A topology preserving geometric deformable model and its application in brain cortical surface reconstruction

Xiao Han
Chenyang Xu
Jerry L. Prince

ABSTRACT

Geometric deformable models, implemented via level-set methods, have advantages over parametric models due to their intrinsic behavior, parameterization independence, topological flexibility, lack of self-intersections, and good numerical stability. But topological flexibility actually hinders the application of geometric deformable models in cases where the model must conform to the known topology of the final object. In this chapter, we present a new geometric deformable model that preserves topology using the simple point concept from digital topology. The new model, which we refer to as topology preserving geometric deformable model (TGDM), conforms to the topology constraint while maintaining other desirable characteristics of standard geometric deformable models including sub-pixel accuracy and production of non-intersecting curves (or surfaces). We then use TGDM to find the inner, central, and outer surfaces of the human brain cortex from magnetic resonance (MR) images. The resulting algorithm is fast and numerically stable, and yields accurate brain surface reconstructions that are guaranteed to be topologically correct and free from self-intersections.

1 Introduction

Deformable models, are curves or surfaces that deform within two-dimensional (2D) or three-dimensional (3D) digital images under the influence of both internal and external forces and user defined constraints. Ever since their introduction by Kass et al. [KWT88], these algorithms have been at the heart of one of the most active and successful research areas in edge detection, image segmentation, shape modeling, and visual tracking. Deformable models are broadly classified as either parametric deformable models (cf., [KWT88, Coh91, XP98]) or geometric deformable models (cf., [CCCD93, MSV95, CKS97, YKOT97, SLTZ98]) according to their
representation and implementation. In particular, parametric deformable models are represented *explicitly* as parameterized contours\(^1\) and implemented in a Lagrangian framework. Geometric deformable models, on the other hand, are represented implicitly as the zero level set of higher-dimensional level set functions and evolve according to an Eulerian formulation known as the level set method [OS88, Set99].

Geometric deformable models were introduced more recently than parametric models, and they have several important advantages [CCCD93, MSV95]. First, they are completely intrinsic and independent of the parameterization of the evolving contour. Thus, there is no need to add or remove nodes from the initial parameterization or adjust the spacing of the nodes as in parametric models. Second, the intrinsic geometric properties of the contour such as the curvature and the normal vector can be easily determined from the level set function. Third, the propagating contour can automatically change topology (e.g., merge or split) without requiring an elaborate mechanism to handle such changes (cf. [MT95, DM00]). Finally, the resulting contours do not contain self-intersections, which are costly to prevent in parametric deformable models (cf. [MKA00]).

Topological flexibility has long been claimed as a major advantage of geometric deformable models over parametric ones. In response to this, in fact, methods to adaptively change the contour topology have also been developed for parametric deformable models [MT95, DM00]. But topological flexibility is not always desirable. In particular, when a specific object (target) is sought and its composition — i.e., the number of components and the homology of each component — is known, then it is most natural to seek the target in a way that yields the correct composition, or topology. For example, in the analysis of 3D brain images — the application that motivated our work on this subject — it is desirable that a reconstruction of the cortical surface has a topology that is consistent with brain anatomy [XPR+99, MKA00]. Recently, in fact, there have been several post-processing methods reported to correct the topology of a cortical segmentation that has the wrong topology [SL01, FLD01, HXBNP02]. In this application, and others like it, topology flexibility should be considered to be a liability rather than an advantage [HFG99, HXBNP02].

In this chapter, we describe a topology preserving mechanism for geometric deformable models that guarantees that all evolving contours (including the final contour) have exactly the same topology as the initial contour. The topology preservation is achieved by maintaining the topology of the digital object encircled by the implicit contour, for which we make use of the *simple point* criterion from digital topology [KR89, SC94, Ber94]. We note that our approach maintains the sub-pixel interpolation property of standard geometric deformable models, which contrasts our method with

\(^1\)In this chapter, “contour” refers to either a “curve” or a “surface.”
the topology preserving region growing method of Mangin et al. [MFB+95]. The topology preserving mechanism we describe can be used with any existing 2D or 3D geometric deformable model, regardless of the internal or external force definition, yielding a large new class of deformable models, which we will refer to as topology-preserving geometric deformable models (TGDM's). As an application of TGDM, we designed and implemented a fully automatic procedure to reconstruct the gray matter/white matter, central, and pial surfaces of the human brain cerebral cortex from magnetic resonance (MR) images. The method is fast, computationally stable, and is guaranteed to yield surfaces that are topologically equivalent to a sphere and do not have self-intersections.

We note that the material presented in this chapter is based on two published conference papers [HXP01, HXTP01].

2 Background

In this section, we first present a brief introduction to geometric deformable models and the evolution of level set functions. We then introduce some basic notions about digital topology and the simple point concepts that will be used to build our topology preserving mechanism. We conclude with a description of isocontour algorithms, which must be implemented correctly in order to yield an accurate and topologically correct representation of the implicit contour.

