LBL-35733
UC-405
Preprint

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Physics Division

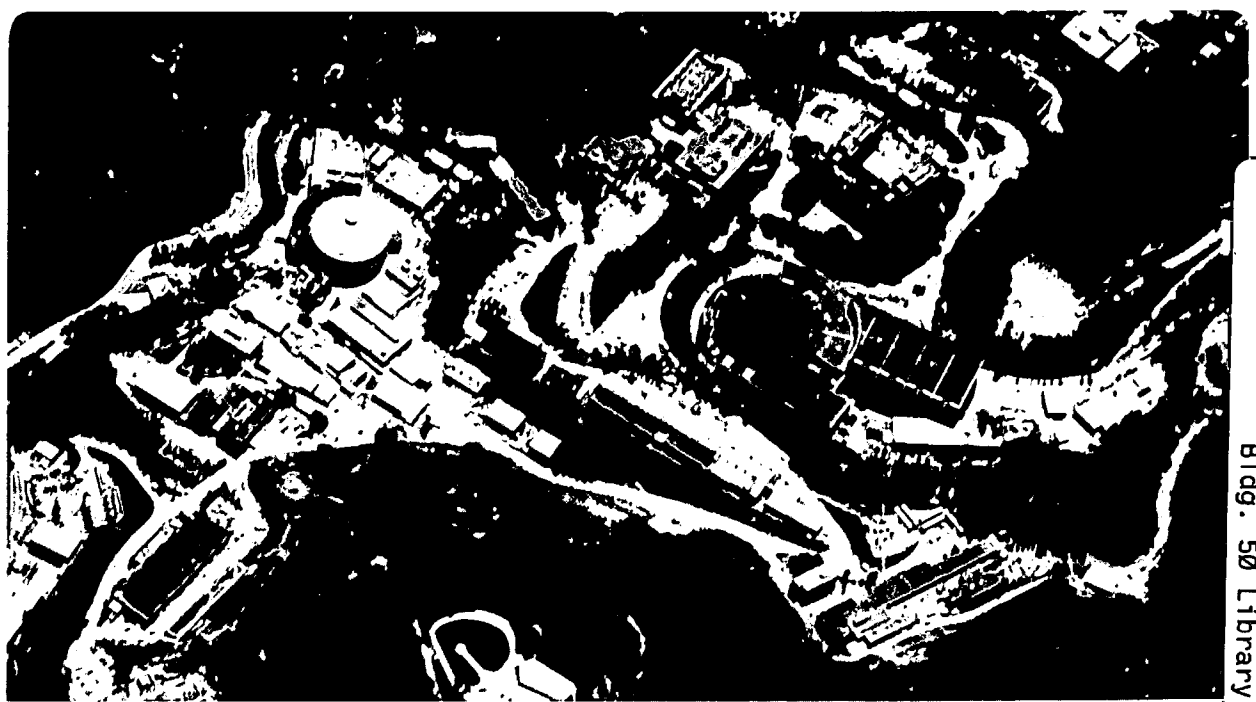
Mathematics Department

To be submitted for publication

Shape Modeling with Front Propagation: A Level Set Approach

R. Malladi, J.A. Sethian, and B.C. Vemuri

June 1994



REFERENCE COPY
Does Not
Circulate
LBL-35733
Copy 1
Bldg. 50 Library.

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

This report has been reproduced directly from the best available copy.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-35733

Shape Modeling with Front Propagation: A Level Set Approach[1]

Ravikanth Malladi
James A. Sethian
Department of Mathematics and Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Baba C. Vemuri
Department of Computer and Information Sciences
University of Florida
Gainesville, FL 32611

June 1994

[1]Supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098 and by the National Science Foundation/ARPA under grant DMS-8919074.

Shape Modeling with Front Propagation: A Level Set Approach *

Ravikanth Malladi,[1] James A. Sethian,[1] and Baba C. Vemuri[2]

[1] Lawrence Berkeley Laboratory
and
Department of Mathematics
University of California, Berkeley, CA 94720.

[2]Department of Computer & Information Sciences
University of Florida, Gainesville, FL 32611.

Abstract

Shape modeling is an important constituent of computer vision as well as computer graphics research. Shape models aid the tasks of object representation and recognition. This paper presents a new approach to shape modeling which retains some of the attractive features of existing methods, and overcomes some of their limitations. Our techniques can be applied to model arbitrarily complex shapes, which include shapes with significant protrusions, and to situations where no *a priori* assumption about the object's topology is made. A single instance of our model, when presented with an image having more than one object of interest, has the ability to split freely to represent each object. This method is based on the ideas developed by Osher and Sethian to model propagating solid/liquid interfaces with curvature-dependent speeds. The interface (front) is a closed, nonintersecting, hypersurface flowing along its gradient field with constant speed or a speed that depends on the curvature. It is moved by solving a "Hamilton-Jacobi" type equation written for a function in which the interface is a particular level set. A speed term synthesized from the image is used to stop the interface in the vicinity of object boundaries. The resulting equation of motion is solved by employing entropy-satisfying upwind finite difference schemes. We present a variety of ways of computing evolving front, including narrow bands, reinitializations, and different stopping criteria. The efficacy of the scheme is demonstrated with numerical experiments on some synthesized images and some low contrast medical images.

*[1] Supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Dept. of Energy under Contract DE-AC03-76SD00098 and by the NSF ARPA under grant DMS-8919074.
[2] Supported in part by NSF grant ECS-9210648.

1 Introduction

In this paper, we describe a modeling technique based on a level set approach for recovering shapes of objects in two and three dimensions from various types of image data. The modeling technique may be viewed as a form of active modeling such as "snakes" [15] and deformable surfaces [34] since, the model which consists of a moving front, may be molded into any desired shape by externally applied halting criteria synthesized from the image data. The "snakes" or deformable surfaces may be viewed as Lagrangian geometric formulations wherein the boundary of the model is represented in a parametric form. These parameterized boundary representations will encounter difficulties when the dynamic model embedded in a noisy data set is expanding/shrinking along its normal field [10] and sharp corners, cusps develop or pieces of the boundary intersect. By exploiting recent advances in interface techniques, our modeling technique avoids this Lagrangian geometric view and instead capitalizes on a related initial value partial differential equation. In this setting, several advantages are apparent, including the ability to evolve the model in the presence of sharp corners, cusps and changes in topology, model shapes with significant protrusions and holes in a seamless fashion, and extension to three dimensions in an extremely straightforward way.

1.1 Background

An important goal of computational vision is to recover the shapes of objects in 2D and 3D from various types of visual data. One way to achieve this goal is via model-based techniques. Broadly speaking, these techniques involve the use of a model whose boundary representation is matched to the image to recover the object of interest. These models can either be rigid, such as correlation-based template matching techniques, or nonrigid, as those used in dynamic model fitting techniques.

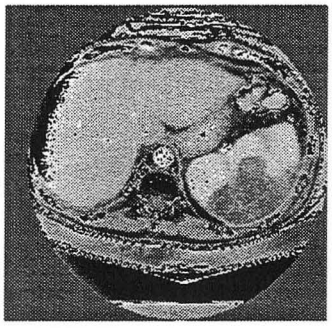
Shape recovery from raw data typically precedes its symbolic representation. Shape models are expected to aid the recovery of detailed structure from noisy data using only the weakest of the possible assumptions about the observed shape. To this end, several variational shape reconstruction methods have been proposed and there is abundant literature on the same (see [4, 27, 35, 38, 17] and references therein). Generalized spline models with continuity constraints are well suited for fulfilling the goals of shape recovery (see [6, 33]). Generalized splines are the key ingredient of the dynamic shape modeling paradigm introduced to vision literature by Kass *et*

al. [15]. Incorporating dynamics into shape modeling enables the creation of realistic animation for computer graphics applications and for tracking moving objects in computer vision. Following the advent of the dynamic shape modeling paradigm [15, 34], considerable research followed, with numerous application specific modifications to the modeling primitives, and external forces derived from data constraints [39, 18, 11, 24, 36, 37].

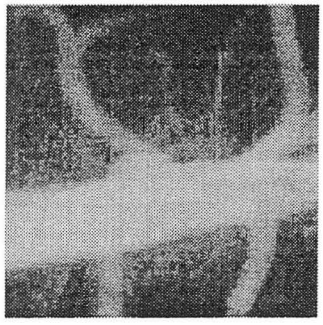
The final recovered shape in these schemes can depend on the initial guess made to start the numerical reconstruction procedure. This is due to the fact that the energy functionals used in the variational formulations are typically nonconvex and hence have multiple local minima. Therefore, the numerical procedures, for convergence to a satisfactory solution require an initial guess which is reasonably close to the desired shape. One solution to this problem in the one-dimensional case has been presented by Amini *et al.* [2]. They use a discrete form of dynamic programming to optimize the univariate variational problem.

The framework of energy minimization (snakes) has been used successfully in the past for extracting salient image contours such as edges and lines by Kass *et al.* [15]. To make the final result relatively insensitive to the initial conditions, Cohen [10] suggested the use of an inflation force which makes the snake behave like an edge seeking active model. Although the inflation force prevents the curve from getting trapped by isolated spurious edges, the active contour model cannot be made to extrude through any significant protrusions that a shape may possess (see figure 1(b)) without resorting to cumbersome resampling techniques. In this paper, we present a technique which overcomes this problem and accurately models bifurcations and protrusions in complex shapes.

Most existing shape modeling schemes require that the topology of the object be known before the shape recovery can commence. However, it is not always possible to specify the topology of an object prior to its recovery. For example, an important concern in object tracking and motion detection applications is topological change resulting from tracking the positions of object boundaries in an image sequence through time. During their evolution, these closed contours may change connectivity and split, thereby undergoing a topological transformation. One such example is the splitting of cell boundary in a sequence of images depicting cell division. A heuristic criterion for splitting and merging of curves in 2D which is based on monitoring deformation energies of points



(a) CT image



(b) DSA image

(c) Shapes with holes

Figure 1: Test bed for our topology-independent shape modeling scheme.

on the elastic curve has been discussed in [26]. In the context of static problems, more recently, particle systems have been used to model surfaces of arbitrary topology [32]. Here, particles can be added and deleted dynamically to enlarge and trim the surface respectively.

