Zeng *Earth, Planets and Space*
DOI 10.1186/s40623-015-0263-6

Earth, Planets and Space
a SpringerOpen Journal

FULL PAPER
Open Access

Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix

Huaien Zeng[1,2]

Abstract

Based on the Lagrangian extremum law with the constraint that rotation matrix is an orthonormal matrix, the paper presents a new analytical algorithm of weighted 3D datum transformation. It is a stepwise algorithm. Firstly, the rotation matrix is computed using eigenvalue-eigenvector decomposition. Then, the scale parameter is computed with computed rotation matrix. Lastly, the translation parameters are computed with computed rotation matrix and scale parameter. The paper investigates the stability of the presented algorithm in the cases that the common points are distributed in 3D, 2D, and 1D spaces including the approximate 2D and 1D spaces, and gives the corresponding modified formula of rotation matrix. The comparison of the presented algorithm and classic Procrustes algorithm is investigated, and an improved Procrustes algorithm is presented since that the classic Procrustes algorithm may yield a reflection rather than a rotation in the cases that the common points are distributed in 2D space. A simulative numerical case and a practical case are illustrated.

Keywords: Weighted 3D datum transformation; Analytical algorithm; Lagrangian extremum; Constraint of orthonormal matrix; Procrustes algorithm

Background

Three-dimensional datum transformation is a frequently used work in geodesy, engineering surveying, photogrammetry, mapping, geographical information science (GIS), machine vision, etc., e.g., Aktuğ (2009), Akyilmaz (2007), El-Mowafy et al (2009), Ge et al (2013), Han and Van Gelder (2006), Horn (1986), Kashani (2006), Neitzel (2010), Paláncz et al (2013), Soler (1998), Soler and Snay (2004), Soycan and Soycan (2008), Zeng (2014). Usually, in order to compute the transformed coordinate, the transformation parameters in the transformation model (e.g., seven-parameter similarity transformation, see Aktuğ 2012, Leick 2004, Leick and van Gelder 1975,) need to be solved with several control points in advance. So far, a large number of algorithms for recovering the parameters have been presented, which can be divided into two classes. One is the numerical iterative algorithm, and the other one is analytical algorithm. The former needs the initial parameter values, linearization, and iterative computation, e.g., Zeng and Tao (2003), Chen et al. (2004), Zeng and Huang (2008), El-Habiby et al. (2009), Zeng and Yi (2011), etc. In the case that the rotation angle is large, the initial values are difficult even never to be obtained in advance, and consequently, it leads to the failure of solution (see Zeng and Yi 2011). We should note that if global optimization algorithms are used, then no initial values are required (see, e.g., Xu 2003a, Xu 2003b, Xu 2002). In contrast, the latter does not involve the initial parameter values, linearization, and iterative computation, and can give the exact solution quickly. However, because of the complexity of mathematical derivation, only several analytical algorithms have been put forward. Grafarend and Awange (2003) presented the Procrustes algorithm which utilized the singular value decomposition technique. Shen et al. (2006) presented a quaternion-based algorithm which utilized the quaternion property and eigenvalue-eigenvector decomposition. Han (2010) presented a stepwise approach to individually calculate the transformation parameters by

Correspondence: zenghuaien_2003@163.com
[1]Key Laboratory of Geological Hazards on Three Gorges Reservoir Area (China Three Gorges University), Ministry of Education, Yichang 443002, China
[2]Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University, Yichang 443002, China

the physical interpretation of similarity transformation. Zeng and Yi (2010) presented a new analytical algorithm based on the good properties of Rodrigues matrix and Gibbs vector.

The present study is organized as follows. In the Methods section, a new analytical algorithm to weighted 3D datum transformation is derived in detail, based on the Lagrangian extremum law with the constraint that the rotation matrix is an orthonormal matrix. In the meanwhile, its stability is discussed when the distribution of 3D control points degenerates into 2D (planar) or even 1D (collinear). The presented algorithm and classic Procrustes algorithm are compared, and an improved Procrustes algorithm is presented since that the classic Procrustes algorithm may yield a reflection rather than a rotation in the cases that the common points are distributed in 2D space. In the Results and discussion section, a simulative numerical case and a practical case are given to demonstrate the presented algorithm, classic Procrustes algorithm, and improved Procrustes algorithm. Lastly, conclusions are made in the Conclusions section.

Methods

Presentation of the basic algorithm

The seven-parameter similarity transformation model can be expressed as

$$a_i = \lambda R b_i + t, \tag{1}$$

subject to

$$R^{\mathrm{T}} R = I_3, \quad \det(R) = +1, \tag{2}$$

where $a_i = [X_i \quad Y_i \quad Z_i]^{\mathrm{T}}$ and $b_i = [x_i \quad y_i \quad z_i]^{\mathrm{T}}$ $i = 1, 2, \cdots, n$ are the 3D coordinates of a common point in target and source coordinate systems of transformation, tagged as system A and system B, respectively. Superscript T stands for transpose, I_3 denotes a 3 × 3 identity matrix, and det means computing the determinant of matrix. λ denotes the scale parameter, $t = [\Delta X \quad \Delta Y \quad \Delta Z]^{\mathrm{T}}$ denotes the three translation parameters, and R denotes the 3 × 3 rotation matrix, which contains the three rotation angles. Supposing R is formed by rotating angles α, β, and γ counterclockwise around the Cartesian X, Y, and Z axes, respectively, then R can be expressed by rotation angles as

$$R = \begin{bmatrix} \cos\gamma\cos\beta & \sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha - \cos\gamma\sin\beta\cos\alpha \\ -\sin\gamma\cos\beta & \cos\gamma\cos\alpha - \sin\gamma\sin\beta\sin\alpha & \cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha \\ \sin\beta & -\cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{3}$$

Using (3), the rotation angles α, β, and γ can be computed if R is recovered as

$$\alpha = -\tan^{-1}\frac{R_{32}}{R_{33}}, \quad \beta = \sin^{-1}(R_{31}), \quad \gamma = -\tan^{-1}\frac{R_{21}}{R_{11}}, \tag{4}$$

where R_{ij} is the element of R in the ith row and jth column.

