

A Fast Level Set based Algorithm for Topology-Independent Shape Modeling *

Ravikanth Malladi,¹ James A. Sethian,² and Baba C. Vemuri¹

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

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

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 the most attractive features of existing methods, and overcomes their prominent limitations. Our technique 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 & 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 the object boundaries. The resulting equation of motion is solved by employing *entropy-satisfying* upwind finite difference schemes. We also introduce a *new algorithm for rapid advancement* of the front using what we call a *narrow-band updation scheme*. This leads to significant improvement in the time complexity of the shape recovery procedure. An added advantage of our modeling 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 low contrast medical images.

*Submitted to the special issue of the *Journal of Mathematical Imaging & Vision* on Topology and Geometry in Computer Vision.

1 Introduction

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, as the name suggests, 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 or nonrigid. In the former case, we have what are popularly known as correlation-based template matching techniques, while the later involves a dynamic model fitting process to the data. In this paper, we present a new approach to dynamic shape modeling which retains the most attractive features of existing methods, and overcomes their prominent limitations. For the rest of this paper we will use the word shape model for a boundary (surface) representation of an object shape.

Shape reconstruction typically precedes the symbolic representation of surfaces and 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 [2, 22, 28, 3, 29, 14]. Generalized spline models with continuity constraints are well suited for fulfilling the goals of shape recovery (see [3, 4, 26]). Generalized splines are the key ingredient of the dynamic shape modeling paradigm introduced by Terzopoulos *et al.*, [27]. 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 [10, 30, 31, 32, 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 an assumption that the object is of a known 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 that may occur in the shape of interest during the shape recovery process. In this paper, we present a modeling scheme that makes no assumption about the object’s topology, and leads to a numerical algorithm whose convergence to the desired shape is completely independent of the shape initialization. Our method can also recover shapes whose topology changes over time, e.g., the cell boundary in cell division.

The framework of energy minimization has also been used successfully in the past for extracting salient image contours – edges and lines. Kass *et al.* [10] used energy-minimizing “snakes” that exhibit an affinity toward image features such as edge points and edge segments, while maintaining piecewise smoothness. The weights of the smoothness and image 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 surface models to image data by means of energy minimization [27]. In this modeling scheme, one 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 [6] defines an inflation force on the active contour. This force makes the model behave like an inflating balloon. The modified contour model will be “stopped” by a strong edge and will simply pass through a spurious edge since the force exerted by it 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 like the one shown in figure 1(b). 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 possess. 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

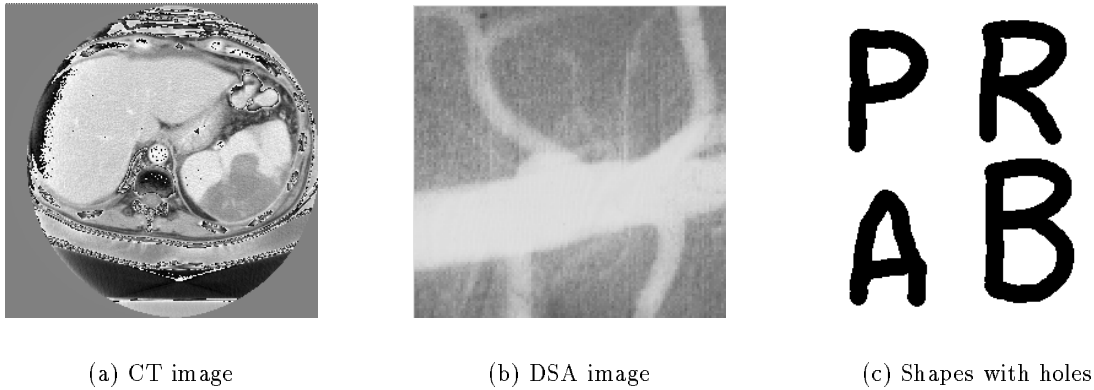


Figure 1: Test bed for our topology-independent shape modeling scheme.

parameters controlling the relative strengths of elasticity and rigidity are adapted to the data(see [21]). 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 [27] 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 segmented into its constituent parts. Instead, we propose a dynamic shape modeling method that will start with a single instance of the model and will automatically “*sprout*” branches during the evolutionary process. Once the shape has been segmented from the image, its constituent part structure may be inferred using the algorithm described in [12].

Most existing surface modeling techniques 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. As a result, most shape recovery schemes make strong assumptions about the object topology. Unknown topology is 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 of curves in 2D which is based on monitoring deformation energies of points on the elastic

curve has been discussed in [20].

In the context of static problems, more recently, particle systems have been used to model surfaces of arbitrary topology [25]. Smoothness and continuity constraints are imposed by subjecting the 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 static as well as dynamic 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 shape of interest (see figure 1(c)), has the ability to split freely to represent each shape [16, 17]. 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 14).

1.1 Overview

In this subsection we briefly outline our scheme to model complex shapes. This method is inspired by ideas first introduced in Osher & Sethian [19, 24] 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. Sethian [23] 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 [19, 24] 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 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 problem of shape recovery. To isolate a shape from its background, we first consider a closed, nonintersecting, initial hypersurface placed

inside (or outside) it. Following the above level set approach, this hypersurface is then made to *flow* along its gradient field with a speed $F(K)$, where K is the curvature of the hypersurface. As in [19], 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 recovery. 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 data. Secondly, we have to construct an *extension* of this speed function to other level sets $\{\psi = C\}$ in order to give a consistent meaning to the image-based speed term at all points 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 [19, 23, 24], has been applied in the area of computer vision for shape characterization by Kimia *et al.* [11, 12], 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 boundary of the object shape to be known, while we recover 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, as in [12], can be achieved once the object shape is extracted.