The schemes described in this paper offer a new approach to some of the above problems. To begin, the convergence to the final result is relatively independent of the shape initialization. The algorithm allows branches to sprout automatically as the front moves. The scheme described in this paper can be applied where no *a priori* assumption about the object's topology is made. A single instance of our model, when presented with an image having more than one shape of interest (see figure 1(c)), has the ability to split freely to represent each shape [19, 20]. We show that by using our approach, it is also possible to extract the bounding contours of shapes with holes in a seamless fashion (see figure 13).

Our method is inspired by ideas first introduced in Osher and Sethian [23, 29], which grew out of work in Sethian [28], to model propagating fronts with curvature-dependent speeds. Two such examples are flame propagation and crystal growth, in which the speed of the moving interface normal to itself depends on transport terms modified by the local curvature. The challenge in these problems is to devise numerical schemes for the equations of the propagating front which will accurately approximate these highly unstable physical phenomena. Osher and Sethian [23] achieve this by viewing the propagating surface as a specific level set of a higher-dimensional function. The

equation of motion for this function is reminiscent of an initial value "Hamilton-Jacobi" equation with a parabolic right-hand side and is closely related to a viscous hyperbolic conservation law.

In our work, we adopt these level set techniques to the problem of shape recovery. To isolate a shape from its background, we first consider a closed, nonintersecting, initial hypersurface placed inside (or outside) it. This hypersurface is then made to flow along its gradient field with a speed $F(K)$, where K is the curvature of the hypersurface. Unknown shapes are recovered by making the front adhere to the object boundaries. This is done by synthesizing a speed term from image data which acts as a halting criterion. Finally, we note that a separate study also applying a level set approach has been performed independently by Caselles *et al.* [7].

The outline of this paper is as follows. In section 2, we briefly explain the level set approach to front propagation problems and the accompanying numerical algorithms. In sections 3 and 4, we discuss the application of this technique to shape recovery problems, consider various speed functions and approaches to the problem, such as the effect of global speed laws, narrow band formulations, reinitialization and stopping criteria. In section 5, we present some experimental results of applying our method to some synthetic and low contrast medical images. We conclude in section 6.

2 Front Propagation Problem

In this section we present the level set technique due to Osher and Sethian [23]. For details and an expository review, see Sethian [29].

As a starting point and motivation for the level set approach, consider a closed curve moving in the plane, that is, let $\gamma(0)$ be a smooth, closed initial curve in Euclidean plane $\Re^2$, and let $\gamma(t)$ be the one-parameter family of curves generated by moving $\gamma(0)$ along its normal vector field with speed $F(K)$, a given scalar function of the curvature K. Let $\mathbf{x}(s,t)$ be the position vector which parameterizes $\gamma(t)$ by s, $0 \leq s \leq S$.

One numerical approach to this problem is to take the above Lagrangian description of the problem, produce equations of motion for the position vector $\mathbf{x}(s,t)$, and then discretize the parameterization with a set of discrete marker particles laying on the moving front. These discrete markers are updated in time by approximating the spatial derivatives in the equations of motion,

and advancing their positions. However, there are several problems with this approach, as discussed in Sethian [28]. First, small errors in the computed particle positions are tremendously amplified by the curvature term, and calculations are prone to instability unless an extremely small time step is employed. Second, in the absence of a smoothing curvature (viscous) term, singularities develop in the propagating front, and an entropy condition must be observed to extract the correct weak solution. Third, topological changes are difficult to manage as the evolving interface breaks and merges. And fourth, significant bookkeeping problems occur in the extension of this technique to three dimensions.

As an alternative, the central idea in the level set approach of Osher and Sethian [23] is to represent the front $\gamma(t)$ as the level set $\{\psi = 0\}$ of a function ψ. Thus, given a moving closed hypersurface $\gamma(t)$, that is, $\gamma(t=0) : [0,\infty) \rightarrow \Re^N$, we wish to produce an Eulerian formulation for the motion of the hypersurface propagating along its normal direction with speed F, where F can be a function of various arguments, including the curvature, normal direction, e.t.c. The main idea is to embed this propagating interface as the zero level set of a higher dimensional function ψ. Let $\psi(\mathbf{x}, t=0)$, where $\mathbf{x} \in \Re^N$ be defined by

$$\psi(\mathbf{x}, t=0) = \pm d \tag{1}$$

where d is the distance from $\mathbf{x}$ to $\gamma(t=0)$, and the plus (minus) sign is chosen if the point $\mathbf{x}$ is outside (inside) the initial hypersurface $\gamma(t=0)$. Thus, we have an initial function $\psi(\mathbf{x}, t=0) : \Re^N \rightarrow \Re$ with the property that

$$\gamma(t=0) = (\mathbf{x} | \psi(\mathbf{x}, t=0) = 0) \tag{2}$$

As illustration, consider the example of an expanding circle. Suppose the initial front γ at $t=0$ is a circle in the xy-plane (figure 2(a)). We imagine that the circle is the level set $\{\psi = 0\}$ of an initial surface $z = \psi(x, y, t=0)$ in $\Re^3$ (see figure 2(b)). We can then match the one-parameter family of moving curves $\gamma(t)$ with a one-parameter family of moving surfaces in such a way that the level set $\{\psi = 0\}$ always yields the moving front (see figures 2(c) & 2(d)).

Our goal is to now produce an equation for the evolving function $\psi(\mathbf{x}, t)$ which contains the embedded motion of $\gamma(t)$ as the level set $\{\psi = 0\}$. Here, we follow the derivation presented in [22]. Let $\mathbf{x}(t), t \in [0,\infty)$ be the path of a point on the propagating front. That is, $\mathbf{x}(t=0)$ is a point on

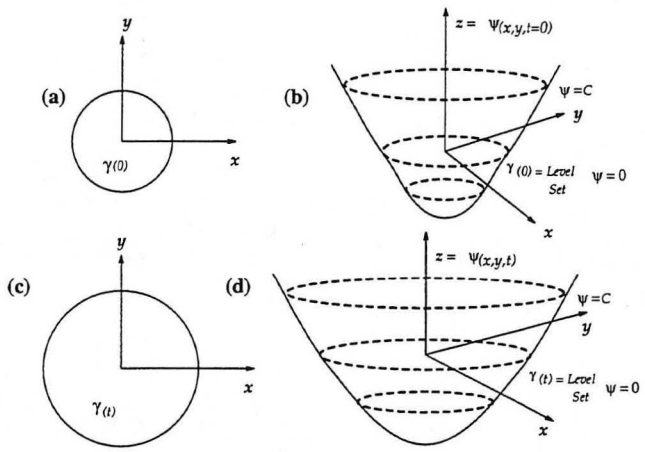


Figure 2: Level set formulation of equations of motion – (a) & (b) show the curve γ and the surface $\psi(x, y)$ at $t = 0$, and (c) & (d) show the curve γ and the corresponding surface $\psi(x, y)$ at time t.

the initial front $\gamma(t = 0)$, and $\mathbf{x}_t = F(\mathbf{x}(t))$ with the vector $\mathbf{x}_t$ normal to the front at $\mathbf{x}(t)$. Since the evolving function ψ is always zero on the propagating hypersurface, we must have

$$\psi(\mathbf{x}(t), t) = 0. \tag{3}$$

By the chain rule,

$$\psi_t + \sum_{i=1}^{N} \psi_{x_i}\, x_{i_t} = 0 \tag{4}$$

where x_i is the ith component of $\mathbf{x}$. Let $(u_1, u_2, \ldots, u_N) = (x_{1_t}, x_{2_t}, \ldots, x_{N_t})$. Since

$$\sum_{i=1}^{N} \psi_{x_i}\, x_{i_t} = (\psi_{x_1}, \psi_{x_2}, \ldots, \psi_{x_N}) \cdot (u_1, u_2, \ldots, u_N) = F(\mathbf{x}(t))|\nabla\psi|, \tag{5}$$

we then have the evolution equation for ψ, namely

$$\psi_t + F \mid \nabla\psi \mid = 0 \tag{6}$$

with a given value of $\psi(\mathbf{x}, t = 0)$. We refer to this as a Hamilton-Jacobi "type" equation because, for certain forms of the speed function F, we obtain the standard Hamilton-Jacobi equation.

There are four major advantages to this Eulerian Hamilton-Jacobi formulation. The first is that the evolving function $\psi(\mathbf{x}, t)$ always remains a function as long as F is smooth. However, the level surface $\{\psi = 0\}$, and hence the propagating hypersurface $\gamma(t)$ may change topology, break, merge, and form sharp corners as the function ψ evolves, see [23].

The second advantage of this Eulerian formulation concerns numerical approximation. Because $\psi(\mathbf{x}, t)$ remains a function as it evolves, we may use a discrete grid in the domain of $\mathbf{x}$ and substitute finite difference approximations for the spatial and temporal derivatives. For example, using a uniform mesh of spacing h, with grid nodes ij, and employing the standard notation that ψ_{ij}^n is the approximation to the solution $\psi(ih, jh, n\Delta t)$, where Δt is the time step, we might write

$$\frac{\psi_{ij}^{n+1} - \psi_{ij}^n}{\Delta t} + (F)(\nabla_{ij}\psi_{ij}^n) = 0. \tag{7}$$

Here, we have used forward differences in time, and let $\nabla_{ij}\psi_{ij}^n$ be some appropriate finite difference operator for the spatial derivative.

The correct technique for approximating the spatial derivative in the above comes from respecting the appropriate entropy condition for propagating fronts, discussed in detail in [29]. As brief motivation for these schemes, consider a periodic cosine curve propagating in its normal direction with speed $F = 1 - \varepsilon K$, where K is the curvature. This problem has been discussed extensively in [28]. For $\varepsilon > 0$, the front stays smooth for all time. For $\varepsilon = 0$, the parameterized analytic solution corresponds to a front which passes through itself and develops a swallowtail solution. In order for the propagating front to correspond to the boundary of an expanding region, we invoke the entropy condition, namely that if the boundary is viewed as a propagating flame, then *once a particle is burnt, it stays burnt.* This entropy condition yields the front which corresponds to the limiting solution as $\varepsilon \to 0$ of the smooth case.

