PDE Software for Digital Image Management

Phase I Final Technical Report

U. S. Army Research Office Contract DAAD19-00-C-0013

Jill Goldschneider
Lydia Ng
Vikram Chalana

Data Analysis Products Division,
MathSoft, Inc.

April 10, 2001

Data Analysis Products Division,
MathSoft, Inc.
1700 Westlake Ave. N, Suite 500
Seattle, WA 98109.9891, USA
Tel: (206) 283-8802
FAX: (206) 283-6310

E-mail: jrgold@statsci.com
lng@statsci.com
vikram@statsci.com

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20010416 071

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 10, 2001	02/01/00 to Final Report 07/31/00

4. TITLE AND SUBTITLE

PDE Software for Digital Image Management

5. FUNDING NUMBERS

Contract #:

DAAD19-00-C-0013

6. AUTHOR(S)

Jill Goldschneider

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MathSoft, Inc.
1700 Westlake Ave North
Suite 500
Seattle, WA 98109

8. PERFORMING ORGANIZATION REPORT NUMBER

None Assigned

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) (Program Mgr Name & Ph #)

U.S. Army Research Office
4300 S. Miami Blvd.
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

.

12b. DISTRIBUTION CODE

(Leave this block blank)

13. ABSTRACT:

Report developed under SBIR contract for topic "Army 99-026".

In Phase I research, we investigated and developed image segmentation and registration algorithms using curvature-based flow-driven nonlinear partial differential equation imaging models and algorithms, including total variation, level-set, and active contour techniques. We implemented the developed algorithms in C and S+, an open and extensible software environment, to allow us to do rapid prototyping and evaluation of the methods. We applied combinations of the methods developed to two applications in medical imaging, prostate segmentation and brain segmentation and registration.

14. SUBJECT TERM

variational problems, level sets, PDE imaging, image segmentation, image registration, medical imaging

15. NUMBER OF PAGES

41

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Contents

1 Final Report

The primary Phase I research and development objectives were to:

1. *Develop curvature-driven, flow-based multiscale image processing methods.* We proposed to develop image enhancement and segmentation methods using nonlinear partial differential equation based models and algorithms, level-set techniques, and active contour models.

2. *Integrate the proposed methods into an open and extensible software environment.* We proposed to integrate the developed methods into our open and extensible software environment featuring S-PLUS and Java-based client tools for processing and displaying images.

3. *Demonstrate the proposed methods on medical and tactical data.* We proposed to apply combinations of the methods developed to specific applications in medical and tactical imaging.

The primary Phase I results are:

1. MathSoft has focused on the development of level-set and variational level-set segmentation techniques based on [1–6].

2. MathSoft has implemented these algorithms in C and linked them to the S-PLUS language to enable scripting and simulation of complex applications. Our implementations can process both two- and three-dimensional image data.

3. MathSoft has applied these algorithms to the segmentation of the surface of the brain from a two-dimensional MRI dataset and the segmentation of the prostate from two-dimensional ultrasound images.

4. Level Set Systems, the Phase I subcontractor, has focused on the development of an active contours technique for curve evolution [7] that makes use of a Mumford-Shah [8] like fitting term.

A detailed discussion of the research conducted and our results is given in this section. We begin with a general overview of level-set methods in Section 2 and its application to image segmentation in Section 3. Implementation of level-set methods requires careful attention to accuracy and robustness as well as computational speed. In Section 3.1, we discuss the implementation issues we have encountered and solutions to them. Preliminary results are given in Section 3.2. Simple segmentation methods often fail in images where there are indistinct or missing boundaries. In Section 3.3, we discuss two incremental improvements we used to address this problem. In Sections 3.4 and 5, two new models using region-based features are introduced which provide further robustness to weak or blurred edges.

2 Introduction to Level-Set Methods

The level-set method developed in [9] has proven to be phenomenally successful in representing and following the motion of interfaces or surfaces, not only in image processing, computer vision and graphics, but also in fluid dynamics, materials science, combustion, and elsewhere. This approach represents a surface as a level-set of a smooth function, and moves the surface solely by evolving the representing function via a nonlinear partial differential equation (PDE). The velocity of the motion often involves curvature. This "implicit" representation of a surface turns out to have major advantages over traditional methods, especially in its ability to handle topological changes such as merging and pinching off. This formulation of the surface motion and the associated numerical methods were abstracted from techniques developed originally for combustion and fluid dynamics simulations.

We now describe briefly the basic level-set approach. The idea is to define a smooth level-set function $\varphi(\vec{x},t)$ on an image point $(\vec{x})$ that represents the evolving surface $\Gamma(t)$ as the set where $\varphi(\vec{x},t) = 0$, with $\varphi(\vec{x},t) < 0$ representing the interior bounded by the surface and $\varphi(\vec{x},t) > 0$ representing the exterior. The evolving surface is captured for all time with $\Gamma(\tau) = \{\varphi(\vec{x},t=\tau) = 0\}$. This deceptively trivial statement is of great significance because topological changes such as merging and breaking are well defined, and are performed automatically and implicitly.

The surface motion is modeled by convecting φ with the desired surface velocity field $\vec{v}$ (and arbitrary elsewhere) via the equation

$$\frac{\partial\varphi}{\partial t} + \vec{v}\nabla\varphi = 0. \tag{1}$$

Actually, only the normal component of v is needed $v_n = \vec{v}\cdot\frac{\nabla\varphi}{|\nabla\varphi|}$, so that Eq. (1) becomes

$$\frac{\partial\varphi}{\partial t} + v_n|\nabla\varphi| = 0. \tag{2}$$

Many physical systems evolve to minimize an energy functional, and the variational formulation leads to a very stable discretization. As mentioned above, these ideas have been used successfully in image processing [10], computer vision [1], and related fields. The energy functional, which involves surface energy and bulk energy, can be represented simply and completely by level-set functions.

Among the advantages of variational level-set methods are: (I) A numerically stable and robust algorithm is guaranteed; (II) a natural physical interpretation can be incorporated into the formulation; (III) constraints such a fixed volume, boundary conditions (based on a penalty method), and normals, curvatures, and other geometric quantities may be prescribed; (IV) shapes may be deformed dynamically by coupling to external physical laws [11].

In Phase I research, we focused on variational level-set methods and their application to the segmentation of the surface of the brain from a three-dimensional MRI dataset. In

Phase I option research, we focused on level-set methods and their application to (inter-patient) brain registration to allow automatic atlas-based segmentation of structures of interest using a three-dimensional MRI dataset.

3 Level Sets for Image Segmentation

The level-set method has been applied to the problem of image segmentation [2]. The basic idea is to start from an initial seed and propagate a contour or front $\Gamma(t)$ until the contour stops at the shape boundary. This is accomplished by defining a speed function v_n based on image gradient features. Note that the propagation of the contour is not done directly. Rather, it is done through the evolution of a level-set function φ. The advantages of this method are firstly that arbitrarily complex shapes (e. g. corners and protrusions) can be recovered, and secondly that more than one shape can be isolated at a time as topological changes are handled implicitly.

In [2], the speed function v_n has the form

$$v_n = k_I \left(\pm 1 - \epsilon\kappa\right) \tag{3}$$

where κ is the local curvature of the level-set. The parameter ϵ controls the smoothness of the propagating front, and hence its robustness to noise. The sign of the first term depends on whether we wish to propagate the interface outwards (positive) or inwards (negative).

The quantity k_I, known as the image speed term, depends on the image gradient

$$k_I = \frac{1}{1 + |\nabla G_\sigma * I(\vec{x})|} \tag{4}$$

where $(G_\sigma * I)$ denotes the convolution of the image with a Gaussian smoothing filter. k_I is close to one in regions of low image gradient and close to zero in regions of high gradient. Hence, k_I forces the propagation to stop near the shape boundary where we expect there to be a large change in the image intensity.

Evolution of the level-set function φ is governed by Eq. (2). We discuss the implementation of this evolution process in the next section.

3.1 Implementation of Level-Set Methods

3.1.1 Numerical Approximation

To prevent smoothing of discontinuities, such as corners, care is needed in the numerical approximation of the evolution equation, Eq. (2). Several entropy-satisfying schemes are outlined in [5]. These make use of upwind schemes where derivatives are calculated using only neighboring pixels from the direction of propagation.

3.1.2 Signed Distances Calculation

The level-set approach requires an initial function $\varphi(\vec{x}, t = 0)$ with the property that the zero level set $\{\varphi = 0\}$ corresponds with the initial front $\Gamma(0)$. A straightforward and computationally expensive technique is to calculate the signed distance function from each pixel, or image grid point, to the initial front. Note that the initial front $\Gamma(0)$ is defined on a continuous domain not just at grid points. As some level-set evolution schemes require periodic re-initialization of the level-set function to the signed distance function, a fast and accurate approximation method is needed. In our research, we investigated two signed distance approximation schemes: the distance transform operator [12] and the Fast Marching method [4].

Distance Transform Operator The distance transform operator [12], has the advantage that it can quickly approximate the distance from each image grid point to the zero level set since it requires only two passes through the image. We found that the major disadvantage of this method is that it assumes that points on the continuous zero level set are located only at regularly spaced lattice grid points. Approximating a point on the zero level set by the closet grid point results in significant loss of accuracy, particularly when re-initialization is done regularly.

Additionally, to use the distance transform for re-initialization, we have to explicitly extract the zero level set from the continuous function φ. One method is to construct a polygonal approximation from the grid cells that contain a zero crossing [2]. This requires a contour tracing procedure to link neighboring cells to form a closed polygon. Contour tracing is a difficult problem due to effects of noise, sharp protrusions and topology changes. We implemented a tracing procedure that effectively performs a depth-first search of the pixel/cell space. Although we have attempted to cover most contingencies, robustness is not guaranteed. Protrusions may be smoothed, and in some cases smaller contours are retained at the expense of larger contours being found.

Due to errors introduced in both the distance transform and contour extraction, we do not recommend this scheme for use in level-set implementations. In our experiments we found that these errors caused the zero level set to be perturbed away from shape boundaries.

Fast Marching The distance calculation from the zero level set can be viewed in terms of front propagation [4]. This simplifies to solving the following Eikonal equation

$$|\nabla T| = 1$$

where $T(\vec{x})$ is the arrival time of the front at image grid point $(\vec{x})$. The Fast Marching method [5] is a fast technique for solving Eikonal equations. Fast Marching requires only one pass through the grid points since it takes advantage of the causality relationship of the front propagation.

The difficulty lies in the initialization stage where we need to provide approximate distances at a set of *trial* points to begin the Fast Marching. We use an initialization method that tags as *trial* all image grid points inside the front with at least one neighbor outside

the front [4]. The approximate distance is calculated by linearly interpolating the level-set function value between the grid point and its neighbors to find the distance to the zero set.

The major advantages of this method are that it has fast computation and that the constructed signed distance has the same zero level set as the given level-set function. We have implemented this method in both in two and three dimensions, and have used it in all our segmentation algorithms.

3.1.3 Image Speed Extension

The image speed term k_I in Eq. (3) has meaning only on the front $\Gamma(t)$, as it was designed to force the propagation to stop near the shape boundary. However, to solve for φ in Eq. (2), k_I must be defined for all grid points. k_I can be extended as follows [2]: for every point P not on the front $\Gamma(t)$, $k_I(P)$ is assigned the value of $k_I(Q)$ where Q is the closest point to P that lies on the front. This extension method is computationally expensive and forms the major bottleneck when executing the algorithm.

The extension of k_I outward from the zero level set is a problem similar to that of the signed distance calculation from the zero level set. In both problems, we need to locate at every grid point the closest point on the zero level set.

The Fast Marching method we used to calculate the signed distance can also be used to simultaneously extend k_I or any other quantity [4]. During the fast marching, a grid point is assigned a k_I value that is the weighted average of the k_I values at the grid points used to compute the distance.

