Acta Numerica (2005), pp. 509–573
DOI: 10.1017/S0962492904000273

Total variation and level set methods in image science

Yen-Hsi Richard Tsai
Department of Mathematics,
University of Texas at Austin, TX 78712, USA
E-mail: ytsai@math.utexas.edu

Stanley Osher*
Department of Mathematics,
University of California, Los Angeles,
Los Angeles, CA 90095-1555, USA
E-mail: sjo@math.ucla.edu

We review level set methods and the related techniques that are common in many PDE-based image models. Many of these techniques involve minimizing the total variation of the solution and admit regularizations on the curvature of its level sets. We examine the scope of these techniques in image science, in particular in image segmentation, interpolation, and decomposition, and introduce some relevant level set techniques that are useful for this class of applications. Many of the standard problems are formulated as variational models. We observe increasing synergistic progression of new tools and ideas between the inverse problem community and the 'imagers'. We show that image science demands multi-disciplinary knowledge and flexible, but still robust methods. That is why the level set method and total variation methods have become thriving techniques in this field.

Our goal is to survey recently developed techniques in various fields of research that are relevant to diverse objectives in image science. 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.*, Alvarez, Guichard, Morel and Lions (1993). This feature places our slightly general definition of the level set method for image science in context. In Section 2 we describe the building blocks of a typical level set method in the continuum

* Research supported by ONR N00014-03-1-0071, NIH U54Rl021813, NSF DMS-0312223, NSF ACI-032197, NSF NYU Subcontract and ONR MURI Stanford Subcontract.

setting. Each important task that we need to do is formulated as the solution to certain PDEs. Then, in Section 3, we briefly describe the finite difference methods developed to construct approximate solutions to these PDEs. Some approaches to interpolation into small subdomains of an image are reviewed in Section 4. In Section 5 we describe the Chan–Vese segmentation algorithm and two new fast implementation methods. Finally, in Section 6, we describe some new techniques developed in the level set community.

CONTENTS

1. Level set methods and image science

The level set method for capturing moving fronts was introduced in Osher and Sethian (1988). (Two earlier conference papers containing some of the key ideas have recently come to light (Dervieux and Thomasset 1979, 1981).) Over the years, the method has proved 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 areas of science in which 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 specialities such as medical imaging and Hollywood-type special effects.

We 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 science will need all these features. We will illustrate this point with some classical applications.

The term 'image science' is used here to denote a wide range of problems related to digital images. It is generally referred to problems related to acquiring images (imaging), image processing, computer graphics, and

computer vision. The type of mathematical techniques involved include discrete math, linear algebra, statistics, approximation theory, partial differential equations, quasi-convexity analysis related to solving inverse problems, and even algebraic geometry. The role of a level set method often relates to PDE techniques involving one or more of the following features: (1) regarding an image as a function sampled on a given grid with the grid values corresponding to the pixel intensity in suitable colour space, (2) regularization of the solutions, (3) representing boundaries, and (4) numerical methods. It is not hard to seek an application of the level set method for image segmentation or to model obstacles in inverse problems, since boundaries and level contours are fundamental objects in image science.

In a later section, we will examine some essential fundamentals of the level set methodology. We refer the reader to the original paper, Osher and Sethian (1988), and a new book, Osher and Fedkiw (2002), for detailed exposition of the level set method. A set of presentation slides is also available from the first author's home page.[1]

An image is considered as a function $u : \Omega \mapsto X$, where Ω is typically a rectangular domain in $\mathbb{R}^2$ and X is some compact space that is determined by the imaging device; *e.g.*, $X = [0, 1]$ if u is a grey value image, and $X = S^1 \times [0, 1]$ if the chromaticity and intensity is used for a colour image. Unless otherwise noted, we will discuss grey level images here.

We write a typical PDE method as

$$\lambda Lu = Ru, \tag{1.1}$$

or

$$u_t + \lambda Lu = Ru, \tag{1.2}$$

where L is some operator applied to the given image, $\lambda \geq 0$ is a predetermined parameter, and R denotes the regularization operator. For example, in the TV deblurring of Rudin and Osher (1994),

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

where K is a compact integral operator, f is the given image, and the restored image is the limit $u(t)$ as $t \longrightarrow \infty$. When L is not invertible, as in the above deblurring model, or when a certain regularity is needed in the image u, a regularization term will be added. In the usual version of total variation methods, regularization usually appears in a form similar to

$$Ru = \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right). \tag{1.3}$$

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

Typically, equations (1.1) and (1.2) are derived either by directly writing down some PDEs whose solutions possess the desired properties, or by devising an energy functional $\mathcal{E}(u)$ and solving for a minimizer. For example, the shock filter of Rudin and Osher (1990) and the inpainting algorithm of Bertalmio, Sapiro, Caselles and Ballester (2000) fall into the first category. The variational approaches seem to be the mainstream for many important problems nowadays, partly owing to the existing mathematical tools, involving calculus of variations and Γ-convergence, available to study such kind of models. The Mumford–Shah multiscale segmentation model (Mumford and Shah 1989) and the total variation (TV) denoising model of Rudin–Osher–Fatemi (ROF) (Rudin, Osher and Fatemi 1992) are successful variational models. Both models have inspired much research activity in this field and will be discussed frequently in this article. The ROF model can be written as

$$\min_u \mathcal{E}_{TV}(u) = \frac{\lambda}{2}\int (f-u)^2 \,\mathrm{d}x + \int |\nabla u| \,\mathrm{d}x, \tag{1.4}$$

where f is the given noisy image. In this set up, the Euler–Lagrange equation for (1.4) defines Lu as $(u-f)$, and $R = \nabla \cdot \frac{\nabla u}{|\nabla u|}$, which is the curvature of the level curve at each point of the image u. We remark that in many other image applications, the unregularized energy functional is nonconvex, and its global minimizer corresponds to the trivial solution. Only a local minimizer is needed. However, in (1.4), we obtain a useful global minimizer.

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 Marquina and Osher (2000), for example, $Ru(x)$ actually denotes the mean curvature of the level set of u passing through x. The effects of (1.3) in noise removal can be explained as follows: the level curves in the neighbourhoods 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 curvature (noise) disappear much more rapidly than those with relatively lower curvatures (this helped motivate the approach taken in Marquina and Osher (2000)). If the $|\nabla u|$ term is dropped (as it usually is) the velocity is inversely proportional 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 and a wide class of second-order nonlinear equations. The

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

viscosity solution theory describes how evolution extends beyond singularities, including the pinching-off of level curves. Chambolle and Lions (1997) provide some analysis of the total variation denoising model. See Chen, Giga and Goto (1991), Crandall, Ishii and Lions (1992) and Evans and Spruck (1991, 1992*a*, 1992*b*, 1995) for more general viscosity theory applied to a wide class of second order equations.

Another interesting category of applications is data interpolation. In the problem of inpainting (see, *e.g.*, Bertalmio *et al.* (2000) and Figure 1.1), the challenge is to repair images which have regions of missing information. The algorithms are motivated in part by 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.*, Caselles, Morel and Sbert (1998)) 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 (Chan and Vese 2001*a*). 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 of this type. Examples also include Chan and Vese (2001*b*) and Paragios and Deriche (1997). In a different, but related, context, Zhao *et al.* use a level set function to interpolate unorganized data sets (Zhao, Osher, Merriman and Kang 2000, Zhao, Osher and Fedkiw 2001).

Many of the above methods rely on the variational level set calculus similar to that of Zhao, Chan, Merriman and Osher (1996) to formulate the energies whose minimizers are interpreted as the solution to the problems,

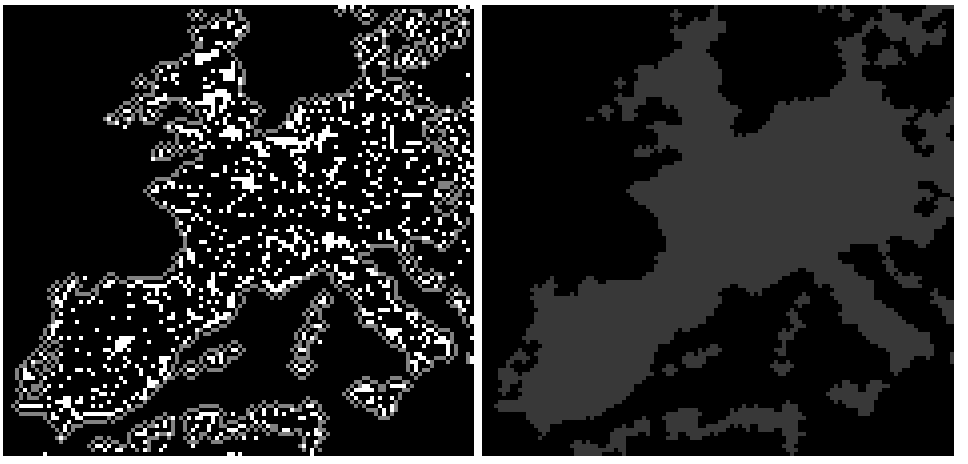

Figure 1.2. Land mass of Europe found using active contours.

and the solutions are level set functions. In general, the energies are variants of the surface integral

$$\int_\Omega F(\phi, u)\delta(\phi)|\nabla\phi|\,\mathrm{d}x,$$

and the volume integral

$$\int_\Omega G(\phi, u)H(-\phi)\,\mathrm{d}x.$$

See Zhao *et al.* (1996) 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 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 classical laws of physics. In fact, in the original level set paper (Osher and Sethian 1988), a level set function was 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 rendering of natural phenomena suitable for special effects in movie productions. The reader can find a detailed description and references in Osher and Fedkiw (2001). Figure 1.3 provides two such simulations.

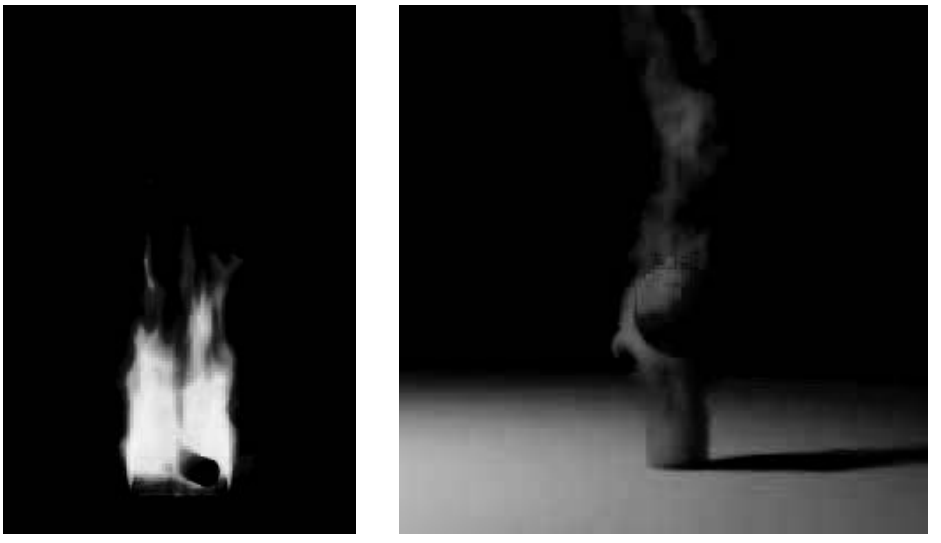

Figure 1.3. Image obtained from `www.cs.stanford.edu/~fedkiw`.

Finally, there is a collection of level set numerics, consisting mostly of approximations to general Hamilton–Jacobi equations and compressible and incompressible fluid dynamics. These methods are not limited only to pure level set formulations. They can also be used to solve other PDE-based image models. The basic numerics started in Osher and Sethian (1988) and Osher and Shu (1991), and generalizations have been carefully documented in Osher and Fedkiw (2002). Some new ones can be found in Enright, Fedkiw, Ferziger and Mitchell (2002), Kao, Osher and Tsai (2002), Sethian and Vladimirsky (2001), Tsai (2002), Tsai, Cheng, Osher and Zhao (2003*a*), Tsai, Giga and Osher (2003*b*) and Tsitsiklis (1995). Additionally, we mention Tornberg and Engquist (2003), which addresses the issue of regularization.

Ideas originating in this type of numerics, for instance, ENO interpolation (Harten, Engquist, Osher and Chakravarthy 1987), have been used to develop wavelet-based methods which minimize ringing, or Gibbs' phenomena at edges (Chan and Zhou 2002).

2. Brief review of the level set method

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 according to the laws of physics associated with the problem. They break up, merge, or disappear during the course of time evolution. These topo-

logical changes are problematic for most conventional methods. The level set method, however, handles these topological changes *'with no emotional involvement'* (Osher and Sethian 1988). Since its introduction, there has developed a powerful level set calculus used to solve a great variety of problems in fluid dynamics, materials sciences, computer vision, computer graphics, to name a few topics. We refer to Osher and Fedkiw (2002) for an extensive exposition of the level set calculus. See also Giga (2002) for a related theoretical exposition.

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

(1) Initialize/re-initialize ϕ at $t = t^n$.

(2) Construct/approximate $H(t, x, \phi, D\phi, D^2\phi)$. (Occasionally higher derivatives also appear for which rigorous viscosity solution theory definitely 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 convention and start our exposition for step (2). Steps (1) and (3) are 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 interfaces represented by a level set function are thus also referred to as curves. 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 $\{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\}$ define the embedded curve Γ. In many situations, we will also regard Γ as the boundary of the sublevel sets $\Sigma = \{\phi \leq 0\}$. See Figures 2.1 and 2.2 for some examples. If we associate a continuous velocity field v whose restriction onto the curve represents the velocity of the curve, then, at least locally in time, the evolution can be described by solving the Cauchy problem
$$\phi_t + v \cdot \nabla\phi = 0, \qquad \phi(x, 0) = \phi_0(x),$$

where ϕ_0 embeds the initial position of the curve. To derive this, let us look at a parametrized curve $\gamma(s,t)$ and assume that $\partial\gamma/\partial t$ is the known dynamics 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 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 the velocity field v.

In general, the velocity v can be a function of position x, t, and some other geometrical properties of the curve, or of other physical quantities

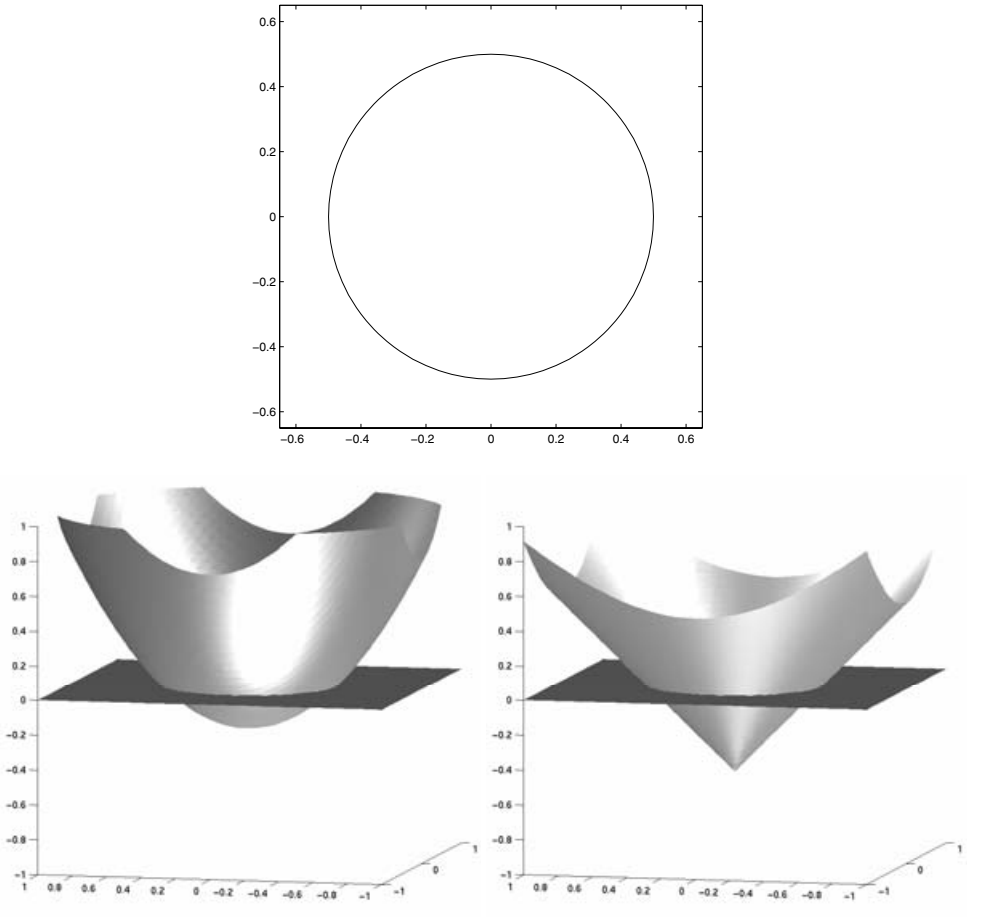

Figure 2.1. A circle embedded by different continuous functions.

that come with the problem. The equation can be written using the normal velocity:

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

We note that these equations are usually fully nonlinear first-order Hamilton–Jacobi or second-order degenerate parabolic equations, and in many cases, the theory of viscosity solutions (Crandall and Lions 1984) can be applied to guarantee well-posedness of the Cauchy problem.