In order to build a correct entropy-satisfying approximation to the difference operator, we exploit the technology of hyperbolic conservation laws. Following [23], we use a modification of an Engquist-Osher schemes [12]. That is, given a speed function $F(K)$, we update the front by the following scheme. First, separate $F(K)$ into a constant advection term F_0 and the remainder $F_1(K)$, that is,

$$F(K) = F_0 + F_1(K) \tag{8}$$

The advection component F_0 of the speed function is then approximated using upwind schemes, while the remainder is approximated using central differences. In one space dimension, we have

$$\phi_i^{n+1} = \phi_i^n - \Delta t \left[\{(\max(D_i^- \phi, 0))^2 + (\min(D_i^+ \phi, 0))^2\}^{1/2} - F_1(K)\nabla\phi_i^n \right]. \tag{9}$$

Extension to higher dimensions are straightforward; we use the version given in [30].

The third advantage of the above formulation is that intrinsic geometric properties of the front may be easily determined from the level function ψ. For example, at any point of the front, the normal vector, is given by

$$\vec{n} = \nabla\psi \tag{10}$$

and the curvature is easily obtained from the divergence of the gradient of the unit normal vector to front, i.e.,

$$K = \nabla \cdot \frac{\nabla\psi}{|\nabla\psi|} = -\frac{\psi_{xx}\psi_y^2 - 2\psi_x\psi_y\psi_{xy} + \psi_{yy}\psi_x^2}{(\psi_x^2 + \psi_y^2)^{3/2}}. \tag{11}$$

Finally, the fourth advantage of the above level set approach is that there are no significant differences in following fronts in three space dimensions. By simply extending the array structures and gradients operators, propagating surfaces are easily handled.

Since its introduction in [23], the above level set approach has been used in a wide collection of problems involving moving interfaces. Some of these applications include the generation of minimal surfaces [8], singularities and geodesics in moving curves and surfaces in [9], flame propagation [25, 40], fluid interfaces [31, 22]. Extensions of the basic technique include fast methods in [1] and extensions to triple points in [3]. The fundamental Eulerian perspective presented by this approach has since been adopted in many theoretical analyses of mean curvature flow, in particular, see [13]. In computer vision, a model for shape theory based on this work has been presented in [16].

3 Shape Recovery with Front Propagation

In this section, we describe how the level set formulation for the front propagation problem discussed in the previous section can be used for shape recovery. First, note that the front represents the boundary of an evolving shape. Since the idea is to extract objects' shapes from a given image, the front should be forced to stop in the vicinity of the desired objects' boundaries. This is analogous to the force criterion used to push the active contour model towards desired shapes [15]. We define the final shape to be the configuration when all the points on the front come to a stop, thereby bringing the computation to an end.

Our goal now is to define a speed function from the image data that can be applied on the propagating front as a halting criterion. As before, we split the speed function F into two components:

$F = F_A + F_G$. The term F_A, referred to as the advection term, is independent of the moving front's geometry. The front uniformly expands or contracts with speed F_A depending on its sign and is analogous to the inflation force defined in [10]. The second term F_G, is the part which depends on the geometry of the front, such as its local curvature. This (diffusion) term smooths out the high curvature regions of the front and has the same regularizing effect on the front as the internal deformation energy term in thin-plate-membrane splines [15] (see the figure (9)). We rewrite equation (6) by splitting the influence of F as

$$\psi_t + F_A \mid \nabla\psi \mid + F_G \mid \nabla\psi \mid = 0. \tag{12}$$

First consider the case when the front moves with a constant speed, i.e., $F_G = 0 \Rightarrow F = F_A$. Define a negative speed F_I to be

$$F_I(x,y) = \frac{-F_A}{(M_1 - M_2)} \{ \mid \nabla G_\sigma * I(x,y) \mid - M_2 \}, \tag{13}$$

where M_1 and M_2 are the maximum and minimum values of the magnitude of image gradient $\mid \nabla G_\sigma * I(x,y) \mid$, $(x,y) \in \Omega$. The expression $G_\sigma * I$ denotes the image convolved with a Gaussian smoothing filter whose characteristic width is σ. Alternately, we could use a smoothed zero-crossing image to synthesize the negative speed function. The zero-crossing image is produced by detecting zero-crossings in the function $\nabla^2 G_\sigma * I$, which is the original image convolved with a Laplacian-of-Gaussian filter whose characteristic width is σ. The value of F_I lies in the range $[-F_A, 0]$ as the value of image gradient varies between M_1 and M_2. From this argument it is clear that, if $\mid \nabla G_\sigma * I(x,y) \mid$ approaches the maximum M_1 at the object boundaries, then the front gradually attains zero speed as it gets closer to the object boundaries and eventually comes to a stop.

If $F_G \neq 0$, then it is not possible to find an additive speed term from the image that will cause the net speed of the front to approach zero in the neighborhood of a desired shape. Instead, we multiply the speed function $F = F_A + F_G$ with a quantity k_I. The term k_I, which is defined as

$$k_I(x,y) = \frac{1}{1 + \mid \nabla G_\sigma * I(x,y) \mid}, \tag{14}$$

has values that are closer to zero in regions of high image gradient and values that are closer to unity in regions with relatively constant intensity. If one desires a speed function that falls to zero

faster than the reciprocal function, the following definition can be employed:

$$k_I(x,y) = e^{-|\nabla G_\sigma * I(x,y)|}. \tag{15}$$

More sophisticated stopping criteria can be synthesized by using the orientation dependent "steerable" filters [14].

4 Extending the Speed Function

The image-based speed terms have meaning only on the boundary $\gamma(t)$, i.e. on the level set $\{\psi = 0\}$. This follows from the fact that they were designed to force the propagating level set $\{\psi = 0\}$ to a complete stop in the neighborhood of an object boundary. However, the level set equation of motion is written for the function ψ defined over the entire domain. Consequently, we require that the evolution equation has a consistent physical meaning for all the level sets, i.e. at every point $(x,y) \in \Omega$. The speed function F_I derives its meaning not from the geometry of ψ but from the configuration of the level set $\{\psi = 0\}$ in the image plane. Thus, our goal is to construct an image-based speed function $\hat{F}_I$ that is globally defined. We call it an *extension* of F_I off the level set $\{\psi = 0\}$ because it extends the meaning of F_I to other level sets [30]. Note that the level set $\{\psi = 0\}$ lies in the image plane and therefore $\hat{F}_I$ must equal F_I on $\{\psi = 0\}$. The same argument applies to the coefficient k_I. With the extensions so defined, the equation of motion for the case $F = F_A$ is given by

$$\psi_t + (F_A + \hat{F}_I) \mid \nabla\psi \mid = 0, \tag{16}$$

and

$$\psi_t + \hat{k}_I(F_A + F_G) \mid \nabla\psi \mid = 0, \tag{17}$$

when $F = F_A + F_G$.

If the level curves are moving with a constant speed, i.e. $F_G = 0$, then at any time t, a typical level set $\{\psi = C\}$, $C \in R$, is a distance C away from the level set $\{\psi = 0\}$ (see figure 3). Observe that the above statement is a rephrased version of *Huygen's principle* which, from a geometrical standpoint, stipulates that the position of a front propagating with unit speed at a given time t

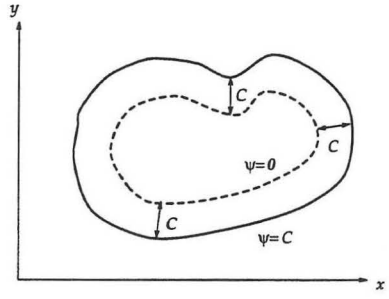


Figure 3: Huygen's principle construction

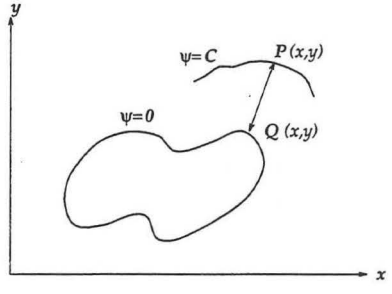


Figure 4: Extension of image-based speed terms to other level sets

should consist of only the set of points located a distance t away from the initial front. On the other hand, for $F_G \neq 0$, the level sets will not remain a constant distance apart.

With this in mind, there are several ways to extend the speed function to the neighboring level sets.

4.1 Global Extension

As a first attempt, we require that the external (image-based) speed function be such that level sets moving under this speed function cannot collide.

We can construct one such extension to the image-based speed function by (see figure 4) letting the value of $\hat{F}_I$ $(\hat{k}_I)$ at a point P lying on a level set $\{\psi = C\}$ be the value of F_I (k_I) at a point Q, such that point Q is closest to P and lies on the level set $\{\psi = 0\}$. Thus, $\hat{F}_I$ $(\hat{k}_I)$ reduces to F_I (k_I) on $\{\psi = 0\}$.

By updating the level set function on a grid, we are moving the level sets without constructing them explicitly. Therefore a straightforward algorithm consists of advancing from one time step to the next as follows:

Algorithm 1

1. At each grid point $(i\Delta x, j\Delta y)$, where Δx and Δy are step sizes in either coordinate directions, the *extension* of image-based speed term is computed. This is done in accordance with the construction described in previous section; i.e., by searching for a point q which lies on the level set $\{\psi = 0\}$, and is closest to the point $(i\Delta x, j\Delta y)$. The value of image-based speed term at the current point is simply its value at the point q.

2. With the value of extended speed term $(\hat{k}_I^n)_{i,j}$ and $\psi_{i,j}^n$, calculate $\psi_{i,j}^{n+1}$ using the upwind, finite difference schemes given in [30].

