

**UCLA**  
**COMPUTATIONAL AND APPLIED MATHEMATICS**

---

**The Level Set Method:  
Applications to Imaging Science**

**Stanley Osher**

**July 2002**

**CAM Report 02-43**

---

**Department of Mathematics  
University of California, Los Angeles  
Los Angeles, CA. 90095-1555**

**<http://www.math.ucla.edu/applied/cam/index.html>**

# The Level Set Method: Applications to Imaging Science

Stanley Osher<sup>1</sup>  
Mathematics Department  
UCLA  
Los Angeles, CA 90095-1555  
sjo@math.ucla.edu

ABSTRACT The level set method for capturing moving fronts was introduced in 1987 by Osher and Sethian [27]. It has proven to be phenomenally successful as a numerical device. For example, as of June 2002, typing in “Level Set Methods” on Google’s search engine gives roughly 2800 responses and the original article has been cited over 530 times (according to web of science). Applications range from capturing multiphase fluid dynamical flows, to graphics, e.g. special effects in Hollywood, to visualization, image processing, control, epitaxial growth, computer vision and include many others. In this chapter we shall give an overview of the numerical technology and of applications in imaging science. These will include surface interpolation, solving PDE’s on manifolds, visibility, ray tracing, segmentation (including texture segmentation) and restoration.

## 1 Introduction

The original idea behind the level set method was a simple one. Given an interface  $\Gamma$  in  $R^n$  of codimension one, bounding a (perhaps multiply connected) open region  $\Omega$ , we wish to analyze and compute its subsequent motion under a velocity field  $v$ . This velocity can depend on position, time, the geometry of the interface (e.g., its normal or its mean curvature), and the external physics. The idea, as devised in 1987 by S. Osher and J.A. Sethian [27] is merely to define a smooth (at least Lipschitz continuous) function  $\varphi(\mathbf{x}, t)$ , that represents the interface as the set where  $\varphi(\mathbf{x}, t) = 0$ . Here  $\mathbf{x} = (x_1, \dots, x_n) \in R^n$ .

---

<sup>1</sup>Supported in part by NSF DMS 0074735, ONR N00014-97-0027 and NIH P20MH65166

The level set function  $\varphi$  has the following properties:

$$\begin{aligned}\varphi(\mathbf{x}, t) &< 0 && \text{for } x \in \Omega \\ \varphi(\mathbf{x}, t) &> 0 && \text{for } x \notin \bar{\Omega} \\ \varphi(\mathbf{x}, t) &= 0 && \text{for } x \in \partial\Omega = \Gamma(t).\end{aligned}$$

Thus, the interface is to be captured for all later time, by merely locating the set  $\Gamma(t)$  for which  $\varphi$  vanishes. This deceptively trivial statement is of great significance for numerical computation, primarily because topological changes such as breaking and merging are well defined and performed “without emotional involvement”.

The motion is analyzed by convecting the  $\varphi$  values (levels) with the velocity field  $v$ . This elementary equation is

$$\frac{\partial\varphi}{\partial t} + \mathbf{v} \cdot \nabla\varphi = 0. \quad (1.1)$$

Here  $\mathbf{v}$  is the desired velocity on the interface and is arbitrary elsewhere.

Actually, only the normal component of  $v$  is needed,  $v_N = \mathbf{v} \cdot \frac{\nabla\varphi}{|\nabla\varphi|}$ , so (1.1) becomes

$$\frac{\partial\varphi}{\partial t} + v_N |\nabla\varphi| = 0. \quad (1.2)$$

Here  $|\nabla\varphi| = \sqrt{\sum_{i=1}^n \varphi_{x_i}^2}$ .

In those cases where  $v_N$  is a constant, equation (1.2) is nonlinear. In fact it is a first order Hamilton-Jacobi equation. If  $v_N \equiv 1$  what the zero level set,  $\Gamma(t)$ , of  $\varphi(\mathbf{x}, t)$  at later time  $t > 0$  represents is the set of all points at a positive distance  $t$  from the original interface  $\Gamma$ . The first thing one notices is that this distance function develops “kinks”.

For example, if one starts with a square  $\Gamma$  and moves it outwards a unit distance, obtaining  $\Gamma(1)$ , then associated with each of four quarter circles of radius 1 centered at each vertex of the original square, we have a continuum of points whose closest point on  $\Gamma$  is the same for the whole continuum – namely the closest vertex of  $\Gamma$ . If we move  $\Gamma(1)$  backwards one unit, we thus have a singular distance function at each vertex of  $\Gamma$ .

