Constructing Iso-Surfaces Satisfying the Delaunay Constraint. Application to the Skeleton Computation.

Dominique Attali
Laboratoire des Images et des Signaux,
ENSIEG, Domaine Universitaire, BP 46
38402 Saint-Martin d'Hères Cedex, FRANCE
E-mail: Dominique.Attali@imag.fr

Jacques-Olivier Lachaud
University of Toronto, CS Dept
10 King's College Road, S. Fleming Bldg
Toronto, Ontario, M5S 3G4, CANADA
E-mail: lachaud@cs.toronto.edu

Abstract

We design a new model for an image iso-surface which lies in the Delaunay graph of its vertices. Within each 8-cube of the image, a set of loops is computed according to the connectedness chosen for inner and outer voxels. Next, a triangulation is computed which respects the local geometry of these loops. Efficiency is obtained through the use of a look-up table which summarizes the algebraic tests that are required of each case. The inclusion of the iso-surface in the Delaunay triangulation has significant consequences. We derive a volume representation of the object, along with its skeleton. An example depicts the complete construction of our iso-surface, volume representation and skeleton computation.

1. Introduction

Currently, we are witnessing a proliferation of high quality volumetric images, coming from a large number of acquisition devices (e.g., MRI, CT scanner). Analysis of this data often requires, as a first step, the computation and display of surfaces which approximate the objects boundaries. *Digital surfaces* and *iso-surfaces* are perhaps the simplest surfaces that can be derived from volumetric images. This paper focuses on iso-surfaces. Iso-surfaces are composed of triangles whose vertices lie in the 3-dimensional Euclidean space $\mathcal{R}^3$. These surfaces present some interesting topological and geometrical properties; in fact they are 2-manifolds in $\mathcal{R}^3$. Unlike digital surfaces, they present a smooth aspect.

Unfortunately, iso-surfaces provide a *surface representation* that might be inadequate in a context other than visualization. A *volume representation* is sometimes necessary. For instance, physical modelling by finite elements requires a partition of the object into elementary blocks (tetrahedra or cubes). On the other hand, the *skeleton* is more appropriate to extract object parameters such as thickness or length.

Unlike existing methods, we propose in this paper to build iso-surfaces that are included in the Delaunay triangulation of their vertices. This property has important consequences. First, the Delaunay triangulation has remarkable properties, e.g. it is optimal for several geometric criterions. Secondly, these iso-surfaces can be used to easily derive volume representations, and skeletons, of objects. Our construction, which is performed locally, remains valid for non isotropic volume and is performed in a time proportional to the size of the image.

2. Preliminary definitions

In this section, we state some elementary definitions of computational geometry which are used throughout the paper [16]. In the following, d designates the dimension of space; when not specified, $d = 3$.

Delaunay graph. Let $E \in \mathcal{R}^d$ be a finite set of points in general position. The *Delaunay triangulation* of E is the set of triangles in 2D (or tetrahedra in 3D) whose circumscribed balls contain no points of E in their interior. In this paper, we will use the term *Delaunay graph*, denoted by $\mathrm{Del}(E)$, for the sub-complex of the Delaunay triangulation formed by triangles, edges and vertices in R^3.

Convex hull. The *convex hull*, denoted by $\mathrm{Hull}(E)$, of an arbitrary set of points $E \in R^d$ is the smallest convex set containing E. If we assume the set E to be finite, the convex hull is a polytope whose boundary is included in the Delaunay graph of E.

Voronoi graph. The *Voronoi diagram* of a set of points E is a partition of the space into regions called *Voronoi regions*. The Voronoi region of $p \in E$ is the set of points of $\mathcal{R}^d$ that are closer to p than to any other point of E. The Voronoi diagram is the dual of the Delaunay triangulation. We use the term *Voronoi graph*, denoted by $\mathrm{Vor}(E)$, for the sub-complex of the Voronoi diagram formed of polygons, edges and vertices in R^3.

