Shape Modeling with Front Propagation: A Level Set Approach *

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

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

[2]Department of Mathematics
University of California, Berkeley, CA 94720

Abstract

Developing shape models is an important aspect of computer vision research. Geometric and differential properties of the surface can be computed from shape models. They also aid the tasks of object representation and recognition. In this paper we present an innovative new approach for shape modeling which, while retaining important features of the existing methods, overcomes most of their limitations. Our technique can be applied to model arbitrarily complex shapes, shapes with protrusions, and to situations where no *a priori* assumption about the object's topology can be 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. Our method is based on the level set ideas developed by Osher & Sethian to follow 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. We move the interface by solving a "Hamilton-Jacobi" type equation written for a function in which the interface is a particular level set. A speed function synthesized from the image is used to stop the interface in the vicinity of the object boundaries. The resulting equations of motion are solved by numerical techniques borrowed from the technology of hyperbolic conservation laws. An added advantage of this scheme is that it can easily be extended to any number of space dimensions. The efficacy of the scheme is demonstrated with numerical experiments on synthesized images and noisy medical images.

*Submitted for publication in the *IEEE Trans. on Pattern Analysis & Machine Intelligence.*

1 Introduction

An important goal of computational vision is to recover the shapes of 2D and 3D objects from various types of visual data. To achieve this goal, shape models that satisfy constraints imposed by sensory data must be synthesized. Shape models aid the computation of certain geometric and differential properties of surfaces. They also serve the purpose of an intermediate stage in object recognition tasks, since they provide a more stable and useful description than the original intensity images. In this paper we present a new approach to shape modeling which, while retaining important features of the existing methods, overcomes most of their limitations.

Shape reconstruction typically precedes the symbolic representation of surfaces. The shape models must recover detailed structure from noisy data using only the weakest among the possible assumptions about the observed shape. Several variational reconstruction methods have been proposed and there is abundant literature on the same [2, 17, 23, 3, 24, 11]. Generalized spline models with continuity constraints are well suited for fulfilling the goals of shape reconstruction (see [3, 4, 21]). Generalized splines are the key ingredient of the dynamic shape modeling paradigm introduced by Terzopoulos *et al.*, [22]. Incorporating dynamics into shape modeling enables the creation of realistic animation in computer graphics applications and for tracking moving objects in computer vision. Following the advent of the dynamic shape modeling paradigm, there was a flurry of research activity in the area, with numerous application specific modifications to the modeling primitives, and external forces derived from data constraints [9, 25, 26, 27, 28, 6, 7]. However, the aforementioned schemes for shape modeling have two serious limitations – the dependence of the final surface shape on the initial guess made to start the numerical reconstruction procedure, and a strong assumption on the object's topology. The first of these deficiencies stems from the fact that the nonconvex energy functionals used in the variational formulations have multiple local minima. As a consequence of this feature, the numerical procedures, for convergence to a satisfactory solution require an initial guess which is "reasonably" close to the desired shape. Existing shape representation schemes have an additional shortcoming in that they

lack the ability to dynamically sense the topological changes during the shape reconstruction process. Our method, which we shall describe presently, makes no assumption about the object's topology, and it leads to a numerical algorithm whose convergence to the desired shape is completely independent of the shape initialization.

The framework of energy minimization has also been used successfully in the problem domain of extracting salient image contours – edges and lines. Kass *et al.* [9] used energy-minimizing "snakes" that are attracted to the image features such as edges points and edge segments, whereas internal spline forces impose a smoothness constraint. The weights of the smoothness and image force terms in the energy functional can be adjusted for different kinds of behavior. Snakes, also referred to as active contour models, are restricted examples of the more general techniques of matching deformable models to image data by means of energy minimization [22]. The scheme seeks to design energy functionals whose local minima comprise the set of alternative solutions available to high-level processes. In the absence of a well-developed high-level mechanism to make a choice among these solutions, an interactive approach is used to explore the alternatives. By adding suitable energy terms to the minimization, the user pushes the model out of a local minimum toward the desired solution. In the problem area of automatic segmentation of noisy images, snakes perform poorly unless they are placed close to the preferred shapes. In a move to make the final result relatively insensitive to the initial conditions, Cohen [5] defines an inflation force on the active contour. This new force makes the model behave like an inflating balloon. The contour model with the above change will be stopped by a strong edge and will simply pass through a spurious edge which is too weak relative to the ambient inflation force.

Although the inflation force prevents the curve from getting "trapped" by isolated spurious edges, the active contour model cannot segment complex shapes with significant protrusions like the one shown in figure (1). Moreover, despite a good initialization, the active contour model, due to its arc-length and curvature minimization properties, cannot be forced to extrude through any significant protrusions that a shape may posses. One possible solution to this problem is to embed the snake model, which is an instance of a 1D thin-plate-membrane-spline, in an adaptive environment wherein the material parameters controlling

Figure 1: Digital subtraction angiogram of an arterial structure

the relative strengths of elasticity and rigidity are adapted (see [16]). The merits of such an approach are suspect since it is not always possible to derive criteria upon which to base the adaptation algorithm. So the problem is one of accurately modeling bifurcations and protrusions in complex structures. In [22] it has been shown that multiple instances of deformable models are required to handle shapes with several distinct parts. This can be very cumbersome, for it involves excessive user interaction and presumes that the shape has already been deciphered into its constituent parts. Instead, we propose a method that will start with a single instance of the model and will *sprout* the branches during the evolutionary process. Once the shape has been segmented from the image, its constituent part structure can be inferred [10].

Most existing surface modeling techniques require that the topology of the object be known before the reconstruction can commence. However, it is not always possible to specify the topology of an object prior to its reconstruction. As a result, most reconstruction schemes end up making strong assumptions about object topology. Vision systems which derive quantitative models of complex object shapes by integrating different visual modalities cannot evade the issue of unknown and unpredictable topologies. Unknown topology is also an

important concern in object tracking and motion detection applications where the positions of object boundaries are tracked in an image sequence through time. During their evolution, these closed contours may change connectivity and split, thereby undergoing a topological transformation. A heuristic criterion for splitting and merging which is based on monitoring deformation energies of points on the elastic curve has been discussed in [15]. More recently, molecular dynamics has been used to model surfaces of arbitrary topology [20]. Smoothness and continuity constraints are imposed by subjecting a particle system to interaction potentials which locally prefer planar or spherical arrangement. Particles can be added and deleted dynamically to enlarge and trim the surface respectively, while the system dynamics strive continually to organize the particles into smooth shapes. The result is a versatile method with applications in surface fitting to sparse data and 3D medical image segmentation.

The scheme described in this paper can be applied to situations where no *a priori* assumption about the object's topology can be 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 [12].

1.1 Overview

In this subsection we briefly outline the scheme we use to model complex shapes. Our method is inspired by ideas first introduced in Osher & Sethian [13, 19] to follow 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. Sethian [18] has shown that direct parameterization of the moving front may be unstable since it relies on local properties of the solution. In contrast, a method which preserves the global properties of the motion is sought. Osher and Sethian [13, 19] achieve this by embedding the surface in a higher-dimensional function. The equation of motion for this function is reminiscent of an initial valued 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 problems of shape reconstruction. To isolate a shape from its background, we first consider a closed, nonintersecting, initial hypersurface placed inside it. Following the level set approach above, this hypersurface is then made to *flow* along its gradient field with some speed $F(K)$, where K is the curvature of the hypersurface. As in [13], we adopt a global approach and view the $(N-1)$ dimensional moving surface *as a level set of a time-dependent function* ψ of N space dimensions. The equations of motion written for this higher dimensional function are then amenable to stable *entropy-satisfying* numerical schemes designed to approximate hyperbolic conservation laws. Topological changes can be handled naturally in this approach, since a particular level set $\{\psi = 0\}$ of the function ψ need not be simply connected. However, there are two problems that need to be surmounted before we can use this design for shape reconstruction. First, it is required that we stop the hypersurface in the neighborhood of the desired shape. We do this by synthesizing a *negative speed* function from the image. Secondly, we have to construct an extension of this speed function to other level sets $\{\psi = C\}$ in the image (see figure 2). In the following sections we outline a possible solution to these problems.

