



# A new algorithm for 3D similarity transformation with dual quaternion

Sebahattin Bektas<sup>1</sup>

Received: 23 November 2021 / Accepted: 9 June 2022  
© Saudi Society for Geosciences 2022

## Abstract

3D coordinate transformation is frequently encountered in geodesy applications. In addition, it is frequently encountered in computer vision, robotics, engineering surveying, and GIS applications. Seven parameter Helmert transformation is one of the widely used 3D transformation algorithms. Nowadays, the quaternion algorithms are finding increasingly common usage due to some disadvantages of Helmert transformation based on Euler angles; in this article, a new 3D Cartesian coordinate transformation with the dual quaternion method is explained in detail, and its advantages over the classical transformation problem algorithm are emphasized. Quaternions are not preferred in 3D transformation applications due to the complexity of the subject. As a result, readers generally stay away from the subject of quaternion. In this article, a new dual quaternion algorithm with eight parameters (DQA) is presented. In addition, a MATLAB function *DQA\_3d\_transformation* will be introduced in the article so that readers can easily perform 3D transformations with dual quaternions. We hope that after reading this article, the reader's perspective on dual quaternions may change slightly.

**Keywords** 3D similarity transformation · Dual quaternion · Euler rotation angle · Least squares method, Gimbal lock

## Introduction

3D coordinate transformations play an important role in many fields of geodesy, photogrammetry, robotic, and computer vision. There are a wide variety of transformation models, for example, Bursa–Wolf and Molodensky–Badekas. They are also called similarity or seven-parameter transformations and they combine a scale change, three axes-rotations, and three origin-shifts in a practical mathematical model of the relationships between points in two different 3D coordinate systems. They differ slightly in their operation; the Molodensky–Badekas transformation uses a centroid but the Bursa–Wolf transformation does not; hence, additional information (the centroid coordinates) is required when using the Molodensky–Badekas transformation; a factor that makes the Bursa–Wolf transformation more popular (Deakin 2006). Even wider 3D coordinate transformation models are available. Most generally, we can talk about the

twelve parameter affine transformation: three translations, three Euler rotation angles, three scale factors, and three skew (affinity or non-perpendicularity) parameters along x, y, z coordinate axes (Amiri-Simkooei 2018; Even-Tzur 2018). In the widely adapted seven parameter-similarity transformation, (the so-called Helmert transformation in geodetic studies), no affinity is involved and only one common scale factor is defined. There are also some other transformations such as 9-parameter affine and 8-parameter affine transformations (Andrei 2006; Even-Tzur 2018, 2020; Wang et al. 2018; Závoti and Kalmár 2016; Zeng 2015).

Rotations through all three axes, 3D vector displacement, and single-scale change are calculated in order to transfer points from one system to another. These seven transformation parameters are known as the seven-parameter transformation or Helmert transformation problem, which is a well-known transformation, not only in engineering but also in other discipline sciences (Zeng and Yi 2011).

Jitka (2011) presents a dual quaternion algorithm for geodetic datum transformation. Its solution is complex and the solution process of transformation is not explicit. Zeng et al. (2018) propose a DQA solution with closed formula. This solution needs an eigen-value decomposition algorithm. Uygur et al. (2020) describe how to solve the symmetric and asymmetric 3D similarity transformation based on quaternion (not dual quaternion). In a recent study, Ioannidou and

---

Responsible Editor: Longjun Dong

✉ Sebahattin Bektas  
sbektas@omu.edu.tr

<sup>1</sup> Faculty of Engineering, Geomatics Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey

Pantazis (2020) include a 9-parameter iterative DQA algorithm. In addition, the results of Helmert transformation method, quaternion method, and dual quaternion method are also compared in that study. As a result of the comparison, they could not mention a certain superiority of the methods over each other. It is seen that 3D similarity transformation has been used in the fields of geophysics and geology in recent years, such as the estimation of geo-hazard (Abdel Azeem et al. 2014), empty region identification (Dong et al. 2021a), and detection of abnormal regions (Dong et al. 2021b)

The most important part of the transformation problem is the stage of determining the transformation parameters between two systems (reverse problem). The transformation parameters are realized by using control points whose coordinates are known in both systems. If the number of control points at hand is more than necessary, a least squares (LS) solution is done (the number of control points should be great than two). If the transformation parameters between the two coordinate systems are known, it is easy to do the transformation between the two systems.

Due to the complexity of the subject, readers generally stay away from the subject of quaternion. We hope that after reading this article, the reader will stay closer to the topic of dual quaternions (Kenwright 2012). Since quaternions are based on complex numbers, (complex numbers are the largest set of numbers), we hope that quaternions will be used much more in many different fields in the future. This article will show a simple algorithm how to do 3D transformation with eight quaternions. Ioannidou and Pantazis (2020). In this way, the unknown parameters are nine: four for each quaternion that represents the rotation and translation and one for scale factor. Contrary to Ioannidou and Pantazis (2020) even if the DQA algorithm seems to have nine parameters, the number of independent parameters is actually eight. Because the scale parameter  $\lambda$  is directly dependent on rotation quaternions.

In terms of creating a base for the subject, first the classical Helmert 3D transformation will be discussed; then, we will move on to the DQA method.