An important contribution of this paper over our previous work in [17] is the design and implementation of a faster numerical technique for solving the governing equation. Our new algorithm exploits the fact that the front, which is a particular level set $\{\psi = 0\}$ of a higher dimensional function, can be advanced by updating the function ψ at a small set of points. This scheme is an alternative to updating the ψ function at all the points in the computational domain. The set of

points at which the update procedure is applied belong to a narrow band lying on either side of the level set $\{\psi = 0\}$. Since the *narrow-band* update strategy involves only a fraction of the total number of points, a significant saving in time is realized, making our method a very attractive alternative to other shape recovery schemes. A complete discussion of the narrow-band techniques for interface propagation may be found in [1].

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 extended to decipher the constituent part structure for high-level processing. 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 recovery and in section 4 we outline the details of our narrow-band approach. Finally in section 5 we present some experimental results of applying our method to some noisy and low contrast medical images. We close with a discussion of advantages of our approach and direction of future research in section 6.

2 Front propagation problem

In this section we present the level set technique due to Osher and Sethian [19]. For details and an expository review, see Sethian [24]. 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. However, there are several problems with this approach, as discussed in Sethian [23]. 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

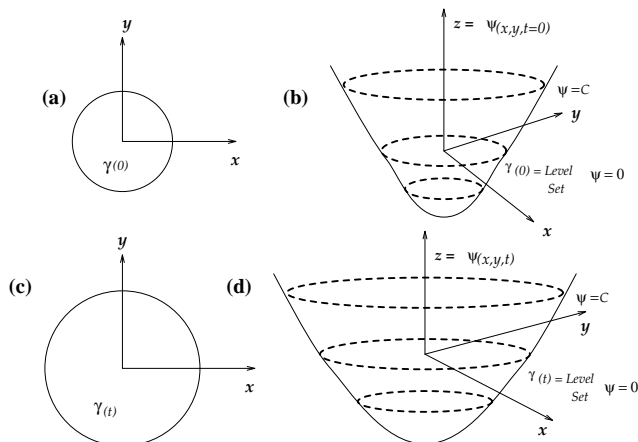


Figure 2: Level set formulation of equations of motion – (a) & (b) show the curve γ and the surface $\psi(x, y)$ at $t = 0$, and (c) & (d) show the curve γ and the corresponding surface $\psi(x, y)$ at time t .

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 [19] 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 (see 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 [19], let $\mathbf{x}(t)$ be the trajectory of a particle located on this level set, so

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

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,

$$\frac{\partial\mathbf{x}}{\partial t} \cdot \mathbf{n} = F, \quad (2)$$

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

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

and substitution yields

$$\psi_t + F |\nabla\psi| = 0, \quad (4)$$

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

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

This approach can also be easily extended to higher dimensions and appropriate expressions can be obtained for the mean curvature and the Gaussian curvature [19].

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

$$\psi_t + |\nabla\psi| = \varepsilon K |\nabla\psi|. \quad (6)$$

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 [19]).

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. There is a fundamental difference between the problems of front propagation and shape recovery. 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 recovery 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 [6]. 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 [10] (see the figure (9)). We rewrite equation (6) by splitting the influence of F as

$$\psi_t + F_A |\nabla \psi| + F_G |\nabla \psi|^2 = 0. \tag{7}$$

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

$$F_I(x, y) = \frac{-F_A}{(M_1 - M_2)} \{ |\nabla G_\sigma * I(x, y)| - M_2 \}, \quad (8)$$

where M_1 and M_2 are the maximum and minimum values of the magnitude of image gradient $|\nabla G_\sigma * I(x, y)|$, $(x, y) \in \Omega$. 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

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

\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

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

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. More sophisticated stopping criteria can be synthesized by using the orientation dependent “steerable” filters [9]. The modified equation of

motion is given by

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

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:

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.*

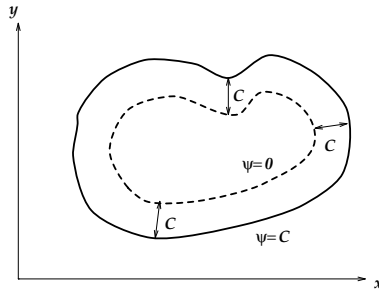


Figure 3: Huygen's principle construction

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

In this section, almost without a change, we present the arguments discussed in Sethian [24]. For complete details of the following scheme, we refer the reader to Osher & Sethian [19, 24].

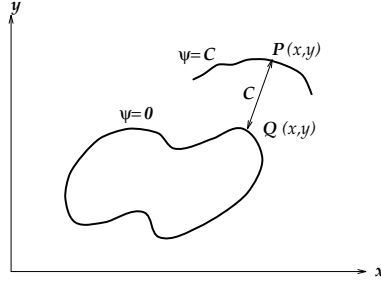


Figure 4: Extension of image-based speed terms to other level sets

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

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

with $\psi(x, y, t = 0) = \pm$ distance from (x, y) to $\gamma(0)$. As shown in Sethian [23], 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 $\varepsilon \rightarrow 0$. This particular weak solution can be selected with the help of a so-called “entropy condition”, see [23]: 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 [19, 24] 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

$$\phi_t + H(\phi_x) = 0, \quad (13)$$

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

$$u_t + [H(u)]_x = 0, \quad (14)$$

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

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

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

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

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,

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

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

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

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

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

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 [19]:

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

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

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

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 [19]). 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):

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

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

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

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

4.1 Narrow-band updation scheme

In this section we first consider a straightforward approach to solve the equations (22) & (23) and subsequently show that the time complexity of our algorithm can be improved by applying the update procedure to a small set of points in the domain. In other words, we propagate the front by solving the governing equation at a small number of points. These points lie in a *narrow band* around the level set $\{\psi = 0\}$.

