*Вісник Національного технічного університету "ХПІ", 2019, № 28 (1353)*
**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

**UDC 004.932:519.652** **DOI: 10.20998/2411-0558.2019.28.10**

***VOLODYMYR MOROZ,*** PhD, Prof., I.I. Mechnikov National University, Odessa,
***MAKSYM SHVANDT,*** BSc, I.I. Mechnikov National University, Odessa

# RECONSTRUCTION AND REGISTRATION OF 3D SURFACES OBTAINED BY SCANNING

In this article the general approach for the three-dimensional surface reconstruction is examined. Also, the registration of three-dimensional surfaces is considered. The process of converting a three-dimensional model from a triangular mesh into a point cloud based on the generation of a random point using the principle of barycentric coordinates was applied. Based on the material studied, a complex procedure is proposed, which includes both correction of errors and defects in surfaces obtained as a result of an object scanning, and the registration of these surfaces previously converted into point clouds. On a specific example of the usage of three-dimensional models in dentistry the proposed procedure was tested, results were obtained and conclusions were made. Figs. 6. Refs. 23 titles.

**Keywords:** 3D-surface; reconstruction; registration; scanning; triangular mesh; point cloud.

**Problem statement.** 3D-models are widely used in medicine, especially in stomatology. In our case there are to 3D-models of human jaw that need to be processed. The first one was obtained from a medical tomographic scanner. This scanner takes up to 500 pictures of jaw from multiple perspectives and then its software combines them into a single 3D-surface as an output file. The second model is obtained by scanning with a usual 3D-scanner of a real dental model of the same jaw. The main task is to align the second surface with respect to the first and to perform the registration of these models. But right after acquisition the scanned dental model contains multiple defects, like polygon self-intersections and holes due to the 3D-scanning technics. These defects and artifacts will affect the accuracy of model aligning and registration: some false matches can occur. That`s why before registration the dental model must be reconstructed and freed from artifacts. Also, most of surface registration algorithms provide best results if the input surface is represented as a point cloud, so it leads to the need of surface conversion from mesh to point cloud. Thus, both the input surfaces must be repaired, converted to point clouds and finally, aligned and registered.

**Analysis of recent research.** Nowadays 3D-surfaces are intensively used in many different areas of object modelling, computer graphics and geometry processing. The 3D-surface can be obtained as a result of digitized scan from 3D-scanner or constructed on the computer manually. It can be

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

represented in two ways: as a mesh or as a point cloud.

A 3D mesh is a set of points in 3D-space, interconnected with edges. By other words it consists of multiple polygons of different shape. For example, most of 3D-scanners use this type of mesh representation. Point cloud is a set of points of 3D-space without any additional information.

The usage of 3D surfaces include many operations on them, but the most common and necessary are the problems of mesh registration and repair. Also during processing it can be necessary to change the shape of the mesh without changing the whole structure. One of the most suitable and commonly used for this type of mesh is a triangular mesh, i.e. all its polygons are represented by triangles [1, 2]. Right after acquisition, the mesh of the model can experience several problems: it can contain different defects, such self-intersections and holes. In production and industrial design, for example, some applications assume that the mesh enclose to solid and does not contain degenerate or nearly degenerate elements [1, 3]. That`s why before using the 3D model, it is important to check its structure and if there are any defects, to repair them.

Another common task for 3D models is the task of surface registration. Three-dimensional registration is a fundamental problem in many industries. For example: medical visualization [4], environment reconstruction in archeology [5], construction of three-dimensional terrain maps [6] or sea bed and in industry, in particular, in robotics [7]. The main tasks of registration are modeling of structures, reconstruction of objects from three-dimensional images from multiple perspectives, alignment of temporary three-dimensional images, in particular for damage monitoring. The registration process will also provide bad results if the meshes contain different defects, because they can affect the registration accuracy. Thus the mesh reconstruction processes is very important.

**The purpose of the article.** The algorithms for surface reconstruction and registration act on their own. But in particular case of human jaw models aligning and registration they need to be combined into a single process. So, the purpose of this research is to develop a complex procedure that will perform both reconstruction and registration phases, including necessary mesh transformations.

