



# A cellular model for multi-objects multi-dimensional homotopic deformations

Yann Cointepas<sup>a</sup>, Isabelle Bloch<sup>a,\*</sup>, Line Garnero<sup>b</sup>

<sup>a</sup>ENST, Département TSI, CNRS URA 820, 46 rue Barrault, 75634 Paris, Cedex 13, France

<sup>b</sup>LENA, CNRS URA 654, 47 Bd de l'Hôpital, 75651 Paris, Cedex 13, France

Received 29 December 1999; received in revised form 23 May 2000; accepted 23 May 2000

---

## Abstract

We deal in this paper with complex three-dimensional scenes, partitioned into several objects of any shape, and with varying local dimensions. We propose a consistent topological modeling of such scenes, based on a cellular model. We introduce a definition of simple elements for any adjacency graph-based geometrical structure. Then we prove a local characterization of simple cells of a cellular model, in the case of two objects, and of any number of objects. This characterization allows to define homotopic deformations of a complex scene. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

*Keywords:* Discrete topology; 3D cellular models; Simple elements; Homotopic deformations

---

## 1. Introduction

Objects that can be modeled by a digital picture are limited by several intrinsic topological properties of the digital grid. Let us consider for instance the classical cubic grid in a three-dimensional discrete space. The smallest object that can be modeled is an elementary cube (voxel). Therefore, it is not possible to model directly structures thinner than a voxel such as a surface. Moreover, attempts to develop a consistent topology based on the adjacency graph failed due to the well-known connectivity paradox [1,2]. This paradox can be avoided in the case of scenes composed of a background and of disjoint object components (by using two different connectivities). But if the scene is more complex and contains for instance nested objects or more complicated configurations, it is difficult to get rid of the paradox [3]. This problem becomes even more complicated if one has to apply deformations to the modeled objects.

In order to go beyond the modeling limits of digital pictures and to get rid of topological paradoxes, we present a cellular complex-based model: the cellular model. Complex scenes composed of many objects with different local dimensions can be represented in one cellular model. This model is an extension of the model we proposed for only one object in Ref. [4]. Then we introduce a new theoretical framework which allows the generalization of discrete homotopic deformation definitions. With this framework, we prove that any scene represented in a cellular model can be homotopically deformed. At last, we propose an optimal implementation in order to use our theoretical results in real applications.

In Section 2, we address jointly two problems: one for representing objects having parts of different local dimensions (Fig. 3), and the other dealing with complex topological arrangement of objects (Fig. 2). In Section 3, we introduce a new definition of simple points which is only based on adjacency graphs and can therefore be applied to any geometrical structure provided with a neighborhood relationship. This definition is used in Section 3.3 to locally characterize the simple cells of a cellular model. In Section 4, we extend the simple point notion to the case of a scene composed of several objects.

---

\* Corresponding author. Tel.: + 33-1-45-81-75-85; fax: + 33-1-45-81-37-94.

E-mail addresses: [yann.cointepas@enst.fr](mailto:yann.cointepas@enst.fr) (Y. Cointepas), [isabelle.bloch@enst.fr](mailto:isabelle.bloch@enst.fr) (I. Bloch), [lenalg@ext.jussieu.fr](mailto:lenalg@ext.jussieu.fr) (L. Garnero).

As a result, all the modeling properties of cellular models can be used jointly with homotopic deformations. Section 5 describes how to implement a cellular model. And we briefly present an application in Section 6.

**2. The cellular model**

Several approaches have been proposed to define discrete surfaces in  $\mathbb{R}^3$  [5]. The methods which provide a direct representation of a surface use a set of *surfels* which is a set of faces between pairs of adjacent voxels [6,7]. Such a structure, composed of both voxels and surfels, allows for the representation of both volumes (set of voxels) and surfaces (set of surfels) in the same model. Unfortunately, such a structure is not free from topological paradoxes, such as the non-respect of Jordan’s theorem, which are well known in digital topology [2,8,9]. A more complex image structure can be introduced in order to get rid of the topological paradoxes. This structure, first introduced and used in 2-D image processing by Kovalevsky [10], is based on *cellular complexes* [11,12].

A *cellular model*  $M = (E, V, P)$  is a cellular complex-based structure embedded in the three-dimensional cubic grid.  $E$  is the set of elements (or *cells*) of different dimensions which compose the cubic grid. These elements are the cubes (or voxels), the faces (or surfels) between two cubes, the segments between two faces and the points between two segments.  $V$  is a *connectivity relationship* ( $V \subset E \times E$ ). Two cells are connected (or neighbors) if one bounds the other. For example, a cube is bounded by six faces, 12 segments and eight points. It has 26 neighbors. A face is bounded by four lines and four points, and it bounds two cubes, so it has 10 neighbors (Fig. 1).  $P$  is a partition of  $E$ . Each element of  $P$  represents an *object*.

Unlike classical digital pictures where connectivity depends on voxel labels [8], the underlying graph of a cellular model is fixed. In other words, the connectivity relationship of a cellular model  $M = (E, V, P)$  does not depend on  $P$ . Therefore, in order to model an object with a cellular model, one just has to decide which cells belong to the object. The membership to the object of all the cells is explicit. The presence of cells of lower dimension makes it possible to control object borders precisely. For example, if an object is composed of two cubes sharing an edge, the membership of the edge determines if the object is connected or not. Thus, if we compare the voxel connectivities between a cellular model and a classical digital pictures, we can say that the cellular model allows to locally choose the adjacency. For example, a cellular object can be composed of parts made of “8-connected” voxels and of parts made of “18-connected” voxels. It is therefore possible to model any scene composed of voxel-based objects. This would have been impossible in some cases with classical images [3] (Fig. 2).

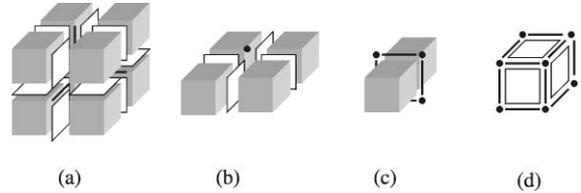


Fig. 1. The four neighborhoods of a cellular model: (a) the neighborhood of a point, (b) the neighborhood of a segment, (c) the neighborhood of a face and (d) the neighborhood of a cube.

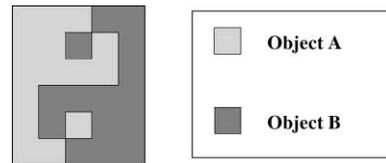


Fig. 2. The connectivity problem. With a classical image, it is not possible to find an appropriate connectivity for each object to be connected, with a cellular model the scene can be represented.

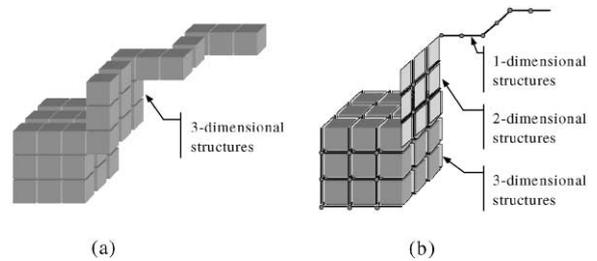


Fig. 3. Representation of objects with different local dimensions. (a) Structures made up with voxels are always three-dimensional. (b) Thin structures can be represented with a cellular model.

The combination of cells of different dimensions in a cellular model makes it possible to represent objects with different local dimensions. Not only is it possible to represent a volume by a set of voxels linked by two-, one- and zero-dimensional cells, or represent a surface by a set of surfels linked by one- and zero-dimensional cells, but both volumes and surfaces can be mixed in the same model. It is thus possible, for example, to have pure two-dimensional structures and three-dimensional objects in the same model (Fig. 3).

