ELSEVIER

Medical Image Analysis 4 (2000) 73–91

MEDICAL IMAGE ANALYSIS

www.elsevier.com/locate/media

T-snakes: Topology adaptive snakes

Tim McInerney*, Demetri Terzopoulos

Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 3H5

Received 3 March 1998; received in revised form 22 March 1999; accepted 26 October 1999

Abstract

We present a new class of deformable contours (snakes) and apply them to the segmentation of medical images. Our snakes are defined in terms of an affine cell image decomposition (ACID). The 'snakes in ACID' framework significantly extends conventional snakes, enabling topological flexibility among other features. The resulting *topology adaptive snakes*, or 'T-snakes', can be used to segment some of the most complex-shaped biological structures from medical images in an efficient and highly automated manner.

Keywords: Deformable contours (snakes); Medical images; Affine cell image decomposition (ACID); Topological flexibility; Complex biological structures

1. Introduction

Segmenting anatomic structures from medical images and reconstructing compact analytic representations of these structures is a challenging problem. This is due to the complexity and variability of the anatomic shapes of interest and the sheer size of the data sets. Furthermore, the shortcomings typical of sampled data, such as sampling artifacts, spatial aliasing, and noise, may cause the boundaries of structures to be indistinct and disconnected. The challenge is to extract the boundary elements belonging to the anatomic structure and integrate these elements into a complete and consistent model of that structure. This process should be performed as efficiently and automatically as possible.

Deformable models (Terzopoulos et al., 1988), which include the popular deformable contours or *snakes* (Kass et al., 1988), are a powerful segmentation technique designed to meet this challenge (see the recent survey by McInerney and Terzopoulos (1996) and the compilation by Singh et al. (1998)). Deformable models tackle the segmentation problem by considering an object boundary as a single, connected structure. They exploit a priori knowledge of object shape and inherent smoothness, usually formulated as internal deformation energies, to compensate for noise, gaps and other irregularities in object boundaries. Their underlying geometric representations provide a compact, analytical description of an object. Moreover, these models support highly intuitive interaction mechanisms that, when necessary, allow medical scientists and practitioners to bring relevant expertise to bear on the model-based image interpretation task.

Their advantages notwithstanding, the classical parametric snakes models have several limitations that mitigate their utility across the full range of medical image analysis problems and limit their potential for automation. A significant shortcoming in certain applications is their topological inflexibility. In this paper we describe a new class of deformable contour models known as *topology adaptive snakes*, or *T-snakes*. Our approach exploits an *affine cell decomposition* (Allgower and Georg, 1990; Munkres, 1984) of the image domain, creating a mathematically sound framework that significantly extends the abilities of standard snake models. The *affine cell image decomposition* (ACID) divides the image domain into a collection of convex polytopes.

We immerse discrete versions of conventional parametric snakes in ACID and incorporate an efficient reparameterization algorithm to produce T-snakes that are able to conform to complex geometries and topologies. The ACID

*Corresponding author.

1361-8415/00/$ – see front matter
PII: S1361-8415(00)00008-6

framework enables T-snakes to maintain the traditional features associated with parametric snakes models, such as user interaction and constraints through energy or force functions, while overcoming several of their limitations. The ACID framework also provides a convenient mechanism for the incorporation of 'hard' geometric and topological constraints. These properties in combination produce an effective and general tool for the efficient, accurate, reproducible, and highly automated extraction and analysis of anatomic structures from medical images.

This paper focuses on the two-dimensional T-snakes formulation first proposed in McInerney and Terzopoulos (1995b,c); however, we have recently extended the ACID framework to 3D, leading to topology adaptive deformable surfaces, or T-surfaces (McInerney and Terzopoulos, 1999). Section 5 will present a brief introduction to T-surfaces and an example of their use.

2. Background

The segmentation of medical images – the partitioning of image points into subsets corresponding to meaningful anatomic structures – is an essential first stage of most image-based medical analysis tasks, including shape analysis, visualization, registration, labeling, and motion tracking. These tasks usually require imaged anatomic structures to be reduced to compact, analytic shape representations. The traditional, manual segmentation of medical images can be extremely labor intensive and time-consuming. Consequently, semi-automatic and, ultimately, fully automated techniques are a desirable goal. Increasing the degree of automation can not only relieve clinicians from much tedious work, but it can also increase the efficiency, accuracy, and perhaps most importantly, reproducibility of the segmentations. However, since erroneous automated interpretations of medical images are usually unacceptable, any successful segmentation technique should support intuitive, efficient interactive guidance or editing by the medical expert.

With this motivation in mind, our goal is to extend considerably the capabilities of standard snakes models (see Appendix A for a review), improving their performance and increasing their degree of automation, while retaining their traditional strengths. An important property to maintain is the ability to design energy or force functions to constrain and interactively guide the model. The most significant limitations to overcome are as follows:

- Standard parametric snakes were designed as interactive models and they usually must be brought close to the boundary of the target object to capture the boundary well. To make snakes more automatic, mechanisms must be added that overcome the initialization sensitivity problem.
- The fixed geometric parameterization of a standard snake in conjunction with the internal deformation (usually arc-length and curvature) energy constraints can limit its flexibility and prevent the snake from conforming to long tubular shapes or shapes with significant branching and protrusions.
- The fixed parameterization of a standard snake also makes it incapable of topological transformations, thus the topology of the object of interest must be known in advance.

Several researchers have attempted to overcome some of these limitations by adding greater functionality to parametric model formulations, such as 'inflation' forces (Terzopoulos et al., 1988; Cohen and Cohen, 1993), the use of automatic snake element subdivision mechanisms (Ivins and Porrill, 1994; Lobregt and Viergever, 1995), and the integration of region-based information (Rougon and Prêteux, 1991; Herlin et al., 1992; Chakraborty et al., 1994; Gauch et al., 1994; Poon et al., 1994; Chakraborty and Duncan, 1995). However, parametric models are incapable of adapting to object topology and segmentation results remain too dependent on initial position. Thus when dealing with images containing complex-shaped objects or objects that are embedded in other objects, the improved parametric models may still require extensive user interaction. Interestingly, in contrast to the 2D case, there are several 3D parametric deformable surface models that are capable of automatically adapting to object topology (Leitner and Cinquin, 1991; Szeliski et al., 1993; Whitaker, 1994; Malladi et al., 1996; Lachaud and Montanvert, 1999). Unlike T-surfaces which are automatically reparameterized via the ACID framework, the reparameterization process of these models is typically based on subdivision rules and not on the intrinsic local geometry of the target object; these triangle refinement mechanisms can create initial position sensitivity that adversely affects segmentation reproducibility.

To introduce topological flexibility, several researchers have developed implicit snakes by adopting Osher and Sethian's (1988) level-set evolution technique to the image segmentation problem (Caselles et al., 1993, 1995; Whitaker, 1994; Malladi et al., 1995; Sapiro et al., 1995). These models are formulated as evolving contours (surfaces in 3D) or 'fronts' which define the level set of some higher-dimensional (hyper-) surface over the image domain. The main feature of this approach is that topological changes are handled naturally, since the level set need not be simply connected; the higher-dimensional surface remains a simple function even as the level set changes topology. While the Osher–Sethian technique is an attractive mathematical framework, partial differential equations governing curvature-dependent front evolution, implicit formulations are not nearly as convenient as explicit, parametric formulations when it comes to incorporating additional control mechanisms including internal deformation energies and external interactive guidance by expert users. Furthermore, the higher dimensional implicit surface

formulation makes it difficult, if not impossible, to impose arbitrary geometric or topological constraints on the level set indirectly through the higher dimensional representation. Therefore, the implicit formulation may potentially limit the ease of use, efficiency and degree of automation achievable in the segmentation task.

In this paper, we develop a 'snakes in ACID' approach that endows a discrete version of a parametric deformable contour model with the topological flexibility inherent to the implicit model formulation, thus gaining the best of both worlds. The ACID automatically and efficiently reparameterizes the T-snakes as they evolve. The T-snakes inherit the features traditionally associated with conventional snakes, such as user interaction and the incorporation of energy/force-based constraints. At the same time, the ACID reparameterization mechanism enables T-snakes to flow into complex shapes, changing topology as necessary. This allows T-snakes to segment objects with complex topologies or objects which contain smaller, embedded objects, as well as automatically merging with other T-snakes introduced by the user. The T-snake will generally produce very similar segmentations no matter where it is initialized within the imaged object of interest. If desired, the underlying parametric formulation enables the use of high-order polynomial finite elements to discretize the contour and/or the integration of image information over the extent of each element.

3. A T-snakes formulation

3.1. Overview

We define our T-snakes model as a closed 2D contour consisting of a set of nodes connected in series. A T-snake is a discrete approximation to a conventional parametric snakes model and retains many of its properties. The internal forces act as a smoothness constraint and users can interact with the model using spring forces and other constraints. An 'inflation' force is used to push the model towards image edges until it is opposed by external image forces. The deformation of the model is governed by discrete Lagrangian equations of motion. We formulate the deformable contour model in Section 3.2.

Unlike traditional snakes, the set of nodes and interconnecting elements of a T-snake does not remain constant during its evolution. We decompose the image domain into a grid of discrete cells. As the model moves under the influence of external and internal forces, we reparameterize the model with a new set of nodes and elements by efficiently computing the intersection points of the model with this ACID grid. We also keep track of the interior region of the model by 'turning on' any grid vertices passed over by the T-snake during its motion. By reparameterizing the model at user-specified iterations of the evolutionary process, we create a simple, elegant and automatic model subdivision technique and the ACID grid provides a framework for robust topological transformations. This allows the model to be relatively insensitive to its initial position and 'flow' into complex shapes with complex topologies in a stable manner. By providing a boundary representation as well as a representation of the interior region of an object, this hybrid snakes model combines the space partitioning, intrinsic parameterization and topological flexibility properties of an implicit formulation with the boundary properties of a parametric model.

The overall motion of a T-snake is analogous to the motion of a propagating front. However, there are two distinct phases to the motion. In the first phase, between reparameterization stages, a T-snake behaves as a standard parametric snake and evolves according to Lagrangian dynamics. This Lagrangian formulation phase allows any data-derived or user-defined force to guide the snake. During the second or reparameterization phase, the snake is reparameterized in terms of the ACID grid and the fixed grid points are used to track the interior of the closed contour model, creating a space partitioning similar to that of an implicit function. This phase provides stability, intrinsic parameterization, and topological adaptability.

Conversion to the traditional parametric snakes model representation is simply a matter of disabling the ACID grid at any time during the evolution process.

3.2. Model description

The first component of T-snakes is a discrete form of the standard snakes model described in Appendix A. A T-snake is defined as a set of N nodes, indexed by $i = 0, \ldots, N-1$, connected in series by a set of N edges or elements. We associate with these nodes time varying positions $\boldsymbol{x}_i(t) = [x_i(t),\ y_i(t)]$, along with tensile forces $\boldsymbol{\alpha}_i(t)$, flexural forces $\boldsymbol{\beta}_i(t)$, inflationary forces $\boldsymbol{\rho}_i(t)$, and external forces $\boldsymbol{f}_i(t)$ that act in the image plane. A periodic boundary condition $\boldsymbol{x}_1(t) = \boldsymbol{x}_N(t)$ is applied to produce a closed contour model.

