Iso-Surface Processing

Topologically Defined Isosurfaces

Jacques-Olivier Lachaud

LIP, ENS-Lyon, URA CNRS 1398
46, allée d'Italie, 69364 LYON Cedex 7
tel: 72.72.85.03 fax: 72.72.80.80
e-mail: jolachau@lip.ens-lyon.fr

Abstract. In this article, we present a new process for defining and building the set of configurations of Marching-Cubes algorithms. Our aim is to extract a topologically correct isosurface from a volumetric image. Our approach exploits the underlying discrete topology of voxels. Our main contribution is to provide a formal proof of the validity of the generated isosurface. The generated isosurface is a closed, oriented surface without singularity with no self-intersection. Furthermore, we demonstrate that it separates the foreground from the background. Finally we show that the graph defining the isosurface is closely linked to the *surfel*-adjacency graph of the digital surface of the same image.

1 Introduction

The Marching-Cubes algorithm, presented in [8], provides a fast solution to extract an isosurface from an image with a given threshold. This algorithm has been intensively used for rendering purposes but has shown limits in other applications, because the extracted isosurface is generally not a "simple" surface. Many authors have contributed to solve this problem (see Section 2.1) but they have often provided an empirical solution or a visual justification. Some authors [16] have noticed that the isosurface cannot always have the same topology that an underlying continuous surface defining the image.

In this work, we propose a topological approach to this problem. First, we recall major definitions on digital geometry in Section 2. Then, we demonstrate formally the coherence of the isosurface (closeness, orientability, no singularity, no self-intersection) in Section 3 along with fundamental properties: the extracted isosurface is the opposite (in terms of orientation) of the isosurface extracted for the opposite image with connectedness considerations, the foreground with a given connectedness is inside the surface and the background (with another connectedness) is outside the surface. The two last properties are proved in Section 4. Section 5 validates our method with examples on synthetic images and on medical images. A link between isosurface and digital surface is established in Section 5.4.

2 Isosurfaces and Digital Geometry

The Marching-Cubes algorithm is based on the tabulation of 256 different configurations. It allows a local isosurface extraction inside a cube defined by eight voxels. The global isosurface is the union of these small pieces of surface computed on each set of eight connected voxels of the image. For a precise survey, see [8]. *We emphasize here that we only modify the table of configurations; that is the reason why the Marching-Cubes algorithm itself remains unchanged.* Furthermore, we provide four different tables of configurations, because several connectednesses may be chosen for the foreground and for the background.

2.1 Topologically Correct Isosurfaces

Our purpose is to build locally a surface by starting from an image of voxels. Each voxel is either a voxel of the foreground or a voxel of the background. The corresponding surface must separate the foreground from the background. The Marching-Cubes is indeed often used to solve this problem. However the coherence of this surface is not guaranteed as was stressed by, for instance, [18] and [2]. The surface has indeed some holes in it. Many researches have been done to empirically correct the topology of the surface [18] [11] [13]. The surfaces generated by these methods are topologically correct (simple and closed 2-manifold). Despite this fact, the generated surface does not represent exactly the underlying volume. This problem was mainly studied by [16] who tested the methods by applying them on several continuous fields.

On the contrary, a topological approach, based on digital geometry considerations, escapes these problems and guarantees the consistency of the generated isosurface. Moreover, as far as we know, no author has ever explored the relationships between digital surfaces (and especially Jordan surfaces) of a set of voxels, and isosurfaces locally computed.

2.2 3D Digital Geometry Definitions

We are only interested in finite pictures. Furthermore we consider only pictures where there are only 0-voxels or only 1-voxels on the border of the support. In the following we use the notions of a *binary picture* (denoted I), and its *inverse picture* (denoted $\overline{I}$). We denote $\mathcal{N}(I)$ the *background* of I (composed of *0-voxels*) and $\mathcal{U}(I)$ the *foreground* of I (composed of *1-voxels*). We suppose that the notions of 6− 18− and 26−*adjacency* between voxels, of *connectedness* between voxels of the same kind, and of *connected components*, are known. Let κ be an adjacency relation. We denote $\mathcal{G}_\kappa(\mathcal{O})$ the *κ-adjacency graph* of a set of 1-voxels $\mathcal{O}$. The *1-map* of $\mathcal{G}_\kappa(\mathcal{O})$ is the canonical embedding of this graph in $\mathbb{R}^3$ according to the localization of the center of the voxels of $\mathcal{O}$. It is denoted $\mathcal{G}^1_\kappa(\mathcal{O})$. We have the similar definitions for 0-voxels considered with another adjacency relation λ. We denote $\mathcal{Z}^1_\kappa$ the 1-map of the whole space $\mathbb{Z}^3$ considered as a set of voxels with the κ-adjacency.

