Topology-preserving Geometric Deformable Model on Adaptive Quadtree Grid

Ying Bai*, Xiao Han†, and Jerry L. Prince*
*Johns Hopkins University, Baltimore MD 21218
† CMS, Inc., St. Louis, MO 63132

Abstract

Topology-preserving geometric deformable models (TGDMs) are used to segment objects that have a known topology. Their accuracy is inherently limited, however, by the resolution of the underlying computational grid. Although this can be overcome by using fine-resolution grids, both the computational cost and the size of the resulting contour increase dramatically. In order to maintain computational efficiency and to keep the contour size manageable, we have developed a new framework, termed QTGDMs, for topology-preserving geometric deformable models on balanced quadtree grids (BQGs). In order to do this, definitions and concepts from digital topology on regular grids were extended to BQGs so that characterization of simple points could be made. Other issues critical to the implementation of geometric deformable models are also addressed and a strategy for adapting a BQG during contour evolution is presented. We demonstrate the performance of the QTGDM method using both mathematical phantoms and real medical images.

1. Introduction

Geometric deformable models (GDMs) [17, 14, 2] are very successful in image segmentation because of their computational stability, topological flexibility, and innate ability to generate simple surfaces without self-intersections. Topology preserving geometric deformable models were recently introduced in order to provide the ability to maintain topology of segmented objects [20, 21, 6] while preserving the other benefits of GDMs. For example, in medical imaging many organs to be segmented have boundary topologies equivalent to that of a sphere. While many applications such as visualization and quantification may not require topologically correct segmentations, there are some applications — e.g., surface mapping and flattening and shape atlas generation — that cannot be achieved without correct topology of the segmented objects.

In Han *et al.* [20], it was observed that the implicit contour of an evolving level set function is homeomorphic to the boundary of the digital object it represents (all grid nodes with non-positive signed distance function values). Accordingly, their topology-preserving geometric deformable model (TGDM) maintains the topology of the implicit contour by controlling the topology of the digital object. This is achieved by applying the *simple point* criterion [3] from the theory of digital topology [16], preventing the level set function from changing sign at non-simple points. In this framework, implicit contours produced by TGDM cannot intersect an edge connecting two grid points more than once, which limits the achievable result resolution of TGDM, as shown in Figs. 1(a)-(c).

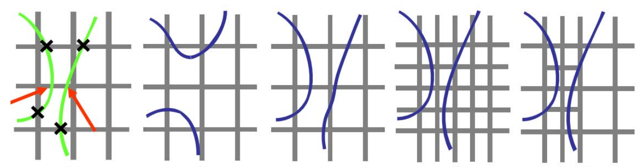

Figure 1. Resolution problem of level set methods. (a) contours irrepresentable due to implicit embedding; (b) SGDM changes topology; (c) TGDM keeps the contours separated by grid nodes; (d) double sized grid resolves the desired contour; (e) quadtree-based adaptive grid also correctly resolves the desired contour.

One way to achieve higher resolution using TGDM is to use a fine-resolution grid, as shown in Fig. 1(d). However, this strategy dramatically increases the computation time and yields much larger contours — i.e., number of vertices generated from the isocontour algorithm applied to the final level set function. Although a multi-resolution scheme [7] can improve the computational efficiency, the resulting contour is not spatially adaptive. One way to address this problem is to use *adaptive grid techniques* [12], which locally refine or deform the grid to concentrate computational efforts where more accuracy is needed. For example, a 2D *moving grid* TGDM method was introduced in [19]. Although improved resolution and topology preservation was demonstrated, this approach proved to be computationally demanding. *Adaptive local refinement* is another adaptive grid technique that is widely used in achieving accurate solution of general PDE's [11, 9]. Local refinement can resolve the above resolution problem as shown

1-4244-1180-7/07/$25.00 ©2007 IEEE

in Fig. 1(e) and the process is computationally efficient. There is extensive literature on adaptive level set methods [4, 9, 11, 10, 8, 18], but no topology preservation mechanism has yet been worked out.

In this paper, we propose a new topology-preserving level set method based on the balanced quadtree grids (BQGs – quadtree grids for which the maximum cell edge length ratio between adjacent grid cells is 2). Using the digital topology framework for the adaptive grid that we recently proposed [22], we are able to define a new characterization of simple points that extends the original characterization on the uniform grid in [3]. We then developed a topology preserving geometric deformable model for adaptive quadtree grids (QTGDM) and demonstrated its behavior and relative performance using both computational phantoms and real medical images.

2. Digital topology on adaptive quadtree grid

The main theoretical development herein is the characterization of "simple point" on BQGs, which is based on the digital topology framework for adaptive grids in [22] and is a generalization of the analogous concepts defined for the uniform grid in [3] . In the following, we first briefly review the basic concepts of neighborhoods and connectivity on BQGs and the difficulties that occur on the interface of cells having different resolutions, and then propose the new characterization of the simple point criterion.