3. Iso-surface

3.1. Definition and marching-cubes algorithm

Let $\mathcal{Z}^d$ be the discrete d-dimensional space. A gray-level *volumetric image* I is a mapping from a set $U \subset \mathcal{Z}^3$ onto the set of real numbers $\mathcal{R}$. Using the canonical embedding, the set U may be mapped from $\mathcal{Z}^3$ into $\mathcal{R}^3$ as a grid of evenly spaced points. Each element of U is called a *voxel*. Thus, the image I may then be viewed as a sampling of a scalar continuous field h on the vertices of this discrete grid: the mapping h coincides with the mapping I on these vertices. The set of points where h takes the value C defines a *surface* S_C (i.e. $S_C = \{M \in \mathcal{R}^3,\ h(M) = C\}$). In the following, we use the term "iso-surface" for any triangulated surface that approaches S_C. The choice of C allows the user to select a specific object in the data volume. In the following, we assume the image I is finite and its border is composed of only outer voxels or of only inner voxels.

The most common method of computing iso-surfaces in images is certainly *marching-cubes* [12]. Its principle is to analyze the image locally using blocks of eight mutually adjacent voxels. These blocks are called *8-cubes*. Within each 8-cube, a set of triangles is found which separate *inner voxels* ($h(v) \geq C$) from *outer voxels* ($h(v) < C$). A *C-vertex* of the iso-surface is defined as a point lying on a grid edge between adjacent, inner and outer, voxels. A linear model is used to estimate where the iso-surface intersects this edge and determine an appropriate position for the C-vertex. Different sets of triangles may be constructed on the set of C-vertices of an 8-cube. The chosen triangulation influences the geometry and the topology of the iso-surface, and hence the properties of the approximation.

Marching-cubes [12] exhibits an arbitrary choice, which depends only on the inner or outer classification of the 8-cube voxels. This leads to 256 possible triangulations within an 8-cube. Building the surface locally may be optimized by pre-computing a table associating each configuration to a triangulation. Some authors refine this choice by using different kinds of interpolation or by exploiting gradient information (see [7] for a survey). However, none of these methods consider the geometry of the C-vertices, which explains why the computed iso-surface does not, in general, coincide with the Delaunay triangulation.

3.2. Proposed method for building iso-surfaces

To build a triangulated iso-surface that is included in the Delaunay graph of its vertices, we use the iso-surface definition suggested in [9] This definition is based on one hand on the digital connectedness of voxels and on the other hand on the location of the C-vertices on the grid. In order that the paper be self-contained, we briefly recall some definitions of digital topology.

Two (different) voxels are said to be *6-adjacent* if their coordinates differ of ± 1 on exactly one coordinate. They are *18-adjacent* (resp. *26-adjacent*) if their coordinates differ of ± 1 on one or two coordinates (resp. one, two or three coordinates). Two voxels a, $b \in F$ are *ρ-connected* (for $\rho \in \{6, 18, 26\}$) if there exists a sequences of ρ-adjacent voxels of F, starting with a and ending with b. The transitive closure of this relation is *ρ-connectedness*, which defines *ρ-components* in F. In the following, a κ-connectedness is associated with the inner voxels, and a λ-connectedness is associated with the outer voxels of I. The pair $\{\kappa, \lambda\}$ is called a *connectedness pair*. Depending on the value chosen for κ and λ (6, 18 or 26), different properties can be shown. Following [9], we restrict our study to the pairs $(6, 18)$, $(6, 26)$, $(18, 6)$ and $(26, 6)$, which are said to be *valid*.

On every face of an 8-cube, *C-edges* are placed between C-vertices so that they do not intersect the connectedness links between inner voxels, or those between outer voxels. The C-edges of in an 8-cube form a set of loops (called *C-loops*) (see Figure 1). The triangulation of C-loops is performed locally and depends on the geometry of their C-vertices. The edges and triangles are built on the boundary of the *convex hull* of both the C-vertices and the inner voxels (resp. outer voxels), when the 6-connectedness has been chosen for outer voxels (resp. inner voxels). This three-dimensional convex set is called the *local C-convex hull*. Figure 2 illustrates this construction. Figure 1 displays some examples of subdivisions for various configurations. It can be shown that, for any of the four valid pairs, the C-loops always belong to the boundary of their local C-convex hull and that edges within a loop are always determined by C-vertices of this loop. The set of C-vertices, edges and triangles thus defined from I forms a 2-manifold in $\mathcal{R}^3$ with no boundary.