We define an *8-cube* as a set of voxels $v_0, \ldots, v_7$ such that for all i and j belonging to $\{0, \ldots, 7\}$, v_i and v_j are 26-connected if $i \neq j$. We classically order the eight voxels such that the coordinates depend on the binary coding of the indice. We denote $\mathrm{Extr}(A)$ the *extremal points* of a convex set A and $\mathrm{Fr}(A)$ the *frontier* of A which is equal to the adherence of A minus the interior of A.

2.3 Extension of Adjacency Graph

Our purpose is to segment volumetric images according to the connectedness. For instance, the obtained surface must not intersect the connection (defined by the adjacency relation) between two voxels of the same kind (i.e. 0-voxels or 1-voxels). So we introduce the following definitions:

Definition 1 (3-map of $\mathbb{Z}^3$). We defined the *3-map* of the space $\mathbb{Z}^3$ considered with the κ-adjacency, that we denoted $\mathcal{Z}^3_\kappa$, as the polyhedral complex derived from the 1-map $\mathcal{Z}^1_\kappa$ such that: (i) the vertices and the edges are those of $\mathcal{Z}^1_\kappa$; vertices are the *0-cells* of this complex, edges are the *1-cells*; (ii) the faces or *2-cells* (resp. the volumes or *3-cells*) are the minimal closed convex cells of dimension 2 (resp. 3) that can be formed using the 1-cells (resp. 2-cells);

Definition 2 (κ-adjacency m-map). We define the *κ-adjacency m-map* of a set of voxels $\mathcal{O}$, for $0 \leq m \leq 3$, and denote $\mathcal{G}^m_\kappa(\mathcal{O})$, as a subset of $\mathcal{Z}^3_\kappa$ such that $\mathcal{G}^m_\kappa(\mathcal{O}) = \{p \in \mathcal{Z}^3_\kappa \mid \dim p \leq m \text{ and } \mathrm{Extr}(p) \text{ are voxels of } \mathcal{O}\}$.

The graph $\mathcal{G}^3_\kappa(\mathcal{O})$ is the "volume" occupied by a set of voxels considered with a given connectedness. Note that $\mathcal{Z}^3_6$ is the standard decomposition of $\mathbb{R}^3$ into unit-cubes, squares, edges and vertices. Let S be a closed and oriented surface, $\mathrm{Int}(S)$ its strict inner part, $\mathrm{Ext}(S)$ its strict outer part. Then S includes a set of κ-connected voxels iff the κ-adjacency 3-map of the voxels is included in $\mathrm{Int}(S)$.

3 Isosurface Generation

3.1 Isosurface Properties

Let I be a binary picture. Let κ be a given connectedness for the 1-voxels and λ for the 0-voxels. We denote by $\mathcal{M}_{\kappa\lambda}$ the process which transforms a binary picture into a surface according to the chosen connectednesses. We now precise the properties which the generated surface $\mathcal{S}_{\kappa\lambda} = \mathcal{M}_{\kappa\lambda}(I)$ must follow:

(i) $\mathcal{S}_{\kappa\lambda}$ is a simple, closed and oriented surface;
(ii) $\mathcal{S}_{\kappa\lambda}$ is embedded in $\mathbb{R}^3$ without self-crossing;
(iii) If $\overline{I}$ is the inverse of the image I, then we must have $\overline{\mathcal{M}_{\kappa\lambda}(I)} = \mathcal{M}_{\lambda\kappa}(\overline{I})$;
(iv) $\mathcal{G}^3_\kappa(\mathcal{U}(I)) \subset \mathrm{Int}(\mathcal{S}_{\kappa\lambda})$;
(v) $\mathcal{G}^3_\lambda(\mathcal{N}(I)) \subset \mathrm{Ext}(\mathcal{S}_{\kappa\lambda})$.

Point (i) specifies the kind of surface we want to obtain. Point (ii) forbids self-crossing. Point (iii) expresses that the isosurface generated for a picture I must be the same that the isosurface generated for the picture $\overline{I}$ up to its orientation. Points (iv) and (v) force the isosurface to separate the κ-components of 1-voxels from the λ-components of 0-voxels. We emphasize that these conditions force the surface to lie within the limits defined by the connections between voxels. *In the following, the complete definitions and demonstrations may be found in [7].*

3.2 Local Reconstruction and Local Graph

In order to build the isosurface, we decompose the binary picture into the set of all 8-cubes of I. For each 8-cube, a small piece of surface is built inside. We show after that the global resulting surface follows the properties described above.

Let $\mathcal{C}_8(v_0, \ldots, v_7)$ be an 8-cube of a binary picture I. An 8-cube has six faces (or *4-faces*), each composed of four voxels. One 4-face is shared by exactly two 8-cubes. One 4-face has an inner side and an outer side for a given 8-cube. We choose the orientation for each inner 4-face as described in Figure 2.a; the 4-faces are then said to be *ordered.* We call an *8-configuration* (resp. *a 4-configuration*) an 8-cube (resp. a 4-face) associated with the value of each voxel (0 or 1). A couple of two 6-connected voxels is called a *link*. The *value* of a link is the value of its voxels. An inner 4-face possesses four links in the clockwise direction. Notice that ordered 4-faces are coded with the Hamming coding.

