

PERFORMING 3D SIMILARITY TRANSFORMATION WITH LARGE ROTATION ANGLES USING CONSTRAINED MULTIVARIATE TOTAL LEAST SQUARES

Realizando transformação de similaridade 3D com grandes ângulos de rotação usando quadrados mínimos multivariados restritos

Wuyong Tao¹ - ORCID: 0000-0003-0821-644X

Xianghong Hua¹

Shaoquan Feng¹

¹ Wuhan University, School of Geodesy and Geomatics, 430079, Hubei, China.

E-mail: wuyong_tao@whu.edu.cn

E-mail: xhhua@sgg.whu.edu.cn

E-mail: 2017202140047@whu.edu.cn

Received in 20th March 2019

Accepted in 8th November 2020

Abstract:

3D similarity transformation is frequently encountered operation in the field of geodetic data processing, and there are many applications that involve large rotation angles. In previous studies, the errors of the coefficient matrix were usually neglected and a least squares algorithm was applied to calculate the transformation parameters. However, the coefficient matrix is composed of the point coordinates in source coordinate system, i.e., the coefficient matrix is also contaminated by errors. Therefore, a total least squares algorithm should be applied. In this paper, a new method is proposed to address the 3D similarity transformation problem with large rotation angles. Firstly, the scale factor and rotation matrix are put together as the parameter matrix to avoid the rank-defect problem. Then, the translation vector is removed and the multivariate model is constructed. Finally, the constraints are introduced according to the properties of the parameter matrix and the constrained multivariate total least squares algorithm is derived to obtain the transformation parameters. The experimental results show that the proposed method has a high computational efficiency.

Keywords: 3D similarity transformation; constraint; large rotation angle; multivariate total least squares

Resumo:

A transformação da similaridade 3D é frequentemente encontrada no campo do processamento geodésico de dados, e há muitas aplicações que envolvem grandes ângulos de rotação. Em estudos anteriores, os erros da matriz de coeficientes foram geralmente negligenciados e um algoritmo de menos quadrados foi aplicado para calcular os parâmetros de transformação. No entanto, a matriz de coeficiente é composta pelas coordenadas de ponto no

How to cite this article: TAO, W.; HUA, X.; FENG, S. Performing 3D similarity transformation with large rotation angles using constrained multivariate total least squares. *Bulletin of Geodetic Sciences*. 26(4): e2020021, 2020.



This content is licensed under a Creative Commons Attribution 4.0 International License.

sistema de coordenadas de origem, ou seja, a matriz de coeficiente também está contaminada por erros. Portanto, um algoritmo total de quadrados mínimos deve ser aplicado. Neste artigo, propõe-se um novo método para abordar o problema de transformação da similaridade 3D com grandes ângulos de rotação. Em primeiro lugar, o fator de escala e a matriz de rotação são montados como a matriz de parâmetros para evitar o problema de defeito de classificação. Em seguida, o vetor de tradução é removido e o modelo multivariado é construído. Finalmente, as restrições são introduzidas de acordo com as propriedades da matriz de parâmetros e o algoritmo multivariado multivariado restrito de menos quadrados é derivado para obter os parâmetros de transformação. Os resultados experimentais mostram que o método proposto tem alta eficiência computacional.

Palavras-chave: transformação de similaridade 3D; restrição; grande ângulo de rotação; multivariar total menos quadrados

1. Introduction

The 3D coordinate transformation is a fundamental problem in the field of geodesy, photogrammetry and map projection etc. The 3D coordinate transformation model with seven parameters (i.e., scale factor, three translation parameters and three rotation angles), also known as similarity transformation, is the most frequently employed. When the rotation angles are small, the Bursa-Wolf model, Molodensky-Badekas model, Veis model and Vaniček-Wells model can be used in the 3D similarity transformation problem (Rapp 1993). Yang (1999) employed the robust least squares in geodetic datum transformation so as to resist the influence of gross errors. You and Hwang (2006) utilized the least squares collocation in geodetic datum transformation so as to compensate the systematic distortion. However, the 3D similarity transformation with large rotation angles is among the frequently encountered operation in a practical project, e.g., the transformation between physical coordinate system and image coordinate system in photogrammetry. Therefore, it is necessary to develop a solution for the 3D similarity transformation with large rotation angles. In Chen et al. (2004), the nine elements of the rotation matrix were used to replace the three rotation angles. The coordinate transformation model with thirteen parameters was established, which can be used to deal with the 3D similarity transformation with large rotation angles, and the constraints were introduced because the rotation matrix is orthogonal. Then, the constrained least squares (CLS) algorithm was applied to estimate the transformation parameters. When both the coefficient matrix and the observation vector are affected by random errors, the total least squares (TLS) algorithm with errors-in-variables (EIV) model should be employed, which was termed by Golub and van Loan (1980). Numerous scholars have paid attention to TLS algorithm. De Moor (1993) considered the structural characteristics of the coefficient matrix, the structured TLS (STLS) algorithm was studied. Yan and Fan (2001) proposed the mixed LS-TLS method, in which the constant columns were separated from the coefficient matrix. However, the STLS algorithm and mixed LS-TLS algorithm are not satisfactory in measurement data processing. For this reason, Xu et al. (2012) extracted the random elements from the coefficient matrix, the weighted total least squares (WTLS) algorithm with partial EIV (PEIV) model was developed. The quantities that need to be estimated were reduced so that the computational efficiency was improved. Subsequently, Shi et al. (2015) improved the WTLS algorithm with PEIV model, the computational efficiency was further improved. In Schaffrin and Felus (2008), the observation vector and parameter vector were written in matrix form and developed the multivariate total least squares (MTLS) algorithm, but for the case of identity weight matrix. Fang (2011) formulated the weighted MTLS (WMTLS) algorithm, and the weight matrix was universal. Zeng et al. (2015) derived the WTLS algorithm with inequality constraints. Zhao and Gui (2017) studied the outlier detection for PEIV model. Because different observation types usually lead to a model with heterogenous observations, variance component estimation for the EIV model was also derived (Xu et al. 2014; Wang and Xu 2016).

The TLS algorithm has been widely applied in the 3D coordinate transformation as well (Fang 2014; Fang 2015). Lu et al. (2014) took the gross errors in observed coordinates into consideration, and constructed the robust

weighted total least squares (RWTLS) algorithm. In Laari et al. (2016) and Okwuashi and Eyoh (2012), the TLS solution based on singular value decomposition was applied in the Bursa-Wolf and Molodensky-Badekas model. Because the aim of the 3D coordinate transformation is to obtain the coordinates of non-common points in the target coordinate system, Li et al. (2012) formulated the total least squares collocation to solve the 3D coordinate transformation problem. Thus, the errors of common points in both the source and target coordinate system could be taken into consideration. Furthermore, both the errors of non-common points in the source coordinate system and the correlations between the coordinates of common and non-common points were also taken into consideration. Both the computation of the transformation parameters and the transformation of the coordinates of non-common points were integratively implemented. In Lu et al. (2015), the weighted total least squares (WTLS) algorithm based on the nonlinear Gauss-Helmert model (Neitzel 2010) was applied in the 3D similarity transformation with large rotation angles. However, the method of Lu et al. (2015) has potential numerical instability because of the existing of the rank-defect problem (Yang et al. 2017). For this reason, Chang (2016) derived the closed-form least-squares solution, but the method would only be applied in the case with rotational invariant covariance structure. Mercan et al. (2018) provided an efficient solution to 3D similarity transformation, which is based on quaternions. In Amiri-Simkooei (2018), the least-squares variance component estimation (LS-VCE) was introduced into EIV model for handling the 3D similarity transformation with hard constraints.