2.1 Geometric deformable models

Geometric deformable models are based on the theory of curve evolution and are implemented using the level set numerical method [OS88, Set99]. Let $I(x) : U \rightarrow \mathcal{R}^+$ be a given image, where $U \subset \mathcal{R}^2$ in 2D and $U \subset \mathcal{R}^3$ in 3D. In geometric deformable models, the evolving contours are embedded as the zero level set of a higher dimensional level set function $\Phi(x, t) : U \times \mathcal{R}^+ \rightarrow \mathcal{R}$, and propagate implicitly through the temporal evolution of $\Phi(x, t)$. By convention, $\Phi(., t)$ is a signed distance function to the contour at each time instant with negative value inside the contour and positive outside. Such a level set function can be computed efficiently from an initial contour using the fast marching method [Tsi95, Set96, Set99].

Many different forms of geometric deformable models exist in the literature (e.g. [CCCD93, MSV95, CKS95, CKS97, CKSS97, KKO+95, KKO+96, TK95, SLTZ98, YTW99, PD00]). One of the most famous is the geodesic active contour model proposed independently by Caselles et al [CKS95, CKS97, CKSS97] and Kichenassamy et al [KKO+95, KKO+96]. It can be
described by the following level set evolution equation:

$$\frac{\partial \Phi(x, t)}{\partial t} = g(x) \kappa(x, t) |\nabla \Phi(x, t)| + cg(x) |\nabla \Phi(x, t)| + \nabla g(x) \cdot \nabla \Phi(x, t), \quad (1.1)$$

where \( \nabla \) is the gradient operator and \( \kappa(x, t) \) is the (mean) curvature of the level set of \( \Phi(\cdot, t) \) that passes through \( x \). The curvature \( \kappa(x, t) \) can be computed directly from the spatial derivatives of \( \Phi \). In this model, \( c \) is a constant and \( g(\cdot) \) is an image-derived metric, which is often a monotonically decreasing function of the gradient magnitude of the image \( I \).

In spite of the large variety of geometric deformable models, we can summarize their evolution equations in the following general form [Set99, XYP01]:

$$\frac{\partial \Phi(x, t)}{\partial t} = F_{\text{prop}}(x, t) |\nabla \Phi(x, t)| + F_{\text{curv}}(x, t) |\nabla \Phi(x, t)| + F_{\text{adv}}(x, t) \cdot \nabla \Phi(x, t), \quad (1.2)$$

where \( F_{\text{prop}} \), \( F_{\text{curv}} \), and \( F_{\text{adv}} \) are speed terms (sometimes called “forces”) that can be spatially varying. In particular, \( F_{\text{prop}} \) is an expansion or contraction force, \( F_{\text{curv}} \) is the part of the force that depends on the intrinsic geometry, especially the curvature of the contour and/or its derivatives, and \( F_{\text{adv}} \) is an advection force that passively transports the contour.

Eq. (1.2) yields the geodesic active contour model when \( F_{\text{prop}}(x, t) = cg(x) \), \( F_{\text{curv}}(x, t) = \kappa(x, t) g(x) \), and \( F_{\text{adv}}(x, t) = \nabla g(x) \). In this case, the model arises as the result of a gradient descent minimization of an energy functional (because that is how the geodesic active contour model was derived). In general, however, one can choose a different form for each speed term in (1.2) to yield a different behavior of the model. As an example, we can choose \( F_{\text{prop}}(x, t) = R(x) \) to be a region force\(^2\) (cf. [PD00, XYP01]) or a binary flow force [YTW99], \( F_{\text{curv}}(x, t) \) to be proportional to the (mean) curvature \( \kappa(x, t) \), and \( F_{\text{adv}}(x, t) = \nabla(x) \) to be a gradient vector flow force [XPH08]. With these choices the evolution equation becomes

$$\frac{\partial \Phi(x, t)}{\partial t} = \omega_R R(x) |\nabla \Phi(x, t)| + \omega_\kappa \kappa(x, t) |\nabla \Phi(x, t)| + \omega_\nabla \nabla(x) \cdot \nabla \Phi(x, t), \quad (1.3)$$

where \( \omega_R \), \( \omega_\kappa \), and \( \omega_\nabla \) are weights for the respective forces. For a binary-valued image \( I \) having values zero or one, it is convenient to define \( R(x) = 2I(x) - 1 \) to provide an expansion force inside the object and a contraction force outside. The model in (1.3) is used in our brain cortical surface reconstruction algorithm.

The numerical solution of Eq. (1.2) [or (1.3)] can be obtained by approximating the time derivative by a forward difference and the spatial derivatives by upwind numerical schemes (for details see [OS88, Set99]),

\(^2\)Also known as a signed pressure force.
which gives:

$$\Phi(x_i, t_{m+1}) = \Phi(x_i, t_m) + \Delta t \Delta \Phi(x_i, t_m),$$  \hspace{1cm} (1.4)$$

where $x_i$, $i = 1, 2, \ldots$, denotes the discrete grid points, $t_m$, $m = 0, 1, \ldots$, denotes the discrete time steps, and $\Delta t = t_{m+1} - t_m$ is the time-step size. Since we are interested in a generic geometric deformable model, we use $\Delta \Phi$ to denote the upwind finite difference approximation to the generic right hand side of (1.2) (see [Set99] for an explicit formula). Then, at each time step $t_{m+1} = (m + 1)\Delta t$, we update the value of the level set function $\Phi(\cdot, t_{m+1})$ at each grid point $x_i$ from its previous values $\Phi(\cdot, t_m)$, until convergence or after a user specified number of time steps.