**The mathematical model.** Triangle meshes are one of the most convinient ways of surface representation; in many geometry processing algorithms they are considered as a collection of triangles without any particular mathematical structure. According to [1, 3] a triangle mesh $M$ consists of a geometric and a topological component. Thus, it is as a pair $(P,\Sigma)$, where $P$ is a set of $N$ point positions $p_i = (x_i, y_i, z_i) \in \mathbf{R}^3$, $1 \le i \le N$ and $\Sigma$ is an abstract simplicial complex which contains all the topological information. The simplicial complex $\Sigma$ is a set of subsets of $\{1,...,N\}$ and these subsets are called simplices. There are three types of them: vertices $v=\{i\}$, edges $e=\{i,j\}$ and triangles $t=\{i,j,k\}$. The simplicial complex $\Sigma$ describes a topology (connectivity) on *P*. Also, as stated in [1, 3, 8] an important topological quality of a surface is whether or not it is manifold, i.e. if for each point the surface is locally homeomorphic to a disk (or a half-disk at boundaries). A triangle mesh is manifold if it does neither contain non-manifold edges or vertices, nor self-intersections. An edge is called non-manifold when it has more than two incident triangles and a vertex is called non-manifold if it was generated by pinching two surface sheets together at that vertex. As mentioned in [1, 3], non-manifold meshes can be problematic for most mesh processing algorithms, since around non-manifold configurations there does not exist any well-defined local geodesic neighborhood. Thus we say that *M* is combinatorially manifold iff $\Sigma$ is a combinatorial manifold [3]. We also say that $\Sigma$ is a combinatorial manifold iff all its vertices are manifold, and a vertex of $\Sigma$ is manifold if its neighborhood is homeomorphic to a disk in the topology of $\Sigma$. We also define an orientation of an edge as an ordering of its two vertices. Two triangles of a mesh sharing an edge $e$ are consistently oriented if they induce different orientations on $e$. A triangle mesh is orientable iff all its triangles can be oriented consistently.

The whole mesh reconstruction algorithm is presented in [3]. It is performed in two successive phases: topology reconstruction and geometry correction. During topology reconstruction phase the algorithm aims to convert the set of input polygons into a single combinatorially manifold and oriented triangle mesh without boundary. The repairing algorithm first converts each mesh polygon to a set of triangles through triangulation [2]. Then the first step in the topology reconstruction amounts to building an explicit connectivity between adjacent triangles. This can be done by comparing the edges and their vertices taken from all triangles in the sorted edge list. Then the algorithm aims to remove topological singularities [3, 9]. This phase is based on two high level operations: cutting and stitching. The cutting operation identifies non-manifold edges and cuts the surface along such edges. The stitching operation involves joining two boundary edges and guarantees that the surface has a manifold topology. If the resulting manifold mesh is made of more than one connected component, only the biggest component is kept. Then the resulting mesh must be oriented. If after the orientation the mesh is still manifold and no cuts were necessary, then we get to geometry correction phase. Typical geometric flaws in a triangle mesh are degenerate elements and self-intersections. In order to repair geometric flaws, the algorithm presented here uses the simplicial neighbourhood approach [3, 11], i.e. the submesh defined by set of triangles that share at least a vertex with

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

the selected one; the second order simplicial neighborhood is the simplicial neighborhood of the simplicial neighborhood, and so on. The degenerate triangles can be determined by checking their corners and edges` length. Than they are removed using edge swap or edge collapse procedures [12 – 14]. Sometimes, it may happen that some degeneracy cannot actually be treated due to topological constraints. In this case, we remove all the triangles belonging to the simplicial neighborhood of *kth* order [11]. Then, we run the swap/collapse procedure within the patch. In other words at each iteration we enlarge the size of the neighborhood of the degeneracies that could not be solved through the swap/collapse procedure. The whole stops with failure after a prescribed number of attempts. After that and before the gap re-triangulation [10], the algorithm checks and possibly removes small disconnected components that detached from the main object [3]. The last step is self-intersection removal. In order not to check every pair of triangles the 3D model is divided into fixed number of voxels and we check every pair only within one voxel or between adjacent ones. For self-intersection removal we also use simplicial neighborhood approach. Then we patch the remaining gaps with new triangles. Summarizing, the overall geometry correction receives a reconstructed manifold mesh, then firstly uses the algorithm for degeneracy removal and only the self-intersections removal procedure.

After having the input surface successfully triangulated and reconstructed, the models can be registered. But as mentioned earlier, most of registration algorithms work with surface, represented as point cloud, which basically is a set of points of three-dimensional space as without any additional information. The given tomographic scan model is initially represented as a point cloud, but the dental model scan is not. Thus, it must be converted from triangular mesh to point cloud. Triangular mesh of the dental model already contains multiple points that represent polygon vertices and edges that connect them. But removing all the edges and just leaving the vertices will result in a surface that has too big spaces between points where triangular faces used to be. So it is necessary to generate some additional points to fill the gaps between the already present ones. The procedure for additional point generation includes next steps [15]. Firstly, it is necessary to decide, in which triangles additional points should be generated. Too small triangles may not require any point generation. So in order to receive uniform point cloud the areas of triangles is computed and triangles with too small areas can be ignored. After the selection of the triangles it is time to generate a random point inside them using the concept of barycentric coordinates. It means that a point can be generated inside a triangle using a linear combination of three numbers and the vertices of that triangle. Let $u$, $v$ and $w$ be the three numbers and $u + v + w \le 1$. We can generate a point inside a

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

triangle by just generating two random numbers $u$ and $v$, setting $w=(1-(u+v))$. And then we multiply these numbers with the coordinates of the vertices of the triangle. Figure 1 show different situations for point generation.