2.1. Neighbors and invalid cases

The concept of *neighbor points* is fundamental in classical digital topology theory [16]. Neighborhoods, defined using distances on the discrete grid, must be defined differently on an adaptive grid because the notion of unit distance is different for cells at different resolutions. In [22], grid points on a quadtree grid are defined to be *edge(E)-neighbors* or *square(S)-neighbors* if they share an edge or a face, respectively, of leaf cells of the quadtree (i.e., cells that have no child cells). Fig. 2 shows an example of neighborhoods on a BQG. The left panel shows a uniform neighborhood and the right panel shows a non-uniform neighborhood on a BQG. The white circle is the root point of the neighborhoods; black squares are the E-neighbors; and white squares are the points that are added to the E-neighbors to yield the S-neighbors. On a balanced quadtree grid, two E-neighbors can exist in the same direction, although they are connected to the root point by different leaf cell edges. Analogous definitions of *neighborhood*, *adjacency*, *path*, and *connectivity* can be found in [22].

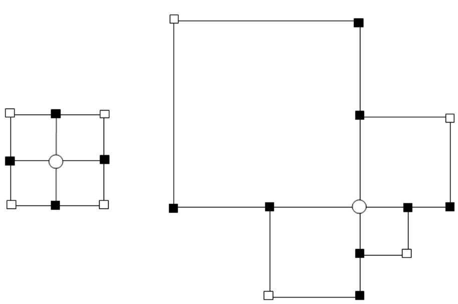

Figure 2. 2D neighborhoods on balanced quadtree grids.

Fig. 3 illustrates a problem in defining unique contour embedding in the quadtree grid. In this figure, the two white points are assumed to belong to the background. They are E-adjacent because they are connected by an edge belonging to the leaf cell on the left. Since the two points are both in the background, there cannot be a contour intersecting any portion of the edge between them. This follows from the principle of digital embedding relative to the coarse cell on the left. The black foreground point defined on the two finer cells on the right, however, implies that there should actually be two contour intersections (indicated by crosses) on the edges of the two leaf cells on the right. This situation is paradoxical and violates the digital embedding principle. We therefore define such level set configurations to be *invalid* [22], and they are not allowed on the quadtree. Because of this, during evolution of a level set function implementing QTGDM, we must prevent both topology changes and invalid configurations.

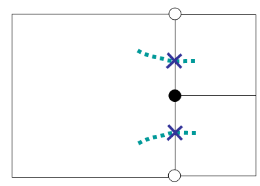

Figure 3. connectivity inconsistency on a transition edge.

2.2. Simple point characterization on adaptive quadtree grid

An efficient algorithm to determine a simple point on a *uniform grid* was presented in [3]. The method requires the definition of a geodesic neighborhood and topological numbers. In this section we follow the spirit of [3] to characterize a simple point on BQGs. Let us denote the domain of digital images on a BQG to be Ω, and the n-neighborhood of a point x on a BQG by $N_n(x)$, and the set comprising the neighborhood of x with x removed by $N_n^*(x)$, where $n \in \{E, S\}$. We define *geodesic neighborhood* and *topological numbers* on BQGs as follows:

DEFINITION 2.1 ***(Geodesic Neighborhood)*** *Let $X \subset \Omega$ and $x \in \Omega$. The geodesic neighborhood of x with respect to X of order k is the set $N_n^k(x, X)$ defined recursively by: $N_n^1(x, X) = N_n^*(x) \cap X$ and $N_n^k(x, X) = \cup\{N_n(y) \cap N_M^*(x) \cap X, y \in N_n^{k-1}(x, X)\}$, where $M = S$ in the balanced quadtree grid.*

DEFINITION 2.2 ***(Topological Numbers)*** *Let $X \subset \Omega$ and $x \in \Omega$. The topological numbers of the point x relative to the set X are: $T_E(x, X) = \#C_E(N_E^2(x, X))$ and $T_S(x, X) = \#C_S(N_S^1(x, X))$ in the balanced quadtree grid, where $C_n(X)$ denotes the set composed of all the n-connected components of X, and $\#$ denotes the cardinality of a set.*

Once the topological numbers are known, the following proposition gives a characterization of simple point on BQGs.

PROPOSITION 2.1 *A point x on a balanced quadtree grid is simple if and only if $T_n(x, X) = 1$ and $T_{\bar{n}}(x, \bar{X}) = 1$, where $(n, \bar{n})$ is a pair of compatible connectivities (cf. [22]) on the balanced quadtree grid.*

