

Comparison of a Point Cloud to a 3D Surface

Eugene Vladimirovich Popov

Engineering Geometry and Computer Graphics Chair

Nizhegorodsky State Architectural and Civil Engineering University, Nizhny Novgorod, Russia

popov_eugene@list.ru

Abstract

This paper deals with the study of fitting a set of unorganized points to a polygonal surface. This problem usually arises when comparing two data sets specified in two different co-ordinate systems. The developed approach uses Principal Component Analysis (PCA) and Stretched grid method (SGM) to substitute a non-linear problem solution with several linear steps. The squared distance (SD) is a general criterion to control the process of convergence of a set of points to a target surface. The described numerical experiment concerns the remote measurement of a large-scale aerial in the form of a frame with a parabolic shape. The experiment shows that the fitting process of a point cloud to a target surface converges in several linear steps. The method is applicable to the geometry measurement of large-scale objects remotely.

Keywords: *Non-contact measurement, point clouds comparison, computer visualization*

1. INTRODUCTION

The geometry measurement of large-scale objects in the industry is very acute. This problem reduces to the comparison a 3D point set given by remote measurement to a continuous theoretical surface. We can classify it as a point-to-surface (PTS) problem. Usually one can treat such comparison as superposition that requires not less than three reference points. However, reference points can be either unknown or meaningless for some classes of product. In this case, the problem comes to a comparison of two geometric objects in 3D space according to a given criterion of optimality. That is, the unknown parameters of the 3D transformation such as translation and rotation are a subject to be found according to objects optimal matching.

All the algorithms of two 3D sets comparison can be classified in the following way:

1. ICP- algorithm (iterative closest point algorithm). The basis of ICP-algorithm is the assumption that two objects have common area where they coincide well enough. It means that in the common, for both of them, area each point of one object has a corresponding point of another object. The basics of the ICP-algorithm were described in work [1]. The main disadvantages of ICP-algorithm are the following:

- the computational complexity of the closest points finding is $O(mN_1N_2)$ where m - number of iterations, N_1 - number of the first object points, N_2 - number of the second object points [2];
- strong dependence on the given initial approximation;
- strong dependence on the density of point clouds;
- the method requires the existence of a large overlap region where the points of one cloud correspond to the points of another cloud.

Recently, many variants of the original ICP approach have been proposed, the most important of which are the following:

- work [3] describes the genetic algorithm of finding the most successful initial approximation which is input data to ICP-algorithm;
- work [4] is also dedicated to the ICP-algorithm modification based on the k-d trees, which allows minimization of the computational complexity to $O(mN_1 \log N_2)$;
- in works [5], [6] algorithms to improve the accuracy and reliability of the ICP-algorithm by imposing certain restrictions of the input data are proposed.

2. Methods based on curvature maps.

This class of methods requires the knowledge of the curvature of the surface given by the point cloud. The algorithm was described in work [7]. The disadvantage of this method is a strong dependence on the point cloud density because it affects the accuracy of the curvature calculation.

3. Other methods.

Work [8] describes the algorithm that does not require the approach of initial data. This algorithm can use a free-form surface; however, it has a very low speed.

The authors of work [9] improved the method of the steepest descent optimization. The disadvantage of the approach is the quadratic computational complexity.

Delaunay triangulation algorithm in combination with Nelder-Mead method are described in works [4] and [10]. The algorithm assumes that the surfaces are single-valued. This algorithm as well as ICP-algorithm depends on the given initial approximation.

The authors of [11] proposed the algorithm based on the least square method. The algorithm requirement is that point clouds have a significant overlap area.

In work [12] the algorithm based on step by step geometric transformation of the point cloud is formulated. The disadvantage of this algorithm is the lack of mathematical rigorousness.

Nowadays there are two trends in the surface superpose problem solution. The first group of methods limits the initial data therefore they work fast. The second group is more general but has a large computational complexity. Hence, we need more algorithms for the comparison of two data sets.

2. INITIAL BACKGROUND

The initial assumption is that there are two sets: $P: P_i(x_i, y_i, z_i)$ - point cloud and Σ a continuous 3D surface (see Fig.1) specified on a bounded domain $D \subset R^3$. It is required to find such Ω_{opt} transformation amongst all possible Ω 3D transformations so that the set $\Omega_{opt}(P)$ could be closest to the surface Σ according to the given distance function ρ . That is

$$\sum_{i=1}^N \rho(\Sigma, \Omega_{opr}(P_i)) = \min_{\Omega} \sum_{i=1}^N \rho(\Sigma, \Omega(P_i)) \quad (1)$$

where $\rho(\Sigma, X)$ - the distance function from the point X to the surface Σ .

