Research Article

Waldemar Odziemczyk*

Application of simulated annealing algorithm for 3D coordinate transformation problem solution

<https://doi.org/10.1515/geo-2020-0038>

received May 25, 2019; accepted February 6, 2020

Abstract: Transformation of spatial coordinates (3D) is a common computational task in photogrammetry, engineering geodesy, geographical information systems or computer vision. In the most frequently used variant, transformation of point coordinates requires knowledge of seven transformation parameters, of which three determine translation, another three rotation and one change in scale. As these parameters are commonly determined through iterative methods, it is essential to know their initial approximation. While determining approximate values of the parameters describing translation or scale change is relatively easy, determination of rotation requires more advanced methods. This study proposes an original, two-step procedure of estimating transformation parameters. In the initial step, a modified version of simulated annealing algorithm is used for identifying the approximate value of the rotation parameter. In the second stage, traditional least squares method is applied to obtain the most probable values of transformation parameters. The way the algorithm works was checked on two numerical examples. The computational experiments proved that proposed algorithm is efficient even in cases characterised by very disadvantageous configuration of common points.

Keywords: 3D coordinate transformation, Rodrigues matrix, quaternions, initial values, simulated annealing, metaheuristic method

1 Introduction

A three-dimensional coordinate transformation is commonly used in photogrammetry, engineering geodesy, geographical information systems or computer visualisations of spatial objects. It consists of converting the spatial coordinates from the original coordinate system to the target one. Similarity transformation is the most frequently used mathematical model, which determines seven parameters, of which three determine translation, three rotation and one change in scale. This method is also known as Helmert transformation. Although it was proposed in 1893 [1] (as a 2D transformation) it is still being analysed and improved [2,3]. The Helmert transformation of point i coordinates is described by equation (1):

$$b_i = \lambda R a_i + t \quad (1)$$

where $a_i = [x_i, y_i, z_i]^T$ is the position vector of a given point in the original system, $b_i = [X_i, Y_i, Z_i]^T$ is the position vector of the point in the target system, $t = [\Delta X, \Delta Y, \Delta Z]^T$ is the translation vector and λ defines scale parameter. R is a 3×3 rotation matrix. Its elements are the functions of three angles of rotation.

Usually, to determine the values of transformation parameters a group of common points is used, for which coordinates are known in both systems. Helmert transformation, which is the subject of this study, requires at least three such points. In some cases it is possible to unambiguously determine the seven transformation parameters on the basis of XYZ coordinates of two common points and one coordinate (e.g. Z) of a third point. In no event can the points be collinear.

There are many algorithms used to determine transformation parameters based on the coordinates of common points. They can be divided into two groups: analytical procedures and iterative numerical procedures. The first group covers the algorithms like Procrustes algorithm [4,5] proposed by Grafarend and Awange, or algorithm based on

* Corresponding author: Waldemar Odziemczyk, Chair of Engineering Geodesy and Control-Measuring Systems, Faculty of Geodesy and Cartography, Warsaw University of Technology, PL. Politechniki 1, 00-661 Warszawa, room. 304, Warsaw 00-661, Poland, e-mail: waldemar.odziemczyk@pw.edu.pl, tel: +48 22 234 6069

distribution of eigenvalues and eigenvectors [6,7]. Analytical algorithms are considered to be complex, therefore they are not customarily applied in practice. They do, however, have the advantage of being able to determine the definitive values of transformation parameters in a direct way. Moreover, they do not require the knowledge of approximated values of those parameters.

Iterative algorithms, being easier to compute, are more commonly used. They usually utilise linearisation of the dependencies binding the transformed coordinates with the transformation parameters. This method, however, requires the knowledge of approximated values of those parameters. While determining approximate values of the parameters describing translation or scale change is relatively easy, determination of rotation requires more advanced methods. This is particularly true for large angles of rotation. This category should include the algorithm utilising modified Gauss–Newton method [8], Pareto algorithm [9], algorithms analysed in reference [10] or algorithm using total least squares (TLS) method [11].

2 Parameterisation of rotation

Definition of parameters describing the rotation components is of particular importance in case of iterative algorithms. Euler angles (φ , ψ , θ) corresponding to each subsequent rotation about the axis of the coordinate system are customarily used. In photogrammetry (especially aerial photogrammetry) ω , φ , κ angles are used. The problematic aspect of using angles of rotation lies in the fact that their value is discontinued after exceeding 360° . Another common characteristic of the abovementioned solutions is the need to calculate trigonometric functions in order to identify elements corresponding to these angles in the rotation matrix. Admittedly, operational efficiency of currently produced computers allows for a slightly less restrictive approach to computational optimisation, nevertheless, in the case of iterative algorithms, by nature requiring multiple iterations of computational tasks, the question of cost/time of the numerical operations should be taken into consideration.

The solution utilising quaternions to describe rotation of the coordinate system is devoid of the mentioned shortcomings. William Rowan Hamilton, an Irish astronomer living in the eighteenth century, is the author of quaternion algebra [12]. Some contribution to the development of the idea of using quaternions to describe transformations in 3D space was made by Olinde Rodrigues, a French banker living in the same time

period [13]. Interesting details about this discovery can be found in reference [14].

