Multi-featured Multi-scale Deformable Model for Three-dimensional Surface Extractio from Volumetric Image

Joo-Young Park, Tim McInerney*† , Demetri Terzopoulos*, Myoung-Hee Ki
Department of Computer Science and Engineering, Ewha Womans University, Seoul, Korea
* Department of Computer Science, University of Toronto, Toronto, Canada
† Dept. of Computer Science, Ryerson Polytechnic University, Toronto, Canada
sindy@mm.ewha.ac.kr, {tim, dt}@cs.toronto.edu, mhkim@mm.ewha.ac.kr

Abstract

Deformable surface models provide a very attractive method for segmenting the three-dimensional shape of complex and various internal organs from volumetric medical images. However, although researchers have succeeded in overcoming some of the limitations of this approach, there still exist several significant unsolved problems. In this paper, we propose a multi-featured multi-scale deformable surface model as a solution to three of these problems: model initialization dependency, inability to extract object concavities, and model self-intersection. The multi-scale approach, in which w progressively resample our deformable triangulated surface model globally and locally in order to match its resolution to the level of a 3-D image pyramid, provides insensitivity to local minima and model initialization as well as the ability to extract object concavities. The multi-featured energy function including internal, external, and geometric self-collision constraint force prevents self-intersection effectively. We have applied our model to the challenging problem of brain cortex boundary extraction and preliminary results are presented.

1. Introduction

The segmentation of volumetric objects from medical images is an essential first step for various subsequent applications such as visualization, quantitative analysis, motion tracking, and image-guided surgery. Over the past several years a variety of segmentation methods have been developed. However, traditional segmentation methods, such as pixel-based clustering, region growing, and edge detection, require additional preprocessing and postprocessing as well as considerable amounts of expert intervention. Furthermore, the subsequent analysis and interpretation of the segmented objects is hampered due to the pixel- o voxel-level structure representations resulting from these methods[13].

Deformable models offer a reasonable approach to solving the problems of segmentation, because the combine many desirable features such as inherent connectivity and smoothness that compensates for noise and boundary irregularities, and the ability to incorporate anatomic knowledge[2,14]. Deformable models in 3-D were first used in computer vision[21] and computer graphics[22]. Many researchers[2,3,7,8,12,14,16] have explored the use of deformable models for segmenting structures in medical image volumes[13].

However, most classifical deformable models have several common limitations. Firstly, deformable models are sensitive to their initial conditions - they wer designed as interactive models and therefore relied upon a user to push or pull them out of local minima. Consequently, in more automatic segmentation scenarios, the initial model must usually be placed close to the boundaries of the target object to guarantee good performance. Secondly, the internal energy constraints of deformable models can limit their geometric flexibilit and prevent them from representing boundary concavities. Thirdly, they can generate non-simple surface geometries due to self-intersections of the model triangles.

On the other hand, level set methods using implicit models [4,10,11] provide topological and geometrical flexibility to overcome these limitations. However, implicit models does not provide geometrical(area, volume, local curvature) and topological information directly and require high computational expense[8,13].

In this paper, we propose a multi-featured multi-scale deformable surface model that overcomes the limitation of classical deformable models in order to segment complex internal organs like the brain cortex. Our deformable model is represented as a closed triangulated surface and integrates multiple features of internal, external and geometric constraints into its energ formulation. The multi-featured energ including geometric constraints help the model prevent self-intersection effectively. We also use a multi-scale approach that adapts the model to the level of the image volume in a 3-D pyramid by global and local resampling. This approach makes the model more efficient, much less sensitive to spurious image features, independent of initial conditions and also extractable to concavities.

In section 2 and 3, we describe the multi-scale approach and the multi-featured energy function employed in our deformable model. In section 4, we present the preliminary results of applying our model to extract brain cortex boundaries. Section 5 concludes the paper with a brief discussion and ideas for further studies.

2. Multi-scale approach

The image potential force, which attracts deformable models towards the boundaries of a target object, is typically computed at the local image area around model vertices. Therefore, deformable models often become stuck on spurious image edge features which do not represent the global energy minimum or a good local energy minimum (i.e. the true boundaries of the object). To avoid these types of local minima, the initial deformable model must be close to the desired boundary. Several global energy minimization methods have been proposed to address this problem for 2- deformable contours. These methods include simulated annealing, genetic algorithms, and markov random fields. However, applying these global energy minimization methods to 3-D deformable surfaces is inefficient due to the very expensive computational load required for convergence.