We apply the following rules on each link of an ordered 4-face according to its value:

- $(0,0)$ or $(1,1)$: there is no vertex created between these two voxels;
- $(0,1)$: there is one vertex between these two voxels and there is an incoming oriented edge on this vertex which stays on this 4-face;
- $(1,0)$: there is one vertex between these two voxels and there is an outgoing oriented edge on this vertex which stays on this 4-face.

Consider the results of these rules for each 4-configuration on Figure 1.a. 14 configurations provide an immediate result and only cases 6 and 9 must be specially handled. Here the connectedness of the foreground and of the background is essential to give a coherent solution to these cases. If κ (resp. λ) is the connectedness of the 1-voxels (resp. 0-voxels), the Figure 1.b shows that a correct solution cannot be obtained for every couple (κ, λ). As a matter of fact, the generated surface must follow point (iii) of Section 3.1. An arbitrary choice in any one of the cases $\{(6,6),(18,18),(18,26),(26,18),(26,26)\}$ would provide a locally different surface in the inverse picture. *In the following, the connectedness couples* $(6,18)$, $(6,26)$, $(18,6)$, $(26,6)$, *will be called* valid couples.

We emphasize that, on a 4-face, no oriented edge intersects the adjacency sub-graphs of the foreground and of the background (when the couple is valid). Let us denote $G_{\kappa\lambda}(\mathcal{C}_8)$ the graph obtained by gathering all the oriented edges generated on the six oriented inner 4-faces. This graph is called the *local graph* of the 8-cube $\mathcal{C}_8$.

Theorem 3. *If* (κ, λ) *is a valid couple, the local graph* $G_{\kappa\lambda}(\mathcal{C}_8)$ *is a set of oriented loops of length* ≥ 3, *each loop being disconnected from the others.*

Proof. We can use a Karnaugh table to represent the whole 8-cube. A close look at Figure 2.b shows that each link is an element of two 4-faces and that each link is systematically covered twice in both directions. According to the four rules above, either no vertex is created on this link (case $(0,0)$ and $(1,1)$), or there is a vertex on this link with an incoming edge $(0,1)$ and an outgoing edge $((1,0)$ when it is covered in the other 4-face). Because all vertices of the graph possess

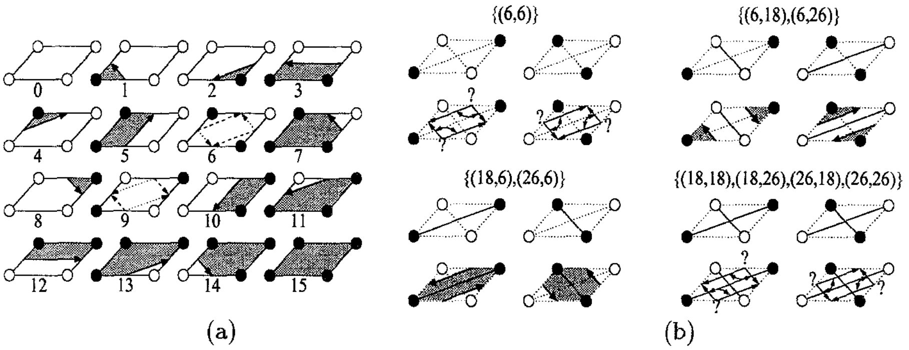

Fig. 1. (a) The 16 different 4-configurations for a 4-face; two cases remain indeterminate, (b) resolution of problematic cases according to the connectedness.

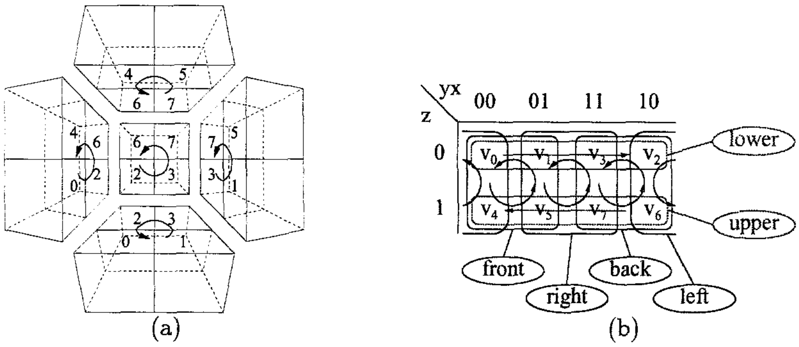

Fig. 2. (a) Orientation chosen for the 4-faces (the front 4-face is not visible), (b) the six oriented inner 4-faces coded in a Karnaugh table.

exactly one incoming edge and one outgoing edge, the graph trivially forms a set of oriented loops with no common vertex or edge. It is easy to show that there is no loop of length 1 or 2. □

Local Graph in case of strict 26-adjacency. By the same way the connectedness was influencing the generation of the local graph on 4-faces, the 26-connectedness modifies the local graph in 8-configurations where two voxels are strictly 26-connected. If $\mathcal{C}_8$ possesses two strictly 26-adjacent 1-voxels (resp. 0-voxels) and $\kappa = 26$ (resp. $\lambda = 26$), the local graph is transformed and the two disconnected loops are triangulated (see Figure 3). This critical modification explains why the embedding of the redefined local graph in $\mathbb{R}^3$ does not intersect the adjacency 3-map of two strictly 26-adjacent voxels.