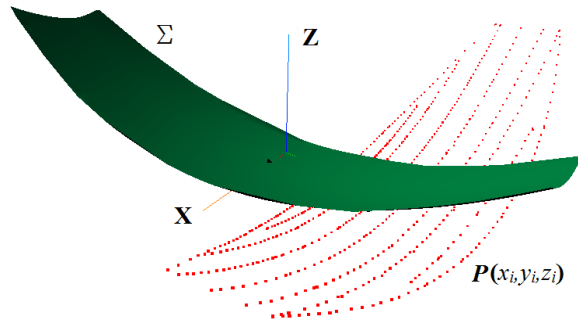


Figure 1: Two data sets

Once the two sets are specified with respect to two different origins we need such transformation of one of them as ‘rigid body’ so that to ensure the satisfaction of the eqn (1). Such 3D transformation is defined by six parameters: the components of the translation vector $\Delta x_c, \Delta y_c, \Delta z_c$ (here C is the geometry center of relocatable set) and three rotation angles $\varphi_x, \varphi_y, \varphi_z$. The numerical solution of this non-linear problem by usual optimization approaches is very complicated for various reasons, namely:

- In general, it is difficult to fit two sets even approximately. Hence, it is impossible to find the initial values of $\Delta x_c, \Delta y_c, \Delta z_c, \varphi_x, \varphi_y, \varphi_z$. It forces us to take them with maximum values that makes the computing process slow down.
- The surface cannot be simply single coherent that increases the number of constraints in the optimization problem.
- Often the surface does not have analytical representation, so its derivatives are unknown or do not exist. That makes it impossible to use efficient numerical algorithms based on the function derivatives.
- The computation time depends on value N (the number of points in P set). Therefore, the computing process becomes very slow when the dense of the point set grows significantly.

We propose a new approach that consists of two stages and the first of them is Principal Component Analysis (PCA). We apply PCA to so-called ‘rough fit’ that actually is the approximate initial fit of two sets. The second stage is the precise fit based on Stretched grid method (SGM) that allows accurate fitting of two sets according to minimum SD criterion in 1-4 linear steps. We demonstrate this approach based on parabolic aerial where Σ – analytic aerial surface, P – source point cloud obtained by measuring with standard electronic tacheometer «Trimble-M3» [13].

3. ROUGH FIT

PCA is often used to map data on a new orthonormal basis in the direction of the largest variance [14]. The largest eigenvector of the covariance matrix always points to the direction of the largest variance of the data.

In our case, the first data set is the point cloud and the second is the continuous surface, therefore, we should represent the surface by another point cloud as well. Further procedure follows the scheme described in work [15]. If the covariance matrix of two point clouds differs from the identity matrix, a rough fit can be obtained by simply aligning the eigenvectors of their covariance matrices. This alignment is obtained in the following way: first, the two point clouds are centered such that the origins of their final bases coincide. The centering of the point cloud simply corresponds to subtracting the centroid coordinates from each of the point coordinates. The centroid of the point cloud corresponds to the average coordinate and is thus obtained by dividing the sum of all point-coordinates by the number of points in the point cloud. Since the rough fit based on PCA simply aligns the directions in which the point clouds vary the most, the second step consists of calculating the covariance matrix of each point cloud. The covariance matrix is an orthogonal 3×3 matrix, the diagonal values of which represent the variances while the off-diagonal values represent the covariance. Third, the eigenvectors of both covariance matrices are calculated. The largest eigenvector is a vector in the direction of the largest variance of the 3D point cloud, and therefore represents the point cloud’s rotation. Further, let A be the covariance matrix, let v be an eigenvector of this matrix, and let λ be the corresponding eigenvalue. The problem of eigenvalues decomposition is then defined as

$$Ax = \lambda x, \quad (2)$$

and further reduces to

$$x(A - \lambda I) = 0. \quad (3)$$

It is clear that (3) only has a non-zero solution if $A - \lambda I$ is singular, and consequently if its determinant equals to zero