The possibility to represent objects with different local dimensions can be useful in applications where complex objects have to be modeled. For example, to build a three-dimensional model of the brain it is necessary to represent the thin parts of the cortex surface which are located in the cortical folds (Fig. 4). A precise model of

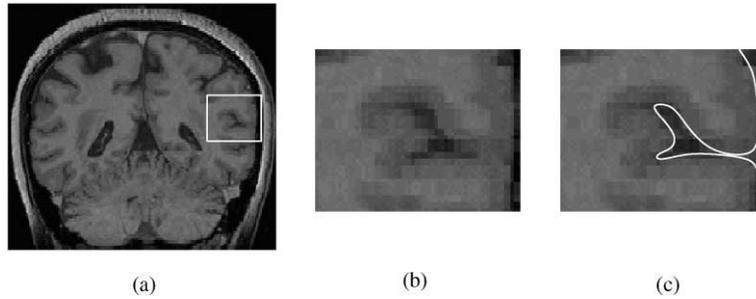


Fig. 4. Partial volume effect can hide cortical surface. (a) MR slice of the head. (b) Zoom on the white rectangle of image (a). A part of the cortical surface is not visible in the MR image. (c) Diagram of the cortex surface superimposed with image (b).

the brain volume and of its surface (which is the cortex surface) can be obtained from a magnetic resonance image (MRI) with a cellular model [13].

Not only a cellular model makes it possible to overcome the modeling limits of classical images, but it also offers good topological properties which allow for the definition of an homotopic deformation of a cellular model. This definition, which is based on the characterization of simple cells, is presented in the following sections.

### 3. Homotopic deformations

Discrete homotopic deformations are widely used in image thinning processes [8,14–16]. The goal of image thinning is to deform an object modeled by an image into a skeleton having the same topology as the object. In general topology, there is no unique definition of topology-preserving deformations. But, in the three-dimensional digital topology framework, this notion is usually defined as the preservation of the number of *connected components* and of the number of *tunnels*. The voxels (or points) that can be modified in an image without changing the topology are called *simple points*. Several authors proposed a local characterization of simple points [8,17–19]. The simplest local characterization consists in counting the connected components in the neighborhood of the considered point. In the following, we extend these notions to cellular models. In order to be as general as possible, we define homotopic deformations for any graph. Then, we use the local properties of the neighborhood graph of the cellular model to find a local characterization of *simple cells*.

#### 3.1. Basic notions

**Definition 3.1.** An *abstract graph*  $G$  is a pair  $(E, V)$  where  $E$  is a discrete set of abstract elements and  $V$  is a set of ordered pairs  $(x, y)$  where  $x \in E$  and  $y \in E$ .  $V$  is called *neighborhood relationship*.

In the following we consider a non-oriented graph  $G = (E, V)$ , i.e.  $(x, y) \in V \Leftrightarrow (y, x) \in V$ , and such that  $\forall x \in E, (x, x) \in V$ . The pair  $(E, V)$  of a cellular model is an example of abstract graph. We note  $O$  a subset of  $E$  and  $\bar{O} = E - O$  the complementary of  $O$ .

**Definition 3.2.** Two elements  $x$  and  $y$  of  $O$  are *neighbors* on  $O$  (denoted by  $x \stackrel{O}{\sim} y$ ) if and only if  $(x, y) \in V$ . The set of neighbors of an element  $s$  of  $E$  (not necessary in  $O$ ) belonging to  $O$  is denoted by  $N_O(s)$ .

**Definition 3.3.** A *path*  $\gamma$  of  $O$  is either the *empty path*  $\gamma_\emptyset$ , or a series  $s_0, s_1, \dots, s_n$  ( $n \in \mathbb{N}$ ) of elements of  $O$  such that  $s_i \stackrel{O}{\sim} s_{i+1} \forall i \in [0, n - 1]$ . The element  $s_0$  (resp  $s_n$ ) is the initial (resp. final) element of  $\gamma$ . The set of paths of  $O$  is denoted by  $\Gamma_O$ .

**Definition 3.4.** Two elements  $x$  and  $y$  of  $O$  belong to the same *connected component* of  $O$  (denoted by  $x \stackrel{O}{\leftrightarrow} y$ ) if and only if there exists a path in  $\Gamma_O$  of which  $x$  is the initial element and  $y$  the final element. The *number of connected components* of  $O$  is denoted by  $N_{cc}(O)$ .

**Definition 3.5.** The *concatenation* of two paths  $\gamma = s_0, s_1, \dots, s_n$  and  $\gamma' = s'_0, s'_1, \dots, s'_m$  of  $\Gamma_O$ , such that  $s_n \stackrel{O}{\sim} s'_0$ , is the path  $\gamma \cdot \gamma' = s_0, s_1, \dots, s_n, s'_0, s'_1, \dots, s'_m$ . The empty path is the null element for concatenation:  $\forall \gamma, \gamma_\emptyset \cdot \gamma = \gamma \cdot \gamma_\emptyset = \gamma$ .

**Definition 3.6.** A *loop*  $\omega$  of  $O$  is a series  $s_0, \dots, s_n$ ,  $n \in \mathbb{N}$ , of elements of  $O$ , such that  $s_n \stackrel{O}{\sim} s_0$  and  $\forall i \in [0, n - 1], s_i \stackrel{O}{\sim} s_{i+1}$ . The set of loops of  $O$  is denoted by  $\Omega_O$ .

In order to define topological equivalence between two objects we need to be able to characterize a tunnel. There is a tunnel in an object if it contains a loop which cannot be continuously deformed into a point. Therefore we need to define continuous deformation for loops in a graph. We will use the notion of elementary deformation of a loop which is, intuitively, the smallest possible continuous deformation. The iteration of elementary

deformations on a loop forms a set of deformations that are similar to the continuous deformations.

**Definition 3.7.** Two loops  $\omega$  and  $\omega'$  of  $O$  are equivalent up to an elementary deformation (denoted by  $\omega \stackrel{O}{\rightleftharpoons} \omega'$ ) if at least one of the following conditions is true:

- $\omega' \stackrel{O}{\rightleftharpoons} \omega$ , (1)

- $\begin{cases} \omega = s_1.\pi.s_2 & \text{where } \pi \in \Gamma_O, \\ \omega' = s_2.s_1.\pi, & s_1, s_2 \in O, \end{cases}$  (2)

- $\begin{cases} \omega = \pi_1.s_1.s_2.s_3.\pi_2 & \text{where } \pi_1, \pi_2 \in \Gamma_O, \\ \omega' = \pi_1.s_1.s_3.\pi_2, & s_1, s_2, s_3 \in O. \end{cases}$  (3)

Condition (1) imposes symmetry on the relation. Condition (2) allows two loops that differ only by a circular permutation on their elements to be equivalent. Condition (3) defines an elementary deformation as a “small deviation” of a loop on three elements which are mutually neighbors (Fig. 5).

An abstract graph has no geometrical basis. Therefore, it is important to define in what sense the elementary deformation can be seen as a continuous deformation. A mapping between a graph and a geometrical structure associates each element of the graph with a geometrical object (cells of a cellular model for example). The arcs of the graph indicate an adjacency between two geometrical objects. Definition 3.7 cannot be used on any geometrical mapping (Fig. 6). In order to be valid, a mapping must respect some rules. In our definitions, the neighborhood relationship is a proximity relationship. In other words, two elements are neighbors if they are “as close as possible”. Formally, any geometrical structure composed of three or less mutually neighboring elements must have the topology of a filled sphere (i.e. one connected component and no tunnel). In this case, any loop in the object corresponding to a pair of neighbors  $(x, y)$  can be continuously deformed into any other loop in the object corresponding to a triplet of neighbors  $(x, y, z)$ . Therefore, the elementary deformation can be seen as a continuous deformation.

We are now able to continuously deform a loop by using a series of elementary deformations. This gives us the following equivalence relationship between loops:

**Definition 3.8.** Two loops  $\omega$  and  $\omega'$  of  $\Omega_O$  are equivalent by deformation if and only if at least one of the following conditions is verified (Fig. 7):

- $\omega' \stackrel{O}{\rightleftharpoons} \omega$  (4)

- there is a series  $\omega_0, \omega_1, \dots, \omega_n$  ( $n \in \mathbb{N}$ ) of loops of  $O$  such that

$$\omega \stackrel{O}{\rightleftharpoons} \omega_0, \omega_0 \stackrel{O}{\rightleftharpoons} \omega_1, \dots, \omega_{n-1} \stackrel{O}{\rightleftharpoons} \omega_n, \omega_n \stackrel{O}{\rightleftharpoons} \omega' \quad (5)$$

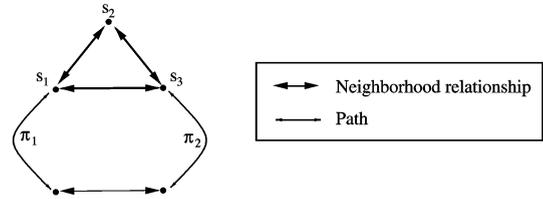


Fig. 5. Elementary deformation of a loop. The loop  $\pi_1.s_1.s_2.s_3.\pi_2$  is equivalent to the loop  $\pi_1.s_1.s_3.\pi_2$ .

