ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

THREE-DIMENSIONAL TRANSFORMATION OF COORDINATE SYSTEMS USING NONLINEAR ANALYSIS – PROCRUSTES ALGORITHM

Corina Daniela Păun *, Valeria Ersilia Oniga, Petre Iuliu Dragomir

* Department of Topography and Cadastre, Technical University of Civil Engineering, Bucharest, Faculty of Geodesy, Romania

DOI: 10.5281/zenodo.291839

ABSTRACT

The coordinates transformation between two projection systems is increasingly approached in geodesy, photogrammetry and computer vision. This issue led to widespread development of algorithms that offer high accuracy and fast processing time. One of this algorithms was presented in this paper, namely the Procrustes algorithm, which involves the usage of seven parameters of transformation (three of translation, three of rotation and one scalar). The main advantages of this algorithm is that you don't have to know the initial values of the parameters, like in the case of iterative numerical methods and the process of equations linearization in order to obtain the equation correction system, isn't necessary. So, to the standard equation of 3D- transformation, the Lagrange function is applied, together with some constraints on rotation matrix (R must be orthogonal) and some derivation conditions. In order to obtain more accurate results in the 3D transformation of coordinates the weights matrix is introduced, which is calculated separately based on variance-covariance matrix, using the error matrix. This paper also investigates the stability of using this algorithm for the registration of two TLS point clouds, by comparing the results with those obtained by applying the 3D conformal transformation. When using the last algorithm, the initial values of the 7- parameters were calculated and an iterative optimization of these parameters was applied.

KEYWORDS: 2 Procrustes algorithm, 7- parameters transformation, TLS, registration, comparison.

INTRODUCTION

The tridimensional coordinates transformation is a process that have frequently been used in geodesy, survey engineering, but also in photogrammetry. The determination of transformation parameters requires knowledge of a minimum number of common points in the two reference systems.

Until now, researchers have been developed and presented a large number of algorithms to determine these parameters that can be divided into two categories: numerical iterative algorithm and analytical algorithm [1].

In the case of algorithms belonging to the first category, knowledge of initial value (approximate) parameters, correction equations linearization and iterative calculation, are necessary. In the case when rotation angles are large, the approximate values of the parameters are difficult to determinate or impossible in some cases, leading to the impossibility of method implementation (method failure) [1].

The algorithms belonging to the second category does not involve knowledge of initial values for the parameters, equation linearization or iterative calculation, so that the solutions are calculated quickly and accurately. But these algorithms have a small drawback, namely the high degree complexity of mathematical derivation, therefore their use and implementation was done fewer times.

One of these algorithms has been implemented by [2], called Procrustes algorithm, and further was improved by [1] introducing the technique of singular values decomposition (SVD).

The singular values decomposition is one of the most complex methods used in linear algebra, specifically matrix calculation. Using this method many problems were solved, such as calculating the rank of a matrix, orthogonal bases for the linear subspace, the least squares, or in this case, determining the values of rotation matrix.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

MATERIALS AND METHODS

The Classic Procrustes Algorithm

The coordinates transformation involves the calculation of seven parameters, three of translation, three of rotation and one scalar and it is given by the following mathematical formulation:

$$XYZ = \lambda R\,xyz + T_{xyz}\,1_n + E \tag{1}$$

where: (X,Y,Z) and (x,y,z) are the coordinates of a point P in the reference coordinate system, respectively arbitrary coordinate system, T_{xyz} is the vector of the three translation parameters, R is the rotation matrix, λ is a scalar,$1_n=[1,1,1,..1]^T$ is a vector with ***n*** components and all elements have a value of 1 - unity vector, E is the error transformation matrix (we added this term because we have supposed that the points have errors).

The rotation matrix is orthogonal and for this reason the following conditions must be met [3]:

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

In other words, R is calculated based on three rotation angles (ω, φ, κ) around coordinate axes *X*,*Y* and *Z* and has the following form:

$$R = \begin{bmatrix} \cos\varphi\cos k & \cos\omega\sin k + \sin\omega\sin\varphi\cos k & \sin\omega\sin k - \cos\omega\sin\varphi\cos k \\ -\cos\varphi\sin k & \cos\omega\cos k - \sin\omega\sin\varphi\sin k & \sin\omega\cos k + \cos\omega\sin\varphi\sin k \\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi \end{bmatrix} \tag{3}$$

