# Disparity Measure Construction for Comparison of 3D Objects' Surfaces

Natalya Dyshkant

Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Vorobyevy gory, Moscow, Russian Federation

**Abstract.** In this paper a problem of 3D objects' surfaces comparison is considered. Each spatial object is given as a set of schlicht surfaces that are described by point clouds. This article discusses a proposed disparity measure to compare such objects and an algorithm to compute it. A method for comparison of mesh functions defined on different point sets is proposed. The theoretical base of the proposed approach is the piecewise-linear approximation of surfaces using Delaunay triangulations for initial point clouds. The presented approach uses Delaunay triangulations of each point clouds, general Delaunay triangulation for both clouds, function interpolation on basis of localization of triangulations in each other and function comparison on single cells of general triangulation. Localization is implemented on basis of minimum spanning trees. As the application of the proposed methodology a problem of 3D face models comparison is considered. It was experimentally verified that the proposed method is numerically efficient.

## 1 Introduction

In pattern recognition, along with k-nearest neighbors, estimate evaluation ([1]) and other algorithms there is a method of pattern matching. A problem of metrics construction for comparison of a given object with the standard occurs in biometric identification (e.g. by 3D face model) and various medical applications.

Any spatial object has a certain geometric shape. Objects' shape can be considered as a set of schlicht surfaces. In such a way a problem of object comparison reduces to a problem of surface comparison.

A method of pointwise description is usually used for specification of complex uneven surfaces. Then a surface is considered as a discrete nonregular ensemble of points. One can receive such description using 3D scanning methods (e.g. http://artec-group.com), topographic mapping and some other.

As the result of rapid progress in objects' 3D scanning techniques problems connected with received surfaces analysis and comparison occur.

Accurate and numerically efficient algorithms of computing disparity measure between surfaces are required in many applications of computer graphics. It is necessary to compare surfaces solving problems of surface classification, surface reconstruction by its separate fragments, etc.

Dyshkant N. (2009).
Disparity Measure Construction for Comparison of 3D Objects' Surfaces.
In *Proceedings of the 2nd International Workshop on Image Mining Theory and Applications*, pages 43-52
DOI: 10.5220/0001957300430052

Known approaches to compare piecewise linear functions that are defined on different discrete sets use hash functions [2]. They have quadratic computational complexity at worst and so are too computationally intensive.

Our method based on constructing of general Delaunay triangulations for union of two discrete sets. As the merging process can be implemented in linear time ([8]) then the total time to compute the proposed measure is comparable with time to construct Delaunay triangulation, i.e. $O(N \log N)$, where $N$ — the total amount of points in two sets. Consequently, the proposed method allows to avoid quadratic search in surface comparison that determines its advantage and novelty.

This paper is organized as follows. In section 2, we describe problem definition and introduce the proposed disparity measure. In section 3, each of stages of the proposed algorithm for disparity measure calculation is described. In section 4 we discuss application of the proposed method for 3D face model comparison. The results of computational experiments are given in section 5.

## 2 Problem Definition and Basic Ideas

A finite point set $G : \{(x^i, y^i) \in \mathbb{R}^2 | i = 1, \ldots, N\},\ N \geq 3$ is called a *nonregular two-dimensional mesh*.

We consider the following problem definition.

Let $G_1 = \{(x_1^i, y_1^i)\}_{i=1}^{N_1}$ and $G_2 = \{(x_1^2, y_1^2)\}_{i=1}^{N_2}$ be nonregular 2D meshes. Suppose $F_1$ and $F_2$ are the mesh functions corresponded to $G_1$ and $G_2$, i.e.

$$F_1 = \{f_1^i\}_{i=1} N_1,\ f_1^i = F_1(x_1^i, y_1^i); \quad F_2 = \{f_2^i\}_{i=1} N_2,\ f_2^i = F_2(x_2^i, y_2^i).$$

It is required to introduce a metrics for comparison such mesh functions and to design a numerically efficient algorithm to compute it.

Let $R$ be a rectangle in $\mathbb{R}^2$. Let $\mu(x, y)$ be a function that defines weight of fragments of $R$ in accordance with significance of function similarity on each fragment.

By $\mathfrak{G}$ denote the set of nonregular 2D meshes contained in $R$. Consider a set $\mathfrak{F}$ of single-valued functions on meshes from $\mathfrak{G}$.

Now we introduce a proximity function $\rho$ over set $\mathfrak{F}$.