We propose a multi-scale approach which provides a fast and reasonable solution to this problem. In the multi-scale approach, we use a pyramid structure of multi-resolution 3-D images and adapt the deformable surface to the multi-resolution images by global and local resampling. A 3-D image pyramid is constructed by a set of multi-level volumetric images from low-level high-resolution images to high-level low-resolution images. The deformable surface is initially constructed using the coarsest triangulated mesh, which is then adapted to the image of lowest resolution in a 3-D image pyramid. Once rough boundaries are extracted by the coarse triangulated mesh, we move down to the lower level of the pyramid and refine the triangulated mesh using global and local resampling in order to adjust it to the resolution of the image. At the lower level, more accurate results can be achieved. This procedure is repeated until the lowest level of pyramid is reached. Figure 1 illustrates a 3-D image pyramid and multi-scale deformable surface model.

This approach allows the deformable model to pass over spurious image features and to quickly find a rough boundary approximation at the initial stage of the deformation. The ability to pass over spurious image edge features makes the model relatively independent of its initial condition. In addition, the multi-scale approach also reduces the total computational time. We describe our 3-D image pyramid in 2.1 and our global and local resampling methods for model refinement in 2.2.

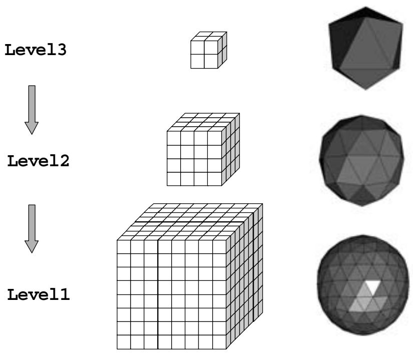


Figure 1. 3-D volumetric image pyramids and multi-scaled deformable surfaces

2.1 3-D image pyramid

In a classical Gaussian pyramid, the successive levels of the pyramid are computed by the convolution of a Gaussian kernel of size 5 voxels. It guarantees a low-cost filtering without a phase translation linked to a reduction factor of 2 for each image dimension[8]. However, the method doesn't consider that voxels can not be bound to be cubic and the reduction factor of pyramid image must be coherent with that of surface resampling. The new algorithm developed by Lachaud, J.[7,8] creates volumetric pyramids of any reduction factor in order to support isotropic convolution and consistency of image filtering/model resampling factor. We use this algorithm to construct our 3-D image pyramid. We now briefl describe the method.

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

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

$$U_{h+1} = \rho U_h \tag{2}$$

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

2.2 Global resampling

The resolution of triangulated deformable surface must match the resolution of the image. The edges of the triangles of model should neither be too long nor too small. If the edges are too long, high-frequency boundaries could be missed, if too small, the model would include redundant information by representing a voxel as several small edges. Therefore, we define the range of model edge lengths in proportion to the unit size of real space at each image pyramid level similar to Lachaud, J [8].

The edge length of mesh at a level h should be at leas larger than the unit voxel size of level *h-1* and smaller than the unit size of level *h+1*. That is, the edge length of mesh at a level h of pyramid must be in the range of $[U_{h-1}, U_{h+1}] = [U_h/\rho, \rho U_h]$. Global resampling controls global resolution of the model by keeping the average edge length of the model in that range. If the average edge length of the model at a pyramid level h is larger than the upper bound of that range, all triangles of the model are subdivided uniformly. The condition for performing global resampling is as follows :

$$L_h \geq \rho U_h \qquad (3)$$

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

Figure 2 illustrates how one of the triangular meshes can be resampled globally. Each edge is bisected to form three new nodes. After every global resampling, the number of the triangles is increased by four times, and the average edge length is reduced to a half of the old one. Thus, in our case, the reduction factor ρ of 3-D pyramid should be 2.

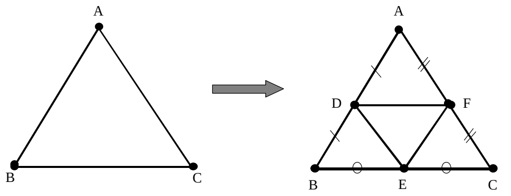


Figure 2. Subdivision of triangle for global resampling

2.3 Local resampling

When the model is deformed to extract the object of interest, some triangles can be expanded, while other triangles can be sh unk. Global resampling can control just the average edge length in a range and does not guarantee the edge lengths of all triangles will be kept in a range, because it subdivides all triangles uniforml depending on the average edge length of model. That is, global resampling cannot guarantee a regularly sampled model.

If the size of all triangles is not regular, there would be large triangles which could not expand themselves further due to the internal forces even though they still have to move further towards the object boundaries. Thus, these triangular areas fail to find the real boundaries. Moreover, very small triangles which may have already converged to the boundaries would be subdivided again in a global resampling stage. This causes not only redundant computational load, but also self-collision between adjacent small triangles. Local resampling is needed to prevent these problems and force the triangulated surface to remain regularly sampled during the deformation.