In this paper, first is Helmert 7-parameter transformation based on Euler rotation angles and second is the advantages of quaternion and disadvantages of using Euler angles in transformation; thirdly, the quaternions are introduced in brief; fourthly, the mathematic model of eight parameters DQA model is established, and then, a new DQA algorithm of 3D Helmert similarity transformation is presented.

### Three-dimensional Helmert similarity transformation

The states of two three-dimensional orthogonal coordinate systems (source or 1<sup>st</sup> and target or 2<sup>nd</sup> system) with respect to each other can be explained with seven datum parameters

(Fig. 1). These—the three translation ( $tx, ty, tz$ ) in the direction of the coordinate axes—are the parameters of the three rotations ( $\epsilon, \psi, \omega$ ) around the coordinate axes and a scale ( $\lambda$ ) between the coordinate systems (Zeng and Yi 2011; Watson 2006; Bektaş 2017).

The coordinates of a P point are the vector  $x$  in the  $(x, y, z)$  1<sup>st</sup> system, the vector  $X$  in the  $(X, Y, Z)$  2<sup>nd</sup> system and, the relationship between them,

$$\underline{X} = \underline{t}_0 + \lambda \underline{R} \underline{x} \tag{1}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{P_i} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \lambda \underline{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{P_i} \quad \underline{R}^* = \begin{bmatrix} 1 & w & -\psi \\ -w & 1 & \epsilon \\ \psi & -\epsilon & 1 \end{bmatrix} \tag{2}$$

It is defined by the relation. Here,

$\underline{t}_0$ :  $(x,y,z)$  coordinates of that origin point of the orthogonal coordinate system in the  $(X, Y, Z)$  system

$$\underline{t}_0 = [t_x, t_y, t_z]^T \tag{3}$$

$R$  is the rotation matrix based on the counter clockwise rotation angles  $\epsilon, \psi, \omega$  around the  $x, y, z$  axes. If the rotation angles between the two systems are very small (in a differential sense), the  $R^*$  rotation matrix can be used instead of the  $R$  rotation matrix.

The rotation matrix  $R$  is equal to the product of the rotation matrix in reverse order, showing the transformation effects of the rotation angles  $\epsilon, \psi, \omega$  around the  $x, y, z$  axes.

$$\begin{aligned} \underline{R}_1(\epsilon) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \epsilon & \sin \epsilon \\ 0 & -\sin \epsilon & \cos \epsilon \end{bmatrix}, \\ \underline{R}_2(\psi) &= \begin{bmatrix} \cos \psi & 0 & -\sin \psi \\ 0 & 1 & 0 \\ \sin \psi & 0 & \cos \psi \end{bmatrix}, \\ \underline{R}_3(\omega) &= \begin{bmatrix} \cos \omega & \sin \omega & 0 \\ -\sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4}$$

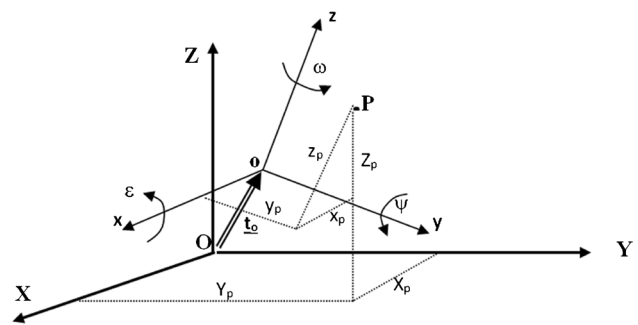


Fig. 1 Three-dimensional coordinate transformation

$$\underline{R} = R_3(\omega)R_2(\psi)R_1(\epsilon)$$

$$\underline{R} = \begin{bmatrix} \cos \psi \cos \omega & \cos \epsilon \sin \omega + \sin \epsilon \sin \psi \cos \omega & \sin \epsilon \sin \omega - \cos \epsilon \sin \psi \cos \omega \\ -\cos \psi \sin \omega & \cos \epsilon \cos \omega - \sin \epsilon \sin \psi \sin \omega & \sin \epsilon \cos \omega + \cos \epsilon \sin \psi \sin \omega \\ \sin \psi & -\sin \epsilon \cos \psi & \cos \epsilon \cos \psi \end{bmatrix} \tag{5}$$

The transformation parameters ( $t_x, t_y, t_z, \lambda, \epsilon, \psi, \omega$ ) must be known in order to make a transformation between two three-dimensional orthogonal coordinate systems. If two points whose three coordinates are known in both coordinate systems and the only one coordinate of another point are known in both systems, an algebraic solution can be made. If at least three control points have all coordinates, the unknown parameters are determined according to the LS method.