It should be observed that by updating the level set function on a grid, we are moving the level sets without constructing them explicitly. A particular level set $\{\psi = 0\}$, which models the shape of interest, is forced to a stop in the vicinity of desired object boundaries. This task is accomplished by applying an image-based speed term on the zero set (equation (11)). An *extension* function is constructed to attribute a consistent meaning to the image-based speed term at points away from the zero set (see discussion in section 3). Therefore, the 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 scheme given in equation (23).
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

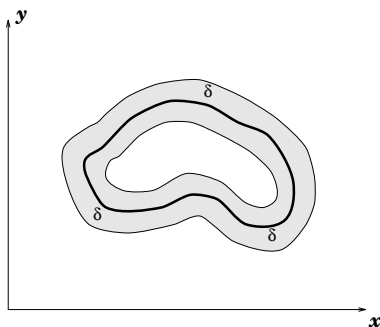


Figure 5: A narrow-band of width δ around the level set $\{\psi = 0\}$.

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 (23).

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

It is easy to see from the above algorithm that the most expensive step 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 (23), 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. A marginal improvement in performance can be achieved by evaluating the *extension* only once every k iterations. In other words, we take k steps in time without recomputing the image-based speed $(\hat{k}_I)_{i,j}$. Alternately, we could down-sample the image and perform our calculations at a lower resolution. On the other hand, we run the risk of losing accuracy while computing on a coarse grid. One approach is to embed our level set algorithm in a multiresolution framework. Instead, we improve the time complexity of our algorithm by updating the level set function ψ only at a small set of points on the grid.

The basic idea behind this new scheme stems from the observation 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 propagating inward or outward; either which way, it cannot move past the narrow-band's boundary.

Therefore, 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 $\gamma(0)$ (see section 2). Note that the reinitialization procedure must 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 equation (23).
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.

The issue of boundary conditions needs to be addressed at this time. In our original method, the level set function ψ was initialized and updated on a square grid. To update $\psi_{i,j}$ at a boundary point (i, j) , it is necessary to specify the boundary conditions in order to numerically approximate the derivatives. We have chosen free-end boundary conditions, i.e., if a derivative computation at a given boundary point (i, j) involves a point that falls outside the computational domain, then that derivative is set to zero. In our new 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 imperative.

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 updation algorithm to a star shaped front propagating with speed $F = -K$, where K is the curvature as in equation (5). The calculation was done on a unit box with 64 points in either direction, and a time step of $\Delta t = 0.0003$ 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 $\{\psi = \pm i0.04\}$, where $i \in [1..5]$. After 40 narrow-band updations (figure 6(b)), only the level sets $\{\psi < |0.08|\}$ move and the rest remain stationary. We note the inconsistency between

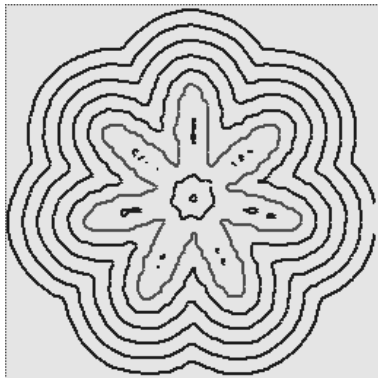
the level sets lying on either side of the narrow-band, making the reinitialization step necessary in order to restore the meaning of the ψ function. Following the reinitialization step, another 40 update steps are applied (figure 6(c)), which “diffuses” the high curvature regions of the front even further. 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.

5 Experimental results

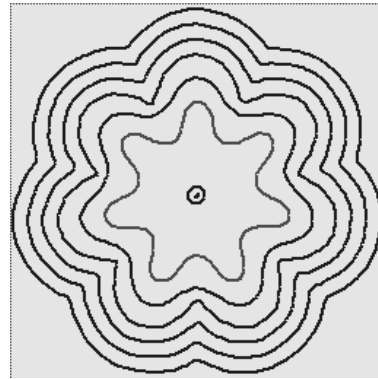
In this section we present several shape recovery results that were obtained by applying the narrow-band level set algorithm to image data. Given an image, our method requires the user to provide an initial contour $\gamma(0)$. As we shall see, there is absolutely no restriction on where the initial 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 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.

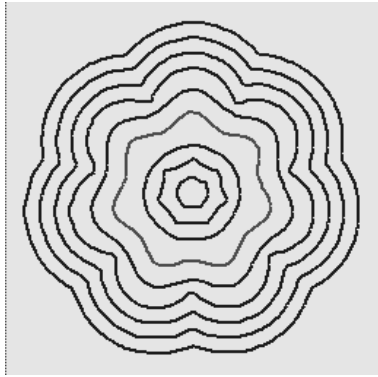
We now present our shape recovery results in 2D. First we consider a 256 X 256 CT (computed tomography) image of an abdominal section shown in figure 7(a). Our intention is to recover the shape of the stomach in this particular slice. The function ψ has been discretized on a 128 X 128 mesh, i.e., calculations are performed at every second pixel. In figure 8(a), we show the closed



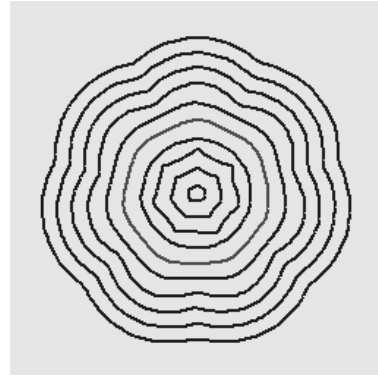
(a) Initialization



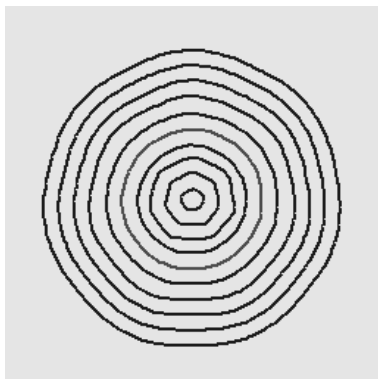
(b) After 40 iterations



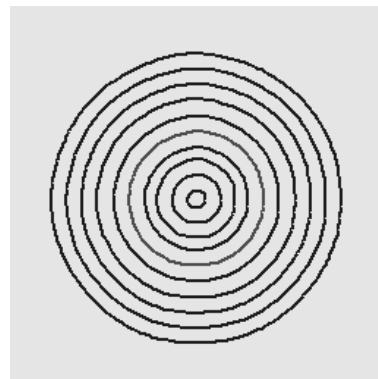
(c) After 80 iterations



(d) After 120 iterations



(e) After 160 iterations



(f) After 200 iterations

Figure 6: Narrow-band update algorithm applied to a star-shaped front propagating with speed $F = -K$. Calculations were done on a 64 X 64 grid with a time step $\Delta t = 0.0003$. ψ was recomputed after every 40 time iterations.