In the initialization phase, if k_I is defined on a finer grid than the calculation grid, sub-grid resolution can be taken into account to define the k_I values at the *trial* calculation grid points. These values are then extended smoothly out by the Fast Marching process. Another advantage of this method is that k_I is extended such that the signed distance function is maintained as the level-set function evolves, preventing the level sets from bunching up or spreading out.

Our implementation of the Fast Marching method can handle seamlessly both two- and three-dimensional datasets and can extend any number of quantities at the same time as it calculates the signed distance. We have used this implementation in all of our level-set segmentation algorithm prototypes.

3.1.4 Level-Set Evolution Algorithms

The flowchart of the basic level-set evolution algorithm is shown in Figure 1. In our implementation, Fast Marching is used in step (1) to calculated the signed distance function and to extend the image speed term k_I.

A more efficient algorithm can be implemented where the level-set function is updated only at a small set of points within a band of width δ around the zero level set of φ^n [2]. The width δ of the narrow band determines the number of points that need to be processed. The zero level set that lies inside the narrow band moves until it collides with the boundary

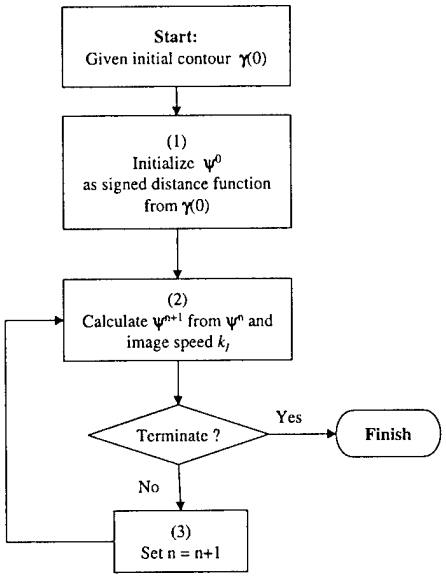


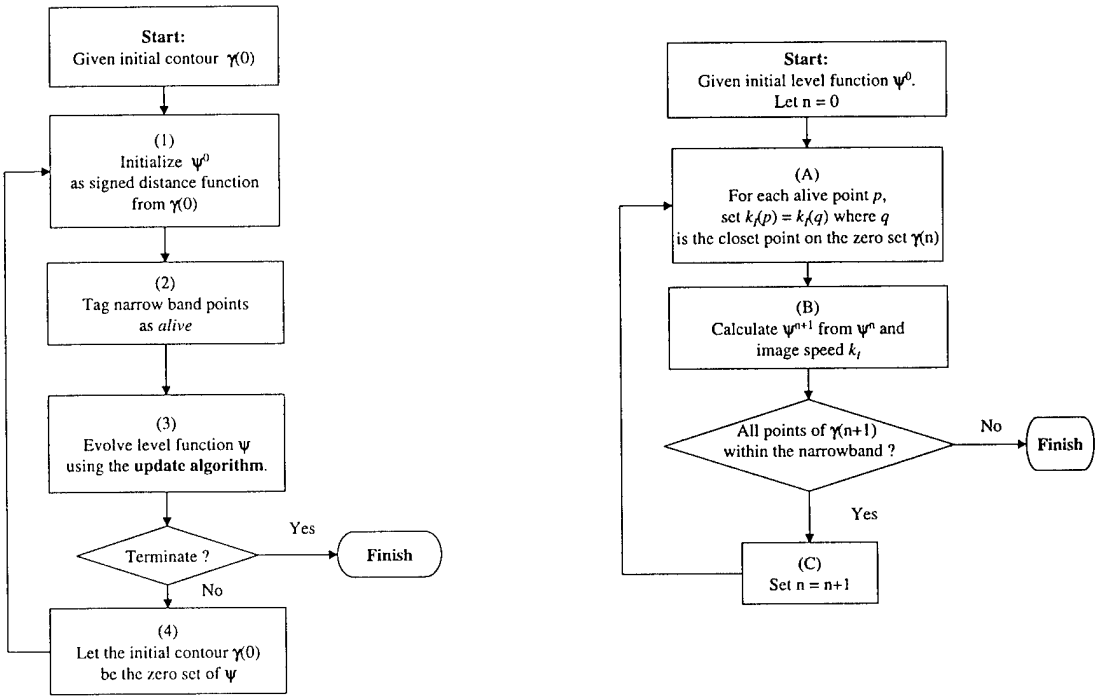
Figure 1: Flowchart of the basic level-set evolution algorithm.

of the narrow band. Upon collision, the level-set function φ is re-initialized by treating the zero level set of φ^n as the initial contour $\Gamma(0)$.

This narrow-band algorithm requires the selection of a parameter l, which is the number of time steps required to move the front by a distance of roughly $\delta/2$. The choice of l requires some experimentation. Alternatively, we can re-initialize φ when we detect a possible zero-passing through a grid point outside of the narrow-band. A flowchart for the narrow-band scheme is shown in Figure 2.

A third alternative is to re-initialize the level-set function at every iteration. Thus, the zero level set will always lie in the center of the narrow-band, and hence collision detection with the narrow-band boundary is not required. Re-initialization at every step is achieved at no extra computational expense as we are already required to extend k_I at every step. Using the Fast Marching technique both are done simultaneously. This scheme is diagrammed in Figure 3.

Identification of the narrow-band is another important implementation issue. Options include using morphological operations or moving along the zero level set and collecting points within a distance of $\delta/2$. Another alternative is to modify the Fast Marching re-initialization method to collect the coordinates of each point being processed, and to terminate the marching when the distance has reached a value greater than $\delta/2$. We have used the latter method in our algorithm prototypes.



(a) Narrow-band algorithm with updating.

(b) Update algorithm.

Figure 2: Flowchart of the narrow-band algorithm. (a) The overall algorithm. (b) The level-set function update algorithm.

3.1.5 Algorithm Framework

We have developed an extensible modular framework for level-set implementations. The framework can seamlessly handle two- and three-dimensional data. New algorithms can be rapidly prototyped by building new modules to extend the existing set of modules. For efficiency and portability the software was developed in C.

3.2 Preliminary Results

3.2.1 Application to Synthetic Data: Single Circle

This simple test image, Figure 4(a), is of a uniform intensity circle on a black background. The initial contour, Figure 4(b), is small square located within the circle, below and to the right of the center of the circle. The zero set expands uniformly until it reaches the boundary of the circle, Figure 4(c). The contour near the boundary stops expanding while the rest of the contour continues to grow until it reaches the rest of the circle boundary, Figure 4(d).

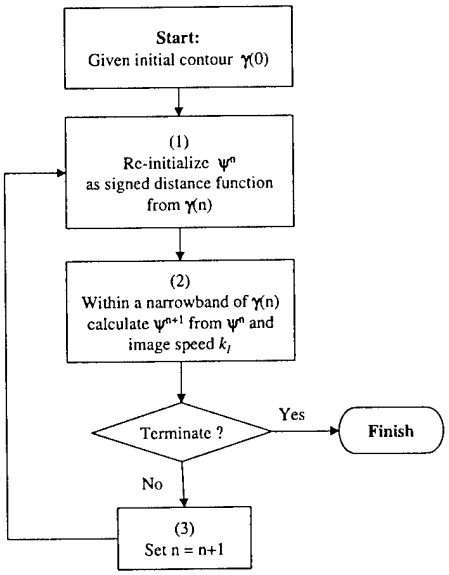


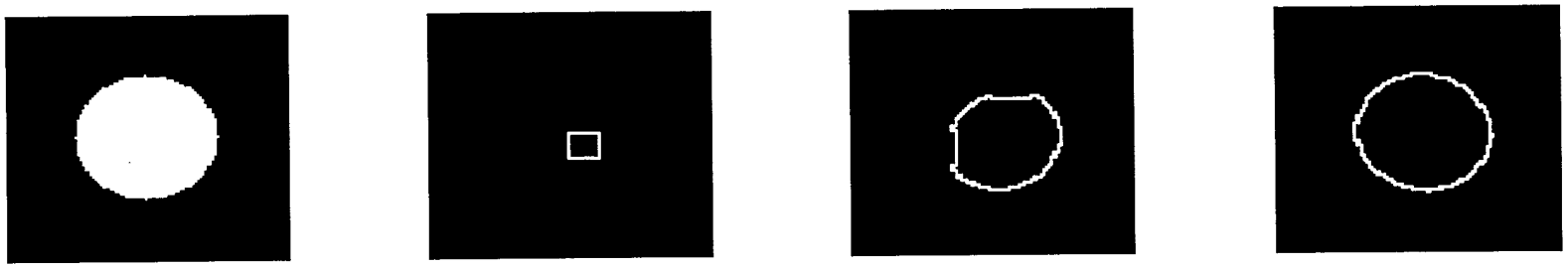
Figure 3: Flowchart of the narrow-band algorithm with re-initialization at every iteration.

3.2.2 Application to Synthetic Data: Two Circles

For this test the image contains two circles of different radii, Figure 5(a). The initial contour is a square that encloses both circles, Figure 5(b). In this example, the zero set contracts uniformly until it reaches the boundary of the circles. The zero set continues to contract around both circles until the contour meets up with itself between the circles, Figure 5(c). At the point, the zero set changes topology and splits into two contours. The contours continue to contract until they reach the boundary of each of the circles, Figure 5(d).

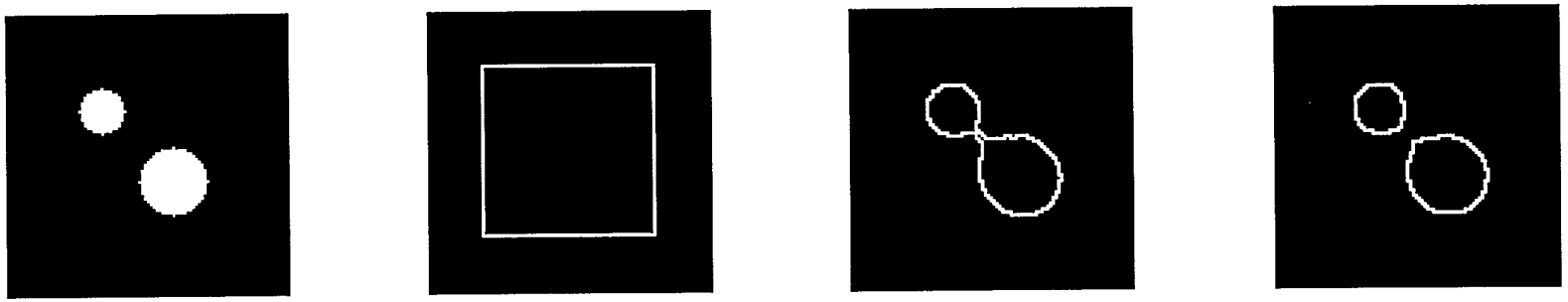
3.2.3 Application to MRI Medical Imagery: Brain

This test image is a cross-section of an MRI brain scan, Figure 6(a). The initial contour is a thin rectangle located within the gray matter, Figure 6(b). Figure 6(c) shows the results obtained using the distance transform operator method for re-initialization. Note that the sharp indentation towards the bottom of the shape has been smoothed. In contrast, Figure 6(d) shows the results obtained using the Fast Marching re-initialization, which captures accurately the indentation.



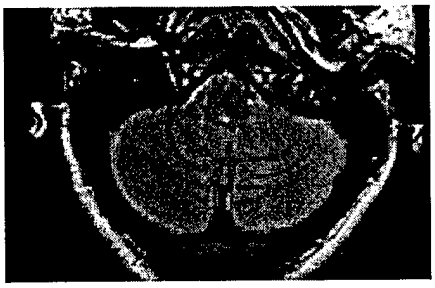
(a) Original image. (b) Initial contour. (c) Iteration 10. (d) Iteration 18.

