|----------------|
| R | 1.0000000000 | 0.0000047797 | − 0.0000043444 |
| | − 0.0000047797 | 1.0000000000 | − 0.0000048370 |
| | 0.0000043443 | 0.0000048371 | 1.0000000000 |
| Rotation angles (") | | | |
| θ_x | | − 0.997716 | |
| θ_y | | 0.896085 | |
| θ_z | | 0.985885 | |
| Translation (m) | | | |
| t_x | | 641.8395 | |
| t_y | | 68.4729 | |
| t_z | | 416.2156 | |
| Scale | | | |
| λ | | 1.000005611 | |
| Standard deviation of unit weight (m) | | | |
| σ | | 0.1140 | |

are listed in Table 2. The point-wise weights of control points are listed in Table 3. The Analytical Dual Quaternion Algorithm (ADQA), Modified Procrustes Algorithm (MPA), and Orthonormal Matrix Algorithm (OMA) are employed to compute the transformation parameters. The computed unit dual quaternion by ADQA is listed in Table 4, and the computed transformation parameters and standard deviation of unit weight by the three algorithms are identical (the rounding errors are small enough to ignore, for instance, less than 0.000001" for the

Table 6 Common points extracted from two neighboring LIDAR point clouds

| Point no. | Unregistered station in the original system (m) | | | Reference station in the target system (m) | | |
|-----------|---|---------|--------|--|---------|--------|
| | x^o | y^o | z^o | x^t | y^t | z^t |
| 1 | -49.007 | 54.453 | 0.978 | -91.406 | 53.344 | 8.320 |
| 2 | -47.365 | 54.435 | -6.242 | -91.297 | 53.222 | 0.916 |
| 3 | -36.514 | 13.733 | 3.642 | -60.158 | 24.280 | 8.948 |
| 4 | -34.881 | 13.859 | -3.608 | -60.135 | 24.278 | 1.521 |
| 5 | -53.378 | -25.872 | -4.187 | -56.298 | -19.186 | 5.700 |
| 6 | -7.324 | -32.695 | -1.389 | -13.269 | -2.677 | -1.444 |
| 7 | 9.587 | -19.650 | 2.449 | -4.666 | 17.245 | -1.605 |
| 8 | -36.532 | -0.319 | 21.980 | -49.939 | 14.297 | 27.119 |
| 9 | -39.932 | -1.307 | 19.965 | -52.769 | 11.523 | 25.906 |
| 10 | -67.051 | -8.834 | 15.017 | -72.929 | -8.630 | 27.146 |
| 11 | -54.124 | -40.688 | 13.216 | -46.500 | -30.291 | 23.078 |
| 12 | -51.943 | -30.962 | -3.965 | -52.581 | -22.934 | 5.676 |
| 13 | -57.712 | -23.376 | 8.397 | -58.972 | -17.511 | 18.862 |
| 14 | -59.650 | -32.625 | 12.037 | -55.429 | -26.155 | 23.077 |
| 15 | -59.512 | -32.705 | 12.071 | -55.313 | -26.131 | 23.039 |
| 16 | -41.466 | 18.246 | 21.085 | -63.467 | 27.962 | 26.981 |
| 17 | -39.133 | 10.234 | 20.247 | -57.673 | 22.069 | 25.782 |
| 18 | -29.781 | -0.026 | -8.062 | -49.687 | 14.083 | -3.666 |

Table 7 Computed dual quaternion by ADQA

| Quaternion | r | s |
|------------|-----------------|-----------------|
| Value | -0.036681390787 | -7.197133335638 |
| | 0.103091603067 | 17.077717584215 |
| | 0.253305902396 | -1.733260783702 |
| | 0.961177775835 | -1.649564727641 |

rotation angle) and listed in Table 5. Thus, the presented algorithm ADQA is valid for the case with small rotation angles and point-wise weights.

