Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 1081–1093
www.elsevier.com/locate/imavis

Topological triangle characterization with application to object detection from images

L.G. Nonato [1], M.A.S. Lizier, J. Batista, M.C.F. de Oliveira, A. Castelo

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-Carlense 400, C.P. 668, 13560-970 São Carlos, SP, Brazil

Received 6 February 2004; received in revised form 28 September 2006; accepted 27 November 2007

Abstract

A novel mathematical framework inspired on Morse Theory for topological triangle characterization in 2D meshes is introduced that is useful for applications involving the creation of mesh models of objects whose geometry is not known *a priori*. The framework guarantees a precise control of topological changes introduced as a result of triangle insertion/removal operations and enables the definition of intuitive high-level operators for managing the mesh while keeping its topological integrity. An application is described in the implementation of an innovative approach for the detection of 2D objects from images that integrates the topological control enabled by geometric modeling with traditional image processing techniques.

Keywords: Object detection; Object modeling from images; Topological triangle characterization; Morse operators; 2D triangular meshes

1. Introduction

A theory of topology for digital spaces [10,22] is the basis of a range of techniques and tools for image segmentation, feature extraction and object detection from images. In this context, topological pixel characterization, i.e., determining the exact situations in which pixels may be removed or inserted from an object representation without changing its topology, plays a fundamental role. Topological tools are also common in geometric modeling, particularly in the construction of two- and three-dimensional mesh approximations of "real" objects. These piecewise linear approximations are typically described in terms of primitive cells (usually triangles in the 2D case, or tetrahedrons in 3D). Topological cell characterization is also essential in this context, because it provides – as in the digital case – the ability to control the topology of the object being modeled.

Mesh models traditionally used in geometric modeling and numerical simulation might be employed to represent image features in terms of triangle primitives, rather than pixels [25]. Such an approach could, for example, enable alternative implementations of image processing operations such as morphing, warping, and detecting and modeling objects from images. An important aspect in this latter case is that the topological control offered by the geometric modeling representations allows more robust object modeling. As precise geometric information about the target models is not easily obtainable from the image, the topological information available can assist in the generation of better approximation models.

The contribution of this work fits exactly into the above scenario: we describe a topological cell characterization that, as previously mentioned, is fundamental to applications in both image segmentation and geometric modeling. Inspired on Morse Theory [18], a novel framework for 2D mesh manipulation is derived. This framework guarantees a robust control of the mesh topology as triangles are inserted/removed and enables the definition of intuitive high-level operators for handling the mesh while keeping

E-mail address: gnonato@icmc.usp.br (L.G. Castelo).
[1] Tel.: +55 16 33739697.

0262-8856/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.imavis.2007.11.011

its topological integrity. Although we are particularly interested in the representation of image features – and thus restrict ourselves to 2D triangular meshes in this work – the mathematical background is useful to other applications involving mesh generation in which precise information on the geometry of the model to be created is not known *a priori*.

The proposed framework is illustrated with an application to a typical image segmentation problem, the detection of 2D objects from images. We show how the integration of techniques from geometric modeling, that operate on piecewise linear object representations, with techniques from image processing operating on digital spaces can improve the effectiveness of a simple procedure such as thresholding to extract meaningful information from images.

Section 2 presents some related works. In Section 3 we provide the basic concepts and notation required to follow the topological triangle characterization introduced in Section 4. From such a characterization, a set of operators is defined that handle all the possible topological changes that may occur as triangles are inserted/removed into a 2D mesh. These operators are then used to define topologically robust and intuitive high-level operations for triangle cutting and gluing. An implementation of the high-level operators, defined in Section 4, requires a suitable data structure for representing the mesh. We introduce such a data structure, called SHE (Singular Handle Edge) in Section 5. An application to object detection in gray scale images is described in Section 6. Final remarks are given in Section 7.

2. Related work

To place in context the potential applicability of the mathematical framework introduced in this paper we shall consider two classes of approaches adopted by techniques targeted at processing, analysis and feature extraction from images. The first class comprises techniques that make use of topology in digital spaces, such as those originated from digital topology [10] and mathematical morphology [22]. Such techniques benefit from topological concepts "adapted" to digital spaces to solve important problems such as skeletonization [16], connectivity [8] and computation of invariant properties [7]. An important issue in this context is the topological characterization of the basic image elements, i.e., pixels. Solutions to the previous problems, and to many others, either explicitly or implicitly employ such a characterization [19,14].

A second class of techniques for image manipulation exist whose conception, in contrast to the previous one, does not exploit topological concepts. To some of these, topology is not an ally but rather an obstacle to be overcome. A typical example is snake for image segmentation and shape extraction. Although highly satisfactory results may be obtained with this technique, it faces severe difficulties when applied to objects with complex topology [12]. Another technique that seeks for closed contours is shown in [1]. This edge-based segmentation process does not exhibit topology control, but ensures that a single, closed contour with no holes is obtained from an image. Traditional image processing and segmentation techniques based on edges or regions [3,11,4,23]; pattern recognition methods [24,5,13] and model-based such as Markovian Random Fields (MRF) and Fractals [6,26,20] do not exploit topological properties either.

Handling topology in the representation of unstructured meshes is a common problem in geometric modeling that has deserved attention from several authors. Typically, the problem is tackled with the definition of a topological data structure and a set of operators for manipulating such a structure. Some topological data structures became popular in the context of two-dimensional mesh generation and representation. The *winged-edge* by Baumgart [2], the *half-edge* by Mäntylä [17], and the *quad-edge* by Guibas and Stolfi [9] are well known examples that rely on topological operators to maintain mesh consistency (incidence and adjacency) and topological control. To manipulate the *quad-edge* data structure Guibas and Stolfi [9] present basically a single topological operator, the Splice. Baumgart [2] and Mäntylä [17] utilize a set of so-called Euler operators to modify the topological entities contained in the *winged-edge* and *half-edge* data structures, respectively.

To implement such an approach we define a topological triangle characterization and a set of topological operators that manage a novel data structure, called SHE – Singular Handle Edge. SHE is similar to the topological data structures described in the literature, except that it keeps an explicit representation of the object border components and explicitly represents singular vertices. The set of operators defined, named Morse operators, differs from existing ones in that its users do not need to be aware of low-level manipulations of vertices and edges, as operators were conceived to give users a high-level view of their functionality in terms of triangle manipulation. Triangle cut and glue operations formalized in terms of the Morse operators support a topologically robust management of the data structure and allow the generation of a mesh directly from images, with no previous pre-processing step, as illustrated in Section 6. The concept of Morse operators has already been employed in geometric modeling [15], however, without a rigorous theoretical framework as the one presented here.

3. Basic concepts

This section introduces the basic concepts and terminology used in the remainder of the text. Definitions and results presented in this and the following sections are restricted to two-dimensional Euclidean space.

A **k-dimensional simplex** or **k-simplex** in $\mathbb{R}^2$, $0 \leqslant k \leqslant 2$, is the convex hull of $k+1$ linearly independent points in $\mathbb{R}^2$ (a set of points $x_0, \ldots, x_k$ are linearly independent if so are the vectors $x_1 - x_0, \ldots, x_k - x_0$). A 0-simplex is called a vertex, a 1-simplex is called an edge and a 2-simplex is called a triangle. If σ is a k-simplex, $k = 1, 2$, then any j-simplex, $0 \leqslant j < k$, contained in σ is called a face of σ.

