Proceedings of ALGORITMY 2005
pp. 290-299

FEATURE EXTRACTION OF 2-MANIFOLD USING DELAUNAY TRIANGULATION

JOSEF KOHOUT [*§], TOMAS HLAVATY [*§], IVANA KOLINGEROVA[†§], VACLAV SKALA[‡§]

Abstract. This paper is dedicated to a proposition of a novel feature extraction method for 2-manifold objects in 3D. The topological description was used as a base of the method. The curvatures of the surface are used to obtain the description independent on the position and the orientation of the object. In the addition, the Morse theory and the Delaunay triangulation are combined together to generate a final description that sufficiently identifies the object.

Keywords. feature extraction, 2-manifold, critical points, Delaunay triangulation

AMS subject classification. 68P30, 57R05, 57R15, 51H20

1. Introduction. The modern time still claims to solve more and more problems by computers. It also can be evident from films in which computers or robots can recognize the world around. The recognition, classification, or retrieval issues just belong to one of the most important. Although the mentioned issues seem to be different at first sight, they solve the same problem, namely the similarity comparison of objects. Unfortunately, a definition of similarity in this context does not exist thus it is very hard to answer the question how much are two objects similar. Each person has a unique perception of the world and objects seem more or less different to him or her.

In spite of that, a definition of the similarity is needed in the computer world. It has to be enunciable by a known data structure. The feature extraction that attempts to extrude important points of an input object into a data structure tries to solve this problem. An algorithm that measures the similarity between those structures also has to exist and it is usually based on a heuristic method.

This paper presents a novel method for the feature extraction of 3D smooth manifold objects. The first part of the paper contains a short overview of existing methods for the description of 3D objects and their classification into several basic groups. The second one is dedicated to a mathematical theory that is needed to understand our method. It contains a short introduction into curvatures, the Morse theory and the Delaunay triangulation. Then the paper continues by the description of our method. The last part of the paper describes experiments that verify correctness and applicability of our method and the paper is closed by a conclusion and our future work.

2. Feature extraction. In our case, the feature extraction (shortly FE) can be specified as a shape description of 3D object by a feature vector given by a data structure. This feature vector serves as a search key in similarity comparing. If an unsuitable FE method had been used, the final results would not be usable. Therefore, the following text is dedicated to properties that an ideal FE method should have.

At first, we have to realize that 3D objects can be saved in various kinds of representations such as polyhedral meshes, volumetric data, parametric or implicit equations,

---

[*] This work is supported by the Ministry of Education of the Czech Republic – project FRVS 1342/2004/G1.

[†] This work is supported by the Ministry of Education of the Czech Republic – project MSMT 235200005.

[‡] This work is supported by the European Union – project 3DTV, NoE FP6-PLT-511568.

[§] University of West Bohemia, Department of Computer Science and Engineering, Univerzitni 8, 306 14 Plzen, Czech Republic ({besoft|thlavaty|skala|kolinger}@kiv.zcu.cz).

etc. The method for FE should accept this fact and it should be independent on the data representation (unfortunately, it is very hard to satisfy in practice). The method also should be invariant under transformation such as translation, rotation and scale of the 3D object. Perhaps it is the most important requirement because 3D objects are usually saved in various poses and scales. The 3D object can be obtained either from a 3D graphics program or from a 3D input device. The second way is more susceptible to errors and, therefore, the FE method also should be insensitive to noise. The last requirement is that it has to be quick to compute and easy to index and to compare. The FE is usually implemented in a database that can contain thousands of objects and a rapid response of the system also belongs to main requirements.

The feature method that would have all the mentioned requirements does not probably exist. The most methods try to find a compromise among ideal properties. They can be classified into several basic groups according to mathematical background of the method. An overview of these methods is introduced e.g. in [10]. Let us briefly cite its basic distribution.

The first group *2D view based methods* are based on more explored methods for 2D images, e.g. [12], [4], [14]. The second one *Histogram based methods* decompose objects into several parts from which representing values are computed and represented by a histogram, e.g. [21], [15], [11], [7]. Another group *Topology based methods* contains methods based on a description of an object topology. Generally, the resultant structure representing an object is a graph, e.g. [9], [19]. The last group *Error based methods* is based on the measuring of volume error between objects. These methods have to solve two problems, namely pose estimation and a calculation of a volume error between objects, e.g. [16].

Our method can be included in the topology based methods. These methods provide the best description of the object's shape. A modified Delaunay triangulation was used as a describing structure where the vertices of the triangulation are found on the base of the Morse theory and surface curvatures. The curvatures were selected purposely because of their invariance to transformations. Furthermore the shape of an object can be described by curvatures exhaustively. The whole method is described later in more details.