[1,8,15,16,20] adapted local resampling in their deformable model. [6]'s local resampling is a two stage recursive process composed of identifying triangles to be resampled and then subdividing. However, this recursive process requires considerable computational time and memory for a stack. The algorithm of [1,15] consists of two steps, a bisection operation and a conforming operation. The conforming operation ensures the desirable triangular mesh properties of confirmity, non-degeneracy and smoothness are maintained. However, this operation also should be recursively applied until the triangulation becomes conforming. Our subdivision algorithm ensures confirmity, non-degeneracy and smoothness without the use of an expensive recursive implementation. In addition, [12,15,16] conside resampling only for big triangles, not for small triangles. Our resampling algorithm considers both.

On the other hand, the methods of [8,20] includes bisection and inversion for too-long edges, melting fo too-short edges. [8] also supports axial and annular transformations for topological changes. In [20], they used the length of the three edges and the three angles of a triangle for finding a ill-conditioned triangle. Thei resampling algorithm also requires iterative procedure.

In [8], they presented three geometric constraints to determine the mesh sampling. They used the distance between non-neighboring vertices as a constraint to detect self-intersection. However, this constraint is not sufficient to detect self-intersection. To detect self-intersection full using only this constraint, a very small time interval must be used. However, a small time interval is inefficient and sensitive to noise. Thus, we check the minimum distance between two non-neighboring triangles in order to detect self-intersection in advance. This constraint is sufficient to prevent the problem. Also in [8], three geometric constraints were applied to maintain mesh uniformity, for self-collision detection, as well as to allow topological transformations. Topological transformations include breaking a connected mesh area into two areas and merging two mesh areas into one. This approach is very flexible and useful to extract objects with complex topology automatically. However, the method which merges two parts when self-collision is detected is not appropriate to extract concave areas like the sulci of brain. Therefore, our local resampling doesn't include topological transformations. Instead, when self-intersection is detected, we apply self-collision protection forces. We describe these forces in Section 3.

We check the edge lengths of each triangle for local resampling and perform local operations such as melting, inversion, or subdivision(bisection, trisection, or quadsection) for each triangle which is too big or too small. Local resampling can be done at every iteration or once per several iterations.

The following conditions are used for local resampling.

$$\left|\overline{v_i} - \overline{v_k}\right| < U_h / \rho \qquad (4)$$

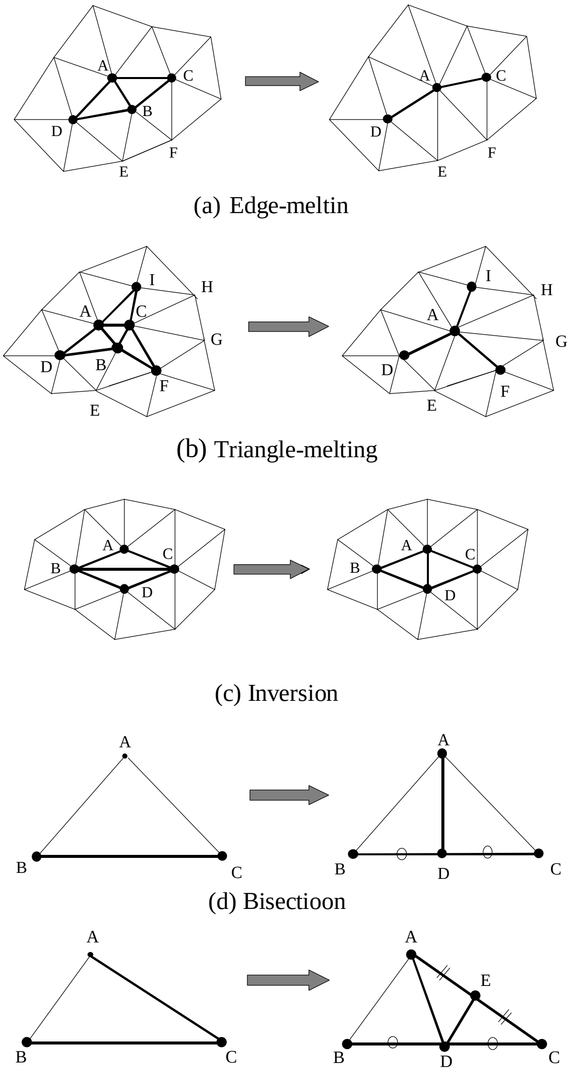
$$\left|\overline{v_i} - \overline{v_k}\right| > \rho U_h \qquad (5)$$

where N is the number of nodes of a model, v_i, v_k ($i,k=1,...,N$) are adjacent model nodes, $\overline{v_i} = (x_i, y_i, z_i)$ is the position of i th node.

(4) and (5) express the violation of the lower and upper bounds of one edge length. For triangles with edges to satisfy condition (4), a triangle-melting or edge-melting operation is applied. For triangles with edges satisfying condition (5), an inversion or subdivision operation is performed. Our local resampling process consists of three steps: First, we trace all triangles of the model in order to find triangles having edges that need melting, and melt the triangle or the edge. Second, we search all triangles to find edges requiring inversion. Third, we trace all triangles to find triangles that need to be subdivided into two, three, or four subtriangles, and subdivide them. These three steps are performed just once and don't require recursive or iterative processes. Three traces of all triangles are sufficient. Figure 3 describes the operations that may be applied at each step for local resampling.

