Quaternion-Based Iterative Solution of Three-Dimensional Coordinate Transformation Problem

Huaien Zeng and Qinglin Yi
College of Civil Engineering and Architecture, China Three Gorges University, Yichang, China
zenghuaien_2003@yahoo.com.cn

Abstract—**Three-dimensional coordinate transformation problem is the most frequent problem in photogrammetry, geodesy, mapping, geographical information science (GIS), and computer vision. To overcome the drawback that traditional solution of the problem based on rotation angles depends strongly on initial value of parameter, which makes the method ineffective in the case of super-large rotation angle, the paper adopts an unit quaternion to represent three-dimensional rotation matrix, then puts forward a quaternion-based iterative solution of the problem. The cases study shows that the quaternion-based solution has no dependence on the initial value of parameter and desirable result with fast speed. Thus it is valid for three-dimensional coordinate transformation of any rotation angle.**

Index Terms—**three-dimensional coordinate transformation, quaternion, rotation matrix, initial value of parameter, parameter adjustment with constraint, improved Gauss-Newton method**

I. INTRODUCTION

Three-dimensional (3D) coordinate transformation is the most common issue in geodesy, photogrammetry, geographical information science (GIS), computer vision and other research areas. It involves transforming spatial data (locations, images, maps, etc.) from an original coordinate system to a target coordinate system by means of mathematical transformation model. Presently, the most frequently model is the similarity transformation model with seven parameters (namely, one scale factor, three translation parameter, and three rotation angles.), also known as Helmert or conformal group $C_7(3)$ transformation, which is employed in the paper. To carry out coordinate transformation, it is critical to calculating the seven parameters, usually by some control points with the coordinates in the both systems.

In geodesy, because the rotation angles are generally very small, namely the two coordinate systems are nearly aligned; the similarity transformation model is simplified to a linear one (e.g., [1]-[2]), whose parameters are easy to solve. A lot of literatures on coordinate transformation from World Geodetic System 1984 (WGS84) to a local system have been published (e.g., [2]-[5]). It is notable that [5] presented a stepwise approach to individually calculate the seven parameters by the geometric properties of similarity transformation.

In photogrametry and computer vision, three-dimensional coordinate transformation is employed to relate image space coordinates to object space coordinates in the so-called "absolute orientation" problem ([6]) or to register multi-station point clouds in a LIDAR surveying ([7]). In these cases, the rotation angles are almost not small and require the solution of nonlinear three-dimensional coordinate transformation model.

Many algorithms have been presented to compute the transformation parameters from the nonlinear over-determined equations of coordinate transformation in least-squares (LS) sense. They can be divided into two categories, i.e., iterative algorithms and analytical algorithms. The former are dominant, e.g., [8]-[11]. The major difference between these algorithms is caused due to the different representations of rotation matrix, which lead to the different linearization models. However, the iterative algorithms traditionally need good initial starting values of parameters and linearization process. It is difficult to implemented in the cases of large rotation angles because the initial values are difficult even impossible to get in advance. At present, the analytical algorithm is rarely seen, of which two key algorithms are presented, known as the Procrustes algorithm ([12]) and a quaternion-based algorithm ([13]). The authors presented a new analytical algorithm based on optimization process and the good properties of Rodrigues matrix ([14]).

To solve the problem that traditional algorithms with the mathematical model based on rotation angles depend strongly on the initial values of parameters, and calculate slowly because of the existing numerous trigonometric computation, the paper will investigate the feasibility of coordinate transformation model with representation of quaternion, and present a efficient algorithm to compute the transformation parameters.

The remainder of the paper is organized as follows. Section II briefly reviews the concept and properties of quaternion, and then derives the representation of rotation matrix by unit quaternion. Section III derives the mathematical model of 3D coordinate transformation inverse problem based on unit quaternion in detail, and presents the solution of transformation parameters. In order to speed up the convergence of iterative calculation,

This work is supported by Youth Science Foundation of China Three Gorges University (No.KJ2009A004) and Talent Research Start-up Fund of China Three Gorges University (No.KJ2009B008).
corresponding author: Huaien Zeng

doi:10.4304/jcp.6.7.1361-1368

we design an improved Gauss-Newton method to substitute the most frequently used classical Gauss-Newton method in the adjustment of geodetic photographic data, etc. The simulative and practical cases are studied to validate the presented algorithm in the next two sections, i.e., Sect. IV and Sect. V respectively. Finally, conclusions are made in Sect. VI.