The details on quaternion theory as well as its application are commonly known. An inquisitive reader may look into a book by Hanson [15]. In the later part of this paper, the author will narrow the scope of the study to the key aspects of applying quaternions to describe transformations of 3D space.

A quaternion is a four-element numerical structure, which when simplified can be expressed as the following equation:

$$Q = [a, b, c, d] \quad (2)$$

Its elements fulfil the following equation:

$$a^2 + b^2 + c^2 + d^2 = 1 \quad (3)$$

A rotation matrix corresponding to the quaternion (Rodrigues' matrix) can be defined as follows:

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2(bc - ad) & 2(ac + bd) \\ 2(bc + ad) & a^2 - b^2 + c^2 - d^2 & 2(-ab + cd) \\ 2(-ac + bd) & 2(ab + cd) & a^2 - b^2 - c^2 + d^2 \end{bmatrix} \quad (4)$$

Each of the four quaternion elements defining Rodrigues' matrix may take values in the range of $\langle -1; 1 \rangle$ (meaningful when quaternion elements are acquired with non-algebraic method). Important characteristic of the rotation matrix (4) in relation to quaternion is the symmetry with respect to zero. It can be easily checked that quaternions Q and $-Q$ return the same rotation matrix.

It has to be noted that in contrast to the rotation matrix created based on angles, in order to calculate all nine elements of rotation matrix it is not necessary to carry out numerically cost-heavy trigonometric calculations. It is a gain in spite of the fact that we have to deal with one more rotation parameter and respect constraint (3). Another important advantage of Rodrigues' matrix is linked to the fact that the functions describing the variability of individual quaternion elements in rotation function are continuous (there is no problem of exceeding 360° as it was the case for angles).

The above-mentioned advantages resulted in Rodrigues' idea becoming widely used in various, often distant scientific and technological disciplines like computer vision [16], astronomy [17,18] or chemistry [19,20].

3 Simulated annealing

Dissemination of computer calculation methods and the related decrease in the cost of computational operations

caused a shift in the way we look at estimation tasks and the algorithms used for those purposes. This new approach resulted in the appearance of new methods from the Monte Carlo or Metaheuristics Family, where numerous repetitions of a computation sequence are required to obtain the results.

Simulated annealing (SA), also known as Monte Carlo annealing, probabilistic hill climbing or stochastic relaxation, is an algorithm which belongs to metaheuristic methods. The idea behind SA comes from thermodynamics and reflects the process of solidification of a liquid into a crystalline solid. As the cooling process progresses molecule mobility decreases. If the cooling rate is sufficiently slow, the molecules can achieve the state of mutual order corresponding to the lowest energy state (e.g. create crystal lattice).

SA algorithm was first published in the study by Metropolis et al. [21]. Since then it has been refined by other scientists, like Kirkpatrick et al. [22]. More information on SA algorithm can be found in reference [23]. The most interesting characteristic of annealing process reflected in the algorithm structure is the ability to avoid local minima corresponding to pseudocrystalline state with energy level higher than minimal. However, this will only take place if the decrease in temperature is sufficiently slow. The idea of solution seeking for one-dimensional objective function is illustrated in Figure 1.

The properties of SA algorithm, especially its ability to solve complex computational problems with local minima, led to a situation, where it is used in fields sometimes very distant from thermodynamics. The famous travelling salesman problem is frequently used as an example. Metaheuristic methods have been used to

solve some optimisation problems in geodesy and surveying techniques e.g. references [24–26] but uses of SA in this field are limited. They include research on geodetic network design [27,28] or terrestrial laser scanning survey design [29]. Although this publication concerns an issue (coordinate transformation) of a much broader spectrum of applications, it proposes a new use for SA algorithm in the field of surveying techniques.

SA is an iterative algorithm, in which constant temperature change of a cooling liquid is replaced by incremented changes realised in subsequent iterations. Its use for solving estimation tasks requires a definition of several essential elements:

1. Objective function corresponding to the molecular energy level during annealing process that will be minimised
2. Cooling scheme comprised assumed initial temperature and dependency defining temperature drop after each iteration
3. Molecule movement model corresponding to the actual temperature
4. Termination criteria for the iterative process. The condition can be formulated on the basis of:
 - final temperature (minimal),
 - acceptable value of the objective function,
 - range of molecule movement that usually corresponds to estimated model parameter changes defining the objective function.

By defining: x_i – vector of the current model parameters in the i th iteration of the estimation task, $f(x_i)$ – objective function defined for the current parameters, and Δx_i – change in parameter vector in the i th iteration, the SA algorithm can be presented using the following scheme:

Figure 1: Idea of solution seeking for one-dimensional objective function.

Figure 2: Flowchart of the SA method.

