



Technical Section

A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images

Joo-Young Park^a, Tim McInerney^{b,c}, Demetri Terzopoulos^c,
Myoung-Hee Kim^{a*}

^a Department of Computer Science and Engineering, Ewha Womans University, 11-1, Daehyun-dong, Seodaemun-ku, Seoul, 120-750, South Korea

^b Department of Computer Science, Ryerson Polytechnic University, Rogers Communication Center, 80 Gould St., Toronto, Ont., Canada

^c Department of Computer Science, University of Toronto, L.P. Pratt Building, 6 King's College Rd., Toronto, Ont., Canada M5S 3H5

Abstract

This paper proposes a non-self-intersecting multiscale deformable surface model with an adaptive remeshing capability. The model is specifically designed to extract the three-dimensional boundaries of topologically simple but geometrically complex anatomical structures, especially those with deep concavities such as the brain, from volumetric medical images. The model successfully addresses three significant problems of conventional deformable models when dealing with such structures—sensitivity to model initialization, difficulties in dealing with severe object concavities, and model self-intersection. The first problem is addressed using a multiscale scheme, which extracts the boundaries of objects in a coarse-to-fine fashion by applying a multiscale deformable surface model to a multiresolution volume image pyramid. The second problem is addressed with adaptive remeshing, which progressively resamples the triangulated deformable surface model both globally and locally, matching its resolution to the levels of the volume image pyramid. Finally, the third problem is solved by including a non-self-intersection force among the customary internal and external forces in a physics-based model formulation. Our deformable surface model is more efficient, much less sensitive to initialization and spurious image features, more proficient in extracting boundary concavities, and not susceptible to self-intersections compared to most other models of its type. This paper presents results of applying our new deformable surface model to the extraction of a spherical surface with concavities from a computer-generated volume image and a brain cortical surface from a real MR volume image. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Deformable surface model; Multiscale approach; Adaptive remeshing; Non-self-intersection force; 3D boundary extraction

1. Introduction

Extracting the boundaries of anatomical structures from medical images is an important preprocessing step for a variety of medical applications, including visualization, quantitative analysis, motion tracking, and the registration of images obtained from two modalities. Traditional boundary extraction methods, such as edge detection, contour following, surface tracking, and isosurface generation via thresholding (e.g. Marching

*Corresponding author. Dept. of Computer Science and Engineering, Ewha Womans University, 11-1 Daehyun-dong, Seodaemun-ku, Seoul, 120-750, South Korea. Tel.: +82-2-3277-2315; fax: +82-2-3277-2306..

E-mail addresses: sindy@computer.org (J.-Y Park), tim@cs.toronto.edu (T. McInerney), dt@cs.toronto.edu dt@cs.nyu.edu (D. Terzopoulos), mhkim@mm.ewha.ac.kr (M.-H. Kim).

Cubes), are sensitive to noise and sampling artifacts and it is often difficult to generate closed, connected boundary surfaces.

In contrast, deformable models, which include the popular deformable contours or snakes [1] and deformable surfaces [2], have proved a powerful technique for the extraction of boundaries from medical images by combining the bottom-up approach of edge detection with the top-down approach of model-based geometric constraints. In general, the behavior of deformable model is governed by internal and external forces. Internal force describes the model as a physical object with stretchy and flexible characteristics. External force describes how the model is attracted to the image data. The model-based approach provides several desirable features such as inherent connectivity and smoothness that counteract noise and boundary irregularities, compact and analytic object representations, and the ability to incorporate prior knowledge of expected anatomic shape [3,4].

There are two possible approaches for extracting the boundaries of anatomical structures from volumetric medical images using deformable models. One is a slice-by-slice approach using 2D deformable contours. Starting with an initial image slice, a snake model is applied to extract the boundary contour of the structure. The resulting snake is then propagated to neighboring slices and used as an initial contour in these slices. This process is repeated until the entire 3D boundary surface is represented as a sequence of 2D contours generated from all slices which contain the object [5–7]. However, this approach often causes discontinuities or inconsistencies between neighboring slices and has difficulties extracting the contours near the first and last image slices bounding the anatomic structure. A fair amount of user interaction is required on many slices as the snake deforms in order to “pull” it out of an incorrect solution. The other approach is to extract the entire boundary surface of the structure all at once using a true 3D deformable surface model or “balloon”. This approach is more efficient, more robust against boundary irregularities and ensures a globally smooth and coherent surface between image slices [3]. Balloon models also maintain the ability of their 2D counterparts for intuitive user interaction via forces applied by the user on cross-sections of the model that are superposed on an image slice. In this paper, we focus exclusively on the 3D deformable surface model approach.

Deformable surface models in 3D were first introduced in computer vision [2] and computer graphics [8]. Numerous researchers [9–12] have since explored the application of these models to boundary extraction from volumetric medical images. Despite their many favorable properties for boundary extraction, most existing deformable surface models (and deform-

able contour models) suffer from three well-known problems:

- *Sensitivity to initialization*: Deformable models were originally designed as interactive models and relied upon the user to guide them toward the most appropriate local minima. Therefore, in more automatic boundary extraction scenarios, the initial model must usually be placed close to the boundaries of the target object to guarantee good performance [4]. If the initial model is placed far from the target object boundary, the model can settle into a local minimum solution that does not represent the true boundary. This is essentially due to the local nature of the external forces which determine the behavior of the models.
- *Inadaptability to boundary concavities*: The smoothness of deformable models and the connectivity of model elements are maintained by internal forces, which regularize edge lengths and minimize curvature. Smoothness and connectivity constraints allow a deformable model to bridge gaps in object boundaries and counteract noise and other boundary irregularities. At the same time however, strong internal forces can limit the geometric flexibility of a deformable model and prevent it from deforming into boundary concavities. Conversely, weak internal forces may allow the model to extract concavities but counteract the desired regularization effect. To improve the adaptability of a deformable model to boundary concavities without resorting to the use of weak internal forces, the model elements should be locally resampled based on the local geometry of the object.
- *Vulnerability to self-intersections*: When deformable models are deformed according to their internal and external forces, some model elements may collide with and intersect other elements. These model self-intersections generate non-simple surface geometries and result in the extraction of incorrect object boundaries. Most conventional deformable models have no mechanism to prevent self-intersections.

These difficulties have been major obstacles to the successful and automatic extraction of very complex-shaped objects. In this paper, we propose a new deformable surface model capable of extracting the boundary surfaces of geometrically complex internal organs, such as the cortex of the brain (which, although topologically simple, exhibits a convoluted boundary shape with severe concavities), from volumetric medical images. Our deformable surface model is represented as a topologically closed structure consisting of a mesh of triangular elements, similar to the geometric deformable model described in [12]. It is deformed by a combination of external and internal forces applied to the model

nodes (i.e. the vertices of the triangular mesh). Our model incorporates three mechanisms in order to overcome the limitations listed above:

- *Multiscale approach:* To overcome sensitivity to the model initialization, we follow [13] and use a pyramid structure of multiresolution image volumes, built bottom-up from the input image volume using successive smoothing processes to remove high frequencies. We then extract the boundaries of the object by applying a multiscale deformable surface model to the image pyramid in a coarse-to-fine fashion. This approach helps to ensure that a good solution, relatively independent of the model initialization, can be quickly and efficiently computed by hiding spurious image features until the model gets near a rough boundary approximation.
- *Adaptive remeshing:* To enable the model to deform into boundary concavities, we use global and local adaptive remeshing. The remeshing matches the resolution of the model to the resolution of the volume image at each level of the pyramid and regularizes the size of the triangular elements. Also, when a region of the model is pulled by external forces towards a desirable image feature, strong internal forces may limit its movement. Local adaptive remeshing refines the triangular mesh in this region, adapting the region shape to the geometry of the object. This ability allows the deformable surface model to extract narrow and deep boundary concavities. In addition, the local geometric operations used in the adaptive remeshing do not require a recursive or iterative process.
- *Self-intersection prevention:* To prevent model self-intersections, the physics-based formulation of our model integrates conventional internal and external forces, as well as a non-self-intersection force. If non-neighboring triangular elements of the model are close enough to intersect, the non-self-intersection force is applied to push the elements apart. This new force not only effectively prevents model self-intersections without a heavy computational load, but also helps the model to adapt boundary concavities (where model self-intersections often occur).

In the remainder of this paper we will first briefly review related deformable model research in Section 2. In Section 3, we will then present the overall procedure to extract object boundaries using our proposed model. Section 4 will describe our multiscale approach based on the volume image pyramid. Sections 5 and 6 will describe adaptive remeshing and our physics-based model formulation, respectively. In Section 7, we will present some preliminary results of applying our model to extract a phantom sphere surface with concavities and an MR brain cortical surface. Finally, Section 8 will

conclude the paper with a brief discussion and suggestions for future research.

2. Related work

There are three general forms of deformable models in the literature today: parametric, implicit, and discrete (i.e. triangular meshes or connected particle systems).

The original deformable models of Kass [1] and Terzopoulos [2,8] were parametric models, and many researchers [6,9–11] have also used this form of deformable model. Parametric deformable models can provide a compact and analytic description of object shape and high tolerance to image noise and large gaps in the object boundary. However, they are not well suited for extracting geometrically complex objects and can also exhibit sensitivity to their initial placement due to their fixed parameterization (i.e. different initial positions can result in different solutions).

Implicit deformable models [14–16] provide topological and geometrical flexibility in an effort to overcome the limitations of parametric models. They are best suited to the recovery of objects with complex shapes and unknown topologies. However, implicit models are more difficult to control than parametric models and discrete models [4] and are not amenable to local reparameterization.

Discrete deformable models [12,17,13,18–22] represent a contour and a surface as a set of vertices connected by edges, and control their deformation through local geometric operations (which may approximate physical forces such as bending or stretching) associated with each vertex rather than the more complex physical modeling of a continuous elastic or plastic structure. Their simple formulation is easily implementable and supports topological and geometric flexibility through the local geometric operations. In addition, the computational cost of a discrete deformable model is proportional to the size and complexity of the object, not the size of the image volume. For these reasons we represent our deformable surface model in discrete form.

Our work is related to the discrete deformable model of Miller et al. [12]. They proposed a closed deformable polyhedral surface model, or balloon, which grows (or shrinks) based on a set of constraints until the balloon reaches the boundary of the object in the volume image. A rough estimate of the object is quickly generated using a low resolution model and then the level of detail desired is achieved by global and local remeshing. However, they did not address the self-intersection problem. Chen and Medioni [18] proposed a dynamic mesh model which can handle complex, non-star-shaped objects without relying on a carefully selected initial state by using adaptive local triangle

mesh subdivision. They applied their model to range images only and again, do not address the self-intersection problem. In addition, neither of the models employed multiscale volume images combined with adaptive size meshing.

MacDonald et al. [17,21] generalized the cost functions defined in [12] and presented an iterative algorithm for simultaneous deformation of multiple curve and surface models with inter-surface constraints and self-intersection avoidance. The multiple models cannot only extract deep, narrow concavities successfully, but are also guaranteed to not self-intersect nor intersect each other. However, the computation of the inter- and intra-surface intersection avoidance constraints is prohibitively expensive and global adaptive remeshing only is employed.