We note that this work on interface motion and hyperbolic conservation laws as discussed in [13, 18, 19], has been applied in the area of computer vision for shape characterization by Kimia *et al.* [10], who unify many diverse aspects of shape by defining a continuum of shapes (reaction/diffusion space), which places shapes within a neighborhood of other similar shapes. This leads to a hierarchical description of a shape which is suitable for its recognition. The key distinguishing feature of our work from that of Kimia *et al.*, is that they assume the object shape to be known, while we reconstruct it from noisy data. In other words, they show that by *evolving a known shape boundary*, explicit clues can be derived towards the goal of developing a hierarchical shape description. In contrast, *we start with an arbitrary function* ψ *and recover complex shapes* by propagating it along its gradient field. Shape characterization can be readily done once the object shape is extracted.

In summary, we present a novel scheme for shape modeling which can be used in both computer vision and computer graphics applications. Given the reconstructed shape, our

approach can also be used for deciphering the constituent part structure. The remainder of this paper is organized as follows: section 2 introduces the curvature-dependent front propagation problem and establishes a link between Hamilton-Jacobi equations and a hyperbolic conservation law. In section 3 we explain our level set algorithm for shape reconstruction and section 4 presents some experimental results of applying our method to some synthesized and real noisy images. We close with a discussion of advantages of our approach and direction of future research in section 5.

2 Front propagation problem

In this section we present the level set technique due to Osher and Sethian [13]. For details and an expository review, see Sethian [19]. 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 R^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 one time step. However, there are several problems with this approach, as discussed in Sethian [18]. 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

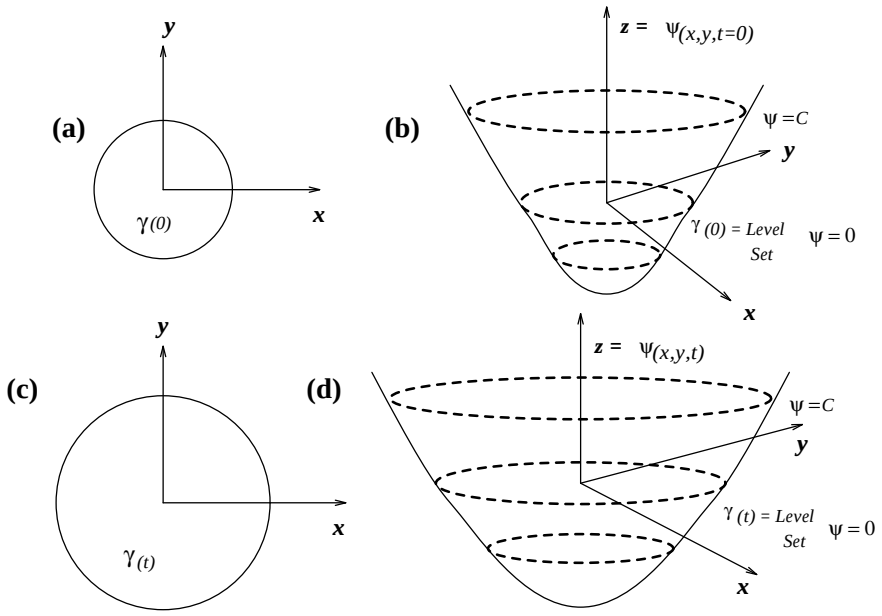


Figure 2: Level set formulation of equations of motion – (a) & (b) show the curve γ and the surface $\psi(x, y)$ at $t = 0$, and (c) & (d) show the curve γ and the corresponding surface $\psi(x, y)$ at time t.

dimensions.

As an alternative, the central idea in the level set approach of Osher and Sethian [13] is to represent the front $\gamma(t)$ as the level set $\{\psi = 0\}$ of a function ψ. To motivate this approach, we 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 R^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 (figures 2(c) & 2(d)).

In the general case, let $\gamma(0)$ be a closed, nonintersecting, $(N-1)$ dimensional hypersurface. Let $\psi(\mathbf{x}, t)$, $\mathbf{x} \in R^N$, be the scalar function such that $\psi(\mathbf{x}, 0) = \pm d(\mathbf{x})$, where $d(\mathbf{x})$ is the signed distance from $\mathbf{x}$ to the hypersurface $\gamma(0)$. We use the plus sign if $\mathbf{x}$ is outside $\gamma(0)$ and minus sign if $\mathbf{x}$ is inside. Each level set of ψ flows along its gradient field with speed $F(K)$. The gradient $\nabla\psi(\mathbf{x}, t)$ is normal to the $(N-1)$ dimensional level set passing through $\mathbf{x}$. Now, we derive the equation of motion for function ψ.

Consider the motion of some level set $\{\psi = C\}$. Following the derivation in [13], let $\mathbf{x}(t)$

be the trajectory of a particle located on this level set, so

$$\psi(\mathbf{x}(t), t) = C. \tag{1}$$

The particle speed $\partial\mathbf{x}/\partial t$ in the direction $\mathbf{n}$ normal to $\gamma(t)$ is given by the speed function F. Thus,

$$\frac{\partial \mathbf{x}}{\partial t} \cdot \mathbf{n} = F, \tag{2}$$

where the normal vector $\mathbf{n}$ is given by $\mathbf{n} = \nabla\psi / \mid \nabla\psi \mid$. By the chain rule we get,

$$\psi_t + \frac{\partial \mathbf{x}}{\partial t} \cdot \nabla\psi = 0 \tag{3}$$

and substitution yields

$$\psi_t + F \mid \nabla\psi \mid = 0, \tag{4}$$

with an initial condition $\psi(\mathbf{x}, 0) = \pm d(\mathbf{x})$. We refer to equation (4) as the level set "Hamilton-Jacobi" formulation. Note that at any time, the moving front $\gamma(t)$ is simply the level set $\{\psi(\mathbf{x}, t) = 0\}$. There are several advantages to this approach. First, since the underlying coordinate system is fixed, discrete mesh points used in the numerical update equations do not move, resulting in a stable computation. Topological changes in the front can be handled naturally by exploiting the property that the level surface $\{\psi = 0\}$ need not be simply connected. $\psi(\mathbf{x}, t)$ always remains a function, even if the level surface $\{\psi = 0\}$ corresponding to the front $\gamma(t)$ changes topology, or forms sharp corners. The geometric and differential properties of $\gamma(t)$ are captured in the function ψ and can be readily extracted. As an example, if $\mathbf{x} \in R^2$, the curvature is given by

$$K = \frac{(\psi_{yy}\psi_x^2 - 2\psi_x\psi_y\psi_{xy} + \psi_{xx}\psi_y^2)}{(\psi_x^2 + \psi_y^2)^{3/2}}. \tag{5}$$

This approach can also be easily extended to higher dimensions and appropriate expressions can be obtained for the mean curvature and the Gaussian curvature [13].

By substituting $F(K) = 1 - \varepsilon K$ as a typical speed function in equation (4), the equation of motion becomes

$$\psi_t + \mid \nabla\psi \mid = \varepsilon K \mid \nabla\psi \mid . \tag{6}$$

Equation (6) resembles a Hamilton-Jacobi equation with viscosity, where "viscosity" refers to the second-order parabolic right-hand side. This equation can be solved using the stable, entropy-satisfying finite difference schemes, borrowed from the literature on hyperbolic conservation laws (see [13]).