3.3 Global Isosurface

We use the notion of *combinatorial manifold* (see [3]) to prove that the generated isosurface fulfills the properties of Section 3.1.

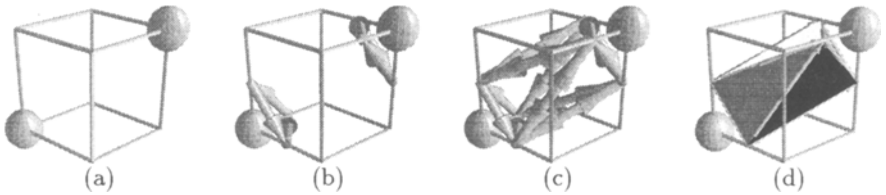

Fig. 3. The configuration of strict 26-adjacency: (a) displays the configuration, (b) shows the oriented loops generated for a couple (26,6), (c) shows the corresponding subdivision, (d) displays the isosurface derived from the subdivision.

Definition 4 (Complete Graph). Let I be a binary picture. By extension, we define the *complete graph* of I, and denote it $G_{\kappa\lambda}(I)$, as the set of all local graphs built from the 8-cubes of I.

We call a *subdivision* of a loop (with a length at least 3) any triangulation of this loop. A manifold derived from another manifold M by replacing every loop of length strictly greater than 3 with any subdivision of this loop is by extension called a *subdivision* of M. Using Theorem 3 we can prove that each edge of the complete graph of I belongs to exactly two loops (with two different orientations). The definition of oriented manifolds, together with the redefinition of the local graph in the configuration of strict 26-adjacency, provide

Theorem 5. *If (κ, λ) is a valid couple, then the complete graph $G_{\kappa\lambda}(I)$ of a binary picture I is an oriented two-dimensional combinatorial manifold without boundary. Moreover, if M is any subdivision of $G_{\kappa\lambda}(I)$, then its canonical embedding is a closed and oriented triangulated surface without singularity. Furthermore this surface is embedded in $\mathbb{R}^3$ without self-crossing.*

4 Correct Subdivision for the Generated Manifold

The Marching-Cubes process $\mathcal{M}_{\kappa\lambda}(I)$ is defined as the canonical embedding of a particular subdivision of $G_{\kappa\lambda}(I)$.

4.1 Generated Volume and Induced Local Surface

In this section, we fix $\lambda = 6$. Moreover we suppose that $\mathcal{C}_8$ does not form the configuration of strict 26-adjacency. In this case, one may notice that there is only one κ-component in $\mathcal{C}_8$. Let $L_1, \ldots, L_k$ be the oriented loops of the local graph $G_{\kappa\lambda}(\mathcal{C}_8)$. We denote O the embedding of the 1-voxels of $\mathcal{C}_8$ and A the embedding of the vertices of the loops (L_i). We define the *generated volume* $V_{\kappa 6}(\mathcal{C}_8)$ as the convex hull of $O \cup A$. It can be shown that $O \cup A$ are exactly the extremal points of $V_{\kappa 6}(\mathcal{C}_8)$. Because the (embeddings of the) loops $L_1, \ldots, L_k$ are Jordan curves of $\mathrm{Fr}(V_{\kappa 6}(\mathcal{C}_8))$, we obtain

Proposition 6. *Each loop L_i splits $Fr(V_{\kappa 6}(\mathcal{C}_8))$ into two open surfaces. One of these two surfaces, called the interior surface of L_i, contains no element of $O \cup A$. Consequently the interior surface of L_i implicitly defines a subdivision for L_i; this subdivision is only defined by vertices of L_i.*

4.2 Marching-Cubes Surface

We can now completely define the triangulated manifold generated by the Marching-Cubes process as well as the corresponding isosurface.