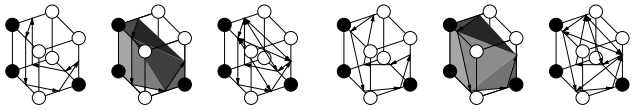


Figure 2. Two different triangulations of a C-loop on the convex hull of the C-vertices and the inner voxels ($(\kappa, \lambda) \in \{(18, 6), (26, 6)\}$).

3.3. Inclusion in the Delaunay graph

We will now present the main result of this paper, which is to show that a such defined iso-surface has the interesting property to be included in the Delaunay triangulation of its vertices. To establish this result, we first show that the computation of the Delaunay graph can be restricted to a computation between vertices of the same 8-cube. In the following, S_C denote an iso-surface, E_C the set of its vertices.

Theorem 1 *Let $\{e_1, e_2, \ldots, e_k\}$ be the k C-vertices of a given 8-cube. Then:*

$$\partial\text{Hull}(\{e_i\}_{i=1,\ldots,k}) \subset \text{Del}(\{e_i\}_{i=1,\ldots,k}) \subset \text{Del}(E)$$

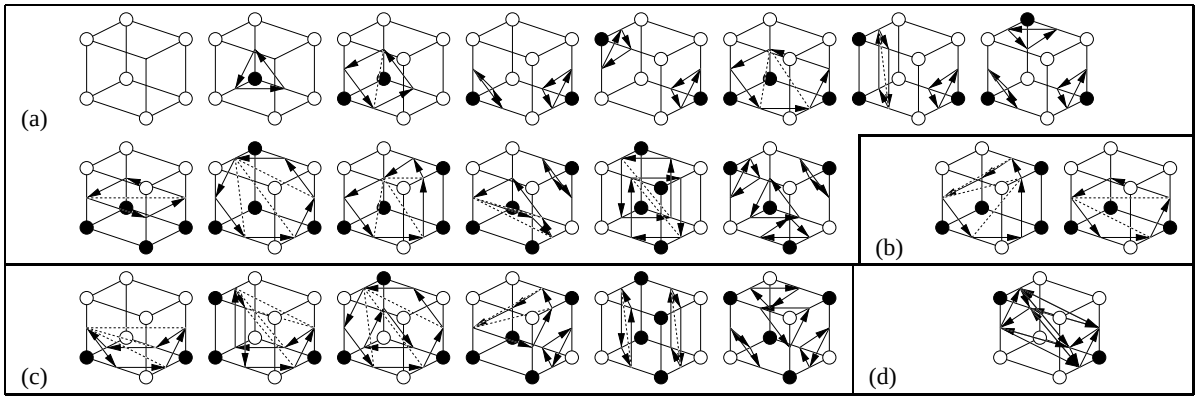


Figure 1. C-loops created for classical configurations and triangulation of these loops. The connectedness pair has an influence on both the construction of the C-loops and their triangulation: (a) C-loops and C-edges created for $(\kappa,\lambda) \in \{(6,18),(6,26)\}$; (b) when $(\kappa,\lambda) \in \{(18,6),(26,6)\}$, these configurations have the same C-loops than in (a), but different C-edges; (c) when $(\kappa,\lambda) \in \{(18,6),(26,6)\}$, these configurations have different C-loops than in (a); (d) special case when $(\kappa,\lambda) \equiv (26,6)$ (a symmetric case exists when $(\kappa,\lambda) \equiv (6,26)$).