Fig. 4 illustrates the computation of topological numbers

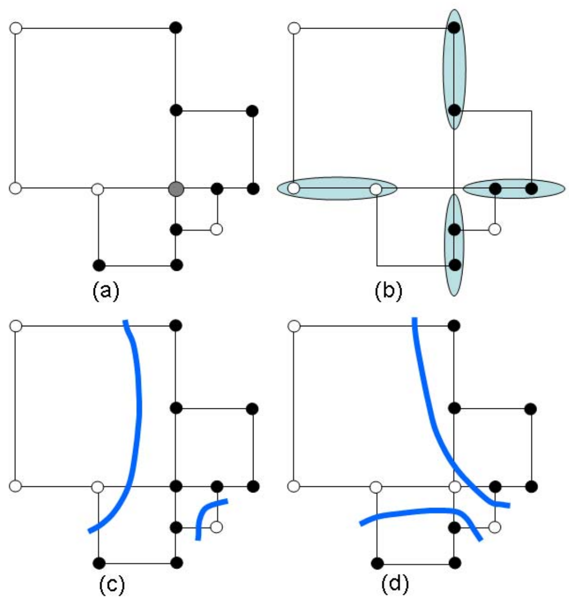

Figure 4. An example of constructing geodesic neighborhood on a BQG.

for a particular example. The root point is the gray point in the center of Fig. 4(a). All points in its neighborhood are marked as either black or white circles representing foreground and background respectively. Assume black circles have E-connectivity and white circles have S-connectivity. The highlighted black and white points in Fig. 4(b) are the first-order E-neighbors in the foreground (black) and the first-order S-neighbors in the background (white), respectively. The remaining points are second-order neighbors. A straightforward computation of the topological numbers requires counting the number of connected components within geodesic neighborhoods, which can be navigated by leaf cell edges on the adaptive grid. For example, when computing the foreground topological number in this case, we start from the gray point and search in the E-connected directions for the first-order neighbors in the foreground. When we search in the upper direction, we find the paired black points (as they are both one leaf cell edge away from the gray point). This pair of points is automatically counted as belonging to the same connected component. Next, the neighbors of these two points in the foreground inside the geodesic neighborhood are also counted into the same connected component, and so on. All the paired points in Fig. 4(b) are counted in this manner. In this example, $T_E(x, X) = 2$ and $T_S(x, \bar{X}) = 2$. Therefore the considered root point is not simple. Fig. 4(c) and Fig. 4(d) show how the topology of the implicit contour changes if the root point is changed from foreground to background.

The use of adaptive grid introduces a special type of points that require special consideration, which are known as *hanging points*. A *hanging point* is defined as a point that is only shared by two leaf cells, and has a one-sided neighborhood, as illustrated in Fig. 5. We can still apply the same

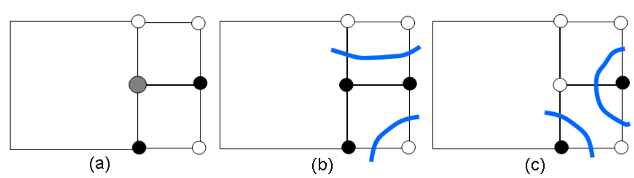

Figure 5. A non-simple hanging point on a BQG.

strategy to build its geodesic neighborhood and compute the topological numbers. In this case, the root point is a hanging point that is also non-simple with $T_E(x, X) = 2$ and $T_S(x, \bar{X}) = 2$. Fig. 5(b) and Fig. 5(c) show the topological change of the embedded implicit contours if this root point changes its status.

It is important to note that the above characterization of a simple point is only valid on a BQG that has no invalid configurations. Therefore, if a level set function is about to change sign at a given node, we must first check to see whether the sign change would create an invalid configuration; if not, then it is appropriate to check the simple point property. The validness constraint can sometimes create a "stuck" situation, as illustrated in Fig. 6. Suppose that both circled white points in Fig. 6(a) should change sign (according to forces acting on the active contour). If they are checked separately, then the hanging point will be determined to be non-simple, and the sign change at the corner point (non-hanging point) will be determined to be invalid — thus neither point can change sign. However, if the points were changed *together*, the two embedded contours indicating before (blue) and after (red) their sign change demonstrate no topological changes, which means their simultaneous sign change should be allowed. We solve this problem by grouping these two points together, and check

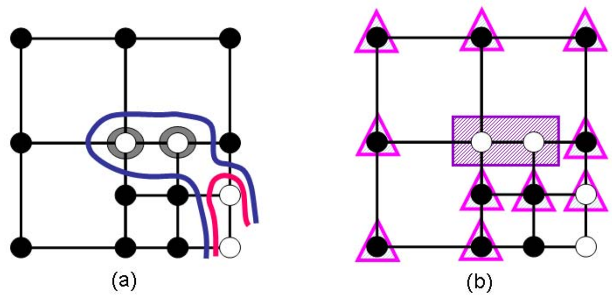

Figure 6. Grouping points when sign change is forbidden by invalid cases.

the criterion in a *union neighborhood* (using the neighbor points marked by triangles) as shown in Fig. 6(b). In this case, the union neighborhood has both foreground and background topological numbers equal to 1 indicating that the sign of the pair can be simultaneously changed. This strategy is only needed when a non-hanging point is forbidden to change sign because of invalidness. Thus, the cardinality of the set of grouped points can be no greater than 5 on a BQG; this is therefore computationally feasible.

3. Quadtree-based TGDM (QTGDM)

In this section, we present the implementation of QTGDM. The overall algorithm is first summarized and the details about several implementation issues are then discussed.

