

FULL PAPER

Open Access



# WTLS iterative algorithm of 3D similarity coordinate transformation based on Gibbs vectors

Huaien Zeng<sup>1,2,3\*</sup> , Guobin Chang<sup>4</sup>, Haiqing He<sup>5</sup> and Kezhao Li<sup>6</sup>

## Abstract

The 3D similarity coordinate transformation is fundamental and frequently encountered in many areas of work such as geodesy, engineering surveying, LIDAR, terrestrial laser scanning, photogrammetry, machine vision, etc. The algorithms of 3D similarity transformation are divided into two categories. One is a closed-form algorithm that is straightforward and fast. However, it cannot provide the accuracy information for the transformation parameters. The other category of algorithm is iterative, and this can offer the accuracy information for the transformation parameters. However, the latter usually needs a good initial value of the unknown. Considering the accuracy information for transformation parameters is essential or indispensable from the viewpoint of uncertainty, this contribution proposes a weighted total least squares (WTLS) iterative algorithm of the 3D similarity coordinate transformation based on Gibbs vectors. It is fast in terms of fewer iterations, reliable and does not need good initial values of transformation parameters. Two cases including the registration of LIDAR points with big rotation angles and a geodetic datum transformation with small rotation angles are demonstrated to validate the new algorithm.

**Keywords:** 3D similarity coordinate transformation, Weighted total least squares (WTLS), Initial value of parameter, Accuracy assessment, Point cloud registration

## Introduction

The 3D similarity coordinate transformation aims to align the coordinate of points in different coordinate systems into a common coordinate system. It usually implements the transformation based on 3D seven-parameter similarity transformation utilizing a set of control points with the coordinates both in the source coordinate system and in the target coordinate system. This work is very popular in many fields, such as geodesy, engineering surveying, LIDAR, terrestrial laser scanning, photogrammetry, machine vision, etc. (Besl and McKay 1992; Crosilla and Beinat 2002; Horn 1987; Jaw and Chuang 2008; Kashani 2006; Krarup 1985; Marx 2017; Paffenholz and Bae 2012;

Walker et al. 1991; Wang et al. 2014; Závoti and Kalmár 2016; Zeng 2014; Zeng et al. 2018).

As far as the algorithm of 3D similarity coordinate transformation is concerned, much related literature has been published. The algorithms proposed in that literature can be divided into two categories. One is the closed-form or analytical algorithm. This category of algorithms utilizes the technologies such as singular value decomposition (SVD) also known as the Procrustes algorithm, e.g., Umeyama (1991), Crosilla and Beinat (2002), Grafarend and Awange (2003), eigenvalue–eigenvector decomposition (a quaternion-based algorithm), e.g., Horn (1987), Shen et al. (2006), orthonormal matrix, e.g., Horn et al. (1988), Zeng (2015). Additionally, Leick and Van Gelder (1975), Zeng and Yi (2010) presented, respectively, a step-wise algorithm based on the physical meaning of the similarity transformation. Wang et al. (2014) proposed a closed-form pairwise registration algorithm of point clouds utilizing the dual quaternion. The closed-form

\*Correspondence: zenghuaien\_2003@163.com

<sup>1</sup> Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University, Yichang 443002, China  
Full list of author information is available at the end of the article

algorithms are fast and efficient because the parameters are recovered directly by exact formulae without solution of nonlinear equations. However, this type of algorithm cannot deal with the general weight matrix of observations (i.e., the coordinates as pseudo-observations) as well as the accuracy estimation of transformation parameters. The other one is the iterative algorithm, which utilizes the numerical (iterative) computation technique to find the unknowns. Lots of iterative algorithms have been presented at present, e.g., Zeng and Tao (2003), Chen et al. (2004), El-Habiby et al. (2009), Zeng and Yi (2011), Zeng et al. (2016), Kurt (2018), Zeng et al. (2019). This type of algorithm usually requires a good initial value of unknown and iterative computation (e.g., Zeng and Tao 2003; Zeng and Yi 2011). But, in some situations, e.g., registration of LIDAR points due to the arbitrary size of rotation angles, it is difficult or even unlikely to obtain a good initial value of parameter. As a result, the algorithm needs many iterations or falls into a local minimum or diverges. Sometimes initial values are not required if global optimization algorithms are performed (e.g., Xu 2003); unfortunately it may lead to much more computation time and burden. The advantage of iterative algorithms over closed-form algorithms is the former can supply the accuracy information of unknowns, which is essential from the viewpoint of uncertainty.

In recent years, the errors-in-variables (EIV) model has evoked a lot of research interest. It considers the errors in all variables; for instance, in the 3D similarity transformation, the errors of target coordinates and source coordinates of control points are considered. Thus it is more reasonable than the traditional Gauss–Markov (GM) model which just considers the errors of target coordinates of control points. Golub and Van Loan (1980) presented the approach dealing with the EIV model and named it total least squares (TLS) technique. Afterward, a few types of TLS algorithms of 2D or 3D similarity coordinate transformation have been proposed. Teunissen (1988) derived a closed-form solution for 2D similarity transformation in the EIV model, i.e., symmetric 2D similarity transformation. Goryn and Hein (1995) presented a TLS solution for the 3D rigid transformation based on the Procrustes algorithm considering homogenous and uncorrelated errors. Schaffrin and Felus (2008) proposed TLS approaches to empirical coordinate transformation, which improved the accuracy in 2D affine transformations. Felus and Burtch (2009) proposed a closed-form weighted TLS (WTLS) solution to the 3D similarity transformation with pointwise weights and uncorrelated errors among points based on the Procrustes algorithm. Neitzel (2010) proposed a TLS solution to 2D similarity transformation within the nonlinear Gauss–Helmert model. Fang (2015) presented a

WTLS algorithm with constraints and universal formula for geodetic transformation. Chang (2015) presented a rigorous solution without presupposed fixing of the scale parameter (which some authors assume to be 1). Mahboub (2016) proposed a WTLS solution to 3D symmetrical similarity transformation without linearization; however, more iterations may be needed than the linearized model method. Mercan et al. (2018) proposed a weighted similarity transformation based on quaternions. The algorithm has seven unknowns including the translation parameters and scaled quaternion, and the iterations may lead to divergence.

As mentioned above, the closed-form solution to 3D similarity coordinate transformation is straightforward and fast, but cannot provide the accuracy of recovered transformation parameters. Iterative solutions to 3D similarity coordinate transformation can deal with general weight and provide the accuracy of recovered transformation parameters; however, it usually needs a good initial value of unknowns. Therefore, this contribution intends to present a new WTLS iterative solution to 3D similarity coordinate transformation based on Gibbs vectors, which is fast and does not need a relatively good initial value of the unknowns; in other words, is not sensitive to the initial value of unknowns.

In the next section, firstly 3D similarity transformation is introduced and the model of similarity transformation based on Gibbs vectors is established. Secondly in order to simplify the transformation model for improvement of computation performance, the translation parameters are derived in the WTLS sense and a WTLS iterative algorithm of 3D similarity transformation based on Gibbs vectors is proposed with detailed derivation. Two cases including the registration of LIDAR points and geodetic datum transformation are studied; the results show the new algorithms are fast and reliable. Finally, conclusions are drawn in the last section.