Such mild singularities are typical of the correct solutions to this Hamilton-Jacobi equation. Fortunately the theory of viscosity solutions, which was invented in [14],[13], exactly picks out this unique Lipschitz continuous solution as do the solutions to the discrete approximations developed in [27],[28],[20] and discussed below. Thus, users can have confidence that their computer simulations give accurate, unique solutions. A particularly interesting result is in [17], where motion by mean curvature, as defined by Osher and Sethian in [27], is shown to be essentially the same motion as is obtained from the asymptotics in the phase field reaction diffusion equation. The motion in the level set method involves no superfluous stiffness as required in phase field methods.

The outline of this chapter is as follows: In section 2 we present the key definition and basic level set technology. In section 3 we discuss the numerical implementation. In section 4 we briefly describe some applications in imaging science, including several very recent results. We present some concluding remarks in section 5.

## 2 Level Set Dictionary and Technology

We list key terms and define them by their level set representation.

1. The interface boundary  $\Gamma(t)$  is defined by  $\{\mathbf{x}|\varphi(\mathbf{x}, t) = 0\}$ . The region  $\Omega(t)$  is bounded by  $\Gamma(t) : \{\mathbf{x}|\varphi(\mathbf{x}, t) < 0\}$  and its exterior is defined by  $\{\mathbf{x}|\varphi(\mathbf{x}, t) > 0\}$ .

2. The unit normal  $\mathbf{N}$  to  $\Gamma(t)$  is given by

$$\mathbf{N} = \frac{\nabla\varphi}{|\nabla\varphi|}.$$

3. The mean curvature  $\kappa$  of  $\Gamma(t)$  is defined by

$$\kappa = -\nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right).$$

4. The Dirac delta function concentrated on an interface is

$$\delta(\varphi)|\nabla\varphi|,$$

where  $\delta(x)$  is a one-dimensional delta function.

5. The characteristic function  $\chi$  of a region  $\Omega(t)$  is

$$\chi = H(-\varphi)$$

where

$$\begin{aligned} H(x) &\equiv 1 && \text{if } x > 0 \\ H(x) &\equiv 0 && \text{if } x < 0 \end{aligned}$$

is a one-dimensional Heaviside function.

6. The surface (or line) integral of a quantity  $P(\mathbf{x}, t)$  over  $\Gamma$  is

$$\int_{R^n} p(\mathbf{x}, t) \delta(\varphi) |\nabla\varphi| d\mathbf{x}.$$

7. The volume (or area) integral of  $p(\mathbf{x}, t)$  over  $\Omega$  is

$$\int_{R^n} p(\mathbf{x}, t) H(-\varphi) d\mathbf{x}.$$

Next we describe some key technological advances which are important in many, if not most, level set calculations.

8. The distance reinitialization procedure replaces a general level set function  $\varphi(\mathbf{x}, t)$  by  $d(\mathbf{x}, t)$  which is the value of the distance from  $\mathbf{x}$  to  $\Gamma(t)$ , positive outside and negative inside. This assures us that  $\varphi$  does not become too flat or too steep near  $\Gamma(t)$ . Let  $d(\mathbf{x}, t)$  be signed distance of  $\mathbf{x}$  to the closest point on  $\Gamma$ . The quantity  $d(\mathbf{x}, t)$  satisfies  $|\nabla d| = 1$ ,  $d > 0$  in  $\Omega$ ,  $d < 0$  in  $(\bar{\Omega})^c$ , and is the steady state solution (as  $\tau \rightarrow \infty$ ) to

$$\begin{aligned} \frac{\partial \psi}{\partial \tau} + \operatorname{sgn}(\varphi)(|\nabla \psi| - 1) &= 0 \\ \psi(\mathbf{x}, 0) &= \varphi(\mathbf{x}, t), \end{aligned} \quad (1.3)$$

where  $\operatorname{sgn}(x) = 2H(x) - 1$  is the one-dimensional signum function. This procedure was designed in [36]. The key observation is that in order to define  $d$  in a band of width  $\epsilon$  around  $\Gamma$ , we need solve (1.3) only for  $\tau = O(\epsilon)$ . It can easily be shown that this can be used globally to construct distance (with arbitrary accuracy) in  $O(N \log N)$  iterations [29]. Alternatively, we may use Tsitsiklis' fast algorithm [40],[41], which is also  $O(N \log N)$ , with a much smaller constant, but which is only first-order accurate. A locally second-order accurate (in the high-resolution sense) fast marching method was proposed in [33]. While this method has a much lower local truncation error than a purely first-order accurate method, it is still globally first-order accurate except for special cases. We might also use the fast sweeping method from [7],[39] which has  $O(N)$  complexity (rigorously proven in [45]) and which is also only first-order accurate. Finally, we note that recently Tsai [38] developed an alternative elegant and interesting higher order approach for computing distance quite accurately.