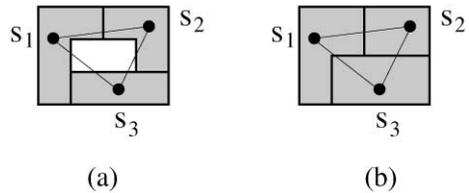


Fig. 6. (a) Example of invalid mapping between a graph and a geometrical structure. According to Definition 3.7, the loop  $s_1.s_2.s_3$  can be deformed into the loop  $s_1.s_2$ . This is not equivalent to a continuous deformation since the loop in the geometrical objects going through the points  $s_1, s_2$  and  $s_3$  cannot be continuously deformed to a loop going through  $s_1$  and  $s_2$ . (b) A valid mapping: any loop can be continuously deformed to any other loop.

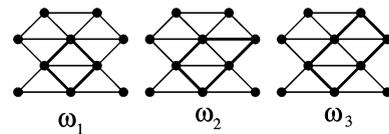


Fig. 7. Loops  $\omega_1, \omega_2$  and  $\omega_3$  are all equivalent since  $\omega_1$  and  $\omega_2$  (resp.  $\omega_2$  and  $\omega_3$ ) are equivalent by elementary deformation.

The “equivalence by deformation” relationship is an equivalence relationship on  $\Omega_O$ . The transitivity is an immediate consequence of Eq. (5). The symmetry is a consequence of the symmetry of  $\omega \stackrel{O}{\rightleftharpoons} \omega'$ . The reflexivity is a consequence of the reflexivity of the neighborhood relationship. For any  $\omega$  in  $\Omega_O$ , the equivalence class of loops of  $O$  which contains  $\omega$  is denoted by  $[\omega]_O$ .

In order to characterize tunnels we need to use the deformation of a loop into a point. Therefore, we need the following definition.

**Definition 3.9.** A loop  $\omega$  of  $\Omega_O$  is reducible in  $O$  if and only if there exists a loop  $\omega' \in [\omega]_O$  such that  $\omega' = s, s \in O$ . Otherwise the loop is said to be irreducible (Fig. 8).

### 3.2. Simple elements in a neighborhood graph

According to the previous definitions, we first define simple elements and their local characterization in the

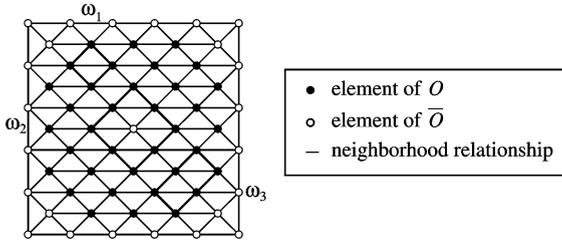


Fig. 8. Loops  $\omega_1$  and  $\omega_3$  are equivalent and reducible on  $O$ . Loop  $\omega_2$  is irreducible on  $O$  because the central point does not belong to  $O$ .

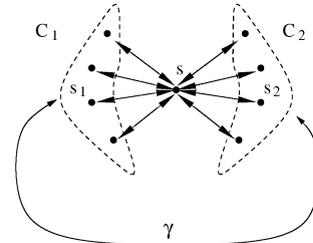


Fig. 9. In the neighborhood of  $s$ , there are two distinct connected components.

case of two objects. In Section 4, we extend these notions to any number of objects. An element of a graph is simple if it can be removed or added to an object of the graph without changing the topology of the scene. Objects of the scene are represented by a partition of the elements of the graph: each element of the partition represents an object. The number of elements in the partition is fixed. Thus, if the partition has two elements, removing an element from an object  $O$  means adding the element to its complementary  $\bar{O}$ . Therefore, we only define the simple elements that can be removed from  $O$  since adding simple elements to  $O$  can be done by exchanging the roles of  $O$  and  $\bar{O}$ .

Let  $G = (E, V)$  be an abstract graph, and  $O$  a non-empty subset of  $E$ . We note  $\bar{O} = E - O$  the complementary of  $O$ . The set of *simple elements* of  $O$ , denoted by  $S_O$ , is a subset of  $O$  such that  $\forall s \in S_O$ , the following conditions are true:

•  $\exists x \in O - \{s\}$  such that  $x \overset{O}{\leftrightarrow} s$ , (6)

•  $\exists x \in \bar{O}$  such that  $x \overset{\bar{O} + \{s\}}{\leftrightarrow} s$ , (7)

•  $\forall x, y \in O - \{s\}, x \overset{O}{\leftrightarrow} y \Rightarrow x \overset{O - \{s\}}{\leftrightarrow} y$ , (8)

•  $\forall x, y \in \bar{O}, x \overset{\bar{O}}{\leftrightarrow} y \Rightarrow x \overset{\bar{O} + \{s\}}{\leftrightarrow} y$ , (9)

•  $\forall \omega \in \Omega_O, \omega$  is irreducible in  $O \Rightarrow$  (10)

$\exists \omega' \in [\omega]_O$  such that  $\omega' \in \Omega_{O - \{s\}}$  and  $\omega$  is irreducible in  $O - \{s\}$ ,

•  $\forall \omega \in \Omega_{O - \{s\}}, \omega$  is irreducible in  $O - \{s\} \Rightarrow$  (11)

$\omega$  is irreducible in  $O$ ,

•  $\forall \omega \in \Omega_{\bar{O} + \{s\}}, \omega$  is irreducible in  $\bar{O} + \{s\} \Rightarrow$  (12)

$\exists \omega' \in [\omega]_{\bar{O} + \{s\}}$  such that  $\omega' \in \Omega_{\bar{O}}$  and

$\omega'$  is irreducible in  $\bar{O}$ ,

•  $\forall \omega \in \Omega_{\bar{O}}, \omega$  is irreducible in  $\bar{O} \Rightarrow$  (13)

$\omega$  is irreducible in  $\bar{O} + \{s\}$ .

The conditions can be divided into two groups: Eqs. (6)–(9) ensure the preservation of the number of connected components, and Eqs. (10)–(13) ensure the preservation of the number of tunnels. Eq. (6) prevents the disappearance of an isolated element of  $O$ ; Eq. (7) prevents the appearance of an isolated element in  $\bar{O}$ , Eq. (8) prevents the separation of one connected component in  $O$ , Eq. (9) prevents the fusion of two connected components of  $\bar{O}$ , Eq. (10) prevents the deletion of a tunnel in  $O$ , Eq. (11) prevents the appearance of a tunnel in  $O$ , Eq. (12) prevents the appearance of a tunnel in  $\bar{O}$  and Eq. (13) prevents the disappearance of a tunnel on  $\bar{O}$ .

We now introduce a condition that is necessary for a graph element to be simple. We then show that this condition is not sufficient in the general case.

**Proposition 3.10.** *If  $s$  is a simple element of  $O$ , then  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ .*

**Proof.** Let  $s$  be a simple element of  $O$ .

If  $N_{CC}(N_O(s) - \{s\}) = 0$ , condition (6) is not verified.

If  $N_{CC}(N_{\bar{O}}(s)) = 0$ , condition (7) is not verified.

If  $N_{CC}(N_O(s) - \{s\}) \geq 2$  then there exist two distinct connected components  $C_1$  and  $C_2$  on  $N_O(s) - \{s\}$ . Therefore, there are two possibilities:

Either  $C_1$  and  $C_2$  are not connected in  $O - \{s\}$ , then  $\exists x \in C_1$  and  $\exists y \in C_2$  which do not verify Eq. (8), because  $C_1$  and  $C_2$  are connected in  $O$  through  $s$ . Therefore  $s$  is not a simple element of  $O$ .