The interior of a simplex σ is the set of points in σ that do not belong to a face of σ.

A **simplicial complex** K is a finite collection of simplices satisfying:

(1) If $\sigma \in K$, then all faces of σ belong to K.
(2) If $\sigma, \tau \in K$, then either $\sigma \cap \tau = \emptyset$ or $\sigma \cap \tau$ is a common face of σ and τ.

The dimension of K, denoted dim K, is -1 if $K = \emptyset$, or the maximum value among the dimensions of the simplices for $K \neq \emptyset$.

A **triangulation** T in $\mathbb{R}^2$ (2D triangulation) is a simplicial complex of dimension 2 where all j-simplex in T, $j = 0, 1$, are contained in at least one 2-simplex of T. The definition of triangulation presented here is quite simple, but sufficient for the development of the work. The subset $|T| = \bigcup_{\tau \in T} \tau$ is called the **underlying space** of the triangulation T. A triangulation T is called **connected** if for any two points x and y in $|T|$, there is a continuous curve in $|T|$ connecting x to y. If $T = T_1 \cup T_2 \cup \ldots \cup T_m$, where each T_i, $i = 1, \ldots, m$, is a connected triangulation with $T_i \cap T_j = \emptyset$, then T is said **non-connected** and each T_i is called a **connected component** of T.

Each bounded and connected component of the subset $\mathbb{R}^2 - |T|$ is called a **hole** of the triangulation T. It is not difficult to see that the number of holes in a triangulation is finite (it may be zero), as the number of triangles is also finite and the subset $\mathbb{R}^2 - |T|$ has exactly one unbounded component.

An edge e (1-simplex) of a 2D triangulation T is an **interior edge** of T if e is shared by two triangles of T, otherwise, if e is contained in a single triangle of T, it is called a **boundary edge**. Let $P = (e_1, \ldots, e_n)$ be a sequence of distinct boundary edges of T such that e_i and e_{i+1}, $i = 1, \ldots, n$, $(e_{n+1} = e_1)$ have a vertex in common. P is said to be a **boundary curve** of T if, as one "walks" from edge e_i to e_{i+1}, the triangles containing e_i and e_{i+1} are always located to the left (or, alternatively, to the right) of e_i and e_{i+1} for every $i = 1, \ldots, n$. A boundary curve $P = (e_1, \ldots, e_n)$ is an **external boundary curve** of T if for each e_i in P, there is a real number $\epsilon > 0$ such that a disk centered in any interior point of e_i with radius ϵ does not intersect any hole of T. If P is a boundary curve and it bounds a hole of T then P is called an **internal boundary curve** of T.

The **star** of a vertex v in a triangulation T is the union of all simplices in T that contain v. The **link** of a vertex v is the union of all edges that do not contain v and are in the triangles forming the star of v. A vertex is **singular** if its link is not homeomorphic to a circle or to a segment.

Let $nv(T)$, $ne(T)$, $nt(T)$, $nc(T)$, and $nh(T)$ be the number of vertices, edges, triangles, connected components, and holes of a triangulation T, respectively. The Euler characteristic of T is given by:

$$\chi(T) = nv(T) - ne(T) + nt(T) = nc(T) - nh(T) \tag{1}$$

The Euler characteristic is a topological invariant that plays an important role in topology and has been extensively used in object classification [10].

4. Topological triangle characterization and Morse operators

Let T be a triangulation in $\mathbb{R}^2$. A triangle τ is called **adjacent** to T if $\tau \notin T$ and $\tau \cap T \neq \emptyset$. A triangle τ adjacent to a triangulation T may be defined as a:

(1) (-1)-**handle** if all edges of τ are in T (Fig. 1(a))
(2) 0-**handle** if either one vertex, or one edge, or two edges of τ are in T (Fig. 1(b))
(3) 1-**handle** if either two vertices, or one edge, and one vertex (opposite to the edge) of τ are in T (Fig. 1(c))
(4) 2-**handle** if three vertices of τ are in T (Fig. 1(d))

Lemma 1 shows how the addition of a k-handle alters the Euler characteristic of a triangulation. The proof is offered in such a way as to illustrate how topology can be controlled by the attachment of handles.

Lemma 1. *Let τ be a k-handle of a triangulation T. Then*

$$\chi(T \cup \tau) = \chi(T) - k \tag{2}$$

Proof. Let τ_v and τ_e be, respectively, the number of vertices and edges of τ that are not in T.

(1) If τ is a (-1)-handle then $\chi(T \cup \tau) = \chi(T) + \tau_v - \tau_e + 1 = \chi(T) + 0 - 0 + 1 = \chi(T) - (-1) = \chi(T) - k$, as depicted in Fig. 2. Attachment of a (-1)-handle always removes a hole from T, adding $+1$ to its Euler characteristic.

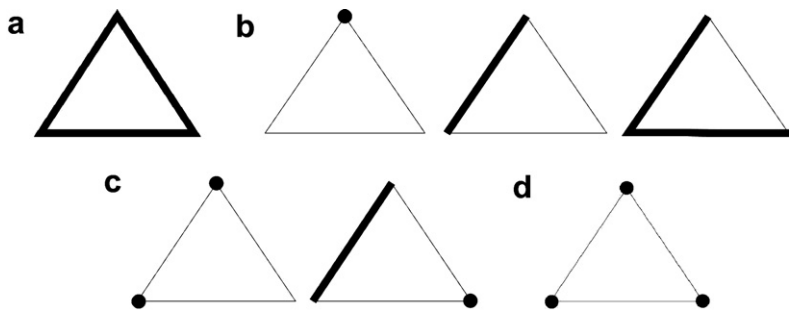

Fig. 1. (a) (-1)-handle; (b) 0-handle; (c) 1-handle; (d) 2-handle.

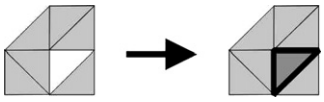

Fig. 2. Attaching a (-1)-handle to T.

(2) If τ is a 0-handle then $\chi(T \cup \tau) = \chi(T) + \tau_v - \tau_e + 1 = \chi(T) + 0 = \chi(T) - k$ for all possible kinds of 0-handles, as shown in Fig. 3. Thus, attachment of a (0)-handle keeps the Euler characteristic of T unchanged.

(3) If τ is a 1-handle then two cases must be considered: the 1-handle is either attached to a single boundary component introducing a new hole in T (Fig. 4(a) and (b)), or it is attached to different boundaries components, joining two connected components (as shown in Fig. 4(c) and (d)). In both cases the corresponding topological change is given by $\chi(T \cup \tau) = \chi(T) + \tau_v - \tau_e + 1 = \chi(T) - 1 = \chi(T) - k$.

(4) If τ is a 2-handle then three different types of attachments are possible, namely: (a) the three vertices are in a single boundary component; (b) two vertices are in a single boundary component and the third one is in a different connected component; (c) each vertex is in a distinct connected component. In the first case, the addition of τ introduces two new holes in T (Fig. 5(a)). In the second case, attaching τ introduces a new hole and removes a connected component from T (Fig. 5(b)). In the last case, two connected components are removed (Fig. 5(c)). In all such cases the Euler characteristic changes according to the following equation: $\chi(T \cup \tau) = \chi(T) + \tau_v - \tau_e + 1 = \chi(T) - 2 = \chi(T) - k$. □

As Lemma 1 contemplates all the topological changes that can be possibly introduced by the insertion of a new triangle to a triangulation, it allows a complete characterization of triangles regarding the topological changes introduced by their insertion or removal. Morse operators are defined in a straightforward manner based on the results of this lemma. They are grouped into four different categories according to the topological change introduced in the triangulation. The following symbols are used to describe the possible actions of each operator over the topological entities they handle: M – make; K – kill; T – triangle; H – hole; S – singular vertex; E – edge identifying; and C – component. The four categories with their respective operators are:

- MO_{-1}={*MEEETKH*};
- MO_0={*MST*,*MET*,*MEET*};
- MO_1={*MTC*,*MSSTH*,*MSETH*,*MSSTKC*,*MSETKC*};
- MO_2={*MSSSTHH*,*MSSSTHKC*,*MSSSTKCC*}.

