# Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation

Tim McInerney,* *Member, IEEE*, and Demetri Terzopoulos, *Member, IEEE*

***Abstract*—Deformable models, which include deformable contours (the popular snakes) and deformable surfaces, are a powerful model-based medical image analysis technique. We develop a new class of deformable models by formulating deformable surfaces in terms of an affine cell image decomposition (ACID). Our approach significantly extends standard deformable surfaces, while retaining their interactivity and other desirable properties. In particular, the ACID induces an efficient reparameterization mechanism that enables parametric deformable surfaces to evolve into complex geometries, even modifying their topology as necessary. We demonstrate that our new ACID-based deformable surfaces, dubbed T-surfaces, can effectively segment complex anatomic structures from medical volume images.**

***Index Terms*— Deformable models, deformable surfaces, segmentation.**

## I. Introduction

The imperfections typical of medical images, such as partial volume averaging, intensity inhomogeneities, limited resolution, and imaging noise, often cause the apparent boundaries of anatomic structures to be indistinct and disconnected. The challenge of segmentation and reconstruction is to identify boundary elements belonging to an anatomic structure of interest and integrate them into a complete and consistent geometric model of that structure. This important task is impeded by the complexity and variability of biological shape and by the sheer size of modern volume images. Yet segmentation and reconstruction must be performed with efficiency and reproducibility and, preferably, with minimal user interaction.

Deformable models [1], which include the popular deformable contours, or snakes [2], and deformable surfaces [1], [3]–[5] are a powerful segmentation technique designed to meet this challenge (see the recent survey [6], which also appears in the compilation [7]). Deformable models offer an attractive approach to the medical image segmentation problem because they combine many desirable features. These include compact analytic representation of object shape and motion, inherent connectivity and smoothness that counteracts noise and boundary irregularities, and the ability to incorporate *a priori* anatomic knowledge. Addressing the fact that few medical images lend themselves to fully automatic analysis with satisfactory results, deformable models furthermore afford the often times essential option of interactive control over the segmentation process. They support interaction mechanisms via an intuitive physics-based paradigm: users can push, pull, anchor, reposition, and otherwise manipulate a deformable model using a mouse or other input devices.

While they have proven very useful in medical image analysis, standard deformable models can be improved. First, they were designed to be manually initialized reasonably close to the boundaries of a target object. It would be helpful to ameliorate the sometimes tedious initialization process by addressing the initialization sensitivity problem. Second, the fixed parameterization of the standard, parametric deformable models in concert with their internal energy constraints can limit geometric flexibility. Hence, they can exhibit reluctance to deform into long tube-like shapes and shapes with significant protrusions or concavities. Third, the topology of the object of interest must be known in advance, since parametric deformable models are incapable of topological transformations without additional machinery.

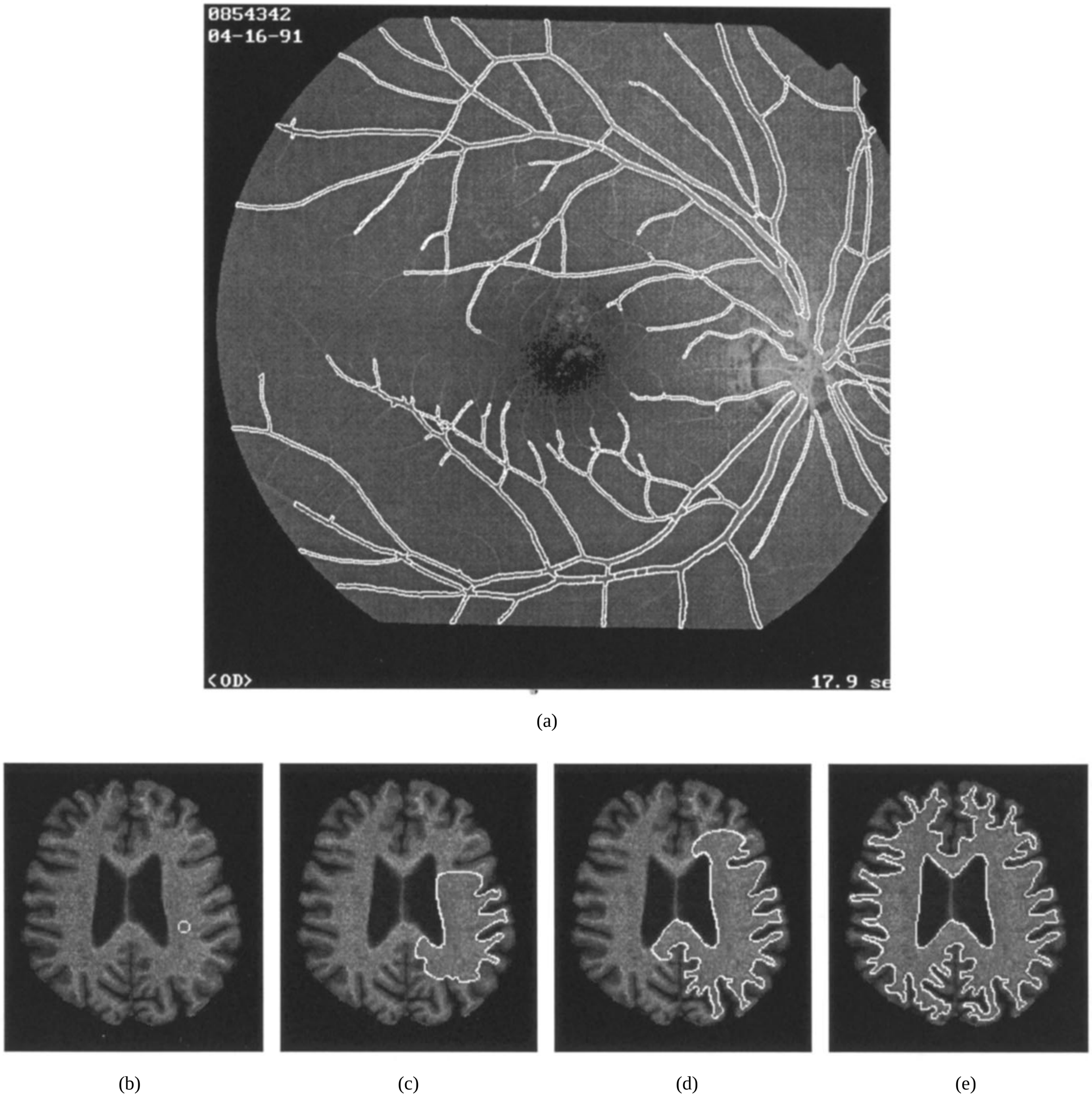
It has been our goal to develop a unified framework that overcomes the limitations of standard deformable models, while retaining their traditional strengths. To this end, we recently introduced a new class of deformable contour models called topology adaptive snakes (T-snakes) [8], [9]. T-snakes exploit an affine cell image decomposition (ACID), a theoretically sound framework that significantly extends the abilities of standard parametric snakes. In particular, the ACID induces an efficient reparameterization mechanism that enables deformable contours to flow or grow into complex geometries, even modifying their topology as necessary. We can initialize a T-snake as a small seed contour anywhere within a target object. The T-snake can dynamically adapt its topology to that of the target object, flow around objects embedded within the target object, and/or automatically merge with other models interactively introduced by the user. Thus, immersing discrete parametric snakes in an ACID enables them to segment and reconstruct even the most complex-shaped biological structures, with a high degree of automation, efficiency, and reproducibility in many medical image analysis scenarios. Finally, an important feature of the ACID framework is that it does not interfere with the physics-based formulation of standard snakes, preserving the intuitive user interaction

Manuscript received March 15, 1999; revised August 9, 1999. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the Information Technologies Research Center of Ontario. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was M. Vannier. *Asterisk indicates corresponding author.*

*T. McInerney is with the Department of Mathematics, Physics, and Computer Science, Ryerson Polytechnic University, Toronto Ont, Canada M5B 2K3.