By $Conv(G)$ denote the convex hull of $G$. Consider $F_1, F_2 \in \mathfrak{F}$. Let $\hat{F_1}$ and $\hat{F_2}$ be continuous functions defined on $Conv(G_1) \cap Conv(G_2)$ such that $\hat{F_1} \equiv F_1$ on $G_1$ and $\hat{F_2} \equiv F_2$ on $G_2$. By $T$ denote the Delaunay triangulation of mesh $G_1 \cup G_2$. We will say that this mesh is the *general* mesh and $T$ is the *general* Delaunay triangulation. Let $A, B, C$ be points of the general mesh. By definition, put

$$V(A, B, C, F_1, F_2) = \iint\limits_{\triangle ABC} \left| \ \hat{F}_1(x, y) - \hat{F}_2(x, y) \ \right| \ \mu(x, y)\ dxdy. \tag{1}$$

The value of $V$ indicates a weighted volume between two surfaces defined by functions $F_1$ and $F_2$ over triangle $\triangle ABC$.

We are interested in case when $Conv(G_1) \cap Conv(G_2) \neq \emptyset$. Otherwise two objects should be reduced to such coordinate system that allows them to be comparable.

Then we introduce a proximity function $\rho$ as

$$\rho(F_1, F_2) = \sum_{\triangle ABC \in T} V(A, B, C, F_1, F_2). \quad (2)$$

So we compute disparity measure for two surfaces summing values of volume between them over all triangles $\triangle ABC$ of general triangulation $T$.

We introduce disparity measure between two surfaces as a spacial volume between the corresponding functions. It is also allowed to use ”weighted” volume. In this case similarity of some surface patches will have weight greater than similarity of others.

Let us remark that two functions $F_1$ and $F_2$ are defined on *different* meshes. The basic idea of the proposed approach is to fill values of each functions at points of the other mesh using construction of two triangulations and their localization in each other.

## 3 Methods

The following stages will be performed to compute the disparity measure 2 between two surfaces given by functions $F_1$ and $F_2$.

1. Delaunay triangulation for each of the meshes $G_1$, $G_2$ is constructed;
2. each of two meshes $G_1$, $G_2$ is located in the triangulation for the other mesh;
3. each of two functions $F_1$, $F_2$ is interpolated on the mesh that the other function is defined on;
4. the general triangulation of both meshes $G_1 \cup G_2$ is constructed on basis of unseparated triangulation merging;
5. after that in each point of the general mesh values of *two* functions are known, and it is possible to make comparison operation on particular cells of the general triangulation, analyzing positional relationship of the spatial triangles given by functions.

Let’s consider each of steps in detail.

### 3.1 Delaunay Triangulation Construction

A triangulation $T$ for a set $G$ is called Delaunay triangulation if the following condition holds: there is no point in $G$ is inside the circumcircle of any triangle in $T$ (see Figure).

The used triangulation construction algorithm based on the paradigm of recursive decomposition (”divide-and-conquer strategy”): division of initial set into two approximately equal subsets, recursive triangulation construction of these subsets and merging of two divided triangulations. The data structure ”nodes with neighbors”, described in [3], can be used.

Computational complexity of this algorithm is $O(N \log N)$.

### 3.2 Point Location in Triangulation

To locate point $Q$ in a Delaunay triangulation $T$ means to declare the triangle of $T$ containing this point. In cases of (i) coincidence of point $Q$ and one of triangulation vertices; (ii) belonging of point $Q$ to one of triangulation edges, it is possible to declare any of triangles incident to the specified vertex or to the specified edge. In case of point $Q$ oversteps the boundaries of $T$ it is possible to declare certain infinite triangle or the nearest triangle to this point.

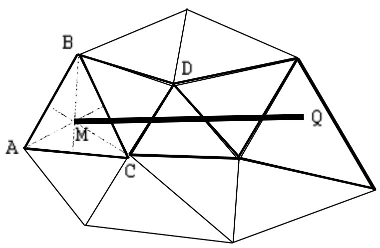


**Fig. 1.** Point location in triangulation.

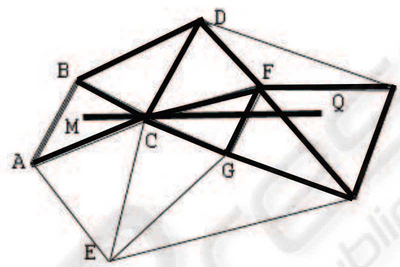


**Fig. 2.** Point $C$ of triangulation belongs to segment $[MQ]$.

Let point $M$ be a point that location in the triangulation is known (e.g. it can be the center of the inscribed circle of any triangle). The idea of algorithm solving point location in triangulation problem consists in gradual transition from $M$ to $Q$ along the straight line $(MQ)$. During each transition step changing on adjacent (neighboring by side) triangle is implemented. Case of belonging of a certain point of $T$ to segment $[MQ]$ is a case of a special interest (see Figure 2). A similar algorithm was described in [4].