The symbols naming an operator provide a summary description of its action on the triangulation. For example, the symbols *MEET* naming an operator in MO_0 indicate that this operator introduces (i.e., makes) two edge identifying and a triangle; the symbols *MSSSTHKC* in MO_2 means that it makes three singular vertices, a triangle and a hole, and kills a component; and the symbol *MSETKC* in MO_1 means that it makes a singular vertex, an edge identifying, and a triangle, and kills a component.

Each category groups operators that introduce similar changes in mesh topology: different handle types result in different topological changes, corresponding to the four possible situations treated in Lemma 1. Thus, operators in category MO_{-1} attach (-1)-handles to the existing triangulation, those in MO_0 attach 0-handles, and so on. In gen-

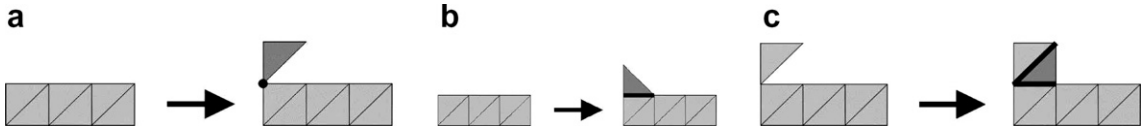

Fig. 3. Attaching a 0-handle to T. (a) By a vertex; (b) by one edge; (c) by two edges.

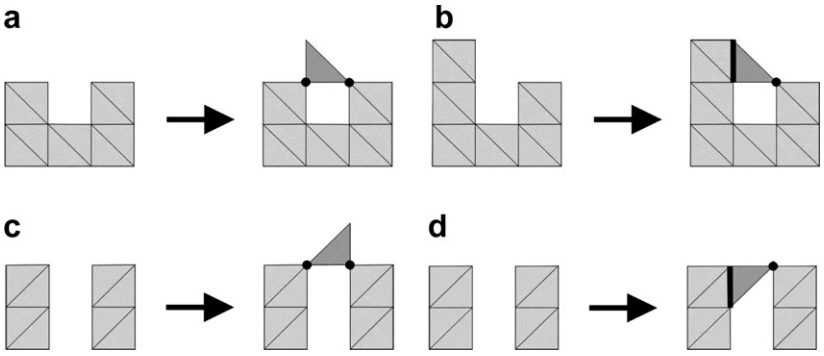

Fig. 4. (a and b) Attaching a 1-handle and generating a hole; (c and d) attaching a 1-handle between two different boundary components.

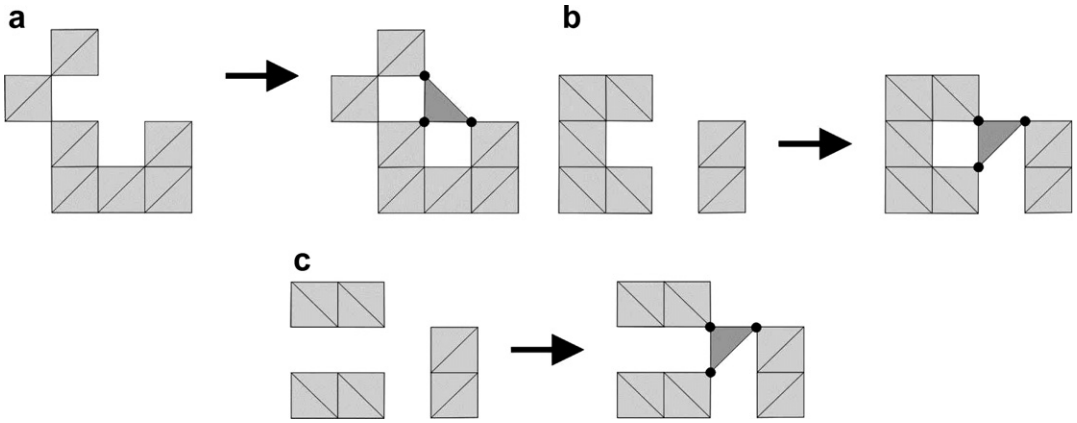

Fig. 5. Attaching a 2-handle to (a) a single boundary component; (b) two different boundary components; (c) three different boundary components.

eral, operators in category MO_k, $k = -1, 0, 1, 2$ add a k-handle to the triangulation, thus changing the mesh's Euler characteristic by an order k, i.e., after applying an MO_k operator the mesh's current Euler characteristic is changed from $\chi(T)$ to $\chi(T) - k$.

The above operators are called direct operators, as they may be used to add triangles to an existing mesh. Operators for removing triangles are described in terms of their inverse operators, which may be obtained by replacing letters M with Ks (and vice versa) in the above description. Inverse operators are also grouped into four categories according to the topological change they introduce. The four categories are:

- IMO_{-1}={*KTC*,*KSSTH*,*KSETH*,*KSSTMC*,*KSETMC*};
- IMO_{-2}={*KSSSTHH*,*KSSSTHMC*,*KSSSTMCC*};
- IMO_0={*KST*,*KET*,*KEET*};
- IMO_1={*KEEETMH*}.

Analogously to the direct operators, operators in category IMO_k, $k = -1, -2, 0, 1$ remove a k-handle from the triangulation, thus changing the mesh's Euler characteristic by an order k, i.e., after applying an IMO_k operator the mesh's current Euler characteristic is changed from $\chi(T)$ to $\chi(T) - k$.

Theoretically, not all operators are required to generate a triangulation with a specific topology. Actually, four of them are enough to generate triangulations of arbitrary topologies, as shown by the proof of Proposition 1 below. However, to build an arbitrary triangulation with only four operators (and their inverses) one must find an appropriate triangle insertion sequence, which may not be an easy task if the triangles comprising the whole triangulation are not known *a priori*, as it is common in several applications. Thus, the whole set of operators becomes necessary to better handle arbitrary topologies and geometries. Although it may not be useful from a practical point of view, the following proposition does offer a theoretical guarantee that only four of the Morse operators are sufficient to generate triangulations.

Proposition 1. *Only four Morse operators are enough to build any triangulation in* $\mathbb{R}^2$.

Proof. Let W be the subspace in $\mathbb{R}^5$ generated by $v - e + f - c + h = 0$. Note that any triangulation may be represented as a vector in W, the set of valid triangulations is a subset of W, and each Morse operator can also be described as a vector in W. For example, operator *MSSTH* may be represented by vector $(1, 3, 1, 0, 1)$, meaning that it introduces one new vertex, three new edges, one new triangle and one new hole in the triangulation.