The behavior of a T-snake is governed by a simplified version of (A.5) in discrete form. The result is a set of first-order ordinary differential equations of motion

$$\gamma_i \dot{\boldsymbol{x}}_i + a\boldsymbol{\alpha}_i + b\boldsymbol{\beta}_i = \boldsymbol{\rho}_i + \boldsymbol{f}_i, \tag{1}$$

where $\dot{\boldsymbol{x}}_i(t)$ is the velocity of node i and γ_i is a damping coefficient. The internal tensile forces

$$\boldsymbol{\alpha}_i(t) = 2\boldsymbol{x}_i(t) - \boldsymbol{x}_{i-1}(t) - \boldsymbol{x}_{i+1}(t) \tag{2}$$

(nodal indexes are evaluated modulo N) are a discrete approximation to the second derivative of the coordinate functions with respect to s (third term of (A.5)) and act to maintain a uniform spacing between model nodes. The parameter a controls the resistance of the contour to stretching deformations. The tensile forces can be made scale invariant by dividing the right hand side of (2) by the

distance between neighboring nodes. The internal flexural forces

$$\boldsymbol{\beta}_i(t) = 2\boldsymbol{\alpha}_i(t) - \boldsymbol{\alpha}_{i-1}(t) - \boldsymbol{\alpha}_{i+1}(t) \tag{3}$$

are a discrete approximation to the fourth derivative of the coordinate functions with respect to s (fourth term of (A.5)). The parameter b controls the resistance of the contour to bending deformations.

On the right hand side of (1), $\boldsymbol{\rho}_i$ and $\boldsymbol{f}_i$ are external forces. Since the model has no inertia, it comes to rest (i.e., $\dot{\boldsymbol{x}}_i = 0$) as soon as the applied forces balance the internal forces. An inflation force is used to push the model towards intensity edges in the image $I(x, y)$, until it is opposed by the image forces. The inflation force is

$$\boldsymbol{\rho}_i(t) = qF(I(\boldsymbol{x}_i(t))\boldsymbol{n}_i(t), \tag{4}$$

where $\boldsymbol{n}_i$ is the unit normal vector to the contour at node i, and q is the amplitude of this force. The binary function

$$F(I(x, y)) = \begin{cases} +1 & \text{if } I(x, y) \geqslant T, \\ -1 & \text{otherwise,} \end{cases} \tag{5}$$

links the inflation force to the image data $I(x, y)$, where T is an image intensity threshold. To calculate a continuous image function $I(x, y)$ we compute the intensity at an arbitrary point (x, y) by bilinearly interpolating the intensities at the four pixels surrounding (x, y). The function F makes the T-snake contract when $I(x, y) < T$ and is used to prevent the T-snake from leaking into the background. Oscillation of the T-snake can be prevented by progressively lowering the magnitude q of the force toward zero once a change of direction is detected or if a T-snake element remains within the same grid cell for a specified number of iterations. Region-based image intensity statistics can be incorporated into the inflation force by extending the function as follows (Ivins and Porrill, 1994; Kapur et al., 1996):

$$F(I(x, y)) = \begin{cases} +1 & \text{if } |I(x,y) - \mu| \leqslant k\sigma, \\ -1 & \text{otherwise,} \end{cases} \tag{6}$$

where μ is the mean image intensity of the target object, σ the standard deviation of the object intensity and k is a user-defined constant. The values of μ and σ are typically known a priori or computed from the image. The inflation force essentially creates an *active* region growing model that provides insensitivity to noise within the region through the connectivity and internal smoothness constraints of the T-snake; smooth, subpixel-accurate region boundaries are produced and a T-snake will pass over small spurious regions, preventing the creation of small holes in the region.

To stop the contour at significant edges, we include the external force

$$\boldsymbol{f}_i(t) = p\nabla P(\boldsymbol{x}_i(t)), \tag{7}$$

where the weight p controls the strength of the force and the potential P is defined by (A.4). The weights p and q are usually chosen to be of the same order, with p slightly larger than q so that a significant edge will stop the inflation but with q large enough so that the model will pass through weak or spurious edges. The image edge force can also be averaged over a local neighborhood centered at $\boldsymbol{x}_i$ to improve robustness against noise.

Another effective external force is an inflation force that makes use of a Chamfer distance map (Borgefors, 1984) or a gradient vector field that approximates the distance and direction to the nearest edge (Prince and Xu, 1996). In this scenario, the inflation force is weighted by the distance to the edge and is directed along a T-snake node normal. Once equilibrium has been achieved the inflation force is turned off and the image edge force is activated. This 'force phasing' approach is an effective means of preventing the T-snake from leaking into neighboring structures when there are significant gaps in the target object edges. Other useful functions F that can be used as inflation force weights include local variance and intensity (Lorigo et al., 1998), texture (Durikovic et al., 1995), or other statistical measures of the target object intensity.

We integrate (1) forward through time using an explicit first-order Euler method. This method approximates the temporal derivatives with forward finite differences. It updates the positions of the model nodes from time t to time $t + \Delta t$ according to the formula

$$\boldsymbol{x}_i^{(t+\Delta t)} = \boldsymbol{x}^{(t)} - \frac{\Delta t}{\gamma}(a\boldsymbol{\alpha}_i^{(t)} + b\boldsymbol{\beta}_i^{(t)} - \boldsymbol{\rho}_i^{(t)} - \boldsymbol{f}_i^{(t)}). \tag{8}$$

The explicit Euler method is simple, but it becomes unstable unless small time steps are used. In our T-snakes application, however, a very reasonable range of time step sizes can be found that produce stable behavior resulting in accurate segmentations.

3.3. Affine cell decomposition

The second component of T-snakes is the affine cell image decomposition. A space decomposition subdivides space into a collection of disjoint connected subsets. A typical subset is a k-dimensional cell (i.e., a set which is homeomorphic to an open disk of dimension k), where the boundary of each cell is defined to be a finite union of lower dimensional cells. The subdivision of space using such subsets is known as a *cellular complex* (Munkres, 1984). Cellular complexes are a powerful tool for constructing definitions and proofs for image topology (Kovalevsky, 1989).

Affine cell decompositions are examples of special cases of cell decompositions obtained by restricting the geometry of the cells to that of a *convex polytope*. There are two main types of affine cell decomposition methods: nonsimplicial and simplicial. Most nonsimplicial methods employ a rectangular tessellation of space. The marching cubes algorithm (Lorenson and Cline, 1987) is an example of

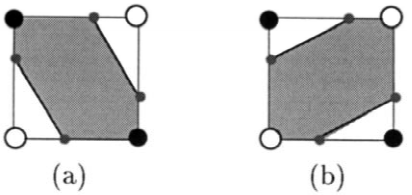

Fig. 1. Example of ambiguous faces of a rectangular cell (black circle, positive cell vertex; open circle, negative cell vertex). Given the diagonal arrangement of vertex polarities, it is unclear which edge intersection should be used.

this type. These methods are easy to implement, but they cannot be used to represent the boundaries of an implicitly defined object unambiguously without the use of a disambiguation scheme (Ning and Bloomenthal, 1993). For example, Fig. 1 shows two possible boundary representations within a rectangular cell of an implicitly defined object. A disambiguation scheme consists of a table lookup to identify ambiguous cases followed by adherence to a disambiguation strategy such as 'preferred polarity' – always separate the positive vertices (and join the negatives) or vice versa.

In a simplicial cell decomposition (Allgower and Georg, 1990), also known as a *triangulation*, space is partitioned into cells defined by open simplices, where an n-simplex is the simplest geometrical object of dimension n: e.g., a triangle in 2D or a tetrahedron in 3D. When using cell decomposition schemes, it is desirable to have a small number of cell types, e.g., *congruent cells*, that differ only by orientation or reflection. If all the cells are identical, computations can be made very simple and efficient. For example, the simplest triangulation of Euclidean space $\Re^n$ with this property is the *Coxeter–Freudenthal* triangulation (Fig. 2). It is constructed by dividing space using a uniform cubic grid and the triangulation is obtained by subdividing each cube into $n!$ simplices.[1]

3.4. Simplicial approximation

Simplicial cell decompositions provide a framework for

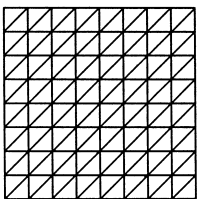

Fig. 2. Freudenthal triangulation.

[1]We have implemented T-snakes using both non-simplicial and simplicial decomposition methods. We will describe the simplicial grid approach in this paper. The formulation of the model using a non-simplicial grid is essentially identical except for the addition of a simple disambiguation scheme during the reparameterization of a T-snake.

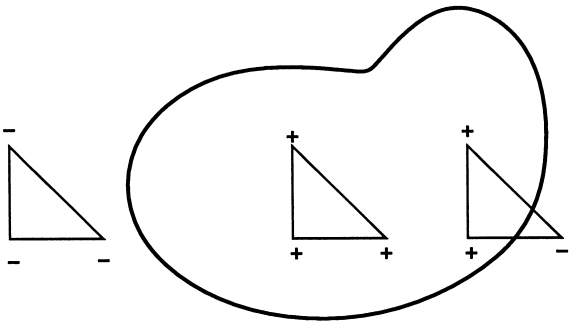

Fig. 3. Simplex classification.

the creation of robust, consistent local polygonal (affine) approximations of the boundary contours of anatomic structures. In 2D, an anatomic structure partitions an image into two open sets of dimension 2 (the interior and exterior points) and one open set of dimension 1 (the boundary points). A simplex σ^2 can be classified in relation to this partitioning of space by testing the 'sign' of its vertices. If the signs are the same for all vertices, the simplex must be totally inside or outside the structure. If the signs are different, the boundary of the structure must intersect the simplex (Fig. 3). In a k-simplex, the negative (inside) vertices can always be separated from the positive (outside) vertices by a single (hyper-) plane; thus an unambiguous polygonalization of the simplex always exists. Furthermore, by the definition of a simplicial complex, a consistent polygonalization of the entire boundary contour will result.

In a 2D image, the set of grid cells that intersect the boundary contour of the anatomic structure are termed the boundary cells. These boundary cells form a two-dimensional combinatorial manifold that has as its dual a one-dimensional manifold that approximates the contour. The one-dimensional manifold is constructed from the intersection of the object boundary contour with the edges of each boundary cell. Using simplicial cell decomposition, the intersection points result in one line segment approximating the boundary contour inside each boundary triangle (Fig. 4). Each line segment intersects a boundary triangle at two distinct edges, separating the inside vertices from the outside vertices. The set of line segments comprises the combinatorial manifold that approximates the boundary contour of the object. We can obtain an approximation to any desired degree of accuracy by decreasing the size of the grid cells.

3.5. Iterative reparameterization

A T-snake is reparameterized every M time steps of the numerical time integration (referred to as a *deformation step*), where M is user-controllable and typically set between 5 and 10. The entire T-snake is set to either expand or shrink during one deformation step. This policy means that although a T-snake element is free to move

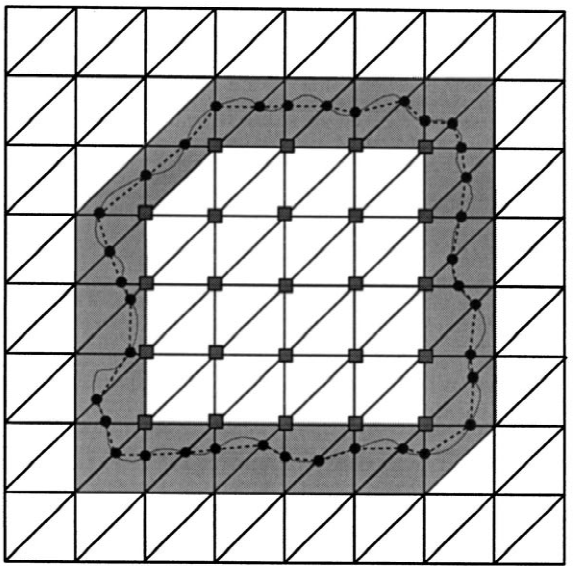

Fig. 4. Simplicial approximation (dashed-line) of an object contour (solid-line) using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded.

forward and backwards during a deformation step, similar to a parametric snake, at the end of the deformation step the element cannot have moved such that a grid vertex previously inside the T-snake is now outside. The reason for this policy is explained later in this section (see phase II). This deformation restriction also applies to the level-set evolution techniques. However, a T-snake can alternate between expansion and shrinkage deformation steps, effectively mimicing the complete freedom of movement over all deformation steps of a standard parametric snake. This allows the user to interact with the T-snake in a manner similar to that of a standard snake.