It is instructive to derive the level set equation via a weak formulation using the area and co-area formula (Evans and Gariepy 1992). Let w be a test function, and let $v_n$be the normal velocity of $\Gamma = \partial\Sigma = \{\phi \leq 0\}$:

$$\begin{aligned}
\int_{\mathbb{R}^2} \frac{\partial\phi}{\partial t} w \,\mathrm{d}x &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^2} \phi w \,\mathrm{d}x = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}}\int_{\{\phi(\cdot,t)<\eta\}} w \,\mathrm{d}x\,\mathrm{d}\eta \\
&= -\int_{\mathbb{R}}\int_{\partial\{\phi(\cdot,t)<\eta\}} w v_n \,\mathrm{d}s\,\mathrm{d}\eta \\
&= -\int_{\mathbb{R}}\int w v_n \delta(\phi(x)-\eta)|\nabla\phi(x)|\,\mathrm{d}x\,\mathrm{d}\eta \\
&= -\int_{\mathbb{R}^2} v_n|\nabla\phi| w \,\mathrm{d}x.
\end{aligned}$$

We typically solve the level set equation on a rectangular domain Ω with Neumann boundary condition on $\partial\Omega$. In general, the level set equations do not admit classical solutions. However, under appropriate regularity conditions on v_n or H, it is possible to uniquely define a special weak solution called the viscosity solution (Crandall and Lions 1983, Crandall *et al.* 1992). For many equations, the viscosity solution corresponds to the uniform limit

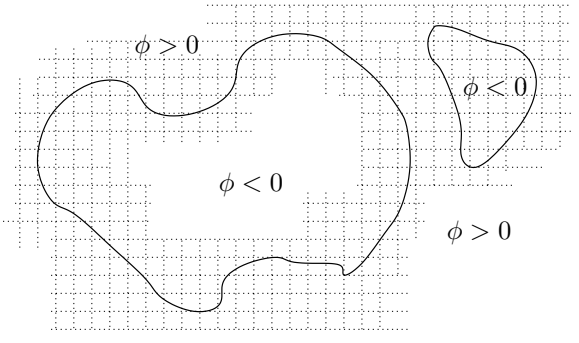

Figure 2.2. Two closed curves that are implicitly embedded by a single level set function defined on the grid.

of the vanishing viscosity solution. It can be shown that the motion of the zero level set of the viscosity solution is a generalization of a smooth motion in the normal direction, and the motion is uniquely defined if no *fattening* occurs; *i.e.*, if $\{\phi = 0\}$ remains a set of measure zero for all time. We refer the interested readers to Evans (1998) and Giga (2002) for more information on this aspect of the level set method. Corresponding to viscosity solution theory, there is a set of simple finite difference methods to construct approximation solutions (Barles and Souganidis 1991, Crandall and Lions 1984).

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

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

If $f \equiv 1$, this integral yields the arc length for curves in two dimensions, and surface area in three dimensions. Volume integrals are defined as

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

where $H(x) = 1$ for $x \geq 0$ and $H(x) = 0$ for $x < 0$. In Sections 3.8 and 3.9, we will review the related numerics proposed by Engquist, Tornberg and Tsai (2004) related to approximating the delta and Heaviside functions.

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 re-initialization (Sussman, Smereka and Osher 1994) by evolving the following PDE to steady state:

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

Here ϕ_0 denotes the level set function before re-intialization. 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 $\mathrm{sgn}(0) = 0$. See Figure 2.3. We will come back to issues related to how proper discretizations of the discontinuous signum function should be carried out in order to achieve efficiency and accuracy.

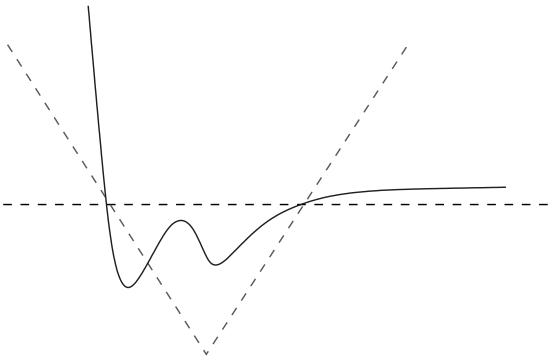

Figure 2.3. Re-initialization.

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.*, Peng, Merriman, Osher, Zhao and Kang (1999*a*), is to run distance re-initialization (2.3) 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 Section 3.7.

We remark that for most applications, the re-initialization is only needed in a neighbourhood around the zero level set, and the diameter of this neighbourhood depends on the discretization of the partial derivatives in the PDE. This implies that only a few time-steps in τ are needed. We also 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 numerical error. Finally, re-initialization 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 neighbourhood of Γ whose restriction on Γ agrees with the known interface velocity. One common way to obtain such a velocity field is to solve the following boundary value problem:

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

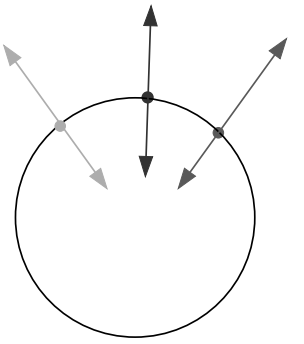

Figure 2.4. Quantities are extended off the zero level set in the normal directions.

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

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

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 2.4. Fast sweeping (Kao *et al.* 2002, Kao, Osher and Qian 2004, Tsai 2002, Tsai *et al.* 2003*a*, Zhao 2005) or fast marching (Tsitsiklis 1995, Sethian 1996) can be used to solve the first equation while a higher-order accurate Hamilton–Jacobi solver can be used for the second (Osher and Shu 1991). 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 change. 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 $\tilde{f}$ satisfies

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

where Q_Γ denotes the 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 neighbourhood 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 centre 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 by Jin, Liu, Osher and Tsai (2005*b*) for such an example. In the next paragraph, we discuss another example of this situation.

Assume that we are interested in quantities defined and parametrized on the surfaces, and we need to know how these quantities redistribute during the evolution of the surfaces. Harabetian and Osher (1998) introduced a method for doing this. 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 parametrized by s and t. We want to evolve, for example, $\Gamma(s,0)$ to time t, by the level set functions

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

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 and finite 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

$$J_t + \nabla \cdot (Jv) = 0$$

in addition to

$$\phi_t + v \cdot \nabla \phi = 0.$$

Finally, we briefly describe the systematic approach that began in Cheng (2000), and was developed in Bertalmio, Cheng, Osher and Sapiro (2001*b*) for solving PDEs on surfaces for image processing and more general applications. A similar approach was later adopted by Xu and Zhao (2003) 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,

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

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 on 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 to 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

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

To perform distance re-initialization on $\tilde{f}$, one can evolve

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

As an example of solving PDEs on surfaces, we consider total variation diminishing flow of an image u, defined on a surface Γ, takes the form

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

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

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

The function u is extended outside Γ as described in Section 2.3. For time-dependent problems, this extension is redone every few time iterations. The PDE needs to be solved only in a small neighbourhood of Γ, as described in Peng *et al.* (1999*a*).

2.5. Level set methods involving variational approaches

Assume that the energy functional $\mathcal{E}$ is an integral operator on u over $\Sigma \subset \Omega \subset \mathbb{R}^d$,

$$\mathcal{E}(u, \Sigma) = \int_\Sigma F(u(x))\, \mathrm{d}x,$$

and the non-positive region of ϕ defines Σ; *i.e.*, $\{\phi \leq 0\} = \Sigma$. The key idea of the variational level set method formulated in Zhao *et al.* (1996) is that the above integral can be written as

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

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. Variational calculus reveals that that the change in ϕ on this functional can be quantified through the boundary integral over $\partial\Sigma = \{\phi = 0\}$.

We follow the review of Burger and Osher (2005) and describe how sensitivity of this type of energies can be studied in the context of level set methods.

Level set method and shape calculus

Shape sensitivity analysis is a classical topic in shape optimization, and defines a natural calculus on shapes. For sufficiently regular shapes (*i.e.*, with C^1 boundary), there are two equivalent ways of introducing shape sensitivities, namely the *deformation method* and the *speed method* (Sokołowski and Zolésio 1992). Owing to its relation to the level set method, we shall use the latter as the basis of the following presentation.

Given a set $\Sigma(t)$ evolving in a velocity field V. Consider an energy $\mathcal{E}(\Sigma)$ that depends on the shape of Σ. The shape sensitivity of $\mathcal{E}$ in the direction of a perturbation V is then given by

$$\mathrm{d}\mathcal{E}(\Sigma; V) = \frac{\mathrm{d}}{\mathrm{d}t} \mathcal{E}(\Sigma(t))|_{t=0}.$$

$\mathrm{d}\mathcal{E}(\Sigma; \cdot)$ is called the shape differential. In the level set framework, $\Sigma(t)$

may be embedded as $\{\phi(\cdot,t)\leq 0\}$. Thus the shape sensitivity is

$$\mathrm{d}\mathcal{E}(\Sigma;V)=\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\{\phi(\cdot,t)\leq 0\})|_{t=0}.$$

Typically, the energies that appear in the image processing applications are either volume integrals,

$$\mathcal{E}(\Sigma)=\int_{\Sigma} g\,\mathrm{d}x,$$

or boundary integrals,

$$\mathcal{E}(\Sigma)=\int_{\partial\Sigma} g\,\mathrm{d}S.$$

For example, in the former case, direct calculation shows that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\Sigma(t))|_{t=0}=\int_{\mathbb{R}^d} gV\cdot\nabla\phi\delta(\phi)\,\mathrm{d}x=\int_{\partial\Sigma(0)} gV_n\,\mathrm{d}S.$$

The shape derivative is hence related back to the variational level set method (Zhao *et al.* 1996). Hence, in a variational level set model, one can choose many different V_n to decrease $\mathcal{E}$. Different choices of V_n may result in minimizing $\mathcal{E}$ in different normed spaces. If $\mathcal{E}$ is nonconvex, the choice of V_n and consequently the descent path might lead to different local minimizers.

In many applications involving shape optimization, *e.g.*, image segmentation, it is desirable to test the sensitivity of the energy function with respect to topological changes in a given shape. The topological derivative of a shape Σ with respect to a spherical perturbation at $x\in\Omega$ (Ω is the computational domain) is given by

$$D_\tau\mathcal{E}(\Sigma;x)=\lim_{R\to 0}\frac{\mathcal{E}(\Sigma\setminus B_R(x))-\mathcal{E}(\Sigma)}{|B_R(x)\bigcap\Omega|},$$

if the limit on the right-hand side exists. Here, $B_R(x)$ denotes the closed ball of radius R centred at x, while $d_\tau\mathcal{E}(\Sigma;x)$ measures the variation with respect to the nucleation of an infinitesimal hole at x. Thus, if $d_\tau\mathcal{E}(\Sigma;x)<0$, then the nucleation of a hole at x will decrease the objective energy functional. One can respectively define the topological derivative of the complement of Σ by

$$D_\tau\mathcal{E}(\Sigma;x)=\lim_{R\to 0}\frac{\mathcal{E}(\Sigma\bigcup B_R(x))-\mathcal{E}(\Sigma)}{|B_R(x)\bigcap\Sigma|}.$$

In this case, we are interested in the sensitivity of the energy function with respect to the introduction of a new connected component to the given shape Σ. One can see the link between the shape derivative and the topological derivative by evaluating $D_\tau\mathcal{E}(\Sigma;x)$ at $\partial\Sigma$.

Burger, Hackl and Ring (2004) successfully incorporate this idea above to solve a class of shape optimization problems. Their idea is to add $d_\tau\mathcal{E}(\Sigma;x)$ as a forcing term in the gradient descent.

Preserving topology

In some applications, one may be interested in preserving the topology of an given initial zero level set of ϕ, *e.g.*, in mapping of brain images, or the optimization of microstructured optical fibres. In particular, one usually needs to prevent disconnected components of $\{\phi < 0\}$ from merging with each other when they get close to each other. This is common behaviour for many level set simulations that compute the viscosity solution (Crandall *et al.* 1992).

An automatic way to incorporate this additional property was recently devised by Alexandrov and Santosa (2005). They proposed adding a penalty term to the original energy,

$$H(\Sigma) = -\int_{\partial\Sigma} \big(\log(d_\Sigma(x + \sigma\nabla d_\Sigma(x)) + \log(-d_\Sigma(x - \sigma\nabla d_\Sigma(x))\big)\, \mathrm{d}s$$

for some small constant $\sigma > 0$. Here, d_Σ denotes the signed distance function to $\partial\Sigma$ with $d_\Sigma(x) \leq 0$ for $x \in \Sigma$. Note that for $x \in \Sigma$, $x + \sigma\nabla d_\Sigma(x)$ is a point projected a distance σ outside Σ, while $x - \sigma\nabla d_\Sigma(x)$ is a point that is projected a distance σ inside Σ. Hence this penalty term forces a minimum distance of σ between connected components of $\{d_\Sigma < 0\}$, and respectively $\{d_\Sigma > 0\}$. Therefore, no topological change can arise. However, this penalty also indirectly regulates the curvatures of $\partial\Sigma$. One can conjure up a scenario in which the desired shape has many slender fingering components such that the thickness of each 'finger' is less than σ. This added penalty will unfortunately prevent structures of the type from being computed.

Another more general method developed to prevent merging can be found in Han, Xu and Prince (2003). This appears to be quite useful in brain mapping.

2.6. 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 reasonable quality without excessive complexity. We refer the readers to Smith, Solis and Chopp (2002), Vese and Chan (2002) and Zhao *et al.* (1996) for level set methods for multiple phases, Burchard, Cheng, Merriman and Osher (2001) and Osher, Cheng, Kang, Shim and Tsai (2002*a*) for higher codimensions, Smereka (2000) for open curves, and Peng *et al.* (1999*a*) and Strain (1999*a*, 1999*b*) for localization. We also refer to Enright *et al.* (2002) for a hybrid particle level set method that is designed to lessen the numerical diffusion effect for some class of problems, particularly two-phase incompressible flows.

3. Numerics

The numerical solution of conservation laws has been an active field of research for quite some time. 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 interpolation 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 (2002) and the extensive references therein. (For simplicity of exposition, we again restrict our discussion to two space dimensions.)

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

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

and 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 (Godunov 1959) developed for conservation laws begins by regarding 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 Bardi and Osher (1991) and Osher and Sethian (1988).

In high-order schemes, cell averages are replaced by more accurate non-oscillatory 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 neighbours. Two commonly used methods to get a kth-order approximation of f in I_i are spectral interpolation, *e.g.*, based on Fourier expansions, and fixed-order polynomial interpolation. Both approaches produce oscillations near the jumps in the function values or their derivatives. We will not comment on the Fourier-based methods since 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 possible points in the 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 a neighbourhood of these $2k-1$ 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 fifth-order WENO. For details, we refer to the original papers by Engquist, Harten and Osher (1987), Harten *et al.* (1987), Jiang and Peng (2000) and Liu, Osher and Chan (1994), and the review article by Shu (1997). Recently, Balsara and Shu (2000) developed even higher-order WENO reconstructions.

There are successful adaptations of this ENO idea/philosophy to other frameworks. See Chan and Zhou (1999, 2002) for ENO wavelet decompositions for image processing, and Cockburn and Shu (1989) 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 equations in questions. This is especially important for nonlinear hyperbolic 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.