LiDAR point cloud registration case

Usually the rotation angles in LiDAR Point Cloud Registration are big. The data in this case are chosen from Wang et al. (2014). 18 common points are extracted from two neighboring LIDAR point clouds and listed in Table 6. The point-wise weights of all points are set to 1 in this case. In other words, the weights of all points are not taken into account. The computed unit dual quaternion by ADQA is given in Table 7, and the computed transformation parameters and standard deviation of unit weight by ADQA, MPA, and OMA are identical (the rounding errors are small enough to ignore) and listed in Table 8. Therefore, the presented algorithm ADQA is valid for the case with big rotation angles.

Table 8 Calculated transformation parameters by ADQA, MPA, and OMA

| Rotation matrix | | | |
|---------------------------------------|----------------|---------------|--------------|
| R | 0.8504164824 | -0.4945070945 | 0.1795954899 |
| | 0.4793809210 | 0.8689811908 | 0.1227420983 |
| | -0.2167619411 | -0.0182872521 | 0.9760531939 |
| Rotation angles (°) | | | |
| θ_x | 1.0733634149 | | |
| θ_y | -12.5189170709 | | |
| θ_z | -29.4100148194 | | |
| Translation (m) | | | |
| t_x | -22.9656 | | |
| t_y | 29.3962 | | |
| t_z | -2.2652 | | |
| Scale | | | |
| λ | 1.000385442 | | |
| Standard deviation of unit weight (m) | | | |
| σ | 0.0301 | | |

Simulated case I

This case is designed to test the stability of the presented algorithm, ADQA, and compare with MPA and OMA. The data are from Zeng (2015). The data are simulated as follows. Six sets of common points in the original system

Table 9 Simulative true coordinates of control points in the original system

| Point number | Set 1 (m): 3D space | | | Set 2 (m): 2D space | | | Set 3 (m): 2D space | | |
|--------------|---------------------|--------|--------|---------------------|--------|--------|---------------------|--------|--------|
| | x^o | y^o | z^o | x^o | y^o | z^o | x^o | y^o | z^o |
| 1 | 10.000 | 30.000 | 5.000 | 10.000 | 30.000 | 5.000 | 10.000 | 30.000 | 57.000 |
| 2 | 20.000 | 30.000 | 12.500 | 20.000 | 30.000 | 12.500 | 20.000 | 30.000 | 67.000 |
| 3 | 30.000 | 30.000 | 15.000 | 30.000 | 30.000 | 15.000 | 30.000 | 30.000 | 77.000 |
| 4 | 10.000 | 20.000 | 9.500 | | | | 10.000 | 20.000 | 42.000 |
| 5 | 20.000 | 20.000 | 11.000 | | | | 20.000 | 20.000 | 52.000 |
| 6 | 30.000 | 20.000 | 10.000 | | | | 30.000 | 20.000 | 62.000 |
| 7 | 10.000 | 10.000 | 14.500 | | | | 10.000 | 10.000 | 27.000 |
| 8 | 20.000 | 10.000 | 4.500 | | | | 20.000 | 10.000 | 37.000 |
| 9 | 30.000 | 10.000 | 4.000 | | | | 30.000 | 10.000 | 47.000 |

| Point number | Set 4(m): 2D space | | | Set5 (m): 1D space | | | Set 6 (m): 1D space | | |
|--------------|--------------------|--------|--------|--------------------|--------|--------|---------------------|-------|-------|
| | x^o | y^o | z^o | x^o | y^o | z^o | x^o | y^o | z^o |
| 1 | 10.000 | 30.000 | 15.000 | 10.000 | 10.000 | 10.000 | 10.000 | 0.000 | 0.000 |
| 2 | 20.000 | 30.000 | 15.000 | 20.000 | 20.000 | 20.000 | 20.000 | 0.000 | 0.000 |
| 3 | 30.000 | 30.000 | 15.000 | 30.000 | 30.000 | 30.000 | 30.000 | 0.000 | 0.000 |
| 4 | 10.000 | 20.000 | 15.000 | 40.000 | 40.000 | 40.000 | | | |
| 5 | 20.000 | 20.000 | 15.000 | 50.000 | 50.000 | 50.000 | | | |
| 6 | 30.000 | 20.000 | 15.000 | 60.000 | 60.000 | 60.000 | | | |
| 7 | 10.000 | 10.000 | 15.000 | 70.000 | 70.000 | 70.000 | | | |
| 8 | 20.000 | 10.000 | 15.000 | 80.000 | 80.000 | 80.000 | | | |
| 9 | 30.000 | 10.000 | 15.000 | 90.000 | 90.000 | 90.000 | | | |

