COMM. MATH. SCI.
Vol. 1, No. 4, pp. 623–656

REVIEW ARTICLE

LEVEL SET METHODS AND THEIR APPLICATIONS IN IMAGE SCIENCE*

RICHARD TSAI† AND STANLEY OSHER‡

In this article, we discuss the question "What Level Set Methods can do for image science". We examine the scope of these techniques in image science, in particular in image segmentation, and introduce some relevant level set techniques that are potentially useful for this class of applications. We will show that image science demands multi-disciplinary knowledge and flexible but still robust methods. That is why the Level Set Method has become a thriving technique in this field.

We begin by reviewing some typical PDE based applications in image processing. In typical PDE methods, images are assumed to be continuous functions sampled on a grid. We will show that these methods all share a common feature, which is the emphasis on processing the level lines of the underlying image. The importance of level lines has been known for some time. See e.g., [2]. This feature places our slightly general definition of the level set method for image science in context. In section two, we describe the building blocks of a typical level set method in the continuum setting. Each important task that one needs to do is formulated as the solution to certain PDEs. Then, in section three, we quickly describe the finite difference methods developed to construct approximate solutions to these PDEs. In section four, we discuss the Chan-Vese segmentation algorithm and two new fast implementation methods. Finally, in section five, we describe some new techniques developed in the level set community as our prospectus for the future.

1. Level Set Methods and Image Science

The level set method for capturing moving fronts was introduced by Osher and Sethian [72] in 1987. (Two earlier conference papers which contained some of the key ideas, have recently come to light [28, 29]). Over the years, the method has proven to be a robust numerical device for this purpose in a diverse collection of problems. One set of problems lies in the field of image science. In this article, we will emphasize not only what has been done in image science using level set techniques, but also in other area of sciences in which the level set methods are applied successfully — the idea is to point out the related formulations and solution methods to the image science communities. These communities include image/video processing, computer vision, and graphics. These are diverse, with specialties such as medical imaging and Hollywood type special effects.

Let us begin with a quick examination of what constitutes a classical level set method: an implicit data representation of a hypersurface (codimension 1 object), a set of PDEs that govern how the surface moves, and the corresponding numerical methods for implementing this on computers. In fact, a typical application in image

*Received: October 28, 2003; accepted (in revised version): November 5, 2003.
†Institute for Advanced Study, Einstein Drive Simonyi Hall Princeton, NJ 08540, (yt-sai@math.ias.edu).
‡University of California, Los Angeles, CA 90095-1555, (sjo@math.ucla.edu).

science may very well need all these features. We will illustrate this point by some classical applications.

The term "image science" (or "imaging science") is used here to denote a wide range of problems related to digital images. It is generally referred to problems related to image processing, computer graphics, and computer vision. The type of mathematical techniques involved range from discrete math, linear algebra, statistics, approximation theory, to partial differential equations, quasi-convexity analysis related to solving inverse problems, and even algebraic geometry. The role of a level set method for image processing often relates to PDE techniques involving one or more of the following features: 0) regarding an image as a function sampled on a given grid with the grid values corresponding to the pixel intensity in suitable color space, 1) regularization of the solutions, 2) representing boundaries, and 3) the numerics developed for the level set methods. Particularly in light of 2), it is not hard to seek an application of the level set method for segmentation. There are, however, efforts which combine different disciplines mentioned above together to accomplish special tasks. For example, Mark Green [47] used a new statistical approach together with Total Variation (TV) denoising [80], and Yves Meyer [62] analyzed function/image representation from a decomposition that is motivated by the TV denoising of [80]. We will see soon that TV denoising is closely related to solving an inverse problem.

In a later section, we will examine some essential fundamentals of the level set methodology. We refer the reader to the original paper [72] and the new book [68] for detailed exposition of the level set method. Also, there is a set of presentation slides available from the first author's home page[1].

We write a typical PDE method as

$$Lu = \lambda Ru,$$

or

$$u_t + Lu = \lambda Ru,$$

where L is some operator applied to the given image, and R denotes the regularization operator. Typically, an image model is obtained by devising an energy functional $E(u)$ and solving for a minimizer. The Mumford-Shah multiscale segmentation model is defined this way [63], as is TV denoising which can be written as:

$$\text{minimum}_u E(u) = \frac{1}{2}\int (u - u_0)^2 dx + \lambda \int |\nabla u| dx,$$

where u_0 is the given noisy image. In this set up, Lu will be the Euler-Lagrange equation subject to the associated natural boundary conditions, which is just $u - u_0$ here and $Ru = \nabla \cdot \frac{\nabla u}{|\nabla u|}$, the curvature of the level curve at each point in this application. When L is not invertible, or when certain regularity in the image u is required, a regularization term will be added. For example, in the TV deblurring of [79],

$$Lu = K * (Ku - u_0),$$

[1] http://www.math.princeton.edu/~ytsai

where K is a compact integral operator , u_0 is the given image, and the restored image is the limit $u(t)$ as $t \longrightarrow \infty$. In the usual version of Total Variation based methods, the regularization is defined as

$$Ru = \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right). \tag{1.1}$$

This is the Euler-Lagrange derivative of the TV term as remarked above. Note that in many image applications, the unregularized energy functional is nonconvex, and its global minimizer corresponds to the trivial solution. Only some local minimizer is needed.

In the development of this type of method, one often qualitatively studies the solutions of the governing PDEs by investigating what action occurs on each of the level sets of a given image. In the TV regularization of [61] for example, $Ru(x)$ actually denotes the mean curvature of the level set of u passing through x. References [14, 20, 38], for example, provided analysis for these types of PDEs.

The effects of (1.1) in noise removal can be explained as follows: the level curves in the neighborhoods of noise on the image have high curvatures. The level curves of the viscosity solution to

$$u_t = \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right) |\nabla u|$$

shrink with the speed of the mean curvature and eventually disappear. Consequently, the level curves with very high curvatures (noise) disappear much faster than those with relatively lower curvatures, (this was the approach taken in [61]). If the $|\nabla u|$ term is dropped (as it usually is) the velocity is also inversely in proportion to the gradient. This means relatively flat edges do not disappear. The analysis of motion by curvature and other geometric motions are all important consequences of viscosity solution theory, originally devised for Hamilton-Jacobi Equations, related to evolution past the singularities, including the pinching-off of level curves. See [20, 26, 38, 39, 40, 42].

Another interesting category of applications is data interpolation. In the problem of inpainting, see e.g. [5] and Figure 1.1, the challenge is to repair images with regions of missing information. The algorithms are developed with strong motivation on connecting the level curves over the "inpainting domain" in an "appropriate way". In a rather orthogonal way, the AMLE (Absolutely Minimizing Lipschitz Extension) algorithm, see e.g. [13], assumes a given set of level curves of an image, and fills in the regions in between the given level curves while trying to minimize the variation of the new data generated.

In many applications such as image segmentation or rendering, level set methods are used to define the objects of interest. For example, a level set function is used to single out desired objects such as the land mass of Europe [18]. The land mass is defined to be the connected region where the level set function is of one sign (see Figure 1.2). There are many successful algorithms. Examples are [17, 75]. In a different, but related, context, Zhao et al. use a level set function to interpret unorganized data sets [108, 105].

Many of the above methods rely on the variational level set calculus similar to that of [107] to formulate the energies whose minimizers are interpreted as the solution to the problems, and the solutions are level set functions. In general, the energies are

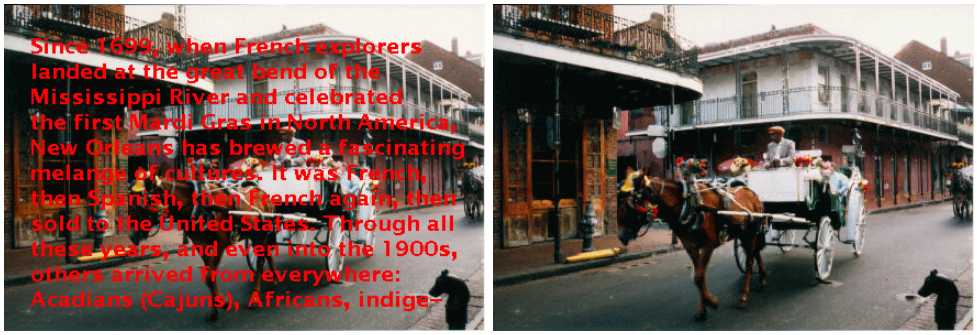

FIG. 1.1. *Image obtained from* `http://mountains.ece.umn.edu/~guille/inpainting.htm`

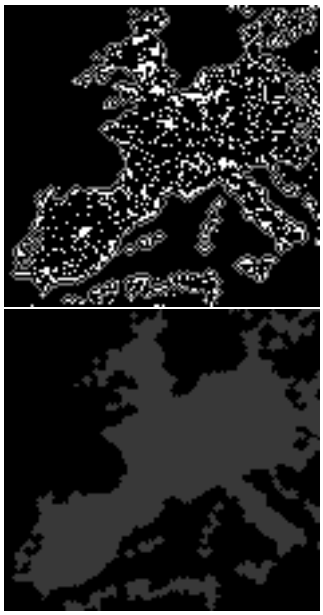

FIG. 1.2. *Land mass of Europe found using active contours.*

variants of the surface integral

$$\int_\Omega F(\phi, u)\delta\phi|\nabla\phi|dx,$$

and the volume integral

$$\int_\Omega G(\phi, u)H(-\phi)dx,$$

see [107] for details and definitions. We shall return to this in a later section on image segmentation.

We notice that in some of the above applications, level set functions are used to separate the domain into different regions. The interfaces separating those regions

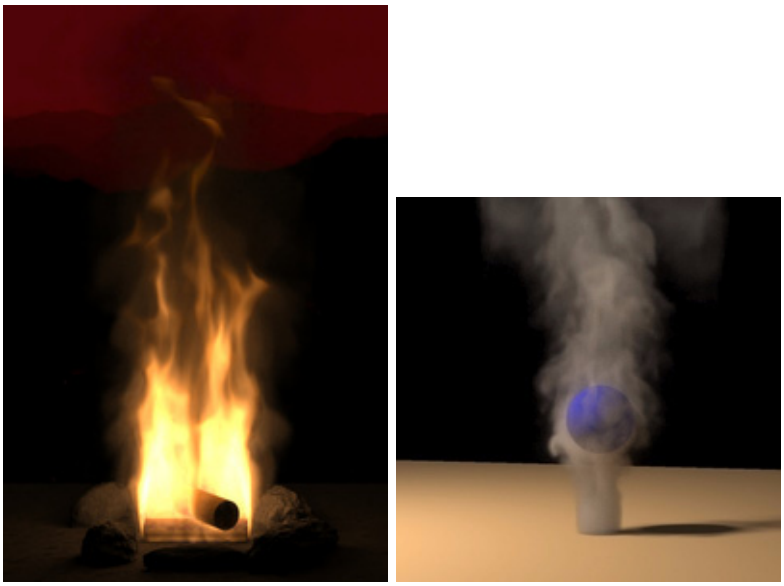

FIG. 1.3. *Image obtained from* `http://www.cs.stanford.edu/~fedkiw`

are defined as the zeros of the level set function. The PDEs that govern the motion of the interface can be derived from a variational principle. In many other cases, the interface motion is governed by well known physics. In fact, in the original level set paper [72], a level set function is used to distinguish burnt and unburnt regions in flame propagation problems. Fedkiw and collaborators used level set methods to simulate diverse physical phenomena such as splashing water, flame propagation, and detonation waves. When the results are rendered on the screen, they become very effective and realistic renderings of natural phenomena suitable for special effects in movie productions. The reader can find a detailed description and references in [69]. Figure 1.3 provides two of such simulations.

Hence, we will define a "level set method" for an image as a method that analyzes and manipulates the level sets of a given continuous function that we call an image.

Finally, there is a collection of level set numerics, consisting of mostly numerics for general Hamilton-Jacobi Equations and compressible and incompressible Fluid Dynamics. These methods are not limited only to the pure level set formulations. They can also be used to solve other PDE based image models. The basic numerics started in [72, 73], and generalizations have been carefully documented in [68]. Some new ones can be found in [34, 56, 81, 98, 100, 101]. Additionally, we mentioned [96], which addresses the issue of regularization.

Ideas originating in this type of numerics, e.g. ENO interpolation [49], have been used to develop wavelet based methods which minimize ringing, or Gibbs' phenomena at edges [19].

2. Level Set Method in a Nutshell

A significant number of problems in science reduce to the study of the evolution of curves, which are usually the boundaries between different media. These curves (or interfaces) move according to their own geometries or the laws of physics associated with the problem. They break up, merge, or disappear during the course of time

evolution. These are the every day challenges of most of the conventional methods. The level set method [72], however, handles these topological changes in the curves *"with no emotional involvement".* Since its first introduction, there has developed a powerful level set calculus used to solve a great variety of problems in fluid dynamics, materials sciences, computer vision, and computer graphics, to name a few topics. We refer to [68] for an extensive exposition of the level set calculus. See also [45] for related theoretical exposition.