Melting : We check the condition (4) for all edges of model. As Figure 3(a), when the length of only one e dge $\overline{AB}$ of ΔABC is shorter than U_h / ρ and the length of other edges $\overline{BC}$, $\overline{CA}$ is equivalent to or longer than U_h / ρ, the edge $\overline{AB}$ is melted as shown Figur 3(a). When more than two edges $\overline{AB}, \overline{BC}$ of ΔABC are shorter than U_h / ρ , the triangle ΔABC is melted a shown Figur 3(b).

Inversion : When a pair of thin and long triangles sharing a long edge to satisfy the condition (5) is found, the edge is inverted. As Figure 3(c), if $BC > \rho U_h$, but all other edges($\overline{AB}$, $\overline{CA}$, $\overline{BD}$, $\overline{DC}$) of two neighboring triangles of $\overline{BC}$ are equivalent to or shorter than ρU_h and $\overline{AD}$ is also equivalent to or shorter than ρU_h , we invert the edge $\overline{BC}$ into $\overline{AD}$.



(a) Edge-meltin

(b) Triangle-melting

(c) Inversion

(d) Bisectioon

(e) Trisection

Figure 3. Operations for local resampling

Subdivision : When we check the lengths of three edges of every triangle, if just one edge satisfys the condition (5), we bisect the triangle based on the long edge. If two edges satisfy the condition (5), we trisect the triangle based on the two long edges. Figure 3(d)(e) shows bisection and trisection. And if all of three edges satisfy the condition, we quadsect the triangle as th global resampling of figure 2. In figure 3(d), for $\overline{BC} > \rho U_h$, $\overline{AB}, \overline{CA} \leq \rho U_h$, we link the midpoint D of $\overline{BC}$

to the node A and subdivide ΔABC into ΔABD , ΔADC . In (e), for $\overline{BC} > \overline{CA} > \rho U_h$ and $\overline{AB} \le \rho U_h$, we link the midpoint D of longest edge $\overline{BC}$ to the node A and then connect the midpoint E of $\overline{CA}$ to D. This subdivision process is performed for each triangle, but the result is the bisection of all edges longer than $\rho \mathrm{U}_h$. Thus this process satisfies conformity without an repeatation.

3. Multi-featured ener function

Our deformable model is composed of a discrete set of nodes. Each node in the model has its own energy function that governs its deformation. The energ function of traditional deformable model consists of an internal force and external force term. The internal force provides elastic and flexible connectedness between model elements. The external force is the force computed from image data in order to move the model towards edges or contour boundaries. However, there is no constraint force to prevent non-simple structures from forming due to self-intersection.

In this paper, we present a multi-featured energy function, which adds to a conventional energ function a geometric self-collision protection force to prevent non-simple structures. The resulting energy function i composed of two internal force terms, two external force terms, and one geometric force terms as follows :

$$E(\overline{v_i}) = E_{int}(\overline{v_i}) + E_{ext}(\overline{v_i}) + E_{geo}(\overline{v_i}) \qquad (6)$$

$$E_{int}(\overline{v_i}) = w_{st}F_{st}(\overline{v_i}) + w_{bd}F_{bd}(\overline{v_i}) \qquad (7)$$

$$E_{ext}(\overline{v_i}) = w_{inf}F_{inf}(\overline{v_i}) + w_{img}F_{img}(\overline{v_i}) \qquad (8)$$

$$E_{geo}(\overline{v_i}) = w_{cp}F_{cp}(\overline{v_i}) \qquad (9)$$

where, $\overline{v_i(t)} = [x_i(t), y_i(t), z_i(t)]$ is the position of node v_i at time t, $E(\overline{v_i})$ is the energy of node i at time t , $E_{int}(\overline{v_i})$, $E_{ext}(\overline{v_i})$, $E_{geo}(\overline{v_i})$ are internal, external, geometric energies, $F_{st}(\overline{v_i})$ and $F_{bd}(\overline{v_i})$ are internal stretching and bending forces, $F_{inf}(\overline{v_i}), F_{img}(\overline{v_i})$, and $F_{cp}(\overline{v_i})$ is a geometric collision protection force. And $w_{st}, w_{bg}, w_{inf}, w_{img}, w_{cp}$ are the individual weighting coefficients that allow the magnitudes of the various parameters to be scaled.