Figure 4: (a) Image of a uniform intensity circle. (b) The initial contour $\gamma(0)$ is a small square located within the circle, below and to the right of the center of the circle. (c) The zero set at iteration 10. The boundary has expanded uniformly to the boundary of the circle. (d) The zero set, at iteration 18. The contour already at the boundary of the circle stops expanding while the rest of the contour continues to grow until it reaches the rest of the circle boundary.

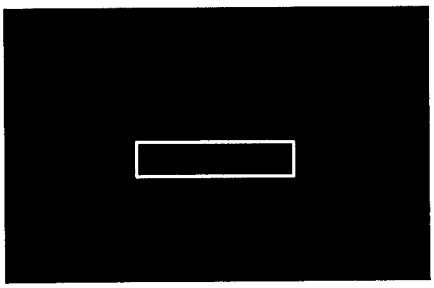


(a) Original image. (b) Initial contour. (c) Iteration 15. (d) Iteration 21.

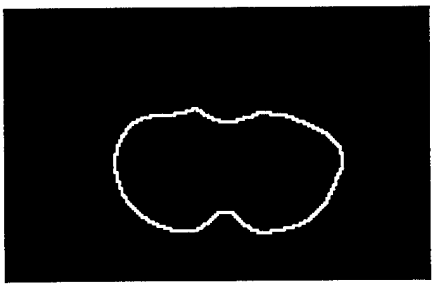
Figure 5: (a) Image of two circles of different radii. (b) The initial contour $\gamma(0)$ is a square that encloses both circles. (c) The zero set at iteration 15. The zero set contracts uniformly until it reaches the boundary of the circles. It continues to contract around both circles until the contour meets up with itself between the circles. (d) The zero set at iteration 21. The zero set changes topology and splits into two contours. The contours continue to contract until they reach the boundary of each of the circles.



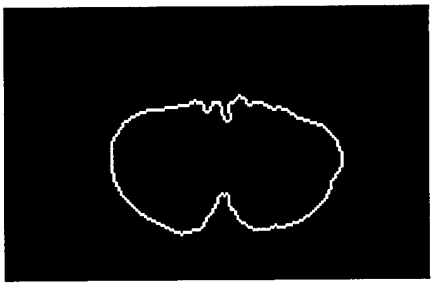
(a) Original image.



(b) Initial contour.



(c) Result when using polygonal approximation and distance transform.



(d) Result when using the Fast Marching re-initialization.

Figure 6: (a) MRI image of a brain. (b) The initial contour $\gamma(0)$ is a square inside the gray matter. (c) The zero set obtained using the polygonal approximation and distance transform. Note that perturbation of the zero set at each re-initialization has caused the indentation at the bottom of the image to be smoothed. (d) The zero set obtained using the Fast Marching re-initialization.

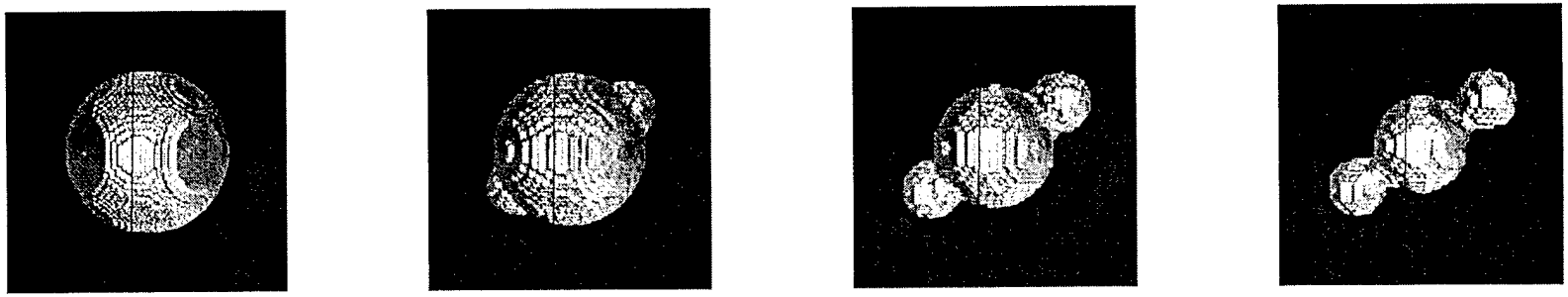
3.2.4 Application to Three-Dimensional Synthetic Data: Two Spheres

In this example, we wish to segment two spheres in a three-dimensional data set with dimensions $60 \times 60 \times 60$. The origin, $(1, 1, 1)$, is placed on the lower left-hand corner. Each of the spheres has a radius of 5 pixels and are located at positions $(20, 20, 20)$ and $(40, 40, 40)$. Figure 7 shows the zero set at different times during the propagation process. The final images were constructed by building a mesh over the surface using the method marching cubes provided in a public domain software called *isosurf*. We have rendered the mesh using another public domain program called *Geomview*. These program are available at:

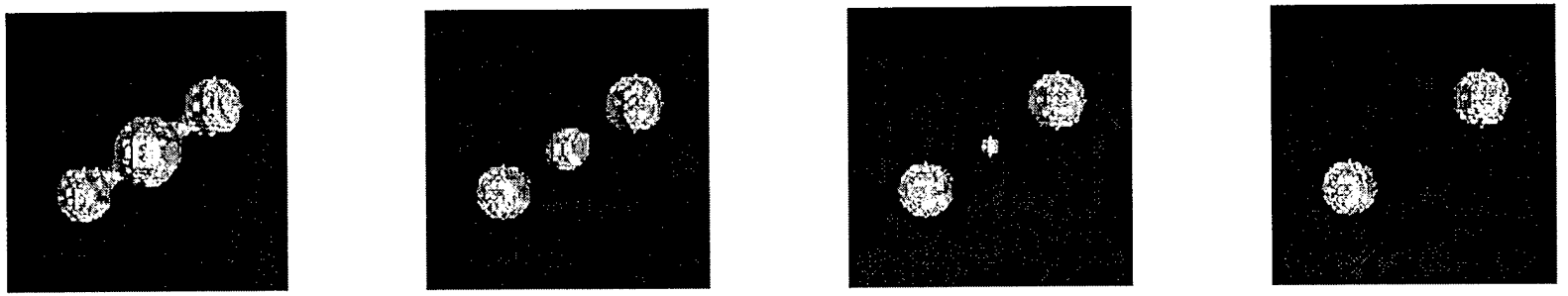
`http://svr-www.eng.cam.ac.uk/~gmt11/software/isosurf/isosurf.html`

http://www.geom.umn.edu/software/geomview/

The initial contour, Figure 7(a), is a sphere of radius 26 pixels located at $(30, 30, 30)$. As the propagation process continues, the zero set contracts uniformly until it reaches the boundary of the spheres, Figure 7(b). The zero set continues to contract around both spheres until the surface meets itself between the spheres, Figure 7(e). At this point, the zero set changes topology and splits to form three spheres, Figure 7(f). Further contraction of the two outer spheres is stopped by image speed k_I, while the center sphere continues to contract to a point, Figure 7(h).



(a) Initial contour. (b) Iteration 15. (c) Iteration 25. (d) Iteration 30.



(e) Iteration 35. (f) Iteration 40. (g) Iteration 45. (h) Iteration 54.

Figure 7: This is a demonstration of the basic level-set segmentation algorithm on a synthetic three-dimensional image containing two spheres. The dimensions of the data set are $60 \times 60 \times 60$. The origin, $(1, 1, 1)$, is placed in the lower left-hand corner. The two spheres have a radius of 5 pixels and are located at positions $(20, 20, 20)$ and $(40, 40, 40)$. (a) The initial contour $\gamma(0)$ is a sphere of radius 26, centered at $(30, 30, 30)$, that encloses both of spheres to be recovered. (b) The zero set at iteration 15. The surface contracts uniformly until it reaches the boundary of the spheres. (c) The zero set at iteration 25. The parts of the zero set around the spheres boundary stops contracting while the rest of the surface continues to shrink. (d) The zero set at iteration 30. (e) The zero set at iteration 35. The zero set continues to contract around the spheres until it meets itself. (f) The zero set at iteration 40. The zero set splits into three surfaces. (g) The zero set at iteration 45. Contraction of the two outer surfaces stops due to the boundaries of the spheres, while the center surface continues to contract. (h) The zero set at iteration 54. The converged solution.

3.3 Improvements

Simple segmentation methods may fail in images where there are indistinct or missing boundaries. In this section, we explore two incremental improvements to address this problem: the incorporation of *a priori* edge information, and the use of geodesic active contours.

3.3.1 Incorporating *A Priori* Edge Information

An example of indistinct boundaries is shown in the ultrasound image of a prostate in Figure 9(a). By calculating k_I, the image speed term, based solely on image gradient information, Figure 9(b), it is difficult to distinguish between the prostate and the background. For this image, the level-set algorithm fails to stop at the shape boundary.

To remedy this, we have used *a priori* edge information obtained from edge detection preprocessing (see the flowchart in Figure 8). The edge image in Figure 9(c) is obtained by using anisotropic diffusion smoothing [13], followed by Canny edge detection [14]. Figure 9(d) is the image speed term (k_I) determined using the thresholded square distance to the closest edge pixel. Using this k_I, a good outline of the prostate can be obtained (Figure 10).

Note that while the original image has dimensions 512×512, the Fast Marching extension method allows us to obtain accurate results using sparser calculation grids. The results shown in Figure 10 were obtained using a calculation grid of 256×256 and 128×128 points. This reduction in grid size provides a large saving in computation time.

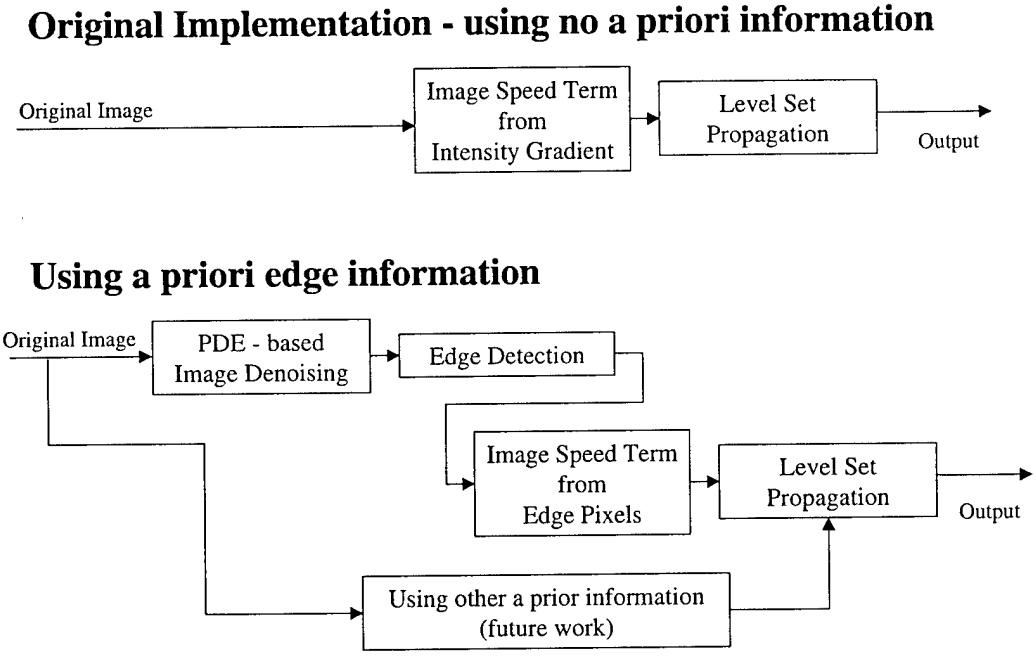
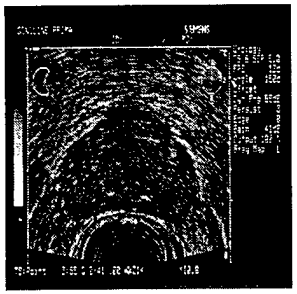
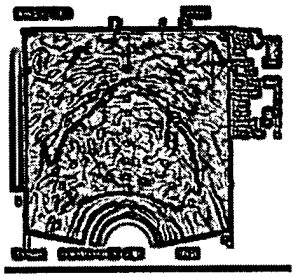


Figure 8: Flow charts of the level-set segmentation process with and without *a priori* information.