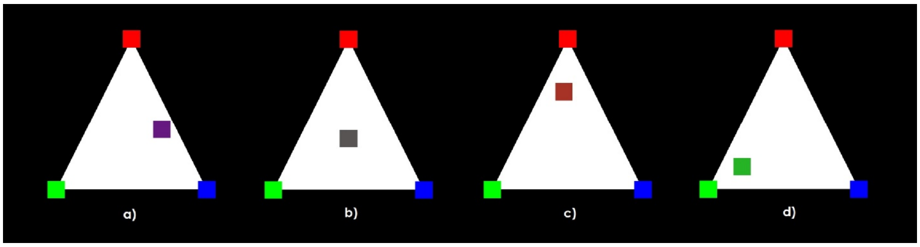



Fig. 1. Case a) $u=0.1,\ v=0.5$; b) $u=0.33,\ v=0.33$; c) $u=0.2,\ v=0.15$; d) $u=0.7,\ v=0.15$

It is important to notice, that there is a certain restriction: if $u+v>1$, then $u=1-u$ and $v=1-v$. This restriction is stated in order to prevent the sampled point from exceeding the Green-Blue edge. Thus using this principle triangular mesh is converted into point cloud.

Pair registration is usually accomplished using one of the many versions of the Iterative Closest Point Method (ICP) proposed by Bessle and Mackay [16]. Due to the non-convexity of the optimization problem, ICP-based methods require the setting of a preliminary approximate transformation in order to increase the possibility of successful surface alignment. To overcome the limitations of the ICP, several methods for establishing global correspondences based on local surface characteristics were proposed in [17, 18]. Thus, there are two large classes of registration methods: feature-based methods and iterative methods. The former are used for initial alignment, the latter serve for more accurate registration of previously roughly aligned surfaces. The suggested mesh registration procedure first uses the methods based on feature points for preliminary registration and then the ICP as the result improvement. The feature points for the preliminary processing are set by user.

The Feature-based method acts as follows: given two point clouds, the original $P_1 \subset \mathbf{R}^3$ and the target $P_0 \subset \mathbf{R}^3$, the problem is to find a rigid transformation $T_{0\leftarrow 1}$ such that $p_0 = T_{0\leftarrow 1}(p_1)$ for the corresponding points $p_0 \in P_0$, $p_1 \in P_1$. In uniform coordinates this is a linear transformation

$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = T_{0\leftarrow 1} \begin{bmatrix} p_1 \\ 1 \end{bmatrix} = \begin{bmatrix} R_{0\leftarrow 1} & t_{0\leftarrow 1} \\ 0^{\tau} & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ 1 \end{bmatrix}, \qquad (1)$$

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

where $R_{0\leftarrow 1}$ is a $3\times 3$ rotation matrix, $t_{0\leftarrow 1}$ is a $3\times 1$ transfer vector and $0^{\tau}$ is a $1\times 3$ zero vector. Additional properties can be assigned to the points $p_i$, such as: surface normal $n_i$, salinity $h_i$ [19], or descriptor $d_i$, where the index denotes the index of the corresponding point.

The task is directly related to search of the source cloud in the destination cloud. From the set of orientations found by matching local data descriptors, transformation can be estimated using a reliable estimation, such as Random Sample Consensus [20]. Key points are chosen as extremes of a salinity measure that determines the type of structures that are searched for in the data and directly affects the frequency and reliability of detection. Fixed and adaptive scale detectors can be distinguished. Local extrema are obtained by suppressing non-maxima, and only those key points with maximum local salinity are preserved. In particular, the key point *p* with salinity *h* is only preserved if $h \geq h_i$ for all $i \in N_{\sigma}(p)$, where $N_{\sigma}(p)$ is the set of point indices that are in the σ-neighbourhood of the point *p*. The points *p* with $|N_{\sigma}(p)| < 10$ are excluded from the identification of key points.

