Physics-Based Shape Deformations for Medical Image Analysis

Ghassan Hamarneh[a,b] and Tim McInerney[c,b]

[a] Mouse Imaging Centre, Hospital for Sick Children, Toronto, M5G 1X8, Canada
[b] Dept. of Computer Science, University of Toronto, Toronto M5S 3H5, Canada
[c] School of Computer Science, Ryerson University, Toronto M5B 2K3, Canada

ABSTRACT

Powerful, flexible shape models of anatomical structures are required for robust, automatic analysis of medical images. In this paper we investigate a physics-based shape representation and deformation method in an effort to meet these requirements. Using a medial-based spring-mass mesh model, shape deformations are produced via the application of external forces or internal spring actuation. The range of deformations includes bulging, stretching, bending, and tapering at different locations, scales, and with varying amplitudes. Springs are actuated either by applying deformation operators or by activating statistical modes of variation obtained via a hierarchical regional principal component analysis. We demonstrate results on both synthetic data and on a spring-mass model of the corpus callosum, obtained from 2D mid-sagittal brain Magnetic Resonance (MR) Images.

1. INTRODUCTION

Controlling non-rigid object deformation at multiple locations and scales in an interactive and intuitive manner is highly desirable in medical image analysis tasks such as segmentation and registration. Most current deformable shape models[1], are boundary-based and although provide excellent local shape control, lack the ability to undergo intuitive global deformation. As a result, it is difficult to incorporate intelligent deformation control operating at the right level of abstraction into the typical deformable model framework of energy minimization. Consequently, these models remain sensitive to initial conditions and spurious image features in image interpretation tasks.

Various hierarchical versions of boundary-based deformable models have been developed[2,3,4,5] but again fail to provide a natural global description of an object - the multi-scale deformation control is constructed upon arbitrary boundary point sets and not upon object-relative geometry. Several global or "volume-based" shape representation or deformation mechanisms do exist[6,7,8,9,10] but are limited either by the type of shapes they can represent, or the type and intuitiveness of the deformations they can carry out. They are also typically not defined in terms of the object but rather the object is unnaturally defined (or deformed) in terms of the representation or deformation mechanism.

Emerging trends in deformable shape modeling include medial-based approaches, which we believe are powerful techniques since they follow the geometry of the object and provide natural and intuitive deformations[11,12]. Additionally, physics based deformable shape models have been developed[10,13]. The attractiveness of these models stems from their ability to inherently handle smoothness and continuity constraints. Furthermore, statistically derived shape models[14,15] are gaining wide acceptance within the medical image analysis community since they constrain the global shape deformations according to the statistical shape variations observed in a training set.

The shape representation and deformation method presented in this paper is motivated by the following desirable characteristics of a deformable model for medical image analysis tasks. *First*, implementing the deformations within a physics-based framework that inherently handles smoothness and continuity constraints and facilitates intuitive user interaction. *Second*, using shape representations and deformations that follow the naturally geometry of the object. *Third*, controlling the deformations of an object shape at multiple locations and multiple scales. *Fourth*, restricting the deformations to produce only feasible shapes.

In this paper, we investigate a method that addresses all of the above points. *First*, the deformable shapes are modeled using physics-based meshes of connected nodes (mass-spring models) that maintain the structural integrity of the body as it deforms and are suitable for intuitive user interaction. *Second*, the mesh nodes and connectivity are based on the medial

Image Processing: Algorithms and Systems II, Edward R. Dougherty, Jaakko T. Astola,
Karen O. Egiazarian, Editors, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol.
5014 (2003) © 2003 SPIE-IS&T · 0277-786X/03/$15.00

axis of the object. *Third*, we use either operator- or statistics-based deformations to control the different types of deformation at multiple locations and scales. *Finally*, statistics-based feasible deformations are derived from a hierarchical (multi-scale) regional (multi-location) principal component analysis.

2. THE DYNAMIC MESH MODEL

We use mesh models to represent object shapes (Figure 1, Figure 2). A mesh is made up of nodes (masses or particles) and springs (elastic links or connecting segments). A Mass m_i, position x_i, velocity v_i, and acceleration a_i are associated with each node n_i. Two terminal nodes n_i and n_j, Hook's spring constant k_s, damping constant k_d, and rest length r_{ij} are associated with each spring s_{ij}

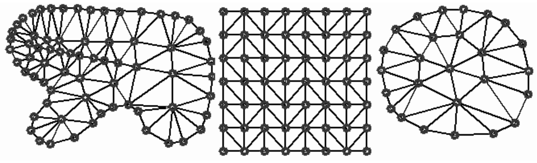


Figure 1. Examples of different synthetic spring-mass structures.

