Digital Surfaces as a Basis for Building Isosurfaces

Jacques-Olivier Lachaud

Equipe INFODIS, lab. TIMC-IMAG
Institut A. Bonniot, Domaine de la Merci
38706 La Tronche Cedex, FRANCE
Jacques-Olivier.Lachaud@imag.fr

Annick Montanvert

Equipe INFODIS, lab. TIMC-IMAG
Institut A. Bonniot, Domaine de la Merci
38706 La Tronche Cedex, FRANCE
Annick.Montanvert@imag.fr

Abstract

This work highlights the relation existing between isosurfaces of an image with a given threshold (as classically computed by a Marching-Cubes algorithm) and digital surfaces of this thresholded image. The first step has been to extend the classical adjacency relation defined between elements of a digital surface. It turns out that the induced surface graphs are 2D combinatorial manifolds without boundary which can easily be mapped into closed and orientable surfaces in $\mathbb{R}^3$. Hence, digital surfaces can be processed to compute corresponding isosurfaces; the converse is also true.

1 Introduction

Volumetric imaging is an active domain of research because of its many applications in the biomedical area. One of its main challenges is the extraction of geometric representations from a volumetric image (a three-dimensional matrix of intensity values). Medical practitioners need these representations to carry out many purposes such as accurate visualization, quantitative analysis, surgical planning and simulation, radiotherapy planning, prosthesis manufacturing. Biologists study them to understand the structures of microscopic entities.

Numerous approaches exist to tackle this issue, and some have the special feature to exploit the discrete nature of images (we refer to these approaches as *discrete approaches*). This feature makes these approaches very fast, and they are thus widely used for visualization at interactive rates. This paper focuses on the two major discrete approaches (digital surface tracking, isosurface construction by Marching-Cubes) and highlights their relations.

Digital surfaces have been introduced by Liu [1]. A digital surface is a set of elements called *surfels*, which can be seen as faces of voxels (a more formal definition follows), and which separates *interior* voxels from *exterior voxels*: the image is thus made binary with a user-given threshold. Artzy *et al.* [2] have proposed an algorithm to extract *closed* digital surfaces by tracking surfels through their adjacencies. An improved version of this algorithm has been presented by Gordon et Udupa [3]. If the image has a size n^3, then the extraction of a connected digital surface can be performed in $O(n^2)$ time complexity. Therefore this method is intensively used in the biomedical area for fast visualization [3]. Drawbacks are the geometrical and topological characteristics of these surfaces. Seen from close quarters, such surfaces look rather jagged because of their digital geometry. Besides, they are generally not 2-manifolds in the Euclidean space $\mathbb{R}^3$.

Isosurfaces as provided by the Marching-Cubes algorithm [4] have also proved to be useful in the same domain (i.e. visualization). Isosurfaces are triangulated surfaces which are meant to approximate the shape of an isopotential inside the image transformed into a continuous field (for instance by tri-linear interpolation). The isopotential value is given by the user and is therefore analogous to a *threshold* over the image. The Marching-cubes algorithm builds an isosurface by scanning all image voxels eight by eight (these eight voxels forming a "cube"); inside each "cube", a precomputed set of triangles is extracted depending on the binary values of the eight voxels ($2^8 = 256$ different configurations). With some improved algorithms (see [5] for a survey), the resulting surface is a 2-manifold in $\mathbb{R}^3$. Consequently, although algorithms derived from Marching-Cubes are slower than digital surface algorithms (their time complexity is in $O(n^3)$), they provide surfaces with interesting properties (both geometrical and topological), and may be used as a preprocessing for segmentation or shape recovery algorithms [6, 7]: deformable models can start their process with this rough initialization.

These two approaches are often used independently.

In this article, we show that several different isosurfaces can easily be derived from a digital surface with connectedness considerations. First, we recall some classical definitions of digital topology. Then, we add an adjacency relation between surfels of a digital image to build a surface graph (classical algorithms of surface tracking [2, 3] can be used to do it). Finally, we show that any surface graph directly defines a 2-manifold in $\mathbb{R}^3$ which corresponds to the intuitive notion of isosurface.