We adopt the narrow band framework [15] in the following implementation and we assume a general GDM model as can be summarized by the following equation:

$$\begin{aligned} \frac{\partial \Phi(\vec{x},t)}{\partial t} &= [F_{\mathtt{prop}}(\vec{x},t) + F_{\mathtt{curv}}(\vec{x},t)]|\nabla\Phi(\vec{x},t)| \\ &\quad + \vec{F}_{\mathtt{adv}}(\vec{x},t)\cdot\nabla\Phi(\vec{x},t) \end{aligned} \tag{1}$$

where $F_{\mathtt{prop}}, F_{\mathtt{curv}}$, and $\vec{F}_{\mathtt{adv}}$ denote user-designed force (or speed) terms that control the model deformation. In particular, $F_{\mathtt{curv}}$, the curvature force, controls the regularity (smoothness) of the implicit contour. $F_{\mathtt{prop}}$ and $\vec{F}_{\mathtt{adv}}$ are two forms of image forces (scalar and vector respectively) that drive the contour to the desired object boundary. The QTGDM algorithm is summarized as follows.

Algorithm 1 (QTGDM Algorithm)

1. Initialize the adaptive grid according to the initial contour topology and adaptation metric (see discussion below). Initialize the level set function to be the signed distance function of the initial contour.

2. Build the narrow band by finding all grid points within a distance threshold of the implicit contour (zero level set of the current level set function).

3. Update the level set function at each point in the narrow band iteratively as follows:

   (a) Compute the new value of $\Phi(\vec{x},t)$ using Eq. 1.

   (b) If there is no sign change, accept the new value and move on to the next point. Otherwise, go to Step 3(c).

   (c) Test whether the sign change at this point yields a valid configuration. If yes, go to Step 3(d). Otherwise, if it is a non-hanging point, group it with the neighbor hanging points that are causing the invalid configuration and go to Step 3(d); if it is a hanging point, move on to the next point.

   (d) Test whether the current point (group) is a simple point (group) by computing two topological numbers. If the point (group) is simple accept the new value. Otherwise, set the level set function to be a small number with the same sign as its original value.

4. If grid adaptation is needed (see below), apply a bottom-up topology-preserving merging followed by a top-down topology-preserving refinement according to a user-defined metric.

5. If the zero level set is near the boundary of the current narrow band, reinitialize the level set function to be a signed distance function and go to Step 2.

6. Test whether the zero level set has stopped moving (i.e., no sign change happens at any point inside the narrowband in two or three consecutive iterations). If yes, stop; otherwise, go to the next iteration. ■

A few comments about QTGDM. First, the reinitialization step is a straightforward extension of the fast marching method to the non-uniform cartesian grid. Different grid sizes are handled by the modified finite difference operator [15]. Second, the final contour must be computed using an adaptive connectivity-consistent marching squares algorithm (cf. [20, 22]) which prevents "cracks" and produces contours with the correct topology. Third, the simple point check can be omitted, and the algorithm becomes a standard geometric deformable model on an adaptive quadtree grid (QSGDM).

Grid adaptation metric

The grid adaptation scheme — how the leaf cell resolutions change in space — has a significant impact on the performance of QTGDM. For segmentation purposes, the metric to control the local distribution of grid nodes should be tailored according to the salient features and geometry of the target object. A widely used metric is the image gradient, as defined in [4]. The resulting computational grid is refined at high gradient regions and coarsened elsewhere. This metric, however, cannot help to reduce the size of the final contour on the adaptive quadtree grid since the grid will be uniformly refined along the object boundary. Our goal is to use coarse-resolution cells to represent "flat regions" and to use fine-resolution cells to represent "convoluted regions". Thus, we use a second-order measure – curvature of the image isophotes — as the metric for adaptation.

The classical definition of the curvature of an isocurve of an image I is given by:

$$\kappa = \mathbf{div}\left(\frac{\nabla I}{|\nabla I|}\right) = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{(I_x^2 + I_y^2)^{3/2}}$$

where I_x,I_y,I_{xx},I_{xy} and I_{yy} denote the first- and second-order partial derivatives of the image I. Proper discretization of this equation provides us with a method to estimate the curvature of the embedded iso-contour in the image. To achieve more robust estimation (in noise, for example), we apply an anisotropic smoothing to the image before the curvature estimation [13]. Once curvature κ is estimated, we define the refinement rule to be:

$$l(x) = i, \quad \text{if} \quad t_{i-1} < \frac{|\kappa(x)|}{\kappa_{max}} \leq t_i$$

where x denotes a quadtree grid node, $l(x)$ denotes the level of the leaf cells sharing node x, and κ_{max} denotes the maximum of $|\kappa(x)|$. If the highest level of the quadtree is l_{max}, then $i = 1, \ldots, l_{max}$. The t_i's are user-selectable thresholds to flexibly tune the grid resolution for different images. Fig. 7 shows one example of using the curvature map to construct an adaptive quadtree grid for the harmonic disk phantom image shown in Fig. 7(a). Fig. 7(b) shows the normalized estimated curvature map; Fig. 7(c) shows the constructed adaptive grid overlaid on the image; Fig. 7(d) shows a closeup view of a "finger" tip of the phantom. It can be seen that the grid resolution is finer in the high-curvature ridge and valley regions, and is coarser in the flat bank regions.

