International Journal of Pure and Applied Mathematics
Volume 18 No. 1 2005, 1-15

A NEW TOOL TO COMPUTE 3D SKELETONS

Dominique Faudot[1] [§], Dominique Rigaudiére[2]
[1,2]LE2I, Informatique
Université de Bourgogne
BP 47870, Dijon Cedex, 21078, FRANCE
[1]e-mail: faudot@u-bourgogne.fr
[2]e-mail: rigaudiere@u-bourgogne.fr

Abstract: The skeleton of a set of points is a continuous or discrete form schematizing the global shape of the cloud. There are many methods to build a skeleton starting from a cloud of points. We present a new one based on binary or ternary trees.

AMS Subject Classification: 05C05
Key Words: 3D skeletons, triangulation, binary and ternary tree

1. Introduction

The idea of form is fundamental to our understanding of the world. Forms allow us to identify objects. The most complex objects are modelled through many simple primitives. Our brain finds it relatively easy to decompose an object into several primitives. However, decomposition of an object as an automatic process is more difficult. A skeleton exists within such a process. A skeleton has the same type of homotopy as that of the form. It is a simple representation of its symmetric axes. A skeleton is a thin form, centered in this object and it describes the form. It can be seen as a medial axis [12]. For instance, Figure 1 represents a skeleton of a box in 3D.

Received: February 24, 2004

[§] Correspondence author

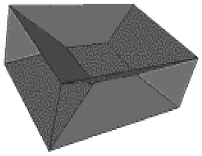

Figure 1: A skeleton of a box

Blum [6] has used a skeleton to create a descriptor of a form and then to sort or measure parameters of an object.

One of the most intuitive formulations of a skeleton was given by Mayya [19], i.e. it is a propagation of a wave from the outline of the object to the centre. The skeleton is then the set of points where waves meet. Moreover, the period of propagation provides the distance between border and centre.

The maximum balls rigorously define a skeleton.

Definition 1. A ball B, whose radius is r, included in an object, is said to be maximal if there is no other ball B', whose radius $r' > r$, is included in the object and B'⊃B (see Figure 2).

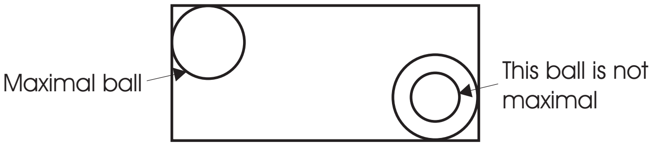

Figure 2: Definition of a maximal ball

Definition 2. Let S be a form and δS the border of S. The skeleton of S, S$_k$(S), is the set of centers of the maximal balls in S.

The skeleton of a form must verify several properties:

- it is included and centered in the object,

- it is thin and has the same type of homotopy of the object,

- the transform generating a skeleton is reversible.

Many methods are used to compute or extract a skeleton. These are sorted in three families: mathematical morphology, distances' map and analytical computation of symmetrical axes.

Mathematical morphology provides a skeleton through the operations of dilation and erosion and through Lantuejoul's Theorem:

$$Sk(X) = \bigcup_{\lambda>0} \bigcap_{\lambda>0} \left(E^{\lambda B}(X) - O^{\gamma B} \left(E^{\lambda B}(X) \right) \right),$$

where B is a unitary ball, O is an open and E is the operation of erosion. In practice, we use

$$Sk(X) = \bigcup_{\lambda>0} \left(E^{\lambda B}(X) - O^{B}\left(E^{\lambda B}(X) \right) \right),$$

e.g. when we compute the skeleton of a rectangle as in Figure 3.

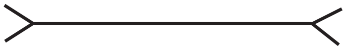

Figure 3: Skeleton of a rectangle

These operations of erosion and dilation are reversible. That is to say, we are able to construct the object from the skeleton, using the operation of extinction.

However, these methods are used in a continuous working space. Nevertheless, we are working fluently in a discrete space. Other works [1], [9] have looked for simple points. These points may be removed while preserving the topology of the object (see Figure 4).