Lachaud and Montanvert [13] proposed a discrete deformable model with automated topology changes and a coarse-to-fine approach to extract complex objects. The model can evolve to quickly extract objects relatively independently of the initial model position, and progressively extract finer and finer details consistent with the image resolution in a pyramid of multi-resolution volume images. However, their remeshing approach is more flexible and useful in automatically extracting objects with complex topologies rather than objects with complex boundary shape including severe concavities. Lachaud and Montanvert also use the distance between non-neighboring vertices as a constraint to detect self-intersections. However, to detect all self-intersections using only this constraint, a very small time-interval must be used, which is inefficient.

The multiscale approach of our model is based on the coarse-to-fine approach of Lachaud and Montanvert [13]. However, unlike their objective, we aim to extract objects with a simple sphere-like topology but highly complicated boundary shape. We have therefore designed adaptive remeshing operations and a physics-based force formulation specifically for this purpose.

3. Boundary extraction by proposed deformable surface

Fig. 1 illustrates the boundary extraction procedure using our proposed deformable surface model.

We first construct a volume image pyramid from the original volume image acquired by medical imaging modalities such as MR, CT, and so on. The volume image pyramid is a hierarchical structure of multi-resolution volume images where a lower resolution volume image is located in a higher level of the pyramid. We then set the level of the pyramid to the highest value and have the user interactively initialize the model by translating, rotating, and scaling an ellipsoid in this volume image. We convert the ellipsoid to a coarse triangulated mesh (an icosahedron) and begin the boundary surface extraction by iterating the following

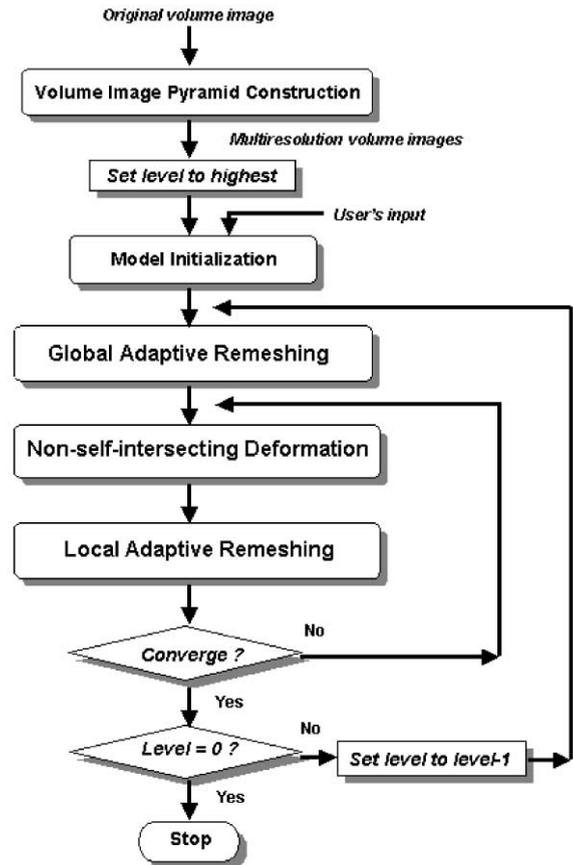


Fig. 1. 3D boundary extraction procedure by our deformable surface model.

three steps: global adaptive remeshing, non-self-intersecting deformation, and local adaptive remeshing.

Global adaptive remeshing refines the simple initial model to match its resolution to the image resolution of the pyramid level. The model is then deformed to extract the boundary surface of the target object using the non-self-intersection force formulation. After every deformation step, the triangular element sizes are regularized using local adaptive remeshing. If the model has converged to the boundary in the current level of the pyramid, we move to the next lower level of the pyramid and repeat these three steps to obtain a more accurate boundary extraction. Otherwise, the last two steps are repeated. These processes are repeated until the finest resolution model has converged to the target object boundary in the lowest level volume image of the pyramid.

4. Multiscale approach

Generally, the external image forces, which attract deformable models towards the boundaries of a target

object, are computed within a local image region around each model node. Therefore, deformable models often become stuck on spurious image edge features not associated with the global energy minimum or even a good local energy minimum (i.e. the true boundaries of the object). To avoid bad local minima, a deformable model must be initialized close to the desired boundary. Several global energy minimization methods have been proposed to address this problem for deformable contours, including simulated annealing, genetic algorithms, and Markov random fields. However, applying these global energy minimization methods to deformable surfaces is, in general, computationally very expensive.

We use a multiscale approach based on a volume image pyramid to provide a solution to this problem that is computationally efficient. As illustrated in Fig. 2, we process the input volume image into a multiresolution pyramid structure and adapt our deformable surface model to the multiresolution images in a coarse-to-fine fashion.

The multiscale approach allows the deformable model to pass over spurious image features and to quickly find a rough boundary approximation in the early stages of the deformation. This approach makes the model relatively independent of its initial condition. Moreover, the multiscale approach reduces the total computational time, as has been repeatedly demonstrated since the earliest application of multiscale techniques to visual reconstruction using (2.5D) deformable surfaces [23]. We describe the method to construct a volume image pyramid in the following subsection, and the method to refine the triangulated meshes in Section 5.

4.1. Volume image pyramid construction

A classical Gaussian pyramid was introduced by Burt et al. [24]. In the 3D Gaussian pyramid, the successive levels are blurred by the convolution with a Gaussian kernel of size 5 voxels. This guarantees low computational cost without phase translation linked to a reduction factor of 2 for each image dimension. However, the method does not consider that voxels may not be cubical and that the reduction factor of the image pyramid must be consistent with the surface remeshing [13]. Consequently, we use the algorithm developed by Lachaud [13], which creates volumetric pyramids of any reduction factor in order to support 3D isotropic convolution, hence consistency of the image filtering and model remeshing. We now describe the method briefly in the following paragraphs.

A 3D pyramid is represented as an ordered list of volumetric discrete images I_0, I_1, \dots, I_m . The original image I_0 is of discrete size (X, Y, Z) and of real size (X_R, Y_R, Z_R) . The coarsest image I_m includes only the lowest frequencies. I_h is the image at level h of the pyramid, and its discrete size is (X_h, Y_h, Z_h) . V is a real continuous image space and U_h is a unit of real space at level h . When the reduction factor of the image pyramid ρ is given, the sizes of the discrete image and the measurement unit at each level h are defined recursively as follows:

$$X_{h+1} = \left\lfloor \frac{X_h}{\rho} \right\rfloor, \quad Y_{h+1} = \left\lfloor \frac{Y_h}{\rho} \right\rfloor, \quad Z_{h+1} = \left\lfloor \frac{Z_h}{\rho} \right\rfloor, \quad (1)$$

$$U_{h+1} = \rho U_h, \quad (2)$$

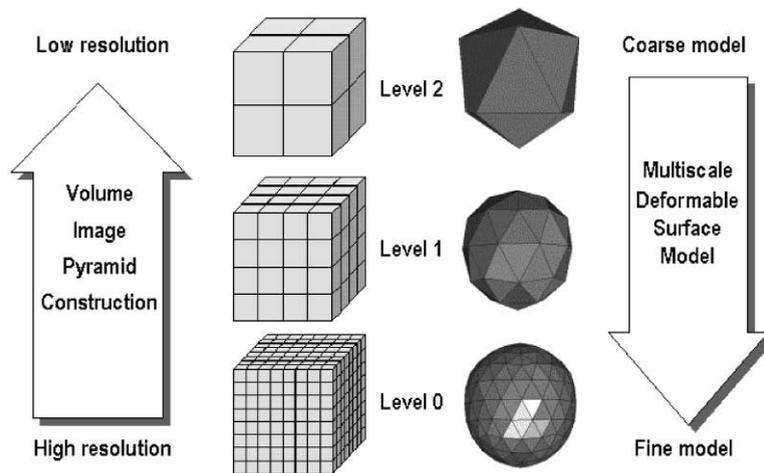


Fig. 2. Multiscale approach based on a volume image pyramid.

where $X_0 = X$, $Y_0 = Y$, $Z_0 = Z$ and $U_0 = \min(X_R/X, Y_R/Y, Z_R/Z)$.

The convolution operation for a voxel (x, y, z) of the discrete image I_{h+1} is defined over points of real image space V as follows:

$$I_{h+1}(x, y, z) = \sum_{i,j,k=-2}^2 w(i, j, k) \times V(T_h(x, y, z) + (iU_h, jU_h, kU_h)), \quad (3)$$

$$T_h(x, y, z) = \left(\left(x + \frac{1}{2} \right) \frac{X_R}{X_h}, \left(y + \frac{1}{2} \right) \frac{Y_R}{Y_h}, \left(z + \frac{1}{2} \right) \frac{Z_R}{Z_h} \right),$$

where $w(i, j, k)$ is a Gaussian convolution kernel of size 5 voxels: $(\frac{1}{16}[1 \ 4 \ 6 \ 4 \ 1])^3$. T_h is the transformation function to embed a voxel (x, y, z) of a discrete image I_h into the real image space V .

Fig. 3 shows an example of MR volume image pyramid constructed by reduction factor 2. Fig. 3(a)–(d) are the transaxial slice images selected from the MR brain volume at each level of the pyramid. The image at level $h + 1$ is generated from the image of level h by the convolution operation of Eq. (3). The original volume size at pyramid level 0 is $256 \times 256 \times 136$ and the volume sizes at level 1, 2, 3 are $128 \times 128 \times 68$, $64 \times 64 \times 34$, and $32 \times 32 \times 17$. The volume sizes at successive higher levels are reduced by half in each dimension while the unit voxel sizes are increased twofold in each dimension. The images at higher levels are progressively blurred. At the highest level, most of the image noise causing deformable model initialization problems has disappeared (see Fig. 3(d)).

The Gaussian filter used in Eq. (3) removes noise well. However, since it smoothes in all directions, edges and other sharp details are also blurred. Other nonlinear blurring strategies [25–27] can be considered to preserve features (such as edges, corners, etc.) adaptively.

5. Adaptive remeshing

In our multiscale approach, the resolution of the triangulated deformable surface model must match the resolution of the volume image at each level of the pyramid. That is, the triangular model elements should be neither bigger nor smaller than the voxels of the volume image. If the elements are too big, high-frequency boundaries including deep concavities could be missed. If the elements are too small, the model would include redundant information by representing a boundary surface within a voxel as several small elements. Therefore, we adapt the resolution of the model to the volume image pyramid in a coarse-to-fine fashion through global and local adaptive remeshing.