3.1 Objective function

The objective of the algorithm is to find the global minimum. Therefore, the objective function $f(x_i)$ must provide an answer whether the current solution (x_i vector) is better than the previous one and thereby if it is closer to the final solution. An equation assuming the maximal value for the end condition can also be used. In that case changing the sign brings the equation down to standard case (searching for minimum).

notion. In such situation it is more important to define the function of temperature change. Various schemata are used in SA algorithms, but they always take the form of:

$$T = f(T_0, i) \quad (5)$$

where T_0 is the initial temperature and i is the iteration number. The fact of a tight link between current temperature and current iteration number makes it possible to replace the temperature parameter with iteration number in some solutions.

3.2 Cooling scheme

Temperature as a parameter of the SA algorithm stems from physical analogies. In case of optimisation tasks not related to liquid solidification it is an abstract

3.3 Molecule movement model

The most distinguishing feature of the SA method is the so-called molecule movement model connected with the way the new solution is obtained. It is generally defined

by two elements. The first of them is the way each current (subsequent) solution is generated, which is defined by equation (6)

$$x_{i+1} = x_i + \Delta x_i \quad (6)$$

Δx_i vector is randomly generated and usually normal distribution $N(0, \sigma)$ is used. Current standard deviation of this distribution σ_i is the function of initial value σ_0 and temperature (or iteration number) in accordance to the equation:

$$\sigma_i = \sigma_0 \beta^t \quad (7)$$

or

$$\sigma_i = \sigma_0 \beta^{T(t)} \quad (8)$$

β coefficient corresponds to the cooling speed and assumes values in the range (0,1) and t defines time or, alternatively, iteration number. The second key element of the molecule movement model is the criterion for accepting a new solution. Decision making process on acceptance of the new solution as the current one (potentially the best) consists of three phases. Firstly, it is checked whether the obtained solution ($x_i + \Delta x_i$) belongs to the task field – in other words, whether it fulfils the formal requirements of the task being solved. Secondly, the value of the objective function is analysed:

$$x_{i+1} = \begin{cases} x_i + \Delta x_i & \text{if } f(x_i + \Delta x_i) < f(x_i) \\ x_i + \Delta x_i & \text{if } f(x_i + \Delta x_i) \geq f(x_i) \text{ with probability } p \\ x_i & \text{otherwise} \end{cases} \quad (9)$$

If the obtained value of the objective function is lower than the current value, the new solution ($x_i = x_i + \Delta x_i$) is accepted unconditionally. Otherwise the obtained vector, even though it corresponds to a worse solution from the objective function's perspective, is accepted with a determined probability p . The most advanced method – based in the thermodynamic origins of the procedure – is the application of Boltzmann distribution. The value of p is then defined by equation (10)

$$p = e^{-\frac{f(x_{i+1}) - f(x_i)}{T_i}} \quad (10)$$

This method gives relatively high p values, which on the one hand provides a higher guarantee of finding the global minimum, but on the other the frequent acceptance of a worse solution considerably slows down the iteration process. A simpler solution would be to assume constant value for p – most commonly in range from 0.001 to 0.2 (like in ref. [28]). In the simplest tasks with uncomplicated spatial distribution of the objective function this element of the algorithm can be omitted, which corresponds to the original version proposed by Metropolis team [21].

3.4 Termination criteria for the iterative process

Termination of the solution seeking process may be linked to acquiring a specific temperature value, which corresponds to a certain number of carried out iterations. Another option may be reaching a satisfactory value for the objective function. In both cases it is essential to determine the critical value. It depends on the required accuracy of the solution. In particular case, SA method may be used to determine an approximate solution, which will subsequently allow for application of other methods utilising e.g. linearisation of the functional dependencies describing the given task. In this case a significantly lower number of iterations is required in order to obtain a satisfactory result.

4 Concept of a hybrid algorithm

The most important advantage of the SA algorithm is the ability to find a solution (global minimum) for a task where local minima are present. Slow cooling rate and non-zero probability of accepting worse solution are crucial in such a case. It is not without significance that the starting point of the procedure has no significant impact on the speed and possibilities of finding a solution. This means that in practice there is no need to know the approximate solution and that the starting point of the process can be assumed freely. These advantages lose their significance in the last stage of the process of finding a solution, when the current solution is the close vicinity of the sought minimum for which the objective function changes in a regular manner. It results from the fact that, in contrast to gradient methods, SA procedure uses only the value of the objective function ignoring completely the character of its distribution. For the same reason and due to the stochastic character of the solution search process itself, when using SA method such solution will always be an approximation. By proper control of termination criteria it is only possible to cause the difference between the obtained result and the global minimum to be negligible for the given task.

For the above reasons, a concept of a hybrid algorithm was born. It is comprised of two extremely distinct algorithms executed sequentially depending on the phase of solution seeking process. First the SA algorithm is used. Its ultimate goal is to find an approximation for the quaternion parameter values that will make it possible to apply the chosen gradient method in the next stage. In this case, it is the widely known least squares (LS) method using linear dependency between

the coordinates of the common points and the seven parameters of 3D transformation.