Definition 7 (Local triangulated manifold). Every 8-cube $\mathcal{C}_8$ of the binary picture I together with a valid connectedness couple (κ, λ) defines a local graph $G_{\kappa\lambda}(\mathcal{C}_8)$ that is a set of oriented loops $L_1, \ldots, L_k$. Note that we have clearly $G_{\lambda\kappa}(\mathcal{C}_8) = \overline{G_{\kappa\lambda}(\overline{\mathcal{C}_8})}$. That is the reason why we define the *local triangulated manifold* $S_{\kappa\lambda}(\mathcal{C}_8)$ by cases:

- case $\lambda = 6$; if $\mathcal{C}_8$ is a configuration of strict 26-adjacency then, either $\kappa = 26$ and $S_{\kappa\lambda}(\mathcal{C}_8)$ is defined as in Section 3.2, or $\kappa = 18$ and $S_{\kappa\lambda}(\mathcal{C}_8) = G_{\kappa\lambda}(\mathcal{C}_8)$; else $S_{\kappa\lambda}(\mathcal{C}_8)$ is defined as the subdivision of $G_{\kappa\lambda}(\mathcal{C}_8)$ implicitly defined by the interior surfaces of $L_1, \ldots, L_k$ (see Proposition 6);
- case $\kappa = 6$; then $S_{\kappa\lambda}(\mathcal{C}_8)$ is defined as $S_{\lambda\kappa}(\overline{\mathcal{C}_8})$ with the opposite orientation (which is already defined by the previous case).

Definition 8 (Marching-Cubes Surface or Isosurface). We define the *isosurface* of the binary picture I considered with valid connectedness (κ, λ), and note $\mathcal{M}_{\kappa\lambda}(I)$, as the canonical embedding of the complete graph of I $(G_{\kappa\lambda}(I))$ with the subdivision of Definition 7.

The following theorem can be obtained by fixing $\lambda = 6$. Its proof is based on the convexity of the generated volume (see [7] for complete demonstrations).

Theorem 9. *Let $(\kappa, \lambda) \in \{(26, 6), (18, 6)\}$. Then the λ-adjacency 3-map of $\mathcal{N}(I)$ is a subset of Ext($\mathcal{M}_{\kappa 6}(I)$) and the κ-adjacency 3-map of $\mathcal{U}(I)$ is a subset of Int($\mathcal{M}_{\kappa 6}(I)$).*

The theorem below states that the isosurface fulfills the requested properties.

Theorem 10. *If (κ, λ) is a valid connectedness couple then $\mathcal{M}_{\kappa\lambda}(I)$ respects properties (i), (ii), (iii), (iv), (v) of Section 3.1.*

Proof. (i) and (ii) come from Theorem 5. Definition 7 proves property (iii). Let I be a binary picture. If $\lambda = 6$ then Theorem 9 proves property (iv) and (v). If $\lambda \neq 6$ then $\kappa = 6$ in order to have a valid couple. In this case, we use property (iii) and the fact that $\text{Ext}(S) = \text{Int}(\overline{S})$ for Jordan surfaces to conclude. □

5 Results and Properties

5.1 Implementation

Locally, the generated isosurface depends only on the local configuration of voxels. This means that we can compute once and for all the 256 different configurations for a given connectedness couple. We will build four different tables, one for each valid connectedness couple. A classical Marching-Cubes algorithm

```
Procedure Build_Local_Surface(Configuration $C$, Connectedness $\kappa$)
Let $E \longleftarrow \emptyset$,
For all inner 4-faces $F$ of $C$ do
    $E_F \longleftarrow$ Extract_edges($F$,$\kappa$,6)
    $E \longleftarrow E \cup E_F$
End for all
$\mathcal{L} \longleftarrow$ Extract_loops($E$)
$H \longleftarrow$ Convex_Hull(get_vertices($\mathcal{L}$) $\cup$ get_1-voxels($C$))
While $\mathcal{L}$ contains a loop $L$ of length $> 3$ do
    $L \longleftarrow$ Remove_a_loop_of_length> 3($\mathcal{L}$)
    $L$ is an ordered set $(a_1, \ldots, a_k)$ of vertices
    Let $exit \longleftarrow false$, $i \longleftarrow 1$, $j \longleftarrow i + 2$
    Repeat
        If $(a_i, a_j) \in H$ then $exit \longleftarrow true$
        Else
            $j \longleftarrow j + 1$
            If $j = k$ then $i \longleftarrow i + 1$, $j \longleftarrow i + 2$
        End If
    Until exit
    Insert_loop($(a_1, \ldots, a_i, a_j, \ldots, a_k)$, $\mathcal{L}$)
    Insert_loop($a_i, \ldots, a_j)$, $\mathcal{L}$)
End While
If Configuration_strict_26-adj($C$) and $\kappa = 26$ then
    $\mathcal{L} \longleftarrow$ build_special_loops_26-adj($\mathcal{L}$)
End If
Return $\mathcal{L}$
```

Fig. 4. Algorithm for building local configurations with couples (18,6) and (26,6).

may thus use directly one of these tables to build a coherent isosurface without any modification in the source code (unlike other methods such that [13] which impose to modify the algorithm). Moreover classical optimizations [1] [17] of the Marching-Cubes are not influenced by these modifications. Figure 4 summarizes the algorithm used to compute one local configuration for the connectedness couples (18,6) and (26,6). The two other tables are computed using the two previous tables and property (iii). The procedure of Figure 4 must be called with the 256 different configurations.