9. Smooth extension of a quantity, e.g.,  $v_n$  on  $\Gamma$  to a neighborhood of  $\Gamma$ . Let the quantity be  $p(\mathbf{x}, t)$ . Solve to steady state ( $\tau \rightarrow \infty$ )

$$\begin{aligned} \frac{\partial q}{\partial \tau} + \operatorname{sgn}(\varphi) \left( \frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla q \right) &= 0 \\ q(\mathbf{x}, 0) &= p(\mathbf{x}, t). \end{aligned}$$

Again, we need only solve this for  $\tau = O(\epsilon)$  in order to extend  $p$  to be constant in the direction normal to the interface in a tube of width  $\epsilon$ . This was first suggested and implemented in [11], analyzed carefully in [46], and further discussed and implemented in both [18], and [29]. A computationally efficient algorithm based on heap sort technology and fast marching methods was devised in [2]. There are many reasons to extend a quantity off of  $\Gamma$ , one of which is to obtain a well-conditioned normal velocity for level contours of  $\varphi$  close to  $\varphi = 0$  [11].

10. The basic level set method concerns a function  $\varphi(\mathbf{x}, t)$  which is defined throughout space. Clearly this is wasteful if one only cares about information near the zero level set. The local level set method defines  $\varphi$

only near the zero level set. We may solve (1.2) in a neighborhood of  $\Gamma$  of width  $m\Delta x$ , where  $m$  is typically 5 or 6. Points outside of this neighborhood need not be updated by this motion. This algorithm works in “ $\varphi$ ” space—so not too much intricate computer science is used. For details see [29]. Thus this local method works easily in the presence of topological changes and for multiphase flow. An earlier local level set approach called “narrow banding” was devised in [1].

11. High codimensional motion. The level set method was originally developed for curves in  $R^2$  and surfaces in  $R^3$ . Problems involving the motion of curves in  $R^3$  are usually done by front tracking. However, the usual problems of merging and pinching off may occur. In [8] a formulation was derived to make use of two level set functions to model a curve in  $R^3$ . In this formulation, a curve is represented by the intersection between the zero level set functions  $\phi$  and  $\psi$ , i.e., where  $\phi = \psi = 0$ . From this, many properties of the curve can be derived, such as the tangent vectors,  $\mathbf{T} = \nabla\psi \times \nabla\phi / |\nabla\psi \times \nabla\phi|$ , the curvature vectors,  $\kappa\mathbf{N} = \nabla\mathbf{T} \cdot \mathbf{T}$ , and even the torsion,  $\tau\mathbf{N} = -\nabla\mathbf{B} \cdot \mathbf{T}$ , where  $\mathbf{N}$  and  $\mathbf{B}$  are the normal and binormal respectively.

Motions of the curve can then be studied under the appropriate system of PDE’s involving the two level set functions. The velocity can depend on external physics, as well as on the geometry of the curve (as in the standard level set approach). The resulting system of PDEs for  $\psi$  and  $\phi$  is

$$\begin{aligned}\phi_t &= -\mathbf{v} \cdot \nabla\phi \\ \psi_t &= -\mathbf{v} \cdot \nabla\psi.\end{aligned}$$

A simple example involves moving the curve according to its curvature vectors, for which  $v = \kappa\mathbf{N}$ . See [8] for many examples and more details. This method is called the vector level set method.

An interesting application of high codimensional motion came in [25]. There a level set based approach for ray tracing and for the construction of wavefronts in geometric optics was developed. The goal, as described in [25], is to solve for an evolving curve in  $(x, y, \theta)$  space (where  $\theta$  is the angle of the normal to the curve) or on  $(x, y, z, \theta, \varphi)$  space (where  $\theta, \varphi$  are the polar coordinate angles of the normal to the surface). This leads to 2 and 3 codimensional problems – evolving curves in 3D and surfaces in 5D. We solve for 2 level set functions in 3D or 3 level set functions in 5D to carry this out. This is all tractable because of new local level set methods.