In this framework, the updating of the level set function $\Phi$ is performed on fixed grid points; thus, no parameterization of the deforming contour is needed. The explicit parametric representation need only be computed after the evolution is completed by taking the zero level set of $\Phi$ at the last time step, which requires an isocontour algorithm. For efficiency, the narrow-band method can be used to update the level set function only at a small subset of points in the neighborhood of the zero level set instead of at all the points in the computational domain [Cho93, AS95, Set99]. This scheme requires recomputing the level set function after a certain number of time steps since the zero level set might move out of the updating region. It is well known that the topology of the embedded contour can change during the evolution of the level set function $\Phi$. This means that the topology of the final contour is unpredictable, and this is the main problem we address in this work.

\section{2.2 Digital topology}

A 2D (resp. 3D) binary image $V \subset \mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) is defined as a square (resp. cubic) array of lattice points. We follow the conventional definition of $n$-neighborhood and $n$-connectivity, where $n \in \{4, 8\}$ in 2D and $n \in \{6, 18, 26\}$ for a 3D image [KR89]. We denote the $n$-neighborhood of a point $x$ by $N_n(x)$, and the set comprising the neighborhood of $x$ with $x$ removed by $N_n^*(x)$. The set of all $n$-connected components of $X \subset V$ is denoted by $\mathcal{C}_n(X)$.

In order to avoid a connectivity paradox, different connectivities, $n$ and $\bar{n}$, must be used in a binary image comprising an object (foreground) $X$ and a background $\bar{X}$. For example, in 2D, if $n$ is chosen to be 4, then $\bar{n}$ must be 8, and vice versa. In 3D, (18, 6) and (26, 6) are the two pairs of compatible connectivities. The following definitions are from [Ber94] and [BEC97].

\textbf{Definition 1 (Geodesic Neighborhood)} Let $X \subset V$ and $x \in V$. The geodesic neighborhood of $x$ with respect to $X$ of order $k$ is the set $N_{k}^{*}(x, X)$ defined recursively by:

$$N_{1}^{*}(x, X) = N_{1}^{*}(x) \cap X \text{ and } N_{k+1}^{*}(x, X) = \bigcup \{N_{k}(y) \cap N_{k}^{*}(x, X), y \in N_{k}^{*-1}(x, X)\},$$

where $M = 8$ in 2D and $M = 26$ in 3D.

\textbf{Definition 2 (Topological Numbers)} Let $X \subset V$ and $x \in V$. The
topological numbers of the point \( x \) relative to the set \( X \) are: 
\[
\begin{align*}
T_4(x, X) &= \#C_4(N^3_4(x, X)) \quad \text{and} \quad T_6(x, X) = \#C_6(N^3_6(x, X)) \quad \text{in} \ 2D; \\
T_8(x, X) &= \#C_8(N^3_8(x, X)), \\
T_{6^+}(x, X) &= \#C_6(N^3_6(x, X)), \\
T_{18}(x, X) &= \#C_{18}(N^3_{18}(x, X)), \\
\text{and} \quad T_{20}(x, X) &= \#C_{20}(N^3_{20}(x, X)) \quad \text{in} \ 3D, \text{where} \ # \ \text{denotes the cardinality of a set.}
\end{align*}
\]

We note that in the above definition of topological numbers in the 3D case, there are two notations for 6-connectivity. This follows the convention introduced in [Ber94], wherein the notation “6+” implies 6-connectivity whose dual connectivity is 18, while the notation “6” implies 6-connectivity whose dual connectivity is 26. This distinction is needed in order to correctly compute topological numbers under 6-connectivity, and does not imply a different definition of connectivity.

Topological numbers are used to classify the topology type of a grid point, especially for the characterization of simple points. A point is simple if its addition to or removal from the object does not change the topology of either the object or the background, in other words, it does not change the number of connected components, the number of cavities, and the number of handles of both the foreground and the background. It is proven in [Ber94] that a point \( x \) is simple if and only if \( T_n(x, X) = 1 \) and \( T_\alpha(x, \hat{X}) = 1 \), where \((n, \alpha)\) is a pair of compatible connectivities.

### 2.3 Isocontour algorithms

The design of a geometric deformable model is not complete without studying the isocontour algorithm that produces an explicit representation of the final contour from the embedding level set function. The choice of a suitable isocontour algorithm is especially critical for our topology-preserving framework where the algorithm must faithfully recover the topology of the implicit contour from the discrete samples of the level set function. In the following, we will focus on the marching cubes isosurface algorithm, and show a modification that is necessary to make it consistent with our topology preservation principle.

The marching cubes (MC) algorithm is a standard isosurface algorithm that produces a triangulated surface whose vertices lie on the edges of the cubic lattice [LC87]. As shown in Fig. 1, the way in which an isosurface intersects a cube is not always unique, which results in the so-called ambiguous face and ambiguous cube cases. The major difference between different MC algorithms lies in how they choose between the two possible tilings for the ambiguous cases. A well-accepted criterion is that the surface tiling should correctly reflect the topology of the true implicit surface. Under the assumption that the embedding function is densely sampled so that it is approximately linear on each cube, face saddle points and body saddle points can be used to produce isosurfaces that are topologically consistent with the embedded implicit surfaces [Nat94]. We note that the saddle points are the critical points of the embedding function — that is, the points where
the first order derivatives of the function are all zero.