Another advantage of LS lies in the consistent evaluation or result accuracy estimation – both in transformation parameters as well as its results (transformed coordinates). As the LS does not include mechanisms of dealing with gross errors, in practical applications it should be supported by the quality control routine. It can be based on LS itself [30]. It is also possible to apply another robust method such as the previously mentioned TLS [11].

5 Numerical realisation of the hybrid algorithm

Spatial coordinate transformation is described by equation (1). Assuming that the a_i vector is a position vector (related to the origin of the coordinate system) the t translation vector describes the translation of the origin point of the original coordinate system. Therefore, the t vector corresponds to the position of that point in the target system. This situation is not favourable, because it is not possible to determine the approximate t components based on the coordinates of the common points – especially if the points lie within a considerable distance from the origin of the coordinate system. Consequently, in practical realisations the concept of a base point for a group of common points is utilised. In this case equation (1) takes the form (11)

$$b_i = \lambda R(a_i - a_0) + b_0 + u \quad (11)$$

where a_0 and b_0 vectors accordingly represent the position of the base point in the original and target system. The $u = [dx, dy, dz]$ vector is a residual vector of the translation. The centre of gravity of the common points or a selected point belonging to this group can be used as the base point. Such approach makes it possible to easily determine all transformation parameters excluding the rotation components.

$$u_{\text{prox}} = [0, 0, 0] \quad \lambda_{\text{prox}} = \frac{\sum |b_i - b_0|}{\sum |a_i - a_0|} \quad (12)$$

5.1 SA used for determining approximated rotation parameters

The following transformed (11) equation is used to determine approximated rotation parameters

$$(b_i - b_0) = \lambda R(a_i - a_0) \quad (13)$$

which is used to formulate the objective function

$$f_{\text{obj}} = \sum \|(b_i - b_0) - \lambda_{\text{prox}} R(a_i - a_0)\|^2 \quad (14)$$

The values sought are quaternion elements $Q = [q_1, q_2, q_3, q_4]$ corresponding to rotation matrix R . As those elements satisfy equation (3) the search area takes on the geometric form of a four-dimensional sphere. By transforming equation (3) it is possible to eliminate one of the elements, which makes it possible to narrow the search to only three of them (e.g. q_2, q_3 and q_4) and the fourth can be obtained through the following equation:

$$q_1 = \sqrt{q_2^2 + q_3^2 + q_4^2} \quad (15)$$

where q_2, q_3 and q_4 elements must satisfy:

$$q_2^2 + q_3^2 + q_4^2 \leq 1 \quad (16)$$

which reduces the search area to a three-dimensional ball. Although two values of q_1 satisfy equation (3) for the three elements:

$$q_1 = \sqrt{q_2^2 + q_3^2 + q_4^2} \quad \text{and} \quad q_1 = -\sqrt{q_2^2 + q_3^2 + q_4^2}$$

but since the rotation matrix is the same for quaternion $Q_1 = [q_1, q_2, q_3, q_4]$ and $Q_2 = [-q_1, -q_2, -q_3, -q_4]$, the search can be limited to $q_1 \geq 0$.

Because the sought parameters are three quaternion elements q_2, q_3 and q_4 , a temperature equivalent in the form of σ parameter is assumed, which corresponds to the sought quaternion components.

The course of the process of finding a solution is defined as the following values:

$\sigma_0 = 1.0$ – initial value of the standard deviation (temperature equivalent)

$\beta = 0.99$ – standard deviation decreasing rate (cooling coefficient)

$p = 0$ – probability of accepting a worse solution

$\sigma_{\text{end}} \leq 0.1$ – termination condition for SA procedure

A relatively quick standard deviation decreasing rate was assumed, which is linked to a low diversity of objective function distribution in the search area. For the same reason it was decided that the original version of the algorithm will be used, eliminating the possibility to accept an inferior solution ($p = 0$). Early termination ($\sigma_{\text{end}} = 0.1$) of the SA procedure is tied to the role that was assigned to this part of the algorithm. Further calculations can be realised using a different method assuming the knowledge of a suitable approximation of the final solution.

5.2 Determining the final values for the seven transformation parameters using least squares method

Knowledge of a suitable approximation of the final values of the transformation parameters opens new possibilities to choose one of many estimation methods. It is possible, for example, to take into account varied accuracy of the coordinates of the common points e.g. ref. [31]. In cases where there is a risk of gross errors occurring in the observations, one of robust estimation methods (e.g. TLS) can be applied.

As it was previously mentioned, LS method was used in the second stage of the hybrid algorithm. As the result of linearisation (11) for any of the common point i it is possible to match a segment of three correction equations. By assuming notation from ref. [8] it is possible to formulate them as equation (17)

$$V_i = B_i \delta x - l_i \quad (17)$$

where $V_i = [V_{xi}, V_{yi}, V_{zi}]^T$ is the correction vector to the coordinates.