One advantage of using these image-driven metrics are that the adaptive grid can sometimes be generated only once before running the GDM, which improves efficiency of the overall method. Dynamic grid adaptation is still needed, however, in situations when the location of the object boundary is hard to predict from the image features. In such cases, we may need to update the metric using information from both the image and the level set function embedding the evolving contour.

Initial contour topology

To guarantee that the final contour has the correct topology, we must start with an initial contour with the correct topology. For example, assuming the topology of the object to be segmented is equivalent to a circle, we can simply start with a circular curve which can be easily initialized on an adaptive grid. However, because GDMs are typically only guaranteed to converge to local minima, it is usually desirable that the initial contour is as close to the final contour as possible so that the GDM can converge to the desired solution. Given a level set function on the uniform grid embedding an initial contour with possibly complex shape and topology, the task is to generate initialization on the adaptive quadtree grid such that the topology of the original initial curve is preserved. It turns out that we can adopt the topology-preserving cell-merging algorithm proposed in [22] to achieve this goal. The grid generation procedure should incorporate this algorithm prior to considering the adaptation metric.

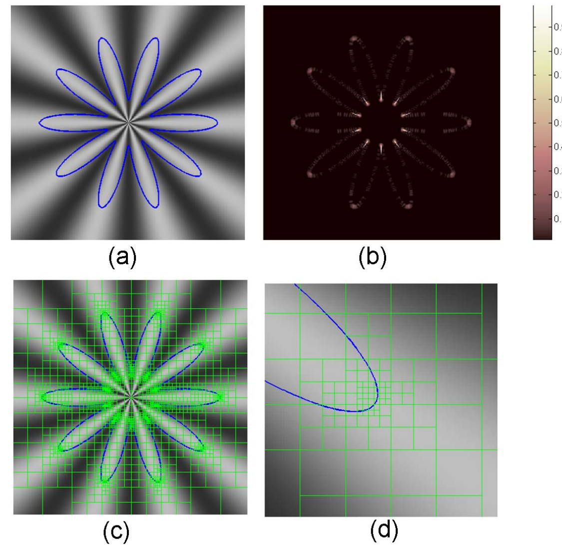

Figure 7. Illustration of grid adaptation for a harmonic disk phantom image.

Dynamic grid refinement with topology constraint

In general segmentation tasks, it is sometimes necessary to use dynamic adaptive grid refinement, where the quadtree discretization adaptively follows the front propagation of the implicit curve, concentrating the computational effort in the area where it is most needed. In order to preserve topology during grid adaptation, we need to design a coarsening and refinement strategy that incorporates a topology constraint. For grid coarsening, we can apply a bottom-up topology-preserving cell-merging algorithm in [22]. For grid refinement, we use a top-down topology-preserving cell-splitting algorithm that has a proper interpolation scheme to initialize the values at the newly generated grid nodes. In particular, when a parent cell is split into four children cells, we use linear interpolation to compute values at all the new nodes except for the case of an ambiguous cell in which the two pairs of diagonal nodes have different sign, causing a topological ambiguity (cf. [20]). To guarantee there is no topology change in refining an ambiguous cell, we must enforce the center node to have the same sign as the two corner nodes that are S-connected, by only aver-

