An Efficient 3D Deformable Model with a Self-Optimising Topology

Andrew J Bulpitt & Nicholas D Efford
School of Computer Studies
University of Leeds, Leeds, LS2 9JT UK
{andy,nde}@dcre.leeds.ac.uk

Abstract

Deformable models are a powerful and popular tool for image segmentation, but in 3D imaging applications the high computational cost of fitting such models can be a problem. A further drawback is the need to select the initial size and position of a model in such a way that it is close to the desired solution. This task may require particular expertise on the part of the operator, and, furthermore, may be difficult to accomplish in three dimensions without the use of sophisticated visualisation techniques.

This article describes a 3D deformable model that uses an adaptive mesh to increase computational efficiency and accuracy. The model employs a distance transform in order to overcome some of the problems caused by inaccurate initialisation. The performance of the model is illustrated by its application to the task of segmentation of 3D MR images of the human head.

1 Introduction

Real images are often so noisy and complex that one cannot expect local low-level operations (e.g., edge detection by convolution) to generate good descriptions of object shape. A higher level approach must be adopted if a smooth, continuous contour or surface is to be obtained. A considerable body of work now exists describing how this can be achieved using models of object shape that deform elastically under the influence of image-derived forces [1, 2, 3, 4, for example]. The material properties of the model constrain deformation and provide some measure of immunity to image noise and missing data.

Deformable model fitting can be viewed as a process of energy minimisation. The traditional, variational approach to the minimisation problem involves finding a solution to the Euler-Lagrange equations that describe model dynamics. Amini *et al.* [5] have raised several issues of concern regarding variational techniques: optimality is not guaranteed because of the non-convexity of the image-derived external force field; hard constraints can be properly incorporated only if they are differentiable, which is often not the case; high order derivatives of discrete, noisy data are computed, and the resulting amplification of high frequency noise increases the chances of numerical instability. Leymarie and Levine [6] note four other major problems: there is an intrinsic bias toward solutions which reduce the size of the model; allowing elasticity to be adaptive leads to computational difficulties; oscillatory or chaotic behaviour is possible if no bounds are placed on the forces that act

BMVC 1995 doi:10.5244/C.9.4

on the model; the steady-state criterion for convergence can actually cause the model to recede from a valid solution.

Attempts have been made to address these issues within the variational framework [6]; however, other workers have sought alternative computational strategies for energy minimisation which do not suffer from the same problems. Amini *et al.* [5] introduced a dynamic programming algorithm for 2D models which takes advantage of the inherently discrete nature of boundary finding in digital images. Their algorithm is numerically stable, guarantees convergence within a finite number of iterations and allows hard constraints to be incorporated easily. However, it has storage requirements of $O(nm^2)$ and a computational complexity of $O(nm^3)$, n being the number of model nodes and m the size of the neighbourhood around each node. It would not be feasible to use this technique for surface fitting in three dimensions. Williams and Shah [7] have developed a simpler, 'greedy' algorithm for fitting 2D models in which each node seeks to minimise its energy irrespective of neighbouring nodes. This has a rather more favourable computational complexity of $O(nm)$.

A further drawback of many deformable models is the need to initialise the model close to the desired solution. For many applications, such as the segmentation of medical images, this requires the intervention of an expert operator. Moreover, with 3D datasets this interactive approach to initialisation may not be feasible, due to the difficulty of simultaneously visualising both model and data in three dimensions.

In this paper, we present an extension into three dimensions of Williams and Shah's 2D active contour model [7]—hereafter referred to as the W-S model. Like the latter, our model uses simple local computations to determine the deformation that takes place at each iteration; unlike the W-S model, it can adapt its topology by refining and decimating the deforming mesh locally, according to the requirements of the data. This technique allows the overall resolution of the model—and hence the computational cost—to be kept to a minimum without the loss of important surface detail. It also prevents undue creasing and stretching of the surface. Delingette [8] employs a similar refinement technique applied to Simplex Meshes. A second important enhancement is the use of a *distance transform*. This removes the need to place the model so close to the desired solution. The distance transform ensures that model nodes are always under the influence of image forces and therefore do not remain stationary if too far from the desired edge. A similar technique has been successfully applied to the related problem of image registration [9].