3. Construct an approximation for the level set $\{\psi = 0\}$ from $\psi_{i,j}^{n+1}$. This is required to visualize the current position of the front in the image plane. A piecewise linear approximation for the front $\gamma(t)$ is constructed as follows. Given a cell $C(i,j)$, if $\max(\psi_{i,j}, \psi_{i+1,j}, \psi_{i,j+1}, \psi_{i+1,j+1}) < 0$ or $\min(\psi_{i,j}, \psi_{i+1,j}, \psi_{i,j+1}, \psi_{i+1,j+1}) > 0$, then $C(i,j) \notin \gamma(t)$ and is ignored, else, the entry and exit points where $\psi = 0$ are found by linear interpolation. This provides two nodes on $\gamma(t)$ and thus, one of the line segments which form the approximation to $\gamma(t)$. The collection of all such line segments constitutes the approximation to the level set $\{\psi = 0\}$, which is used for future evaluation of the image-based speed term in the update equation.

4. Replace n by $n + 1$ and return to step 1.

4.2 Global Extension with Reinitialization

The above construction can create a discontinuous velocity extension away from the zero level set, since the distance function is not differentiable. One solution to this is to reinitialization the level set function every fixed number of time steps to keep the level sets evenly spaced around the front. A straightforward way to do this is to recompute the distance from each point of the grid to the zero level set. However, this is an $O(N^3)$ operation, if we assume that there are N points in each coordinate direction, plus approximately $O(N)$ points on the interfaces.

An alternative to this reconstruction is provided by [31], based on an idea of Morel. The idea is simply to iterate on the level set function at a given time according to the following equation:

$$\psi_{k+1} = \psi_k + S(\psi)(1 - |\nabla\psi|). \tag{18}$$

In the limit as $k \to \infty$, this convergences to the distance function, with some error in relocating the original zero level set. For details, see [5].

The most expensive step in either of these algorithms is the computation of the *extension* for image-based speed term. This is because at each grid point, we must search for the closest point lying on the level set $\{\psi = 0\}$. Moreover, if $F_G = 0$, then the stability requirement for the explicit method for solving our level set equation is $\Delta t = O(\Delta x)$. For the full equation (12), the stability requirement is $\Delta t = O(\Delta x^2)$. This could potentially force a very small time step for fine grids. These two effects, individually and compounded, make the computation exceedingly slow. In the case of reinitializing using the above iteration formula, additional labor is involved.

4.3 Narrow-Band Extension with Reinitialization

As a efficient alternative, we observe that the front can be moved by updating the level set function at a small set of points in the neighborhood of the zero set instead of updating it at all the points on the grid. In figure (5) the bold curve depicts the level set $\{\psi = 0\}$ and the shaded region around it is the narrow band. The narrow band is bounded on either side by two curves which are a distance δ apart, i.e., the two curves are the level sets $\{\psi = \pm\delta/2\}$. The value of δ determines the number of grid points that fall within the narrow band. Since, during a given time step the value of ψ_{ij} is not updated at points lying outside the narrow band, the level sets $\{|\ \psi\ | > \delta/2\}$ remain stationary. The zero set which lies inside moves until it collides with the boundary of the narrow band. Which boundary the front collides with depends on whether it is moving inward or outward; either which way, it cannot move past the narrow band. A complete discussion of the narrow band techniques for interface propagation may be found in [1].

As a consequence of our update strategy, the front can be moved through a maximum distance of $\delta/2$, either inward or outward, at which point we must rebuild an appropriate (a new) narrow band. We reinitialize the ψ function by treating the current zero set configuration, i.e., $\{\psi = 0\}$, as the initial curve $\gamma(0)$. Chopp [8] observed that the reinitialization step can be made cheaper by treating the interior and exterior mesh points as sign holders. Note that the reinitialization procedure must

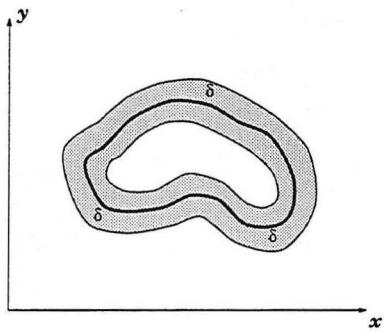


Figure 5: A narrow band of width δ around the level set $\{\psi = 0\}$.

account for the case when $\{\psi = 0\}$ changes topology. This procedure will restore the meaning of ψ function by correcting the inaccuracies introduced as a result of our update algorithm. Once a new ψ function is defined on the grid, we can create a new narrow band around the zero set, and go through another set of, say l, iterations in time to move the front ahead by a distance equal to $\delta/2$. The value of l is set to the number of time steps required to move the front by a distance roughly equal to $\delta/2$. This choice depends on some experimentation. Thus, a faster algorithm for shape recovery consists of the following steps:

Algorithm 2

1. Set the iteration number $m = 0$ and go to step 2.

2. At each grid point (i, j) lying inside the narrow band, compute the *extension* $\hat{k}_I$ of image-based speed term.

3. With the above value of extended speed term $(\hat{k}_I^m)_{i,j}$ and $\psi_{i,j}^m$, calculate $\psi_{i,j}^{m+1}$ using the upwind, finite difference scheme given in [30].

4. Construct a polygonal approximation for the level set $\{\psi = 0\}$ from $\psi_{i,j}^{m+1}$. A contour tracing procedure is used to obtain a polygonal approximation. Given a cell (i, j) which contains $\gamma(t)$, this procedure traces the contour by scanning the neighboring cells in order to find the next cell which contains $\gamma(t)$. Once such a cell is found, the process is repeated until the contour closes on itself. The set of nodes visited during this tracing process constitutes the polygonal approximation to $\gamma(t)$. In general, to collect all the closed contours, the above tracing procedure is started at a new, as yet unvisited, cell which contains the level set

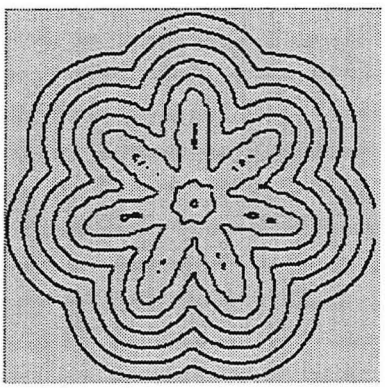
$\{\psi = 0\}$. A polygonal approximation is required in step 2 for the evaluation of image-based speed term and more importantly, in step 6 for reinitializing the ψ function.

5. Increment m by one. If the value of m equals l, go to step 6, else, go to step 2.

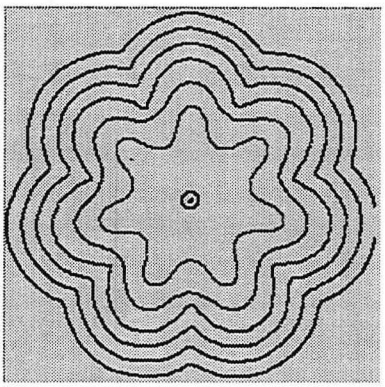
6. Compute the value of signed distance function ψ by treating the polygonal approximation of $\{\psi = 0\}$ as the initial contour $\gamma(0)$. As mentioned earlier, a more general method of reinitialization is required when $\{\psi = 0\}$ changes topology. Go to step 1.

In this approach, since we only update ψ at points lying in the narrow band, the issue of specifying boundary conditions for points lying on the edge of the band becomes pertinent. With our relatively simple speed motion, the free-end boundary condition is adequate, however, in more complex applications such as crystal growth, and flame propagation, accurate specification of boundary conditions is necessary [1].

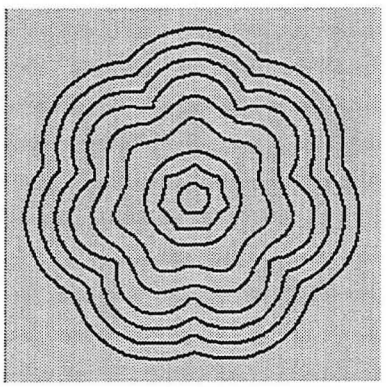
We now show that this new faster approach provides a correct approximation to the propagating front problem. In figure (6), we show the result of applying narrow-band algorithm to a star shaped front propagating with speed $F = -K$, where K is the curvature as in equation (11). The calculation was done on a unit box with 64 points in either direction, and a time step of $\Delta t = 0.00003$ was employed. The width of the narrow band has been set to $\delta = 0.075$, and the ψ function was recomputed once every ($l =$) 40 time steps. In figure 6(a), we show the initial curve along with the level sets $\{| \psi | < 0.2\}$. After 40 narrow-band updates (figure 6(b)), only the level sets $\{| \psi | < 0.0375\}$ move and the rest remain stationary. We note the inconsistency between the level sets lying on either side of the narrow band, making the reinitialization step necessary in order to restore the meaning of the ψ function. Following the reinitialization step, another 40 update steps are applied (figure 6(c)), which "diffuses" the high curvature regions of the front even further. In subsequent figures, the results of repeatedly applying the same strategy are shown. Finally, in figure 6(f), the peaks and troughs on the front get completely diffused, and it attains a smooth circular configuration after 4 reinitialization steps and a total of 200 time steps.