Table 10 Simulative true values of transformation parameters

| t_x (m) | t_y (m) | t_z (m) | θ_x (°) | θ_y (°) | θ_z (°) | λ |
|-----------|-----------|-----------|----------------|----------------|----------------|-----------|
| 30 | 30 | 10 | 71 | 78 | 73 | 1.000016 |

are given in Table 9 firstly. For Set 1, common points are distributed in 3D space. For Sets 2 to 4, common points are distributed in 2D space. For Sets 5 and 6, common points are distributed in 1D space. Then true transformation parameters are designed to suit big rotation angle and listed in Table 10. At last, the simulative true coordinates of control points in the target system are computed by Eq. (1) and listed in Table 11. ADQA, MPA, and OMA are employed to compute the transformation parameters. The results of Set 1 to Set 4 by ADQA, MPA, and OMA are identical and listed in Table 12. For Set 5 and Set 6, the results by ADQA, MPA, and OMA are different and listed in Table 13. It is obviously seen that for Set 1 to Set 4, i.e., 3D or 2D space, the three algorithms are valid to compute the correct transformation parameters. And for Set 5 and Set 6, i.e., 1D space, the three algorithms are valid to compute all correct translation and scale parameters, however, only 0–2 correct rotation angles. Thus, ADQA is stable when all common points (at least three) are located in 2D or 3D space, however, become unstable

when all common points are located in 1D space (i.e., on a line). In order to compute 7 parameters correctly all common points (at least three) should be distributed in 2D or 3D space. In other words, all common points should not be distributed on a line; otherwise only 4–6 of the 7 parameters can be computed correctly.

Simulated case II

In this case two schemes are designed to test the reliability of ADQA with different noises (Scheme 1) and with different rotation angles (Scheme 2). The settings of two schemes are listed in Table 14.

For Scheme 1, the coordinates of control points in the target system are computed by the given coordinates of control points in the original system and fixed transformation parameters. In order to simulate the errors of control points in the original system and in the target system, normal distributed noises $N(0, \sigma_0^2)$ are produced and added into the coordinates. The standard deviation σ_0 is varied in the zone [0.0001 m, 0.01 m] with a

Table 11 Simulative true coordinates of control points in the target system

| Point number | Set 1 (m): 3D space | | | Set 2 (m): 2D space | | | Set 3 (m): 2D space | | |
|--------------|---------------------|--------|--------|---------------------|-------|--------|---------------------|--------|--------|
| | x^t | y^t | z^t | x^t | y^t | z^t | x^t | y^t | z^t |
| 1 | 52.116 | 7.239 | 14.222 | 52.116 | 7.239 | 14.222 | 94.294 | 37.450 | 17.742 |
| 2 | 58.807 | 9.608 | 24.512 | 58.807 | 9.608 | 24.512 | 103.013 | 41.272 | 28.201 |
| 3 | 61.443 | 9.072 | 34.463 | 61.443 | 9.072 | 34.463 | 111.732 | 45.093 | 38.659 |
| 4 | 49.949 | 17.746 | 16.493 | | | | 76.310 | 36.628 | 18.693 |
| 5 | 51.773 | 16.629 | 26.376 | | | | 85.029 | 40.450 | 29.151 |
| 6 | 51.570 | 14.060 | 36.090 | | | | 93.748 | 44.271 | 39.610 |
| 7 | 48.187 | 28.543 | 18.797 | | | | 58.325 | 35.806 | 19.643 |
| 8 | 40.683 | 20.745 | 27.902 | | | | 67.044 | 39.627 | 30.102 |
| 9 | 40.886 | 18.466 | 37.650 | | | | 75.763 | 43.449 | 40.560 |