Introducing the following matrix form of the coordinates as

$$A = [a_1 \quad a_2 \quad \cdots \quad a_n], \; B = [b_1 \quad b_2 \quad \cdots \quad b_n], \tag{5}$$

then Eq. (1) is rewritten as

$$A = \lambda R B + t\mathbf{1}_n, \tag{6}$$

where $\mathbf{1}_n = \underbrace{[1 \quad \cdots \quad 1]}_{n}$, i.e., a row vector with n elements and all elements are 1. It is obvious that in order to determine the seven parameters, the number n of common points must be greater than or equal to 3.

Considering the coordinates include errors, Eq. (7) is transformed as

$$A = \lambda R B + t\mathbf{1}_n + E, \tag{7}$$

where E is the transformation error matrix. The criterion of the least squares can be constructed by the Lagrangian extremum law with the constraint of Eq. (2), i.e., orthonormal matrix as follows. It is worthy of note that the constraint det(R) = + 1 is not imposed, since it can be separately treated at some extra computation as in the Stability of the basic algorithm and its modification section.

$$L(\lambda, t, \mathrm{R}, \Lambda) = \mathrm{tr}\left(EPE^{\mathrm{T}}\right) + \mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R - I_3\right)\right) = \min, \tag{8}$$

where tr denotes trace operation of matrix, Λ is a symmetric Lagrangian multiplier matrix, and P represents the weight matrix that every point has an isotropic weight and is independent of each other. Substituting the expression of E easily obtained from Eq. (7) into Eq. (8), one can obtain

$$\begin{aligned} L(\lambda, t, \mathrm{R}, \Lambda) &= \mathrm{tr}\left(EPE^{\mathrm{T}}\right) + \mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R - I_3\right)\right) \\ &= \mathrm{tr}\left((A - \lambda RB - t\mathbf{1}_n)P(A - \lambda RB - t\mathbf{1}_n)^{\mathrm{T}}\right) + \mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R - I_3\right)\right) \\ &= \min. \end{aligned} \tag{9}$$

If and only if the following conditions are satisfied, the Lagrangian extremum exists.

$$\frac{\partial L}{\partial t} = 0, \tag{10}$$

$$\frac{\partial L}{\partial \lambda} = 0, \tag{11}$$

$$\frac{\partial L}{\partial R} = 0, \tag{12}$$

$$\frac{\partial L}{\partial \Lambda} = 0. \tag{13}$$

By Eqs. (9) and (10), one gets

$$\begin{aligned} L(\lambda, t, \mathrm{R}, \Lambda) &= \mathrm{tr}\left((A-\lambda RB-t\mathbf{1}_n)P\left(A^{\mathrm{T}}-\lambda B^{\mathrm{T}}R^{\mathrm{T}}-\mathbf{1}_n{}^{\mathrm{T}}t^{\mathrm{T}}\right)\right) \\ &\quad + \mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right) \\ &= \mathrm{tr}\left((A-\lambda RB)P\left(A^{\mathrm{T}}-\lambda B^{\mathrm{T}}R^{\mathrm{T}}\right)-(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}t^{\mathrm{T}} \right. \\ &\quad \left. -t\mathbf{1}_nP\left(A^{\mathrm{T}}-\lambda B^{\mathrm{T}}R^{\mathrm{T}}\right)+t\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}t^{\mathrm{T}}\right) \\ &\quad + \mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right), \end{aligned} \tag{14}$$

$$\begin{aligned} \frac{\partial L}{\partial t} &= -(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}-(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}+2\cdot\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}t \\ &= -2(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}+2\cdot\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}t = 0, \end{aligned} \tag{15}$$

and thus,

$$t = \left(\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}\right)^{-1}(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}. \tag{16}$$

Obviously, t is the function form of λ and R.
Substituting Eq. (16) into Eq. (7), one gets

$$\begin{aligned} E &= A-\lambda RB-\left(\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}\right)^{-1}(A-\lambda RB)P\mathbf{1}_n{}^{\mathrm{T}}\mathbf{1}_n \\ &= (A-\lambda RB)\left(I_n-\left(\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}}\right)^{-1}P\mathbf{1}_n{}^{\mathrm{T}}\mathbf{1}_n\right), \end{aligned} \tag{17}$$

where $I_n - (\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}})^{-1}P\mathbf{1}_n{}^{\mathrm{T}}\mathbf{1}_n$ is the centering matrix. Let $\Delta A = A(I_n - (\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}})^{-1}P\mathbf{1}_n{}^{\mathrm{T}}\mathbf{1}_n)$, $\Delta B = B(I_n - (\mathbf{1}_nP\mathbf{1}_n{}^{\mathrm{T}})^{-1}P\mathbf{1}_n{}^{\mathrm{T}}\mathbf{1}_n)$, and thus, they are the centralized coordinate matrix, and then, Eq. (17) is written as