(a) Original image.

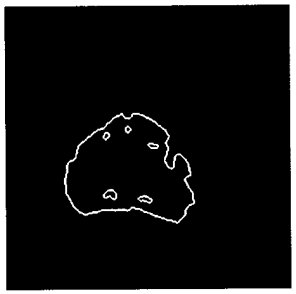


(b) Image speed term, k_I, from the image gradient.

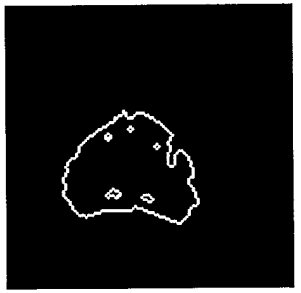
(c) Edge pixels from preprocessing; anisotropic diffusion followed by Canny edge detection.

(d) Image speed term, k_I, from edge pixels found in (c).

Figure 9: (a) This ultrasound image of a prostate has indistinct boundaries. (b) Image speed term k_I obtained using the image gradient. (c) Edge image obtained from anisotropic diffusion followed by Canny edge detection [14]. (d) Image speed term k_I calculated using edge information.



(a) One-quarter resolution (256×256).



(b) One-sixteenth resolution (128×128).

Figure 10: The zero set obtained for the prostate data using Figure 9(d) as the image speed term. Note that the original data is 512×512 while calculation of the zero set was performed on grid sizes of (a) 256×256 and (b) 128×128.

3.3.2 Geodesic Active Contours

A partial solution to the indistinct boundary problem can be found in [15, 16], where the propagation speed function is given as:

$$v_n = -\nabla \cdot \left(k_I \frac{\nabla \varphi}{\|\nabla \varphi\|} \right) = -\frac{\nabla\ k_I \cdot \nabla\ \varphi}{\|\nabla \varphi\|} - k_I \kappa. \tag{5}$$

The second term in Eq. (5) acts as smoothing term to reduce the length of the contour. The first term acts like a doublet which attracts the contour to the potential wells in k_I. This technique is known as a geodesic active contour in [15], and conformal length shortening in [16].

An additional constant inflation (deflation) term is also suggested to speed up convergence [15, 16]:

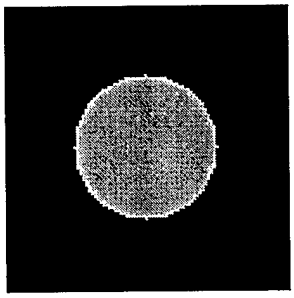
$$v_n = -\frac{\nabla\ k_I \cdot \nabla\ \varphi}{\|\nabla \varphi\|} + k_I (c - \kappa). \tag{6}$$

Note that the image speed terms in Eq. (6) have meaning only on the front $\Gamma(t)$. Hence at each iteration of the algorithm we need to extend all components of ∇k_I and well as k_I. This additional extension requires only a small amount of extra work in our Fast Marching re-initialization method.

The major differences between using the speed functions in Eq. (3) and Eq. (5) are illustrated in the following simple example that uses a synthetic test image consisting of a uniform intensity circle on a black background. In this example, Figure 11, we use an initial contour $\gamma(0)$ of a circle that is the same size as the one we wish to recover, but we shift it to the upper left-hand corner by a few pixels.

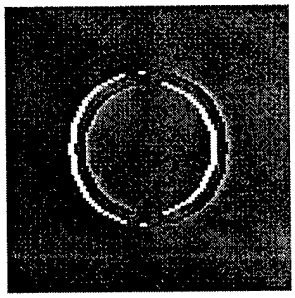
The results obtained using the original speed function, Eq. (3), are shown in Figure 12, which illustrates the behavior of the old speed function once the shape boundary has been passed. Using a constant inflation term in Figure 12(b), the contour expands to meet the circle boundary except in the top-right region where it leaks into the background. Using a constant deflation term in Figure 12(c), the contour contracts to meet the circle boundary except in the bottom-left region where it shrinks inside the shape.

For the same initial contour, Figure 13 shows the results obtained using the new speed function, Eq. (5), which allows movement both inwards and outwards to try follow the boundary of the circle. Note that the bottom-right boundary is slow to converge, Figures 13(b) and 13(c). This is partially due to the curvature term acting to pull the contour away from the boundary. In this example, convergence does not occur until the 57th iteration. The rate of convergence can be improved by adding a constant inflation component to the speed function as in Eq. (6), where $c > 0$. With c set to 0.05 for the same test, convergence was reached in 37 iterations, a significant improvement when compared to 57 iterations when $c = 0$.

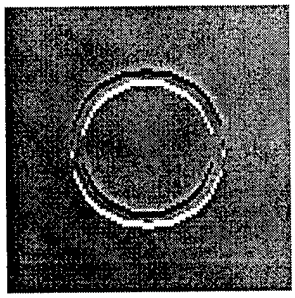


(a) Test image.

(b) Edge potential k_I.

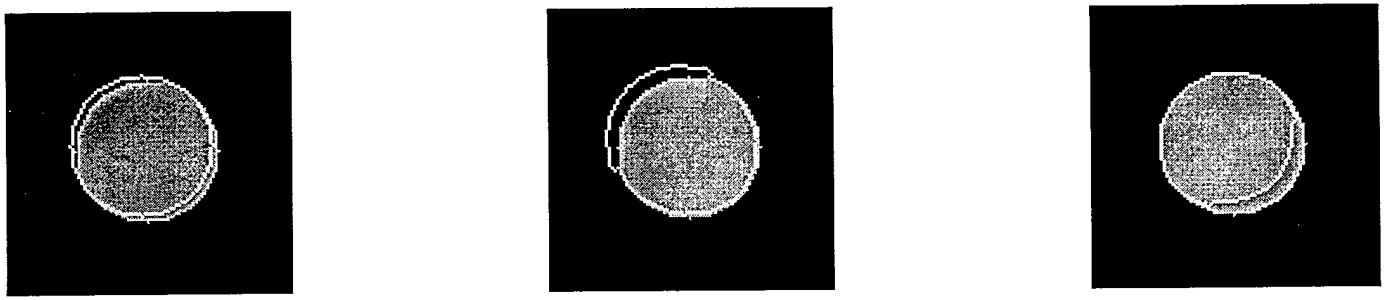


(c) Column derivative $\partial k_I/\partial x$.



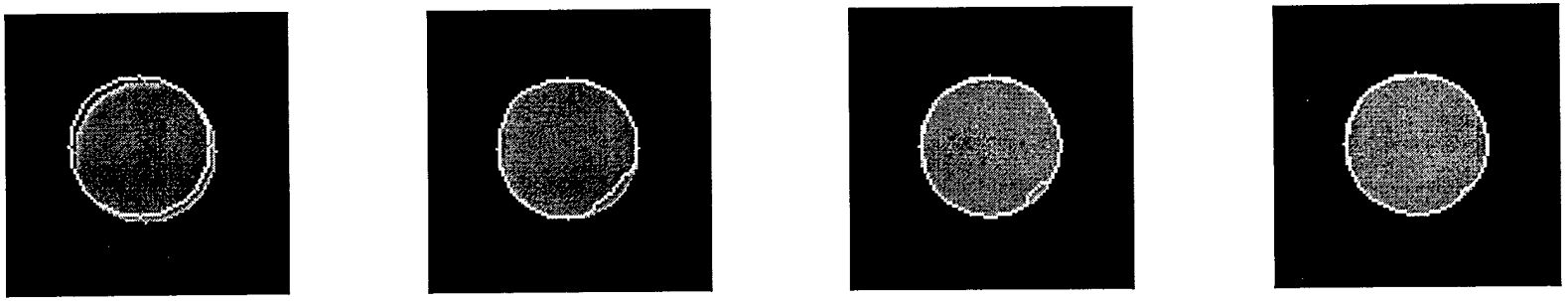
(d) Row derivative $\partial k_I/\partial y$.

Figure 11: (a) A simple test image consisting of a uniform intensity circle on a black background. (b) The image of the edge potential k_I calculated using Eq. (4). The grayscale values range from black (0) to white (1). Images of the (c) column and (d) row derivatives of k_I. The grayscale values range from black (-0.5) to white (0.5).



(a) Initial contour. (b) Original method: inflation. (c) Original method: deflation.

Figure 12: Segmentation using speed function Eq. (3). (a) The initial contour $\gamma(0)$ is a circle that is the same size as the circle we wish to recover but offset by a few pixels. (b) Using a constant inflation term, the contour expands to meet the circle boundary, except in the top-right region where it "leaks" into the background. (c) Using a constant deflation term, the contour contracts to meet the circle boundary, except in the bottom-left region where it "shrinks" inside the circle.



(a) Initial contour. (b) Iteration 20. (c) Iteration 40. (d) Iteration 57.

Figure 13: Segmentation using speed function Eq. (5). (a) The initial contour $\gamma(0)$ is a circle that is the same size as the circle we wish to recover but offset by few pixels. (b) The zero set at iteration 20. The contour shifts to meet the boundary, except in the bottom-right region. (c) The zero set at iteration 40. The contour converges slowly near the bottom-right region. (d) The zero set at iteration 57. The converged solution.

3.4 Segmentation Using Region Statistics

In Section 3.2, we observed that edge-based segmentation methods fail when there is insufficient intensity change at the object boundaries. Two incremental improvements of incorporating *a priori* edge information and using geodesic active contours were discussed in the last section. Another solution is to use region-based features. The use of region-based features avoids the need to calculate image gradients that are extremely sensitive to noise, and provides greater robustness to the effects of weak or blurred edges.

For example, the region-based method in [6] assumes that an image consists of a finite number of regions, where each region is delineated by a predetermined set of features or statistics (e. g. means, variances, and textures). In [6], an initial curve $\Gamma(0)$ is evolved to maximize the difference between the value of the feature(s) inside the curve from the value on the outside. The curve evolution is implemented in a level-set framework, where $\Gamma(t)$ is embedded as the zero level set of the level-set function φ.

3.5 Bimodal Images Segmentation

Suppose we wish to segment a bimodal image, i. e. an image that consists of two regions, possibly multiply connected, delineated by a particular feature or statistic. By multiply connected, we mean that each region may consist of many disconnected parts; for example, this page consists of two regions, one white and one black, where each region has many parts that are disconnected, such as letters. The approach of [6] is to minimize the following energy functional:

$$E(\Gamma) = -\frac{1}{2}(u-v)^2 \tag{7}$$

where u is the value of the feature for the region inside the curve Γ and v is the value for the region outside Γ. At each iteration, Γ is evolved in a steepest descent manner such that:

$$\frac{d\Gamma}{dt} = (u-v)(\nabla u - \nabla v). \tag{8}$$

If u and v represent the mean intensities, then we have:

$$\frac{d\Gamma}{dt} = (u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right)\vec{N} \tag{9}$$

where I is the image intensity, A_u, A_v are the area of the regions inside and outside Γ and $\vec{N}$ denotes the outward unit normal of Γ. This evolution can be translated to a level-set evolution such that:

$$\frac{\partial\varphi}{\partial t} - (u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right)|\nabla\varphi| = 0. \tag{10}$$

The use of a level-set framework allows complex shapes to be segmented and topological changes such as merging and splitting to be handled implicitly. Since this approach uses

region-based statistics, the initial curve $\Gamma(0)$ can be placed anywhere on the image domain (e. g. the initial curve may overlap object boundaries).