We use the edge lengths of the triangular elements to determine the resolution of the model. That is, the range of the edge lengths is adjusted in proportion to the unit voxel size U_h of real image space V at each level h of the image pyramid. The length of the unit voxel in the x direction, y direction, and z direction is U_h and its diagonal length is $\sqrt{3}U_h$ as shown in Fig. 4(a). We specify the minimum and maximum distance d_{\min}^h , d_{\max}^h which is allowed between adjacent nodes at each level h of the image pyramid as follows:

$$d_{\min}^h = U_h, \quad (4)$$

$$d_{\max}^h = 2\sqrt{3}U_h. \quad (5)$$

Thus, an edge is at least no shorter than the length of a unit voxel in the x, y, z direction (Fig. 4(a)) and at most no longer than twice the diagonal length of a unit voxel (Fig. 4(b)). This range allows neither the redundancies that more than one edge are included in a voxel, nor the details smoothed over by one edge spanning more than two voxels.

Global adaptive remeshing controls global resolution of the model by adjusting the average edge length of the model in this range. Local adaptive remeshing forces the

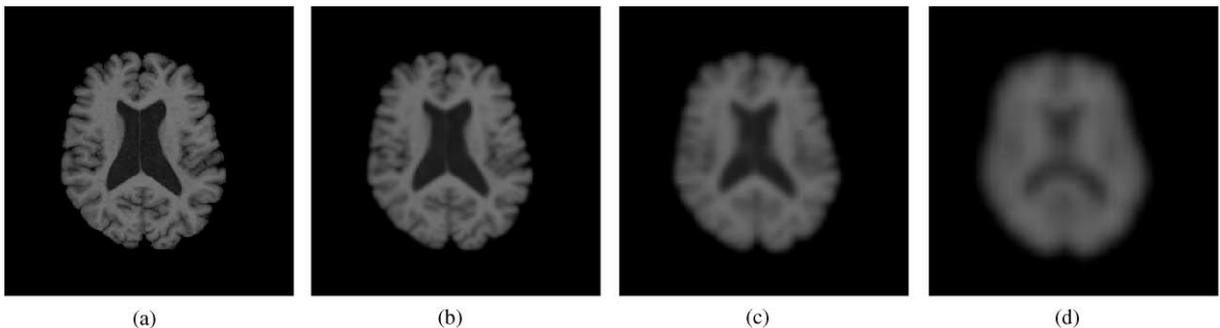


Fig. 3. An example of volume image pyramid. (a)–(d) Transaxial slice images selected from the MR brain volume data at each level 0, 1, 2, 3 of the pyramid.

model to remain regularly sampled during the deformation by keeping the edge lengths of each triangular element in this range.

5.1. Global adaptive remeshing

Our initial model is an icosahedron with 12 nodes, 20 triangles, and 30 edges. Before we begin deforming the

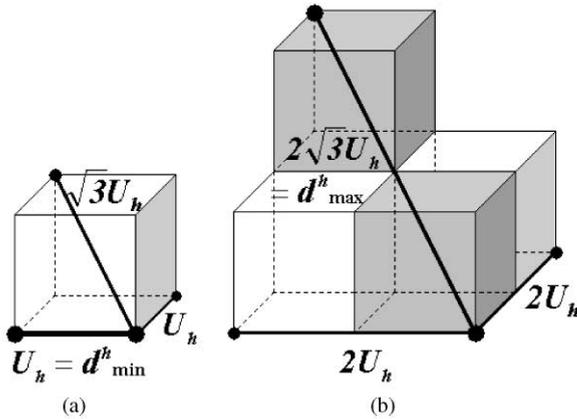


Fig. 4. Minimum (a) and maximum (b) edge length at the level h of the image pyramid.

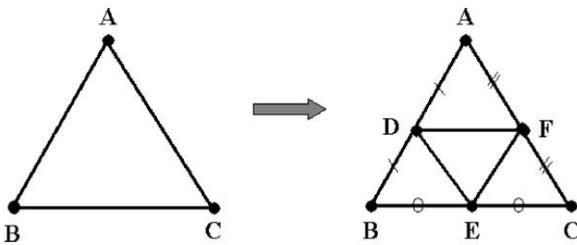


Fig. 5. Triangle subdivision for global adaptive remeshing.

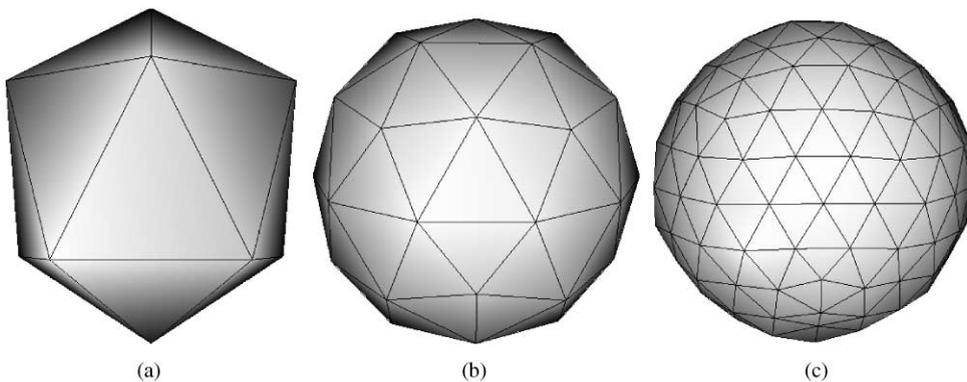


Fig. 6. Refinement of a triangulated surface model by global adaptive remeshing. (a) Initial icosahedron with 20 triangles. (b) (c) The models with 80 and 320 triangles refined after 1 and 2 iterations of global adaptive remeshing.

model, this initial model is globally remeshed to match the resolution of the coarsest volume image. Subsequently, we perform global adaptive remeshing upon each move to the next level of the pyramid. That is, the triangle edge lengths are adapted to the voxel size of the current volume image.

Global adaptive remeshing subdivides all triangles of the model uniformly until the average edge length of the model is smaller than the upper bound of the range allowed at a pyramid level h . That is, all model elements are subdivided uniformly until

$$L_h < d_{\max}^h, \tag{6}$$

where L_h is the average edge length of the model at a pyramid level h .

Fig. 5 illustrates how a triangle is subdivided during global remeshing. Each edge is bisected to form three new nodes. The new nodes are then connected in order to create four triangles from the original triangle. After every global adaptive remeshing, the number of the triangles increases fourfold, and the average edge length is halved. Thus, we set the reduction factor of the volume image pyramid to $\rho = 2$. Fig. 6 shows an initial icosahedron and the triangulated surface models refined after 1 and 2 iterations of the global adaptive remeshing.

5.2. Local adaptive remeshing

Global adaptive remeshing adapts the global resolution of the model to the resolution of the image only when moving from one level of the pyramid to the next and not during model deformation. It matches the average edge length to the volume image resolution. Consequently, once the model starts deforming within the current volume image, the local resolution of the model may change. That is, some triangles of the model may expand to extract the object of interest, while others may shrink resulting in regions of the model that are

irregularly sampled. If the sizes of triangles are irregular, internal forces may prevent large triangles from expanding further to capture highly curved boundaries or concavities. Furthermore, global remeshing at this point would subdivide triangles that have already accurately conformed to boundaries and therefore not only cause needless computation but may also cause intersections between the resulting adjacent small triangles.

Local adaptive remeshing adapts the local resolution of the model to the resolution of the image by keeping the edge lengths of all triangular elements in a range. It forces the triangulated surface to remain regularly sampled during the deformation. Therefore, it makes the model adaptable to geometrically complex boundaries by responding to image features flexibly.

Several authors have employed some operations for local remeshing in their deformable models [12,13,18,28,29]. In [12,18,28], the operations for subdividing overly large triangles were considered, but small triangles were not treated. In [12], large triangles are subdivided into four subtriangles and adjacent triangles are subdivided into two subtriangles to keep the topology consistent. This scheme was implemented by a two-stage recursive process. The recursive process requires considerable computational time and memory to maintain a stack. The subdivision algorithms of [18,28] consist of two steps, a bisection operation and a conformation operation. The conformation operation was intended to ensure that the desirable triangular mesh properties of conformity, non-degeneracy and smoothness are maintained. However, these operations are also recursively applied. The methods described in [13,29] include bisection and inversion operations for overly long edges as well as melting for overly short edges. In [29], the authors used the length of the three edges and the three angles of a triangle to find an ill-conditioned triangle. However, their remeshing algorithms also require iterative procedures.

In addition to operations for remeshing overly long or short edges, Lachaud and Montanvert [13] suggest axial and annular transformations for topological changes. They applied three geometric constraints to maintain mesh uniformity for self-intersection detection, as well as to allow topological transformations that include breaking a connected mesh into pieces and merging two meshes or two regions of the same mesh upon self-intersection. This approach is very flexible and useful in automatically extracting objects with complex topologies. However, the method that merges two mesh regions when self-intersection is detected is not appropriate when extracting concave areas such as brain sulci—an incorrect cortical boundary may be generated (i.e. instead of reconstructing the concavity it may be smoothed over).

Our local adaptive remeshing includes operations for both overly large and small triangles: *melting* (*edge-melting*, *triangle-melting*), *inversion*, and *subdivision* (*bisection*, *trisection*, or *quadsection*). However, we do not allow topological transformations for the reason mentioned above. Instead, when self-intersection is detected, we apply a non-self-intersection force (see Section 6). In addition, our local adaptive remeshing algorithm does not use expensive recursive or iterative processes.

Our local adaptive remeshing scheme checks the edge lengths of each triangle at every deformation step or once every several steps. For triangles with edges shorter than minimum length at pyramid level h , a *triangle-melting* or an *edge-melting* operation is applied. For triangles with edges longer than the maximum length, an *inversion* or a *subdivision* operation is performed. Formally, the local adaptive remeshing ensures the model satisfies the following condition:

$$d_{\min}^h \leq |\mathbf{x}_i - \mathbf{x}_k| \leq d_{\max}^h, \quad (7)$$

where $\mathbf{x}_i, \mathbf{x}_k$ ($i, k = 1, \dots, N$) are the positions of adjacent model nodes.

Local adaptive remeshing is implemented as a three-pass process. These three passes are performed once and do not require recursive or iterative processes. Three traces of all triangles are sufficient. Figs. 7 and 8 describes the local adaptive remeshing operations that may be applied at each pass.