2 Digital topology definitions

An *image* I is a couple $(\mathcal{E}, h)$ where h is a mapping from a subset $\mathcal{E}$ of $\mathbb{Z}^3$, called the *support* of I, toward a set $D(I)$ called the *value domain* of I. An image whose domain is $\{0,1\}$ is called a *binary image*. Any thresholded image is a binary image. In the following, I is any binary image with a finite support and I^- is its negative. A *voxel* v is an element of $\mathcal{E}$; $h(v)$ is the *value* of the voxel v in I. A voxel of value 0 (resp. 1) is a *0-voxel* (resp. *1-voxel*). The *background* $\mathcal{N}(I)$ (resp. *foreground* $\mathcal{U}(I)$) of I is the subset of 0-voxels (resp. 1-voxels) of $\mathcal{E}$. We only consider images whose border is either a subset of the foreground or a subset of the background. Under these hypotheses, I is a *scene over* $\mathbb{Z}^3$ as defined in [8].

We use the classical definitions of *ρ-adjacency* in a digital space for $\rho \in \{6, 18, 26\}$. Definitions of *strict ρ-adjacency*, *ρ-connectedness* in a set of voxels and *ρ-components* of a set are supposed to be known (refer to [8] otherwise). Two ρ-adjacent (resp. strictly ρ-adjacent) voxels u and v are denoted $\rho(u,v)$ (resp. $\check{\rho}(u,v)$). We define an *8-cube* $\mathcal{C}_8$ as a set of eight voxels such that $\forall u, v \in \mathcal{C}_8, u \neq v \Rightarrow 26(u,v)$. An 8-cube has six *4-faces*, each composed of 4 voxels. Two strictly 26-adjacent m-voxels are contained in a unique 8-cube of I, say C. C forms a *strict 26-configuration for m-voxels* if all other voxels of C are $(1-m)$-voxels.

For any 6-adjacent voxels v and v', the pair $\{v, v'\}$ is called a *surfel* (surfel for surface element). The oriented pair (v, v') is called an *oriented surfel*. A *digital surface* is a non-empty set of surfels. The *boundary* $\partial(A, B)$ of any two disjoint sets of voxels A and B is the set of surfels $\{(v, v') \in (A, B) \mid 6(v, v')\}$.

Definition 2.1 ($\kappa\lambda$-boundary) Let $\mathcal{O}$ be a κ-component of $\mathcal{U}(I)$ and $\mathcal{Q}$ a λ-component $\mathcal{N}(I)$. The (oriented) digital surface $\partial(\mathcal{O}, \mathcal{Q})$ is called a *$\kappa\lambda$-boundary* of I if not empty.

The surfels (v, v') such that $v \in \mathcal{U}(I)$ and $v' \in \mathcal{N}(I)$ are called the *bels* (for boundary elements) of I. The set of all bels of I is denoted $\mathcal{B}(I)$.

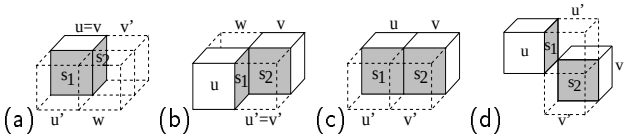

Figure 1: Illustration of $\sigma_{\kappa.\lambda}$-adjacency between surfels: (a) point (i); (b) point (ii); (c) point (iii); (d) point (iv).

The four connectedness couples $(18,6)$, $(6,18)$, $(26,6)$ and $(6,26)$, called *valid couples*, are *Jordan pairs* (of $\mathbb{Z}^3$): for these connectedness couples (κ, λ) it has indeed been demonstrated [9, 10] that any $\kappa\lambda$-boundary has a κ-connected interior and a λ-connected exterior, and that any 6-path from the interior to the exterior contains a bel (see [8] for more formal definitions). Therefore, $\kappa\lambda$-boundaries of an image follow a Jordan-like theorem and separate κ-components of the foreground from λ-components of the background.

3 Digital surface and surface graph

We define a local adjacency relation between bels of I. This adjacency relation allows the tracking of digital surfaces. Unless otherwise stated, (κ, λ) is a valid couple (hence a Jordan pair), κ (resp. λ) is the connectedness of the 1-voxels (resp. 0-voxels).

Definition 3.1 ($\sigma_{\kappa.\lambda}$-adjacency) Let $s_1 = (u, u')$ and $s_2 = (v, v')$ be two bels. These surfels are said to be $\sigma_{\kappa.\lambda}$-adjacent if one of the statements below is true (see Figure 1):

(i) $u = v$ and, either $\lambda(u', v')$ or the voxel w such that $6(u', w)$ and $6(w, v')$ is a 0-voxel;

(ii) $u' = v'$ and, either $\kappa(u, v)$ or the voxel w such that $6(u, w)$ and $6(w, v)$ is a 1-voxel;

(iii) $6(u, v)$ and $6(u', v')$;

(iv) $\check{\kappa}(u, v)$ and $\check{\lambda}(u', v')$.