Let $x_1 = (3, 3, 1, 1, 0)$, $x_2 = (1, 3, 1, 0, 1)$, $x_3 = (2, 3, 1, 0, 0)$, and $x_4 = (1, 2, 1, 0, 0)$ be the vector representations of operators *MTC*, *MSSTH*, *MST*, and *MET*, respectively. A straightforward computation shows that x_1, x_2, x_3, and x_4 are in W and are linearly independent. So, *MTC*, *MSSTH*, *MST*, and *MET* define a base to subspace W, thus generating any triangulation.

Note that other sets of four linearly independent operators could generate the topological triangulation classes. □

It is also worth pointing out that if singular vertices were not allowed, only four Morse operators would enable the description of triangle cut and glue operations. But, as shown in Proposition 2 bellow, singular vertices are required to handle general 2D meshes.

Proposition 2. *It is not possible to generate a triangulation with holes, by gluing triangles, without adding singular vertices to the triangulation during the process.*

Proof. Suppose, without any loss of generality, that T is a connected triangulation without holes constructed by gluing triangles, for example, by using operators *MTC*, *MET*, and *MEET*. To generate a hole in T it is necessary to attach a 1-handle to it. The only operators capable of doing that are *MSSTH* and *MSETH*, and both introduce singular vertices. The same occurs if two holes are added to the triangulation by the operator *MSSSTHH*. □

5. The Singular Handle-Edge data structure

The *Singular Handle-Edge* (SHE) data structure for representing two-dimensional meshes is tailored for manipulation by Morse Operators, as it can handle singular vertices and includes an explicit representation of the boundary curves of a triangulation. Although explicit representation of boundary curves is not common in topological data structures, it is essential for a correct implementation of the Morse operators. Moreover, it simplifies storage of the contouring conditions that typically appear in numerical simulations. Another favorable feature of the explicit representation of boundary curves is that it restricts the search for where to insert a triangle in a gluing operation, as new triangles are always attached to boundary curves. SHE is organized in terms of eight explicitly represented entities (or nodes), which are:

- **SHE** – Represents the whole mesh;
- **Mesh** – Represents each connected component in the mesh;
- **Vertex** – Represents the vertices in the mesh;
- **Half-edge** – Represents an edge contained in a face;
- **Sing** – Represents each singular component incident to a vertex;
- **Face** – Represents the faces or cells in the mesh;
- **Boundary-curve** – Represents the boundary curves in the mesh;
- **Boundary-edge** – Represents the boundary edges.

The entities Half-edge, Sing, and Boundary-edge are stored in circular dynamic lists, whilst the remaining entities are stored in non-circular lists. The relationships amongst the entities are shown in Fig. 6. SHE points to the list of Meshes and to the list of Vertices. Each Mesh points to the lists of its Faces and Boundary-curves (Fig. 6(a)). Each Vertex points to its Sing list (Fig. 6(b)). The Half-edge is always oriented and points to its tail vertex. The Half-edge node also points to the Half-edge in its adjacent face if it is not in a boundary, otherwise it points to a Boundary-edge entity (Fig. 6(c)). Each Sing points to a Half-edge that "leaves" its associated vertex. If the vertex is a singular one then it has an associated Sing circular list, and each element in this list points to a Half-edge in the boundary of each component forming the singularity (Fig. 6(d)). Each Face node points to the circular list of Half-edges that defines its orientation (Fig. 6(e)). The Boundary-curve points to a circular list of its Boundary-edges. Each Boundary-edge points to the Boundary-curve in which it is contained and to the Half-edge it represents (Fig. 6(f)).

5.1. Implementation

The data structure and a complete set of methods to access its stored information have been implemented in C++. Geometric data can also be modified through specific methods. Topological consistence is ensured with the use of the Morse operators to manipulate the structure, and the implementation enforces their use, so that mesh topology can only be manipulated through them. The operators were implemented as friend functions of the classes defining the data structure. A mechanism called **iterator** is offered to traverse the elements of SHE, which allows scanning the lists with a simple *FOR* loop. Methods defined for each class are responsible for initializing and finishing the *iterator*. For example, the following code could be employed to print the coordinates of all vertices in the mesh by traversing its Vertex list:

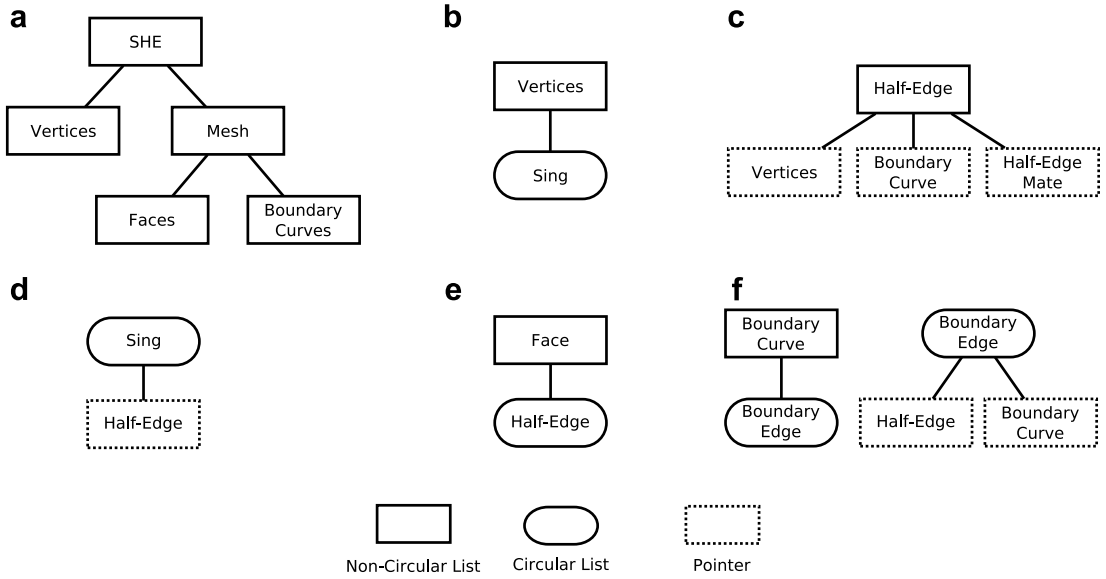

Fig. 6. Relationships amongst the entities constituting the SHE data structure.

```
Iterator<SHE_Vertex> iv;
for   (iv=s->Begin_vertex();   iv.NotFin-
ish(); ++iv)
  cout << iv->Get_x() << iv->Get_y();
```

where *s* is a pointer to a SHE object. The main advantage of the iterator is that it encapsulates the lists from the user and standardizes access to them.

5.2. Complexity

Due to the adjacency and incidence relations stored in SHE the time to obtain the neighborhood of a given topological element is proportional to the number of elements in that neighborhood. Such analysis is typically found in texts describing topological data structures [17]. In this section, we focus on the complexity analysis of the Morse operators on SHE.

Surely the most expensive operation involving the Morse operators is locate where to insert a new triangle. Without an auxiliary structure, the cost of attaching a new triangle to the mesh is proportional to the time spent traversing the boundary lists plus the time to modify the structure in the neighborhood of the region where the triangle is to be inserted. If M is the number of boundary edges and L is the cost of the structure's local update, then the total cost is $O(M+L)$ for each new triangle to be inserted. As SHE offers efficient access to the neighborhood of the elements, complexity of the Morse operators is dominated by the search in the boundary curves. A geometrical search tree [21] can be employed to improve the search process, so that attachment of a new triangle can be executed in $O(\log(M)+L)$.