Two types of key point detectors can be identified. The first one uses a covariance matrix of points:

$$C_p = \frac{1}{\sum_i w_i} \sum_{i \in N_s(p)} w_i \left(p_i - \mu\right)\left(p_i - \mu^{\tau}\right),\ \mu = \frac{1}{\sum_i w_i} \sum_i w_i p_i. \qquad (2)$$

The second type uses a covariance matrix of normals,

$$C_n = \frac{1}{\sum_i w_i} \sum_{i \in N_s(p)} w_i n_i n_i^{\tau}, \qquad (3)$$

where $w_i$ are the weights assigned to individual points, $N_s(p)$ is the set of points adjacent to *p*, $\{i \mid \|p_i - p\| \leq s\}$. The eigenvalues of these covariance matrices in the increasing order are $\lambda_1$, $\lambda_2$, $\lambda_3$ and $q_1$, $q_2$, $q_3$ are the corresponding eigenvectors. The following measures of salinity as functions of eigenvalues are considered: $\min\left(\frac{\lambda_1}{\lambda_2}, \frac{\lambda_2}{\lambda_3}\right)$, $\lambda_1$, $\lambda_2$, $\lambda_3$, $\frac{\lambda_1}{\lambda_2}$, $\frac{\lambda_2}{\lambda_3}$. Despite the intuitive geometric sense of the salinity measure, this may not express their quality in terms of the repetition of the corresponding feature points. Therefore, several such measures are evaluated in order to select the most appropriate one for the given problem. For example, [21] uses the least eigenvalue $\lambda_3$ of the matrix $C_p$ and several methods based on the matrix $C_n$ were implemented in Point Cloud Library [22]. For the key points found in this stage, local reference systems are established and descriptors are calculated. Local reference systems are a key tool to achieve the desired level of descriptor invariance. The Cartesian coordinate system is the most common. Using counting systems offers several benefits; for example, three-dimensional point distribution may be captured by a descriptor to increase its distinguishing power. The general approach used by many methods is to establish at the beginning of a count the covariance matrix of signs from the eigenvectors.

A fairly common is the descriptor with $3 \times 3$ spatial containers in the plane and 8 polar containers (fig. 2). Descriptor is created by projecting the points in the neighborhood and the corresponding normals into three planes, drawn on pairs of basis vectors, and accumulates projections on the histograms. Each landmark point casts weighted voices to the two closest polar containers given by the normal projection and to the four closest spatial containers given by the projection of the point. The scales are proportional to the relative proximity to each container of the histogram and inversely proportional to the local density of the surface sampling.

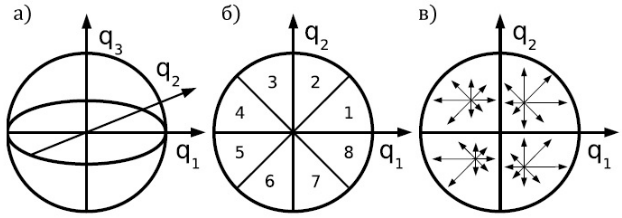



Fig. 2. Descriptor: a) spherical support with a local reference system, b) 8 orientation containers, c) 4 spatial containers

Invariant descriptors are used to determine the previous correspondences between the source and the destination point cloud. Let $p_1$, $p_0$ be a pair of points from the source and the target, respectively, and $d_1$, $d_0$ their descriptors. Then, if $d_0$ is among the three nearest neighbors of $d_1$ from the set of descriptors of the target and $d_1$ is among the three nearest neighbors of $d_0$ from the set of descriptors of the source, the correspondence is set. The position is estimated from a set of prior matches of locally optimized RANSAC – the pairs of matches, received randomly from the set, are searched for; surface positions are generated and we accept only that, what maximizes the number of matched matches. The maximum number of iterations is estimated by establishing the probability of non-rejections $\mu = 1/100$. To generate surface positions and to refine positions from agreed matches we calculate

$$\underset{R_{0\leftarrow1},t_{0\leftarrow1}}{\operatorname{argmin}} \sum_i \left( \left\| R_{0\leftarrow1} p_{1,i} + t_{0\leftarrow1} - p_{0,i} \right\|^2 + \left\| R_{0\leftarrow1} n_{1,i} - n_{0,i} \right\|^2 \right) \tag{4}$$

for the corresponding positions $p_{0,i}$, $p_{1,i}$ and normal vectors $n_{0,i}, n_{1,i}$ [23].