This adjacency relation induces a $\sigma_{\kappa.\lambda}$-connectedness and $\sigma_{\kappa.\lambda}$-components in $\mathcal{B}(I)$.

Points (i), (ii), (iii) are classical definitions of adjacency between bels sharing an edge for couples (18,6) and (6,18) [8] (or [2, 3] when oriented). As far as we know, only Perroton et Miguet [11, 10] have introduced *one* adjacency link between surfels in case of Figure 1d and proved the validity of a surface tracking algorithm for these digital surfaces. Point (iv) builds *six* links in a strict 26-configuration for 1-voxels (resp. 0-voxels) when $\kappa = 26$ (resp. $\lambda = 26$) (the tracking algorithm of [11] is still valid). From this, we define:

Definition 3.2 (Surface graph) Let Σ be a $\kappa\lambda$-boundary of I. The *$\sigma_{\kappa.\lambda}$-surface graph* $G_{\sigma_{\kappa.\lambda}}(\Sigma)$ is

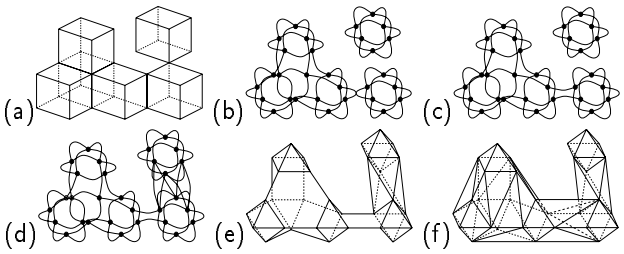

Figure 2: (a) Voxel representation of an image; (b) its $\sigma_{6.\lambda}$-surface graph; (c) its $\sigma_{18.6}$-surface graph; (d) its $\sigma_{26.6}$-surface graph; (e) the $\sigma_{26.6}$-surface graph displayed as a combinatorial manifold; (f) convex subdivision in $\mathbb{R}^3$ of (e).

the finite graph whose vertices are the surfels of Σ and whose arcs correspond to two $\sigma_{\kappa.\lambda}$-adjacent surfels of Σ. The *$\sigma_{\kappa.\lambda}$-surface graph of I* $G_{\sigma_{\kappa.\lambda}}(\mathcal{B}(I))$ is the (disjoint) union of all the $\sigma_{\kappa.\lambda}$-surface graphs defined by the $\kappa\lambda$-boundaries of I.

With a little abuse (and to simplify notations), this (abstract) graph is identified to its corresponding graph embedded in $\mathbb{R}^3$, where each surfel is associated with its centroid and each arc is associated with an open segment between centroids.

Arcs defined by Definition 3.1.(i-iii) are called *1-arcs* (the intersection of the two bels is a 1-cell). Arcs defined by Definition 3.1.(iv) are called *0-arcs* (the intersection is a 0-cell). An illustration of surface graphs for several Jordan pairs is given in Figure 2a-d. An interesting property (which cannot be achieved when bel adjacency is restricted to bels sharing an edge) is:

Theorem 3.1 *Let (κ, λ) be a valid couple (hence a Jordan pair of $\mathbb{Z}^3$). Every $\kappa\lambda$-boundary in every binary image is a $\sigma_{\kappa.\lambda}$-component of $\mathcal{B}(I)$ (i.e., $(\kappa, \lambda, \sigma_{\kappa.\lambda})$ is a* Jordan triple *in the terminology of [8]).*

Proof. For (6,18) (and (18,6) with I^-), a demonstration is in [8] because $\sigma_{18.6}$ is indeed a bel adjacency. For (26,6) (and (6,26)) we only sketch a demonstration: each 26-component is divided into its 18-components. On these 18-components the $\sigma_{26.6}$-surface graph F (say) corresponds to the $\sigma_{18.6}$-surface graph. Every strict 26-configuration for 1-voxels induces $\sigma_{26.6}$-adjacencies between bels of different $\sigma_{18.6}$-components of F, thus connecting them together. □