Or,  $\exists \gamma \in \Gamma_O, \exists s_1 \in C_1, \exists s_2 \in C_2$  such that  $\omega = s_1 . s_2 . \gamma \in \Omega_O$  (Fig. 9). For any number  $n$  of deformations applied to  $\omega$ , we will always obtain, except for a circular permutation, a loop  $\omega_n$  such that

$$\omega_n = \pi_1 . s'_1 . s . s'_2 . \pi_2$$

where  $s'_1 \in C_1, s'_2 \in C_2$  and  $\pi_1, \pi_2 \in \Gamma_O$  (14)

**Proof.** Let us prove it by recurrence on  $n$ . If  $n = 0$ , we have immediately  $\omega_0$  by putting  $s'_1 = s_1, s'_2 = s_2, \pi_1 = \gamma_0, \pi_2 = \gamma$ . If condition (14) is true for  $n = N$  with  $N \in \mathbb{N}$  then it is true for  $n = N + 1$  because an elementary

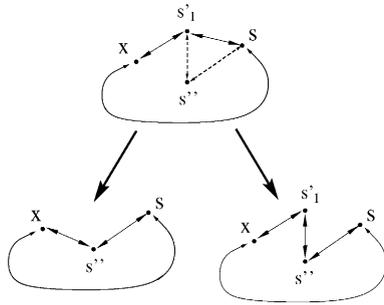


Fig. 10. Two possibilities for elementary deformation of a loop.

deformation cannot delete  $s$  from the loop, because  $s'_1$  and  $s'_2$  are not neighbors. Moreover, the elementary deformation can only delete  $s'_1$  (resp.  $s'_2$ ) if there exists  $s''$  in  $C_1$  (resp.  $C_2$ ) which would become the new predecessor (resp. successor) of  $s$  in  $\omega_{N+1}$  (Fig. 10).  $\square$

Therefore, if  $N_{CC}(N_O(s) - \{s\}) \geq 2$ , any loop of  $[O]_\omega$  contains  $s$  and at least two other distinct elements, and therefore is irreducible and cannot belong to  $\Omega_{O-\{s\}}$ .

Thus  $\omega$  is irreducible in  $O$  and this contradicts Eq. (10). Therefore,  $s$  is not a simple element of  $O$ .

If  $N_{CC}(N_O(s)) \geq 2$  we can prove in the same way as before that Eq. (12) is not respected and therefore  $s$  is not a simple element of  $O$ .  $\square$

The reciprocal of Proposition 3.10 is not true. If  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ , the four connectivity conditions (6)–(9) are true. Moreover, let us show that Eq. (10) is true. If Eq. (10) was false, we would have

$$\exists \omega \in \Omega_O \text{ such that } \omega \text{ is irreducible in } O$$

$$\text{and such that } \forall \omega' \in \Omega_{O-\{s\}} \cap [\omega]_O,$$

$$\omega' \text{ is irreducible in } O - \{s\}. \tag{15}$$

If  $\omega'$  exists, since it is equivalent to  $\omega$  and irreducible in  $O$ , then  $\omega$  is irreducible in  $O$ . It contradicts Eq. (15). Therefore  $\omega'$  does not exist, that means that all loops in  $[\omega]_O$  contain  $s$  and either  $s$  is isolated ( $N_{CC}(N_O(s) - \{s\}) = 0$ ), or there are at least two distinct connected components in  $N_O(s) - \{s\}$ . It contradicts the assumption that  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ . Therefore Eq. (10) is true.

It is possible to prove in the same way that Eq. (12) is also true. But it is possible to find a set  $E$ , a neighborhood relationship  $V$  and a subset  $O \in E$  such that  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ , and Eq. (11) is false or Eq. (13) is false (Fig. 11). This is because it is not possible to detect the appearance or disappearance of a tunnel with only the neighborhood of an element. To make it possible, it is necessary to have more assumptions about  $E$  and  $V$ .

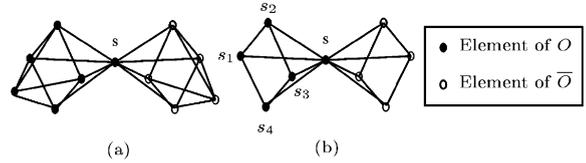


Fig. 11. (a)  $s$  is a simple element of  $O$ . (b)  $s$  is not a simple element of  $O$  because the loop  $s_1-s_2-s_3-s_4$  is reducible on  $O$  but not on  $O - \{s\}$ . In both cases,  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_O(s)) = 1$ .

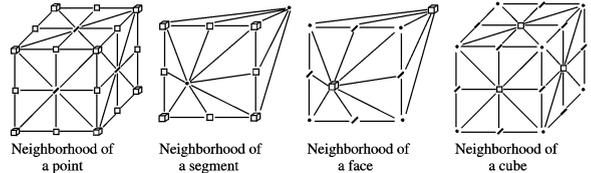


Fig. 12. The different neighborhoods of a cellular complex. Three-dimensional representation of the elements connected to a central element.

### 3.3. Homotopic deformations of a cellular model

The neighborhood graph of a cellular model makes Proposition 3.10 a necessary and sufficient condition. The neighborhood of a cellular model element depends on the type of this element. There are four types of elements: cubes, faces, lines and points. But if we see the neighborhoods as a graph where the vertices are the elements of the complex and the arcs are the connectivity relationships, we can notice that there are only two different neighborhood graphs. Fig. 12 shows that the different neighborhood types form only two distinct polyhedron shapes when we represent the connectivity graph. The polyhedra are made up of triangles.

**Proposition 3.11.** *If the neighborhood in  $E$  of an element  $s$  of  $O$  can be represented as a polyhedron made up with triangles, then  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1 \Leftrightarrow s$  is simple.*

**Proof.** We suppose  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ . We have seen that we only have to show that conditions (11) and (13) are true. We will prove it for condition (11), the demonstration for Eq. (13) is similar. We will need the following lemma:

**Lemma 3.12.** *If  $s \in E$  is an element whose neighborhood in  $E$  can be represented by a polyhedron made up with triangles and such that  $N_{CC}(N_O(s) - \{s\}) = N_{CC}(N_{\bar{O}}(s)) = 1$ , we have  $\forall \omega \in \Omega_{N_O(s)}$ ,  $\omega$  is reducible in  $N_{O-\{s\}}(s)$ .*

**Proof.** We will use a geometrical approach that uses recurrence on the number  $n$  of elements of  $N_{\bar{O}}(s)$ .

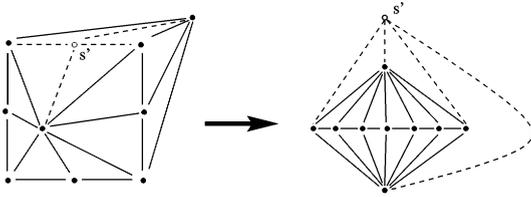


Fig. 13. Planar representation of a neighborhood.

If  $n = 1$ , there exists only one element  $s'$  of  $\bar{O}$  in  $N_E(s)$ . Since  $N_E(s)$  can be represented as a polyhedron and since  $s'$  is a vertex of the polyhedron, the other vertices are elements of  $O$  and can be represented as a planar structure made up with triangles. In this structure only the external vertices are connected to an element of  $\bar{O}$  (Fig. 13). This structure represents the elements of  $N_{O-\{s\}}(s)$  and is made up with triangles that represent triplets of elements, any pair of which being neighbors. Since an elementary deformation of a loop is based on such a triplet of elements, we can deduce that any loop of  $N_{O-\{s\}}(s)$  is irreducible.

Let us suppose that the lemma is true for  $n = N$  where  $N \geq 2$ . To increment  $n$  by one, we must move an element from  $N_{O-\{s\}}(s)$  to  $N_{\bar{O}}(s)$ . On the planar representation it is done by removing a point on the boundary of the structure because only the external points are connected to  $N_{\bar{O}}(s)$ . Thus, we do not change the properties of the planar structure and therefore any loop of  $N_{O-\{s\}}(s)$  is reducible.

To prove Proposition 3.11, we will suppose that Eq. (11) is false:  $\exists \omega \in \Omega_{O-\{s\}}$  such that  $\omega$  is irreducible in  $O - \{s\}$  and reducible in  $O$ . Therefore,  $\omega$  is reducible in  $O \Rightarrow \exists \omega' \in [\omega]_{O-\{s\}} \cap \Omega_{N_{O-\{s\}}(s)}$ . According to the lemma,  $\omega'$  is reducible in  $N_{O-\{s\}}(s)$ , therefore  $\omega$  is reducible in  $O - \{s\}$ , which contradicts the supposition.  $\square$

#### 4. Homotopic deformation of several objects

The definition of simple elements and the local characterization can be extended to a scene composed of more than two objects.