Crandall and Lions (1983) introduced viscosity solution theory for a class of Hamilton–Jacobi equations requiring Lipschitz-continuous initial data and for which the Hamiltonian $H(x, u, p)$ is Lipschitz-continuous and *non-decreasing in* u. Later, in Crandall and Lions (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 (1985) extended the results to include variable coefficients. Osher and Sethian (1988) contributed to the numerics of Hamilton–Jacobi equations in their level set paper. This was later generalized and completed in the paper by Osher and Shu (1991), in which they provided a family of numerical Hamiltonians related to the ENO schemes for conservation laws. WENO schemes using the numerical Hamiltonians described in Osher and Shu (1991) were introduced in Jiang and Peng (2000). The method of lines using TVD Runge–Kutta time discretization is used (Shu and Osher 1988). 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 Osher and Shu (1991) (see also Osher and Sethian (1988)), and suppressing the dependence of H on x and y, we recommend using the Local Lax–Friedrichs (LLF) numerical Hamiltonian:

$$\hat{H}^{LLF}(p^+, p^-, q^+, q^-) = H\left(\frac{p^+ + p^-}{2}, \frac{q^+ + q^-}{2}\right) - \frac{1}{2}\alpha^x(p^+, p^-)(p^+ - p^-) - \frac{1}{2}\alpha^y(q^+, q^-)(q^+ - q^-), \tag{3.1}$$

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, and the intervals $[A, B]$ and $[C, D]$ are *a priori* bounds of ϕ_x and ϕ_y. This Hamiltonian is used together with ENO or WENO interpolation to obtain higher-order methods.

3.5. Curvature

In many applications, the mean curvature term

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

for the level set function ϕ or the image function u appears as a regularization. We will use u in our following discussion. This term is usually approximated by finite differencing centred 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 useful for numerical computations.) A popular choice would be $|\nabla u|_\epsilon = (|\nabla u|^2 + \epsilon^2)^{1/2}$, $0 < \epsilon \ll 1$. Under these settings,

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

$$n^x_{i\pm1/2,j} := \frac{D^{\pm}_x u_{i,j}}{\sqrt{(D^{\pm}_x u_{i,j})^2 + D^0_y (S^{\pm}_x u_{i,j})^2 + \epsilon^2}},$$

$$n^{\epsilon}_{i,j\pm1/2} := \frac{D^{\pm}_y u_{i,j}}{\sqrt{D^0_x (S^{\pm}_y u_{i,j})^2 + (D^{\pm}_y u_{i,j})^2 + \epsilon^2}},$$

where

$$S^{\pm}_x u_{i,j} = \frac{u_{i\pm1,j} + u_{i,j}}{2} \quad \text{and} \quad S^{\pm}_y u_{i,j} = \frac{u_{i,j\pm1} + u_{i,j}}{2}$$

are the averaging operators in the x and y direction. In practice, we choose ϵ to be the same scale as the mesh size.

It is important to point out that one can not prove convergence to the viscosity solution from this discretization using $\epsilon = a\Delta x + b\Delta y$, for two fixed nonnegative constants a and b. However, in practice, this approximation seems to work well. To be safe, we recommend taking $\epsilon = \mathcal{O}(\Delta x^p, \Delta y^p)$ for $0 < p < 1$. A general approximation theory for this type of degenerate elliptic or parabolic equations is outlined by Barles and Souganidis (1991). Following this theory, a numerical discretization needs to be monotone, consistent and stable in order to achieve convergence. Recently, Oberman proposed a convergent numerical discretization for the mean curvature term on two-dimensional Cartesian grids (Oberman 2004). In his work, an extra degree of freedom is introduced: the curvature term is not only discretized with Δx and Δy, but also with $\Delta\theta$, which is the angle between two adjacent vectors formed by the grid points in the stencil. The last term discretizes the angle of the normal of the level sets of u. Hence, the resulting scheme enlarges the stencil as one refines the grid, which makes it a bit impractical.

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:

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

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

end up having

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n} - \Delta t\, H^{LLF}(x_i, y_j, D_+^x \phi_{i,j}^n, D_-^x \phi_{i,j}^n, D_+^y \phi_{i,j}^n, D_-^y \phi_{i,j}^n). \tag{3.2}$$

Typically, we use the third-order TVD Runge–Kutta scheme of Shu and Osher (1988), or the fourth-order schemes of Spiteri and Ruuth (2005) 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 third-order TVD RK scheme below. We wish to advance $u_t = L(u)$ from $t_n$to t_{n+1}:

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

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.

Solving eikonal equations

In geometrical optics (Keller 1962), the eikonal equation

$$\sqrt{\phi_x^2 + \phi_y^2} = r(x, y) \tag{3.3}$$

is derived from the leading term in an asymptotic expansion

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

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

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

is the viscosity solution to equation (3.3) with the boundary condition

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

Rouy and Tourin (1992) proved the convergence to the viscosity solution of an iterative method solving equation (3.3) with the Godunov numerical Hamiltonian approximating $|\nabla\phi|$. They also noticed that the Godunov numerical Hamiltonian can be written in the following simple form for this eikonal equation:

$$H_G(p_-, p_+, q_-, q_+) = \sqrt{\max\{p_-^+, p_+^-\}^2 + \max\{q_-^+, q_+^-\}^2}, \tag{3.4}$$

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 (1993) provided a link to the time-dependent eikonal equation 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 Osher (1993)). As a consequence, one can try to solve the time-dependent equation by the level set formulation (Osher and Sethian 1988) with high-order approximations to the partial derivatives (Osher and Shu 1991, Jiang and Peng 2000). Crandall and Lions (1984) proved that the discrete solution obtained with a consistent, monotone Hamiltonian converges to the desired viscosity solution.

Tsitsiklis (1995) combined heap sort with a variant of the classical Dijkstra algorithm to solve the steady state equation of the more general problem

$$|\nabla\phi| = r(x).$$

This was later rederived in Sethian (1996) and Helmsen, Puckett, Colella and Dorr (1996). It has become known as the fast marching method, and has complexity $\mathcal{O}(N \log N)$, where N is the number of grid points. Osher and Helmsen (2005) 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.

The sweeping idea

Danielsson (1980) 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. Boué and Dupuis (1999) suggested a similar 'sweeping' approach to solve the steady state equation which, experimentally, results in a $\mathcal{O}(N)$ algorithm for the problem at hand. This 'sweeping' approach has recently been used in Tsai (2002) and Zhao *et al.* (2000) to compute the distance function to an arbitrary data set in computer vision. Zhao (2005) proved that the fast sweeping algorithm achieves 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 $[0, 1]$:

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

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

$$\sqrt{\max(\max(D_- u_i, 0)^2, \min(D_+ u_i, 0)^2)} = 1, \quad u_0 = u_n = 0 \tag{3.5}$$

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

$$u_i^{(1)} = \begin{cases} i/n, & \text{if } i < n, \\ 0, & \text{if } i = n. \end{cases}$$

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 (3.5) 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 neighbouring 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 numerical Hamiltonian is essential in the algorithm as described here, since it determines what neighbouring 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. For sweeping applied to very general class of Hamilton–Jacobi equations, we recommend the simple and versatile Lax–Friedrichs method, which we mention in Section 3.7 below. See Kao *et al.* (2004) for details.

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 Dynamic Surface Extension of Steinhoff, Fang and Wang (2000), we can define functions that map each point in $\mathbb{R}^3$ to the space of (local) representations of surfaces (previously referred to as surface elements). We can further define the distance between a point P and a surface element $\mathcal{S}$ by

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

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 *enforce the first arrival property* of the viscosity solution to 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 parametrized surface $\Upsilon : I_s \times I_t \mapsto \mathbb{R}^3$, our algorithm provides a good initial guess for Newton's iterations on the orthogonality identity:

$$F(s_*, t_*; x) = \begin{pmatrix} (x - \Upsilon(s_*, t_*)) \cdot \Upsilon_s(s_*, t_*) \\ (x - \Upsilon(s_*, t_*)) \cdot \Upsilon_t(s_*, t_*) \end{pmatrix} = 0,$$

where $\Upsilon(s_*, t_*)$ is the closest point on the surface to x. The initial guess in this case is simply the closest point of the neighbours of 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 neighbour P_l of E, compute $u_l^{\text{tmp}} = \text{dist}(E, W(P_l))$.
(4) If $\text{dist}(E, W(P_l)) < \min_k u(P_k)$, set $u_l^{\text{tmp}} = \infty$. This is to enforce the monotonicity of the solution.
(5) Set $u(E) = \min_l u_l^{\text{tmp}} = u_\lambda^{\text{tmp}}$ and $W(E) = W(P_\lambda)$.

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

In general, if only the level set function is available, one can construct a suitable interpolant of the level set function and try to compute the closest points. This was proposed in Chopp (2001), where a bicubic interpolation of the level set function is constructed and Newton's method is used to find the closest points on the zero level set of the interpolant.

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,

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

and the equations using Bellman's formulae for convex Hamiltonians, we refer readers to the recent papers by Tsai *et al.* (2003*a*) and Kao *et al.* (2002). Recently, a simple sweeping algorithm, based on the Lax–Friedrichs scheme (3.1), has been shown to work in great generality (Kao *et al.* 2004). Special conditions at the grid boundaries must be enforced in order for this central scheme to compute the correct solution. Accurate estimates of the bounds on the partial derivatives of the Hamiltonian increase the resolution and the efficiency of this algorithm. The main advantage of this algorithm is in the ease of implementation, especially for equations involving complicated and nonconvex Hamiltonians.

Higher-resolution sweeping methods have also been devised (Zhang, Zhao and Qian 2004). Essentially, the idea is to reconstruct the derivatives of the solution using the grid values that have been updated as a correction in the new approximation. The higher-order approximations of the derivatives require a larger stencil, leading to a larger numerical domain of dependence; together with the non-monotonicity in the reconstruction, more iterations are needed for convergence to the discretized nonlinear system. The complexity of these algorithms is still an open question. Nevertheless, it seem to be lower than that of a straightforward time marching to steady state.

3.8. Discretization of delta functions supported along the zero level set

In the level set formulation, the evaluation of a surface integral along the zero level set of ϕ requires singular integrals involving Dirac delta func-

tions. Careless quadratures for this type of integrals might lead to error that prevents convergence (Tornberg and Engquist 2003). Here we review the approaches proposed by Engquist *et al.* (2004).

Let s be a parametrization of Γ and let $\mathrm{d}s$ be the corresponding surface area measure. Define $\delta(\Gamma, x)$, $x \in \mathbb{R}^d$ as a delta function supported on Γ such that

$$\int_{\mathbb{R}^d} \delta(\Gamma, x)\, f(x)\, \mathrm{d}x = \int_{\mathbb{R}^d} f(x)\delta(\phi(x))|\nabla\phi(x)|\, \mathrm{d}x = \int_{\Gamma} f(X(s))\, \mathrm{d}s, \quad (3.6)$$

where $X(s) \in \Gamma$. The following techniques are based on replacing the distribution function δ by a class of continuous functions δ_ε in the approximation of integral defined in (3.6), and replacing the integral over the domain by a Riemann sum. δ_ε is chosen to be the linear hat function that has two discrete moments:

$$\delta_\varepsilon^L(x) = \begin{cases} \frac{1}{\epsilon}(1 - |\frac{x}{\epsilon}|), & 0 \le |x| \le \varepsilon, \\ 0, & |x| > \varepsilon. \end{cases} \quad (3.7)$$

Here, discrete moments of a function are defined in analogy to the usual notion of moments at continuous level; δ_ε is said to have q discrete moments if

$$h \sum_{j=-\infty}^{\infty} \delta_\varepsilon(x_j - \bar{x})(x_j - \bar{x})^r = \begin{cases} 1, & r = 0, \\ 0, & 1 \le r < q, \end{cases} \quad (3.8)$$

for any $\bar{x} \in \mathbb{R}$, and grid points $\{x_j\}$. It is shown in Tornberg (2002) and Tornberg and Engquist (2003) that the overall approximation is of first order in h if $\varepsilon = \sqrt{h}$. For a very narrow support, such as $\epsilon = C_0 h$, the δ_ε function is not sufficiently resolved and the error must instead be analysed directly by taking into account discrete effects of the computational grid.

Engquist *et al.* (2004) proposed two regularized delta functions built from the linear hat function (3.7). One is the product formula, following Peskin (2002), that requires explicit parametrization of Γ:

$$\delta_\varepsilon(\Gamma, x) = \int_{\Gamma} \prod_{k=1}^{d} \delta_{\varepsilon_k}(x^{(k)} - X^{(k)}(s))\, \mathrm{d}s. \quad (3.9)$$

Here δ_{ε_k} corresponds to the one-dimensional regularized δ function, and $X(s) = (X^{(1)}(s), \ldots, X^{(d)}(s))$ is a point on Γ. The other method is the level set formulation

$$\delta_\varepsilon(\Gamma, x) = \delta_\varepsilon(\phi(x))|\nabla\phi(x)|.$$

Both approaches use a pointwise variable regularization parameter dependent on the gradient of the level set function; *i.e.*, $\epsilon = \epsilon(x, \phi_x, \phi_x)$. The authors showed that with these approaches and with δ_ε^L as the building block, it is possible to approximate the singular integrals (3.6) on a uniform

Cartesian grid with at least first-order accuracy in h, while keeping minimum support (with $|\epsilon(x, \nabla\phi)| \leq Ch$). The first approach seems to yield approximations that are second-order accurate if ϕ is the distance function to Γ. We refer readers to Engquist *et al.* (2004) for the explicit formula derived from their first approach. We describe their second approach here owing to its simplicity. In short, $\delta(\phi(x))$ is approximated pointwise by

$$\delta^L_{\varepsilon(x,\nabla\phi)}(\phi(x)),$$

where

$$\epsilon(x, \nabla\phi) = h|\nabla\phi(x)|_{\ell_1}, \tag{3.10}$$

and $|\nabla\phi(x)|_{\ell_1} = \sum_{j=1}^{d} |\phi_{x_j}|$.

3.9. Regularization of characteristic functions

In the level set method, the average of a function g over the set $\{\phi \geq 0\}$ translates to an integral involving the Heaviside function:

$$\int_\Omega g(x)H(\phi(x))\,\mathrm{d}x.$$

Following the discussion in the previous subsection, one can regularize the Heaviside function by

$$H_\epsilon(x) = \begin{cases} 1, & x \geq \epsilon, \\ \frac{1}{2}(1 + \frac{x}{\epsilon}), & |x| < \epsilon, \\ 0, & x \leq -\epsilon, \end{cases} \tag{3.11}$$

with the same type of pointwise scaling:

$$\epsilon(x, \nabla\phi) = \frac{h}{2}|\nabla\phi(x)|_{\ell_1}.$$

It can be shown that the resulting approximation to the volume integral is second-order accurate in the mesh size h.

The signum function used in equation (2.2) is discontinuous:

$$\mathrm{sgn}(z) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0, \end{cases}$$

and may introduce grid effects when discretized improperly on the grid. Ideally, a smooth monotone function that passes through zero should replace the signum function, since we only care about the direction of the characteristics and the steady state of the solution in a neighbourhood of its zero level set. With a bounded smooth function such as

$$\tanh(\gamma_0 x), \quad \gamma_0 > 0$$

the accuracy of the solution to (2.2), for smooth zero level set, is then determined by the order of the discretization. However, the characteristics emanate from the zero level set at a speed that is 0 at the interface and smoothly increases as the bicharacteristics are getting farther away. On a grid with N grid points, this usually implies that the number of time-steps needed for steady state on this grid is proportional to N. In many applications, one is only interested in a thin band of width C/N around the zero level set. Therefore, if other operations involve $\mathcal{O}(N)$ operations, this regularization might be an attractive option. (See Section 2.2.)

4. Image interpolation

Consider an old photo with scratches. One can try to restore the original photo by filling in the scratched regions with certain values so that the overall image looks 'right'. This is a complicated interpolation problem. The main difficulties are as follows.

(1) The interpolation domain may be non-simply connected and have irregular boundaries.
(2) The interpolation procedure must allow discontinuities along some meaningful geometrical structures.
(3) Ultimately, the interpolation result is subject to human psycho-visual inspection.

Classically, (2) and (3) relate to the discussion of the function space to which images belong and which norm should be used. A severe problem would be that the interpolation domain is too large so that one essentially has to 'generate' new information.