- *Melting*: In the first pass, we trace all triangles of the model in order to find triangles with edges shorter than d_{\min}^h . If there is a triangle with only one edge shorter than d_{\min}^h and the other two edges satisfy condition (7), we perform an *edge-melting* operation. In Fig. 7(a), \overline{AB} of ΔABC is shorter than d_{\min}^h , but edges \overline{BC} , \overline{CA} are equal to or longer than d_{\min}^h . Thus, edge \overline{AB} is melted. For a triangle with more than two edges shorter than d_{\min}^h , we perform a *triangle-melting* operation as in Fig. 7(b). After this pass, all short edges violating condition (7) disappear.
- *Inversion*: The second pass traces all triangles to find pairs of long thin triangles with a shared edge longer than d_{\max}^h and all other edges satisfying condition (7). If such a pair is found, we invert the shared edge. In Fig. 8(a), ΔABC and ΔDCB share edge \overline{BC} which is longer than d_{\max}^h , while edges $\overline{AB}, \overline{CA}, \overline{BD}, \overline{DC}$ are equal to or shorter than d_{\max}^h . \overline{AD} also satisfies condition (7). The shared edge \overline{BC} is swapped into \overline{AD} by an *inversion* operation. This pass removes a most but not all overly large triangles from the model.
- *Subdivision*: The third pass checks all triangles to find overly large triangles untreated in the second pass. The triangles may have one or more edges longer than d_{\max}^h . For a triangle with just one such edge, we

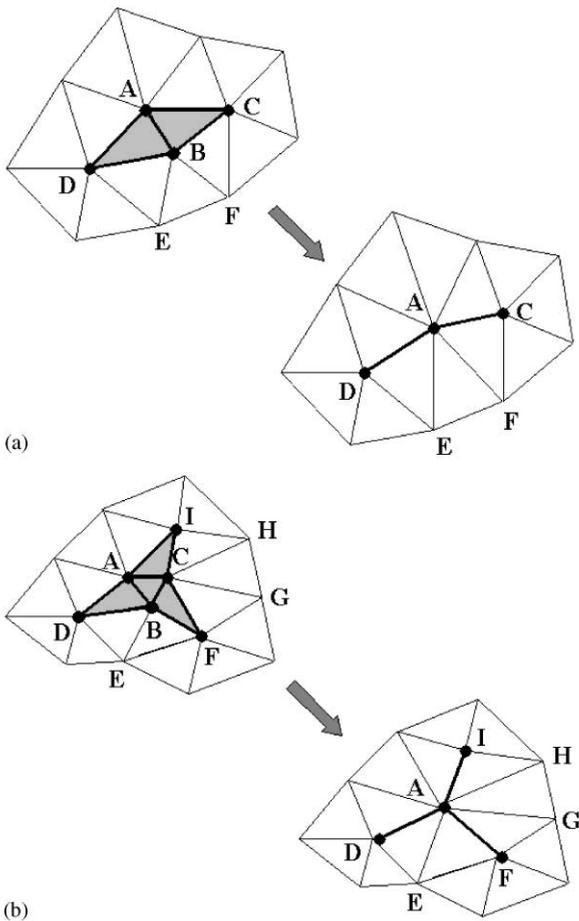


Fig. 7. Operations for local adaptive remeshing (1): (a) edge-melting, (b) triangle-melting.

perform a *bisection* operation. The *bisection* operation divides the triangle into two subtriangles by splitting the long edge into two equal parts. In Fig. 8(b), \overline{BC} is longer than d_{\max}^h and \overline{AB} , \overline{CA} satisfy condition (7). We bisect $\triangle ABC$ into $\triangle ABD$, $\triangle ADC$ by linking the midpoint D of \overline{BC} to node A . For a triangle with two edges longer than d_{\max}^h , we perform a *trisection* operation. The *trisection* operation divides the triangle into three subtriangles by splitting the longest and second longest edge in half. In Fig. 8(c), for $\overline{BC} > \overline{CA} > d_{\max}^h$ and $\overline{AB} \leq d_{\max}^h$, we link the midpoint D of the longest edge \overline{BC} to node A and then connect the midpoint E of \overline{CA} to D . For a triangle with all three edges longer than d_{\max}^h , a *quadsection* is executed as in the case of global adaptive remeshing (Fig. 5). Through this pass, all remaining overly large triangles are removed and all edges longer than d_{\max}^h are bisected. This process achieves *conformity* for the triangulation of the

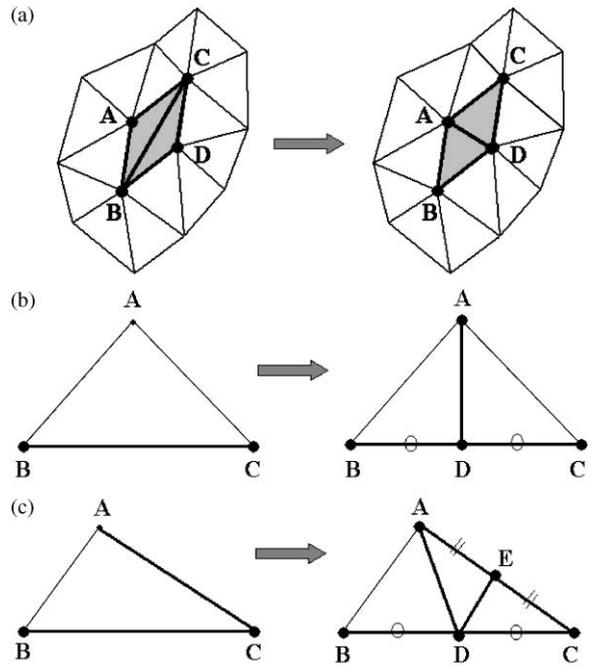


Fig. 8. Operations for local adaptive remeshing (2): (a) inversion, (b) bisection, (c) trisection.

model without the use of recursion. That is, any two adjacent triangles share only either a node or an edge without an additional conforming operation. The example of Fig. 9 shows the *subdivision* satisfies conformity.

Edge-melting, *triangle-melting* and *inversion* operations may cause degeneration of the triangular mesh. To prevent mesh degeneration, we use a constraint for minimum node connectivity (i.e. a node should be connected to at least three nodes). In Fig. 6(a), the melting of edge \overline{AB} decreases the connectivity of node C and D by one. If the connectivity of node C or D equals 3, the melting of edge \overline{AB} would cause a node connectivity conflict that the connectivity of node C or D becomes two. In Fig. 7(b), if the connectivity of node D or F or I equals 3, the melting of triangle $\triangle ABC$ would also cause a node connectivity conflict. Therefore, in Fig. 7(a), if the connectivity of node C equals 3 (i.e. node C is just connected to node A , B , and F , we replace three adjacent triangles $\triangle CAB$, $\triangle CBF$, and $\triangle CFA$ with one triangle $\triangle ABF$ before *edge-melting*. And also in Fig. 8(a), if the connectivity of node B or C equals 3, the *inversion* of edge \overline{BC} causes a node connectivity conflict. In this case, the *inversion* is rejected. We prevent mesh degeneration by checking the minimum node connectivity constraint before every *edge-melting*, *triangle-melting* and *inversion* operation, and then avoiding node connectivity conflicts.

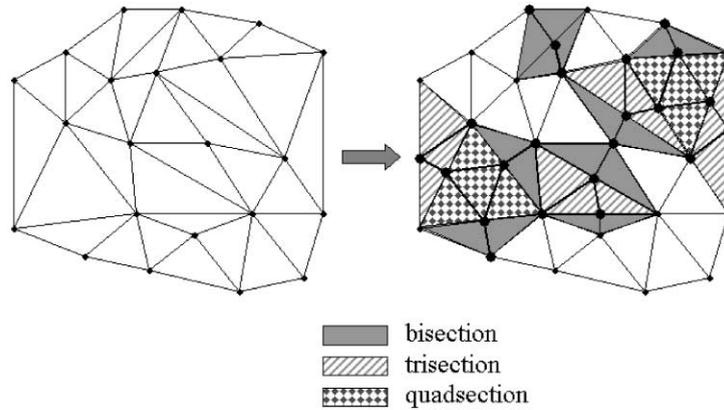


Fig. 9. An example of triangle subdivision satisfying conformity.

6. Force formulation

The behavior of our deformable model is governed by internal, external, and non-self-intersection forces. Internal forces provide flexible connections between model elements. External forces are computed from the image data in order to attract the model towards image features such as edges or contour boundaries. Non-self-intersection forces prevent the formation of non-simple structures due to self-intersection.

The dynamics of every node i ($i = 1, \dots, N$) in our model are described by the discrete Lagrange equations of motion

$$m_i \mathbf{a}_i + \gamma_i \mathbf{v}_i = \mathbf{f}_{\text{int},i} + \mathbf{f}_{\text{ext},i} + \mathbf{f}_{\text{nsi},i}, \quad (8)$$

where $\mathbf{x}_i = [x_i, y_i, z_i]$ is the position vector of the node i , $\mathbf{v}_i = d\mathbf{x}_i/dt$ and $\mathbf{a}_i = d^2\mathbf{x}_i/dt^2$ are the velocity and acceleration vectors of the node i , m_i is the mass of the node (we assume the same mass for all nodes), and γ_i is the damping coefficient which controls the rate of the dissipation of kinetic energy. $\mathbf{f}_{\text{int},i}$, $\mathbf{f}_{\text{ext},i}$, $\mathbf{f}_{\text{nsi},i}$ are internal, external and non-self-intersection forces respectively applied to node i . The computation of the forces is specified in the following subsections.

The initial position for each node i is specified by the user and its initial velocity is set to zero. We implement the actual deformation process as a numerical time integration process in which the whole state of the deformable model is calculated at a sequence of discrete positions in time, as in [19]. After computing the total force $\mathbf{f}_i(t)$ that acts on node i at each time step t , we then compute the current acceleration $\mathbf{a}_i(t)$, new velocity $\mathbf{v}_i(t + \Delta t)$,

and new position $\mathbf{x}_i(t + \Delta t)$ using the explicit Euler method

$$\begin{aligned} \mathbf{f}_i(t) &= \mathbf{f}_{\text{int},i}(t) + \mathbf{f}_{\text{ext},i}(t) + \mathbf{f}_{\text{nsi},i}(t) - \gamma_i \mathbf{v}_i(t), \\ \mathbf{a}_i(t) &= \mathbf{f}_i(t)/m_i, \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \mathbf{a}_i(t)\Delta t, \\ \mathbf{x}_i(t + \Delta t) &= \mathbf{x}_i(t) + \mathbf{v}_i(t + \Delta t)\Delta t. \end{aligned} \quad (9)$$

The deformation process of the model is continued until it is stabilized. We classify every model node into “active” and “inactive” based on its activity during last n time steps. If the ratio of “inactive” nodes to the total model nodes increases over a threshold value (δ), the model is regarded as stabilized.

6.1. Internal force

The internal force is intended to maintain the simple topology of the initial model. We define the internal force $\mathbf{f}_{\text{int},i}$ as a weighted combination of two forces: a stretching force $\mathbf{f}_{\text{st},i}$ and a bending force $\mathbf{f}_{\text{bd},i}$ where

$$\mathbf{f}_{\text{int},i} = w_{\text{st}} \mathbf{f}_{\text{st},i} + w_{\text{bd}} \mathbf{f}_{\text{bd},i} \quad (10)$$

and where w_{st} and w_{bd} are stretching and bending coefficients. The stretching force controls the distance between neighboring nodes and the bending force controls the curvature of the model surface. The internal force helps prevent the model from leaking out of gaps in the boundary of object, as well as helping to prevent the model from sticking to spurious image features.