##### 4.1. Simple deformations and homotopy sets

Given a graph  $G = (E, V)$  we define an object as an element of a partition  $P$  of  $E$ . Thus,  $P$  represents a complete scene. We define the deformation of a scene as a one-to-one mapping between two partitions of  $E$ . In this case, the “simplest” deformation consists in moving an element from an object  $A$  to an object  $B$ . Such an elementary deformation is called an  $(A, B)$ -deformation. An  $(A, B)$ -deformation is said simple if it does not change the topology of neither  $A$  nor  $B$ , it is said  $A$ -simple if it

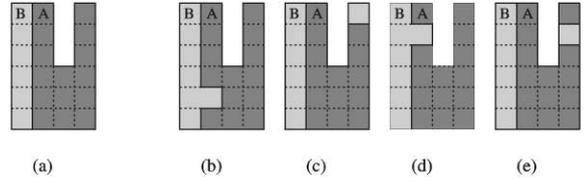


Fig. 14. Different types of  $(A, B)$ -deformations. (a) Objects  $A$  and  $B$ . (b) Simple deformation. (c)  $A$ -simple deformation. (d)  $B$ -simple deformation. (e) Non-simple deformation. (For simplifying the illustrations, we use pixel-based representation and, in configuration where no topological ambiguities occurs, we assume 8-connectivity for all objects.)

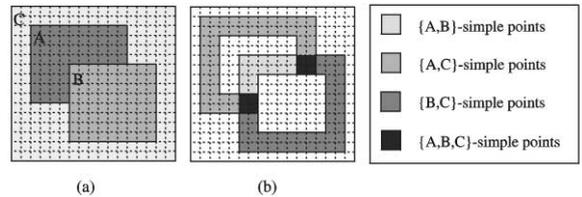


Fig. 15. Simple points depend on several objects of a scene. (a) A scene with three objects.  $A$  and  $B$  are considered with 4-connectivity (just here for sake of illustration),  $C$  is considered to be 8-connected. (b) The simple points of the scene.

changes only the topology of  $B$ , it is said  $B$ -simple if it changes only the topology of  $A$  and it is said not simple if it does not preserve the topology of neither  $A$  nor  $B$  (Fig. 14).

In order to be homotopic, an  $(A, B)$ -deformation must preserve the topology of all the objects. Therefore, the set of all simple  $(A, B)$ -deformations for all  $A$  and  $B$  represent all the elementary homotopic deformations. The set of all homotopic elementary deformations that can be applied to an element of  $E$  can be used to define simple elements in a scene composed of several objects:

**Definition 4.1.** Given a graph  $G = (E, V)$ , a partition  $P$  of  $E$  and a non-empty subset  $Q$  of  $P$ , an element  $e \in E$  is said  $Q$ -simple if for all  $(A, B) \in Q \times Q$ , any  $(A, B)$ -deformation of  $e$  is simple (Fig. 15).

Since  $Q$  is a subset of a partition, it represents a set of objects. In other words, Definition 4.1 says that an element of  $E$  is  $Q$ -simple if changing its membership to any object of  $Q$  does not change the topology of the scene. An immediate consequence of this definition is that union and intersection of sets of objects preserve “simplicity”:

**Proposition 4.2.**

$\forall e \in E, \forall Q \subset P, \forall R \subset P$ , if  $e$  is  $Q$ -simple and  $e$  is  $R$ -simple then  $e$  is  $(Q \cup R)$ -simple and  $e$  is  $(Q \cap R)$ -simple. (16)

The union of all sets  $Q \subset P$  (i.e. the maximal set) such that  $e$  is  $Q$ -simple is called the *homotopy set* of  $e$  and is denoted  $H_e$ . This set completely defines the homotopic properties of an element according to the entire scene because it contains exactly all objects in which  $e$  can be deformed without changing the topology of the scene:

**Proposition 4.3.** *Given a graph  $G = (E, V)$ , a partition  $P$  of  $E$  and two elements  $A$  and  $B$  of  $P$ , an  $(A, B)$ -deformation  $d$  of an element  $e \in A$  is simple if and only if  $B$  belongs to  $H_e$ .*

**Proof.** If  $d$  is simple, then  $e$  is  $\{A, B\}$ -simple and therefore  $B$  is in  $H_e$ .

Conversely, let us assume that  $B \in H_e$ . Since  $d$  is an  $(A, B)$ -deformation, then  $e \in A$  before the deformation because an  $(A, B)$ -deformation moves an element from  $A$  to  $B$ . An  $(A, A)$ -deformation does not change the scene and therefore preserves the topology. As a consequence,  $H_e$  contains  $A$ . Thus  $A$  and  $B$  belong to  $H_e$ . Therefore, by Definition 4.1,  $d$  is simple because  $e$  is  $H_e$ -simple.  $\square$

The homotopy set of an element contains all the objects to which the element can belong without changing the topology of the scene. Therefore, it contains at least the object to which the element belongs in the scene. If it contains no other object, no elementary homotopic deformation can be applied to the element. On the contrary, if it contains more than one object, the element can be moved to any of these objects without changing the topology of the scene. Thus, the homotopy set of an element can be used to find all the elementary homotopic deformations of this element. Therefore, the homotopy sets of all elements define the set of elementary homotopic deformations of a scene. As for simple elements in scenes composed of two objects, the modification of a scene changes its homotopy properties and therefore can modify the homotopy sets of the scene. Thus, it is not generally possible to modify two elements in parallel. A global homotopic deformation of a scene must be done by iteratively applying elementary homotopic deformations. Two scenes are topologically equivalent if there exists a series of elementary simple deformations transforming one of the two scenes into the other. Thus, simple  $(A, B)$ -deformations are the basis of topological equivalence. Since there is an equivalence between the set of all simple  $(A, B)$ -deformations and homotopy sets (Proposition 4.3), both can be used when dealing with homotopic deformations. The homotopy sets are more general than  $(A, B)$ -deformations since the homotopy sets of an element  $e$  makes it possible to find all the simple deformations of  $e$  but the contrary is not true.

In the binary case, it is possible to locally characterize simple elements. The simple elements correspond to the elementary simple deformations of a scene having two

objects. We show in the following section that it is possible to extend the local characterization of simple elements in order to locally characterize the elementary homotopic deformations and the homotopy sets of a scene.

#### 4.2. Local characterization of the homotopy set

In this section, we consider the local characterization of simple elements in the case of two objects presented in Section 3.3 and we apply it to each object of a scene composed of any number of objects in order to obtain a local characterization of the homotopy sets of the scene.

We defined topology preserving deformations for an object  $O$  embedded in a background  $\bar{O}$ . If we consider  $O$  as an element of a partition  $P$  of  $E$  and  $\bar{O}$  as the complementary of  $O$  in  $E$  (i.e. the union of the elements of all other objects of the partition), we obtain directly a definition of homotopy for a unique object of a partition. In the general case, an elementary deformation does not only modify one object but it cannot modify the topology of more than two objects. An  $(A, B)$ -deformation can only change the topology of  $A$  or  $B$  because any other object  $C$  remains unchanged and its background  $\bar{C}$  on  $E$  is also unchanged by this transformation because all the elements of  $A$  and  $B$  belong to  $\bar{C}$ , thus moving an element from  $A$  to  $B$  does not change the elements of  $\bar{C}$ . Therefore, it is possible to locally characterize simple  $(A, B)$ -deformations by checking topology preservation for both  $A$  and  $B$ :

**Proposition 4.4.** *An  $(A, B)$ -deformation  $d$  of an element  $e$  is simple if and only if  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$ .*

**Proof.** If  $d$  is simple, it does not change neither the number of connected components nor the number of tunnels of  $A$ . Therefore  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = 1$  (Proposition 3.11). It is the same for  $B$ , therefore  $N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$ .

The reciprocal is also a direct consequence of Proposition 3.11. If  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$ , then the number of connected components and the number of tunnels are preserved for a modification between  $A$  and  $\bar{A}$  and between  $B$  and  $\bar{B}$ . Since  $A \subset \bar{B}$ ,  $B \subset \bar{A}$  and  $d$  is a deformation between  $A$  and  $B$ , therefore  $d$  is simple because it preserves the topology of all objects of the scene.  $\square$

With Proposition 4.4, it is possible to locally characterize the homotopy set of each element of a scene:

**Proposition 4.5.** *Given a graph  $G = (E, V)$ , a partition  $P$  of  $E$  and an element  $e$  of  $E$ , we have  $\forall B \in P, B \in H_e$  if and*

only if  $e \in B$  or  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$  where  $A$  denotes the element of  $P$  such that  $e \in A$ .