To add robustness to noise, a curvature-based term can added to second term of Eq. (10):

$$\frac{\partial \varphi}{\partial t} - \left\{(u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right) - \alpha\kappa\right\}|\nabla\varphi| = 0. \tag{11}$$

Note that this approach does not require the extension of the image speed in contrast to other methods described in this report. However, in our experiments we have found that the level sets may bunch up making the curvature calculation noisy. Therefore, regular re-initialization of the level-set function φ to the signed distance function (from the zero level set) is recommended.

As described in [6], more than one feature can be used to segment a bimodal image. The approach is then to maximally separate the two feature vectors using some appropriate distance measure.

Extension to trimodal images is also described in [6]. To segment an image into three regions/classes at least two features are needed. Two (coupled) level sets are evolved to maximize the area of the triangle formed by the feature vectors. In general, to segment an image into N classes, at least $N-1$ features and $N-1$ level sets are require.

3.5.1 Implementation Issues

When dealing with an image, updating the level set at every image grid point can be computationally intensive. To improve efficiency, the narrow-band scheme described in Section 3.1 can also be applied here. In this scheme, the level-set function is updated only at a small set of points within a band of width δ around the zero level set of φ^n.

In our narrow-band implementation, φ^n is re-initialized to the signed distance function at every iteration. This strategy means that the zero level set will always remain in the middle of the narrow-band. Therefore collision detection with the narrow-band boundary is not required.

In the narrow-band scheme, the choice of the time step Δt is important. If Δt is too small, the algorithm will be slow to converge; if Δt is too large, the zero level set may move out of the narrow-band. A good choice of Δt is the one that keeps things moving if a large proportion of the zero level set is in a region of approximately constant intensity. In our experiment, we found a satisfactory choice of Δt is:

$$\Delta t = \frac{\min\left(A_u, A_v\right)}{(u-v)^2}.$$

Note that in images where the region we wish to segment is multiply connected, the narrow-band scheme may miss regions which are never within δ of the zero level set.

3.5.2 Application to Microscopy Imagery: Red Blood Cells

Figure 14(a) shows a microscopy image of red blood cells. The number and size of red blood cells (RBC) in a sample can be used to diagnose anemia and other diseases. In this example, we apply the region statistics method [6] to segment the RBCs from the background.

The initial curve is a circle located in the top-right region of the image, Figure 14(a). The curve evolves to separate maximally the mean intensity inside the curve and the mean intensity outside the curve. Note that the evolving curve can move both in and out. For example, consider the RBCs near the initial curve. In Figure 14(b), it can be seen that the zero level set has move inwards to locate the RBCs inside the initial curve as well as locating RBCs outside the initial curve.

Figure 14(c) shows the zero level set after 800 iterations. This example illustrates the major advantage of using level sets: topological changes are handled automatically allowing all RBCs to be segmented. The level-set framework also makes it simple to extract the RBCs for post-processing. The RBCs are regions where $\varphi > 0$. These are shown as dark grey regions in Figure 14(d). The dark gray value is the steady state mean for regions outside Γ, while the light gray is the steady state mean for regions inside Γ.

Individual RBCs can be separated using connected component labeling [12]. In Figure 15(a), each individual RBCs is assigned a different gray value. Note that we have also eliminated all RBCs that are not wholly within the image.

From Figure 15(a), it can be seen that except for two cases, the segmentation method was able to separate RBCs which were very close together in the original image. Using the image in Figure 15(a), simple statistics can be calculated, Figure 16. The image in Figure 15(b) shows the smallest and largest cell detected. The small "cell" is actually a platelet and the large "cell" is really two cells close together.

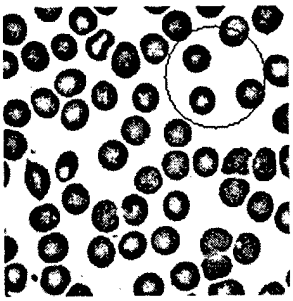
3.5.3 Application to Prostate Boundary Segmentation

Measuring the size of the prostate and identifying accurately the prostate boundary is an important part in the diagnosis and treatment of prostate cancer. Three-dimensional transrectal ultrasound (TRUS) imaging is routinely used for prostate examination. Ultrasound images are typically low contrast and images of soft tissues (e. g. the prostate) suffer from blurred boundaries. In this section, we apply the region statistics method of [6] to the problem of prostate boundary segmentation from TRUS images.

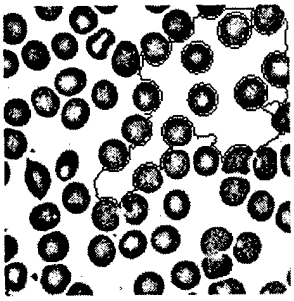
Results using two-dimensional segmentation are shown in Figures 17 and 18. Figure 17(a) shows one of the center slices of a TRUS image set. The initial curve is a large circle centered around the center of the prostate. The curve evolves in a manner that tries to maximize the difference between the mean intensity inside the curve and the mean outside the curve, Figure 17(b).

Figure 17(d) shows the zero level set after 200 iterations. It can be seen than the general shape of the prostate has been captured. The ragged edges on the bottom-left are due to the blurred boundaries in the original image. These edges can be smoothed out in post-

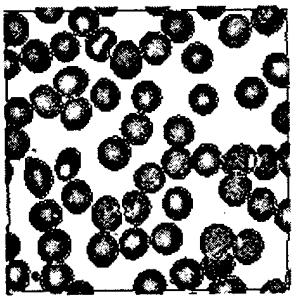
processing. However, too much smoothing may lead to the underestimation of the prostate size. The size of the prostate can be estimated by counting the number of pixels where $\varphi < 0$. For this image, the prostate is approximately $37,298$ pixels in area.



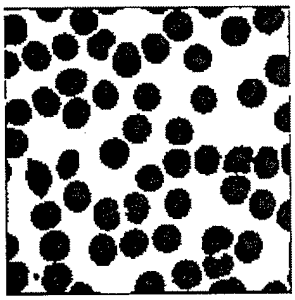
(a) Initial contour.



(b) Zero set, iteration 400.

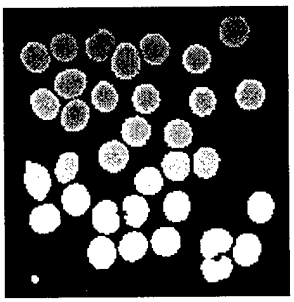


(c) Zero set, iteration 800.

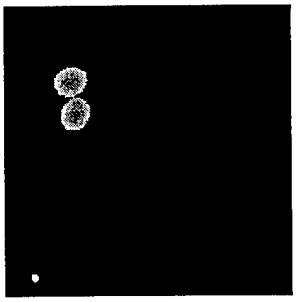


(d) Segmented red blood cells.

Figure 14: Red blood cells segmentation from a 169×169 microscopy image. (a) The initial contour $\Gamma(0)$ is a circle near the top-right corner of the image. (b) The zero level set at iteration 25. The contour has evolved to maximally separate the mean intensity inside the contour from the mean intensity on the outside. Note that the contour has split several times to segment red blood cells individually. (c) The zero level set at iteration 800. All the red blood cells have been isolated. (d) The reconstructed image. The dark gray region represents the region inside the contour of image (c). The dark gray value is the mean intensity of all pixels inside the contour, while the light gray value is the mean intensity of all pixels outside of the contour.



(a) Connected component labeling.



(b) Smallest/largest segmented cell.

Figure 15: Connected component analysis of Figure 14(d). (a) The result from connected component labeling. Each disjoint region is represented by a different gray value. Note that all regions not wholly within the image has been removed. (b) The smallest (white) and largest (gray) segmented regions. The small region is actually a platelet. The large region is really two red blood cells that are close together in the original image.

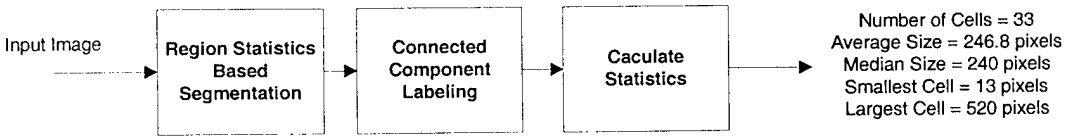
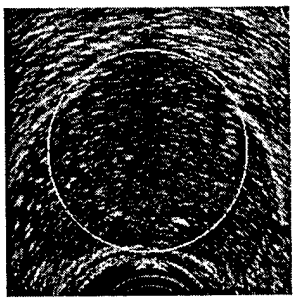
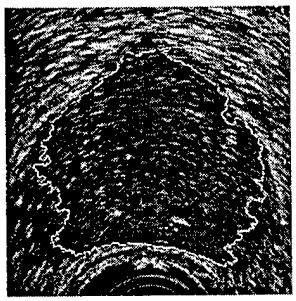


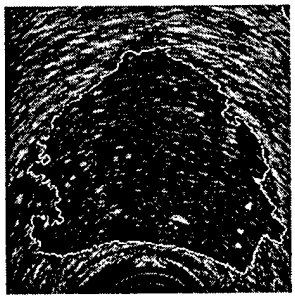
Figure 16: Flowchart of the red blood cell counting process.



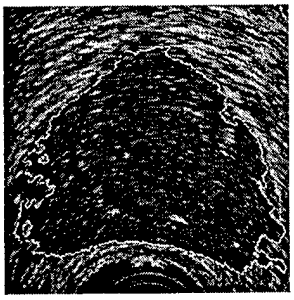
(a) Initial contour.



(b) Zero set, iteration 25.

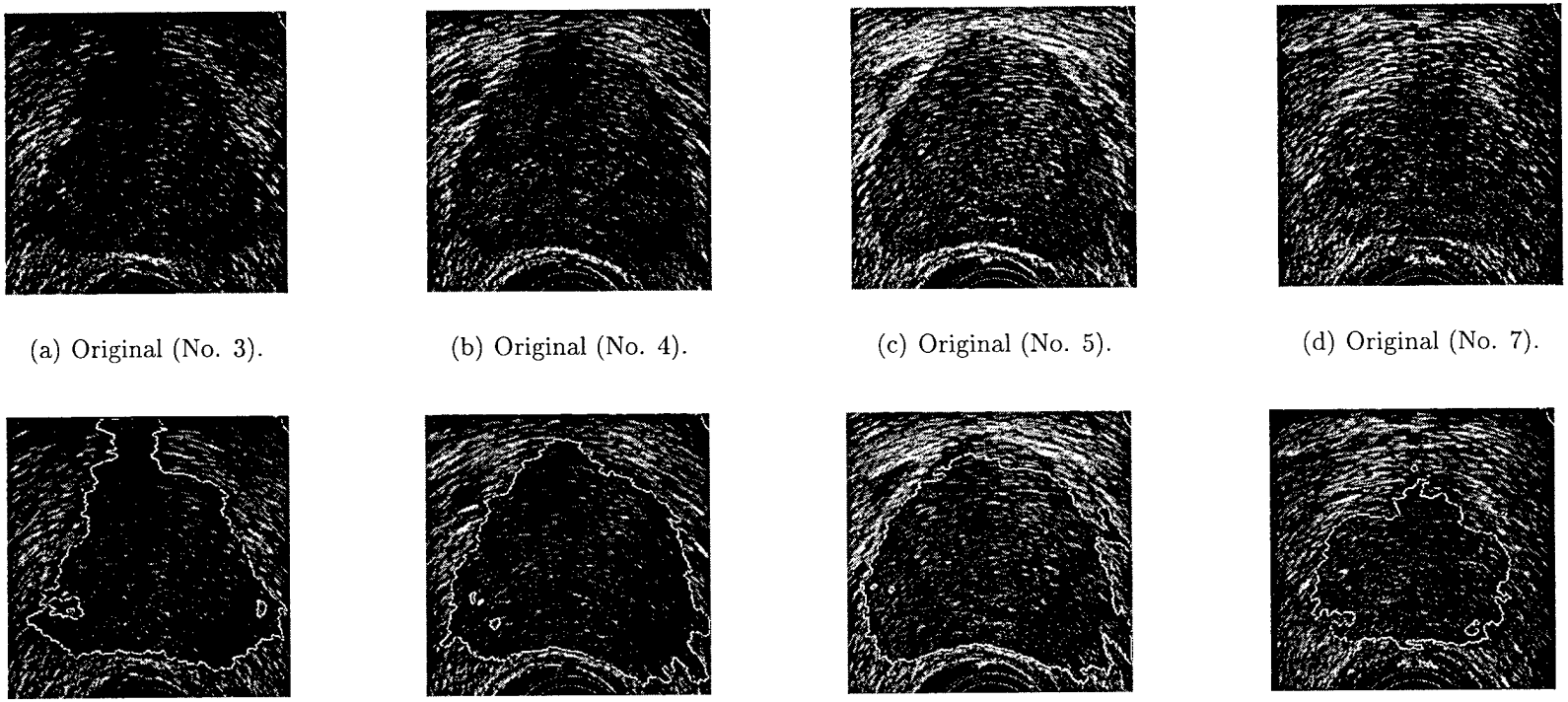


(c) Zero set, iteration 50.