If the elements of this matrix are known, the values of the three rotation angles can be determinate, using the following formulas:

$$\omega = -\tan^{-1}\left(\frac{r_{32}}{r_{33}}\right), \quad \varphi = \sin^{-1}(r_{31}), \quad \kappa = -\tan^{-1}\left(\frac{r_{21}}{r_{11}}\right) \tag{4}$$

For the first equation, the minimum condition (least squares) was apply, which will resolve based on Lagrangian multiplier matrix Λ . Taking into account the constraints given by Eq. 2, Lagrange function will write, depending on the terms of equation 1 (*tr-trance*):

$$L(\lambda, T, R, \Lambda) = tr(EE^T) + tr(\Lambda(R^T R - I_3)) = \min \tag{5}$$

The method of Lagrangian multiplier, solves the constrained optimization problem (Eq.1) by transforming it into a non-constrained optimization (see Eq. 6). So, we compute the derivates of Lagrangian function (Eq. 8) and the result is the optimum values of (*λ, T, R*). In other words, by introducing this function the linearization of correction equations will not be necessary attending to solve the system.

So, the expression of E will be replaced (written on eq. 1) in eq. 5 and will get:

$$\begin{aligned} L(\lambda, T, R, \Lambda) &= tr(EE^T) + tr(\Lambda(R^T R - I_3)) = \\ &= tr((A - \lambda RB - T1_n)(A - \lambda RB - T1_n)^T) + tr(\Lambda(R^T R - I_3)) = \min \end{aligned} \tag{6}$$

For an easier writing of the above equation, will make the following notations:

$$A = [X_i \quad Y_i \quad Z_i]; \quad B = [x_i \quad y_i \quad z_i] \tag{7}$$

The Lagrangian function exists if the followings condition are satisfied:

$$\frac{\partial L}{\partial \lambda} = 0; \quad \frac{\partial L}{\partial T} = 0; \quad \frac{\partial L}{\partial R} = 0; \quad \frac{\partial L}{\partial \Lambda} = 0. \tag{8}$$

Based on the above conditions, the following equations results:

$$\begin{aligned} &L(\lambda, T, R, \Lambda) = tr((A - \lambda RB - T1_n)(A^T - \lambda B^T R^T - 1_n{}^T T^T)) + tr(\Lambda(R^T R - I_3)) = \\ &= tr((A - \lambda RB)(A^T - \lambda B^T R^T) - (A - \lambda RB)1_n{}^T T^T - T1_n(A^T - \lambda B^T R^T) \\ &+ T1_n 1_n{}^T T^T) + tr(\Lambda(R^T R - I_3)) = \min \end{aligned} \tag{9}$$

$$\frac{\partial L}{\partial T} = -(A - \lambda RB)1_n{}^T - (A - \lambda RB)1_n{}^T + 2\,1_n 1_n{}^T T = -2(A - \lambda RB)1_n{}^T + 2\,1_n 1_n{}^T T = 0 \tag{10}$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Eq. 10 will give the translation vector, which is given by the following equation:

$$T = (1_n 1_n^T)^{-1}(A - \lambda RB)1_n^T \quad (11)$$

It is noted that the vector T depends on the scalar and the rotation matrix R (as expected). Next, Eq.11 is replaced in the translation vector in Eq. 1, obtaining:

$$E = A - \lambda RB - (1_n 1_n^T)^{-1}(A - \lambda RB)1_n^T 1_n = (A - \lambda RB)(I_n - (1_n 1_n^T)^{-1}1_n^T 1_n) \quad (12)$$

where: $I_n - (1_n 1_n^T)^{-1}1_n^T 1_n = I_n - \frac{1}{n}1_n^T 1_n$, is called the centering matrix.

Further, for an easier writing of the equation, the following notations are done:

$$\Delta A = A(I_n - \frac{1}{n}1_n^T 1_n), \quad \Delta B = B(I_n - \frac{1}{n}1_n^T 1_n) \quad (13)$$

Eq. 13 are the centralized coordinate matrix of two coordinate systems, reduces the coordinates to the centering matrix. Eq. 12 will be rewritten as:

$$E = \Delta A - \lambda R \Delta B \quad (14)$$

Replacing Eq. 14 into Eq. 6, we obtain:

$$\begin{aligned} L(\lambda, R, \Lambda) &= tr((\Delta A - \lambda R\Delta B)(\Delta A - \lambda R\Delta B)^T + tr(\Lambda(R^T R)) \\ &= tr(\Delta A\Delta A^T - 2\lambda \Delta A\Delta B^T R^T + \lambda^2 R\Delta B\Delta B^T R^T) + tr(\Lambda(R^T R)) \end{aligned} \quad (15)$$

The derivation of Lagrange function (Eq.15) shall be conform matrix operations, e.g. the matrix trance: $tr(\Delta A\Delta B^T R^T) = tr((\Delta A\Delta B^T R^T)^T) = tr(R\Delta B\Delta A^T)$.