$$E = \Delta A-\lambda R\Delta \mathrm{B}. \tag{18}$$

Substituting Eq. (18) into Eq. (8), one gets

$$\begin{aligned} L(\lambda, \mathrm{R}, \Lambda) &= \mathrm{tr}\left((\Delta A-\lambda R\Delta B)P(\Delta A-\lambda R\Delta B)^{\mathrm{T}}\right)+\mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right) \\ &= \mathrm{tr}\left((\Delta A-\lambda R\Delta B)P\left(\Delta A^{\mathrm{T}}-\lambda\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)\right)+\mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right) \\ &= \mathrm{tr}(\Delta AP\Delta A^{\mathrm{T}}-\lambda\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}-\lambda R\Delta BP\Delta A^{\mathrm{T}} \\ &\quad +\lambda^2R\Delta BP\Delta B^{\mathrm{T}}R^{\mathrm{T}})+\mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right) \\ &= \mathrm{tr}(\Delta A^{\mathrm{T}}P\Delta A-2\lambda\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}+\lambda^2R\Delta BP\Delta B^{\mathrm{T}}R^{\mathrm{T}}) \\ &\quad +\mathrm{tr}\left(\Lambda\left(R^{\mathrm{T}}R-I_3\right)\right). \end{aligned} \tag{19}$$

Derivation of Eq. (19) makes use of the properties of trace operation, i.e.,

$$\mathrm{tr}\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right) = \mathrm{tr}\left(\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)^{\mathrm{T}}\right) = \mathrm{tr}\left(R\Delta BP\Delta A^{\mathrm{T}}\right). \tag{20}$$

Substituting Eq. (19) into Eq. (11), one gets

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= -2\mathrm{tr}\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)+2\lambda\mathrm{tr}\left(R\Delta BP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right) \\ &= -2\mathrm{tr}\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)+2\lambda\mathrm{tr}\left(\Delta BP\Delta B^{\mathrm{T}}R^{\mathrm{T}}R\right) \\ &= -2\mathrm{tr}\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)+2\lambda\mathrm{tr}\left(\Delta BP\Delta B^{\mathrm{T}}\right) \\ &= 0, \end{aligned} \tag{21}$$

$$\lambda = \frac{\mathrm{tr}\left(\Delta AP\Delta B^{\mathrm{T}}R^{\mathrm{T}}\right)}{\mathrm{tr}\left(\Delta BP\Delta B^{\mathrm{T}}\right)}. \tag{22}$$

Obviously, λ is the function form of R.
Substituting Eq. (19) into Eq. (12), one gets

$$\begin{aligned} \frac{\partial L}{\partial R} &= 2(\Delta A-\lambda R\Delta B)P\left(-\lambda\Delta B^{\mathrm{T}}\right)+R\left(\Lambda+\Lambda^{\mathrm{T}}\right) \\ &= -2\lambda\Delta AP\Delta B^{\mathrm{T}}+2\lambda^2R\Delta BP\Delta B^{\mathrm{T}}+2R\Lambda = 0, \end{aligned} \tag{23}$$

and thus,

$$R = \lambda\Delta AP\Delta B^{\mathrm{T}}\left(\lambda^2\Delta BP\Delta B^{\mathrm{T}}+\Lambda\right)^{-1}. \tag{24}$$

Substituting Eq. (19) into Eq. (13), one gets the following equation and the derivation process is given in the Appendix.

$$R^{\mathrm{T}}R-I_3 = 0, \tag{25}$$

further substituting Eq. (24) into Eq. (25), one gets

$$\lambda^2\left(\lambda^2\Delta BP\Delta B^{\mathrm{T}}+\Lambda\right)^{-1}\Delta BP\Delta A^{\mathrm{T}}\Delta AP\Delta B^{\mathrm{T}}\left(\lambda^2\Delta BP\Delta B^{\mathrm{T}}+\Lambda\right)^{-1} = I_3, \tag{26}$$

so

$$\lambda^2\Delta BP\Delta B^{\mathrm{T}}+\Lambda = \lambda\left(\Delta BP\Delta A^{\mathrm{T}}\Delta AP\Delta B^{\mathrm{T}}\right)^{\frac{1}{2}}, \tag{27}$$

and substituting Eq. (27) into Eq. (24), one gets

$$R = \Delta AP\Delta B^{\mathrm{T}}\left(\Delta BP\Delta A^{\mathrm{T}}\Delta AP\Delta B^{\mathrm{T}}\right)^{-\frac{1}{2}}. \tag{28}$$

Let

$$D = \Delta AP\Delta B^{\mathrm{T}}, \tag{29}$$

then Eq. (28) can be rewritten as

$$R = D\left(D^{\mathrm{T}}D\right)^{-\frac{1}{2}}. \tag{30}$$

Note that $D^{\mathrm{T}}D$ is symmetric, non-negative definitive and so has non-negative real eigenvalues. The inverse of the square root of $D^{\mathrm{T}}D$ can thus be computed using eigenvalue-eigenvector decomposition.

$$\left(D^{\mathrm{T}}D\right)^{-\frac{1}{2}} = \frac{\nu_1\nu_1{}^{\mathrm{T}}}{\sqrt{d_1}}+\frac{\nu_2\nu_2{}^{\mathrm{T}}}{\sqrt{d_2}}+\frac{\nu_3\nu_3{}^{\mathrm{T}}}{\sqrt{d_3}}, \tag{31}$$