II. QUATERNION AND 3D ROTATION MATRIX

A. Concept and Properties of Quaternion

Quaternion was a mathematic concept invented by Hamilton in 1843, which is represented as follows [15].

$$Q = q_1 + iq_2 + jq_3 + kq_4, \tag{1}$$

where q_1 is the real part, q_2, q_3 and q_4 are the imaginary part, i, j and k are imaginary units, and they meet the relationships: ① $i^2 = j^2 = k^2 = -1$, ② $ij = -ji = k$, ③ $jk = -kj = i$, ④ $ki = -ik = j$. The corresponding conjugate quaternion can be denoted as

$$Q^* = q_1 - iq_2 - jq_3 - kq_4. \tag{2}$$

In order to simplify the description, Q is expressed as $\begin{pmatrix} q_1 & q^T \end{pmatrix}^T$ in the column vector form with respect to the bases $\begin{pmatrix} 1 & i & j & k \end{pmatrix}$, where $q = \begin{pmatrix} q_2 & q_3 & q_4 \end{pmatrix}^T$ denotes a 3D vector, q_1 denotes a scalar, and T the transpose. The norm of quaternion Q is defined as

$$\|Q\| = \sqrt{q_1{}^2 + q_2{}^2 + q_3{}^2 + q_4{}^2}. \tag{3}$$

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

According to the definition of quaternion, it is easily proved that the following properties are satisfied for quaternion

$$P + Q = \begin{pmatrix} p_1 + q_1 & p^T + q^T \end{pmatrix}^T, \tag{4}$$

$$PQ = p_1q_1 - p \cdot q + p \times q + p_1q + q_1p, \tag{5}$$

$$OPQ = (OP)Q = O(PQ), \tag{6}$$

$$O(P+Q) = OP + OQ, \tag{7}$$

$$(OP)^* = P^*O^*, \tag{8}$$

$$QQ^* = \|Q\|, \tag{9}$$

$$Q^{-1} = Q^* / \|Q\|, \tag{10}$$

where O, P and Q are quaternions, Q^{-1} denotes the inverse of the quaternion Q, and the symbols $\cdot$ and $\times$ stand for the dot product and cross product, respectively. The dot and cross product of vectors are defined as

$$p \cdot q = p^T q, \tag{11}$$

$$p \times q = c(p)q, \tag{12}$$

where

$$c(p) = \begin{bmatrix} 0 & -p_4 & p_3 \\ p_4 & 0 & -p_2 \\ -p_3 & p_2 & 0 \end{bmatrix}. \tag{13}$$

The product quaternion PQ can be expressed in the column vector and matrix form as

$$PQ = C(P)Q = \overline{C(Q)}P, \tag{14}$$

where

$C(P) = \begin{bmatrix} p_1 & -p^T \\ p & p_1I + c(p) \end{bmatrix}$,

$\overline{C(Q)} = \begin{bmatrix} q_1 & -q^T \\ q & q_1I - c(q) \end{bmatrix}$, and I denotes a 3×3 identity matrix.

B. 3D Rotation Matrix Represented by Quaternion

Supposing vector s is produced of vector p by means of rotation angle of θ around axis OA, and the OA-axis unit vector is r (see Fig. 1), a well-known method to represent the rotation of p to s is derived with quaternion [16]

$$S = QPQ^* = C(Q)\overline{C(Q^*)}P, \tag{15}$$

where P and S are the quaternion forms of vectors p and s with scalar both as zero, Q is a unit quaternion formed by θ and r as

$$Q = \cos(\theta/2) + r\sin(\theta/2), \tag{16}$$

where, $r = ir_1 + jr_2 + kr_3$, and $r_1^2 + r_2^2 + r_3^2 = 1$.

$C(Q)\overline{C(Q^*)}$ in (16) can be expanded as $\begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix}$, where

$$R = \left(q_1^2 - q^T q\right)I + 2\left(qq^T + q_1c(q)\right). \tag{17}$$

R in (17) is the 3D rotation matrix, whose elements are composed of the unit quaternion Q.

III. QUATERNION-BASED ITERATIVE SOLUTION OF 3D COORDINATE TRANSFORMATION PROBLEM

A. Mathematic Model

The seven-parameter similarity transformation model can be expressed as [2]

$$a_i = \lambda R b_i + t, \tag{18}$$

where $a_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T$ and $b_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ ($i = 1, 2, \cdots, n$) are two sets of co-located 3D coordinates in two different systems, tagged as system A and system B

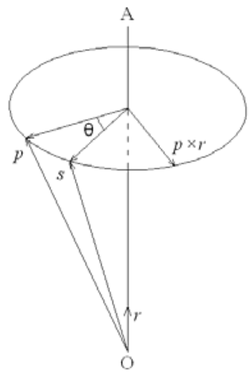


Figure 1. Rotation of vector and the physical meaning of quaternion

respectively, $t = \begin{bmatrix} \Delta X & \Delta Y & \Delta Z \end{bmatrix}^T$ denotes three translation parameters, λ denotes the scale parameter and R denotes the 3×3 rotation matrix, which contains the three rotation angles. It is Obvious that in order to determine the seven parameters, the number n of co-located coordinates a_i, b_i must be greater than or equal to three.