In this paper, the nine elements in the rotation matrix are treated as the unknown parameters. Thus, the complex formula derivation and tedious computation of trigonometric functions are avoided. The scale factor and rotation matrix are put together as the parameter matrix to avoid the rank-defect problem. Then, the translation parameters are eliminated and the multivariate model is constructed. Although the multivariate model had been constructed for the 3D coordinate transformation by Felus and Burtch (2009), the orthogonal Procrustes algorithm was applied so that their method is only suitable for the condition where the errors of observed coordinates are symmetrical and independent. Here, we develop the constraints according to the properties of the parameter matrix, and the constrained MTLs (CMTLS) algorithm is derived to determine the parameter matrix. Because we eliminate the translation parameters, the constants that result in a large number of computation will not appear in the coefficient matrix. Furthermore, we do not need to obtain the Jacobian matrix whose calculation is time-consuming for calculating the cofactor matrix of the coefficient matrix. Therefore, the proposed algorithm is computationally cheap. The main contributions of our work are summarized as follows:

The random errors of both the coefficient matrix and observation matrix are taken into consideration. Thus, the more reliable posteriori standard deviation can be obtained.

The translation vector is eliminated and calculated separately. The multivariate model is constructed to estimate the parameters. As a consequence, we avoid calculating the Jacobian matrix, and the coefficient matrix will not contain a large amount of constants. This reduce the computation time of the algorithm.

2. 3D Similarity Transformation Model with Large Rotation Angles

In the 3D similarity transformation model with large rotation angles, the nine elements in the rotation matrix are treated as the unknown parameters in addition to the three translation and one scale parameters. The nine elements need to be estimated, rather than the three rotation angles (Lu et al. 2015). Let (X, Y, Z) denote the coordinates in the target system and (x, y, z) denote the coordinates of the corresponding points in the source system. The mathematical model of the transformation from the source coordinate system to the target coordinate system reads

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = k\mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \quad (1)$$

where k is the scale factor, X_0, Y_0, Z_0 are the three translation parameters, and \mathbf{R} is the rotation matrix.

The rotation matrix is the product of three individual rotation matrices as follow

$$\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix} \quad (2)$$

If one wants to estimate the three rotation angles, the rotation matrix needs to be linearized, i.e., seeking the partial derivative of the rotation matrix with respect to the three rotation angles. The formula is rather complicated, and the tedious computation of the trigonometric functions will also be introduced. Therefore, Chen et al. (2004) directly estimated the nine elements in the rotation matrix. Then, to preserve the orthogonality of the rotation matrix, the following constraints should be applied

$$\begin{bmatrix} r_1^2 + r_2^2 + r_3^2 \\ r_4^2 + r_5^2 + r_6^2 \\ r_7^2 + r_8^2 + r_9^2 \\ r_1r_4 + r_2r_5 + r_3r_6 \\ r_1r_7 + r_2r_8 + r_3r_9 \\ r_4r_7 + r_5r_8 + r_6r_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

3. The Constrained Multivariate Total Least Squares Algorithm

Transposing equation (1), we can obtain

$$\begin{bmatrix} X & Y & Z \end{bmatrix} = k \begin{bmatrix} x & y & z \end{bmatrix} \mathbf{R}^T + \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix} \quad (4)$$

For n number of common points, we have

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} = k \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \mathbf{R}^T + \mathbf{I}_n \otimes \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix} \quad (5)$$

where \mathbf{I}_n denotes the $n \times 1$ vector of ones and \otimes denotes the Kronecker product operator.

Designing a matrix \mathbf{G} so that $\mathbf{G}(\mathbf{I}_n \otimes \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}) = \mathbf{0}$, thus we can eliminate the translation vector. Through deduction, we have

$$\begin{aligned} \mathbf{G}(\mathbf{I}_n \otimes \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}) &= (\mathbf{G} \otimes \mathbf{1})(\mathbf{I}_n \otimes \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}) \\ &= (\mathbf{G} \cdot \mathbf{I}_n \otimes \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}) \end{aligned} \quad (6)$$

We need to make $\mathbf{G} \cdot \mathbf{I}_n = \mathbf{0}$, so we can obtain

$$\mathbf{G} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (7)$$

where \mathbf{G} is an $(n-1) \times n$ matrix. Then, multiplying equation (5) by matrix \mathbf{G} , we obtain

$$\mathbf{G} \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} = k\mathbf{G} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \mathbf{R}^T \quad (8)$$

$$\text{Let } \mathbf{T} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, \mathbf{S} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}. \text{ Equation (8) becomes}$$

$$\mathbf{GT} = \mathbf{GS}k\mathbf{R}^T \quad (9)$$

The rotation matrix and scale factor are put together as the parameter matrix. That is $\xi = k\mathbf{R}^T$. The aim is to avoid the rank-defect problem. The reason for the rank-defect problem was present in Yang et al. (2017). Let the coefficient matrix $\mathbf{A} = \mathbf{GS}$ and the observation matrix $\mathbf{Y} = \mathbf{GT}$. Then, equation (9) can be rewritten in the following form

$$\mathbf{Y} = \mathbf{A}\xi \quad (10)$$

According to the Kronecker product and the property of the vectorized operator, we obtain

$$\text{vec}(\mathbf{A}) = \text{vec}(\mathbf{GSI}_3) = (\mathbf{I}_3 \otimes \mathbf{G})\text{vec}(\mathbf{S}) \quad (11)$$

$$\text{vec}(\mathbf{Y}) = \text{vec}(\mathbf{GTI}_3) = (\mathbf{I}_3 \otimes \mathbf{G})\text{vec}(\mathbf{T}) \quad (12)$$

where $\text{vec}(\cdot)$ is the mathematical operator to convert a matrix to a column vector by stacking one column of this matrix underneath the previous one, and \mathbf{I}_3 denotes a 3×3 identity matrix. According to the cofactor propagation law, the cofactor matrices of the coefficient matrix and observation matrix are

$$\mathbf{Q}_A = (\mathbf{I}_3 \otimes \mathbf{G})\mathbf{Q}_S(\mathbf{I}_3 \otimes \mathbf{G}^T) \quad (13)$$

$$\mathbf{Q}_Y = (\mathbf{I}_3 \otimes \mathbf{G})\mathbf{Q}_T(\mathbf{I}_3 \otimes \mathbf{G}^T) \quad (14)$$

where \mathbf{Q}_T is the cofactor matrix of the coordinates in the target system, i.e., the cofactor matrix of \mathbf{T} . \mathbf{Q}_S is the cofactor matrix of the coordinates in the source system, i.e., the cofactor matrix of \mathbf{S} .