The stretching force acts as if all nodes are linked to adjacent nodes by elastic springs [20]. The force increases as lengths between nodes are stretched or compressed relative to a rest length d_{rest} . Thus, the stretching force at node i acts to regularize the lengths

between itself and all adjacent nodes, where

$$\mathbf{f}_{st,i} = \sum_{j=1}^{n(\mathbf{x}_i)} \left[(|\mathbf{x}_{ij} - \mathbf{x}_i| - d_{rest}) \frac{(\mathbf{x}_{ij} - \mathbf{x}_i)}{|\mathbf{x}_{ij} - \mathbf{x}_i|} \right], \quad (11)$$

where \mathbf{x}_{ij} is the position of the j th adjacent node of \mathbf{x}_i , $j = 1, \dots, n(\mathbf{x}_i)$, and $n(\mathbf{x}_i)$ is the number of adjacent nodes of node i . We set the rest length d_{rest} to an average edge length for the whole triangles at time $t - \Delta t$, because edge lengths are regularized in a range by local adaptive remeshing.

The bending force essentially minimizes curvature. That is, it acts to smooth the model based on an estimate of local curvature. The discrete approximation to curvature can be computed in various ways [12,13,23]. We approximate curvature by computing the distance vector between a node position \mathbf{x}_i and the barycenter $\mathbf{c}(\mathbf{x}_i)$ of its neighboring nodes [13]:

$$\mathbf{f}_{bd,i} = \mathbf{c}(\mathbf{x}_i) - \mathbf{x}_i - \frac{1}{n(\mathbf{x}_i)} \sum_{j=1}^{n(\mathbf{x}_i)} (\mathbf{c}(\mathbf{x}_{ij}) - \mathbf{x}_{ij}), \quad (12)$$

where the barycenter $\mathbf{c}(\mathbf{x}_i)$ is computed as an average position of all neighboring nodes

$$\mathbf{c}(\mathbf{x}_i) = \frac{1}{n(\mathbf{x}_i)} \sum_{j=1}^{n(\mathbf{x}_i)} \mathbf{x}_{ij}. \quad (13)$$

6.2. External force

The external force $\mathbf{f}_{ext,i}$ is a weighted summation of two forces: an edge force $\mathbf{f}_{edg,i}$ and balloon force $\mathbf{f}_{bal,i}$

$$\mathbf{f}_{ext,i} = w_{edg} \mathbf{f}_{edg,i} + w_{bal} \mathbf{f}_{bal,i}, \quad (14)$$

where w_{edg} and w_{bal} are edge and balloon coefficients. The edge force is designed to attract the model towards significant 3D intensity edges in image. However, the edge force influences the model only near intensity edges. If the model is initially positioned far from the object boundary, it will not be attracted towards. For

this reason, we also use a ‘‘balloon inflation’’ force to push the model towards the object boundary.

The edge force is computed as follows:

$$\mathbf{f}_{edg,i} = \nabla |O_{MD} * I(\mathbf{x}_i)|, \quad (15)$$

where ∇ is the gradient operator and $O_{MD} * I(x, y, z)$ denotes the 3D intensity edge field produced from a volume image $I(x, y, z)$ by 3D Monga–Deriche (MD) edge-detection operator O_{MD} . We compute the gradient operator ∇ at any model points \mathbf{x}_i by tri-linear interpolation using values at the eight surrounding pixels.

The effectiveness of the edge force is dependent upon the performance of the edge detection operator. We use the 3D Monga–Deriche (MD) operator [30] because it is less sensitive to noise than other simple gradient or sobel edge detectors and thus detects real object boundaries and fewer spurious edges. Fig. 10 shows the 3D intensity edge fields detected from the MR brain volume images at each level of the pyramid (Fig. 3) by 3D Monga–Deriche operator.

The balloon force inflates or deflates the model locally until it lies within a region of the image with the desired intensity range, assuming that the image data can be separated into object and non-object. The balloon force takes the form

$$\mathbf{f}_{bal,i} = B(I(\mathbf{x}_i)) \mathbf{n}_i, \quad (16)$$

where \mathbf{n}_i is the unit normal vector to the model surface at the node \mathbf{x}_i , computed by normalizing the sum of the normal vectors of adjacent triangles. $B(I(\mathbf{x}_i))$ is the binary thresholding operator for the image intensity $I(\mathbf{x}_i)$.

$$B(I(\mathbf{x}_i)) = \begin{cases} (+1), & T_{low} \leq I(\mathbf{x}_i) \leq T_{high}, \\ (-1), & I(\mathbf{x}_i) < T_{low} \text{ or } I(\mathbf{x}_i) > T_{high}, \end{cases} \quad (17)$$

where T_{low} and T_{high} are the lower and upper thresholds for desired intensity values of the object of interest,

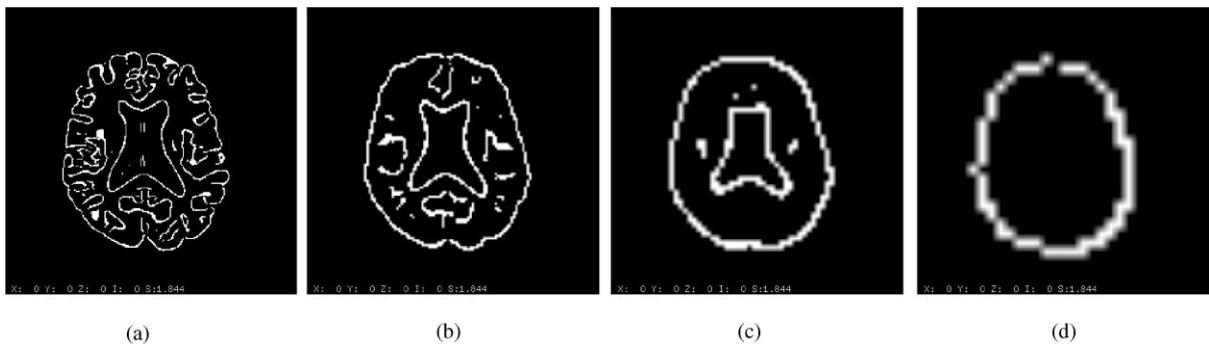


Fig. 10. The intensity edge fields detected from MR brain volume images at pyramid level 0, 1, 2, 3 (shown in Fig. 3) by 3D Monga–Deriche operator.

given by user. As the balloon coefficient w_{bal} is positive, the model will inflate on the desired intensity range, while the model will deflate out of the desired intensity range. If w_{bal} is negative, the model will act in the opposite direction.

6.3. Non-self-intersection force

The non-self-intersection force is intended to prevent a model element from intersecting other non-neighborhood model elements. At every deformation step we search for pairs of candidate model triangles that are close enough to intersect. We then apply the non-self-intersection force to separate the two triangles.

To find candidate triangle pairs, we check the smallest distance between all two non-neighborhood triangles in the model. The smallest distance $d_{\text{cls}}(T_k, T_l)$ between two triangles, T_k and T_l , is determined by computing the two points (one in each triangle) having minimum Euclidean distance. If the smallest distance $d_{\text{cls}}(T_k, T_l)$ is smaller than the minimum distance allowed for non-neighborhood triangles D_{min} , the non-self-intersection force of the triangle T_k against T_l , $\mathbf{f}_{\text{nsi}}(T_k, T_l)$ is computed as follows:

$$\mathbf{f}_{\text{nsi}}(T_k, T_l) = \begin{cases} \frac{\|d_{\text{cls}}(T_k, T_l) - D_{\text{min}}\|}{d_{\text{cls}}(T_k, T_l)} (\mathbf{p}(T_k, T_l) - \mathbf{p}(T_l, T_k)) & \text{if } d_{\text{cls}}(T_k, T_l) < D_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where $\mathbf{p}(T_k, T_l)$ is the point in T_k which is the closest to T_l , $d_{\text{cls}}(T_k, T_l)$ is the Euclidean distance between $\mathbf{p}(T_k, T_l)$ and $\mathbf{p}(T_l, T_k)$, and D_{min} should be at least as large as d_{min}^h . The force causes the two triangles to repel each other.

In practice, we apply the non-self-intersection force of the triangle T_k to its three nodes. The non-self-intersection force applied to a node i of the triangle T_k against T_l , $\mathbf{f}_{\text{nsi},i}(T_k, T_l)$, has the same direction to $\mathbf{f}_{\text{nsi}}(T_k, T_l)$, but the magnitude is adjusted according to the distance between the closest point $\mathbf{p}(T_k, T_l)$ and the node position \mathbf{x}_i ,

$$\mathbf{f}_{\text{nsi},i}(T_k, T_l) = w_{\text{nsi}} \frac{1}{\|\mathbf{p}(T_k, T_l) - \mathbf{x}_i\|} \mathbf{f}_{\text{nsi}}(T_k, T_l), \quad (19)$$

where w_{nsi} is non-self-intersection coefficient. In Fig. 11, the non-self-intersection force applied to node A is larger than the forces applied to B and C , because the distance of A from $\mathbf{p}(T_k, T_l)$ is closer than B and C .

Obviously, it would be prohibitively expensive to compute the distances between all non-neighborhood triangles in the model to find self-intersection candidates. In practice, we do not need to check distances between triangles which are very far away from each

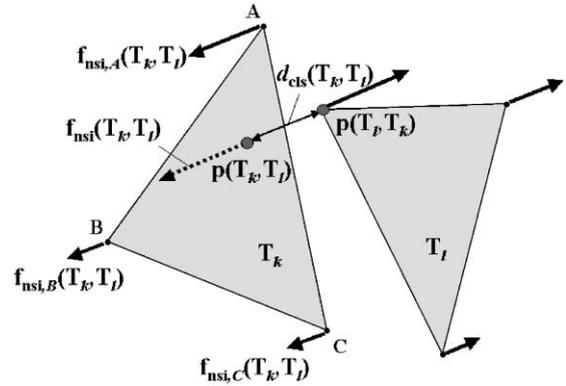


Fig. 11. Non-self-intersection force.

other. We use an octree structure of model nodes to search for self-intersection candidates efficiently by traversing 3D space hierarchically. An octree structure is a tree based on a node with eight children and each node of an octree represents a cube in physical space. Consequently, our non-self-intersection force ensures that the model can prevent self-intersections efficiently without heavy computational load.

7. Experiments

We have developed a prototype system of the described model on an SGI Octane/SE R10000 workstation and have used it to extract: (1) a sphere with concavities from a computer-generated volume image and (2) the complex brain cortical surface from a real MR volume image. We initialized our model using a superquadric function which can be quickly translated, scaled, and rotated by the user in the volume image (Fig. 12). The superquadric is then converted into an icosahedron which is then further remeshed globally to adjust the resolution of the model to the image resolution. This section presents preliminary results of these experiments.