δx – defines the unknowns vector $\delta x = [dq_1, dq_2, dq_3, dq_4, dx, dy, dz, d\lambda]^T$ – corrections to the approximated values of transformation parameters, B is a 3×8 coefficient matrix, created from individual coordinate derivatives (components of b_i vector in equation (11)) of individual unknowns, and $l_i = [l_x, l_y, l_z]$ is the vector of residuals:

$$l_i = b_i - (\lambda R(a_i - a_0) + b_0 + u)$$

where b_i determines the coordinates of the i common point in the target system, λ – current approximation of scale change coefficient, u – current approximation of translation parameters and rotation matrix R was acquired using equation (4) based on the current quaternion component values $[q_1, q_2, q_3, q_4]$.

Having $n \geq 3$ common points makes it possible to match a correction equation set in the form of

$$V = B \delta x - l \quad (18)$$

$$\text{where } V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix}, B = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix}, l = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}.$$

To calculate the vector of the unknowns it is necessary to include equation (3) corresponding to the conditional matrix (19)

$$W = [q_1, q_2, q_3, q_4, 0, 0, 0, 0] \quad (19)$$

Unknowns δx will be obtained by solving the system of normal equation (20)

$$\delta x = N^{-1}L \quad (20)$$

$$\text{where } N = \begin{bmatrix} B^T B & W^T \\ W & 0 \end{bmatrix} \quad L = \begin{bmatrix} B^T l \\ 0 \end{bmatrix}.$$

The obtained unknowns are then used for correcting the previous approximation of transformation parameters $x_{i+1} = x_i + \delta x$. Afterwards the next iteration is performed. The iteration process finishes when the vector norm of the unknowns approaches zero within the assumed distance ε :

$$|\delta x| \leq \varepsilon \quad (21)$$

However, one must remember that the parameter ε does not define the precision at which the unknowns are determined but only the residual effect of nonlinearity of the transformed coordinates in regard to the linear approximation of those functions.

6 Numerical test

The way the proposed algorithm operates will be illustrated on two numerical examples.

The first test object consists of eight points. Four of them (1–4) are common (reference) points and four others (5–8) are control points. Control points are not used to calculate transformation parameters. Figure 3 shows distribution of both groups of points on XOY plane.

Figure 3: Test object 1 points placement.

Table 1: Common and control points coordinates – example 1

Pt	System 1 [m]			System 2 [m]		
	x_1	y_1	z_1	x_2	y_2	z_2
1	21.374	98.615	112.938	3045.691	2835.122	39.727
2	134.752	42.840	0.428	3028.006	2666.808	37.048
3	36.630	-29.441	-22.770	3130.473	2681.592	105.495
4	-48.285	37.710	86.821	3125.372	2835.167	93.628
5	94.863	-11.198	86.676	3107.415	2725.879	-9.495
6	15.988	52.207	-48.539	3062.508	2711.755	152.780
7	79.922	62.372	132.735	3056.076	2796.734	-19.903
8	25.632	98.525	21.368	3028.778	2771.709	103.788

Spatial coordinates of all points are presented in Table 1.

Solution seeking process for the SA method is best illustrated using the so-called Markov chain. It consists of a sequence of states (in this case, sets of q_2 , q_3 , q_4 values) corresponding to subsequent, at the time best, objective function values. Due to the stochastic character of the solution seeking process every time the procedure is started it takes different way to approach the final solution. For that reason it is not possible to obtain two identical Markov chains by two independent starts of the SA procedure. Analysing Markov chains of SA is

essential while optimising the key parameters of the procedure in the particular task.

To better illustrate how the SA algorithm is working out approximate rotation parameters the procedure was repeated three times. The Markov chains describing course of the solution seeking process for the test 1 were juxtaposed in Table 2. In the case of SA method, the total number of iterations was constant, as it is dependent on the initial value of the standard deviation value for quaternion elements – σ_0 , standard deviation decreasing rate – β and termination condition for the iterative process. For the values assumed here (previous chapter) it was equal to 230.

Table 2: Course of solution seeking process – example 1

Trial no.	SA					LS	
	it.	q_2	q_3	q_4	f_{obj}	it.	$ \delta x $
1	2	-0.61014	0.29932	0.26327	3.0×10^4	1	6.8×10^3
	6	-0.06746	0.80368	0.46698	2.9×10^4	2	1.5×10^3
	17	-0.33752	0.67297	0.41731	1.4×10^4	3	2.5×10^1
	55	-0.33125	0.77385	0.46125	6.5×10^3	4	3.4×10^{-3}
	70	-0.42668	0.79239	0.29698	9.5×10^2		
2	16	-0.61091	0.22384	-0.30276	3.5×10^4	1	5.3×10^3
	17	-0.56810	0.75727	0.15112	2.3×10^3	2	3.8×10^2
	167	-0.44612	0.82186	0.30560	4.3×10^2	3	1.8×10
	219	-0.42750	0.81745	0.30802	3.4×10^2	4	1.8×10^{-4}
3	2	-0.46998	0.42775	-0.35724	2.9×10^4	1	6.6×10^3
	20	-0.55534	0.52513	0.04127	1.3×10^4	2	1.4×10^3
	21	-0.26761	0.89556	0.12314	9.2×10^3	3	3.5×10^1
	71	-0.41424	0.86637	0.01111	6.4×10^3	4	1.9×10^{-2}
	126	-0.57854	0.67501	0.37977	4.8×10^3		
	147	-0.38718	0.79575	0.37317	2.2×10^3		
	149	-0.46441	0.85084	0.15977	1.9×10^3		
	178	-0.43432	0.81917	0.16067	1.4×10^3		
	190	-0.37388	0.84222	0.29749	1.4×10^3		
	196	-0.45269	0.78088	0.28901	6.0×10^2		

Note: Values in bold are the final values of the first stage of the calculation process (SA section) or starting values for the second stage (LS section). The starting values of the LS stage can be treated as a rate of the SA stage.