| Point number | Set 4(m): 2D space | | | Set5 (m): 1D space | | | Set 6 (m): 1D space | | |
|--------------|--------------------|--------|--------|--------------------|--------|--------|---------------------|--------|--------|
| | x^t | y^t | z^t | x^t | y^t | z^t | x^t | y^t | z^t |
| 1 | 60.227 | 13.048 | 14.899 | 44.537 | 25.929 | 18.493 | 30.608 | 28.012 | 19.782 |
| 2 | 60.835 | 11.060 | 24.681 | 59.073 | 21.858 | 26.985 | 31.216 | 26.023 | 29.563 |
| 3 | 61.443 | 9.072 | 34.463 | 73.610 | 17.787 | 35.478 | 31.824 | 24.035 | 39.345 |
| 4 | 54.410 | 20.941 | 16.865 | 88.146 | 13.716 | 43.971 | | | |
| 5 | 55.017 | 18.953 | 26.647 | 102.683 | 9.645 | 52.463 | | | |
| 6 | 55.625 | 16.965 | 36.429 | 117.219 | 5.573 | 60.956 | | | |
| 7 | 48.592 | 28.834 | 18.831 | 131.756 | 1.502 | 69.449 | | | |
| 8 | 49.200 | 26.846 | 28.613 | 146.292 | -2.569 | 77.941 | | | |
| 9 | 49.808 | 24.857 | 38.394 | 160.829 | -6.640 | 86.434 | | | |

Table 12 Computed transformation parameters for Set 1 to Set 4 by ADQA, MPA, and OMA

| Parameters | Set 1 | Set 2 | Set 3 | Set 4 |
|----------------|-----------|-----------|-----------|-----------|
| t_x (m) | 30.000215 | 29.997125 | 29.999564 | 29.999778 |
| t_y (m) | 30.000014 | 29.999418 | 30.000156 | 30.000191 |
| t_z (m) | 9.999992 | 10.000804 | 9.999562 | 9.999647 |
| θ_x (°) | 70.998025 | 70.994443 | 70.999494 | 71.000802 |
| θ_y (°) | 77.999873 | 77.996704 | 77.999588 | 78.000742 |
| θ_z (°) | 73.001648 | 73.000253 | 73.000571 | 72.999769 |
| λ | 1.000012 | 1.000049 | 1.000025 | 1.000028 |
| σ (m) | 0.000315 | 0.000197 | 0.000313 | 0.000294 |

constant interval 0.0001 m, so there are 100 different values of σ_0 . For each value of σ_0 , the noises are produced and added 100 times, and the transformation parameters are computed 100 times. The average computed transformation parameters of 100 times and the error regarding to true transformation parameters are computed based on Monte Carlo simulation method. The results by ADQA, MPA, and OMA are the same and depicted in Fig. 1. It is seen from Fig. 1 that the errors of transformation parameters and standard deviation of unit weight σ all keep a trend of overall increase with the increase of

σ_0 . From the view point of improving the transformation accuracy, high accurate control points (with small errors in the coordinates) should be chosen.

For Scheme 2, the coordinates of control points in the target system are computed by the given coordinates of control points in the original system, fixed scale and translations, and varied rotation angles θ_0 . θ_0 is varied in the zone $[1^\circ, 85^\circ]$ with a constant interval 1° , so there are 85 different values of θ_0 . In order to simulate the errors of control points in the original system and in the target system, normal distributed noises $N(0, 0.001^2)$ are produced and added into the coordinates. For each value of θ_0 , the noises are produced and added 100 times, and the transformation parameters are computed 100 times. The average computed transformation parameters of 100 times and the error regarding to true transformation parameters are computed based on Monte Carlo simulation method. The results by ADQA, MPA, and OMA are the same and depicted in Fig. 2. Figure 2 indicates that the errors of transformation parameters and standard deviation of unit weight σ all keep a trend of overall increase with the increase of θ_0 . From the view point of improving the transformation accuracy, it is preferred to decrease the rotation angles as much as possible.