(d) Zero set, iteration 200.

Figure 17: Prostate segmentation from a 280×280 ultrasound image. This is the sixth slice of an eight slice trans-rectal ultrasound image set. (a) The initial contour is a large circle centered near the center of the prostate. (b) The zero set at iteration 25. The contour has evolved to maximally separate the mean intensity inside the contour from the mean intensity outside of the contour. (c) The zero set at iteration 50. (d) The zero set at iteration 200. The ragged edges at the bottom-left are due to blurred edges in the original image.



(a) Original (No. 3). (b) Original (No. 4). (c) Original (No. 5). (d) Original (No. 7).

(e) Result (No. 3). (f) Result (No. 4). (g) Result (No. 5). (h) Result (No. 7).

Figure 18: Segmentation results for slice 3, 4, 5, and 7 in the trans-rectal ultrasound image set. (a) - (d) The original 280×280 images. (e) - (h) Results after 200 iteration of the region statistics algorithm described in Section 3.4.

Segmentation results for other images in the TRUS set are shown in Figure 18:

- **Slice 3** (Figure 18(e)) - interference from an artifact has cause the segmentation to also capture the dark region at the top.
- **Slice 4** (Figure 18(f)) - most of the prostate has been captured. There is slight underestimation on the top-right. This is due to the region being lighter in color.
- **Slice 5** (Figure 18(g)) - again most of the prostate has been captured with underestimation along the top boundary. The dark fringes on the right have caused "tails" in the segmentation results.
- **Slice 7** (Figure 18(h)) - the prostate is barely visible in the original image. The segmentation algorithm was able to capture most of the prostate with some underestimation at the top.

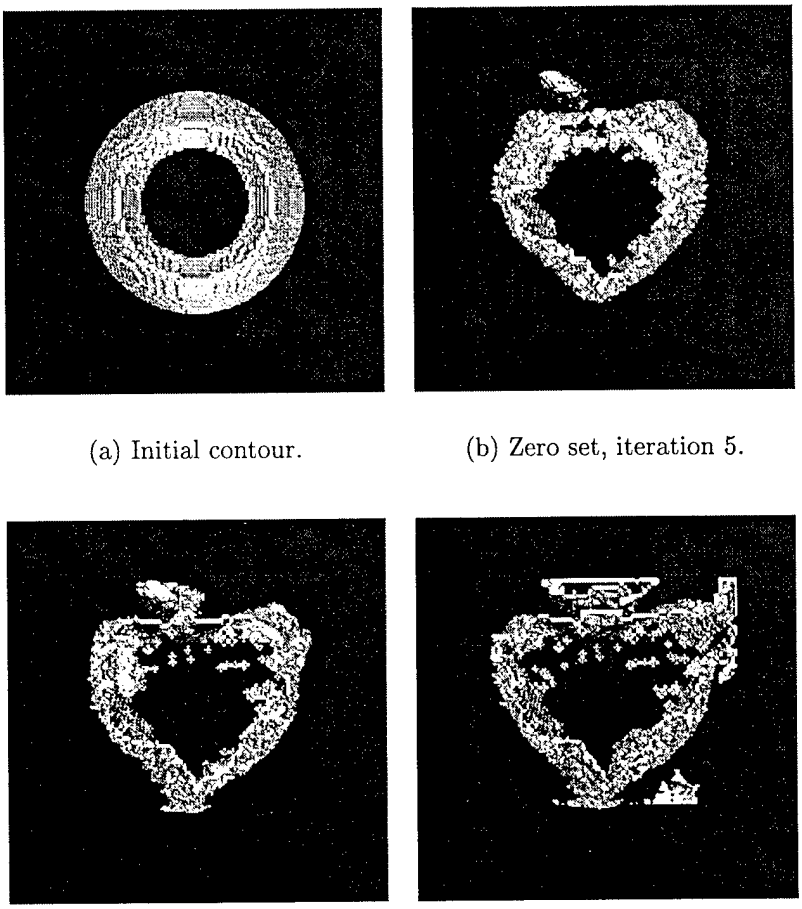
Theoretically, all images in a TRUS set can be used together to reconstruct the prostate in three dimensions. In practice, this is a difficult problem as there are typically only eight slices in a TRUS image set, therefore resolution in the third dimension is poor. To address this problem, intermediate slices can be interpolated from the original images. However, the images needs to be downsampled to reduce the huge discrepancy in the grid spacing between the dimensions.

In our experiment, we have linearly interpolated ten intermediate slice between each pair of original images. The images were also downsampled to give a dataset of size $70 \times 70 \times 45$. Note that we have use only slices 3 to 7 of the TRUS set, as the prostate is barely visible in the start and end slices. We then apply the region-based segmentation method of [6].

The initial surface $\Gamma(0)$ is a partial sphere, Figure 19(a). The surface evolves to maximally separate the mean intensity inside from the mean intensity outside, Figures 19(b)-19(d). The object at the top is due the dark semi-circle which appears in all the images.

From Figure 19(d) some of the major characteristics of a prostate are visible. The prostate is approximately as long as it is wide and roughly conical in shape. The bumps of the two lateral lobes at the top of the prostate can be seen in the reconstruction.

Figure 20 shows the contours of the surface in Figure 19(d) overlaid on the original downsampled slices. From these images it can be seen most of the problem with the 3D segmentation are due to blurry nature of slices 3 and 7. Although far from ideal these results shows promise and we propose to investigate the use of this region-based segmentation method as part of Phase II work.



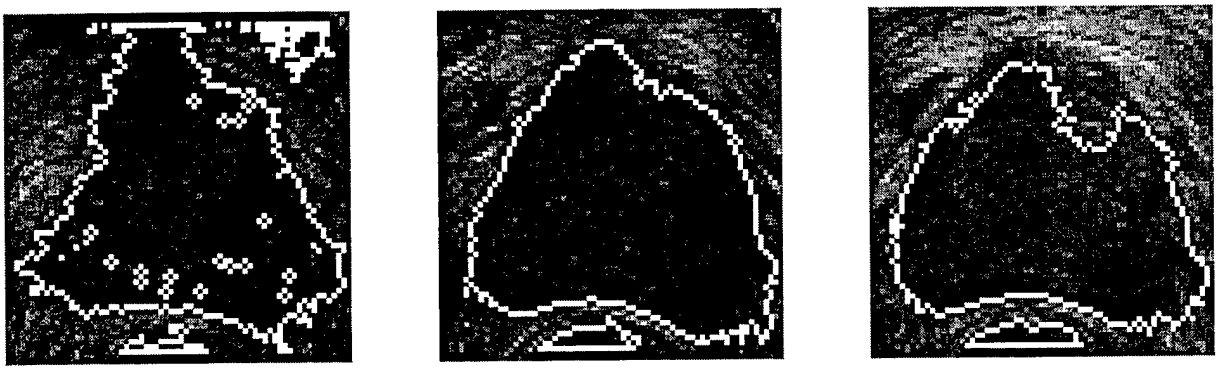
(a) Initial contour.

(b) Zero set, iteration 5.

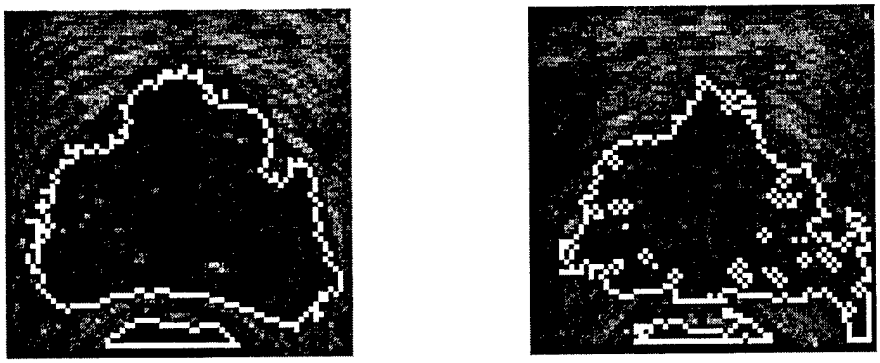
(c) Zero set, iteration 20.

(d) Zero set, iteration 100.

Figure 19: Three-dimensional segmentation of the prostate from a set of trans-rectal ultrasound images. (a) The initial surface is a partial sphere centered near the center of the prostate. (b) The zero set at iteration 5. The surface evolves to separate the mean intensity inside the surface from the mean intensity on the outside. (c) The zero set at iteration 20. (d) The zero set at iteration 100. The object at the top is due to the dark semi-circle which appears at the bottom of all images. The ragged edges at the front are due to the blurriness of the image in slice 7, Figure 18(d).



(a) Slice 3. (b) Slice 4. (c) Slice 5.



(d) Slice 6. (e) Slice 7.

Figure 20: Three-dimensional segmentation of the prostate from a set of trans-rectal ultrasound images. Each image shows the contour of surface in Figure 19(d) onto each of the original downsampled slices.

4 Level Sets for Deformable Image Registration

Investigation of algorithms for PDE-based deformable image registration [17, 18] was the focus of the Phase I option. These algorithms have a unique approach to non-rigid image registration in their use of a novel curve evolution approach. In the framework of these two algorithms, an image volume is formulated in terms of a level set, where voxels of a particular intensity make up one level set, that is, an image volume is viewed as a set of iso-intensity contours.