7.1. Extraction of a sphere with concavities from a synthetic volume image

We first tested our model on a computer-generated volume image of a sphere with concavities. A volume image pyramid of 4 levels was constructed from the original volume image with dimensions $128 \times 128 \times 128$ using a reduction factor of 2. The size of the remaining levels of the image pyramid are $64 \times 64 \times 64$, $32 \times 32 \times 32$, and $16 \times 16 \times 16$. We set the deformation parameters to the following values: $w_{\text{st}} = 5.0$, $w_{\text{bd}} = 10.0$, $w_{\text{edg}} = 10.0$, $w_{\text{bal}} = 15.0$, $w_{\text{nsi}} = 10.0$. Local

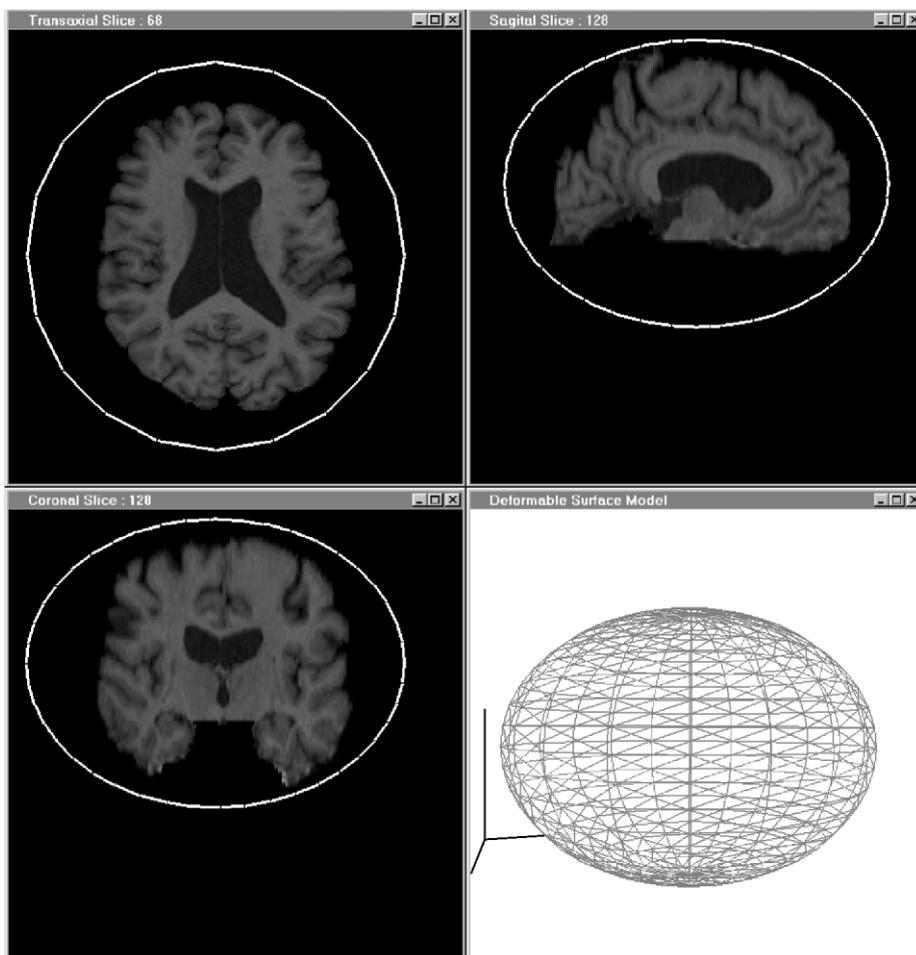


Fig. 12. Model initialization using a superquadric. Top left, top right, bottom left: the transaxial, sagittal, and coronal cross-sectional images of the volume data overlaid by the cross-sectional contours of the superquadric, determining initial position and size of the model. Bottom right: 3D view of the superquadric.

adaptive remeshing was executed every five deformation steps.

In Fig. 13, (a) shows an initial icosahedron and (c), (e), (g), (i) show the sphere boundary surface with concavities extracted at each level of the volume image pyramid. Fig. 13(b), (d), (f), (h) and (j) also shows the icosahedron and the models overlaid on the transaxial slice images at each level. At the coarsest resolution, shown in Fig. 13(c) and (d), the model found the rough shape of the sphere with 162 nodes and 320 triangles very quickly after 20 iterations. At finer resolutions, the more detailed concave structure was gradually extracted. The model was refined with 642 nodes and 1280 triangles after total of 50 iterations at level 2 (Fig. 13(e) and (f)) and with 3135 nodes and 6266 triangles after 140 iterations at level 1 (Fig. 13(g) and (h)). At the finest level after 150 iterations, a complete boundary surface

with a deep concavity was extracted without any model self-intersections as shown in Fig. 13(i) and (j). The final model contains 12,706 nodes and 25,408 triangles.

7.2. Extraction of brain cortical surface from a MR volume image

We applied our model to extract the cerebral cortex boundary from a volumetric MR brain image with dimensions $256 \times 256 \times 136$. The brain image has been preprocessed to remove the skin, bone, fat and other extracranial tissues. For the preprocessing, a semiautomatic software package such as BrainSuite [31] which provides interactive mathematical morphological operations can be used. Multi-scale watershed algorithm based on the generalized anisotropic nonlinear (GAN) diffusion scheme [32] may be also applied. We then

constructed a volume image pyramid of 4 levels using a reduction ratio of 2. The image resolutions at level 1, 2, 3 are $128 \times 128 \times 68$, $64 \times 64 \times 34$, and $32 \times 32 \times 17$. We set the deformation parameters to the following values: $w_{st} = 5.0$, $w_{bd} = 10.0$, $w_{edg} = 10.0$, $w_{bal} = 10.0$, $w_{nsi} = 10.0$. Local adaptive remeshing was executed every deformation step.

Fig. 14 shows top, side, and front views of the brain cortical surfaces extracted at each level of the image

pyramid. Fig. 15(a)–(d) also shows the models overlaid on the transaxial slice images at each level. At the coarsest level, shown in Figs. 14(a) and 15(a), the model found a coarse outline of the cortex very quickly after 50 iterations. At the next level after 350 iterations, the model progressed into interhemispheric fissure, so that left and right hemispheres were roughly identified (Fig. 14(b) and Fig. 15(b)). At level 1 and 0 after 450 and 550 iterations, many concavities including

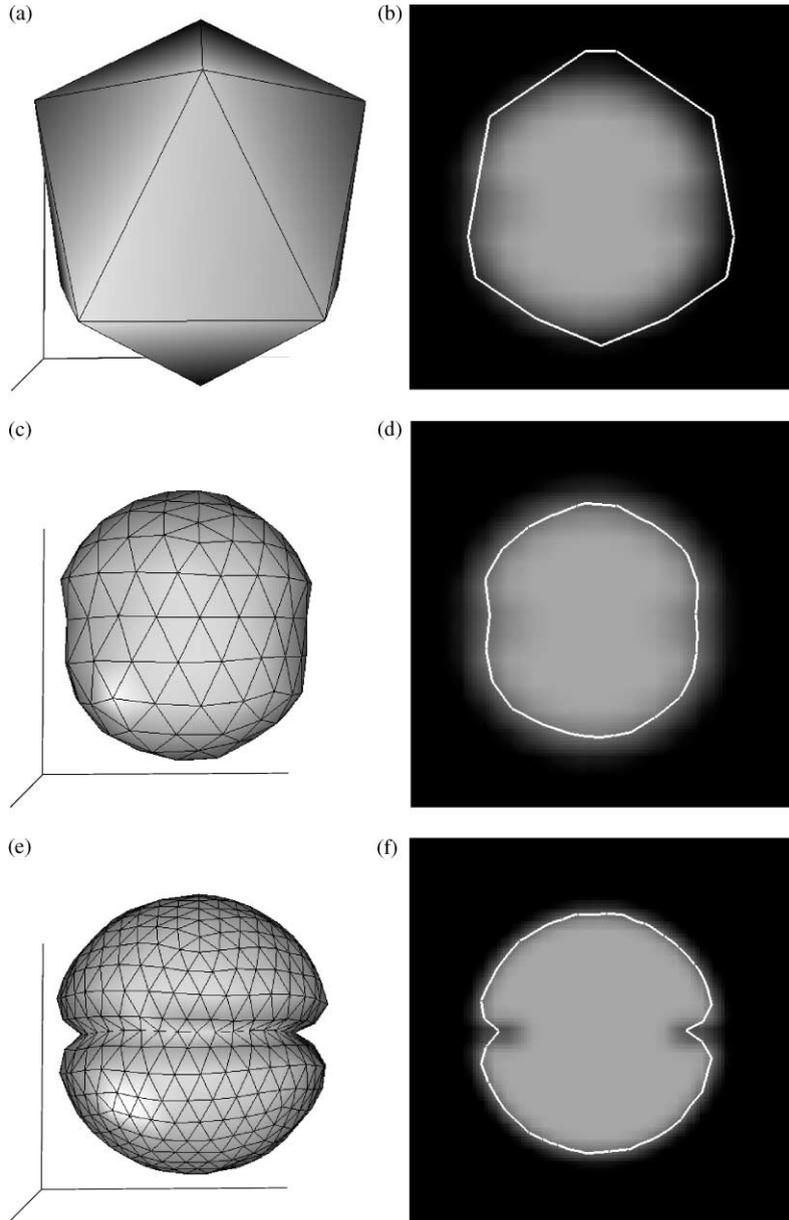


Fig. 13. Phantom sphere boundary surfaces with concavities extracted at each level of a computer-generated volume image pyramid. (a) Initial icosahedron. (c)(e)(g)(i) The deformable surface models converged after 20, 50, 140, 150 iterations at level 3, 2, 1, 0. (b)(d)(f)(h)(j) Initial icosahedron and the models overlaid on transaxial slice images.

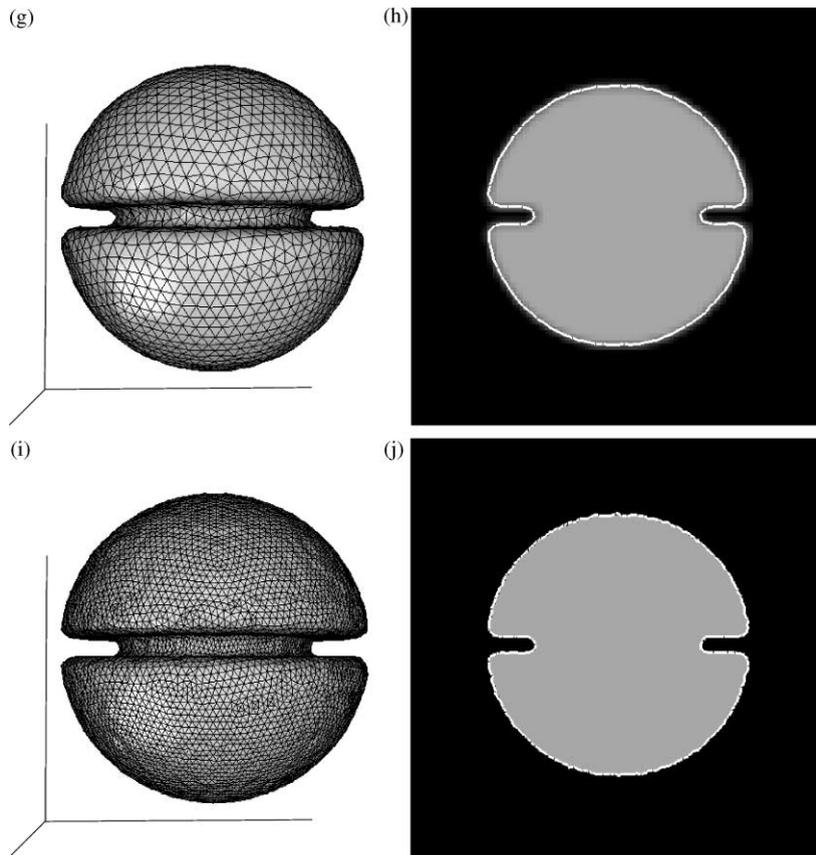


Fig. 13. (continued)

the lateral fissure, central sulcus, and other small sulci were gradually extracted in detail without any model self-intersections (Fig. 14(c), (d) and Fig. 15(c), (d)).