The basic equation of three-dimensional similarity transformation is

$$\underline{X} = \underline{t}_o + \lambda \underline{R} \underline{x} \tag{6}$$

Above the expression is not linear with respect to the unknown's transformation parameters; the approximate values of the unknowns are selected and linearized by opening to the Taylor series. In cases where approximate values are not chosen properly or cannot be chosen, the adjustment process is repeated and iteration is done. If there is no information about the approximate values of the transformation parameters, the zero value for the translation and rotation parameters can be taken as zero ( $t_{x0}, t_{y0}, t_{z0}, \epsilon_0, \psi_0, \omega_0 = 0$ ) for the scale parameter  $\lambda_0=1$  can be taken. The derivative matrices required for linearization are as follows:

$$\underline{R}_\epsilon = \frac{\partial \underline{R}}{\partial \epsilon} = \begin{bmatrix} 0 & -\sin \epsilon \sin \omega + \cos \epsilon \sin \psi \cos \omega & \cos \epsilon \sin \omega + \sin \epsilon \sin \psi \cos \omega \\ 0 & -\sin \epsilon \cos \omega - \cos \epsilon \sin \psi \sin \omega & \cos \epsilon \cos \omega - \sin \epsilon \sin \psi \sin \omega \\ 0 & -\cos \epsilon \cos \psi & \sin \epsilon \cos \psi \end{bmatrix}$$

$$\underline{R}_\psi = \frac{\partial \underline{R}}{\partial \psi} = \begin{bmatrix} -\sin \psi \cos \omega & \sin \epsilon \cos \psi \cos \omega & \cos \epsilon \cos \psi \cos \omega \\ \sin \psi \sin \omega & -\sin \epsilon \cos \psi \sin \omega & \cos \epsilon \cos \psi \sin \omega \\ \cos \psi & \sin \epsilon \sin \psi & -\cos \epsilon \sin \psi \end{bmatrix} \tag{7}$$

$$\underline{R}_\omega = \frac{\partial \underline{R}}{\partial \omega} = \begin{bmatrix} -\cos \psi \sin \omega & \cos \epsilon \cos \omega - \sin \epsilon \sin \psi \sin \omega & \sin \epsilon \cos \omega + \cos \epsilon \sin \psi \sin \omega \\ \cos \psi \cos \omega & -\cos \epsilon \sin \omega - \sin \epsilon \sin \psi \cos \omega & -\sin \epsilon \sin \omega + \cos \epsilon \sin \psi \cos \omega \\ 0 & 0 & 0 \end{bmatrix}$$

Assuming that only the coordinates in the 2nd system are loaded with error in the transformation model

$$\underline{X} + v_x = \underline{t}_o + \lambda \underline{R} \underline{x} \tag{8}$$

If this nonlinear equation is opened to Taylor series and linearization is done, a linear correction equation system is obtained in the following structure.

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \underbrace{\begin{bmatrix} I_{3 \times 3} & R \\ y & \lambda_0 R_\epsilon \\ z & \lambda_0 R_\psi \\ z & \lambda_0 R_\omega \end{bmatrix}}_A \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta \lambda \\ \delta \epsilon \\ \delta \psi \\ \delta \omega \end{bmatrix} + \underbrace{\begin{bmatrix} t_{x0} \\ t_{y0} \\ t_{z0} \end{bmatrix} + (\lambda_0) \underline{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{-l} \tag{9}$$

Correction equations to be written for a control point are shown above. If these correction equations are written for all control points, a linearized functional model of the indirect measure compensation in the form of  $v = A \delta x - l$  is obtained.

$P_{3n \times 3n}$  weight matrix of control points coordinates, ( $n$ : number of control points) in accordance with principle of adjustment LS

$$v^T P v = \min .$$

Normal equations are established

$$A^T P A \delta x - A^T P l = 0 \tag{10}$$

$$\delta x = (A^T P A)^{-1} A^T P l \tag{11}$$

$\delta x$  is the vector of unknowns, the transformation parameters

$$\delta x = (A^T A)^{-1} A^T l \text{ or } \delta x = A \setminus l : \text{ if } (P = I) \tag{12}$$

$$\delta x = [\delta_x, \delta_y, \delta_z, \delta \Delta, \delta \epsilon, \delta \psi, \delta \omega]^T$$

The exact transformation parameters are found by adding the calculated transformation parameters to their approximate values.

$$\begin{aligned}
 t_x &= t_{x0} + \delta_x & t_y &= t_{y0} + \delta_y & t_z &= t_{z0} + \delta_z \\
 \omega &= \omega_0 + \delta\omega & \psi &= \psi_0 + \delta\psi & \varepsilon &= \varepsilon_0 + \delta\varepsilon \\
 \lambda &= \lambda_0 + \delta\lambda_0
 \end{aligned}
 \tag{13}$$

If the initial approximate values of the transformation parameters are not suitable enough, these calculated parameters will not provide transformation equality Eq. (8). In this case, the exact transformation parameters found at the end of the first adjustment are approximated and the adjustment is renewed, that is, iteration. The iteration process continues until the result checks hold. The appropriateness of initial approximate values affects the number of iterations.

After the transformation parameters are calculated with sufficient accuracy, the corrections to be brought to the control point coordinates are calculated by placing the  $v = A \delta x - l$  functional model established for the transformation. In the transformation of new points, these adjustment transformation parameters are used. If necessary, precision calculations are made for the transformation process.

### Advantages of quaternions and disadvantages of using Euler angles in rotation

The representation of a rotation as a quaternion (4 numbers) is always more meaningful and more simple than the representation of a rotation as an orthogonal matrix (9 numbers) consisting of Euler rotation angles.