### Formulation and WTLS iterative algorithm of 3D similarity transformation

In this section, the basic EIV model of 3D similarity transformation is introduced, and in order to avoid transcendental function and functional constraints, the model of similarity transformation based on Gibbs vectors is established. Lastly, a WTLS iterative algorithm of 3D similarity transformation based on Gibbs vectors is put forward.

#### 3D similarity transformation in EIV model without functional constraints

Assume that a set of control points are given with their coordinates in both source and target coordinate systems.

Then the 3D similarity transformation in EIV model can be expressed as

$$\mathbf{p}_i^t - \mathbf{e}_i^t = \lambda \mathbf{R}(\mathbf{p}_i^o - \mathbf{e}_i^o) + \mathbf{t} \quad (1)$$

subject to

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

where  $\mathbf{p}_i^o = [x_i^o \ y_i^o \ z_i^o]^T$  and  $\mathbf{p}_i^t = [x_i^t \ y_i^t \ z_i^t]^T$  are the 3D coordinate vectors of control point  $i$  ( $i = 1, 2, \dots, n$ ) in the source coordinate system and the target coordinate system (denoted with superscript  $o$  and  $t$ ), respectively.  $\mathbf{e}_i^o$  and  $\mathbf{e}_i^t$  are the error vectors of  $\mathbf{p}_i^o$  and  $\mathbf{p}_i^t$ , respectively. And superscript  $T$  is transpose of matrix,  $\det$  represents determinant computation of matrix,  $\mathbf{I}_3$  is a  $3 \times 3$  identity matrix.  $\lambda$  denotes the scale factor,  $\mathbf{t} = [t_x \ t_y \ t_z]^T$  is the vector of three translation parameters and  $\mathbf{R}$  represents the rotation matrix of size  $3 \times 3$ , which is traditionally expressed by three rotation angles. Assume  $\mathbf{R}$  is produced by counterclockwise coordinate-frame rotations  $\theta_x, \theta_y, \theta_z$  ( $\theta_x$  applied first,  $\theta_z$  applied last). Then  $\mathbf{R}$  can be represented by rotation angles as

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

If  $\mathbf{R}$  is known, and the 3 main diagonal elements are positive,  $\theta_x, \theta_y, \theta_z$  can be calculated quickly by (3) as

$$\theta_x = -\tan^{-1} \frac{\mathbf{R}_{32}}{\mathbf{R}_{33}}, \theta_y = \sin^{-1}(\mathbf{R}_{31}), \theta_z = -\tan^{-1} \frac{\mathbf{R}_{21}}{\mathbf{R}_{11}}, \quad (4)$$

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

For  $n$  control points, the 3D similarity transformation in EIV model is constructed easily from Eq. (1).

$$\mathbf{p}^t - \mathbf{e}^t = \mathbf{B}\mathbf{p}^o - \mathbf{B}\mathbf{e}^o + \mathbf{C}\mathbf{t}, \quad (5)$$

where  $\mathbf{p}^o = [\mathbf{p}_1^{oT} \ \mathbf{p}_2^{oT} \ \dots \ \mathbf{p}_n^{oT}]^T$ ,  $\mathbf{p}^t = [\mathbf{p}_1^{tT} \ \mathbf{p}_2^{tT} \ \dots \ \mathbf{p}_n^{tT}]^T$ ,  $\mathbf{B} = \mathbf{I}_n \otimes \lambda \mathbf{R}$ ,  $\mathbf{C} = \mathbf{1}_n \otimes \mathbf{I}_3$ ,  $\mathbf{e}^o = [\mathbf{e}_1^{oT} \ \mathbf{e}_2^{oT} \ \dots \ \mathbf{e}_n^{oT}]^T$ ,  $\mathbf{e}^t = [\mathbf{e}_1^{tT} \ \mathbf{e}_2^{tT} \ \dots \ \mathbf{e}_n^{tT}]^T$ , and  $\mathbf{1}_n = \underbrace{[1 \ 1 \ \dots \ 1]}_n^T$ .  $\mathbf{I}_n$  is the identity matrix of size  $n \times n$ , the symbol  $\otimes$  means the Kronecker product.

The objective of 3D similarity transformation in EIV model is to recover the seven transformation parameters in the principle of total least squares, i.e.,

$$\mathbf{e}^{tT} \mathbf{W} \mathbf{e}^t + \mathbf{e}^{oT} \mathbf{W} \mathbf{e}^o = \min, \quad (6)$$

where  $\mathbf{W}$  of size  $3n \times 3n$  is a weight matrix of observations, adopting the point (i.e., row or column) weight in

Felus and Burtch (2009) as follows, since the point weight is more reasonable than the identity matrix weight usually employed in practice.

$$\mathbf{W} = \begin{bmatrix} w_1 \mathbf{I}_3 & & & \\ & w_2 \mathbf{I}_3 & & \\ & & \ddots & \\ & & & w_n \mathbf{I}_3 \end{bmatrix}, \quad (7)$$

where  $w_i$  is the weight of point  $i$  ( $i = 1, 2, \dots, n$ ). Note that the components of  $\mathbf{R}$  in Eq. (3) are trigonometric functions of rotation angles, which causes a high computation burden in the solution of parameters. In some particular situations, for instance in geodesy, the small rotation angles are small usually at the level of seconds; the elements of rotation matrix can be reduced to rotation angles (in radians) or constants (one or zero) and the 3D similarity transformation model is simplified to a linear one. Evidently this reduced treatment causes model error and then is not applicable to big rotation angle cases. In this contribution, the nonlinear 3D similarity transformation model is adopted to maintain validity for

any rotation angles.

Apart from using rotations, there are other representation of  $\mathbf{R}$ , such as unit quaternion (e.g., Horn 1987; Shen et al. 2006; Zeng and Yi 2011; Závoti and Kalmár 2016), dual quaternion (e.g., Walker et al. 1991; Wang et al. 2014; Zeng et al. 2018, 2019), direction cosine matrix (e.g., Chen et al. 2004; Wang et al. 2018), and the Rodrigues matrix or Gibbs vector (e.g., Zeng and Yi 2010; Závoti and Kalmár 2016; Zeng et al. 2016; Kurt 2018). The first three approaches to the representation of  $\mathbf{R}$  introduce functional constraints, for instance the norm of quaternion is unity as the unit quaternion representation is concerned. Additional functional constraints increase the complexity of formulation and computation. However, the 4th approach is free of transcendental function and functional constraints, and then improves the speed of computation. For this sake, this paper adopts the Rodrigues matrix or Gibbs vector to represent the rotation matrix  $\mathbf{R}$ . Due to the orthogonality of rotation matrix, it can be represented by a Rodrigues matrix as

$$\mathbf{R} = (\mathbf{I}_3 + \mathbf{S})(\mathbf{I}_3 - \mathbf{S})^{-1}, \quad (8)$$

where

$$\mathbf{S} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \quad (9)$$

is a skew-symmetric matrix, and  $\mathbf{v} = (a \ b \ c)^T$  is a Gibbs vector. If Eq. (8) is expanded, it is rewritten as

$$\mathbf{R} = \frac{1}{1+a^2+b^2+c^2} \begin{bmatrix} 1+a^2-b^2-c^2 & 2(ab-c) & 2(ac+b) \\ 2(ab+c) & 1-a^2+b^2-c^2 & 2(bc-a) \\ 2(ac-b) & 2(bc+a) & 1-a^2-b^2+c^2 \end{bmatrix}. \quad (10)$$

It is easy to see that the rotation matrix  $\mathbf{R}$  is expressed in terms of polynomials without trigonometric functions and functional constraints.