After performing rough registration, ICP method for results improvement is used. Let $A$ be the set of $N_a$ points marked $\vec{a}_i : A = \{\vec{a}_i\}$, $i = 1,...,N_a$. The distance between the point $\vec{p}$ and the set of points $A$ is:

$$d(\vec{p}, A) = \min_{i\in\{1,...,N_a\}} d(\vec{p}, \vec{a}_i). \tag{5}$$

The nearest point $\vec{a}_j$ of the set $A$ satisfies the equality $d(\vec{p},\vec{a}_i) = d(\vec{p}, A)$; $d(\vec{r}_1,\vec{r}_2)$ is the Euclidean distance between two points $\vec{r}_1 = (x_1, y_1, z_1)$ and $\vec{r}_2 = (x_2, y_2, z_2)$. Let $l$ be the segment joining the points $\vec{r}_1$ and $\vec{r}_2$. The distance between the point $\vec{p}$ and segment $l$ is:

$$d(\vec{p}, l) = \min_{u+v=1} \| u\vec{r}_1 + v\vec{r}_2 - \vec{p} \|, \ u \in [0,1], \ v \in [0,1]. \tag{6}$$

Let $L$ be a set of $N_l$ segments, marked as $l_i$ and $L = \{l_i\}$, $i = 1,...,N_l$. The distance between the point $\vec{p}$ and the set of segments $L$ is:

$$d(\vec{p}, L) = \min_{i\in\{1,...,N_l\}} d(\vec{p}, l_i). \tag{7}$$

Let $t$ be a triangle defined by three points $\vec{r}_1 = (x_1, y_1, z_1)$, $\vec{r}_2 = (x_2, y_2, z_2)$ and $\vec{r}_3 = (x_3, y_3, z_3)$. The distance between the point $\vec{p}$ and some triangle $t$ can be defined in the same way.

A single quaternion is a 4-vector $\vec{q}_R = [q_0, q_1, q_2, q_3]^\tau$, where $q_0 \geq 0$, $q_0 + q_1 + q_2 + q_3 = 1$. The $3\times3$ rotation matrix is generated by a single quaternion:

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \quad (8)$$

Let $\vec{q}_T = [q_4, q_5, q_6]^\tau$ be a transfer vector. Full registration status vector is $\vec{q} = [\vec{q}_R \mid \vec{q}_T]$. Let $P = \{\vec{p}_i\}$ be a the set of points that must be aligned with the set of points of the model $X = \{\vec{x}_i\}$, $N_x = N_p$ and each point $\vec{p}_i$ corresponds to the point $\vec{x}_i$ with the same index. The RMS objective function to be minimized:

$$f(\vec{q}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| \vec{x}_i - R(\vec{q}_R)\vec{p}_i - \vec{q}_T \right\|^2. \quad (9)$$

The center of mass $\vec{\mu}_p$ of the measured set of points *P* and the center of mass $\vec{\mu}_x$ of the pint set *X* is:

$$\sum\nolimits_{px} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left[ (\vec{p}_i - \vec{\mu}_p)(\vec{x}_i - \vec{\mu}_x)^\tau \right] = \frac{1}{N_p} \sum_{i=1}^{N_p} \left[ \vec{p}_i \vec{x}_i^\tau \right] - \vec{\mu}_p \vec{\mu}_x^\tau. \quad (10)$$

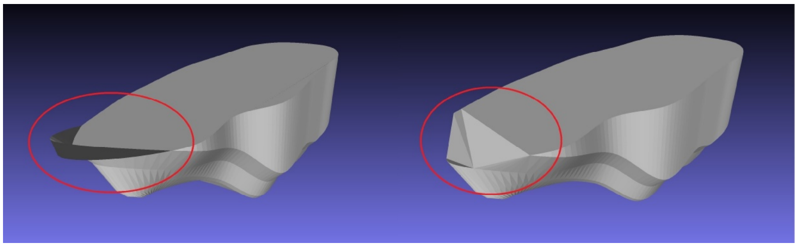
Cyclic components of an antisymmetric matrix $A_{ij} = \left(\Sigma_{px} - \Sigma_{px}^\tau\right)_{ij}$ are used to form a column vector $\Delta = [A_{23} A_{31} A_{12}]^\tau$. This vector is used to form a symmetric matrix $4 \times 4$ - $Q(\Sigma_{px})$:

$$Q(\Sigma_{px}) = \begin{bmatrix} tr(\Sigma_{px}) & \Delta^\tau \\ \Delta & \Sigma_{px} + \Sigma_{px}^\tau - tr(\Sigma_{px}) I_3 \end{bmatrix}, \quad (11)$$

where $I_3$ is a $3 \times 3$ unit matrix. The unit eigenvector $\vec{q}_R = [q_0, q_1, q_2, q_3]^\tau$, which corresponds to the maximum eigenvalue of the matrix $Q(\Sigma_{px})$, is chosen as the optimal rotation vector $\vec{q}_T = \vec{\mu}_x - R(\vec{q}_R)\vec{\mu}_p$.