Furthermore, for a given axis and angle, one can easily construct the corresponding quaternion, and conversely, for a given quaternion, one can easily read off the axis and the angle. Both of these are much harder with rotation matrices consisting of Euler angles.

Major disadvantages of using Euler angles in transformation;

- since the Euler angles are composed of sinus and cosines trigonometric functions, they are always dual-solution,
- the sinus function changes rapidly in small rotation angles,
- there is a possibility of gimbal lock problem when using Euler angles

Gimbal lock problem may occur when using Euler rotation angles (when the rotation around the Y axis is  $\pm 90$  degrees). Two axes overlap and lose their independence, which is called “loss of degree of freedom.” Quaternions are used to avoid this problem. Quaternions have a 4th value in addition to Euler angles. This value is a scalar value, which prevents conflicts. For further information on gimbal locks, please refer to Velsink (2015) and Uygur et al. (2020).

A quaternion can be represented as “ $q_1 i + q_2 j + q_3 k + q_0$ ”

Thanks to the scalar value next to the 3 vector values, each rotation matrix indicates to a single angle in the image set, and the conflict problem is solved.

### Mathematical model of 3D transformation based on dual quaternion

Quaternion was invented by Irish mathematician Hamilton in 1843 (Liu 2004), which is generally expressed as follows:

$$q = q_1 i + q_2 j + q_3 k + q_0 \tag{14}$$

where  $q_1, q_2, q_3,$  and  $q_0$  are real numbers,  $i, j,$  and  $k$  are basic quaternion units, and they meet following properties (Zeng 2019) .

$$\begin{aligned}
 i^2 &= j^2 = k^2 = -1 \\
 ij &= -ji = k \\
 jk &= kj = i \\
 ki &= -ik = j \\
 ijk &= -1
 \end{aligned}
 \tag{15}$$

in order to evaluate the rotation matrix from quaternions

$$R = (qo^2 - q^T \cdot q) I_{3 \times 3} + 2 (q \cdot q^T \cdot) + q_0 \cdot C(q) \tag{16}$$

where

$$q = [q_1 \ q_2 \ q_3]^T$$

$I_{3 \times 3}$  is the unit matrix

$$C(q) = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \tag{17}$$

The relations between the elements of the rotation matrix R and the quaternions are as follows:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{18}$$

$$\begin{aligned}
 q_0^2 &= \frac{1}{4} (1 + r_{11} + r_{22} + r_{33}) \\
 q_1^2 &= \frac{1}{4} (1 + r_{11} - r_{22} - r_{33}) \\
 q_2^2 &= \frac{1}{4} (1 - r_{11} + r_{22} - r_{33}) \\
 q_3^2 &= \frac{1}{4} (1 - r_{11} - r_{22} + r_{33})
 \end{aligned}
 \tag{19}$$

The rotation angles from the quaternions

$$\begin{aligned} \varepsilon &= \operatorname{atan2}(-2(q_0 q_1 + q_2 q_3), (q_0^2 + q_1^2 - q_2^2 - q_3^2)) \\ \psi &= \operatorname{asin}(2(-q_0 q_2 + q_3 q_1)) \\ \omega &= \operatorname{atan2}(-2(q_0 q_3 + q_2 q_1), (q_0^2 + q_1^2 - q_2^2 - q_3^2)) \end{aligned} \tag{20}$$

The essence of this work is the introduction of a new dual quaternion algorithm. In order not to get away from the essence, we will not go into further detail on quaternions. There is a wealth of information about quaternions in the literature (Jitka 2011; Hamilton 1853; Liu 2004; Wang et al. 2014; Zeng 2019).

Coordinate transformation with the dual quaternions method extends to the 1882s. Clifford (2007) created dual quaternions for both rotation and translation on a single model.

$$q = r + s \cdot \varepsilon \tag{21}$$

where  $r$  and  $s$  are quaternions.

$$s.t. \ \varepsilon^2 = 0 \text{ and } \varepsilon \neq 0$$

In order to determine a new position of a point, through the dual quaternions, the translation and rotation properties for the unit dual quaternion were defined as  $q_r = r$  and  $s = t \cdot r$

$$s = [s_0 \ s_1 \ s_2 \ s_3]^T$$

with  $r$  the unit quaternion that represents the rotation and  $t = (tx, ty, tz, 0)$  the quaternion that represents the translation (Zeng 2019; Jitka 2011). Utilizing the dual quaternion theory, the R matrix can be determined from the relation:

$$\begin{aligned} R &= (r_0^2 - r^T \cdot r) I_{3 \times 3} + 2(r \cdot r^T) + r_0 \cdot C(r) \\ r &= [r_1 \ r_2 \ r_3]^T \end{aligned} \tag{22}$$

$$Q(r) = \begin{bmatrix} r_0 I + C(r) & r \\ -r^T & r_0 \end{bmatrix} \quad W(r) = \begin{bmatrix} r_0 I - C(r) & r \\ -r^T & r_0 \end{bmatrix} \tag{23}$$

Finally, we get DQA transformation equality as follows:

$$\begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}_{P_i} = 2 \cdot W_{(r)}^T \cdot s + \lambda W_{(r)}^T \cdot Q_{(r)} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}_{P_i} \tag{24}$$