D. Terzopoulos is with the Department of Computer Science, University of Toronto, Toronto Ont, Canada M5S 3G4.

Publisher Item Identifier S 0278-0062(99)09630-5.



(a)

(b) (c) (d) (e)

Fig. 1. Segmentation with T-snakes. (a) T-snakes segmenting blood vessels in a retinal angiogram. Geometric flexibility allows the T-snakes to grow into the complex vessel shapes. (b)–(e) T-snake segmenting gray-matter/white-matter interface and ventricles in an MR brain image slice. The initially circular T-snake (b) changes its topology to a highly deformed annular region (e).

mechanism and the ability to incorporate constraints through energy or force functions.

While deformable contours, in general, and T-snakes, in particular, have proved to be a successful boundary integration and feature extraction technique, their two-dimensional (2-D) formulation limits their ability to process three-dimensional (3-D) image data. Although slice-to-slice contour propagation [10], [11] can improve efficiency and increase automation, there are many segmentation scenarios where this technique is not effective. Three dimensional deformable surfaces or balloons, on the other hand, are potentially faster, make more effective use of the 3-D data, and, in many situations, require less user input and guidance. Several variants have been developed [12]–[17].

In this paper, we present a natural extension of our ACID framework that is suitable for deformable surfaces. In particular, we develop topology adaptive deformable surfaces, dubbed T-surfaces [18], for use on volume images. After a brief review of the planar T-snakes formulation in the next section, we develop its 3D extension in Section III. In Section IV, we present segmentation experiments using T-surfaces, demonstrating their potential for efficient, accurate, and reproducible extraction and analysis of anatomic structures from medical image volumes. Section V discusses T-surfaces in comparison to competing techniques. Section VI concludes the paper.

## II. Review of T-Snakes

Discrete versions of conventional parametric snakes [2], T-snakes are immersed in an ACID which supports efficient reparameterization. As shown in Fig. 1, they exhibit significant geometric and topological flexibility.

As a T-snake deforms under the influence of external and internal forces, it is periodically reparameterized with a new set of nodes and elements. This is done by efficiently computing the intersection points of the model with the superposed affine cell grid. During the reparameterization, the interior of a T-snake is also tracked by turning on any grid cell vertices that passed from the exterior to the interior of the model during its motion. Reparameterizing a T-snake at prespecified intervals as it flows yields an elegant automatic model subdivision technique and the grid provides a framework for robust topological transformations. Thus, the T-snake is relatively invariant to its initial placement. It flows into complex shapes, changing its topology whenever necessary, all in a stable manner. Conversion to a conventional parametric snakes model is simply a matter of disabling the grid at any time. By providing a boundary representation that also keeps track of the interior region of an object, T-snakes combine the space partitioning and topological flexibility of an implicit model with an explicit parametric boundary model.

There are three components to the T-snakes formulation as follows (see [9] for the details).

The first component is a discrete form of the conventional snakes model described in [2]. That is, a T-snake behaves like a standard parametric snake between reparameterizations and it is free to deform in any direction. The T-snake nodes act as a dynamic interconnected particle system. Associated with each node is a time varying position along with internal tensile forces and flexural forces to maintain model smoothness, inflationary forces to drive the T-snake toward the target object boundary, and external forces to stop the T-snake at the object boundary. T-snake evolution is governed by a simplified version of the Lagrange equations of motion in discrete form.

The second component of T-snakes is the ACID. There are two main types of affine cell decomposition methods: nonsimplicial and simplicial. In a simplicial decomposition, space is partitioned into cells defined by open simplices, where an $n$ simplex is the simplest geometrical object of dimension $n$: a triangle in 2-D and a tetrahedron in 3-D. For example, the Coxeter–Freudenthal triangulation (Fig. 2) is constructed by dividing space using a uniform cubic grid and the triangulation is obtained by subdividing each cube into $n!$ simplices. Simplicial cell decompositions provide a framework for creating robust consistent local polygonal (affine) approximations of the boundary contours of anatomic structures. The vertices of each simplex can be classified as either inside an anatomic structure or outside the structure. Simplicial cells containing both inside and outside vertices are termed boundary cells. In these cells the inside vertices can always be separated from the outside vertices by a single line or plane. Thus, an unambiguous polygonalization of the simplex always exists and a consistent polygonalization of the entire boundary contour will result.

The third component of T-snakes is the reparameterization process. A T-snake alternates between deformation steps and reparameterization steps. At the beginning of a 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. 3). To reestablish the correspondence between the T-snake and the grid, a new simplicial approximation of the deformed T-snake is computed, using a robust two-phase reparameterization algorithm. In phase I of the algorithm, the intersection points between the T-snake elements and the grid cell edges are computed. These intersection points will become the nodes of the new T-snake. In phase II, grid cell vertices that have moved from the exterior to the interior of the T-snake are marked as on. In this manner, the interior of a T-snake is continuously tracked.

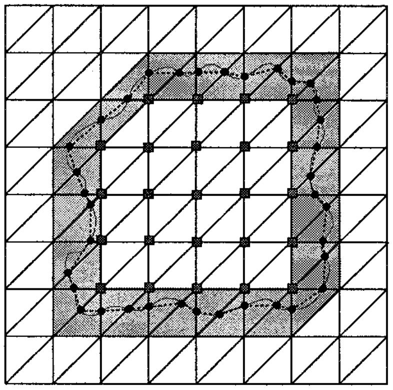
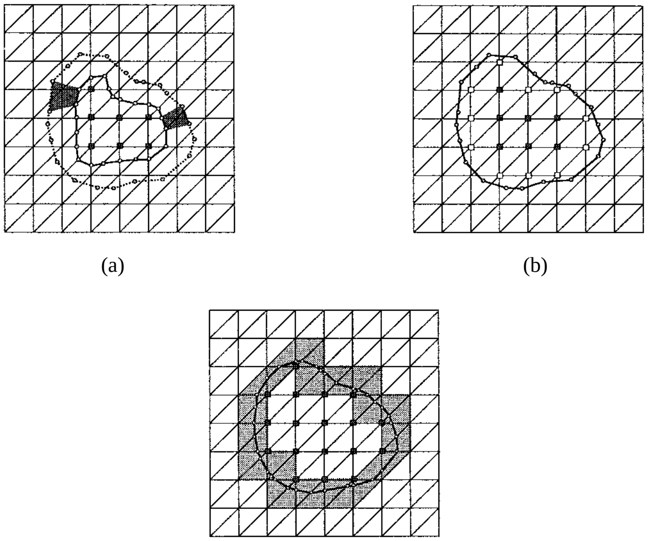


Fig. 2. Simplicial approximation (dashed line) of an object contour (solid line), using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded.



(a)

(b)

(c)

Fig. 3. Illustration of the T-snake reparameterization process. (a) Shaded regions show examples of grid vertices that are turned on by the expanding contour. (b) New inside grid vertices (white) added to current inside vertices (dark). (c) New contour after one deformation step showing new grid intersections, inside grid vertices, and boundary grid cells (gray shaded).