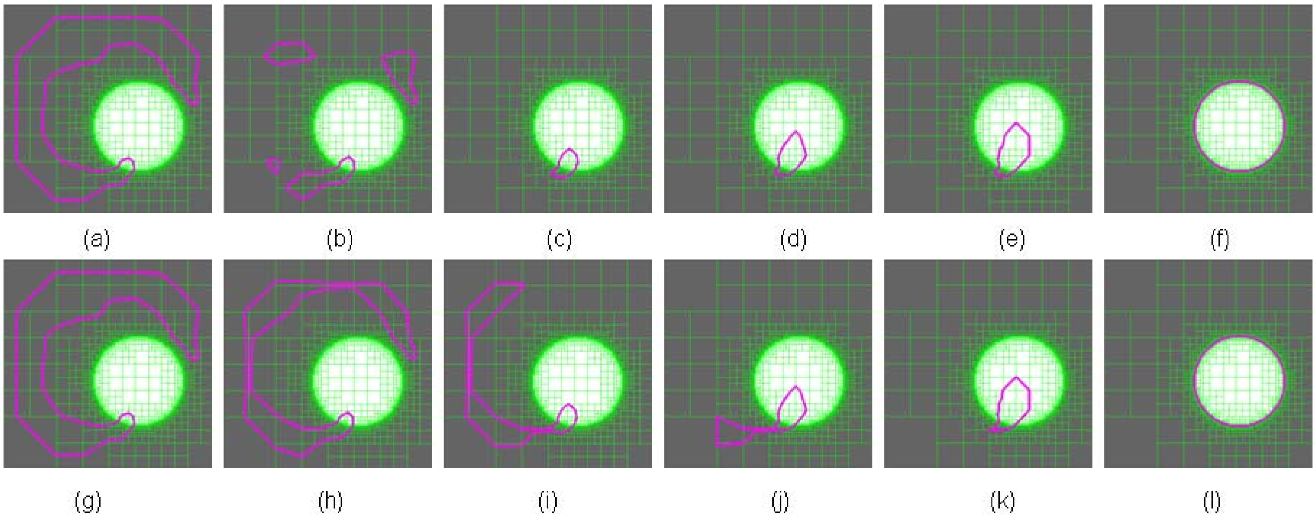

Figure 8. Circle experiment: (a)-(f) propagation of SGDM on a BQG; (e)-(i) propagation of TGDM on a BQG.

aging the values of these two corner nodes to initialize the center node.

4. Experiments

In this section, we present several experiments to demonstrate the benefits of using QTGDM.

In the first experiment, we used a circle phantom image and a fixed BQG to compare the behavior of QSGDM and QTGDM. Both models apply a region force that expands inside the white circular cell and contracts outside. The top row in Fig. 8 shows the propagation of QSGDM on a BQG. It changes topology after the first iteration, later splitting into four curves, as shown in Fig. 8(b). After that three of the curves disappear, as shown in Fig. 8(c). On the other hand, as shown in the bottom row, QTGDM maintains the same topology throughout its evolution. In this case, both algorithms achieve the same result in the end.

In the second experiment, we tested QTGDM on a phantom image comprising two ellipses. The image is of size 256×256. In Fig. 9(a), the initial contours are shown as red curves. The blue curves show the SGDM result using a computational grid the same size of the original image. We use this result as the ground truth in comparison. We now chose a coarse computational grid of size 128×128, and the result is shown in Fig. 9(b). We then chose a coarser grid of size 64×64 and applied both SGDM and TGDM with the same forces as before. The results are shown in Fig. 9(c) and Fig. 9(d), respectively. Without a topology constraint, the two curves are wrongly merged in Fig. 9(c). Fig. 9(e) shows the QSGDM result. The adaptive grid we use has its finest resolution equivalent to the original image resolution, as shown in Fig. 9(f). We see that the grid is dense in highly curved regions where large errors or topology change can occur. Table 1 summarizes the error and contour vertex number of the above results. We see that the QSGDM result achieves better accuracy with fewer vertices as compared to the result from using uniform grid with size 128×128.

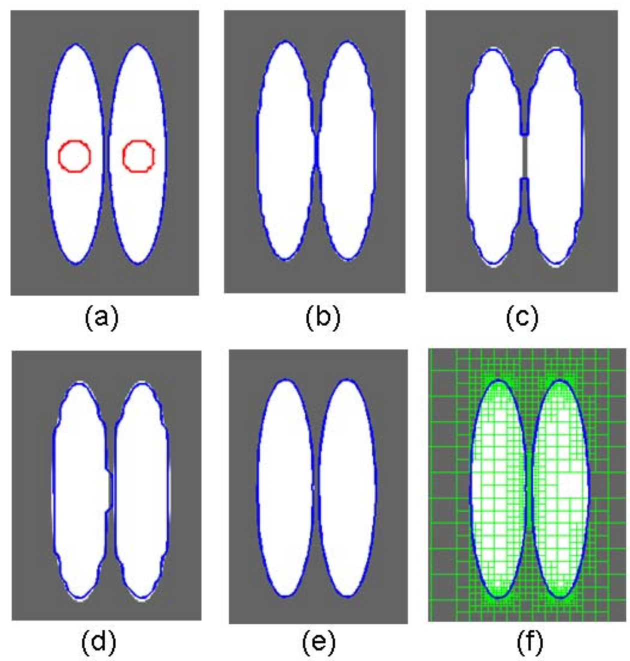

Figure 9. Ellipses experiment: (a) initial contour and 256×256 grid SGDM result as ground truth; (b) 128×128 grid SGDM result; (c) 64×64 grid SGDM result; (d) 64×64 grid TGDM result; (e) QSGDM result; (f) adaptive grid overlaid on result (e).

In the third experiment, a quantitative evaluation was

Table 1. Ellipses experiment results

	Mean Err	Max Err	Vertex Num	Savings
result (a)	0	0	698	0%
result (b)	0.288	1.259	350	49.8%
result (c)	1.01	9.41	161	76.9%
result (d)	0.70	2.70	172	75.3%
result (e)	0.154	0.689	254	63.6%

performed by using an image comprising a harmonic disk object. In this case, we know the true contour, and we apply a SGDM algorithm on different resolution uniform grids and our adaptive grid (with finest resolution equivalent to that of the finest uniform grid). The results are shown in Fig. 10 and the statistics are shown in Table 2. Results show that the adaptive grid method is fast, saves over 80% of vertices in the final contour, and achieves higher accuracy than the 128×128 sized grid.