The following section describes the key features of the model. Sections 3 and 4 present the methods used to adapt mesh topology. Section 5 presents the results of using the model to extract surfaces from 3D MR images of the human head. Note that the head is not a challenging target for segmentation; it can be found with relative ease in MR images, e.g., by applying the marching cubes algorithm [10] to thresholded voxel data. We use the head here because it is a familar shape containing both concave and convex regions of varying complexity, and we are interested in how the model performs in these regions. Our conclusions appear in section 6.

2 3D Model

Our model extends the 2D W-S model [7] into three dimensions. The W-S model was selected as a basis for the work for reasons of numerical stability; ease of implementation;

the ease with which hard constraints can be added; and speed (it is much faster than dynamic programming methods).

A triangular mesh is employed in the model to represent the model surface. Each node is surrounded by six neighbours. The initial shape of the mesh is nominally that of a sphere surrounding the object of interest. During each iteration, a 3D neighbourhood surrounding each node of the mesh is examined and the energy associated with each position in the neighbourhood is computed. The node is then moved to the position in the neighbourhood with the lowest energy. The energy minimised by the algorithm is given by

$$E = \sum_{i=1}^{N} (\alpha_i E_{\text{cont}} + \beta_i E_{\text{curv}} + \gamma_i E_{\text{img}} + E_{\text{ext}}), \tag{1}$$

where E_{cont} and E_{curv} are the first and second-order continuity constraints that represent the internal energy of the model, E_{img} is the energy due to external image forces and E_{ext} is the energy arising from any external constraints applied to the model, such as invalid positions of a node or a minimum spacing between nodes. The parameters α, β and γ control the relative influence of the internal and external forces experienced by each node of the model.

The first term, E_{cont}, measures the distance of a node from the centre of gravity of its neighbours. Since energy is being minimised, this helps ensure that the nodes remain evenly spaced within their neighbourhood and thereby prevents the bunching of nodes. This measure also helps to reduce the chance of crossover between nodes in the mesh—which would result in self-intersection of the surface. We assume

$$E_{\text{cont}} = |\mathsf{P}_i - \frac{\sum_n \mathsf{P}_j}{n}|, \tag{2}$$

where P_i is the position vector of the current node i and n is the number of neighbours to that node. The distance measure is normalised using the mean spacing of nodes in the surrounding neighbourhood in order to allow for regions with a higher density of nodes.

The second term, E_{curv}, approximates the curvature of the surface, using the distance of a node from the average plane of its neighbours (Figure 1). The average plane is constructed using the normals, n_i, and areas, A_i, of the elements containing the node, and their centroid, x_i:

$$\hat{\mathsf{n}} = \frac{\mathsf{n}}{|\mathsf{n}|}, \quad \mathsf{n} = \frac{\sum_{\mathcal{E}} A_i \mathsf{n}_i}{\sum_{\mathcal{E}} A_i}, \quad \mathsf{x} = \frac{\sum_{\mathcal{E}} A_i \mathsf{x}_i}{\sum_{\mathcal{E}} A_i}, \tag{3}$$

where $\mathcal{E}$ is the set of all elements containing the node. The distance, d, of the node to the average plane, p, is then

$$d = |\hat{\mathsf{n}} \cdot (\mathsf{p} - \mathsf{x})|. \tag{4}$$

The distance is normalised using the size of the neighbourhood of points used to construct the plane. This metric is similar to that used in the decimation algorithm developed by Schroeder *et al.* [11].

The external constraints (E_{ext}) are used to prevent the surface moving outside the volume defined by the image.

Two types of image potential provide the image forces on the model. Initially, the distance transform of the image [12] is used. This reduces the problems caused by the initial position of the model being too far away from the desired solution, as every voxel in the image will have a potential associated with it and all neighbouring voxels are unlikely to share

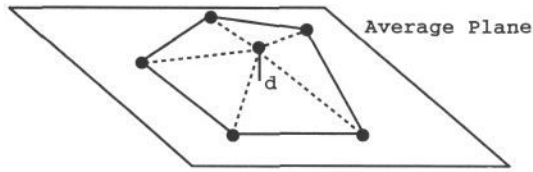

Figure 1: Distance to average plane.

the same potential. A node therefore need not remain stationary even if $\alpha E_{\text{cont}} + \beta E_{\text{curv}}$ is constant over the search neighbourhood. The distance transform can also be used to control the rate at which the model deforms. This is achieved by using the image term to force all nodes to be an equal distance from the surface on each iteration. This approach prevents some parts of the model reaching the surface much faster than other areas which can result in creasing of the surface.