The dynamics of each node v_i according to Newton's law is given by a simplified form of the discrete Lagrange's equations of motion :

$$\gamma \dot{v}_i = F_{\mathrm{int}}(v_i) + F_{ext}(v_i) + F_{geo}(v_i) \qquad (10)$$

where γ is a damping factor and $\dot{v}_i$ denotes the velocit of node v_i. $F_{\mathrm{int}}, F_{ext}, F_{geo}$ is the internal, external, geometric forces. We integrate the dynamic system (10) forward through time using an explicit Euler method. The energy of the model is minimal when it reaches a stable position.

3.1 Internal force

The internal energy force consists of a stretching force and a bending force. The stretching force acts as if all nodes is linked with adjacent nodes by an elastic spring. Thus, the stretching force increases as lengths between nodes are stretched or compressed relative to a rest length d_{rest} .

$$F_{st}(\overline{v_i}) = \sum_{j=1}^{m_i} \left(\left| \overline{v_{ij}} - \overline{v_i} \right| - d_{rest} \right) \frac{(\overline{v_{ij}} - \overline{v_i})}{\left| \overline{v_{ij}} - \overline{v_i} \right|} \qquad (11)$$

where $\overline{v_{ij}}$ is the position of j th adjacent node of v_i , $j = 1, ..., m_i$. m_i is the number of adjacent nodes of node v_i .

The bending force is essentiall curvature regularization and forces model to be smooth based on an estimate of local curvature. The discrete approximation of the curvature can be computed in various ways[8,16,23]. We approximate the curvature by computing the distance vector between a node position and the baricenter of neighboring nodes of the node like [8].

$$F_{bd}(\overline{v_i}) = \overline{c(v_i)} - \overline{v_i} - \frac{1}{m_i} \sum_{j=1}^{m_i} \left(\overline{c(v_{ij})} - \overline{v_{ij}} \right) \qquad (12)$$

where, $\overline{c(v_i)}$ is the baricenter for adjacent nodes of node v_i .

3.2 External force

We set the external force as a summation of inflation/deflation, and image forces.

Assuming that the image data can be separated into object and non-object, inflation/deflation force is used to inflate or deflate the model locally as long as it does not lie on the desired intensity range.

$$F_{inf}(\overline{v_i}) = B(I(\overline{v_i}))\overline{n_i} \qquad (13)$$

where, n_i is the unit normal vector of node v_i and computed by normalizing sum of normal vectors of adjacent triangles. $B(I(\overline{v_i}))$ is the binary thresholding operator for the image intensity $I(\overline{v_i})$.

The image edge force attracts the model toward significant 3D intensity edges in some region of an image function $I(x, y, z)$. The effect of this force is dependant upon the performance of the edge detection operator. We apply the 3D Monga-Deriche(MD) operator[17] to produce a 3D intensity edge field.

$$F_{img}(\bar{v}_i) = \nabla\left|O\_MD * I(\bar{v}_i)\right| \qquad (14)$$

3.3 Geometric force

Deformable surfaces typically use internal stretching and bending forces for regularization but they do not prevent non-simple(self-intersecting) surface geometries. A self-collision protection force is a geometric force used to maintain the simple geometric structure of the model i not destroyed.

We compute the smallest distance between two non-neighboring triangles to find self-colliding candidates. We then apply a self-collision protection force to th candidate triangles :

$$F_{cp}(t_i) = (d_{closest}(t_i, t_j) - D_{min})^2,$$
$$if\ d_{closest}(t_i, t_j) < D_{min} \qquad (15)$$
$$F_{cp}(t_i) = 0, \qquad otherwise$$

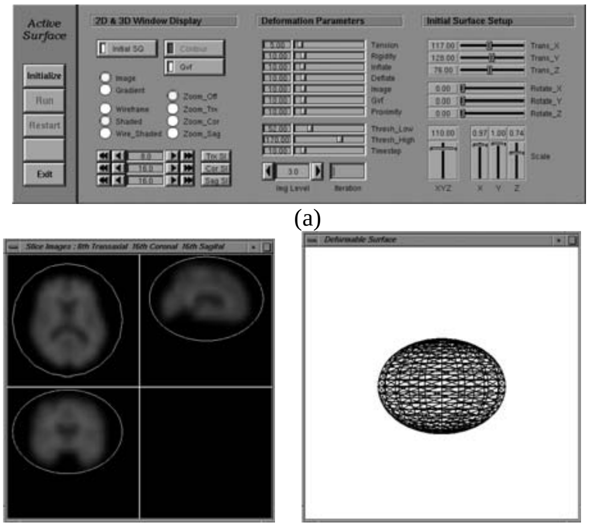
where $d_{closest}(t_i, t_j)$ is the Euclidean distance between the closest two vertices in the i' th and j' th triangle and D_{min} is the minimum distance allowed for non-neighboring triangles. The closest two vertices of i'th and j'th triangle is approximated by determining two vertices with the minimum distance among three vertices to three lines, three lines to three lines, three vertices to a triangle, and three lines to a triangle. Afte the $F_{cp}(t)$ is computed for every triangle t, we get $F_{cp}(\bar{v})$ by averaging self-collision protection forces of adjacent triangles $F_{cp}(t)$.