Because the rotation matrix is an orthogonal matrix, we can derive

$$\xi\xi^T = k^2\mathbf{R}^T\mathbf{R} = k^2\mathbf{I}_3 \quad (15)$$

with

$$\xi = \begin{bmatrix} \xi_1 & \xi_4 & \xi_7 \\ \xi_2 & \xi_5 & \xi_8 \\ \xi_3 & \xi_6 & \xi_9 \end{bmatrix} \quad (16)$$

Then, according to equation (15), we can write

$$\begin{bmatrix} \xi_1^2 + \xi_4^2 + \xi_7^2 & \xi_1\xi_2 + \xi_4\xi_5 + \xi_7\xi_8 & \xi_1\xi_3 + \xi_4\xi_6 + \xi_7\xi_9 \\ \xi_1\xi_2 + \xi_4\xi_5 + \xi_7\xi_8 & \xi_2^2 + \xi_5^2 + \xi_8^2 & \xi_2\xi_3 + \xi_5\xi_6 + \xi_8\xi_9 \\ \xi_1\xi_3 + \xi_4\xi_6 + \xi_7\xi_9 & \xi_2\xi_3 + \xi_5\xi_6 + \xi_8\xi_9 & \xi_3^2 + \xi_6^2 + \xi_9^2 \end{bmatrix} = \begin{bmatrix} k^2 & 0 & 0 \\ 0 & k^2 & 0 \\ 0 & 0 & k^2 \end{bmatrix} \quad (17)$$

As we can see, the diagonal elements of $\xi\xi^T$ have the same value and the off-diagonal elements are equal to 0. Thus, the new constraints for the reduced observation equations are as follows

$$\begin{bmatrix} \xi_1^2 + \xi_4^2 + \xi_7^2 - \xi_2^2 - \xi_5^2 - \xi_8^2 \\ \xi_1^2 + \xi_4^2 + \xi_7^2 - \xi_3^2 - \xi_6^2 - \xi_9^2 \\ \xi_1\xi_2 + \xi_4\xi_5 + \xi_7\xi_8 \\ \xi_1\xi_3 + \xi_4\xi_6 + \xi_7\xi_9 \\ \xi_2\xi_3 + \xi_5\xi_6 + \xi_8\xi_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (18)$$

In order to adjust the nonlinear model, we need to linearize equations (10) and (18). Whilst the random errors in the coefficient matrix and observation matrix are taken into consideration

$$Y - E_Y = (A - E_A)\xi_0 + (A - E_A)\delta\xi \quad (19)$$

$$C\text{vec}(\delta\xi) + W_0 = 0 \quad (20)$$

with

$$C = \begin{bmatrix} 2\xi_1^0 & -2\xi_2^0 & 0 & 2\xi_4^0 & -2\xi_5^0 & 0 & 2\xi_7^0 & -2\xi_8^0 & 0 \\ 2\xi_1^0 & 0 & -2\xi_3^0 & 2\xi_4^0 & 0 & -2\xi_6^0 & 2\xi_7^0 & 0 & -2\xi_9^0 \\ \xi_2^0 & \xi_1^0 & 0 & \xi_5^0 & \xi_4^0 & 0 & \xi_8^0 & \xi_7^0 & 0 \\ \xi_3^0 & 0 & \xi_1^0 & \xi_6^0 & 0 & \xi_4^0 & \xi_9^0 & 0 & \xi_7^0 \\ 0 & \xi_3^0 & \xi_2^0 & 0 & \xi_6^0 & \xi_5^0 & 0 & \xi_9^0 & \xi_8^0 \end{bmatrix} \quad (21)$$

$$W_0 = \begin{bmatrix} (\xi_1^0)^2 + (\xi_4^0)^2 + (\xi_7^0)^2 - (\xi_2^0)^2 - (\xi_5^0)^2 - (\xi_8^0)^2 \\ (\xi_1^0)^2 + (\xi_4^0)^2 + (\xi_7^0)^2 - (\xi_3^0)^2 - (\xi_6^0)^2 - (\xi_9^0)^2 \\ \xi_1^0\xi_2^0 + \xi_4^0\xi_5^0 + \xi_7^0\xi_8^0 \\ \xi_1^0\xi_3^0 + \xi_4^0\xi_6^0 + \xi_7^0\xi_9^0 \\ \xi_2^0\xi_3^0 + \xi_5^0\xi_6^0 + \xi_8^0\xi_9^0 \end{bmatrix} \quad (22)$$

where $\xi_i^0 (i = 1, 2, \dots, 9)$ are the elements of ξ_0 , which is the initial value of the parameter matrix. E_A is the random error matrix of the coefficient matrix, and E_Y is the random error matrix of the observation matrix.

Although $E_A\delta\xi$ is the second-order smaller quantity, the iterative computation could diverge or converge to a wrong stationary point if it is neglected (Pope 1972). However, this pitfall can be avoided by using the estimate $E_{A(j)}$ of the j th iteration instead of E_A in equation (19), as suggested by Li et al. (2012). Noting $A_{(j)} = A - E_{A(j)}$, we obtain

$$\begin{cases} Y - E_Y = A\xi_0 - E_{A(j)}\xi_0 + A_{(j)}\delta\xi \\ C\text{vec}(\delta\xi) + W_0 = 0 \end{cases} \quad (23)$$

Next, the CMTLS algorithm will be derived. We start with the objective function

$$e_A^T Q_A^{-1} e_A + e_Y^T Q_Y^{-1} e_Y = \min \quad (24)$$

where $e_A = \text{vec}(E_A)$ whose cofactor matrix is Q_A , and $e_Y = \text{vec}(E_Y)$ whose cofactor matrix is Q_Y . The corresponding Lagrange target function reads

$$\begin{aligned} \Phi(\mathbf{e}_A, \mathbf{e}_Y, \delta\hat{\boldsymbol{\xi}}, \boldsymbol{\lambda}, \mathbf{u}) &= \mathbf{e}_A^T \mathbf{Q}_A^{-1} \mathbf{e}_A + \mathbf{e}_Y^T \mathbf{Q}_Y^{-1} \mathbf{e}_Y + \\ & 2\text{vec}^T(\boldsymbol{\lambda})\text{vec}(\mathbf{Y} - \mathbf{E}_Y - \mathbf{A}\boldsymbol{\xi}_0 + \mathbf{E}_A\boldsymbol{\xi}_0 - \mathbf{A}_{(j)}\delta\hat{\boldsymbol{\xi}}) + 2\mathbf{u}^T (\mathbf{C}\text{vec}(\delta\hat{\boldsymbol{\xi}}) + \mathbf{W}_0) \end{aligned} \quad (25)$$

where $\boldsymbol{\lambda}$ denotes a matrix of Lagrange multipliers, \mathbf{u} denotes a vector of Lagrange multipliers.

The solution of this target function is derived via the Lagrange necessary condition, namely

$$\left. \frac{1}{2} \frac{\partial \Phi}{\partial \mathbf{e}_A} \right|_{\hat{\mathbf{e}}_A, \hat{\mathbf{e}}_Y, \delta\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{u}}} = \mathbf{Q}_A^{-1} \hat{\mathbf{e}}_A + (\boldsymbol{\xi}_0 \otimes \mathbf{I}_{n-1}) \text{vec}(\hat{\boldsymbol{\lambda}}) = 0 \quad (26)$$

$$\left. \frac{1}{2} \frac{\partial \Phi}{\partial \mathbf{e}_Y} \right|_{\hat{\mathbf{e}}_A, \hat{\mathbf{e}}_Y, \delta\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{u}}} = \mathbf{Q}_Y^{-1} \hat{\mathbf{e}}_Y - \text{vec}(\hat{\boldsymbol{\lambda}}) = 0 \quad (27)$$

$$\left. \frac{1}{2} \frac{\partial \Phi}{\partial (\text{vec}(\delta\hat{\boldsymbol{\xi}}))} \right|_{\hat{\mathbf{e}}_A, \hat{\mathbf{e}}_Y, \delta\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{u}}} = -(\mathbf{I}_3 \otimes \mathbf{A}_{(j)}^T) \text{vec}(\hat{\boldsymbol{\lambda}}) + \mathbf{C}^T \hat{\mathbf{u}} = 0 \quad (28)$$

$$\left. \frac{1}{2} \frac{\partial \Phi}{\partial (\text{vec}(\boldsymbol{\lambda}))} \right|_{\hat{\mathbf{e}}_A, \hat{\mathbf{e}}_Y, \delta\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{u}}} = \text{vec}(\mathbf{Y} - \mathbf{A}\boldsymbol{\xi}_0) - \hat{\mathbf{e}}_Y + (\boldsymbol{\xi}_0^T \otimes \mathbf{I}_{n-1}) \hat{\mathbf{e}}_A - (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}) \text{vec}(\delta\hat{\boldsymbol{\xi}}) = 0 \quad (29)$$

$$\left. \frac{1}{2} \frac{\partial \Phi}{\partial \mathbf{u}} \right|_{\hat{\mathbf{e}}_A, \hat{\mathbf{e}}_Y, \delta\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{u}}} = \mathbf{C}\text{vec}(\delta\hat{\boldsymbol{\xi}}) + \mathbf{W}_0 = 0 \quad (30)$$

where hats indicate “estimated” quantities.