For example, let us consider the cost of the local update for attaching the highlighted triangle to the mesh depicted in Fig. 7(a) with operator *MSSTH*. This operator will introduce a hole and a boundary curve to the triangulation. This new boundary curve can be constructed from the old one as follows: instanciate a new boundary curve, transfer the curve segment between the two singular vertices from the current boundary curve to the new one, and close both curves with the appropriate edges of the new triangle. As the SHE data structure keeps all the adjacency and incidence relationships, the transfer of the curve segment can be accomplished by straightforward pointer manipulation.

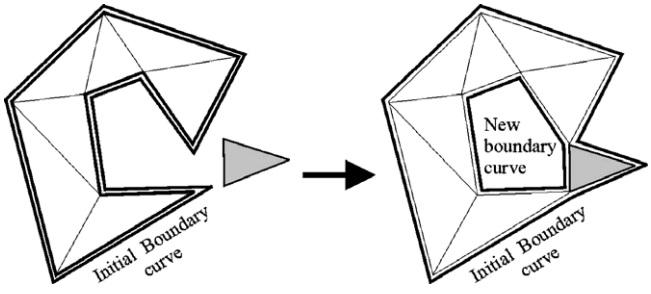

Fig. 7. Creating a new boundary curve.

The Sing entities associated to both singular vertices must also be updated.

6. Application to object detection in gray scale images

We illustrate how the topological Morse operators and its associated SHE data structure, which realize the topological characterization introduced in Section 4, provide a powerful tool to detect target objects and segment regions of interest from images. We shall make our point using the MRI gray level image in Fig. 8, which has intensity values in the range 0–255. This is handled with a software that creates a mesh model from a given image according to user input parameters that define a mesh resolution and a target threshold range. The mesh resolution specifies the coarseness of an imaginary grid that is superposed to the image, as depicted in Fig. 9(a) and (b). Resolution defines how many pixels are contained in a triangle, the primitive mesh element. In the grid depicted in Fig. 9(a), grid resolution is 48 pixels, implying that a triangle has an horizontal (vertical) edge length equal to 48 pixels. In Fig. 9(b) grid resolution is 24 pixels. The threshold range selected provides an initial approximation of the region of interest in the image.

From these two inputs, a mesh may be derived as follows. From each triangle in the imaginary grid it is determined whether one of its vertices corresponds to a pixel value within the user-defined threshold. If so, the corresponding triangle is added to the mesh triangulation by the appropriate Morse operator. The software automatically chooses the correct operator based on the geometric intersection between the triangle being inserted and the mesh, evaluating how this intersection relates with the topological entities. This is done for all grid triangles. Using the Morse operators to construct the mesh ensures

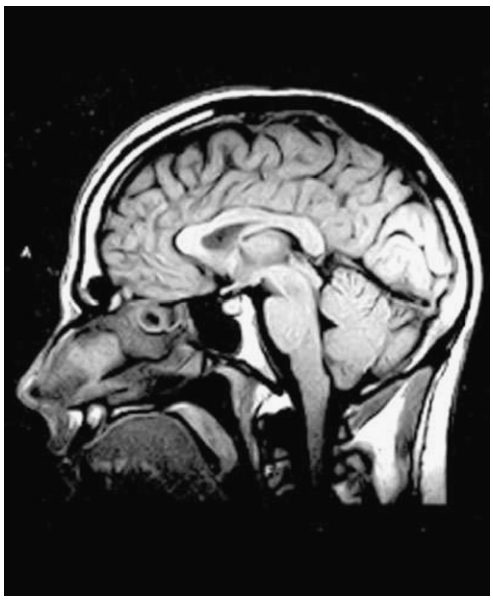

Fig. 8. A sagittal MRI image of the head. Source: http://www.rrvr.net/images/AIRIS/brain.jpg.

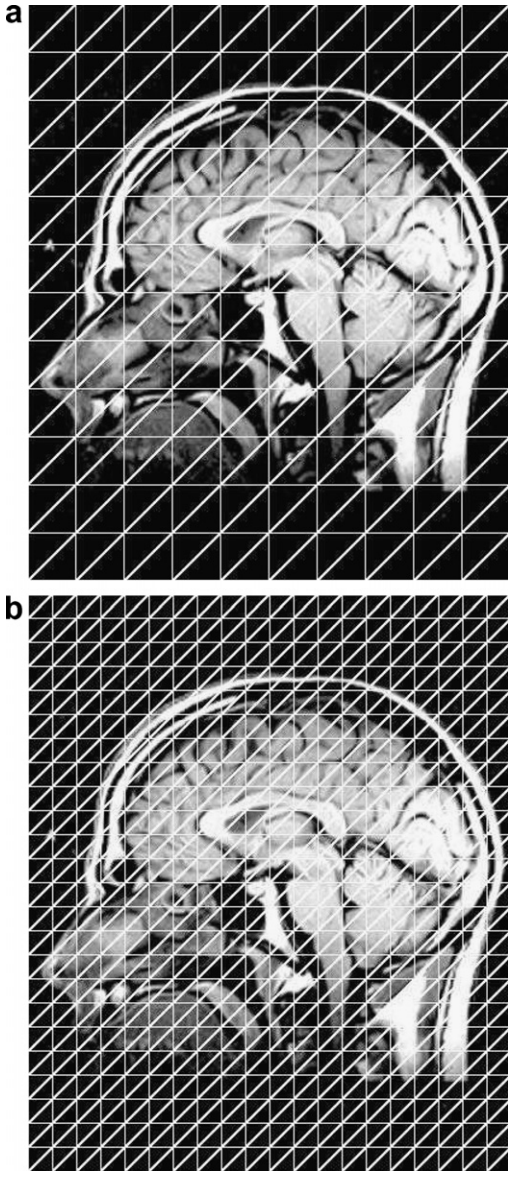

Fig. 9. Original image with superposed grid to generate approximation mesh. (a) Grid resolution is 48 pixels (length of horizontal/vertical edges); (b) grid resolution is 24 pixels.

full control over the mesh topology, i.e., one keeps track of the exact number of connected components and holes in the mesh after each triangle insertion. Thus, a user can also specify a desired topology, establishing the desired number of components and holes in the resulting mesh. Undesired components and holes may be removed using the appropriate operators. The software used to produce the examples below implements this approach operating over the SHE data structure.

Let us assume a user wishes to extract the region corresponding to the brain and medullae from the image in Fig. 8, isolating it from the outer regions that correspond to other elements such as the nose and skull. The first task for the user is to search for a suitable threshold range that successfully covers the target region. Fig. 10(a) shows the shaded mesh model representation of all the connected components detected in this image, for a threshold interval going from 120 to 235, with a grid resolution of two pixels. Just to provide a view of the mesh model, Fig. 10(b) depicts a zoom of the mesh corresponding to the region delimited by the rectangle in Fig. 10(a).

The above threshold range detects the brain and medulla as desired, however, regions that clearly do not correspond to brain are also detected and included in the mesh model. In addition to components that do correspond to identifiable "objects" in the image (such as brain, skull bone and tissue, nose), noisy regions were also detected as isolated components, as it would be expected from any threshold detecting technique. The resulting mesh has a total of 88 distinct components – each component corresponds to a single connected region – and 155 holes. The advantage of our approach over other segmentation techniques is that the underlying topological framework gives the user full control over components and holes. Moreover, such a framework is completely transparent to the user.