When a T-snake collides with itself or with another T-snake, or splits into two or more parts (or shrinks and disappears), a topological transformation must take place. T-snake topology changes are automatically performed via the ACID [Fig. 1(b)–(e)]. By keeping track of the inside grid vertices (and hence the boundary grid cells) and reestablishing the correspondence of the T-snake with the grid after every deformation step, the boundary or isocontour of the new T-snake(s) can always be unambiguously determined. New elements are constructed based on the signs (i.e., inside or outside) of the grid vertices in each boundary cell and from the intersection points computed in phase I, such that the inside and outside grid vertices in these cells are separated by a single line.

## III. Formulation of T-Surfaces

The main components of the 3-D T-surface formulation are analogous to those for the 2-D T-snakes. We describe each component in detail in the following sections.

### A. Model Description

The first component of T-surfaces, like T-snakes, is a discrete form of the conventional parametric deformable surfaces [19], [5]. A T-surface is a closed elastic triangular mesh, consisting of a set of nodes and triangular elements. We associate with each node $i$ a time varying position $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]$, along with internal tensile forces $\boldsymbol{\alpha}_i(t)$ and flexural forces $\boldsymbol{\beta}_i(t)$, inflationary forces $\boldsymbol{\rho}_i(t)$, and external forces $\mathbf{f}_i(t)$ [20]. The behavior of a T-surface is governed by a simplified version of the Lagrange equations of motion [20] in discrete form

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

where $\dot{\mathbf{x}}_i$ is the velocity of node $i$ and $\gamma_i$ is a damping coefficient that controls the rate of dissipation of the kinetic energy of the nodes.

A general nonlinear strain energy for a parametric deformable surface is a function of the differential area and curvature at each point [21]. A more practical version of this deformation energy is the linear combination of the well-known membrane and thin-plate functionals [22]. This linearized functional approximates the more general nonlinear strain energy functional for small deformations near the actual minimum (where higher order terms tend to zero), but is well behaved for large deformations and its quadratic form leads to computational benefits. The respective variational derivatives of the membrane and thin-plate functionals correspond to the Laplacian $L(\mathbf{x}) = \mathbf{x}_{uu} + \mathbf{x}_{vv}$ and squared Laplacian (biharmonic) $L^2(\mathbf{x}) = L \circ L(\mathbf{x}) = \mathbf{x}_{uuuu} + 2\mathbf{x}_{uuvv} + \mathbf{x}_{vvvv}$ (where $u, v$ represent the surface parameterization) and give rise to the internal tensile and flexural forces, respectively. We approximate the Laplacian at each node using the umbrella operator resulting in the internal tensile force

$$\boldsymbol{\alpha}_i(t) = L(\mathbf{x}_i(t)) = \frac{1}{m} \sum_{j \in \mathcal{N}(i)} x_j(t) - x_i(t) \tag{2}$$

where $x_j$ are the neighbors of the node $x_i$, and $m = \#\mathcal{N}(i)$ is the number of these neighbors (the valence). The parameter $a$ is used to control the strength of this force (i.e., the resistance of the model to stretching deformations). Currently, we approximate the Laplacian by taking only the local mesh topology at a node into account. A more accurate approximation would adjust the neighbor node weighting to reflect the local geometry of the model node as well [23]. To compute the internal flexural force at a model node, we approximate the squared Laplacian by convolving the umbrella operator over the node and its neighbors. The parameter $b$ is then used to control the resistance of the model to bending deformations.

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

$$\boldsymbol{\rho}_i(t) = qF(I(\mathbf{x}_i(t))\mathbf{n}_i(t) \tag{3}$$

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

$$F(I(\mathbf{x})) = \begin{cases} +1, & \text{if } I(\mathbf{x}) \geq T \\ -1, & \text{otherwise} \end{cases} \tag{4}$$

links the inflation force to the image data $I(\mathbf{x})$, where $T$ is an image intensity threshold. The function $F$ makes the T-surface contract when $I(\mathbf{x}) < T$ and is used to prevent the model from leaking into the background. Oscillation of the model can be prevented by progressively lowering the magnitude $q$ of the force toward zero once a change of direction is detected or if a model 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 [25], [26]:

$$F(I(\mathbf{x})) = \begin{cases} +1, & \text{if } |I(\mathbf{x}) - \mu| \leq k\sigma \\ -1, & \text{otherwise} \end{cases} \tag{5}$$

where $\mu$ is the mean image intensity of the target object, $\sigma$ the standard deviation of the object intensity, and $k$ is a user defined constant. The values of $\mu$ and $\sigma$ 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 model. Smooth subvoxel-accurate object boundaries are produced and a T-surface will pass over small spurious regions, preventing the creation of small holes in the region.

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

$$\mathbf{f}_i(t) = p\nabla P(\mathbf{x}_i(t)) \tag{6}$$

where the weight $p$ controls the strength of the force and the potential $P$ is defined by

$$P(\mathbf{x}) = -c\|\nabla[G_\sigma * I(\mathbf{x})]\|. \tag{7}$$

$G_\sigma * I$ denotes a Gaussian smoothing filter of standard deviation $\sigma$ and $c$ scales the potential. 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 $\mathbf{x}_i$ to improve robustness against noise.

Another effective external force is an inflation force that makes use of a Chamfer distance map [27] or a gradient vector field that approximates the distance and direction to the nearest edge [28]. In this scenario, the inflation force is weighted by the distance to the edge and is directed along a model 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 model from leaking into neighboring structures when there are significant gaps in the target object edges. Alternative functions $F$ that can be used as inflation force weights are based on local variance as well as intensity [29], texture [30], or other statistical measures of the target object intensity.

To calculate a continuous image function $I(\mathbf{x})$ for either (5) or (6) we compute the intensity at an arbitrary point $(\mathbf{x})$ by trilinear interpolation. We also 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

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

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

### B. 3-D ACID

The second component of T-surfaces is the extension of the ACID framework to three dimensions, using simplicial (tetrahedral) cells or nonsimplicial (e.g., hexahedral) cells. Most nonsimplicial methods employ a rectangular tessellation of space. The marching cubes algorithm [31] is an example of 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 [32]. For example, Fig. 4 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 an adherence to a disambiguation strategy such as preferred polarity: always separate the positive vertices (and join the negatives) or vice versa. We have implemented T-surfaces, using both nonsimplicial and simplicial decomposition methods. We will describe the simplicial grid approach in this paper. The formulation of T-surfaces using a nonsimplicial grid is essentially identical, except for the addition of the disambiguation scheme.

In a 3-D simplicial decomposition, space is partitioned into tetrahedral cells using the Coxeter–Freudenthal triangulation. We construct the grid by dividing the image volume using a uniform cubic grid and subdividing each cube into six tetrahedra [Fig. 5(a)]. As in the 2-D case, 3-D simplicial decompositions provide an unambiguous framework for the creation of robust consistent local polygonal approximations of the surface of anatomic structures. The polygonal approximation is constructed from the intersection of the object surface with the edges of each boundary cell (i.e., tetrahedra containing vertices both inside the structure and outside). The intersection points result in one triangle or one quadrilateral (which can be subdivided into two triangles) approximating the object surface inside each boundary cell [Fig. 5(b), (c)], where each triangle or quadrilateral intersects a tetrahedral cell on three or four distinct edges, respectively. The triangle (or quadrilateral) separates the positive vertices of the tetrahedral cell from the negative vertices. The set of all these triangles constitutes the polygonal approximation of the object surface. As in the case of T-snakes, we can obtain an approximation to any desired degree of accuracy by decreasing the size of the grid cells.