Table 3: Residuals on common points and control points for LS solution – example 1

Pt	Δx [m]	Δy [m]	Δz [m]
1	0.0069	-0.0043	0.0046
2	-0.0054	-0.0054	-0.0031
3	0.0001	0.0052	-0.0005
4	-0.0016	0.0045	-0.0010
5	-0.0105	-0.0055	-0.0074
6	0.0036	-0.0097	0.0011
7	-0.0092	-0.0048	-0.0052
8	0.0139	-0.0035	0.0065

Note: Values in bold are the final values of the first stage of the calculation process (SA section) or starting values for the second stage (LS section). The starting values of the LS stage can be treated as a rate of the SA stage.

The *it. nr* column presents the iteration numbers for which the SA procedure found the better solution.

The last (final) set of quaternion parameters set is marked in bold. As it can be seen, each of the three trials delivered quite similar parameters values. Differences in q_2 , q_3 and q_4 don't exceed 10% of their values.

The unknowns vector norm of the LS first iteration (also marked in bold) can be treated as an assessment of the SA result. The final solution being the result of application of LS was always identical. Residuals on common (bold) and control points are presented in Table 3.

Due to proper number and distribution of the first test object common points, finding transformation parameters is a well-defined and relatively easy task. To test the effectiveness of the SA algorithm in a more demanding case another test object was invented. The second test object consists of three common points and three control points. The placement of common points was selected in a very disadvantageous way. They form an elongated triangle. The layout of all the points is presented in Figure 4 and their coordinates in Table 4. All the measures in Figure 4 are in meters.

The coordinates in system 2 were obtained by rotation and translation of the established points in system 1 using equation (1), with rotation matrix R created on the basis of commonly known equations used in photogrammetry presented *i.a.* in reference [32] (equation (2.3)) for rotation angles $\omega = -16^\circ$, $\varphi = 195^\circ$, $\kappa = 45^\circ$ respectively. A translation vector $t = [500.00, 500.00, 200.00]$ was set. Coordinates in system 2 were distorted with errors generated based on normal distribution with standard deviation $\sigma_{xyz} = \pm 0.02$ m.

As in the previous test the procedure was repeated three times. The Markov chains describing course of the solution seeking process for test 2 were juxtaposed in Table 5.

Figure 4: Test object 2 points placement.**Table 4:** Common and control points coordinates – example 2

Pt	System 1 [m]			System 2 [m]		
	x_1	y_1	z_1	x_2	y_2	z_2
1	10.000	10.000	0.000	486.664	499.534	195.260
2	150.000	180.000	0.000	280.933	471.031	122.094
3	75.000	96.000	0.000	386.592	481.140	159.728
4	74.000	99.000	12.000	381.316	478.475	170.616
5	157.000	174.000	10.000	276.691	480.367	130.959
6	142.000	184.000	5.000	282.131	462.747	128.003

Residuals on common and control points are presented in Table 6.

Despite the significant disadvantage resulting from the placement of common points the SA procedure was able to deliver angular transformation parameters that were good enough, so that the subsequently introduced LS procedure was able to provide the exact solution after a few iterations. The Markov chains (columns q_2 , q_3 and q_4) presented in Table 5 point to a high diversity in the way the solution was reached. Relatively big differences present themselves in the final quaternion component values. It is especially true of q_3 component. The previous test and other experiments carried out by the author that are not presented in this paper indicate that these discrepancies are closely linked to point set geometry and can be reduced to small extent by increasing the precision of the SA method (e.g. by slowing down the standard deviation decreasing rate or delaying the termination of solution seeking process by decreasing value of σ_{end}).

Disadvantageous geometry of the common points is also responsible for large residuals on the control points shown in Table 6.