Typically, one can write a general level set algorithm in three steps enumerated below:

1. Initialize/reinitialize ϕ at $t = t^n$.
2. Construct/approximate $H(t, x, \phi, D\phi, D^2\phi)$. (Occasionally higher derivatives also appear for which viscosity solution theory does not apply).
3. Evolve

$$\phi_t + H(t, x, \phi, D\phi, D^2\phi) = 0,$$

for $t = t_n + \Delta t$.

For image applications, ϕ above can either be the image itself (e.g. deblurring applications) or an extra function that is used to process the given image (e.g. segmentation applications).

We will discuss the key components of the three steps in the following sections. More precisely, we will follow a conventional approach of describing the level set method and start our exposition for Step 2. Step 1 and Step 3 is typically implemented by suitable numerical methods that will be reviewed in the next section.

2.1. Basic formulation. For simplicity, we discuss the conventional level set formulation in two dimensions. The interface represented by a level set function is thus also referred to as a curve. However, the methodology presented in this section can be naturally extended to any number of space dimensions. There, the interface that is represented is generally called a hypersurface (in three dimensions, it is simply called a surface). We will use the words interface and curves, interchangeably.

In the level set method, the curves are implicitly defined as the zeros of a Lipschitz continuous function ϕ. This is to say that $\{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\}$ corresponds to the location of the embedded curve. See Figures 2.1, 2.2, and 2.3 for some examples. If we associate a continuous velocity field $\mathbf{v}$ whose restriction onto the curve represents the velocity of the curve, then at least locally in time, the evolution can be described as the Cauchy problem

$$\phi_t + \mathbf{v} \cdot \nabla\phi = 0, \qquad \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}),$$

where ϕ_0 embeds the initial position of the curve. To derive this, let us look at a parameterized curve $\gamma(s, t)$ and assume that $\partial\gamma/\partial t$ is the known dynamic of this curve. If we require that $\gamma(s, t)$ be the zero of the function ϕ for all time, i.e. $\phi(\gamma(s, t), t) = 0$ for all $t \geq 0$, then at least formally, the following equation

$$\phi_t + \frac{\partial\gamma}{\partial t} \cdot \nabla\phi(\gamma, t) = 0$$

is satisfied along γ. Extending $\partial\gamma/\partial t$ continuously to the whole domain will create such a velocity field.

In general, the velocity $\mathbf{v}$ can be a function of position $\mathbf{x}$ and some other geometrical quantities of the curve or other physical quantities that come with the problem.

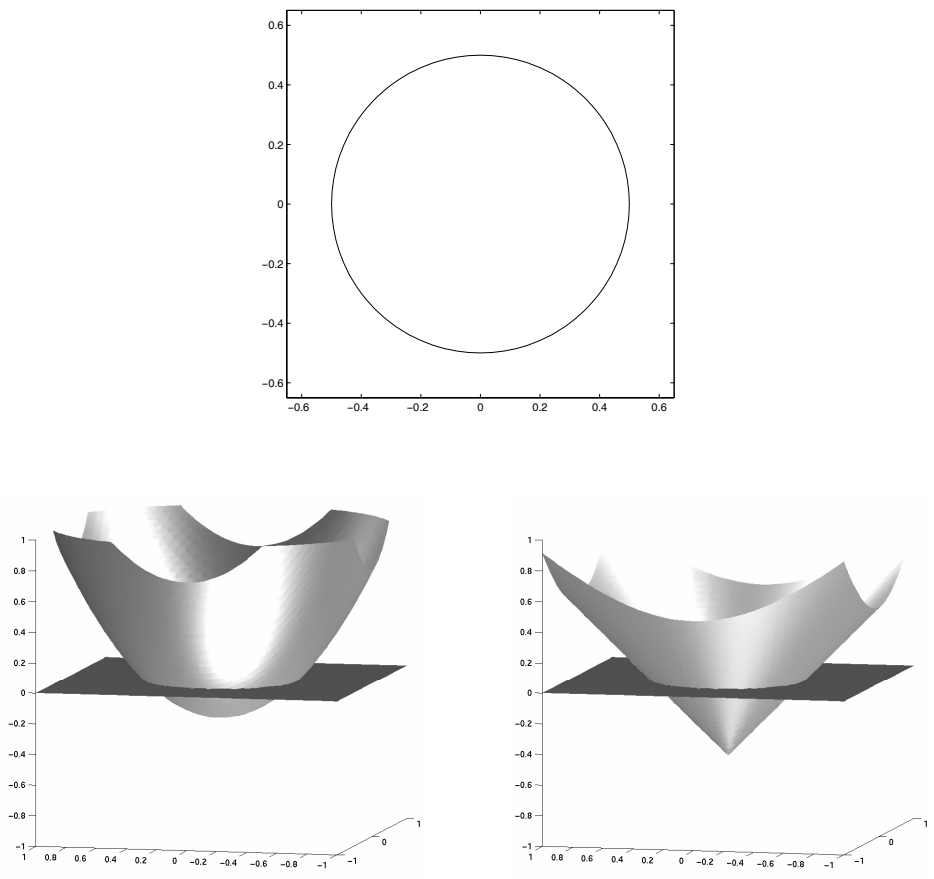

FIG. 2.1. *A circle embedded by different continuous functions.*

The equation can be written using the normal velocity:

$$v_n = \mathbf{v} \cdot \frac{\nabla u}{|\nabla u|}, \qquad \phi_t + v_n |\nabla \phi| = 0. \tag{2.1}$$

We note that these equations are usually fully nonlinear first order Hamilton-Jacobi or second order degenerate parabolic equations, and with suitable restrictions, the theory of viscosity solutions [25] can be applied to guarantee well-posedness of the Cauchy problem.

Corresponding to the viscosity solution theory, there is a set of simple finite difference methods to construct approximation solutions. We refer the interested readers to [41, 45].

Finally, in the level set formulation, the surface integral of function f along the zero level set is defined via the surface integral

$$\int_{\mathbb{R}^d} f(x)\delta(\phi)|\nabla \phi| dx.$$

If $f \equiv 1$, this integral yields the arclength for curves in two dimensions, and surface

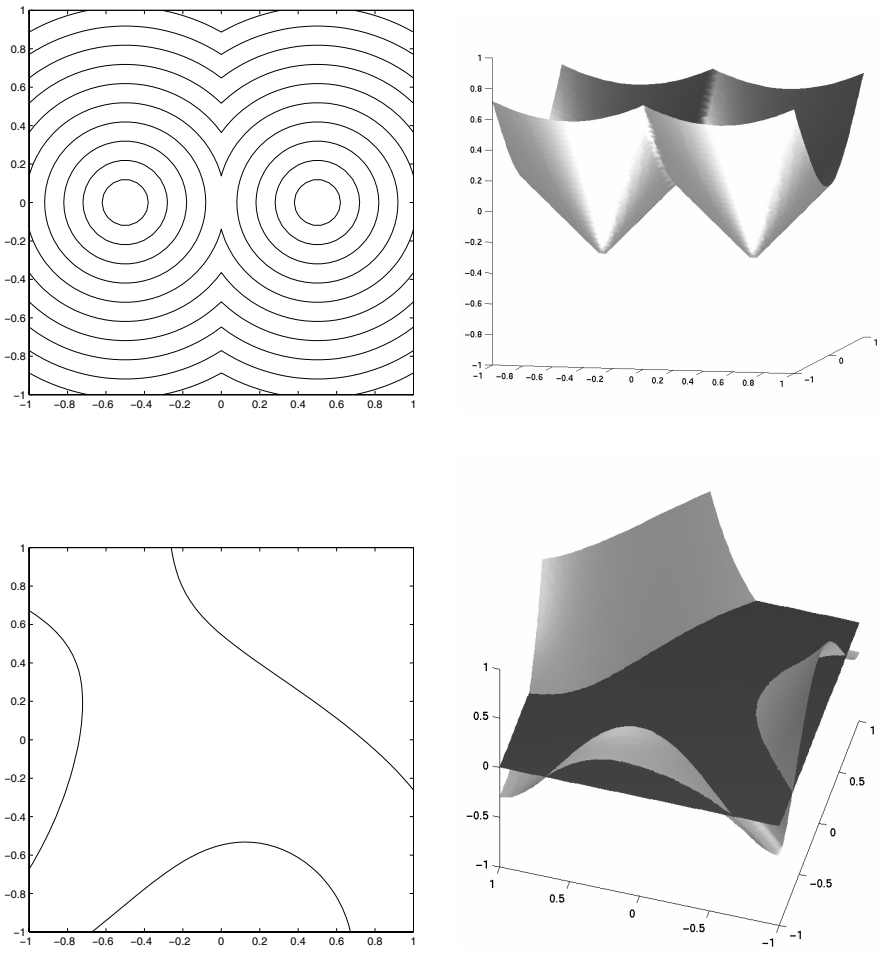

FIG. 2.2. *More complicated curves and their embedding level set functions.*

area in three dimensions. Volume integrals are defined as

$$\int_{\mathbb{R}^d} f(x)H(\phi)|\nabla\phi|dx,$$

where $H(x) = 1$ for $x \geq 0$ and $H(x) = 0$ for $x < 0$. We refer the discussion on the related computational issues to [33].

2.2. Reshaping the level set function. In many situations, the level set function will develop steep or flat gradients leading to problems in numerical approximations. It is then needed to reshape the level set function to a more useful form, *while keeping the zero location unchanged.* One way to do this is to perform what is called distance reinitialization [94] by evolving the following PDE to steady state:

$$\phi_\tau + \text{sgn}(\phi_0)(|\nabla\phi| - 1) = 0, \quad \phi(x, \tau = 0) = \phi_0(x). \tag{2.2}$$

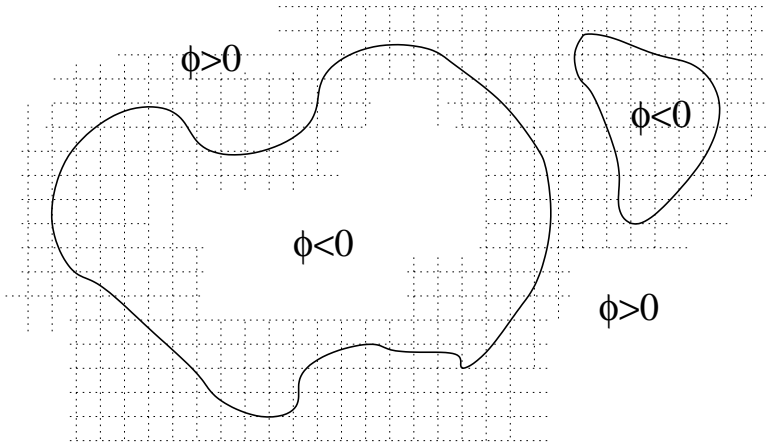

FIG. 2.3. *Two closed curves that are implictly embedded by a single level set function defined on the grid.*

Here ϕ_0 denotes the level set function we have before the reintialization. If we evolve the solution to steady state over the computational domain, the solution ϕ becomes the signed distance function to the interface $\{\phi_0 = 0\}$. One can understand the mechanism of this approach from the following scenario: in the region in which ϕ_0 is positive, $\phi_\tau < 0$ whenever $|\nabla\phi| > 1$; therefore, the value of ϕ will decrease, and consequently, $|\nabla\phi|$ will become closer to 1. Notice that $\phi_\tau \equiv 0$ wherever $\phi_0 \equiv 0$, since $\text{sgn}(0) = 0$. See figure 2.4.

Another equivalent approach is to solve the eikonal equation

$$|\nabla\phi| = 1$$

with the boundary condition $\phi = 0$ on $\{\phi_0 = 0\}$. A common numerical approach, e.g. [76] is to run distance reinitialization (2.4) with a high order accurate method for a short amount of time, so that in a thin tube around $\{\phi_0 = 0\}$, ϕ is now the distance function. Then fix the values of ϕ in this tube as boundary conditions, and use fast sweeping or fast marching methods to solve the eikonal equations. We shall discuss the sweeping method in the next section.

We remark first that for most applications, the reinitialization is only needed for a neighborhood around the zero level set, and the diameter of this neighborhood depends on the discretization of the partial derivatives in the PDE. This implies that only a few time steps in τ are needed. Next we note that it is important to solve (2.2) using a high order discretization method. Otherwise, the location of the original interface will be perturbed noticeably by the numerical diffusion. Finally, reinitialization globally in the computational domain will prevent new zero contours from appearing. Thus, one needs to be careful if emergence of new level contours is of interest. In many image segmentation tasks, this is important, and we shall comment on this in a later section.