In this article, we will call the problem of interpolating over 'narrow' domains the inpainting problem, and the other the 'disocclusion' problem. Essentially, interpolation algorithms rely on the regularity of certain *suitable quantities*. Considering grey-scale images, a natural quantity of consideration would be the level lines of the given image functions. One would think of properly connecting the level lines from a neighbourhood of the inpainting domain into it. Pioneering works of importance to this area are Caselles *et al.* (1998) and Masnou and Morel (1998).

Bertalmio *et al.* (2000) proposed an algorithm designed to project the gradient of the smoothness of the image intensity in the direction of the image level lines. The resulting model is a third-order PDE,

$$u_t = \nabla^\perp u \cdot \nabla(\Delta u),$$

where u is the image intensity, and $\nabla^\perp$ denotes the differential operator $(-\partial_y, \partial_x)$ and Δ is the standard Laplacian operator. At steady state,

$$\nabla^\perp u \cdot \nabla(\Delta u) = 0$$

inside the inpainting domain, implying that the gradient of Δu has to be perpendicular to the level line of u. In other words, the image value u is convected along the level curves of the quantity Δu. Later, Bertalmio, Bertozzi and Sapiro (2001*a*) established the connection of the image intensity u in this model to the stream function in a 2D incompressible fluid, where Δu can be interpreted as the vorticity of the fluid.

Chan and Shen proposed a variational model for inpainting:

$$J_\lambda[u] = \int_{E \bigcup D} |\nabla u| \, \mathrm{d}x + \lambda \int_E |f - u|^2 \, \mathrm{d}x. \tag{4.1}$$

Here, E is the region which is not to be interpolated, and D is the region with missing data. Imposing Neumann boundary condition at the boundary of $E \bigcup D$, the gradient equation is

$$u_t = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \chi_E(x) \lambda (u - f), \quad \text{for} \quad x \in E \bigcup D,$$

where χ_E is the characteristic function of E. One immediately sees the clear connection to the TV denoising model 1.4. This algorithm interpolates a given image so that the total variation in the inpainting domain is minimized. A mental application of the co-area formula reveals that the level lines stemming from E are connected in D with minimal arc lengths. This algorithm performs denoising and inpainting simultaneously.

The models that we have just described use the regularity of some local geometrical quantities for interpolation over the inpainting regions. In reality, the human vision may use more global quantities of a given image for judging whether any particular inpainting algorithm generates suitable solutions. One good test is to see how an inpainting algorithm connects the missing boundaries of a given set of shapes; whether a straight horizontal bar will be reconnected from the image with its middle part removed; or where a curved boundary can be restored. Therefore, many current efforts in devising new inpainting algorithms or in comparing different algorithms concentrate on this aspect. Of course, it is also possible to propose an inpainting model that is based on the regularity of statistical properties of a given image or images, especially when inpainting textures.

There have been efforts to incorporate more global quantities for inpainting. For example, Chan, Kang and Shen (2002) and Esedoglu and Shen (2002) replaced the total variation term in (4.1) by Euler's elastica:

$$e(\Gamma) = \int_\Gamma (\alpha + \beta \kappa^2) \, \mathrm{d}s = \alpha \, \mathrm{length}(\Gamma) + \beta \int_\Gamma \kappa^2 \, \mathrm{d}s.$$

Figure 4.1 shows an inpainting result from Esedoglu and Shen (2002).

In Bertalmio, Vese, Sapiro and Osher (2003), texture is first separated from a given image (described in Section 6.1), leaving a 'cartoon'-like

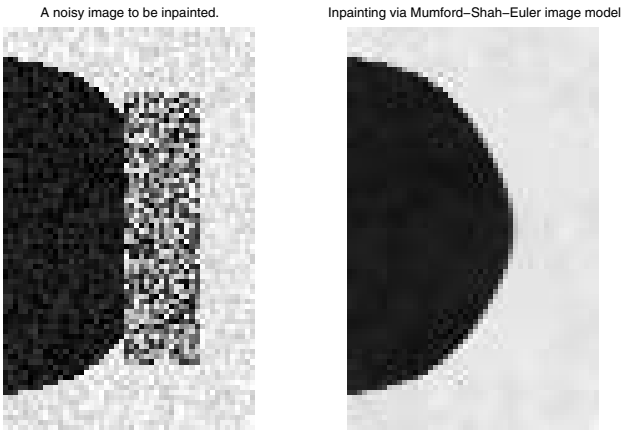

Figure 4.1. This is an inpainting result from Esedoglu and Shen (2002).

component of the original image. A statistical approach is adopted to 'synthesize' texture for the inpainting domain so that some statistical regularity across the whole image is maintained. After the separation of texture, an inpainting procedure based on local geometric regularity can be performed on the remaining cartoon-like image. Finally, the inpainting is done by putting the synthesized texture together with the cartoon inpainting result. See also Ballester, Bertalmio, Caselles, Sapiro and Verdera (2001) and Ballester, Caselles and Verdera (2003) for related work in inpainting and disocclusion. It is also possible to 'inpaint' in the time-frequency domain with a regularity constraint on the spatial domain. See, *e.g.*, Chan, Shen and Zhou (2004).

Finally, the level set method has also been used for interpolation of unorganized points, curves and/or surface patches by Zhao *et al.* (2000, 2001). Briefly, one finds a level set function whose zero level set passes through a given unorganized set S. The unsigned distance function d_S to the data set is used for fast visualization and analysis. This distance function can be efficiently constructed using the generalized closest point algorithm described in Section 3.7. Then a minimal surface/convection-type model, resembling geodesic snakes, is used for shape reconstruction from the data set. More precisely, Zhao *et al.* (2000, 2001) construct a local minimizer of the following energy:

$$\mathcal{E}(\Gamma) = \left(\int_\Gamma d_S^p(x)\, \mathrm{d}s \right)^{1/p},$$

using gradient descent with an initial guess constructed from $\{d_S = \epsilon_0\}$ by a fast tagging algorithm (Zhao *et al.* 2001). The positive constant ϵ_0 is determined from the sampling density of S. No *a priori* knowledge is assumed about the topology of the shape to be reconstructed. See Zhao and Osher (2003) for a recent review article.

5. 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 tumour 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 weapons. There are many other obvious applications. Mathematically, given an image $u : \Omega \subset \mathbb{R}^2(\text{or } \mathbb{R}^3) \mapsto \mathbb{R}^+$, we want to find closed sets Ω_i satisfying

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

such that $\mathcal{F}(u, \Omega_i) = 0$, where $\mathcal{F}$ is some functional that defines the segmentation goals and $\Omega_i^{(0)}$ denotes the interior of Ω_i. As in the example of finding tumours, typically, N is taken to be 2 (sometimes $N = 3$ when volumetric data is given), and Ω_1 is taken to be the region corresponding to the tumour, while Ω_2 contains everything else. It is then natural to devise a level set method to perform this task, by representing, for example, Ω_1 as the region in which ϕ is nonnegative. 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 in segmenting stealth fighter jets from both conventional radar signals and also infrared images.

Very often, the definition of what belongs to the 'desired' regions depends on the grey-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.

5.1. The Chan–Vese algorithm

This is closely related to the classical Mumford–Shah algorithm (Mumford and Shah 1989), but uses a simple level set framework for its implementation. We present the original Chan–Vese segmentation algorithm (Chan and Vese 2001*a*), and discuss various aspects of this algorithm.

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

$$\mathcal{E}(\phi,c_1,c_2;f)=\mu\int_\Omega \delta(\phi)|\nabla\phi|\,\mathrm{d}x \tag{5.1}$$
$$+\lambda_1\int_\Omega |f-c_1|^2H(\phi)\,\mathrm{d}x+\lambda_2\int_\Omega |f-c_2|^2(1-H(\phi))\,\mathrm{d}x.$$

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 the L^2-norm. The first term in the energy measures the arc length of the segment boundaries. Thus, minimizing this quantity provides some stability to 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

$$\partial_\phi\mathcal{E}=-\delta_\epsilon(\phi)\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}-\lambda_1(f-c_1)^2+\lambda_2(f-c_2)^2\right]=0, \tag{5.2}$$

with natural boundary condition

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

$$c_1(\phi)=\frac{\int_\Omega f(x)H_\epsilon(\phi(x))\,\mathrm{d}x}{\int_\Omega H_\epsilon(\phi(x))\,\mathrm{d}x}, \tag{5.3}$$

and

$$c_2(\phi)=\frac{\int_\Omega f(x)(1-H_\epsilon(\phi(x)))\,\mathrm{d}x}{\int_\Omega(1-H_\epsilon(\phi(x)))\,\mathrm{d}x}. \tag{5.4}$$

Discretization
A common approach to solving the minimization problem is to perform gradient descent on the regularized Euler–Lagrange equation (5.2); that is, solving the following time-dependent equation to steady state:

$$\frac{\partial\phi}{\partial t}=-\partial_\phi\mathcal{E}$$
$$=\delta_\epsilon(\phi)\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}-\lambda_1(f-c_1)^2+\lambda_2(f-c_2)^2\right]. \tag{5.5}$$

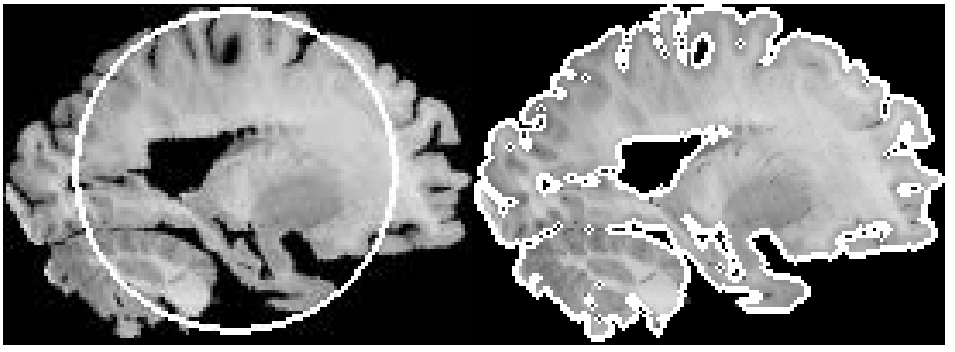

Figure 5.1. Brain segmentation in 2D.

Here, we remind the readers that $c_1(\phi)$ and $c_2(\phi)$ are defined in (5.3) and (5.4).

In the Chan–Vese algorithm, the authors regularized the Heaviside function used in (5.3) and (5.4),

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

and defined the delta function as its derivative:

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

Equation (5.5) 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 (5.5) is discretized as described in the previous section using the value of $\phi^n_{i\pm1,j\pm1}$, 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 a 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 written as

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

where $\alpha_\kappa \geq 0$ comes from the discretization of the curvature term. If the scheme were fully explicit, $\alpha_\kappa = 0$ and G would depend on $\phi^n_{i,j}$. In the original 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 5.1 and 5.2 for some results of this algorithm applied to brain segmentation.

Finally, it is also possible but usually not advisable in this (unusual) case because new zero level sets are likely to develop spontaneously, to replace the δ function in front of the curvature term by $|\nabla\phi|$ (Marquina and

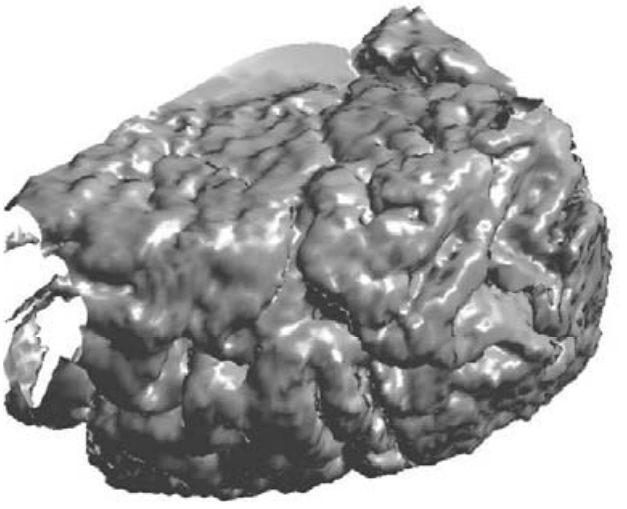

Figure 5.2. Brain segmentation in 3D.

Osher 2000). The equation then becomes independent of the choice of the level set function used, *i.e.*, the problem becomes morphological (Alvarez *et al.* 1993).

5.2. Fast segmentation algorithms

Recently, Gibou and Fedkiw (2005), and later Song and Chan (2002), 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 node, 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

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

here V corresponds to the fitting term in the Euler–Lagrange equation

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

(Note that the case $V = 0$ is implicitly defined.) In this algorithm, step (3) is essential for regularizing the segment boundaries. Without it, fractal-like boundaries may develop.

In Song and Chan's algorithm, the key observation is that only the signs of the level set function matter in the energy functional. This can easily be seen from the model defined in equation (4.1), in which 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:

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

Hence, the sign of $\phi^n(x)$ is flipped *only if the energy* (5.1) *is non-increasing.* This provides stability of the algorithm at the cost of some speed of implementation.

We remark that there is a close connection between these two 'level set' methods to the 'Γ-convergence' methods of Ambrosio and Tortorelli (1990). The Chan–Vese segmentation method can be approximated by the following variational problem:

$$E_\epsilon(u, c_1, c_2; f) := \mu \int \epsilon |\nabla u|^2 + \frac{1}{\epsilon} W(u)\, \mathrm{d}x \qquad (5.6)$$
$$+ \lambda_1 \int u^2 (f - c_1)^2 + \lambda_2 \int (1 - u)^2 (f - c_2)^2\, \mathrm{d}x,$$

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 term u^2 and $(1 - u)^2$ correspond respectively to $H(\phi)$ and $1 - H(\phi)$ in (5.1), and $\epsilon |\nabla u|^2$ corresponds to the regularization of of the length of $\partial\Omega_i$. Intuitively, 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.

5.3. Segmentation of multiple 'phases'

There are successful efforts to generalize the level set methods for multiphase computation. For example, in Zhao *et al.* (1996), 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 is expensive when the number of phases is large.

Vese and Chan (2002) use the sign of the level set functions ϕ_j as a binary coding for the phases, each assigned a nonnegative 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,

$$\Omega_0 = \{\phi_0 \geq 0\} \bigcap \{\phi_1 \geq 0\}, \qquad \Omega_1 = \{\phi_0 \leq 0\} \bigcap \{\phi_1 \geq 0\},$$
$$\Omega_2 = \{\phi_0 \geq 0\} \bigcap \{\phi_1 \leq 0\}, \qquad \Omega_3 = \{\phi_0 \leq 0\} \bigcap \{\phi_1 \leq 0\}.$$

In 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

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

It appears that the gradient descent algorithm using this formulation is quite sensitive to the initial configurations and tends to get stuck in some undesirable local minima. There is also the potential misidentification of what is supposed to be categorized as one single phase to two or more 'different' phases, since the 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. 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.

5.4. Discussion

One of the successful features reported in Chan and Vese (2001*a*) is the emergence of new interior contours. 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 Figure 5.3.

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, with the possibility of the emergence of a 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

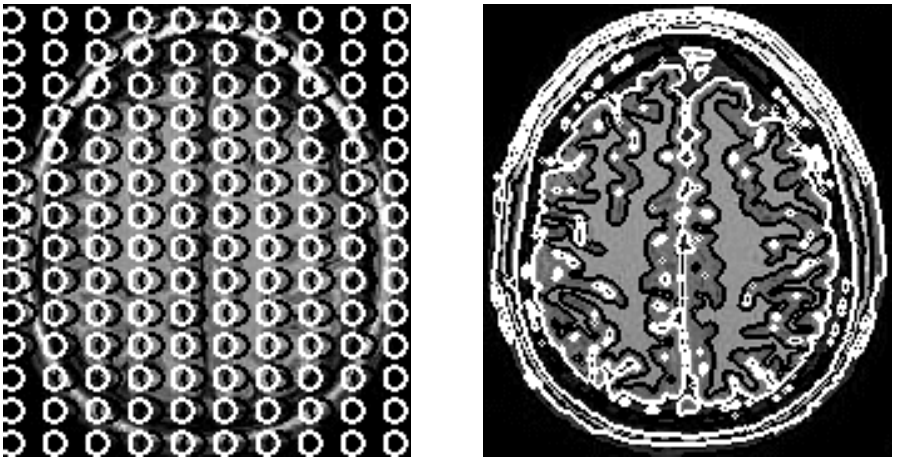

Figure 5.3. Initialization.