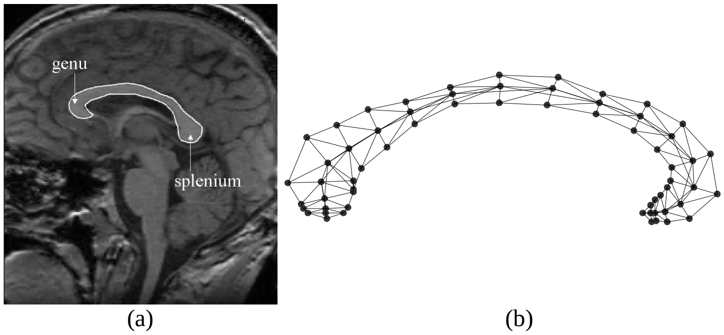


Figure 2. (a) Mid-sagittal MR brain image with the corpus callosum (CC) outlined in white. (b) CC mesh model.

Applying Newton's second law of motion and simulating the dynamics by time integration causes the mesh nodes to move and the object's shape to deform. Newton's second law of motion for the node n_i, states that $a_i = f_i / m_i$, where f_i is the total force acting on n_i

$$f_i = f_i^{Hook} + f_i^{viscous} + f_i^{user} + f_i^{image} \tag{1}$$

A spring s_{ij} will cause

$$f_i^{Hook} = -k_s \left(\left\| x_i - x_j \right\| - r_{ij} \right) \frac{x_i - x_j}{\left\| x_i - x_j \right\|} - \left(k_d \left(v_i - v_j \right)^T \frac{x_i - x_j}{\left\| x_i - x_j \right\|} \right) \frac{x_i - x_j}{\left\| x_i - x_j \right\|} \tag{2}$$

to be exerted at n_i and $-f_i^{Hook}$ on n_j. Viscous drag at n_i is given by

$$f_i^{viscous} = -k_v v_i \,. \tag{3}$$

A single user applied force f_i^{user} is implemented as the dynamic force resulting from a spring connecting a mesh node to the (varying) position of the user's point of application. Image forces can be implemented as

$$f_i^{image} \propto k_{ext} \nabla \left(\left\| \nabla I_s \left(x_i \right) \right\| \right) \tag{4}$$

where $I_s \left(x_i \right)$ is the intensity of a pixel at the location of node n_i in a smoothed version of the image. Image forces that attract the model to an image boundary are calculated only for boundary mesh nodes (similarly image forces that attract medial model nodes to medial features can also be applied).

Following the calculation of the node forces we compute the new acceleration, velocity, and position of each node given the old velocity and position values, as follows (explicit Euler solution with time step Δt)

$$a_i = f_i / m_i \ , \quad v_i = v_i^{old} + a_i \Delta t \ , \quad x_i = x_i^{old} + v_i \Delta t \tag{5}$$

3. SHAPE DEFORMATION

In this section we describe how shape deformation is implemented using external forces and internal spring actuation.

3.1. Shape deformation using external forces

As explained in section 2, deformations can be produced via the application of external forces such as user interaction (Figure 3) or image forces.

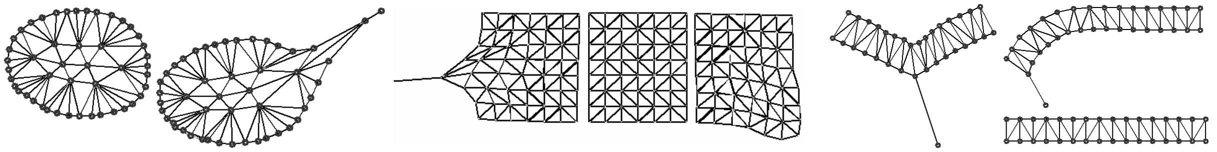


Figure 3. Examples of deformations via user interaction ('mouse' forces).

3.2. Deformations using spring actuation

Other forces result from spring actuation (in a manner analogous to muscle actuation in animals). Two nodes connected by a spring will normally change position until the spring is at its rest length. To actuate a spring we change its rest length while continuously simulating the mesh dynamics.