### Deduction of translation parameters and simplified 3D similarity transformation model

The similarity transformation parameters in preceding EIV model based on the Rodrigues matrix or Gibbs vector theoretically can be recovered by the TLS derivation; however, it is found in computation practice that the normal equation is singular because the numerical magnitude of translation parameter  $\mathbf{t}$  and those of the Gibbs vector  $\mathbf{v}$  as well as scale factor  $\lambda$  probably differ greatly from each other, which prevents a solution to the equation. For this reason, we seek to eliminate the translation parameters  $\mathbf{t}$  from the original model.

Lagrangian extremum principle is a reliable way to solve the 3D similarity transformation problem in the sense of TLS. In order to obtain the explicit solution of  $\mathbf{t}$  regarding  $\lambda$  and  $\mathbf{R}$  as known parameters, the Lagrangian extremum problem which just considers  $\mathbf{t}$  as unknown is established as

$$\min_{\mathbf{e}^o, \mathbf{e}^t, \mathbf{t}, \mathbf{k}} \left\{ L(\mathbf{e}^o, \mathbf{e}^t, \mathbf{t}, \mathbf{k}) = \mathbf{e}^{oT} \mathbf{W} \mathbf{e}^o + \mathbf{e}^{tT} \mathbf{W} \mathbf{e}^t + 2\mathbf{k}^T (\mathbf{p}^t - \mathbf{e}^t - \mathbf{B} \mathbf{p}^o + \mathbf{B} \mathbf{e}^o - \mathbf{C} \mathbf{t}) \right\}, \quad (11)$$

where  $\mathbf{k}$  is a vector of the Lagrangian multiplier. In order to obtain the solution, partial derivations with respect to all variables should be set to zero. From the partial derivations with respect to  $\mathbf{e}^o$ , one gets

$$\mathbf{e}^o = -\mathbf{W}^{-1} \mathbf{B}^T \mathbf{k}. \quad (12)$$

From the partial derivations with respect to  $\mathbf{e}^t$ , one gets

$$\mathbf{e}^t = \mathbf{W}^{-1} \mathbf{k}. \quad (13)$$

Inserting Eqs. (12) and (13) into the partial derivations with respect to  $\mathbf{k}$  which is identical to Eq. (5) and arranging the terms, one obtains

$$\mathbf{k} = (\mathbf{W}^{-1} + \mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T)^{-1} (\mathbf{p}^t - \mathbf{B} \mathbf{p}^o - \mathbf{C} \mathbf{t}). \quad (14)$$

Denoting

$$\mathbf{W}_1 = (\mathbf{W}^{-1} + \mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T)^{-1} \quad (15)$$

substituting  $\mathbf{B} = \mathbf{I}_n \otimes \lambda \mathbf{R}$  into Eq. (15), the following formula can be easily derived:

$$\mathbf{W}_1 = (1 + \lambda^2)^{-1} \mathbf{W}. \quad (16)$$

Combining  $\mathbf{C}^T \mathbf{k} = \mathbf{C}^T \mathbf{W}_1 (\mathbf{p}^t - \mathbf{B} \mathbf{p}^o - \mathbf{C} \mathbf{t})$  from Eq. (14) with  $\mathbf{k}^T \mathbf{C} = 0$  from the partial derivative of  $\mathbf{t}$ , one obtains the solution of  $\mathbf{t}$  as

$$\mathbf{t} = (\mathbf{C}^T \mathbf{W}_1 \mathbf{C})^{-1} \mathbf{C}^T \mathbf{W}_1 (\mathbf{p}^t - \mathbf{B} \mathbf{p}^o). \quad (17)$$

Inserting Eq. (16) into Eq. (17), one obtains the following formula and its proof is given in Additional file 1: Appendix S1:

$$\mathbf{t} = \overline{\mathbf{p}^t} - \lambda \mathbf{R} \overline{\mathbf{p}^o}, \quad (18)$$

where

$$\overline{\mathbf{p}^t} = \frac{\sum_{i=1}^n w_i \mathbf{p}_i^t}{\sum_{i=1}^n w_i} \quad (19)$$

and

$$\overline{\mathbf{p}^o} = \frac{\sum_{i=1}^n w_i \mathbf{p}_i^o}{\sum_{i=1}^n w_i} \quad (20)$$

are the weighted coordinates of the barycenter in the target coordinate system and the source coordinate system, respectively.

Inserting Eq. (17) into Eq. (5) and carrying out arrangements, one gets

$$\begin{aligned} & \left( \mathbf{I}_{3n} - \mathbf{C} (\mathbf{C}^T \mathbf{W}_1 \mathbf{C})^{-1} \mathbf{C}^T \mathbf{W}_1 \right) \mathbf{p}^t - \mathbf{e}^t \\ &= \left( \mathbf{I}_{3n} - \mathbf{C} (\mathbf{C}^T \mathbf{W}_1 \mathbf{C})^{-1} \mathbf{C}^T \mathbf{W}_1 \right) \mathbf{B} \mathbf{p}^o - \mathbf{B} \mathbf{e}^o. \end{aligned} \quad (21)$$

Denoting

$$\mathbf{D} = \mathbf{I}_{3n} - \mathbf{C}(\mathbf{C}^T \mathbf{W}_1 \mathbf{C})^{-1} \mathbf{C}^T \mathbf{W}_1, \quad (22)$$

thus Eq. (21) is rewritten as

$$\mathbf{D}\mathbf{p}^t - \mathbf{e}^t = \mathbf{D}\mathbf{B}\mathbf{p}^o - \mathbf{B}\mathbf{e}^o. \quad (23)$$

It is easily proved that

$$\mathbf{D}\mathbf{B} = \mathbf{B}\mathbf{D}, \quad (24)$$

and the proof is given in Additional file 2: Appendix S2.