5.2 Results on Synthetic Data

Figure 5 shows the building process for a configuration considered with different connectednesses. Note that the result is identical whether one of the connectedness is 18 or 26. This is due to the fact that the 26-connectedness behaves differently than the 18-connectedness only for configurations of strict 26-adjacency. Figure 3 shows the behaviour of the algorithm in the configuration of strict 26-adjacency. If the connectedness of the foreground is 26, then a triangulation is done over the vertices of the local graph; if not, the local graph is not modified.

We test our modified Marching-Cubes algorithm on synthetic images in order to highlight the influence of connectedness on the results. We exhibit here an example where the image represents a cube with eight connected vertices. Figure 6 shows the image and the various results for the couples (26,6), (18,6) and anyone of $\{(6,18),(6,26)\}$.

One may notice that we never use the fact that a vertex of the isosurface stands at the middle of the two voxels which define it. *All our properties are*

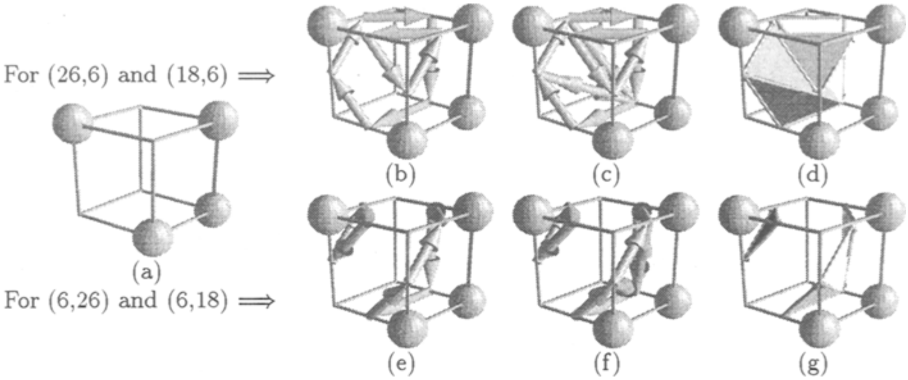

Fig. 5. An example of a configuration with the corresponding local surface: (a) displays the configuration, (b) (resp. (e)) shows the oriented loops generated for a couple (18,6) or (26,6) (resp. (6,18) or (6,26)), (c) and (f) show the corresponding subdivisions, (d) and (g) display the isosurfaces derived from the subdivisions.

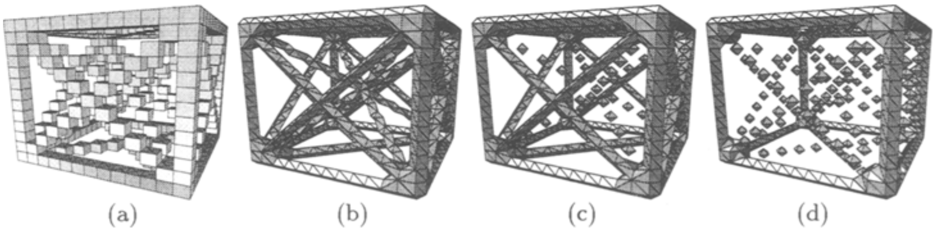

Fig. 6. Test of the Marching-Cubes over a "connection" cube: (a) displays the image, (b) result of the Marching-Cubes algorithm with $(\kappa, \lambda) = (26, 6)$, (c) with $(\kappa, \lambda) = (18, 6)$, (d) with $(\kappa, \lambda) \in \{(6, 18), (6, 26)\}$.

still valid if created vertices stand in the open segment linking the two bordering voxels. This ability for the vertex to lie anywhere on this segment is critical in visualization, where the surface must be as smooth as possible [8].

5.3 Results on Medical Data

Figure 7.a shows an isosurface of a computed tomography. The size of the image is 256 × 256 × 113. Note that we make no use of the interpolation to smooth the result. With this method, the edge length of the isosurface is extremely regular. Now, because the computed isosurface is a "simple" surface, it can be considered as a deformable surface and exploited as an initialization for another process. For instance, we have used the snake-like algorithm described in [6] to deform and smooth the isosurface according to physical constraints. The result (see Figure 7.b) is better than a direct visualization of a Marching-Cubes. For instance, the isosurface computed by the Marching-Cubes possesses 354 connected components (and about 295,000 vertices) whereas the deformed surface possesses only 45 connected components (and about 191,000 vertices).