The whole ICP method algorithm for point cloud can be presented as follows: the input set of points *P* from $N_p$ points $\{p_i\}$ and the target set of points *X* from $N_x$ points $\{p_i\}$ are given. Then iteration is initialized by assigning the values $P_0 = P$, $\vec{q}_0 = [1,0,0,0,0,0,0]^\tau$ and $k = 0$. Registration vectors are determined from the initial data set $P_0$, so the final registration is a complete transformation. Next steps are run till match within tolerance $\tau$. Firstly, the closest points are calculated: $Y_k = C(P_k, X)$ (computational complexity is $O(N_p N_x)$ the worst case scenario, $O(N_p \log N_x)$ in average). Then the registration is calculated: $(\vec{q}_k, d_k) = Q(P_0, Y_k)$ $\left(O\left(N_p\right)\right)$. After that registration is applied: $P_{k+1} = \vec{q}_k(P_0)$ $\left(O\left(N_p\right)\right)$. The iteration process stops when the change of root mean square error becomes less than the specified threshold $\tau > 0$: $d_k - d_{k+1} < \tau$.

**Results**. The whole surface reconstruction and repair procedure was implemented using the C++ language and open source point cloud processing library PCL. Fig. 3, *a* demonstrates the work of the mesh repair procedure.

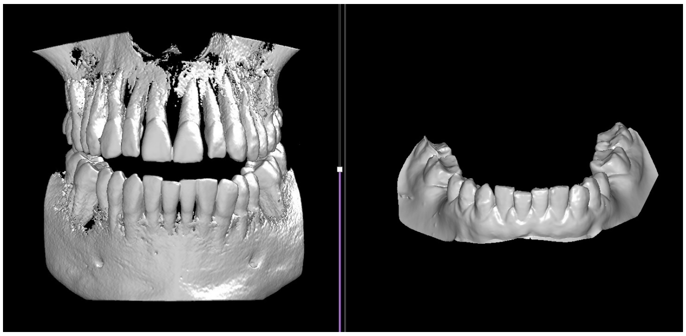


*a*) *b*)

Fig. 3. Raw model *a*) contains self-intersections; the model *b*) is free of them

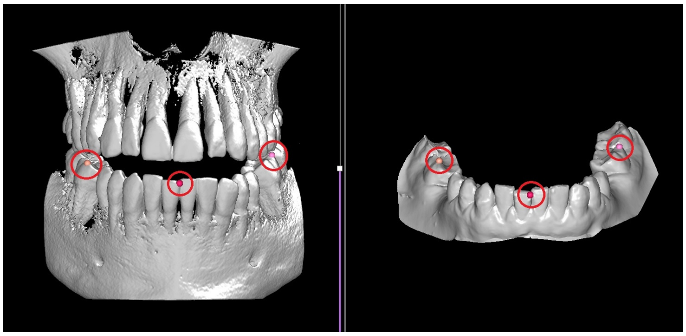
Now we present the result of surface registration phase on the example of medical images. Fig. 4 shows input surfaces, a) is a model received as a tomographic scan (initially represented as a point cloud) and b) is a dental cast. The dental cast model is already repaired through reconstruction phase and converted to point cloud. To perform a rough registration first, we set some feature points by hand (fig. 5). The results of it are shown on fig. 6, *a*. Then the specifying registration using the ICP is performed (fig. 6, *b*). As it can be seen the results received using ICP are much better than those received using rough registration even given the absence of gums on the surface obtained by tomographic scanning.

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**



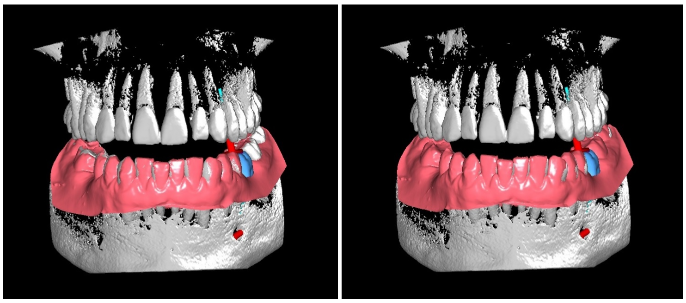
*a*) *b*)

Fig. 4. Input surfaces. Model a) is a tomographic scan, model b) is a scan of a dental cast



*a*) *b*)

Fig. 5. Definition of the key points on a) tomographic scan and dental cast scan b)



*a*) *b*)

Fig. 6. a) is the rough registration using key points, b) is specifying registration using ICP

**Prospects for further research.** Further studies are planned in the search for optimization of difference of the registration surfaces by constructing the best sampling of the surfaces. The best grid sampling is a prerequisite for minimizing the standard deviation of the surfaces.