2.3. Extending quantities off the normals of the interface. In many models, one can only derive the interface velocity v_n in equation (2.1) along Γ. It is necessary to create a continuous velocity field defined on the whole domain Ω or at least in a tubular neighborhood of Γ whose restriction on Γ agrees with the known

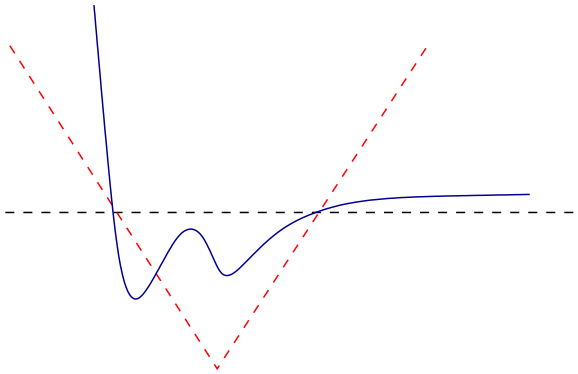

FIG. 2.4. *The result of the reinitialization of the solid curve is depicted by the cusp shown in the dashed line. The horizontal dashed line denotes the axis* $y = 0$.

interface velocity. One common way to obtain such a velocity field is to solve the following boundary value problem:

$$\operatorname{sgn}(\phi)\nabla w \cdot \nabla\phi = 0, \quad \text{with} \quad w|_\Gamma = v_n, \tag{2.3}$$

or equivalently, to solve for the steady state of the time dependent equation:

$$w_t + \operatorname{sgn}(\phi)\nabla w \cdot \nabla\phi = 0, \tag{2.4}$$

with any initial data w_0 whose restriction on Γ matches v_n.

The interpretation of this approach is that v_n will be propagated as a constant along the characteristics of the PDE (2.3), emanating from Γ, parallel to the surface normals. See figure 2.5. Fast sweeping [56, 98, 100, 106] or fast marching [102, 82] can be used to solve the first equation while a higher order accurate Hamilton-Jacobi solver can be used for the second [73]. In the next section, we will briefly describe some popular discretizations.

2.4. Tracking quantities defined on the fronts using level set method. So far we have described the basic level set method that enables us to move curves and surfaces normal to themselves by the prescribed velocities. We have concentrated on describing how the physical location of the curves and surfaces changes. In many applications, including image processing and computer vision, we need to track quantities that are defined on the surfaces. In this section, we review some techniques for doing this.

Let $\tilde{f} : \Gamma \mapsto X$ denote the quantity defined on Γ, the zero level set of ϕ, and $\tilde{f}$ satisfy

$$\tilde{f}_t + Q_\Gamma \tilde{f} = 0, \quad \tilde{f}(x, t = 0) = \tilde{f}_0(x), \tag{2.5}$$

where Q_Γ denotes some differential operator on Γ. This equation determines how $\tilde{f}$ is changing on Γ. Let $f : U \subset \mathbb{R}^d \mapsto X$ be a function defined in a neighborhood U of Γ, and $f|_\Gamma \equiv \tilde{f}$. Here $\mathbb{R}^d$ is the ambient space of Γ; i.e. $\phi : \mathbb{R}^d \mapsto \mathbb{R}$, and $\Gamma = \{x : \phi(x) = 0\}$. In a typical level set method, instead of solving (2.5) directly on Γ, one solves the corresponding PDE in $\mathbb{R}^d$:

$$f_t + Qf = 0,$$

so that the restriction of $f(t)$ to Γ matches with $\tilde{f}(t)$ for $t \geq 0$. At this point, it is natural to ask what Q is, given the Q_Γ? In many applications, the form of Q *is* the center of the study, and it might be more convenient to track an alternative quantity, g in order to obtain an equation that is easier to solve. See the recent paper [54] for such an example. In the next paragraph, we discuss another example of this situation.

Of the surfaces. In [48], Osher and Harabetian introduced a method for tracking the parametrization of closed curves. Let ϕ denote the level set function that embeds the surface of interest. The idea is to introduce an auxiliary function ψ such that (ϕ, ψ) forms a coordinate system near the zero level set of ϕ.

Let the family of closed curves $\Gamma(s,t) = (x(s,t), y(s,t))$ be parameterized by s and t. We want to evolve for example, $\Gamma(s,0)$ to time t, by the level set functions:

$$\phi(x(s,t), y(s,t), t) \equiv 0, \;\; \psi(x(s,t), y(s,t), t) \equiv s.$$

However, ψ is not a single valued function over a closed curve if it is defined this way. The authors then proposed to evolve the Jacobian

$$J = \det \begin{bmatrix} \varphi_x & \varphi_y \\ \psi_x & \psi_y \end{bmatrix}$$

instead of ψ to circumvent this problem. J has to be nonzero so that we can express (x_s, y_s) by $(-\phi_y, \phi_x)/J$. Thus, in order to track the tangential motion we evolve

$$J_t + \nabla \cdot (J\mathbf{v}) = 0$$

in addition to

$$\phi_t + \mathbf{v} \cdot \nabla\phi = 0.$$

Finally, we briefly describe the systematic approach that was pioneered in Cheng's thesis [21], and later on in [7] for solving PDE's on surfaces for image processing purpose. A similar approach was adopted by [109] to study surfactants on interfaces that move in time. For simplicity, we assume the zero level set to be fixed in time.

Consider the surface gradient $Q_\Gamma = \nabla_\Gamma$ that maps scalar functions defined on Γ to the tangent bundle of Γ. The key notion is to replace ∇_Γ by a suitable projection of the gradient operator ∇ in $\mathbb{R}^d$. The corresponding projection operator is a linear operator defined by:

$$\mathcal{P}_v = \mathcal{I} - \frac{v \otimes v}{|v|^2},$$

or equivalently, as a matrix, $\mathcal{P}_v$ can be written as

$$(\mathcal{P}_v)_{ij} = \delta_{ij} - \frac{v_i v_j}{|v|^2},$$

where v is a vector in $\mathbb{R}^d$, and δ_{ij} is the Kronecker delta function. For $x \in \Gamma$, and v the normal of Γ at x, $\mathcal{P}_v$ projects vectors onto the tangent plane of Γ at x.

Recall that $\Gamma = \{\phi = 0\}$, and $\nabla\phi$ is parallel to the normal of Γ. It can be proved that ∇_Γ and $\mathcal{P}_{\nabla\phi}\nabla$ are equivalent on Γ. Thus, for scalar functions f,

$$\nabla_\Gamma f = P_{\nabla\phi}\nabla f,$$

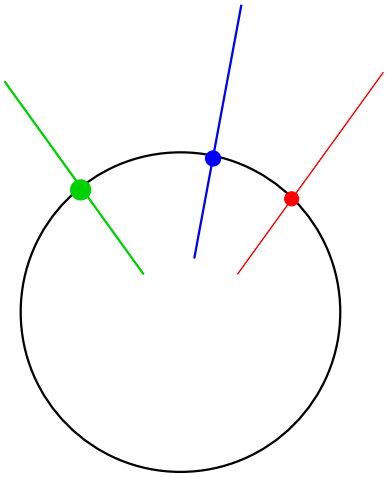

FIG. 2.5. *Quantities are extended off the zero level set in the normal directions.*

and for surface divergence of vector fields F,

$$\nabla_\Gamma \cdot F = P_{\nabla\phi} \nabla \cdot F.$$

Let us illustrate this approach with a few examples. Consider a continuous function $\tilde{f}$ defined on Γ, a surface in $\mathbb{R}^3$, and a given vector field v defined in the tangent bundle of Γ. If the zeros of $\tilde{f}$ embed the curve of interest (call it C) on Γ, then by solving

$$\tilde{f}_t + v \cdot \nabla_\Gamma \tilde{f} = 0,$$

one obtains the evolution of the curve constrained on the surface. Correspondingly, the extension f of $\tilde{f}$ in $\mathbb{R}^3$ is another level set function, whose zero level set intersects with that of ϕ on C, and the corresponding PDE in $\mathbb{R}^3$ is

$$f_t + v \cdot \mathcal{P}_{\nabla\phi} \nabla f = 0,$$

or, by property of $\mathcal{P}_v$,

$$f_t + \mathcal{P}_{\nabla\phi} v \cdot \nabla f = 0.$$

To perform distance reinitialization on $\tilde{f}$, one can evolve

$$f_\tau + \operatorname{sgn}(f_0)(|\mathcal{P}_{\nabla\phi} \nabla f| - 1) = 0.$$

As an example, total variation diminishing flow of an image u, defined on a surface Γ, takes the form

$$\mathcal{E}(u) = \int_{\mathbb{R}^3} |\mathcal{P}_{\nabla\phi} \nabla u| \delta(\phi) |\nabla\phi| dx,$$

and the corresponding gradient descent equation becomes

$$u_t = \mathcal{P}_{\nabla\phi} \nabla \cdot \left(\frac{\mathcal{P}_{\nabla\phi} \nabla u}{|\mathcal{P}_{\nabla\phi} \nabla u|} \right),$$

where the right hand side corresponds to the geodesic curvature, and can also be written as

$$\nabla \cdot \left(\frac{\mathcal{P}_{\nabla\phi} \nabla u}{|\mathcal{P}_{\nabla\phi} \nabla u|} |\nabla \phi| \right) \frac{1}{|\nabla \phi|}.$$

2.5. Limitations of the Level Set Methods. The original idea in the level set method is to use the sign of a given function to separate the given domain into two disjoint regions, and use the continuity of the level set function near its zero to define the boundary of these disjoint regions. One realizes that it can be more complicated to extend this idea to handle non-simple curves, and multiple phases. An equally important issue is to solve the problem at hand in obtaining a reasonable quality without excessive complexity. We refer the readers to [87, 103, 107] for level set methods for multiple phases, [11, 66] for higher codimensions and [86] for open curves, and [76, 93, 92] for localization. We also refer to [34] for a hybrid particle level set method that is designed to lessen the numerical diffusion effect for some class of problems.

3. Numerics for conservation laws and Hamilton-Jacobi Equations

Numerical solution of conservation laws has been an active field of research for quite some time. In particular, UCLA researchers have been major contributors to the field. The finite difference methods commonly used in the level set methods (in particular, those related to Hamilton-Jacobi equations) are developed under the general philosophy of the Godunov procedure and the nonlinear ENO reconstruction techniques for avoiding oscillations in calculations. As a result, upwinding and ENO interpolations become the indispensable parts of the algorithms documented here.

In what follows, we will first describe the Godunov procedure in the context of solving conservation laws and Hamilton-Jacobi equations. We will also describe the ENO interpolation and compare the differences between its usage in conservation laws schemes and in Hamilton-Jacobi solvers. We refer the details to the book of Osher and Fedkiw [68] and the extensive references therein.

Let us introduce some notations that we shall use in this section. Let $\phi^n_{i,j}$ denote the value of $\mathbf{x}_{i,j} = (x_0 + i\Delta x, y_0 + j\Delta y) \in \Omega$ at time $t_n = t_0 + n \cdot \Delta t$. We shall assume that $\Delta x = \Delta y$.

DEFINITION 3.1. *(Finite difference operators) Given the values of u on the grid we first define the forward and backward difference operators:*

$$D^{\pm}_x u_{i,j} := \pm \frac{u_{i\pm1,j} - u_{i,j}}{\Delta x},$$

and

$$D^{\pm}_y u_{i,j} := \pm \frac{u_{i,j\pm1} - u_{i,j}}{\Delta y};$$

also the central difference operators:

$$D^0_x u_{i,j} := \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x},$$

and

$$D^0_y u_{i,j} := \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}.$$

3.1. The Godunov procedure. The Godunov procedure [46] developed for conservation laws started by looking at grid values as cell averages of the solution at time t_n. We then "build" a piecewise constant function whose value in each cell is the cell average. We solve the Riemann problem at cell boundaries "exactly" for an appropriate time step Δt. This involves following the characteristics and making sure that the Rankine-Hugoniot and entropy conditions are satisfied. Finally, we average the function at $t = t_n + \Delta t$ in each cell, and repeat the above steps.

In the context of certain conventional Hamilton-Jacobi equations, piecewise constant cell averages are replaced by a piecewise linear function that is continuous at the cell boundaries, and point values are updated. This is described in [4].

In high order schemes, cell averages are replaced by more accurate nonoscillatory reconstruction on the functions or the fluxes. We perform this reconstruction by ENO/WENO methods.

3.2. ENO/WENO interpolation. We want to approximate the value of the function f in the interval $I_i := [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, using the given values (or averaged values) of f on the grid nodes x_i and its neighbors. Two commonly used methods to get a k-th order approximation of f in I_i are spectral interpolation that is based on Fourier expansions and fixed order polynomial interpolation. Both approaches produce oscillations near the jumps in the function values. We will not comment on the Fourier based methods since the they are not particularly useful in this connection. Conventional polynomial interpolations usually use the function values on *all the grid points* within a certain fixed distance from x_i, *regardless of the smoothness of the interpolated function.* ENO interpolation, on the other hand, is a nonlinear procedure that is built on a "progression" of Newton's divided differences. By "progression", we mean that the procedure starts by building a linear reconstruction of f in I_i using *either* $f(x_i)$ and $f(x_{i-1})$ or $f(x_i)$ and $f(x_{i+1})$, depending on which pair of values will give a smoother reconstruction. Suppose the reconstruction from $f(x_i)$ and $f(x_{i-1})$ is selected, we then carry out the reconstruction using the values of f on either x_{i-2}, x_{i-1}, x_i or x_{i-1}, x_i, x_{i+1}. This procedure is iterated until the desired order of approximation is achieved. Newton's interpolation is natural in this framework, since one can incrementally compute the divided differences for interpolation. In addition, we can use the values of the divided differences as an indicator of the smoothness of the functions in the intervals formed by the grid points that are considered as stencil.