The number of unknown parameters appears to be nine in the above transformation equation. These are four for each quaternion that represents the rotation and translation and one for scale factor, whereas in reality, the number of unknown parameters is eight because the scale parameter  $\lambda$  is directly dependent on rotation quaternions.

$$\lambda = r_0^2 + r_1^2 + r_2^2 + r_3^2 \tag{25}$$

If we put above Eq. (25) into Eq. (24), we get DQA transformation equality with eight parameters as follows:

$$\begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}_{P_i} = 2 \cdot W_{(r)}^T \cdot s + (r_0^2 + r_1^2 + r_2^2 + r_3^2) W_{(r)}^T \cdot Q_{(r)} \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}_{P_i} \tag{26}$$

As above, the LS method is used to calculate both the rotation angles and the translation parameters as well as their error.

For the  $i$ -th point, DQA transformation can be written from Eq. (26)

The transformation equation (Eq. (26)) is not linear. Approximate values of the unknowns are required to linearize the transformation equation. When choosing the first approximate values of the unknowns, only  $r_{0,0} = 1$  and others can be taken as zero.

$$r_{1,0} = r_{2,0} = r_{3,0} = s_{0,0} = s_{1,0} = s_{2,0} = s_{3,0} = 0$$

The transformation model in Eq. (26) is then linearized with respect to these quaternion elements ( $r_0, r_1, r_2, r_3, s_0, s_1, s_2, s_3$ ) as follows:

We assume that only the coordinates of the 2nd system are loaded with error for the LS adjustment

$$v = A_0 \delta x - l_0 \tag{27}$$

where  $(\cdot)_0$  denotes the approximate value;

- $n$  number of control points
- $A$  is the  $4n \times 8$  matrix of coefficient of the parameters of interest
- $\delta x_{8 \times 1}$  is the vector of the parameters
- $v_{4n \times 1}$  is the vector of corrections
- $l_0$   $4n \times 1$  is the vector of fixed terms.

If at least three control points have all coordinates, the unknown parameters are determined according to the least squares (LS) method.

Equation (27) correction equations to be written for each control point are shown above.

A linearized functional model of the indirect measure compensation in the form of  $v = A \delta x - l$  is obtained.

$P$  weight matrix of control points coordinates in accordance with principle of LS adjustment

$$v^T P v = \min . \tag{28}$$

Normal equations are established

$$A^T P A \delta x - A^T P l = 0 \tag{29}$$

$\delta x = (A^T P A)^{-1} A^T P l$  :  $\delta x$  is the vector of unknowns, the transformation parameters

$$\delta x = (A^T A)^{-1} A^T l \quad \text{or} \quad \delta x = A \setminus l \quad : \text{if } (P = I) \tag{30}$$

$$\delta x = [\delta r_0 \ \delta r_1 \ \delta r_2 \ \delta r_3 \ \delta s_0 \ \delta s_1 \ \delta s_2 \ \delta s_3]^T$$

The exact quaternions are found by adding the calculated differential quaternions to their approximate values.

$$r_0 = r_{0,0} + \delta r_0 \quad r_1 = r_{1,0} + \delta r_1 \quad r_2 = r_{2,0} + \delta r_2 \quad r_3 = r_{3,0} + \delta r_3 \tag{31}$$

$$s_0 = s_{0,0} + \delta s_0 \quad s_1 = s_{1,0} + \delta s_1 \quad s_2 = s_{2,0} + \delta s_2 \quad s_3 = s_{3,0} + \delta s_3 \tag{32}$$

and scale parameter

$$\lambda = r_0^2 + r_1^2 + r_2^2 + r_3^2 \tag{33}$$

If the initial approximate values of the transformation parameters are not suitable enough, these calculated parameters will not provide transformation equality. In this case, the exact transformation parameters found at the end of the first adjustment are approximated and the adjustment is renewed, that is, iteration. The iteration process continues until the result checks hold. The appropriateness of initial approximate values affects the number of iterations.

After the transformation parameters are calculated with sufficient accuracy, the corrections to be brought to the control point coordinates are calculated by placing the  $v = A \delta x - l$  functional model established for the transformation. In the transformation of new points, these adjustment transformation parameters are used. If necessary, precision calculations are made for the transformation process.

According to the presented algorithm, the quaternions and scale values found as a result of calculation are normalized values. Their actual values ( $\bar{r}, \bar{s}, \bar{\lambda}$ ) are determined below.

$$\bar{r}_i = r_i / \sqrt{\lambda}, \quad \bar{s}_i = s_i / \sqrt{\lambda} \quad (i = 0, 1, 2, 3) \tag{34}$$

$$\bar{\lambda} = \lambda^2 \tag{35}$$

After construct  $W(\bar{r})$  and  $Q(\bar{r})$  matrix by Eq. (23), finally, we get dual quaternion transformation equality with eight parameters as follows:

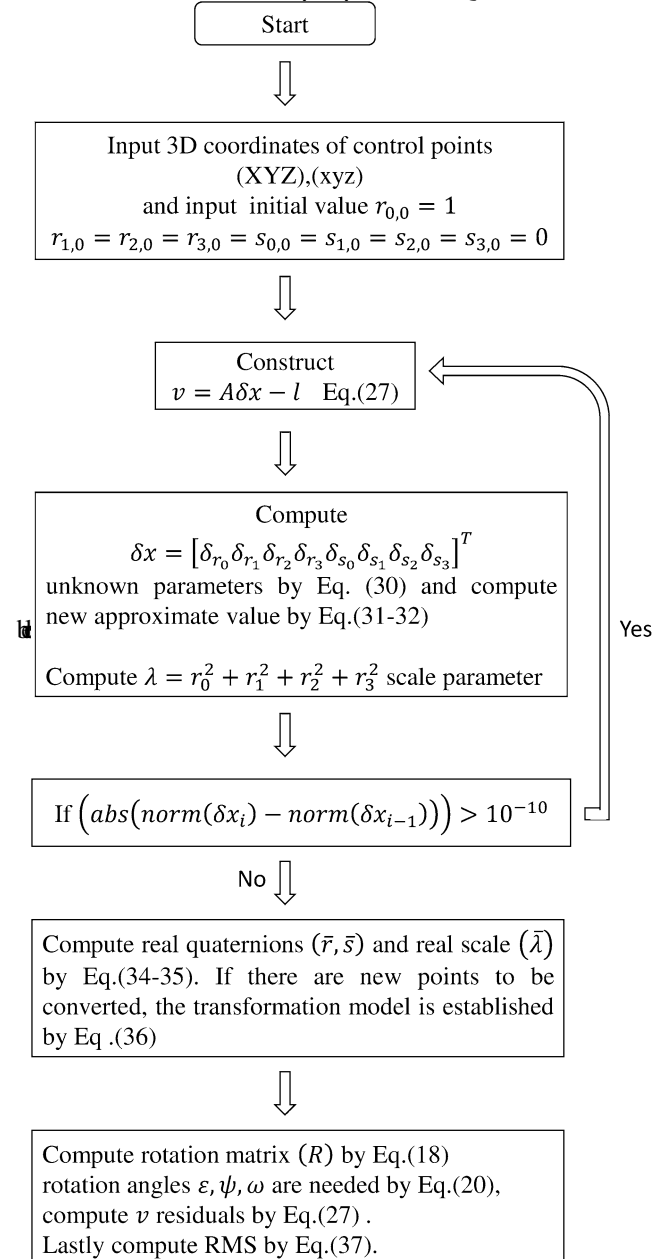
$$\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}_{P_i} = 2 \cdot W_{(\bar{r})}^T \cdot \bar{s} + \bar{\lambda} W_{(\bar{r})}^T \cdot Q_{(r)} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}_{P_i} \tag{36}$$

Compute corrections  $v = A \delta x - l$

Root mean square error

$$rms = \sqrt{\frac{v^T P v}{3n - 7}} \tag{37}$$

The flowchart of proposed DQA method is



Therefore, the algorithm of proposed DQA method is presented in Table 1.

### DQA\_3d\_transformation MATLAB function

The function *DQA\_3d\_transformation* performs both reverse and direct problem solving based on dual quaternions between two Cartesian coordinate systems. The 3D



**Table 1** A dual quaternion algorithm

---

*Initiation:* input 3D coordinates of control points (XYZ), (xyz) and initial value  
 $r_{0,0} = 1$  and  $r_{1,0} = r_{2,0} = r_{3,0} = s_{0,0} = s_{1,0} = s_{2,0} = s_{3,0} = 0$   
*Step 1* Construct  $v = A \delta x - l$  by Eq. (27)  
*Step 2* Compute  $\delta x = [\delta r_0 \delta r_1 \delta r_2 \delta r_3 \delta s_0 \delta s_1 \delta s_2 \delta s_3]^T$  unknown parameters by Eq. (30) and compute new approximate value by Eqs. (31–32)  
*Step 3* Compute  $\lambda = r_0^2 + r_1^2 + r_2^2 + r_3^2$  scale parameter  
*Step 4* Compute norm ( $\delta x_i$ ) (subscript i denotes the iterative number).  
 If  $(\text{abs}(\text{norm}(\delta x_i) - \text{norm}(\delta x_{i-1})) > \text{predefined threshold (e.g., } 10^{-10}\text{)})$ , turn to Step 1, otherwise turn to Step 5  
*Step 5* Compute real quaternions  $(\bar{r}, \bar{s})$  and scale  $(\bar{\lambda})$  by Eqs. (34–35). If there are new points to be converted, the transformation model is established by Eq. (36) and lastly transformation is done.  
*Step 6* Compute R by Eq.(18) furthermore, if rotation angles  $\epsilon, \psi, \omega$  are needed by Eq. (20), compute v residuals by Eq. (27). Lastly compute rms by Eq. (37).