Then Eq. (23) is re-expressed as

$$\Delta\mathbf{p}^t - \mathbf{e}^t = \mathbf{B}\Delta\mathbf{p}^o - \mathbf{B}\mathbf{e}^o, \quad (25)$$

where

$$\Delta\mathbf{p}^t = \mathbf{D}\mathbf{p}^t \quad (26)$$

and

$$\Delta\mathbf{p}^o = \mathbf{D}\mathbf{p}^o \quad (27)$$

are centralized coordinate vectors, i.e., the difference of original coordinates and the weighted coordinates of the barycenter, in the target coordinate system and source coordinate system, respectively.  $\mathbf{D}$  is a centering matrix.

$$\frac{\partial \mathbf{B}}{\partial b} = \mathbf{I}_n \otimes \lambda \frac{\partial \mathbf{R}}{\partial b}, \quad (31)$$

$$\frac{\partial \mathbf{B}}{\partial c} = \mathbf{I}_n \otimes \lambda \frac{\partial \mathbf{R}}{\partial c}, \quad (32)$$

$$\begin{aligned} \frac{\partial \mathbf{R}}{\partial v_x} &= \frac{\partial \mathbf{S}}{\partial v_x} (\mathbf{I}_3 - \mathbf{S})^{-1} + (\mathbf{I}_3 + \mathbf{S})(\mathbf{I}_3 - \mathbf{S})^{-1} \frac{\partial \mathbf{S}}{\partial v_x} (\mathbf{I}_3 - \mathbf{S})^{-1} \\ &= \frac{\partial \mathbf{S}}{\partial v_x} (\mathbf{I}_3 - \mathbf{S})^{-1} + \mathbf{R} \frac{\partial \mathbf{S}}{\partial v_x} (\mathbf{I}_3 - \mathbf{S})^{-1} \\ &= (\mathbf{I}_3 + \mathbf{R}) \frac{\partial \mathbf{S}}{\partial v_x} (\mathbf{I}_3 - \mathbf{S})^{-1}, \quad v_x = a, b, c, \end{aligned} \quad (33)$$

$$\frac{\partial \mathbf{S}}{\partial a} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad (34)$$

$$\frac{\partial \mathbf{S}}{\partial b} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad (35)$$

$$\frac{\partial \mathbf{S}}{\partial c} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (36)$$

Denoting

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \mathbf{B}}{\partial \lambda} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) & \frac{\partial \mathbf{B}}{\partial a} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) & \frac{\partial \mathbf{B}}{\partial b} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) & \frac{\partial \mathbf{B}}{\partial c} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) \end{bmatrix} \quad (37)$$

It is obvious to see that the translation parameter  $\mathbf{t}$  is eliminated.

#### WTLS iterative algorithm of 3D similarity transformation in errors-in-variables model

Linearizing Eq. (25) by Taylor's formula, one obtains

$$\begin{aligned} \Delta\mathbf{p}^t - \mathbf{e}^t &= \mathbf{B}^j \Delta\mathbf{p}^o - \mathbf{B}^j \mathbf{e}^{oj} + \frac{\partial \mathbf{B}}{\partial \lambda} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) d\lambda \\ &\quad + \frac{\partial \mathbf{B}}{\partial a} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) da + \frac{\partial \mathbf{B}}{\partial b} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) db \\ &\quad + \frac{\partial \mathbf{B}}{\partial c} (\Delta\mathbf{p}^o - \mathbf{e}^{oj}) dc - \mathbf{B}^j d\mathbf{e}^{oj}, \end{aligned} \quad (28)$$

where superscript  $j$  denotes the approximation from iteration  $j$  of the corresponding variable,  $d\mathbf{x} = [d\lambda \ da \ db \ dc]^T$  is the correction vector of unknowns  $\mathbf{x} = [\lambda \ a \ b \ c]^T$ , and

$$\frac{\partial \mathbf{B}}{\partial \lambda} = \mathbf{I}_n \otimes \mathbf{R}, \quad (29)$$

$$\frac{\partial \mathbf{B}}{\partial a} = \mathbf{I}_n \otimes \lambda \frac{\partial \mathbf{R}}{\partial a}, \quad (30)$$

and considering

$$\mathbf{e}^o = \mathbf{e}^{oj} + d\mathbf{e}^{oj}, \quad (38)$$

Eq. (28) is rewritten as

$$\Delta\mathbf{p}^t - \mathbf{e}^t = \mathbf{B}^j \Delta\mathbf{p}^o - \mathbf{B}^j \mathbf{e}^o + \mathbf{A}^j d\mathbf{x}. \quad (39)$$

The WTLS solution process of  $d\mathbf{x}$  in Eq. (39) is similar to that of  $\mathbf{t}$  in Eq. (5) by the Lagrangian extremum principle shown in the preceding subsection. Replacing  $\mathbf{p}^t, \mathbf{p}^o, \mathbf{C}, \mathbf{t}$  with  $\Delta\mathbf{p}^t, \Delta\mathbf{p}^o, \mathbf{A}^j, d\mathbf{x}$  in the solution process of  $\mathbf{t}$ , respectively, one can easily get the solution as

$$d\mathbf{x} = (\mathbf{A}^{jT} \mathbf{W}_1 \mathbf{A}^j)^{-1} \mathbf{A}^{jT} \mathbf{W}_1 (\Delta\mathbf{p}^t - \mathbf{B}^j \Delta\mathbf{p}^o), \quad (40)$$

$$\mathbf{e}^o = -\mathbf{W}^{-1} \mathbf{B}^T \mathbf{k}', \quad (41)$$

$$\mathbf{e}^t = \mathbf{W}^{-1} \mathbf{k}', \quad (42)$$

where

$$\mathbf{k}' = \mathbf{W}_1 (\Delta\mathbf{p}^t - \mathbf{B}^j \Delta\mathbf{p}^o - \mathbf{A}^j d\mathbf{x}). \quad (43)$$



Next, we can utilize an iterative algorithm to compute the final solution of  $\mathbf{x}$ . Set the initial value  $\mathbf{x}^0 = [1 \ 0 \ 0 \ 0]^T$ , the initial value of the unknown  $\mathbf{e}^0$  as  $[0 \ 0 \ \cdots \ 0]^T$  and compute the  $\mathbf{dx}$ , if the absolute values of all elements of  $\mathbf{dx}$  are less than a small tolerance value  $\tau$  (e.g.,  $1.0 \times 10^{-10}$ ), stop the iteration, otherwise update the  $\mathbf{x}$  with previous iterative value of  $\mathbf{x}$  plus  $\mathbf{dx}$  and repeat the iterative computation until the stop criterion is satisfied. After iteration, the variance factor of unit weight is estimated as

$$\sigma = \pm \sqrt{\frac{\mathbf{e}^{0T} \mathbf{W} \mathbf{e}^0 + \mathbf{e}^{tT} \mathbf{W} \mathbf{e}^t}{3n - 7}}, \quad (44)$$

or if  $\mathbf{A}^j \mathbf{dx}$  is a sufficiently small term, the approximate formula as follows is adopted:

$$\sigma \approx \pm \sqrt{\frac{(\Delta \mathbf{p}^t - \mathbf{B}^j \Delta \mathbf{p}^o)^T \mathbf{W}_1 (\Delta \mathbf{p}^t - \mathbf{B}^j \Delta \mathbf{p}^o)}{3n - 7}}. \quad (45)$$

By Eq. (40) the estimated covariance matrix of transformation parameters  $\mathbf{x}$  is derived as

$$\mathbf{D}_x = \sigma^2 (\mathbf{A}^{jT} \mathbf{W}_1 \mathbf{A}^j)^{-1}. \quad (46)$$

Similarly by Eq. (17) the estimated covariance matrix of transformation parameters  $\mathbf{t}$  is derived as

$$\mathbf{D}_t = \sigma^2 (\mathbf{C}^T \mathbf{W}_1 \mathbf{C})^{-1}. \quad (47)$$

To sum up, the above presented WTLS iterative algorithm is listed in Table 1.