3 Shape reconstruction 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 reconstruction. There is a fundamental difference between the problems of front propagation and shape reconstruction. In the former, the front represents a solid/liquid interface (crystal growth) or a boundary separating burnt and unburnt regions (flame propagation). In these cases the computation is alive as long as there remains a physical domain into which the front can be moved. For example, the flame front can be moved as long as there is a region to be burnt and it hasn't crossed the physical domain in which the solution is sought. On the contrary, in shape reconstruction the front represents the boundary of an evolving shape. Since the idea is to extract object shapes from a given image, the front should be forced to stop in the vicinity of the desired object boundaries. This is analogous to the force criterion used to push the active contour model towards desired shapes. 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 stopping criterion. In general the function F can be split 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 [5]. 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 [9]. We rewrite equation (6) by splitting the influence of F as

$$\psi_t + F_A \mid \nabla\psi \mid + F_G \mid \nabla\psi \mid = 0. \tag{7}$$

First consider the case when the front moves with a constant speed, i.e. $F = F_A$. To this if we add a *negative speed term* synthesized from the image, such that their sum tends to zero near large image gradient locations, we will achieve our goal of bringing the front to a stop in the neighborhood of object boundaries. To this end, we define a negative speed F_I to be

$$F_I(x,y) = \frac{-F_A}{(M_1 - M_2)} \left\{ \mid \nabla G_\sigma * I(x,y) \mid - M_2 \right\}, \tag{8}$$

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 equation of motion with the addition of image-based speed is

$$\psi_t + (F_A + \hat{F}_I) \mid \nabla\psi \mid = 0. \tag{9}$$

$\hat{F}_I$ is called an *extension* of F_I to points away from the boundary $\gamma(t)$, i.e. at points $(x,y) \in (\Omega - \gamma(t))$, and is equal to F_I on $\gamma(t)$. We shall return to the issue of extension shortly. 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 the front gradually attains zero speed as it gets closer to the object boundaries and eventually comes to a stop.

In the case when the front moves with a speed that is a function of local curvature, i.e. $F_G \neq 0$, 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

$$k_I(x,y) = \frac{1}{1 + \mid \nabla G_\sigma * I(x,y) \mid}, \tag{10}$$

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. The modified equation of motion is given by

$$\psi_t + \hat{k}_I(F_A + F_G) \mid \nabla\psi \mid = 0. \tag{11}$$

We now come to an important juncture in our discussion. The image-based speed term, be it F_I or k_I, has 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 equation of motion (9) is written for the function ψ, which is made up of infinitely many level curves. In other words, equations (9) & (11) control the evolution of a family of level sets. Therefore, it is imperative that the net speed used in the evolution equation has a consistent physical meaning for all the level sets, i.e. at every point $(x, y) \in \Omega$. Speed functions such as F_G which are functions of geometric properties of the surface $z = \psi(x, y)$, can be readily computed at any $(x, y) \in \Omega$. However, F_I is not such a function. It 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. 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.

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 should consist of only the set of points located a distance t away from the initial front. On the other hand, if $F_G \neq 0$, the level sets will violate the property that they are a constant distance away from each other. However, they will never collide and cross each other if the speed function $F = F_A + F_G$ is continuous (see [8]). With the above remarks in mind we state the following:

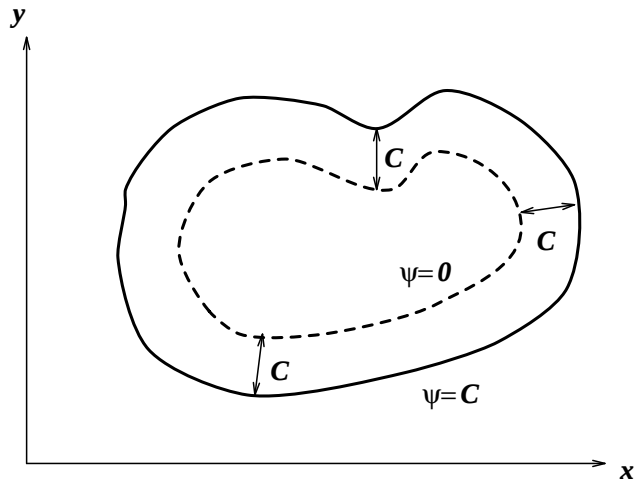


Figure 3: Huygen's principle construction

Property 1 *Any external (image-based) speed function that is used in the equation of motion written for the function ψ should not cause the level sets to collide and cross each other during the evolutionary process.*

Recall that the function $\psi(\mathbf{x}, t)$ has been initialized to $d(\mathbf{x})$, where $d(\mathbf{x})$ is the signed distance from a point $\mathbf{x} \in \Omega$ to the boundary $\gamma(0)$. Since we cannot attribute any geometric meaning to the function F_I (k_I) at points away from the level set $\{\psi = 0\}$, we look for a meaning that is consistent with property (1). Therefore, the question to ask is: what is the value of $\hat{F}_I$ (or $\hat{k}_I$) at a point (x, y) lying on a level set $\{\psi = C\}$? We answer this question in the following construction (see figure 4).

Construction 1 *The value of $\hat{F}_I$ ($\hat{k}_I$) at a point P lying on a level set $\{\psi = C\}$ is exactly the value of F_I (k_I) at a point Q, such that point Q is a distance C away from P and lies on the level set $\{\psi = 0\}$.*

It is easy to see that $\hat{F}_I$ reduces to F_I on $\{\psi = 0\}$. We use the same construction to determine the value of $\hat{k}_I$ at a point (x, y) lying on some level set $\{\psi = C\}$. Note that if the definition of a speed function adheres to construction 1, then it will also be consistent with the property 1. Thus, having ascribed the intent of pseudodifferential equations (9) & (11) in the context of shape modeling, we can use finite difference schemes to solve them

numerically. Since ψ can develop corners and sharp gradients, numerical schemes borrowed from hyperbolic conservation laws are used to produce stable upwind schemes. Moreover, the equations of motion can be solved on a uniform mesh and the level sets can be moved without their explicit construction.

4 Numerical solution and experimental results

In this section, almost without a change, we present the arguments discussed in Sethian [19]. For complete details of the following scheme, we refer the reader to Osher & Sethian [13, 19].

The equation (6) poses an initial valued problem. It is rewritten here as

$$\psi_t + (\psi_x^2 + \psi_y^2)^{1/2} = \varepsilon \nabla \cdot \left(\frac{\nabla \psi}{\mid \nabla \psi \mid} \right) \tag{12}$$

with $\psi(x, y, t = 0) = \pm$ distance from (x, y) to $\gamma(0)$. As shown in Sethian [18], for $\varepsilon > 0$, the parabolic right-hand side diffuses sharp gradients and forces ψ to stay smooth at all values of t. For $\varepsilon = 0$, the boundary moves with unit speed, and a corner must develop from smooth initial data. Once a corner develops, it is not clear how to propagate the front in the normal direction, since the derivative is not defined at the corner. A variety of "weak" solutions which propagate the curve beyond the occurrence of a singularity are possible. Of all such weak solutions, one is interested in the one that is the limit of smooth solutions as $\varepsilon \to 0$. This particular weak solution can be selected with the help of a so-called "entropy condition", see [18]: If the front is viewed as a burning flame, then *once a particle is burnt it stays burnt.* Thus, approximations to the spatial derivative are sought that do not artificially smooth sharp corners and which pick out the correct entropy solution when singularities develop. The schemes given in [13, 19] are motivated by the fact that the entropy condition for propagating fronts is identical to the one for hyperbolic conservation laws, where stable, consistent, entropy-satisfying algorithms have a rich history.