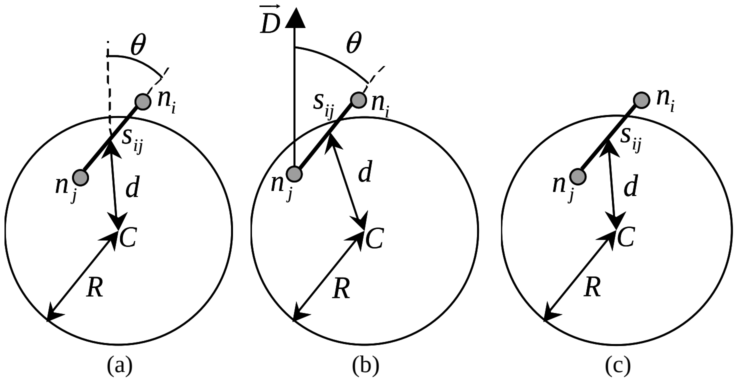
3.2.1. Operator-based localized deformations

Bulging (radial bulge), stretching (directional bulge), bending, tapering, and scaling deformations are implemented using spring actuation. These operator-based deformations can be applied at different locations and scales with varying amplitudes.

To perform a (radial) bulge deformation we specify a center C and a radius R of a deformation region (Figure 4a) as well as a deformation amplitude K. We then update the rest length r_{ij} of each spring s_{ij} if at least one of its terminal nodes, n_i or n_j, lies within the deformation region, as follows

$$r_{ij} = \left(\left(1-\frac{d}{R}\right)\left(1-\frac{2\theta}{\pi}\right)(K-1)+1\right)r_{ij}^{old} \qquad (6)$$

where $\theta \in \left[0, \frac{\pi}{2}\right]$ is the angle between s_{ij} and the line L connecting the midpoint of the spring with C, and d is the length of L (Figure 4a). The resulting effect of the above equation is that springs closer to C and with directions closer to the radial direction are affected more (Figure 5).



(a) (b) (c)

Figure 4. Definition of variables for (a) radial bulge, (b) directional bulge, and (c) localized scaling.

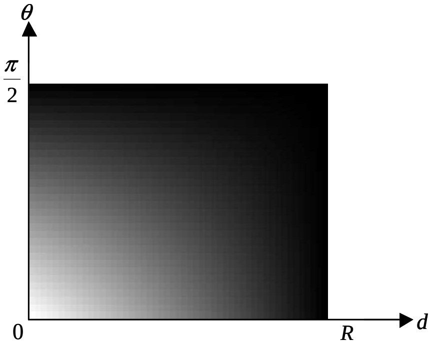


Figure 5. The factor (white=*K*, black=1) by which r_{ij}^{old} is multiplied as a function of θ and d.

To perform a stretch (directional bulge) we again specify a deformation region and amplitude as well as a direction $\vec{D}$ (Figure 4b). We update the rest length of each spring as in equation (6) where $\theta \in \left[0, \frac{\pi}{2}\right]$ is now defined as the angle between s_{ij} and $\vec{D}$ (Figure 4b). The resulting effect in this case is that springs closer to C and with directions closer to the stretch deformation direction are affected more (Figure 5).

A localized scaling deformation is independent of direction and requires only the specification of a deformation region and amplitude (Figure 4c). The rest length update equation then becomes

$$r_{ij} = \left(\left(1 - d/R\right)(K-1) + 1\right) r_{ij}^{old}. \tag{7}$$

To perform localized bending, we specify bending amplitude K and two regions surrounding the medial axis (Figure 6). The rest lengths of the springs on one side of the medial are increased according to

$$r_{ij}^{1} = \left(\frac{d_1}{R_1}\left(1 - \frac{2\theta_1}{\pi}\right)(K-1) + 1\right) r_{ij}^{1,old} \tag{8}$$

while the rest lengths on the other side are decreased according to

$$r_{ij}^{2} = \left(\frac{d_2}{R_2}\left(1 - \frac{2\theta_2}{\pi}\right)\left(\frac{1}{K} - 1\right) + 1\right) r_{ij}^{2,old}. \tag{9}$$

To perform localized tapering, we specify tapering amplitude K and a region with a base (Figure 7). The rest lengths on one side of the base are increased according to

$$r_{ij}^{1} = \left(\frac{d_1}{R_1}(K-1) + 1\right) r_{ij}^{1,old} \tag{10}$$

while those on the other side are decreased according to

$$r_{ij}^{2} = \left(\frac{d_2}{R_2}\left(\frac{1}{K} - 1\right) + 1\right) r_{ij}^{2,old} \tag{11}$$

Different examples of localized operator-based deformations are shown in Figure 8.

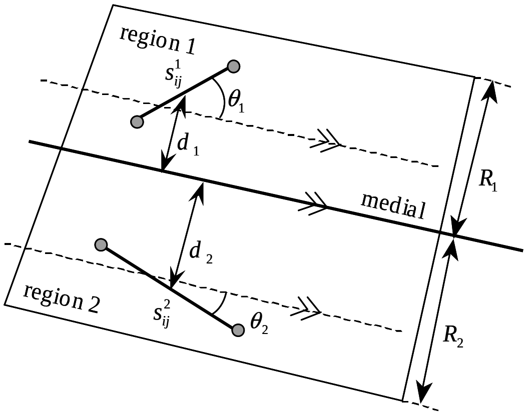


Figure 6. Definition of variables for the localized bending deformation operator.