In this work, we need an isosurface algorithm that can correctly represent the topology of a binary object under the prescribed digital connectivity rule. For this purpose, we proposed the use of a connectivity consistent MC (CCMC) algorithm [HXP01]. In this algorithm, the coordinates of surface intersections are still computed using linear interpolation, but the resolution of ambiguous faces and cubes now depends on the pre-defined digital connectivity rule. In particular, we choose the tilings in Figs. 1(c) and 1(e) for the corresponding ambiguous cases respectively if the black points are assumed to be 18-connected while the white points are 6-connected. If the black points are 26-connected, then Figs. 1(c) and 1(f) should be used instead.

The corresponding isocontour algorithm in 2D can be called the connectivity consistent marching squares (CCMS) algorithm. The only ambiguous case that needs special care is an ambiguous square (e.g., the front face of the cube in Fig. 1(a)). The correct tiling should separate the white points while connect the black ones if the black points are 8-connected, and vice versa.

3 Topology Preserving Geometric Deformable Model

3.1 Algorithm

Our topology preserving mechanism exploits the binary nature of the object that is delineated by the level set function. In this framework, topology
changes at the zero level set are directly related to the sign changes of the level set function. A constraint can then be imposed to keep the topology unchanged while the implicit contour deforms. The resulting deformable model behaves exactly as the unconstrained model except at places where topology changes can occur. In particular, the new model still deforms continuously, and sub-pixel accuracy is maintained.

Assume that the implicit contour is embedded as the zero level set of a level set function. Then at a given time step each grid point is either inside or outside the contour depending on the sign of the level set function at that point. We arbitrarily assign points with zero distance to be inside the contour. In this way, the implicit contour defines a unique digital object on the computational grid lattice, which consists of all the inside points. We can then relate the topology change of the implicit contour to that of the digital object. The topology of the digital object can only change if the inside/outside status at a point is changed or, equivalently, if the level set function changes sign at a grid point. Therefore, to preserve the topology of the object and hence the topology of the implicit contour, the level set function can only be allowed to change sign at simple points. This is the principle of the topology preserving constraint that we impose. We now give a detailed description of its implementation.

We start with the narrow band implementation of geometric deformable models since it is computationally fast, but modify it to conform to the topology constraint. In the following implementation, it is necessary to store a binary-valued indicator function \( B \), defined on the digital grid, to record the dynamically changing digital object. For a grid point \( x_i \), \( B(x_i) \) equals 1 if \( \Phi(x_i, t_m) \leq 0 \), and equals 0 otherwise, where \( t_m \) is the last time the point \( x_i \) is visited. The array \( B(\cdot) \) is initialized using \( \Phi(\cdot, 0) \), and is updated whenever the level set function \( \Phi \) undergoes a sign change at a grid point \( x_i \). The algorithm is summarized below, where \( x_i \) is used to denote a general grid point and \( y_i \) denotes a narrow band point.

**Algorithm 1 (Topology Preserving Narrow Band Algorithm)**

1. **Initialize** — Set \( m = 0 \) and \( t_m = 0 \). Initialize \( \Phi(\cdot, t_0) \) to be the signed distance function of the initial contour. Initialize the binary indicator array \( B \).

2. **Build the Narrow Band** — Find all grid points \( y_i, i \in \{1, \ldots, Q\} \) such that \( |\Phi(y_i, t_m)| < W_{nb} \), where \( W_{nb} \) is the user-specified narrow band width, and \( Q \) denotes the total number of narrow band points.

3. **Update** — For \( i = 1, \ldots, Q \), compute the level set function at the narrow band point \( y_i \) at time \( t_{m+1} = t_m + \Delta t \) by:

   (a) Using (1.4), compute \( \Phi_{\text{temp}}(y_i) = \Phi(y_i, t_m) + \Delta t \Phi(y_i, t_m). \)
(b) If \( \text{sign}(\Phi_{\text{temp}}(y_i)) = \text{sign}(\Phi(y_i, t_m)) \), then set \( \Phi(y_i, t_{m+1}) = \Phi_{\text{temp}}(y_i) \), keep \( B(y_i) \) unchanged, and go to Step 3(f). Otherwise continue to Step 3(c).

(c) Compute the topological numbers \( T_n(y_i, X) \) and \( T_n(y_i, \tilde{X}) \), where \((n, \tilde{n})\) is the chosen digital connectivity pair, \( X = \{ x_i | B(x_i) = 1 \} \), and \( \tilde{X} = \{ x_i | B(x_i) = 0 \} \).

(d) If the point is simple — i.e., \( T_n(y_i, X) = T_n(y_i, \tilde{X}) = 1 \) — then set \( \Phi(y_i, t_{m+1}) = \Phi_{\text{temp}}(y_i), \ B(y_i) := (B(y_i) + 1) \mod 2 \), and go to Step 3(f). Otherwise continue to Step 3(e).

(e) Point \( y_i \) is not simple. To preserve the topology, we do not allow the sign change and set \( \Phi(y_i, t_{m+1}) = \epsilon \cdot \text{sign}(\Phi(y_i, t_m)) \), where \( \epsilon \) is a small positive number used to avoid the arbitrariness of a zero value. Also, \( B \) does not change.

(f) Increase \( i \). If \( i > Q \), go to Step 4.