Proof. The first inclusion is immediate. To prove the second inclusion, we compute the intersection of the smallest sphere B enclosing e_i and e_j with the discrete grid. This computation can be achieved by simple geometrical considerations: we search points v such that the straight-lines (ve_i) and (ve_j) are orthogonal. Let P be the smallest straight parallelepiped enclosing e_i and e_j. The vertices of P are located on the sphere B. As P remains inside the considered 8-cube (see Fig. 3), the sphere B does not intersect grid edges of other 8-cubes. Consequently, the sphere B contains no points from $E \setminus \{e_1, e_2, ..., e_k\}$ in its iterior, which implies the second inclusion. □

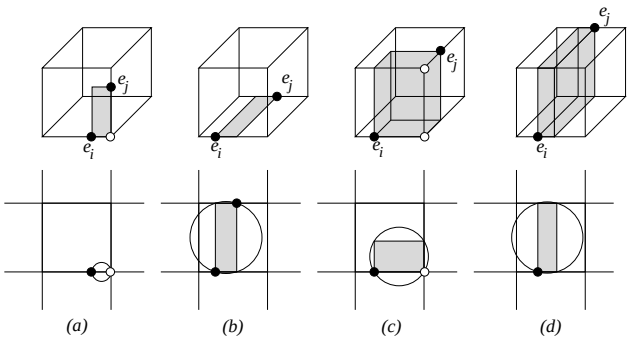


Figure 3. To examine the intersection of the sphere B with diameter $[e_ie_j]$, four different cases must be considered, which correspond to different positions of the C-vertices e_i and e_j on the 8-cube. The smallest parallelepiped P enclosing e_i and e_j has been drawn in gray. It always lies inside the 8-cube. The second line shows the intersection of the sphere B, the parallelepiped P and the 8-cube with the plane $\{z = 0\}$.

Theorem 2 *Let S_C be an iso-surface and E_C the set of its vertices. Then: $S_C \subset \mathrm{Del}(E_C)$.*

Proof. Each edge $[ee']$ of the iso-surface belongs to an 8-cube. Let $\{e_1, e_2, ..., e_k\}$ denotes the vertices of this 8-cube. By construction, $[ee'] \subset \partial\mathrm{Hull}(\{e_1, e_2, ..., e_k\})$. Theorem 1 concludes. □

3.4 Tabulation of configurations

Because the set of triangles built inside each 8-cube depends on the geometry of the C-vertices, a static table of 256 configurations cannot be used to store all possible triangulations (as opposed to the marching-cubes for instance). We can nevertheless simplify the computation of the convex hull within each 8-cube with a careful case study of the different configurations.

First, we notice that only C-loops with at least four C-vertices must be triangulated. Therefore, 56 configurations do not depend on the geometry of the C-vertices. For each other configuration, which contains at least one C-loop with more than three C-vertices, several sets of triangles are stored. At run-time, simple algebraic tests will determine which set of triangles corresponds to a particular geometry. To exhibit these algebraic tests for each configuration, we make use of the following lemma:

Lemma 1 *Let $L = \{e_0, e_1, ..., e_{k-1}\}$ be a C-loop in a given 8-cube. Let us assume that the connectedness is one of $(26,6)$ or $(18,6)$. Then an edge e_ie_j belongs to the local C-convex hull iff*

$$\forall k, l / i < k < j, j < l < i \Rightarrow e_i\vec{e}_l \cdot (e_i\vec{e}_k \wedge e_i\vec{e}_j) < 0,$$

indices taken modulo k.

This lemma comes from the fact that all edges of a C-loop belong to the local C-convex hull, and that the

shape of the local C-convex hull within a C-loop depends only on the vertices of this loop. A symmetric lemma holds when the connectedness is one of $(6, 26)$ or $(6, 18)$ (the inequality is just inverted).

Table 1 displays the algebraic tests required for each configuration. All other configurations can be derived from the pictured ones either by rotation or by inversion of inner and outer voxels (and connectedness). The table also displays configurations whose algebraic tests are identical to the one presented. In most cases, only several of the different calculations must be performed, since most edges cannot exist simultaneously.

The computation of iso-surfaces following the Delaunay constraint cannot be as fast as an entirely tabulated algorithm. Most algebraic tests can nevertheless be computed quickly, except for the last case which may require a dozen inequality computations.

4. Application to the skeleton computation

The skeleton is a convenient representation of objects widely used in the field of image analysis. The notion of skeleton was first introduced by Blum [2] under the name of *medial axis transform*. The skeleton $\mathrm{Sk}(X)$ of an object X is defined as the set of centers of its *maximal disks* [6]. A disk is said to be maximal in an object if there are no other disks included in the object that contain it. This definition holds in any dimension and can be applied in discrete as well as in continuous space. In continous space, the skeleton has remarkable properties [5]. It is a graph made of points and curves in 2D and points, curves and surfaces in 3D. It has the same homotopy type as the object. Assuming the skeleton points to be labelled with the radius of their associated maximal disk, the skeleton provides a reversible encoding of the object.