Euler–Lagrange equation (5.2). 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. We remark that it is possible to speed up the gradient flow in the Chan–Vese algorithm by a splitting method described in Eyre (1998).

There are many new (and old) 'level set' 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 Gibou and Fedkiw (2005) and Song and Chan (2002), these types of methods typically appear to be faster, and in some cases more flexible in handling multiple phases. These trends seem to be going against the original spirit and *raison d'être* 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 of 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). See Merriman, Bence and Osher (1994) for a precise analysis of this.

One should ask 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

and Shiota (1993) and also in the applications related to Euler's elastica. In these types of applications, the 'conventional' level set approach certainly has an 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 segmentation methods. So far, popular choices have been those variants arising 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, owing to their origin, retain special orientations and are anisotropic. This application appears, 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.*, $\mathcal{R}(n) = \operatorname{div}(\gamma(n))$. (In the common TV regularization, $\gamma(n) = n$.) We refer to Soravia (1994), Osher and Merriman (1997), Peng, Osher, Merriman and Zhao (1999*b*) and Esedoglu and Osher (2005) for more details.

6. Pushing the limit

In this section, we describe recent work corresponding to the classical applications we listed above.

6.1. Image decomposition

Many important tasks in image science involve the decomposition of given images into different components. Again, we start with total variation denoising model (1.4)

$$\min_u \mathcal{E}(u) = \lambda \int (u - f)^2 \, \mathrm{d}x + \int |\nabla u| \, \mathrm{d}x.$$

One can re-interpret this model as finding a decomposition of the given image f into a sum of two functions: $f = u + v$, with u corresponding to the 'clean' image that one wishes to reconstruct from f, and v contains the unwanted noise that is separated from f. The segmentation model of Mumford and Shah essentially proposes a similar decomposition, with the additional constraint on a lower-dimensional set that is interpreted as the edge of the resulting segmentation. If one considers the special setting in which images take on only two values c_1 and c_2, and the boundaries between the two constant regions are rectifiable, then the total variation of u corresponds to the length of the boundaries weighted by the jump $|c_1 - c_2|$. In this context, the link between the two models is especially clear. This connection was pointed out by Vese and Osher (2002) and was described in Osher (2003).

In his inspiring book of 2001, Meyer examined the total variation model of Rudin *et al.* (1992) more closely and proposed a decomposition in which the noise and texture part, v, is written as the divergence of a vector field; *i.e.*, $v = \mathrm{div} g$ with the norm $\|v\|_*$ defined as the infimum of L^∞ norms of such vectors g. The proposed decomposition finds u as the solution to the following minimization problem:

$$\min_u \lambda \|f - u\|_* + \int |\nabla u| \, \mathrm{d}x.$$

The motivation of Meyer is that the L_2 norm used in the first integral in (1.4) to measure the noise and texture part of f can be improved by using the dual norm of $\|\cdot\|_{BV}$ (with proper completion of the space BV). This book triggered a sequence of interesting studies and useful algorithms.

Vese and Osher (2003) approximated Meyer's $\|\cdot\|_*$ norm by an L^p norm, and proposed a modified variational model:

$$\min_{u,g_1,g_2} \mathcal{E}(u, g_1, g_2) = \lambda \int \left(f - \left(u + \frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \right) \right)^2 \mathrm{d}x + \mu \left(\int (g_1^2 + g_2^2)^{\frac{p}{2}} \right)^{\frac{1}{p}} + \int |\nabla u| \, \mathrm{d}x.$$

Osher, Sole and Vese (2003) assumed the Hodge decomposition of the vector field g: $g = \nabla P + Q$, where Q is divergence-free. With this assumption, and the H^{-1} norm square in place of $\|\cdot\|_*$, they proposed the model

$$\min_{u,g_1,g_2} \mathcal{E}(u, g_1, g_2) = \lambda \int |\nabla \Delta^{-1}(u_0 - u)|^2 \, \mathrm{d}x + \int |\nabla u| \, \mathrm{d}x.$$

Later, the first decomposition was combined with other texture synthesis technique to inpaint textured images (Bertalmio *et al.* 2003).

See Haddad and Meyer (2004) for a recent review of the related variational models.

6.2. Inverse scale space and PDE-based multi-resolution image analysis

It is possible to construct a hierarchical decomposition of a given image using the 'length scale' parameter λ in the TV denoising model. Tadmor, Nezzar and Vese (2004) study the convergence properties of this type of decomposition using $\lambda = \lambda_0 2^j$. More precisely, the decomposition starts with $f = u_0 + v_0$, where u_0 is the minimizer of the standard TV denoising model

$$\min_{u \in L^2(\Omega)} \mathcal{E}_{TV}(f, u, \lambda_0) = \int_\Omega |\nabla u| \, \mathrm{d}x + \lambda_0 \int_\Omega |f - u|^2 \, \mathrm{d}x$$

and then iteratively performs the same decomposition for the residual v_j:

$$u_{j+1} = \mathrm{arginf}_{u \in L^2(\Omega)} \mathcal{E}_{TV}(v_j, u, \lambda_0 2^j), \quad \text{and} \quad v_{j+1} = v_j - u_{j+1}.$$

This procedure thus leads to a nonlinear hierarchical decomposition

$$f = \sum_{j=0}^{k} u_j + v_k.$$

The same strategy was first proposed in Scherzer and Groetsch (2001).

Instead of using the L^2 norm in the fitting term, Esedoglu and Chan (Esedoglu and Chan 2004) reported interesting results for the model

$$\mathcal{E}_{EC} = \min_{u \in L^1(\Omega)} \int_\Omega |\nabla u| \, \mathrm{d}x + \lambda \|f - u\|_{L_1},$$

which was studied in Alliney (1996) and Nikolova (2002).

Figure 6.1 shows the graph of the fidelity term in $\mathcal{E}_{EC}$ and $\mathcal{E}_{TV}$ as functions of λ, computed from a given image containing features of different sizes (length scales). The fidelity term in $\mathcal{E}_{EC}$ appears to be a piecewise smooth function of λ, and, strikingly, the discontinuities seem to correspond to some visually drastic change in scale space, *e.g.*, to the disappearances of objects of certain fixed length scales. The intensity of the remaining parts seems to remain constant within each connected smooth part of the graph of $\mathcal{E}_{EC}$. Figure 6.1 shows an example of such decomposition. It is worth noting that in the case of the L^2 decomposition, the intensity of every part of u fades gradually with the increment in λ. See Figure 6.2. It is especially intriguing to realize the inferred connection of this decomposition with human perception of the size of objects in images. As it is pointed out by Esedoglu and Chan (2004), the L^1 scale space suggests a way to select the scale parameter λ_j using the discontinuities in the fidelity term.

Bregman distance and inverse scale space

Recently, Osher, Burger, Goldfarb, Xu and Yin (2005) proposed a different approach to multi-resolution image analysis in scale space. Their proposed method, surprisingly, can be interpreted as a rather unique application of a powerful method, known as Bregman iteration, for constructing the minimizer of convex problems.

Rather than varying λ in the total variation denoising model (1.4), the authors proposed to iteratively 'fortify' selected parts of a given image and subsequently perform the standard TV decomposition using the modified image. Their algorithm can be described as follows.

- Let $u_1 = \arg\min \mathcal{E}(u) + \frac{\lambda}{2}\|f - u\|_{L^2}^2$.
- Define $f = u_1 + v_1$.
- Then inductively, let

$$u_k = \arg\min \mathcal{E}(u) + \frac{\lambda}{2}\|f + v_{k-1} - u\|_{L^2}^2$$

  and $f + v_{k-1} = u_k + v_k$.

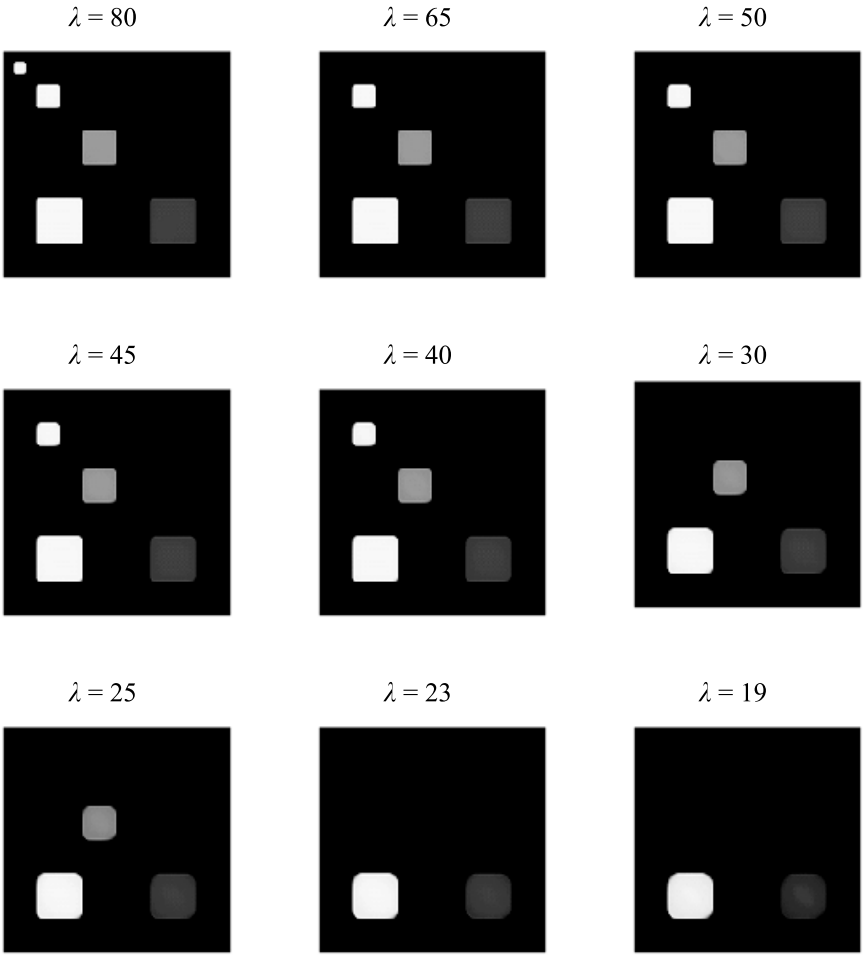

Figure 6.1. Inverse scale space using L^1-BV decomposition.

In other words, the 'noise' v_{k-1} is added back to f and ROF minimization is performed with f replaced by $f + v_{k-1}$ to decompose this function into 'signal' (u_k)+ 'noise' (v_k). Intuitively, if v_k contains both some structural information (*e.g.*, edges) of the optimized cleaned image u as well as the noise, then in the subsequent ROF decomposition, the fitting term will effectively have an inhomogeneous weighting on the locations of the support of v_{k-1}. Noise should be cleaned out from u more rapidly than the structural parts. The authors proved that as $k \to \infty$, $u_k \to f$ monotonically in L_2. In other words, k can be regarded as a parameter for scale; the larger k, the finer the scale information incorporated in u_k. For denoising purpose, the authors observed that one can find a k_0 such that u_{k_0} resembles the ideal

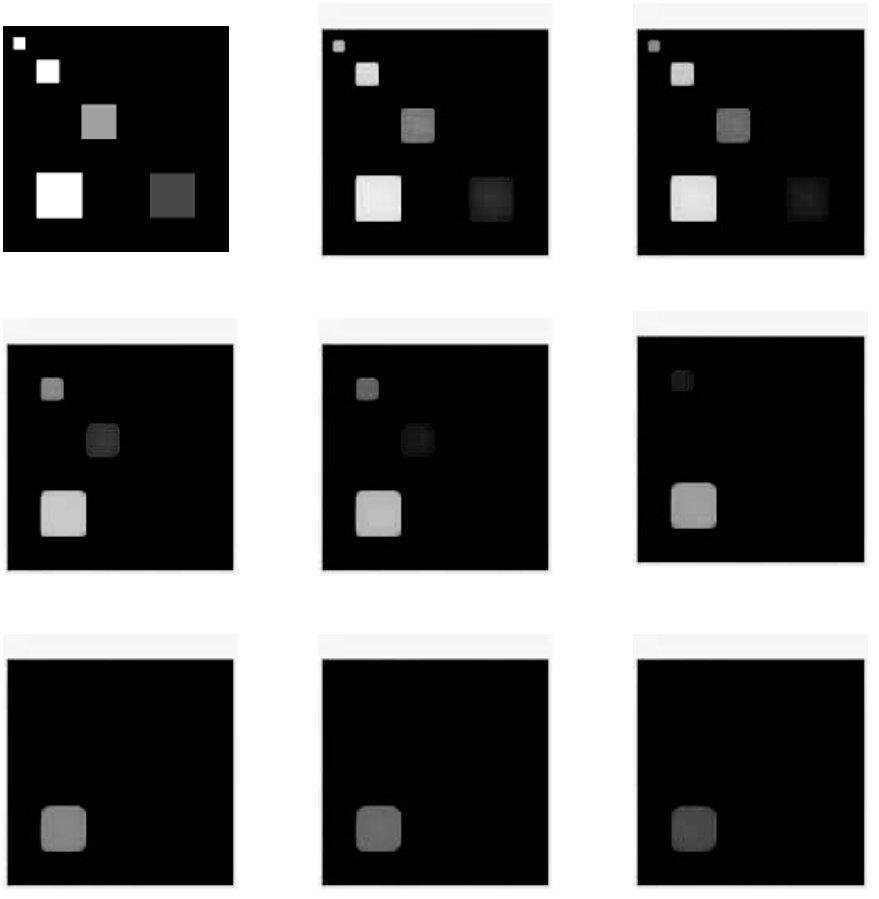

Figure 6.2. Inverse scale space using L^2-BV decomposition.

cleaned image much better than the standard ROF solution. Figure 6.3 shows a decomposition of this sort.

As mentioned above, the procedure outlined above is identical to an iterative procedure using the so-called Bregman distance. Briefly, define a sequence $\{u_k\}$ defined by: Let $u_0 = 0,\ p_0 = 0$, for $k = 1, 2, \ldots$

Compute $u_k = \arg\min Q_k(u)$

$$Q_k : u \to \mathcal{E}(u) - \mathcal{E}(u_{k-1}) - \langle p_{k-1}, u - u_{k-1}\rangle + \frac{\lambda}{2}\|f - u\|_{L^2}^2$$

where $\langle \cdot, \cdot \rangle$ denotes the usual duality product and p_k is the subgradient of $\mathcal{E}(u_k)$. Then compute using the update equation

$$p_k = p_{k-1} + \lambda(f - u_k). \tag{6.1}$$

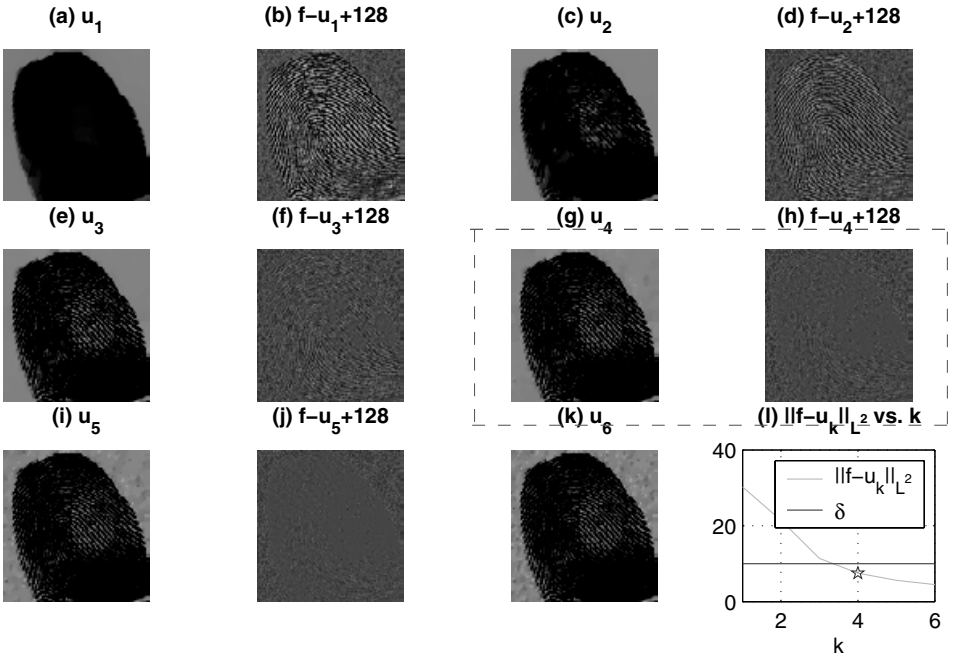

Figure 6.3. Bregman iterative refinement on ROF denoising, 2D finger example. Multi-step. The first k for $\|f - u_k\|_{L^2} < \delta$ is the optimal result.