Now, the goal is to isolate the brain, which is known to consist of a single connected region with no holes. In this particular case, it is also clear that the region corresponding to the brain is the larger one in the image. The user may set a parameter "number of components" to one, causing only the largest component (the one with the highest number of triangles) to be kept, and the other ones to be removed using the inverse Morse operators described in Section 4. It is worth noting that criteria other than size could be used to keep/remove components. This is just an implementation choice. Fig. 11(a) shows the resulting shaded mesh model, which provides a very good approximation model for the brain region contained in the original image. The border is depicted in Fig. 11(b).

The mesh now consists of a single component, which corresponds well to the regions of brain and medulla. However, the model still has "non-existing" (i.e., in the "real" object) holes (117, in fact). Our framework allows these holes to be removed by running the program with a parameter "number of holes" set to zero. Fig. 12(a) and (b) show the resulting mesh and its border, when the number of holes is set to zero.

Further processing may be applied. For example, if all mesh triangles that include a singular vertex are removed, mesh regions that are loosely connected may be separated into different components. The effect of disconnecting the loosely connected components in the mesh from Fig. 12 is shown in Fig. 13. Note that the region at the bottom right-hand corner of the image were removed after eliminating a total of 109 singular vertices. Such an operation is immediate because the SHE data structure representing the mesh keeps an explicit representation of all its singular vertices.

We now compare the proposed method with three other segmentation techniques: a texture-based, Markovian Random Field (MRF) [6], a traditional K-means clustering algorithm [24] and a border tracker [1]. Fig. 14(a) and (c)

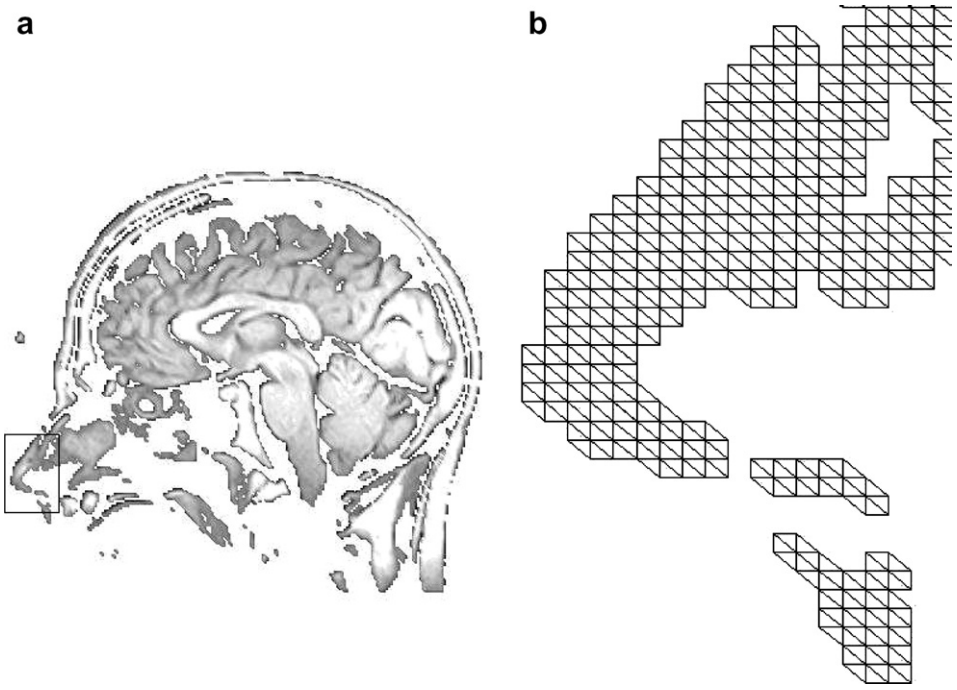

Fig. 10. (a) Shaded mesh model representation of connected components detected in the image for the user defined threshold 120–135; (b) the mesh that approximates the delimited region in (a), zoomed.

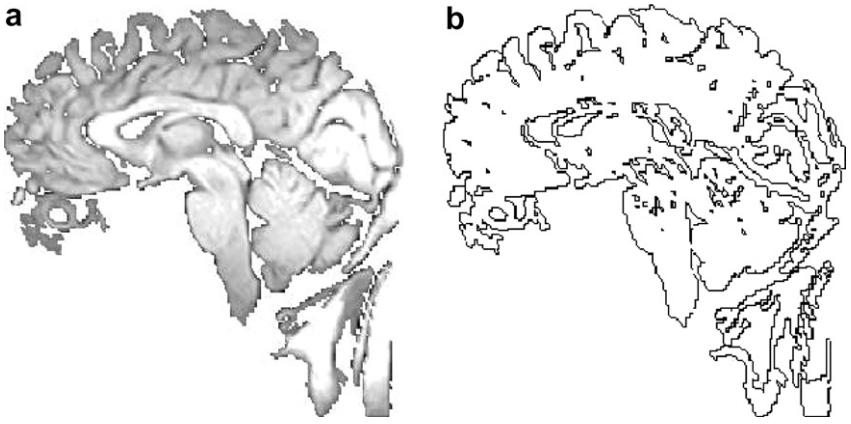

Fig. 11. (a) Mesh model (shaded) of the regions corresponding to brain and medulla, obtained after keeping only the largest connected component of all components depicted in Fig. 10(a); (b) the component border.

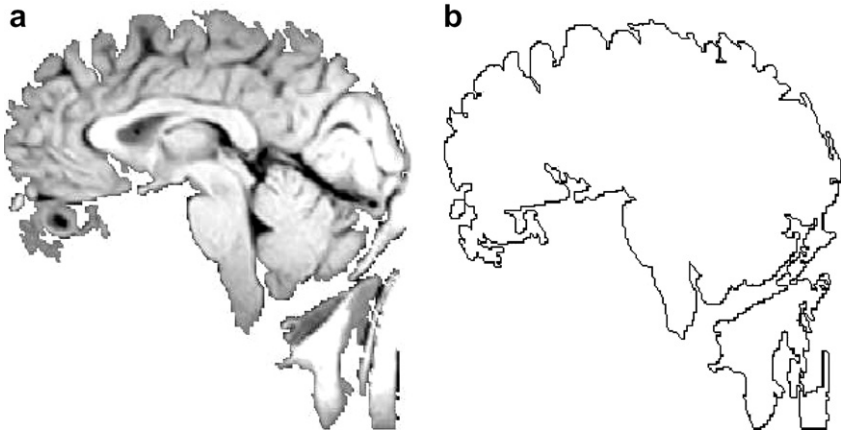

Fig. 12. (a) The same component from Fig. 11 after removing all holes; (b) the component border.

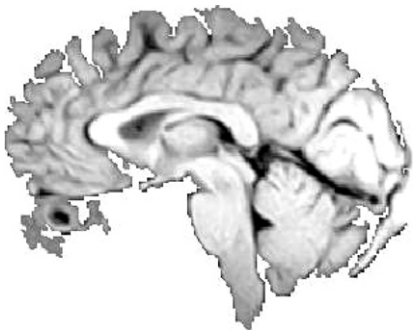

Fig. 13. The brain model of Fig. 12 after disconnecting loosely connected regions by removing singular vertices from the component. This produces two components, corresponding to the brain and the medullae (in the lower right region in Fig. 12), of which only the larger component (the brain) is shown here.

are the segmented images by the MRF and K-means clustering methods, respectively. Five distinct classes have been considered in both cases. While the Markovian approach models textures within the image, the K-means clustering in this case has been performed with the pixel intensity as a single feature. Fig. 14(b) and (d) show the boundaries extracted from their respective counterparts 14(a) and (c). The lack of topological control in both techniques yields a great amount of small contours, many of which are of no anatomical meaning. In order to obtain the same behaviour as the one provided by the topological control, segmentation by region based techniques such as MRF and K-means would require some post-processing, which, in general, is highly user dependent.