The main idea of [17] is to evolve one image into another by letting the level-sets evolve in the normal direction until the first image becomes the second image. This evolution is described in terms of a non-linear hyperbolic PDE:

$$\frac{\partial \vec{V}(\vec{x})}{\partial t} = \left(I_2(\vec{x}) - I(\vec{V}(\vec{x}))\right) \frac{\nabla I(\vec{V}(\vec{x}))}{\|\nabla I(\vec{V}(\vec{x}))\|} \tag{12}$$

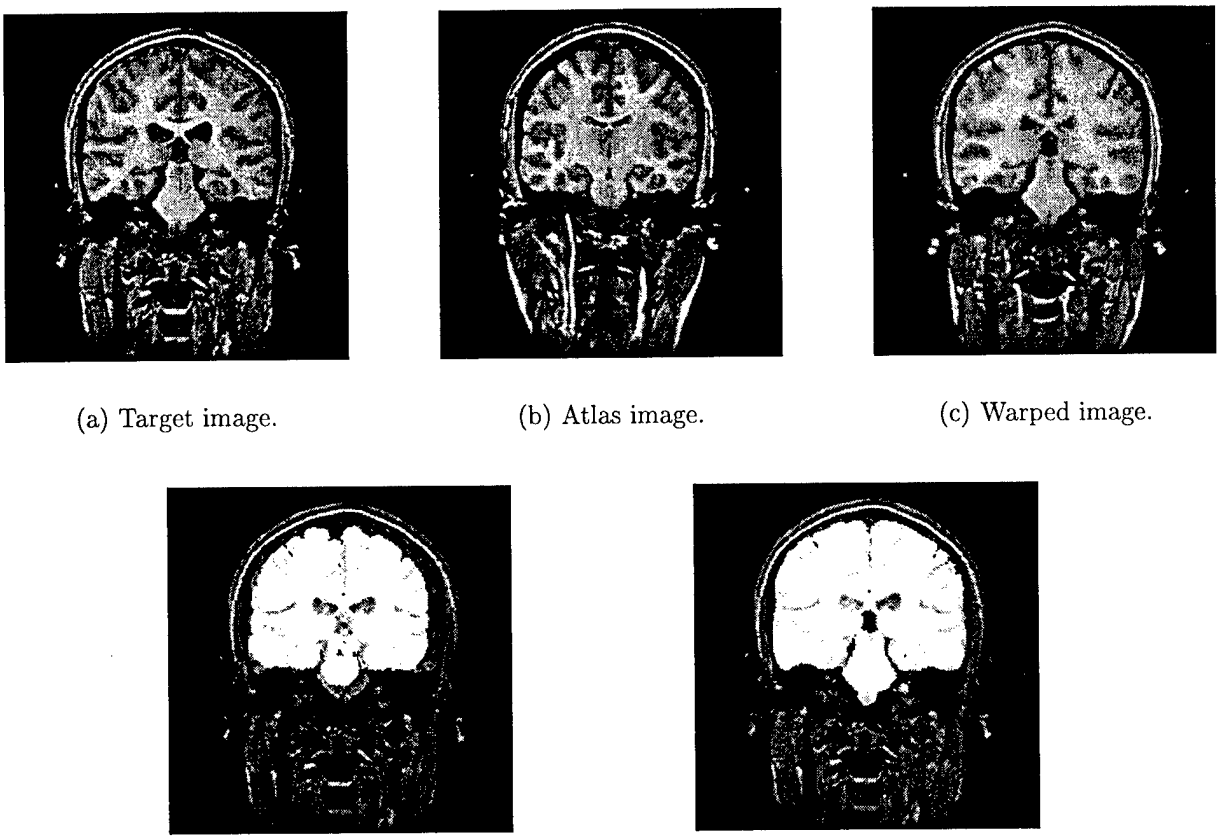
where $\vec{V}(\vec{x})$ is the displacement vector at grid point $\vec{x}$. $I_2(\vec{x})$ represents the image intensity of the reference or target image and $I(\vec{x})$ represents the intensity of the subject (warped) image.

The main idea of [18] is that a three-dimensional grid of "demons" deforms an image by pushing the contours in the normal direction. The orientation and magnitude of the displacement is derived from the instantaneous optical flow equation.

The main difference between the two algorithms is in their implementations. The Vemuri algorithm [17] uses derivatives from the test (moving) image, while the Thirion algorithm [18] uses derivatives from the reference (stable/target). Being intensity-based, both algorithms are sensitive to variation in the intensity between the two volumes. In our experiments we must first apply histogram-based intensity normalization to correct for the variation. We have implemented these algorithms in a multi-resolution framework. In addition, we have improved Thirion's algorithm [18] by using a bijective implementation, where both the forward and reverse deformations are calculated. At each iteration, corrections are made to make ensure that the forward and reverse transforms are compatible.

We have used our prototype for the medical imaging application of brain registration. The ability to perform (inter-patient) brain registration allows automatic atlas-based segmentation of structures of interest. In longitudinal studies of the same patient, deformable registration allows one to perform change detection so that one may track the process of a treatment. Figure 21 illustrates the registration procedure for brain volume segmentation on a sample dataset.

In our validation experiments [19] (carried out under a different project), an MRI brain atlas was warped (registered using Thirion's algorithm [18]) to a subject brain, and then the labels in the atlas were automatically warped as well. We tested this algorithm using an atlas and images of 20 normal brains (T1 volumes of MRI images) and compared our results to a three-dimensional mathematical morphology algorithm. Performance results showed over 90 percent accuracy. Our study found that the morphological algorithm had an average



(a) Target image.

(b) Atlas image.

(c) Warped image.

(d) Original brain mask on target image.

(e) Deformed brain mask on target image.

Figure 21: Atlas-based brain volume registration and segmentation procedure on a sample dataset. (a) The target image (the image to be segmented). (b) The atlas image (the image to be warped) for which the brain volume has been manually outlined. (c) The warped altas image matching the target image. (d) The pre-deformed atlas brain label image superimposed on the target. (e) The deformed atlas brain label image superimposed on the target.

similarity index of 0.918, while the Thirion/atlas-based algorithm had an average similarity index of 0.953. We also carried out an initial test of tracking a small brain structure, the caudate, in longitudinal data using the same non-rigid registration algorithm.

5 Segmentation Using Active Contours without Edges - LSS. Inc.

Recently, a new model [20] for active contours has been proposed that is based upon using a Mumford-Shah [8] like functional. In contrast to classical active contours models or snakes, the stopping term in this method has no dependence upon image gradient or edge strength ∇I. Rather, the stopping term is related to the segmentation of the image. This new method is advantageous in that objects can be detected that are not necessarily defined by sharp edges. Furthermore, this method lends itself readily to a level-set formulation, wherein the evolving curve Γ is equated to the zero level set of a function φ.

For a given image I, defined on a domain Ω and formed by two regions of piecewise constant intensities, the main idea behind this algorithm is to minimize an energy based segmentation. The key idea is the formulation of a fitting energy

$$\int_{\text{in}(\Gamma)} |I - c_1|^2 d\vec{x} + \int_{\text{out}(\Gamma)} |I - c_2|^2 d\vec{x}$$

where the constants c_1 and c_2 are averages of the image I inside and outside the boundary curve Γ, respectively. Clearly, the above term is minimized when the evolving curve Γ perfectly delineates the regions formed by the two intensities. Additional regularizing terms, dependent upon the length and area within Γ, may be added to the fitting energy. All terms may, in fact, be rewritten in terms of the level-set function φ and the resulting general fitting energy is:

$$\begin{aligned} F(\varphi, c_1, c_2) = & \\ & \mu \int_\Omega \delta(\varphi) |\nabla \varphi| d\vec{x} + \nu \int_\Omega H(\varphi) d\vec{x} \\ & + \int_\Omega |I - c_1|^2 H(\varphi) d\vec{x} + \int_\Omega |I - c_2|^2 (1 - H(\varphi)) d\vec{x} \end{aligned}$$

where H and δ are the Heaviside and Dirac functions and μ, ν are constant fixed parameters for the length and area terms, respectively. The associated Euler-Lagrange equations for φ can be derived by minimizing F with respect to φ and parameterizing the descent direction by an artificial time.

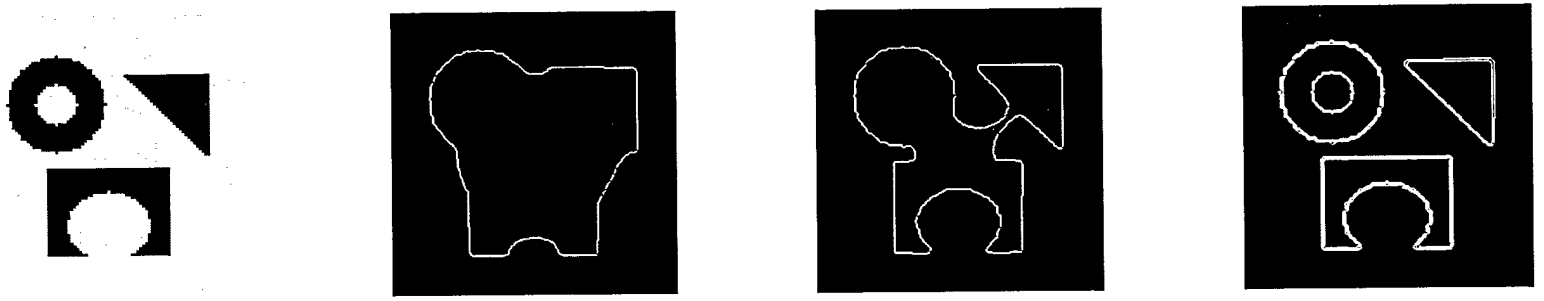
This "active contours without edges" model has the advantage of detecting interior contours automatically. An example of this is shown in Figure 22. The original image for these figures is shown in Figure 22(a). The three distinct objects within this image are successfully captured by the active contours without edges model. In Figure 23, it can be seen that the new model works well on the same image with the addition of noise.

The active contours without edges model can be easily extended to three dimensions. Some examples of this are shown in Figures 24 and 25. In Figure 24, the algorithm has been applied to an image that contains two balls. In Figure 25, the algorithm has been applied to an image a cube with a semi-spherical region cut out from it. In both examples, the initial zero level set of φ is a large sphere.

Figure 26 shows the segmentation of the prostate from a three-dimensional transrectal ultrasound image (TRUS). A two-dimension cross-sectional snapshot, Figure 26(a), shows the prostate. The original image was cropped to remove extraneous text.

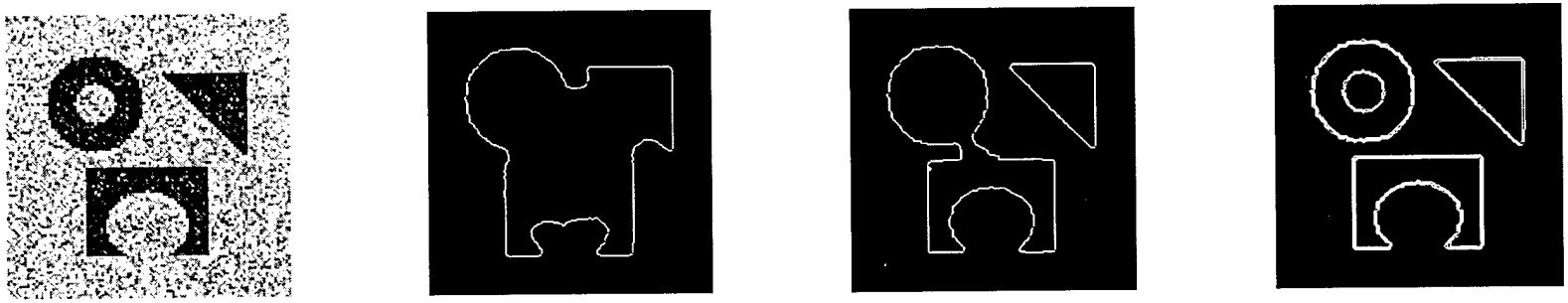
Figure 27 and Figure 28 show the segmentation of the brain fro a three-dimensional image. The grid size use for Figure 27 is $80 \times 80 \times 7$, while the grid size use for Figure 28 is $29 \times 29 \times 19$.

Again, the algorithm performs well, as many of the grooved details are captured by the (blue) evolving surface.