For conservative schemes approximating conservation laws, this ENO reconstruction is performed on the flux function f or the cell averages $\bar{u}$ by first reconstructing the integral of the solution u. For Hamilton-Jacobi equations, we perform the ENO reconstruction on the solution u.

In the ENO reconstruction procedure, only one of the k candidate stencils (grid points used for the construction of the scheme) covering $2k-1$ cells is actually used. If the function is smooth in the neighborhood of this $2k-1$ cells, we can actually get a $(2k-1)$-th order approximation, if we use all these grid values. This is the idea behind the WENO reconstruction. In short, WENO reconstruction uses a convex linear combination of all the potential stencils. The weights in the combination are determined so that the WENO reconstruction procedure behaves like ENO near discontinuities. As a result, WENO method use smaller stencils to achieve the same order of accuracy as ENO in smooth regions. Currently, our choice of scheme is the 5th order WENO. For details, we refer to the original papers [32, 49, 52, 59, 84, 85], and the review articles [83]. Recently, Shu and Balsara [3] developed even higher order WENO reconstructions.

There are successful adaptations of this ENO idea/philosophy to other frameworks. See [16, 19] for ENO wavelet decompositions for image processing, and [24] for an application of the ENO philosophy in Discontinuous Galerkin methods.

3.3. Numerics for equations with Hamiltonians $H(x,u,p)$ nondecreasing in u. We repeat here that any discussion of the numerical schemes cannot be detached from the solution theory of the equation in questions. This is especially important for nonlinear equations, since in general, discontinuities in the function values or in the derivatives develop in finite time. We are usually seeking a particular type of weak solution.

In 1983, Crandall and Lions introduced viscosity solution theory for a class of Hamilton-Jacobi Equations requiring Lipschitz continuous initial data and for which the Hamiltonians $H(x,u,p)$ is Lipschitz continuous and *non-decreasing in u*. Later in [25] in 1984, they proved the convergence to the viscosity solution of monotone, consistent schemes for Hamilton-Jacobi Equations with H independent of x and u. Souganidis [89] extended the results to include variable coefficients. Osher and Sethian contributed to the numerics of Hamilton-Jacobi Equation in their level set paper in 1988 [72]. This line of work was later generalized and completed in the paper by Osher and Shu [73] in 1993, in which the authors provided a family of numerical Hamiltonians analogous to the ENO schemes for conservation laws. WENO schemes using the numerical Hamiltonians described in [73] were introduced in [52]. The method of lines using TVD Runge-Kutta time discretization is used [84]. We first discretize the spatial derivatives and compute the appropriate approximation to the Hamiltonians,

$$\hat{H}(p_-, p_+; q_-, q_+),$$

with $p_\pm$, $q_\pm$ representing the left/right approximations of the derivatives, obtained from ENO/WENO reconstruction of the solution. They are higher order versions of the forward and backward divided differences of the grid functions:

$$p_\pm \sim D_x^\pm u_{i,j} := \pm \frac{u_{i\pm1,j} - u_{i,j}}{\Delta x},$$

and

$$q_\pm \sim D_y^\pm u_{i,j} := \pm \frac{u_{i,j\pm1} - u_{i,j}}{\Delta y}.$$

3.4. The Lax-Friedrichs schemes for the level set equation. Following the methods originally conceived for HJ equations $\phi_t + H(D\phi) = 0$ in [73], see also [72], and suppressing the dependence of H on x and y, we use the Local Lax-Friedrichs (LLF) flux

$$\begin{aligned}\hat{H}^{LLF}(p^+, p^-, q^+, q^-) = H(\frac{p^+ + p^-}{2}, \frac{q^+ + q^-}{2}) \\ -\frac{1}{2}\alpha^x(p^+, p^-)(p^+ - p^-) - \frac{1}{2}\alpha^y(q^+, q^-)(q^+ - q^-),\end{aligned}$$

for the approximation of H. In the above scheme,

$$\alpha^x(p^+, p^-) = \max_{p \in I((p^+,p^-), C \le q \le D} |H_{\phi_x}(p,q)|,$$

$$\alpha^y(q^+, q^-) = \max_{q \in I((q^+,q^-), A \le p \le B} |H_{\phi_y}(p, q)|,$$

$$I(a, b) = [\min(a, b), \max(a, b)],$$

and $p^\pm$, $q^\pm$ are the forward and backward approximations of ϕ_x and ϕ_y respectively.

3.5. Curvatures. In many applications, the curvature term

$$\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \text{ or } \nabla \cdot \frac{\nabla u}{|\nabla u|}$$

for the level set function ϕ or the image function u appears as a regularization. This term is approximated by finite differencing centered at each grid point. For convenience, let $(n^x_{i,j}, n^y_{i,j})$ denote the values of $\nabla u/|\nabla u|_\epsilon$ at the grid point $x_{i,j}$, and $\nabla u/|\nabla u|_\epsilon$ is a smooth approximation of $\nabla u/|\nabla u|$ (This avoids the issue of singularity at $|\nabla u|$ and is better for numerical computations). The curvature $\kappa_{i,j}$ is approximated by

$$\kappa^\epsilon_{i,j} := \frac{n^x_{i+1/2,j} - n^x_{i-1/2,j}}{\Delta x} + \frac{n^y_{i,j+1/2} - n^y_{i,j-1/2}}{\Delta y},$$

and

$$n^x_{i\pm 1/2,j} : = \frac{D^\pm_x u_{i,j}}{\sqrt{(D^\pm_x u_{i,j})^2 + D^0_y (S^\pm_x u_{i,j})^2 + \epsilon^2}},$$

$$n^\epsilon_{i,j\pm 1/2} := \frac{D^\pm_y u_{i,j}}{\sqrt{D^0_x (S^\pm_y u_{i,j})^2 + (D^\pm_y u_{i,j})^2 + \epsilon^2}},$$

where

$$S^\pm_x u_{i,j} = \frac{u_{i\pm 1,j} + u_{i,j}}{2}, \text{ and } S^\pm_y u_{i,j} = \frac{u_{i,j\pm 1} + u_{i,j}}{2}$$

are the averaging operators in the x and y direction and $\varepsilon > 0$ is quite small.

3.6. Time discretization. From the previous subsections, we know how to discretize the terms involving spatial derivatives. What remains is to discretize in time in order to evolve the system; i.e. we need to solve the following ODE system:

$$\frac{\partial}{\partial t}\phi_{i,j} = -\tilde{H}(\phi^n_{i-1,j}, \phi^n_{i+1,j}, \phi^n_{i,j}, \phi^n_{i,j-1}, \phi^n_{i,j+1}),$$

where $\tilde{H}$ is the numerical approximation of $H(\mathbf{x}, \phi, D\phi, D^2\phi)$. For example, if we use local Lax-Friedrichs for $H(\phi_x, \phi_y)$, and forward Euler for time, we end up having:

$$\phi^{n+1}_{i,j} = \phi^n_{i,j} - \Delta t\, H^{LLF}(x_i, y_j, D^x_+ \phi^n_{i,j}, D^x_- \phi^n_{i,j}). \tag{3.1}$$

Typically, we use 3rd order TVD Runge-Kutta scheme of [84], or the fourth order schemes of [90] to evolve the system, since higher order accuracy can be achieved while using larger time steps. To keep this description self-contained, we describe the 3rd order TVD RK scheme below: we wish to advance $u_t = \text{rhs}(u)$ from $t_n$to t_{n+1}.

1. $u_1 = u^n + \Delta t \cdot \text{rhs}(u^n)$;
2. $u_2 = \frac{3}{4}u^n + \frac{1}{4}u_1 + \frac{1}{4}\Delta t \cdot \text{rhs}(u_1)$;
3. $u_{n+1} = \frac{1}{3}u^n + \frac{2}{3}u_2 + \frac{2}{3}\Delta t \cdot \text{rhs}(u_2)$.

3.7. Algorithms for constructing the distance function. In the following subsections, we review some of the solution methods for the eikonal equation:

$$|\nabla u| = r(x, y), \quad u|_\Gamma = 0.$$

We present a fast Gauss-Seidel type iteration method which utilizes a monotone upwind Godunov flux for the Hamiltonian. We show numerically that this algorithm can be applied directly to equations of the above type with variable coefficients.

3.8. Solving eikonal equations. In geometrical optics [57], the eikonal equation

$$\sqrt{\phi_x^2 + \phi_y^2} = r(x, y) \tag{3.2}$$

is derived from the leading term in an asymptotic expansion

$$e^{i\omega(\phi(x,y)-t)} \sum_{j=0}^{\infty} A_j(x, y, t)(i\omega)^{-j}$$

of the wave equation:

$$w_{tt} - c^2(x, y)(w_{xx} + w_{yy}) = 0,$$

where $r(x, y) = 1/|c(x, y)|$, is the function of slowness. The level sets of the solution ϕ can be thus be interpreted as the first arrival time of the wave front that is initially Γ. It can also be interpreted as the "distance" function to Γ.

We first restrict our attention to the case in which $r = 1$. Let Γ be a closed subset of $\mathbb{R}^2$. It can be shown easily that the distance function defined by

$$d(\mathbf{x}) = \text{dist } (\mathbf{x}, \Gamma) := \min_{p\in\Gamma} |\mathbf{x} - p|, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2,$$

is the viscosity solution to equation (3.2) with the boundary condition

$$\phi(x, y) = 0 \text{ for } (x, y) \in \Gamma.$$

Rouy and Tourin [78] proved the convergence to the viscosity solution of an iterative method solving equation (3.2) with the Godunov numerical Hamiltonian approximating $|\nabla\phi|$. The Godunov numerical Hamiltonian function can be written in the following simple form for this eikonal equation:

$$H_G(p_-, p_+, q_-, q_+) = \sqrt{\max\{p_-^+, p_+^-\}^2 + \max\{q_-^+, q_+^-\}^2}, \tag{3.3}$$

where $p_\pm = D_\pm^x \phi_{i,j}$, $q_\pm = D_\pm^y \phi_{i,j}$, and $x^+ = \max(x, 0)$, $x^- = -\min(x, 0)$. The task is then to solve

$$H_G = 1$$

on the grid.

Osher [65] provided a link to time dependent eikonal equations by proving that the t-level set of $\phi(x, y)$ is the zero level set of the viscosity solution of the evolution equation at time t

$$\psi_t + |\nabla\psi| = 0$$

with appropriate initial conditions. In fact, the same is true for a very general class of Hamilton-Jacobi equations (see [65]). As a consequence, one can try to solve the time-dependent equation by the level set formulation [72] with high order approximations on the partial derivatives[73, 52]. Crandall and Lions proved that the discrete solution obtained with a consistent, monotone Hamiltonian converges to the desired viscosity solution [25].

Tsitsiklis [102] combined heap sort with a variant of the classical Dijkstra algorithm to solve the steady state equation of the more general problem

$$|\nabla\phi| = r(\mathbf{x}).$$

This was later rederived in [82] and [50]. It has become known as the fast marching method whose complexity is $\mathcal{O}(N\log(N))$, where N is the number of grid points. Osher and Helmsen[70] have extended the fast marching type method to somewhat more general Hamilton-Jacobi equations. Since the fast marching method is by now well known, we will not give details here on its implementation in this paper.

3.9. The sweeping idea. Danielsson [27] proposed an algorithm to compute Euclidean distance to a subset of grid points on a two dimensional grid by visiting each grid node in some predefined order. In [9], Boué and Dupuis suggest a similar "sweeping" approach to solve the steady state equation which, by experience, results in a $\mathcal{O}(N)$ algorithm for the problem at hand. This "sweeping" approach has recently been used in [98] and [108] to compute the distance function to an arbitrary data set in computer vision. In [106], it was proven that the fast sweeping algorithm achieves a reasonable accuracy in a (small) finite number of iterations independent of grid size. Using this "sweeping" approach, the complexity of the algorithms drops from $\mathcal{O}(N\log N)$ in the fast marching to $\mathcal{O}(N)$, and the implementation of the algorithms becomes a bit easier than the fast marching method in that no heap sort is needed.