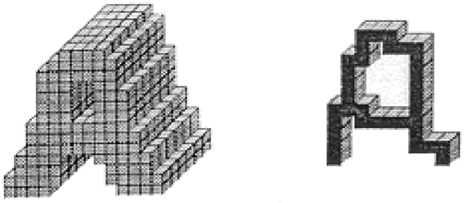

Figure 4: Discreet skeletons

The second family of methods of to extract skeleton requires the computation of a distances' map, which gives for every point in an object, the distance of the nearest point of the border [32], [5].

Definition 3. Let S be an object and P a point of S. $\forall P \in S : \rho(P) = \min\left(d(P,Q)\right)$ with $Q \in \partial S$. The set of $\rho(P)$is called the map of distances of S.

We keep out the local maximum of the distances' map. The links of the previous elements (maximum) gives a continuous skeleton of the object. The Euclidean distance is not the best distance [24]. Therefore, the distance of Chanfrein [29] or chess is used. But these skeletons do not produce "good" results: they do not look like a wave.

Other methods seem to give good results like [15], which proposes a decomposition of the object in parallel blocks in computing a representation with vectors.

Mestelskii [20] Evans and al. [14] compute 2D maximal disks within the form, extracted from the bisectrices of the elements of the polygonal representation of the form. Therefore, the form may be constructed using thick curves (Figure 5). In [14] a tree is generated representing the skeleton.

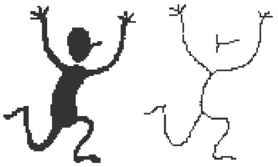

Figure 5: A 2D image and its simplified skeleton

Nevertheless, these methods only work in 2D. Sherbrooke and et al. [31] compute a 3D skeleton with a graph of the adjacency of the faces, vertices and edges.

Many recent works are merging to a Voronoï diagram or Delaunay triangulation [30]. Our approach is the same, and we will present it in this paper.

In Section 2, we will present a Voronoï diagram and Delaunay triangulation from which we will extract a skeleton, as explained in Section 3.

Then we will present our new 2D method based on a tree. This method first builds a triangulation of a set of points. Then it constructs a tree with this triangulation and finally computes the skeleton using the nodes of the tree. Our method is based on the bijection between the set of points, the triangulation and the skeleton. We will extend our method to 3D. Finally, we will outline some problems caused by the triangulation of Delaunay.

2. Skeleton Extraction with Voronoï Diagrams and Delaunay Triangulation

The Voronoï diagram of a set of points divides the space following the rule of the nearest neighbor.

Definition 4. (George et al, [17]) Let M be a set of n points in E^d. M_1,..., M_n are called a site of Voronoï.

A region of Voronoï $\mathrm{V}(\mathrm{M}_i)$ is

$$V(M_i) = \left\{ X \in E^d / d\left(X, M_i\right) \leq d\left(X, M_j\right) \ \forall j \neq i \right\},$$

where d is the Euclidean distance. The set of $\mathrm{V}(\mathrm{M}_i)$ is the diagram of Voronoï.

V(M$_i$) may be seen as the set of centers of maximal balls passing through M$_i$, whose interiors do not contain any other point of M. The Delaunay triangulation is the dual of Voronoï triangulation (see Figure 6).