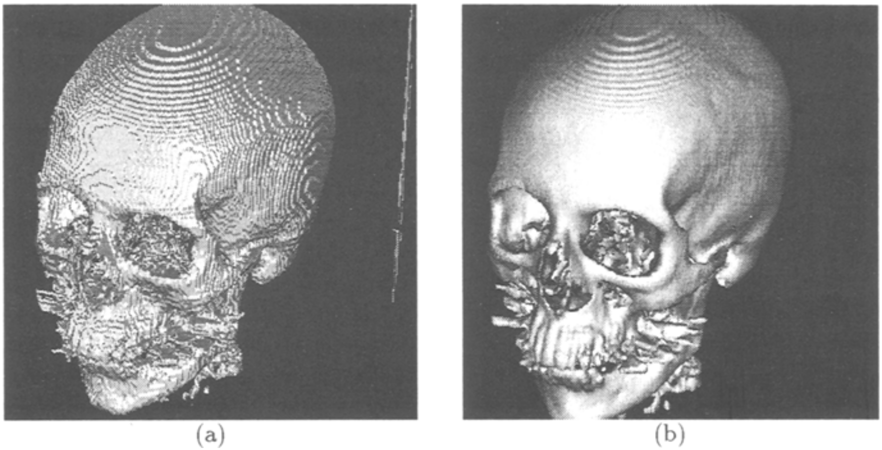

Fig. 7. Results on medical data: (a) isosurface of a Computed Tomography, (b) after processing of a snake-like algorithm.

5.4 Properties of Isosurfaces

In this section we prove that the complete graph of I, considered with the connectedness couple (κ, λ), is exactly the graph of surfel-adjacency of the corresponding digital surface of I defined with the same connectedness couple.

A description of digital surfaces in $\mathbb{Z}^n$ may be found in [4]. A digital surface that lies between a κ-component $\mathcal{O}$ of $\mathcal{U}(I)$ and a λ-component $\mathcal{Q}$ of $\mathcal{N}(I)$ is a *$\kappa\lambda$-boundary* of I. The first interesting result is that these surfaces follow a Jordan-like theorem for couples (18,6) and (6,18) [14] and for couples (26,6) and (6,26) [10]. We can establish a parallel between that fact and the coherence of generated isosurfaces for these connectedness couples.

The study of algorithm of digital surface tracking has provided the notion of adjacency between surfels or *surfel-adjacency.* The surfel adjacency is closely linked to the underlying adjacency between the voxels which defined the surfels. Refer to [5] for precise definitions and to [15] for the theory of boundaries in $\mathbb{Z}^n$.

Definition 11 (Surfel-Adjacency). Let $s_1 = (u, u')$ and $s_2 = (v, v')$ be two surfels. These surfels are said to be $\sigma_{\kappa\lambda}$-adjacent if

(i) $u = v$ and either u' is λ-adjacent to v' or $w \in \mathcal{Q}$ (see Figure 8.a);
(ii) $u' = v'$ and either u is κ-adjacent to v or $w \in \mathcal{O}$ (Figure 8.c);
(iii) $6(u, v)$ and $6(u', v')$ (Figure 8.b);
(iv) u is strictly κ-adjacent to v and u' is strictly λ-adjacent to v' (Figure 8.d).

Points (i), (ii), (iii) are classical definitions of the surfel-adjacency for the connectedness couples (18,6) and (6,18). For the couples (26,6) and (6,26), as far as we know, only the authors of [10] have introduced an adjacency between surfels of two strictly 26-adjacent voxels, by arbitrarily connecting two surfels of these voxels. They have proved the validity of a surface tracking algorithm for these digital surfaces in [12].

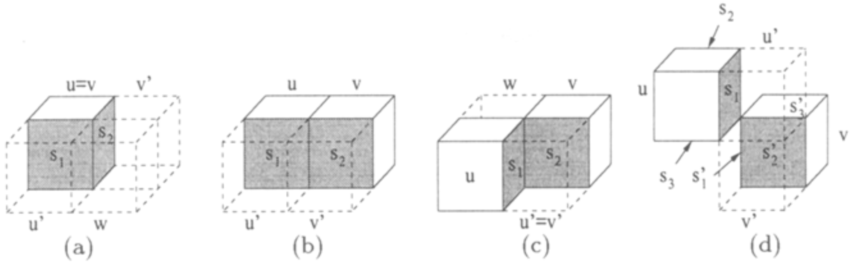

Fig. 8. Description of surfel-adjacency of digital surfaces.