Table 1 Simulative true coordinates of control points in system B

Point number	Set 1 (m)			Set 2 (m)			Set 3 (m)		
	x	*y*	*z*	*x*	*y*	*z*	*x*	*y*	*z*
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)			Set 5 (m)			Set 6 (m)		
	x	*y*	*z*	*x*	*y*	*z*	*x*	*y*	*z*
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			

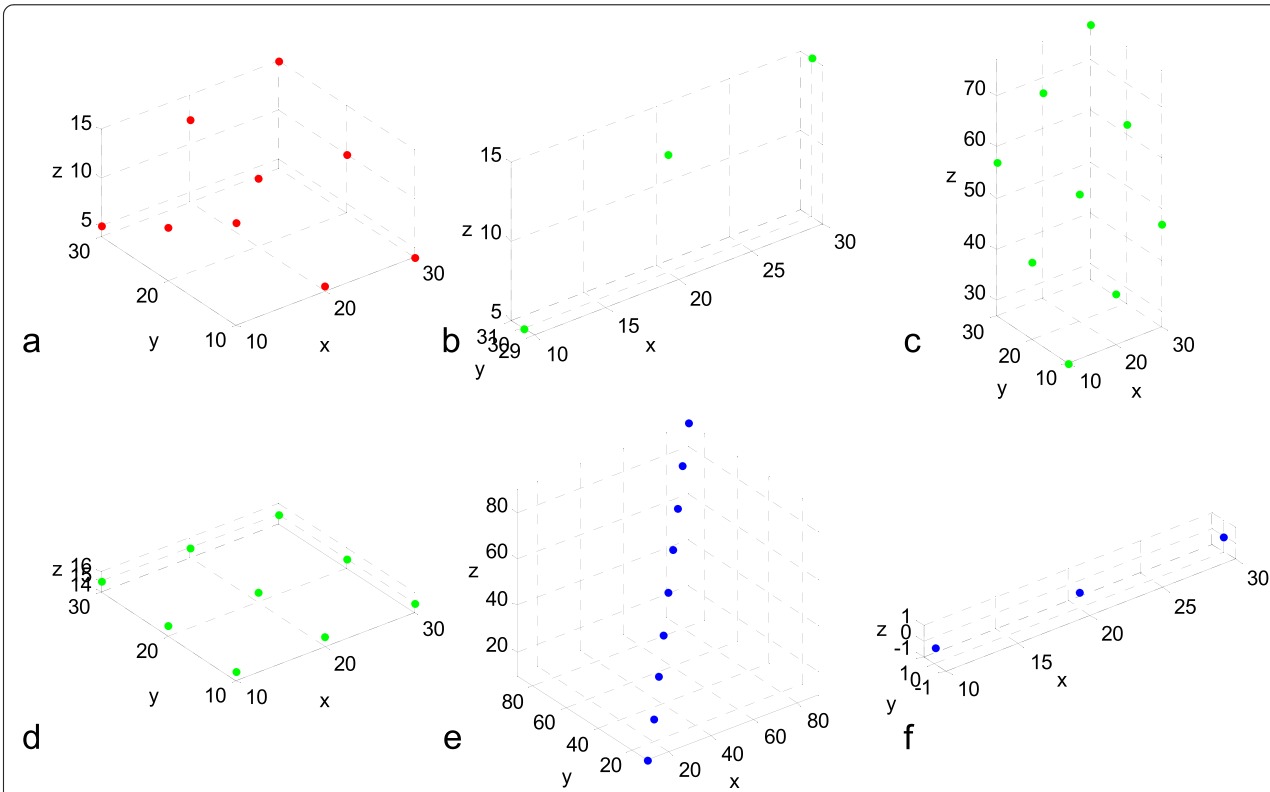

Fig. 1 The distribution of control point in system B. **a** Set 1. **b** Set 2. **c** Set 3. **d** Set 4. **e** Set 5. **f** Set 6

Table 2 Simulative theoretical values of transformation parameters

ΔX (m)	ΔY (m)	ΔZ (m)	α	β	γ	λ
30	30	10	71°	78°	73°	1.000016

where d_i and v_i for $i = 1, 2, 3$ are the eigenvalues and corresponding eigenvectors of the matrix $D^{\mathrm{T}}D$. So, Eq. (30) can be written as

$$R = D\left(\frac{v_1 {v_1}^{\mathrm{T}}}{\sqrt{d_1}} + \frac{v_2 {v_2}^{\mathrm{T}}}{\sqrt{d_2}} + \frac{v_3 {v_3}^{\mathrm{T}}}{\sqrt{d_3}}\right) \tag{32}$$

Stability of the basic algorithm and its modification

Obviously, the construction of the inverse of the square root of $D^{\mathrm{T}}D$, i.e., Eq. (31), fails if one or two of d_i for $i = 1, 2, 3$ equals to 0. Assume that the eigenvalues of the matrix $D^{\mathrm{T}}D$ satisfies the following condition.

$$d_1 \leq d_2 \leq d_3. \tag{33}$$

When the common points are distributed in a plane, i.e., 2D space, the matrix $D^{\mathrm{T}}D$ is singular and of rank 2, and thus,

$$d_1 = \mathbf{0}, \quad d_3 \geq d_2 > \mathbf{0}. \tag{34}$$

One can compute the rotation matrix as follows.

$$D_1 = \left(\frac{v_2 {v_2}^{\mathrm{T}}}{\sqrt{d_2}} + \frac{v_3 {v_3}^{\mathrm{T}}}{\sqrt{d_3}}\right), \tag{35}$$

$$D_2 = \left(DD_1 (DD_1)^{\mathrm{T}} - I_3\right) v_1 {v_1}^{\mathrm{T}}, \tag{36}$$

$$R = DD_1 \pm \frac{D_2}{\sqrt{|\mathrm{tr}(D_2)|}}, \tag{37}$$

where the sign ± is chosen so that $\det(R) = +1$ is satisfied.