So, the function L is derived based on scalar **λ**, resulting:

$$\frac{\partial L}{\partial \lambda} = -2tr(\Delta A\Delta B^T R^T) + 2\lambda tr(R\Delta B\Delta B^T R^T) = -2tr(\Delta A\Delta B^T R^T) + 2\lambda tr(\Delta B\Delta B^T) = 0 \quad (16)$$

From Eq. 16, the value of scalar can be determined:

$$\lambda = \frac{tr(\Delta A\Delta B^T R^T)}{tr(\Delta B\Delta B^T)} \quad (17)$$

Next, the function L will be derived based on the rotation matrix, using the Eq. 15:

$$\frac{\partial L}{\partial R} = 2(\Delta A - \lambda R\Delta B)(-\lambda \Delta B^T) + R(\Lambda + \Lambda^T) = -2\lambda \Delta A\Delta B^T + 2\lambda^2 R\Delta B\Delta B^T + 2R\Lambda = 0 \quad (18)$$

The rotation matrix is given by the following equation:

$$R = \lambda \Delta A\Delta B^T(\lambda^2 \Delta B\Delta B^T + \Lambda)^{-1} \quad (19)$$

By derivation of Lagrange function ($L(\lambda, R, \Lambda)$), based on Lagrangian multiplier matrix (Λ), we get the following equation, that represent the constraint required at the beginning of the algorithm:

$$R^T R - I_3 = 0 \quad (20)$$

Replacing the Eq. 19 in Eq. 20, one gets:

$$\lambda^2(\lambda^2 \Delta B\Delta B^T + \Lambda)^{-1}\Delta B\Delta A^T \Delta A\Delta B^T(\lambda^2 \Delta B\Delta B^T + \Lambda)^{-1} = I_3 \quad (21)$$

$$\lambda^2 \Delta B\Delta B^T + \Lambda = \lambda(\Delta B\Delta A^T \Delta A\Delta B^T)^{\frac{1}{2}} \quad (22)$$

From Eq. 19 and 22, we obtained the formula for the calculation of the rotation matrix coefficients:

$$R = \Delta AP\Delta B^T(\Delta BP\Delta A^T \Delta AP\Delta B^T)^{-\frac{1}{2}} \quad (23)$$

After some notations, the rotation matrix becomes:

$$R = D(D^T D)^{-\frac{1}{2}}, \text{ where: } D = \Delta A\Delta B^T \quad (24)$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

The Weight Procrustes Algorithm

The determination of the weight matrix

For determining the weight matrix of each point, the determination of the variance-covariance matrix is necessary.

In order to determine the dispersion for each set of coordinates, the generation of variance-covariance matrix has to be done. So, the following steps are necessary:

-compute the errors matrix:

$$E = A - (\lambda R \square B + T_{xyz} \square 1_n) \tag{25}$$

-compute the mean for each set of coordinates, for both systems (reference and arbitrary system);

$$M_{AX} = \frac{\left(\sum_{i=1}^{n} X_i\right)}{n}; M_{AY} = \frac{\left(\sum_{i=1}^{n} Y_i\right)}{n}; M_{AZ} = \frac{\left(\sum_{i=1}^{n} Z_i\right)}{n} \tag{26}$$

$$M_{Bx} = \frac{\left(\sum_{i=1}^{n} x_i\right)}{n}; M_{By} = \frac{\left(\sum_{i=1}^{n} y\right)}{n}; M_{Bz} = \frac{\left(\sum_{i=1}^{n} z_i\right)}{n} \tag{27}$$