If we substitute (17) into (18), we can obtain the quaternion-based non-linear 3D coordinate transformation model. In terms of linearization of the model, we obtain the observation equation. However, the corresponding normal equation in actual adjustment usually doesn't avoid ill-posed property. For this reason, we transform (18) to another form by means of baseline vector, namely difference of coordinates which eliminates the three translation parameters as follows

$$\Delta a_i = \lambda R \Delta b_i, \tag{19}$$

where $\Delta a_i = a_i - a_0$, $\Delta b_i = b_i - b_0$, a_0 and b_0 denote two sets of co-located 3D coordinates for the starting point of all baselines, and the starting point can be often supposed to be a point with high accuracy. By means of linearization of (19), the observation equation are obtained as follows

$$V_i = B_i \delta x - l_i, \tag{20}$$

where $V_i = \begin{bmatrix} V_{xi} & V_{yi} & V_{zi} \end{bmatrix}^T$ denotes correction of Δa_i, $\delta x = \begin{bmatrix} dq_1 & dq_2 & dq_3 & dq_4 & d\lambda \end{bmatrix}^T$ denotes correction of unkowns $x = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \lambda \end{bmatrix}^T$, B_i is a 3×5 coefficient matrix as

$$B_i = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} & K_1 \\ B_{21} & B_{22} & B_{23} & B_{24} & K_2 \\ B_{31} & B_{32} & B_{33} & B_{34} & K_3 \end{bmatrix}, \tag{21}$$

$l_i = \begin{bmatrix} l_{xi} & l_{yi} & l_{zi} \end{bmatrix}^T$ is a constant matrix, and the elements of B_i and l_i are listed in Appendix A.

Because Q is a unit quaternion, there is a constraint accompanying (20) as follows

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{22}$$

Linearizing (22), we obtain

$$C\delta x - W_x = 0, \tag{23}$$

where

$C = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & 0 \end{bmatrix}$,

$W_x = (1 - q_1^2 - q_2^2 - q_3^2 - q_4^2)/2$.

B. Classical Solution of the Transformation Parameters

When the number of co-located points $n \geq 3$, we can establish $3(n-1)$ observation equations like (20) as

$$V = B\delta x - l, \tag{24}$$

where $V = \begin{bmatrix} V_1 \\ \vdots \\ V_{n-1} \end{bmatrix}$, $B = \begin{bmatrix} B_1 \\ \vdots \\ B_{n-1} \end{bmatrix}$, $l = \begin{bmatrix} l_1 \\ \vdots \\ l_{n-1} \end{bmatrix}$, with a constraint, i.e., (23). Then we can solve the transformation parameters by means of parameter adjustment with constraint [17], and the expression of solution δx can be written as

$$\delta x = (N_{bb}^{-1} - N_{bb}^{-1} C^T N_{cc}^{-1} C N_{bb}^{-1}) W + N_{bb}^{-1} C^T N_{cc}^{-1} W_x \tag{25}$$

where $N_{bb} = B^T \Sigma B$, $W = B^T \Sigma l$, $N_{cc} = C N_{bb} C^T$, Σ denotes the weight matrix of observations. To simplify the calculation, in this paper we suppose the weight matrix of observations is an identity matrix, namely $\Sigma = I$.

Because it is difficult or even impossible to get the initial value (i.e., approximation) of parameter in advance, the classic Gauss-Newton method (see [18]) is usually employed to solve the parameters iteratively, i.e., we firstly give rough approximation of x, then solve the correction δx by means of parameter adjustment with constraint (using (25)), and give the approximation of x of next iteration as $x + \delta x$, then repeat the above procedure until the δx is less than a given tolerance, or other termination conditions are satisfied.

C. Improved Solution of the Transformation Parameters

Whereas the classic Gauss-Newton method depends strongly on the initial value of parameter, i.e., if the initial values of parameters are poor, the solution will fail because of iterative non-convergence. For the sake, a improved Gauss-Newton method is presented, which uses the k-th iterative solution δx^k of classic Gauss-Newton method, then adds a adaptive variable step-size s^k in the next iteration as follows

$$x^{k+1} = x^k + s^k \delta x^k, \tag{26}$$

which satisfies $V^T(x^{k+1})V(x^{k+1}) < V^T(x^k)V(x^k)$, where $V(x^k)$ is the k-th iterative correction of coordinates. The calculation formula of s^k is as follows [19]

$$\begin{aligned} s^k = 0.5 + 0.25[R(x^k) - R(x^k + \delta x^k)]/ \\ [R(x^k) + R(x^k + \delta x^k) - 2R(x^k + 0.5\delta x^k)], \end{aligned} \tag{27}$$

where $R(x^k)$ is k-th iterative objective function as follows

$$R(x^k) = f^T(x^k) f(x^k) - 2 f^T(x^k) l(x^k), \tag{28}$$

where

$$f(x^k) = \begin{bmatrix} f_1(x^k) \\ \vdots \\ f_{n-1}(x^k) \end{bmatrix}, \quad f_i(x^k) = \Delta a_i - \lambda^k R^k \Delta b_i,$$

$l(x^k) = \begin{bmatrix} l_1(x^k) \\ \vdots \\ l_{n-1}(x^k) \end{bmatrix}$, $l_i(x^k)$ is the k-th iterative l_i, ($i = 1, 2, \cdots, n-1$).