3. Calculation of Curvatures. A curve in the space can be described in several different ways. A unique description of a curve can be achieved by a curvature ${}^1k$ and torsion ${}^2k$ that can describe an arbitrary shape of the curve exhaustively (except of its position). A similar description for a surface is formulated in this section. However, a theory about surfaces and their curvatures is very wide and, therefore, only basic facts needed to define curvatures are shortly described here. A more detailed theory can be found in a book dedicated to the differential geometry.

Let us suppose that we have a smooth surface defined by the parametric equation:

(1) $$\mathbf{r}(u,v) = (x(u,v), y(u,v), z(u,v)), \quad [u,v] \in \Omega$$

and a curve on this surface defined by a parametric equation:

(2) $$u = u(t),\ v = v(t)$$

Let us also suppose that all required conditions are fulfilled. When the curve and the surface are defined, we can express the first and the second fundamental forms:

(3) $$I\!: \mathrm{d}s^2 = E\,\mathrm{d}u^2 + 2F\,\mathrm{d}u\,\mathrm{d}v + G\,\mathrm{d}v^2, \quad II: \ -\mathrm{d}\mathbf{r}\cdot\mathrm{d}\mathbf{n} = L\,\mathrm{d}u^2 + 2M\,\mathrm{d}u\,\mathrm{d}v + N\,\mathrm{d}v^2$$

where **r** represents the equation of the surface and **n** represents the normal of the surface. The letters E, F, G and L, M, N denote coefficients of the given fundamental form (see [17] for more details).

These forms describe a surface in the same way as the curvature and the torsion describe a curve. However, additional coefficients are needed in practice. Perhaps one of the most important is the radius of the curvature and the normal curvature in the given point.

All curves on a surface which pass through a point **p** have the same osculating plane at **p** and the same curvature at **p**. The radius of the curvature of the curve of the section cut by a plane passing through a point **p** of the surface **r**(*u,v*) is :

$$(4) \qquad r = \frac{E\,\mathrm{d}u^2 + 2F\,\mathrm{d}u\,\mathrm{d}v + G\,\mathrm{d}v^2}{\left|L\,\mathrm{d}u^2 + 2M\,\mathrm{d}u\,\mathrm{d}v + N\,\mathrm{d}v^2\right|}\cos\vartheta\,, \qquad (\,II \neq 0\,)$$

where ϑ is the angle between the plane of section and the normal to the surface at **p** (see Fig. 1). Note that the curvature is generally defined as the inverse of its radius and if we speak about a normal radius we can also speak about a normal curvature.

Let us suppose a regular point **p** and a tangent line υ in this point are given. Then according to Meusnier's theorem, for all curves on the surface having the tangent line υ in the point **p** the centre of the radius of curvature lie on the circle with the radius R_n. The circles of curvatures of all these curves lie on the sphere of radius R_n (see Fig. 1).

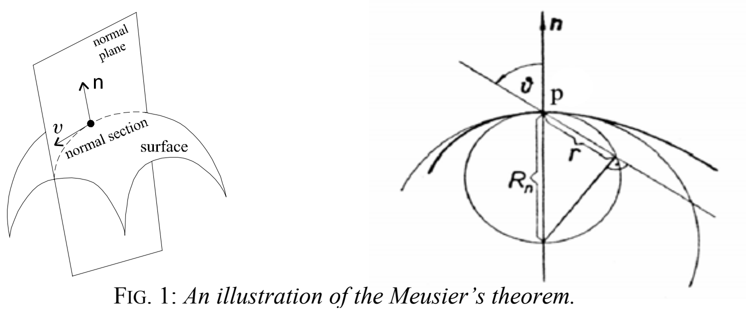


FIG. 1: *An illustration of the Meusier's theorem.*

The curves of normal sections of a surface for which the corresponding normal curvatures have extreme values are called the principal curves of normal section and their radii of curvature R_1 and R_2 are called the principal radii of curvature at the point considered on the surface.

Another Euler theorem says that the normal curvature $1/R_n$ of a curve at a point **p** is given by the formula:

$$(5) \qquad \frac{1}{R_n} = \frac{\cos^2\delta}{\varepsilon_1 R_1} + \frac{\sin^2\delta}{\varepsilon_2 R_2}\,,\ \varepsilon_1, \varepsilon_2 = \pm 1$$

where δ is the angle between the plane of the curve and the plane of the first principal curvature in their normal sections. The constants ε_1, ε_2 have a geometric meaning and its value determines a type of point **p**. Three basic types are defined: parabolic (R_1 *or* $R_2 \rightarrow \pm\infty$), elliptic ($\varepsilon_1 = \varepsilon_2$) and hyperbolic ($\varepsilon_1 \neq \varepsilon_2$) point. The Dupin's indicatrix helps us to determine the radius of the normal curvature in an arbitrary direction as is shown on the Fig. 2. The direction is given by the angle δ measured from the direction of the first principal curvature.