**Proof.** If  $B \in H_e$  either  $e \in B$  then the proposition is verified, or  $e \notin B$ ; in that case any  $(A, B)$ -deformation is simple (Proposition 4.3). Therefore, according to Proposition 4.4,  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$ .

We showed in the proof of Proposition 4.3 that  $H_e$  contains the object which contains  $e$ . Thus, if  $e \in B$  then  $B \in H_e$ . If  $N_{CC}(N_A(e)) = N_{CC}(N_{\bar{A}}(e)) = N_{CC}(N_B(e)) = N_{CC}(N_{\bar{B}}(e)) = 1$  then any  $(A, B)$ -deformation of  $e$  is simple (Proposition 4.4), therefore  $B \in H_e$  (Proposition 4.3).  $\square$

Local characterization of simple elements in the case of a scene composed of several objects is not more complicated than in the binary case. The only difference comes from the possibility for an element to belong to several objects after a modification. It is therefore necessary, for each modification from an object  $A$  to an object  $B$ , to locally check topology preservation of both  $A$  and  $B$ .

The local properties of the adjacency graph of the cellular model make it possible to use local characterization of homotopy sets (see Section 3.3). It is therefore possible to homotopically deform any scene that can be modeled by a cellular model. In the following section we present a cellular implementation method which is optimized for both size of the data and speed of the algorithms using heavily the adjacency graph such as homotopic deformation algorithms.

### 5. Cellular model implementation

A cellular model  $M = (E, V, P)$  can be separated in, at least, two distinct parts. The first part represents the model structure and is composed of the set of cells  $E$  and the adjacency relationship  $V$ . The second part contains  $P$  and represents a scene embedded in the structure  $(E, V)$ . Thus, we obtain a framework similar to classical images. The structure of an image is composed of a set of elements (pixels or voxels for example) provided with one adjacency relationship. The image data represent the other part of the framework. This similarity is not just theoretical, it makes it possible to build a computer implementation which is similar to many classical images implementations.

#### 5.1. Implementation of the cellular model structure

To implement the structure of the cellular model, we first order the set of cells by choosing a one-to-one mapping between the set of cells and  $\{0, 1, \dots, N - 1\}$

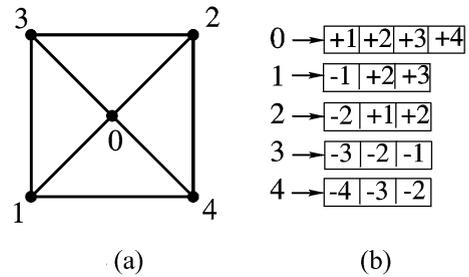


Fig. 16. Possible implementation of an adjacency graph. (a) The elements are numbered. (b) Each element is provided with a set of index offsets. Each offset represents a neighboring element.

where  $N$  is the number of cells. Once ordered, each cell  $s$  of the set can be identified by its index  $i_s$ . In this case given a cell  $s$  as a reference cell, any cell  $s'$  of the set can be uniquely identified by  $i_{s'} - i_s$ . Therefore, the set of cells which are neighbors of  $s$  can be expressed by a set of index offsets. This gives a possibility to represent the adjacency graph of the cellular model by providing each cell with a set of index offsets (Fig. 16). In this case, the amount of data needed to represent the graph is important and grows with the number of cells. But it is possible to find an order which allows to use the same structure with only a small fixed size data structure.

We can order the cells with the aim of having only a few different sets of index offsets for all cell neighborhoods. Thus we only need to have these few sets of neighbors in memory and to link each cell to the appropriate one. An appropriate order can be found by isolating a group of cells which can be used as a basis to build the whole cellular model. Fig. 17a shows such a group of cells (called *basic cells*). It is possible to build a cellular model by putting a group of basic cells on each node of the cubic grid (Fig. 17b). This structure gives an easy way for choosing an order for all the cells. First, the groups of basic cells are sorted. Since they are on a cubic grid we can use a classical order based on coordinates: a group of coordinates  $(x, y, z)$  in a cubic grid of size  $S_x \times S_y \times S_z$  is given the index  $g = x + y.S_x + z.S_x.S_y$ . Then, the basic cells in every group are sorted with a unique order. We use the order presented in Fig. 17c. With the group indices and the basic cell index in its group, we can give a unique index to every cell of the cellular model. Since all groups have exactly eight elements, a cell  $s$  whose group index is  $g$  and index in the group is  $t$ , can be given the index  $i_s = 8.g + t$ .

With this order, the number of different set of index offsets needed to represent neighborhoods is reduced to eight. Therefore, we only need to have these eight neighborhood sets in memory instead of one set for each cell. Moreover, all cells which have the same index in the group of basic cells have also the same neighborhood set. In other words, there is a one-to-one mapping between

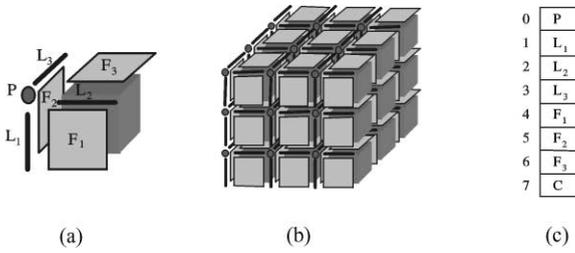


Fig. 17. (a) Group of basic cells used to build a cellular model. (b) Cellular model obtained by repeating the group of (a) in a  $3 \times 3 \times 3$  cubic grid. (c) Order of the basic cells.

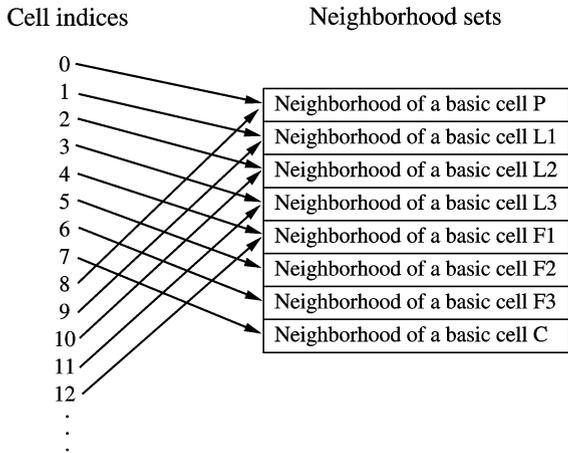


Fig. 18. Implementation of the cellular model graph structure. Only eight neighborhood sets are in memory.

the basic cells and the neighborhood sets. Therefore, we can compute very easily the link between a cell  $s$  and its neighborhood set with only the index  $i_s$ . The index in the basic cells group of  $s$  is  $i_s$  modulo eight. This group index can be used directly to find the appropriate neighborhood set (Fig. 18). Thus, implementation of the adjacency graph of the cellular model uses a very small fixed size amount of memory. Moreover, algorithms using this implementation use only a low computation time to go through the adjacency graph. Given a cell of index  $i_s$ , only a few basic operations are needed to find the index of one of its neighbors: a modulo operation to find the appropriate neighborhood set and an addition to obtain the index of the neighbor. Therefore, this implementation is optimized according to both data size and algorithms speed.

Since the graph structure of the cellular model uses very few memory, the memory usage of a cellular model is determined by the data associated with the cells to represent the scene. The best way to represent a partition is to use a set of labels and to associate each cell with one label of the set. Therefore, the memory size of the cellular

model is directly proportional to the number of cells. This is again similar to classical images, the size of which depends on the number of voxels. However, if we compare classical images and cellular models, it appears that a cellular model uses more memory. For example, if we build a cellular model to represent the same scene as an image, we must use a cube for each voxel of the image and represent the cells of lower dimension. Since we use the group of eight cells of Fig. 17a as an “atom” for building the cellular model, the size of the cellular model is eight times the size of the image.

The implementation we propose is very close to a possible implementation of classical images. The main difference is that the neighborhood set of a voxel depends on the label of the voxel. However, it is possible to use an object-oriented language such as C++ in order to use the same code for both classical images and cellular models when the algorithm is mainly based on the adjacency graph. For example, we implemented a unique function for finding the connected components of an object in a cellular model or in a classical image using a two pass chamfer method. But it is also possible to adapt more sophisticated algorithms which use the geometrical properties of the images. For example, we defined a chamfer distance between the cells of a cellular model which allows the computation of distance maps and geodesic distances on cellular model objects.