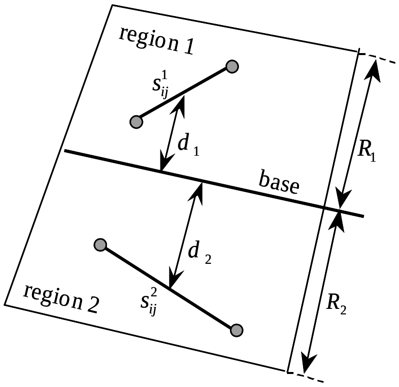


Figure 7. Definition of variables for the localized tapering deformation operator.

3.2.2. Statistical or learned deformations

Statistical or learned deformations are also implemented via spring actuation. To facilitate intuitive deformations, springs are designed to be of different types: stretch springs, bend springs, or thickness springs. Stretch springs connect neighboring medial nodes, bending springs are hinge springs that connect non-consecutive medial nodes, and thickness springs connect medial nodes with boundary nodes (Figure 9). Actuating the stretch springs causes stretch deformations, actuating hinge springs causes bend deformations, and actuating thickness springs causes bulging, squashing, or tapering deformations. Feasible mesh deformations are obtained by actuating springs according to the outcome of a statistical analysis performed on the spring lengths of a training set (discussed in section 3.4).

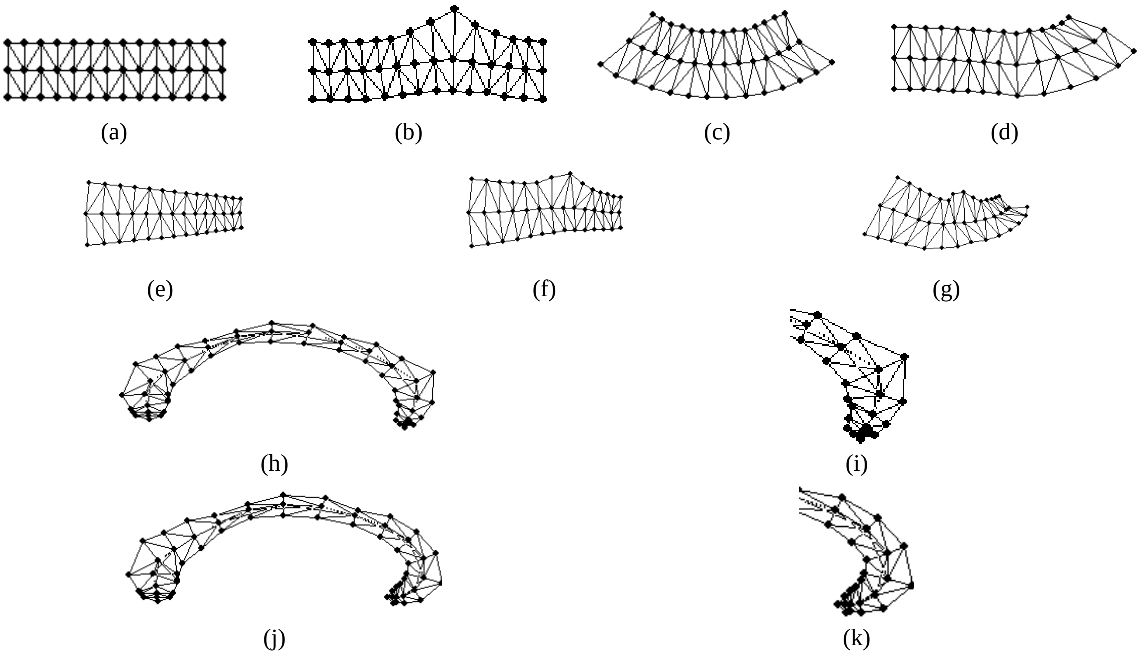


Figure 8. Examples of localized deformations. (a) Initial synthetic object, (b) bulge, (c) bend, (d) bend at another location, (e) tapering, (f) tapering followed by a bulge, and (g) tapering followed by a bulge and a bend deformations. CC model (h) before and (j) after a localized bend. (i,k) Close up versions of (h,j).