This force causes them to push each other in the opposite direction. D_{min} should be at least larger than U_h. This force is very similar the proximity constraint proposed by D.MacDonald and et. al.[9]. However, while they implemented the process in a brute-force way, we improved their approach to perform the process efficiently by using an octree structure of triangles.

4. Implementation and experimental results

We have implemented an interactive system on a Silicon Graphics Octane/MXE SE R10000 processor and use it to segment complex objects from 3-D volumetri medical images. This section describes the design of this experimental system and presents a preliminary result using this system for the extraction of the brain cortex boundary.

As shown in figure 4 (a), the system provides an interface that allows the user to control the parameters fo initialization and deformation of the deformable surface model interactively. The 2-D window of figure 4 (b) displays the cross-sectional images of the 3-D volumetric data in three orthogonal views and overlays the cross-sectional contours of the model embedded in the data. The 3-D window of figure 4 (c) displays the deformable surface in wireframe, shading, or both. Currently the user uses a superquadric to initialize the deformable surface model ((b)(c) of figure 4). The superquadric can be translated, scaled, and rotated to determine an initial position and size of the model. The superquadric is then converted into an icosahedron (the initial model) which is then further resampled globally to adjust the resolution of the model to the resolution of images.



(a)

(b) (c)

Figure 4. User interface for deformable surface model : (a) the interface fo controlling initialization and deformation parameters interactively (b) the window for displaying 2-D image slices (c) the window for displaying 3-D deformable surface model : Currently on (b), the lowest resolution image of the 3-D pyramid of 4 levels is shown. On (b)(c), superquadric, which the user use to determine an initial position and sizes of model, is placed.

We tested our model on a volumetric image of brain MR of discrete sizes 256× 256× 136. The brain image has been preprocessed to remove the skull and the brain stem. Since the reduction ratio is 2, the data size in level 1,2,3 is 128 × 128× 64, 64× 64× 34, and 32× 32× 17. The initial superquadric of the model was placed on the most blurred data of level 3 as shown in figure 4(b). And we set the deformation parameters to the following values : $w_{st}, w_{bg}, w_{inf}, w_{img}, w_{cp}$=5.0, 10.0, 10.0, 10.0, 10.0.

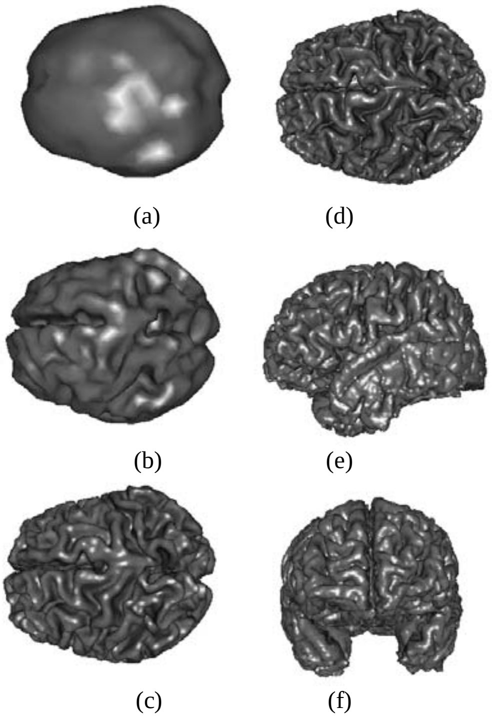


Figure 5. Brain cortex boundary extracted at each level of a 3-D image pyramid by multi-featured multi-scale deformable surface model : (a) 100 iterations at level 3(601 nodes, 1198 triangles), (b) 400 iterations at level 2(3333 nodes, 6662 triangles), (c) 500 iterations at level 1(19260 nodes, 38516 triangles), (d)(e)(f) top, front, and side view after 600 iterations at level 0 (103081 nodes, 206158 triangles)

Figure 5 shows the brain cortex boundary extracted at each level of a 3-D image pyramid. At the coarsest resolution of figure 5 (a), the model found a rough shape of the cortex very quickly. At the finer level, the mor detailed concave structures such a sulci were extracted gradually. Finally, we could get the fine and good result at the level 0. The final surface model in figure 5 (d)(e)(f) contains 103,081 nodes, 206,158 triangles. Figure 6 also shows the final result overlaid on the 2-D transaxial, sagital, coronal images. A lot of concave area such a sulci were extracted very deeply without any self-collision. On the contrary, when we applied the model without self-collision protection force as figure 7(a)(b), lots of colliding and penetrating were generated between close non-neighbor triangles. In addition, when we applied the model without local resampling as figure 7(c), the model could not extract many local concave area compared to figure 6(b).