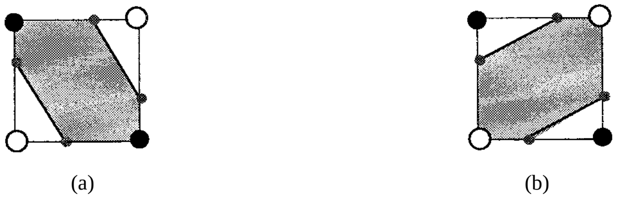


Fig. 4. (a) and (b) Example of ambiguous faces of a cube (black circle: positive cell vertex, open circle: negative cell vertex). Given the diagonal arrangement of vertex polarities, it is unclear which edge/surface intersection should be used.

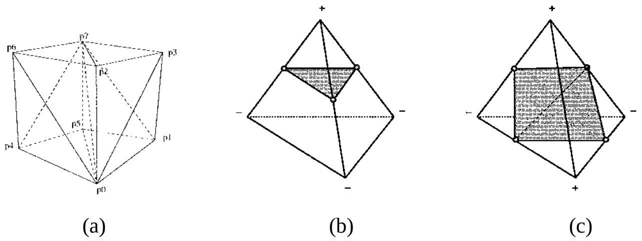


Fig. 5. (a) Cube divided into six tetrahedra: $\rho_0 = (p_0, p_1, p_3, p_7)$, $\rho_1 = (p_0, p_1, p_5, p_7)$, $\rho_2 = (p_0, p_2, p_3, p_7)$, $\rho_3 = (p_0, p_2, p_6, p_7)$, $\rho_4 = (p_0, p_4, p_5, p_7)$, $\rho_5 = (p_0, p_4, p_6, p_7)$. (b) Intersection of object boundary with grid cells.

### C. T-Surface Reparameterization

The third component of T-surfaces is a reparameterization process analogous to the T-snakes case. The time derivatives in (1) are approximated by finite differences. A T-surface is then 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 five and ten. The entire T-surface is set to either expand or shrink during one deformation step. This policy means that although a T-surface triangle element is free to move forward and backward during a deformation step, similar to a standard parametric deformable model, at the end of the deformation step the element cannot have moved such that a grid vertex previously inside the T-surface 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, an ACID-based model can alternate between expansion and shrinkage deformation steps, effectively mimicking the complete freedom of movement over all deformation steps of a parametric deformable model. In 3-D, this allows the user to interact with a T-surface by, for example, applying forces to a T-surface via cross sections of the surface that are overlaid on 2-D image slices.

We have explored several T-surface reparameterization algorithms. One such algorithm is detailed in [20] and this algorithm is used for all experiments described in Section IV. However, we are in the process of extending the newer and more efficient two-phase algorithm, briefly outlined for T-snakes, to 3-D. The 3-D version is exactly analogous to the 2-D case, except that we are computing intersections between triangle elements and grid cell edges in $\Re^3$ rather than edge elements and grid cell edges in $\Re^2$ .The new algorithm yielded a 15-fold increase in efficiency for T-snakes (most segmentation experiments involving T-snakes now require under a second to complete) and we expect considerable improvement for T-surfaces. For this reason, we will describe the new algorithm below.

*1) Phase I:* In phase I of the algorithm, we determine if a T-surface triangular element intersects a grid cell edge. Standard robust ray-triangle intersection algorithms are used and local neighborhood searches can be employed to speed up the process. 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-surface [see Fig. 3(a) and (c) for a 2-D illustration]. The intersection point may be unused and therefore is 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-surface 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-surface intersects itself or when multiple T-surfaces intersect. In these cases, we take the lower numbered vertex of the grid edge (each vertex in the grid is assigned a number) and determine on which side (i.e, half space) of the plane, formed by the T-surface triangle element, this vertex lies. Consequently, every intersection point is given a sign associating it with the lower numbered vertex, either inside or outside (using the outward pointing element normal as the reference direction). Every intersection point is then 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 triangle element is off (which indicates that is it is now in the interior of the T-surface), we also store it on a queue for processing in Phase II. Phase I of the reparameterization process is simple, efficient, and inherently parallel: each T-surface element can be processed independently.

*2) Phase II:* During a deformation step, as the T-surface expands some grid vertices that were outside of the closed T-surface will now be contained inside [see Fig. 3(a) for a 2-D illustration]. 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 model. Note that the inflation force pushes a T-surface in a direction normal to the surface at each model 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-surface. We solve this problem by mimicking the physically correct behavior for a propagating flame front. This behavior is selected by adhering to a so-called entropy condition [33]: once a grid vertex is turned on, it remains on.

As mentioned in Phase I, we compute the grid cell edge intersection points for each T-surface element and enqueue grid cell vertices to be labeled on or inside for processing in Phase II. In Phase II, we dequeue these vertices and check their corresponding grid cell edge data structure. If an intersection point is stored in the data structure (indicating that the T-surface 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 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-surface 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-surface). Phase II of the reparameterization process is simple, highly efficient, and inherently parallel. Typically, only a small number of grid vertices will be enqueued in Phase I for processing in phase II.

### D. T-Surface Equilbrium

In contrast to the level-set evolution techniques which accede control to a higher dimensional implicit function, we retain an explicit parametric model formulation. An explicit model formulation allows us to track and control the evolution of the T-surface. Consequently, reparameterizations can be performed very efficiently and constraints can be easily imposed on the model. Furthermore, a T-surface is considered to have reached its equilibrium state when all of its triangle elements have been inactive for a specified number of deformation steps. Element activity or movement is easily measured via the grid, using a flame propagation analogy. Elements are assigned a temperature, based on the number of deformation steps the element (and the grid cell in which it is defined) has remained valid (i.e., has remained a boundary cell). An element is considered inactive when its temperature falls below a preset freezing point. Frozen elements can be removed from the computation. This adjustable mechanism allows the system to maintain a manageable computational burden for many segmentation scenarios. Once a T-surface has reached equilibrium the grid can be disabled, if desired, and the T-surface run as a standard (discrete) deformable model.

## IV. Volume Image Segmentation with T-Surfaces

We have implemented a prototype system using T-surfaces. We have used the system to segment and reconstruct a wide

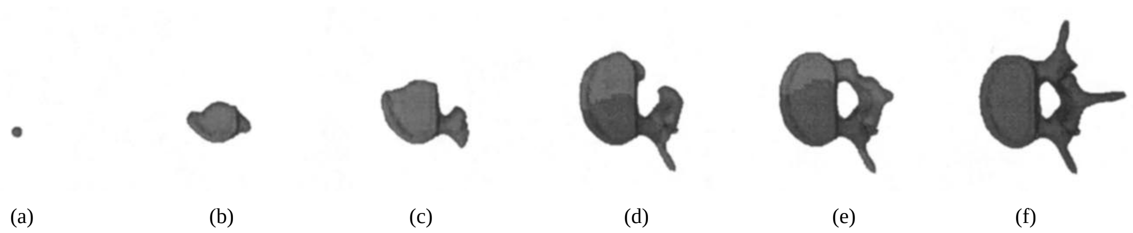


Fig. 6. T-surface segmenting vertebra phantom from CT image volume. The dark shaded regions are frozen and have been removed from the computation.