Actually this procedure works effectively in great generality, for example, deblurring/denoising of images, recovering unknown coefficients for elliptic equations. Of course $\|f - u\|_{L^2}^2$ is replaced by another appropriate fitting term in those examples.

Here we recognize that we are using the Bregman distance between u, u_{k-1}, defined as follows,

$$D(u, v) = \mathcal{E}(u) - \mathcal{E}(v) - \langle u - v, p \rangle, \quad p \in \partial \mathcal{E}(v),$$

where $\partial J(v)$ is the subgradient of the (perhaps) nonstrictly convex function $J(u)$. We have

$$Q_k(u) = D(u, u_{k-1}) + \frac{\lambda}{2} \|f - u\|_{L^2}^2.$$

It was shown in Osher *et al.* (2005) that we obtain a unique sequence of minimizers u_k and subgradients p_k satisfying (6.1) above. The Bregman distance and the associated iteration was not typically used in this fashion in the past. Rather it was used to minimize functions $H(u, f)$ where H is a (usually complicated) convex function of u having a unique minimum: see, *e.g.*, Cetin (1989).

Osher *et al.* (2005) showed that $\{u_k\}$ defined in the sequence satisfies

$$\|u_k - f\|_{L^2}^2 \leq \|u_{k-1} - f\|_{L^2}^2$$

and if $f \in BV(\Omega)$, then

$$\|u_k - f\|_{L^2}^2 \leq \frac{\mathcal{E}(f)}{k},$$

i.e., u_k converges monotonically to f in L^2 with L^2 rate $O\big(\frac{1}{\sqrt{k}}\big)$.

Of course this convergence is not particularly useful to us as a denoising algorithm. The function f is typically noisy. The key denoising result obtained in Osher *et al.* (2005) is as follows.

Let $g \in BV(\Omega)$. Then

$$D(g, u_k) < D(g, u_{k-1})$$

as long as

$$\|f - u_k\|_{L^2}^2 \geq \tau \|g - f\|_{L^2}^2$$

for any $\tau > 1$.

This gives a stopping rule for our iterative procedure. If we have an estimate of the variance of the noise, *i.e.*,

$$f = g + n,$$

where $g \in BV(\Omega)$ is the denoised image and n is the noise, with

$$\|n\|_{L^2} = \sigma,$$

then we stop at the first k for which

$$\|f - u_{k+1}\|_{L^2} \leq \sigma.$$

Burger, Osher and Xu (2005) elegantly reformulated their new algorithm into a continuous flow in scale space, involving the solution of an integro-differential equation. For $\lambda = \Delta t$, $k\Delta t = t$, consider the Bregman iterations written in the form

$$\frac{p(t) - p(t - \Delta t)}{\Delta t} = f - u(t),$$

where

$$p = -\nabla \cdot \frac{\nabla u}{|\nabla u|} = \partial \mathcal{E},$$

and $\mathcal{E}(u) = \int |\nabla u| \, \mathrm{d}x$; *i.e.*, p is the subgradient of $\mathcal{E}$. Letting $\Delta t \downarrow 0$, we arrive at the differential equation

$$\frac{\mathrm{d}p}{\mathrm{d}t} = f - u(t)$$
$$u(t) = u(p(t)), \quad u(0) = p(0) = 0.$$

So if the flow exists and is well behaved, we have an inverse scale space. This means that we start at $u(0) = 0$ and converge as $t \to \infty$, *i.e.*, $\lim_{t\to\infty} u(t) = f$. We go from the smoothest possible image to the noisy image f. The goal is to use the flow to denoise the image, *i.e.*, to get closer initially to the denoised image g, until t crosses a threshold.

As an example, consider the analytically easier case $\mathcal{E}(u) = \frac{1}{2}\int |\nabla u|^2$that is detailed in Scherzer and Groetsch (2001). Then $p = -\Delta u$ with $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$. There is a unique solution for u, given $\int_\Omega p = 0$, $u = -\Delta^{-1}p$. (We also normalize so $\int_\Omega u = \int_\Omega f = 0$.) Simple manipulation leads us to the equation

$$\frac{\mathrm{d}}{\mathrm{d}t}(u - f) = \Delta^{-1}(u - f)$$

or

$$u = f - \mathrm{e}^{\Delta^{-1}t} f \to f.$$

For example, if Ω is the unit square, then we may expand

$$f = \sum_{i,j=1}^{\infty} \tilde{f}_{ij} \cos(\pi i x) \cos(\pi j y),$$

$$u = \sum_{i,j=1}^{\infty} \tilde{u}_{ij} \cos(\pi i x) \cos(\pi j y),$$

where $\tilde{u}_{ij} = \tilde{f}_{ij}\big(1 - \mathrm{e}^{-\frac{t}{(i^2+j^2)\pi}}\big)$.

We refer the reader to Burger *et al.* (2005) for an extension to the important case $\mathcal{E}(u) = \sqrt{|\nabla u|^2 + \epsilon^2}$.

6.3. Diffusion-generated motion and the Esedoglu–Tsai algorithm

Recently, Esedoglu and Tsai, partially motivated by the algorithms presented above, proposed a type of fast segmentation algorithm (Esedoglu and Tsai 2004). Their main algorithm can be regarded as a splitting scheme for the Modica–Mortola functional (5.6) using a thresholding approach similar to the MBO scheme (Merriman *et al.* 1994). The segmentation will be represented by a function v such that $\{v = 0\}$ and $\{v = 1\}$ represent the disjoint partitions in the segmentation. Their algorithm consists of three steps.

(1) Evolve

$$w_t = \Delta w - \frac{\lambda}{\sqrt{\pi\delta t}}\big(w(c_1 - f)^2 + (w - 1)(c_2 - f)^2\big)$$

for $t \in (t_n, t_n + \delta t]$ using $w(t_n) = v_n$ and the periodic boundary condition.

(2) Set

$$v_{n+1} = \begin{cases} 0 & \text{if } w(x, t_n + \delta t) \in (-\infty, \frac{1}{2}], \\ 1 & \text{if } w(x, t_n + \delta t) \in (\frac{1}{2}, \infty). \end{cases}$$

(3) Update c_1 and c_2 by

$$c_1 = \frac{\int_D v f \, \mathrm{d}x}{\int_D v \, \mathrm{d}x}, \quad \text{and} \quad c_2 = \frac{\int_D (1-v) f \, \mathrm{d}x}{\int_D (1-v) \, \mathrm{d}x}.$$

The stiffness of the phase field model (5.6) is resolved by the splitting and the projection to equilibrium (step (2)). Step (1) involves solving linear PDEs with standard Laplacian and can be solved using any mature numerical scheme such as a Fourier method or a multigrid method. The authors studied the consistency of this algorithm by using an asymptotic expansion near the boundary ($\{v = 1/2\}$) and proposed a modified scaling $\tilde{\lambda} = \lambda/\sqrt{\pi \delta t}$ so that the length parameter in the final algorithm scales independently of any other parameters. See Figure 6.4.

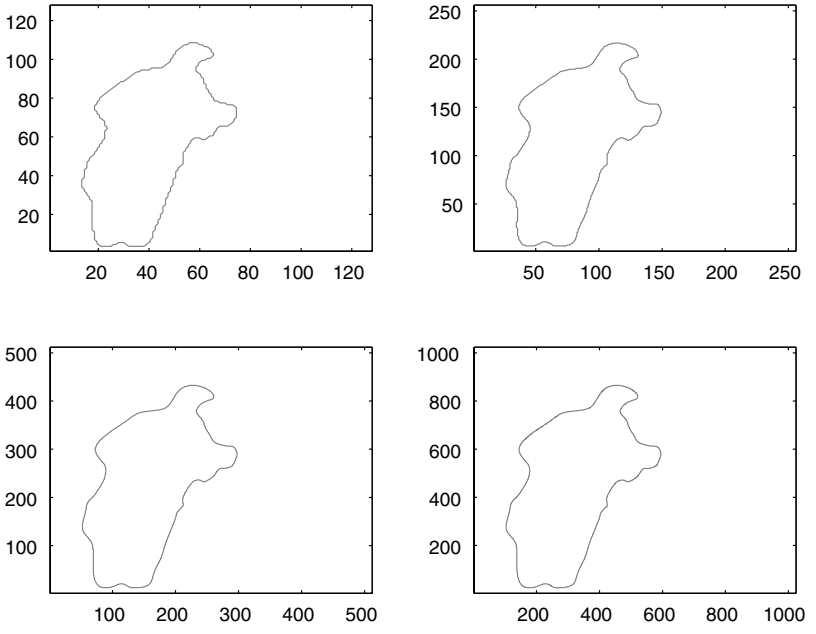

Figure 6.4. Segmentation results in a succession of grid refinements. With the same parameters δt and λ, the algorithms converge in three iterations and produce virtually identical segmentations.

6.4. Computer graphics and beyond

We will see that these efforts combine many different ideas to manipulate more complicated geometrical objects. However, the basic principle and spirit remains unchanged. Burchard *et al.* (2001) and Cheng, Burchard, Merriman and Osher (2002) provided a level set framework to represent and move curves on implicit surfaces or in three-dimensional space. This framework was then generalized to process images and even more general quantities such as vector fields that are defined on nonflat surfaces (Bertalmio *et al.* 2001*b*). Figure 6.5 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, Zhao *et al.* (1996) used multiple level sets to solve a multiphase minimal surface problem. Vese and Chan (2002) further generalized the idea and applied it to image segmentations. This was discussed in the previous section. Smereka (2000) used multiple level sets to define spirals and study the formation of screw dislocations in crystal growth. Liao, Bergsneider, Vese, Huang and Osher (2002) used this approach in brain morphing. Additionally, Smith *et al.* (2002) 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 Burchard *et al.* (2001) and Cheng *et al.* (2002), 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 (Osher *et al.* 2002*a*). This approach makes the manipulation of even more complicated curves and surfaces possible: see Figure 6.6.

Figure 6.5. The image on the right is the denoised and inpainted result from the left.

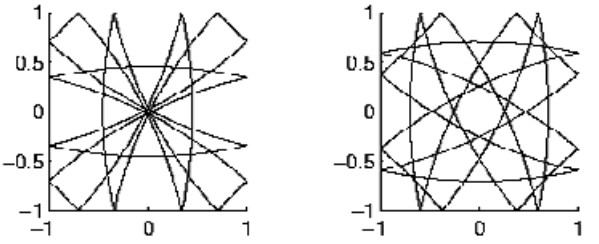

Figure 6.6. These figures shows some complicated curves with self-intersections using the approach in Osher *et al.* (2002*a*).

Visibility

The problem of visibility involves the determination of regions in space visible to a given observer when obstacles to that sight are present. When the observer is replaced by a light source in the simplified geometrical optics setting with perfectly absorbing boundary condition at the obstacles, the problem translates to that of finding illuminated regions.

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 areas related to the accumulation on surfaces of quantities that propagate like light also need visibility information. Examples include etching (Adalsteinsson and Sethian 1997), the formation of huge ice spikes on the Peruvian Andes mountains (Betterton 2001), and shape from shading models (Jin, Yezzi, Tsai, Cheng and Soatto 2005*a*): see Figure 6.7.

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. A very versatile level set method for the visibility problem has recently been developed by the authors and collaborators (Tsai, Cheng, Burchard, Osher and Sapiro 2004). The underlying basic algorithm can be regarded as a multi-level implicit ray tracer that works with volumetric data. Given a level set function ψ describing the obstacles D that obstruct the lines-of-sight, the visibility function $\phi(y; x_0)$ constructed by the algorithm in Tsai *et al.* (2004) takes the form

$$\phi(y; x_0) = \min_{z \in \mathcal{L}(y, x_0)} \psi(z), \tag{6.2}$$

where $\mathcal{L}(y, x_0)$ is the integral curve of the vector field r, connecting y and x_0. The simplicity of this formulation and the associated algorithm facilitates many further extensions and applications.

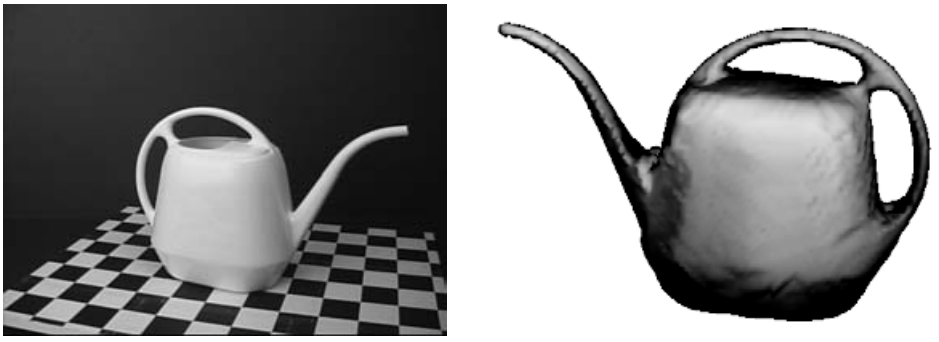

Figure 6.7. The picture on the right shows the reconstructed surface from multiple images on the right (Jin *et al.* 2005*a*).

These algorithms have been applied successfully in reconstructing surfaces from multiple images of different views (Jin *et al.* 2005*a*). They can also be applied directly to some surface renderers, *e.g.*, the 'non-photo-realistic' renderer of Hertzmann and Zorin (2000). In the algorithm defined in Tsai *et al.* (2004), the boundaries of visible and invisible regions, both silhouette and swath[2] (Duguet and Drettakis 2002), are implicitly represented in the framework of Burchard *et al.* (2001) and Cheng *et al.* (2002), mentioned above. Figure 6.8 shows an accumulated visibility result of a path above the Grand Canyon. Figure 6.9 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 a simple Hamilton–Jacobi equation and Tsai *et al.* (2004) offers a near-optimal solution method on the grid. The dynamics of the visibility with respect to moving vantage points 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 involving the visibility numerically and efficiently (Cheng and Tsai 2004).

Let D be the non-reflecting occluders in a domain Ω. Cheng and Tsai (2004) considered the following three central questions that are important in a variety of applications.

- *What is the optimal location x_0 for an observer such that a maximum volume of Ω is visible?*

[2] A swath is a set consisting of the points of intersection of rays which are tangent to an occluder with yet another occluder.

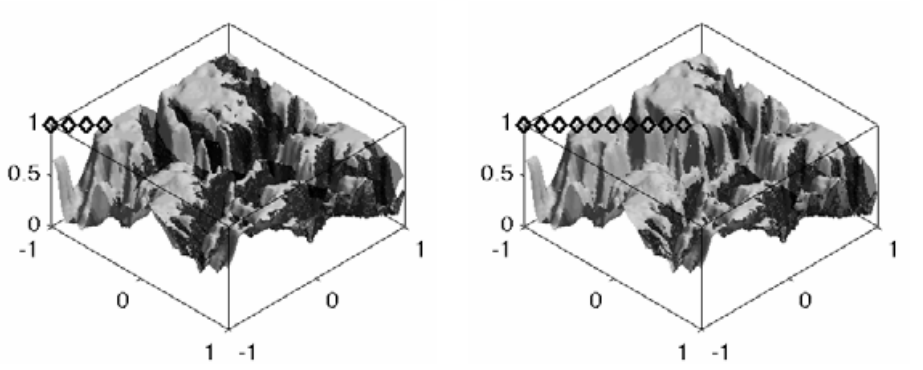

Figure 6.8. The black regions are invisible to the path indicated by the diamonds.