First, consider the one-dimensional version of the level set equation, with constant normal velocity $F_A = 1$. Then the equation (6) becomes a standard Hamilton-Jacobi equation

$$\phi_t + H(\phi_x) = 0, \tag{13}$$

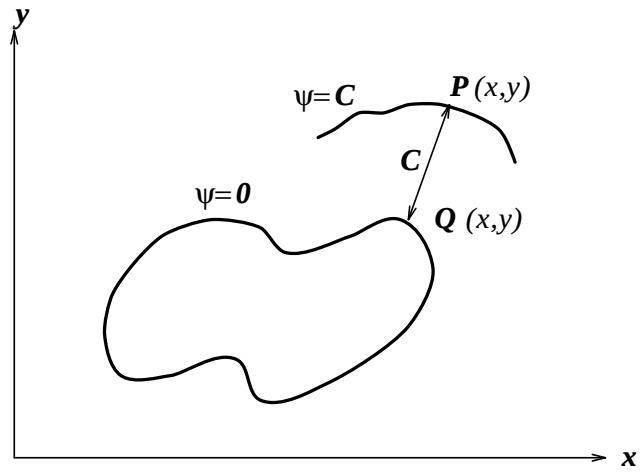


Figure 4: Extension of image-based speed terms to other level sets

where $H(\phi_x) = -(\phi_x^2)^{1/2}$, and with a given initial value of ϕ. Let $u = \phi_x$. Taking the derivative with respect to x, equation (13) becomes

$$u_t + [H(u)]_x = 0, \tag{14}$$

where $H(u) = -(u^2)^{1/2}$. Equation (14) is a scalar hyperbolic conservation law in one space variable. Solutions can develop discontinuous jumps, known as *shocks*, even with smooth initial data. In order to make sense of the solution after shocks form, an integral version of the conservation law which admits discontinuous solutions is studied. Both sides of equation (14) are integrated in an arbitrary interval $[a, b]$ to produce

$$\frac{d}{dt}\int_a^b u(x,t)dt = H[u(a,t)] - H[u(b,t)]. \tag{15}$$

u is known as a *weak solution* of the conservation law if it satisfies the above integral equation. Note that u need not be differentiable to satisfy the integral form of the conservation law.

When will a numerical algorithm approximate the correct, entropy-satisfying solution to equation (15)? The answer is found in the following definition:

Definition 1 *Let u_i^n be the value of u at a mesh point $i\Delta x$ at time $n\Delta t$. A three-point difference scheme is said to be in conservation form if there exists a function $g(u_1, u_2)$ such that the scheme can be written in the form*

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{g(u_i^n, u_{i+1}^n) - g(u_{i-1}^n, u_i^n)}{\Delta x}, \tag{16}$$

where $g(u,u) = H(u)$.

This definition is natural; the scheme must approximate the hyperbolic conservation law, subject to the consistency requirement $g(u,u) = H(u)$. In order to guarantee that the scheme picks out the correct entropy-satisfying weak solution, *monotonicity* is required, i.e., that u_i^{n+1} be an increasing function of the arguments u_{i-1}^n, u_i^n, and u_{i+1}^n. The main fact is: *A conservative, monotone scheme produces a solution that satisfies the entropy condition.* Equation (16) is a scheme for the slope u, which must be converted into a scheme for ϕ itself. First write equation (13) with a forward difference in time as,

$$\phi_i^{n+1} = \phi_i^n - \Delta t H(u). \tag{17}$$

Since the numerical flux function g approximates H, the solution to equation (17) may be approximated by

$$\begin{aligned} \phi_i^{n+1} &= \phi_i^n - \Delta t g(u_{i-1/2}, u_{i+1/2}) \\ &= \phi_i^n - \Delta t g(D_x^- \phi_i, D_x^+ \phi_i), \end{aligned} \tag{18}$$

where g is an appropriate numerical flux function and the standard definitions of the forward and the backward difference operators have been used, namely,

$$\begin{aligned} D_x^- \phi_i &= \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}, \\ D_x^+ \phi_i &= \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}. \end{aligned} \tag{19}$$

Finally, an appropriate numerical flux function g is required. In the special case where $H(u)$ may be written as a function of u^2, i.e., $H(u) = f(u^2)$ for some function f, one can use the Hamilton-Jacobi flux function given in [13]:

$$\begin{aligned} g(u_{i-1/2}, u_{i+1/2}) &= g(D_x^- \phi_i, D_x^+ \phi_i) \\ &= f((\max(D_x^- \phi, 0))^2 + (\min(D_x^+ \phi, 0))^2). \end{aligned} \tag{20}$$

This conservative monotone scheme is *upwind* method in that it differences in the direction of propagating characteristics. This is important, since it imposes boundary conditions on the

walls of a finite-sized computational box. An upwind scheme automatically differences in the outward-flowing direction at the box walls if the boundary is expanding, thus information flows out. In the case when $F_A = 1$, so that $f(u^2) = -(u^2)^{1/2}$, equation (18) reduces to

$$\phi_i^{n+1} = \phi_i^n - \Delta t\{(\max(D_x^- \phi, 0))^2 + (\min(D_x^+ \phi, 0))^2\}^{1/2}. \tag{21}$$

This algorithm produces the correct entropy-satisfying weak solution to the propagating front problem.

The above discussion can be extended to problems in more than one space dimension (see Osher & Sethian [13]). Recall that the original intent was to solve equations (9) and (11). In two dimensions, the scheme given in equation (21) is extended by differencing in each direction to produce the following numerical scheme for equation (9):

$$\begin{aligned} \psi_{i,j}^{n+1} &= \psi_{i,j}^n - \Delta t[F_A + (\hat{F}_I)_{i,j}]\{(\max(D_x^- \psi_{i,j}, 0))^2 + (\min(D_x^+ \psi_{i,j}, 0))^2 \\ &\quad +(\max(D_y^- \psi_{i,j}, 0))^2 + (\min(D_y^+ \psi_{i,j}, 0))^2\}^{1/2}. \end{aligned} \tag{22}$$

Similarly, the numerical scheme for equation (11) is,

$$\begin{aligned} \psi_{i,j}^{n+1} &= \psi_{i,j}^n - \Delta t F_A(\hat{k}_I)_{i,j}\{(\max(D_x^- \psi_{i,j}, 0))^2 + (\min(D_x^+ \psi_{i,j}, 0))^2 \\ &\quad +(\max(D_y^- \psi_{i,j}, 0))^2 + (\min(D_y^+ \psi_{i,j}, 0))^2\}^{1/2} - \Delta t F_G \hat{k}_I \mid \nabla \psi \mid . \end{aligned} \tag{23}$$

The second term $F_G \hat{k}_I \mid \nabla \psi \mid$ is not approximated in the above equation; one may use a straightforward central difference approximation to this term.

4.1 Experimental results

In this section we present several shape reconstruction results that were obtained by applying the level set algorithm to image data. We also outline some of the implementation details of our algorithm. Given an image, our method requires the user to provide an initial contour $\gamma(0)$. As we shall see, there is absolutely no restriction on where the initial contour can be placed in the image plane as long as it is inside a desired shape or encloses all the constituent shapes. This feature is of paramount importance in the context of automatic shape recovery. Our front seeks the object boundaries by either propagating outward in the normal direction

or propagating inward in the negative 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.

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. In our implementation, since we usually work with $2^k X 2^k$ images, the computational domain is a square one with $\Delta x = \Delta y = h$. 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 the update equations from the previous section to move the front.