### Numerical analysis and discussion

Two cases are studied. One is the registration of LIDAR points involving big rotation angles and the weight matrix is not considered, i.e., an identity matrix is

adopted. The other one is 3D coordinate transformation in geodesy, which encounters small rotation angles and point-weighting is adopted.

#### Case 1 (registration of LIDAR points)

This data are adopted from Wang et al. (2014). The point features extracted from two neighboring LIDAR point clouds are listed in Table 2. In order to verify the new algorithm, the 18 points are divided into control points and check points. The first 10 points in Table 2 are arbitrarily chosen as control points and the last 8 points are check points. The distribution of all points is depicted in Fig. 1. In Fig. 1 plus signs denote control points and cross signs denote check points. The weight matrix adopted is an identity matrix in this case; in other words, the weights of all points are the same, and the correlation among the three coordinates of each point is not considered.

The transformation parameters in the WTLS sense are computed by the new algorithm and Algorithm 4 presented in Felus and Burtch (2009), henceforth described as the Felus–Burtch Algorithm. All results are listed in Table 3. It is seen from Table 3, that the new algorithm needs only 6 iterations to converge. And for two algorithms, the results are exactly the same. Thus the new algorithm is fast and reliable. Additionally, the Felus–Burtch Algorithm does not provide the accuracies of recovered transformation parameters; however, the new algorithm does. It gives the accuracy information which is shown in Table 3. For more details, refer to the covariance matrix of recovered transformation parameters offered by the new algorithm, which is listed in Table 4. The predicted errors of coordinates of control points by the new algorithm are listed in Table 5. It is worth mentioning that the formula to compute  $\tilde{\mathbf{E}}_Y$  in Algorithm 4 presented from Felus and Burtch (2009) is wrong. The first term namely the inverse of weight matrix should be removed from the formula. If the LS transformation parameters and predicted error of coordinates of control

**Table 1** WTLS iterative algorithm of 3D similarity transformation in EIV model

Initiation:

Input 3D coordinates vector  $\mathbf{p}^o, \mathbf{p}^t$  of control points and pointwise weight matrix  $\mathbf{W}$ , set the initial value of the unknown  $\mathbf{x}$  as  $[1 \ 0 \ 0 \ 0]^T$  and the initial value of the unknown  $\mathbf{e}^0$  as  $[0 \ 0 \ \cdots \ 0]^T$

Iterative computation:

Step 1. Calculate the centering matrix  $\mathbf{D}$  by Eq. (22), and then calculate the centralized coordinate vectors  $\Delta \mathbf{p}^o, \Delta \mathbf{p}^t$  by Eqs. (27) and (26)

Step 2. Compute  $\mathbf{R}$  by Eq. (8) or Eq. (10),  $\mathbf{W}_1$  by Eq. (16), and  $\mathbf{A}^j$  by Eqs. (29) to (37)

Step 3. Compute  $\mathbf{dx}$  by Eq. (40). If absolute values of all the elements of  $\mathbf{dx}$  are less than a threshold  $\tau$  ( $1.0 \times 10^{-10}$  is set in the paper), turn to Step 4, otherwise turn to Step 2 with the undated value  $\mathbf{x} = \mathbf{x} + \mathbf{dx}$  and  $\mathbf{e}^0$  by Eq. (41) and Eq. (43)

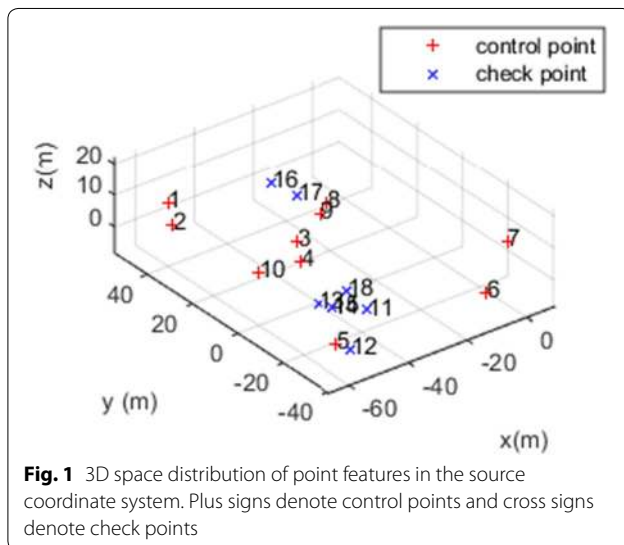
Step 4. Compute  $\mathbf{R}$  by Eq. (8) or Eq. (10). If rotation angles  $\theta_x, \theta_y, \theta_z$  are needed, compute them by Eq. (4). Compute  $\mathbf{t}$  by Eqs. (18) to (20)

Step 5. Estimate the variance factor of unit weight  $\sigma$  by Eq. (44) or Eq. (45), estimate the covariance matrix of transformation parameters  $\mathbf{x}$  and  $\mathbf{t}$  by Eqs. (46) and (47), respectively

Output the scale factor  $\lambda$  (the first element of  $\mathbf{x}$ ), the Gibbs vector  $\mathbf{v} = (a \ b \ c)^T$  (the latter three elements of  $\mathbf{x}$ ), the rotation matrix  $\mathbf{R}$  and the translation parameter  $\mathbf{t}$ . If required, output rotation angles  $\theta_x, \theta_y, \theta_z$ . Output the variance factor of unit weight  $\sigma$  and the covariance matrices of transformation parameters  $\mathbf{x}$  and  $\mathbf{t}$

**Table 2** Point features extracted from two neighboring LIDAR point clouds

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



points are computed with the algorithm in Wang et al. (2014), one finds that LS variance factor of unit weight is 0.0234 m, which is bigger than that of WTLS. Therefore, the WTLS solution is better than the LS solution from the viewpoint of variance factor of unit weight.

Next, the new algorithm in this contribution and the Felus–Burtch Algorithm are used to calculate the target coordinate of check points once the transformation parameters are recovered by the 3D similarity

transformation model, i.e., Eq. (1). The errors of computed coordinates of check points are obtained by subtracting the computed coordinates of check points from the known coordinates of check points. The errors are listed in Table 6.