The total fitting process took CPU time of about 15 minutes (859,780 ms) : 2,150 ms for 100 iterations at level 3, 32,620 ms for 300 iterations at level 2, 102,570 ms for 100 iterations at level 1, 722,440 ms for 100 iterations at level 0. The total fitting time is a sum of 1,500 ms for global resampling, 22,290 ms for local resampling, 308,680 for computing internal/external forces, 527,410 for protecting self-collision. The time consumed for local resampling is relatively very shor compared to other force computing times, because it doesn't require recursive o repeatative computation. On the other hand, the time cost for self-collision protection is the most expensive, although we use an octree structure for effective computation. Additionally, we took about 10 minutes for detecting edges by 3D Monga-Deriche(MD) operator. However, the whole processing time of about an hour is very reasonable compared to other researches[9,23]. D. MacDonald[6] reported a time of about 100 hours on a SGI Origin 200 R10000 processo for the segmentation of cortical surface and C.Xu[23] reported a time of 4.5 - 6.4 hours on a SGI O2 R10000 processor for the reconstruction of the central layer of th cortex.

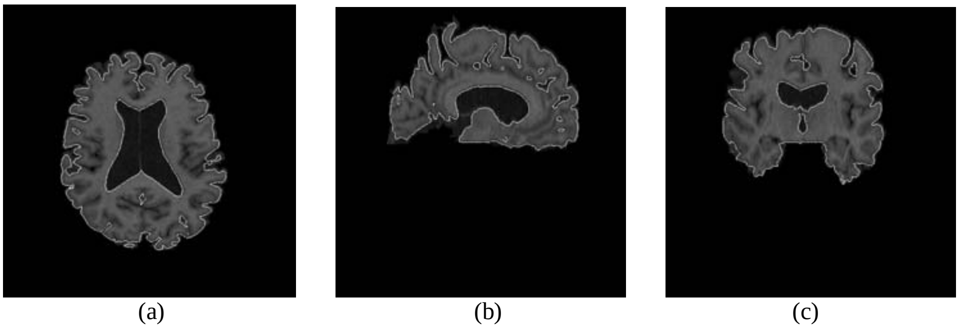


Figure 6. Triangulated surface overlaid on the 2-D cross-sectional images after total 600 iterations on a 3-D image pyramid : (a) transaxial (b) sagital (c) coronal view

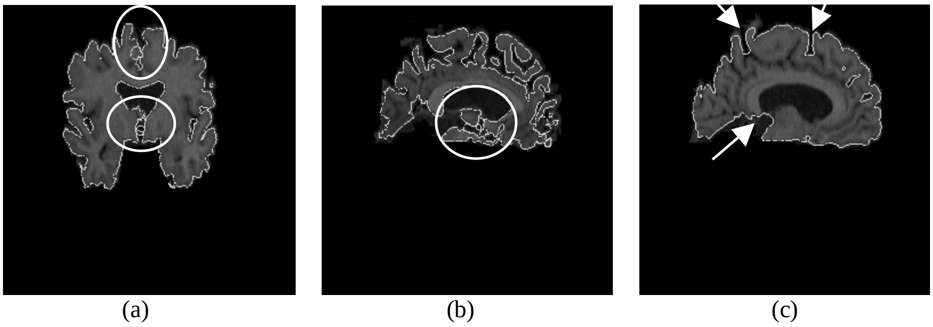


Figure 7. Results extracted by th deformable surface model without (a)(b) self-collision protection force and (c) local resampling : (a)(b) annotated circles include self-colliding triangles (c) arrows indicate the concave area failed to extract compared with (b)

5. Conclusion

We developed a multi-featured multi-scale deformable surface model for the segmentation of complex-shaped objects from volumetric medical images. Our model is resampled globally and locally on a pyramid structure of 3-D multi-resolution images and deformed by the multi-featured energy forces including traditional internal, external forces and a geometric constraint force.

The global resampling allows a coarse model to be fitted initially to the low resolution data and efficientl reconstruct the rough overall object shape, while a finer model can then be used to capture the detail from the higher resolution data. The approach greatly reduces the whole computation time for extracting object models as well as providing a model that is independent of initialization. The local resampling of the model improves the efficiency, compactness, and accuracy of the object reconstruction. As demonstrated by the experimental result, the multi-scale model without local resampling failed to extract local protrusive or concave area due to the magnitude of the internal forces relative to the external forces in these areas. Our local resampling method including melting, inversion and subdivision, can be performed efficiently in O(n) time without an additional recursive conforming operation. On the other hand, we not only overcame the self-intersection problem by using an effective self-collision protection force, but also drastically reduced time needed to compute this time-consuming force by using an octree structure of triangles.