Skeleton computation has been intensively studied both in discrete and continous space. In discrete space, many algorithms have been proposed that can be divided in two main families:

- algorithms based on morphological thinnings that preserve homotopy [10, 13, 14, 11],
- algorithms using the distance map that preserve reversibility [4, 18].

There also exists mixed approaches using anchor points [17]. A drawback of these techniques is the difficulty of computing, in a reversible fashion, a discrete, single pixel wide skeleton which preserves homotopy. For particular binary objects, such a set of pixels simply does not exist.

In this paper, we focus on methods that directly compute the skeleton in continuous space. We use the method proposed by Boissonnat [3]. The idea is to partition the object into Delaunay elements. The skeleton is then defined as the dual of this partition. The input of this method is a closed surface S whose set of vertices is E_S. If the surface S is included in the Delaunay graph $\mathrm{Del}(E_S)$, it is possible to distinguish exactly two different types of Delaunay tetrahedra : *inner tetrahedra* lying inside S and *outer tetrahedra* lying outside S. The skeleton is defined as the dual of the inner Delaunay elements. As such, the skeleton is formed of polygons, edges and vertices defining surfaces in 3D. This method works only if the boundary S is included in the Delaunay graph of its vertices $\mathrm{Del}(E_S)$. The iso-surfaces constructed in the previous part verify this condition and can therefore be used to compute the skeleton.

A simplification step follows the computation of the skeleton. Its goal is to remove peripheral branches associated to non significant parts of the object. The simplification method used in this paper is a direct 3D extension of the method proposed in [1].

Depending on the choice of inner and outer connectedness, different types of iso-surfaces can be generated for the same volumetric data and the same threshold C. We use the connectedness pair $(6, 18)$ in order to minimize the Euler constant and the number of loops in the object and improve the efficiency of the simplification step.

Our results are presented in figure 4. The first volume was generated synthetically and corresponds to the discretization of an ellipsoid. As expected, the skeleton computation provides an ellipse. The second and third volumes were real images corresponding respectively to a bone sample and a human heart. The volume sizes were $30 \times 30 \times 30$, $30 \times 30 \times 30$ and $85 \times 85 \times 98$. The number of vertices on each iso-surfaces was 1708, 3272 and 36640. The running times for computing both the iso-surface and the skeleton was 3 s, 6 s and 115 s on an SGI O2 R4000. We emphasize that the surface portions making the skeleton have a smooth aspect. Such a smooth aspect is particularly tricky to achieve with purely discrete approaches.

5. Conclusion

In this paper, we have constructed a new type of iso-surfaces: they have the property to be included in the Delaunay graph of their vertices. The proof uses simple geometrical considerations and holds for non isotropic voxels. A table of algebraic tests has been proposed in order to speed up the computation. Iso-surfaces were used to compute the skeleton. Future work includes the use of the skeleton in order to extract parameters and segment objects.

Acknowledgments

We are grateful to Stuart Andrews for his careful reviewing of the manuscript. Our deepest thanks to Francoise Peyrin for the bone sample volume and to Yves Usson for the heart volume. The bone sample was acquired at the European Synchrotron Radiation Facility (ESRF) by computed microtomography [15]. The heart data was obtained with a polarized light optical bench [8]. This work was partly performed within the