Lastly, in order to test the dependence of the new algorithm on the initial values of parameters, six sets of initial values of parameters are adopted which are listed in Table 7. From Table 7, it is seen that the maximum biased angle from the correct solution is 2.5°, 18.9°, 29.4°, 44.5°, 59.4°, 74.9° for Set 1, Set 2, Set 3, Set 4, Set 5, Set 6, respectively. Set 3 gives the default initial values of parameters. The corresponding initial values of the Gibbs vector elements are computed from the initial values of rotation angles in Table 7. Then the new algorithm is performed with six sets of initial values of parameters, respectively. Results show that the algorithm successively finds the correct solutions with six sets of initial values of parameters. Number of iterations is 5, 5, 6, 6, 8, and 8 for Set 1, Set 2, Set 3, Set 4, Set 5, Set 6, respectively. It is evidently seen that with the increase of maximum biased angle from Set 1 to Set 6, the number of iterations increases slowly from 5 to 8. Thus the new algorithm is not sensitive to the initial values of parameters. Further, the iterative process is drawn in Fig. 2. From this figure, it is seen that the convergence rate is fast since that the logarithm (base 10) of objective function, i.e.,  $\log_{10}(\mathbf{e}^{oT}\mathbf{W}\mathbf{e}^o + \mathbf{e}^{tT}\mathbf{W}\mathbf{e}^t)$

**Table 3 Recovered parameters for Case 1 using the new algorithm and the Felus–Burtch algorithm, respectively**

Parameters	The new algorithm	The Felus–Burtch algorithm
Iterations	6	–
Scale		
$\lambda$	$1.0002101164 \pm 0.0002001329$	1.0002101164
Gibbs vector		
$a$	$-0.0381487705 \pm 0.0001517110$	–
$b$	$0.1072667832 \pm 0.0001625734$	–
$c$	$0.2637168674 \pm 0.0001124502$	–
Rotation angles		
$\theta_x$ (°)	1.0693156620	1.0693156620
$\theta_y$ (°)	– 12.5193487938	– 12.5193487938
$\theta_z$ (°)	– 29.4297272328	– 29.4297272328
Translation		
$t_x$ (m)	$-22.9747 \pm 0.0074$	– 22.9747
$t_y$ (m)	$29.4056 \pm 0.0074$	29.4056
$t_z$ (m)	$-2.2626 \pm 0.0074$	– 2.2626
$\sigma$ (m)	0.0165797705	0.0165797705

**Table 4 Covariance matrixes of recovered transformation parameters obtained by the new algorithm**

$D_x = 10^{-7} \times$	0.4005319716	0.0000000000	0.0000000000	0.0000000000
	0.0000000000	0.2301623730	– 0.1041878824	– 0.0074983064
	0.0000000000	– 0.1041878824	0.2643009705	– 0.0034785756
	0.0000000000	– 0.0074983064	– 0.0034785756	0.1264504316
$D_t = 10^{-4} \text{ m}^2 \times$	0.5498931099	0.0000000000	0.0000000000	
	0.0000000000	0.5498931099	0.0000000000	
	0.0000000000	0.0000000000	0.5498931099	

**Table 5 Predicted errors at 10 control points in the source system and target system by the new algorithm**

Point no.	Error of reference station (target system) (m)			Error of unregistered station (source system) (m)		
	$x^t$	$y^t$	$z^t$	$x^o$	$y^o$	$z^o$
1	0.0093	0.0054	– 0.0027	– 0.0111	– 0.0001	0.0003
2	0.0096	0.0015	– 0.0026	– 0.0095	0.0034	0.0006
3	0.0057	0.0058	– 0.0057	– 0.0089	– 0.0024	0.0039
4	0.0052	0.0034	– 0.0021	– 0.0065	– 0.0004	0.0007
5	0.0095	0.0073	0.0028	– 0.0110	– 0.0016	– 0.0053
6	0.0015	0.0069	– 0.0045	– 0.0056	– 0.0053	0.0033
7	– 0.0045	0.0075	– 0.0064	– 0.0011	– 0.0089	0.0061
8	– 0.0013	– 0.0014	– 0.0015	0.0015	0.0006	0.0019
9	– 0.0341	– 0.0198	– 0.0020	0.0381	0.0003	0.0105
10	– 0.0009	– 0.0166	0.0247	0.0141	0.0145	– 0.0220

linearly decreases for all six sets of initial values. So the new algorithm is very fast in terms of fewer iterations. It is worth noting that for this big rotation angle case, no good initial values of parameters are required while usually good initial values of parameters are needed, as shown in Zeng and Yi (2011).

#### Case 2 (geodetic datum transformation)

The data are from Grafarend and Awange (2003). The 3D coordinates of 7 points in the source coordinate system and the target coordinate system, respectively, are given in Table 8. The 3D space distribution of 7 points in the source coordinate system is shown in Fig. 3. In

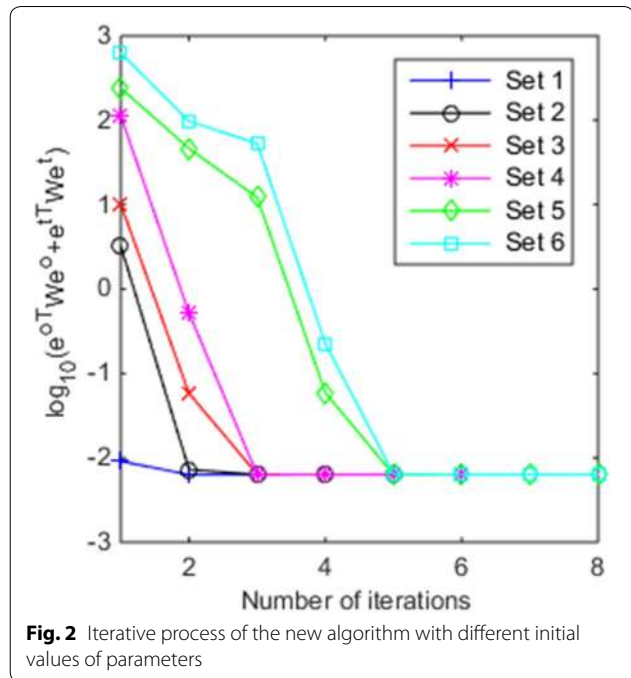


**Table 6** Errors of computed target coordinates of check points by the two algorithms

Point no.	Errors of computed target coordinates (m)		
	$x^t$	$y^t$	$z^t$
11	0.0071	−0.0060	0.0379
12	0.0433	0.0259	0.0167
13	−0.0055	−0.0549	0.0118
14	0.0345	0.0687	−0.0609
15	0.0816	0.0456	−0.0182
16	−0.0139	−0.0062	−0.0012
17	−0.0093	−0.0592	0.0198
18	−0.0496	0.0221	−0.0098

order to validate the new algorithm, the 7 points are divided into control points and check points. Four points are selected as control points from Fig. 3 based on the consideration that usually, in order to obtain good transformation parameters, control points should be distributed evenly and enclose as much of the surveying area as possible. The blue asterisk denotes the control points, and the red circle denotes the check points in Fig. 3. The selected four control points are shown with bold font in Table 8. The weight matrix adopted is a pointwise one, i.e., for each point, it has isotropic weight and its different coordinate components are not correlated, and for all points, they are independent of each other. The pointwise weights are listed in Table 9.