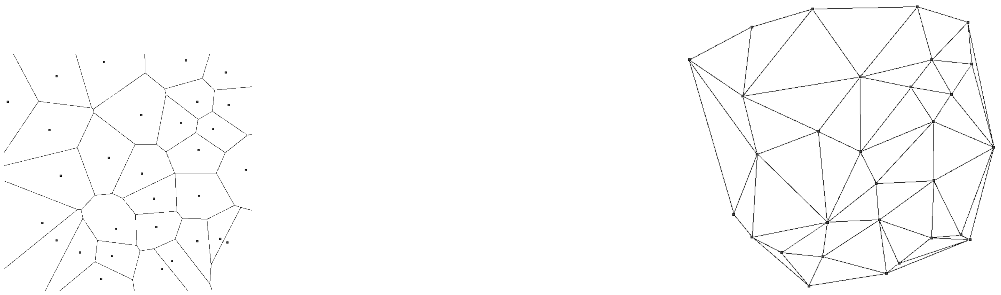

Figure 6: A Voronoï diagram and its triangulation of Delaunay

Every face of the Delaunay triangulation in M contains a circumscribed sphere that does not contain any M$_i$. This triangulation is the best one for the properties of granularity and smoothness [8].

Many algorithms compute using Voronoï and Delaunay, [17], [18], [15], [8] and many others extract skeletons from Voronoï and Delaunay [19], [11], [23], [13], [3] We will present four of them that we have tested in 2D. However, we have to note that: $S_k\left(\partial S\right) \subset \lim\limits_{E \to \partial S} \mathrm{Vor}(E)$, where E is the set of points that samples the form S, Vor is the Voronoï diagram and S$_k$ is the skeleton.

First Method, [28]. The skeleton Sk$_1$(X) is composed with vertices of Voronoï inside the object. The homotopy of the object is no longer preserved.

$$\mathrm{Sk}_1\left(\partial S\right) = \lim_{E \to \partial S}\left(\bigcup_{s \in S, s \text{ vertex of } \mathrm{Vor}\left(E\right)} \{s\}\right).$$

Second Method, [23]. the skeleton Sk$_2$(X) is composed with points of Voronoï inside the form

$$\mathrm{Sk}_2\left(\partial S\right) = \lim_{E \to \partial S}\left(\bigcup_{s \in S, s \in \mathrm{Vor}(E)} \{s\}\right)$$

For instance, Figure 7 shows a rectangle, which is discretized with 8 points and its Voronoï diagram and Delaunay triangulation.

Figure 8 shows us skeletons obtained with methods 1 and 2.

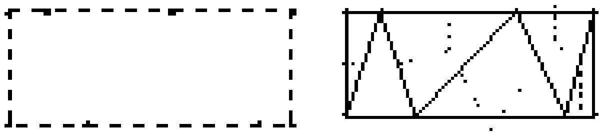

Figure 7: Rectangle with 8 points, Voronoï diagram and Delaunay triangulation

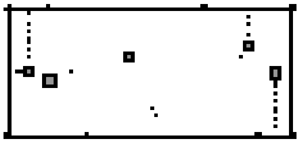

Figure 8: Skeletons obtained with methods 1 and 2. In dotted lines is the skeleton of [23] and in black squares skeleton of [28]

Third Method, [10]. Sk$_3$(X) is built from elements of Voronoï inside the form. As elements, we mean edges in 2D or plans in 3D. The skeleton is always inside the form but it may be empty (see Figure 9).

$$\mathrm{Sk}_3(\partial S) = \bigcup_{F \subset S,\, F \text{ element of Vor}\,(E)} \mathrm{F}\,.$$

Fourth Method, [7]. The elements inside the triangulation of Delaunay are the elements of this triangulation strictly inside the form. If we have ∂S $\subset$Del(S) then Sk$_4$(∂S) = $\bigcup_{v \text{ element inside Del}(E)}$ Dual(v).

The skeleton is never empty but is not always defined and is not always inside the form (see Figure 9). Finally, we can say $Sk_1(S) \subset Sk_3(S) \subset Sk_2(S)$.

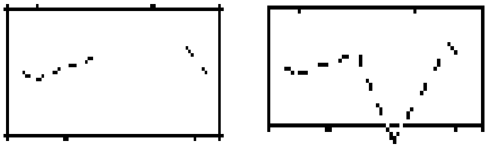

Figure 9: Skeletons obtained using third and fourth methods

We note that branches are built that have no link with the real skeleton. So we have to simplify this skeleton [4], [21], [22].