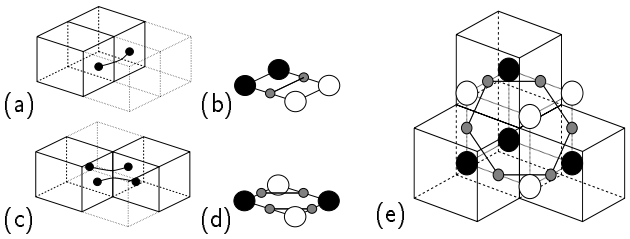

Figure 3: (a) (resp. (b)) displays $\sigma_{\kappa.\lambda}$-adjacencies inside a 4-face in the digital space (resp. in the lattice space), for any (κ, λ); (c) and (b) also, but only for $\sigma_{18.6}$- or $\sigma_{26.6}$-adjacency; (e) displays a loop defined by an 8-cube: all arcs are 1-arcs (the voxel and lattice representations are mixed).

4 Surface graph and Marching-Cubes isosurface

In this section, we exhibit the link existing between the surface graph and a surface generated by an algorithm similar to the Marching-Cubes (often called an *isosurface*): we show indeed that any surface graph can be transformed into a *2-manifold in $\mathbb{R}^3$ without boundary* (i.e., a closed and orientable surface without self-intersection in $\mathbb{R}^3$ which may have several connected components).

The *lattice representation* $I^\star$ of an image I maps voxels to their centroids at integer coordinates in $\mathbb{R}^3$: a *$\star$-point* $v^\star$ is associated to each voxel v, a *$\star$-edge* (open segment) to each surfel, a *$\star$-face* (open square) to each 4-face and a *$\star$-cube* (open cube) to each 8-cube. Any point belonging to a $\star$-edge is called a *midpoint* and can be assimilated to the surfel $\{u, v\}$ corresponding to the $\star$-edge. This midpoint is said to be *separating* if $\{u, v\}$ is an (unoriented) bel of I. See Figure 3 for a visual interpretation of the two following lemmas:

Lemma 4.1 *Any vertex of a surface graph of I is a separating midpoint. Any 1-arc (resp. 0-arc) of a surface graph is included in a $\star$-face (resp. $\star$-cube) of I.*

Lemma 4.2 *Let $\mathcal{C}_8$ be any 8-cube. Let SG be the subgraph of $G_{\sigma_{\kappa.\lambda}}(\mathcal{B}(I))$* induced by *$\mathcal{C}_8$ (i.e., the vertices of SG are the vertices of $G_{\sigma_{\kappa.\lambda}}(\mathcal{B}(I))$ whose surfels are defined by voxels of $\mathcal{C}_8$, idem for the arcs). Then the set of arcs of SG can be arranged into a set of loops, such that each 1-arc is visited exactly once and each 0-arc is visited exactly twice.*

Note that vertices created by a Marching-Cubes algorithm on an image for a threshold are separating midpoints of the thresholded image. In fact, there is

one-to-one correspondence between these vertices and the vertices of $G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$.

If G is a finite graph and L a set of loops over G. Then L forms an *umbrella* around a vertex $v \in G$ if all the loops of L containing v can be arranged into a circular permutation $(L_0, \ldots, L_{l-1})$ such that L_i has one common arc with L_{i+1}, indices taken modulo l.

Theorem 4.1 *The set of arcs of the $\sigma_{\kappa,\lambda}$-surface graph of I can be arranged into a set of loops such that each arc is visited exactly twice. Moreover this set of loops forms an umbrella around each vertex of the surface graph. Consequently, the $\sigma_{\kappa,\lambda}$-surface graph of I together with this set of loops, denoted $\Diamond G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ is a 2D combinatorial manifold without boundary [12] (see Figure 2e).*

Proof. Each 0-arc belongs to exactly one $\star$-cube (Lemma 4.1) and can only be visited by loops defined over it (Lemma 4.2). Each 1-arc belongs to one $\star$-face which is the face of two $\star$-cubes. Hence it belongs to two loops defined over these two $\star$-cubes. Any vertex of the surface graph belongs to four 8-cubes; inside each 8-cube, this vertex belongs to exactly one loop; two arcs of this loop are incident to this vertex, and each of these arcs also belongs to a loop on an adjacent 8-cube. This point shows that an umbrella can be formed around each vertex. The loops of $\Diamond G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ are the ones defined in each 8-cube of the image (Lemma 4.2). □