When the surface is close to its final solution, the image grey level gradient is used to provide a driving force. This results in a better fit than that achieved by using the distance transform alone, as is shown in section 5.2.

The image term, E_{img}, is derived from the image intensity at a node, I_{node}. This is normalised using the maximum and minimum values of image intensity, I, within the search neighbourhood, $\mathcal{N}$, i.e.,

$$E_{\text{img}} = \frac{I_{\text{node}} - \min(I, \mathcal{N})}{\max(I, \mathcal{N}) - \min(I, \mathcal{N})}. \tag{5}$$

Large fluctuations in the value of E_{img} are prevented in regions where the image values are almost constant by ensuring that $\max(I, \mathcal{N}) - \min(I, \mathcal{N}) > 3$.

A further function of the distance transform is to control the values of α and β at each node of the model. A higher value of α prevents the bunching of nodes around strong edges in the image, but this may also prevent the model from deforming towards features of interest between such edges. Likewise a high value of β prevents excessive bending of the surface but may result in an overly smooth surface. The distance of the node from an edge is therefore used to relax the constraints on the nodes in the model as it approaches a solution, thereby allowing the nodes to seek out finer details in the image. For the results presented in section 5

$$\alpha_i = \begin{cases} 1 & E_{\text{img}} > 10 \\ \frac{E_{\text{img}}}{10} & E_{\text{img}} \leq 10 \end{cases} \tag{6}$$

3 Mesh Refinement

The computational cost of fitting the model is obviously dependent on the resolution of the mesh that is used. It is therefore desirable to have the lowest resolution possible, but this, of course, has to be traded against the need for sufficient detail in the fitted surface. To overcome this problem, our model is able to refine itself locally by adding extra nodes to the mesh in areas where they are required. Such areas are regions of more complex shape or places where the mesh has become 'overstretched', causing a loss of surface detail. In the human head examples illustrated here, the smoother portions of the surface (e.g., the top and back of the head) can be modelled adequately with a fairly coarse mesh, but the ears, eyes, nose and mouth all require more detail if they are to be represented accurately.

The criteria used to select elements for refinement are based on three metrics. The first of these uses the local curvature of the surface at a node. If the curvature or the change of curvature at that node is above a threshold value, then the elements containing the node are refined.

The second metric uses the distance of the model surface surrounding a node from the surface in the image when the model is close to its final state. The metric measures the distance transform of the image at the position of the node and also at points between neighbouring nodes along the edges of the mesh. This additional information is used to ensure that despite neighbouring nodes lying close to the image surface, the element between them does not smooth over important image information. If any of the points are found to be above a threshold distance from the image surface, this suggests that this region of the mesh may not have reached its best solution and the node's elements are refined as before.

The final metric measures the area of the elements in the mesh and refines elements which are large relative to neighbouring elements in the mesh. This allows elements that have become stretched and may therefore not fit the underlying surface accurately to be refined.

The mesh refinement technique used here is based on Rivara's local refinement algorithm [13]. A similar approach is adopted in [14]. The process consists of two steps; the bisection of an element, and a conforming operation which ensures that the properties of conformity, non-degeneracy and smoothness are maintained.

The bisecting operation bisects an element by its longest edge. This method of bisection has been shown to prevent the interior angles of an element from becoming acute [15]. It also improves the shape regularity of the elements.

The conforming operation is then used to ensure that the mesh possesses the properties required by the finite element method—where two adjacent elements must only share either a node or an edge [16].

4 Mesh decimation

During the deformation process many of the nodes of the mesh may become *redundant*, i.e., they are no longer required to produce an adequate description of the surface. Nodes may also bunch together, thereby inhibiting further deformation locally. The removal of such nodes is desirable, to improve both the efficiency of model fitting and also the quality of the resulting surface.

The decimation procedure used here is based on that of Schroeder *et al.* [11]. Two criteria are used to decide if a node should be removed from the mesh. The first is curvature-based, similar to that described for mesh refinement in section 3. In this case, the curvature (equation 4) must be less than a predetermined value for a node to be removed, indicating that the surface is very smooth and the removal of a node will therefore not signiifcantly alter the shape of the surface. The second criterion for decimation is when the average distance between a node and its neighbours falls below a predetermined value, indicating that the nodes are bunching together.

If a node is removed, the region surrounding that node must then be retriangulated in order to maintain a conforming mesh. The triangulation procedure adopted here is that described by Schroeder *et al.*.