### 3 Numerical Methods

In the important special case where  $v_N$  in Eq. 2 is a function only of  $\mathbf{x}, t$ , and  $\nabla\varphi$  (e.g.,  $v_N = 1$ ), Eq. 2 becomes a Hamilton-Jacobi (H-J) equation whose solutions generally develop kinks (jumps in derivatives). We seek the unique viscosity solution. Many good references exist for this important subject;

see, e.g. [4],[15]. The appearance of these singularities in the solution means that special, but not terribly complicated, numerical methods have to be used, usually on uniform Cartesian grids. This was first discussed in [27] and numerical schemes developed there were generalized in [20],[28]. The key ideas involve monotonicity, upwind differencing, essentially nonoscillatory (ENO) schemes, and weighted essentially nonoscillatory (WENO) schemes. In this section we present some of the details and motivation.

Typically a level set equation involves a velocity field  $\mathbf{v}$  which depends on the geometry of the zero level set and external quantities which can be thought of as depending on  $\mathbf{x}$  and  $t$ . A prototype is

$$\mathbf{v} = \mathbf{a}(\mathbf{x}) \frac{\nabla\varphi}{|\nabla\varphi|} - \mu(\mathbf{x}) \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) \quad (1.4)$$

or a more “linear” version is

$$\mathbf{v} = \mathbf{a}(\mathbf{x}) - \mu(\mathbf{x}) \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) \quad (1.5)$$

Here  $\mu > 0$  and the equations

$$\varphi_t + \mathbf{a}(\mathbf{x})|\nabla\varphi| = \mu(\mathbf{x})|\nabla\varphi| \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) \quad (1.6)$$

$$\varphi_t + \mathbf{a}\mathbf{x} \cdot \nabla\varphi = \mu(\mathbf{x})|\nabla\varphi| \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) \quad (1.7)$$

involve convection (in the normal direction with normal velocity  $\mathbf{a}(\mathbf{x})$  in (1.6) or  $\mathbf{a}(\mathbf{x}) \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right)$  in (1.7)) and curvature dependent motion (defined on as the right hand side with velocity  $-\mu(\mathbf{x}) \frac{\nabla\varphi}{|\nabla\varphi|} \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right)$  or equivalently normal velocity  $-\mu(\mathbf{x}) \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right)$ ).

For simplicity of exposition only we will compute problems in  $R^2$ , so  $\mathbf{x} = (x, y)$ . Also, we will describe the methods on a uniform Cartesian grid,  $x_j = j\Delta x$ ,  $j = 0, \pm 1, \pm 2 \dots$   $y_k = k\Delta y$ ,  $k = 0, \pm 1, \pm 2 \dots$  and again for simplicity only, we take  $\Delta x = \Delta y = \Delta$ . Also  $t^n = n\Delta t$ ,  $n = 0, 1, 2 \dots$ . We shall be using  $\varphi_{j,k}^n$  as an approximation to  $\varphi(j\Delta, k\Delta, n\Delta t)$ . The finite difference scheme we use will involve the following operators:

$$D_{\pm}^x \varphi_{jk} = \pm \frac{(\varphi_{j\pm 1, k} - \varphi_{jk})}{\Delta}, D_{\pm}^y \varphi_{jk} = \pm \frac{(\varphi_{j, k\pm 1} - \varphi_{jk})}{\Delta}.$$

We shall discretize (1.6) and (1.7) in three stages

(1) Approximate the convection term using ideas developed in [27],[28], borrowed from their origins in the numerical solution of conservation laws [19],[34],[35].

(2) Approximate the curvature term by central difference methods.

(3) Use TVD Runge-Kutta schemes, derived in [34] to do the time discretization.

We begin with the approximation of the convection term in (1.7). A simple first order accurate upwind difference approximation to  $\mathbf{a}(x) \cdot \nabla \varphi$  is

$$(a_1(x_j, y_k, t^n))^+ D_-^x \varphi_{jk} + (a_1(x_j, y_k, t^n))^- D_+^x \varphi_{jk} \\ + (a_2(x_j, y_k, t^n))^+ D_-^y \varphi_{jk} + a_2(x_j, y_k, t^n)^- D_+^y \varphi_{jk}.$$

Here  $a = (a_1, a_2)$  and  $x^+ = \max(x, 0)$ ,  $x^- = \min(x, 0)$ . The forward and backwards differences are chosen so as to follow the direction of propagation of the characteristics.