Table 1 shows the model components (the number of model nodes, the number of triangular meshes, and average edge length) observed at each level of the image pyramid. When the model is transferred from h level to $h - 1$ in the image pyramid, the number of model nodes and triangular meshes increases about four times. On the other side, the average edge length of the model decreases by half, that is, by the same degree as the unit voxel size of the image is reduced. Therefore, the model resolution matches the image resolution for every level of the pyramid.

In Table 2, we show the amount of CPU time (in ms) taken to extract brain cortical surface at each level of the image pyramid for each computational step: global remeshing, local remeshing, internal/external force computation, and non-self-intersection force computation. The total CPU time of about 21 min (1,279,640 ms) was spent mostly at level 0 (1,028,600 ms), since the computational time for fitting model is directly propor-

tional to the number of model nodes. The total time consumed for local adaptive remeshing is small relative to time for computing the forces, because it does not require recursive or repetitive computation. On the other hand, computing the non-self-intersection force is the most expensive step, despite the use of an octree structure for efficient computation. Additionally, it took about 10 min to detect edges using the 3D Monga–Deriche (MD) operator. However, the whole processing time is very reasonable compared to times reported in the literature [21,33]. Xu [33] reported a time of 4.5–6.4 h on an SGI O2 R10000 system to reconstruct the central layer of the cortex. MacDonald [21] reported a time of about 100 h on an SGI Origin 200 R10000 system to identify both the inner and outer surfaces of the cerebral cortical gray matter.

For comparison, we applied the model with the non-self-intersection force disabled ($w_{\text{nsi}} = 0.0$), resulting in numerous instances of model self-intersection (cf. Fig. 16(a) to (b)). We also applied the model without local adaptive remeshing and many concave regions successfully extracted by the locally remeshed model were smoothed over (cf. Fig. 16(c) to (d)).

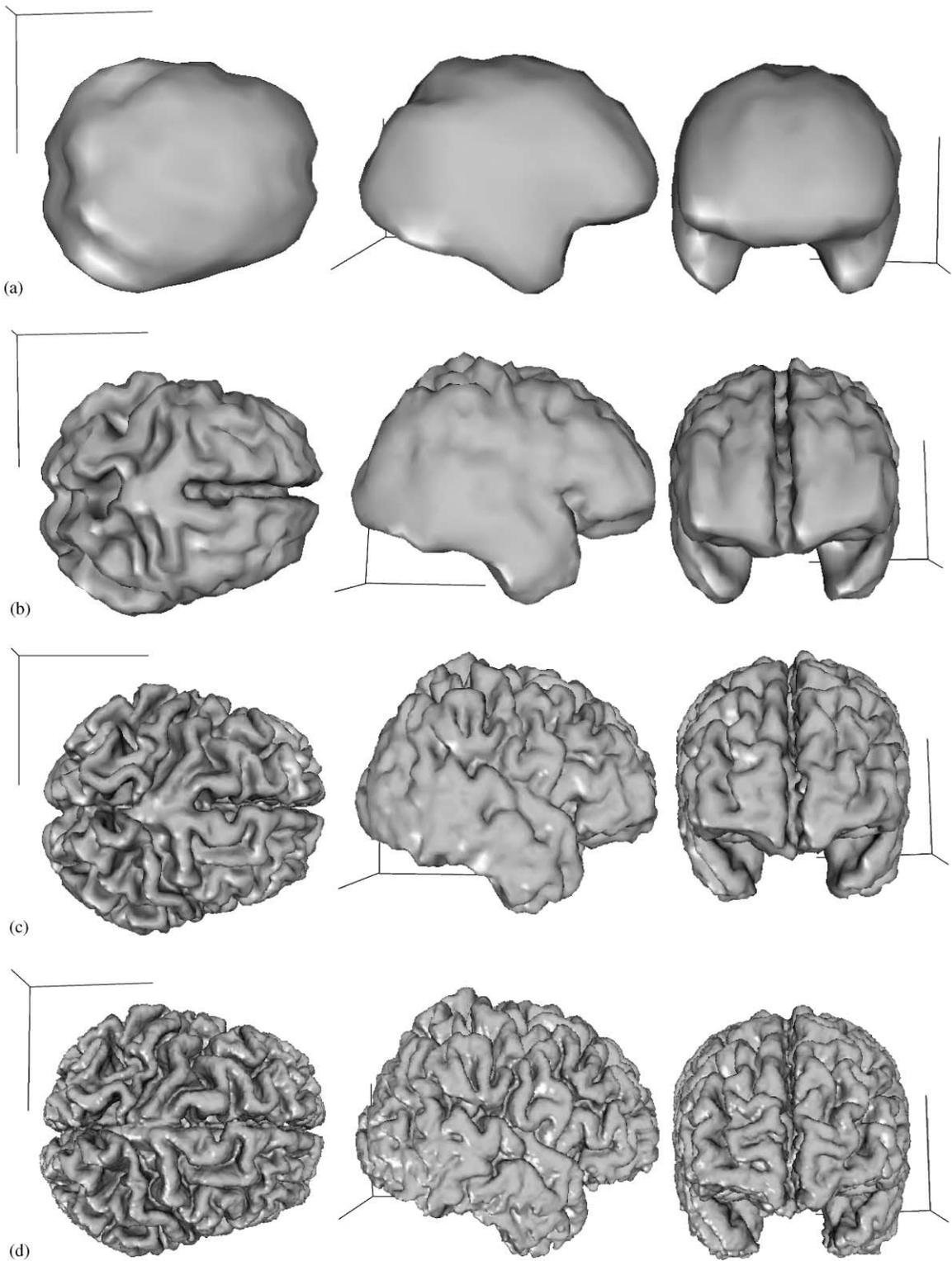


Fig. 14. Brain cortical surfaces extracted at each level of a MR volume image pyramid. (a) (b) (c) (d) Top, side, front views of the model converged after 50, 350, 450, 550 iterations at level 3, 2, 1, 0.

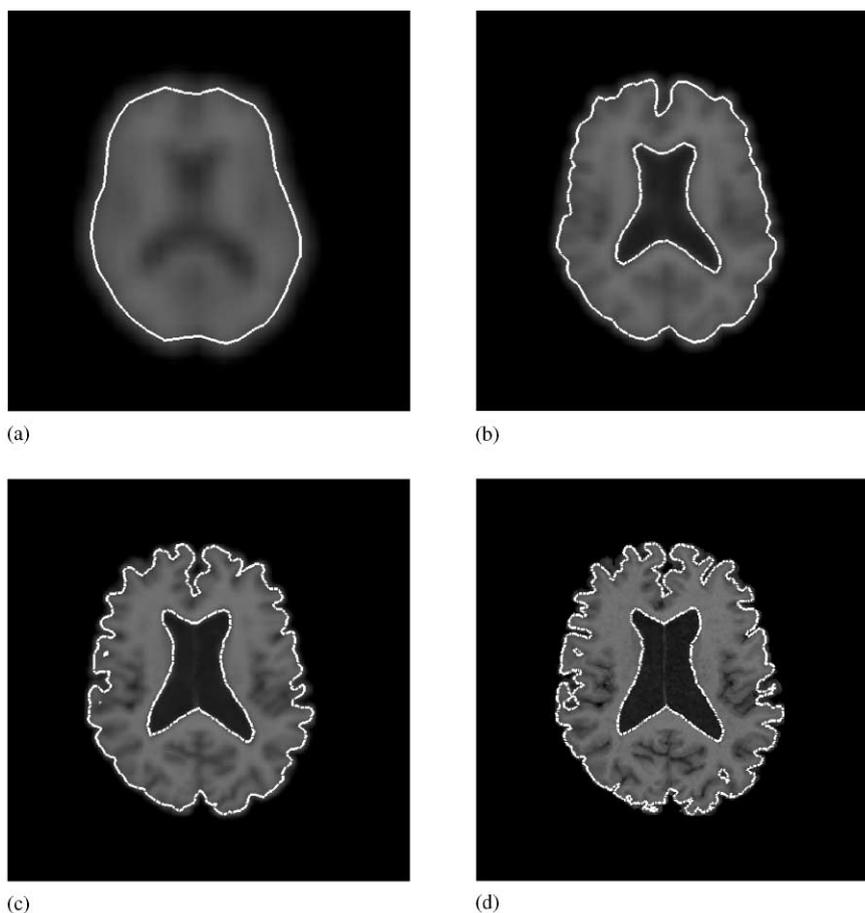


Fig. 15. Transaxial slice images overlaid by the brain cortical surfaces extracted at each level of the pyramid. (a) (b) (c) (d) Results after 50, 350, 450, 550 iterations at level 3, 2, 1, 0.

Table 1
Model components at each level of the image pyramid on brain cortical surface extraction

Pyramid level	3	2			1	0
iterations	50	150	250	350	450	550
No. of nodes	608	2966	3216	3390	18,291	97,615
No. of triangles	1212	5928	6428	6776	36,578	195,226
Average edge length	10.564	5.735	5.717	5.717	2.916	1.449

Table 2
Computational time(ms) for each processing step on brain cortical surface extraction

Pyr. level	3	2			1	0	Total
iterations	50	150	250	350	450	550	
Global rem.	20	40	—	—	260	1630	1950
Local rem.	480	3560	3830	4060	24,410	138,020	174,360
Int./ext. for.	590	6010	6830	7520	42,070	245,700	308,720
NSI. for.	1920	14,140	17,060	19,450	98,790	643,250	794,610
Total	3010	23,750	27,720	31,030	165,530	1,028,600	1,279,640