5 MR Image Segmentation

We use the model here to extract the outer skin surface in 3D magnetic resonance images of the human head. This is a somewhat contrived example which serves as a means of evaluating model performance on shapes of moderate complexity.

5.1 Distance Transform Calculation

Calculation of the distance transform requires knowledge of the approximate position of object boundaries in the image. We acquire this by first smoothing the image using a median filter, in order to remove unwanted speckle noise. The filtered image is then thresholded and a simple 3D edge operator [17] is applied to identify voxels in the vicinity of the skin surface. This procedure works because, in our MR data, the contrast between the skin surface and the background is high. Finally, a distance transform is produced by convolving the edge image with the following 3D chamfer kernel [12]:

Forward

$$\begin{matrix} C & B & C \\ B & A & B \\ C & B & C \end{matrix} \qquad \begin{matrix} B & A & B \\ A & 0 & * \\ B & * & * \end{matrix} \qquad \begin{matrix} * & * & * \\ * & * & * \\ * & * & * \end{matrix}$$

Backward

$$\begin{matrix} * & * & * \\ * & * & * \\ * & * & * \end{matrix} \qquad \begin{matrix} * & * & B \\ * & 0 & A \\ B & A & B \end{matrix} \qquad \begin{matrix} C & B & C \\ B & A & B \\ C & B & C \end{matrix}$$

For Euclidean distances, A, B and C should be in the ratio $1 : \sqrt{2} : \sqrt{3}$. Integers with approximately this ratio are used in this case.

5.2 Results

We present results here for a $128 \times 128 \times 128$ MR image of the human head. The inital mesh is a sphere of radius 60 voxels containing 4098 nodes, placed at the centre of the image. The initial values of the mesh parameters are $\alpha = 1.0$, $\beta = 1.0$ and $\gamma = 1.4$. The model is allowed to deform until the number of nodes moving at each iteration remains almost constant. For these data, 50 iterations are typically required, consuming approximately 200 seconds of CPU time on a Silicon Graphics workstation (100 MHz R4000 processor, 64 Mb main memory).

Figure 2 is a series of snapshots of the model evolving towards a solution. In a–e, deformation was driven by the distance transform alone. In f, the grey level gradient supplied the external forces—and, as a result, the level of detail around the mouth and ears has improved.

Figure 3 shows a comparison of surfaces produced with (a) and without (b) any refinement of the mesh. Clearly, more detail is obtained around the ears and mouth by using the refinement process.

Figure 4 illustrates the mesh refinement/decimation procedure. One can see that the model has a higher resolution in the most complex areas of the surface (the eyes, ears and

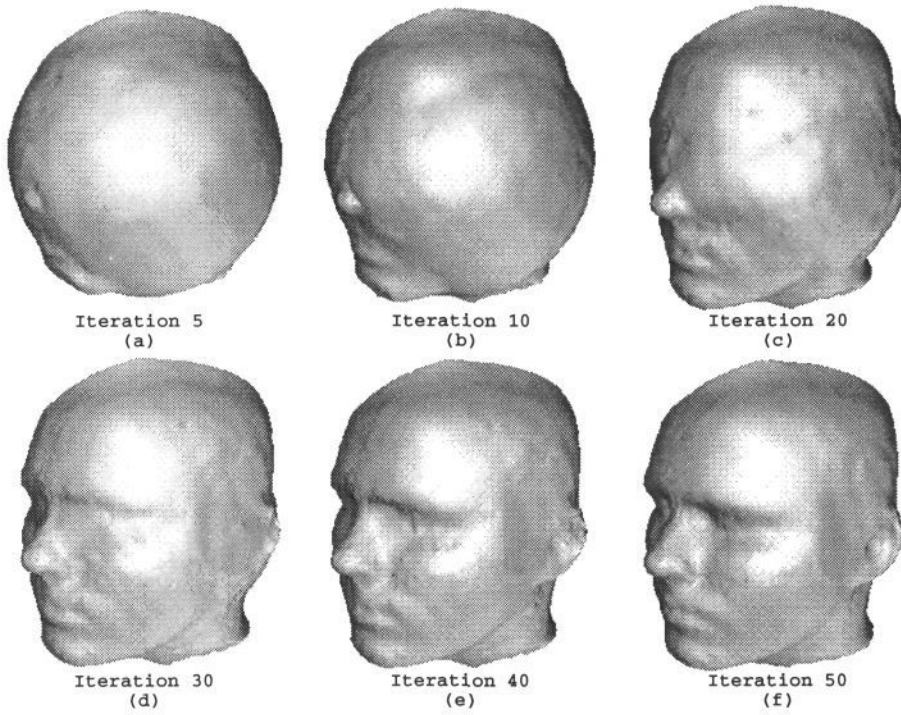

Figure 2: Surfaces produced during deformation.