3. Our Method

Aldous [2] established a link between a triangulation of a 2D convex set of points and a binary tree.

To explain, we take it that we have a polygon and its triangulation. We choose a face to enter the triangulation (always the same one. However, this choice has no importance if we do not want a bijection between triangulation and the tree). This face is the root of the tree. Each edge of the triangle, in which we are, is cutting the overall space in two sub-spaces. Each sub-space is corresponding with a branch of our tree and the edge is a son of the knot of the root (see Figure 10).

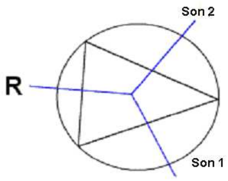

Figure 10: Branch with two sons

If we continue this process with each son, we obtain a binary tree, whose root is the first edge of the first triangle and whose leaves represent the edges of the convex hull. Figure 11 shows an example of a triangulation and its tree.

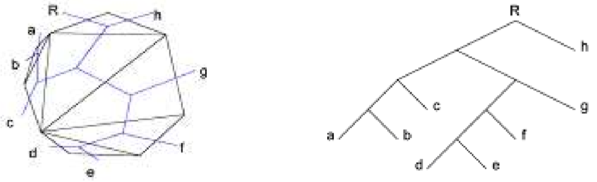

Figure 11: An example of a triangulation and its tree

We have already seen that there is a bijection between a Delaunay triangulation and the set of points. Here, we have a bijection between a Delaunay triangulation and a tree. Therefore, we obtained a bijection between a set of points and a tree. The nodes of the tree represent the skeleton. To obtain this skeleton we consider the tree without the leaves because leaves represent branches going out of the triangulation.

Moreover, the nodes are computed in the iso centre of the triangles. Therefore, if we use the skeleton in a reconstruction process, the elements around the skeleton will be equidistant from every point of the set of points.

Our method combines the advantages of the third previous skeleton extraction method (skeleton always defined and inside the form) and those of the fourth one (homotopy with the initial form). Figure 12 represents the skeleton of the previous example. We can notice that it comes nearer to the real skeleton (the branches, which go outside of the form, are not drawn).

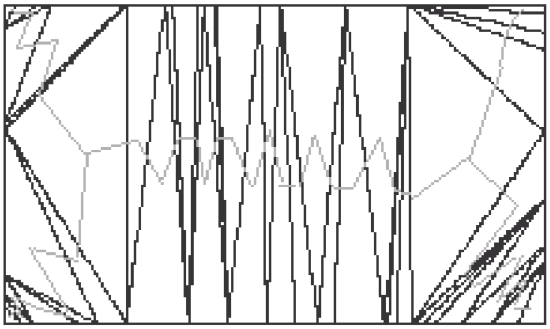

Figure 12: Skeleton obtained in a rectangle (in black: the triangulation of Delaunay, in grey: the skeleton)

There are many algorithms in extracting a skeleton in 3D from a Delaunay triangulation, e.g. [25]. Our method, described above, can be extended to a space of greater size. The algorithm, for a 3D space, produces a bijection between the cloud of 3D points and a tree whose nodes will have up to three sons.

In 3D, the initial stage of the algorithm is always a tetrahedrisation of Delaunay, to profit from the properties. One of the significant properties concerns the distance between the vertices of a tetrahedron. To construct this tetrahedrization, we choose to project the points in a space of greater size (see Figure 13).

Always from a point of view of bijection between the cloud of points and the tree, we must determine the face of the algorithm in a single way, for a tetrahedrization, which will correspond to the root of the tree. We choose the face of the tetrahedrons of the convex hull whose first co-ordinate of the iso center is minimal (in the event of equality, the choice is then made over the

Figure 13: A cloud of 3D points and its Delaunay tetrahedrisation

second, then if necessary, over the third co-ordinate).