- compute the dispersion;

$$D_{A^T} = M\left\{\begin{bmatrix} X_1 - M_{AX} \\ \dots \\ X_i - M_{AX} \\ Y_1 - M_{AY} \\ \dots \\ Y_i - M_{AY} \\ Z_1 - M_{AZ} \\ \dots \\ Z_i - M_{AZ} \end{bmatrix} \begin{bmatrix} (X_1 - M_{AX})^T & \dots & (X_i - M_{AX})^T & (Y_1 - M_{AY})^T & \dots & (Y_i - M_{AY})^T & (Z_1 - M_{AZ})^T & \dots & (Z_i - M_{AZ})^T \end{bmatrix}\right\} \tag{28}$$

$$D_{B^T} = M\left\{\begin{bmatrix} x_1 - M_{Ax} \\ \dots \\ x_i - M_{Ax} \\ y_1 - M_{Ay} \\ \dots \\ y_i - M_{Ay} \\ z_1 - M_{Az} \\ \dots \\ z_i - M_{Az} \end{bmatrix} \begin{bmatrix} (x_1 - M_{Ax})^T & \dots & (X_i - M_{Ax})^T & (y_1 - M_{Ay})^T & \dots & (y_i - M_{Ay})^T & (z_1 - M_{Az})^T & \dots & (z_i - M_{Az})^T \end{bmatrix}\right\} \tag{29}$$

-compute the variance-covariance matrix (errors dispersion matrix)

$$D_{vec}E^T = M\left\{[vecE^T - M\left\{vecE^T\right\}][vecE^T - M\left\{vecE^T\right\}]^T\right\} =$$
$$= M\left\{[vecA^T - M\left\{vecA^T\right\} - \lambda(vecRB^T - M\left\{vecRB^T\right\})] \times [vecA^T - M\left\{vecA^T\right\} - \lambda(vecRB^T - M\left\{vecRB^T\right\})]^T\right\} \tag{30}$$

$$D_{vec}E^T = M\left\{[vecA^T - M\left\{vecA^T\right\}][vecA^T - M\left\{vecA^T\right\}]^T\right\} + \lambda^2 M\left\{[vecRB^T - M\left\{vecRB^T\right\}][vecRB^T - M\left\{vecRB^T\right\}]^T\right\} -$$
$$-2\lambda M\left\{[vecA^T - M\left\{vecA^T\right\}][vecRB^T - M\left\{vecRB^T\right\}]^T\right\} \tag{31}$$

$$D_{vec}E^T = D_{vec}A^T + D_{vec}\lambda RB^T - 2D_{vec}(A^T, vec\lambda RB^T) \tag{32}$$

where: $vecA^T = [X_1, Y_1, Z_1, ..., X_n, Y_n, Z_n]$ and $vecB^T = [x_1, y_1, z_1, ..., x_n, y_n, z_n]$ transforms the transpose matrix of the points coordinates in both systems, into a vector with $3n \times 1$ dimension (n is the number of common points).

In Eq. 32, the following developments are done:

- it is known that [4]:

$$vecAB = (I_q \otimes A)vecB, \quad for \quad A \in \square^{n \times m}, B \in \square^{m \times q} \tag{33}$$

So based on the above formula, ***vecRB^T*** becomes:

$$vecRB^T = (I_n \otimes \lambda R)vecB^T, \quad for \quad R \in \square^{3 \times 3}, B^T \in \square^{3 \times n} \tag{34}$$

Further, the Eq. 34 is replaced into Eq. 32 and the variance-covariance matrix becomes:

$$D_{vec}E^T = D_{vec}A^T + (I_n \otimes \lambda R)D_{vec}B^T(I_n \otimes \lambda R)^T - 2D_{vec}(A^T, (I_n \otimes \lambda R)vecB^T) \tag{35}$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Eq. 35 can be interpreted in detail: $D_{vec}A^T$ and $D_{vec}B^T$ are the dispersion matrix (variance-covariance) of the coordinates sets from the reference and arbitrary system, respectively; $D_{vec}(A^T,(I_n\otimes\lambda R)vecB^T)$ is the dispersion computed between vector $vecA^T$ and the term $(I_n\otimes\lambda R)vecB^T$ given by *Kronecker-Zehfuss* determination.