Fig. 15 illustrates the detected object (15(a)) and its respective contour (15(b)) produced by the border tracker method. This result somehow resembles that shown in Fig. 13, in which all holes and loosely connected regions have been removed. The major difference is that while in the former the single closed contour is an inherent – an the only possible – outcome of the border tracker approach (which always looks for contours of such nature), in the proposed method a single component is just one choice – out of many – made possible by the topological control provided. Furthermore, the proposed topological approach is less prone to noise and provides not only the contours but also the content of the region.

The mesh models thus extracted may be used in applications that typically handle geometric models. As an illustration, Fig. 16 shows an elastic deformation being applied to the brain model extracted from Fig. 8. This type of deformation is widely used in problems in which a model must be adjusted to fit a second model, for example to assess how much two objects differ. A measure of the deformation required to fit the two models may be employed as an estimate measure of their differences. In Fig. 16(a) the

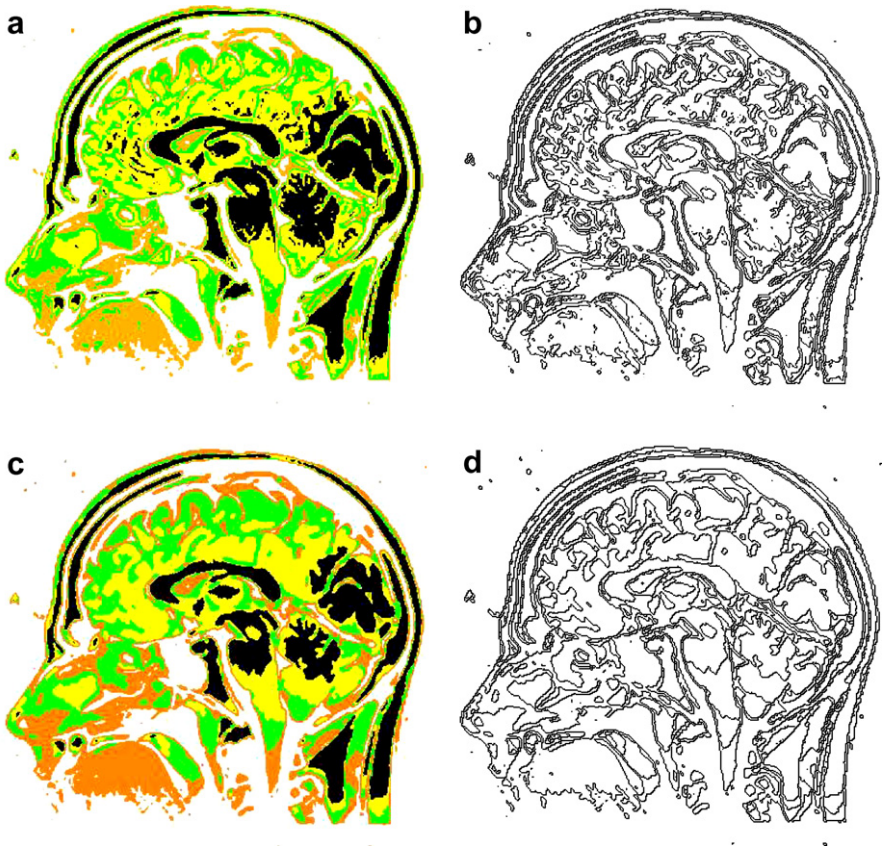

Fig. 14. Segmentation by MRF and K-means clustering for five classes. (a) Segmented image by K-means clustering; (b) extracted boundaries; (c) segmented image by MRF; (d) extracted boundaries.

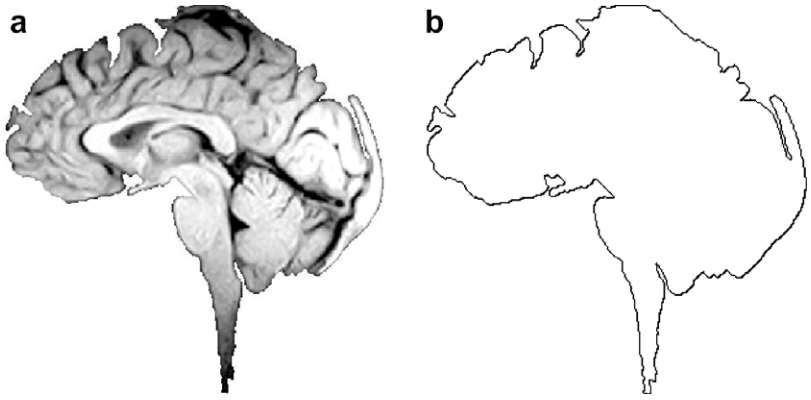

Fig. 15. Image segmentation by border tracking: (a) Single region with no holes; (b) component border.

brain model with a single hole introduced is shown. Different stages of the deformation, achieved by displacing the hole, are depicted in Fig. 16(b)–(d). Fig. 17(a) shows the original mesh model (prior to deformation) of the region delimited in Fig. 16(a). The deformed mesh corresponding to the same region in Fig. 16(d) is depicted in Fig. 17(b).

7. Conclusions and further work

A robust topological framework for triangle characterization in two-dimensional meshes has been introduced and its application illustrated for the problem of detecting objects and constructing two-dimensional models from images. Comparative results have shown that the topological control may, according to the application domain, be a usefull feature, as unimportant objects can be easily removed by specifying a desired topoloy. Traditional geometric modeling techniques rely on available knowledge on object geometry and topology to build a model. However, when building models from images little or no geometrical information about the target objects is known *a priori*. Nevertheless, topological