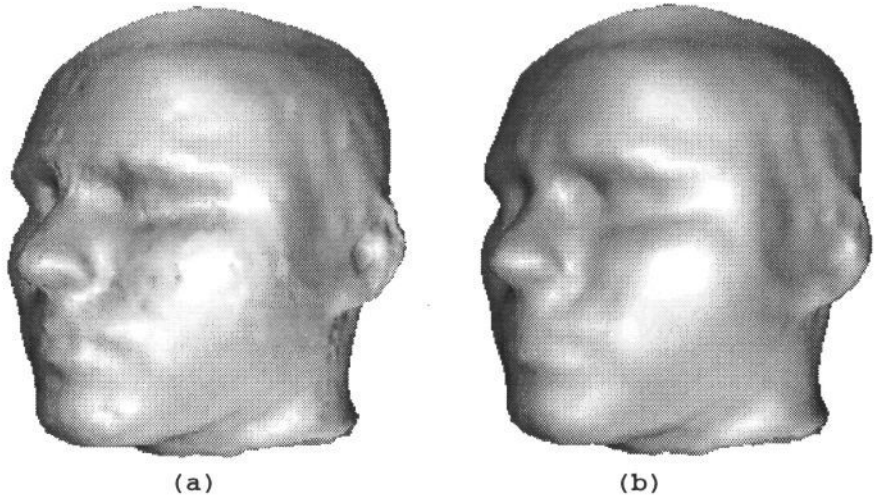

Figure 3: Comparison of surfaces produced with and without refinement or decimation.

nose) and a lower resolution in smoother regions (the top and back of the head). The final mesh employs 6684 nodes. By comparison, a contour stitching algorithm [18] will generate over 10,000 nodes for the same dataset.