After performing the specific operations, the dispersion matrix of errors E is obtained, having the size of $3n\times 3n$. Within this matrix, only the elements on the main diagonal are taken into consideration.

$$D_{vec}E^T=\begin{bmatrix}\sigma_{x_1}^2 & & & & & & \\ & \sigma_{y_1}^2 & & & & & \\ & & \sigma_{y_1}^2 & & & & \\ . & . & . & . & . & . & \\ & & & \sigma_{x_n}^2 & & & \\ & & & & \sigma_{x_n}^2 & & \\ & & & & & \sigma_{x_n}^2 & \end{bmatrix} \tag{36}$$

From the above matrix, using only the elements of the main diagonal the dispersion of each point is calculated like a geometric mean:

$$\sigma_i^2=\sqrt{\sigma_{x_i}^2+\sigma_{y_i}^2+\sigma_{y_i}^2},\quad i=\overline{1,n} \tag{37}$$

Finally, after performing all operations, the error matrix E result (size $n\times n$),with elements only on the principal diagonal:

$$E=\begin{bmatrix}\sigma_1^2 & & & \\ & \sigma_2^2 & & \\ . & . & . & . \\ & & & \sigma_n^2\end{bmatrix} \tag{38}$$

The weight matrix P is equal to the inverse error matrix E.

$$P=E^{-1}=\begin{bmatrix}p_1 & & & \\ & p_2 & & \\ . & . & . & . \\ & & & p_n\end{bmatrix} \tag{39}$$

The transformation parameters calculation by using the weight Procrustes algorithm

The determination of seven transformation parameters will be achieved by applying weight Procrustes algorithm, using the weight matrix calculated in previous section.

The weight Procrustes algorithm is based on Lagrange function, that solves the problem of the minimum condition, without requiring the linearization of the equation system.

The equations used to calculate the weighted transformation parameters, are the same as in case of the classical algorithm, noting that the weights matrix will be introduce.

$$L(\lambda,T,R,\Lambda)=tr(EPE^T)+tr(\Lambda(R^TR-I_3))=\min \tag{40}$$

By applying a equations from section 2 (Eq. 9 - Eq. 22) and a derivation conditions, (Eq. 8) the translation vector, the scalar and the rotation matrix are given by the following equations:

$$T=(1_nP1_n^T)^{-1}(A-\lambda RB)P1_n^T \tag{41}$$

$$\lambda=\frac{tr(\Delta AP\Delta B^TR^T)}{tr(\Delta BP\Delta B^T)} \tag{42}$$

$$R=\lambda\Delta AP\Delta B^T(\lambda^2\Delta BP\Delta B^T+\Lambda)^{-1} \tag{43}$$

If the Lagrange function is derived ($L(\lambda,R,\Lambda)$) according to Lagrangian multiplier matrix (Λ),the constraint equation is obtained:

$$R^TR-I_3=0 \tag{44}$$

Substituting the Eq. 44 into the rotation matrix equation (Eq. 43), we obtain:

$$\lambda^2(\lambda^2\Delta BP\Delta B^T+\Lambda)^{-1}\Delta BP\Delta A^T\Delta AP\Delta B^T(\lambda^2\Delta BP\Delta B^T+\Lambda)^{-1}=I_3 \tag{45}$$

$$\lambda^2\Delta BP\Delta B^T+\Lambda=\lambda(\Delta BP\Delta A^T\Delta AP\Delta B^T)^{\frac{1}{2}} \tag{46}$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Based on Eq.43 and Eq. 46, the equation for calculating the coefficients of the rotation matrix is obtained:

$$R = \Delta A P \Delta B^T (\Delta B P \Delta A^T \Delta A P \Delta B^T)^{-\frac{1}{2}} \quad (47)$$

After some notations the R matrix becomes:

$$R = D(D^T D)^{-\frac{1}{2}}, \text{ where: } D = \Delta A \Delta B^T \quad (48)$$

The rotation angles can be computed later as in the case of the classic algorithm, using the Eq. 4 or more simply, by applying the singular values decomposition algorithm.

THE CASE STUDY

For this case study, we considered the three dimensional coordinates of five points, manually chose and measured from two point clouds acquired with ScanStation 2 terrestrial laser scanner, representing natural tie points (Fig. 1). The points coordinates were measured after the two point clouds were represented in the same coordinate system, namely the one of the left point cloud and can be found in [5]. The first point cloud, colored in red, is considered as reference and the second one, colored in blue, as arbitrary. The tie points distribution is random, but were chosen so as to cover approximately the entire point cloud.

The ScanStation2, produced by Leica Geosystems, is a terrestrial scanner system using laser-pulsed technology for distance measurement with a precision of 6 mm at a distance of 50 m, which has an integrated 1 MP resolution digital camera [5].