From equations (26) and (27), we can obtain

$$\hat{\mathbf{e}}_A = -\mathbf{Q}_A (\boldsymbol{\xi}_0 \otimes \mathbf{I}_{n-1}) \text{vec}(\hat{\boldsymbol{\lambda}}) \quad (31)$$

$$\hat{\mathbf{e}}_Y = \mathbf{Q}_Y \text{vec}(\hat{\boldsymbol{\lambda}}) \quad (32)$$

Inserting equations (31) and (32) into equation (29), we derive

$$\begin{aligned} \text{vec}(\hat{\boldsymbol{\lambda}}) &= (\mathbf{Q}_Y + (\boldsymbol{\xi}_0^T \otimes \mathbf{I}_{n-1}) \mathbf{Q}_A (\boldsymbol{\xi}_0 \otimes \mathbf{I}_{n-1}))^{-1} (\text{vec}(\mathbf{Y} - \mathbf{A}\boldsymbol{\xi}_0) - (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}) \text{vec}(\delta\hat{\boldsymbol{\xi}})) \\ &= \mathbf{Q}_y^{-1} (\text{vec}(\mathbf{Y} - \mathbf{A}\boldsymbol{\xi}_0) - (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}) \text{vec}(\delta\hat{\boldsymbol{\xi}})) \end{aligned} \quad (33)$$

where $\mathbf{Q}_y = \mathbf{Q}_Y + (\boldsymbol{\xi}_0^T \otimes \mathbf{I}_{n-1}) \mathbf{Q}_A (\boldsymbol{\xi}_0 \otimes \mathbf{I}_{n-1})$.

Inserting equation (33) into equation (28), we derive

$$(\mathbf{I}_3 \otimes \mathbf{A}_{(j)}^T) \mathbf{Q}_y^{-1} (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}) \text{vec}(\delta\hat{\boldsymbol{\xi}}) + \mathbf{C}^T \hat{\mathbf{u}} = (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}^T) \mathbf{Q}_y^{-1} \text{vec}(\mathbf{Y} - \mathbf{A}\boldsymbol{\xi}_0) \quad (34)$$

Combining equation (34) and equation (30), we can obtain

$$\begin{bmatrix} \mathbf{N} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \text{vec}(\delta\hat{\boldsymbol{\xi}}) \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{W} \\ -\mathbf{W}_0 \end{bmatrix} \quad (35)$$

where $\mathbf{N} = (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}^T) \mathbf{Q}_y^{-1} (\mathbf{I}_3 \otimes \mathbf{A}_{(j)})$, $\mathbf{W} = (\mathbf{I}_3 \otimes \mathbf{A}_{(j)}^T) \mathbf{Q}_y^{-1} \text{vec}(\mathbf{Y} - \mathbf{A}\boldsymbol{\xi}_0)$.

Thus, we get the CMTLS solution

$$\text{vec}(\delta\hat{\boldsymbol{\xi}}) = \mathbf{N}^{-1} (\mathbf{W} + \mathbf{C}^T (\mathbf{C}\mathbf{N}^{-1}\mathbf{C}^T)^{-1} (-\mathbf{W}_0 - \mathbf{C}\mathbf{N}^{-1}\mathbf{W})) \quad (36)$$

Then, the updated parameter matrix is calculated as

$$\boldsymbol{\xi} = \boldsymbol{\xi}_0 + \delta\hat{\boldsymbol{\xi}} \quad (37)$$

After the parameter matrix is calculated, we can take the value of any diagonal elements of the matrix $\xi\xi^T$ to calculate the scale factor. For example

$$\hat{k} = \sqrt{\xi\xi^T(1,1)} \quad (38)$$

where $\xi\xi^T(1,1)$ denotes the first diagonal element of $\xi\xi^T$.

Thus, the rotation matrix is given by

$$\hat{R} = \frac{\xi^T}{\hat{k}} \quad (39)$$

Because the observation matrix Y is an $(n-1) \times 3$ matrix and the parameter matrix ξ is a 3×3 matrix, according to Schaffrin and Felus (2008) and Zeng et al. (2015), the standard deviation is estimated as

$$\hat{\sigma} = \sqrt{\frac{e_A^T Q_A^{-1} e_A + e_Y^T Q_Y^{-1} e_Y}{3(n-1) - 3 \cdot 3 + 5}} = \sqrt{\frac{vec(Y - A\xi)^T Q_Y^{-1} vec(Y - A\xi)}{3n - 7}} \quad (40)$$

In addition, we can calculate the estimated random error vectors of S and T with

$$\hat{e}_S = vec(\hat{E}_S) = Q_S (\hat{E}_S \otimes G^T) Q_A^{-1} \hat{e}_A \quad (41)$$

$$\hat{e}_T = vec(\hat{E}_T) = Q_T (I_3 \otimes G^T) Q_Y^{-1} \hat{e}_Y \quad (42)$$

where \hat{E}_S is the estimated random error matrix of S and \hat{E}_T is the estimated random error matrix of T .

After the scale factor is computed by using equation (38) and the rotation matrix is computed by using equation (39), the adjusted coordinates are used to calculate the translation vector of each point as follows

$$A = (T - \hat{E}_T) - \hat{k}(S - \hat{E}_S)\hat{R}^T \quad (43)$$

Here, we observe that the calculated translation vectors of all the points are the same, so any row Δ_i of Δ can be considered as the translation vector.

The calculation step is summarized as follows:

Step 1: The prepared data include the coordinates in the source system, the coordinates in the target system, the cofactor matrix Q_S of the coordinates in the source system, and the cofactor matrix Q_T of the coordinates in the target system.

Step 2: Compute the initial value of the parameter matrix ξ_0 and set the initial random error matrix E_{A_0} of the coefficient matrix as zero matrix.

$$vec(\xi_{(0)}) = ((I_3 \otimes A^T) Q_Y^{-1} (I_3 \otimes A))^{-1} (I_3 \otimes A^T) Q_Y^{-1} vec(Y)$$

$$E_{A_{(0)}} = \mathbf{0}_{(n-1) \times 3}$$

Step 3: Using the initial value, we start the following iterative process:

$$Q_y = Q_Y + (\xi_{(j)}^T \otimes I_{n-1}) Q_A (\xi_{(j)} \otimes I_{n-1})$$

$$A_{(j)} = A - E_{A_{(j)}}$$

$$N = (I_3 \otimes A_{(j)}^T) Q_y^{-1} (I_3 \otimes A_{(j)})$$

$$W = (I_3 \otimes A_{(j)}^T) Q_y^{-1} vec(Y - A\xi_0)$$

$$vec(\delta\hat{\xi}_{(j+1)}) = N^{-1} (W + C^T (CN^{-1}C^T)^{-1} (-W_0 - CN^{-1}W))$$

$$\xi_{(j+1)} = \xi_{(j)} + \delta\hat{\xi}_{(j+1)}$$

$$\hat{e}_{A_{(j+1)}} = -Q_A (\xi_{(j)} \otimes I_{n-1}) Q_y^{-1} (vec(Y - A\xi_{(j)}) - (I_3 \otimes A_{(j)}) vec(\delta\hat{\xi}_{(j+1)}))$$

Step 4: Repeat the step (3) until $\|\delta\hat{\xi}_{(j+1)}\| \leq \varepsilon$ (ε is a preset threshold).