We notice that the linear interpolant through the data points  $(x_{j-1}, \varphi_{j-1,k}), (x_j, \varphi_{jk})$  which is

$$I_{j-\frac{1}{2},k}(x) = \varphi_{j,k} + (x - x_j) D_- \varphi_{jk}$$

has the property

$$\frac{d}{dx} I_{j-\frac{1}{2}}^1(x) \Big|_{x=x_j} = D_- \varphi_{jk}.$$

To get higher order, say  $m^{\text{th}}$  order, space accuracy, we must use a polynomial of degree  $m$  which interpolates  $(x_{j-1}, \varphi_{j-1,k}), (x_j, \varphi_{jk})$  and  $m-1$  additional data points. These should be of the form

$$(x_{j-\nu}, \varphi_{j-\nu,k}), (x_{j-\nu+1}, \varphi_{j-\nu+1,k}) \dots (x_{j-\nu+m}, \varphi_{j-\nu+m,k})$$

with  $1 \leq \nu \leq m$ . The question is: which polynomial to choose? This question was originally answered in [19]. We choose the smoothest polynomial i.e. the one with the smaller 2<sup>nd</sup>, then 3<sup>rd</sup> then...  $m^{\text{th}}$  divided differences. This is done inductively. For example,

$$I_{j-\frac{1}{2},k}^2(x) = I_{j-\frac{1}{2},k}^1(x) + \frac{(x-x_j)(x-x_{j-1})}{2} m(D_-^2 \varphi_{jk}, D_- D_+ \varphi_{jk})$$

where

$$m(x, y) = x \text{ if } |x| \leq |y| \\ = y \text{ otherwise.}$$

This general procedure is called ENO (essentially nonoscillatory) interpolation. For more details see [19].

Thus the  $m^{\text{th}}$  order ENO approximation to  $\mathbf{a}(x) \cdot \nabla \varphi$  is:

$$(a_1(x_j, y_k, t^n))^+ \left( \frac{d}{dx} I_{j-\frac{1}{2},k}^m(x) \right)_{x=x_j} \\ + (a_1(x_j, y_k, t^n))^- \left( \frac{d}{dx} I_{j-\frac{1}{2},k}^m(x) \right)_{x=x_j} \\ + (a_2(x_j, y_k, t^n))^+ \left( \frac{d}{dy} I_{j,k-\frac{1}{2}}^m(y) \right)_{y=y_k} \\ + (a_2(x_j, y_k, t^n))^- \left( \frac{d}{dy} I_{j,k+\frac{1}{2}}^m(y) \right)_{y=y_k}.$$



We typically use  $m = 3$  in real applications.

For general nonlinear Hamilton-Jacobi equations of the form

$$\varphi_t + H(x, y, t, \varphi, \varphi_x, \varphi_y) = 0 \quad (1.8)$$

where  $H$  is the Hamiltonian, we can again use ENO interpolation to approximate  $H$ . We began by constructing a numerical Hamiltonian having the following properties. (For simplicity of exposition, we ignore the  $(x_j, y_k, \varphi_{jk}^n)$  dependence):

$$\hat{H}(D_+^x \varphi_{jk}, D_-^x \varphi_{jk}, D_+^y \varphi_{jk}, D_-^y \varphi_{jk}) = \hat{H}(u^+, u^-; v^+, v^-) \quad (1.9)$$

$\hat{H}$  is a Lipschitz continuous function which is

(1) Consistant:  $\hat{H}(u, u; v, v) = H(u, v)$

(2) Monotone:  $\hat{H}(\downarrow, \uparrow; \downarrow, \uparrow)$ , i.e.  $\hat{H}$  is nonincreasing in its first and third arguments and nondecreasing in its second and fourth arguments.

There are four monotone numerical Hamiltonians of interest:

$$\hat{H}^G(u^+, u^-; v^+, v^-) = \text{ext}_{u \in I(u^-, u^+), v \in I(v^-, v^+)} H(u, v) \quad (1.10)$$

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

$$\text{ext}_u I(a, b) = \begin{cases} \min_{a \leq u \leq b} & \text{if } a \leq b \\ \max_{a \geq u \geq b} & \text{if } a > b \end{cases} . \quad (1.11)$$

See [3],[28] for a derivation and motivation. This is the canonical monotone upwind scheme. As an example, if

$$H(u, v) = \sqrt{u^2 + v^2}$$

then

$$\hat{H}^G(u^+, u^-, v^+, v^-) = \sqrt{\max((u^+)^-, ((u^-)^+)^2 + \max((v^+)^-, ((v^-)^+)^2)},$$

see [30].