A starting face of the algorithm is then known without ambiguity. Once the face of initialization is determined, the construction of the tree can start. The procedure proceeds in a similar way to the 2D version. However, it is necessary to define an orientation of the tetrahedrons, as we will see from now on. One can then carry out the procedure of dividing space, in an iterative way to build the whole tree. The procedure stops when there are no more tetrahedrons that do not have a neighbor not already traversed. These tetrahedrons correspond then to leaves of the tree. One finally obtains a tree with the highest number of ternary nodes, starting from the initial cloud of points (see Figure 14). The three faces are ordered in the following way (Figure 15).

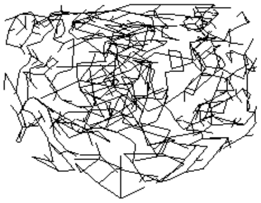

Figure 14: 3D tree obtained on the previous cloud of points

The first one is the one whose third co-ordinate (in Z) of the iso-centre is minimal (in the event of equality between two faces, the remaining face is selected). Then, the choice between the two faces is determined by the trigonometric direction based on the first point of view from the face, from which we enter the tetrahedron.

The Delaunay triangulation can induce some problems. Indeed, the unicity

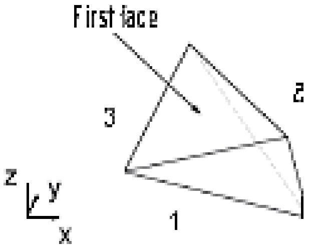

Figure 15: Ordering the faces of a tetrahedron

of the triangulation is guaranteed only if no more than three co-cyclic points exist (in 2D), or no more than four co-spherical points (in 3D) (see Figure 16). To the contrary, there are several triangulations of Delaunay of the set of points. We affect a very small displacement to the concerned points of the cloud, in order to avoid this configuration. It results, however, in the creation of "flat" tetrahedrons, which will lead to the appearance of unwanted nodes and branches (in the tree). As one can see in Figure 17, which represents the result of our algorithm on a box, sampled by its 8 vertices, "flat" tetrahedrons on the sides of the objects are generated which lead to the creation of unwanted branches. It is thus necessary, in the algorithm, to introduce a test on the tetrahedrons to decide whether to process them or not. For that, we call upon the quality standard Q$_k$ of a tetrahedron [17], which is characteristic of its form.

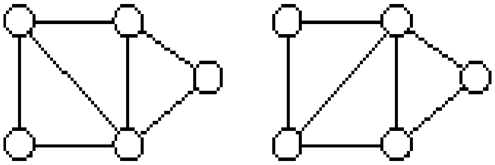

Figure 16: Two triangulations of Delaunay for the same set of points

Definition 5. (Quality Standard) The quality standard Q$_k$ is defined as follows:

$Q_k = \beta \frac{h_s^3}{V_k}$, where β is a constant $\left(\beta = \frac{\sqrt{3}}{216}\right)$, $h_s = \sqrt{\sum\limits_{i=1}^{6} L_i^2}$, with Li the

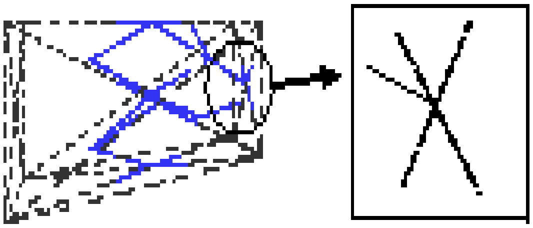

Figure 17: Flat tetrahedrons

length of the side of the tetrahedron and V$_k$ the volume of the tetrahedron. V$_k$ can be obtained by the resolution of

$$V_k = \frac{1}{6}\begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - x_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - x_1 & z_3 - x_1 & z_4 - x_1 \end{vmatrix}.$$

The value of Q$_k$ varies from 1 (for an equilateral tetrahedron), to ∞ (for a flat tetrahedron). One will thus prefer to use $^1/_{Q_k}$varying from 0 to 1 for a flat tetrahedron to equilateral. This criterion allows us to avoid the problem of the nodes and branches created in the tree, following the existence of several tetrahedrizations of Delaunay (see results Figure 18).