It is worth remembering that even though all tests presented in Table 5 were successful, due to the random

Table 5: Course of solution seeking process – example 2

Trial no.	SA					LS	
	it.	q_2	q_3	q_4	f_{obj}	it.	$ \delta x $
1	11	0.25970	−0.87668	0.11922	2.3×10^4	1	1.1×10^4
	33	0.30390	−0.27214	0.69446	2.0×10^4	2	3.1×10^3
	42	0.07751	0.00543	0.95193	5.8×10^3	3	9.0×10^1
	66	−0.12058	0.17849	0.88496	2.1×10^3	4	2.5×10^{-1}
	83	−0.47744	0.31399	0.75136	1.7×10^3	5	1.8×10^{-4}
	141	−0.43279	0.46618	0.69776	3.1×10^2		
2	202	−0.39075	0.36523	0.74209	8.5×10^1		
	3	−0.21799	0.08426	0.96925	9.2×10^3	1	8.9×10^3
	13	−0.27265	0.26343	0.82109	3.5×10^2	2	1.4×10^3
	131	−0.08587	−0.26250	0.90688	1.4×10^2	3	6.7×10^1
3						4	1.1×10^{-1}
						5	8.0×10^{-5}
	3	−0.39293	0.06396	0.57403	1.9×10^4	1	1.1×10^4
	8	−0.64238	0.73165	0.13101	2.9×10^3	2	2.7×10^3
	22	−0.46757	0.41313	0.61078	2.2×10^3	3	1.4×10^2
	57	−0.35814	0.39209	0.75010	6.5×10^1	4	3.2×10^{-1}
						5	2.0×10^{-4}

Note: Values in bold are the final values of the first stage of the calculation process (SA section) or starting values for the second stage (LS section). The starting values of the LS stage can be treated as a rate of the SA stage.

Table 6: Residuals on common and control points for LS solution – example 2

Pt	Δx [m]	Δy [m]	Δz [m]
1	0.0052	0.0012	0.0018
2	0.0050	0.0007	0.0018
3	−0.0102	−0.0019	−0.0036
4	−0.0258	−0.0669	−0.0051
5	−0.0350	−0.0535	0.1078
6	0.0188	−0.0404	−0.0588

Note: Values in bold are the final values of the first stage of the calculation process (SA section) or starting values for the second stage (LS section). The starting values of the LS stage can be treated as a rate of the SA stage.

nature of the SA procedure itself one cannot eliminate the possibility of a failure, which will be understood as obtaining a solution distant from the final one. The probability of a failure increases the more the SA parameters are optimised for processing speed. In such case an attempt to continue the process using LS will lead to a situation, where two subsequent iterations i and $i + 1$ will produce an inequality $|\delta x_{i+1}| \geq |\delta x_i|$, which will be interpreted as an error and will terminate the LS iteration process. If SA parameters have been selected correctly, a repeated attempt to solve the task is the best countermeasure.

7 Conclusions

The performed computational tests confirmed the predicted effectiveness of the SA procedure in solving a fundamental 3D transformation problem, namely to determine approximate values of parameters defining the change in orientation. While the first numerical test used to check and demonstrate the algorithm using four regularly distributed common points can be considered as easy, the second one utilised an extremely disadvantageous placement of three common points. Poor geometry of common points caused the distribution of the objective function in the space of the sought parameters to be less clear. In result, quaternion elements being rotation parameters obtained as a result of SA procedure in the second test were much more differentiated among subsequent trials.

It should be noted that the length of the Markov chains presented in Tables 2 and 5 did not exceed 10 nodes for 230 iterations. Such nature of the process results from the used parameters which caused the SA procedure to progress relatively quickly ($\beta = 0.99$) and finish relatively early ($\sigma_{\text{end}} = 0.1$). The selection of parameters proved sufficient even in the case of unfavourable geometry. The unfavourable geometry of the common points in test 2 also caused that, on average, one more iteration of LS were needed to obtain the final solution.

The author is aware of the shortcomings of the LS method, of which the most often referenced one is insufficient robustness in case of presence of gross errors. That is why its application in the described algorithm is of demonstrative value only. Nothing prevents software used for real 3D tasks from utilising a robust estimation method e.g. TLS.

Due to the high probable diversity of cases as well as high dependency of the course of the SA procedure and the precision of the delivered results on its control parameters, one should provide an advanced user the possibility to modify these parameters in order to optimise the course of calculations for specific cases.

In the case of a numerous common points their number can be limited in order to accelerate the SA algorithm operation process (only approximated values are needed).

Parameters of SA procedure were optimised for potentially least optimal case and were assumed as constant. It seems that in case of more advantageous distribution of common points and larger number of them it is possible to accelerate the solution seeking process. Formulating rules for customisation of SA parameters with regard to the task's characteristic needs requires further research.