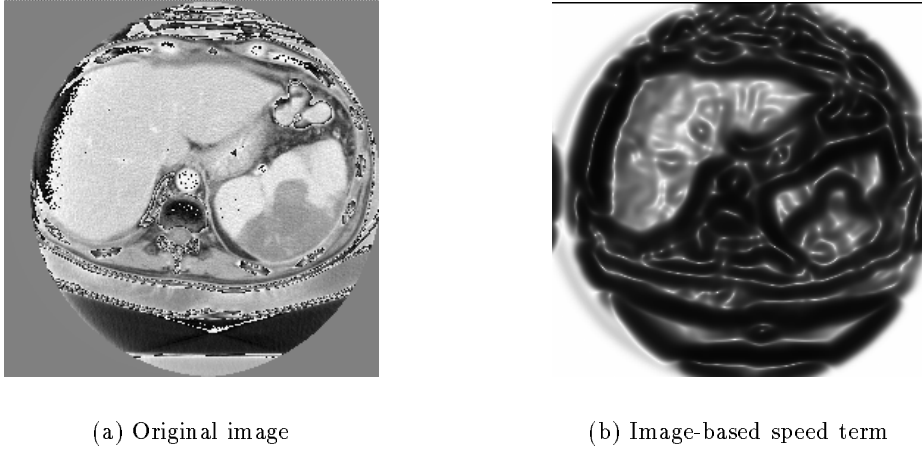
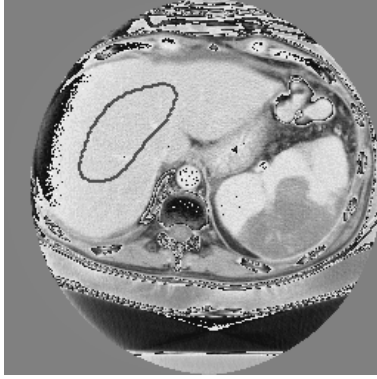


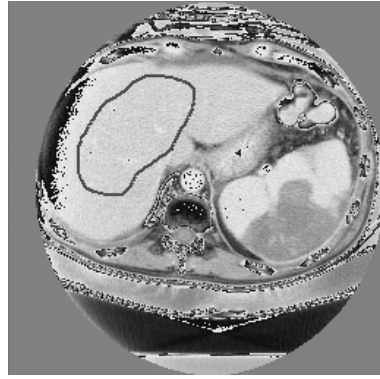
Figure 7: Image-based speed term $k_I(x, y) = \frac{1}{1+|\nabla G_{\sigma} * I(x, y)|}$, with $\sigma = 3.25$, synthesized from the CT image.

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 updation algorithm to move the front with a time step size set to $\Delta t = 0.0005$ and the ψ function recomputed after every 50 time steps. Figure 7(b) shows the image-based speed term which is synthesized according to equation (10). This term when applied on the propagating front, acts as a stopping criterion, thereby causing the front to come to rest in the neighborhood of the desired shape. 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). It should be noted 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.

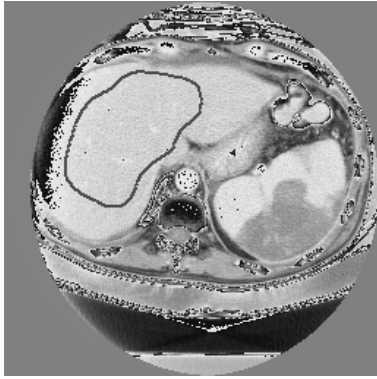
As mentioned in section 3, smoothness of the front can be controlled by choosing an appropriate curvature component in the speed function $F = 1 - \varepsilon K$. The objective of our next experiment is



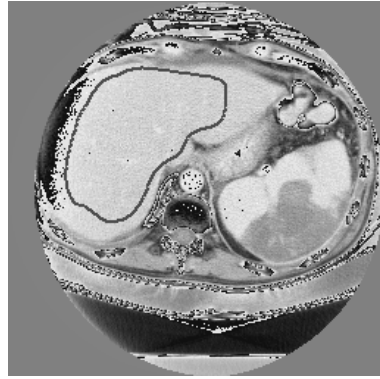
(a) Initialization



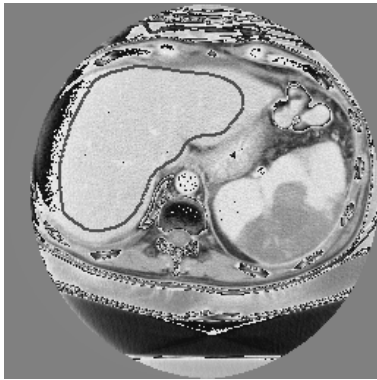
(b) After 100 iterations



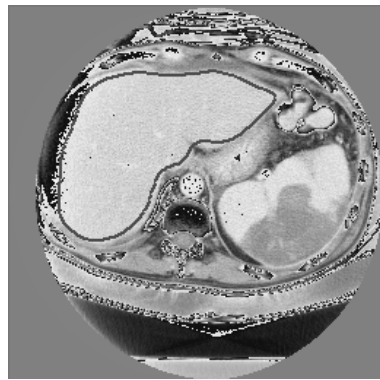
(c) After 175 iterations



(d) After 300 iterations



(e) After 450 iterations



(f) After 575 iterations

Figure 8: Recovery of the stomach shape from a CT image of an abdominal section. Narrow-band computation was done on a 128 X 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.