---

similarity transformation parameters are eight quaternions. The quaternions are estimated from control points coordinates with least-squares sense. Three translations, three rotation angles, and one scale factor are computed from these quaternions. The function works for any translation, rotation angle, and scale factor and does not need approximate values. The use of this function is in the form of  $[q, L, T, \text{angles}, \text{res}, \text{rms}, \text{xyz3}] = \text{DQA\_3d\_transformation}(\text{xyz1}, \text{xyz2})$  where xyz1 denotes the [x y z] matrix of coordinates in the source system; xyz2 denotes the [X Y Z] matrix of coordinates in the target system; if there is also a new point to be converted, these points are added to the xyz1 matrix. The output of the function is explained in the help of this function. The function DQA\_3d\_transformation can be downloaded freely from MATLAB file-exchange. The link below can be used to download the function. [https://www.mathworks.com/matlabcentral/fileexchange/90711-dual-quaternions-3d-transformation?s\\_tid=srchtitle](https://www.mathworks.com/matlabcentral/fileexchange/90711-dual-quaternions-3d-transformation?s_tid=srchtitle)

**Numerical example**

The data are chosen (Grafarend and Awange (2003)) as seven control points in this case; the 3D coordinates of control point in 1<sup>st</sup> and 2<sup>nd</sup> systems are listed in Table 2.

The weight of control points is equal ( $P=I$ ). In order to make comparisons between different transformation

methods, the transformation problem in question is solved separately with classical Helmert 3D transformation, quaternion-based transformation, and dual quaternion-based transformation (3 different ways). The same results were obtained in all methods. It was observed that there was no significant difference between the different method solutions. The results are listed in Table 3. Residual (correction) matrix of control points is listed in Table 4.

The same residual values were obtained in the 5 different calculations above. When Table 3 is examined, it was observed that the results were compatible and there was no significant difference between them. It is observed that there are only minor differences between Jitka (2011) and others. It is thought that these small differences may be due to the algorithm used.

**Conclusion**

Expecting accuracy in coordinate transformation operations depends on other factors besides choosing a good transformation model. As the number of parameters increases in transformation models, the transformation accuracy is higher. In all coordinate transformations, if the coordinates consist of large numbers, there is a loss of precision even if the calculations are performed with double precision. It is recommended to use the point coordinates of both systems

**Table 2** Coordinate of control points (XYZ)<sub>1</sub> and (XYZ)<sub>2</sub> system

Station name	X <sub>1</sub>	Y <sub>1</sub>	Z <sub>1</sub>	X <sub>2</sub>	Y <sub>2</sub>	Z <sub>2</sub>
Solitude	4157222.5430	664789.3070	4774952.0990	4157870.2370	664818.6780	4775416.5240
Buoch Zeil	4149043.3360	688836.4430	4778632.1880	4149691.0490	688865.7850	4779096.5880
Hohenneuffen	4172803.5110	690340.0780	4758129.7010	4173451.3540	690369.3750	4758594.0750
Kuehlenberg	4177148.3760	642997.6350	4760764.8000	4177796.0640	643026.7000	4761228.8990
Ex Mergelacc	4137012.1900	671808.0290	4791128.2150	4137659.5490	671837.3370	4791592.5310
Ex Hof Asperg	4146292.7290	666952.8870	4783859.8560	4146940.2280	666982.1510	4784324.0990
Ex Kaisersbach	4138759.9020	702670.7380	4785552.1960	4139407.5060	702700.2270	4786016.6450

**Table 3** Computed transformation parameters (identical weight) with different methods

	Helmert 3D transformation	Quaternion transformation (Uygur et al. 2020)	Dual quaternion transformation		
			(Jitka 2011)	(Zeng 2019)	Proposed method
$t_x$ [m]	641.8804	641.8804	641.8908	641.8804	641.8804
$t_y$ [m]	68.6553	68.6553	68.6570	68.6553	68.6553
$t_z$ [m]	416.3982	416.3981	416.4101	416.3981	416.3981
$\lambda$	1.000005582	1.000005582	1.000005582	1.000005582	1.000005582
$\varepsilon$ [°]	-0.00027736	-0.00027736	-0.00027736	-0.00027736	-0.00027736
$\psi$ [°]	0.00024825	0.000248247	0.000248250	0.000248248	0.000248247
$\omega$ [°]	0.00027585	0.0002758589	0.000275858	0.000275859	0.0002758589
$r_0$		0.9999999999	-0.9999999	1.000000000	0.9999999999
$r_1$		-0.000002407	0.000002420	0.000002420	0.000002420
$r_2$		-0.000002166	0.000002166	-0.000002166	-0.000002166
$r_3$		0.0000024204	-0.00000240	-0.000002407	-0.000002407
$s_0$				320.9406	320.9406
$s_1$				34.3289	34.3289
$s_2$				208.1983	208.1983
$s_3$				-0.00020124	-0.00020124
<i>rms</i> [m]	0.0772	0.0772	0.0772	0.0772	0.0772

by shifting them to the center of gravity separately. Thus, loss of sensitivity will be relatively prevented since working with small coordinates. The precision of the transformation process depends on the distribution and the distance from the center of gravity of the new point to be transformed. Better results can be obtained in cases where common points are homogeneously distributed and surround the new points to be transformed. When moving away from control points, sensitivity decreases in coordinate transformation processes (Bektaş 2017).