References

- [1] Helmert FR. Die Europäische Langengradmessung in 52 Grad Breite von Greenwich bis Warchau. I. Heft. Veroff. des Konigl. Berlin: Preussischen Geodätischen Instituts und Centralbureaus der Int. Erdmessung; 1893. p. 47–50. (in German).
- [2] Teunissen PJ. The non-linear 2D symmetric Helmert transformation: an exact non-linear least-squares solution. *B Geo.* 1988;62(1):1–16.
- [3] Watson GA. Computing Helmert transformations. *J Comput Appl Math.* 2006;197(2):387–94. doi: 10.1016/j.cam.2005.06.047.
- [4] Grafarend EW, Awange JL. Nonlinear analysis of the three-dimensional datum transformation [conformal group $\mathbb{C}^7(3)$]. *J Geodesy.* 2003;77(1–2):66–76. doi: 10.1007/s00190-002-0299-9.
- [5] Păun CD, Oniga VO, Dragomir PI. Three-dimensional transformation of coordinate systems using nonlinear analysis – Procrustes algorithm. *Int J Eng Sci Res Technol.* 2017;6(2):355–63. doi: 10.5281/zenodo.291839.
- [6] Zeng H. Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix. *Earth Planets Space.* 2015;67(1):105. doi: 10.1186/s40623-015-0263-6.
- [7] Hanson AJ. Extensions and Exact Solutions to the Quaternion-Based RMSD Problem. arXiv preprint, 2018, arXiv:1804.03528.
- [8] Zeng H, Yi Q. Quaternion-based iterative solution of three-dimensional coordinate transformation problem. *JCP.* 2011;6(7):1361–8.
- [9] Paláncz B, Awange JL, Völgyesi L. Pareto optimality solution of the Gauss–Helmert model. *Acta Geod Geophys.* 2013;48(3):293–304. doi: 10.1007/s40328-013-0027-3.
- [10] El-Habiby MM, Gao Y, Sideris MG. Comparison and analysis of non-linear least squares methods for 3-D coordinates transformation. *Surv Rev.* 2009;41(311):26–43. doi: 10.1179/003962608X389988.
- [11] Akyilmaz O. Total least-squares solution of coordinate transformation. *Surv Rev.* 2007;39(303):68–80. doi: 10.1179/003962607X165005.
- [12] Hamilton WR. On quaternions; or a new system of imaginaries in algebra. *Philos Mag.* 1844;25:489–95.
- [13] Rodrigues O. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J Math Pure Appl.* 1840;1re série, tome 5:380–440. (in French).
- [14] Altmann SL. Hamilton Rodrigues and the quaternion scandal. *Math Mag.* 1989;62(5):291–308. doi: 10.1080/0025570X.1989.11977459.
- [15] Hanson AJ. Visualizing Quaternions. Morgan-Kaufmann/Elsevier; 2006.
- [16] Mukundan R. Quaternions: from classical mechanics to computer graphics, and beyond. Proceedings of the 7th Asian Technology conference in Mathematics. 2002. p. 97–105.
- [17] Arribas M, Elipse A, Palacios M. Quaternions and the rotation of a rigid body. *Celest Mech Dyn Astr.* 2006;96(3–4):239–51. doi: 10.1007/s10569-006-9037-6.
- [18] Shuster MD, Natanson GA. Quaternion computation from a geometric point of view. *J Astronaut Sci.* 1993;41(4):545–56.
- [19] Coutsias EA, Seok C, Dill KA. Using quaternions to calculate RMSD. *J Comput Chem.* 2004;25(15):1849–57. doi: 10.1002/jcc.20110.
- [20] Popov P, Grudinin S. Rapid determination of RMSDs corresponding to macromolecular rigid body motions. *J Comput Chem.* 2014;35(12):950–6. doi: 10.1002/jcc.23569.
- [21] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. *J Chem Phys.* 1953;21(6):1087–92. doi: 10.1063/1.1699114.
- [22] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. *Science.* 1983;220(4598):671–80. doi: 10.1126/science.220.4598.671.
- [23] van Laarhoven PJM, Aarts EHL. Simulated annealing: Theory and applications. Dordrecht, The Netherlands: Reidel; 1987.
- [24] Yetkin M. Metaheuristic optimisation approach for designing reliable and robust geodetic networks. *Surv Rev.* 2013;45(329):136–40.
- [25] Yetkin M, Berber M. Implementation of robust estimation in GPS networks using the artificial bee colony algorithm. *Earth Sci Inf.* 2014;7(1):39–46. doi: 10.1007/s12145-013-0131-5.
- [26] Yetkin M. Application of robust estimation in geodesy using the harmony search algorithm. *J Spat Sci.* 2018;63(1):63–73. doi: 10.1080/14498596.2017.1341856.

- [27] Berné JL, Baselga S. First-order design of geodetic networks using the simulated annealing method. *J Geodesy*. 2004;78(1–2):47–54. doi: 10.1007/s00190-003-0365-y.
- [28] Baselga S. Second order design of geodetic networks by the simulated annealing method. *J Surv Eng*. 2011;137(4):167–73. doi: 10.1061/(ASCE)SU.1943-5428.0000053.
- [29] Jia F, Lichti D. Comparison a of simulated annealing, genetic algorithm and particle swarm optimization in optimal first-order design of indoor TLS networks. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci*. 2017;IV-2/W4:75–82. doi: 10.5194/isprs-annals-IV-2-W4-75-2017.
- [30] Prószyński W. Measuring the robustness potential of the least-squares estimation: geodetic illustration. *J Geodesy*. 1997;71(10):652–9. doi: 10.1007/s001900050132.
- [31] Kwaśniak M, Prószyński W. The stochastic approach to coordinate transformation for the needs of engineering surveys. *Geodezja i Kartogr*. 1998;47(3–4):267–80.
- [32] Schut GH. An Introduction to Analytical Strip-Triangulation with a Fortran Program. Vol. 1. National Research Council of Canada Report, No. NRC; 1973. p. 3148.