The recovered transformation parameters in the WTLS sense by the new algorithm and the Felus–Burtch

**Fig. 2** Iterative process of the new algorithm with different initial values of parameters

Algorithm are listed in Table 10. It is seen from Table 10 that the new algorithm needs only 2 iterations to complete computation. The results are identical for the two algorithms if the decimal rounding is ignored. Thus the new algorithm is fast and reliable for this case. As mentioned in the preceding case, the new algorithm can provide the accuracies of recovered transformation parameters while the Felus–Burtch Algorithm cannot. The accuracy information is shown in Table 10. For

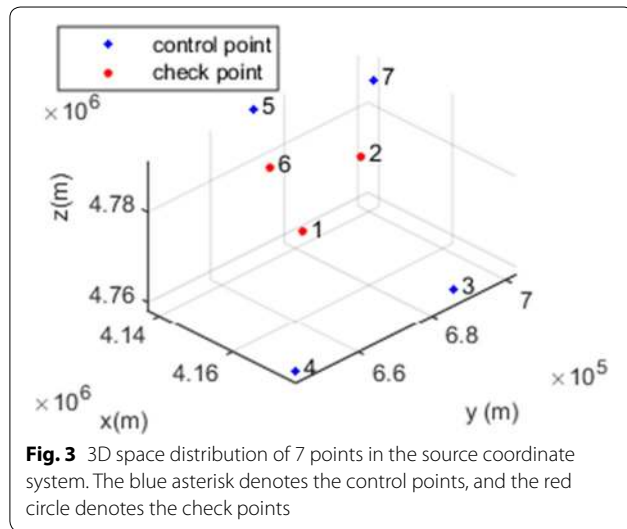
**Table 7** Six sets of initial values of transformation parameters

Parameters	Set 1	Set 2	Set 3 (default)	Set 4	Set 5	Set 6
Scale						
$\lambda$	1	1	1	1	1	1
Gibbs vector (from initial rotation angles)						
$a$	−0.0210	−0.1981	0	0.0688	−0.2513	−0.7442
$b$	0.0874	0.0453	0	−0.2867	−0.2235	0.2915
$c$	0.2400	0.2565	0	0.2401	−0.3192	−0.1960
Rotation angles						
$\theta_x$ (°)	0	20	0	0	20	76
$\theta_y$ (°)	−10	−10	0	32	30	−10
$\theta_z$ (°)	−27	−27	0	−27	30	30
Biased angles (from the correct solution)						
For $\theta_x$ (°)	−1.1	18.9	−1.1	−1.1	18.9	74.9
For $\theta_y$ (°)	2.5	2.5	12.5	44.5	42.5	2.5
For $\theta_z$ (°)	2.4	2.4	29.4	2.4	59.4	59.4
Number of iterations	5	5	6	6	8	8

The values in italics denote the maximum biased angle from the correct solution for each set of initial values

**Table 8** Coordinates of 7 points in the source system and target system

Point no.	Source system (local system) (m)			Target system (WGS-84 system) (m)		
	$x^o$	$y^o$	$z^o$	$x^t$	$y^t$	$z^t$
1 Solitude	4157222.543	664789.307	4774952.099	4157870.237	664818.678	4775416.524
2 Buoch Zeil	4149043.336	688836.443	4778632.188	4149691.049	688865.785	4779096.588
3 Hohenneuffen	4172803.511	690340.078	4758129.701	4173451.354	690369.375	4758594.075
4 Kuehlenberg	4177148.376	642997.635	4760764.800	4177796.064	643026.700	4761228.899
5 Ex Mergelaec	4137012.190	671808.029	4791128.215	4137659.549	671837.337	4791592.531
6 Ex Hof Asperg	4146292.729	666952.887	4783859.856	4146940.228	666982.151	4784324.099
7 Ex Kaisersbach	4138759.902	702670.738	4785552.196	4139407.506	702700.227	4786016.645



more details, refer to the covariance matrix of recovered transformation parameters offered by the new algorithm, which are listed in Table 11. The predicted errors of coordinates of control points by the new algorithm are listed in Table 12. Table 12 shows as far as absolute values are concerned, the predicted errors of coordinates of the new algorithm control points in the source coordinate system are approximately identical to those in the target coordinate system. It is because the point weights are the same for the two systems, the scale factor is very close to 1 and the rotation angles are very small resulting in almost an identity rotation matrix.

Next, the new algorithm and the Felus–Burtch Algorithm are employed to calculate the target coordinates of check points. Then errors of computed coordinates

**Table 10** Recovered parameters for Case 2 using the new algorithm and the Felus–Burtch Algorithm, respectively

Parameters	The new algorithm	The Felus–Burtch algorithm
Iterations	2	–
Scale		
$\lambda$	$1.0000062604 \pm 0.8265 \times 10^{-6}$	1.0000062604
Gibbs vector		
$a$	$2.6896 \times 10^{-6} \pm 0.5939 \times 10^{-6}$	–
$b$	$-2.2310 \times 10^{-6} \pm 0.6482 \times 10^{-6}$	–
$c$	$-2.6177 \times 10^{-6} \pm 0.5187 \times 10^{-6}$	–
Rotation angles		
$\theta_x (")$	–1.109526838	–1.109526839
$\theta_y (")$	0.920338884	0.920338882
$\theta_z (")$	1.079870444	1.079870445
Translation		
$t_x (m)$	$639.3602 \pm 0.0270$	639.3602
$t_y (m)$	$72.4921 \pm 0.0270$	72.4921
$t_z (m)$	$412.2363 \pm 0.0270$	412.2363
$\sigma (m)$	0.0579705587	0.0579705589

of the check points are computed. The errors are the same for the two algorithms and are listed in Table 13.

## Conclusions

The paper presents a WTLS iterative algorithm of the 3D similarity coordinate transformation based on the Gibbs vector considering that transformation model is compact and without redundant constraints of parameter, such as the norm of the unit quaternion is 1. The

**Table 9** Pointwise weighting

Point no. $i$	1	2	3	4	5	6	7
$w_i$	2.170137	2.097755	2.208968	2.201671	2.182928	2.268808	2.643404

**Table 11 Covariance matrix of transformation parameters obtained by the new algorithm**

$D_x = 10^{-12} \times$	0.6830762558	0.0000000000	0.0000000000	0.0000000000
	0.0000000000	0.3527666780	− 0.1693925312	− 0.1326418580
	0.0000000000	− 0.1693925312	0.4202274973	0.1112063825
	0.0000000000	− 0.1326418580	0.1112063825	0.2690705785
$D_t = 10^{-3} \text{ m}^2 \times$	0.7276425140	0.0000000000	0.0000000000	
	0.0000000000	0.7276425140	0.0000000000	
	0.0000000000	0.0000000000	0.7276425140	

**Table 12 Predicted error coordinates of control points in the source system and target system by the new algorithm**

Point no.	Error of source coordinate (local system) (m)			Error of target coordinate (WGS-84 system) (m)		
	$x^o$	$y^o$	$z^o$	$x^t$	$y^t$	$z^t$
3	0.0119	0.0379	− 0.0089	− 0.0119	− 0.0379	0.0089
4	− 0.0268	− 0.0127	0.0192	0.0268	0.0127	− 0.0192
5	0.0198	− 0.0206	− 0.0063	− 0.0198	0.0206	0.0063
7	− 0.0040	− 0.0041	− 0.0034	0.0040	0.0041	0.0034

**Table 13 Errors of computed target coordinates of check points by the two algorithms**

Point no.	Error of computed target coordinate (m)		
	$x^t$	$y^t$	$z^t$
1	− 0.1335	− 0.1670	− 0.1705
2	− 0.0942	0.0356	− 0.0296
6	− 0.0353	− 0.0371	0.0302

cases studies show that whether the rotation angles are big or small, the new algorithm is fast in terms of fewer iterations, reliable and does not require good initial value of unknowns or not sensitive to the initial value. The new algorithm can offer accuracy of the unknowns, i.e., transformation parameters while the analytical TIS algorithm, e.g., the Felus–Burtch Algorithm cannot. This is meaningful and indispensable from the viewpoint of the accuracy assessment. Additionally, it is shown that the TLS solution is advantageous over the LS solution from the viewpoint of variance factor of unit weight. In the near future, we intend to present new algorithms that deal with more general weighting of observations.