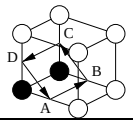
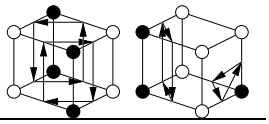
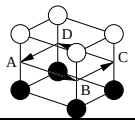
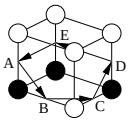
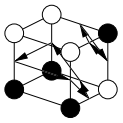
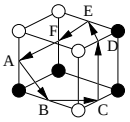
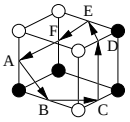
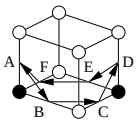
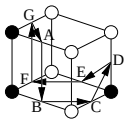
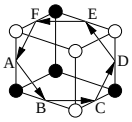
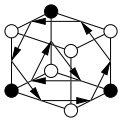
configuration	connectedness	algebraic test ⇒ added edges *(excluded edges)*	other configurations
	$(26,6)\ (18,6)$ (inverted for $(6,26)\ (6,18)$)	$bd < ac \Rightarrow$ AC *(BD)* $bd > ac \Rightarrow$ BD *(AC)*	
	$(26,6)\ (18,6)$ (inverted for $(6,26)\ (6,18)$)	$b+d < a+c \Rightarrow$ AC *(BD)* $b+d > a+c \Rightarrow$ BD *(AC)*	
	$(26,6)\ (18,6)$ (inverted for $(6,26)\ (6,18)$)	$b < \frac{a(1-c)}{a(1-c)+cd}$, $b < \frac{a}{a(1-c)+ce} \Rightarrow$ AC *(BD, BE)* $d > (a-eb)/b$, $d > (e-a)(1-c) \Rightarrow$ AD *(BE, CE)* $c > \frac{a(1-b)}{a(1-b)+db}$, $c > \frac{(e-d)(1-b)}{db+e(1-b)} \Rightarrow$ BD *(CE, AC)* $a < bce/(1-b+bc)$, $a < eb-db \Rightarrow$ BE *(AC, AD)* $d < (e-a)(1-c)$, $d < \frac{e(1-b)(1-c)}{1-b+bc} \Rightarrow$ CE *(AD, BD)*	
	$(26,6)\ (18,6)$	true $\Rightarrow$ AD *(BE, CE, BF, CF)* $b < \frac{a(c-d)}{a(c-d)-c} \Rightarrow$ AC *(BD)* $b > \frac{a(c-d)}{a(c-d)-c} \Rightarrow$ BD *(AC)* $a < \frac{1-f+def-e}{e(d-1)} \Rightarrow$ DF *(AE)* $a > \frac{1-f+def-e}{e(d-1)} \Rightarrow$ AE *(DF)*	
	$(6,26)\ (6,18)$	true $\Rightarrow$ CF *(AD, AE, BD, BE)* $b > a/(a(1-c)+cf) \Rightarrow$ AC *(BF)* $b < a/(a(1-c)+cf) \Rightarrow$ BF *(AC)* $c < \frac{def-df+d-e}{(1-e)(1-f)} \Rightarrow$ DF *(CE)* $c > \frac{def-df+d-e}{(1-e)(1-f)} \Rightarrow$ CE *(DF)*	
	$(26,6)\ (18,6)$	true $\Rightarrow$ AD *(BE, BF, CE, CF)* $d < \frac{a(1-b)(1-c)}{bc} \Rightarrow$ AC *(BD)* $d > \frac{a(1-b)(1-c)}{bc} \Rightarrow$ BD *(AC)* $a < \frac{def}{(1-e)(1-f)} \Rightarrow$ DF *(AE)* $a > \frac{def}{(1-e)(1-f)} \Rightarrow$ AE *(DF)*	
	$(26,6)\ (18,6)$	true $\Rightarrow$ AD *(BE, BF, BG, CE, CF, CG)* true $\Rightarrow$ DG *(AE, AF, BE, BF, CE, CF)* $b < \frac{1-c+dac}{1-c+cd} \Rightarrow$ AC *(BD)* $b > \frac{1-c+dac}{1-c+cd} \Rightarrow$ BD *(AC)* $f > \frac{edg-e+1}{1-e+ed} \Rightarrow$ DF *(EG)* $f < \frac{edg-e+1}{1-e+ed} \Rightarrow$ EG *(DF)*	
	$(26,6)\ (18,6)$ (inverted for $(6,26)\ (6,18)$)	$b < \frac{a(1-c)}{a(1-c)+cd}$, $b < \frac{a(1-ce)}{a(1-c)+c}$, $b < \frac{af}{a(f-c)+c} \Rightarrow$ AC *(BD, BE, BF)* $b > \frac{a(1-c)}{a(1-c)+cd}$, $b > \frac{d(ce-1)+1-c}{(1-c)(1-d)}$, $c > \frac{(1-b)(1-fd)}{1-b+db} \Rightarrow$ BD *(AC, CE, CF)* ... $a > b(1-d)/(1-e)$, $a > \frac{b(1-df)}{b(1-f)+f}$, $a > \frac{d(ce-1)+1-c}{(1-c)(1-e)}$, $d > (1-c)(1-a)/f \Rightarrow$ AD *(BE, BF, CE, CF)* ...	