At the beginning of the deformation step, the T-snake nodes are defined in terms of the edges of the grid boundary cells. At the end of the deformation step, the nodes have moved relative to the grid cell edges (Fig. 5(a)). We then re-establish the correspondence of the model with the grid by computing a new simplicial approximation of the deformed T-snake. This new simplicial approximation is computed using a robust two-phase reparameterization algorithm. We have explored several reparameterization algorithms. One such algorithm is detailed in

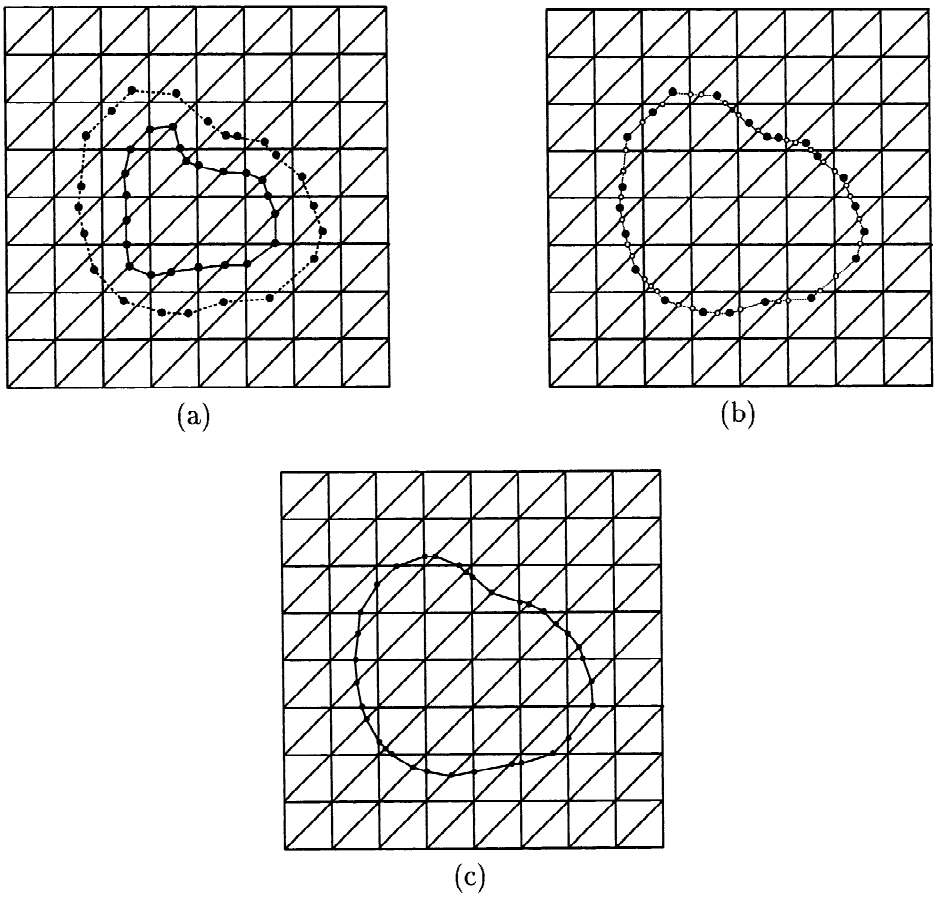

Fig. 5. Phase II of T-snake reparameterization: (a) T-snake expands (solid line) and moves relative to the grid during the deformation step (dashed line), (b) new model nodes are computed, (c) new T-snake nodes and elements.

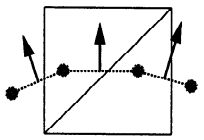

Fig. 6. 2D example illustrating the computation of the 'sign' of a grid cell edge and model element intersection point. The T-snake element normal is used to determine which grid edge vertex is on the inside half-space of the T-snake element and which is on the outside half-space.

(McInerney, 1997). We are currently using a newer and simpler algorithm which is more readily extensible to higher dimensions.

3.5.1. Phase I

In phase I of the reparameterization algorithm, we perform an efficient local search and intersection test of each T-snake element with the grid cell edges. If an intersection point is found for a particular grid cell edge, it is stored in a data structure associated with this edge. This intersection point may become a node of the updated T-snake (Fig. 5(b,c)). The intersection point may be unused and therefore discarded if after the second phase of the reparameterization process both grid vertices of the grid cell edge are 'on'. This means that both grid vertices must be inside the T-snake and the edge joining them is therefore not a boundary edge.

Several intersection points may be found for a particular grid cell edge. This situation occurs when, during a deformation step, a T-snake intersects itself or when multiple T-snakes intersect. In these cases, we take the lower-numbered vertex of the grid edge and determine on which side of the line formed by the T-snake element it lies (Fig. 6). Consequently, every intersection point is given a 'sign' – either inside or outside (using the outward pointing T-snake element normal as the reference direction). Thus, every intersection point is compared against the existing intersection point (if any) of a grid cell edge. If the intersection points are of opposite signs, they cancel each other out. If they are of the same sign, the new intersection point replaces the existing point. If no intersection point currently exists for the grid edge, the new intersection point is stored in the grid edge data structure.

Finally, if the grid vertex on the inside half-space of the T-snake element is 'off', we also store it on a queue for processing in phase II. Phase I of the reparameterization process is simple, efficient and is inherently parallel; each T-snake element can be processed independently.

3.5.2. Phase II

During a deformation step, as the T-snake expands, some grid vertices that were outside the closed T-snake will now be contained inside (Fig. 7(a)). We then update the state of these grid vertices from 'off' to 'on'. In this manner, we are able to determine and track the interior region of the T-snake. The total set of 'on' or inside grid vertices unambiguously defines the boundary of the T-snake; they are used to track the grid boundary cells continuously (i.e., cells having both inside and outside grid vertices) throughout the evolution of the T-snake and hence determine which intersection points should be used to form the nodes and elements of the new T-snake. Furthermore, the inflation force pushes a T-snake in a direction normal to the contour at each node. This form of evolution can result in singularities and self-intersections (although this is ameliorated by the smoothing effect of the internal forces). In these situations, it is not clear how to evolve the T-snake. We solve this problem by mimicing the physically correct behavior for a propagating flame front. This behavior is selected by adhering to a so-called *entropy condition* (Osher and Sethian, 1988): once a grid vertex is turned on, it remains on.

As mentioned in phase I, during a deformation step, for each T-snake element we compute its grid cell edge intersection points and enqueue grid cell vertices to be labeled 'on' or inside for processing in phase II. In phase

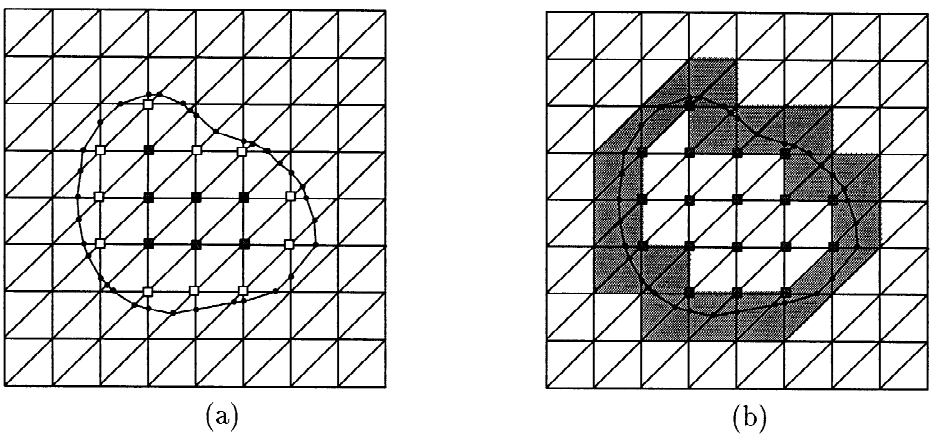

Fig. 7. Phase II of T-snake reparameterization: (a) new T-snake nodes along with new inside grid vertices (light-shaded), (b) new T-snake and active boundary grid cells (shaded).

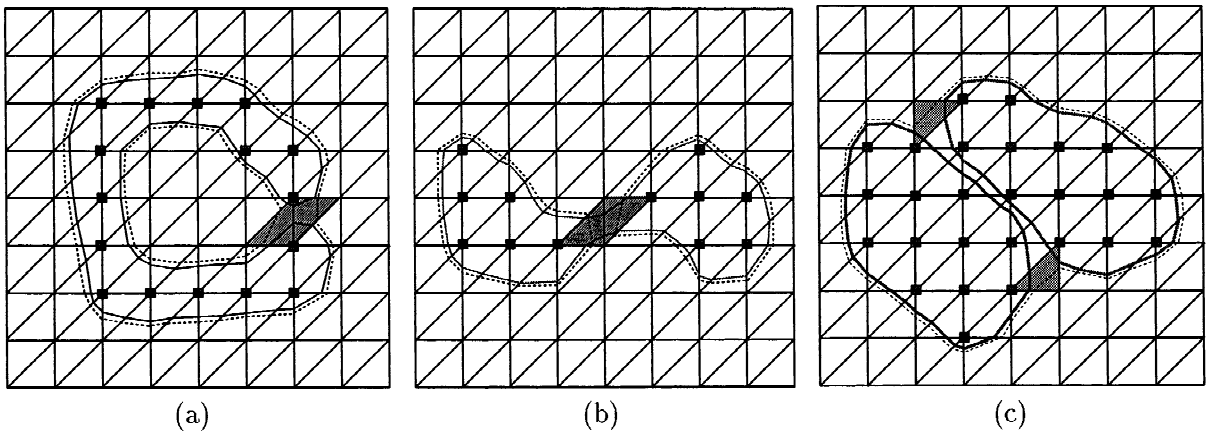

Fig. 8. Examples of T-snake topological transformations: (a) self-intersection, (b) shrinking and splitting, (c) merging. The resulting T-snake(s) after the transformations are shown as the dotted line(s). Node reconnections occur automatically in the shaded triangles so that inside and outside grid vertices are separated by a T-snake element.

II, we dequeue these vertices and check their corresponding grid cell edge data structures. If an intersection point is stored in an edge data structure of a grid vertex (indicating that the T-snake has moved such that the grid vertex is now inside) and the grid vertex is off, we turn it on. For all grid vertices that were turned on, we then use them as seed vertices in a standard region fill algorithm to turn on any neighboring vertices that are in the interior of the T-snake and are still off. A neighboring vertex is turned on if the path of grid edges connecting it to a seed vertex contains no intersection points (i.e., indicating that it is inside the T-snake). Phase II of the reparameterization process is simple, highly efficient and inherently parallel. Typically only a small number of grid vertices are enqueued in phase I for processing in phase II. Fig. 7(a,b) illustrates the second phase of the reparameterization process.