The quaternion-based solution of non-linear 3D coordinate transformation parameters (hereinafter, "quaternion method") is finally summarized as

Step 1. Initiate x, e.g., set the initial value of λ to 1, and the initial values of q_1, q_2, q_3 and q_4 to 0.5 respectively, or set one of them to 1 and the others to 0.

Step 2. Establish observation equation and constraint equation, and solve δx by means of parameter adjustment with constraint (using (25)), if every element of δx is less than given tolerance τ (in this paper, τ is given 1.0×10^{-9}), turn to Step 5.

Step 3. Firstly Compute $R(x^k)$ by using (28), then compute s^k by using (27), lastly compute x^{k+1} by using (26).

Step 4. Calculate $R(x^{k+1})$ by using (28), and if $\left|R(x^{k+1}) - R(x^k)\right| < \varepsilon,$ where ε is a given tolerance, (in the paper, it is set to 1.0×10^{-9}), turn to Step 5, else initiate x with x^{k+1}, continue Step 2.

Step 5. Substitute the solution of x and the coordinates a_0 and b_0 into (18), obtain t, finally output x and t.

Substituting the solution of the unit quaternion Q into (17), we obtain the rotation matrix R. Supposing R is formed by rotating angles α, β, γ counterclockwise around the Cartesian X, Y and Z axes respectively, then R can be expressed by rotation angles as

$$R = \begin{bmatrix} \cos\gamma\cos\beta & \sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha - \cos\gamma\sin\beta\cos\alpha \\ -\sin\gamma\cos\beta & \cos\gamma\cos\alpha - \sin\gamma\sin\beta\sin\alpha & \cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha \\ \sin\beta & -\cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{29}$$

Using (29), the rotation angles α, β, γ can be computed as

$$\alpha = -\tan^{-1}\frac{R_{32}}{R_{33}}, \beta = \sin^{-1}(R_{31}), \gamma = -\tan^{-1}\frac{R_{21}}{R_{11}}. \tag{30}$$

where R_{ij} is the element of R in the i-th row and j-th column.

If we substitute (29) into (19), we obtain the transformation model based on rotation angle. Further we can also establish the observation equation with the rotation angle in terms of linearization, and gain the least squares solution by means of parameter adjustment. Similarly to the 5 steps of quaternion method, finally we can get solution of the seven transformation parameters. This method is the classical one based on rotation angle (hereinafter, "angle algorithm").

IV. SIMULATIVE CASE STUDY AND DISCUSSION

The simulative data and demonstrative process are made as follows. Firstly, the simulative true values of coordinates in system B and transformation parameters are given. Secondly, coordinates in system A (simulative true values) are computed by using (18). Thirdly, the transformation parameters (calculated values) are solved with quaternion method using the above simulative coordinates. Finally, the correctness of the method is proved by comparing the calculated values and simulated values of the parameters and the transformation residuals of coordinates.

Simulated true values of coordinates in system B in system A are listed in Table I. The simulative transformation parameters have three sets, tagged as set 1, set 2, and set 3 corresponding small rotation angle, large rotation angle and super-large rotation angle in Table II. Simulated true values of coordinates in system A are listed Table III.

Supposing that point 1 is the starting point of baselines and then solving the transformation parameters with quaternion method using the simulative coordinates of point 1, 2, 4, 6, 8, (the other four points are reserved for residual calibration of coordinate), we obtain the result as shown in Table IV. To compare quaternion method with angle method, the solution of angle method is also listed in Table IV. The transformation residuals of coordinates in Table IV (using all 9 points) are the differences between the simulative true values and calculated values of coordinates in system A, of which the latter is obtained by substituting coordinates in system B and calculated transformation parameters into (18).

It is clearly seen in Table IV that although the errors and transformation residuals get larger and larger with the increase of rotation angles (from set 1 to set 2 and to set 3), it doesn't change the validation of the quaternion method (all the errors and residuals are too small to be neglectable), and the iteration keep a fast speed (9 to 11 times) regardless of the increase of rotation angles. For small and large angles transformation (set 1 and set 2), rotation angle method is correct, and its solution speed is fast (less than 16 times), but for super large angle transformation (set 3), it can not solve the parameters because of its strong dependence on initial values of parameters.

To validate the superiority of the improved Gauss-Newton method to the classic Gauss-Newton method, the iterative process of them are compared in Fig. 2. The left column is about the relationship of step-size with iteration times, and the right column is about the relationship of objective function with iteration times in Fig. 2. As seen in Fig. 2, for small angle case (set 1), the classic Gauss-Newton method failed after iterative 30 times, showing divergent trend, and with two fluctuations in a few steps of iteration, but the improved Gauss-Newton method converged after iterative 11 times, and with no fluctuations. For large angle case (set 2), the classic Gauss-Newton method succeeded in convergence after iterative 13 times with fluctuations in a few steps of iteration, however the improved Gauss-Newton method converged after iterative 11 times, and with no fluctuations. For super large angle case (set 3), the classic Gauss-Newton method succeeded in convergence after iterative 18 times with fluctuations, however the improved Gauss-Newton method converged after iterative 9 times with no fluctuations. The analysis above indicates that the adaptive step-size strategy is very important, which accelerates the convergence rate and avoids the iterative fluctuation. Thus, the improved Gauss-Newton method is valid compared to the classic Gauss-Newton method.