Table 1. Case study for configurations whose triangulation is ambiguous. The center of axes is at the lower left voxel of the 8-cube. The X axis is directed to its lower right neighbor; the Y axis is directed to its upper right neighbor; the Z axis is vertical. Each vertex has one degree of freedom on its segment; this free coordinate is denoted by the lower case of its name (its range is the open interval $]0,1[$). The non-free coordinates are set to either 0 or 1. For each configuration, the connectedness pair is given. The algebraic tests necessary to verify the existence of C-edges are displayed. For each of these edges, the C-edges that cannot exist at the same time are listed. Other configurations with similar tests are also displayed. We have not listed all the different tests necessary to solve the lowest configuration; we have rather provided several tests that are representative. The phrase "inverted for $(6,18)(6,26)$" means that every inequality must be inverted when the connectedness couple is one of $(6,18)$ or $(6,26)$.

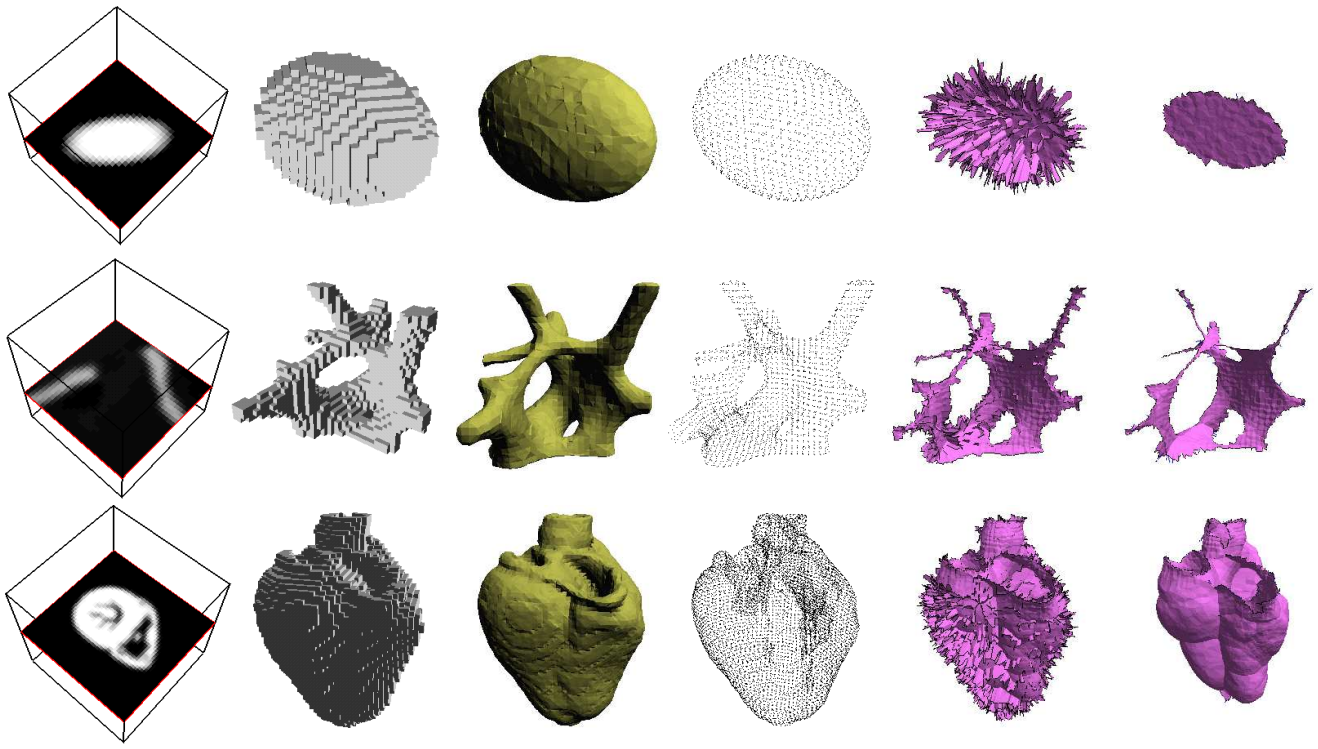


Figure 4. (a) Volumetric image. (b) Digital surface. (c) Iso-surface. (d) C-vertices of the iso-surface. (e) skeleton extracted from the Voronoi graph of the C-vertices. (f) Simplified skeleton.

framework of the joint incentive action "Beating Heart" of the research groups ISIS, ALP and MSPC of the French National Center for Scientific Research (CNRS). Jacques-Olivier Lachaud is supported by a Lavoisier post-doctoral grant from the French Ministry of Foreign Affairs.

References

[1] D. Attali and A. Montanvert. Modeling noise for a better simplification of skeletons. In *International Conference on Image Processing*, volume 3, pages 13–16, Lausanne, Switzerland, 1996.

[2] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380, Cambridge, MA, 1967. M.I.T. Press.

[3] J. D. Boissonnat. Shape reconstruction from planar cross sections. *Comput. Vision, Graph. and Image Proc.*, 44:1–29, 1988.

[4] G. Borgefors. Centres of maximal disks in the 5-7-11 distance transform. In *8th Scandinavian Conf. on Image Analysis*, pages 105–111, Tromsø, Norway, 1993.

[5] J. W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CVGIP: Image Understanding*, 59(1):116–124, 1994.