3.6. Topological transformations

When a T-snake collides with itself or with another T-snake, or when a T-snake breaks into two or more parts, a topological transformation must take place.[2] In order to effect consistent topological changes, consistent decisions must be made about disconnecting and reconnecting T-snake nodes. The grid and the reparameterization process provides us with an automatic and unambiguous mechanism to perform reconnections. By tracking the interior grid vertices (and hence tracking the boundary cells), adhering to the entropy condition, and re-establishing the correspondence of the model with the grid after a deformation step, we can always unambiguously determine the boundary or 'isocontour' of the new T-snake(s). We simply compute new T-snake elements from the signs of the grid vertices (i.e., whether they are off or on) in each boundary cell and from the intersection points computed in phase I such that the inside and outside grid vertices of these cells are separated by the T-snake element (Fig. 8). Thus, analogous to the evolving level set of an implicit function, the simplicial grid and the reparameterization process guarantees that topological transformations are handled automatically, consistently and efficiently.

3.7. The T-snake algorithm

Unlike the level-set evolution techniques which accede control to a higher dimensional implicit function, T-snakes retain an explicit parametric model formulation. The explicit formulation allows us to track and control the evolution of the T-snake. Consequently, reparameterizations can be performed very efficiently and constraints can be easily imposed on the model. The T-snake algorithm is as follows:

For each deformation step (M time steps):

1. For M time steps: (a) compute the external and internal forces acting on T-snake nodes and elements; and (b) update the node positions using (8).
2. Perform reparameterization phase I.
3. Perform reparameterization phase II.
4. For all current T-snake elements, determine if the T-snake element is still valid. A T-snake element is valid if its corresponding grid cell is still a boundary cell. Discard invalid T-snake elements and unused nodes.
5. Use the grid vertices turned on in phase II above (if any) to determine new boundary cells and hence new T-snake nodes and elements.

The T-snake is considered to have reached its equilibrium state when all of its elements have been inactive for a user-specified number of deformation steps. T-snake element activity or movement is measured via the grid, again using a flame propagation analogy. Model elements are assigned a 'temperature' based on the number of deforma-

[2] Another topological transformation can occur when a T-snake shrinks down to nothing and disappears. This property can be utilized in automatic segmentation scenarios (McInerney and Terzopoulos, 1995c).

tion steps during which the element (and its corresponding boundary cell) has remained valid. An element is considered inactive when its temperature falls below a pre-set 'freezing point'. Once a T-snake has reached equilibrium, the ACID grid can be deactivated and the model run as a standard parametric snake. The internal energy constraints will then create more evenly spaced model nodes.

4. Applying T-snakes to medical images

This section presents segmentation experiments using T-snakes. We have segmented from a variety of medical images a range of anatomic structures with complex shapes and topologies, demonstrating the usefulness of the T-snake approach. In most of the segmentation examples presented, the segmentations complete in under a second on an SGI workstation (with the exception of the segmentation of the white matter of the brain which requires about 3 s), demonstrating the efficiency of the T-snake reparameterization mechanism. All T-snake parameters are currently set manually by experimentation. This process is performed once for a specific image modality or for a specific anatomic structure and requires only a few minutes of experimentation. The time step Δt and deformation step M parameters are set to achieve maximum T-snake efficiency. The parameter settings are usually similar for a given image modality although the shape of the object often dictates the ratio of external forces to internal forces – a higher ratio is sometimes needed to force the T-snake into narrow protrusions.

4.1. Geometric flexibility

Using the ACID grid to reparameterize a T-snake at each iteration of the evolution process is a simple, elegant and automatic model subdivision technique. This process allows a T-snake to segment and reconstruct objects with significant protrusions, tubular objects, or objects with bifurcations (Fig. 9). Furthermore, the ACID grid parameterizes the fitted T-snake in terms of the intrinsic local object geometry. This property can significantly increase segmentation efficiency and reproducibility by making the T-snake much less sensitive to its initial placement than a standard parametric snake. A small T-snake may be seeded practically anywhere within a target object or a large T-snake may be initialized around the object – simpler and much less time consuming tasks than initializing a standard snake close to the target boundary – and still produce similar segmentations with similar intrinsic parameterizations (Fig. 10).[3]

4.2. Topological adaptability

As described in Section 3, the ACID grid provides a mathematically sound framework for robust topological transformations. This feature allows a T-snake to seamlessly split or merge and adapt to the topology of the target object (Figs. 11(a–d), 12 and 13). Topological adaptability combined with geometric flexibility can potentially significantly increase the automation of the segmentation process. Note that in the example showing the segmentation of the white matter in an MR image slice of the brain (Fig. 12), the T-snake is effectively acting as an active region growing model as discussed in Section 3.2.

4.3. Multiple T-snakes

Multiple T-snakes can be statically or dynamically created (or destroyed) (Fig. 14). This feature can be useful in several scenarios. Firstly, multiple T-snakes can evolve concurrently on different CPUs to improve segmentation efficiency. Secondly, users can seed T-snakes on several objects or on part of an object that is blocked by an edge to the object body. Finally, target objects can potentially be identified automatically, seeded with multiple T-snakes, and then automatically segmented.

4.4. Topology preservation

Rather than allow the T-snake to alter its topology,

[3] Noise and spurious image features can still affect T-snake behavior and, consequently, affect the result of the segmentation.

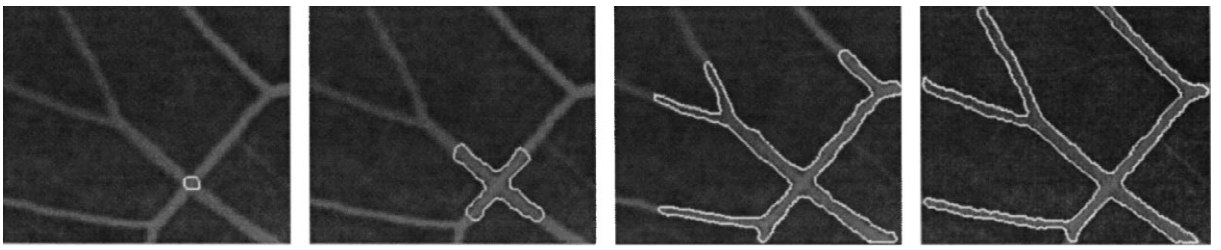

Fig. 9. Segmentation of the blood vessels in a clipped portion of retinal angiogram. The image sequence shows a snake flowing and branching along a vessel. A pixel-resolution ACID grid was used with model parameters: $p = 51.0$, $q = 50.0$, $a = 35.0$, $b = 5.0$, $\Delta t = 0.003$ and $M = 10$ (the number of time steps between reparameterization).

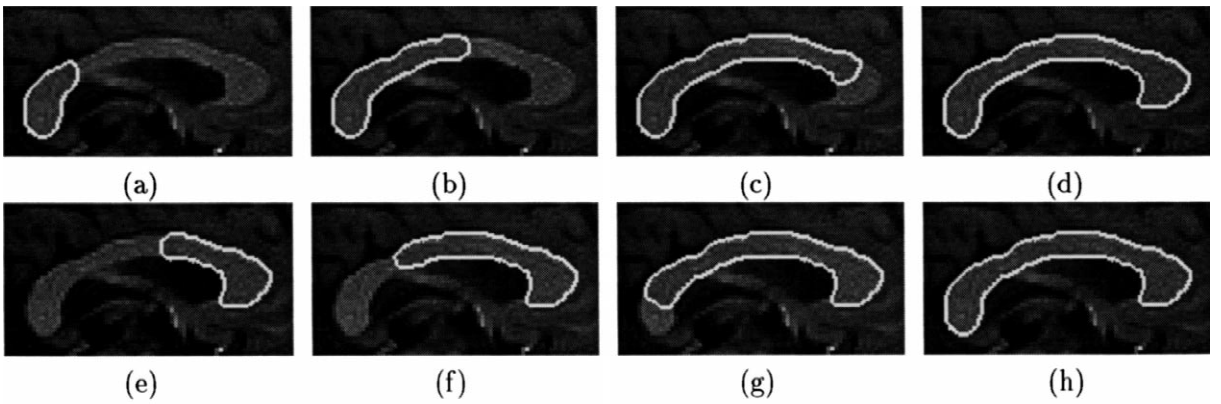

Fig. 10. Segmentation of the corpus callosum from MR brain image slice. The top sequence shows the T-snake initialized in the left part of the corpus callosum while the bottom sequence shows the T-snake initialized in the right part. Highly similar segmentation results are generated. A 128×128 squared cell ACID grid was used with model parameters: $p = 41.0$, $q = 40.0$, $a = 20.0$, $b = 20.0$, $\Delta t = 0.003$ and $M = 10$. Note that a user-defined barrier constraint was used to prevent the T-snake from leaking into the fornix.

another useful constraint that can be easily imposed is to ensure that the T-snake *maintains* its topology, guaranteeing that no self-intersections will occur (Fig. 15). This constraint can be used, for example, to ensure the correctness of object skeletonization; a promising technique for the analysis and interpretation of medical images. In standard parametric snake models, it is difficult to guarantee that no self-intersections of the fitted snake have occurred. Consequently, incorrect object topology can be generated during discretization. Providing to the thinning algorithms a smooth segmented object which is guaranteed to have the topology of the real object can potentially result in more robust skeletonizations.

The ACID framework provides a robust mechanism to identify and prevent self-intersections (and therefore possible topology changes) of the T-snake. The result is a simple, efficient technique for imposing a global topological constraint on a T-snake which guarantees that the topology of the fitted T-snake matches the topology of the real object. We implement the global topological constraint

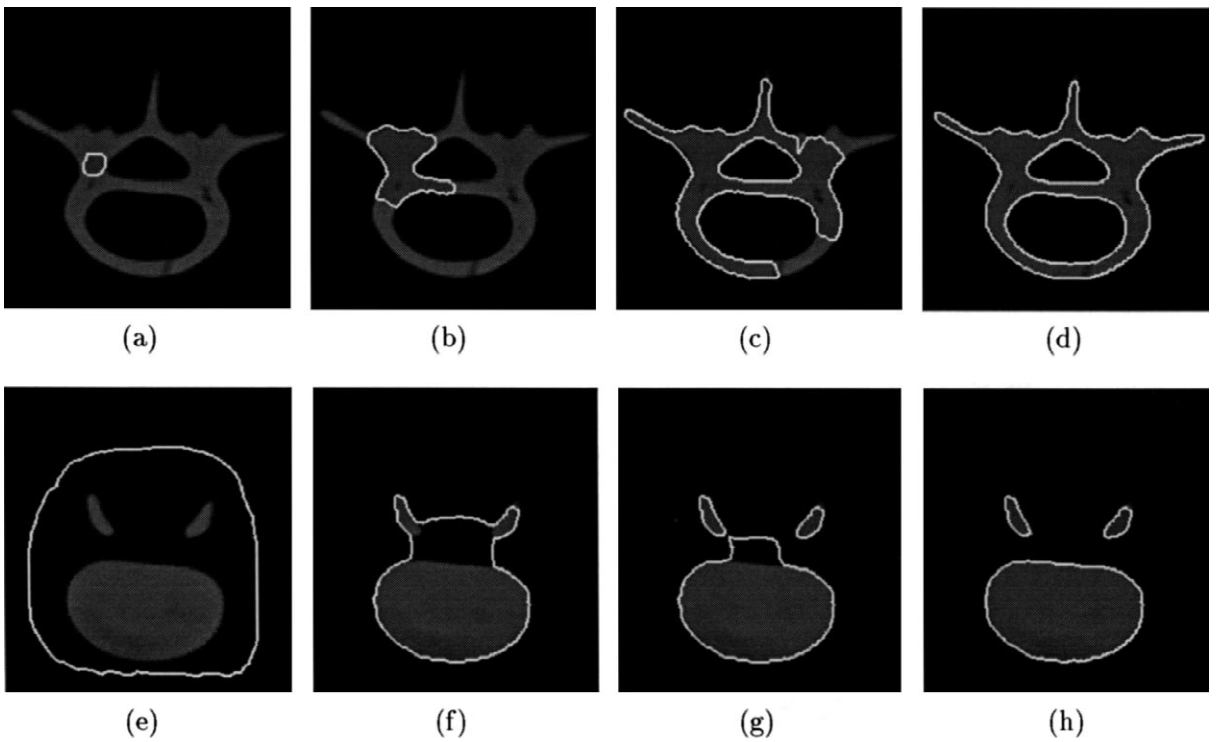

Fig. 11. Segmentation of two cross-sectional images of a human vertebra phantom illustrating the topological flexibility of T-snakes. 128×120 image slices from a CT image volume of a human vertebra phantom. A 50×50 squared cell ACID grid is used and model parameters: $p = 51.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.005$ and $M = 10$ (time steps per deformation step).