Step 5: Compute the scale factor and rotation matrix by equations (38) and (39), and compute the translation vector using equation (43).

Step 6: Compute the standard deviation using equation (40).

4. Experiments and Analysis

4.1 Three-dimensional coordinate transformation

In this section, the simulation experiment is performed to illustrate the advantages of the derived CMTLS algorithm. Matlab is used to perform the experiment. We randomly choose ten points as the points in the source system. The true values of the scale factor, translation parameters and rotation angles are set as

$$[k \ X_0 \ Y_0 \ Z_0 \ \alpha_1 \ \alpha_2 \ \alpha_3] = [1.01 \ 6 \ 7 \ 8 \ 30^\circ \ 45^\circ \ 60^\circ] \quad (44)$$

Thus, we can calculate the coordinates of the ten points in the target system. The true values of the coordinates in the two coordinate systems are shown in Figure 1. The blue points are the points in the source coordinate system and the red points are the points in the target coordinate system.

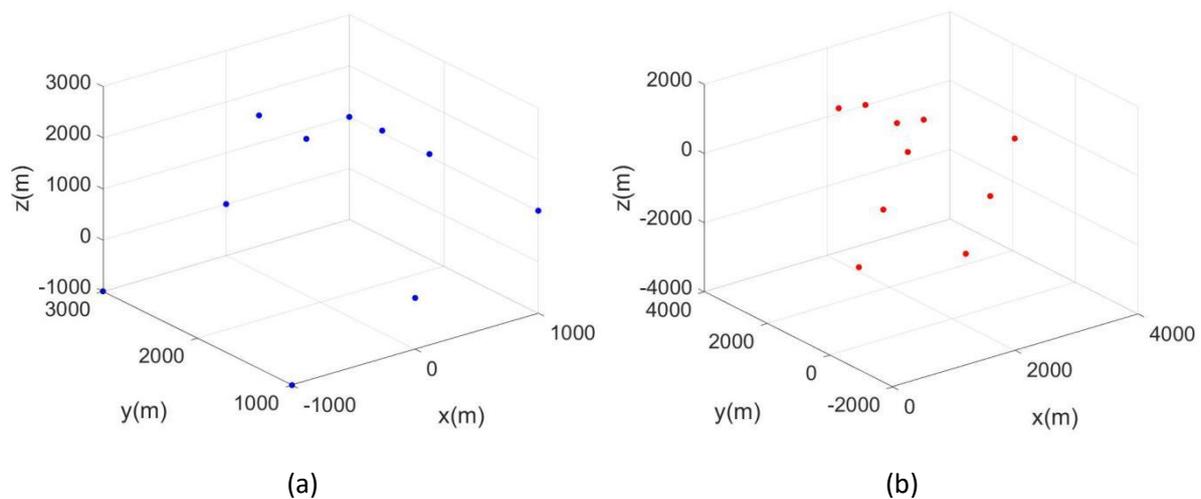


Figure 1: The ten points in the source coordinate system (a) and target coordinate system (b).

The random errors with different variances are added to the true values of the coordinates. The coordinates of point 1 to point 5 in the target system contain the random errors with zero mean and standard deviation of 0.03m. The coordinates of point 6 to point 10 in the target system contain the random errors with zero mean and standard deviation of 0.06m. The coordinates of point 1 to point 5 in the source system contain the random errors with zero mean and standard deviation of 0.09m. The coordinates of point 6 to point 10 in the source system contain the random errors with zero mean and standard deviation of 0.12m.

At first, the WMTLS (Fang 2011) and CMTLS are implemented to obtain the parameter matrix. The priori standard deviation is set as 0.03 to determine the cofactor matrices \mathbf{Q}_S and \mathbf{Q}_T . The matrix $\hat{\xi}\hat{\xi}^T$ is calculated

according to the calculated parameter matrix, as shown in Table 1.

Table 1: The calculated results of the two methods

Method	WMTLS	CMTLS
$\xi\xi^T$	$\begin{bmatrix} 1.0201 & -0.0001 & 0.0000 \\ -0.0001 & 1.0202 & -0.0001 \\ 0.0000 & -0.0001 & 1.0201 \end{bmatrix}$	$\begin{bmatrix} 1.0201 & 0.0000 & 0.0000 \\ 0.0000 & 1.0201 & 0.0000 \\ 0.0000 & 0.0000 & 1.0201 \end{bmatrix}$

Because the rotation matrix is an orthogonal matrix, the diagonal elements of the matrix $\xi\xi^T$ should be equal and the off-diagonal elements should be equal to 0. From the calculated results of Table 1, we note that the results of the CMTLS are consistent with the constraints. However, in the results calculated by the WMTLS, the diagonal elements are not equal, and non-zero values exist in the off-diagonal elements. Therefore, the results of the CMTLS are better and the additional constraints are necessary in this context.

Next, the CWLS, CWTLS and CMTLS are implemented. The calculated results of the three algorithms are compared to perform analysis. In order to have an insight to the three algorithms, the CWLS and CWTLS for the 3D similarity transformation are presented in Appendix. We carried out 1000 simulation experiments and randomly extract the calculated results of one simulation experiment. The results of the CWLS are as follows:

$$\hat{k} = 1.00997767, \hat{\Delta} = [6.05115997 \quad 6.94359596 \quad 8.04445603],$$

$$\hat{R} = \begin{bmatrix} 0.35358869 & 0.92676295 & 0.12682848 \\ -0.61234939 & 0.12683902 & 0.78034613 \\ 0.70710908 & -0.35358491 & 0.61235157 \end{bmatrix}$$

The results of the CWTLS are as follows:

$$\hat{k} = 1.00997271, \hat{\Delta} = [6.03809733 \quad 6.96173005 \quad 8.02453490],$$

$$\hat{R} = \begin{bmatrix} 0.35357755 & 0.92676721 & 0.12682840 \\ -0.61235068 & 0.12683161 & 0.78034632 \\ 0.70711353 & -0.35357640 & 0.61235134 \end{bmatrix}$$

The results of the CMTLS are as follows:

$$\hat{k} = 1.00997271, \hat{\Delta} = [6.03809733 \quad 6.96173005 \quad 8.02453490],$$

$$\hat{R} = \begin{bmatrix} 0.35357755 & 0.92676721 & 0.12682840 \\ -0.61235068 & 0.12683161 & 0.78034632 \\ 0.70711353 & -0.35357640 & 0.61235134 \end{bmatrix}$$

As we can see, the results of the CWTLS are the same as those of the CMTLS. It indicates that the CMTLS is equivalent to the CWTLS in terms of the transformation parameter estimates. The results of the CWLS are slightly different from the results of the CWTLS and CMTLS. This is because the CWLS ignores the errors of the coefficient matrix. However, the estimated standard deviations are different. As shown in Figure 2, the standard deviations calculated by the CWLS are obviously higher than those calculated by the CWTLS and CMTLS. The mean value of the standard deviations calculated by the CWLS for 1000 simulations is 0.0787, which is higher than the priori standard

deviation. The mean values of the standard deviations calculated by the CWTLS and CMTLS for 1000 simulations are all 0.0296, which is close to the priori standard deviation. Therefore, the standard deviations obtained by the CWTLS and CMTLS are more reasonable. This advantage is significant for the statistical analysis of the results. The standard deviations calculated by the CWTLS and CMTLS are useful for gross error detection and hypothesis testing etc.