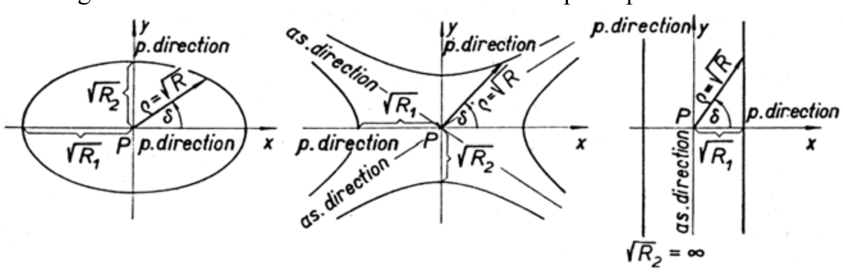


FIG. 2: *An illustration of the Dupin's indicatrix (taken from [17]).*

Last terms that are mentioned here are the Gaussian curvature K and Mean curvature H. They are derived from principal curvatures and can be expressed as:

$$(6) \quad K = \pm\frac{1}{R_1}\cdot\frac{1}{R_2} = \frac{LN - M^2}{EG - F^2}\,, \qquad H = \frac{1}{2}\left(\frac{1}{\varepsilon R_1} + \frac{1}{\varepsilon R_2}\right) = \frac{EN - 2FM + GL}{2(EG - F^2)^2}\,,\ \varepsilon = \pm 1$$

The values of these curvatures are usually used for a classification of the points on the surface. The following table shows the meaning of their values.

	K < 0	K = 0	K > 0
H < 0	saddle ridge	ridge	peak
H = 0	minimal surface	flat	none
H > 0	saddle valley	valley	pit

A short introduction about curvatures was mentioned. The theory about the differential geometry is, indeed, wider. In our case, it is only important to know a geometric meaning of the mentioned terms and the fact that the curvatures are invariant to transformation.

4. Morse theory. In the 1930's M. Morse proved so called Morse theory. Originally, Morse theory was applied on a class of mathematical objects called smooth manifolds (such as a plane, a circle, the surface of a sphere, etc.) and provides a tool for understanding the topology of objects with limited information. It is based on the Taylor series. This theory is briefly introduced in the following text (for more details see e.g. in [1], [2], [13], [18]).

Let $f(x)$ be a real function and x be a real value. For x near a given value x_0 the value of the function $f(x)$ with infinitely many derivatives at x_0 can be expressed by the Taylor series:

(7) $$f(x) = c_0(x_0) + c_1(x_0)\cdot(x - x_0) + c_2(x_0)\cdot(x - x_0)^2 + \dots, \quad c_k(x_0) = \frac{f^{(k)}(x_0)}{k!}$$

Suppose $x_0 = 0$ (i.e., it represents the origin), then the equation can be simplified:

(8) $$f(x) = c_0(x_0) + c_1(x_0)\cdot x + c_2(x_0)\cdot x^2 + \dots$$

If $c_0(0) = 0$ and $c_1(0)$ is different from zero, then for $\boldsymbol{x}$ close to the origin the equation (8) can be approximated as:

(9) $$f(x) \approx c_1(x_0)\cdot x$$

That point is called a *regular point* of $f(x)$.

If $c_0(0) = 0$, $c_1(0) = 0$ and $c_2(0)$ is different from zero, then for x close to the origin the equation (8) can be approximated as:

(10) $$f(x) \approx c_2(x_0)\cdot x^2$$

That point is called a *non-degenerate critical point* of $f(x)$.

If $c_0(0) = 0$, $c_1(0) = 0$, $c_2(0) = 0$ and $c_3(0)$ is different from zero, then for x close to the origin the equation (8) can be approximated as:

(11) $$f(x) \approx c_3(x_0)\cdot x^3$$

That point is called a *degenerate critical point* of $f(x)$. Note that degenerate or non-degenerate critical point is also called a *singularity of the function* $f(x)$.

Let us think in a general way. Let us suppose that x is not a single variable but it is a point in E^N. Then the point $\mathbf{x}$ is non-degenerate if the gradient of the function vanish at this point, i.e. $grad(f(\mathbf{x})) = \mathbf{0}$, and the determinant $det(\mathrm{H}(f(\mathbf{x}))) \neq 0$.