[6] L. Calabi and W.E. Hartnett. Shape recognition, prairie fires, convex deficiencies and skeletons. Technical Report No. 1, Parke Mathematical Laboratories Inc., Carlisle, MA, 1966.

[7] A. Van Gelder and J. Wilhelms. Topological Considerations in Isosurface Generation. *ACM Transactions on Graphics*, 13(4):337–375, 1994.

[8] P.-S. Jouk, Y. Usson, G. Michalowicz, and F. Parazza. Mapping of the orientation of myocardial cells by means of polarized light and confocal scanning laser microscopy. *Microscopy Research and Technique*, 30:480–490, 1995.

[9] J.-O. Lachaud. Topologically Defined Iso-surfaces. In *6th Discrete Geometry for Computer Imagery*, volume 1176 of *LNCS*, pages 245–256, Lyon, France, 1996. Springer-Verlag.

[10] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies - a comprehensive survey. *IEEE Trans. on PAMI*, 14(9):869–885, 1992.

[11] Ta-Chih Lee and Rangasami L. Kashyap. Building skeelton models via 3-d medial surface/axis thinning algorithms. *CVGIP: Graphical Models and Image Proc.*, 56(6):462–478, 1994.

[12] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[13] G. Malandain and G. Bertrand. Fast characterization of 3D simple points. In *11th ICPR*, pages 232–235, 1992.

[14] S. Miguet and V. Marion-Poty. A new 2-D and 3-D thinning algorithm based on successive border generations. In *4th Discrete Geometry for Computer Imagery*, pages 195–206, Grenoble, 1994.

[15] F. Peyrin, M. Pateyron, A.M. Laval-Jeantet, and P. Cloetens. Analysis of bone micro-architecture from high resolution 3d synchrotron radiation microtomography images. In *10th Conference of the European Society of Biochemistry*, page 211, Louvain, Belgique, 1996.

[16] F. P. Preparata and M. I. Shamos. *Computational Geometry : an Introduction.* Springer-Verlag, 1988.

[17] V. Ranwez and P. Soille. Order independent homotopic thinning. In G. Bertrand, M. Couprie, and L. Perroton, editors, *Discrete Geometry for Computer Imagery'99*, volume 1568 of *LNCS*, pages 337–346. Springer-Verlag, 1999.

[18] E. Thiel. *Les distances de chanfrein en analyse d'images : Fondements et Applications.* PhD thesis, Université Joseph Fourier - Grenoble I, 1994.