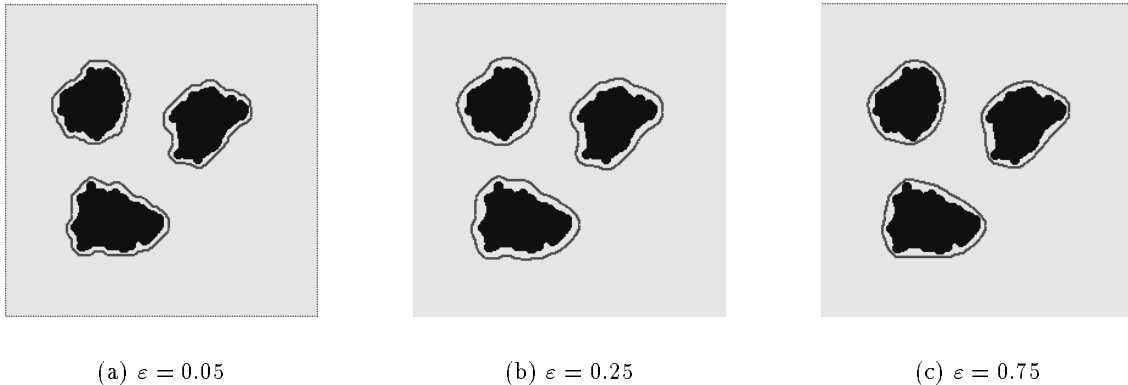


Figure 9: Smoothness control in shape recovery can be achieved by varying the curvature component in the speed $F = \hat{k}_I(1.0 - \varepsilon K)$.

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 are then made to propagate with speed $F = \hat{k}_I(1.0 - \varepsilon K)$. First, as shown in figure 9(a), we 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 controlled-continuity provided by thin-plate-membrane splines [26, 10].

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 to bring the limitations of the later into focus. We first attempt to reconstruct the complex arterial structure using a snake model with *inflation forces* [6]. 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, 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 focus. 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 10(g)), the snake “snaps” back into a relatively “bumpless” configuration in figure 10(h). This inadequacy stems from snake’s arc-length (elasticity) and curvature (rigidity) minimizing property. 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 11(a), the front is made to propagate in the normal direction. We employ the narrow-band updation scheme with a band width of $\delta = 0.075$ to move the front. It can be seen that in subsequent frames the front literally “flows” into the branches and finally in 11(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 recovery 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. All calculations were carried out on a 64 X 64 grid and a time step $\Delta t = 0.001$ was used.

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 envelops all the objects. The front is then advanced in the direction of 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(c) & 12(d)) and subsequently splits into four separate closed curves (figure 12(e)). While the first three closed segments of $\{\psi = 0\}$ recover the three distinct shapes, the one in the middle (see figure 12(e)), since it does not enclose any object, eventually disappears. Figure 12(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. To show the working of our algorithm, in figures 13(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 was set to 0.001.

Lastly, we show that our approach can also be used to recover shapes with holes. We do this by applying our method to extract the shapes of hand-written characters. The characters “A” and “B” in the figure (14) are examples of shapes with holes. Recovery of hand-written character shapes finds application in the area of character recognition. These shapes can also serve as an input stage to algorithms computing medial axis transform [13, 5] or skeletons [18, 15]. The attractive feature here is the seamless way in which the outer and inner boundaries of a given shape can be recovered without separate initializations. In figure 14(a), we show the initial contour which encloses all the characters. This contour is then made to propagate inward with a constant speed. Figures 14(b) shows an intermediate stage in the front’s evolution and in 14(c), the front splits into four separate contours. The calculation comes to a halt when in figure 14(d), the level set $\{\psi = 0\}$ recovers the outer boundaries of four separate characters. Unlike the characters “A” & “B” for which we need to extract the inner boundary for their complete shape description, for the characters “S” & “H”, the outer boundary completely describes their shape. In the second stage of our computation, we treat the zero set configuration in figure 14(d) as an initial state, and propagate all the fronts inward by *momentarily relaxing the image-based speed term*. This causes the zero set to move into the character shapes as shown in figure 14(e). Finally in figure 14(f), the zero set comes to stop in the neighborhood of holes that are present in the shapes of characters “A” & “B”, thereby achieving a complete shape recovery. The calculations for this experiment were done on a 128 X 128 grid and the time step Δt was set to 0.00025.

6 Summary and discussion

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 [19] to the problem of shape recovery. With this approach, complex shapes can be recovered from images. Unlike the variational formulations for shape recovery 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 in applications such as automatic shape recovery 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 recovery, we force our front to come to a stop in the neighborhood of object boundaries by synthesizing a speed function from noisy images. An *extension* is defined to attribute a consistent meaning to this speed term at all points in the computational domain. A significant improvement in time complexity is realized by adopting the *narrow-band* update strategy. In the narrow-band scheme, the front is advanced by updating the ψ function in a narrow band around the zero set. The efficacy and versatility of our method is demonstrated by a series of shape recovery experiments on a set of synthetic and noisy medical images.