$$\det(A - \lambda I) = 0. \quad (4)$$

The eigenvalues can simply be obtained by solving (4), whereas the corresponding eigenvectors are obtained by substituting the eigenvalues into (2). Once the eigenvectors are known for each point cloud, the fit is achieved by aligning these vectors. Then, let us assume that matrix T_{Σ} represents the transformation that would align the largest eigenvector of the target point cloud related to the surface Σ with the X -axis. Now, let us suppose that matrix T_P represent the transformation that would align the largest eigenvector of the source point cloud P with the X -axis as well. Finally, we can align the source point cloud with the target point cloud easily if we take into account coincidence of both principal component systems (X_{pr}, Y_{pr}, Z_{pr}) of source and target point clouds (see Fig.2).

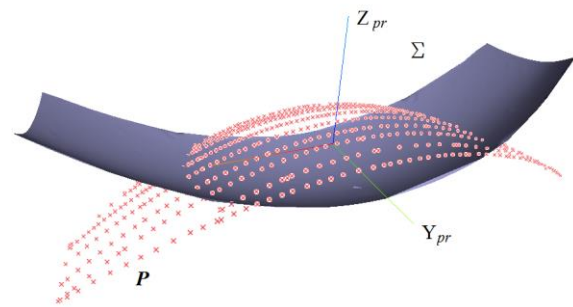


Figure 2: Two data sets in common principal component system

Unfortunately, we cannot always determine the direction of collinear principal component axes uniquely with PCA (see Fig.2). Therefore, we correct their directions in this issue manually by rotating the source point cloud about axes X_{pr} , Y_{pr} , Z_{pr} consequently to meet the minimum of SD criterion. In our sample, we rotate the point cloud about X_{pr} axis (see Fig. 3.)

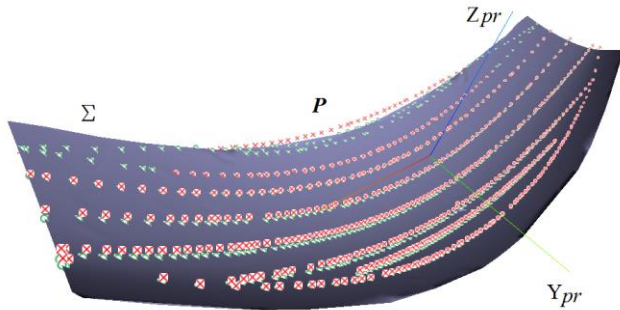


Figure 3: The rough fit of two data sets in common principal system

The rough fit cannot obtain real minimum solution according to the SD criterion therefore the next stage is the precise fit.

4. PRECISE FIT

The precise fit stage is based on SGM. SGM described in work [16] is a numerical technique for finding approximate solutions of various mathematical and engineering problems that can be related to an elastic grid behavior. In our case, we apply SGM to drag in the source point cloud as a ‘rigid body’ to the target surface by the set of elastic springs. Each elastic spring for our cloud connects the nearest neighbor point on the target surface q_i of each point p_i in the source point cloud (see Fig. 4). We find the neighbor point on the target surface by normal projection of the source point onto the target surface. This approach is similar to ICP point-to-point technique described in [15] but is much easier and has another physical meaning.

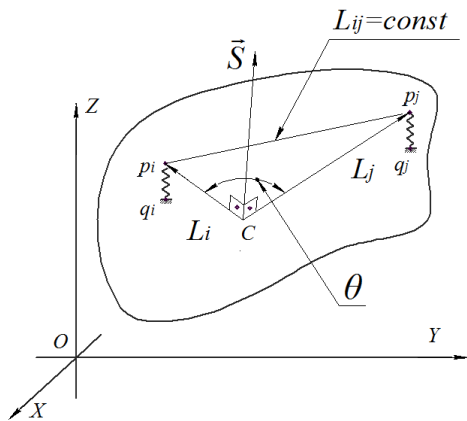


Figure 4: The scheme of the precise fit

The aim of the precise fit is to find functions $\Delta x_c, \Delta y_c, \Delta z_c, \varphi_x, \varphi_y, \varphi_z$ that obtain the minimum to exp (1). If we apply classical motion equation, we should further resolve non-linear equation system consisted of transcendental functions. Fortunately, we can

take into account linear dependence of displacement of points on the cloud rotation as a rigid body.