This sweeping idea is best illustrated by solving the eikonal equation in $[0,1]$:

$$|u_x| = 1,\ u(0) = u(1) = 0.$$

Let $u_i = u(x_i)$ be the grid values and $x_0 = 0$, $x_n = 1$. We then solve the discretized nonlinear system

$$\sqrt{\max(\max(D_- u_i, 0)^2, \min(D_+ u_i, 0)^2)} = 1,\ \ u_0 = u_n = 0 \tag{3.4}$$

by our sweeping approach. Let us begin by sweeping from -1 to 1, i.e. we update u_i from $i = 0$ increasing to $i = n$. This is "equivalent" to following the characteristics emanating from x_0. Let $u_i^{(1)}$ denote the grid values after this sweep. We then have

$$u_i^{(1)} = \begin{cases} i/n, & \text{if } i < n \\ 1/n & i = n \end{cases}.$$

In the second sweep, we update u_i from $i = n$ decreasing to 0, using $u_i^{(1)}$. During this sweep, we follow the characteristics emanating from x_n. The use of (3.4) is essential, since it determines what happens when two characteristics cross each other. It is then not hard to see that after the second sweep,

$$u_i = \begin{cases} i/n, & \text{if } i \le n/2 \\ (n-i)/n & \text{otherwise.} \end{cases}$$

Thus, to update u_o, one only uses the immediate neighboring grid values and does not need the heap sort data structure. More importantly, the algorithm follows the characteristics with certain directions simultaneously, in a parallel way, instead of a sequential way as in the fast marching method. The Godunov flux is essential in the algorithm, since it determines what neighboring grid values should be used to update u on a given grid node o. At least in the examples presented, we only need to solve a simple quadratic equation and run some simple tests to determine the value to be updated. This simple procedure is performed in each sweep, and solution is obtained after a few sweeps.

3.10. Generalized closest point algorithms. In this subsection, we describe an algorithm that can be applied for constructing a level set implicit representation for a surface which is defined explicitly. It can also be used to extend the interface velocity to the whole computational domain.

In the spirit of the Steinhoff et.al. Dynamic Surface Extension [91], we can define functions that map each point in $\mathbb{R}^3$ to the space of (local) representations of surfaces (heretherto referred as surface elements). We can further define the distance of a point P and a surface element $\mathcal{S}$

$$\operatorname{dist}(P, \mathcal{S}) := \min_{y \in S}(P, y).$$

The 'surface element' can be for example the tangent plane, the curvature, or a NURB description of the surface.

Instead of propagating distance values away from the interface, *we propagate the surface element information along the characteristics* and impose conditions that e*nforce the first arrival property* of the viscosity solution of the eikonal equation. The challenge is to compute the exact distance from a given surface element and to derive the "upwinding" criteria for propagating the surface information throughout the grids.

Given a smooth parameterized surface $\Sigma : I_s \times I_t \mapsto \mathbb{R}^3$, our algorithm provides good initial guess for Newton's iterations on the orthogonality identity:

$$F(s_*, t_*; \mathbf{x}) = \begin{pmatrix} (\mathbf{x} - \Sigma(s_*, t_*)) \cdot \Sigma_s(s_*, t_*) \\ (\mathbf{x} - \Sigma(s_*, t_*)) \cdot \Sigma_t(s_*, t_*) \end{pmatrix} = 0,$$

where $\Sigma(s_*, t_*)$ is the closest point on the surface to $\mathbf{x}$. The initial guess in this case is simply the closest point of the neighbors of $\mathbf{x}$.

Let W denote the function that maps each point in space to its closest surface element on S. We can then write the algorithm as follows:

Algorithm: Let u be the distance function on the grids, and W be the corresponding generalized closest point function.

1. Initialize: give the exact distance to u, and the exact surface elements to W at grids near Γ. Mark them so they will not be updated. Mark all other grid values as ∞.
2. Iterate through each grid point E with index (i,j,k) in each sweeping direction or according to the fast marching heap sort.
3. For each neighbor P_l of E, compute $u_l^{tmp} = \operatorname{dist}(E, W(P_l))$
4. If $\operatorname{dist}(E, W(P_l)) < \min_k u(P_k)$, set $u_l^{tmp} = \infty$. This is to enforce the monotonicity of the solution.
5. Set $u(E) = \min_l u_l^{tmp} = u_\lambda^{tmp}$ and $W(E) = W(P_\lambda)$.

This procedure can be used e.g., to convert triangulated surfaces to implicit surfaces.

3.11. Further generalizations. For further generalizations of the sweeping method to solve more complicated Hamilton-Jacobi equations, such as those which arise in computing distance on a manifold:

$$H(u_x, u_y) = \sqrt{au_x^2 + bu_y^2 + 2cu_xu_y} = r(x, y), \text{ for } a, b > 0, \ ab > c^2,$$

and the equations using Bellman's formulae, we refer the readers to the recent papers [100, 56]. Recently, a very simple sweeping algorithm, based on the Lax-Friedrichs scheme, has been shown to work in great generality [55].

4. Segmentation Algorithms

The task of image segmentation is to find a collection of non-overlapping subregions of a given image. In medical imaging, for example, one might want to segment the tumor or the white matter of a brain from a given MRI image. In airport screening, one might wish to segment certain "sensitive" shapes, such as guns. There are many other obvious applications. Mathematically, given an image $u : \Omega \subset \mathbb{R}^2$(or $\mathbb{R}^3$)$\mapsto\mathbb{R}^+$, we want to find closed sets Ω_i satisfying

$$\Omega = \bigcup_{i=0}^{N-1} \Omega_i, \text{ and } \bigcap_{i=0}^{N-1} \Omega_i^{(0)} = \emptyset,$$

such that $\mathcal{F}(u, \Omega_i) = 0$, where $\mathcal{F}$ is some functional that defines the segmentation goals. Here, $\Omega_i^{(0)}$ denotes the interior of Ω_i. As in the example of finding tumors, typically, N is taken to be 2 (sometimes $N = 3$), and Ω_0 is taken to be the region corresponding to the tumor, while Ω_1 contains everything else. It is then natural to devise a level set method to perform this task, by representing, for example, Ω_0 as the region in which ϕ is non-negative. A slightly more general statement would be to perform segmentation from a given set of images u_j that come from different sources. For example, one might be interested to segment stealth fighter jets from both the conventional radar signals and also the infrared images.

Very often, the definition of what belongs to the "desired" regions depends on the gray scale intensity of the given image, and the problem of finding such regions is formulated as a variational problem; i.e. the solution minimizes some "energy". In a standard level set method, ϕ is used to represent Ω_i and $\partial\Omega_i$. This is the setting of our discussion. In this section, we describe some level set segmentation methods based on this type of definition.

4.1. Variational level set method. Assume that the energy functional $\mathcal{E}$ is an integral operator on u over Ω_0:

$$\mathcal{E}(u, \Omega_0) = \int_{\Omega_0} F(u(\mathbf{x}))d\mathbf{x},$$

and the non-positive region of ϕ defines Ω_0; i.e. $\{\phi \leq 0\} = \Omega_0$. The key idea of the variational level set method formulated in [107] is that the above integral can be written as

$$\int_\Omega F(u(\mathbf{x}))d\mathbf{x} = \int_{\mathbb{R}^2} \chi_\Omega(\mathbf{x})F(u(\mathbf{x}))d\mathbf{x} = \int_{\mathbb{R}^2} H(-\phi)F(u)d\mathbf{x},$$

where H is the Heaviside function: $H(x) = 1$ if $x \leq 0$ and $H(x) = 0$ elsewhere. One can then try to find the minimizer ϕ for this energy.

4.2. The Chan-Vese algorithm. This is closely related to the classical Mumford-Shah algorithm [63], but uses a simple level set framework for its implementation. We present the original Chan-Vese segmentation algorithm [18], and discuss various aspects of this algorithm.

4.2.1. Basic formulation. The minimization problem is:

$$\min_{\phi\in BV(\Omega),c_1,c_2\in\mathbb{R}^+} \mathcal{E}(\phi,c_1,c_2;u_0),$$

where the energy is defined as

$$\mathcal{E}(\phi,c_1,c_2;u_0) = \mu\int_\Omega \delta(\phi)|\nabla\phi|dx + \lambda_1\int_\Omega |u_0-c_1|^2H(\phi)dx + \lambda_2\int_\Omega |u_0-c_2|^2(1-H(\phi))dx. \quad (4.1)$$

Intuitively, one can interpret from this energy that each segment is defined as the subregions of the images over which the average of the given image is "closest" to the image value itself in L_2-norm. The first term in the energy measures the arclength of the segment boundaries. Thus, minimizing this quantity provides stability of the algorithm as well as preventing fractal like boundaries from appearing.

If one regularizes the δ function and the Heaviside function by two suitable smooth functions δ_ϵ and H_ϵ, then formally, the Euler-Lagrange equations can be written as

$$\partial_\phi\mathcal{E} = -\delta_\epsilon(\phi)\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} - \nu - \lambda_1(u_0-c_1)^2 + \lambda_2(u_0-c_2)^2\right] = 0, \quad (4.2)$$

with natural boundary condition

$$\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0 \text{ on } \partial\Omega.$$

$$c_1(\phi) = \frac{\int_\Omega u_0(\mathbf{x})H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}}, \quad (4.3)$$

and

$$c_2(\phi) = \frac{\int_\Omega u_0(\mathbf{x})(1-H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}{\int_\Omega(1-H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}. \quad (4.4)$$

4.2.2. Discretization. A common approach to solve the minimization problem is to perform gradient descent on the regularized Euler-Lagrange equation (4.2); i.e. solving the following time dependent equation to steady state:

$$\frac{\partial\phi}{\partial t} = -\partial_\phi\mathcal{E} = \delta_\epsilon(\phi)\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} - \nu - \lambda_1(u_0-c_1)^2 + \lambda_2(u_0-c_2)^2\right]. \quad (4.5)$$

Here, we remind the readers that $c_1(\phi)$ and $c_2(\phi)$ are defined in (4.3) and (4.4).

In Chan-Vese algorithm, the authors regularized the Heaviside function used in (4.3) and (4.4):

$$H_{2,\epsilon}(z) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan(\frac{z}{\epsilon})\right),$$

and define the delta function as the derivative of it:

$$\delta_{2,\epsilon}(z) = H'_{2,\epsilon}(z).$$

Equation (4.5) is then discretized by a semi-implicit scheme; i.e. to advance from $\phi^n_{i,j}$ to $\phi^{n+1}_{i,j}$, the curvature term right hand side of (4.5) is discretized as described in the previous section using the value of $\phi^n_{i_\pm,j_\pm}$, except for the diagonal term $\phi_{i,j}$, which uses the implicitly defined $\phi^{n+1}_{i,j}$. The integrals defining $c_1(\phi)$ and $c_2(\phi)$ are approximated by simple Riemann sum with the regularized Heaviside function defined above. ϕ_t is discretized by the forward Euler method: $(\phi^{n+1}_{i,j} - \phi^n_{i,j})/\Delta t$. Therefore, the final update formula can be conceptually written as

$$\phi^{n+1}_{i,j} = \frac{1}{1+\alpha_\kappa}\left(\phi^n_{i,j} + G(\phi^n_{i-1,j}, \phi^n_{i+1,j}, \phi^n_{i,j-1}, \phi^n_{i,j+1})\right),$$

where $\alpha_\kappa \geq 0$ comes from the discretization of the curvature term. If the scheme is fully explicit, $\alpha_\kappa = 0$ and G would depend on $\phi^n_{i,j}$. In the paper, the authors used $\Delta x = \Delta y = 1$, $\epsilon = 1$, and $\Delta t = 0.1$. This implies that the delta function is really a regular bump function that puts more weight on the evolution of the zero level set of ϕ. See Figures 4.1 and 4.2 for some results of this algorithm applied to brain segmentation.

Finally, it is also possible but not advisable in this (unusual) case because new zero level sets are likely to develop spontaneously, (see section 1.4.5) to replace the δ function in front of the curvature term $|\nabla\phi|$ [61]. The equation then becomes independent of the choice of the level set function used, i.e. the problem becomes morphological [2].

4.3. Fast one-pass segmentation algorithms. Recently, Gibou and Fedkiw [44], and Song and Chan [88], proposed some fast methods that are based on the Chan-Vese level set segmentation formulation. These algorithms are built upon flipping the values of ϕ at each grid point/pixel from positive to negative or vice versa according to *a rule* $\mathcal{R}$, and contain 4 main steps:

1. Initialize $\phi^0 : \Omega \mapsto \{-1, 1\}$.
2. Advance: for each grid point, set $\phi^{n+1}(x) = -\phi^n(x)$ if $\mathcal{R}(\phi^{n+1}, \phi^n, x) = 1$.
3. (Perform regularization if needed.)
4. Repeat until $\phi^{n+1} \equiv \phi^n$.

For example, in Gibou and Fedkiw's algorithm, $\mathcal{R}(\phi^{n+1}, \phi^n) = 1$ if

$$V(\phi^n) \cdot \text{sign}(\phi^n) < 0;$$

here V corresponds to the fitting term in the Euler-Lagrange equation:

$$V(\phi^n, x) := -\lambda_1(u_0 - c_1(\phi^n))^2 + \lambda_2(u_0 - c_2(\phi^n))^2.$$

(Note that the case $V = 0$ is implicitly defined). In this algorithm, Step 3 essentially provides regularization to the segment boundaries. Without it, fractal-like boundaries may develop.