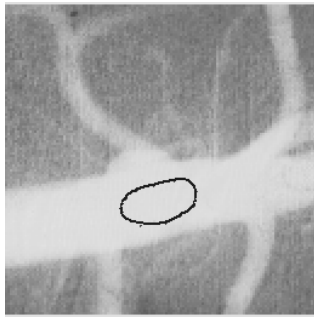
Our future research efforts will be focussed on extending this method to other application domains. It is easy to envision a similar scheme to reconstruct the surface structures of 3D objects from 3D medical image data. However, it is not clear how to adapt the level set formulation in its present form to reconstruct surface shapes from sparse range data. The issue of time complexity is an important one, more so in 3D. We have seen in section 4 that the stability requirement for solving the level set equations forces a very small time step for fine grids and that the evaluation of the *extension* is computationally very expensive. The narrow-band strategy has drastically reduced the computational cost in 2D, but implementing a similar scheme in 3D could prove cumbersome. One approach is to embed our 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. Alternately, a parallel implementation can be devised. We are currently pursuing such schemes.

References

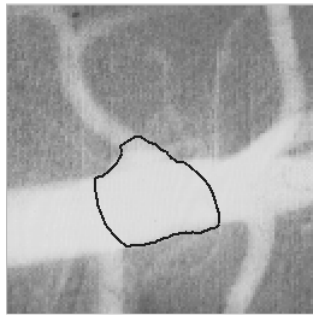
- [1] D. Adalsteinsson, “A narrow-band approach to level set techniques for propagating interfaces,” LBL Report, Oct. 1993.
- [2] R. M. Bolle and B. C. Vemuri, “On three-dimensional surface reconstruction methods,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI 13, No. 1, pp. 1–13, 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] H. Blum, “A transformation for extracting new descriptors of shape,” *Models for the Perception of Speech and Visual Form* (W. Wathen-Dunn, ed.), Cambridge MA: MIT Press, 1967.
- [6] L. D. Cohen, “On Active Contour Models and Balloons,” *Computer Vision, Graphics, and Image Processing*, Vol. 53, No. 2, pp. 211–218, March 1991.
- [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] 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.
- [10] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” *International Journal of Computer Vision*, pp. 321–331, 1988.
- [11] 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.

- [12] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, “Shapes, shocks, and deformations I: The components of shape and reaction-diffusion space,” to appear in *International Journal of Computer Vision*, 1992.
- [13] D. T. Lee, “Medial Axis Transformation of a Planar Shape,” *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-4, No. 4, pp. 363–369, 1982.
- [14] 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.
- [15] F. Leymarie and M. D. Levine, “Simulating the grass-fire transform using an active contour model,” *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-14 No. 1, pp. 56–75, 1992.
- [16] 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, California, pp. 246–258, July 1993.
- [17] 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.
- [18] N. Mayya and V. T. Rajan, “An efficient shape representation scheme using Voronoi skeletons,” IBM Research Report, RC 19161 (83458), Sept. 1993.
- [19] 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.
- [20] R. Samadani, “Changes in connectivity in active contour models,” *Proceedings of the Workshop on Visual Motion*, pp. 337–343, Irvine California, March 1989.
- [21] 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.

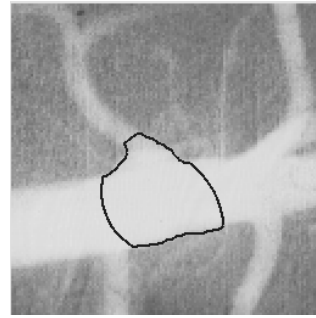
- [22] L.L. Schumaker, "Fitting Surfaces to Scattered data," in *Approximation Theory II*, G.G. Lorentz, C.K. Chui, and L.L. Schumaker, (eds.). New York: Academic Press, 1976, pp. 203–267.
- [23] J. A. Sethian, "Curvature and the evolution of fronts," *Commun. in Mathematical Physics*, Vol. 101, pp. 487–499, 1985.
- [24] J. A. Sethian, "Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws," *Journal of Differential Geometry*, Vol. 31, pp. 131–161, 1990.
- [25] R. Szeliski and D. Tonnesen, "Surface modeling with oriented particle systems," *Computer Graphics SIGGRAPH*, Vol. 26, No. 2, pp. 185–194, July 1992.
- [26] 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.
- [27] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Artificial Intelligence*, 36, pp. 91–123, 1988.
- [28] 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.
- [29] 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.
- [30] B. C. Vemuri and R. Malladi, "Surface gridding with intrinsic parameters," *Pattern Recognition Letters*, Vol. 13, No. 11, pp. 805–812, November 1992.
- [31] 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.
- [32] 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.