It should be observed that by updating the level set function on a grid, we are moving the level sets without constructing them explicitly. To find the position of the front and to compute the image-based speed terms, at each time step we need to find the level set $\{\psi = 0\}$. We construct a piecewise linear approximation for $\gamma(t)$ as follows. Given a cell (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 that cell cannot contain $\gamma(t)$, so we ignore that cell. Otherwise, we find the entrance and exit points where $\psi = 0$ by linear interpolation. This provides two nodes on $\gamma(t)$ and thus one of the line segments which form our approximation to $\gamma(t)$. The collection of all such line segments constitutes our approximation to the level set $\{\psi = 0\}$, which is used for future evaluation of the image-based speed term in the update equation (23). These line segments are also used to display the current position of the front in the image plane.

The stability requirement for the explicit method for solving our level set equation is $\Delta t = O(\Delta x^2)$ for the equation (23). If $F_G = 0$, then the stability requirement is $\Delta t = O(\Delta x)$. This could potentially force a very small time step for fine grids making the computation excruciatingly slow. Since the algorithm to extend the image-based speed terms to other level sets is relatively more expensive than the front propagation algorithm, we could improve the performance by evaluating the extension only once every k iterations. In other words,

we take k steps in time without recomputing the force field $(\hat{k}_I)_{i,j}$. Alternately, we could down-sample the image and perform our calculations at a lower resolution. In this approach we run the risk of losing accuracy. However, we show that the results obtained with down-sampled images are very promising. We can get the best of both worlds by embedding our level set algorithm in a multiresolution framework. We refrain from continuing this line of discussion any further since the details of such schemes are beyond the scope of this paper.

We now present our shape reconstruction results in 2D. We first consider a 256 X 256 image with a single shape. The function ψ has been discretized on a 64 X 64 mesh, i.e. calculations are performed at every fourth pixel. In figure 5(a), we show the closed contour that the user places around the shape at time $t = 0$. The function ψ is then made to propagate in the negative normal direction. Figures 5(b) & 5(c) depict the configuration of the level set $\{\psi = 0\}$ at two intermediate time instants. The final result is achieved after 80 time iterations and is shown in figure 5(d). It should be noted that *our method does not require that the initial contour be placed close to the object boundary.* In order to quantize the loss in accuracy which results at lower resolutions, we perform the same reconstruction on a 256 X 256 mesh. The result is shown in 5(e). In figure 5(e), we also plot the level sets $\{\psi = \pm C\}$ to verify that the image-based speed term does not violate the property 1.

In our second 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 branches and significant protrusions. In this experiment we compare the performance of our scheme with the active contour model and bring its limitations into focus. We first attempt to reconstruct the complex arterial structure using a snake model with inflation forces. In figures 6(a) through 6(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 6(c), 6(f), & 6(i) corresponding to three distinct initializations, one better than the preceding. 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. Two prominent limitations of the snake model immediately come into light. The first is the dependence of final result on the initial configuration. This is due to the existence of multiple

local minima in the (nonconvex) energy functional which the numerical procedure explicitly minimizes. The second feature is the inability of snake model to attain a stable shape with protrusions. Observe how in the third case, despite a good initialization (figure 6(g)), the snake "snaps" back into a relatively "bumpless" configuration in figure 6(h). This inadequacy stems from snake's arc-length (elasticity) and curvature (rigidity) minimizing nature. Snake prefers regular shapes because shapes with protrusions have very high deformation energies. Now, we apply our level set algorithm to reconstruct the same shape. After initialization in figure 7(a), the front is made to propagate in the normal direction. It can be seen that in subsequent frames the front literally "flows" into the branches and finally in 7(f) it completely reconstructs the complex tree structure. The advantages of our scheme are quite apparent from this example. Since our front advancement process does not involve optimization of any quantity, the shape reconstruction results we obtain are independent of initialization. In addition, a *single instance of our shape model* "sprouts" branches and recovers all the connected components of a given shape. In figures 8(a)-(i) we plot the other level sets to elucidate the process of extending the image-based speed function to points away from the zero set. All calculations were carried out on a 64 X 64 grid and the time step Δt is set to 0.001.

Lastly, in figure (9) we depict a situation when the front undergoes a topological transformation to reconstruct the constituent shapes in an image. The image consists of three distinct shapes. Initial curve is placed in such a way that it envelops all the objects. The front is then advanced in the direction of negative normal. The level set $\{\psi = 0\}$ first wraps itself tightly around the objects (see figures 9(c) & 9(d)) and subsequently splits into four separate closed curves (figure 9(e)). While the first three closed segments of $\{\psi = 0\}$ recover the three distinct shapes, the one in the middle (see figure 9(e)), since it does not enclose any object, eventually disappears. Figure 9(f) 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. Again, to show the working of our algorithm, in figures 10(a)-(i) we show the level sets $\{\psi = \pm iC\}$, $i \in [-5..+5]$, with $C = 0.05$. The function ψ was discretized on a 64 X 64 grid and Δt is set to 0.001.

5 Concluding remarks

In this paper we presented a novel shape modeling scheme. Our approach while retaining the desirable features of existing methods for shape modeling, overcomes most of their deficiencies. We adopt the level set techniques first introduced in Osher & Sethian [13] to the problem of shape recovery. With this approach, complex shapes can be reconstructed. Unlike the variational formulations for shape reconstruction which rely on energy minimization, the final result in our method is completely independent of the initial state. This is a very desirable feature to have, specially if the problem is to recover object shapes from noisy images. 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 extendibility to higher dimensions. The equations of motion governing our evolutionary system resemble an initial valued Hamilton-Jacobi equation with a parabolic right-hand side and are amenable to stable entropy-satisfying numerical solution schemes. Thus, the result is a very general shape modeling algorithm which we believe will find numerous applications in the areas of computer vision and computer graphics.

In the context of 2D shape reconstruction, we force our front to come to a stop in the neighborhood of object boundaries by synthesizing a negative speed term from noisy images. It is easy to envision a similar scheme to reconstruct the surface structures of 3D objects from 3D medical image data. However, it is not clear how to adapt the level set formulation in its present form to reconstruct surface shapes from sparse range data. The other issue is one of time complexity. We have seen in the previous section that the stability requirement for solving the level set equations forces a very small time step for fine grids. On the other hand, shape reconstruction results on a coarse grid suffer from loss of accuracy. To salvage this situation we propose to embed out level set algorithm in a multiresolution framework, where the loss of accuracy at rapidly-converging coarse grid computation is compensated by highly accurate slowly-converging fine grid calculation. We are currently working on one such scheme.

References

[1] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, Vol. 46, pp. 1–21, 1989.

[2] 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.

[3] T.E. Boult and J.R. Kender, "Visual surface reconstruction using sparse depth data," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* June 1986, pp. 68–76.

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

[5] L. D. Cohen, "On Active Contour Models and Balloons," *Computer Vision, Graphics, and Image Processing*, Vol. 53, No. 2, pp. 211–218, March 1991.

[6] L. D. Cohen and I. Cohen, "Deformable models for 3D medical images using finite elements and balloons," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. , Urbana Illinois, June 1992.

[7] 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.

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

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

[10] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations I: The components of shape and reaction-diffusion space," Technical Report LEMS-105, Division of Engineering, Brown University, June 1992.

[11] 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.

[12] R. Malladi, J. A. Sethian, and B. C. Vemuri, "A topology-independent shape modeling scheme," in *Proceedings of SPIE Conference on Geometric Methods in Computer Vision II*, San Diego, July 1993.

[13] 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.

[14] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1988.