The above case is the ideal one, but in the case that the common points are distributed in an approximate plane, although

$$d_3 \geq d_2 > d_1 > \mathbf{0}, \tag{38}$$

and the order of magnitude of d_i for $i = 1, 2, 3$ differs greatly from each other. In other words, the matrix $D^{\mathrm{T}}D$ is approximately singular, and the condition number of matrix $D^{\mathrm{T}}D$ is very big (usually the threshold is set to 100). The Eq. (31) is ill-conditioned and leads to a biased solution. At this time, one can also obtain the exact solution by Eqs. (35)–(37).

Table 3 Simulative true coordinates of control points in system A

Point number	Set 1 (m)			Set 2 (m)			Set 3 (m)		
	X	*Y*	*Z*	*X*	*Y*	*Z*	*X*	*Y*	*Z*
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)			Set 5 (m)			Set 6 (m)		
	X	*Y*	*Z*	*X*	*Y*	*Z*	*X*	*Y*	*Z*
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 4 Calculated transformation parameters and mean error

Parameters	Set 1			Set 2			Set 3		
	PA	CPA	IPA	PA	CPA	IPA	PA	CPA	IPA
ΔX (m)	30.000215	30.000215	30.000215	29.997125	64.907810	29.997125	29.999564	29.884451	29.999564
ΔY (m)	30.000014	30.000014	30.000014	29.999418	−17.354508	29.999418	30.000156	31.848392	30.000156
ΔZ (m)	9.999992	9.999992	9.999992	10.000804	−1.797635	10.000804	9.999562	9.420190	9.999562
α (°)	70.998025	70.998025	70.998025	70.994443	−70.994443	70.994443	70.999494	60.478813	70.999494
β (°)	77.999873	77.999873	77.999873	77.996704	77.996704	77.996704	77.999588	43.508921	77.999588
γ (°)	73.001648	73.001648	73.001648	73.000253	73.000253	73.000253	73.000571	−89.744632	73.000571
λ	1.000012	1.000012	1.000012	1.000049	1.000049	1.000049	1.000025	1.000025	1.000025
ME (m)	0.000315	0.000315	0.000315	0.000197	0.000197	0.000197	0.000313	0.000313	0.000313
Parameters	Set 4			Set 5			Set 6		
	PA	CPA	IPA	PA	CPA	IPA	PA	CPA	IPA
ΔX (m)	29.999778	29.999778	29.999778	30.000278	30.000278	30.000278	30.000000	30.000000	30.000000
ΔY (m)	30.000191	30.000191	30.000191	30.000389	30.000389	30.000389	30.000333	30.000333	30.000333
ΔZ (m)	9.999647	9.999647	9.999647	10.000083	10.000083	10.000083	10.000333	10.000333	10.000333
α (°)	71.000802	71.000802	71.000802	−45.000000	−35.145218	−54.854782	unsolved	−61.858215	−61.858215
β (°)	78.000742	78.000742	78.000742	16.444350	−24.651859	−24.651859	77.998588	77.998588	77.998588
γ (°)	72.999769	72.999769	72.999769	15.645706	6.418995	6.418995	72.998563	72.998563	72.998563
λ	1.000028	1.000028	1.000028	1.000016	1.000016	1.000016	1.000008	1.000008	1.000008
ME (m)	0.000294	0.000294	0.000294	0.000296	0.000296	0.000296	0.000407	0.000407	0.000407

When the common points are distributed in a line, i.e., 1D space, the matrix $D^{\mathrm{T}}D$ is singular and of rank 1, and thus,

$$d_2 = d_1 = \mathbf{0}, \quad d_3 > \mathbf{0}. \tag{39}$$

The rotation matrix is impossible to recover in a whole; however, one can recover at most two rotation angles by the following formula.

$$R = \frac{v_3 {v_3}^{\mathrm{T}}}{\sqrt{d_3}}, \tag{40}$$

and the utilization of Eq. (40) make it feasible to compute the translation parameter and scale parameter.

The above case is the ideal one, but in the case that the common points are distributed in an approximate line, although

$$d_3 > d_2 > d_1 = \mathbf{0}, \tag{41}$$

and the order of magnitude of d_i for $i = 2, 3$ differs greatly from each other (e.g., $d_3 > 100d_2$). The computation by Eqs. (35)–(37) is biased, and no exact solution can be found like the case that the common points are distributed in a line space. At this time, one can carry out the computation by Eq. (40).