Although current resampling methods provide effective result and proper triangulation, it has the tendency to generate too many nodes and triangles without relation to the complexity of the local area. Fo this reason, we are currently exploring the use of other metrics for deciding which triangles should to be resampled, such as curvature. We currently use an octree structure to find self-colliding triangle candidates, but we have under consideration more efficient spatial decomposition techniques like k-d tree[18], BSP-tree[19] or hierarchical bounding volumes lik OBBTrees[5], k-DOPs[6]. 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 ou model that will force the model to correctly penetrate and segment the brain sulci in highly noisy regions.

References

[1] Chen, Y. and Medioni, G., "Surface Description of Complex Objects from Multiple Range Images", *Proc. of CVPR'94*, pp.153-158, 1994

[2] Cohen, L.D. and Cohen, I., "Finite element methods for active contour models and balloons for 2D and 3D images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.15, no 11, pp.1131-1147, 1993

[3] Davatzikos, C.A. and Prince, J.L., "An Active Contour Model for Mapping the Cortex", *IEEE Trans. on Medical Imaging,* vol.14, no 1, pp.65-80, 1995

[4] Faugeras, O. and Keriven R., "Variational principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem", *INRIA Technical Report,* no.3021, 1996

[5] Gottschalk, S., Lin M.C., and Manocha, D., "OBBTree : A Hierarchical Structure for Rapid Interference Detection", *Proc. of ACM SIGGRPH'96*, 1996

[6] Klosowski, J.T., Held, M., et al., "Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs", *IEEE Trans. on Visualization and Computer Graphics* , vol.4, no.1, pp.21-36, Jan-Mar. 1998

[7] Lachaud, J.-O. and Montanvert, A., "Volumetric Segmentation using Hierarchical Representation and Triangulated Surface", *Research Report 95-37, Laboratoire de l'Informatique du Parallelisme,* ENS Lyon, France, Nov. 1995

[8] Lachaud, J.-O. and Montanvert, A., " Deformable meshes

with automated topology changes for coarse-to-fine three-dimensional surface extraction", *Medical Image Analysis*, vol.3, no.1, pp.1-21, 1999

[9] MacDonald, D., Avis, D., and Evans, A.C., "Proximit Constraints in Deformable Models for Cortical Surface Identification", *Medical Image Computing and Computer-Assisted Intervention(MICCAI) '98*, pp. 650-659, Oct. 1998

[10] Malladi, R., Sethian J.A., and Vemuri, B.C., “Shape Modeling with Front Propagation : A Level Set Approach”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.17, no.2, Feb.1995

[11] Malladi, R., and Sethian J.A., “A Real-Time Algorithm for Mecical Shape Recovery”, *Proc. of International Conference on Computer Vision*, Jan. 1998

[12] McInerney, T. and Terzopoulos, D., ``A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis," *Journal of Computerized Medical Imaging and Graphics*, vol.19, no.1, pp.69-83, Jan.,1994, Special Issue on Cardiopulmonary Imaging.

[13] McInerney, T. and Terzopoulos, D., "Deformable Models in Medical Image Analysis : A Survey", *Medical Image Analysis*, vol.1,no.2, pp.91-108, 1996

[14] McInerney, T. and Terzopoulos, D., "Medical Image Segmentation Using Topologically Adaptable Surfaces", *Proc. CVRMed'97*, Mar. 1997

[15] Metaxas, D., Kho, E., "Efficient shape representation using deformable models with locally adaptive finite elements", *Proc. of SPIE, Geometric Methods in Computer Vision II*, vol.2031, pp.160-171, 1993

[16] Miller, J.V., Breen, D.E., Lorensen, W.E., O'Bara, R.M. and Wozny, M.J., "Geometrically Deformed Models : A Method for Extracting Closed Geometric Models from Volume Data", *Computer Graphics(Proc. SIGGRAPH'91 Conf., Las Vegas, NY)*, vol.25, no.4, pp.217-226, July, 1991

[17] Monga, O. and Deriche, R., "3D Edge Detection Using Recursive Filtering", *Proc. IEE Conf. Comp. Vision and Pattern Recognition*, June 1989

[18] Naylor, B., Amanatides, J. and Thibault, W., “Merging bsp trees yield polyhedral modeling results”, *Proc. of ACM SIGGRAPH*, pp.115-124, 1990

[19] Samet, H., *Spatial Data Structures : Quadtree, Octrees and Other Hierarchical Methods*, Addison Wesley, 1989

[20] Suzuki, H., Sakurai, Y.,Kanai T., Kimura, F., “Interactive Mesh Dragging with Adaptive Remeshing Technique”, *Pacific Graphics’98*, pp.188-197, 1998

[21] Terzopoulos, D., "Constraints on deformable models : Recovering 3D shape and nonrigid motion", *Artificial Intelligence*, vol.36, no.1, pp.91-123, 1988

[22] Terzopoulos, D. and Fleischer, K., "Deformable models", *The Visual Computer*, vol.4, no.6, pp.306-331, 1988