On the other hand, no study showing the superiority of the DQA method over classical 3D methods in terms of accuracy has been noted. This is to be expected, because although they have different transformation methods, the aim is to find the existing 7 parameters between two coordinate systems according to LS solution. The presented iterative DQA algorithm is faster convergence than Helmert

method. DQA does not need approximate values of quaternions for solution. The disadvantages of the Helmert method based on Euler angles were mentioned. The DQA method removes these disadvantages. Finding translation and rotation parameters as one meaning at once is the superiority of DQA method over other transformation methods. Furthermore, it is considered that the proposed DQA method can be applied to other models such as 9-parameter affine and general 12-parameter affine transformations.

**Acknowledgements** I thank the Editor and three anonymous reviewers for their remarks and constructive comments which clearly helped to improve this article

## Declarations

**Conflict of interest** The authors declare no competing interests.

## References

- Amiri-Simkooei AR (2018) Parameter estimation in 3D affine and similarity: implementation of variance component estimation. *J Geod* 92(11):1285–1297. <https://doi.org/10.1007/s00190-018-1119-1>
- Andrei O. (2006) 3D affine coordinate transformations. M.Sc. Thesis in Geodesy, No. 3091 TRITA-GIT EX 06-004. Stockholm, Sweden: School of Architecture and the Built Environment, Royal Institute of Technology (KTH).
- Azeem MA, Mekkawi M, Gobashy M (2014) Subsurface structures using a new integrated geophysical analysis, South Aswan, Egypt. *Arab J Geosci* 7:5141–5157. <https://doi.org/10.1007/s12517-013-1140-x>

**Table 4** Residual matrix of control points

Residual matrix			
Station name	Vx [m]	Vy [m]	Vz[m]
Solitude	0.0940	0.1351	0.1402
Buoch Zeil	0.0588	-0.0497	0.0137
Hohenneuffen	-0.0399	-0.0879	-0.0081
Kuehlenberg	0.0202	-0.0220	-0.0874
Ex Mergelacc	-0.0919	0.0139	-0.0055
Ex Hof Asperg	-0.0118	0.0065	-0.0546
Ex Kaisersbach	-0.0294	0.0041	0.0017



- Bektaş S (2017) Dengeleme Hesabı (Adjustment Calculus). Ondokuz Mayıs University Press, ISBN:975-7636-54-1, Samsun
- Clifford WK (2007) Mathematical Papers. AMS Chelsea Publishing; American Mathematical Society, Providence
- Deakin RE (2006) A note on the Bursa-Wolf and Molodensky-Badekas transformations. In: Technical Report, School of Mathematical and Geospatial Sciences. RMIT University, Melbourne, pp 1–21
- Dong L, Tong X, Qingchun H, Tao Q (2021a) Empty region identification method and experimental verification for the two-dimensional complex structure. *Int J Rock Mech Min Sci* 147(3):104885. <https://doi.org/10.1016/j.ijrmms.2021.104885>
- Dong L, Tong X, Ma J (2021b) Quantitative investigation of tomographic effects in abnormal regions of complex structures. *Engineering* 7(7):1011–1022. <https://doi.org/10.1016/j.eng.2020.06.021>
- Even-Tzur G (2018) Invariance property of coordinate transformation. *J Spat Sci* 63(1):23–34. <https://doi.org/10.1080/14498596.2017.1316688>
- Even-Tzur G (2020) Coordinate transformation with variable number of parameters. *Surv Rev* 52(370):62–68. <https://doi.org/10.1080/00396265.2018.1517477>
- Grafarend E, Awange J (2003) Nonlinear analysis of the three-dimensional datum transformation [conformal group C7(3)]. *J Geod* 77(1–2):66–76. <https://doi.org/10.1007/s00190-002-0299-9>
- Hamilton WR (1853) Lectures on quaternions: containing a systematic statement of a new mathematical method. University Press, Berlin
- Ioannidou S, Pantazis G (2020) Helmert transformation problem. From Euler Angles Method to Quaternion Algebra. *ISPRS Int J Geo-Inf* 9:494
- Jitka P (2011) Application of dual quaternions algorithm for geodetic datum transformation. *Aplimat J Appl Math* 4(2):225–236
- Kenwright B (2012) A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies, The 20th International Conference on Computer Graphics, Visualization and Computer Vision (2012), pp 1–13
- Liu JF (2004) Three dimensional rotation represented by quaternion. *College Phys* 23(4):39–43 (in Chinese)
- Uygur SÖ, Aydın C, Akyılmaz 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>
- Velsink H (2015) Extendable linearised adjustment model for deformation analysis. *Surv Rev* 47(345):397–410. <https://doi.org/10.1179/1752270614Y.0000000140>
- 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(358):69–81. <https://doi.org/10.1080/00396265.2016.1234806>
- Watson G (2006) Computing Helmert transformations. *J Comput Appl Math* 197:387–394
- 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 H (2015) Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix. *Earth Planets Space* 67(1):105. <https://doi.org/10.1186/s40623-015-0263-6>
- Zeng H (2019) Iterative solution of Helmert transformation based on a unit dual quaternion. *Acta Geodaetica et Geophysica* 54(1):123–141. <https://doi.org/10.1007/s40328-018-0241-0>
- Zeng H, Yi Q (2011) Quaternion-based iterative solution of three-dimensional coordinate transformation problem. *J Comput* 6:1361–1368 [CrossRef]
- Zeng H, 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>