Comparison to classic Procrustes algorithm and improvement of the classic Procrustes algorithm

The classic Procrustes algorithm presented by Grafarend and Awange (2003) is a well-known analytical algorithm of 3D datum transformation. It is also based on the Lagrangian extremum law, similarly to the presented algorithm in this paper. But differently, it does not constrain the orthonormal matrix condition. For the Procrustes algorithm, due to utilization of the singular value decomposition technique, the computed rotation matrix always satisfies the constraint condition $R^{\mathrm{T}}R = I_3$; however, in the cases that the common points are

Table 5 Number of solvable transformation parameters

Space	PA				CPA				IPA			
	Translation	Angle	Scale	Total	Translation	Angle	Scale	Total	Translation	Angle	Scale	Total
3D	3	3	1	7	3	3	1	7	3	3	1	7
2D	3	3	1	7	0–3	0–3	1	1–7	3	3	1	7
1D	3	0–2	1	4–6	3	0–2	1	4–6	3	0–2	1	4–6

Table 6 Coordinates of control points in systems B and A

Station name	System B (local system) (m)			System A (WGS-84) (m)		
	x	*y*	*z*	*X*	*Y*	*Z*
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

distributed in a rigid or approximate plane, this situation that $\det(R) = -1$ rather than $\det(R) = +1$ usually happens. This means the computed R is a reflection instead of a rotation. For the presented algorithm in this paper, the constraint condition $R^{\mathrm{T}}R = I_3$ is imposed in the computation, and thus, it is always satisfied. And in the cases that the common points are distributed in a rigid or approximate plane, the sign ± in Eq. (36) is properly chosen so that $\det(R) = +1$ is satisfied.

To recover the exact rotation matrix by Procrustes algorithm when $\det(R) = -1$, the computation formula of R, i.e., Eq. (22) in Grafarend and Awange (2003) should be improved as

$$X_{3l} = U\tilde{V}^{\mathrm{T}}, \tag{42}$$

where $\tilde{V} = [\,V_1 \quad V_2 \quad -V_3\,]$, V_1, V_2, and V_3 are the column matrix of V, and V_3 is the column matrix that corresponds to the singular value that is 0.

Therefore, the general improved computation formula of R, i.e., Eq. (22) in Grafarend and Awange (2003) is

$$X_{3l} = U\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det\left(UV^{\mathrm{T}}\right) \end{bmatrix}V^{\mathrm{T}}, \tag{43}$$

which is stable for the cases that the common points are distributed in 3D and 2D spaces including approximate 2D space. In the case that the common points are distributed in 1D space, Eq. (43) can recover at most two rotation angles and be used to compute the translation parameter and scale parameter.

Results and discussion

Simulative case

The case data is simulated as follows. In order to investigate the stability performance of the presented algorithm (PA) in this paper, the classic Procrustes algorithm (CPA) and improved Procrustes algorithm (IPA) in the cases that the control points are distributed in 3D, 2D, and 1D spaces, six sets of control point in system B is first given in Table 1, of which set 1 is distributed in 3D space, sets 2, 3, and 4 are distributed in 2D space, and sets 5 and 6 are distributed in 1D space. The distribution of six sets of control point is depicted intuitively in Fig. 1. Set 2 has only three control points, and it is necessary for the least point number to solve the seven parameters. Secondly, the theoretical seven parameters are given in Table 2. For the sake of an efficient test of the algorithms, the rotation angles are designed to be big angles. Thirdly, the coordinates of control points in system A are computed by Eq. (1), and the result is listed in Table 3. In this case, the stability of the three algorithms is focused, so the weight matrix is designated to identity matrix for easy demonstration.

Next, the transformation seven parameters are recovered by the three algorithms, and the result is listed in Table 4. ME in Table 4 is the mean error and computed by

$$\mathrm{ME} = \sqrt{\frac{\mathrm{tr}\left(E^{\mathrm{T}}PE\right)}{3n-7}}. \tag{44}$$

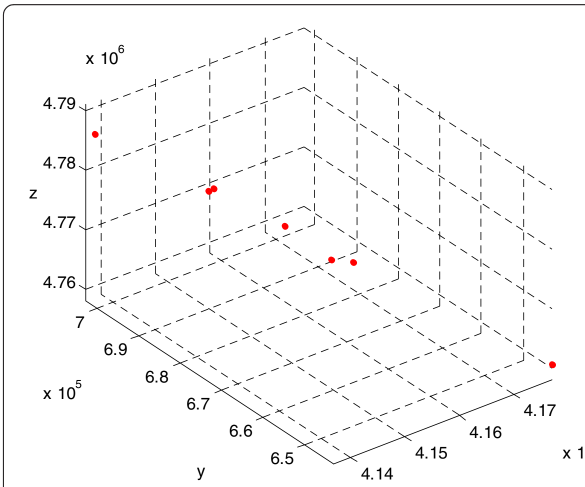

Fig. 2 The distribution of seven control points in local system

Table 7 Calculated transformation parameters (identity weight matrix)

	PA			CPA		
Rotation matrix						
R	1.0000000000	0.0000048146	−0.0000043328	1.0000000000	0.0000048146	−0.0000043328
	−0.0000048146	1.0000000000	−0.0000048408	−0.0000048146	1.0000000000	−0.0000048408
	0.0000043327	0.0000048408	1.0000000000	0.0000043327	0.0000048408	1.0000000000
Rotation angles (″)						
α		−0.998496121			−0.998496121	
β		0.893693325			0.893693325	
γ		0.993086229			0.993086229	
Translation (m)						
ΔX		641.8805			641.8805	
ΔY		68.6551			68.6551	
ΔZ		416.3982			416.3982	
Scale						
λ		1.000005583			1.000005583	
ME (m)		0.0773			0.0773	

It is seen from Table 4 that the results of seven parameters are identical and accurate for the three algorithms in the case of set 1, i.e., the case that the control points are distributed in 3D space. For the cases that the control points are distributed in 2D space, PA and IPA have the identical and accurate results of seven parameters, but CPA has one correct result of seven parameters for set 4 and two wrong results of rotation angles for sets 2 and 3, because reflections rather than rotations are found, and all translation and scale parameters are correct. For the cases that the control points are distributed in 1D space, i.e., sets 5 and 6, the three algorithms all recover the correct translation and scale parameters and at most recover two rotation angles. The number of solvable transformation parameters for the three algorithms is counted and listed in Table 5.