This theorem shows that every loop belongs to one 8-cube: loops of the combinatorial manifold can thus be computed locally. To obtain a *triangulation*, the loops of $\Diamond G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ must be subdivided (i.e., triangulated) into loops composed of three arcs. We call *convex subdivision* the following scheme to subdivide each loop: any loop $L = (a_1, \ldots, a_k)$ is built from an 8-cube $\mathcal{C}_8$ (Lemma 4.2). Every (a_i) is a separating midpoint. For $\kappa \in \{18, 26\}$, let H be the closed convex hull in $\mathbb{R}^3$ of firstly the points (a_i), secondly the $\star$-points of the 1-voxels of $\mathcal{C}_8$. The loop L is a Jordan curve of the boundary of H. Consequently H induces a subdivision of L into planar elements defined by a subset of the points (a_i). We decompose L into loops (L_j) according to these subsets. Any L_j either has three arcs or has a planar embedding (and is then randomly subdivided). For $\lambda \in \{18, 26\}$, the process is symmetrical. The Figure 4 depicts a convex subdivision inside an 8-cube. With this *triangulation*, we infer:

Theorem 4.2 *Let $\triangle G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ be the convex subdivision of $\Diamond G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$. Then the embedding of $\triangle G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ in the Euclidean space such that its vertices are mapped into midpoints, its arcs into open edges between midpoints, its loops (composed of three arcs) into the open triangles bordered by the edges, is a 2-manifold in $\mathbb{R}^3$ without boundary (see Figure 2f). It is also orientable, because every closed non-orientable 2-manifold intersects itself in $\mathbb{R}^3$.*

Proof. The umbrella around each vertex of $\triangle G_{\sigma_{\kappa,\lambda}}(\mathcal{B}(I))$ ensures that each vertex has a neighborhood locally homeomorphic to $\mathbb{R}^2$. On an edge, the two faces sharing it also define a suitable neighborhood. The argument is trivial for a point on a face. Now, it can be proved that this surface do not intersect itself [6] (self-intersection can only occur between loops defined in the same 8-cube). □

It can be shown [6] that this 2-manifold separates the embeddings of the κ-components of 1-voxels from the λ-components of 0-voxels (i.e., the surface does not intersect the "connections" between two connected 1-voxels or 0-voxels). The 2-manifold generated for the negative image with inverse connectedness couple $(\sigma_{\lambda,\kappa})$ is exactly the same. Because of these properties, this 2-manifold is called a $\kappa\lambda$-isosurface. Figure 5 illustrates the extraction of three different $\kappa\lambda$-isosurfaces from a binary image.

5 Conclusion

The two previous theorems imply that a 2-manifold in $\mathbb{R}^3$ can be built from any $\sigma_{\kappa,\lambda}$-surface graph of an image *locally* (when (κ, λ) is a Jordan pair). The Marching-Cubes algorithm has exactly the same purpose and is indeed built over the same vertices (i.e., the separating midpoints). Now, a surface tracking algorithm (such [2] or [11] for $\{26, 6\}$), which stores bel adjacencies in memory, builds a surface graph of a $\kappa\lambda$-boundary in $O(n^2)$ time complexity (for an image of size n^3). The traversal of the surface graph has the same complexity and each convex subdivision can be made in $O(1)$. Hence, the computation of a closed 2-manifold in $\mathbb{R}^3$ (i.e., a Marching-Cubes isosurface) that separates a κ-component of the foreground from a λ-component of the background can be done in $O(n^2)$. Note that the classical hole problem in the Marching-Cubes algorithm [5] is solved in passing.

It is a significant improvement to the Marching-Cubes algorithm (whose time complexity is $O(n^3)$) when we do not need to compute the whole isosurface of an image but merely *one of its components*. Moreover, it shows that a digital surface can be transformed into an isosurface easily and that the converse can also be realized.