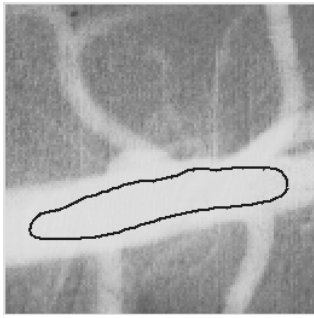
(a) Initialization 1



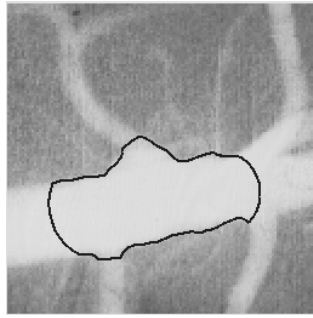
(b) 500 iterations



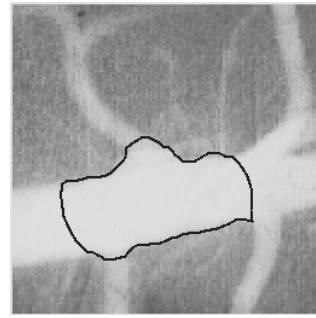
(c) 1000 iterations



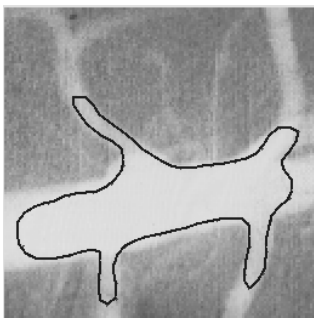
(d) Initialization 2



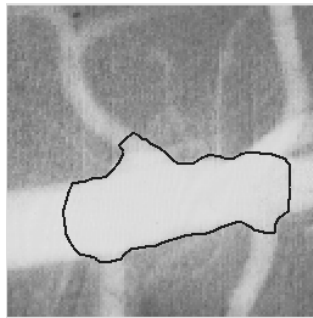
(e) 500 iterations



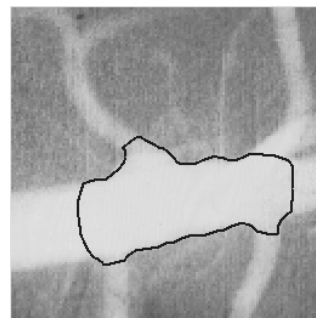
(f) 1000 iterations



(g) Initialization 3

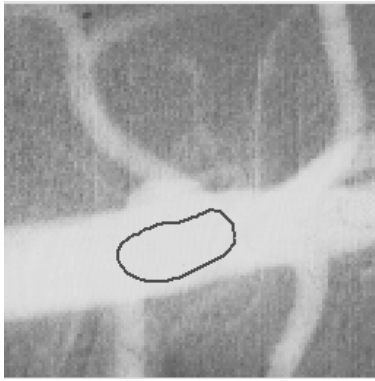


(h) 500 iterations

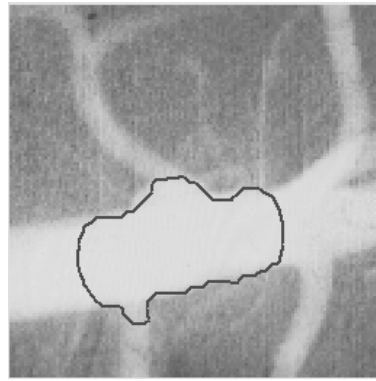


(i) 1000 iterations

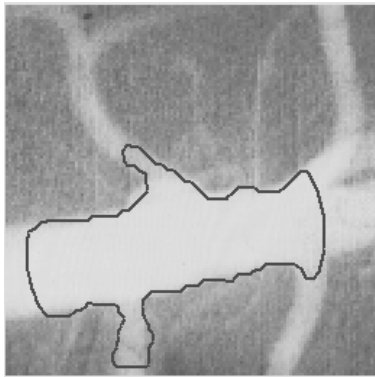
Figure 10: An unsuccessful attempt to reconstruct a complex shape with “significant” protrusions using an active contour model. Three different (poor) results are shown in parts (c), (f), & (i) corresponding to three distinct initializations in parts (a), (d), & (g) respectively.



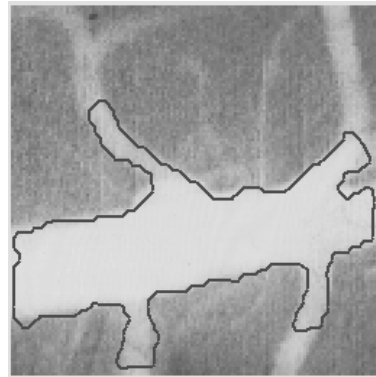
(a) Initialization



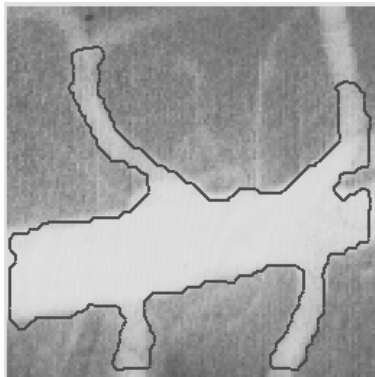
(b) After 60 iterations



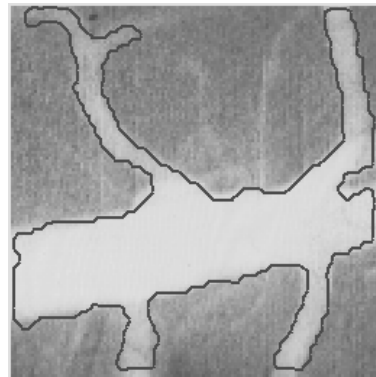
(c) After 123 iterations



(d) After 200 iterations

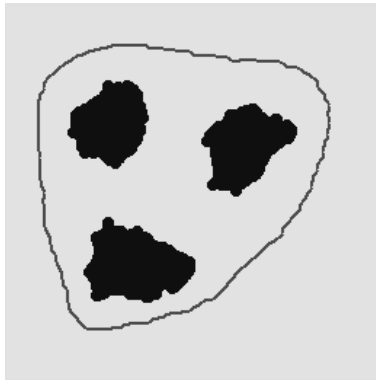


(e) After 275 iterations

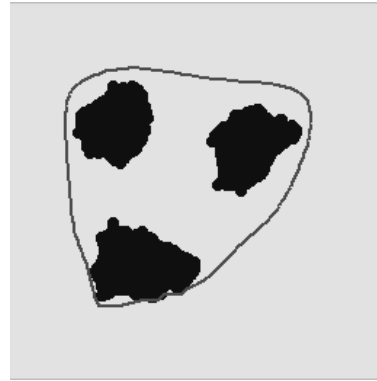


(f) After 391 iterations

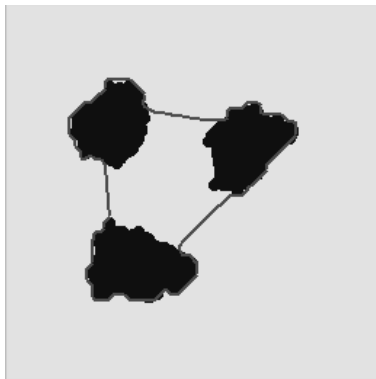
Figure 11: Reconstruction of a shape with “significant” protrusions: an arterial tree structure. Computation was done on a 64 X 64 grid with a time step $\Delta t = 0.001$. A narrow-band updation scheme was used with a band width of $\delta = 0.075$.