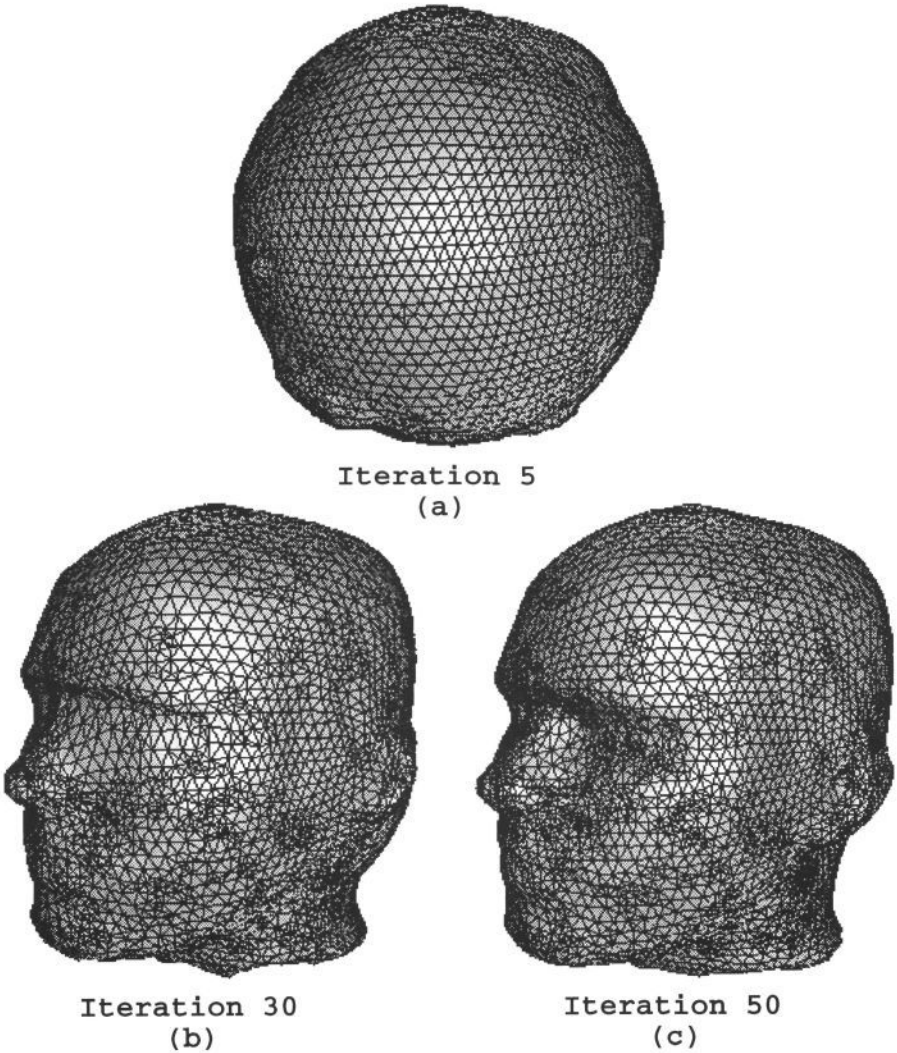

Figure 4: Refined mesh produced by the adaptive model.

Figure 5 compares surfaces produced with and without the adaptation of parameter α in equation 1. In a, nodes bunch around the edges of the eyes because α is too low. This produces large mesh elements stretched across the eyes, and hence a loss of detail. In c, too high a value of α leads to a smoother surface with the loss of some fine detail.

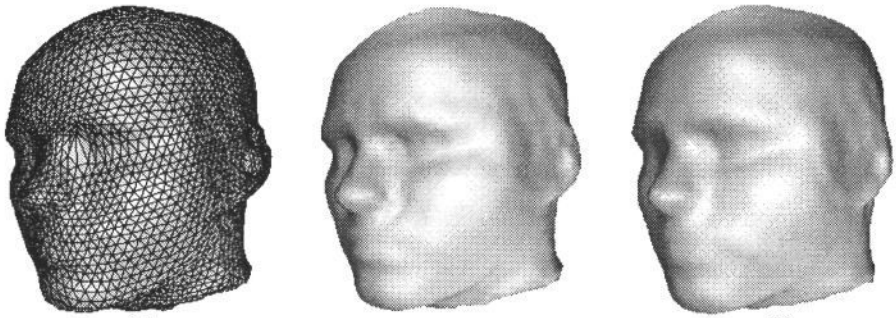

Figure 5: Effect of parameter α on model fitting.

6 Conclusions

The results presented here show how improvements to the performance and accuracy of a 3D deformable model are possible through

- use of a fast algorithm to compute deformation
- use of a distance transform to drive initial deformation
- automatic refinement of the mesh in areas of high detail
- automatic decimation of the mesh in smooth areas, where detail is low

The local adaptation of the parameters of the model can also be used to improve the surfaces produced; local adaptation of parameters prevents the bunching of nodes and also the stretching of elements which may result in the loss of important information.

The rules which govern mesh refinement, mesh decimation and parameter adaptation are clearly a very important factor in the successful application of the model and are hence the subject of current investigation.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[2] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.

[3] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *Proceedings of the Fourth International Conference on Computer Vision*, pages 518–523, 1993.

[4] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[5] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[6] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634, June 1993.

[7] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, January 1992.

[8] H. Delingette. Adaptive and deformable models based on simplex meshes. In *Proceedings of the IEEE Workshop on Motion of Non-Ridgid and Articulated Objects*, pages 152–157, 1994.

[9] N. A. Harrison D. L. G. Hill, D. J. Hawkes and C. F. Ruff. A strategy for automated multimodality image registration incorporating anatomical knowledge and imager characteristics. In *Proceedings of the 13th International Conference on Information Processing in Medical Imaging*, pages 182–196, 1993.

[10] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolutions 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[11] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, 1992.

[12] G. Borgefors. Distance transformations in arbitary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[13] M. C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20:745–756, 1984.

[14] D. Metaxas and E. Koh. Efficient shape representation using deformable models with locally adaptive finite elements. In *Proceedings SPIE: Geometric Methods in Computer Vision II*, volume 2031, pages 160–171, 1993.

[15] I. G. Rosenberg and F. Stenger. A lower bound on the angles of triangles constructed by bisecting the longest side. *Math. Comp.*, 29:390–395, 1975.

[16] G. Dhatt and G. Touzot. *The Finite Element Method Displayed*. John Wiley & Sons, 1984.

[17] S. W. Zucker and R. A. Hummel. An optimal three-dimensional edge operator. Technical Report 79-10, McGill University, April 1979.

[18] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Transactions on Graphics*, 11(3):228–258, 1992.