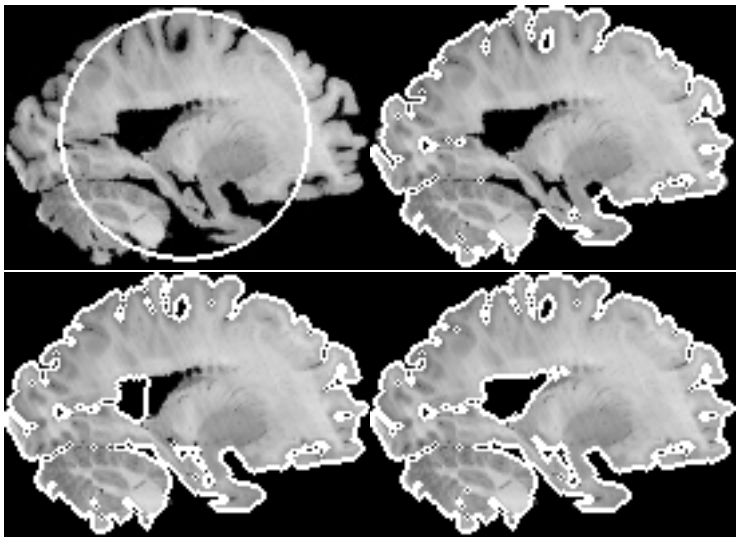

FIG. 4.1. *Brain segmentation in 2D.*

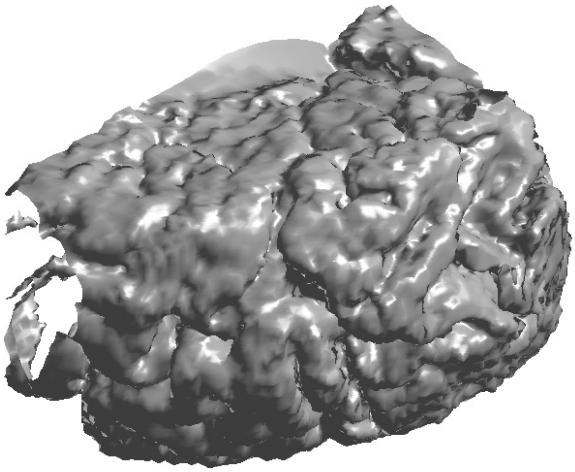

FIG. 4.2. *Brain segmentation in 3D.*

In Song and Chan's algorithm, the key observation is that only the signs of the the level set function matter in the energy functional. This can easily be seen from the model defined in equation (4.1), in which one sees that the energy is a function of $H(-\phi)$. In this algorithm, $\mathcal{R}(\phi^{n+1}, \phi^n)$ can be interpreted as the logical evaluation of the following inequality:

$$\mathcal{E}(\phi^{n+1}, c_1, c_2; u_0) \leq \mathcal{E}(\phi^n, c_1, c_2; u_0).$$

Hence, the sign of $\phi^n(x)$ is flipped *only if the energy (4.1) is non-increasing.* This provides stability of the algorithm when compared to Gibou and Fedkiw's in which there is no checking on the energy descent, at the cost of some speed of implementation.

We remark that there is a close connection between these two "level set" based methods to the "Γ-convergence" base methods. The Chan-Vese segmentation method can be approximated by the following variational problem:

$$E_\epsilon(u, c_1, c_2; u_0) := \mu \int \epsilon|\nabla u|^2 + \frac{1}{\epsilon} W(u)dx + \lambda_1 \int u^2(u_0 - c_1)^2 + \lambda_2 \int (1-u)^2(u_0 - c_2)^2 dx,$$

where $w(u) = u^2(1-u)^2$, and ϵ is a small positive number. Due to the strong potential $\epsilon^{-1}W(u)$, u will quickly be attracted to either 1 or 0, and consequently, the terms u^2 and $(1-u)^2$ correspond respectively to $H(\phi)$ and $1 - H(\phi)$ in (4.1), and $\epsilon|\nabla u|^2$ corresponds to the regularization of the length of $\partial\Omega_i$. Essentially, one can interpret the Gibou-Fedkiw or Song-Chan algorithm as performing a one step projection to the steady state that results from the stiff potential W. This is delineated in the current work of Esedoglu and Tsai [37].

4.4. Segmentation of multiple "phases". There are efforts to generalize the level set methods for multiphase computation. For example, in [107], each partition Ω_i is represented by a level set function ϕ_i. It is then important to enforce the constraints that 1) the regions represented do not overlap ($\bigcap_{i=1}^N \{\phi_i < 0\} = \emptyset$), and 2) there are no unclaimed regions; i.e. every point in Ω belongs to certain Ω_i ($\Omega = \bigcup_{i=1}^N \{\phi_i \leq 0\}$). Interesting formulae are derived in the variational setting to enforce these two conditions. However, this approach may be expensive when the number of phases is large.

In [103], the authors use the sign of the level set functions ϕ_j as a binary coding for the phases, each assigned a non-negative integer value. Suppose there are four phases, Ω_i, $i =, \cdots, 3$, and two level set functions ϕ_0 and ϕ_1 are used for their representation. One can then write, for instance,

$$\Omega_0 = \{\phi_0 \geq 0\} \bigcap \{\phi_1 \geq 0\},$$

$$\Omega_1 = \{\phi_0 \leq 0\} \bigcap \{\phi_1 \geq 0\},$$

$$\Omega_2 = \{\phi_0 \geq 0\} \bigcap \{\phi_1 \leq 0\},$$

$$\Omega_3 = \{\phi_0 \leq 0\} \bigcap \{\phi_1 \leq 0\}.$$

To full generality, write the phase number i in binary format $i = \sum_{k=0}^{n-1} c_k \cdot 2^k$, where c_k takes on either 0 or 1. Then one way of using $\{\phi_k\}_{k=0}^{n-1}$ level set functions to represent Ω_i is to identify

$$\Omega_i = \bigcap_{k=0}^{n-1} \{\mathbf{x} \in \Omega : (1 - c_k) \cdot \phi_k(\mathbf{x}) \geq 0\}.$$

A drawback of this approach is the potential mis-identification of what is supposed to be categorized as one single phase to two or more "different" phases, since the

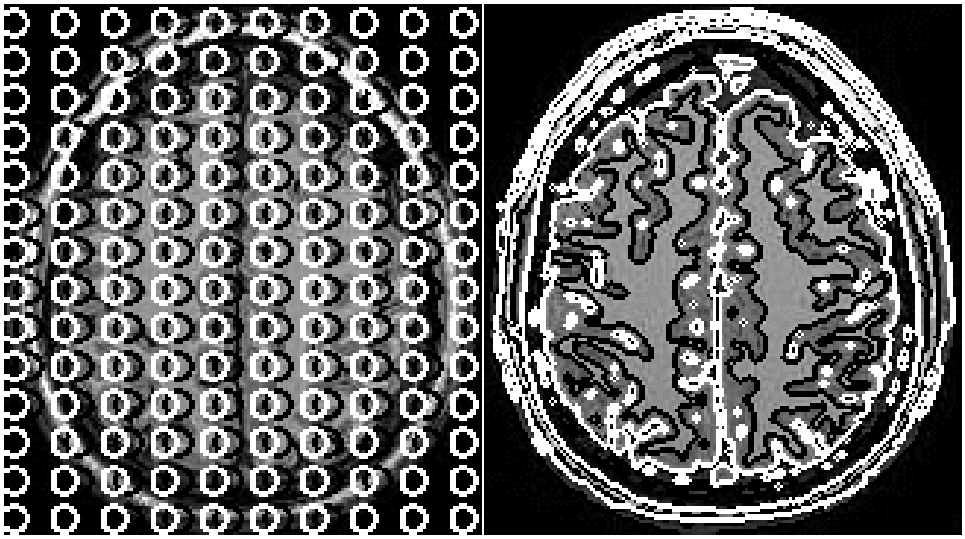

FIG. 4.3. *Initialization.*

formulation really comes with 2^n phases with n level set functions. In the Chan-Vese algorithm for example, it is possible that the image u has the same average in two different segments. Another drawback is the possible miscalculation of the arc length/surface area of each phase, when two phase boundaries are forced to collapse into one and may be given more weight than others. Related to the above drawbacks, an important but so far untouched (to the best of our knowledge) problem in the level set world is to determine the optimal number of phases in certain segmentation problems.

4.5. Discussion. One of the successful features reported in [18] is the emergence of new interior contour. As we mentioned earlier, if one enforces the level set function to be the distance function to the existing interfaces or replace the delta function by $|\nabla\phi|$ and computes locally, then the existing interfaces are only allowed to merge or disappear. The authors attributed the possibility of new interior contour emergence to their particular choice of delta function that has non-compact support. One common approach in getting around this problem is to initially seed many small circles that are densely distributed throughout the given image and let them gradually merge and evolve to a number of larger contours. See figures 1.4.3

This approach seems to capture the interior contour pretty well. While the statements about the nonlocal effect of the particular delta function used in Chan-Vese are valid, more careful study is called for to compare the degree of regularization, and diameter of the interior of any segmentation, to the possibility of the emergence of new interior contour.

We would also like to comment that the iterative approach adopted by Chan and Vese can be regarded as a version of Gauss-Jacobi iterations for the nonlinear Euler-Lagrange equation (4.2). This statement can be supplemented by looking at the same approach applied to the linear equation:

$$u_t = \Delta u.$$

The complexity of both approaches is proportional to N^2, the total number of pixels. This is verified by the computations of Song in [36]. We remark that it is possible to speed up the gradient flow in the Chan-Vese algorithm by a splitting method described in [43].

There are many new (and old) "level set based" segmentation algorithms that discard the continuity of the level set function and propose, instead, to model the segmentation problem as a completely discrete, pixel-by-pixel, algorithm. As in [44] and [88], these type of methods typically appear to be faster, and in some cases, more flexible in handling multiple phases. This trend seems to be going against the original spirit and *raison d'etre* of PDE based level set methods for image processing — the geometry of the interface is approximated at higher order accuracy through the assumed continuity of the level set function over the grid. This fact resonates with the criticism over phase field models for segmentation that there is no accurate representation of the interface, unless one refines the grid and resolves the stiff parameter ϵ^{-1} (something that is typically impossible to do for many image applications).

One should ask the question whether accurate representation of the phase boundaries is really needed for the problem at hand. Of course, there are applications in which geometrical quantities of the phase boundaries play important roles in the model; e.g. in the disocclusion application of Nitzberg-Mumford-Shiota [64] and also in the applications related to Euler's Elastica. In these type of applications, the "conventional" level set approach certainly has the advantage. In the cases where the geometrical quantities are not of importance, the piecewise constant model may be quite useful.

Our last comment is on the regularization term of the level set based segmentation methods. So far, popular choices have been the variants coming from minimizing the length of the interface. In denoising, as we have seen, this corresponds to L_1 regularization of the image gradient. It is possible that the features to be segmented, due to their origin, retain special orientations and are anisotropic. These type of applications exist, for example, in material sciences. In this case, one should look into the possible alternatives. We point out that Wulff energy is one such possible candidate. There, the regularization operator $\mathcal{R}$ is a function of the normal of the interface, i.e. $\mathcal{R}(\vec{n}) = div(\gamma(\vec{n}))$. In the common TV regularization, $\gamma(\vec{n}) = \vec{n}$. We refer to details to [71, 77] and [67].

5. Pushing the Limit

In this section, we will describe recent work corresponding to the classical applications that we listed above. We will see that this new work combines different ideas together to manipulate more complicated geometrical objects. However, the basic principle and spirit remains unchanged.

We start with total variation denoising. In [62], Meyer examined the total variation model of [80] more closely and proposed to decompose an image, u_0, into two portions, $u_0 = u + v$, where v contains the texture and noise parts of u_0 and can be written as the divergence of a vector field; i.e. $v = \text{div } g$ with the norm of v, $||v||_*$ defined as the infimum of L^∞ norms of such vectors g. Soon afterwards, in [104] and also [74], the authors proposed a numerical algorithm of approximating such a decomposition and combined it with other texture synthesis techniques to inpaint textured images [6].