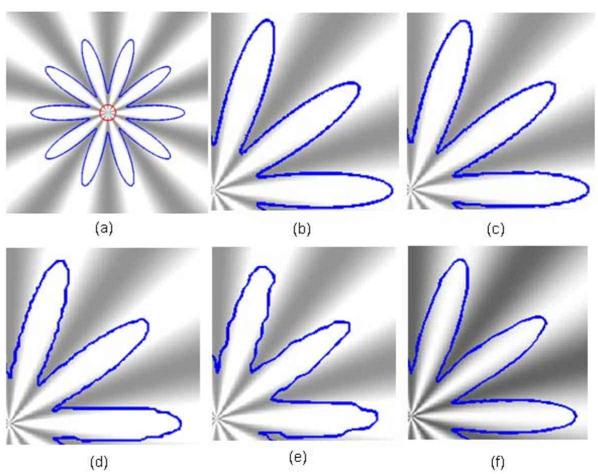

Figure 10. Harmonic disk experiment: (a) image phantom with initial contour (red) and truth final contour (blue); (b) 512 × 512 grid SGDM result; (c) 256 × 256 grid SGDM result; (d) 128 × 128 grid SGDM result; (e) 64 × 64 grid SGDM result; (f) QSGDM result.

Table 2. Harmonic disk experiment results

	Mean Err	Max Err	Vertex Num	Time (sec)
result (b)	0.08	0.52	3865	45.99
result (c)	0.20	1.13	1933	8.52
result (d)	0.57	3.05	963	2.22
result (e)	1.33	5.69	483	0.25
result (f)	0.36	1.46	718	4.58

In the fourth experiment, we applied QTGDM to segment a real CT image of carpal bones. We used a binary-flow GDM model [1], which tries to separate the mean intensity of the region inside the evolving contour from the mean of the outside. This approach requires dynamic grid refinement, as the underlying region force is changing as the curve evolves. Fig. 11(a) shows the initial contour and the initial adaptive grid overlaid on the image. Figs. 11(b)-(e) show a cropped view of the evolution of the curve and the associated dynamic grid adaptation in the bone joint part. Fig. 11(f) shows the final contour and the final adaptive grid overlaid on the image. Note that applying a uniform grid SGDM at original resolution on this image yields a segmentation with wrong topology (cf. [20]), due to the close adjacency of the two bones.

In the last experiment, we applied the proposed method to find the boundary of the white matter in a 2D slice of a high-resolution 512×512 MR brain image. Fig. 12(a) shows the original image with the initial contours. Figs. 12(b)-(d) are the close-up views that compare the results of a fine grid SGDM, a coarse grid SGDM, and a QSGDM. The yellow curve is the result on a uniform 512×512 grid that has 5266 vertices. The red curve is the result on a uniform 256×256 grid which has 2486 vertices. The blue curve is the QTGDM result (with finest resolution equivalent to the 512×512 grid resolution) which has 2317 vertices. We see that the blue curve follows the yellow curve better in most folded areas, while the red curve misses those anatomical features. Figs. 12(e)-(f) show two cases that cannot be represented by the coarser grid, but is correctly resolved by the adaptive grid.

5. Conclusion and Future Work

We have proposed a topology-preserving geometric deformable model for the adaptive quadtree grid (QTGDM); it is based on new digital topology concepts that we have developed for adaptive quadtree grids. Experiments show that the proposed method correctly preserves the digital topology of the implicit contour(s), saves computation time, and yields fewer vertices in the final contour(s). The proposed method has the potential to be extended into three dimensions in a consistent fashion and be used to further improve the accuracy of segmentation in volumetric data. Future work also includes addressing the issue of sensitivity to initialization as discussed in [5].

References

[1] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proc. ICCV'99*, pages 898–903, Corfu, Greece, 1999. 7

[2] A. Yezzi, S. Kichenassamy, P. Olver, and A. Tannenbaum. A geometric snake models for segmentation of medical imagery. *IEEE Trans. Med. Imag.*, 16:199–209, 1997. 1

[3] G. Bertrand. Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recognition Letters*, 15:1003–1011, 1994. 1, 2

[4] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Proc. CVPR'91*, pages 70–75, Lahaina, HI, 1991. 2, 4

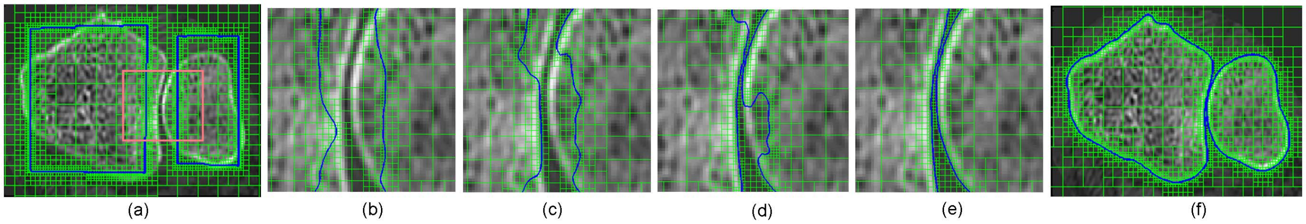

Figure 11. Carpal bones experiment: (a)initialization; (b) 4th iteration; (c) 8th iteration; (d) 16th iteration; (e) final iteration; (f) final result.