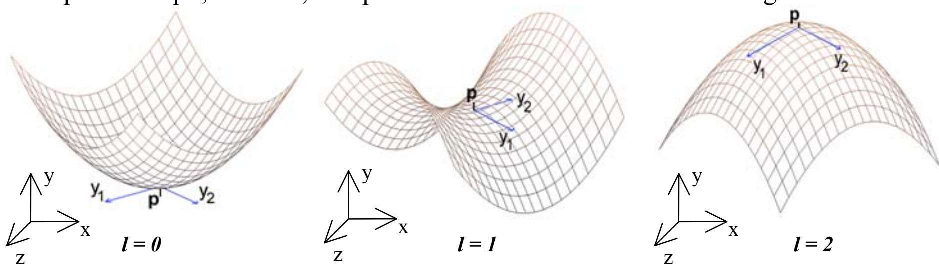
Let us denote the number of negative eigenvalues of Hessian $\mathrm{H}(f(\mathbf{x}))$ as the index l. Then the following lemma can be introduced. Let $\mathbf{p} \in E^N$ be a non-degenerate critical point of a Morse function $f(\mathbf{x})$ with the index l and let $c = f(\mathbf{p})$. Then there is a local coordinate system $\mathbf{y} = (y_1, y_2, \dots, y_N)$ in a neighbourhood U of the point $\mathbf{p}$, in which $\mathbf{p}$ represents the origin of this coordinate system:

(12) $$f(\mathbf{y}) = c - y_1^2 - y_2^2 - \dots - y_l^2 + y_{l+1}^2 + \dots + y_N^2$$

A practical meaning of this theory can be shown on a 2D differentiable manifold ($N = 2$) where the Morse function $f(\mathbf{x})$ represents the height from an xz plane. A surface shape of neighbourhood U of point $\mathbf{p}$ can be estimated on the basis of the index l (see Fig. 3). Note that in this case the Hessian matrix has the rank 2 with the following values:

$$\mathbf{H}(f(\mathbf{x})) = \begin{bmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial z \\ \partial^2 f/\partial z \partial x & \partial^2 f/\partial z^2 \end{bmatrix} \tag{13}$$

According to eigenvalues of the matrix **H** we can classify type of the non degenerate critical point as a pit, a saddle, or a peak. It is also demonstrated in the Fig. 3.

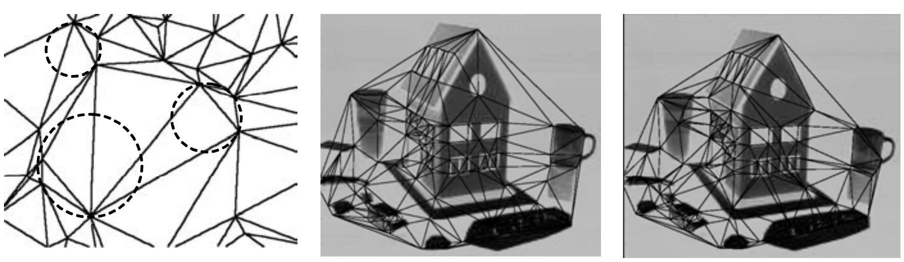


$f(\mathbf{y}) = c + y_1^2 + y_2^2$ $f(\mathbf{y}) = c - y_1^2 + y_2^2$ $f(\mathbf{y}) = c - y_1^2 - y_2^2$

FIG. 3: *Examples of surface shapes of neighborhood U for different values of eigenvalues* ***H****(* *f(****x****)), the Morse function f(****x****) is represented as a height from a xz plane.*

In our method, we combine mentioned theories together. The combination of Gaussian and mean curvatures is used as the Morse function *f*(**x**). Then we can find important points with the help of Hessian matrix of the given function (i.e., points in which are extreme values of the Gaussian or mean curvature). These points represent important changes of the surface curvature and are independent on the position and orientation of the object.

5. Delaunay triangulation on the object surface. In previous sections, we described how to evaluate the given model in 3D and how to find points with some important features on its surface. Now, we need to find a proper connectivity of the chosen important points, i.e. we need to encode the simplified topology of the model.
In this section, we introduce the Delaunay triangulation that we decided to use for the description of the object topology. The Delaunay triangulation DT(S) of a set of points S in 2D was proposed by a Russian scientist B. N. Delone [5]. It is such a triangulation that the circum-circle of any triangle does not contain any other point of *S* in its interior – see Fig. 4a.



a) An example of the Delaunay triangulation with several circum-circles.

b) The Delaunay triangulation of significant points detected in two similar images (taken from [1]).

FIG. 4: *The Delaunay triangulation and its use for pattern recognition.*

Due to good properties [8] of the Delaunay triangulation, it is one of the fundamental topics in the computational geometry and it is used in many areas such as terrain modeling (GIS), scientific data visualization and interpolation, robotics, pattern recognition, meshing for finite element methods (FEM), natural sciences, computer graphics and multimedia, etc.