In [11, 22], the authors provided a level set framework to represent and move curves on implicit surfaces or in free three dimensional space. This framework was then generalized to process images and even more general quantities such as vector

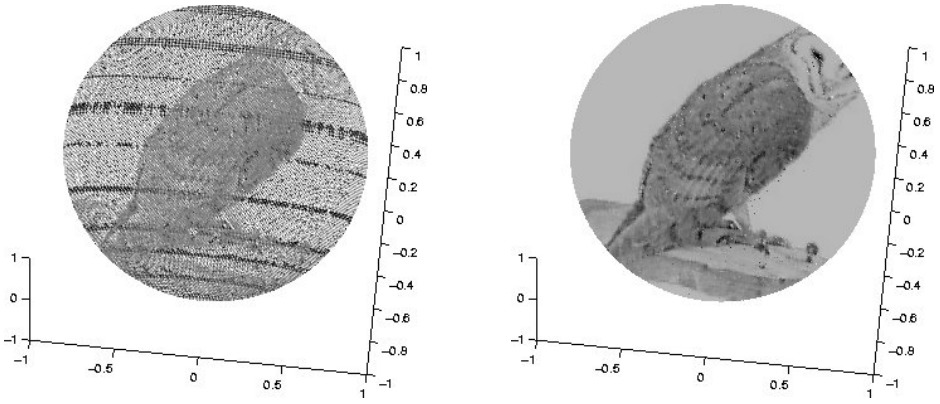

FIG. 5.1. *The image on the right is the denoised and inpainted result from the left.*

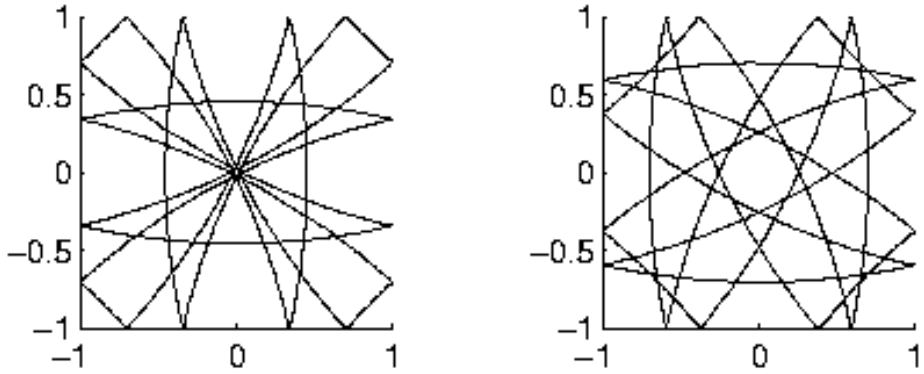

FIG. 5.2. *This figures shows some complicated curves with self intersections using the approach in [66].*

fields that are defined on nonflat surfaces [7]. Figure 5.1 shows inpainting over a sphere. This is one of the pioneering works on more complicated geometries in the level set framework. Generally speaking, the key is to raise the space dimension and/or the number of level set functions. For example, in [107], the authors used multiple level sets to solve a multiphase minimal surface problem. Chan-Vese further generalized the idea and applied it to image segmentations [103]. This was discussed in the previous section. Smereka [86] used multiple level sets to define spirals and study the formation of screw dislocations in crystal growth. Liao et al. [58] used this approach in brain morphing. Additionally, [87] also had an interesting level set approach to the multiphase computation that could be used in image segmentation.

As a last example, in the framework of [11, 22], a curve is represented as the intersection of two implicit surfaces, and the differential operators on surfaces are approximated by the projections of the related operators in the ambient space. This was then generalized to work on even more complicated geometrical objects commonly seen in dynamic geometrical optics [66]. This approach makes the manipulation of even more complicated curves and surfaces possible, see Figure 5.2.

FIG. 5.3. *The picture on the right shows the reconstructed surface from multiple images on the right.*

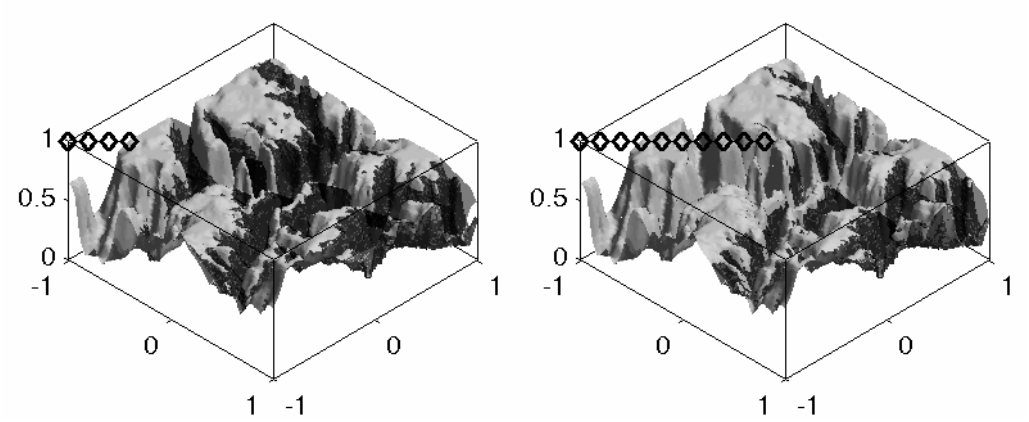

FIG. 5.4. *The black regions are invisible to the path indicated by the diamonds.*

5.1. Computer Graphics and Beyond. The problem of finding the visible/invisible region of a given surface configuration is a fundamental one. One of the most straightforward applications is in surface rendering. Typically, explicit ray tracing techniques have been used to render a "realistic" projection of the visible part of the given surfaces on the image plane. Not surprisingly, some applications related to the accumulation on surfaces of quantities that propagate as does light also need visibility information. Examples include etching [1], the formation of huge ice spikes on the Peruvian Andes Mountains [8], and shape from shading models [53] (see Figure 5.3).

We point out here that in many of the applications listed above, the data (i.e. surfaces) are given implicitly. It is therefore, natural to work directly with the implicit data without converting to a different explicit representation. An interesting level set method for the visibility problem has recently been developed by the authors and collaborators [99]. The underlying basic algorithm can be regarded as a multi-level implicit ray tracer that works with volumetric data. It has been applied successfully in reconstructing surfaces from multiple images of different views [53]. It can also be applied directly to some surface renderers, e.g. the "non-photo-realistic" renderer of [51]. In the algorithm defined in [99], the boundaries of visible and invisible regions,

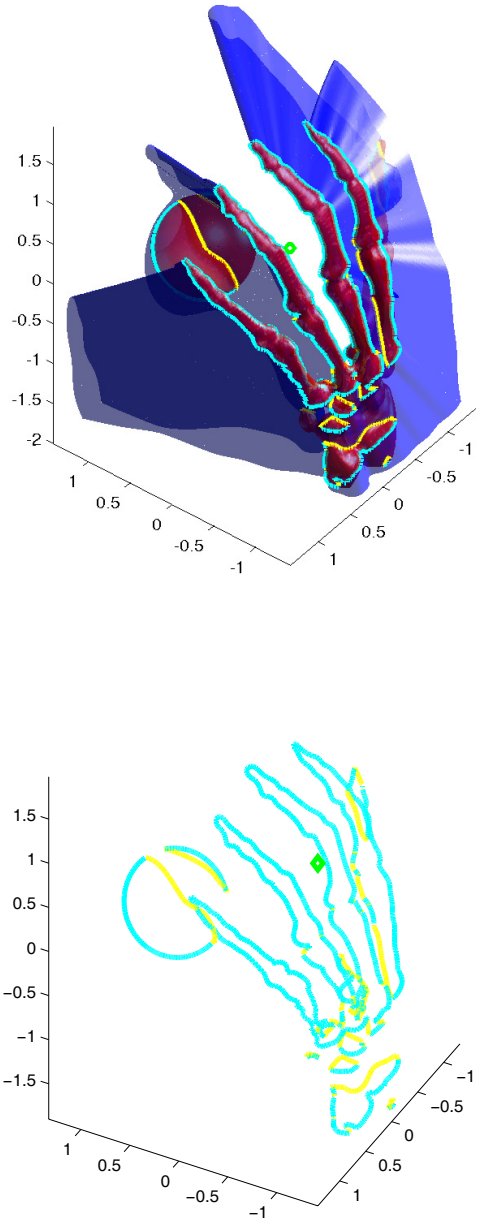

FIG. 5.5. *The blue surface borders the visible and invisible regions. The light blue and yellow curves indicate the silhouettes and swaths.*

both silhouette and swath [30], are implicitly represented in the framework of [11, 22], mentioned above. Figure 5.4 shows an accumulative visibility result of a path above the Grand Canyon. Figure 5.5 shows a result and the silhouette.

This implicit framework for visibility offers many other advantages. For example, the visibility information can be interpreted as the solution of simple first order PDEs

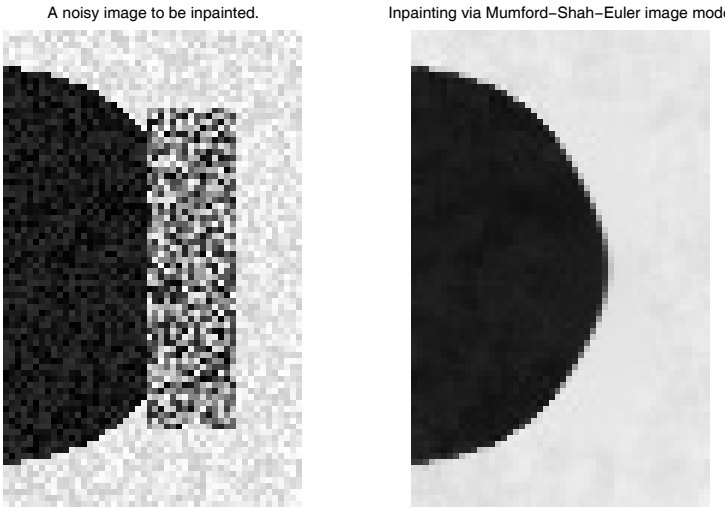

FIG. 5.6. *This is an inpainting result of [35].*

and [99] offers a near optimal solution method on the grid. The dynamics of the visibility with respect to moving vantage point or dynamic surfaces can be derived and tracked implicitly within the same framework. Furthermore, using the same framework and the well developed level set calculus and numerics, one can start solving variational problems on the visibility numerically and efficiently [97]. This will then relate to classical "guarding cameras" or "pursuer-evader" problems in computational geometry and robotics.

6. Current Trends

Currently, higher order nonlinear PDEs are increasingly appearing in image science. For example, in image inpainting of [15, 35, 60], a fourth order PDE is derived from regularizing the level set curvature of a given image. Figure 5.6 shows an inpainting result from [35].

In computer graphics, Tasdizen et al. [95] proposed to perform anisotropic diffusion on the normals of a given level set surface model. In general, fourth order equations are much harder to analyze, since they generally do not have a maximum principle as second order parabolic equations do. Another interesting paper of Burchard [10] discusses the diffusion operators constrained in color space.

There is also a trend of devising extremely fast (almost one-pass and discrete) algorithms for image science applications from level set PDE models. This began in [44, 88].

Multiscale methods have been successfully applied to several different applications in image science. For example, wavelet analysis has been quite successful in image compression. Multigrid methods provide fast algorithms to solve certain linear equations. Fast Multipole Methods also have been used in image science, see e.g. [12]. Recently, a general framework (HMM) for a class of multiscale problems has been advocated by W. E and B. Engquist [31]. An HMM-level set method for front propagation in random media is introduced by [23]. The authors believe that a combined multiscale-level set method will be useful in modeling image sequences that are

obtained from highly noisy media.

Acknowledgments. The authors thank Li-Tien Cheng, Frédéric Gibou, Jackie Shen, and Luminita Vese for providing their results for this paper. The work of the first author is partially supported by the National Science Foundation under agreement No. DMS-0111298. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The second author was supported by NSF grant DMS 0312222 and ONR MURI grant N00014-02-1-0728.

REFERENCES

[1] D. Adalsteinsson and J.A. Sethian, *An overview of level set methods for etching, deposition, and lithography development*, IEEE Transactions on Semiconductor Devices, 10(1), February, 1997.

[2] F. Alvarez, F. Guichard, J.M. Morel, and P.L. Lions, *Axioms and fundamental equations of image processing*, Arch. Rat. Mech. and Analys., 123:199–257, 1993.

[3] D.S. Balsara and C.W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, J. Comput. Phys., 160(2):405–452, 2000.

[4] M. Bardi and S. Osher, *The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations,* SIAM J. Math. Anal., 22(2):344–351, 1991.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in ACM SIGGRAPH, 417–424, 2000.

[6] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*, UCLA CAM Report, 02(47), 2002.

[7] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys., 174(2):759–780, 2001.

[8] M.D. Betterton, *Theory of structure formation in snowfields motivated by penitentes, suncups, and dirt cones*, Physical Review E, 63, 2001.

[9] M. Boué and P. Dupuis, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, SIAM J. Numer. Anal., 36(3):667–695, 1999.

[10] P. Burchard, *Total variation geometry I: Concepts and motivation*, UCLA CAM Report, 02(01), 2002.

[11] P. Burchard, L.T. Cheng, B. Merriman, and S. Osher, *Motion of curves in three spatial dimensions using a level set approach*, J. Comput. Phys., 170:720–741, 2001.

[12] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, B.C. McCallum, W.R. Fright, and T.R. Evans, *Reconstruction and representation of 3d objects with radial basis functions*, in SIGGRAPH, 2001.

[13] V. Caselles, J.M. Morel, and C. Sbert, *An axiomatic approach to image interpolation*, IEEE Trans. Image Processing, 7(3):376–386, 1998.

[14] A. Chambolle and P.L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76(2):167–188, 1997.

[15] T. Chan, S.H. Kang, and J.H. Shen, *Euler's elastica and curvature based inpainting*, SIAM J. Appl. Math., 63(2):564–592, 2002.

[16] T. Chan and H.M. Zhou, *Adaptive ENO-wavelet transforms for discontinuous functions*, UCLA CAM Report, 99(21), 1999.

[17] T.F. Chan and L.A. Vese, *A level set algorithm for minimizing the Mumford and Shah model functional in image processing*, Proceedings of the 1st IEEE Workshop on "Variational and Level Set Methods in Computer Vision", 161–168, 2001.