Thus, after point location stage is finished, there is a path consisting of triangulation triangles, each of them (except the initial one) is adjacent with previous. We say that it is *location path*. On Figures 1 and 2 triangles of location path are outlined.

Complexity of one point location depends on quantity of triangles located along segment $[MQ]$ and is $O(\sqrt{N})$ on the average and $O(N)$ at worst.

### 3.3 Mesh Location in Triangulation

To locate a two-dimensional mesh $G$ in a triangulation $T$ means to locate all points of $G$ in this triangulation.

We propose a mesh location algorithm that uses spanning tree for graph $T$. In this case location paths will pass along edges of spanning tree.

As spanning tree for graph $T$ does not contain cycles and passes through all points of the mesh $G$, the algorithm will work correctly: it will not loop and performs location of absolutely all points of mesh.

The proposed method for comparison of schlicht surfaces uses the general triangulation of two meshes constructed by merger method on one of the subsequent stages.

This method uses minimum spanning trees (MST) of both meshes. So it is justified to use exactly *minimum* spanning trees for mesh location. Then location paths will be optimal (see Fig. 3, 4).

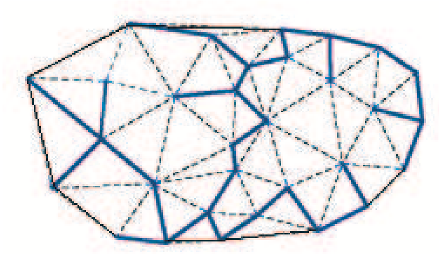

**Fig. 3.** Minimum spanning tree for Delaunay graph.

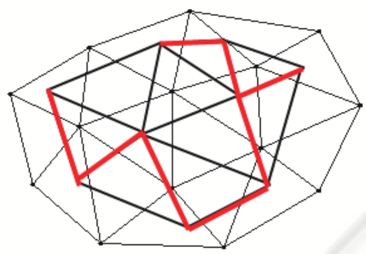

**Fig. 4.** Mesh location in triangulation.

It is known, that it is possible to construct minimum spanning tree for a set $G$ from Delaunay triangulation for $G$ in linear time. Linear time is reached owing to clean-up operation proposed by Cheriton and Tarjan in [5], and to data structure "fibonacci heap", defined by Fredman and Tarjan in [6], [7].

It was experimentally verified that computational complexity of mesh location stage is $O(N)$.

By means of the described algorithm each of the meshes $G_1$ and $G_2$ is located in the triangulation for the other mesh, and it is possible to consider a problem of interpolation for function given on one mesh at points of the other mesh.

### 3.4 Function Interpolation

Let point $V_0(x_0, y_0)$ be located in a certain triangle $\triangle(V_1(x_1, y_1), V_2(x_2, y_2), V_3(x_3, y_3))$: such that $F(x_1, y_1) = f_1$, $F(x_2, y_2) = f_2$, $F(x_3, y_3) = f_3$. For interpolation of function $F$ value at the point $V_0$ linear interpolation and barycentric coordinates can used: $f_0 = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$, here $\lambda_1\ \lambda_2\ \lambda_3 \geq 0$ and

$$\begin{cases} x_0 & = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3; \\ y_0 & = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3; \\ 1 & = \lambda_1 + \lambda_2 + \lambda_3. \end{cases}$$

Results of the described method are shown on Figures 5 - 8. The triangulated surface defined by function $F_1$ on grid $G_1$ is represented by (1) (on the top), the triangulated surface defined by function $F_2$ on grid $G_2$ is represented by (2) (at the bottom of Figure 5) and the surface received after interpolation of function $F_2$ on grid $G_1$ is represented by (3) (at the bottom). As shown on Figures, two triangulations represented by (2) and (3) define the same surface (the bottom one).

Computational complexity of mesh interpolation stage is $O(N)$.