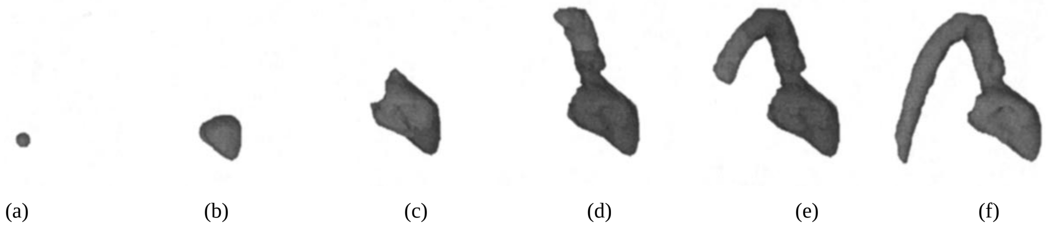


Fig. 7. T-surface segmenting canine LV and aorta from CT image volume.

range of anatomic structures with complex shapes and topologies to demonstrate its potential.

We initialize a T-surface, using an implicitly defined superquadric function which can be quickly scaled, bent, and tapered, if desired, and placed inside the object of interest (or scaled such that it contains the object). The superquadric is then converted into a T-surface and the model will then expand (or shrink if it surrounds the object) and adapt to the object boundaries. Multiple T-surface seeds can also be dynamically created and placed throughout the object. They will then adapt and automatically merge to form one model of the object.

All T-surface 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 $\Delta t$ and deformation step $M$ parameters are set to achieve maximum T-surface efficiency. For many segmentation scenarios we have found that setting $\Delta t$ to 0.001 and $M$ is to 10 works well. The resolution of the grid is usually set to as coarse as possible so that the target object can be accurately reconstructed as efficiently as possible. Objects with narrow protrusions or fine surface structures often require a voxel-resolution mesh. The force 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-surface into narrow protrusions. The internal tensile force strength is typically set to approximately 0.5–0.75 of the inflation force strength. In the following experiments, the internal flexural force was not utilized. In general, however, this force, which is computed over a larger T-surface neighborhood than the tensile force, produces smoother reconstructions and acts to prevent the T-surface from leaking into neighboring structures and is therefore useful when segmenting noisy image volumes.

### A. Geometric Flexibility and Topological Adaptability

In the first example, we apply a T-surface to a 120 × 128 × 52 CT image volume of a human vertebra phantom to demonstrate the topological adaptability of the T-surface (Fig. 6). We use a 32 × 30 × 13 cell grid (where each cubical cell is divided into six tetrahedra) with model parameters: $p = 20.0$, $q = 80.0$, $a = 60.0$, and time step $\Delta t = 0.001$.

In the second example we segment and reconstruct the left-ventricular (LV) chamber and aorta from a CT image volume of a canine heart. The image volume dimensions are 128 × 128 × 118 and a 20 × 20 × 20 resolution grid was used (model parameters: $p = 61.0$, $q = 60.0$, $a = 40.0$, $\Delta t = 0.002$). We manually seed the LV with a small T-surface and the segmentation then proceeds automatically (Fig. 7). The inflation force is weighted with LV region image intensity statistics to reinforce the image edge forces. The final result, after disabling the grid and converting to a classical deformable surface, is shown at the top of Fig. 8. While there are a few image slices that may require manual editing, most of the model fits very accurately and the entire process takes under 4 min on an SGI Indigo 2 workstation using the older reparameterization algorithm described in [20]. We expect significantly better performance using the new reparameterization algorithm. Segmentation with T-surfaces is highly reproducible. As long as the T-surface is seeded within the bright region of the LV, it produces highly similar results (measured using visual inspection only).

In the third example, we have used a T-surface to segment and reconstruct the vascular system of the brain from an MRA image volume. The data consist of a stack of 100 slices each with $512^2$ pixels, 2 bytes per pixel (voxel size 0.429 687 5 mm × 0.429 687 5 mm × 0.7 mm). The image volume was interpolated to produce cubical voxels, resulting in 162 slices. A T-surface is seeded at the root of the vessel tree and then flows into the vessels, automatically extracting the vascular system (Fig. 9). A voxel-resolution grid was used with model parameters: $p = 00.0$, $q = 80.0$, $a = 60.0$, $\Delta t = 0.0005$. The segmentation takes about 1 h on an SGI Indigo 2 workstation, primarily due to disk thrashing induced by the massive image data set relative to the memory in the workstation. We expect considerable performance improvement on a workstation with sufficient memory and our new reparameterization algorithm. When combined with image statistics-weighted inflation forces, a T-surface behaves as an active region growing model that is able

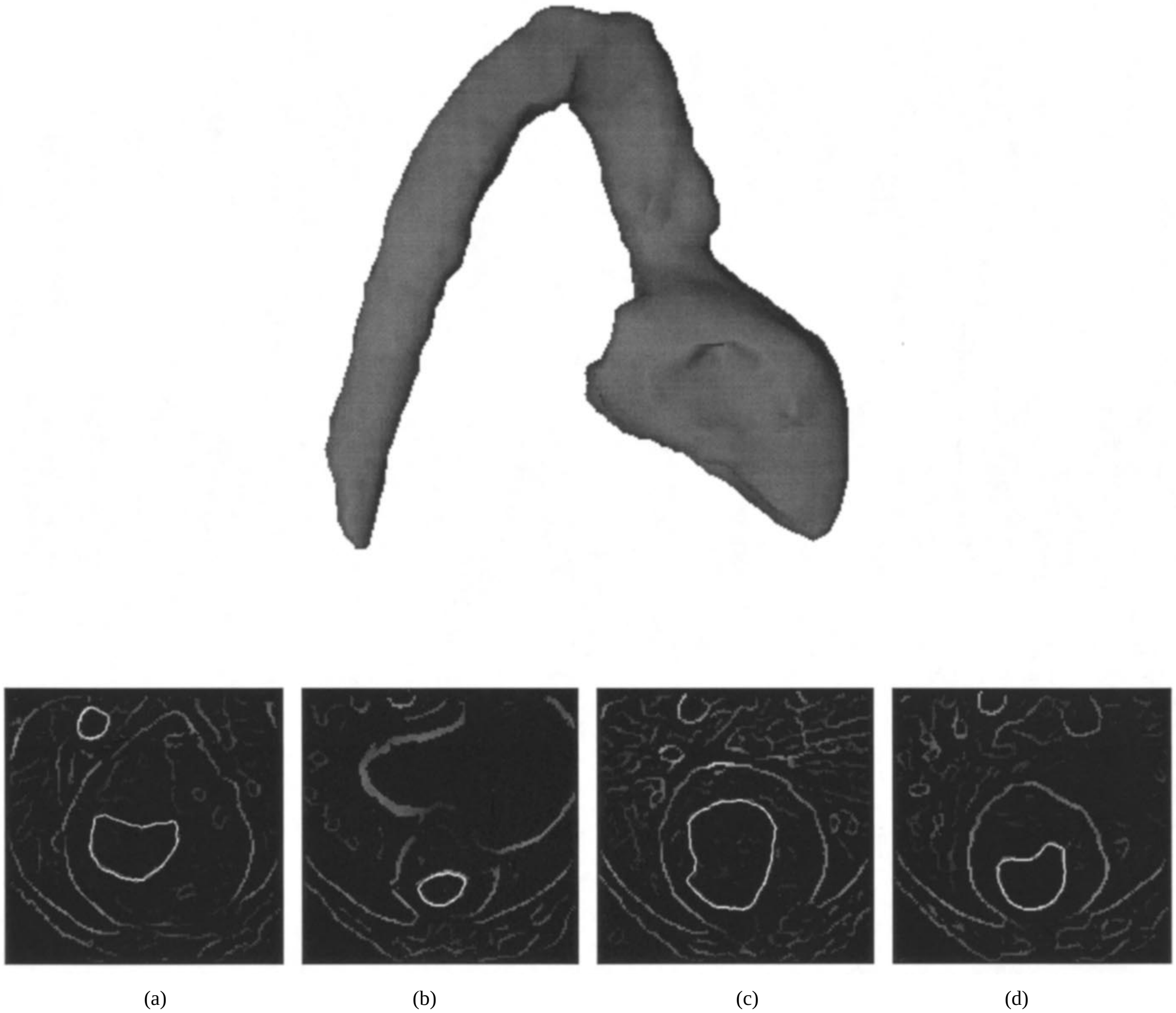


Fig. 8. Final fitted T-surface and four cross sections of the model overlaid on edge-detected image slices.