Table 13 Computed transformation parameters for Set 5 to Set 6 by ADQA, MPA, and OMA

| Parameters | Set 5 | | | Set 6 | | |
|----------------|------------|------------|------------|------------|------------|-----------|
| | ADQA | MPA | OMA | ADQA | MPA | OMA |
| t_x (m) | 30.000278 | 30.000278 | 30.000278 | 30.000000 | 30.000000 | 30.000000 |
| t_y (m) | 30.000389 | 30.000389 | 30.000389 | 30.000333 | 30.000333 | 30.000333 |
| t_z (m) | 10.000083 | 10.000083 | 10.000083 | 10.000333 | 10.000333 | 10.000333 |
| θ_x (°) | 55.330072 | -54.854782 | -45.000000 | -61.858215 | -61.858215 | Unsolved |
| θ_y (°) | 41.175912 | -24.651859 | 16.444350 | 77.998588 | 77.998588 | 77.998588 |
| θ_z (°) | -51.523256 | 6.418995 | 15.645706 | 72.998563 | 72.998563 | 72.998563 |
| λ | 1.000016 | 1.000016 | 1.000016 | 1.000008 | 1.000008 | 1.000008 |
| σ (m) | 0.000296 | 0.000296 | 0.000296 | 0.000407 | 0.000407 | 0.000407 |

Table 14 The settings of two test schemes

| Parameters | Scheme 1 | Scheme 2 |
|--|--------------------|------------------|
| Coordinates of control points in the original system | Set 1 in Table 9 | Set 1 in Table 9 |
| λ | 1.000016 | 1.000016 |
| t_x (m) | 30 | 30 |
| t_y (m) | 30 | 30 |
| t_z (m) | 10 | 10 |
| θ_x (°) | 50 | θ_0 |
| θ_y (°) | 30 | θ_0 |
| θ_z (°) | 80 | θ_0 |
| Noise | $N(0, \sigma_0^2)$ | $N(0, 0.001^2)$ |

Conclusions

The rigid transformation, including rotation and translation, can be described by a unit dual quaternion. Based on this good property of unit dual quaternion the point-wise weighted 3D coordinate transformation is formulated. The model constructed, i.e., Equation (48), is not differential as traditional ones, like MPA and OMA, utilize \tilde{C} to centralize the model and eliminate the three translations. The Analytical Dual Quaternion Algorithm (ADQA) of the point-wise weighted 3D coordinate transformation is derived step by step and summed up.

Four numerical experiments involving two real world cases, namely, geodetic datum transformation and LIDAR point cloud registration, and two simulative cases are demonstrated. The results show that the presented algorithm, i.e., ADQA, as well as MPA and OMA, is valid for the transformation with point-wise and no matter how big the rotation angles are. Meanwhile, the results also indicate that in order to recover all parameters the distribution of common points should be in 3D or 2D space. In other words, all common points should not on a line. From the perspective of improving the transformation accuracy, high

accurate control points (with small errors in the coordinates) should be chosen, and it is preferred to decrease the rotation angles as much as possible.

Appendix Proof of symmetric property of matrix $B^T C$

Equations (58) and (59) are rewritten, respectively, as

$$B = \begin{bmatrix} 0 & -\sum_{i=1}^n \alpha_i z_i^t & \sum_{i=1}^n \alpha_i y_i^t & \sum_{i=1}^n \alpha_i x_i^t \\ \sum_{i=1}^n \alpha_i z_i^t & 0 & -\sum_{i=1}^n \alpha_i x_i^t & \sum_{i=1}^n \alpha_i y_i^t \\ -\sum_{i=1}^n \alpha_i y_i^t & \sum_{i=1}^n \alpha_i x_i^t & 0 & \sum_{i=1}^n \alpha_i z_i^t \\ -\sum_{i=1}^n \alpha_i x_i^t & -\sum_{i=1}^n \alpha_i y_i^t & -\sum_{i=1}^n \alpha_i z_i^t & 0 \end{bmatrix}, \tag{88}$$