V. Actual Case Study and Discussion

In order to demonstrate the application of the presented algorithm in the paper and compare it with the famous Procrustes algorithm presented by E. W. Grafarend and J. L. Awange (see [12]), an actual case is investigated in this section. The Cartesian coordinates of seven stations, as listed in Table V, are taken from [12]. Using these coordinates, the transformation parameters are computed with the presented algorithm, as shown in Table VI. In the process, the barycenter of all seven stations is selected as the starting point of baselines to keep consistence with [12]. To compare with the Procrustes algorithm, the results reported in [12] are also listed in Table VI. The rotation angles corresponding to the Procrustes algorithm are not directly obtained from [12] but calculated by the authors according to the computed result of rotation matrix. The residuals are given in Table VII.

TABLE I. Simulative True Values of Coordinates in System B

Point no.	System B (m)		
	x	y	z
1	10.000	30.000	5.000
2	20.000	30.000	12.500
3	30.000	30.000	15.000
4	10.000	20.000	9.500
5	20.000	20.000	11.000
6	30.000	20.000	10.000
7	10.000	10.000	14.500
8	20.000	10.000	4.500
9	30.000	10.000	4.000

TABLE II. Simulative True Values of Transformation Parameters

Set no.	ΔX (m)	ΔY (m)	ΔZ (m)	α	β	γ	λ
Set 1	30	30	10	$47'$	$32'$	$55'$	1.000016
Set 2	30	30	10	27°	21°	24°	1.000016
Set 3	30	30	10	71°	78°	73°	1.000016

TABLE III. Simulative True Values of Coordinates in System A

Point no.	System A (Set 1) (m)			System A (Set 2) (m)			System A (Set 3) (m)		
	x	y	z	X	Y	Z	X	Y	Z
1	40.437	59.903	14.682	53.325	51.360	5.028	52.116	7.239	14.222
2	50.367	59.847	22.275	61.051	51.648	14.850	58.807	9.608	24.512
3	60.343	59.721	24.867	69.312	49.212	20.514	61.443	9.072	34.463
4	40.235	49.967	19.319	47.733	46.333	13.009	49.949	17.746	16.493
5	50.219	49.828	20.911	56.101	43.353	17.841	51.773	16.629	26.376
6	60.227	49.654	20.005	64.737	39.011	20.593	51.570	14.060	36.090
7	40.028	40.038	24.455	42.087	41.578	21.407	48.187	28.543	18.797
8	50.117	39.740	14.549	51.686	32.334	16.672	40.683	20.745	27.902
9	60.120	39.573	14.142	60.269	28.265	19.840	40.886	18.466	37.650

TABLE IV. DIFFERENCES BETWEEN CALCULATED VALUES AND SIMULATIVE TRUE VALUES OF TRANSFORMATION PARAMETERS

Transformation Parameters	**Set 1**		**Set 2**		**Set 3**	
	Quaternion method	***Angle method***	***Quaternion method***	***Angle method***	***Quaternion method***	***Angle method***
ΔX (m)	3.8×10^{-10}	5.2×10^{-6}	4.6×10^{-7}	-5.7×10^{-6}	4.4×10^{-7}	
ΔY (m)	-7.9×10^{-11}	-9.2×10^{-7}	8.3×10^{-8}	-2.2×10^{-5}	1.8×10^{-7}	
ΔZ (m)	-2.7×10^{-10}	-3.3×10^{-6}	7.1×10^{-8}	-6.3×10^{-6}	1.2×10^{-7}	
α (″)	-1.5×10^{-6}	-3.2×10^{-2}	-7.7×10^{-4}	-5.8×10^{-2}	-1.4×10^{-3}	
β (″)	9.4×10^{-7}	-2.7×10^{-2}	5.0×10^{-5}	2.6×10^{-2}	-9.1×10^{-4}	
γ (″)	-2.4×10^{-6}	-3.8×10^{-2}	-1.9×10^{-3}	-6.3×10^{-2}	4.0×10^{-3}	
λ	-3.0×10^{-13}	-1.0×10^{-8}	-1.1×10^{-8}	5.7×10^{-7}	-5.6×10^{-9}	
max. order of magnitude of residuals (m)	10^{-10}	10^{-6}	10^{-7}	10^{-5}	10^{-7}	
Iteration times	11	12	11	16	9	divergence