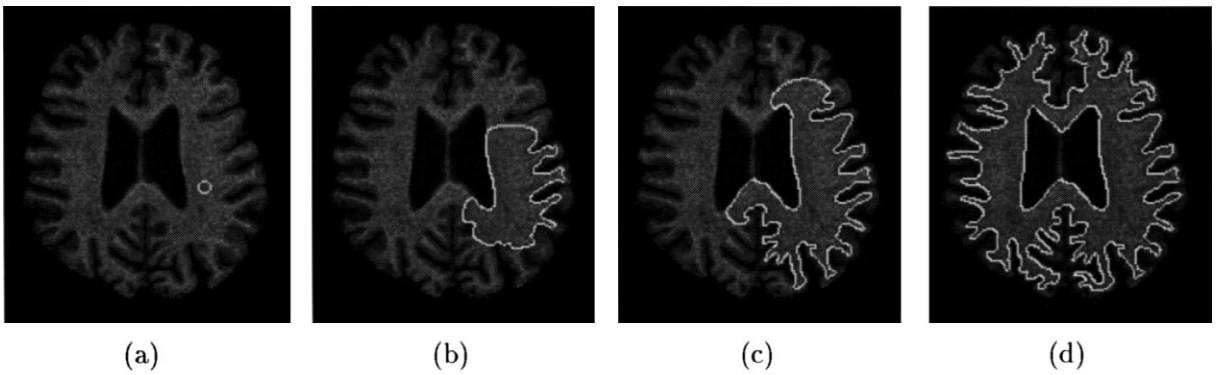

Fig. 12. T-snake used to segment gray-matter/white-matter interface in MR brain image slice. A pixel-resolution ACID grid was used with model parameters: $p = 0.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.003$ and $M = 10$. Statistics of white matter pixel intensity are used to weight the T-snake inflation force.

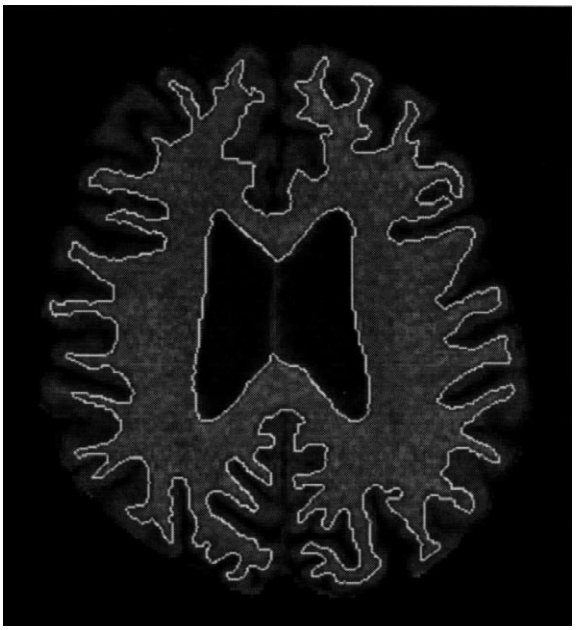

Fig. 13. Closeup of T-snake segmentation.

using topology-preserving T-snake deformations. A topology-preserving deformation is implemented by ensuring that no T-snake element outside of a small, contiguous local neighborhood of the element can turn on an ACID grid vertex that is close to this local T-snake neighborhood (McInerney, 1997). The T-snake has an initial topology of a circle and its evolution consists of a sequence of topology-preserving deformations. We are able to efficiently perform the topology-preserving deformations, since our model is defined in terms of a cell complex; labeling T-snake elements surrounding an ACID grid vertex as 'local' or 'non-local' is a simple operation.

4.5. T-snake element cooling process

We have incorporated an element 'cooling' process into the T-snake implementation. As described in Section 3.7, T-snake elements are assigned a temperature based on the number of deformation steps the element (and its corresponding boundary grid cell) has remained valid. The temperature attribute provides a measure of element activity or movement. When the temperature of an element falls below a user-set freezing point, the element is removed from the computational process and stored in a table. This adjustable mechanism allows the system to maintain a small, manageable computational burden for many segmentation scenarios. Fig. 16 illustrates the cooling process. As the T-snake flows along the object, only a few elements are active (hot) at one time. The remaining elements are inactive (frozen) and do not contribute to the computational load.

4.6. Interactive control

The ACID framework allows T-snakes to maintain the

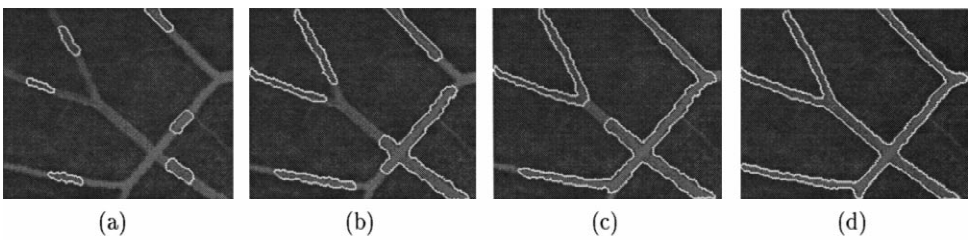

Fig. 14. Segmentation of the blood vessels in angiogram of retina using using multiple T-snakes.

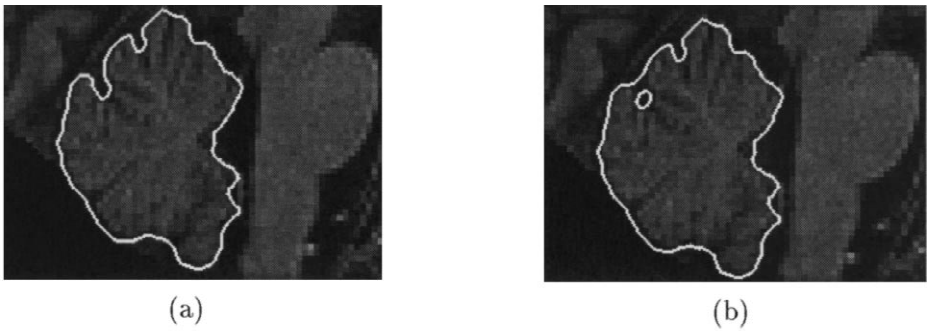

Fig. 15. Segmentation of cerebellum: (a) with topology preservation constraint, (b) without topology preservation constraint.

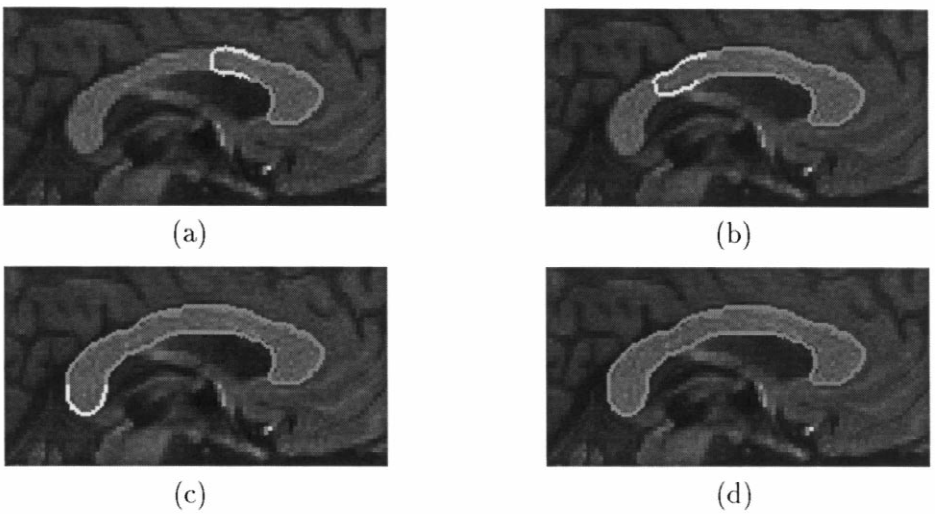

Fig. 16. Segmentation of the corpus callosum incorporating the element cooling process. The active portions of the T-snake are shown as white while the frozen elements are shown as gray.

intuitive interactive capabilities associated with standard snakes. Users can exert attraction or repulsion forces using mouse-controlled springs, anchored springs, 'volcanos', 'magnets', etc. (Kass et al., 1988). For example, Fig. 17 shows a segment of a T-snake being pulled by a mouse-controlled spring force into the correct position. Fig. 18(a,b) shows the same segment being pulled close enough to the opposite side of the T-snake such that a topological transformation takes place and the T-snake breaks into two parts. Fig. 18(c,d) shows the reverse of this process. One possibility for preventing accidental topology changes is to convert the T-snake to a standard parametric snake by simply discarding the ACID grid thus disabling the reparameterization process. This conversion can be performed at any time during the T-snake evolution. Conversely, a closed snake can be converted back to a T-snake at any time during its evolution.

Other useful T-snake interaction mechanisms are geometric point constraints. Geometric point constraints are incorporated into the T-snakes model either as soft

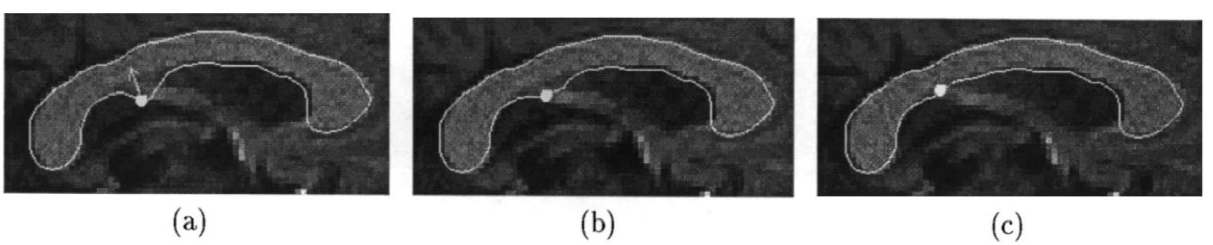

Fig. 17. T-snake segment (indicated by the circle) interactively pulled into the correct position by a mouse-controlled spring force.