One of these properties is that if no four points lie on a circle, the *DT*(*S*) is unique. Let us assume we already know how to construct two-dimensional *DT*(*S*) on the object surface, i.e.

for our chosen points in 3D. Then if we have two objects and one of them is translated or rotated in comparison with the second one, the Delaunay triangulations of important points of these objects are exactly the same. It is clear that the first object can be deformed or it can have even different number of points than the second object. In such a case, indeed, Delaunay triangulations differ. Fig. 4b shows an example of two similar images and the *DT*(*S*) of their significant points. We can identify identical parts in both triangulations. It seams that we can evaluate the similarity of two Delaunay triangulations using some sophisticated heuristic.

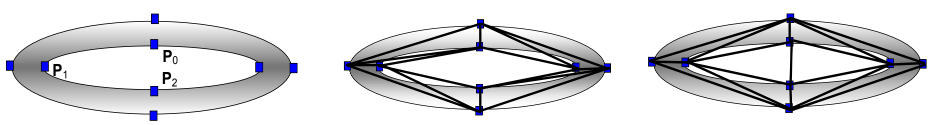
At present, we have no such an approach, however, we would like to investigate this problem in our future work. Nevertheless, there exist several papers dealing with the pattern recognition based on *DT*(*S*) showing us that finding this heuristics is not unsolvable problem, e.g. [1] or [6].

The Delaunay triangulation also contains the most equiangular triangles of all possible triangulations, i.e. it maximizes the smallest angle. That is important for purposes of interpolation because an interpolation of values on too narrow triangles is very often improper because of the numerical instability.

Let us also mention that the Delaunay triangulation can be computed in $O(N \cdot \log(N))$ time in the worst case (N is the number of points to triangulate). Algorithms with $O(N)$ expected time also exist. Su et al. [20] brought the comparison of the best-known algorithms.

We have just described the Delaunay triangulation of points in 2D and its properties. Generalization into 3D is straightforward: the *DT*(*S*) of points in 3D is a tetrahedronization such that circum-sphere of any tetrahedron does not contain any other point of *S* in its interior. The properties of the *DT*(*S*) in 2D and 3D are similar.

As our input is a set of important points in 3D, we could simply use any of existing algorithms for the construction of the Delaunay triangulation in 3D in order to get the required connectivity. Such a result would, indeed, somehow encode the topology of a given object. The problem is, however, that such an approach might connect two points that are quite distant in the meaning of curvatures behavior. Fig. 5 shows an example of such unwanted connectivity. Although the first principal curvature increases on a surface curve from the point $\mathbf{p}_0$ to $\mathbf{p}_1$ and then it monotonously decreases until the point $\mathbf{p}_2$ is reached, the edge $\mathbf{p}_0, \mathbf{p}_2$ exists in the resulting triangulation.



a) A deformed torus and its important points. *b) The expected triangulation.* *c) The resulting triangulation.*

FIG. 5: *The connectivity problem.*

The solution is to construct a 2D triangulation of important points in 3D that is unfolded on the surface of a given object. In such a case, we can use Cartesian distances no more. Distances have to be measured on the surface, i.e. the distance of two points is the length of the shortest curve connecting these points that can be found on the surface. As the surface is covered by triangles, we measure the length of a poly-line formed by edges in the triangulation.

It is a very difficult task to find a mapping function such that it maps all points into a two dimensional parametric domain, especially if we stress additional requirements on this function such as preservation of distances between two points, angles formed by two vectors or area sizes. There exist many papers dealing with this problem. Many solutions are sophisticated approximations limited to some kind of objects; very often they use a unit sphere as the output domain.

Let us, therefore, propose a simple heuristics suitable for models represented by triangular meshes. The basic idea is to exploit the original triangular mesh and reduce it successively in order to obtain a surface triangulation with important points only. As this triangulation is an arbitrary one, we afterwards apply local improvements to get the Delaunay triangulation.

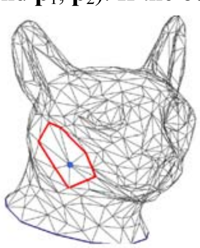
During the entire process, each edge e in the current triangulation represents a poly-line path (on the surface) between the ending points of this edge. The edge e is evaluated by the cost function $g(e)$ equal to the length of appropriate poly-line. At the beginning, indeed, $g(e)$ = the Cartesian length of the edge e. The goal is to minimize $g(e)$ for each edge.

After the initialization, in the main stage, we successively remove unimportant points (in random order). The removal of an unimportant point, say **p**, consists of two steps: cavity construction and cavity retriangulation. In both steps, the coordinates of points are not considered. Instead of it, we use the knowledge of their connectivity and the evaluation of edges between them.