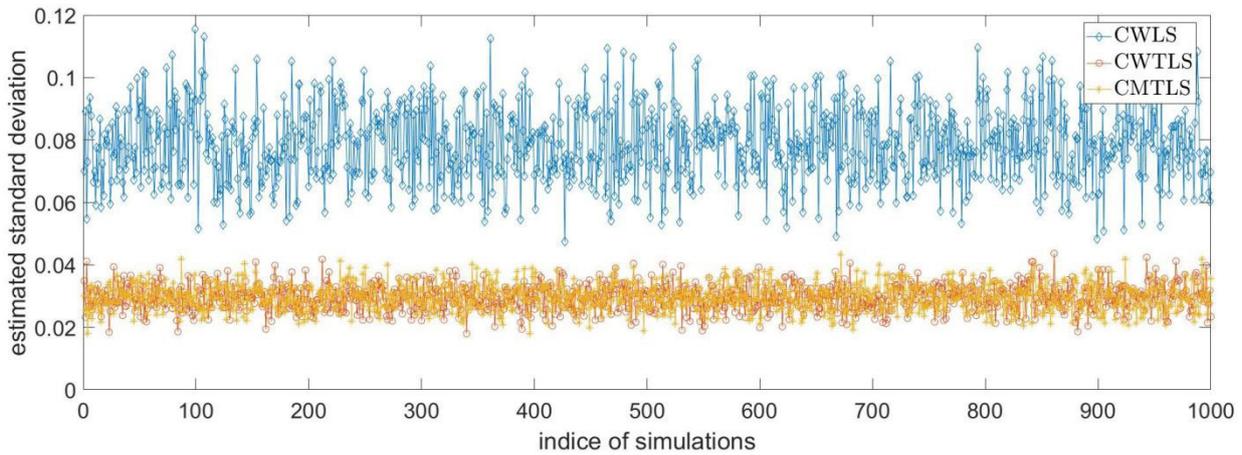


Figure 2: The standard deviations calculated by the three algorithms for 1000 simulations

Although the calculated results of the CWTLS are the same as those of the CMTLS, the computational efficiency of two methods is different. The cofactor matrix of the coefficient matrix cannot be directly obtained when implementing the CWTLS. We need to compute the jacobian matrix of the coefficient matrix with respect to the coordinates in the source coordinate system. Then, we can obtain the cofactor matrix of the coefficient matrix according to the cofactor propagation law. In contrast, the cofactor matrices of the coefficient matrix and observation matrix can be obtained without extra computation within the CMTLS, because the matrix G is given and easy to compile code, as shown in equations (13) and (14). Therefore, compared with the CWTLS, the CMTLS is easier to implement. In addition, for the CWTLS, there are numerous constants in the coefficient matrix, which result in a large computational burden.

In order to compare the computational efficiency of the CWTLS and CMTLS, we simulate five common points, ten common points, fifteen common points and twenty common points, respectively. The computation time is plotted as a function of the number of the common points. As shown in Figure 3, the computation time is obviously less when implementing the CMTLS. With the increasing number of the common points, the growth of the computation time is also slow. However, when implementing the CWTLS, the growth of the computation time is significant. Hence the CWTLS algorithm is relatively time-consuming. For big data processing, the computational efficiency is very important.

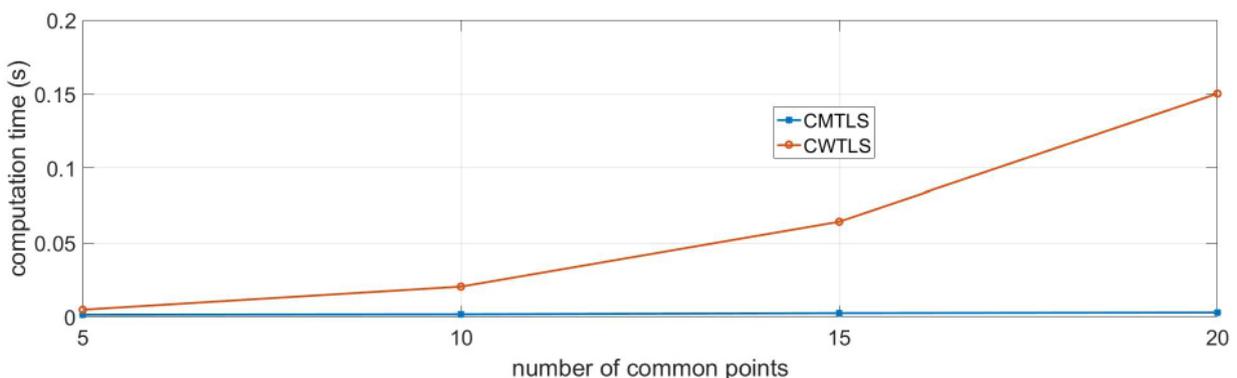


Figure 3: Computation time under different number of common points

In some practical project, the point coordinates are obtained by control network adjustment, so the cofactor matrix of the point coordinates should be fully-populated, i.e., the cofactor matrix is a non-diagonal matrix. For this reason, a case with non-diagonal cofactor matrices \mathbf{Q}_S and \mathbf{Q}_T is presented. The data are the twenty common points used before. The random errors with standard deviation of 0.01m are added to the true values of the coordinates. The priori standard deviation is set as 0.01. Thus, the cofactor matrix of each point is the identity matrix. We assume that the point coordinates are correlated. The empirical covariance function in Li et al. (2013) is employed to construct the cofactor matrices. The correlation coefficient of two arbitrary points is

$$\rho_{ij} = \frac{1}{1 + (d_{ij} / d_0)^2} \quad (45)$$

where d_{ij} is the distance between points i and j , and d_0 is a constant to govern the correlation strength, which is set as 1000 in this experiment. The maximum, minimum and mean correlation coefficients are respectively 0.9412, 0.0513 and 0.2813. Note that the correlation coefficient is only computed between two points in the same coordinate system.