[18] T.F. Chan and L.A. Vese, *Active contours without edges*, IEEE Transactions on Image Processing, 10(2):266–277, 2001.

[19] T.F. Chan and H.M. Zhou, *ENO-wavelet transforms for piecewise smooth functions*, in SIAM J. Numerical Analysis, to appear.

[20] Y.G. Chen, Y. Giga, and Shun'ichi Goto, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, J. Differential Geom., 33(3):749–786, 1991.

[21] L.T. Cheng, *The level set method applied to geometrically based motion, materials science, and image processing*, PhD Thesis, Univeristy of California, Los Angeles, 2000.

[22] L.T. Cheng, P. Burchard, B. Merriman, and S. Osher, *Motion of curves constrained on surfaces using a level set approach*, J. Comput. Phys., 175:604–644, 2002.
[23] L.T. Cheng and W. E, *The heterogeneous multi-sclae method for interface dynamics*, J. Sci. Comput., 2002, to appear.
[24] B. Cockburn and C.W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws*, II. General framework. Math. Comp., 52(186):411–435, 1989.
[25] M.G. Crandall and P.L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Mathematics of Computation, 43:1–19, 1984.
[26] M.G. Crandall, H. Ishii, and P.L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. (N.S.), 27(1):1–67, 1992.
[27] P.E. Danielsson, *Euclidean distance mapping*, Computer Graphics and Image Processing, 14:227–248, 1980.
[28] A. Dervieux and F. Thomasset, *A finite element method for the simulation of rayleigh-taylor instability*, Lecture Notes in Mathematics, 771:145–158, 1979.
[29] A. Dervieux and F. Thomasset, *Multifluid incompressible flows by a finite element method*, Lecture Notes in Physics, 11:158–163, 1981.
[30] F. Duguet and G. Drettakis, *Robust epsilon visibility*, in John Hughes, editor, Proceedings of ACM SIGGRAPH 2002; Annual Conference Series. ACM Press / ACM SIGGRAPH, 2002.
[31] W. E and B. Engquist, *The heterogeneous multi-scale methods*, UCLA CAM Report, 02(15), 2002; Comm. Math. Sci., to appear.
[32] B. Engquist, A. Harten, and S. Osher, *A high order essentially nonoscillatory shock capturing method* Large scale scientific computing (Oberwolfach, 1985), 197–208, Birkhäuser Boston, Boston, MA, 1987.
[33] B. Engquist, A.K. Tornberg, and Y.H. Tsai, *Dirac-delta functions in level set methods*, in preparation.
[34] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183:83–116, 2002.
[35] S. Esedoglu and J.H. Shen, *Digital inpainting based on the Mumford-Shah-Euler image model*, UCLA CAM Report, 01(26), 2001.
[36] S. Esedoglu, B. Song, and R. Tsai, private communication.
[37] S. Esedoglu and Y.H. Tsai, *Decoupled phase-field methods for segmentation problems*, in preparation.
[38] L.C. Evans and J. Spruck, *Motion of level sets by mean curvature*, I. J. Differential Geom., 33(3):635–681, 1991.
[39] L.C. Evans and J. Spruck, *Motion of level sets by mean curvature*, II. Trans. Amer. Math. Soc., 330(1):321–332, 1992.
[40] L.C. Evans and J. Spruck, *Motion of level sets by mean curvature*, III. J. Geom. Anal., 2(2):121–150, 1992.
[41] L.C. Evans, *Partial differential equations*, American Mathematical Society, Providence, RI, 1998.
[42] L.C. Evans and J. Spruck, *Motion of level sets by mean curvature*, IV. J. Geom. Anal., 5(1):77–114, 1995.
[43] D. Eyre, *Unconditionally gradient stable time marching: the Cahn-Hilliard equation*, in Computational and mathematical models of microstructural evolution (San Francisco, CA, 1998), volume 529 of Mater. Res. Soc. Sympos. Proc., 39–46, MRS, Warrendale, PA, 1998.
[44] F. Gibou and R. Fedkiw, *A fast level set based algorithm for segmentation*, 2002; submitted to CVPR, Nov. 4, 2002, private communication.
[45] Y. Giga, *Surface evolution equations — a level set method*, Lipschitz Lecture Notes 44, Univ. of Bonn, 2002.
[46] S.K. Godunov, *A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics*, Mat. Sb. (N.S.), 47 (89):271–306, 1959.
[47] M.L. Green, *Statistics of images, the TV algorithm of Rudin-Osher-Fatemi for image denoisng and an improved denoising algorithm*, UCLA CAM Report, 02(55), 2002.
[48] E. Harabetian and S. Osher, *Regularization of ill-posed problems via the level set approach*, SIAM J. Appl. Math., 58(6):1689–1706 (electronic), 1998.
[49] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes*, III. J. Comput. Phys., 71(2):231–303, 1987.
[50] J. Helmsen, E. Puckett, P. Colella, and M. Dorr, *Two new methods for simulating photolithography development in 3d*, in SPIE 2726, 253–261, 1996.

[51] A. Hertzmann and D. Zorin, *Illustrating smooth surfaces*, in ACM SIGGRAPH, 517–526, 2000.
[52] G.S. Jiang and D.P. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 21(6):2126–2143 (electronic), 2000.
[53] H. Jin, A. Yezzi, Y.H. Tsai, L.T. Cheng, and S. Soatto, *Estimation of 3d surface shape and smooth radiance from 2d images: a level set approach*, Journal of Scientific Computing, to appear.
[54] S. Jin, H.L. Liu, S. Osher, and R. Tsai, *Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation*, 2003, www.levelset.com/download/density.pdf, submitted to Journal of Computational Physics.
[55] C.Y. Kao, S. Osher, and J.L. Qian, *Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations*, UCLA CAM Report, 03(38), 2003.
[56] C.Y. Kao, S. Osher, and Y.H. Tsai, *Fast sweeping methods for Hamilton-Jacobi equations*, UCLA CAM Report, 02(66), 2002.
[57] J.B. Keller, *Geometrical theory of diffraction*, J. Opt. Soc. Amer., 52:116–130, 1962.
[58] W.H. Liao, M. Bergsneider, L. Vese, S.C. Huang, and S. Osher, *From landmark matching to space and open curve matching: A level set approach*, UCLA CAM Report, 02(59), 2002.
[59] X.D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115(1):200–212, 1994.
[60] M. Lysaker, S. Osher, and X.C. Tai, *Noise removal using smoothed normals and surface fitting*, UCLA CAM Report, 03(03), 2003.
[61] A. Marquina and S. Osher, *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22(2):387–405 (electronic), 2000.
[62] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations*, x+122, American Mathematical Society, Providence, RI, 2001; The fifteenth Dean Jacqueline B. Lewis memorial lectures, Rutgers University.
[63] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42(5):577–685, 1989.
[64] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, segmentation and depth*, volume 662 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1993.
[65] S. Osher, *A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations*, SIAM J. Math. Anal., 24(5):1145–1152, 1993.
[66] S. Osher, L.T. Cheng, M. Kang, H. Shim, and Y.H. Tsai, *Geometric optics in a phase-space-based level set and Eulerian framework*, J. Comput. Phys., 179(2):622–648, 2002.
[67] S. Osher and S. Esedoglu, *Decomposition of images by the anisotropic rudin-osher-fatemi model*, UCLA CAM Report, 03(34), 2003.
[68] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.
[69] S. Osher and R.P. Fedkiw, *Level set methods: an overview and some recent results*, J. Comput. Phys., 169(2):463–502, 2001.
[70] S. Osher and J. Helmsen, *A generalized fast algorithm with applications to ion etching*, in preparation.
[71] S. Osher and B. Merriman, *The Wulff shape as the asymptotic limit of a growing crystalline interface*, Asian J. Math., 1(3):560–571, 1997.
[72] S. Osher and J.A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79(1):12–49, 1988.
[73] S. Osher and C.W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Numer. Anal., 28(4):907–922, 1991.
[74] S. Osher, A. Sole, and L. Vese, *Image decomposition and restoration using total variation minimization and the H^{-1} norm*, UCLA CAM Report, 02(57), 2002.
[75] N. Paragios and R. Deriche, *A PDE-based Level Set Approach for Detection and Tracking of Moving Objects*, Technical Report 3173, INRIA, France, May 1997.
[76] D.P. Peng, B. Merriman, S. Osher, H.K. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155(2):410–438, 1999.
[77] D.P. Peng, S. Osher, B. Merriman, and H.K. Zhao, *The geometry of Wulff crystal shapes and its relations with Riemann problems*, in Nonlinear partial differential equations (Evanston, IL, 1998), 251–303, Amer. Math. Soc., Providence, RI, 1999.
[78] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM J. Numer. Anal., 29(3):867–884, 1992.

[79] L. Rudin and S. Osher, *Total variation based restoration with free local constraints*, Proceedings ICIP, IEEE, Austin, TX.

[80] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60(1-4):259–68, 1992.

[81] J.A. Sethian and A. Vladimirsky, *Ordered upwind methods for static Hamilton-Jacobi equations*, Proc. Natl. Acad. Sci. USA, 98(20):11069–11074 (electronic), 2001.

[82] J.A. Sethian, *Fast marching level set methods for three dimensional photolithography development*, in SPIE 2726, 261–272, 1996.

[83] C.W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, ICASE Report 97-65, NASA, 1997.

[84] C.W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77(2):439–471, 1988.

[85] C.W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, II. J. Comput. Phys., 83(1):32–78, 1989.

[86] P. Smereka, *Spiral crystal growth*, Phys. D, 138(3-4):282–301, 2000.

[87] K.A. Smith, F.J. Solis, and D.L. Chopp, *A projection method for motion of triple junctions by levels sets*, Interfaces Free Bound., 4(3):263–276, 2002.

[88] B. Song and T. Chan, *Fast algorithm for level set based optimization*, UCLA CAM Report, 02(68), 2002.

[89] P.E. Souganidis, *Approximation schemes for viscosity solutions of Hamilton-Jacobi equations*, J. Differential Equations, 59(1):1–43, 1985.

[90] R.J. Spiteri and S.J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, preprint.

[91] J. Steinhoff, M. Fang, and L. Wang, *A new Eulerian method for the computation of propagating short acoustic and electromagnetic pulses*, J. Comput. Phys., 157:683–706, 2000.

[92] J. Strain, *Fast tree-based redistancing for level set computations*, J. Comput. Phys., 152(2):664–686, 1999.

[93] J. Strain, *Semi-Lagrangian methods for level set equations*, J. Comput. Phys., 151(2):498–533, 1999.

[94] M. Sussman, P. Smereka, and S. Osher, *A level set method for computing solutions to incompressible two-phase flow*, J. Comput. Phys., 114:146–159, 1994.

[95] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, *Geometric surface smoothing via anisotropic diffusion of normals*, UCLA CAM Report, 02(56), 2002.

[96] A.K. Tornberg and B. Enquist, *Regularization techniques for numerical approximation of PDEs with singularities*, UCLA CAM Report, 02(37), 2002; J. Sci. Comp., to appear.

[97] R. Tsai, L.T. Cheng, and S. Osher, *A level set framework for visibility related variational problems*, in preparation.

[98] Y.H.R. Tsai, *Rapid and accurate computation of the distance function using grids*, J. Comput. Phys., 178(1):175–195, 2002.

[99] Y.H.R. Tsai, L.T. Cheng, P. Burchard, S. Osher, and G. Sapiro, *Dynamic visibility in an implicit framework*, UCLA CAM Report, 02(06), 2002.

[100] Y.H.R. Tsai, L.T. Cheng, S. Osher, and H.K. Zhao, *Fast sweeping methods for a class of Hamilton-Jacobi equations*, SIAM J. Numer. Anal., 2003, to appear.

[101] Y.H.R. Tsai, Y. Giga, and S. Osher, *A level set approach for computing discontinuous solutions of Hamilton-Jacobi equations*, Math. Comp., 72(241):159–181 (electronic), 2003.

[102] J. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Transactions on Automatic Control, 40(9):1528–1538, 1995.

[103] L.A. Vese and T.F. Chan, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, International Journal of Computer Vision, 50(3), 2002.

[104] L.A. Vese and S.J. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, UCLA CAM Report, 02(20), 2002.

[105] H.K. Zhao, S. Osher, and R. Fedkiw, *Fast surface reconstruction using the level set method*, in "First IEEE Workshop in Variation and Level Set Methods in Computer Vision", 194–202, Vancouver, Canada, 2001. ICCV.

[106] H.K. Zhao, *Fast sweeping method for eikonal equations*, Math. Comp., 2003, to appear.

[107] H.K. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion*, Journal of Computational Physics, 127:179–195, 1996.

[108] H.K. Zhao, S. Osher, B. Merriman, and M. Kang, *Implicit and non-parametric shape reconstruction from unorganized points using a variational level set method*, Computer Vision and Image Understanding, 80:295–319, 2000.

[109] J.J. Xu and H.K. Zhao, *An Eulerian formulation for solving partial differential equations along a moving interface*, J. Sci. Comp., 2003, to appear.