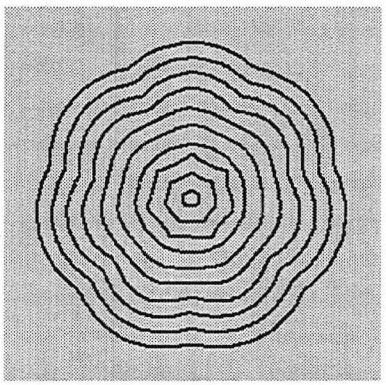
(a) $t = 0.0000$



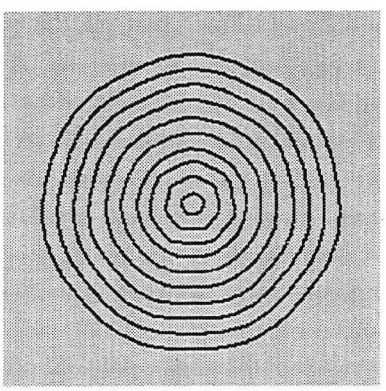
(b) $t = 0.0012$



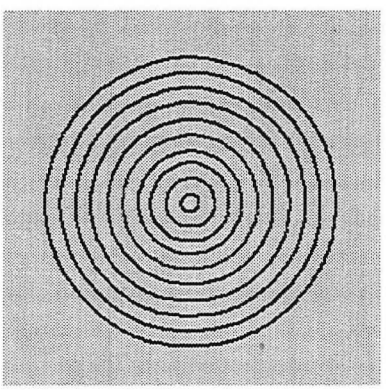
(c) $t = 0.0024$



(d) $t = 0.0036$



(e) $t = 0.0048$



(f) $t = 0.0060$

Figure 6: Narrow-band algorithm applied to a star-shaped front propagating with speed $F = -K$. Calculations were done on a 64×64 grid with a time step $\Delta t = 0.00003$. ψ was recomputed after every 40 time iterations.

4.4 Straightforward Narrow-Band Extension

The narrow-band approach, in addition to being computationally efficient, allows us to return to the original construction of the speed function extension and replace it with a more mathematically appealing version. Since the narrow-band mechanism periodically "recalibrates" the front, we can in fact simply move each level set with the speed determined by the image gradient as given in equations (14) and (15). In other words, for points inside the narrow band, the external speed values are picked directly from their corresponding image locations. Thus, we can ignore the previous extension velocity and provide a purely geometric one based on the local image gradient. Although this may cause many other level sets to temporarily stop, the narrow-band reinitialization resets them all around the zero level set. This will ensure that the zero level set is drawn close to the object boundary as well as retain other desirable properties of the level set approach, such as topological merge and split. Also, since the extension computation does not involve any search, the time complexity of this approach is identical to that of a basic narrow-band front propagation algorithm. We currently use this computationally efficient algorithm, and suggest it for others interested in this work.

5 Shape Recovery Results

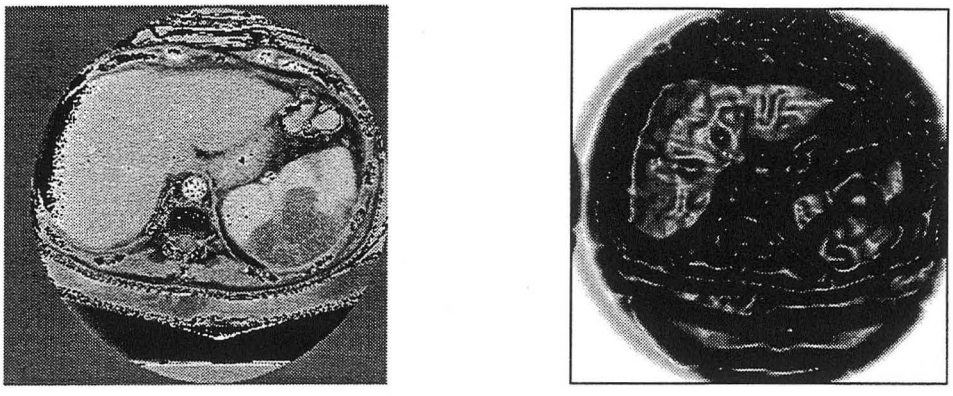
In this section we present several shape recovery results that were obtained by applying the narrow-band level set algorithm to image data. Given an image, our method requires the user to provide an initial contour $\gamma(0)$. The initial contour can be placed anywhere in the image plane. However, it must be placed inside a desired shape or enclose all the constituent shapes. Our front seeks the object boundaries by either propagating inward or outward in the normal direction. This choice is made at the time of initialization. Note that after the specification of initial shape of $\gamma(0)$, our algorithm does not require any further user interaction. On the other hand, the user may interact with the model by varying the smoothness control parameter ε until a desired amount of smoothness is achieved in a given shape.

The initial value of the function ψ i.e., $\psi(\mathbf{x}, 0)$ is computed from $\gamma(0)$. We first discretize the level set function ψ on the image plane and denote $\psi_{i,j}$ as the value of ψ at a grid point $(i\Delta x, j\Delta y)$,

where Δx and Δy are step sizes in either coordinate directions. We define the distance from a point (i, j) to the initial curve to be the shortest distance from (i, j) to $\gamma(0)$. The magnitude of $\psi_{i,j}$ is set to this value. We use the plus sign if (i, j) is outside $\gamma(0)$ and minus sign if (i, j) is inside. Once the value of $\psi_{i,j}$ is computed at time $t = 0$ by following the above procedure, we use algorithms from the previous section to move the front.

We now present our shape recovery results in 2D. First, we consider a 256×256 CT (computed tomography) image of an abdominal section shown in figure 7(a), with the goal of recovering the shape of the stomach in this particular slice. The function ψ has been discretized on a 128×128 mesh, i.e., calculations are performed at every second pixel. In figure 8(a), we show the closed contour that the user places inside the desired shape at time $t = 0$. The function ψ is then made to propagate in the normal direction with speed $F = \hat{k}_I(-1.0 - 0.025K)$. We employed the narrow-band update algorithm to move the front with a time step size set to $\Delta t = 0.0005$, and the ψ function was recomputed after every 50 time steps. Figure 7(b) shows the image-based speed term which is synthesized according to equation (14). Note that in figure 7(b), $k_I(x, y)$ values lying in the interval [0..1] have been mapped into the interval [0..255]. In figures 8(b) through 8(e) we depict the configuration of the level set $\{\psi = 0\}$ at four intermediate time instants. The final result is achieved after 575 time iterations and is shown in figure 8(f). We emphasize that our method does not require that the initial contour be placed close to the object boundary. In addition, observe how the front overshoots all the isolated spurious edges present inside the shape (see figure 7(b)) and settles in the neighborhood of edges which correspond to the true shape. This feature is a consequence of εK component in the speed which diffuses regions of high curvature on the front and forces it to attain a smooth shape.

As mentioned in section 3, smoothness of the front can be controlled by choosing an appropriate curvature component in the speed function $F = 1 - \varepsilon K$. The objective of our next experiment is to demonstrate smoothness control in the context of shape recovery. In figures 9(a) through 9(c), we show the results of applying our narrow-band shape recovery algorithm to an image consisting of three synthetic shapes. Initialization was performed by drawing a curve enclosing each one of the three shapes. We compute the signed distance function $\psi(x, y)$ from these curves. The level sets of ψ are then made to propagate with speed $F = \hat{k}_I(1.0 - \varepsilon K)$. First, as shown in figure 9(a), we



(a) Original image

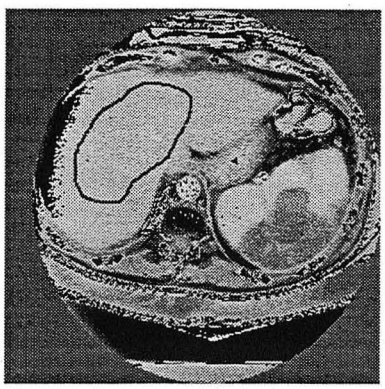
(b) Image-based speed term

Figure 7: Image-based speed term $k_I(x,y) = \frac{1}{1+|\nabla G_\sigma * I(x,y)|}$, with $\sigma = 3.25$, synthesized from the CT image.

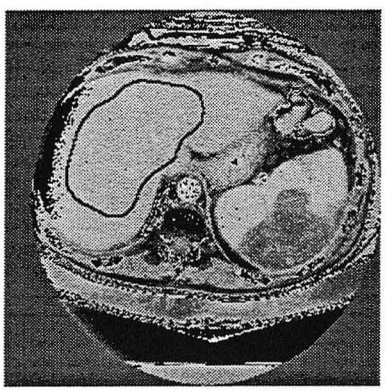
perform shape recovery with the value of $\varepsilon = 0.05$. The process is repeated with different values of ε; 0.25 in figure 9(b) and 0.75 in figure 9(c). Clearly, with every increment in the value of ε, the level set $\{\psi = 0\}$ attains a configuration that is relatively smoother. This is analogous to the smoothness provided by the second order term in the internal energy of a thin flexible rod [15].

In our third experiment we recover the complicated structure of an arterial tree. The real image has been obtained by clipping a portion of a digital subtraction angiogram. This is an example of a shape with extended branches or significant protrusions. In this experiment we compare the performance of our scheme with the active contour model. First, an attempt is made to reconstruct the arterial structure using a snake model with inflation forces [10]. In figures 10(a) through 10(i), we show a sequence of pictures depicting the snake configuration in the image. We present the final equilibrium state of the snake in figures 10(c), 10(f), & 10(i) corresponding to three distinct initializations, each better than the preceding one – in terms of the closeness to the desired final shape. In all three cases the active contour model, even after 1000 time iterations, barely recovers the main stem of the artery and completely fails to account for the branches. Due to the existence of multiple local minima in the (nonconvex) energy functional which the numerical procedure explicitly minimizes, the final result depends on the initial guess. Observe how in the third case, despite a