Then, the CWLS, CWTLS and CMTLS are performed to calculate the transformation parameters. The results of the CWLS are as follows:

$$\hat{k} = 1.01000135, \hat{\mathbf{A}} = [6.00053850 \quad 6.98611275 \quad 8.00795856],$$

$$\hat{\mathbf{R}} = \begin{bmatrix} 0.35355997 & 0.92677411 & 0.12682696 \\ -0.61237080 & 0.12683006 & 0.78033078 \\ 0.70710489 & -0.35355886 & 0.61237145 \end{bmatrix}$$

The results of the CWTLS are as follows:

$$\hat{k} = 1.01000293, \hat{\mathbf{A}} = [5.99430186 \quad 6.99458201 \quad 8.00556967],$$

$$\hat{\mathbf{R}} = \begin{bmatrix} 0.35355541 & 0.92677543 & 0.12683003 \\ -0.61237264 & 0.12682506 & 0.78033015 \\ 0.70710559 & -0.35355719 & 0.61237161 \end{bmatrix}$$

The results of the CMTLS are as follows:

$$\hat{k} = 1.01000293, \hat{\mathbf{A}} = [5.99430186 \quad 6.99458201 \quad 8.00556967],$$

$$\hat{\mathbf{R}} = \begin{bmatrix} 0.35355541 & 0.92677543 & 0.12683003 \\ -0.61237264 & 0.12682506 & 0.78033015 \\ 0.70710559 & -0.35355719 & 0.61237161 \end{bmatrix}$$

It can be seen that the results of the CMTLS and CWTLS are the same, and the results of the CWLS are slightly different from those of the CMTLS and CWTLS. The estimated standard deviation and the computation time are listed in Table 2. As we can see, the standard deviations calculated by the CWTLS and CMTLS are closer to the priori standard deviation, while the one calculated by the CWLS is much far away from the priori standard deviation. We can also see that the computation time of the CMTLS is significantly smaller than that of the CWTLS. For the case with non-diagonal cofactor matrices, the CMTLS still yields the more reliable standard deviation in comparison with the CWLS and has a significantly better computational efficiency in comparison with the CWTLS.

Table 2: The estimated variance components and the computation time

Method	CWLS	CWTLS	CMTLS
Standard deviation	0.0283	0.0134	0.0134
Computation time (s)	/	9.0860	0.1642

In our algorithm, the scale factor and rotation matrix are put together as the parameter matrix, and the translation vector is calculated by equation (43), so our algorithm can only get the variances of the parameters (the elements in the parameter matrix) rather than the variances of the transformation parameters. This is a drawback of the proposed algorithm.

4.2 Point cloud registration

In order to indicate the importance of computational efficiency, a case involved point cloud registration is provided. The iterative closest point (ICP) algorithm (Besl and McKay, 1992) is the most popular registration method. It alternates between two steps until the convergence condition is reached. The first step is to establish correspondences, and the second step is to estimate the transformation. Here, the CWTLS and CMTLS algorithms are used for the second step. The TLS adjustment has already been used for the ICP-like algorithm by Tao et al. (2018) and Ge and Wunderlich (2016).

The data are the point clouds of the “Man head” model, as shown in Figure 4(a). The two point clouds respectively contain 11283 and 10862 points. We down-sample them to 25%, 50% and 75% of their original points. The computation time required by the CWTLS and CMTLS algorithms is recorded under different sampling rates, which is shown in Figure 4(b). The registration result without down-sampling is also illustrated in Figure 4(a). As we can see, the computation time of the CWTLS algorithm is significantly more than that of the CMTLS algorithm. For the point cloud registration, it often involves a large number of points, so the computation efficiency is crucial.

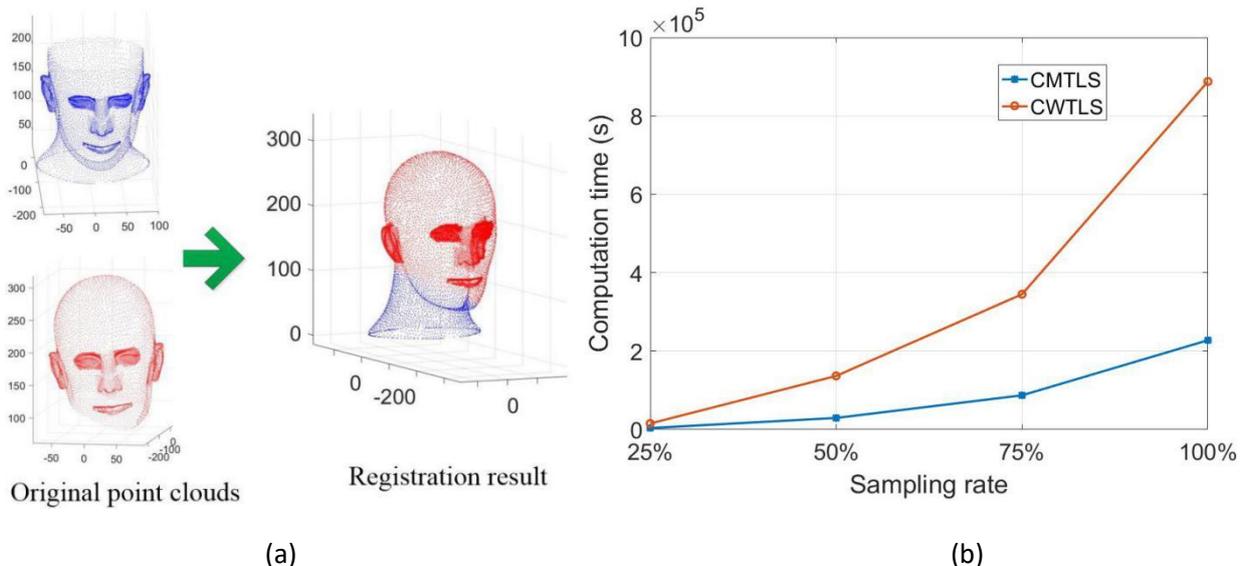


Figure 4: (a) the original points and the registration result, and (b) the computation time under different sampling rates

5. Conclusion

The 3D similarity transformation with large rotation angles is studied in this paper. A new method is developed to obtain the transformation parameters with multivariate model. Considering the property of the parameter matrix, the CMTLS algorithm is derived in this paper. Through the experiments, we can conclude:

Because the rotation matrix is an orthogonal matrix, the parameter matrix, which is the product of the scale factor and transposed rotation matrix, should satisfy the proper constraints. By comparing the results of the WMTLS and CMTLS, the matrix $\hat{\zeta}\hat{\zeta}^T$ calculated by the CMTLS is more consistent with the constraints. Hence the constraints are necessary.

The CMTLS algorithm and CWTLS algorithm take the errors of the coordinates in the two coordinate systems into consideration. The calculated standard deviations are more reliable, while the standard deviation calculated by the CWLS are greatly bigger than the priori standard deviation.

The CWTLS algorithm is more time-consuming due to a large number of constants existing in the coefficient matrix and the calculation of jacobian matrix. Hence, the computational efficiency of the CMTLS is higher compared to the CWTLS, so it is useful to process big data.

ACKNOWLEDGMENT

This research was partly supported by the National Natural Science Foundation of China (41674005; 41374011), and partly supported by the China Scholarship Council (CSC) scholarship (201906270179).

AUTHOR'S CONTRIBUTION

Wuyong Tao derived the algorithm and wrote the paper; Xianghong Hua performed the experiments and analyzed the results; Shaoquan Feng checked the paper.

REFERENCES

- Amiri-Simkooei, A.R. 2018. Parameter estimation in 3D affine and similarity transformation: implementation of variance component estimation. *J Geod*, 92(11), pp. 1285-1297.
- Besl, P. J. and McKay, D. N. 1992. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2), pp. 239–256.
- Chen Y. Shen, Y. Liu, D. 2004. A simplified model of three dimensional-datum transformation adapted to big rotation angle. *Geomatics and Information Science of Wuhan University*, 29(12), pp. 1101-1105 (In Chinese with English Abstract).
- Chang, G. 2016. Closed form least-squares solution to 3D symmetric helmert transformation with rotational covariance structure. *Acta Geodaetica et Geophysica*, 51(2), pp. 237-244.
- De Moor, B. 1993. Structured total least squares and L2 approximation problem. *Linear Algebra and Its Application*, 188, pp. 163-205.