Unfortunately, the formula in (1.10) often becomes complicated near sonic points, i.e. those for which  $H_u = 0$  or  $H_v = 0$ .

A simpler monotone numerical Hamiltonian is

$$\begin{aligned} \hat{H}^{LF}(u^+ u^-; v^+, v^-) &= H\left(\frac{u^+ + u^-}{2}; \frac{v^+ + v^-}{2}\right) \\ &- \alpha \frac{(u^+ - u^-)}{2} - \beta \frac{(v^+ - v^-)}{2} \end{aligned} \quad (1.12)$$

where  $\alpha, \beta$  are positive constants chosen so that

$$\begin{aligned} \alpha &> \max_{\substack{u \in [A, B] \\ v \in [C, D]}} |H_u(u, v)|, & \beta &> \max_{\substack{u \in [A, B] \\ v \in [C, D]}} |H_v(u, v)| \end{aligned} \quad (1.13)$$

where  $[A, B]$  is the value range for  $u^\pm$  and  $[C, D]$  is the value range for  $v^\pm$ .

A third candidate is:

$$\begin{aligned} \hat{H}^{LLF}(u^+, u^-, v^+, v^-) &= H\left(\frac{u^+ + u^-}{2}, \frac{v^+ + v^-}{2}\right) \\ &- \alpha^+(u^+, u^-) \frac{(u^+ - u^-)}{2} - \beta^+(v^+, v^-) \frac{(v^+ - v^-)}{2} \end{aligned} \quad (1.14)$$

where

$$\begin{aligned} \alpha(u^+, u^-) &= \max_{\substack{u \in I(u^-, u^+) \\ v \in [C, D]}} |H_u(u, v)|, & \beta(v^+, v^-) &= \max_{\substack{v \in I(v^-, v^+) \\ u \in [A, B]}} |H_v(u, v)| \end{aligned} \quad (1.15)$$

The fourth and final candidate is

$$\hat{H}^{RF}(u^+, u^-, v^+, v^-) = \begin{cases} H(u^*, v^*) & \text{if } H_1(u, v) \text{ and } H_2(u, v) \\ & \text{do not change sign in} \\ & u \in I(u^-, u^+) \text{ and} \\ & v \in I(v^-, v^+); \\ \hat{H}^{RF}(u^+, u^-, v^+, v^-) - \alpha(u^+, u^-) \frac{u^+ - u^-}{2} & \text{else if } H_2(u, v) \text{ does not} \\ & \text{change sign in } u \in [A, B] \\ & \text{and } v \in I(v^-, v^+); \\ H\left(u^*, \frac{v^+ + v^-}{2}\right) - \beta(v^+, v^-) \frac{v^+ - v^-}{2} & \text{else if } H_1(u, v) \text{ does not} \\ & \text{change sign in } v \in [C, D] \\ & \text{and } u \in I(u^-, u^+); \\ \hat{H}^{LLF}(u^+, u^-, v^+, v^-) & \text{otherwise,} \end{cases} \quad (1.16)$$

where  $u^*, v^*$  are defined by upwinding

$$u^* = \begin{cases} u^+ & \text{if } H_1(u, v) \leq 0, \\ u^- & \text{if } H_1(u, v) \geq 0; \end{cases} \quad v^* = \begin{cases} v^+ & \text{if } H_2(u, v) \leq 0, \\ v^- & \text{if } H_2(u, v) \geq 0. \end{cases} \quad (1.17)$$

The four Hamiltonians may be ordered monotonically increasing in terms of increasing numerical diffusion and increasing programming simplicity as

$$\hat{H}^G < \hat{H}^{RF} < \hat{H}^{LLF} < \hat{H}^{LF}$$

i.e. the left side is generally less diffusive and more complicated to program than the right side of each inequality.