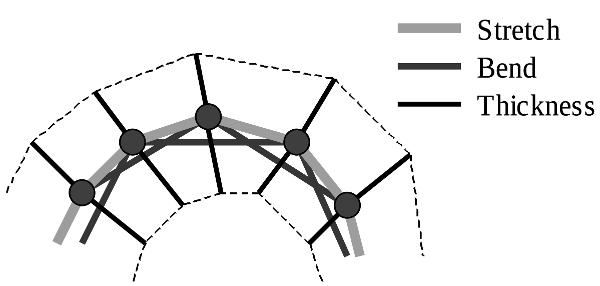


Figure 9. Spring types used for statistics-based deformations.

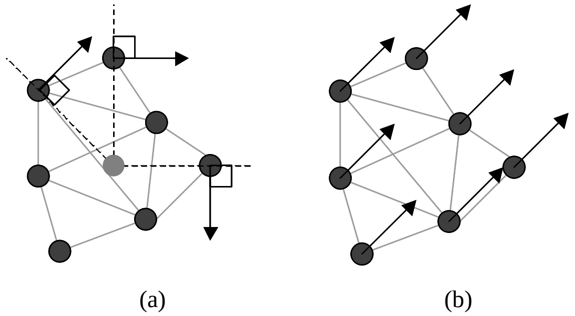


Figure 10. External forces for performing a (a) rotation (light gray circle marks center of mass) and a (b) translation.

3.3. Affine transformations

Rotation and translation are implemented via the application of external forces. Scaling is implemented by muscle actuation. Scaling by a factor of S is performed by changing the rest length of all the springs, i.e. $r_{ij} = S \cdot r_{ij}^{old}$. Rotation forces are applied on all nodes in a direction normal to the line connecting each node with the center of mass of the model, with a consistent clockwise/counter clockwise direction (Figure 10a). Translation forces are applied on all nodes in the direction of the desired translation (Figure 10b). Examples are shown in Figure 11.

3.4. Hierarchical Regional PCA

To produce feasible (i.e. similar to what has been observed in a training set) shape deformations at different locations and scales, we perform a principal component analysis (PCA) of the spring lengths corresponding to the desired localized deformations as explained below.

The set of rest lengths for the stretch springs (Figure 9) in a single example model are collected in a vector $\mathbf{r}_S$, i.e.

$$\mathbf{r}_S = \left\{ r_{ij} \forall i,j : s_{ij} \in \text{stretch springs} \right\} \tag{12}$$

and similarly for the bending and left and right thickness springs (Figure 9)

$$\begin{aligned} \mathbf{r}_B &= \left\{ r_{ij} \forall i,j : s_{ij} \in \text{bend springs} \right\} \\ \mathbf{r}_{TL} &= \left\{ r_{ij} \forall i,j : s_{ij} \in \text{left thickness springs} \right\} \\ \mathbf{r}_{TR} &= \left\{ r_{ij} \forall i,j : s_{ij} \in \text{right thickness springs} \right\} \end{aligned} . \tag{13}$$

This gives

$$\begin{aligned} \mathbf{r}_S &= \left[\mathbf{r}_S^1, \mathbf{r}_S^2, \ldots, \mathbf{r}_S^{N_S} \right] \\ \mathbf{r}_B &= \left[\mathbf{r}_B^1, \mathbf{r}_B^2, \ldots, \mathbf{r}_B^{N_B} \right] \\ \mathbf{r}_{TL} &= \left[\mathbf{r}_{TL}^1, \mathbf{r}_{TL}^2, \ldots, \mathbf{r}_{TL}^{N_T} \right] \\ \mathbf{r}_{TR} &= \left[\mathbf{r}_{TR}^1, \mathbf{r}_{TR}^2, \ldots, \mathbf{r}_{TR}^{N_T} \right] \end{aligned} \tag{14}$$

where N_S, N_B, N_T are the numbers of stretch, bend, and left/right thickness springs and the springs are ordered spatially (i.e. moving from one end of the medial to the other we encounter $\mathbf{r}_S^1, \mathbf{r}_S^2, \ldots, \mathbf{r}_S^{N_S}$).

Performing global (traditional) PCA on corresponding variables in a training set gives (details on obtaining the corpus callosum training set can be found in section 4)

$$\begin{aligned} \mathbf{r}_S &= \overline{\mathbf{r}}_S + M_S \mathbf{w}_S \\ \mathbf{r}_B &= \overline{\mathbf{r}}_B + M_B \mathbf{w}_B \\ \mathbf{r}_{TR} &= \overline{\mathbf{r}}_{TR} + M_{TR} \mathbf{w}_{TR} \\ \mathbf{r}_{TL} &= \overline{\mathbf{r}}_{TL} + M_{TL} \mathbf{w}_{TL} \end{aligned} \tag{15}$$

where the columns of M_S , M_B , M_{TR} , M_{TL} are the main modes of spring length variation. Associated with each mode is the variance it explains.