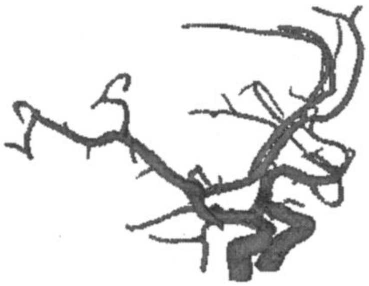


Fig. 9. T-surface segmentation of cerebral vasculature from MRA image volume.

to integrate edge information and filter out noise through the model smoothness constraints.

Although MR angiography produces high contrast images, the arteries exhibit a large variation in gray-scale values. This large intensity variation makes it difficult to segment the images with simple thresholding techniques. A possible solution to this problem is the use of an adaptive threshold. An adaptive threshold can be computed for the active portions of the T-surface in each branch of the arteries, based on a local statistical analysis. This solution is possible using T-surfaces since we maintain an explicit model formulation. One drawback of T-surfaces for segmenting narrow branching structures, such as arteries, is that the grid resolution is dictated by the smallest artery. In order to flow into tiny branches, a voxel-resolution or even subvoxel-resolution grid must be used. A model that is able to locally adapt its resolution to match the size of the arterial branch may be better suited for images containing a large range of artery diameters.

In the final two examples we use a T-surface with an image statistics-weighted inflation force to segment the major brain ventricles and the cerebral cortex from a preprocessed MR image volume (the skull and brain stem have been manually removed from the image) (Figs. 10 and 11). The 256 × 256 ×85 (0.9375 mm × 0.9375 mm × 1.5 mm) MR image volume was interpolated to produce cubical voxels resulting in 136 slices rather than 85. A 128 × 128 × 68 resolution grid was used for the ventricles (model parameters $p = 30.0$, $q = 48.0$, $a = 40.0$, $\Delta t = 0.0008$, and $N = 10$). Since the T-surface provides a boundary and interior representation, the volume occupied by the model can be be quickly and accurately calculated. Brain ventricle volumes are useful indicators of many brain pathologies. Fig. (c)–(g) shows several cross sections of the fitted model. Note that the T-surface had difficulty penetrating into the narrow inferior horn of the

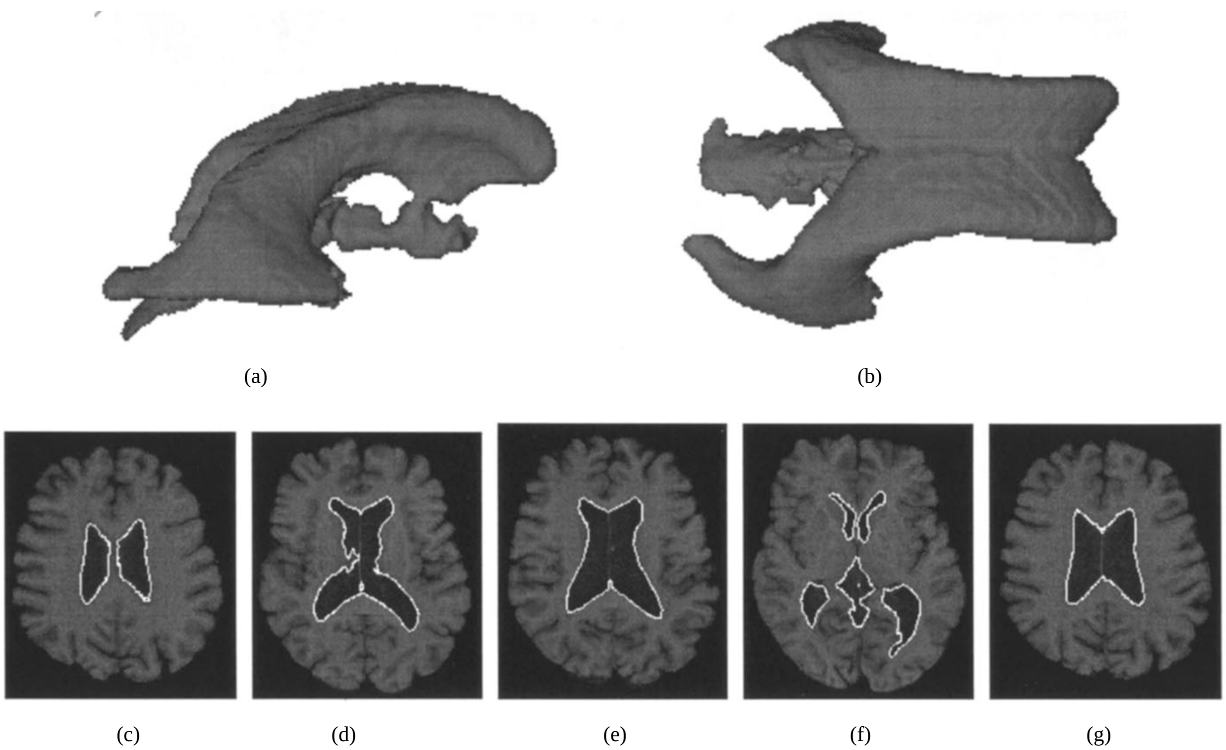


Fig. 10. T-surface segmentation of ventricles from MR image volume of the brain.

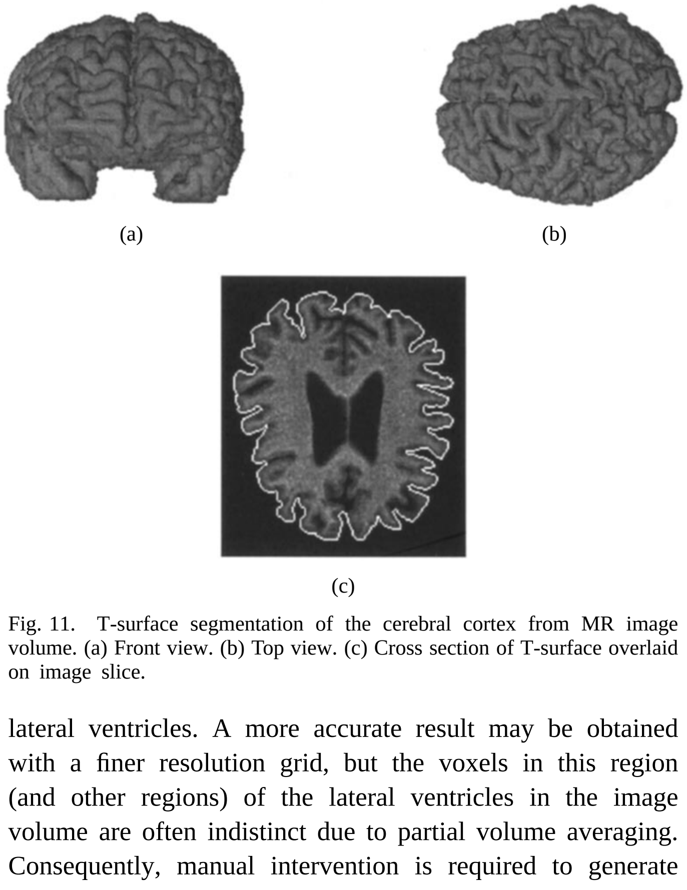


Fig. 11. T-surface segmentation of the cerebral cortex from MR image volume. (a) Front view. (b) Top view. (c) Cross section of T-surface overlaid on image slice.

lateral ventricles. A more accurate result may be obtained with a finer resolution grid, but the voxels in this region (and other regions) of the lateral ventricles in the image volume are often indistinct due to partial volume averaging. Consequently, manual intervention is required to generate an accurate segmentation in some regions. T-surfaces fully support user interaction through their intuitive physics-based paradigm.