If this definition of $\sigma_{26.6}$-adjacency is sufficient for a surface tracking algorithm where only one connection is required, it has no real topological meaning. That is why we introduce the point (iv) of Definition 11. This definition states that two surfels are $\sigma_{26.6}$-adjacent if they share a vertex and if they do not lie on the same plane (see Figure 8.d). As a matter of fact, we have $\sigma_{26.6}(s_i, s'_j)$ when $i \neq j$. With this definition, the surface tracking algorithm of [12] is still valid and we can deduce the following properties:

Theorem 12. *Let $\Sigma(\mathcal{O}, \mathcal{Q})$ be the $\kappa\lambda$-boundary of $\mathcal{O}$ and $\mathcal{Q}$. Let $G_{\kappa\lambda}(\mathcal{O} \cup \mathcal{Q})$ be the restriction of the complete graph of I to the vertices of $\mathcal{O} \cup \mathcal{Q}$. Then the $\sigma_{\kappa\lambda}$-adjacency graph of $\Sigma(\mathcal{O}, \mathcal{Q})$ is identical to the graph $G_{\kappa\lambda}(\mathcal{O} \cup \mathcal{Q})$.*

Corollary 13. *Let Σ be the digital surface defined by $\mathcal{U}(I)$ and $\mathcal{N}(I)$. Let $G_{\kappa\lambda}(I)$ be the complete graph of I. Then the $\sigma_{\kappa\lambda}$-adjacency graph of Σ is identical to the graph $G_{\kappa\lambda}(I)$.*

A very interesting consequence of this theorem is that a Marching-Cubes over a component of the foreground and a component of the background can be computed in $O(s)$ time if s is the number of surfels of I, by using a surface tracking algorithm to build the complete graph. The computation of the classical Marching-Cubes is in $O(n^3)$, where n^3 is the size of the volumetric image. Usually, a digital surface has $O(n^2)$ elements. It may thus be interesting to use this property when computing one isosurface of the volumetric image I.

6 Conclusion

We have presented a new topological and formally proved approach for extracting consistent isosurfaces from volumetric images. For a given connectedness couple, a corresponding isosurface is built which separates foreground elements (d-cells of the foreground) from background elements (d'-cells of the background). The identity between the underlying graph of the isosurface and the surfel-adjacency graph of the digital surface has been established. The algorithm has been implemented and validated with synthetic data. It is at the moment used in segmentation of medical images. Additional informations may be found in [7].

Acknowledgements

The author thanks Jean-Marc Nicod and Serge Miguet for providing a parallel version of the original Marching-Cubes algorithm (see [9] for a description). Tests on real data were performed by this algorithm with modified configuration tables. The author also thanks the reviewers for helpful comments and suggestions.

References

1. J.-H. Chuang and W.-C. Lee. Efficient Generation of Isosurfaces in Volume Rendering. *Computer & Graphics*, 19(6):805–813, 1995.
2. M.J. Dürst. Additionnal Reference to Marching Cubes. *Computer Graphics*, 22(2):72–73, 1988.
3. J. Françon. Discrete Combinatorial Surfaces. *CVGIP: Graphical Models and Image Processing*, 57(1):20–26, January 1995.
4. G.T. Herman. "Discrete Multidimensional Jordan Surfaces". *CVGIP*, 54(6):507–515, November 1992.
5. T.Y. Kong and J.K. Udupa. A justification of a fast surface tracking algorithm. *CVGIP: Graphical Models and Image Processing*, 54(6):507–515, November 1992.
6. J-O. Lachaud and A. Montanvert. "Volumic Segmentation using Hierarchical Representation and Triangulated Surface". In *6th European Conference on Computer Vision*, pages 137–146, Cambridge, UK, April 1996.
7. J.O. Lachaud. "Topologically Defined Isosurfaces". Research Report 96-20, LIP - ENS Lyon, France, 1996.
8. W. E. Lorensen and H. E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". *Computer Graphics*, 21:163–169, January 1987.
9. S. Miguet and J-M. Nicod. "A load-balanced parallel implementation of the Marching-Cube algorithm". In *HPCS*, July 1995.
10. S. Miguet and L. Perroton. "Discrete surfaces of 26-connected sets of voxels". In *5th Discrete Geometry for Computer Imagery*, September 1995.
11. G.M. Nielson and B. Hamman. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In IEEE, editor, *Visualization'91*, pages 83–90, San Diego, 1991.
12. L. Perroton. A 26-connected object surface tracking algorithm. *Géométrie discrète en imagerie, fondements et applications*, 1:1–10, September 1993.
13. S. Röll, A. Haase, and M. von Kienlin. "Fast Generation of Leakproof Surfaces from Well-Defined Objects by a Modified Marching Cubes Algorithm". *Computer Graphics Forum*, 14(2):127–138, January 1995.
14. A. Rosenfeld, T.Y. Kong, and A.Y. Wu. "Digital Surfaces". *CVGIP*, 53(4):305–312, July 1991.
15. J.K. Udupa. Multidimensional Digital Boundaries. *CVGIP: Graphical Models and Image Processing*, 56(4):311–323, July 1994.
16. A. van Gelder and J. Wilhelms. Topological Considerations in Isosurface Generation. *ACM Transactions on Graphics*, 13(4):337–375, October 1994.
17. J. Wilhelms and A. van Gelder. Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics*, 11(3):201–227, July 1992.
18. G. Wyvill, C. McPheeters, and B. Wyvill. Data Structure for Soft Objects. *The visual Computer*, 2:227–234, 1986.