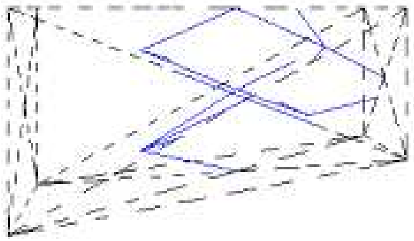

Figure 18: Result after removing flat tetrahedrons

4. Conclusions and Prospects

In this paper, we have presented a method giving a representation of a convex set of points in the form of a tree. We proposed its use for the extraction of skeletons by using the triangulation of Delaunay, which enables us to associate information on the relative position of the points to this tree. Indeed, the organization of the nodes of the tree is related to the triangles and thus takes into account the properties of vicinity of the points. A representation of the skeleton of the initial cloud can then be easily built in a 2D field.

Moreover, this representation directly gives us the elements of the skeleton in the form of segments (branches of the tree), with in addition, their spatial organization. That allows us, for example, to approximate many segments by a parametric curve, in order to create a skeleton whose continuity will be higher (Figure 19), see [27], [26].

Figure 19: A Bezier skeleton and its representation an implicit surface

Our algorithm thus makes it possible to determine the skeleton of a 2D object sampled by points of its border. It works with both a convex object and a non-convex object. Indeed in the second case, one only needs to use an algorithm providing a non-convex triangulation and to apply our method to obtain the skeleton of the object as a tree. The task is not so easy in 3D. One must also carry out a post-processing on this tree in order to be able to use it as a skeleton. It would be very interesting to be able, starting from the tree, to generate the various elements that we require for the skeleton (segments, planes and so on). For that, a phase of connection of the nodes of the tree must be developed. This is not commonplace and requires the installation of criteria on the connection of the nodes of the tree.

Another solution is the binarization of our 3D-tree. We are able to build a skeleton from a 2D-tree. Then, we hope to calculate a 3D tree from many 2D trees and Boolean operations on these trees.

References

[1] Z. Aktouf, *Opérateurs Topologiques pour Images Binaires 3D*, Ph.D. Thesis, Univ. Paris Sud. (1997).

[2] D. Aldous, Triangulating the circle at random, **101**, No. 3 (1994), 223-233.

[3] D. Attali, *Squelettes et Graphes de Voronoï 2D et 3D*, Thèse de Doctorat, Université Joseph Fourier, Grenoble I, France (1995).

[4] D. Attali, A. Montanvert, Computing and simplifying 2D and 3D continuous skeletons, *Computer Vision and Image Understanding*, **67**, No. 3 (1997), 161-273.

[5] E. Bittar, N. Tsingos, et al, Automatic reconstruction of unstructured 3D data: Combining a medial implicit surfaces, *Eurographics'95*, **14**, No. 3 (1995), 457-468.

[6] H. Blum, A transformation for extracting new descriptors of shape, *Models for the Perception and Visual Form*, MIT Press (1967), 362-380.

[7] J.D. Boissonnat, B. Geiger, Three dimensional reconstruction of complex shapes based on the Delaunay triangulation, *Raport de Recherche*, Inria: Institut National de Recherche en Informatique et en automatique, RR-1697 (1992).

[8] J.D. Boissonnat, M. Yvinec, *Géométrie Algorithmique*, Ediscience International (1995).

[9] G. Borgefors, I. Nyström, et al, Computing skeletons in three dimensions, *Pattern Recognition*, **32** (1999), 1225-1236.

[10] J.W. Brandt, Convergence and continuity criteria for discrete aroximations of the continuous planar skeletons, *CVGIP: Computer Vision, Graphics and Image Processing*, Image Understanding, **59**, No. 1 (1994), 116-124.

[11] J.W. Brandt, V.R. Algazi, Continuous skeleton computation by Voronoï diagram, *CVGIP : Computer Vision, Graphics and Image Processing*, Image Understanding, **55**, No. 3 (1992), 329-338.