Now to obtain an  $m^{\text{th}}$  order ENO approximation to  $H(x, y, t, \varphi, \varphi_x, \varphi_y)$  we merely use

$$\hat{H}(x_j, y_k, t^n, \varphi_{jk}, \left(\frac{d}{dx} I_{j+\frac{1}{2}, k}^m(x)\right)_{x=x_j}, \left(\frac{d}{dx} I_{j-\frac{1}{2}, k}^m(x)\right)_{x=x_j}; \quad (1.18)$$

$$\left( \frac{d}{dy} I_{j,k+\frac{1}{2}}^m(y) \right)_{y=y_k} , \left( \frac{d}{dy} I_{j,k-\frac{1}{2}}^m(y) \right)_{y=y_k} .$$

In [22] and then later [21],[20] the idea was put forward of taking a weighted combination of polynomials to approximate  $\varphi_x, \varphi_y$ . The formalism is still as in equation 1.18, but using different polynomials  $I^m$ . The weights are chosen so as to default to ENO in the presence of discontinuities, but to obtain optimal accuracy in smooth regions. See [20] for the details. The method seems robust and is desirable for problems which are of one scale and where high order accuracy is important.

We now know how to approximate the convective terms in (1.6) and (1.7). The right hand side diffusive term is handled with central differencing of the following sort

$$|\nabla\varphi| \approx \sqrt{\frac{1}{2}(D_+^x\varphi_{jk})^2 + (D_-^x\varphi_{jk})^2 + (D_+^y\varphi_{jk})^2 + (D_-^y\varphi_{jk})^2} \tag{1.19}$$

$$\frac{\partial}{\partial x} \frac{\varphi_x}{|\nabla\varphi|} \approx \left( \frac{D_+^x\varphi_{jk}}{\sqrt{(D_+^x\varphi_{jk})^2 + (D^y\varphi_{jk})^2 + \epsilon}} \right)$$

where  $(D^y\varphi_{jk})^2 = \frac{1}{4} ((D_+^y\varphi_{jk})^2 + (D_+^y\varphi_{j+1,k})^2 + (D_-^y\varphi_{jk})^2 + (D_-^y(\varphi_{j+1,k}))^2)$  and  $\epsilon > 0$  is a small parameter, used to avoid dividing by zero,  $\epsilon \approx 10^{-10}$  in our calculations.

An analogous expression is used to approximate  $\frac{\partial}{\partial y} \frac{\varphi_y}{|\nabla\varphi|}$ .

This discretization results in a second order accurate approximation to the diffusion term, which is adequate for all reasonable calculations.

Finally, we use a third order total variation diminishing (TVD) Runge-Kutta scheme, devised by Shu-Osher in [34] to perform the time integration.

If we have a semidiscrete approximation of the form

$$\frac{\partial\varphi_{jk}}{\partial t} = L_{jk},$$

then the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> order TVD Runge-Kutta methods are

$$\begin{aligned} \varphi_{jk}^{n+1} &= \varphi_{jk}^n + \Delta t L_{jk}^n && \text{(1st order)} && (1.20) \\ \left\{ \begin{aligned} \varphi_{jk}^{n+\frac{1}{2}} &= \varphi_{jk}^n + \Delta t L_{jk}^n && \text{(2nd order)} \\ \varphi_{jk}^{n+1} &= \frac{\varphi_{jk}^{n+\frac{1}{2}} + \varphi_{jk}^n}{2} + \frac{\Delta t}{2} L_{jk}^{n+\frac{1}{2}} \end{aligned} \right. \\ \left\{ \begin{aligned} \varphi_{jk}^{n+\frac{1}{3}} &= \varphi_{jk}^n + \Delta t L_{jk}^n \\ \varphi_{jk}^{n+\frac{2}{3}} &= \frac{3}{4}\varphi_{jk}^n + \frac{1}{4}\varphi_{jk}^{n+\frac{1}{3}} + \frac{1}{4}\Delta t L_{jk}^{n+\frac{1}{3}} \\ \varphi_{jk}^{n+1} &= \frac{1}{3}\varphi_{jk}^n + \frac{2}{3}\varphi_{jk}^{n+\frac{2}{3}} + \frac{2}{3}\Delta t L_{jk}^{n+\frac{2}{3}} \end{aligned} \right. && \text{(3rd order)} \end{aligned}$$

The time step restriction is of the form

$$\frac{\Delta t}{\Delta} \leq c_1 + c_2 \Delta$$

with  $c_2$  proportional to  $|\mu|$  and  $c_1$  proportional to  $\max_{u \in [A,B], v \in [C,D]} (|H_u|, |H_v|)$

## 4 Imaging Science Applications

There are far too many applications of geometric level set methods in imaging science to discuss here. I will confine myself to a few classical and new applications which I find to be of special interest.