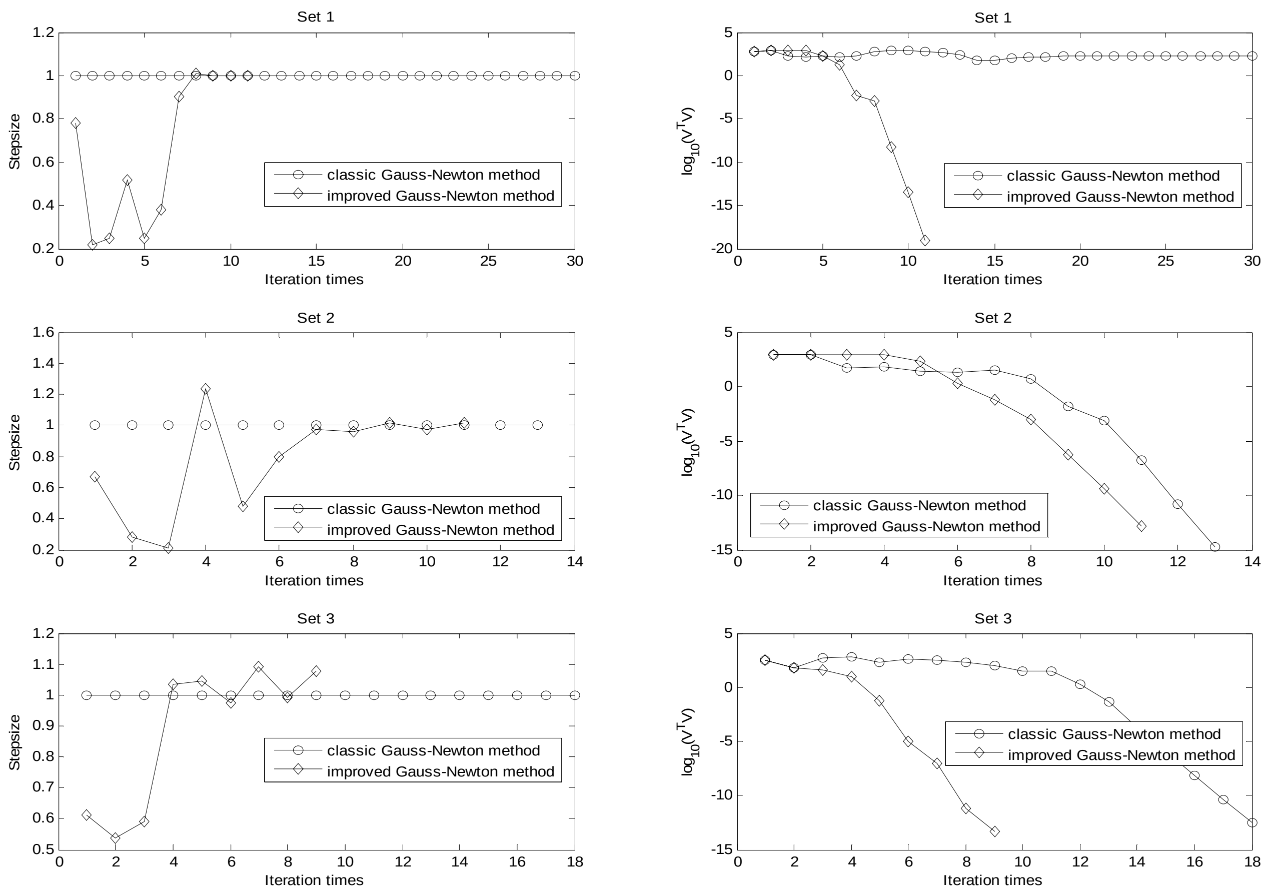


Figure 2. Iterative process comparison of the improved Gauss-Newton method with the classic Gauss-Newton method

TABLE V. CARTESIAN COORDINATES IN SYSTEM B AND A

Station Name	**System B (local system) (m)**			**System A (WGS-84) (m)**		
	x	y	z	X	Y	Z
Solitude	4157222.543	664789.307	4774952.099	4157870.237	664818.678	4775416.524
Buoch Zeil	4149043.336	688836.443	4778632.188	4149691.049	688865.785	4779096.588
Hohenneuffen	4172803.511	690340.078	4758129.701	4173451.354	690369.375	4758594.075
Kuehlenberg	4177148.376	642997.635	4760764.800	4177796.064	643026.700	4761228.899

Ex Mergelaec	4137012.190	671808.029	4791128.215	4137659.549	671837.337	4791592.531
Ex Hof Asperg	4146292.729	666952.887	4783859.856	4146940.228	666982.151	4784324.099
Ex Kaisersbach	4138759.902	702670.738	4785552.196	4139407.506	702700.227	4786016.645

TABLE VI. TRANSFORMATION PARAMETERS RESULTS OF THE PRESENTED ALGORITHM IN THIS PAPER AND I-LESS PROCRUSTES ALGORITHM

	The presented algorithm in this paper			**Procrustes algorithm**		
Rotation matrix						
R	1.000000000	0.000004815	-0.000004333	0.999999999	-0.000004332	0.000004814
	-0.000004815	1.000000000	-0.000004841	-0.000004814	0.999999999	-0.000004841
	0.000004333	0.000004841	1.000000000	0.000004332	0.000004841	0.999999999
Rotation angles (″)						
α		-0.998502748			-0.998527928	
β		0.893691145			0.893539141	
γ		0.993093503			0.992958778	
Translation (m)						
ΔX		641.8804			641.8804	
ΔY		68.6554			68.6553	
ΔZ		416.3982			416.3982	
Scale						
λ		1.000005583			1.000005583	