## 5.2. Cellular homotopic deformation algorithms

### 5.2.1. Simple cell detection

Checking if a cell  $s$  is simple or not can be done by counting connected components on a graph. The nodes of the graph represent the cells which are in the neighborhood of  $s$  and the arcs are the adjacency relationships between these cells. Thus, the graph depends on the structure of the neighborhood of  $s$ ; our cellular model implementation uses eight different neighborhoods (each one represents the neighborhood of a basic cell), therefore there are eight different neighborhood graph implementations. These graphs are intensively used by the cellular deformation algorithm, it is thus necessary to build these eight graphs when the cellular model is created because their structures depend only on the cellular model structure. To build these graphs we have to associate to each neighbor  $v_i$  of a cell  $s$ , the set of all cells which are in the neighborhood of  $v_i$  and in the neighborhood of  $s$  (Fig. 19).

In order to check if a cell  $s$  is simple or not in the binary case, one just has to count the connected components of the object and of the background in the neighborhood of  $s$ .  $s$  is simple if and only if there are exactly two connected components: one for the object and one for the background. Since the neighborhood graph of each cell is fixed, the “simplicity” of a cell  $s$  depends only on the

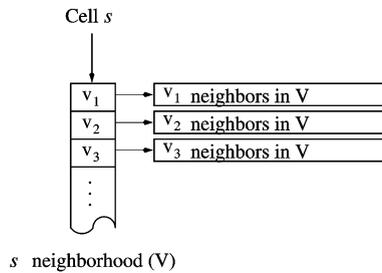


Fig. 19. Implementation of the neighborhood graph of each cell  $s$ .

membership to the object  $O$  of the neighbors of  $s$ . Therefore, the “simplicity” criteria is a binary function with binary parameters:

$$\text{simplicity}(s, O) = f(b_1, b_2, \dots, b_N). \quad (17)$$

In this equation, each binary value  $b_i$  represents the membership of one neighbor of  $s$  to the object  $O$ . If  $N$  is the number of neighbors of  $s$ , there is exactly  $2^N$  different combinations of  $f$  parameters. If  $N$  is small enough, it is possible to precalculate  $f$  in a boolean array of size  $2^N$ . Each index  $i$  of the array correspond to a configuration of  $f$  parameters ( $i = b_1 + 2.b_2 + 2^2.b_3 + \dots + 2^{N-1}.b_N$ ) and the value at index  $i$  contains the result of function  $f$ . It is therefore possible to check if a cell is simple by testing each neighbor of  $s$  only once. This optimization can be partially applied to cellular models. Indeed, the facets and the segments have 10 neighbors which implies arrays with 1024 elements to represent  $f$  which is completely tractable. Unfortunately, cubes and points have 26 neighbors which implies arrays with more than 67 millions of elements, the use of such huge arrays is not possible in practical image processing systems; in this case it is possible to use a mixed approach based on classical connected component algorithms for points and cubes, and on precalculated arrays for segments or facets. By using the mixed approach (1 h 0 min 50 s), we obtained about 20% better time performance as compared to a fully classical approach (1 h 16 min 35 s) in a complete cortex segmentation process (see Section 6). These execution times are obtained on a Sun Ultra Sparc 250 computer. In this process, the check of simple elements does not represent the most time consuming part of the algorithm (which is the cost function computation), therefore one can expect to gain more than 20% time performance with the mixed approach. Other optimization techniques, such as binary decision trees, can be considered for cubes and points in applications which require a very fast calculation of the simple cells.

In the case of a scene composed of more than two objects, we must compute the homotopy set of a cell  $s$  which relies on Proposition 4.5 and is also based on

connected components on the neighborhood graph of  $s$ . It is possible to use the method proposed for the binary case. Thus we obtain the following algorithm to compute the homotopy set  $H_s$  of a cell  $s$ :

```

A = object to which s belongs
Q = set of objects in the neighborhood of s
Hs = {A}
If simplicity(s, A)
  For each object O of Q
    If simplicity(s, O)
      Add O to Hs
    End If
  End For
End If

```

### 5.2.2. Deformation algorithm

In this section we introduce the main homotopic deformation algorithm of a binary cellular complex without constraint related to a specific application. This algorithm uses three main functions:

**initial\_model**: This function is used to build the initial model which is to be deformed. The topology of the initial model will be preserved during the deformations.

**selection\_criterion**: The choice of the cells which must be modified is done by this function. It has the responsibility to guide the deformations to the desired solution.

**stop\_criterion**: This criterion must decide if a cellular model should continue to be deformed or not.

$M$  is a cellular model

$P$  are the parameters of the algorithm

$M = \text{initial\_model}(P)$

**While** stop\_criterion( $M, P$ ) is false

**For** each cell  $s$  to  $M$

**If**  $s$  is simple and selection\_criterion( $s, M, P$ ) is true change the label of  $s$

**End If**

**End For**

**End While**

When this algorithm is implemented, it is important to note that it is possible to obtain different results with the same initial\_model( $P$ ), selection\_criterion( $s, M, P$ ) and stop\_criterion( $M, P$ ) functions. This is due to a well-known problem in classical image homotopic deformations: the order used to modify the simple cells can influence the result because the modification of a cell can change the “simplicity” of its neighbors. With the cellular model it is possible to limit this problem because the cells with the same dimension are never neighbors. Therefore, it is possible to modify all the cells of a given dimension in any order (even in parallel). Thus, if we modify the points, then the segments, then the facets, etc.,

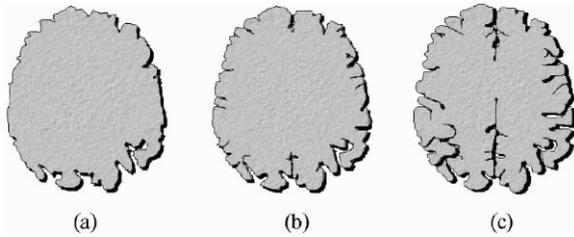


Fig. 20. Diagram of the segmentation process. (a) The model is initialized on the outer brain volume. (b) Intermediate step during the deformation process. (c) Final step: the external surface of the model has penetrated inside the folds. Black lines are the result of the segmentation superimposed on an image representing the membership to the cortex (only a part of one slice is shown, but this process is 3D).

there is no privileged direction due to the traversing order.

This deformation algorithm can be easily extended to a cellular model composed of several objects. We just have to choose, for each cell  $s$  considered by the algorithm, an object in the homotopy set of  $s$  and to make  $s$  belong to this object.

## 6. Application to cortex segmentation

In previous works [20,21], we used an homotopic cellular deformable model to segment the cerebral cortex in MR images of the brain. The cortex is a ribbon of gray matter which surrounds all other brain tissues. It has a very complex shape with many folds separated by both very thin (two-dimensional) and volumic structures. We showed that these structures can be accurately modeled by a cellular model. It is also well known that the cortex topology is almost spherical [22–25], therefore it is necessary to use homotopic deformations during the segmentation process to impose the topology of the results.

We used a cellular model initialized on the outer brain volume and deformed inside the external part of the cortex folds (Fig. 20). The deformation was done according to both volumic and surfacic constraints. In this process, the use of a cellular model allowed us to build an homotopic deformable model where local dimensions adapt to the cortical folds geometry: surfaces evolves in thin parts and volumes are deformed in volumic parts. The description of the cortex segmentation process is beyond the scope of this paper, further information can be found in Refs. [13,26]. Fig. 21 shows a detail of a result obtained from a three-dimensional MRI of the head, the surface of the resulting cellular model is superimposed on a membership to the cortex image. It shows that both volumic and surfacic structures have been deformed in the brain.

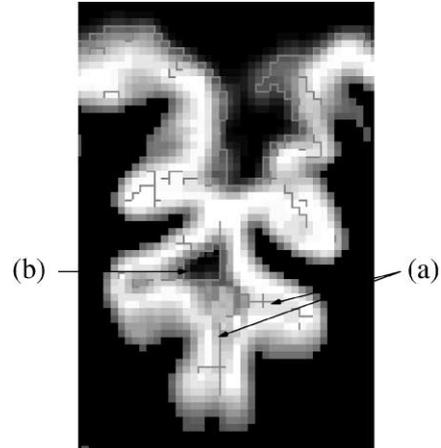


Fig. 21. Cortex segmentation process. Surfacing parts are deformed where the cortex folds are very tight (a) and volumic structures are deformed where there is enough space (b).