Actual case

The case data is from Grafarend and Awange (2003). The coordinates of control points in system B (local system) and A (WGS-84 system) is listed in Table 6. The distribution of seven control points in local system is depicted in Fig. 2. From this figure, it is seen that the distribution of control points are in an approximate plane. In the process of PA computation, condition number of matrix $D^{\mathrm{T}}D$ is 2.5×10^{11}, so Eq. (32) is seriously ill-conditioned and yields a biased solution if not processed. When the weight matrix is an identity matrix, the computed results of seven parameters with PA and CPA are listed in Table 7. For the situation that the weight matrix is a point-wise matrix, i.e., every point has isotropic weight and is independent of each other, the point-wise matrix is generated by the way introduced in Grafarend and Awange (2003) and is listed in Table 8. The computed results of seven parameters with PA and CPA are listed in Table 9.

It is seen from Tables 7 and 9 that the results of PA and CPA are identical if the bias caused by decimal rounding is ignored. Hence, the PA is comparable with CPA.

Conclusions

The numerical case study shows that the presented new algorithm and improved Procrustes algorithm are both stable and reliable for the cases that the control points are distributed in 3D, and 2D including approximate 2D space, and can recover at most two angles as well as all translation and scale parameters for the cases that the control points are distributed in 1D space. The classic Procrustes algorithm also can compute all translation and scale parameters for all the cases that the control

Table 8 Point-wise weight matrix

Values						
2.170137	0	0	0	0	0	0
0	2.097755	0	0	0	0	0
0	0	2.208968	0	0	0	0
0	0	0	2.201671	0	0	0
0	0	0	0	2.182928	0	0
0	0	0	0	0	2.268808	0
0	0	0	0	0	0	2.643404

Table 9 Calculated transformation parameters (point-wise weight matrix)

	PA			CPA		
Rotation matrix						
R	1.0000000000	0.0000047797	−0.0000043444	1.0000000000	0.0000047797	−0.0000043444
	−0.0000047797	1.0000000000	−0.0000048370	−0.0000047797	1.0000000000	−0.0000048370
	0.0000043443	0.0000048371	1.0000000000	0.0000043443	0.0000048371	1.0000000000
Rotation angles (″)						
α		−0.997716185			−0.997716186	
β		0.896085615			0.896085615	
γ		0.985885069			0.985885070	
Translation (m)						
ΔX		641.8395			641.8395	
ΔY		68.4729			68.4729	
ΔZ		416.2156			416.2156	
Scale						
λ		1.000005611			1.000005611	
Mean error (m)		0.1140			0.1140	

points are distributed in 3D, 2D, and 1D spaces; however, it may yield a reflection rather than a rotation in the cases that the common points are distributed in 2D space. The numerical case study also shows the presented algorithm in this paper, and improved Procrustes algorithm is both stable and reliable when the rotation angles are big. And the presented algorithm in this paper is comparable with the classic Procrustes algorithm when the point-wise weight matrix is involved.

Appendix

Let

$$K = R^{\mathrm{T}}R–I_3, \tag{45}$$

$$K^{\mathrm{T}} = R^{\mathrm{T}}R–I_3 = K, \tag{46}$$

then K is a symmetric matrix. Since that Λ is a symmetric Lagrangian multiplier matrix, Λ and K can be described as follows.

$$\Lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_2 & \lambda_4 & \lambda_5 \\ \lambda_3 & \lambda_5 & \lambda_6 \end{bmatrix}, \tag{47}$$

$$K = \begin{bmatrix} \kappa_1 & \kappa_2 & \kappa_3 \\ \kappa_2 & \kappa_4 & \kappa_5 \\ \kappa_3 & \kappa_5 & \kappa_6 \end{bmatrix}. \tag{48}$$

$$\begin{aligned} \mathrm{tr}(\Lambda K) &= \lambda_1\kappa_1 + \lambda_2\kappa_2 + \lambda_3\kappa_3 + \lambda_2\kappa_2 + \lambda_4\kappa_4 + \lambda_5\kappa_5 \\ &\quad + \lambda_3\kappa_3 + \lambda_5\kappa_5 + \lambda_6\kappa_6 \\ &= \lambda_1\kappa_1 + 2\lambda_2\kappa_2 + 2\lambda_3\kappa_3 + \lambda_4\kappa_4 + 2\lambda_5\kappa_5 + \lambda_6\kappa_6, \end{aligned} \tag{49}$$

$$\begin{aligned} \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\Lambda} &= \begin{bmatrix} \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_1} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_2} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_3} \\ \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_2} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_4} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_5} \\ \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_3} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_5} & \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\lambda_6} \end{bmatrix} \\ &= \begin{bmatrix} \kappa_1 & 2\kappa_2 & 2\kappa_3 \\ 2\kappa_2 & \kappa_4 & 2\kappa_5 \\ 2\kappa_3 & 2\kappa_5 & \kappa_6 \end{bmatrix}. \end{aligned} \tag{50}$$