4. \textit{Reinitialize} — If the zero level set of \( \Phi(\cdot, t_{m+1}) \) is near the boundary of the current narrow band, reinitialize \( \Phi(\cdot, t_{m+1}) \) to be the signed distance function of its own zero level set.

5. \textit{Convergence Test} — Test whether the zero level set stops movement.
   If yes, stop; otherwise, set \( m = m + 1 \). If reinitialization is performed in Step 4, then go back to Step 2 to rebuild the narrow band; otherwise, go back to Step 3.

Note that in the above algorithm, the \textit{sign} function \( \text{sign}(\cdot) \) returns a value of \(-1\) if its argument is zero, which reflects our convention that a zero-valued grid point belongs to the interior of the zero level set.

Compared with a standard narrow band algorithm \cite{Cho93, AS95}, our new algorithm differs only in the updating step, which performs a simple point criterion check whenever the level set function is going to change sign at a grid point. The sign change is prohibited if the point is not a simple point, and the evolution of the level set function at that point is truncated. We want to point out that there can be some arbitrariness in the specific result of this algorithm depending on the order in which the points are visited in the narrow band. This situation is well known in skeletonization algorithms where the result depends on the order of simple point removal. Currently, we do not have a criterion to prefer one ordering over another one; but, we feel that this problem is not as significant as in skeletonization since the overall motion of the deforming contour is controlled by the speed (force) terms. The simple point criterion only takes place at locations where topological changes are bound to occur otherwise, which is ordinarily a very small portion of the overall contour. In the results reported in this chapter, we followed exactly the same ordering as in the standard narrow band implementation, where points are ordered by their natural coordinates.
After the level set iterations have converged, we extract the final contour using the CCMC (or CCMS) algorithm. In these algorithms, the contour location is computed by linear interpolation of the level set function, but the tiling for the ambiguous cases is selected based on the chosen digital connectivity pair. If the level set function value is exactly zero at a grid point, it is explicitly changed to \(-\epsilon\) before interpolation to prevent a singularity in the resulting explicit contour. This prevents the anomalous “touching” or “point sharing” artifact that is produced by most other isocontour algorithms and has a negligible effect on the accuracy of the resulting contour.

3.2 Examples

In this section, we present several examples that apply our new topology preserving geometric deformable model in 2D and 3D. Since a new model can be obtained by imposing the topology preservation constraint on an existing geometric deformable model (GDM), we refer to a standard model without the topology constraint as an SGDM and the corresponding (i.e., with the same set of speed terms) topology preserving model as a TGDM. Note that for TGDM’s, the CCMC or CCMS algorithm must be used in order to correctly extract the final curves or surfaces from the level set function, while the SGDM’s require a standard isocontour algorithm (preferably using face saddle points in 2D and both face and body saddle points in 3D). In these examples, we chose \((n, \bar{n}) = (4, 8)\) as the pair of 2D digital connectivities and \((n, \bar{n}) = (18, 6)\) for 3D.

Fig. 2 shows a 2D example that illustrates the topology preservation ability of the TGDM model. Here, the underlying geometric deformable model was the geodesic active contour model of Eq. 1.1. Fig. 2(a) (and 2(e)) shows the original image (65 × 140 pixels) consisting of two circular cells placed side-by-side. The two initial curves are also shown in this figure. Figs. 2(b) and 2(c) show the location of the implicit curves of the SGDM at
FIGURE 3. Segmentation of a hand phantom using both SGDM and TGDM (see text for details).

an intermediate and the final stage. Because of the weak edge between the two cells, the two initial curves are merged into one final curve — an undesirable result in this example. Figs. 2(f) and 2(g) show the corresponding deformations when the topology preservation constraint is enforced (and otherwise the two models are identical in internal and external force terms). We note that TGDM keeps the two curves separated throughout the evolution and also correctly finds the boundary of each cell. Figs. 2(d) and 2(h) demonstrate the sub-pixel resolution of the SGDM and the TGDM results respectively.

Fig. 3 shows another 2D example in which the two deformable models of the previous example were applied to find the boundary of a hand-shaped object. The original image (220 × 190 pixels) and the initial curve are shown in Fig. 3(a) and also 3(e). Figs. 3(b) and 3(c) illustrate the deformation of the SGDM contour at an intermediate and the final stage. Again, without the topology preservation constraint, the initial curve changes topology and gives two separate curves as the final result [a larger outer curve and a disjoint inner curve as shown in Fig. 3(c) and zoomed up in Fig. 3(d)]. We note that the two middle fingers in this hand become one “finger” with a hole in it in the final segmentation. The corresponding deformations of the TGDM contour are illustrated in Figs. 3(f) and 3(g). TGDM keeps the boundary of each finger separated, and the final contour correctly reflects the shape of the hand, as can be seen clearly in the zoomed view of Fig. 3(h).

Next, we applied a 3D version of the geometric deformable model of Eq. (1.3) to find the boundary surface of the 3D object depicted in Fig. 4(a). The object is actually a piece of white matter (WM) segmented from a
magnetic resonance (MR) brain image. Due to data noise, the WM piece has a handle in it, which is the wrong topology from an anatomical standpoint. In fact, we desire a topology equivalent to that of a sphere. We applied both SGDM and TGDM starting from two different initializations: a large sphere that encloses the whole object and a small ellipsoid that intersects with the object. A 2D slice showing the object and the two initial surfaces is shown in Fig. 4(b).