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

[16] R. Samadani, "Adaptive snakes: control of damping and material parameters," *Proceedings of SPIE Conference on Geometric Methods in Computer Vision*, Vol. 1570, pp. 202–213, San Diego California, July 1991.

[17] 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.

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

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

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

[21] 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.

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

[23] 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.

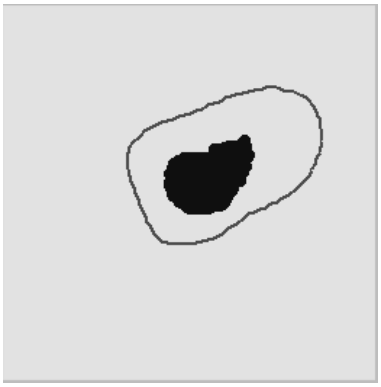
[24] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Int. Journal of Image and Vision Computing*, 4, pp. 107–114, 1986.

[25] B. C. Vemuri and R. Malladi, "Deformable models: Canonical parameters for surface representation and multiple view integration," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724–725, Maui Hawaii, June 1991.

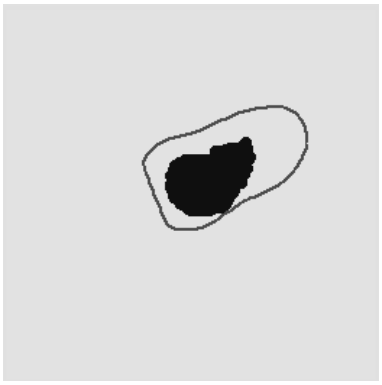
[26] B. C. Vemuri and R. Malladi, "Surface griding with intrinsic parameters," *Pattern Recognition Letters*, Vol. 13, No. 11, pp. 805–812, November 1992.

[27] B. C. Vemuri and R. Malladi, "Constructing intrinsic parameters with active models for invariant surface reconstruction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, in press.

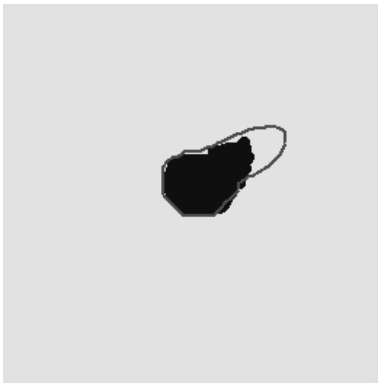
[28] Y. F. Wang and J. F. Wang, "Surface reconstruction using deformable models with interior and boundary constraints," in *Proceedings of ICCV*, pp. 300–303, Osaka, Japan, 1990.



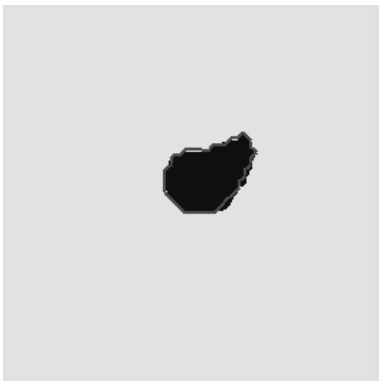
(a) Initialization



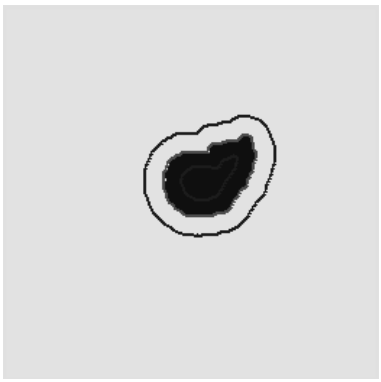
(b) After 22 iterations



(c) After 49 iterations

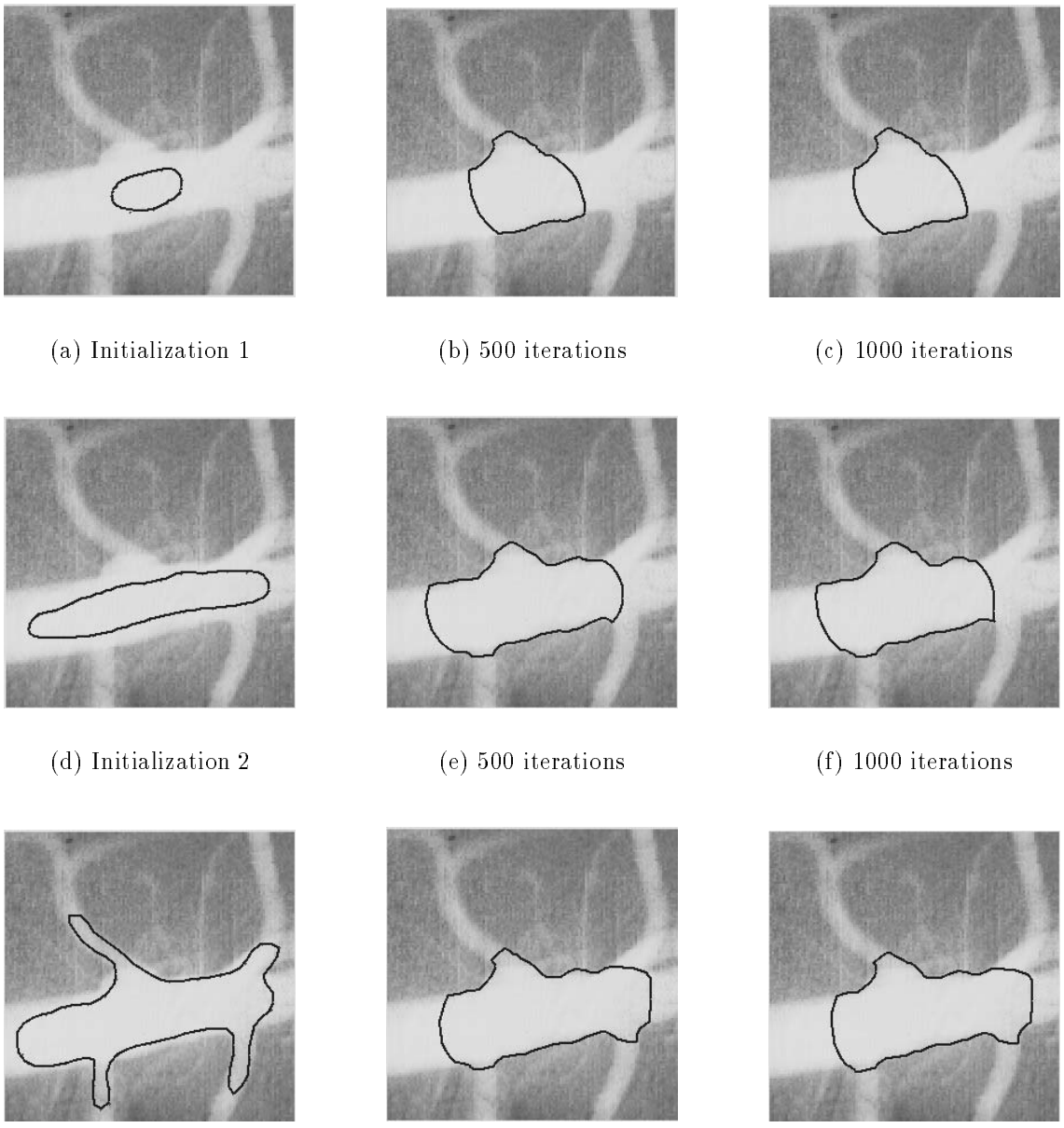


(d) After 80 iterations



(e) Fine grid computation

Figure 5: Shape reconstruction results at different resolutions: mesh size for parts (a), (b), (c), & (d) is 64 X 64 and the time step $\Delta t = 0.008$. Computation in part (e) was performed on a 256 X 256 mesh with a step size $\Delta t = 0.00005$. The other level sets shown are $\{\psi = \pm 0.05\}$.