**Conclusion.** The suggested complex procedure of the surface reconstruction and registration can be very useful in medicine, industrial design and many other areas that widely use 3D models. The mesh repair phase presents fine input mesh repair quality as it gets completely free of self-intersections, singular elements and other flaws. It also demonstrates fine speed of work and hardware usage in terms of memory consumption. The surface registration phase also provides good results. Two stage process using feature points and then ICP provides much more accurate than it can be using only one of methods, as shown by tests. It also performs fine in terms of hardware usage. Thus, the surface received after both reconstruction and registration is suitable for further analysis and usage and the whole complex procedure proves to be a fine tool for mesh processing.

**References**

**1.** Botsch, M., Pauly, M., Kobbelt, L., Alliez, P., L´evy, B., Bischoff, S., and Roessl, C. (2008), "Geometric Modeling Based on Polygonal Meshes". *In Eurographics 2008 Full-Day Tutorial*, pp. 4, 8-10, 25-40.

**2.** Frey, P.J., and George, P.L. (2008), *Mesh Generation. Application to Finite Elements*. Wiley, Second Edltion, pp. 11-13, 21-30, 235-273.

**3.** Attene, M. (2010), "A lightweight approach to repairing digitized polygon meshes". *The Visual Computer*, Vol. 26 (11), pp. 1393-1406.

**4.** Audette, M.A., Ferrie, F.P., and Peters. T.M. (1999), "An Algorithmic Overview of Surface Registration Techniques for Medical Imaging". *Oxford University Press, Medical Image Analysis*, pp. 1-18.

**5.** De Reu, J., Plets, G., Verhoeven, G. et al. (2013), "Towards a three-dimensional cost-effective registration of the archaeological heritage". *Journal of Archaeological Science*, Vol. 40, pp. 1108-1121.

**6.** Hebert, M., Caillas, C., Krotkov, E., Kweon , I.S., and Kanade, T. (1989), "Terrain Mapping for a Roving Planetary Explorer". *Robotics Institute*, Paper 405, pp. 997-1002.

**7.** Matrinez, C., Boca, R., Zhang, B., Chen, H., and Nidamarthi, S. (2015), "Automated bin picking system for randomly located industrial parts". *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 1-6.

**8.** Attene, M., and Falcidieno, B. (2006), "Remesh: An interactive environment to edit and repair triangle meshes". *Shape Modeling and Applications*, pp. 271-276.

**9.** Guéziec, A., Taubin, G., Lazarus, F., and Horn, B. (2001), "Cutting and stitching: Converting sets of polygons to manifold surfaces". *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7 (2), pp. 136-151.

**10.** Barequet, G., and Sharir, M. (1995), "Filling gaps in the boundary of a polyhedron". *In Computer-Aided Geometric Design*, Vol. 12 (2), pp. 207-229.

**11.** Rourke, C.P., and Sanderson, B.J. (1972), *Introduction to piecewise linear topology*, Springer-Verlag.

**12.** Botsch, M., and Kobbelt. L. (2001), "A robust procedure to eliminate degenerate faces from triangle meshes". *Vision, Modeling and Visualization*, pp. 283-290.

**ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)**

**13.** Guibas, L., Salesin, D., and Stolfi, J. (1989), "Epsilon geometry: building robust algorithms from imprecise computations". *ACM Symposium on Computational Geometry*, pp. 208-217.
**14.** Dey, T.K., Edelsbrunner, H., Guha, S., and Nekhayev, D. (1999), "Topology preserving edge contraction". *Publications de l'Institut Mathematique* (Beograd), Vol. 20 (80), pp. 23-45.
**15.** David de la Iglesia (2017), "3D point cloud generation from 3D triangular mesh. Medium". Available at: https://medium.com/@daviddelaiglesiacastro/3d-point-cloud-generation-from-3d-triangular-mesh-bbb602ecf238
**16.** Besl, P.J., and McKay, N.D. (1992), "A Method for Registration of 3-D Shapes". *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 14 (2), pp. 239-255.
**17.** Rusu, R.B., Blodow, N., and Beetz, M. (2009), "Fast Point Feature Histograms (FPFH) for 3D Registration". *IEEE International Conference on Robotics and Automation*, pp. 3212-3217.
**18.** Mian, A., Bennamoun, M., and Owens, R. (2010), "On the Repeatability and Quality of Keypoints for Local Featurebased 3D Object Retrieval from Cluttered Scenes". *International Journal of Computer Vision*, pp. 348–361.
**19.** Lee, C.H., Varshney, A., and Jacobs, D.W. (2005), *Mesh Saliency*, ACM SIGGRAPH, 8 p.
**20.** Fischler, M.A., and Bolles, R.C. (1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *New York: Communications of the ACM*, Vol 24, pp. 381-395.
**21.** Zhong, Y. (2009), "Intrinsic shape signatures: A shape descriptor for 3D object recognition". *Computer Vision Workshops (ICCV Workshops), IEEE 12th International Conference*, pp. 689-696.
**22.** Guibas, L., Salesin, D., and Stolfi, J. (1989), "Epsilon geometry: building robust algorithms from imprecise computations". *ACM Symposium on Computational Geometry*, pp. 208-217.
**23.** Petricek, T., and Svoboda, T. (2017), "Point cloud registration from local feature correspondences - Evaluation on challenging datasets". *PLoS ONE*, Vol. 12 (11): e0187943, 16 p.