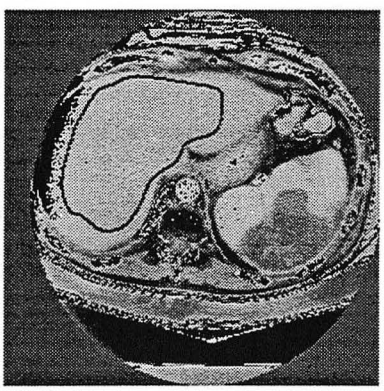
(a) $t = 0.0000$



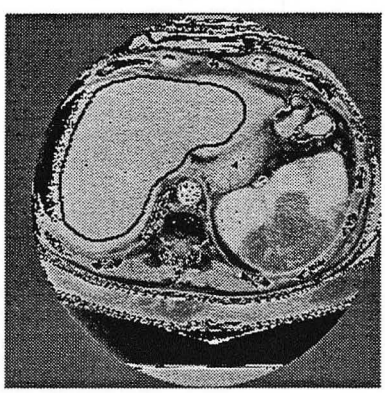
(b) $t = 0.0500$



(c) $t = 0.0875$



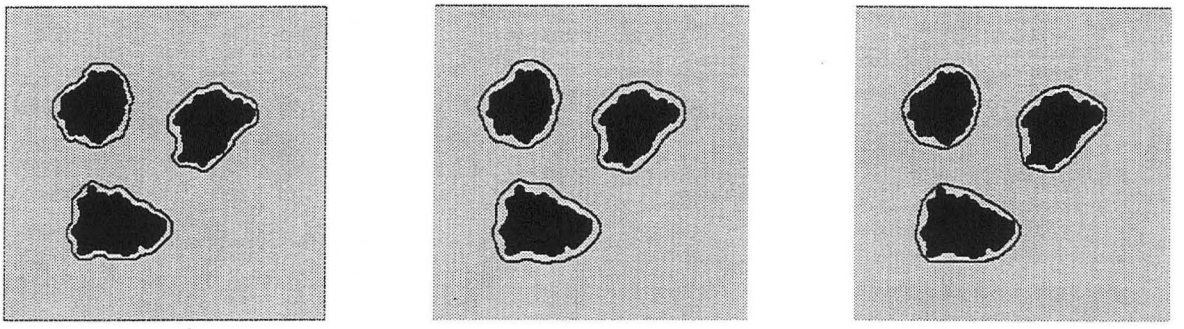
(d) $t = 0.1500$



(e) $t = 0.2250$

(f) $t = 0.2875$

Figure 8: Recovery of the stomach shape from a CT image of an abdominal section. Narrow-band computation was done on a 128×128 grid – the front was made to propagate with speed $F = \hat{k}_I(-1.0 - 0.025K)$ and the time step Δt was set to 0.0005. ψ was recomputed once every 50 time steps.



(a) $\epsilon = 0.05$ (b) $\epsilon = 0.25$ (c) $\epsilon = 0.75$

Figure 9: Smoothness control in shape recovery can be achieved by varying the curvature component in the speed $F = \hat{k}_I(1.0 - \epsilon K)$.

good initialization (figure 10(g)), the snake snaps back into a relatively bumpless configuration in figure 10(h). This is due to the snake's arc-length (elasticity) and curvature (rigidity) minimizing property. Snakes prefer regular shapes because shapes with protrusions have very high deformation energies. Note that it is important to maintain a balance between the image-based force and the inflation force. Therefore, we cannot increase the latter arbitrarily. One possible way to account for significant protrusions in a shape is via an adaptive resampling of the first order "balloon-snake" model. This however is a cumbersome solution to the problem.

Now, we apply our level set algorithm to reconstruct the same shape. After the initialization in figure 11(a), the front is made to propagate in the normal direction. We employ the narrow-band algorithm with a band width of $\delta = 0.045$ to move the front. It can be seen that in subsequent frames the front evolves into the branches and finally in 11(h) it completely reconstructs the complex tree structure. Thus, a single instance of our shape model sprouts branches and recovers all the connected components of a given shape. Calculations were carried out on a 128×128 grid and a time step $\Delta t = 0.00025$ was used. The plots of $\psi(\mathbf{x}, t = 0)$ and $\psi(\mathbf{x}, t = 0.375)$ are shown in figure 11(b) and 11(i) respectively.

In the next experiment, we depict a situation when the front undergoes a topological transformation to reconstruct the constituent shapes in an image. The image shown in figure 12(a) consists

of three distinct shapes. Initial curve is placed in such a way that it envelopes all the objects. The front is then advanced in the direction of the negative normal. Alternately, we could perform the initialization by placing a curve in each one of the individual shapes and propagating them in the normal direction. We choose the former option. The level set $\{\psi = 0\}$ first wraps itself tightly around the objects (see figures 12(d) – 12(f)). Subsequently it changes connectivity and splits twice – in figure 12(g) and figure 12(h) thereby recovering three shapes. Figure 12(i) shows the final result. Again it should be noted that a single instance of our shape model dynamically splits into three instances to represent each object. The function ψ was discretized on a 64×64 grid and Δt was set to 0.00025.

Next, we show that our approach can also be used to recover shapes with holes. The shapes in the figure (13) are examples of shapes with holes. The outer and inner boundaries of a given shape are recovered without requiring separate initializations. In figure 13(a), we show the initial contour which encloses both the shapes. This contour is then made to propagate inward with a constant speed. Figures 13(b)-13(d) are intermediate stages in the front evolution and in figure 13(e), it splits into two separate contours. The calculation comes to a halt when in figure 13(f), the level set $\{\psi = 0\}$ recovers the outer boundaries of two disconnected shapes. In the second stage of our computation, we treat the zero set configuration in figure 13(f) as an initial state, and propagate the front inward by momentarily relaxing the image-based speed term. This causes the zero set to move into the shapes as shown in figure 13(g), and recover the holes, thereby achieving a complete shape recovery (see 13(h)). The calculations for this experiment were done on a 128×128 grid and the time step Δt was set to 0.00025.

In our last experiment, we recover the shape of a flat superquadric using the level set front propagation scheme in 3D. Volume data for this experiment consists of 32 slices each with a particular cross section of the superquadric. The image-based speed term k_I is computed from these images according to an equation in 3D which is analogous to equation (14). A sphere, which is the level surface $\{\psi = 0\}$ of a function $\psi(x, y, z) = x^2 + y^2 + z^2 - 0.01$, forms our initialization (see figure 14(a)). This initial surface is moved with speed $F = \hat{k}_I$ by updating the value of ψ on a discrete 3D grid. The initial surface expands smoothly in all directions until a portion of it collides with the superquadric boundary. At points with high gradient, the $\hat{k}_I$ values are close to zero and

cause the zero set to locally come to stop near the boundary of the superquadric shape. This situation is depicted in figures 14(b)- 14(e), wherein the initial spherical shape transforms into a flat superquadric. Finally, in figure 14(f), all the points on our shape model are stopped, thereby recovering the entire shape of the flat superquadric. Calculations were done on a $32 \times 32 \times 32$ grid with a time step $\Delta t = 0.0025$.

6 Concluding Remarks

In this paper we have presented a new shape modeling scheme. Our approach retains some of the desirable features of existing methods for shape modeling and overcomes some of their deficiencies. We adopt the level set techniques first introduced in Osher and Sethian [23] to the problem of shape recovery. With this approach, complex shapes can be recovered from images. The final result in our method is relatively independent of the initial guess. This is a very desirable feature to have, specially in applications such as automatic shape recovery from image data. Moreover, our scheme makes no *a priori* assumption about the object's topology. Other salient features of our shape modeling scheme include its ability to split and merge freely without any additional bookkeeping during the evolutionary process, and its easy extensibility to higher dimensions. We believe that this shape modeling algorithm will have numerous applications in the areas of computer vision and computer graphics.

References

[1] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," submitted for publication, *Journal of Computational Physics*, 1994.

[2] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 9, pp. 855–867, 1990.

[3] J. Bence, B. Merriman, and S. Osher, "Motion of multiple triple junctions: A level set approach," to appear in *Journal of Computational Physics*, 1994.

[4] R. M. Bolle and B. C. Vemuri, "On three-dimensional surface reconstruction methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 13, No. 1, pp. 1–13, 1991.

[5] A. Bourlioux and J. A. Sethian, "Projection methods coupled to level set interface methods," to be submitted, *Journal of Computational Physics*, 1994.

[6] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA.

[7] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," Internal report no. 9210, CEREMADE, Université de Paris-Dauphine, France.

[8] D. L. Chopp, "Computing minimal surfaces via level set curvature flow," *Journal of Computational Physics*, Vol. 106, pp. 77–91, 1993.

[9] D. L. Chopp and J. A. Sethian, "Curvature flow and singularity development," submitted for publication in *Journal of Experimental Mathematics*, 1993.

[10] L. D. Cohen, "On active contour models and balloons," *Computer Vision, Graphics, and Image Processing*, Vol. 53, No. 2, pp. 211–218, March 1991.

[11] H. Delingette, M. Hebert, and K. Ikeuchi, "Shape representation and image segmentation using deformable models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 467–472, Maui Hawaii, June 1991.

[12] B. Engquist and S. Osher, "Stable and entropy satisfying approximations for transonic flow calculations," *Math. Comp.*, Vol. 34, 45, 1980.

[13] L. C. Evans and J. Spruck, "Motion of level sets by mean curvature. I," *Journal of Differential Geometry*, Vol. 33, pp. 635–681, 1991.

[14] W. T. Freeman and E. H. Adelson, "Steerable filters for early vision, image analysis, and wavelet decomposition," in *Proceedings of ICCV*, pp. 406–415, Osaka, Japan, 1990.

[15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, pp. 321–331, 1988.

[16] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Toward a computational theory of shape: An overview," in *Proceedings of ECCV*, Antibes, France, 1990.

[17] D. Lee and T. Pavlidis, "One-dimensional regularization with discontinuities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 10, pp. 822–829, 1986.

[18] R. Malladi, "Deformable models: Canonical parameters for surface representation and multiple view integration," Masters thesis, Dept. of CIS, University of Florida, Gainesville, May 1991.

[19] R. Malladi, "A topology-independent shape modeling scheme," Doctoral dissertation, Dept. of CIS, University of Florida, Gainesville, December 1993.

[20] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery," in *Proceedings of Third European Conference on Computer Vision*, LNCS Vol. 800, pp. 3–13, Stockholm, Sweden, May 1994.

[21] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," Center for Pure and Applied Mathematics, Report PAM-589, Univ. of California, Berkeley, August 1993.

[22] W. Mulder, S. Osher, and J. A. Sethian, "Computing interface motion in compressible gas dynamics," *Journal of Computational Physics*, Vol. 100(2), pp. 209–228, 1992.

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

[24] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, July 1991.

[25] C. Rhee, L. Talbot, and J. A. Sethian, "Dynamical behavior of a premixed turbulent open V-Flame," submitted for publication, *Journal of Fluid Mech.*, 1994.