Figs. 4(c) and 4(d) are the final surfaces obtained by SGDM. The two results are the same since standard geometric deformable models are insensitive to initialization. The final surfaces both have a handle, however, which is the incorrect topology. With the sphere as the initialization, TGDM gives the final surface shown in Fig. 4(e), and with the ellipsoid, it gives the result shown in Fig. 4(f). Both surfaces have the correct topology, but the topology is preserved in different ways. The surface obtained from the sphere initialization yields a thin membrane across the tunnel (dual of the handle as viewed from the background) through the original object, while the ellipsoid initialization makes a cut in the handle.

Our final example applies SGDM, TGDM, and a parametric deformable surface model (PDM) to extract the central cortical surface from an initial fuzzy segmentation of a brain MRI image volume. (More details about cortical surface reconstruction can be found in the next section.) In this experiment, we used exactly the same initialization, the same external forces and similar internal forces for the geometric deformable models as in the parametric model. The results are presented in Fig. 5.

Fig. 5(a) shows the final surface extracted from the parametric model. The SGDM and TGDM surfaces look very similar, but on close examination
there are important differences. The parametric model result, for example, has self-intersections as shown in Fig. 5(b), while the TGDM surface does not, as shown in Fig. 5(c). Also, the genus (number of handles) of the SGDM result is 40, while that of both the parametric model result and the TGDM result is 0. Thus, TGDM produces both the correct topology and a valid manifold; hence it is the only model that gives a legal cortical surface reconstruction.

4 Brain Cortical Surface Reconstruction

In this section we describe an automatic cortical surface reconstruction procedure that incorporates the topology preserving geometric deformable surface model. The method begins by finding a topologically correct representation of the WM isosurface that is close to the desired GM/WM interface. Three implementations of the new topology preserving geometric deformable surface model with well-chosen internal and external forces are then used to find three characteristic surfaces of the cortex in sequence. In the following, we first give a brief introduction of the brain cortex segmentation problem, and then present our method in detail.

4.1 Background

The brain cortex is a highly convoluted layer of gray matter (GM) that lies between the white matter (WM) and cerebrospinal fluid (CSF), as illustrated in Fig. 6. Reconstructing the surface of the cerebral cortex from magnetic resonance (MR) images is an important step in brain visualization, quantitative analysis of brain geometry, multimodal registration, surgical planning, and unfolding and mapping the cortex [DFS99, FSD99]. Conventionally, the interface between GM and WM is first sought, largely
because this interface is readily visualized by T1-weighted MR images. In some cases, the central cortical surface is sought because this surface best represents the overall cortical geometry and algorithms designed to find it tend to be robust to noise and other imaging artifacts. Despite their highly convoluted geometries, when either surface is connected across the corpus callosum and diencephalon, it has a topology equivalent to that of a sphere. Unfolding the cortex and mapping it to a sphere (or other topologically equivalent surface) can then be accomplished for visualization, measurement, and the establishment of a global coordinate system on the cortex.

Estimation of the interface between the GM and the cerebrospinal fluid (CSF), or the pial surface, is also an important step in the analysis of brain cortex [MKAE00, ZSSD99]. The GM/WM interface and the pial surface together establish a segmentation of the cortical gray matter, and the distance between the two surfaces measures the cortical thickness, which varies with cortical location.

In the literature, deformable models have been widely applied for the cortical surface reconstruction problem (e.g., [XPR+99, DFS99, MKAE00, ZSSD99]). However, they typically face difficulties caused by imaging noise, image intensity inhomogeneities, the partial volume effect, and the highly convoluted nature of the brain cortex itself. For example, due to the partial volume averaging effect, opposing banks of sulci are often blurred together, which makes it difficult to reconstruct the pial interface in these areas. Also, to accurately reflect the true geometry of the cortical surface, reconstructions that connect the hemispheres across the corpus callosum and diencephalon should not self-intersect and should be topologically equivalent to a sphere [DFS99, MKAE00]. Although a large amount of research effort has been spent on improving the performance of deformable models in this area, it remains a difficult problem to produce surface representations of
the brain cortex that are topologically correct, geometrically accurate, and non-intersecting. The method described here is a continuation of the work of [XPR+99], but improves the previous work in several aspects, which includes a better model initialization procedure and compensation for partial volume effect. A significant change was made by replacing the previous parametric deformable model by TGDM, which brings the advantage of speed, stability, and producing surfaces without self-intersections. We now describe the method in detail.

4.2 Preprocessing and surface initialization

The MR brain images were T1-weighted with voxel size 0.9375 × 0.9375 × 1.5 mm$^3$. The volumes were preprocessed to remove extracranial tissue, the cerebellum, and the brain stem. Each volume was then interpolated to isotropic voxels, and robustly segmented into GM, WM, and CSF fuzzy membership functions [PP99], which represent a partial volume segmentation of the image volume. These membership functions were used in all subsequent processing (rather than using the raw MR data) because they offer robustness to noise and the partial volume effect. The WM membership function was then automatically processed to fill the ventricles and subcortical GM structures such as the thalamus, hypothalamus, caudate nucleus, and putamen [HXRP01].