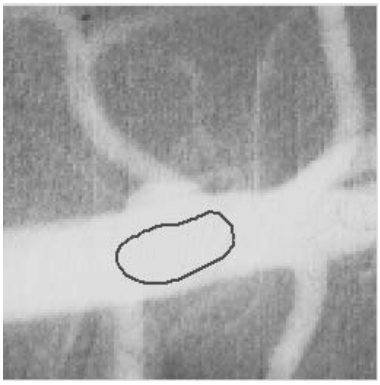


(a) Initialization 1 (b) 500 iterations (c) 1000 iterations

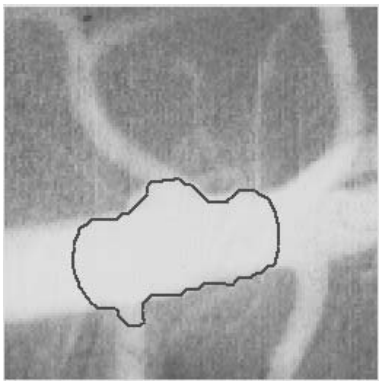
(d) Initialization 2 (e) 500 iterations (f) 1000 iterations

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

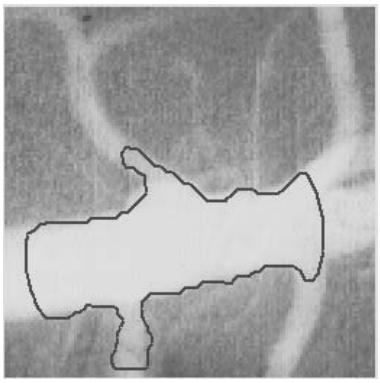
Figure 6: An unsuccessful attempt to reconstruct a complex shape with "significant" protrusions using an active contour model. Three different (poor) results are shown in parts (c), (f), & (i) corresponding to three distinct initializations in parts (a), (d), & (g) respectively.



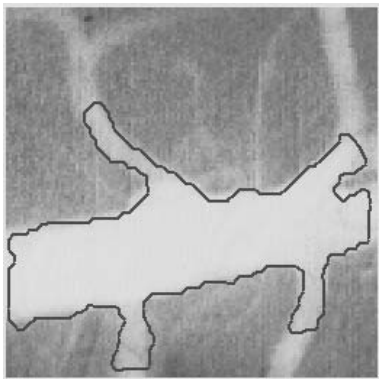
(a) Initialization



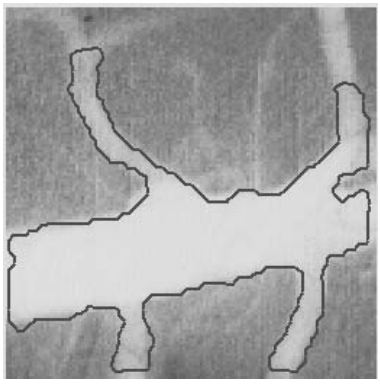
(b) After 60 iterations



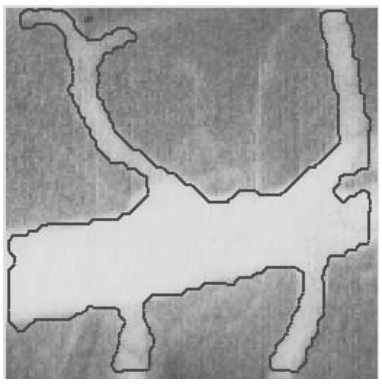
(c) After 123 iterations



(d) After 200 iterations

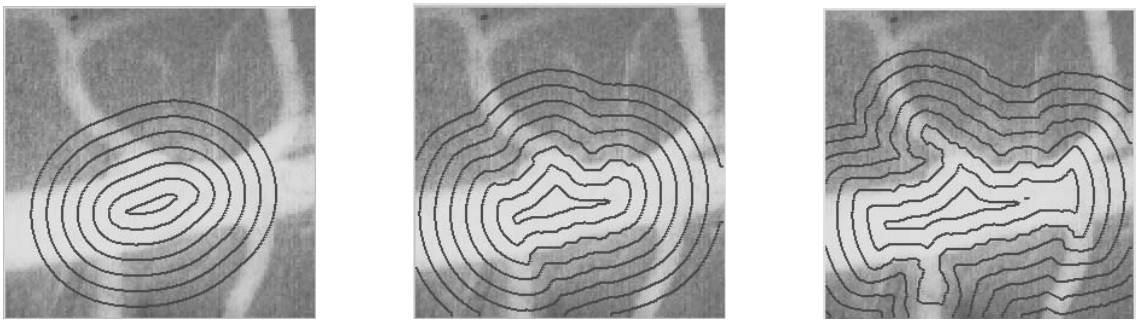


(e) After 275 iterations

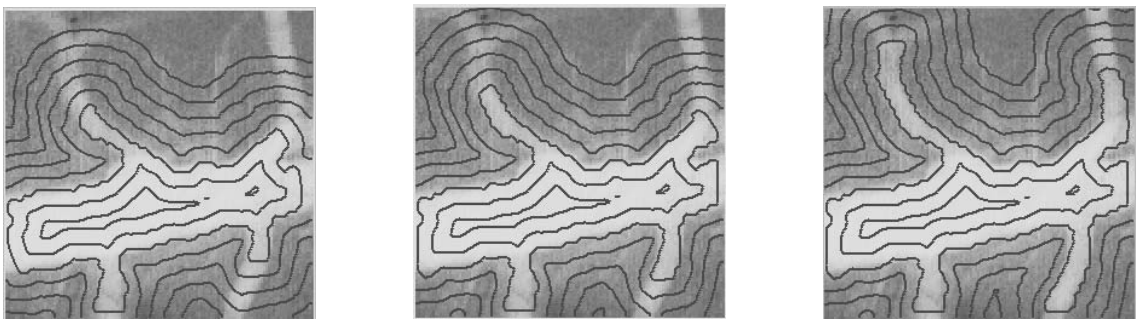


(f) After 391 iterations

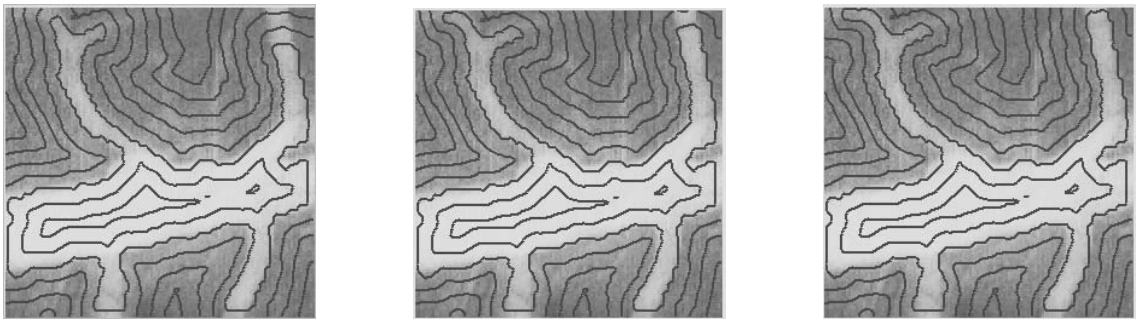
Figure 7: Reconstruction of a shape with "significant" protrusions: an arterial tree structure. Computation was done on a 64 X 64 grid with a time step $\Delta t = 0.001$.