- Felus, Y.A. Burtch, R.C. 2009. On symmetrical three-dimensional datum conversion. *GPS Solution*, 13(1), pp. 65-74.
- Fang, X. 2011. Weighted total least squares solution for applications in geodesy. *Dissertation, Leibniz University Hanover*, pp. 39-41.
- Fang, X. 2014. A total least squares solution for geodetic datum transformation. *Acta Geod Geophys*, 49(2), pp. 189-207.
- Fang, X. 2015. Weighted total least-squares with constraints: a universal formula for geodetic symmetrical transformation. *J Geod*, 89(5), pp. 459-469.
- Golub, G.H. van Loan, C.F. 1980. An analysis of the total least squares problem. *SIAM J Numer Anal*, 17(6), pp. 883-893.
- Ge, X. and Wunderlich, T. 2016. Surface-based matching of 3D point clouds with variable coordinates in source and target system. *ISPRS Journal of Photogrammetry and Remote Sensing*, 111, pp. 1-12.
- Kwuashi, O. Eyoh, E. 2012. 3D coordinate transformation using total least squares. *Academic Research International*, 3(1), pp. 2223-9944.
- Li, B. Shen, Y. Li, W. 2012. The seamless model for three-dimensional datum transformation. *Science China Earth Science*, 42(7), pp. 1047-1054.
- Li, B. Shen, Y. Zhang, X. Li, C. Lou, L. 2013. Seamless multivariate affine error-in-variables transformation and its application to map rectification. *International Journal of Geographical Information Science*, 27(8), pp. 1572-1592
- Lu, J. Chen, Y. Li, B. Fang, X. 2014. Robust total least squares with reweighting iteration for three-dimensional similar transformation, *Survey Review*, 46(334), pp. 28-36.
- Lu, J. Chen, Y. Fang, X. Zheng, B. 2015. Performing 3D similarity transformation using the weighted total least-squares Method. *Munich: International Association of Geodesy Symposia*, 140, pp. 71-77.
- Laari, P.B. Ziggah, Y.Y. Annan, R.F. 2016. Determination of 3D transformation parameters for the Ghana Geodetic reference network using ordinary least squares and total least squares technology. *International Journal of Geomatics and Geoscience*, 7(3), pp. 245-261.
- Mercan, H. Akyilmaz, O. Aydin, C. 2018. Solution of the weighted symmetric similarity transformations based on quaternions. *J Geod*, 92(10), pp. 1113-1130.
- Neitzel, F. 2010. Generalization of total least- squares on example of unweighted and weighted 2D similarity transformation. *J Geod*, 84(12), pp. 75-762.
- Pope, A. 1972. Some pitfalls to be avoided in the iterative adjustment of nonlinear problems. *Proceedings of the 38th Annual Meeting of the American Society Photogrammetry, May 12-17, Washington DC*, pp. 449-473.
- Rapp, R. 1993. Geometric geodesy part II. *Ohio: The Ohio State University*, pp. 60-75.
- Schaffrin, B. Felus, Y.A. 2008. On the multivariate total least squares approach to empirical coordinate transformations: three algorithm. *J Geod*, 82(6) , pp. 373-383.
- Shi, Y. Xu, P. Liu, J. Shi, C. 2015. Alternative formulae for parameter estimation in partial error-in-variables model. *J Geod*, 89(1), pp. 13-16.
- Tao, W., Hua, X., Yu, K., He, X. and Chen, X. 2018. An improved point-to-plane registration method for terrestrial laser scanning data. *IEEE Access*. 6, pp. 48062-48073.
- Wang, L. Xu, G. 2016. Variance component estimation for partial errors-in-variables models. *Stud. Geophys. Geod*, 60(1), pp. 35-55.
- Xu, P. Liu, J. Shi, C. 2012. Total least squares adjustment in partial errors-in-variables: algorithm and statistical analysis. *J Geod*, 86(8), pp. 661-675.
- Xu, P. Liu, J. 2014 Variance components in error-in-variables models: estimability, stability and bias Analysis. *J Geod*. 88(8), pp. 719-734.
- Yang, Y. 1999. Robust estimation of geodetic datum transformation. *J Geod*, 73(5),pp. 268-274.

Yan, S., and Fan, J. 2001. The solution set of the mixed LS-TLS problem. *International Journal of Computer Mathematics*, 77(4), pp. 545-561.

You, R. Hwang, H. 2006. Coordinate transformation between two geodetic datums of taiwan by least-squares collocation. *Journal of Surveying Engineering*, 132(2), pp. 64-70.

Yang, J. Wang, Y. Wang, Q. Tao, Y. 2017. Solution for rank-defect EIV model based on TLS estimation," *Survey review*, 49(354) , pp. 171-175.

Zeng, W. Liu, J. Yao, Y. 2015. On partial error-in-variables models with inequality constraints of parameters and variables. *J Geod*, 89(2), pp. 111-119.

Zhao, J. Gui, Q. 2017. Outlier detection in partial errors-in-variables model. *Boletim de Ciências Geodésicas*, 23(1), pp. 1-20.

6. APPENDIX

The CWLS algorithm and CWTLS algorithm for the 3D similarity transformation are presented in this section. The observation equation is

$$\mathbf{e}_y = \mathbf{B}\mathbf{X} - \mathbf{y} \quad (\text{A1})$$

where $\mathbf{y} = \text{vec}(\mathbf{T})$, $\mathbf{B} = [\mathbf{I}_3 \otimes \mathbf{S} \quad \mathbf{I}_3 \otimes \mathbf{I}_n]$, $\mathbf{X} = [\text{vec}^T(\hat{\mathbf{t}}) \quad \hat{\mathbf{A}}]^T$ and \mathbf{e}_y denotes the residual vector of \mathbf{y} .

We linearize equation (A1)

$$\mathbf{e}_y = \mathbf{B}\delta\mathbf{X} - \mathbf{L} \quad (\text{A2})$$

where $\mathbf{L} = \mathbf{y} - \mathbf{B}\mathbf{X}_0$, $\delta\mathbf{X} = \mathbf{X} - \mathbf{X}_0$ and \mathbf{X}_0 denotes the initiate value of \mathbf{X} .

Considering the constraints, we obtain

$$\begin{cases} \mathbf{e}_y = \mathbf{B}\delta\mathbf{X} - \mathbf{L} \\ [\mathbf{C} \quad \mathbf{0}_{5 \times 3}] \delta\mathbf{X} + \mathbf{W}_0 = 0 \end{cases} \quad (\text{A3})$$

where $\mathbf{0}_{5 \times 3}$ is a 5×3 matrix whose elements are all zero, the meanings of other letters are the same as those mentioned above.

If the errors of the coefficient matrix \mathbf{B} are neglected, the CWLS algorithm can be used to solve equation (A3). The cofactor matrix of \mathbf{L} is \mathbf{Q}_T when implementing the CWLS algorithm. If the errors of the coefficient matrix are considered, the CWTLS algorithm can be used to solve equation (A3). The cofactor matrix of the coefficient matrix \mathbf{B} is $\mathbf{Q}_B = \mathbf{M}\mathbf{Q}_S\mathbf{M}^T$, and the cofactor matrix of \mathbf{L} is still \mathbf{Q}_T . \mathbf{M} is the jacobian matrix of the coefficient matrix with respect to the coordinates in the source coordinate system. The CWLS algorithm and CWTLS algorithm have been studied extensively. Therefore, no longer described here. It is worth noting that the translation vector is not eliminated and the parameter matrix $\hat{\mathbf{t}}$ is still the product of the scale factor and transposed rotation matrix.