The filled WM volume was binarized with a threshold of 0.5, since a membership value of 0.5 indicates the interface between two classes. The binary volume was then processed to correct its topology as described in [HXBNP02]. The 0.5-isosurface of the resulting volume would yield a topologically correct explicit representation of the WM boundary surface if computed using the CCMC algorithm. We note that both the topology correction, the CCMC algorithm, and TGDM must assume the same underlying digital connectivity rule. In all the results presented herein, we assume 18-connectivity for the foreground object (i.e., the white matter) and 6-connectivity for the background.

We then embed the topology correct WM boundary surface into a signed distance function using the fast marching method. The topology preserving deformable surface model can then be applied to get the desired cortical surfaces. The general deformable model given by Eq. (1.3) is used to find all three surfaces, but the forces are designed differently, as we now discuss.

4.3 Nested surface reconstruction algorithm

GM/WM surface. The first TGDM model is targeted at the inner GM/WM surface. Although the initialization step already provides a very close and topologically correct representation of the GM/WM boundary, we need to smooth out its staircase artifacts while simultaneously maintaining the accuracy of the surface and keeping the correct topology. To accomplish
this, we use the curvature force and the regional signed pressure force but not the gradient vector flow force. The curvature force aims to regularize the surface, and is proportional to the mean curvature $\kappa$ of the surface. The signed pressure force is defined as

$$R(x) = 2\mu_{WM}(x) - 1,$$

where $\mu_{WM}$ denotes the new WM membership function obtained after filling the ventricles and subcortical GM structures. Since $\mu_{WM} \in [0, 1]$, $R(x)$ falls in the range of $[-1, 1]$. If $\mu_{WM} > 0.5$ then $R(x) > 0$, whereas if $\mu_{WM} < 0.5$ then $R(x) < 0$. Therefore, $R(x)$ provides outward balloon forces when the surface resides within the WM and inward forces when it resides outside the WM. Thus, $R(x)$ will force the surface to the GM/WM boundary. The weights for the two forces are chosen to be $\omega_R = 1$ and $\omega_e = -0.02$. These are presently determined empirically, and will be optimized in future work. After convergence, the final level set function, denoted by $\Phi_{in}$, gives an implicit representation of the inner cortical surface, i.e., the GM/WM interface.

**Anatomically consistent GM editing.** The GM/WM interface is relatively unaffected by the partial volume effect. Reconstruction of both the central and pial surfaces, however, is very difficult when opposing sulcal banks are blurred together. Before proceeding to estimate these surfaces, we present our anatomically consistent editing (ACE) approach, which automatically modifies the GM membership function in tight sulci to produce evidence of CSF where it is otherwise obscured by the partial volume effect.

The first step in ACE is to detect the medial axis of the sulci, as illustrated in Fig. 7(a). Since the level set function of a geometric deformable model is itself a signed distance function, we can use $\Phi_{in}$ to identify points that are outside the GM/WM interface and are equidistant to two (or more) GM/WM surface patches, that is, the outer skeletal points. In particular, we extract the outside skeleton by taking the discrete Laplacian of $\Phi_{in}$, clip negative values, set it to zero whenever $\Phi_{in}$ is negative, and then normalize the result to the range $[0, 1]$. Locations where the resulting function — which we denote by $L(x)$ — is large represent the outside skeleton.

We now use the outside skeleton function $L(x)$ to modify the original GM membership function. One problem that must be addressed is the presence of medial axes that approach the GM/WM interface. Although the normalized Laplacian decreases as it approaches the interface, we do not want to risk actually modifying the GM membership function very near its interior surface. Accordingly, modifications to the GM membership function are only made at a distance greater than 1 mm from the GM/WM interface — an arbitrary distance chosen because it represents a lower bound on cortical thickness. Following this rationale, we produce an ACE GM membership function as follows

$$\mu_{GM}(x) = \begin{cases} 
[1 - L(x)]\mu_{GM}(x) & \Phi_{in}(x) > 1\text{mm} \\
\mu_{GM}(x) & \text{otherwise} 
\end{cases}$$

(1.5)
The result of the GM editing is illustrated in Figs. 7(b) and 7(c), where a 2-D cross-sectional view is shown. Clearly, there is a marked improvement in the appearance of sulcal gaps in the ACE result as compared to the original GM membership function. We note that ACE does not make any assumptions about the maximum cortical thickness, so it will not restrict the cortical width anywhere in the cortex (cf. [MKAE00, ZSSD99]. We also note that it only has an effect within sulci, and even then only when there is actual gray matter on the outside skeleton.

Central cortical surface. The ACE GM membership function better represents the true anatomy of the brain cortex. We now use this membership function to derive forces for a second TGDM model to find the central surface of the cortex.

As shown in [XPR+99], the use of a gradient vector flow (GVF) force makes it easy to find the central layer of a thick sheet. Suppose we take the ACE GM membership function $\mu'_{\text{GM}}$ itself as an edge map, and compute the gradient vector flow from it. Then, the resulting GVF force $\mathbf{v}(\cdot)$ will point to the center of the thick “edge”, that is, the center of the GM sheet. We refer the reader to [XP98, XPR+99] for details about computing the GVF force from a given edge map.