$$C = \begin{bmatrix} 0 & \sum_{i=1}^n \alpha_i z_i^o & -\sum_{i=1}^n \alpha_i y_i^o & \sum_{i=1}^n \alpha_i x_i^o \\ -\sum_{i=1}^n \alpha_i z_i^o & 0 & \sum_{i=1}^n \alpha_i x_i^o & \sum_{i=1}^n \alpha_i y_i^o \\ \sum_{i=1}^n \alpha_i y_i^o & -\sum_{i=1}^n \alpha_i x_i^o & 0 & \sum_{i=1}^n \alpha_i z_i^o \\ -\sum_{i=1}^n \alpha_i x_i^o & -\sum_{i=1}^n \alpha_i y_i^o & -\sum_{i=1}^n \alpha_i z_i^o & 0 \end{bmatrix}; \tag{89}$$

thus,

$$B^T C = -BC = N = \begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{14} \\ N_{21} & N_{22} & N_{23} & N_{24} \\ N_{31} & N_{32} & N_{33} & N_{34} \\ N_{41} & N_{42} & N_{43} & N_{44} \end{bmatrix}, \tag{90}$$

where N_{ij} ($i, j = 1, 2, 3, 4$) is the element of N in the i th row and j th column listed as

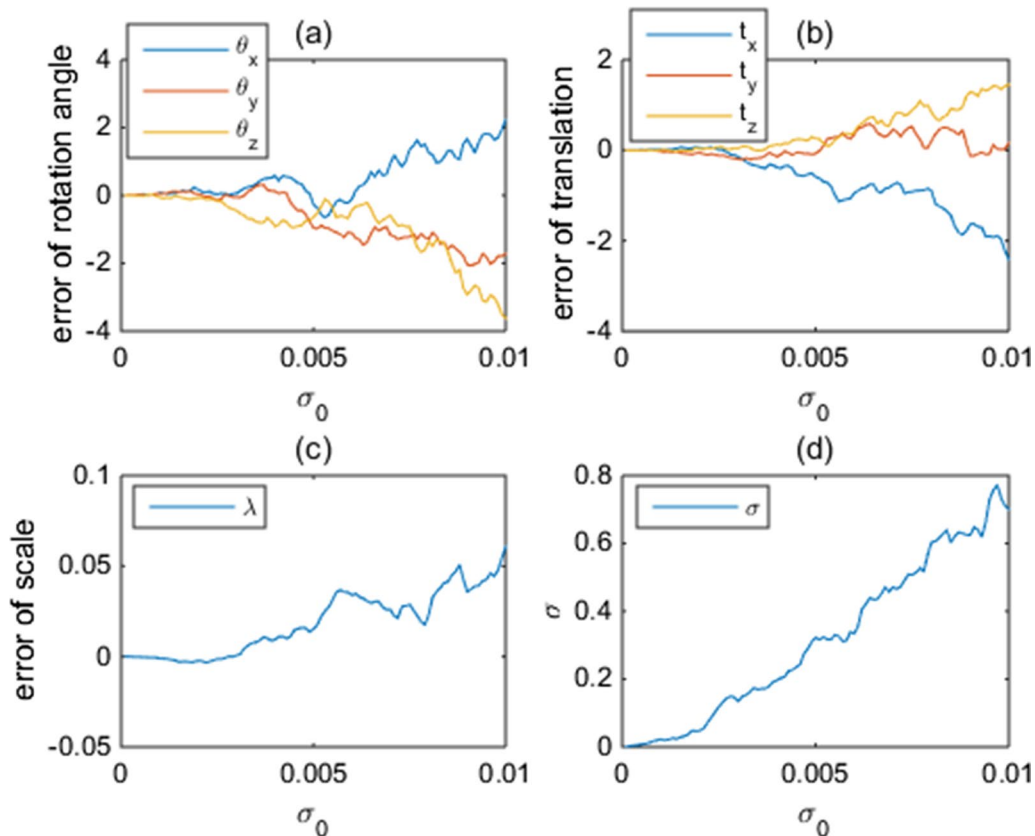


Fig. 1 The errors of computed transformation parameters and standard deviation of unit weight caused by different normal distributed random noises $N(0, \sigma_0^2)$. Each x axis value represents the different σ_0 in meters. Each y axis value represents different accuracy index values: **(a)** error of rotation angle, in degrees. **(b)** Error of translation, in meters. **(c)** error of scale, in no unit. **(d)** σ in meters