By Eqs. (19) and (13), one gets

$$\frac{\delta L}{\delta\Lambda} = \frac{\delta \mathrm{tr}(\Lambda K)}{\delta\Lambda} = 0, \tag{51}$$

and then,

$$\begin{bmatrix} \kappa_1 & 2\kappa_2 & 2\kappa_3 \\ 2\kappa_2 & \kappa_4 & 2\kappa_5 \\ 2\kappa_3 & 2\kappa_5 & \kappa_6 \end{bmatrix} = 0. \tag{52}$$

Further,

$$\kappa_i = 0 \ \ (i = 1, 2, \cdots, 6). \tag{53}$$

So,

$$K = R^{\mathrm{T}}R–I_3 = 0. \tag{54}$$

Competing interests

The author declares that he has no competing interests.

Acknowledgments

The study is supported by the Open Foundation of Hubei Key Laboratory of Construction and Management in Hydropower Engineering, China Three Gorges University (Grant No. 2014KSD13), the National Natural Science Foundation of China (Grant No. 41104009), the Open Research Fund Program of the Key Laboratory of Geospace Environment and Geodesy, Ministry of Education, China (Grant No. 11-01-04), and the 2015 Open Foundation of the Key Laboratory of Precise Engineering and Industry Surveying, National Administration of Surveying, Mapping, and Geoinformation of China. The author is grateful for the support and good working atmosphere provided by his research team in China Three Gorges University. Last but not the least, the author's special thanks are given to Dr. Koji Matsumoto and two anonymous reviewers for their valuable comments and suggestions, which enhance the quality of this manuscript.

Received: 3 March 2015 Accepted: 4 June 2015
Published online: 30 June 2015

References

Aktuğ B (2009) Inverse and compound datum/frame transformations. J Surv Eng 135(2):46–55
Aktuğ B (2012) Weakly multicollinear datum transformations. J Surv Eng 138(4):184–192
Akyilmaz O (2007) Total least squares solution of coordinate transformation. Surv Rev 39(303):68–80
Chen Y, Shen YZ, Liu DJ (2004) A simplified model of three dimensional-datum transformation adapted to big rotation angle. Geomat Info Sci Wuhan Uni 29(12):1101–1105
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
El-Mowafy A, Fashir H, Al-Marzooqi Y (2009) Improved coordinate transformation in Dubai using a new interpolation approach of coordinate differences. Surv Rev 41(311):71–85
Ge Y, Yuan Y, Jia N (2013) More efficient methods among commonly used robust estimation methods for GPS coordinate transformation. Surv Rev 45(330):229–234
Grafarend EW, Awange JL (2003) Nonlinear analysis of the three-dimensional datum transformation [conformal group C7(3)]. J Geod 77:66–76
Han JY, Van Gelder HWB (2006) Step-wise parameter estimations in a time-variant similarity transformation. J Surv Eng 132(4):141–148
Han JY (2010) Noniterative approach for solving the indirect problems of linear reference frame transformations. J Surv Eng 136(4):150–156
Horn BKP (1986) Robot vision. MIT Press, Cambridge, Massachusetts and McGraw-Hill, New York
Kashani I (2006) Application of generalized approach to datum transformation between local classical and satellite-based geodetic networks. Surv Rev 38(299):412–422
Leick A (2004) GPS satellite surveying, 3d/Ed. Wiley, Hoboken
Leick A, van Gelder BHW (1975) On similarity transformations and geodetic network distortions based on Doppler satellite observations. Rep. No. 235. Dept. of Geodetic Science, Ohio State Univ, Columbus, Ohio
Neitzel F (2010) Generalization of total least-squares on example of weighted 2D similarity transformation. J Geod 84(12):751–762. doi:10.1007/s00190-010-0408-0
Paláncz B, Awange JL, Völgyesi L (2013) Pareto optimality solution of the Gauss-Helmert model. Acta Geod Geophys 48:293–304
Shen YZ, Chen Y, Zheng DH (2006) A quaternion-based geodetic datum transformation algorithm. J Geod 80:233–239
Soler T (1998) A compendium of transformation formulas useful in GPS work. J Geodyn 72(7–8):482–490
Soler T, Snay RA (2004) Transforming positions and velocities between the International Terrestrial Reference Frame of 2000 and North American Datum of 1983. J Surv Eng 130(2):49–55
Soycan M, Soycan A (2008) Transformation of 3D GPS Cartesian coordinates to ED50 using polynomial fitting by robust re-weighting technique. Surv Rev 40(308):142–155
Xu PL (2002) A hybrid global optimization method: the one-dimensional case. J Comput Appl Math 147:301–314
Xu PL (2003a) A hybrid global optimization method: the multi-dimensional case. J Comput Appl Math 155:423–446
Xu PL (2003b) Numerical solution for bounding feasible point sets. J Comput Appl Math 156:201–219
Zeng HE (2014) Planar coordinate transformation and its parameter estimation in the complex number field. Acta Geod Geophys 49(1):79–94
Zeng HE, Huang SX (2008) A kind of direct search method adopted to solve 3D coordinate transformation parameters. Geomat Info Sci Wuhan Uni 33(11):1118–1121
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 of Computers 6(7):1361–1368
Zeng WX, Tao BZ (2003) Non-linear adjustment model of three-dimensional coordinate transformation. Geomat Info Sci Wuhan Uni 28(5):566–568