(a) Initialization (b) 60 iterations (c) 123 iterations

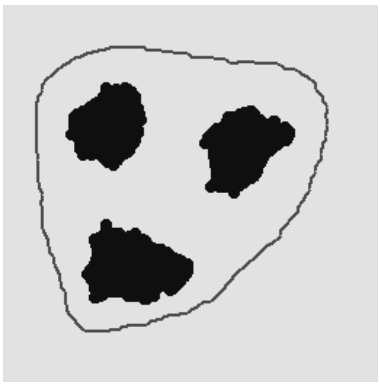


(d) 170 iterations (e) 200 iterations (f) 275 iterations

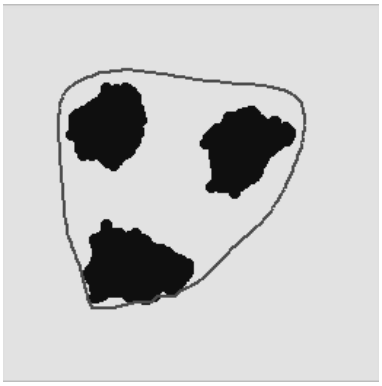


(g) 320 iterations (h) 350 iterations (i) 390 iterations

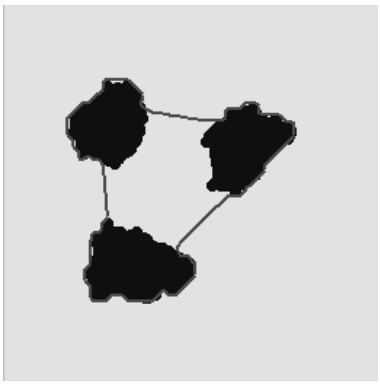
Figure 8: Reconstruction of a shape with protrusions: figure shows the level sets $\{\psi = \pm iC\}$, $i \in [-5..+5]$, and $C = 0.05$.



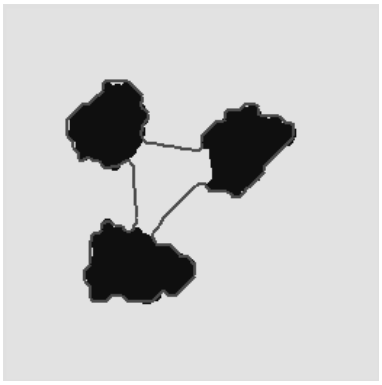
(a) Initialization



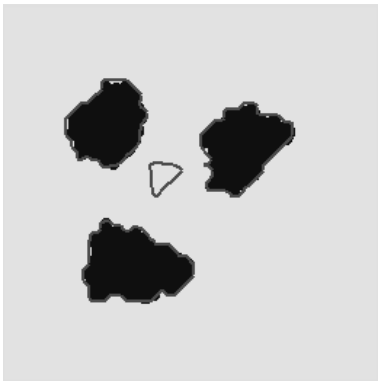
(b) After 30 iterations



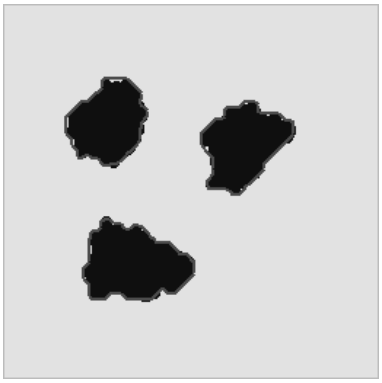
(c) After 60 iterations



(d) After 100 iterations

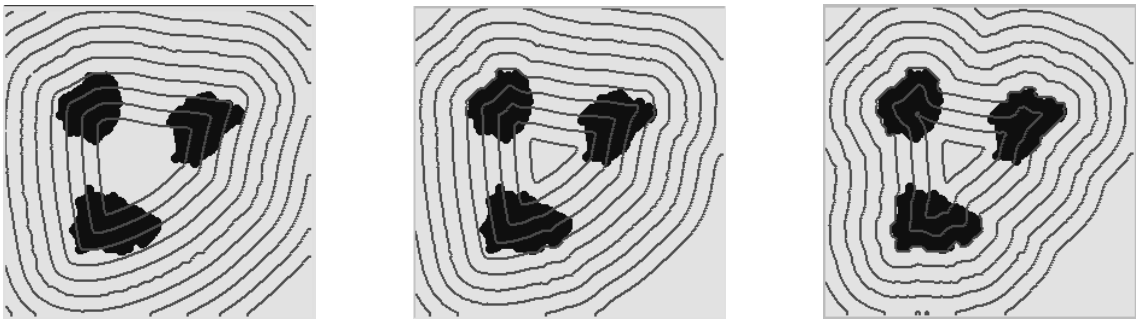


(e) After 125 iterations

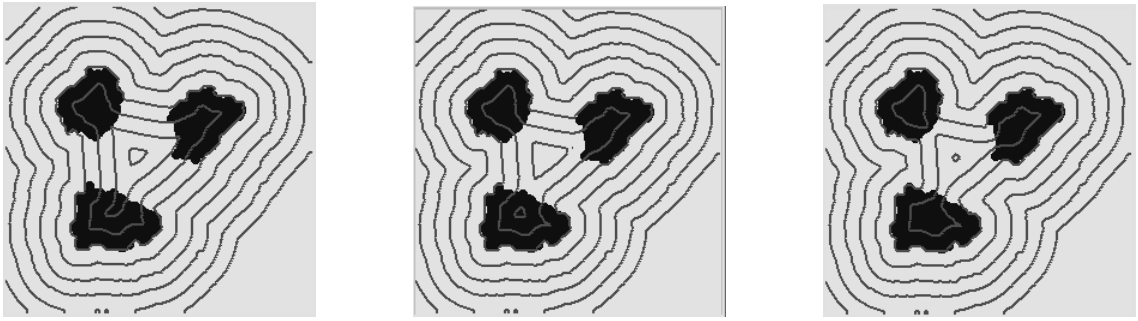


(f) After 140 iterations

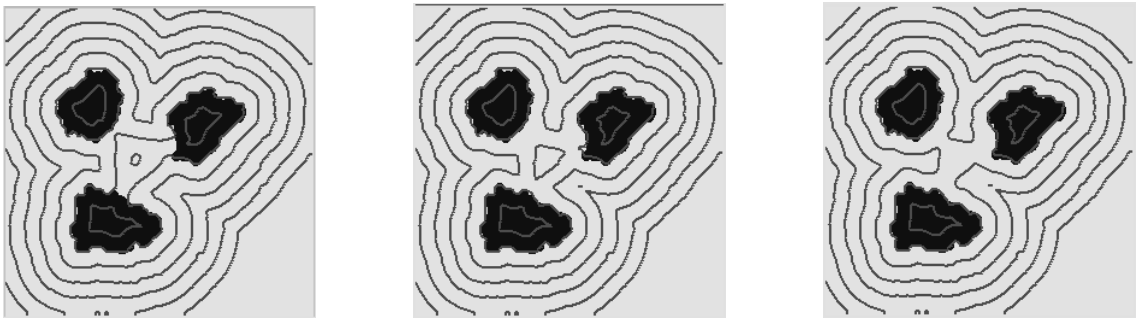
Figure 9: Topological split: a single instance of the shape model splits into three instances to reconstruct the individual shapes. Computation was done on a 64 X 64 mesh with a time step $\Delta t = 0.001$.



(a) Initialization (b) 60 iterations (c) 89 iterations



(d) 100 iterations (e) 120 iterations (f) 130 iterations



(g) 154 iterations (h) 170 iterations (i) 185 iterations

Figure 10: Topological split example: level sets shown are $\{\psi = \pm iC\}$, $i \in [-5..+5]$, with $C = 0.05$.