Taking into account the ‘rigid body’ rotation of point cloud due to precise fit, we can write the displacement of an arbitrary point p_i as follows

$$\begin{Bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{Bmatrix} = \begin{bmatrix} B_{11}^{(i)} & B_{12}^{(i)} & B_{13}^{(i)} \\ B_{21}^{(i)} & B_{22}^{(i)} & B_{23}^{(i)} \\ B_{31}^{(i)} & B_{32}^{(i)} & B_{33}^{(i)} \end{bmatrix} \begin{Bmatrix} \Delta x_j \\ \Delta y_j \\ \Delta z_j \end{Bmatrix} + \begin{Bmatrix} \Delta x_c \\ \Delta y_c \\ \Delta z_c \end{Bmatrix}, \quad (5)$$

where $\Delta x_i, \Delta y_i, \Delta z_i$ – displacements of an arbitrary point i ;

$\Delta x_j, \Delta y_j, \Delta z_j$ – displacements of point j ;

$\Delta x_c, \Delta y_c, \Delta z_c$ – displacements of the point cloud centroid.

We can calculate the components of the normalized matrix B for an arbitrary point of the cloud as a rotation matrix about the unit vector $\vec{s}(u, v, w)$ at the angle θ (see Fig. 4) by the following expressions

$$\begin{aligned} B_{11}^{(i)} &= (u^2 + (1-u^2)\cos\theta) \cdot F, \\ B_{12}^{(i)} &= (uv(1-\cos\theta) - w\sin\theta) \cdot F, \\ B_{13}^{(i)} &= (uw(1-\cos\theta) + v\sin\theta) \cdot F, \\ B_{21}^{(i)} &= (uv(1-\cos\theta) + w\sin\theta) \cdot F, \\ B_{22}^{(i)} &= (v^2 + (1-v^2)\cos\theta) \cdot F, \\ B_{23}^{(i)} &= (vw(1-\cos\theta) - u\sin\theta) \cdot F, \\ B_{31}^{(i)} &= (uw(1-\cos\theta) - v\sin\theta) \cdot F, \\ B_{32}^{(i)} &= (vw(1-\cos\theta) + u\sin\theta) \cdot F, \\ B_{33}^{(i)} &= (w^2 + (1-w^2)\cos\theta) \cdot F. \end{aligned} \quad (6)$$

Here $F = L_i / L_j$, where L_i, L_j – vectors to points i and j respectively from the point cloud centroid.

Due to exp (5) we can calculate the displacement of each point in the point cloud if we know the displacement of single point number j only.

The further step is to write the expression for the potential energy of entire connecting lines between the cloud points and the springs including (Fig. 4) that takes the following form

$$\Pi = D \sum_{m=1}^n R_m^2, \quad (7)$$

where n - total number of springs,

R_m - the length of spring number m ,

D - an arbitrary constant ($D = 1$ in our case).

Then, let us suppose that co-ordinate vector $\{X\}$ of all the points of the cloud is associated with a final cloud position, when the source cloud is fit to the target surface and the vector $\{X\}'$ is associated with an initial point cloud position. Thus, vector $\{X\}$ will look in the following way

$$\{X\} = \{X\}' + \{\Delta X\}, \quad (8)$$

where $\{\Delta X\}$ - vector of the co-ordinate increment of entire points.

To determine vector $\{\Delta X\}$ we should derive function (9) by incrementing vector $\{\Delta X\}$ with form (10) taken into account, i.e.

$$\frac{\partial \Pi}{\partial \Delta X_{kt}} = 0, \quad (9)$$

where k – number of current point,
 t – number of current co-ordinate.

After transformations using exps (5), (7), (8), (9) and keeping all lengths L_{ij} constant (see Fig.4) we can obtain the following linear equation system 6×6

$$K \cdot \Delta x = Q, \quad (10)$$

where vector Δx has only 6 unknown components to be found, namely $\Delta x_j, \Delta y_j, \Delta z_j, \Delta x_c, \Delta y_c, \Delta z_c$.

Using exp (5) we can calculate the displacements of entire points of the cloud. The final fit of two sets is presented in Fig.5.