[12] F. Chin, J. Snoeyink, et al, Finding the medial axis of a simple polygon in linear time, *Geometry: Discrete and Computational Geometry*, **21**, No. 3 (1999), 405:420

[13] T. Culver, J. Keyser, Accurate computation of the medial axis of a polyhedron, In: *Proceedings of the Fifth Symposium on Solid Modeling and Applications, SSMA-99* (1999), 179-190.

[14] G. Evans, A. Middleditch, et al, Stable computation of the 2D medial axis transform, *International Journal of Computational Geometry and Alications*, **8**, No. 5, 6 (1998), 577-598.

[15] K.C. Fan, D.F. Chen, et al, Skeletonization of binary images with nonuniform width via block decomposition and contour vector matching, *Pattern Recognition*, **31**, No. 7 (1998), 823-838.

[16] T.P. Fang, L.A. Piegl, Delaunay triangulation in three dimensions, *IEEE Computer Graphics and Alications*, **15**, No. 5 (1995), 62-69.

[17] P.L. George, H. Borouchaki, *Triangulation de Delaunay et Maillage*, Hermes (1997).

[18] P. Kauffmann, J.C. Spehner, Sur l'algorithme de Fortune, *Revue Internationale de CFAO et d'Informatique Graphique*, **10**, No. 4 (1995), 321-336.

[19] N. Mayya, V.T. Rajan, Voronoï diagrams of polygons: A framework for shape representation, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition* (1996), 638-643.

[20] L.M. Mestelskii, Fat curves and representation of planar figures, *Siggraph'200* (2000), 9-21.

[21] R.L. Ogniewicz, *Discrete Voronoï Skeletons*, Ph.D. Thesis, Institut Fédéral de Technologie, Zurich, Suisse (1992).

[22] R.L. Ogniewicz, Skeleton-space: A multiscale shape description combining region and boundary information, In: *Proceedings of Computer Vision and Pattern Recognition* (1994), 746-751.

[23] R.L. Ogniewicz, M. Ilg, Voronoï skeletons: Theory and alications, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition* (1992), 63-69.

[24] R.L. Ogniewicz, O. Kübler, Hierarchic Voronoï skeletons, *Pattern Recognition*, **28**, No. 3 (1995), 343-359.

[25] J.M. Reddy, G.M. Turkiyyah, Computation of 3D skeletons using a generalized Delaunay triangulation technique, *Computer Aided Design*, **27**, No. 9 (1995), 677-694.

[26] D. Rigaudière, D. Faudot, et al, Shape modeling with skeleton based implicit primitives, *Graphicon'2000* (2000), 174-178

[27] D. Rigaudière, G. Gesquière, et al, New implicit primitives used in reconstruction by skeletons, *WSCG'99*, **2** (1999), 433-439.

[28] M. Schmitt, Some examples of algorithms analysis in computational geometry by means of mathematical morphology techniques, *LNCS in Computer Science, Geometry and Robotics*, **391** (1989), 225-246.

[29] Y.M. Sharaiha, A graph-theoretic aroach to chamfer distance transformations, *Discrete Geometry for Computer Imaging* (1993), 68-181.

[30] D.J. Sheehy, C.G. Armstrong, et al, Computing the medial surface of a solid from a domain Delaunay triangulation, In: *SMA '95: Proceedings of the 3-rd Symposium on Solid Modeling and Alications* (1995), 201-212.

[31] E.C. Sherbrooke, N.M. Patrikalakis, et al, An algorithm for the medial axis transform of 3D polyhedral solids, *IEE Transactions on Visualization and Computer Graphics*, **2**, No. 1 (1996), 44-61.

[32] Y. Usson, A. Montanvert, Reconstruction tridimensionnelle à partir des distances discrètes, *Colloque Géométrie Discrète en Imagerie: Fondements et Alications* (1993), 20-21.