### Supplementary information

Supplementary information accompanies this paper at <https://doi.org/10.1186/s40623-020-01179-1>.

**Additional file 1: Appendix S1.** Proof of Eq. (18).

**Additional file 2: Appendix S2.** Proof of Eq. (24).

### Acknowledgements

The first author thanks his research team for valuable suggestions and providing a good atmosphere. All the authors thank editor Severine Rosat and two anonymous reviewers for valuable comments and suggestions, which enhanced the quality of this manuscript.

### Authors' contributions

HZ carried out the theoretical studies including establishing the 3D similarity coordinate transformation based on the Gibbs vector, deriving the WTLS solution of transformation parameters, additionally drafted the manuscript. GC participated in deriving the solution of transformation parameters. HH designed the case study and helped to draft the manuscript. KL carried out the case computation. All the authors read and approved the final manuscript.

### Funding

The study is supported jointly by the Open Foundation of Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University (Grant No. 2019KSD09), 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), the Open Foundation of Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University (Grant No. 2015KLA06), the Open Foundation of the Key Laboratory of Precise Engineering and Industry Surveying, National Administration of Surveying, Mapping and Geoinformation of China (Grant No. PF2015-14), Hubei Provincial Natural Science Foundation of China (Grant No. 2016CFB443), and National Natural Science Foundation of China (Grant No. 41104009).

### Availability of data and material

The case data were collected from Wang et al. (2014) and Grafarend and Awange (2003).

### Competing interests

The authors declare that they have no competing interests.

**Author details**

<sup>1</sup> Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University, Yichang 443002, China. <sup>2</sup> National Field Observation and Research Station of Landslides in the Three Gorges Reservoir Area of Yangtze River, China Three Gorges University, Yichang 443002, China. <sup>3</sup> Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, China. <sup>4</sup> School of Environmental Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China. <sup>5</sup> School of Geomatics, East China University of Technology, Nanchang 330013, China. <sup>6</sup> School of Surveying and Landing Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China.

Received: 10 January 2020 Accepted: 16 April 2020

Published online: 29 April 2020

**References**

- 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 (2015) On least-squares solution to 3D similarity transformation problem under Gauss-Helmert model. *J Geod* 89:573–576
- Chen Y, Shen YZ, Liu DJ (2004) A simplified model of three dimensional-datum transformations adapted to big rotation angle. *Geomatics Inform Sci Wuhan Univ* 29(12):1101–1105
- 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
- El-Habiby MM, Gao Y, Sideris MG (2009) Comparison and analysis of non-linear least squares methods for 3-D coordinates transformation. *Surv Rev* 41(311):26–43
- 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(1):65–74
- Golub GH, Van Loan CF (1980) An analysis of the total least squares problem. *SIAM J Numer Anal* 17(6):883–893
- Goryn D, Hein S (1995) On the estimation of rigid body rotation from noisy data. *IEEE Trans Pattern Anal Mach Intell* 17(12):1219–1220
- 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
- Jaw JJ, Chuang TY (2008) Registration of ground-based LIDAR point clouds by means of 3D line features. *J Chin Inst Eng* 31(6):1031–1045
- Kashani I (2006) Application of generalized approach to datum transformation between local classical and satellite-based geodetic networks. *Surv Rev* 38(299):412–422
- Krarpup T (1985) Contribution to the geometry of the Helmert transformation. Geodetic Institute, Denmark
- Kurt O (2018) An integrated solution for reducing ill-conditioning and testing the results in non-linear 3D similarity transformations. *Inverse Probl Sci Eng* 26(5):708–727
- Leick A, Van Gelder BHW (1975) On similarity transformations and geodetic network distortions based on Doppler satellite observations. Report No. 235, Dep. of Geodetic Sci., The Ohio State University, Columbus
- 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
- Mercan H, Akyilmaz O, Aydin C (2018) Solution of the weighted symmetric similarity transformations based on quaternions. *J Geod* 92:1113–1130
- Neitzel F (2010) Generalization of total least-squares (TLS) on example of unweighted and weighted 2D similarity transformation. *J Geod* 84(12):751–762
- Paffenholz JA, Bae KH (2012) Geo-referencing point clouds with transformational and positional uncertainties. *J Appl Geod* 6:33–46
- Schaffrin B, Felus YA (2008) On the multivariate total least-squares approach to empirical coordinate transformations. Three algorithms. *J Geod* 82(6):373–383
- Shen YZ, Chen Y, Zheng DH (2006) A quaternion-based geodetic datum transformation algorithm. *J Geod* 80:233–239
- Teunissen PJG (1988) The non-linear 2D symmetric Helmert transformation: an exact non-linear least-squares solution. *Bull Géodésique* 62:1–16
- Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans Pattern Anal Mach Intell* 13:376–380
- Walker MW, Shao L, Volz RA (1991) Estimating 3-D location parameters using dual number quaternions. *CVGIP Image Underst* 54:358–367
- Wang YB, Wang YJ, Wu K, Yang HC, Zhang H (2014) A dual quaternion-based, closed-form pairwise registration algorithm for point clouds. *ISPRS J Photogramm Remote Sens* 94:63–69
- 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
- Xu PL (2003) A hybrid global optimization method: the multi-dimensional case. *J Comput Appl Math* 155:423–446
- 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 (2014) Planar coordinate transformation and its parameter estimation in the complex number field. *Acta Geod Geophys* 49(1):79–94
- 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 WX, Tao BZ (2003) Non-linear adjustment model of three-dimensional coordinate transformation. *Geomat Inform Sci Wuhan Univ* 28(5):566–568
- Zeng H, Yi Q (2010) A new analytical solution of nonlinear geodetic datum transformation. In: Proceedings of the 18th international conference on geoinformatics
- Zeng H, Yi Q (2011) Quaternion-based iterative solution of three-dimensional coordinate transformation problem. *J Comput* 6(7):1361–1368
- Zeng H, Yi Q, Wu Y (2016) Iterative approach of 3D datum transformation with a non-isotropic weight. *Acta Geod Geophys* 51:557–570
- Zeng H, Fang X, Chang G, Yang R (2018) A dual quaternion algorithm of the Helmert transformation problem. *Earth Planets Space* 70(1):26. <https://doi.org/10.1186/s40623-018-0792-x>
- Zeng H, 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

**Publisher's Note**

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

**Submit your manuscript to a SpringerOpen<sup>®</sup> 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)