$$N_{11} = \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i x_i^o - \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i y_i^o - \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{91}$$

$$N_{22} = -\sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i x_i^o + \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i y_i^o - \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{92}$$

$$N_{33} = -\sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i x_i^o - \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i y_i^o + \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{93}$$

$$N_{44} = \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i x_i^o + \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i y_i^o + \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{94}$$

$$N_{12} = N_{21} = \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i x_i^o + \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i y_i^o, \tag{95}$$

$$N_{13} = N_{31} = \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i x_i^o + \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{96}$$

$$N_{14} = N_{41} = \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i y_i^o - \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{97}$$

$$N_{23} = N_{32} = \sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i y_i^o + \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{98}$$

$$N_{24} = N_{42} = -\sum_{i=1}^n \alpha_i z_i^t \sum_{i=1}^n \alpha_i x_i^o + \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i z_i^o, \tag{99}$$

$$N_{34} = N_{43} = \sum_{i=1}^n \alpha_i y_i^t \sum_{i=1}^n \alpha_i x_i^o - \sum_{i=1}^n \alpha_i x_i^t \sum_{i=1}^n \alpha_i y_i^o. \tag{100}$$

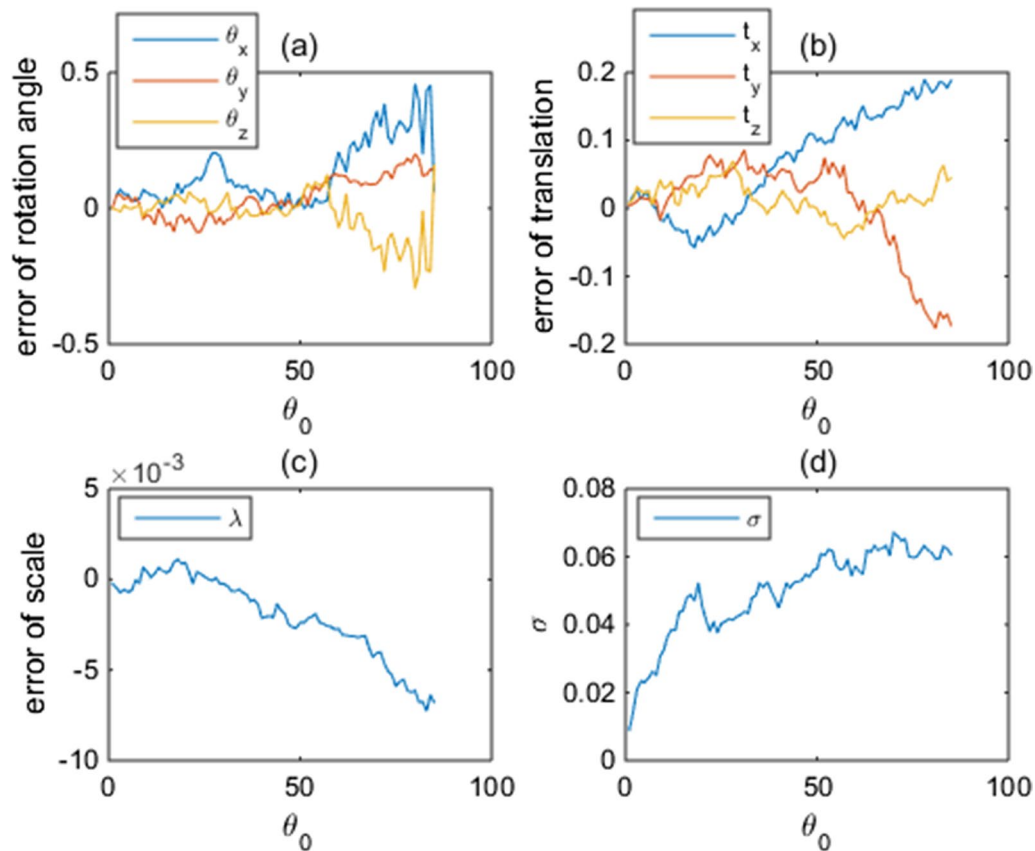


Fig. 2 The errors of computed transformation parameters and standard deviation of unit weight with different rotation angle θ_0 . Each x axis value represents the different θ_0 in degrees. Each y axis value represents different accuracy index: **(a)** error of rotation angle, in degrees. **(b)** Error of translation, in meters. **(c)** Error of scale, in no unit. **(d)** σ in meters

Therefore, $\mathbf{B}^T \mathbf{C}$ is a symmetric matrix.