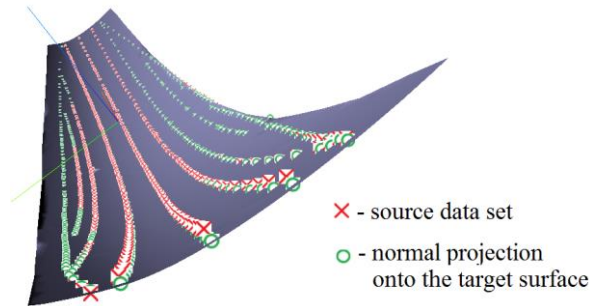


Figure 5: The final fit of two data sets.

5. CONCLUSION

In spite of linear nature of the precise fit, the process needs some iterations to converge because of some disparity of two sets after rough fit. Table 1 shows the matching error of SD against the number of iterations.

Table 1: The process convergence

Iterations	Relative Error of SD,%	Time, sec
2	4.045	4.64
8	1.5257	18.30
17	0.0984	46.72
20	0.0107	55.00
29	0.0064	85.55

As we can see, the process meets minimum SD criterion very quickly. The final error (about 0.01%) means that the fit precision is about 1-2 mm for the aerial with about 30m of overall dimension.

6. ACKNOWLEDGMENTS

This work was supported by grant RFFI (Russian Fond Fundamentalnyh Issledovaniy) under Grant # 15-07-01962.

7. REFERENCES

- Vaillant M., Glaunes J., 2005 Surface matching via currents. *Lecture Notes in Computer Science: Information Processing in Medical Imaging* — Vol. 3565. - Pp. 1-5
- Friedman J., Bentley J.L., Finkel R.A., 1977 An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*. - Vol.3, no.3.- pp. 209-226
- Brunstrom K., Stoddart A. J., 1996 Genetic algorithms for free-form surface matching Proc. *ICPR*. - pp. 689-693.
- Dyshkant N., 2010 An algorithm for calculation the similarity measures of surfaces represented as point clouds. *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications* -Vol.20, no.4-Pp. 495-504.
- Y. Liu, L. Li, and Y. Wang, 2004 Free Form Shape Matching Using Deterministic Annealing and Softassign, in *17th International Conference on Pattern Recognition*, Cambridge, United Kingdom, August 23-26.
- Y. Liu, L. Li, and B. Wei, 2004 3D Shape Matching Using Collinearity of Constraint, in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 26 - May 1.
- Gatzke T., Zelinka S., Grimm C., Garland M., 2005 Curvature Maps for Local Shape Comparison. In: *Shape Modeling International*. — Pp. 244-256
- Bergevin, R., Laurendeau, D., Poussart D., 1995 Registering Range Views of Multi-Part Objects, *Computer Vision and Image Understanding*. *IEEE Trans. Pattern Analysis and Machine Intell.*, 61(1):1–16, Jan.
- Sitnik R., Kujawinska M., 2002. Creating true 3d-shape representation merging methodologies. *Three-Dimensional Image Capture and Applications V, Proceedings of SPIE* Vol. 4661, pp. 92-99.
- Nelder J.A., Mead R., 1965 A simplex method for function minimization. *Computer Journal*. - Vol. 7. Pp. 308-313
- Gruen A., Akca D., 2005 Least Squares 3D Surface and Curve Matching. *ISPRS Journal of Photogrammetry and Remote Sensing*. — Vol. 59. — Pp. 151-174.
- Popov E.V., Rotkov S.I. The Optimal Superpose of a Finite Point Set With a 3D Surface, *Proceedings of the International Conference on Physics and Technology CPT2013*, 12-19 of May 2013, Larnaca, Ciprus, 2013
- Trimble M3 Mechanical Total Station, <http://www.trimble.com/Survey/trimblem3.aspx>
- B. Draper, W. Yambor, and J. Beveridge, “Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures,” *Empirical Evaluation Methods in Computer Vision*, 2002, pp. 1–14.
- Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn, “A Survey of Rigid 3D Point cloud Registration Algorithms”, *AMBIENT 2014 : The Fourth International Conference on Ambient Computing, Applications, Services and Technologies, IARIA*, 2014, pp. 8–13.
- Popov E.V., On Some Variational Formulations for Minimum Surface. *Transactions of Canadian Society of Mechanics for Engineering*, Univ. of Alberta, vol.20, N 4, 1997, pp. 391–400.

About the author

Eugene Vladimirovich Popov,
 Doctor of Science, professor of Nizhegorodsky State Architectural and Civil Engineering University,
 Engineering Geometry and Computer Graphics Chair.
 e-mail: popov_eugene@list.ru