A 64 × 64 × 34 resolution grid was used for the cortex (model parameters: $p = 0.0$, $q = 48.0$, $a = 40.0$, $\Delta t = 0.0008$). The T-surface was initialized to surround the cortex and then shrinks and conforms to the CSF/gray-matter interface. A topology preservation constraint was applied to the T-surface to ensure that the T-surface maintains a spherical topology. This global topology constraint is implemented using a series of topology preserving T-surface deformations [20]. The automatically subdividing T-surface is able to penetrate, with good success, into the narrow and deep cavities of the highly convoluted cortex [Fig. 11(c)]. A more accurate result could be obtained with a finer resolution grid, at the expense of increased computation time and a larger number of model triangles. We can also use the grid to easily and efficiently implement a coupled surfaces propagation technique, as described in [34], to make more effective use of *a priori* anatomical knowledge and segment the CSF/gray-matter and gray/white-matter interface, simultaneously.

## V. Discussion

In this section, we compare T-surfaces with alternative techniques.

A common method for segmenting image volumes is the use of thresholding to generate an isosurface of the target object. Although simple and fast, thresholding is highly noise sensitive, is not guaranteed to produce closed surfaces, and provides no local control. It must be used in concert with morphological and other filtering operations to perform corrections, resulting in tedious user interaction and user-dependent segmentations.

Region growing is another common technique and, like thresholding, is also difficult to control. Furthermore, it does not make use of image edge information, has no inherent noise suppresion capabilities, can leak through gaps in boundaries, and, again, provides limited ability to locally control the segmentation. By contrast, T-surfaces integrate both edge and region information, provide smoothing (noise suppression), can fill in gaps in boundaries, can guarantee topology, and, perhaps most importantly, can be locally controlled through intuitive interaction mechanisms.

Several researchers have attempted to overcome the limitations of parametric deformable surface models by adding greater functionality, or by using discrete deformable meshes with automatic model refinement and topology adaptation mechanisms [12], [14], [17], [35], [36]. Like T-surfaces, these models can support user interaction through energy/force-based constraints. Unlike T-surfaces, which are automatically reparameterized (or retriangulated) via the ACID framework, the retriangulation process of these models is typically based on triangle or particle subdivision rules. This type of mechanism can create initial position sensitivity and therefore adversely affect segmentation reproducibility. T-surface parameterization is intrinsic. That is, the ACID grid triangulates the fitted T-surface in terms of the intrinsic local geometry of the target object, providing much lower sensitivity to initial conditions. However, a possible advantage the discrete deformable mesh models have over T-surfaces is their potential for simple local retriangulation. Currently, a T-surface is globally reparameterized (although a form of local reparameterization is possible by activating locally defined regions of the grid and deactivating the remainder).

To introduce geometric and topological flexibility, several researchers have developed implicit deformable curves and surfaces by adopting Osher and Sethian's [33] level-set evolution technique to the image segmentation problem [37], [38], [15], [39], [16], [40], [41]. These models are formulated as fronts, evolving 2-D contours or 3-D surfaces which define the level set of some higher dimensional hypersurface over the image domain.[1] 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 hypersurface remains a simple function even as the level-set changes topology. The reparameterization process is inherent in the level-set formulation, while reparameterization in T-surfaces is an explicit process. While the level-set technique is an attractive mathematical framework, implicit formulations are much less convenient than explicit surface formulations when it comes to incorporating additional control mechanisms, such as internal deformation energies, external force functions tailored to the specific image features of a target object, and external interactive guidance by expert users. Furthermore, in the implicit formulation it is difficult to impose arbitrary geometric or topological constraints on the level set indirectly, through its higher dimensional representation.

A T-surface is a hybrid model that combines aspects of both explicit (parametric) deformable surface models and implicit surface models. The ACID framework introduces aspects of the latter, primarily topological flexibility. The primary advantage of retaining an explicit model formulation, however, is user control. In medical image analysis it is essential for a user to be able to interactively control the segmentation process. Automatic methods can fail for many reasons: local visual evidence of object boundaries may be too low or even absent, or it may be overrun by a neighboring object. In these cases, recognition of the object boundary must be performed by an expert and simple intuitive guidance or correction of the segmentation is required. By retaining the explicit, Lagrangian formulation, T-surfaces can be dynamically manipulated using virtual forces such as interactive springs, anchored springs, and volcanos [2]. In addition, an explicit model formulation supports the use of more sophisticated control mechanisms. For example, in 3-D an initial T-surface can be constructed from the convex hull of a set of user defined points. These points can be used as anchors and represent known object boundary points. External forces can then be progressively applied to a T-surface, starting from the anchor points and spreading over the entire T-surface [42]. Another possibility is the local control over the evolution of the T-surface by pulling regions of the T-surface, using an implicitly defined shape primitive such as a cylinder or ellipse. A final example is to temporarily disable or limit the flow in a user specified direction, but enable it in orthogonal directions. This type of control may be useful, for example, when segmenting curved, elongated structures.

## VI. Conclusion

Deformable surface models offer a flexible and powerful approach to medical image analysis. Nevertheless, standard parametric deformable surface models suffer from several limitations that prevent their application to the full range of medical image analysis problems and inhibit their potential degree of automation. We have proposed T-surfaces as a solution to many of these limitations. This new class of deformable models, which work in concert with an ACID, can be used to segment, reconstruct, and analyze complex anatomic structures from medical image volumes in an efficient, reproducible, and highly automated manner. At the same time, our T-surface models retain the advantages of standard parametric deformable models, such as user interaction and the incorporation of constraints through energy or force functions. The ACID framework complements and extends the standard models and users can dynamically disable it in situations where more direct control and interaction is required.

Opportunities for future research include improving model performance through the use of the new reparameterization algorithm, improving interactive control over the flow of a T-surface through the use of mouse-controlled spring forces, user defined point constraints and implicitly defined shape primitives, as well as an interactive T-surface editing ca-

[1] There is a mathematical relationship between the implicit models and simple types of parametric deformable models. The energy minimization of a deformable model involves variational principles which can be equivalently formulated as solutions to partial differential equations (Euler–Lagrange equations). Caselles [39] pointed out that setting the rigidity term to zero in the energy functional of a snake model establishes an equivalence between this parametric model and a geometric curve evolution, since they both minimize the length of the contour in the metric induced by the image. In three dimensions, minimization of the area functional for a surface (which is often approximated in parametric deformable model formulations using the well-known membrane functional) is equivalent to mean curvature surface evolution.

pability. Finally, since ACID-based deformable models are naturally multiresolution models (i.e., simply by changing the grid resolution), this feature can be used with multiscale image preprocessing techniques [2], [1] to further improve the efficiency and robustness of the segmentation.

## Acknowledgment

The MR brain image volume was provided courtesy of C. Davatzikos of Johns Hopkins School of Medicine. The MRA image volume was provided courtesy of A. R. Sanderson, H. Bushwell, and D. Parker of the University of Utah, Department of Radiology and the Department of Computer Science.