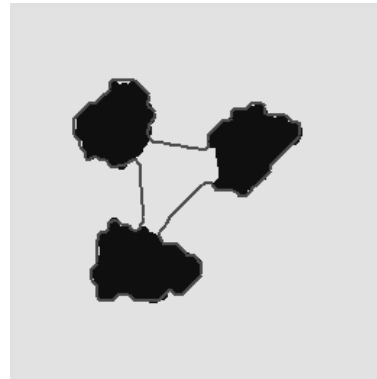
(a) Initialization



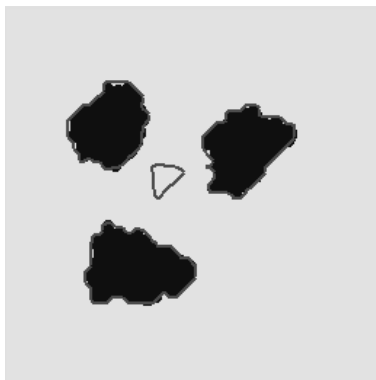
(b) After 30 iterations



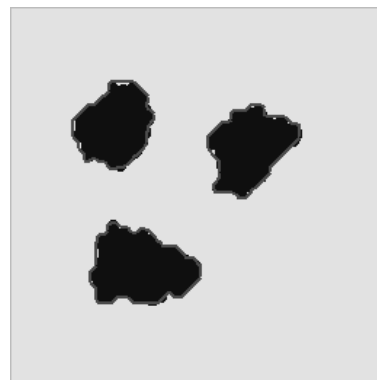
(c) After 60 iterations



(d) After 100 iterations

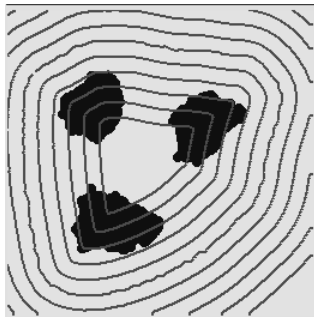


(e) After 125 iterations

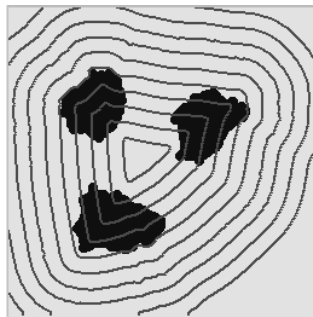


(f) After 140 iterations

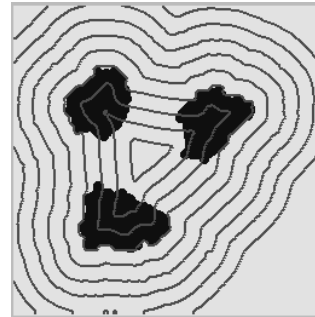
Figure 12: Topological split: a single instance of the shape model splits into three instances to reconstruct the individual shapes. Computation was done on a 64×64 mesh with a time step $\Delta t = 0.001$.



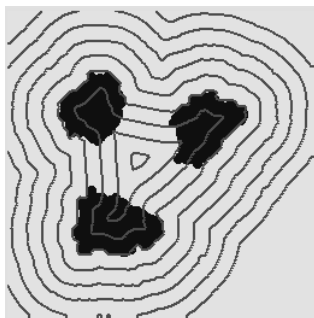
(a) Initialization



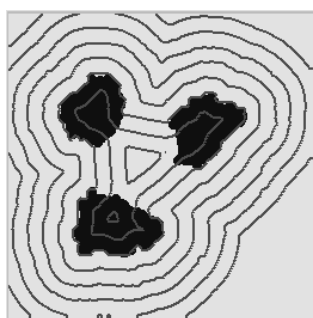
(b) 60 iterations



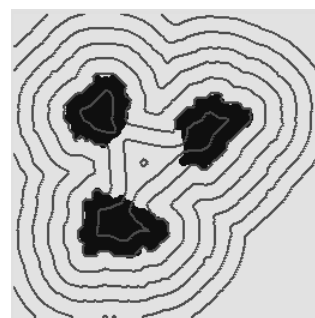
(c) 89 iterations



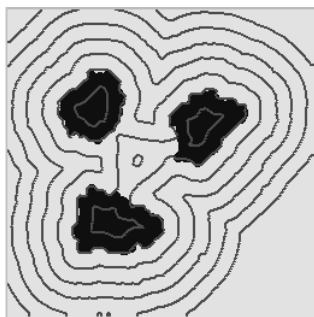
(d) 100 iterations



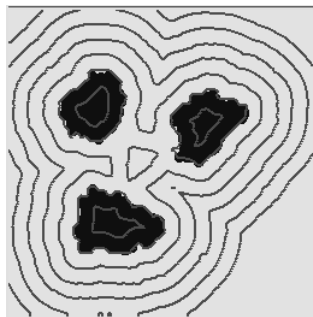
(e) 120 iterations



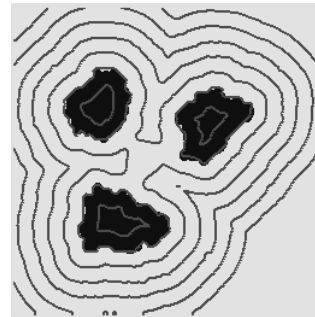
(f) 130 iterations



(g) 154 iterations



(h) 170 iterations



(i) 185 iterations

Figure 13: Topological split example: level sets shown are $\{\psi = \pm iC\}$, $i \in [-5..+5]$, with $C = 0.05$.



(a) Initialization



(b) 545 iterations



(c) 760 iterations



(d) Outer boundaries



(e) Second stage



(f) Inner boundaries

Figure 14: 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 X 128 grid and the time step Δt was set to 0.00025.