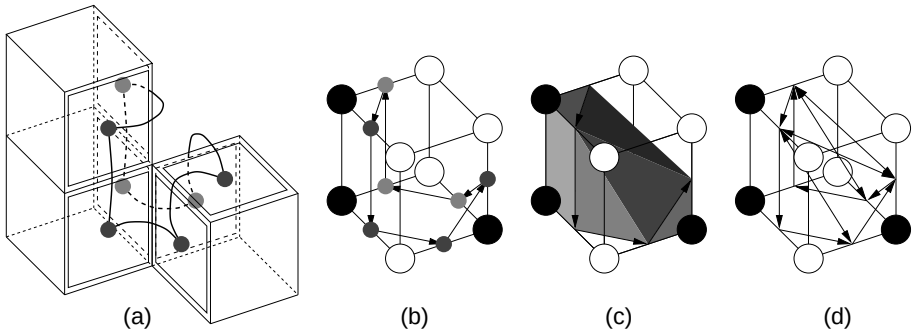

Figure 4: These figures display the creation of a loop and its convex subdivision: (a) set of bels (depicted with thicker squares) belonging to an 8-cube, centroids of these bels along with their adjacencies (case $\sigma_{18.6}$ or $\sigma_{26.6}$); (b) lattice representation of this 8-cube, loop built on the graph of bel-adjacencies; (c) closed convex hull defined on this 8-cube (the loop forms a Jordan curve on the surface of this set); (d) convex subdivision on the loop determined by this set.

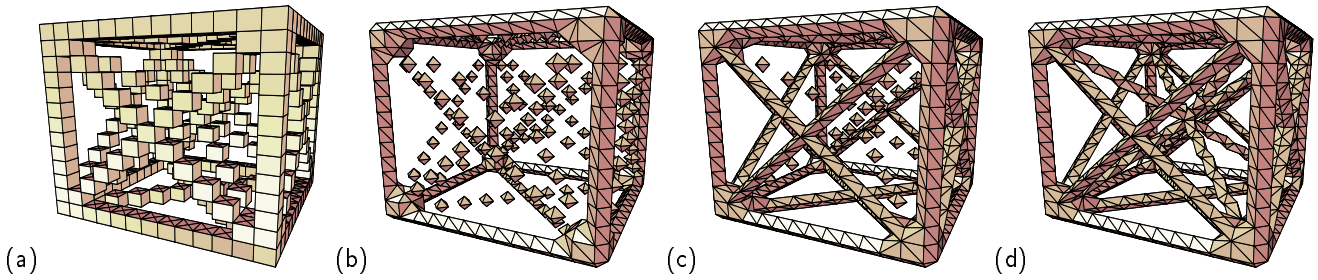

Figure 5: (a) Digital surface of a "connection" cube; (b) the 2-manifold derived from the $\sigma_{6.\lambda}$-surface graph (the $\sigma_{6.18}$- and the $\sigma_{6.26}$-surface graphs are the same on this image); (c) the 2-manifold derived from the $\sigma_{18.6}$-surface graph; (d) the 2-manifold derived from the $\sigma_{26.6}$-surface graph.

References

[1] H.K. Liu. Two and three dimensional boundary detection. *Computer Graphics and Image Processing*, 6(2):123–134, April 1977.

[2] E. Artzy, G. Frieder, and G.T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Graphics and Image Processing*, 15:1–24, 1981.

[3] D. Gordon and J.K. Udupa. Fast surface tracking in three-dimensional binary images. *Computer Vision, Graphics, and Image Processing*, 45(2):196–241, February 1989.

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

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

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

[7] C. Xu, D.L. Pham, and J.L. Prince. Finding the brain cortex using fuzzy segmentation, isosurfaces, and deformable surface models. In J. Duncan and G. Gindi, editors, *Proc. of 15th Int. Conf. Information Processing in Medical Imaging (IPMI'97), Poultney, Vermont, USA*, volume 1230 of *Lecture Notes in Computer Science*. Springer-Verlag, June 1997.

[8] J.K. Udupa. Multidimensional Digital Boundaries. *CVGIP: Graphical Models and Image Processing*, 56(4):311–323, July 1994.

[9] G.T. Herman and D. Webster. A topological proof of a surface tracking algorithm. *Computer Vision, Graphics, and Image Processing*, 23:162–177, 1983.

[10] S. Miguet and L. Perroton. Discrete surfaces of 26-connected sets of voxels. In *Proc. of 5th Discrete Geometry for Computer Imagery*, Clermont-Ferrand, France, September 1995.

[11] L. Perroton. A 26-connected object surface tracking algorithm. *Géométrie discrète en imagerie, fondements et applications*, 1:1–10, September 1993.

[12] J. Françon. Discrete Combinatorial Surfaces. *CVGIP: Graphical Models and Image Processing*, 57(1):20–26, January 1995.