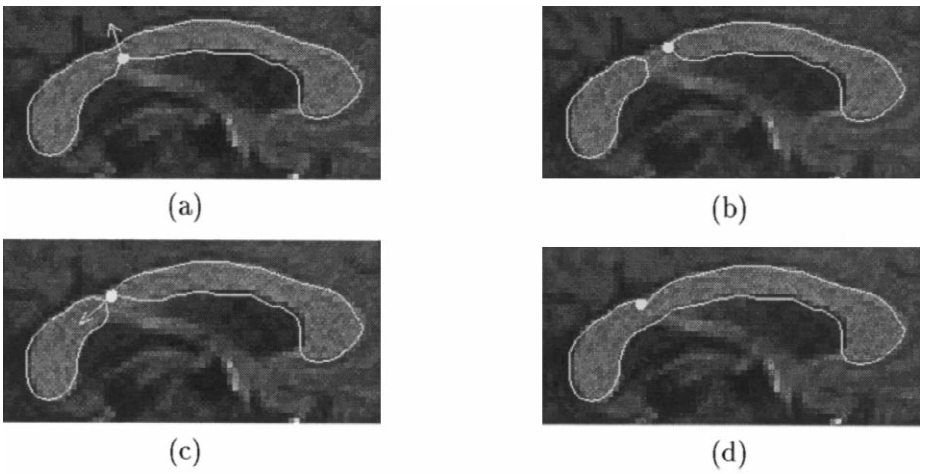

Fig. 18. T-snake segment interactively pulled until topological transformation occurs. In (a,b) the T-snake has broken into two parts. In (c,d) the T-snake has been 'mended'.

constraints to be satisfied approximately or as hard constraints that must never be violated. Soft constraints are incorporated into the T-snakes physics-based formulation as force functions. The anchored spring force is an example of a soft point constraint.

In the following experiments, we demonstrate the use of geometric point constraints in an interactive segmentation scenario. Both 'soft' and 'hard' constraints are employed to segment neuronal cells in an EM photomicrograph. We begin by defining a series of points on the cell boundary, forming a closed polygon from these points and converting the polygon into an initial T-snake (Fig. 19(a)). The T-snake deforms and localizes the cell boundary (Fig. 19(b)) while being constrained to pass through the initial user-

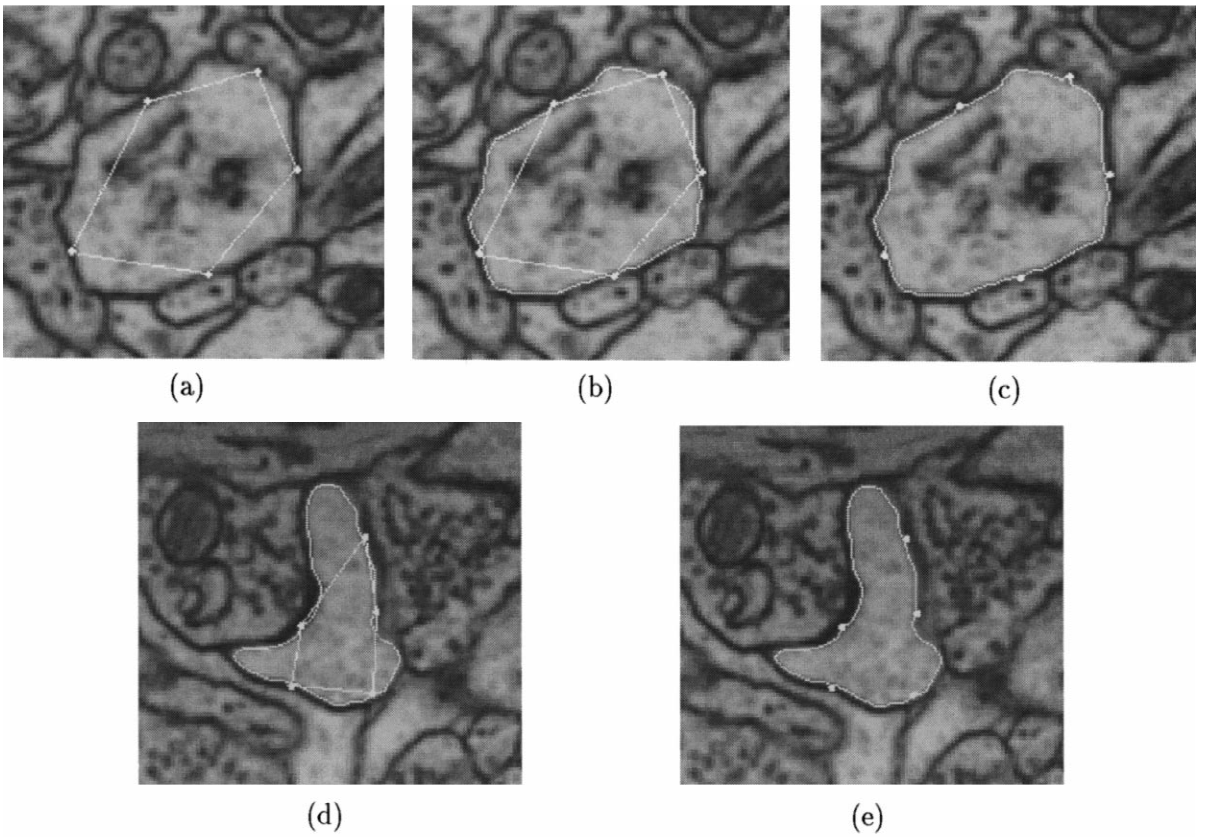

Fig. 19. T-snake (derived from closed polygon) used to segment neuronal cells. The T-snake is constrained to pass through the initial user-specified points.

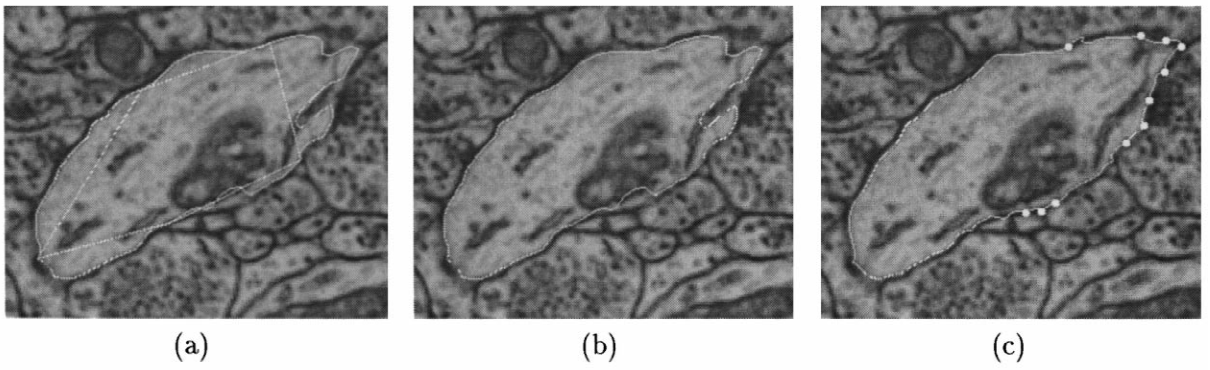

Fig. 20. (a,b) T-snake formed from user-specified closed polygon deforms to segment neuronal cell. (c) Soft point constraints are used to pull T-snake off spurious edges.

defined points (Fig. 19(c)). A second example is shown in Fig. 19(d,e). In a third experiment we make use of soft geometric point constraints in the form of anchored springs. The springs exert forces on T-snake nodes that are within a specified neighborhood of the springs. In Fig. 20(a,b) the initialization and result of the T-snake segmentation without spring constraints is shown. The T-snake has attached itself to several interior parts of the cell. Fig. 20(c) shows the addition of spring constraints (the springs were added dynamically) which result in a correct segmentation of the cell boundary. All experiments use a 160×122 cell ACID grid on the 640×488 pixel image and model parameters: $p = 51.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.005$ and $M = 10$.

Other useful hard geometric constraints include barriers or 'forbidden zones' which can be used to force a T-snake to take on the shape of the impacted region of the circle or ellipse, allowing the user to 'shore-up' object boundary sections with sparse edges (this constraint was used in Fig. 10 to prevent the T-snake from 'leaking' into a neighboring structure (the fornix)).

As the above experiments have demonstrated, T-snakes may be easily initialized with a few points such that they are close to the target object boundary. However, T-snakes (and the level-set techniques) are essentially region-based. They are best suited for objects where the image intensity is homogeneous or of a consistent texture within the object. Standard snakes or interactive semi-automatic tracing tools such as intelligent scissors (Mortensen and Barrett, 1995; Barrett and Mortensen, 1997) may be more appropriate for certain non-homogeneous images.

4.7. Constraints based on image statistics

Section 3.2 described an image intensity statistics-weighted inflation force that can be used along with image edge forces, allowing a T-snake to effectively integrate edge information with region information. The integration of region and edge information can sometimes be effective in preventing a T-snake from leaking into regions surrounding the target object. Fig. 21(a) shows a CT image slice of a canine heart. The bright region is the left ventricle (LV). Note that in this particular slice the LV intensity diminishes considerably in some regions, resulting in very weak edges (Fig. 21(b)). Fig. 22(a–d) shows a T-snake segmenting the LV using the statistically weighted inflation force defined in (5).

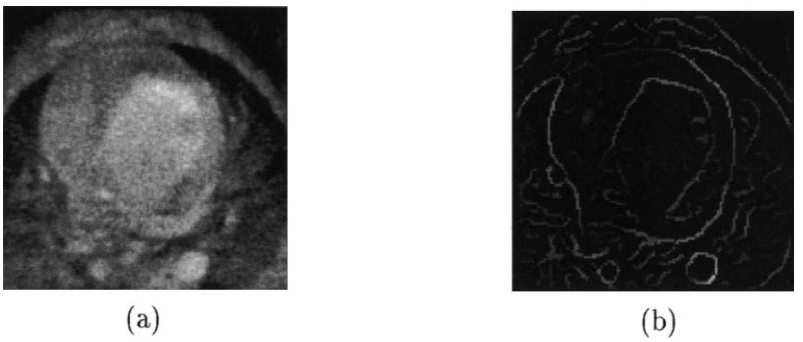

Fig. 21. CT image slice of LV and edge-detected version.

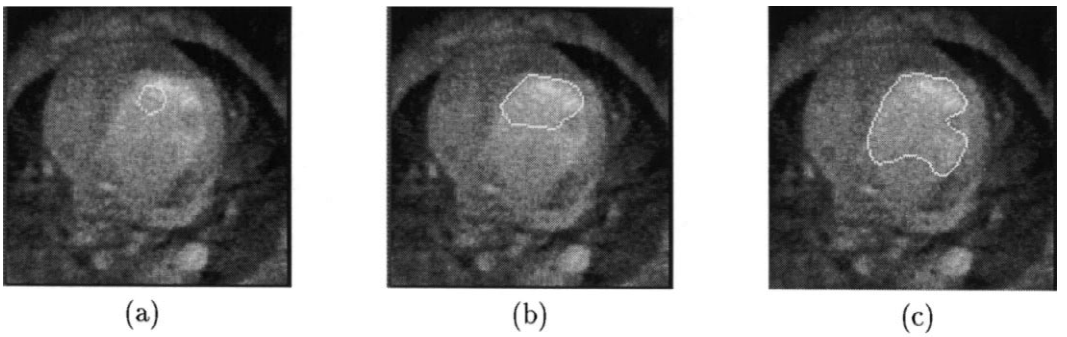

Fig. 22. T-snake segmenting LV image slice using statistically weighted inflation force and image edge forces.