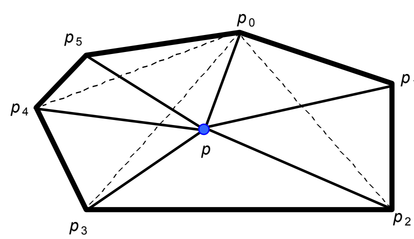
In the first step, all triangles sharing the point **p** are found. Their edges that do not coincide with this point form a 2D polygon – see Fig 6. Let us suppose that there exist at most three candidates for the shortest path between vertices $\mathbf{p}_i$ and $\mathbf{p}_j$: the first two possible paths lead through edges of the polygon, one in the clockwise and another in the counter-clockwise orientation. The third path goes through the point **p**. It is clear that it is not always true that there is no other shorter path on the surface. Therefore, the resulting triangulation is only an approximation of the exact solution.

For each vertex $\mathbf{p}_i$ on this polygon a set of all possible diagonals is constructed and these temporary edges are evaluated as follows. We set the length of temporary edge connecting vertices $\mathbf{p}_i$ and $\mathbf{p}_j$ as the sum of lengths of edges $\mathbf{p}_i, \mathbf{p}$ and $\mathbf{p}_j, \mathbf{p}$. E.g., in Fig. 6b, $g(\mathbf{p}_0, \mathbf{p}_2) = g(\mathbf{p}_0, \mathbf{p}) + g(\mathbf{p}_2, \mathbf{p})$. The result is that we have three different paths from $\mathbf{p}_0$, $\mathbf{p}_2$ with likely different length.

When we remove $\mathbf{p}_1$ the number of paths decreases. In such a case, we have to verify whether the “direct” path (i.e. the edge $\mathbf{p}_0$, $\mathbf{p}_2$) is shorter than its counterpart with one stop (i.e. edges $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_1$, $\mathbf{p}_2$). If the outcome is negative, the length has to be updated.



a) An unimportant point and the cavity caused by the removal of this point.



b) Polygonal cavity and all possible diagonals that can be constructed from the point p_0.

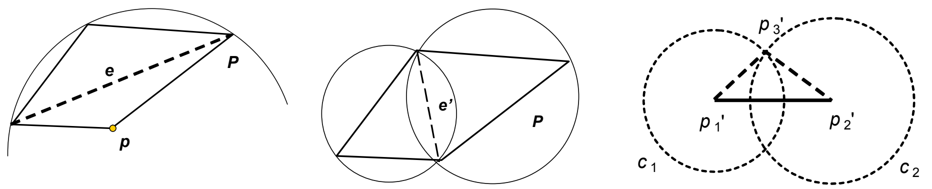
FIG. 6: *The construction of the cavity.*

At the end, all triangles sharing the point **p**, all edges going to this point and the point itself are removed from the triangulation and we obtain a cavity.

In the second step, we have to retriangulate this cavity. We have generated all potential edges in previous step. However, they intersect one another and we have to decide which to choose. As the distance between two points on the surface is equal to the length of shortest surface path (i.e. poly-line) between these two points, it is necessary to ensure that the sum of lengths of edges inside the cavity is as small as possible. The brute-force is to generate all possible triangulations and then to choose the minimal one. However, this approach is very slow, it can be proved that it is the NP problem. Fortunately, we do not need an exact solution. The greedy heuristic is sufficient enough.

The result of the proposed decimation algorithm is an arbitrary surface triangulation in 3D. As we used the greedy approach for the retriangulation, thus preferring the shortest lengths, triangulations of similar objects will be likely similar as well. If we want to get the Delaunay triangulation on surface, we need to apply local transformations in such a manner to ensure that all edges are locally optimal.

In a plane, we say that the edge *e* is locally optimal if and only if the polygon *P* formed by two triangles sharing this edge is not convex or the circum-circle of one of these two triangles does not contain the far point of the second triangle in its interior. If the edge *e*, i.e. the first diagonal of the convex polygon *P*, is not optimal, it is removed from the triangulation and the second diagonal *e′* is inserted into the triangulation – see Fig. 7a,b. Then it is necessary to check all four edges of the polygon *P* on optimality.



a) Non-optimal edge e. *b) Optimal edge e′.* *c) The construction of the point P_3'.*

FIG. 7: *Flipping the diagonals to convert an arbitrary triangulation into the Delaunay one.*

As our input is a set of triangles in 3D, we have to modify the check for optimality in the following way. Let us have triangles $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_4$ in 3D. We can map the common edge $\mathbf{p}_1$, $\mathbf{p}_2$ into a new edge $\mathbf{p}_1'$, $\mathbf{p}_2'$ in 2D, where $\mathbf{p}_1' = (0, 0)$, $\mathbf{p}_2' = (d, 0)$ and *d* denotes the length of an edge connecting $\mathbf{p}_1$ and $\mathbf{p}_2$. Let us remark that by the length of the edge *e*, we mean the value $g(e)$. For the point $\mathbf{p}_3$, let us construct then two circles c_i centered in $\mathbf{p}_i'$ with radius equal to the length of edge $\mathbf{p}_i$ and $\mathbf{p}_3$. The point $\mathbf{p}_3' \in$ 2D is taken as one of two intersections of these two circles – see Fig. 7c. The point $\mathbf{p}_4$ is mapped in a similar way. Indeed, we pick the intersection in the opposite half-plane.