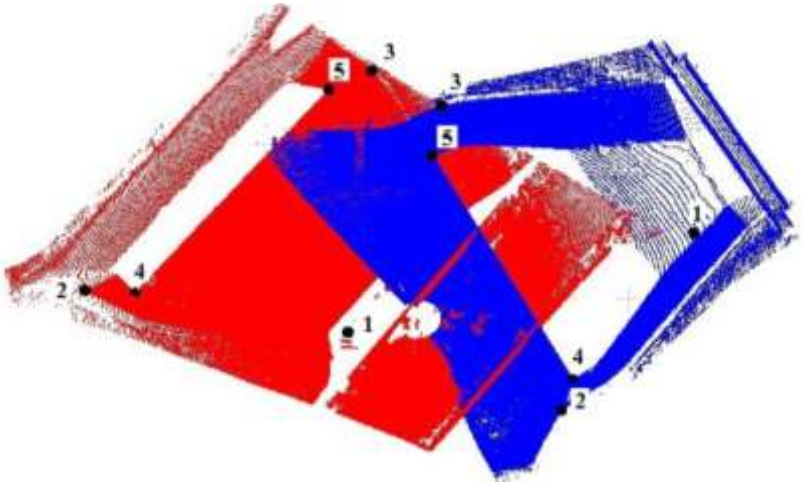

Figure 1. Graphical representation of the TLS point clouds in the left point cloud coordinate system and the five tie points pairs [5]

In this section, we present the classical and the weighted Procustes algorithms in the process of TLS point clouds registration.

In the first stage, the seven transformation parameters were computed (Table 1), the place of the weight matrix being taken by the identity matrix. So, the dispersion matrix of errors was calculated using the scalar and the rotation matrix values. The resulting dispersion matrix with a size of 3n x3n, will be the errors matrix for these 5 points and only the elements from the main diagonal will be considered (section 3.1). Using the Eq. 42, the root mean square error (RMS) for each of the five points was calculated, the result being a diagonal matrix with a size of *n x n*, called error matrix ***E***.

By applying the inverse of this matrix, the weights specify to each given point will be calculated. In this case, every point has isotropic weight and is independent of each other [2], weight matrix is generated and is given in Table 2.

In the second stage, the weight Procrustes algorithm for the transformation parameters determination using the weights of each point (section 3.2), was computed. The new set for the seven transformation parameters is presented in Table 3. For a comparison purpose, we have also introduced the results of the 3D conformal transformation obtained in [5] (Table 5).

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Table 1.The transformation parameters calculated by using the Classic Procrustes Algorithm

	Values		
Rotation matrix R	0.38268 -0.92388 0.0016915	0.92388 0.38268 0.0017449	-0.0022594 0.00089502 1.0000
Rotation angle	**ω** -0.051281 **φ** -0.12945 **k** -67.5		
Translation **Tx** **Ty** **Tz**	 -19.896 21.22 -3.8812		
Scale	1.0007		
Mean error	0.044124		

Table 2.Weights matrix

0.040119	0	0	0	0
0	0.089652	0	0	0
0	0	0.081375	0	0
0	0	0	0.093759	0
0	0	0	0	0.000458

Table 3. The transformation parameters calculated by using the Weight Procrustes Algorithm

	Values		
Rotation matrix R	0.38287 -0.9238 0.00089171	0.9238 0.38287 0.0018461	-0.0020468 0.00011697 1.00000000
Rotation angle	**ω** -0.041807 **φ** -0.14975 **k** -67.564		
Translation **Tx** **Ty** **Tz**	 -19.879 21.243 -3.8737		
Scale	1.002		
Mean error	0.012523		

In the success of this algorithm, the Eq. 2 carries a very important role, because it is a necessary condition to be met by the rotation matrix.

Both Procustes algorithms, classical and weighted, were implemented into Matlab programming language, the implementation and processing times being much faster than any other 3D coordinate transformation method, i.e. 3D conformal transformation.

In order to establish the success and the stability of the Procustes algorithm in the process of TLS point clouds registration, the 7- transformation parameters obtained by using both methods (classical and weighted), will be compared with those obtained by using the 3D conformal transformation.

The 3D conformal transformation is one of the iterative methods, which involves performing a preliminary determination of the 7- transformation parameters (initial values) and an iteratively solve of the unknowns (the parameters corrections).