TABLE VII. TRANSFORMATION RESIDUALS OF THE PRESENTED ALGORITHM IN THIS PAPER AND I-LESS PROCRUSTES ALGORITHM (M)

	The presented algorithm in this paper			**Procrustes algorithm**		
Station Name	X	Y	Z	X	Y	Z
Solitude	0.0940	0.1351	0.1402	0.0940	0.1351	0.1402
Buoch Zeil	0.0588	-0.0497	0.0137	0.0588	-0.0497	0.0137
Hohenneuffen	-0.0399	-0.0879	-0.0081	-0.0399	-0.0879	-0.0081
Kuehlenberg	0.0202	-0.0220	-0.0874	0.0202	-0.0220	-0.0874
Ex Mergelaec	-0.0919	0.0139	-0.0055	-0.0919	0.0139	-0.0055
Ex Hof Asperg	-0.0118	0.0065	-0.0546	-0.0118	0.0065	-0.0546
Ex Kaisersbach	-0.0294	0.0041	0.0017	-0.0294	0.0041	0.0017

VI. CONCLUDING REMARKS

To overcome the drawback that angle method depends strongly on initial value of parameter, especially on rotation angles, which makes the method ineffective in the case of super-large rotation angle due to the beyond estimation in advance, this paper uses quaternion to represent 3D rotation matrix, then presents the quaternion-based iterative method in terms of linearization. The iterative method designs an adaptive step-size based on the classic Gauss-Newton method, which accelerates the convergence rate and avoids the iterative fluctuation. The cases study shows that the method has no dependence on initial value of parameter and satisfactory result with fast speed, and is suitable for coordinate transformation of any rotation angle.

APPENDIX THE ELEMENTS OF B_i AND l_i

$$B_{11} = 2\lambda(-q_4\Delta x_i + q_3\Delta z_i),$$
$$B_{12} = 2\lambda(q_3\Delta y_i + q_4\Delta z_i),$$
$$B_{13} = 2\lambda(-2q_3\Delta x_i + q_2\Delta y_i + q_1\Delta z_i),$$
$$B_{14} = 2\lambda(-2q_4\Delta x_i - q_1\Delta y_i + q_2\Delta z_i),$$
$$B_{21} = 2\lambda(q_4\Delta x_i - q_2\Delta z_i),$$
$$B_{22} = 2\lambda(q_3\Delta x_i - 2q_2\Delta y_i - q_1\Delta z_i),$$
$$B_{23} = 2\lambda(q_2\Delta x_i + q_4\Delta z_i),$$
$$B_{24} = 2\lambda(q_1\Delta x_i - 2q_4\Delta y_i + q_3\Delta z_i),$$
$$B_{31} = 2\lambda(-q_3\Delta x_i + q_2\Delta y_i),$$
$$B_{32} = 2\lambda(q_4\Delta x_i + q_1\Delta y_i - 2q_2\Delta z_i),$$
$$B_{33} = 2\lambda(-q_1\Delta x_i + q_4\Delta y_i - 2q_3\Delta z_i),$$
$$B_{34} = 2\lambda(q_2\Delta x_i + q_3\Delta y_i),$$
$$K_1 = [1-2(q_3^2+q_4^2)]\Delta x_i + 2(q_2q_3 - q_1q_4)\Delta y_i + 2(q_1q_3 + q_2q_4)\Delta z_i,$$
$$K_2 = 2(q_1q_4 + q_2q_3)\Delta x_i + [1-2(q_2^2+q_4^2)]\Delta y_i + 2(q_3q_4 - q_1q_2)\Delta z_i,$$
$$K_3 = 2(q_2q_4 - q_1q_3)\Delta x_i + 2(q_1q_2 + q_3q_4)\Delta y_i + [1-2(q_2^2+q_3^2)]\Delta z_i,$$
$$l_{xi} = \Delta X_i - \lambda K_1,$$
$$l_{yi} = \Delta Y_i - \lambda K_2,$$
$$l_{zi} = \Delta Z_i - \lambda K_3.$$

ACKNOWLEDGMENT

The authors thank Prof. S. X. Huang of School of Geodesy and Geomatics, Wuhan University for his valuable comments and suggestions, which enhanced the quality of this manuscript. The first author is grateful to the support and good working atmosphere provided by his research team in China Three Gorges University.

REFERENCES

[1] A. Leick, *GPS satellite surveying*, 3rd edn. Wiley, Hoboken, 2004.

[2] Y. Z. Shen, L. M. Hu, and B. F. Li, "Ill-posed problem in determination of coordinate transformation parameters with small area's data based on Bursa model," *Acta Geod. et Cartogr. Sinica*, vol.35, no.2, pp.95-98, May 2006 (in Chinese).

[3] T. Soler, and R. A. Snay, "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, 2004.