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 and GC conceived the original ideas; JW, SL, and HH participated in deriving formulae; HZ, ZW, and RY implemented case study; HZ and HH worked on preparing the original draft; HZ and RY reviewed and edited the manuscript; HZ supervised the research work. All the authors read and approved the final manuscript.

Funding

This study is supported jointly by National Natural Science Foundation of China (Grant Nos. 42074005, 42074001, 41861062), the 2021 Science and Technology Project of Hubei Geological Bureau (Grant No. KJ2021-16), 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), Wang et al. (2014), and Zeng (2015).

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Key Laboratory of Geological Hazards on Three Gorges Reservoir Area, Ministry of Education, China Three Gorges University, Yichang 443002, China. ²Hubei Key Laboratory of Disaster Prevention and Mitigation, China Three Gorges University, Yichang 443002, China. ³College of Civil Engineering and Architecture, China Three Gorges University, Yichang 443002, China. ⁴School of Geomatics, East China University of Technology, Nanchang 330013, China. ⁵School of Environmental Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China. ⁶School of Civil Engineering, Chongqing University, Chongqing 400045, China.

Received: 25 August 2022 Accepted: 27 October 2022

Published online: 17 November 2022

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
- Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 14(2):239–256
- Chang G (2016) Closed form least-squares solution to 3D symmetric Helmert transformation with rotational invariant covariance structure. *Acta Geod Geophys* 51(2):237–244
- 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
- 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
- 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>
- Kanatani K, Niitsuma H (2012) Optimal computation of 3-D similarity: Gauss-Newton vs Gauss-Helmert. *Comput Stat Data Anal* 56:4470–4483
- 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
- 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
- Ligas M, Prochniewicz D (2019) Procrustes based closed-form solution to the point-wise weighted rigid-body transformation in asymmetric and symmetric cases. *J Spat Sci*. <https://doi.org/10.1080/14498596.2019.1684394>
- Mahboub V (2016) A weighted least-squares solution to a 3-D symmetrical similarity transformation without linearization. *Stud Geophys Geod* 60:195–209
- Marx C (2017) A weighted adjustment of a similarity transformation between two point sets containing errors. *J Geod Sci* 7(1):105–112. <https://doi.org/10.1515/jogs-2017-0012>
- 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
- Päun 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>
- Ruffhead AC (2021) Derivation of rigorously-conformal 7-parameter 3D geodetic datum transformations. *Surv Rev* 53(376):8–15. <https://doi.org/10.1080/00396265.2019.1665614>
- Shen YZ, Chen Y, Zheng DH (2006) A quaternion-based geodetic datum transformation algorithm. *J Geod* 80:233–239
- 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 Understand* 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
- Wang Q, Chang G, Xu T, Zou Y (2018) Representation of the rotation parameter estimation errors in the Helmert transformation model. *Surv Rev* 50:69–81
- Yang R, Deng C, Yu K, Li Z, Pan L (2022) A new way for Cartesian coordinate transformation and its precision evaluation. *Remote Sens* 14:64. <https://doi.org/10.3390/rs14040864>
- 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 Planets Space* 67:105. <https://doi.org/10.1186/s40623-015-0263-6>
- 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 Planets 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 Planets Space* 72:53. <https://doi.org/10.1186/s40623-020-01179-1>
- Zeng HE, He HW, Chen LG, Chang GB, He HQ (2022) Extended WTLS iterative algorithm of 3D similarity transformation based on Gibbs vector. *Acta Geod Geophys* 57:43–61

Publisher's Note

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

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)