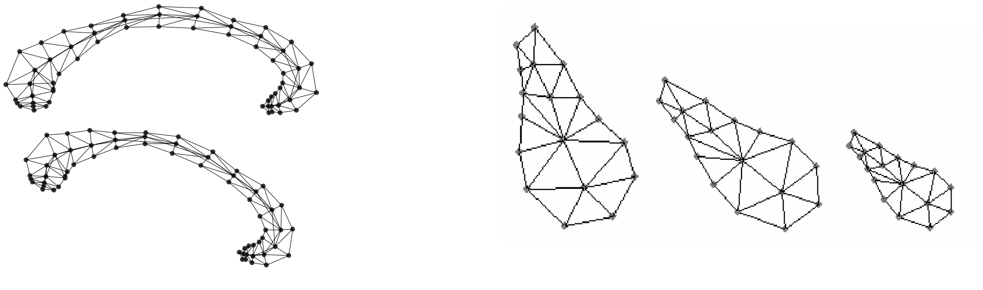


Figure 11. Affine transformation via external forces. (a) Rotating a model of the corpus callosum. (b) Rotating and scaling a synthetic model.

For capturing the shape variations at different locations and scales, we study the variations in the rest lengths of the springs in the desired localized region. Furthermore, to decompose the variations into different types of general deformations, each statistical analysis of the spring length in a localized region is restricted to a specific type of deformation springs (Figure 9). Accordingly, the PCA becomes a function of the deformation type, location and scale.

For example, to analyze the local variation in object length (stretch), we perform a statistical analysis on the lengths of the stretch springs of that local region. In general, for a single deformation/location/scale- specific PCA we obtain

$$\mathbf{r}_{def,loc,scl} = \overline{\mathbf{r}}_{def,loc,scl} + M_{def,loc,scl}\mathbf{w}_{def,loc,scl} \tag{16}$$

where def is the deformation type being either, S (for stretch), B (for bend), TL (for left thickness) or TR (for right thickness). The location and scale, set by the choice of loc and scl respectively, determine which springs are included in the analysis according to

$$\mathbf{r}_{def,loc,scl} = \left[\mathbf{r}_{def}^{loc}, \mathbf{r}_{def}^{loc+1}, \ldots, \mathbf{r}_{def}^{loc+scl-1}\right]. \tag{17}$$

For example, for the bending deformation at location 'five' with scale 'three' ($def, loc, scl = B, 5, 3$) we have

$$\mathbf{r}_{def,loc,scl} = \mathbf{r}_{B,5,3} = \left[\mathbf{r}_B^5, \mathbf{r}_B^6, \mathbf{r}_B^7\right]. \tag{18}$$

The average values of the spring lengths are calculated according to

$$\overline{\mathbf{r}}_{def,loc,scl} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{r}(j)_{def,loc,scl} \tag{19}$$

where $\mathbf{r}(j)_{def,loc,scl}$ is $\mathbf{r}_{def,loc,scl}$ obtained from the j^{th} training example and N is the number of training examples. The columns of $M_{def,loc,scl}$ are the eigenvectors, $m_{def,loc,scl}$, of the covariance matrix $C_{def,loc,scl}$. That is

$$\{C\mathbf{m} = \lambda\mathbf{m}\}_{def,loc,scl} \tag{20}$$

where

$$\left\{C = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{r}(j) - \overline{\mathbf{r}})(\mathbf{r}(j) - \overline{\mathbf{r}})^T\right\}_{def,loc,scl} \tag{21}$$

and where $\{\ \}_{def,loc,scl}$ denotes deformation type-, location-, and scale- specific PCA variables.

The data set needs to be aligned only with respect to scale. The statistical analysis of spring lengths is independent of orientation and translation. See the different examples in Figure 12 to Figure 15.

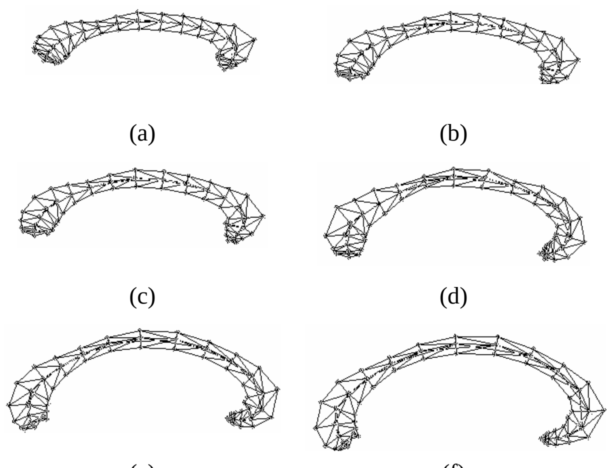

(a) (b) (c) (d) (e) (f)

Figure 12. Sample corpus callosum mesh model deformations (1st PC for all deformation types over the entire CC) derived from the hierarchical regional PCA.