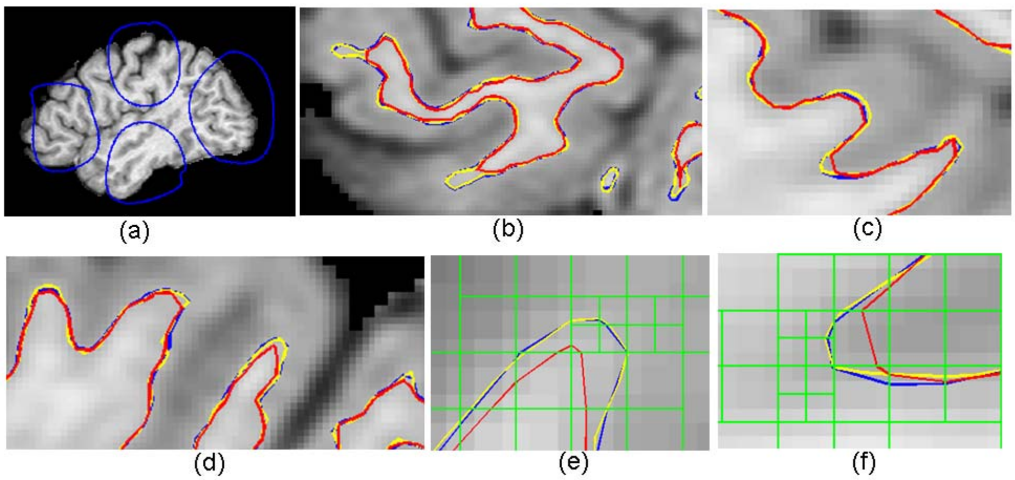

Figure 12. Cortical surfaces (see text for details).

[5] F. Segonne, J. P. Pons, E. Grimson, and B. Fischl . Active contour under topology control - genus preserving level sets. In *Proc. CVBIA 2005*, pages 135–145, 2005. 7

[6] Ganesh Sundaramoorthi and Anthony J. Yezzi. More-than-topology-preserving flows for active contours and polygons. In *ICCV*, pages 1276–1283, 2005. 1

[7] J. Weickert and G. Kuhne. Fast methods for implicit active contour models. In S. Osher and N. Paragios, editors, *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, 2003. 1

[8] M. Droske, B. Meyer, C. Schaller, and M. Rumpf. An adaptive level set method for medical image segmentation. In M. F. Insana and R. M. Leahy, editors, *Proc. IPMI 2001*, LNCS 2082, pages 416–422. Springer Verlag, 2001. 2

[9] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flow. *J. Comput. Phys.*, 148:81–124, 1999. 1, 2

[10] M. Xu, P. M. Thompson, and A. W. Toga. An adaptive level set segmentation on a triangulated mesh. *IEEE Trans. Medical Imaging*, 23(2):191–201, 2004. 2

[11] R. B. Milne. *Adaptive Level Sets Methods Interfaces*. PhD thesis, Dept. Math., UC Berkely, 1995. 1, 2

[12] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation*. CRC Press, Boca Raton, FL, 1994. 1

[13] R. Malladi and J. A. Sethian. Image processing via level set curvature flow. In *Proc. Natl. Acad. Sci. U.S.A.*, pages 7046–7050, 1995. 5

[14] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. PAMI*, 17:158–175, 1995. 1

[15] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, 1999. 4

[16] T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *CVGIP: Image Understanding*, 48:357–393, 1989. 1, 2

[17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1997. 1

[18] V. Sochnikov and S. Efrima. Level set calculations of the evolution of boundaries on a dynamically adaptive grid. *Int. J. Numer. Meth. Engng*, 56:1913–1929, 2003. 2

[19] X. Han, C. Xu, and J. L. Prince. A 2D moving grid geometric deformable model. In *Proc. of CVPR (CVPR'03)*, pages I:153–160, Madison, Wisconsin, June 2003. 1

[20] X. Han, C. Xu, and J. Prince. A topology preserving level set method for geometric deformable models. *IEEE Trans. Patt. Anal. Machine Intell.*, 25:755–768, 2003. 1, 4, 5, 7

[21] X. Han, D.L. Pham, D. Tosun, M.E. Rettmann, C. Xu, and J. L. Prince. Cruise: Cortical reconstruction using implicit surface evolution. *NeuroImage*, 23:997–1012, 2004. 1

[22] Y. Bai, X. Han, and J. L. Prince. Octree-based topology-preserving isosurface simplification. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2006)*, New York, June 2006. 2, 3, 4, 5