To make sure the surface does not move out of the cortex, we also apply a regional force using [XPR+99]

$$R(x) = \begin{cases} 0, & \text{if } \|2\hat{\mu}'_{\text{WM}}(x) + \mu'_{\text{GM}}(x) - 1\| > 0.5; \\ 2\hat{\mu}'_{\text{WM}}(x) + \mu'_{\text{GM}} - 1, & \text{otherwise.} \end{cases} $$

This region force pushes the surface outward if it is in the WM, inward if in the CSF, but has no effect within the GM. Thus, the surface moves toward the central layer of the GM under the influence of GVF forces alone. We also apply a small curvature force to keep the surface smooth. The relative strengths of the individual forces are determined by their coefficients, which we fix to be $\omega_R = \omega_V = 1$ and $\omega_K = -0.02$ for all the brain studies.
Using these internal and external forces, the second TGDM is applied starting from the GM/WM surface $\Phi_{in}$. To ensure the correct relative geometry of the surfaces, we also apply a barrier constraint that prevents the central surface from staying inside the GM/WM interface (which could be otherwise caused by noise in GVF). In particular, during the narrow band update we do not allow the evolving level set function to get larger than the initialization $\Phi_{in}$ at any grid point. After convergence, the output level set function gives the signed distance from the central cortical surface, which is denoted by $\Phi_{central}$.

**Pial surface.** TGDM is now used with new forces to find the pial surface, starting from the central surface obtained in the last step. Here, we use a GVF force that is designed to transport the surface towards the GM/CSF interface. We first compute a new edge map $\text{edge}_{outer}$ that has large values at the GM/CSF interface, as follows

$$\text{edge}_{outer}(x) = \|\nabla(\mu'_GM(x) + \mu'_WM(x))\|.$$  

We then compute the new GVF force $v(\cdot)$ from the edge map $\text{edge}_{outer}$. To accelerate the movement of the deformable surface inside the GM and to help it stay at the GM/CSF interface, the following region force $R(x)$ is also used

$$R(x) = 2\mu'_{GM} - 1.$$  

This region force makes the surface expand if inside the GM and contract if outside the GM (i.e., inside the CSF). Mean curvature is still used as the internal force.

A third TGDM is run with the above forces and the weights as before: $\omega_R = \omega_v = 1$ and $\omega_x = -0.02$. We also apply a barrier constraint as in the previous section to make the outer surface stays outside of the central surface. The converged level set function of this TGDM model, denoted by $\Phi_{out}$, embeds the pial surface as its zero level set.

### 4.4 Results

We have applied our new cortical surface reconstruction brain method on 21 MR brain images obtained from the Baltimore Longitudinal Study on Aging [RGD+00]. Figs. 8(a)-(c) show the reconstructed GM/WM, central, and pial surfaces, respectively, for one of these data sets. Fig. 9 shows these surfaces overlaying various cross sections of the original MR data. From these figures, it can be seen that the estimated surfaces follow the folds of the cortex very well. It can also be seen that ACE helps the outer surface stay inside tight sulci.

To estimate the accuracy of the proposed method, we computed a set of landmark errors on six of the reconstructed central cortical surfaces. The landmarks, ten on each brain, were manually picked on several major sulci and gyri (see [XPR+99]). The landmark error was then computed as
the minimum distance between the given landmarks and the reconstructed surfaces. The overall average landmark error produced is about 0.87 mm, which shows significant improvement over our previously reported 1.22 mm in [XPR+99], where a parametric deformable surface model was used to extract the central surface and without using the ACE. The new method also exhibits substantially increased stability, as indicated by a much lower standard deviation of 0.50 mm in average as compared to the previous 1.01 mm (see [HXTP01] for details).

We also ran a program to detect self-intersections on the final surface meshes. The new results have no self-intersections anywhere, as expected. In contrast, most cortical surface reconstructions obtained from parametric deformable models that do not explicitly prohibit them will have self-intersections. This is especially true for the pial surface, where many parts of the surface are essentially back-to-back in the tight sulci. Those methods that explicitly prohibit self-intersections (cf. [DFS99] and [MKAE00]) are very computationally intensive. Another way to try to reduce self-intersections with parametric models is to increase the internal forces, but this can have a very negative effect on the surface accuracy. Our method prevents self-intersections while simultaneously maintaining low internal forces to permit highly accurate (subvoxel) surface reconstructions.

Given the initial fuzzy segmentation, our method currently takes about 40 minutes on an SGI O2 workstation (174 MHz, R10000 processor) to reconstruct all the three surfaces. Thus, it is comparable to the geometric deformable model described in [ZSSD99], and much faster than parametric models reported in the literature, even those that do not impose self-intersection detection. Given that most algorithms currently report many hours to obtain similar results, we believe that our algorithm represents an important development for brain image analysis.
5 Conclusion

Deformable models constitute a set of powerful image segmentation tools. The two class of deformable models, parametric and geometric, both have their advantages and disadvantages with respect to specific applications. In recent years, many efforts have been conducted to overcome their shortcomings and combine their advantages. In this work, we propose a method to overcome the topology arbitrariness of standard geometric deformable models, which is critical for applications where topology correctness is of major concern. The significance of the proposed technique is demonstrated by its application in the brain cortex segmentation tasks. Future work includes reducing the sensitivity to model initialization and investigating other areas of applications.

Acknowledgments: We thank Drs. Susan Resnick and Dzung Pham for their guidance in using the brain image data. This work was supported in part by NSF/ERC grant CISST#9731748 and NIH/NINDS grant R01NS37747.
6 References