[4] I. Kashani, "Application of generalized approach to datum transformation between local classical and satellite-based geodetic networks". *Surv. Rev.*, vol.38, no.299, pp.412–422, 2006.

[5] J. Y. Han, H. W. B. Van Gelder, "Step-wise parameter estimations in a time-variant similarity transformation". *J. Surv. Eng.* 132(4):141–148, 2006.

[6] E. M. Mikhail, J. S. Bethel, C. J. McGlone, *Introduction to modern photogrammetry*. Wiley, Chichester, 2001.

[7] J. J. Jaw , and T. Y. Chuang, "Registration of ground-based LIDAR point clouds by means of 3D line features" *J. Chinese Inst. of Eng.*, 31(6), 1031-1045, 2008.

[8] W. X. Zeng and B. Z. Tao, "Non-linear adjustment model of three-dimensional coordinate transformation", *Geomatics and Information Science of Wuhan University*, vol.28, no.5, pp.566-568, Oct. 2003 (in Chinese).

[9] Y. Chen, Y. Z. Shen, D. J. Liu, "A simplified model of three dimensional-datum transformation adapted to big rotation angle", *Geomatics and Information Science of Wuhan University*, vol.29, no.12, pp.1101-1105, Dec. 2004 (in Chinese).

[10] J. L. Yao, B. M. Han, Y. X. Yang, "Applications of Lodrigues Matrix in 3D Coordinate Transformation", *Geomatics and Information Science of Wuhan University*, vol.31, no.12, pp.1094-1096, Dec. 2006 (in Chinese).

[11] H. E. Zeng and S. X. Huang, "A kind of direct search method adopted to solve 3D coordinate transformation parameters.", *Geomatics and Information Science of Wuhan University*, vol.33, no.11, pp.1118-1121, Nov. 2008 (in Chinese).

[12] E. W. Grafarend and J. L. Awange, "Nonlinear analysis of the three-dimensional datum transformation[conformal group C7(3)]". *J. Geod.*, 2003, 77:66-76.

[13] Y. Z. Shen, Y. Chen, D. H. Zheng, "A quaternion-based geodetic datum transformation algorithm", *J. Geod.*, 2006, 80: 233–239.

[14] H. E. Zeng, Q. L. Yi, "A New Analytical Solution of Nonlinear Geodetic Datum Transformation", in the *Proc. of 18th International Conference on Geoinformatics*, Beijing, china, June 18-20, 2010, in press.

[15] J. F. Liu, "Three dimensional rotation represented by quaternion", *College Phys* 23(4):39–43, 2004 (in Chinese).

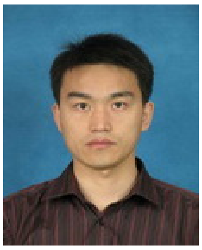
[16] Y. L. Xiao. *Principles of Spacecraft Flight Dynamics*, Beijing: Astronautics Press, 1995 (in Chinese).

[17] Staff Room of Adjustment, Wuhan Technological University of Survey and Mapping, *Adjustment Base*, 3rd ed., Beijing: Surveying and Mapping Press, 1996 (in Chinese).

[18] D. J. Liu, J. N. Huang, "Nonlinear least squares adjustment iterative method", *Science and Technology of Wuhan Technological University of Survey and Mapping*, 1987, no.4, pp,25-31(in Chinese).

[19] X. Z Wang, *Theory and Application of Non-linear Model in Parameter Estimation*. Wuhan: Wuhan University Press, 2002(in Chinese).

[20] F. Zhang, X. B. Cao, J. X. Zou, "A new large-scale transformation algorithm of quaternion to eluer angle", *Journal of Nanjing University of Science and Technology*, vol.26, no.4, pp.376-380, Aug.2002 (in Chinese).



Huaien Zeng was born in Ezhou city of Hubei province, China, in 1979. He obtained a Master of Engineering degree and a Ph.D degree in geodesy and surveying engineering from School of Geodesy and Geomatics, Wuhan University, Wuhan city, China, in 2005 and 2008 respectively.

He is currently a instructor and tutor for master degree gainer in China Three Gorges University, Yichang city, China. He has published sixteen articles, e.g., "a kind of direct search method adopted to solve 3D coordinate transformation parameters.", *Geomatics and Information Science of Wuhan University*, vol.33, no.11, pp.1118-1121, Nov. 2008 (in Chinese). His current research interests include geodesy, photogrammetry and remote sensing, geographical information science (GIS), and their application.

Dr. Zeng likes traveling, playing basketball, table tennis, and Chinese chess.

Qinglin Yi was born in Songzi county, Hubei province, China, in 1966. He obtained a Bachelor of Engineering degree in surveying and mapping engineering from Wuhan Technological University of Survey and Mapping, Wuhan city, China, in 1986.

He is currently a senior engineer and tutor for master degree gainer in China Three Gorges University, Yichang city, China. His current research interests include surveying engineering and disaster prevention and mitigation engineering.

Mr. Yi is a member of educational and professional development committee, China GPS technology application association, and a board director of association of disaster defense, Hubei province, China.