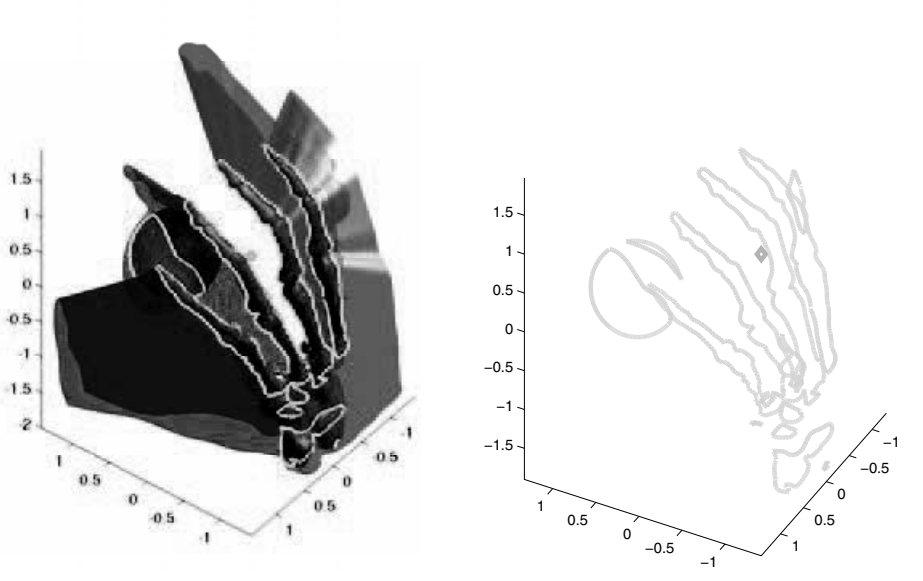

Figure 6.9. The surface borders the visible and invisible regions. The curves indicate the silhouettes and swaths.

A larger class of problems emerges when variations and extensions involving the observer and the space – multiple observers, moving observers, optimality under different measures – are taken into account. For example:

- *What are the optimal locations* $\{x_i\}$ *for a collection of observers, so that jointly a maximum volume of* Ω *is visible?*
- *What is the optimal path* $\gamma(t)$ *of an observer, travelling from* A *to* B*, so that a maximum volume of* Ω *is visible?*

In most situations, it is useful to think of an observer as a light source. Consequently, Cheng and Tsai (2004) approach solving the three central questions by *maximizing the volume of illuminated regions in* Ω*, or maximizing the averaged illumination* (*exposure*) *in* Ω. Two ideas are formulated as the two main variational problems below.

Problem 6.1. (Volume-based visibility optimization) Define $V(x_o)$ as the volume of Ω visible from x_o. Find $x_o \in \mathcal{A} \subseteq \Omega$ such that $V(x_o)$ is maximized. Mathematically,

$$\max_{x_o \in \mathcal{A} \subseteq \Omega} V(x_0) = \int_{\Omega \setminus D} H(\phi(y; x_0))\, \mathrm{d}y.$$

The approach is to introduce an artificial time variable τ and flow x_o from a given initial location to a local maximum. The computation of the gradient of $V(x_o)$ replies heavily on the Lipschitz-continuity of ϕ for accuracy. This, of course, can be generalized to multiple observers and different weighting in space:

$$\max_{x_J \in \mathcal{A} \subseteq \Omega} V(\{x_j\}) = \int_{\Omega \setminus D} w(y, x_1, \cdots x_j) H(\phi(y; \{x_j\})\, \mathrm{d}y.$$

Here $\phi(y; \{x_j\})$ represents the joint visibility of $\{x_j\}$; *i.e.*, $\phi(y; \{x_j\}) \geq 0$ is y is visible to any of x_j.

Define a function $\mathcal{X}$ that counts how many times a given point y can be seen from a collection of observers. This concept can be extended to construct an optimal path for surveillance. Consider the amount of time a point y is exposed to an observer travelling at unit speed along a path $\gamma : [0, 1] \to \mathbb{R}^d$, parametrized by τ,

$$\mathcal{X}(y; \gamma) = \int_0^1 H \circ \phi(y; \gamma(\tau)) |\gamma'(\tau)|\, \mathrm{d}\tau,$$

which we will refer to as the exposure due to γ on x. Points outside obstacles can be said to be viewed in a more uniform manner by an observer moving along γ if the deviation of the exposure from being constant is small for some constant C, as we see in the following problem.

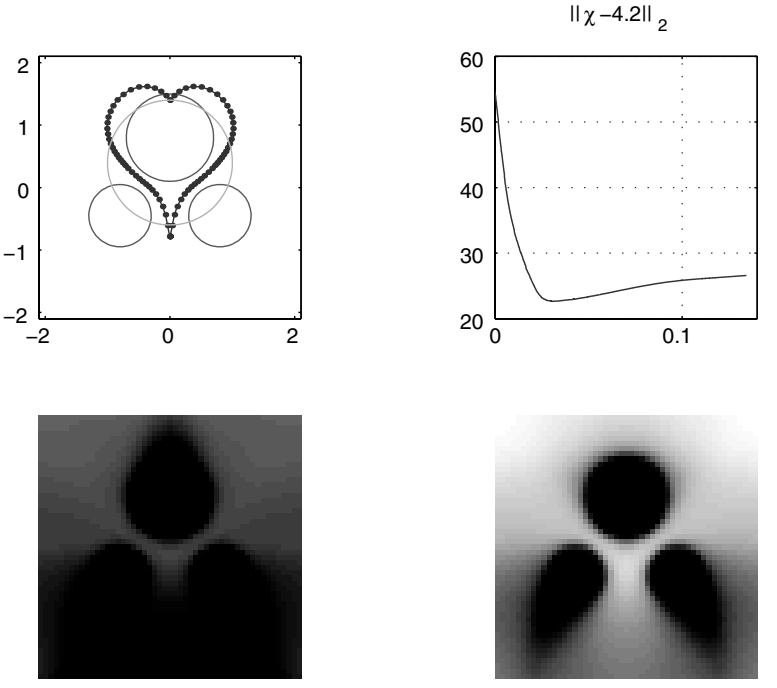

Figure 6.10. The upper left figure shows the occluders (three disjoint circles), the initial curve (the curve intersecting the three occluders), and the optimized curve (dotted curve). The constant C is chosen to be 4.2, the curvature regularization term is 0.05. The images in the second row show the exposure of the initial and the optimized paths.

Problem 6.2. (Exposure-based visibility optimization) Given $p_0, p_1 \in \mathbb{R}^d$, and a constant C, find $\gamma : [0,1] \mapsto \mathbb{R}^d$ with $\gamma(0) = p_0$ and $\gamma(1) = p_1$ minimizing the energy

$$E(\gamma, C) = \frac{1}{2}\|\mathcal{X}(\cdot;\gamma) - C\|_{L^2}^2 + \lambda \int_0^1 |\gamma'(\tau)|\,\mathrm{d}\tau. \tag{6.3}$$

Finally, Cheng and Tsai (2004) considered a time-dependent problem driven by the presence of an evader $y(t)$. The objective is to keep the evader from vanishing into the occlusion. The 'inescapability' of the evader from the pursuer is quantified as the distance between the evader and the observer, and the distance the evader is from the occlusion. Again, taking the advantage of the continuity of the visibility representation, Cheng and

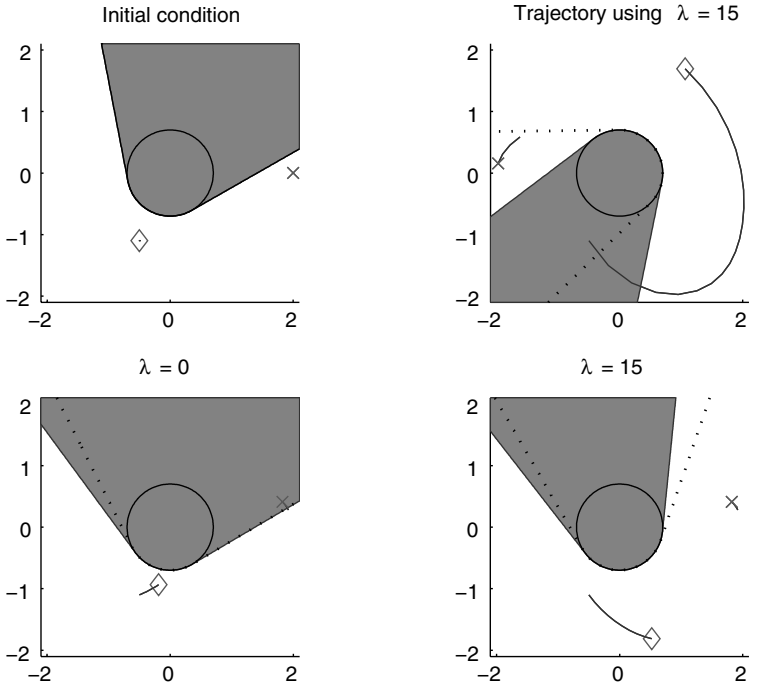

Figure 6.11. Past trajectories due to the absence and presence of the visibility gradient term. The diamonds and crosses indicate the current locations of the observer and the evader, respectively. The lower left plot is with the absence of the gradient term and should be compared to the lower right plot. The upper right plot is a longer time simulation when the gradient term is present.

Tsai defined

$$\mathcal{I}(x_o, y) = \frac{1}{2}|x_0 - y|^2 - \lambda\phi(y; x_0),$$

and formulated the corresponding problem.

Problem 6.3. (Inescapability) Find $x(t)$ so that $I(x_o(t), y(t))$ is strictly decreasing with a prescribed rate.

Various aspects of Problems 6.1 and 6.2 are studied in Cheng and Tsai (2004). Figure 6.10 shows a circular initial path being deformed to a locally optimized path for uniform visibility. Figure 6.11 shows a comparison of the past trajectories of x_o with and without the consideration of maximizing inescapability.

7. Current trends

Currently, higher-order nonlinear PDEs are increasingly appearing in image science. For example, in image inpainting of Chan *et al.* (2002), Esedoglu and Shen (2002) and Lysaker, Osher and Tai (2004), a fourth-order PDE is derived by regularizing the level set curvature a given image. In computer graphics, Tasdizen, Whitaker, Burchard and Osher (2003) proposed performing anisotropic diffusion on the normals of a given level set surface model. In general, fourth-order equations are much harder to analyse, since they rarely have a maximum principle.

An interesting paper of Burchard (2002) discusses the diffusion operators constrained in colour space, which involves a new vector-valued extension of TV minimization.

There are many imaging applications formulated as inverse obstacle problems using level set formulations. Medical imaging contains many such applications. Recently, level set optimization methods have been used for the morphological registration of medical images by Droske and Rumpf (2003/04) and by Vemuri, Ye, Chen and Leonard (2003), where objective functionals similar to elastic energies have been minimized using level set gradient methods. We expect to see more advances in this area.

In many computer graphics simulations using level set formulations, we see an emergence of semi-Lagrangian methods owing to the ease of incorporating them into some adaptive gridding (see Enright, Losasso and Fedkiw (2005), Falcone and Ferretti (2002) and also Losasso, Fedkiw and Osher (2004)). There are efforts to develop Newton method optimization techniques for finding the minima of variational image models. We refer the interested readers to the recent review paper of Burger and Osher (2005) for these problems and applications involving optimal design.

We anticipate increasing efforts in the analysis of the mathematical image models as well as numerical analysis of various aspects of the corresponding algorithms.

Acknowledgements

The authors thank Li-Tien Cheng, Selim Esedoglu, Frédéric Gibou, Jackie Shen, and Luminita Vese for providing their results for this paper.

The first author would like to thank the National Center for Theoretic Study, Taiwan for hosting his stay while parts of this research was being performed.

REFERENCES

D. Adalsteinsson and J. Sethian (1997), 'An overview of level set methods for etching, deposition, and lithography development', *IEEE Trans. Semiconductor Manufacturing* **10**(1), 167–184.

O. Alexandrov and F. Santosa (2005), 'A topological preserving level set method', *J. Comput. Phys.*, to appear.

S. Alliney (1996), 'Recursive median filters of increasing order: A variational approach', *IEEE Trans. Signal Process.* **44**(6), 1346–1354.

F. Alvarez, F. Guichard, J.-M. Morel and P.-L. Lions (1993), 'Axioms and fundamental equations of image processing', *Arch. Rat. Mech. Anal.* **123**, 199–257.

L. Ambrosio and V. M. Tortorelli (1990), 'Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence', *Comm. Pure Appl. Math.* **43**(8), 999–1036.

C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro and J. Verdera (2001), 'Filling-in by joint interpolation of vector fields and gray levels', *IEEE Trans. Image Process.* **10**(8), 1200–1211.

C. Ballester, V. Caselles and J. Verdera (2003), 'Disocclusion by joint interpolation of vector fields and gray levels', *Multiscale Model. Simul.* **2**(1), 80–123 (electronic).

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

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

G. Barles and P. E. Souganidis (1991), 'Convergence of approximation schemes for fully nonlinear second order equations', *Asymptotic Anal.* **4**(3), 271–283.

M. Bertalmio, A. L. Bertozzi and G. Sapiro (2001*a*), 'Navier–Stokes, fluid dynamics, and image and video inpainting', *Proc. of ICCV, IEEE* **1**, I355–I362.

M. Bertalmio, L.-T. Cheng, S. Osher and G. Sapiro (2001*b*), 'Variational problems and partial differential equations on implicit surfaces', *J. Comput. Phys.* **174**(2), 759–780.

M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester (2000), Image inpainting, in *ACM SIGGRAPH*, ACM, pp. 417–424.

M. Bertalmio, L. Vese, G. Sapiro and S. Osher (2003), 'Simultaneous structure and texture image inpainting', *IEEE Trans. Image Process.* **12**(8), 882–889.

M. D. Betterton (2001), 'Formation of structure in snowfields: Penitentes, suncups, and dirt cones', *Phys. Rev. E* **63**, 05629-1-12.

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

P. Burchard (2002), 'Total variation geometry I: Concepts and motivation', UCLA CAM Report # 02-01.

P. Burchard, L.-T. Cheng, B. Merriman and S. Osher (2001), 'Motion of curves in three spatial dimensions using a level set approach', *J. Comput. Phys.* **170**, 720–741.

M. Burger and S. Osher (2005), 'A survey on level set methods for inverse problems and optimal design', *Inverse Problems*, to appear.

M. Burger, B. Hackl and W. Ring (2004), 'Incorporating topological derivatives into level set methods', *J. Comput. Phys.* **194**(1), 344–362.

M. Burger, S. Osher and J.-J. Xu (2005), 'Iterative refinement and inverse scale space methods for image restoration', in preparation.

V. Caselles, J. Morel and C. Sbert (1998), 'An axiomatic approach to image interpolation', *IEEE Trans. Image Process.* **7**(3), 376–386.

A. Cetin (1989), 'Reconstruction of signals from Fourier transform samples', *Signal Processing* **16**, 129–148.

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

T. Chan and H. M. Zhou (1999), 'Adaptive ENO-wavelet transforms for discontinuous functions', UCLA CAM Report # 99-21.

T. F. Chan and L. A. Vese (2001*a*), 'Active contours without edges', *IEEE Trans. Image Process.* **10**(2), 266–277.

T. F. Chan and L. A. Vese (2001*b*), 'A level set algorithm for minimizing the Mumford and Shah model functional in image processing', *Proc. 1st IEEE Workshop on 'Variational and Level Set Methods in Computer Vision'*, pp. 161–168.

T. F. Chan and H.-M. Zhou (2002), 'ENO-wavelet transforms for piecewise smooth functions', *SIAM J. Numer. Anal.* **40**(4), 1369–1404.

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

T. Chan, J. Shen and H.-M. Zhou (2004), 'Total variation wavelet inpainting', UCLA CAM Report # 04-47.

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

L.-T. Cheng (2000), 'The level set method applied to geometrically based motion, materials science, and image processing', UCLA CAM Report # 00-20.

L.-T. Cheng and Y.-H. R. Tsai (2004), 'Visibility optimization using variational approaches', UCLA CAM Report # 04-03.

L.-T. Cheng, P. Burchard, B. Merriman and S. Osher (2002), 'Motion of curves constrained on surfaces using a level set approach', *J. Comput. Phys.* **175**, 604–644.

D. L. Chopp (2001), 'Some improvements of the fast marching method', *SIAM J. Sci. Comput.* **23**(1), 230–244 (electronic).