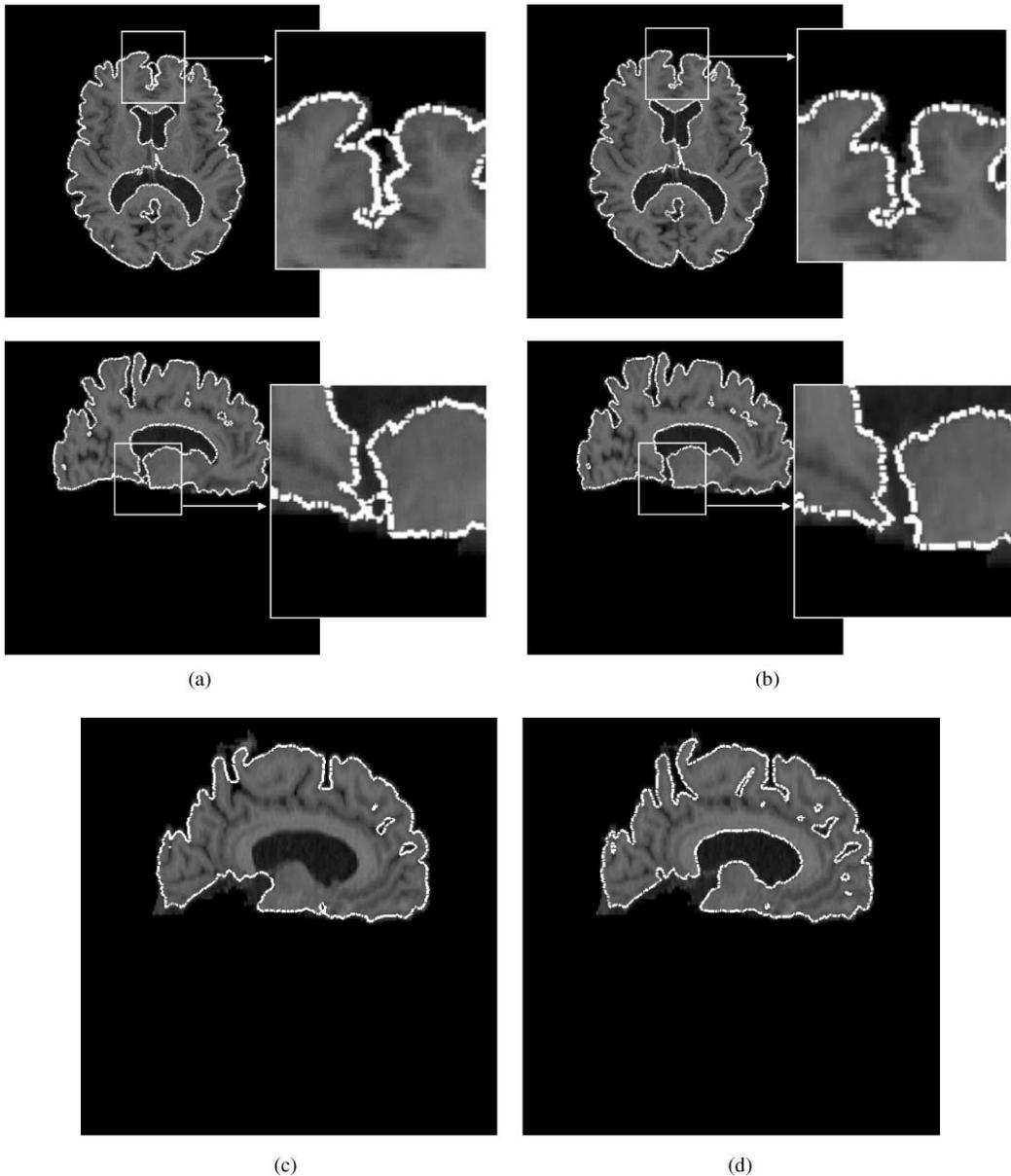


Fig. 16. Results obtained using the deformable surface model (a) without non-self-intersection force, (b) with non-self-intersection force, (c) without local adaptive remeshing and (d) with local adaptive remeshing.

8. Conclusion

We have developed a multiscale deformable surface model with an adaptive remeshing capability and a non-self-intersection capability for extracting the boundaries of complex-shaped anatomic structures from volumetric medical images. Through global and local adaptive remeshing, our model adapts its resolution to a pyramid of multiresolution volume images derived from the input

volume image in a coarse-to-fine fashion. It then deforms in accordance with a physics-based formulation including conventional internal and external forces, as well as a non-self-intersection force.

The multiscale approach allows a coarse model to be fitted quickly to low resolution data and efficiently reconstruct the rough overall shape of the anatomic structure, while a series of progressively finer models subsequently capture the detail in higher resolution data.

The approach not only greatly reduces the overall computation time for extracting object boundaries, but also provides a model that is relatively independent of initialization. Adaptive remeshing enables the model to progress into deep concavities successfully, and consequently improves the efficiency, compactness, and accuracy of the reconstructed model. As demonstrated by our experimental result, the deformable surface model without local adaptive remeshing failed to extract localized protrusive or concave surface features. Our local adaptive remeshing method, which includes melting, inversion and subdivision operations, can be performed efficiently in $O(n)$ time without an additional recursive conformation step. Furthermore, we not only overcame the self-intersection problem by using an effective non-self-intersection force, but also drastically reduced its computation time by exploiting an octree structure of triangles.

Although our current adaptive remeshing methods provide effective results and proper triangulation, it has the tendency to generate too many nodes and triangles relative to the complexity of local areas. For this reason, we are currently exploring the use of other metrics such as curvature instead of edge length for deciding which triangles should be remeshed. And an further improved adaptive remeshing is also under consideration in order that triangles from different refinement levels can be joined, preserving the mesh consistency with no degeneration of triangle shape, as in $\sqrt{3}$ -subdivision [34]. We currently use an octree structure to find self-intersection triangle candidates, but more efficient spatial decomposition or hierarchical bounding volume techniques can be adopted to reduce its time cost. Finally, we are considering further model improvements in order to extract the brain sulci more accurately. For example, we intend to explore the use of interactively placed point and curve constraints with our model that will force the model to correctly penetrate and segment the brain sulci in highly noisy regions.

Acknowledgements

This work was partially supported by the Korean Ministry of Information and Communication under the University S/W Research Center Program and partially supported by the Korean Ministry of Science and Technology under the National Research Laboratory Program.

References

- [1] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. *International Journal of Computer Vision* 1988;1(4):321–31.
- [2] Terzopoulos D, Witkin A, Kass M. Constraints on deformable models: recovering 3D shape and nonrigid motion. *Artificial Intelligence* 1988;36(1):91–123.
- [3] McInerney T, Terzopoulos D. Deformable models in medical image analysis: a survey. *Medical Image Analysis* 1996;1(2):91–108.
- [4] McInerney T, Terzopoulos D. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging* 1999;18(10):840–50.
- [5] Carlbom I, Terzopoulos D, Harris K. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Transactions on Medical Imaging* 1994;13(2):351–62.
- [6] Cohen LD, Cohen I. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1993;15(11):1131–47.
- [7] Durikovic R, Kaneda K, Yamashita H. Dynamic contour: a texture approach and contour operations. *The Visual Computer* 1995;11:277–89.
- [8] Terzopoulos D, Fleischer K. Deformable models. *The Visual Computer* 1988;4(6):306–31.
- [9] Staib LH, Duncan JS. Model-based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging* 1996;15(1):720–31.
- [10] Davatzikos CA, Prince JL. An active contour model for mapping the cortex. *IEEE Transactions on Medical Imaging* 1995;14(1):65–80.
- [11] McInerney T, Terzopoulos D. A dynamic finite element surface model for segmentation and tracking in multi-dimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics* 1995;19(1):69–83.
- [12] Miller JV, Breen DE, Lorensen WE, O'Bara RM, Wozny MJ. Geometrically deformed models: a method for extracting closed geometric models from volume data. *Proceedings of SIGGRAPH 1991 Annual Conference on Computer Graphics, Las Vegas, NV, vol. 25, July 1991*. pp. 217–26.
- [13] Lachaud J-O, Montanvert A. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis* 1999;3(1):1–21.
- [14] Whitaker R. Volumetric deformable models. In: Robb RA, editor, *Proceedings of Conference on Visualization in Biomedical Computing (VBC'94)*, SPIE Proceedings, vol. 2359, March 1994.
- [15] Malladi R, Sethian JA. A real-time algorithm for medical shape recovery, *Proceedings of International Conference on Computer Vision, January 1998*.
- [16] Caselles V, Kimmel R, Sapiro G. Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1997;19(4).
- [17] MacDonald D, Avis D, Evans A. Multiple surface identification and matching in magnetic resonance images. In: Robb RA, editor, *Proceedings of Conference on Visualization in Biomedical Computing (VBC'94)*, SPIE Proceedings, vol. 2359, 1994. p. 160–9.
- [18] Chen Y, Medioni G. Surface description of complex objects from multiple range images. *Proceedings of*

- Computer Vision and Pattern Recognition Conference (CVPR'94), 1994. p. 153–8.
- [19] Terzopoulos D, Vasilescu M. Adaptive surface reconstruction. Proceedings of Sensor Fusion III: 3-D Perception and Recognition, SPIE Proceedings, vol. 1383, 1990. p. 257–64.
- [20] Vasilescu M, Terzopoulos D. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. Proceedings of Computer Vision and Pattern Recognition Conference (CVPR'92), June 1992 p. 829–32.
- [21] MacDonald D, Avis D, Evans AC. Proximity constraints in deformable models for cortical surface identification. Proceedings of First International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'98), October 1998. p. 650–9.
- [22] Montagnat J, Delingette H. Volumetric medical image segmentation using shape constrained deformable models. Proceedings of Second International Conference on Computer Vision, Virtual Reality and Robotics in Medicine, 1997. p. 13–22.
- [23] Terzopoulos D. Multilevel computational processes for visual surface reconstruction. Computer Vision, Graphics, and Image Processing 1983;24:52–96.
- [24] Burt PJ. Fast filter transforms for image processing. Computer Graphics and Image Processing 1981;16:20–51.
- [25] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12(7):629–39.
- [26] Alvarez L, Lions P-L, Morel J-M. Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal of Numerical Analysis 1992;29(3):845–66.
- [27] Niessen WJ, Romeny BMH, Florack LMJ, Viergever MA. A general framework for geometry-driven evolution equations. International Journal of Computer Vision 1997;21(3):187–205.
- [28] Metaxas D, Kho E. Efficient shape representation using deformable models with locally adaptive finite elements. Proceedings of Geometric Methods in Computer Vision II, SPIE Proceedings, vol. 2031, 1993. p. 160–71.
- [29] Suzuki H, Sakurai Y, Kanai T, Kimura F. Interactive mesh dragging with adaptive remeshing technique. Proceedings of Pacific Graphics'98, 1998. p. 188–97.
- [30] Monga O, Deriche R. 3D edge detection using recursive filtering. Proceedings of IEEE Conference, Computer Vision and Pattern Recognition, June 1989.
- [31] Shattuck DW, Leahy RM. BrainSuite: An automated cortical surface identification tool. Proceedings of Third International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2000), October 2000.
- [32] Dam E, Nielsen M. Non-linear diffusion for interactive multi-scale watershed segmentation. Proceedings of Third International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2000), October 2000.
- [33] Xu C, Pham DL, Prince JL, Etemad ME, Yu DN. Reconstruction of the central layer of the human cerebral cortex from MR images. Proceedings of First International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'98), October 1998. p. 481–88.
- [34] Kobbelt L. $\sqrt{3}$ -subdivision. Proceedings of SIGGRAPH 2000 Annual Conference on Computer Graphics, New Orleans, LA, July 2000. p. 103–12.