Let us briefly discuss how to deal with noisy data sets. It is obvious that many "important" points are detected. When the triangulation of these points is constructed, we can inspect them in order to find out whether we really need them all. If the value (in our case curvatures) stored in the point **p** does not differ too much from values stored in points connected with **p** and, moreover, the total area covered by triangles sharing this point is small enough, then we can additionally remove the point **p** from the triangulation.

The proposed solution that we have just described is simple and it can work quite fast, especially for large number of important points. On the contrary, our approach is limited to models represented by surface triangular meshes. As this representation is, however, very often used, this drawback is not serious.

6. Experiments. In previous sections, we proposed an approach that can find the description of the shape of a given model in 3D. The result of the model evaluation is the Delaunay triangulation of points that are defined by a significant change of the Gaussian and Mean curvatures. In each point, a normal vector, vectors of principal curvatures and its type (i.e. elliptic, parabolic or hyperbolic point) are stored. In this section, we show that such a description of object is sufficient. For better understanding, we present also results for planar curves. Let us note that in 2D, there is no problem of connectivity of important points and also we have only one curvature.

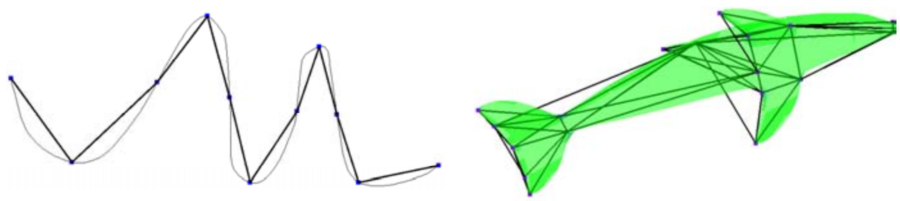
Fig. 8 shows examples of an open planar curve and a model in 3D. It shows also behaviours of their curvature functions. Fig. 9 shows the position of points with the significant change of the curvature, i.e. points where the curvature reaches a local extreme, and their connectivity. The number of these points depends on the level of smoothing of the curvature function and can be further reduced in post-processing.

The proof whether the proposed description is sufficient, i.e. that it encodes both geometry and topology of a model properly, can be done by backward reconstruction of the model from this description. Fig. 10 shows several examples of planar curves reconstruction based on Bezier bicubic splines weighted by curvatures. The detail description is out of the scope of this paper. It can be seen that even for a small number of points the shape of curve is well preserved. To demonstrate the strength and necessity of curvatures, right image in Fig. 10 shows also the result of reconstruction based on cardinal splines (the knowledge of curvatures is not exploited there). In our opinion, the result is poor.

a) An open planar curve and its curvature function (below). The correspondence between the curve and its curvature function is highlited.

b) The dolphin model shaded according to Gaussian curvature with the vectors of principal curvatures.

FIG. 8: *Examples of models in 2D and 3D and the behavior of curvatures.*



a) An open planar curve. *b) The dolphin model.*

FIG. 9: *Important points and their connectivity.*

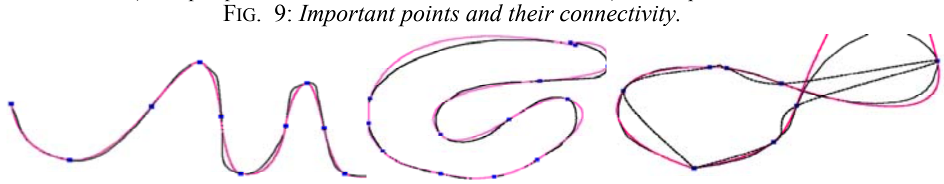


FIG. 10: *Examples of curves and their reconstruction using the proposed description. Right image shows also a cardinal spline of important points.*

Unfortunately, we have no example of reconstructed surface yet. The problem in 3D is, without any doubt, more complex. However, the description of a model in 3D contains more information (e.g. types of important points, etc.) that can be used for a successful reconstruction. As there is also no significant difference between 2D and 3D theory, let us claim that the proposed description of objects is sufficient.

7. Conclusion and future work. In this paper, we have proposed a method for describing a given smooth 2-manifold model in 3D. This method is based on the Delaunay triangulation of points on the surface with important change of curvature. The use of curvatures ensures invariance of the description to transformations and the Delaunay triangulation uniquely encodes model topology.