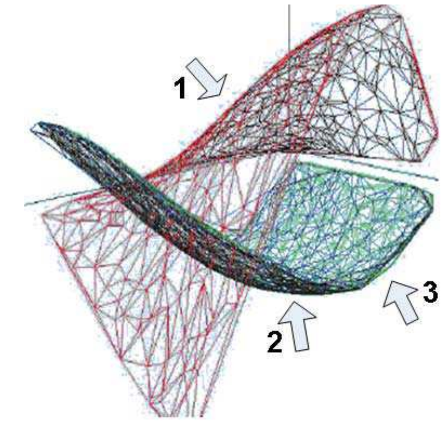


**Fig. 5.** Linear interpolation, $N_1 = N_2 = 1\,000$.

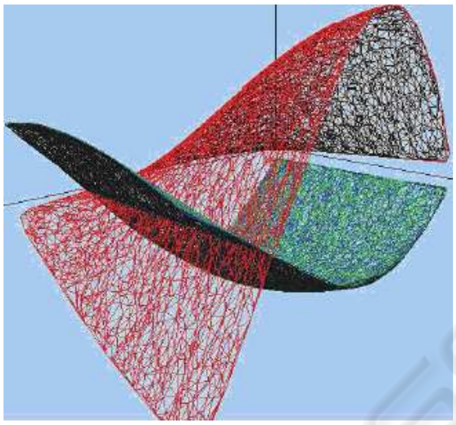

**Fig. 6.** Linear interpolation, $N_1 = N_2 = 10\,000$.

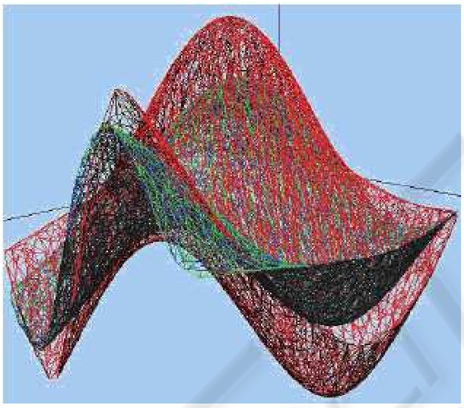

**Fig. 7.** Linear interpolation, $N_1 = N_2 = 10\,000$.

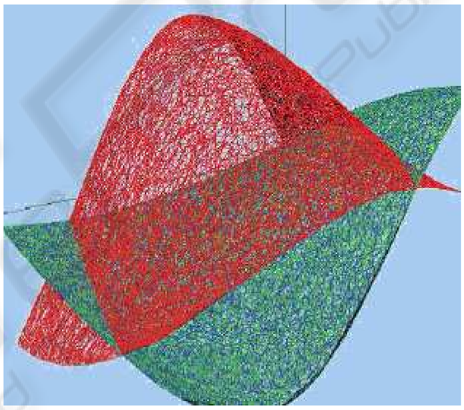

**Fig. 8.** Linear interpolation, $N_1 = N_2 = 15\,000$.

By means of the described method values of function $F_1$ are interpolated at all points of mesh $G_2$ and values of function $F_2$ are interpolated at points of mesh $G_1$.

### 3.5 Function Comparison at Cells of General Triangulation

After interpolation stage values of *two* functions are known at each point of the general mesh $G = G_1 \cup G_2$: one of them has been given, and the second one is received by interpolation.

Let's construct the general Delaunay triangulation $T$ for general mesh $G$. As locations for points of the meshes $G_1$ and $G_2$ in triangles of triangulations for $G_2$ and $G_1$ are known usage of the triangulation merge algorithm proposed by L.Mestetkiy, E.Tsarik in [8] is the most efficient here.