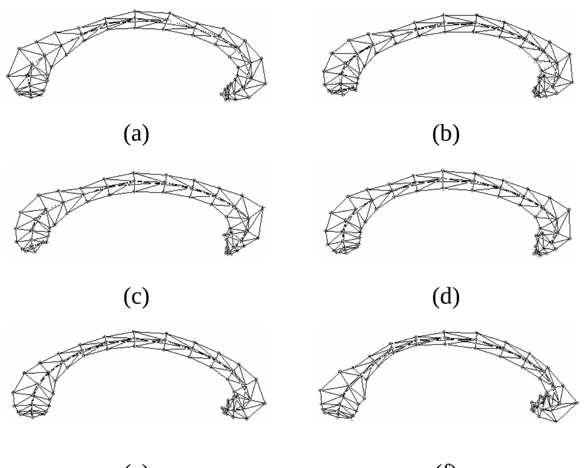

(a) (b) (c) (d) (e) (f)

Figure 13. Sample CC mesh model deformations (2nd PC for all deformation types over the entire CC) derived from the hierarchical regional PCA.

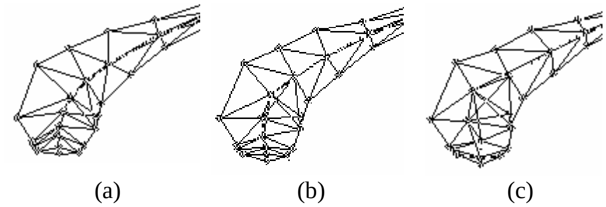


Figure 14. Statistical CC mesh model deformations: Stretching the splenium.

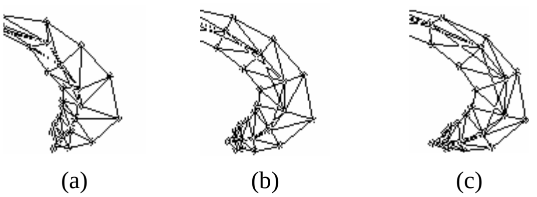


Figure 15. Statistical CC mesh model deformations: Bending the genu.

4. MESH GENERATION FROM REAL DATA

From 51 MRI brain volumes, we extracted the mid-sagittal slices from the coronal slices. We then used human expert segmented corpus callosum images (Figure 16a) to compute the set of spatially ordered boundary coordinates (Figure 16b). We calculated a pruned (using morphological operations) skeleton to produce a medial axis (Figure 16c-d) represented by spatially ordered coordinates. We then sampled the medial and boundary coordinates (we experimented with critical point detection algorithm[16], fitting line segments[17], in addition to uniform/equal arc length sampling and non-uniform sampling). We then constructed the mesh by finding the boundary points closest to the line normal to the sampled medial points. Since Delaunay triangulation does not guarantee correspondence between the meshes in different examples, we hand crafted the spring connections and applied it to all the training data (Figure 16e-f).