## 7. Conclusion

In this paper we proposed an original three-dimensional model to deal with complex scenes composed of several objects having complex shapes and different local dimensions. This cellular model is based on a cellular complex structure embedded in the cubic grid. The cellular structure allows to extend the modeling possibilities of classical images and the embedding in the cubic grid makes the cellular model geometrically close to classical images. It is therefore possible to use a cellular model to do image processing such as segmentation of complex objects with different local dimensions. For this purpose we showed that a cellular model can be used as an homotopic deformable model.

In order to define homotopic deformation of a cellular model, we introduced a new definition of topology preserving deformations which is only based on an adjacency graph structure. This definition generalizes those proposed in digital picture framework because it does not rely on a particular geometrical structure such as the cubic grid. With our definition, any geometrical structure provided with an adjacency graph and verifying some local conditions can identify simple elements. Moreover, we extended the notion of topology preserving deformation to the case of models composed of more than only one object and one background. To do this we introduced two new notions: simple elementary deformation and homotopy set. Then we showed that these two notions can be locally characterized and therefore used in an homotopic deformation process.

Then, we proposed an implementation of the cellular model which is optimized for both memory usage and speed. Therefore, the cellular model can be used in any

application considering objects with complex geometry such as cortex segmentation from MR image.

## References

- [1] V.A. Kovalevsky, Topological foundations of shape analysis, NATO ASI Ser., Ser. F 126 (1994) 21–36.
- [2] T. Pavlidis, Structural Pattern Recognition, Springer, New York, USA, 1977.
- [3] L. Latecki, Multicolor well-composed pictures, Pattern Recognition Lett. 16 (1995) 425–431.
- [4] Y. Cointepas, I. Bloch, L. Garnero, A discrete homotopic deformable model dealing with objects with different local dimensions, Discrete Geometry for Computer Imaging, Noisy-le-Grand, France, March 1999, pp. 258–271.
- [5] G. Bertrand, M. Couprie, Some structural properties of discrete surfaces. Proceedings of DGCI'97, Lecture Notes in Computer Science, vol. 1347, Montpellier, France, December 1997, pp. 113–124.
- [6] E. Artzy, G. Frieder, G.T. Herman, The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm, Comput. Vision Graphics Image Process.: Graphical Models Image Process. 15 (1981) 1–24.
- [7] G.T. Herman, Discrete multidimensional Jordan surfaces, Comput. Vision Graphics Image Process.: Graphical Models Image Process. 54 (6) (1992) 507–515.
- [8] T.Y. Kong, A. Rozenfeld, Digital topology: introduction and survey, Comput. Vision Graphics Image Process. 48 (1989) 357–393.
- [9] V.A. Kovalevsky, Discrete topology and contour definition, Pattern Recognition Lett. 2 (5) (1984) 281–288.
- [10] V.A. Kovalevsky, Finite topology as applied to image analysis, Comput. Vision Graphics Image Process. 46 (1989) 141–146.
- [11] C. Godbillon, *Éléments de topologie algébrique*, Hermann, Paris, 1971.
- [12] J.R. Munkers, *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park, CA, 1984.
- [13] Y. Cointepas, I. Bloch, L. Garnero, A Precise Segmentation of the Cerebral Cortex from 3D MRI using Cellular Model and Homotopic Deformations, Advanced Signal Processing for Medical Magnetic Resonance Imaging and Spectroscopy, Hong Yan (Chapter 10), in press.
- [14] D.G. Morgenthaler, Three dimensional simple points: serial erosion, parallel thinning, and skeletonization, Technical Report TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, 1981.
- [15] P.K. Saha, B. Chanda, D.D. Majumder, Principles and algorithms for 2-D and 3-D shrinking, Technical Report TR/KBCS/2/91, Indian Statistical Institute, Calcutta, India, 1991.
- [16] Y.F. Tsao, K.S. Fu, A parallel thinning algorithm for 3-D pictures, Comput. Vision Graphics Image Process. 17 (1981) 315–331.
- [17] G. Bertrand, G. Malandain, A new characterization of three-dimensional simple points, Pattern Recognition Lett. 15 (1994) 169–175.
- [18] G. Bertrand, A boolean characterization of 3-D simple points, Pattern Recognition Lett. 17 (2) (1996) 115–124.
- [19] P.K. Saha, B.B. Chaudhuri, Detection of 3-D simple points for topology preserving transformations with application to thinning, IEEE Trans. Pattern Anal. Mach. Intell. 16 (10) (1994) 1028–1032.
- [20] Y. Cointepas, I. Bloch, L. Garnero, Cellular complexes: a tool for 3D homotopic segmentation in brain images, Proceedings of IICIP'98, Vol. 3, Chicago, USA, 1998, pp. 832–836.
- [21] Y. Cointepas, I. Bloch, L. Garnero, Joined segmentation of cortical surface and brain volume in MRI using a homotopic deformable cellular model, 3-D Digital Imaging and Modeling, Ottawa, Canada, October 1999, pp. 240–248.
- [22] C. Davatzikos, Using a deformable surface model to obtain a shape representation of the cortex, IEEE Trans. Med. Imaging 15 (6) (1996) 785–795.
- [23] P.C. Teo, G. Sapiro, B.A. Wandell, Creating connected representations of cortical gray matter for functional MRI visualization, IEEE Trans. Med. Imaging 16 (6) (1997) 852–863.
- [24] J.-F. Mangin, O. Coulon, V. Frouin, Robust brain segmentation using histogram scale-space analysis and mathematical morphology, Medical Image Computing and Computer Assisted Intervention (MICCAI'98), Lecture Notes in Computer Science, Cambridge, USA, Springer, Berlin, October 1998, pp. 1230–1241.
- [25] G. Le Goualher, E. Procyk, D.L. Collins, R. Venugopal, C. Barillot, A.C. Evans, Automated extraction and variability analysis of sulcal neuroanatomy, IEEE Trans. Med. Imaging 18 (3) (1999) 206–217.
- [26] Y. Cointepas, Modélisation topologique et segmentation tridimensionnelles du cortex cérébral partir d'IRM pour la résolution des problèmes directs et inverses en EEG et en MEG, Ph.D. Thesis, Ecole Nationale Supérieure des Télécommunications, October 1999. (ENST 99 E 25).

**About the Author**—YANN COINTEPAS received Ph.D. in “Image and Signal Processing” from ENST Paris (Signal and Image Processing Department) in 1999. He received a DEA (Diploma of Advanced Studies, equivalent to postgraduate qualification) of “Image and Signal Processing” in 1996 and a DEA of “Cognitive Science” in 1994. He has a four-year university degree (Maîtrise) in Computer Science. His research interests include discrete 3D modeling, discrete geometry, multimodality medical imaging, neuroscience and computer science.

**About the Author**—ISABELLE BLOCH is a Professor at ENST (Signal and Image Processing Department), where she is responsible for the Image Processing and Interpretation Group. She graduated from Ecole des Mines de Paris in 1986, received Ph.D. from ENST Paris in 1990, and the “Habilitation à Diriger des Recherches” from University Paris 5 in 1995. Her research interests include 3D image and object processing, 3D and fuzzy mathematical morphology, discrete 3D geometry and topology, decision theory, information fusion in image processing, fuzzy set theory, evidence theory, structural pattern recognition, spatial reasoning and medical imaging.

**About the Author**—LINE GARNERO was born in France, on September 4, 1955. She received the Doctorat de 3ème Cycle and the Doctorat d'Etat from the University of Paris XI (Orsay) in 1981 and 1987, respectively. She is Chargé de Recherche at the CNRS (National Center of Scientific Research). She worked at the Institute of Optics in Orsay from 1981 to 1996. Her research field was the image reconstruction techniques for microwave, X-ray or g tomography. She works now in the laboratory of Cognitive Neurosciences and Cerebral Imaging in La Salpêtrière Hospital, Paris, where her main research concerns the reconstruction of brain electrical activity from electroencephalographic (EEG) or magnetoencephalographic (MEG) data, and the fusion IRMf/EEG/MEG.