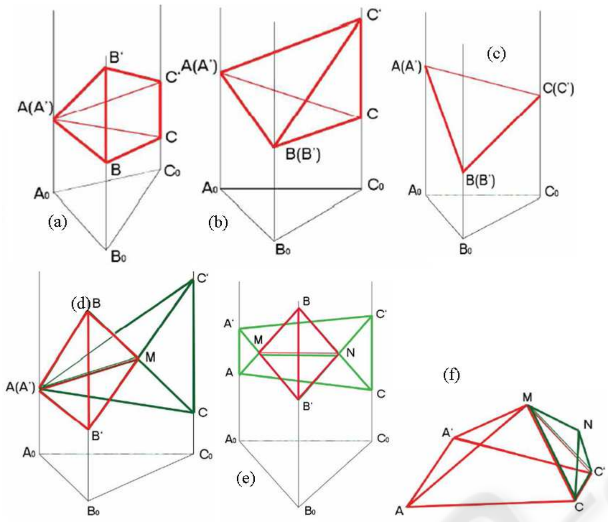


**Fig. 9.** Comparison of functions given at three points: (a) $a = 0$, $b > 0$, $c > 0$; (b) $a = 0$, $b = 0$, $c > 0$; (c) $a = 0$, $b = 0$, $c > 0$; (d) $a = 0$, $b < 0$, $c > 0$; (e) $a > 0$, $b < 0$, $c > 0$; (f) wedge volume as sum of volumes of triangular or quadrangular pyramids.

Let $\triangle A_0B_0C_0$ be a triangle of the general triangulation $T$ and let $\triangle ABC$ and $\triangle A'B'C'$ be spatial triangles corresponding to functions $F_1$ and $F_2$. As a disparity measure of surfaces we will use the sum of volumes of difference between prisms $A_0B_0C_0ABC$ and $A_0B_0C_0A'B'C'$ over all triangles $\triangle A_0B_0C_0$ of the general triangulation $T$.

Let $a$, $b$, $c$ be differences of coordinates of axis $Oz$ of points $A'$ and $A$, $B'$ and $B$, $C'$ and $C$ respectively. We analyze positional relationships of the spatial triangles $\triangle ABC$ and $\triangle A'B'C'$ and consider all possible cases (see Fig. 9). For computing of volume of difference between prisms it is necessary to calculate volume of a pyramid — triangular or quadrangular (see Fig. 9a-c), or total volume of two triangular pyramids (see Fig. 9d), or total volume of a triangular pyramid and a wedge (see Fig. 9e) where wedge volume is searched as the sum of volumes of quadrangular and triangular pyramids (see Fig. 9f).

Summing over all triangles of general triangulation value of difference volume, we obtain the difference measure 2 between the given surfaces.

## 4 Comparison of Human Face Surfaces

As the application of the proposed methodology we have considered a problem of comparison of 3D face models (see Fig. 10).

Using elementary manipulations of one model (shifts and rotations by small angles) it is possible to improve the received result considerably, i.e. to find such model position that two models constitute a maximum matching so that the corresponding disparity measure will be minimal (see Fig. 11).

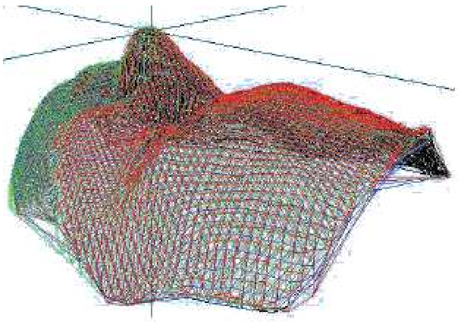

**Fig. 10.** Comparison of 3D face models, disparity measure is $39\,234, 254$.

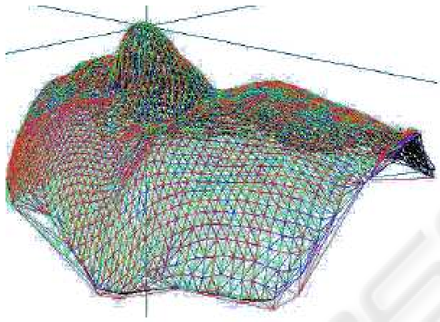

**Fig. 11.** Comparison of 3D face models, disparity measure is $20\,238, 8$.

In [9] the author considered a problem of quantitative estimation of facial asymmetry in 3D models. To solve this problem reflection of the initial model is constructed and two models are compared using the presented approach. The received disparity measure is called initial quantitative estimation of asymmetry. Then the correction stage of facial asymmetry plane is performed and a final value of estimation is received.

## 5 Computing Experiments

The proposed method of surface comparison was implemented, and there also has been made multiple computing experiments for all stages of algorithm.

As experimental estimations have shown, each of stages except stage of triangulation constructions is implemented for linear time in number of mesh points. The stage of triangulation construction is implemented for time $O(N \log N)$, which defines computational complexity of the proposed approach.

Running time for different stages of algorithm during surface comparison are adduced in tables (1)-(3). The three-dimensional portraits consisting approximately from $3\,000$ points were used here. Computing experiments were carried out using AMD Athlon $2\,600+$ processor and $512$ Mb operative memory.