B. Cockburn and C.-W. Shu (1989), 'TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework', *Math. Comp.* **52**(186), 411–435.

M. G. Crandall and P.-L. Lions (1983), 'Viscosity solutions of Hamilton–Jacobi equations', *Trans. Amer. Math. Soc.* **277**(1), 1–42.

M. G. Crandall and P.-L. Lions (1984), 'Two approximations of solutions of Hamilton–Jacobi equations', *Math. Comp.* **43**, 1–19.

M. G. Crandall, H. Ishii and P.-L. Lions (1992), 'User's guide to viscosity solutions of second order partial differential equations', *Bull. Amer. Math. Soc.* (N.S.) **27**(1), 1–67.

P.-E. Danielsson (1980), 'Euclidean distance mapping', *Computer Graphics and Image Processing* **14**, 227–248.

A. Dervieux and F. Thomasset (1979), A finite element method for the simulation of Rayleigh–Taylor instability, in *Approximation Methods for Navier–Stokes Problems*, Vol. 771 of *Lecture Notes in Mathematics*, Springer, pp. 145–158.

A. Dervieux and F. Thomasset (1981), Multifluid incompressible flows by a finite element method, in Vol. 11 of *Lecture Notes in Physics*, pp. 158–163. Lectures Notes in Physics, Vol.141, 158-163 (1981)

M. Droske and M. Rumpf (2003/04), 'A variational approach to nonrigid morphological image registration', *SIAM J. Appl. Math.* **64**(2), 668–687 (electronic).

F. Duguet and G. Drettakis (2002), Robust epsilon visibility, in *Proc. ACM SIGGRAPH 2002* (J. Hughes, ed.), Annual Conference Series, ACM Press/ACM SIGGRAPH, pp. 567–575.

B. Engquist, A. Harten and S. Osher (1987), A high order essentially nonoscillatory shock capturing method, in *Large Scale Scientific Computing: Oberwolfach, 1985*, Birkhäuser, Boston, MA, pp. 197–208.

B. Engquist, A.-K. Tornberg and Y.-H. Tsai (2004), 'Discretization of Dirac-δ functions in level set methods', UCLA CAM Report # 04-16. Under review.

D. Enright, R. Fedkiw, J. Ferziger and I. Mitchell (2002), 'A hybrid particle level set method for improved interface capturing', *J. Comput. Phys.* **183**, 83–116.

D. Enright, F. Losasso and R. Fedkiw (2005), 'A fast and accurate semi-Lagrangian particle level set method', UCLA CAM Report # 03-58. *Computers and Structures* **83**, 479-490.

S. Esedoglu and T. Chan (2004), 'Aspects of total variation regularized L^1 function approximation', UCLA CAM Report # 04-07.

S. Esedoglu and S. Osher (2005), 'Decomposition of images by the anisotropic Rudin–Osher–Fatemi model', *Comm. Pure Appl. Math.*, to appear.

S. Esedoglu and J. Shen (2002), 'Digital inpainting based on the Mumford-Shah-Euler image model', *European J. Appl. Math.* **13**, 353–370.

S. Esedoglu and Y.-H. Tsai (2004), 'Threshold dynamics for the piecewise Mumford-Shah functional', UCLA CAM Report # 04-63.

L. C. Evans (1998), *Partial Differential Equations*, AMS, Providence, RI.

L. C. Evans and R. F. Gariepy (1992), *Measure Theory and Fine Properties of Functions*, Studies in Advanced Mathematics, CRC Press, Boca Raton, FL.

L. C. Evans and J. Spruck (1991), 'Motion of level sets by mean curvature I', *J. Differential Geom.* **33**(3), 635–681.

L. C. Evans and J. Spruck (1992*a*), 'Motion of level sets by mean curvature II', *Trans. Amer. Math. Soc.* **330**(1), 321–332.

L. C. Evans and J. Spruck (1992*b*), 'Motion of level sets by mean curvature III', *J. Geom. Anal.* **2**(2), 121–150.

L. C. Evans and J. Spruck (1995), 'Motion of level sets by mean curvature IV', *J. Geom. Anal.* **5**(1), 77–114.

D. J. Eyre (1998), Unconditionally gradient stable time marching: The Cahn–Hilliard equation, in *Computational and Mathematical Models of Microstructural Evolution: San Francisco, CA, 1998*, Vol. 529 of *Mater. Res. Soc. Sympos. Proc.*, MRS, Warrendale, PA, pp. 39–46.

M. Falcone and R. Ferretti (2002), 'Semi-Lagrangian schemes for Hamilton–Jacobi equations, discrete representation formulae and Godunov methods', *J. Comput. Phys.* **175**(2), 559–575.

F. Gibou and R. Fedkiw (2005), A fast level set based algorithm for segmentation, in *Proc. 4th Hawaii International Conference on Statistics, Mathematics and Related Fields*, pp. 281–291. Stanford Technical Report (2002).

Y. Giga (2002), Surface evolution equations: A level set method, Vol. 44 of *Lipschitz Lecture Notes*, University of Bonn.

S. K. Godunov (1959), 'A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics', *Mat. Sb.* (N.S.) **47 (89)**, 271–306.

A. Haddad and Y. Meyer (2004), 'Variational methods in image processing', UCLA CAM Report # 04-52.

X. Han, C. Xu and J. Prince (2003), Topology preserving geometric deformable models for brain reconstruction, in *Geometric Level Set Methods in Imaging, Vision, and Graphics* (S. Osher and N. Paragios, eds), Springer, New York, pp. 421–438.

E. Harabetian and S. Osher (1998), 'Regularization of ill-posed problems via the level set approach', *SIAM J. Appl. Math.* **58**(6), 1689–1706 (electronic).

A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy (1987), 'Uniformly high-order accurate essentially nonoscillatory schemes III', *J. Comput. Phys.* **71**(2), 231–303.

J. Helmsen, E. Puckett, P. Colella and M. Dorr (1996), Two new methods for simulating photolithography development in 3D, in *SPIE 2726*, pp. 253–261.

A. Hertzmann and D. Zorin (2000), Illustrating smooth surfaces, in *ACM SIGGRAPH*, ACM, pp. 517–526.

G.-S. Jiang and D. Peng (2000), 'Weighted ENO schemes for Hamilton–Jacobi equations', *SIAM J. Sci. Comput.* **21**(6), 2126–2143 (electronic).

H. Jin, A. Yezzi, Y.-H. Tsai, L. T. Cheng and S. Soatto (2005*a*), 'Estimation of 3D surface shape and smooth radiance from 2D images: A level set approach', *J. Sci. Comput.*, to appear.

S. Jin, H. Liu, S. Osher and R. Tsai (2005*b*), 'Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation', `www.levelset.com/download/density.pdf`, *J. Comput. Phys.*, to appear.

C. Y. Kao, S. Osher and J. Qian (2004), 'Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations', *J. Comput. Phys.* **196**, 367–391.

C. Y. Kao, S. Osher and Y.-H. Tsai (2002), 'Fast sweeping methods for Hamilton–Jacobi equations', UCLA CAM Report # 02-66.

J. B. Keller (1962), 'Geometrical theory of diffraction', *J. Opt. Soc. Amer.* **52**, 116–130.

W.-H. Liao, M. Bergsneider, L. Vese, S.-C. Huang and S. Osher (2002), 'From landmark matching to shape and open curve matching: A level set approach', UCLA CAM Report # 02-59.

X.-D. Liu, S. Osher and T. Chan (1994), 'Weighted essentially non-oscillatory schemes', *J. Comput. Phys.* **115**(1), 200–212.

F. Losasso, R. Fedkiw and S. Osher (2004), 'Spatially adaptive techniques for level set methods and incompressible flow', UCLA CAM Report # 04-67.

M. Lysaker, S. Osher and X.-C. Tai (2004), 'Noise removal using smoothed normals and surface fitting', *IEEE Trans. Image Process.* **13**(10), 1345–1357.

A. Marquina and S. Osher (2000), '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).

S. Masnou and J.-M. Morel (1998), Level-lines based disocclusion, in *Proc. 5th IEEE International Conference on Image Processing: Chicago, IL*, pp. 259–263.

B. Merriman, J. K. Bence and S. J. Osher (1994), 'Motion of multiple functions: A level set approach', *J. Comput. Phys.* **112**(2), 334–363.

Y. Meyer (2001), *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, AMS, Providence, RI. The fifteenth Dean Jacqueline B. Lewis memorial lectures, Rutgers University.

D. Mumford and J. Shah (1989), 'Optimal approximations by piecewise smooth functions and associated variational problems', *Comm. Pure Appl. Math.* **42**(5), 577–685.

M. Nikolova (2002), 'Minimizers of cost functionals involving nonsmooth data fidelity terms', *SIAM J. Numer. Anal.* **40**(3), 965–994.

M. Nitzberg, D. Mumford and T. Shiota (1993), *Filtering, Segmentation and Depth*, Vol. 662 of *Lecture Notes in Computer Science*, Springer, Berlin.

A. Oberman (2004), 'A convergent upwind difference scheme for motion by mean curvature', *Numer. Math.* **99**, 365–379.

S. Osher (1993), 'A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations', *SIAM J. Math. Anal.* **24**(5), 1145–1152.

S. Osher (2003), Level set methods, in *Geometric Level Set Methods in Imaging, Vision, and Graphics* (S. Osher and N. Paragios, eds), Springer, New York, pp. 3–20.

S. Osher and R. Fedkiw (2002), *Level Set Methods and Dynamic Implicit Surfaces*, Springer, New York.

S. Osher and R. P. Fedkiw (2001), 'Level set methods: an overview and some recent results', *J. Comput. Phys.* **169**(2), 463–502.

S. Osher and J. Helmsen (2005), 'A generalized fast algorithm with applications to ion etching', in preparation.

S. Osher and B. Merriman (1997), 'The Wulff shape as the asymptotic limit of a growing crystalline interface', *Asian J. Math.* **1**(3), 560–571.

S. Osher and J. A. Sethian (1988), 'Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations', *J. Comput. Phys.* **79**(1), 12–49.

S. Osher and C.-W. Shu (1991), 'High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations', *SIAM J. Numer. Anal.* **28**(4), 907–922.

S. Osher, M. Burger, D. Goldfarb, J.-J. Xu and W. Yin (2005), 'An iterative regularization method for total variation based image restoration', *Multiscale Model. Simul.*, to appear.

S. Osher, L.-T. Cheng, M. Kang, H. Shim and Y.-H. Tsai (2002*a*), 'Geometric optics in a phase-space-based level set and Eulerian framework', *J. Comput. Phys.* **179**(2), 622–648.

S. Osher, A. Sole and L. Vese (2003), 'Image decomposition and restoration using total variation minimization and the H^{-1} norm', *Multiscale Model. Simul.* **1**, 344–370.

N. Paragios and R. Deriche (1997), A PDE-based level set approach for detection and tracking of moving objects, Technical Report 3173, INRIA, France.

D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang (1999*a*), 'A PDE-based fast local level set method', *J. Comput. Phys.* **155**(2), 410–438.

D. Peng, S. Osher, B. Merriman and H.-K. Zhao (1999*b*), The geometry of Wulff crystal shapes and its relations with Riemann problems, in *Nonlinear Partial Differential Equations: Evanston, IL, 1998*, AMS, Providence, RI, pp. 251–303.

C. S. Peskin (2002), The immersed boundary method, in *Acta Numerica*, Vol. 11, Cambridge University Press, pp. 479–517.

E. Rouy and A. Tourin (1992), 'A viscosity solutions approach to shape-from-shading', *SIAM. J. Numer. Anal.* **29**(3), 867–884.

L. Rudin and S. Osher (1994), 'Total variation based restoration with free local constraints', *Proceedings ICIP, IEEE, Austin, TX* pp. 31–35.

L. I. Rudin and S. Osher (1990), 'Feature-oriented image enhancement using shock filters', *SIAM J. Numer. Anal.* **27**(4), 919–940.

L. Rudin, S. Osher and E. Fatemi (1992), 'Nonlinear total variation based noise removal algorithms', *Phys. D* **60**(1–4), 259–68.

O. Scherzer and C. Groetsch (2001), Inverse scale space theory for inverse problems, in *Scale-space and Morphology in Computer Vision: Proc. 3rd International Conference Scale Space*, Springer, pp. 317–325.

J. Sethian (1996), Fast marching level set methods for three dimensional photolithography development, in *SPIE 2726*, pp. 261–272.

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

C.-W. Shu (1997), Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, ICASE Report 97-65, NASA.

C.-W. Shu and S. Osher (1988), 'Efficient implementation of essentially non-oscillatory shock-capturing schemes', *J. Comput. Phys.* **77**(2), 439–471.

P. Smereka (2000), 'Spiral crystal growth', *Phys. D* **138**(3–4), 282–301.

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

J. Sokołowski and J.-P. Zolésio (1992), *Introduction to Shape Optimization*, Vol. 16 of *Springer Series in Computational Mathematics*, Springer, Berlin.

B. Song and T. Chan (2002), 'Fast algorithm for level set based optimization', UCLA CAM Report # 02-68.

P. Soravia (1994), 'Generalized motion of a front propagating along its normal direction: a differential games approach', *Nonlinear Anal.* **22**(10), 1247–1262.

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

R. Spiteri and S. Ruuth (2005), 'A new class of optimal high-order strong-stability-preserving time discretization methods', preprint.

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

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

J. Strain (1999*b*), 'Semi-Lagrangian methods for level set equations', *J. Comput. Phys.* **151**(2), 498–533.

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

E. Tadmor, S. Nezzar and L. Vese (2004), 'A multiscale image representation using hierarchical (BV, L^2) decompositions', *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal* **02**(4), 554–579.

T. Tasdizen, R. Whitaker, P. Burchard and S. Osher (2003), 'Geometric surface processing via normal maps', *ACM Trans. Graphics* **22**(4), 1012–1033.

A. Tornberg (2002), 'Multi-dimensional quadrature of singular and discontinuous functions', *BIT* **42**, 644–669.

A. Tornberg and B. Engquist (2003), 'Regularization techniques for numerical approximation of PDEs with singularities', *J. Sci. Comput.* **19**, 527–552.

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

Y.-H. R. Tsai, L.-T. Cheng, P. Burchard, S. Osher and G. Sapiro (2004), 'Visibility and its dynamics in a PDE based implicit framework', *J. Comput. Phys.* **199**(06), 260–290.

Y.-H. R. Tsai, L.-T. Cheng, S. Osher and H.-K. Zhao (2003*a*), 'Fast sweeping methods for a class of Hamilton–Jacobi equations', *SIAM J. Numer. Anal.* **41**(2), 673–699.

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

J. Tsitsiklis (1995), 'Efficient algorithms for globally optimal trajectories', *IEEE Trans. Automatic Control* **40**(9), 1528–1538.

B. Vemuri, J. Ye, Y. Chen and C. Leonard (2003), 'Image registration via level set motion: applications to atlas-based segmentation', *Medical Image Analysis* **7**, 1–20.

L. A. Vese and T. F. Chan (2002), 'A multiphase level set framework for image segmentation using the Mumford and Shah model', *Internat. J. Computer Vision*.

L. A. Vese and S. J. Osher (2003), 'Modeling textures with total variation minimization and oscillating patterns in image processing', *J. Sci. Comput.* **19**(1–3), 553–572.

L. Vese and S. Osher (2002), 'The level set method links active contours, Mumford-Shah segmentation, and Total Variation restoration', UCLA CAM Report # 02-05.

J.-J. Xu and H.-K. Zhao (2003), 'An Eulerian formulation for solving partial differential equations along a moving interface', *J. Sci. Comput.* **19**(1–3), 573–594.

Y.-T. Zhang, H.-K. Zhao and J. Qian (2004), 'High order fast sweeping methods for static Hamilton–Jacobi equations', UCLA CAM Report # 04-37.

H.-K. Zhao (2005), 'Fast sweeping method for eikonal equations', *Math. Comp.*, to appear.

H.-K. Zhao and S. Osher (2003), Visualization, analysis and shape reconstruction of sparse data, in *Geometric Level Set Methods in Imaging, Vision, and Graphics* (S. Osher and N. Paragios, eds), Springer, New York, pp. 361–380.

H.-K. Zhao, T. Chan, B. Merriman and S. Osher (1996), 'A variational level set approach to multiphase motion', *J. Comput. Phys.* **127**, 179–195.

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

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