5. 3D extension: T-surfaces

We have extended the ACID framework to three dimensions using tetrahedral (or hexahedral cells combined with a disambiguation scheme) to create topology adaptive surfaces (T-surfaces) – a discrete deformable closed-surface model that is a 3D generalization of T-snakes (McInerney and Terzopoulos, 1999). T-surfaces share all of the features and properties of their 2D counterpart. A T-surface is represented as a closed oriented triangular surface mesh and, analogous to the T-snake, it is a discrete form of a standard parametric deformable surface model (McInerney and Terzopoulos, 1995a; Terzopoulos et al., 1988). As in the T-snake, we associate with the model nodes time varying positions $\boldsymbol{x}_i(t) = [x_i(t),\ y_i(t),\ z_i(t)]$, along with tensile forces, flexural forces, inflationary forces, and external forces. The behavior of the T-surface is governed by a 3D form of (1).

The T-surface reparameterization process is analogous to the T-snakes case. In phase I, ACID grid edge intersection points and their signs are computed for each model element (triangle) and potential inside grid vertices are enqueued. In phase II, inside grid vertices are turned on and new model nodes are computed using the intersection points from phase I. Old model elements and unused model nodes are discarded. Topological transformations are again handled as in the T-snakes case. That is, by tracking the interior grid vertices (and hence tracking the

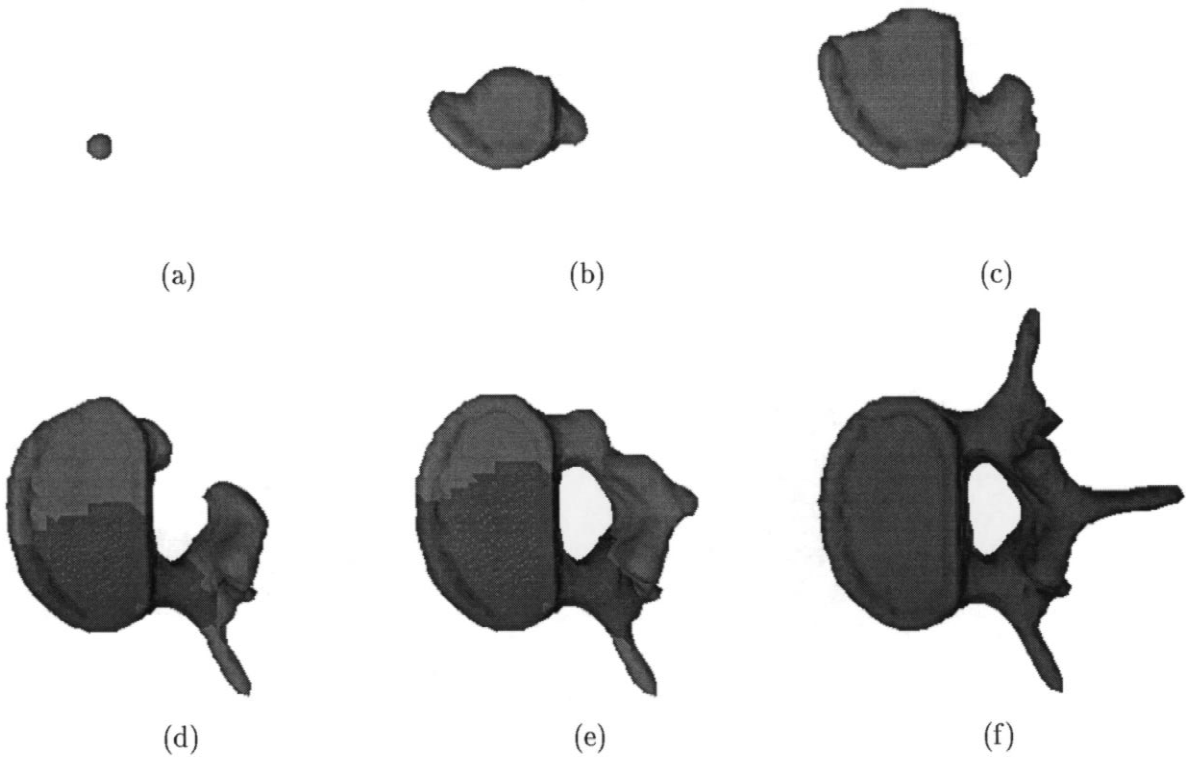

Fig. 23. T-surface segmenting vertebra phantom from CT volume image.

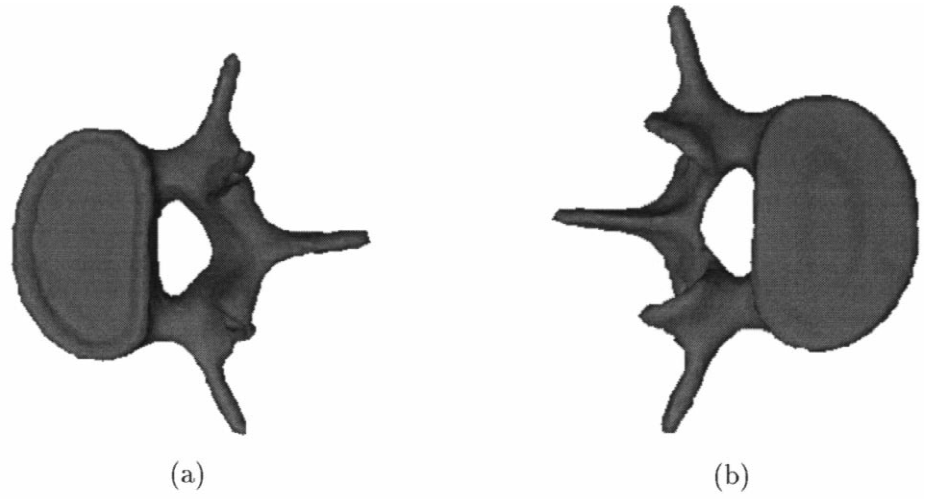

Fig. 24. Two views of the fitted T-surface.

boundary tetrahedra) and re-establishing the correspondence of the T-surface with the grid after each deformation step, the boundary or 'isosurface' of the new model is uniquely determined.

We have applied T-surfaces to a variety of medical images in order to segment anatomic structures with complex shapes and topologies. Fig. 23 presents a representative example where we apply a T-surface to a $120 \times 128 \times 52$ CT image volume of a human vertebra phantom, which demonstrates the topological adaptability of the 3D model. We use a $32 \times 30 \times 13$ cell ACID grid (where each cubical cell is divided into six tetrahedra). Fig. 24 shows two views of the final result after conversion to a standard deformable surface. Fig. 25 shows several cross-sections of the fitted model.

Full details and additional results are described in (McInerney and Terzopoulos, 1999).

6. Conclusion

Deformable models overcome many of the limitations of traditional image segmentation techniques by exploiting analytical representations of object shape, by incorporating a priori knowledge, and by providing intuitive interactive capabilities. A challenge has been to increase the degree of automation of deformable models, reducing the labor-intensive aspects of interactive anatomic structure segmentation to a basic minimum, while maximizing segmentation accuracy and reproducibility. In an effort to meet this challenge, we have introduced an affine cell image decomposition (ACID) framework for deformable models. In this paper, we have developed our approach in terms of a novel deformable contour model that we call T-snakes.

The ACID framework provides an elegant mechanism

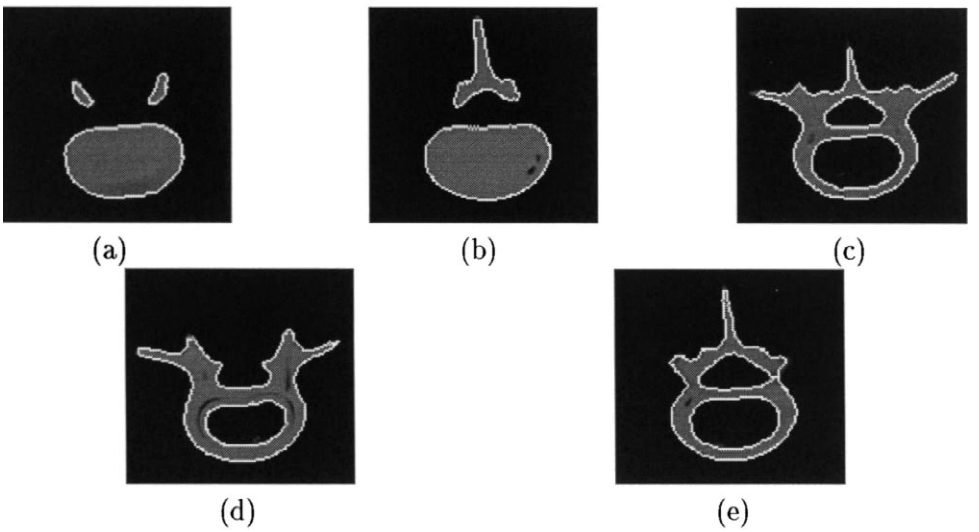

Fig. 25. Several cross-sections of T-surface overlaid on CT vertebra image volume slices.

for contour reparameterization, allowing T-snakes to segment and reconstruct objects with significant protrusions, tubular objects, or objects with bifurcations. The ACID grid also provides a mathematically sound basis for robust topological transformations. This property enables T-snakes to seamlessly split or merge, adapting to the topology of a target object. The ACID reparameterization mechanism makes the model relatively insensitive to initial placement, significantly improving the efficiency and automation of the segmentation process.

The advantage of retaining an explicit active contour formulation as opposed to an implicit, level-set formulation is that constraints expressed as energy or force functions can easily be incorporated into the energy-minimization framework. Constraints may be introduced either as soft constraints to be satisfied approximately or as hard constraints that must never be violated.

Using the ACID framework, contour evolution is performed by tracking and recording the interior region of a T-snake as it expands or contracts under the influence of pressure forces. The strength of these pressure forces can be directly linked to the local or global statistics of the image pixel intensity values of the target object. These statistical constraints allow ACID-based models to behave as active region growing models that can effectively integrate edge information with region-based information.

We have demonstrated the power and flexibility of ACID-based deformable models to segment complex objects from medical images in a convenient, efficient, and highly automated manner.

Acknowledgements

The CT heart images used in this paper were provided courtesy of Dr. Eric Hoffman of the University of Pennsylvania Medical School and were redistributed to us by Dr. Dmitry Goldgof, CSE Department, University of South Florida. The EM neuronal tissue images were provided courtesy of Dr. Kristen Harris of the Harvard Medical School.

Appendix A. Review of parametric snakes

This appendix briefly reviews parametric snakes models. The reader is referred to (Kass et al., 1988; McInerney, 1997) for additional details.

Snakes models are typically defined as a time-varying, parametric contour $\boldsymbol{v}(s, t) = (x(s, t), y(s, t))^{\mathrm{T}}$ in the image plane $(x, y) \in \Re^2$, where the contour coordinates $x(s, t)$ and $y(s, t)$ are functions of the parametric variable $s \in [0, 1]$ and time t. Boundary conditions may be used to specify the topology of the contour. In particular, free boundary conditions specify an open curve, whereas the periodic boundary condition $\boldsymbol{v}(0, t) = \boldsymbol{v}(1, t)$ produces a closed snake.