## References

[1] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Artif. Intell.*, vol. 36, no. 1, pp. 91–123, 1988.

[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comp. Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[3] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara, and M. J. Wozny, "Geometrically deformed models: A method for extracting closed geometric models from volume data," in *Proc. Computer Graphics SIGGRAPH'91 Conf.*, Las Vegas, NV, July, 1991, vol. 25, no. 4, pp. 217–226.

[4] I. Cohen, L. D. Cohen, and N. Ayache, "Using deformable surfaces to segment 3D images and infer differential structures," *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 242–263, 1992.

[5] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis," *Comput. Med. Imag. Graph.*, vol. 19, no. 1, pp. 69–83, Jan. 1995.

[6] ———, "Deformable models in medical image analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, 1996.

[7] A. Singh, D. Goldgof, and D. Terzopoulos, *Deformable Models in Medical Image Analysis*. Los Alamitos, CA: IEEE Computer Society, 1998.

[8] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proc. Fifth Int. Conf. Computer Vision (ICCV'95)*. Los Alamitos, CA: IEEE Computer Society, 1995, pp. 840–845.

[9] ———, "Topology adaptive snakes," *Med. Image Anal.*, to be published.

[10] A. Singh, L. von Kurowski, and M. Y. Chiu, "Cardiac MR image segmentation using deformable models," in *Biomedical Image Processing and Biomedical Visualization*. Bellingham, WA: SPIE, 1993, vol. 1905 pp. 8–28.

[11] I. Carlbom, D. Terzopoulos, and K. Harris, "Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections," *IEEE Trans. Med. Imag.*, vol. 13, pp. 351–362, 1994.

[12] F. Leitner and P. Cinquin, "Complex topology 3D objects segmentation," in *Model-Based Vision Development and Tools.* Bellingham, WA: SPIE, 1991, vol. 1609, pp. 16–26.

[13] H. Delingette, M. Hebert, and K. Ikeuchi, "Shape representation and image segmentation using deformable surfaces," *Image Vision Comput.*, vol. 10, no. 3, pp. 132–144, Apr. 1992.

[14] R. Szeliski, D. Tonnesen, and D. Terzopoulos, "Modeling surfaces of arbitrary topology with dynamic particles," in *Proc. Conf. Computer Vision Pattern Recognition (CVPR'93)*. Los Alamitos, CA: IEEE Computer Society, 1993, pp. 82–87.

[15] R. Whitaker, "Volumetric deformable models," in *Proc. Third Conf. on Visualization in Biomedical Computing (VBC'94*, R. A. Robb, Ed. Bellingham, WA: SPIE, 1994, vol. 2359.

[16] R. Malladi, R. Kimmel, D. Adalsteinsson, G. Sapiro, V. Caselles, and J. A. Sethian, "A geometric approach to segmentation and analysis of 3D medical images," in *IEEE Workshop Mathematical Methods Biomedical Image Analysis,*, Los Alamitos, CA: IEEE Computer Society, 1996, pp. 244–252.

[17] J.-O. Lachaud and A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction," *Med. Image Anal.*, vol. 3, no. 2, pp. 187–207, 1999.

[18] T. McInerney and D. Terzopoulos, "Medical image segmentation using topologically adaptable deformable surface models," in *Proc. Second Int. Conf. Computer Vision Virtual Reality Robotics Medicine*, Grenoble, France, Mar. 1997, pp. 23–32.

[19] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, Mar. 1991.

[20] T. McInerney, "Topologically adaptable deformable models for medical image analysis," Ph.D. dissertation, Dept. Comp. Sci., Univ. of Toronto, Toronto, Ont, Canada, 1997.

[21] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Computer Graphics (Proc. SIGGRAPH'87 Conf.*, Anaheim, CA, July, 1987, vol. 21, pp. 205–214.

[22] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 413–424, Apr. 1986.

[23] G. Taubin, "A signal processing approach to fair surface design," in *Proc. SIGGRAPH'95* (Annual Conference Series). New York: ACM, 1995, pp. 351–358.

[24] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. SIGGRAPH'9* (Annual Conference Series). New York: ACM, 1999, to be published.

[25] J. Ivins and J. Porrill, "Statistical snakes: Active region models," in *Proc. 5th Br. Machine Vision Conf. (BMVC'94)* BMVA, 1994, pp. 377–386.

[26] T. Kapur, WEL. Grimson, WM. Wells III, and R. Kikinis, "Segmentation of brain tissue from magnetic resonance images," *Med. Image Anal.*, vol. 1, no. 2, 1996.

[27] G. Borgefors, "Distance transformations in arbitrary dimensions," *Comp. Vision, Graph., Image Processing*, vol. 27, pp. 321–345, 1984.

[28] J. L. Prince and C. Xu, "A new external force model for snakes," in *Proc. 1996 Image Multidimensional Signal Processing Workshop*, 1996, pp. 30–31.

[29] L. Lorigo, O. Faugeras, W. E. L. Grimson, R. Keriven, and R. Kikinis, "Segmentation of bone in clinical knee MRI using texture-based geodesic active contours," in *Medical Image Computing and Computer-Assisted Intervention: Proc. 1st Int. Conf. (MICCAI'98)* (Lecture Notes in Computer Science), W. Wells, A. Colchester, and S. Delps, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. 1496, pp. 1195–1204.

[30] R. Durikovic, K. Kaneda, and H. Yamashita, "Dynamic contour: A texture approach and contour operations," *Visual Comp.*, vol. 11, pp. 277–289, 1995.

[31] W. E. Lorensen and H. E. Cline, "Mar.ing cubes, a high resolution 3D surface construction algorithm," *Comp. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[32] P. Ning and J. Bloomenthal, "An evaluation of implicit surface tilers," *IEEE J. Comp. Graph. Appl.*, pp. 33–41, Nov. 1993.

[33] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," *J. Computat. Phys.*, vol. 79, pp. 12–49, 1988.

[34] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan, "Segmentation and measurement of the cortex from 3D MR images," in *Medical Image Computing and Computer-Assisted Intervention: Proc. 1st Int. Conf. (MICCAI'98* (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 1995, vol. 1496, pp. 519–530.

[35] J. Montagnat and H. Delingette, "Volumetric medical image segmentation using shape constrained deformable models," in *Second International Conf. on Computer Vision, Virtual Reality and Robotics in Medicine.* Berlin Germany: Springer-Verlag, 1997, pp. 13–22.

[36] D. MacDonald, D. Avis, and A. C. Evans, "Proximity constraints in deformable models for cortical surface identification," in *Medical Image Computing and Computer-Assisted Intervention: Proc. 1st Int. Conf. (MICCAI'98)*. Berlin Germany: Springer-Verlag, 1998, pp. 650–659.

[37] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours," *Numerische Mathematik*, vol. 66, 1993.

[38] R. Malladi, J. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, Feb. 1995.

[39] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *Proc. Fifth International Conf. on Computer Vision (ICCV'95)*. Los Alamitos, CA: IEEE Computer Society, 1995, pp. 694–699.

[40] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Trans. Med. Imag.*, vol. 16, pp. 199–209, Apr. 1997.

[41] V. Caselles, R. Kimmel, and G. Sapiro, "Minimal surfaces based object segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, Apr. 1997.

[42] W. Neuenschwander, P. Fua, G. Szeḱely, and O. Kubler, "Velcro surfaces: Fast initialization of deformable models," *Comp. Vision Image Understanding*, vol. 65, no. 2, pp. 237–245, Feb. 1997.