[26] R. Samadani, "Changes in connectivity in active contour models," *Proceedings of the Workshop on Visual Motion*, pp. 337–343, Irvine California, March 1989.

[27] L.L. Schumaker, "Fitting surfaces to scattered data," in *Approximation Theory II*, G.G. Lorentz, C.K. Chui, and L.L. Schumaker, (eds.). New York: Academic Press, 1976, pp. 203–267.

[28] J. A. Sethian, "Curvature and the evolution of fronts," *Commun. in Mathematical Physics*, Vol. 101, pp. 487–499, 1985.

[29] J. A. Sethian, "Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws," *Journal of Differential Geometry*, Vol. 31, pp. 131–161, 1990.

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

[31] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," UCLA CAM Report 93-18, 1993.

[32] R. Szeliski and D. Tonnesen, "Surface modeling with oriented particle systems," *Computer Graphics SIGGRAPH*, Vol. 26, No. 2, pp. 185–194, July 1992.

[33] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 8, No. 2, pp. 413–424, 1986.

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

[35] D. Terzopoulos, "The computation of visible surface representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI 4, Vol. 10, pp. 417–438, 1988.

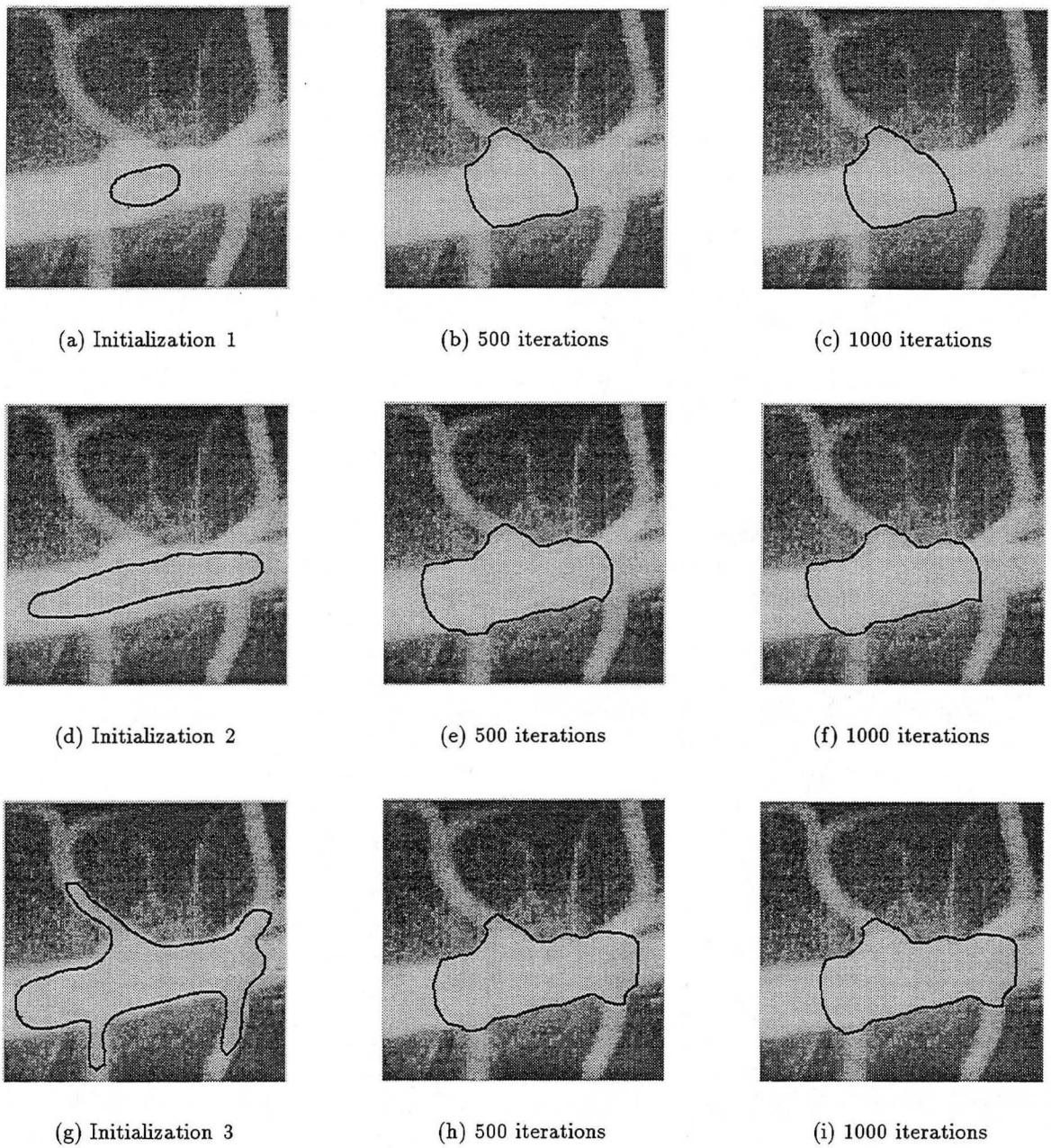
[36] B. C. Vemuri and R. Malladi, "Surface griding with intrinsic parameters," *Pattern Recognition Letters*, Vol. 13, No. 11, pp. 805–812, November 1992.

[37] B. C. Vemuri and R. Malladi, "Constructing intrinsic parameters with active models for invariant surface reconstruction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 7, pp. 668–681, July 1993.

[38] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Int. Journal of Image and Vision Computing*, 4, pp. 107–114, 1986.

[39] Y. F. Wang and J. F. Wang, "Surface reconstruction using deformable models with interior and boundary constraints," in *Proceedings of ICCV*, pp. 300–303, Osaka, Japan, 1990.

[40] J. Zhu and J. A. Sethian, "Projection methods coupled to level set interface techniques," *Journal of Computational Physics*, Vol. 102(1), pp. 128–138, 1992.

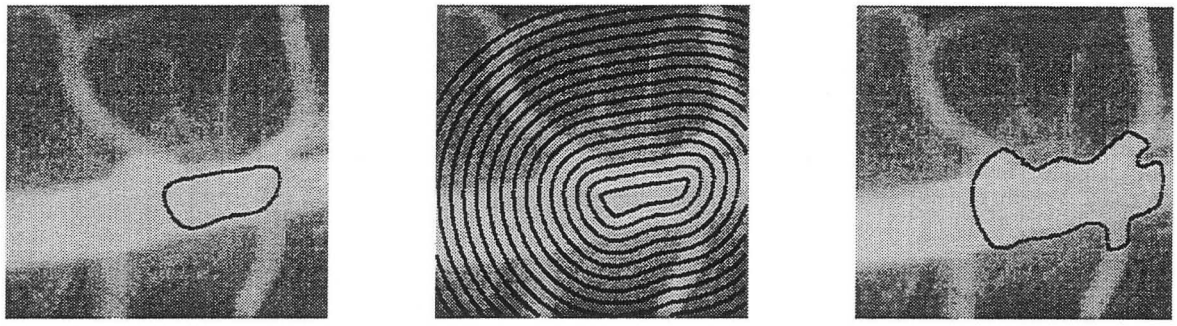


(a) Initialization 1 (b) 500 iterations (c) 1000 iterations

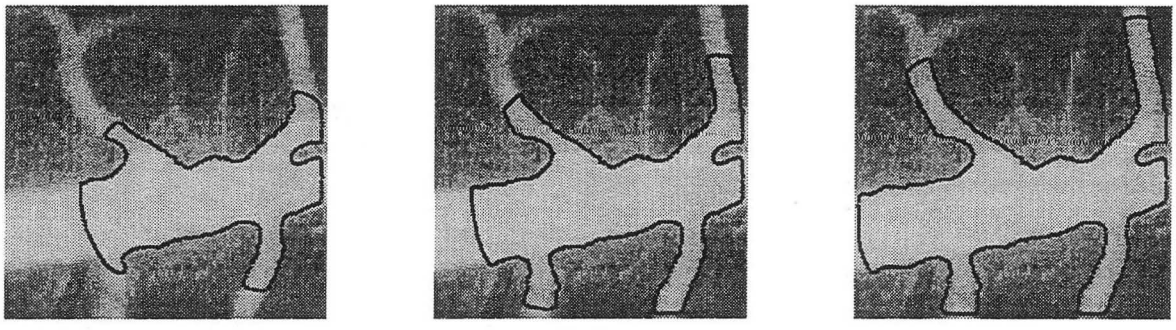
(d) Initialization 2 (e) 500 iterations (f) 1000 iterations

(g) Initialization 3 (h) 500 iterations (i) 1000 iterations

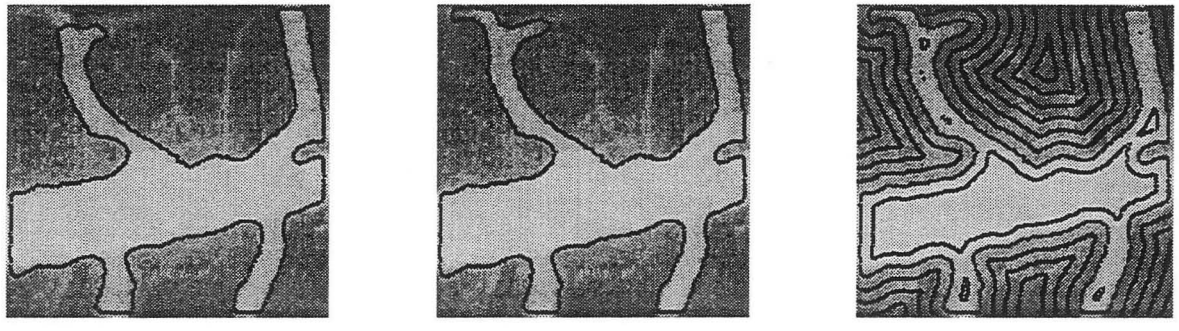
Figure 10: An unsuccessful attempt to reconstruct a complex shape with significant protrusions using an active contour model. Three different results are shown in parts (c), (f), & (i) corresponding to three distinct initializations in parts (a), (d), & (g) respectively. The following parameter values were employed in this experiment: γ (damping) = 1.0, Δt = 0.50, w_1 (elasticity) = 0.035, w_2 (rigidity) = 0.015, coefficient of inflation force = 0.50, and coefficient of image force = 2.50.