In Table 4, the residual of tie points coordinates when applying the CP and WCP, respectively are presented, as well as the differences between them. It can be seen that the differences are quite small, of centimeters or even millimeters.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Table 4. Residual of tie points coordinates when applying the CP and WP transformation and the differences between them

No	Residual of tie points coordinates when applying the Classic- Procrustes algorithm			Residual of tie points coordinates when applying the Weight- Procrustes algorithm			Differences		
	X(m)	*Y(m)*	*Z(m)*	*X(m)*	*Y(m)*	*Z(m)*	*diffX(m)*	*diffY(m)*	*diffZ(m)*
1	0.03730	-0.01477	-0.01384	0.02162	0.00376	-0.02000	0.01569	-0.01853	0.00616
2	-0.03019	0.06842	-0.01171	-0.02200	0.05775	-0.01526	-0.00819	0.01067	0.00355
3	0.02944	0.00920	0.01463	-0.01169	-0.00168	0.00259	**0.04113**	0.01088	0.01204
4	0.02000	-0.04846	0.02371	0.02193	-0.05537	0.02090	-0.00194	0.00691	**0.00281**
5	-0.05654	-0.01440	-0.01278	-0.10234	-0.02113	-0.02866	**0.04580**	0.00673	0.01587

The differences between the tie points residuals obtained by applying the CP and WCP algorithms, shows that weighted Procrustes transformation provides better results, because the residuals of tie points coordinates are smaller and the root mean squared error when applying this algorithm has a values of **1.25 cm**, compared to the value of **4. 41 cm** achieved when applying the classical algorithm.

The values of the transformation parameters, obtained by applying the three methods of transformation described above, i.e. 3D Conformal, Classical and Weighted Procrustes algorithms, as well as the differences between them, are given in Table 5.

Table 5. Parameters of 3D conformal transformation, Classical Procrustes and Weighted Procrustes respectively and the differences between them

Parameter	Values obtained by 3D conformal transformation	Values obtained by Classical Procrustes	Values obtained by Weighted Procrustes	Differences 3D Conformal - CP	Differences 3D Conformal- WP
λ	1.000675	1.0007	1.002	**-0.00002**	-0.00133
ω	-0.051281º	-0.051281º	-0.041807º	0.00000º	-0.00947º
φ	-0.129454º	-0.12945º	-0.14975º	0.00000º	0.02030º
k	-67.500083º	-67.5º	-67.564º	**-0.00008º**	**0.06392º**
T_X	-19.896 m	-19.896 m	-19.879 m	0.00000 m	-0.01700 m
T_Y	21.220 m	21.220 m	21.243 m	0.00000 m	**-0.02300 m**
T_Z	-3.881 m	-3.8812 m	-3.8737 m	**0.00020 m**	-0.00730 m

CONCLUSION

In this paper, the steps for 3D coordinate transformation between two coordinate systems when applying the classical and the weighted Procrustes algorithms, were presented. Then, an analysis of their potential in the process of TLS point clouds registration, based on pairs of common points, randomly chosen and evenly distributed in the point clouds, was performed. So, the 7- transformation parameters were calculated by using both methods, classical and weighted Procustes algorithm and the results were compared to those obtained by using the 3D conformal transformation method.

We conclude by saying that the results contain quite small differences and the goal was achieved. In other words, Procrustes algorithm has a higher complexity, but if we are looking for a faster method, this is the better option.

REFERENCES

[1] Huaien Zeng, "Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix", SPRING OPEN JOURNAL EARTH, PLANET AND SPACE, 2015.

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

[3] Schonemann P. H. - A generalized solution of the orthogonal Procrustes problem. Psychometrika, vol. 31, no. 1, USA, 1966.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

[4] Grafarend E, Schaffrin B - Ausgleichungsrechnung in Linearen Modellen. B.I. Wissenschaftsverlag, Mannheim, 1993.
[5] Oniga, E., Savu, A., Negrila, A., – The Evaluation of CloudCompare Software in the Process of TLS Point Clouds Registration, GeoCAD 2016, Alba Iulia, Romania.
[6] S. R. Baig, F. U. Rehman, and M. J. Mughal, "Performance Comparison of DFT, Discrete Wavelet Packet and Wavelet Transforms in an OFDM Transceiver for Multipath Fading Channel,", 9th IEEE International Multitopic Conference, pp. 1-6, Dec. 2005.
[7] N. Ahmed, Joint Detection Strategies for Orthogonal Frequency Division Multiplexing, Dissertation for Master of Science, Rice University, Houston, Texas. pp. 1-51, Apr. 2000.