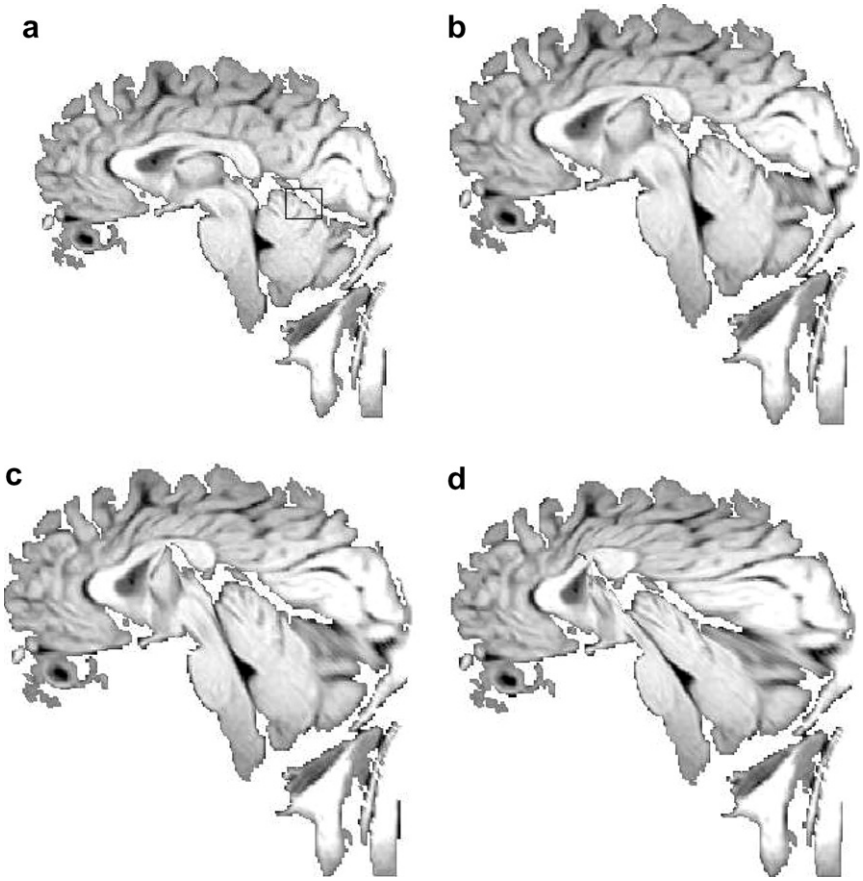

Fig. 16. (a)–(d) Different stages of an elastic deformation applied to the mesh model of the brain.

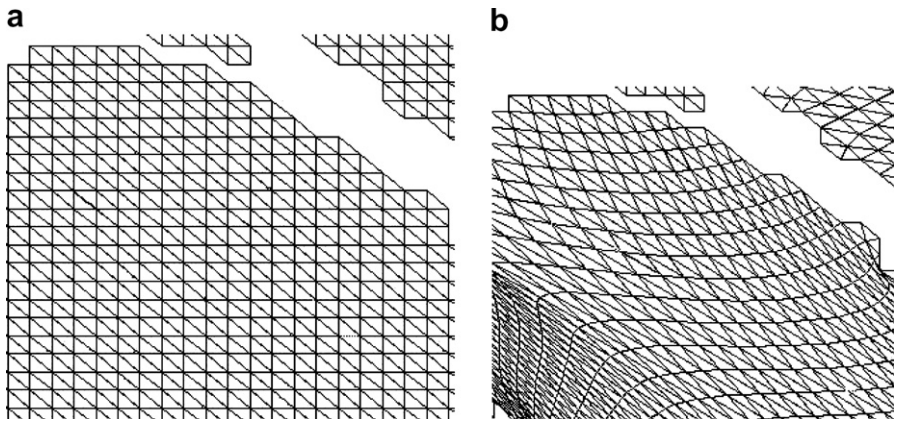

Fig. 17. (a) Mesh model of the delimited region in Fig. 16(a); (b) deformed mesh model of the same region in Fig. 16(d).

information is available, even though it is usually ignored by traditional techniques that extract information from images, such as deformable models. The strength of our approach lies in the way it exploits topological information to enable the creation of the desired models, combining techniques from geometric modeling and digital image processing.

We show that the formal treatment given to triangle characterization provides a high degree of topological control in modeling objects from images. The set of Morse operators and supporting data structure allow triangle insertion and removal to be accomplished in a unified and consistent manner, at a suitable level of abstraction for the user and without taking into account the specificities of different applications. The capability of precisely describing and tracking the topological changes introduced whenever a triangle is inserted or removed from a given mesh is useful for a range of applications in different fields. Another potential application not discussed here is in the implementation of popular mesh generation procedures for numerical simulation.

Acknowledgments

The authors acknowledge the financial support of FAPESP – the State of São Paulo Research Funding Agency – and CNPq, – the Brazilian Federal Research Funding Agency. They are also grateful to Igor Prata Soares for his elastic deformation software, which was used to generate the deformed models in Figs. 16 and 17.

References

[1] J. Batista, R. Freitas, An adaptive gradient-based boundary detector for MRI images of brain, in: Image Processing and its Applications 1999, vol. 456, July 1999, pp. 440–445.
[2] B.G. Baumgart, A polyhedron representation for computer vision, AFIPS National Computer Conference 44 (1975) 589–596.
[3] J. Canny, A computational approach to edge detection, in: RCV87, 1987, pp. 184–203.
[4] M. Codrea, O. Nevalainen, Note: an algorithm for contour-based region filling, Computers and Graphics 29 (3) (2005) 441–450.
[5] D. Comaniciu, Image segmentation using clustering with saddle point detection, in: ICIP, issue 3, 2002, pp. 297–300.
[6] M. Commer, E. Delp, The {EM/MPM} algorithm for segmentation of textured images: analysis and further experimental results, IEEE Transactions on Image Processing 9 (10) (2000) 1731–1744.
[7] J.L. Díaz de León, J.H.S. Azuela, On the computation of the euler number of a binary object, Pattern Recognition 29 (3) (1996) 471–476.
[8] S.B. Gray, Local properties of binary images in two dimensions, IEEE Transactions on Computers C-20 (1971) 551–561.
[9] L. Guibas, J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of voronoi, ACM Transactions on Graphics 4 (2) (1985) 74–123.
[10] G.T. Herman, Geometry of Digital Spaces, Birkhauser, Boston, 1998.
[11] A. Huertas, G.G. Medioni, Detection of intensity changes with subpixel accuracy using laplacian–gaussian masks, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (5) (1986) 651–664.
[12] L. Ji, H. Yan, Robust topology-adaptive snakes for image segmentation, Image and Vision Computing 20 (2) (2002) 147–164.
[13] S. Kollias, D. Kalogeras, A multiresolution probabilistic neural network for image segmentation, in: Proceedings of the ICASSP'94, April 1994, pp. 569–572.
[14] T.Y. Kong, A. Rosenfeld, Digital topology: introduction and survey, Computer Vision Graphics and Image Processing 48 (3) (1989) 357–393.
[15] H. Lopes, A. Castelo, G. Tavares, Handlebody representation for surfaces and Morse operators, in: Curves and Surfaces in Computer Vision and Graphics III, vol. 1830, 1992, pp. 270–283.
[16] G. Malandain, S. Fernández-Vidal, Euclidean skeletons, Image and Vision Computing 16 (5) (1998) 317–327.
[17] M. Mäntylä, An Introduction to Solid Modeling, Computer Science Press, Inc., New York, USA, 1987.
[18] J.W. Milnor, Morse Theory, Princeton University Press, 1963.
[19] L.G. Nonato, A. Castelo, R. Minghim, J.E.S. Batista, Morse operators for digital planar surfaces and their applications in image segmentation, IEEE Transactions on Image Processing 13 (2) (2004) 216–227.
[20] S. Novianto, Y. Suzuki, J. Maeda, Near optimum estimation of local fractal dimension for image segmentation, Pattern Recognition Letters 24 (2003) 365–374.
[21] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
[22] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, Inc., 1983.

[23] Y. Sun, S. Ozawa, Efficient wavelet-based image retrieval using coarse segmentation and fine region feature extraction, IEICE Transactions on Information and Systems e88-d (5) (2005) 1021.
[24] R. Wilson, M. Spann, A new approach to clustering, Pattern Recognition 23 (12) (1990) 1413–1425.
[25] Y. Yang, M.N. Wernick, J.G. Brankov, A fast approach for accurate content-adaptive mesh generation, IEEE Transactions on Image Processing 12 (8) (2003) 866–881.
[26] H. Zhang, Z. Yuan, Z. Cai, Z. Bian, Segmentation of {MRI} using hierarchical markov random field, Journal of Software 13 (9) (2002) 1779–1786.