(a) $t = 0.0000$ (b) $\psi(\mathbf{x}, 0)$ (c) $t = 0.0625$

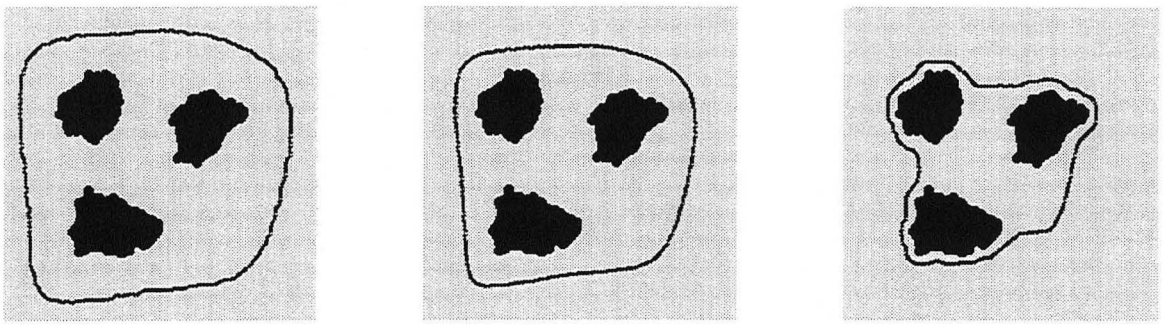


(d) $t = 0.1250$ (e) $t = 0.1875$ (f) $t = 0.2500$

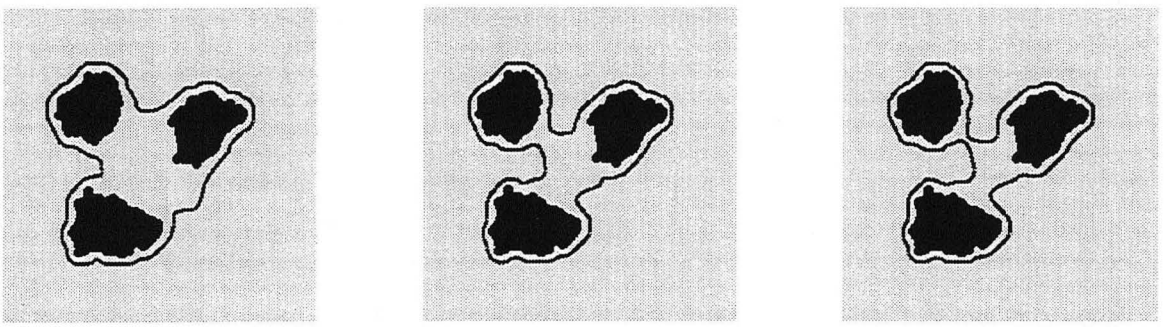


(g) $t = 0.3050$ (h) $t = 0.3750$ (i) $\psi(\mathbf{x}, 0.375)$

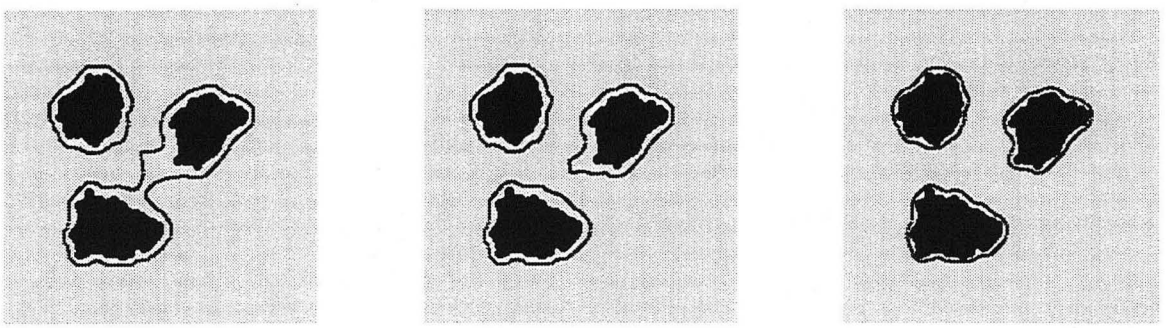
Figure 11: Reconstruction of a shape with significant protrusions: an arterial tree structure. Computation was done on a 128×128 grid with a time step $\Delta t = 0.00025$. The narrow-band algorithm was used with a band width of $\delta = 0.045$.



(a) $t = 0.0000$ (b) $t = 0.0250$ (c) $t = 0.0875$



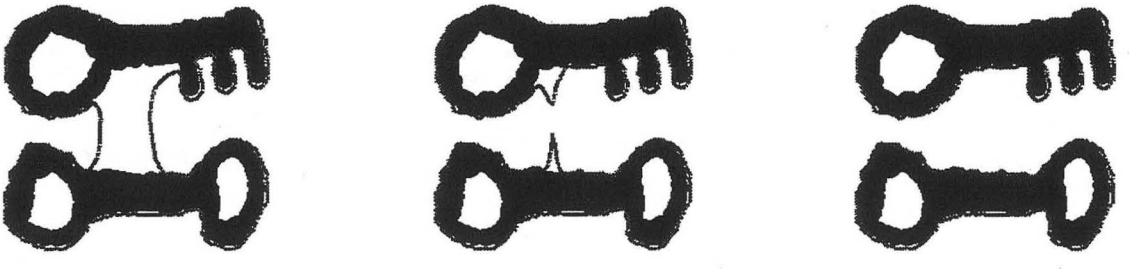
(d) $t = 0.1250$ (e) $t = 0.1625$ (f) $t = 0.1750$



(g) $t = 0.1875$ (h) $t = 0.2000$ (i) $t = 0.2500$

Figure 12: Topological split: a single instance of the shape model splits into three instances to reconstruct the individual shapes. Computation was done on a 64×64 mesh with a time step $\Delta t = 0.00025$.

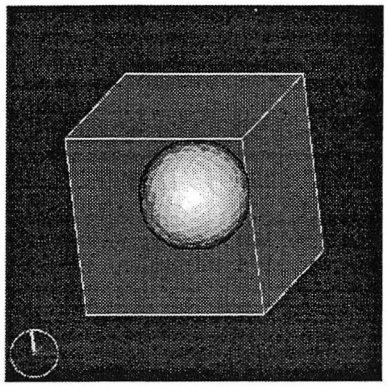
(a) $t = 0.0000$ (b) $t = 0.0500$ (c) $t = 0.1000$



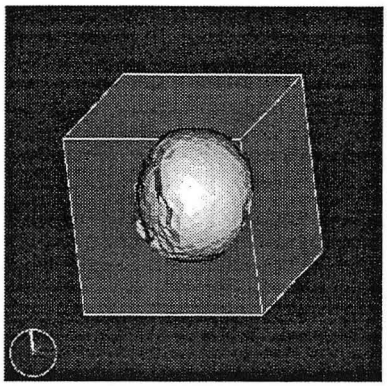
(d) $t = 0.1750$ (e) $t = 0.2137$ (f) $t = 0.2400$

(g) $t = 0.2500$ (h) $t = 0.2700$ (i) $t = 0.2950$

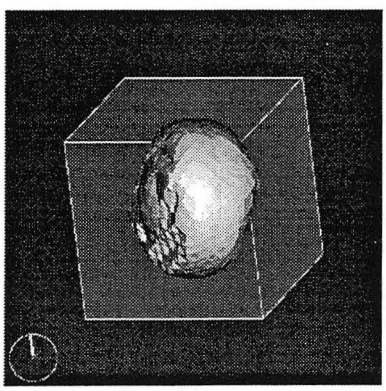
Figure 13: Shapes with holes: a two-stage scheme is used to arrive at a complete shape description of both simple shapes and shapes with holes. Computation was performed on 128×128 grid and the time step Δt was set to 0.00025.



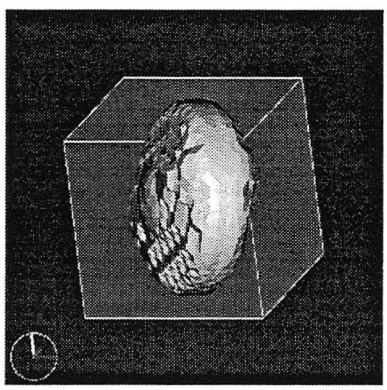
(a) $t = 0.0000$



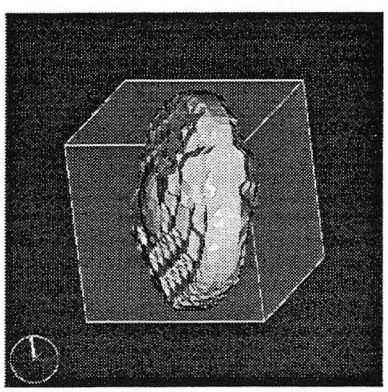
(b) $t = 0.0500$



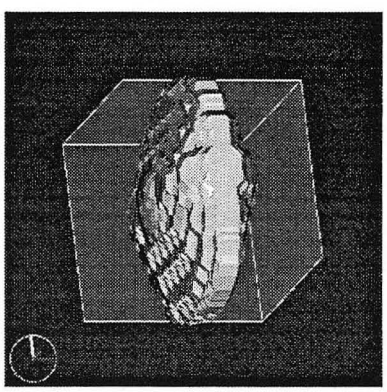
(c) $t = 0.1000$



(d) $t = 0.1750$



(e) $t = 0.2250$



(f) $t = 0.3000$

Figure 14: Shape recovery in 3D: a flat superquadric shape. Calculations were done on a $32 \times 32 \times 32$ grid with a time step $\Delta t = 0.0025$.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720