Let us remark that in case one of two models is stored in a database (verification problem) then the total running time will cut in half because construction of Delaunay triangulation and minimum spanning tree for stored model can be implemented during preprocessing stage.

In addition, we have performed computing experiments on 3D face database that estimate approximation accuracy of scanned surfaces. These experiments have showed how much surfaces can be simplified without loss of accuracy for disparity measure (2) by comparing initial triangular meshes and their simplified representation. The problem

**Table 1.** Running time for different stages of algorithm. Comparison of face surfaces consisting of about 3 000 points.

| Stage of algorithm | Time (sec) |
|---|---|
| Construction of two triangulations | 0,124 |
| Construction of two MSTs | 0,203 |
| Location of triangulations | 0,015 |
| Function interpolation | $< 0{,}001$ |
| Construction of general triangulation | 0,031 |
| Computing disparity measure $\int \mid F_1 - F_2 \mid$ | 0,031 |
| Total time | 0,405 |

**Table 2.** Time for minimum spanning tree construction for graph of Delaunay triangulation using Cheriton and Tarjan algorithm.

| Number of points | 20 000 | 40 000 | 60 000 | 80 000 | 100 000 |
|---|---|---|---|---|---|
| Time (sec) | 0,906 | 1,812 | 2,562 | 3,531 | 4,468 |

**Table 3.** Time for location of one mesh in triangles of the triangulation for the other mesh.

| Number of points in the mesh $G_1$ | 10 000 | 25 000 | 50 000 | 75 000 | 100 000 |
|---|---|---|---|---|---|
| Time (sec) | 0,031 | 0,093 | 0,171 | 0,234 | 0,312 |

**Table 4.** Running time of linear interpolation of both functions.

| Number of points in both meshes $G_1$ and $G_2$ | 50 000 | 100 000 | 150 000 | 200 000 |
|---|---|---|---|---|
| Time (sec) | 0,015 | 0,031 | 0,046 | 0,062 |

of measuring error on simplified surfaces were considered in detail by F.Cignoni et al in [10].

## 6 Conclusions

A new approach to 3D objects' surface comparison is proposed in this paper. A new disparity measure between two surfaces is introduced. The approach is based on the piecewise-linear approximation of surfaces using Delaunay triangulations for initial point clouds.

The proposed method has the following advantages: numerical efficiency, possibility of paralleling. Besides, the described approach possesses some universality in comparison with others as it is suitable for comparison of any models given by functions on discrete sets. The proposed measure can be adapted for each concrete application, for example, by means of introducing measure on a surface. So the considered methodology gives mathematical apparatus for construction of common and specific metrics for surface comparison.

## Acknowledgements

The author is grateful to her scientific adviser Prof. Leonid Mestetskiy for useful discussions and important observations. This work is supported by the Russian Foundation for Basic Research (grants 08-01-00670, 08-07-00305-a).

## References

1. Yu. I. Zhuravlev and V. V. Nikiforov, Recognition Algorithms Based on Estimate Evaluation, Kibernetika, No. 3, 111 (1971).
2. Petrenko, D. A., S. A. Triangulation comparison using hash-functions (in russian). In Herald of Tomsk State University 280 (2003).
3. Scvortsov, A. V. Delaunay triangulation and its applications (in Russian). Tomsk (2002).
4. Ernst, M., I. S. Fast randomized point location without preprocessing in two- and three-dimensional delauney triangulation. In Proceedings of the 11th Annual Symposium on Computational Geometry. Los Alamos, New Mexico (1996).
5. Cheriton, D., Tarjan, R. Finding minimum spanning trees. In SIAM J.Comput. (1976).
6. Fredman, M., Tarjan, R. Data Structures and Network Algorithms. Society for Industrial and Applied Mathematics, London (1989), 2nd edition.
7. Tarjan, R. Fibonacci heaps and their uses in improved network optimization algorithms. In Journal of the ACM (1987).
8. Mestetskiy, L., Tsarik, E. Delaunay triangulation: recursion without space division of vertices (in russian). In GraphiCon, International Conference on computer graphics. Moscow (2004).
9. Dyshkant, N., Mestetskiy, E. Asymmetry estimation in 3D faces (in russian). In Intellectual Data Processing'08, pp.94-96, Simferopol (2008).
10. P. Cignoni, C. Rocchini and R. Scopigno Metro: measuring error on simplified surfaces. Computer Graphics Forum, Blackwell Publishers, vol. 17(2), pp. 167-174, (1998).