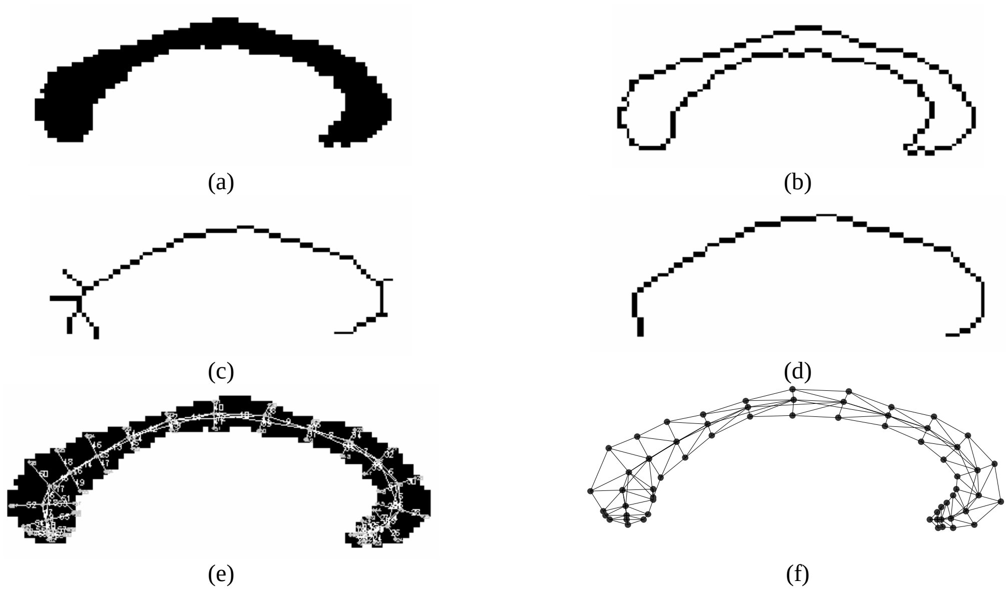


Figure 16. CC mesh generation: (a) expert segmented CC images, (b) extracted boundary pixels, (c) skeleton, (d) pruned skeleton/medial axis, (e) spring connections, and (f) final CC mesh model.

5. CONCLUSION

A key requirement of a deformable model-based medical image analysis system is the ability to intelligently schedule and control the type, location, extent, and order of intuitive model deformations during the fitting process. In this paper we investigated the use of a physics-based shape representation and deformation technique to meet such a requirement. This investigation is the first step for using the proposed shape representation and deformation method as the lower layers of a recently developed multi-layered intelligent model-fitting system[18].

Several interesting issues are currently under consideration for further exploration. For example, the circular deformation region may be too restrictive for more complex-shaped mesh models. We are also investigating the extension of the controlled physics based deformations to 3D meshes, and the use of a more stable and efficient algorithm for animating the mass-spring system[19]. We are also exploring other triangulation algorithms to automate the generation of corresponding meshes.

REFERENCES

1. T. McInerney and D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey", *Medical Image Analysis*, Vol. 1, No. 2, pp. 91-108, 1996.
2. J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, and M.J. Wozny, "Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data", *Proceedings of SIGGRAPH*, Vol. 25, No. 4, pp. 217-226, 1991.
3. J. Montagnat and H. Delingette, "Volumetric Medical Image Segmentation using Shape Constrained Deformable Models", *Proceedings of Computer Vision, Virtual Reality and Robotics in Medicine*, pp. 13-22, 1997.
4. C. Mandal, B.C. Vemuri, and H. Qin, "A New Dynamic FEM-Based Subdivision Surface Model for Shape Recovery and Tracking in Medical Images", *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, Vol. 1496, pp. 753-760, 1998.
5. J.O. Lachaud and A. Montanvert, "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.
6. A.H. Barr, "Global and Local Deformations of Solid Primitives", *Proceedings of SIGGRAPH, Computer Graphics*, Vol. 18, No. 3, pp. 21-30, 1984.
7. T.W. Sederberg and S.R. Parry, "Free-Form Deformation of Solid Geometric Models", *Proceedings of SIGGRAPH'86*, Vol. 4, No. 20, pp. 151-160, 1986.
8. S. Coquillart, "Extended Free Form Deformations: A Sculpting Tool for 3D Geometric Modeling", *Proceedings of SIGGRAPH*, Vol. 24, No. 2, pp. 187-196, 1990.
9. K. Singh and E. Fiume, "Wires: A Geometric Deformation Technique", *Proceedings of SIGGRAPH'98*, Vol. 99, No. 1, pp. 405-414, 1998.
10. D. Terzopoulos and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 703-714, 1991.
11. S. Pizer, D. Fritsch, "Segmentation, Registration, and Measurement of Shape Variation via Image Object Shape", *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, pp. 851-856, 1999.
12. G. Hamarneh and T. McInerney, "Controlled Shape Deformations via Medial Profiles", *Proceedings of Vision Interface*, pp. 252-258, 2001.
13. D. Molloy and P. Whelan, "Active-Meshes", *Pattern Recognition Letters*, Vol. 21, pp. 1071-1080, 2000.
14. T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham, "Active Shape Models – Their Training and Application", *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 38-59, 1995.
15. G. Szekely, A. Kelemen, Ch. Brechbuehler, and G. Gerig, "Segmentation of 3D Objects from MRI Volume Data using Constrained Elastic Deformations of Flexible Fourier Surface Models", *Medical Image Analysis*, Vol. 1, No. 1, pp. 19-34, 1996.
16. P. Zhu and P.M. Chirlian, "On Critical Point Detection of Digital Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17(8), pp. 737-748, 1995.
17. A. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
18. G. Hamarneh, T. McInerney, and D. Terzopoulos, "Deformable Organisms for Automatic Medical Image Analysis", *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, pp. 66-75, 2001.
19. M. Desbrun, P. Schröder, A. Barr, "Interactive Animation of Structured Deformable Objects", *Proceedings of Graphics Interface*, pp. 1-8, 1999.