*Статтю подав д-р техн. наук, проф. ОНУ імені І.І. Мечникова Гунченко Ю.А.*

*Надійшла (received) 03.11.2019*

Moroz Volodymyr, PhD in Applied Mathematics, Professor
Odessa I.I. Mechnikov National University
Str. Dvoryans'ka, 2, Odesa, Ukraine, 65072
Phone (+380) 67-484-6975, e-mail v.moroz@onu.edu.ua

Shvandt Maxim, BSc
Odessa I.I. Mechnikov National University
Str. Dvoryans'ka, 2, Odesa, Ukraine, 65072
Phone (+380) 48-723-5254, e-mail maxim.shvandt@gmail.com

УДК 004.932:519.652

**Реконструкція та реєстрація 3D-поверхонь, отриманих в результаті сканувaня / Мороз В.В., Швандт М.А.** // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2019. – № 28 (1353). – С. 117 – 130.

Досліджено загальний підхід до реконструкції тримірних поверхонь, представлених у вигляді триангуляційної сітки. Також в статті розглянуто реєстрації тримірних поверхонь. Застосований процес перетворення тримірної моделі із триангуляційної сітки на хмару точок, що базується на генерації випадкової точки разом із використанням принципу барицентричних координат. На основі вивченого матеріалу запропоновано комплексну процедуру, що включає в себе виправлення помилок та дефектів у поверхонь, отриманих в результаті сканування об'єкта, а також реєстрацію цих поверхонь, попередньо перетворених на хмари точок. На конкретному прикладі використання тримірних моделей у стоматології вище вказана процедура була протестована, були отримані результати та зроблені висновки. Іл.: 6. Бібліогр.: 23 назв.

**Ключові слова:** 3D-поверхня; реконструкція; реєстрація; алгоритм; сканування; триангуляційна сітка; хмара точок.

УДК 004.932:519.652

**Реконструкция и регистрация 3D-поверхностей, полученых в результате сканирования / Мороз В.В., Швандт М.А.** // Вестник НТУ "ХПИ". Серия: Информатика и моделирование. – Харьков: НТУ "ХПИ". – 2019. – № 28 (1353). – С. 117 – 130.

Исследован общий подход к реконструкции трехмерных поверхностей, представленных в виде триангуляционной сетки. Также в статье рассмотрена регистрация трехмерных поверхностей. Применен процесс превращения трехмерной модели из триангуляционной сетки в облако точек, который базируется на генерации случайной точки вместе с использованием принципа барицентрических координат. На основе изученного материала предложена комплексная процедура, которая включает в себя исправление ошибок и дефектов в поверхностях, полученных в результате сканирования объекта, а также регистрацию этих поверхностей, предварительно превращенных в облака точек. На конкретном примере использования трехмерных моделей в стоматологии вышеуказанная процедура была протестирована, были получены результаты и сделаны выводы. Ил.: 6. Библиогр.: 23 назв.

**Ключевые слова:** 3D-поверхность; реконструкция; регистрация; алгоритм; сканирование; триангуляционная сетка; облако точек.

UDC 004.932:519.652

**Reconstruction and registration of 3D surfaces obtained by scanning // Volodymyr Moroz, Maksym Shvandt** // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkov: NTU "KhPI". – 2019. – № 28 (1353). – P. 117 – 130.

In this article the general approach for the three-dimensional surface reconstruction is examined. Also, the registration of three-dimensional surfaces is considered. The process of converting a three-dimensional model from a triangular mesh into a point cloud based on the generation of a random point using the principle of barycentric coordinates was applied. Based on the material studied, a complex procedure is proposed, which includes both correction of errors and defects in surfaces obtained as a result of an object scanning, and the registration of these surfaces previously converted into point clouds. On a specific example of the usage of three-dimensional models in dentistry the proposed procedure was tested, results were obtained and conclusions were made. Figs.: 6. Refs.: 23 titles.

**Keywords:** 3D-surface; reconstruction; registration; algorithm; scanning; triangular mesh; point cloud.