Our future work, indeed, includes a verification of the description in 3D and then we focus on developing a method suitable for the comparison of these descriptions in order to decide whether the appropriate objects are similar or not. This task involves finding a method for the comparison of triangulations. We would like also to investigate problems related to processing of noisy data sets.

REFERENCES

[1] X. YU H. BAI, E. R. HANCOCK, *Graph Matching using Spectral Embedding and Alignment*, in Proceedings of 17th International Conference on Pattern Recognition, Cambridge, Vol. 3, (2004).

[2] S. BIASOTTI, M. MORTARA, M. SPAGNUOLO, *Surface compression and reconstruction using Reeb graphs and shape analysis*, in Proc. of Spring Conf. on Computer Graphics, (2000), 175-184.

[3] K. L. BOYER, R. SRIKANTIAH, P. J. FLYNN, *Salience Sequential Surface Organization for Free-Form Object Recognition*, Computer Vision and Image Understand., Volume 88, Number 3, (2002).

[4] CH. M. CYR, B. B. KIMIA, *3D Object Recognition Using Shape Similarity Based Aspect Graph*, International Conference on Computer Vision, (2001).

[5] B. DELAUNAY, *Sur la sphere vide*, Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk 7 (1934), 793-800.

[6] A. M. FINCH, E.R. HANCOCK, *Matching Deformed Delaunay Triangulations*, in Proceedings of International Symposium on Computer Vision, IEEE, Coral Gables, Florida, (1995), 31-36.

[7] T. FUNKHOUSER, P. MIN, M. KAZHDAN, J. CHEN, A. HALDERMAN, D. DOBKIN, *A Search Engine for 3D Models*, ACM Transactions on Graphics, Volume 22, Issue 1, (2003), 83 – 105.

[8] J. E. GODMAN, J. O'ROURKE, *Handbook of discrete and computational geometry*, CRC Press, (1997).

[9] M. HILAGA, Y. SHINAGAWA, T. KOHMURA, T. L. KUNII, *Topology Matching for Fully Automatic Similarity Estimation of 3D shapes*, SIGGRAPH, (2001), 203-212.

[10] T. HLAVATY, V. SKALA, *A Survey of Methods for 3D Model Feature Extraction*, Bulletin of IV. Seminar "Geometry and Graphics in Teaching Contemporary Engineer", (2003), 5-8.

[11] M. KAZHDAN, B. CHAZELLE, D. DOBKIN, A. FINKELSTEIN, T. FUNKHOUSER, *A Reflective Symmetry Description for 3D Models*, To appear in Algorithmica, (2003).

[12] J. LÖFFLER, *Content-based Retrieval of 3D models in Distributed Web Databases by Visual Shape Information*, int. Conf. on Information Visualisation, (2000).

[13] J. MILNOR, *Morse theory*, Princeton University Press. New Jersey, (1963).

[14] P. MIN, J. CHEN, T. FUNKHOUSER, *A 2D Sketch Interface for a 3D Model Search Engine*, SIGGRAPH 2002 Technical Sketches, (2002).

[15] F. Mokhtarian, N. Khalili, P. Yuen: Multi scale free form 3D object recognition using 3D models. Image and Vision Computing, 19, (2001), 271-281.

[16] M. NOVOTNI, R. KLEIN, *A Geometric Approach to 3D Object Comparison*, Int. Conf. on Shape Modeling and Applications, SMI, (2001), 154-166.

[17] K. REKTORYS AND AT AL., *Survey of Applicable Mathematics*, (in Czech) Nakl. Prometheus, ISBN 8085849720, (1995).

[18] Y. SHINAGAWA, T. L. KUNII, Y. L. KERGOSIEN, *Surface Coding Based on Morse Theory*, IEEE Computer Graphics and Applications, Vol. 11, No. 5, (1991).

[19] K. SIDDIGI, A. SHOKOUFANDEH, S. J. DICKINSON, S. W. ZUCKER, *Shock graphs and shape matching*, Int. J. Computer Vision, (1998), 222-229.

[20] P. SU, R. L. DRYSDALE, *A Comparison of Sequential Delaunay Triangulation Algorithms*, in Proceedings of 11th annual symp. on Computational geometry, ACM Press, New York, USA, (1995), 61-70.

[21] J. W. H. TANGELDER, R. C. VELTKAMP, *Polyhedral Model Retrieval Using Weighted Point Sets*, International Journal of Image and Graphics, Vol. 3, No. 1, (2003), 209-229.

[22] Y XIAO, H. YAN, *Text region extraction in a document image based on the Delaunay tessellation*, Pattern Recognition, Vol. 36, No. 3, (2003), 799-809.