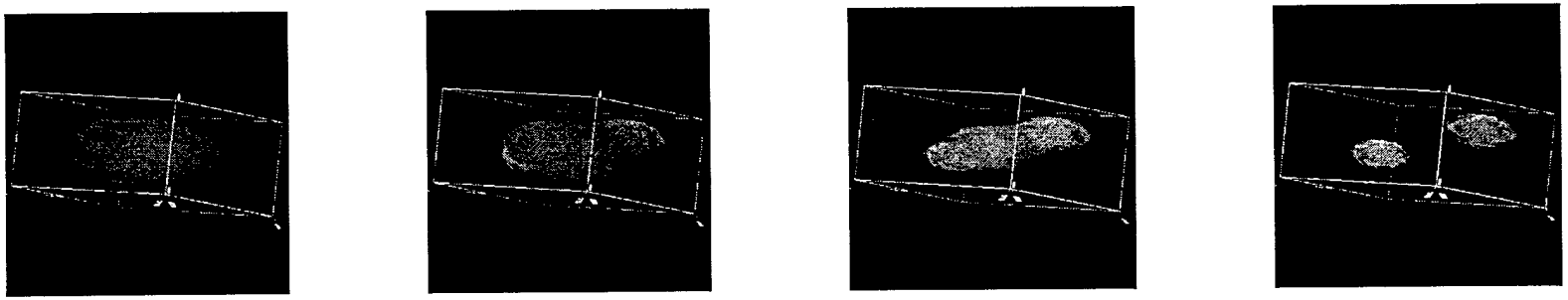
(a) Original image. (b) Iteration 300. (c) Iteration 700. (d) Iteration 1900.

Figure 22: Results from the active contours without edges model. (a) The original image contains three objects. The initial contour is a circle that encompasses all three objects. Segmentation results after (b) 300, (c) 700, and (d) 1900 iterations.



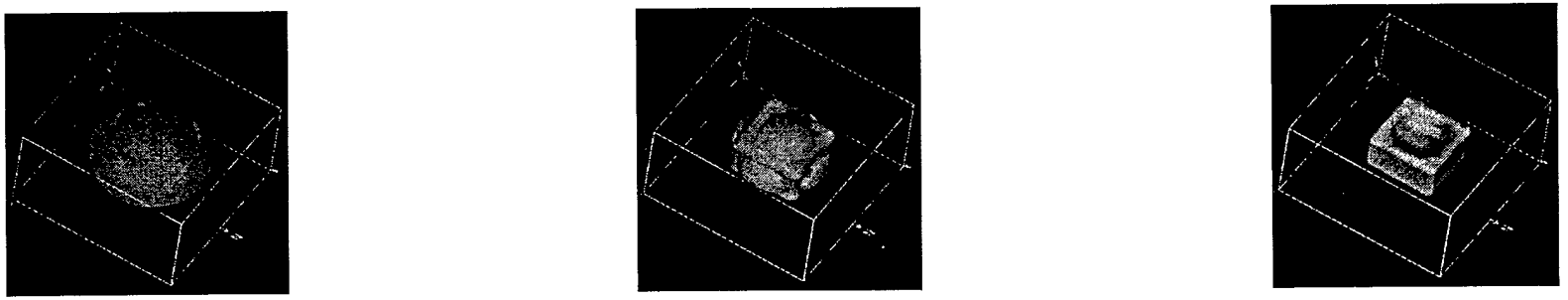
(a) Original image. (b) Iteration 600. (c) Iteration 1000. (d) Iteration 1900.

Figure 23: Results from the active contours without edges model. (a) The original noisy image contains three objects. The initial contour is a circle that encompasses all three objects. Segmentation results after (b) 600, (c) 1000, and (d) 1900 iterations.



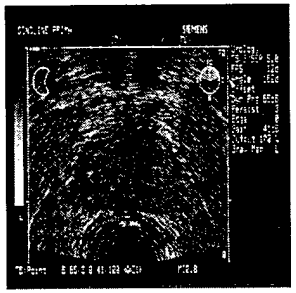
(a) Initial contour. (b) Iteration 400. (c) Iteration 500. (d) Iteration 600.

Figure 24: The original image contains two spheres. Results from the active contours without edges model: (a) initial contour, (b) 400, (c) 500, and (d) 600 iterations.

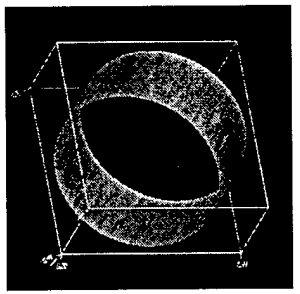


(a) Iteration 400. (b) Iteration 500. (c) Iteration 800.

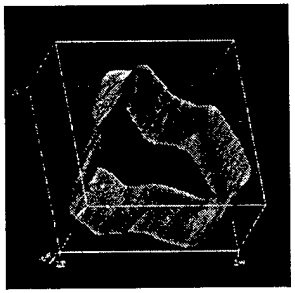
Figure 25: The original image contains a box with a semi-spherical region cut out from it. Results from the active contours without edges model: (a) 400, (b) 500, and (c) 800 iterations.



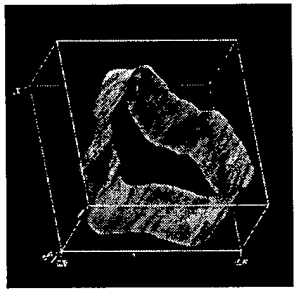
(a) One slice of the original image.



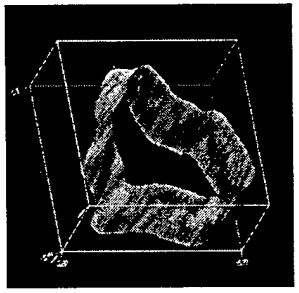
(b) Initial contour.



(c) 1000 iterations.

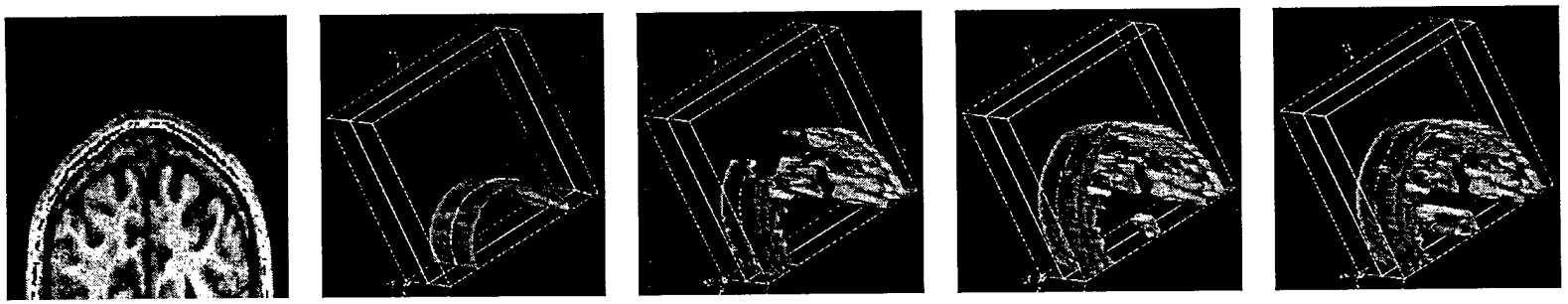


(d) 2000 iterations.



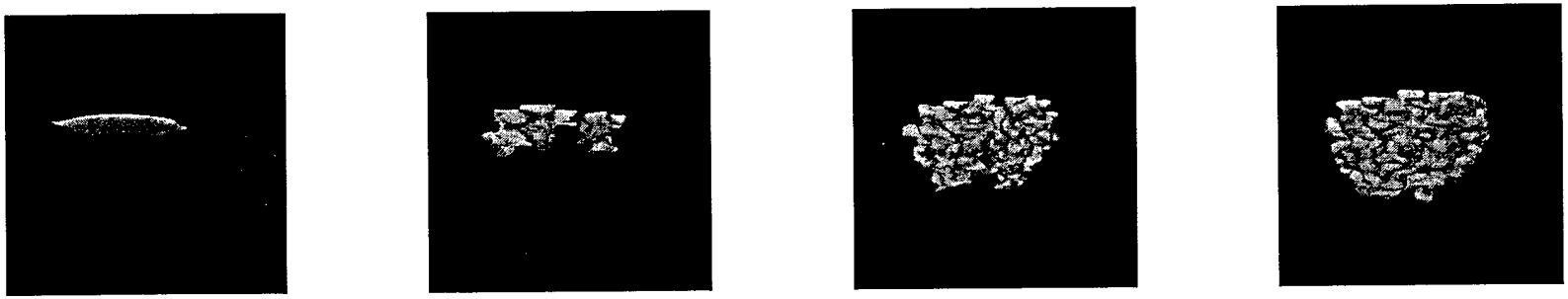
(e) 3000 iterations.

Figure 26: Segmentation results of a three-dimensional transrectal ultrasound image of a prostate. (a) The original 280×280 image. (b) Initial contour. (c) The contour after 1000 iterations. (d) The contour after 2000 iterations. (e) The contour after 3000 iterations.



(a) Original image. (b) Initial contour. (c) Iteration 400. (d) Iteration 800. (e) Iteration 1200.

Figure 27: Results from the active contours without edges model on (a) a three-dimensional brain image. (b) The initial contour. Segmentation results after (c) 400, (d) 800, and (e) 1200 iterations.



(a) Initial contour. (b) Iteration 60. (c) Iteration 100. (d) Iteration 200.

Figure 28: Results from the active contours without edges model on a three-dimensional brain image using a coarser grid of $29 \times 29 \times 19$. (a) The initial contour. Segmentation results after (b) 60, (c) 100, and (d) 200 iterations.

References

[1] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerical Mathematics*, vol. 66, pp. 1–31, 1993.

[2] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 158–174, Feb. 1995.

[3] J. A. Sethian and J. D. Strain, "Crystal growth and dendritic solidification," *Journal of Computational Physics*, vol. 98, pp. 231 – 253, 1992.

[4] D. Adalsteinsson and J. A. Sethian, "The fast construction of extension velocities in level set methods," *Journal of Computational Physics*, vol. 148, pp. 2–22, 1999.

[5] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science.* No. 3 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1999.

[6] A. Yezzi, A. Tsai, and A. Willsky, "Binary and ternary flows for image segmentation," in *Proceedings of the IEEE International Conference on Image Processing*, 1999.

[7] T. Chan and L. Vese, "Active contours without edges," tech. rep., CAM Report 98-53 University of California, Los Angeles, 1998. Revised 1999.

[8] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications Pure and Applied Mathematics*, vol. 42, no. 4, pp. 577–685, 1989.

[9] S. J. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[10] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 255–268, 1992.

[11] M. Kang, B. Merriman, S. Osher, and P. Smereka, "A level set approach for the motion of soap bubbles with curvature dependent velocity or acceleration," tech. rep., CAM Report 96-19, University of California, Los Angeles, 1996.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Machine Vision.* Addison-Wesley, 1991.

[13] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, July 1990.

[14] J. F. Canny, "Finding edges and lines in images," Master's thesis, Massachusetts Institute of Technology, June 1983.

[15] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contour," *International Journal on Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[16] S. Kichenassamy, A. Kumar, P. J. Olver, A.Tannenbaum, and A. Yezzi, "Conformal curvature flows: From phase transitions to active vision," *Arch. Rat. Mech. Anal.*, vol. 134, pp. 275–301, 1996.

[17] B. C. Vemuri, J. Ye, Y. Chen, and C. M. Leonard, "A level-set based approach to image registration," in *Proc. of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 86–93, 2000.

[18] J.-P. Thirion, "Fast non-rigid matching of 3D medical image," tech. rep., Research Report RR-2547, Epidure Project, INRIA Sophia, May 1995.

[19] V. Chalana, L. Ng, L. Rystrom, J. Gee, and D. Haynor, "Validation of brain segmentation and tissue classification algorithms for T1-weighted MR images," in *Proceedings of SPIE Medical Imaging Conference*, Feb. 2001.

[20] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, 1999.