The shape of the contour is dictated by the energy functional

$$\mathscr{E}(\boldsymbol{v}) = \mathscr{S}(\boldsymbol{v}) + \mathscr{P}(\boldsymbol{v}), \tag{A.1}$$

and the final shape and position of the contour corresponds to a local minimum of $\mathscr{E}$. The first term of the functional

$$\mathscr{S}(\boldsymbol{v}) = \int_0^1 w_1(s) \left| \frac{\partial \boldsymbol{v}}{\partial s} \right|^2 + w_2(s) \left| \frac{\partial^2 \boldsymbol{v}}{\partial s^2} \right|^2 \mathrm{d}s \tag{A.2}$$

is the internal deformation energy, characterizing the deformation of an elastic contour. Two physical parameter functions $w_1(s)$ and $w_2(s)$ determine the extent to which the snake can stretch or bend at any parametric coordinate s. The second term in (A.1) consists of external energy potentials that couple the snake to the image $I(x, y)$ and support user interaction. Traditionally,

$$\mathscr{P}(\boldsymbol{v}) = \int_0^1 P(\boldsymbol{v}(s, t))\, \mathrm{d}s, \tag{A.3}$$

where $P(x, y)$ denotes a scalar potential function defined on the image plane. To couple snakes to images, external potentials are designed whose local minima coincide with intensity extrema, edges, and other image features of interest. For example, the contour will be attracted to intensity edges in $I(x, y)$ by choosing a potential

$$P(x, y) = -c \left\| \nabla [G_\sigma * I(x, y)] \right\|, \tag{A.4}$$

where G_σ denotes a Gaussian smoothing filter of standard deviation σ, and c scales the potential.

Few medical images lend themselves to fully automated processing with completely satisfactory results; hence, it is often essential for a user to be able interactively to control the segmentation process. Furthermore, the equilibrium state of the snake will, in general, represent a single local minimum energy solution out of numerous possible local minima. The user must have the ability to pull the snake out of one local minimum into another. This ability is conveniently realized in the energy minimization framework through the use of user-defined external constraint potentials such as interactive springs, anchored springs, and 'volcanos' (Kass et al., 1988).

While it is natural to view energy minimization as a static problem, a potent approach to computing the local minima of a functional such as (A.1) is to construct a dynamical system that is governed by the functional and allow the system to evolve to equilibrium. The dynamical system may be constructed by applying Lagrangian mechanics. A simple example is a dynamic snake with a mass density $\mu(s)$ and a damping density $\gamma(s)$. The Lagrange equations of motion for a snake with the internal energy (A.2) and external energy (A.3) is

$$\mu\frac{\partial^2 \boldsymbol{v}}{\partial t^2} + \gamma\frac{\partial \boldsymbol{v}}{\partial t} - \frac{\partial}{\partial s}\left(w_1\frac{\partial \boldsymbol{v}}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2\frac{\partial^2 \boldsymbol{v}}{\partial s^2}\right) = -\nabla P(\boldsymbol{v}(s,t)). \quad \text{(A.5)}$$

The first two terms on the left hand side of this partial differential equation represent inertial and damping forces, the next two terms represent the internal tensile and flexural forces, while the right hand side represents the external forces. Equilibrium is achieved when the internal and external forces balance and the contour comes to rest (i.e., $\partial \boldsymbol{v}/\partial t = \partial^2 \boldsymbol{v}/\partial t^2 = 0$).

In shape recovery problems not involving time-varying data, the mass density μ is often set to zero, resulting in simplified equations of motion and a snake that comes to rest as soon as the internal forces balance the external forces. We can solve the resulting first-order dynamics equation by representing the continuous geometric model $\boldsymbol{v}$ in terms of linear combinations of local-support or global-support basis functions, such as finite elements or finite differences, to produce a discretized version of the equation. We can then use a semi-implicit first-order Euler method to integrate the system forward through time (Kass et al., 1988).

References

Allgower, E.L., Georg, K., 1990. Introduction to Numerical Continuation Methods, Springer, Berlin, Heidelberg.

Barrett, W.A., Mortensen, E.N., 1997. Interactive live-wire boundary extraction. Medical Image Analysis 1 (4), 331–341.

Borgefors, G., 1984. Distance transformations in arbitrary dimensions. Comput. Vis. Graphics Image Processing 27, 321–345.

Caselles, V., Catte, F., Coll, T., Dibos, F., 1993. A geometric model for active contours. Numerische Mathematik 66.

Caselles, V., Kimmel, R., Sapiro, G., 1995. Geodesic active contours. In: Proc. Fifth International Conf. on Computer Vision (ICCV'95), Cambridge, MA, June, 1995. IEEE Computer Society Press, Los Alamitos, CA, pp. 694–699.

Chakraborty, A., Duncan, J.S., 1995. Integration of boundary finding and region-based segmentation using game theory. In: Bizais, Y., Barillot, C., Paola, R. Di, (Eds.), Information Processing in Medical Imaging: Proc. 14th Int. Conf. (IPMI'95), Ile de Berder, France, June, 1995. Computational Imaging and Vision, Vol. 3. Kluwer, Dordrecht, The Netherlands, pp. 189–200.

Chakraborty, A., Staib, L.H., Duncan, J.S., 1994. Deformable boundary finding influenced by region homogeneity. In: Proc. Conf. Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA, June, 1994. IEEE Computer Society Press, Los Alamitos, CA, pp. 624–627.

Cohen, L.D., Cohen, I., 1993. Finite element methods for active contour models and balloons for 2D and 3D images. IEEE Trans. Pattern Anal. Machine Intell. 15 (11), 1131–1147.

Durikovic, R., Kaneda, K., Yamashita, H., 1995. Dynamic contour: a texture approach and contour operations. Visual Computer 11, 277–289.

Gauch, J.M., Pien, H.H., Shah, J., 1994. Hybrid boundary-based and region-based deformable models for biomedical image segmentation. In: Mathematical Methods in Medical Imaging III. Vol. 2299 of SPIE Proc. SPIE, San Diego, CA, pp. 72–83.

Herlin, I.L., Nguyen, C., Graffigne, C., 1992. A deformable region model using stochastic processes applied to echocardiographic images. In: Proc. Conf. Computer Vision and Pattern Recognition (CVPR'92), Urbana, IL, June, 1992. IEEE Computer Society Press, Los Alamitos, CA, pp. 534–539.

Ivins, J., Porrill, J., 1994. Statistical snakes: Active region models. In: Proc. 5th British Machine Vision Conf. (BMVC'94). BMVA Press, pp. 377–386.

Kapur, T., Grimson, W.E.L., Wells III, W.M., 1996. Kikinis, R., 1996. Segmentation of brain tissue from magnetic resonance images. Med. Image Anal. 1 (2), 109–127.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. Int. J. Comput. Vis. 1 (4), 321–331.

Kovalevsky, V.A., 1989. Finite topology as applied to image analysis. Comput. Vis. Graphics Image Processing 46, 141–161.

Lachaud, J.-O., Montanvert, A., 1999. Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction. Med. Image Anal. 3 (2), 187–207.

Leitner, F., Cinquin, P., 1991. Complex topology 3D objects segmentation. In: Model-Based Vision Development and Tools. Vol. 1609 of SPIE Proc. SPIE, Bellingham, WA, pp. 16–26.

Lobregt, S., Viergever, M.A., 1995. A discrete dynamic contour model. IEEE Trans. Med. Imaging 14 (1), 12–24.

Lorenson, W.E., Cline, H.E., 1987. Marching cubes, a high resolution 3D surface construction algorithm. Comput. Graphics 21 (4), 163–169.

Lorigo, L., Faugeras, O., Grimson, W.E.L., Keriven, R., Kikinis, R., 1998. Segmentation of bone in clinical knee MRI using texture-based geodesic active contours. In: Wells, W., Colchester, A., Delps, S. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI'98. Vol. 1496 of Lectures Notes in Computer Science. Springer, Berlin, pp. 1195–1204.

Malladi, R., Sethian, J., Vemuri, B.C., 1995. Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Anal. Machine Intell. 17 (2), 158–175.

Malladi, R., Kimmel, R., Adalsteinsson, D., Sapiro, G., Caselles, V., Sethian, J.A., 1996. A geometric approach to segmentation and analysis of 3D medical images. In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, San Francisco, CA, June 1996. IEEE Computer Society Press, Los Alamitos, CA, pp. 244–252.

McInerney, T., 1997. Topologically Adaptable Deformable Models for Medical Image Analysis. Ph.D. Dissertation. Department of Computer Science, University of Toronto, Toronto, ON, Canada.

McInerney, T., Terzopoulos, D., 1995a. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. Comput. Med. Imaging Graphics 19 (1), 69–83.

McInerney, T., Terzopoulos, D., 1995b. Medical image segmentation using topologically adaptable snakes. In: Ayache, N. (Ed.), Proc. First International Conf. on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), Nice, France, April 1995. Vol. 905 of Lectures Notes in Computer Science. Springer, Berlin, Germany, pp. 92–101.

McInerney, T., Terzopoulos, D., 1995c. Topologically adaptable snakes. In: Proc. Fifth International Conf. on Computer Vision (ICCV'95), Cambridge, MA, June 1995. IEEE Computer Society Press, Los Alamitos, CA, pp. 840–845.

McInerney, T., Terzopoulos, D., 1996. Deformable models in medical image analysis: a survey. Med. Image Anal. 1 (2), 91–108.

McInerney, T., Terzopoulos, D., 1999. Topology adaptive deformable surfaces for medical image volume segmentation. IEEE Trans. Med. Imaging 18 (10), 840–850.

Mortensen, E.N., Barrett, W.A., 1995. Intelligent scissors for image composition. In: Proc. SIGGRAPH'95, Los Angeles, CA, August 1995. Computer Graphics Proc., Annual Conf. Series 1995. ACM SIGGRAPH, New York, NY, pp. 191–198.

Munkres, J.R., 1984. Elements of Algebraic Topology, Addison-Wesley, Menlo Park, CA.

Ning, P., Bloomenthal, J., 1993. An evaluation of implicit surface tilers. IEEE Comput. Graphics Appl. 33–41.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulation. J. Comput. Phys. 79, 12–49.

Poon, C.S., Braun, M., Fahrig, R., Ginige, A., Dorrell, A., 1994. Segmentation of medical images using an active contour model incorporating region-based images features. In: Robb, R.A. (Ed.), Proc. Third Conf. on Visualization in Biomedical Computing (VBC'94), Rochester, MN, October 1994. Vol. 2359 of SPIE Proc. SPIE, Bellingham, WA, pp. 90–97.

Prince, J.L., Xu, C., 1996. A new external force model for snakes. In: 1996 Image and Multidimensional Signal Processing Workshop, pp. 30–31.

Rougon, N., Prêteux, F., 1991. Deformable markers: Mathematical morphology for active contour models control. In: Image Algebra and Morphological Image Processing II. Vol. 1568 of SPIE Proc. SPIE, Bellingham, WA, pp. 78–89.

Sapiro, G., Kimmel, R., Caselles, V., 1995. Object detection and measurements in medical images via geodesic deformable contours. In: Vision Geometry IV. Vol. 2573 of SPIE Proc. SPIE, Bellingham, WA, pp. 366–378.

Singh, A., Goldgof, D., Terzopoulos, D., 1998. In: Deformable Models in Medical Image Analysis, IEEE Computer Society, Los Alamitos, CA.

Szeliski, R., Tonnesen, D., Terzopoulos, D., 1993. Modeling surfaces of arbitrary topology with dynamic particles. In: Proc. Conf. Computer Vision and Pattern Recognition (CVPR'93), New York, NY, June 1993. IEEE Computer Society Press, Los Alamitos, CA, pp. 82–87.

Terzopoulos, D., Witkin, A., Kass, M., 1988. Constraints on deformable models: recovering 3D shape and nonrigid motion. Artif. Intell. 36 (1), 91–123.

Whitaker, R., 1994. Volumetric deformable models. In: Robb, R.A. (Ed.), Proc. Third Conf. on Visualization in Biomedical Computing (VBC'94), Rochester, MN, October 1994. Vol. 2359 of SPIE Proc. SPIE, Bellingham, WA.