### 4.1 Dynamic Visibility

The problem is easily stated: Given a collection of closed surfaces representing objects in space, determine quickly the regions (in space or on the surfaces) that are visible to an observer. This question is crucial to applications in fields as diverse as rendering, visualization, etching, and the solution of inverse problems. In a 3D virtual-reality environment, knowing the visible region speeds up the rendering by enabling us to skip costly computations on occluded regions.

As the theme of this book emphasizes, representing a family of surfaces implicitly as the zero level set of a *single* function  $\varphi(\vec{x})$ ,  $\vec{x} = (x, y, z)$ , has several advantages, particularly when things are moving. Topological changes are easily handled (without “emotional involvement”), and geometric quantities, such as normals and curvatures, are also easily computed dynamically. Most published work in computer graphics and computer vision uses explicit surfaces, usually constructed with triangles, but this is changing. The upcoming SIGGRAPH conference (in San Antonio, July 2002) will have many LSM-related papers and a full-day course on LSM and PDE-based methods in graphics. For a recent detailed report on the visibility problem with explicit surfaces, see [16].

In our approach to the dynamic visibility problem, as described in [37], we begin by laying down a simple Cartesian grid—this is step one in almost all level set applications. Next, we compute the signed distance function  $\varphi(\vec{x})$  to the occluding region  $\Omega$  (which generally has many disjoint pieces). Here we can use the optimally fast algorithm of Tsitsiklis [40],[41]. The function  $\varphi$  approximately satisfies the eikonal equation:

$$\sqrt{\varphi_x^2 + \varphi_y^2} = |\nabla\varphi| = 1 \quad (1.21)$$

with

$$\begin{aligned} \varphi(\mathbf{x}) &< 0 \text{ in } \Omega, \\ \varphi(\mathbf{x}) &> 0 \text{ in } \Omega^c, \end{aligned}$$

and

$$\varphi(\mathbf{x}) = 0 \text{ on } \partial\Omega. \quad (1.22)$$

Then, for a given vantage point  $\mathbf{x}_0$ , we use a simple, easily parallelizable, multiresolution algorithm of optimal complexity to compute the visibility function  $\psi_{\mathbf{x}_0}(\mathbf{x})$ :

$$\psi_{\mathbf{x}_0}(\mathbf{x}) \geq 0 \Leftrightarrow \mathbf{x} \text{ is visible to } \mathbf{x}_0. \quad (1.23)$$

Next, we allow  $\mathbf{x}$  to move with velocity  $d\mathbf{x}_0/dt$ . Along the way, we obtain fairly elegant geometric-based formulae for the motion of the horizon. This is defined to be the set of visible points  $\mathbf{x}$  lying in  $\partial\Omega$  for which  $\mathbf{x} - \mathbf{x}_0$  is orthogonal to  $\partial\Omega$ , i.e., for which

$$\psi_{\mathbf{x}_0}(\mathbf{x}) = 0 = \varphi(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_0) \cdot \nabla\varphi(\mathbf{x}). \quad (1.24)$$

We do the same for the motion of points on the cast horizon – that is, points that lie both on the extension of the ray connecting a horizon point  $\mathbf{x}_0$  and to  $\mathbf{x}$  and on an occluder.

$\psi_{\mathbf{x}_0(t)}(\mathbf{x}, t)$  can be found quickly at each discrete time step. As  $\mathbf{x}_0(t)$  flies through a region, we can compute the invisible region at time  $t$  by computing the intersection of the sets

$$S_\tau = \{\mathbf{x} | \psi_{\mathbf{x}_0(\tau)}(\mathbf{x}, \tau) < 0\}. \quad (1.25)$$

## 4.2 Interpolation from Unorganized Points via the Level Set Method

Hong-Kai Zhao and I have written a review article on this subject in chapter 16 of this book. See that chapter and [48],[47] for details. Briefly, one finds a level set function whose zero level set passes through a collection of unorganized data points, curves and/or surface patches. Among the advantages of this method is the following: The interpolating surfaces obtained this way can easily be denoised. I shall discuss this in section 4.4 below.

## 4.3 PDE's on Implicit Surfaces

M. Bertalmio, L.-T. Cheng, G. Sapiro and myself have also written a review article on this subject Chapter 18. See that chapter and [5],[6] for details. We believe that this new technique will enable a user to solve PDE's on very complicated surfaces without intricate gridding. One interesting related effort involves moving curves on fixed surfaces in [12]. There we fix one of the two level set functions  $\psi$ , defined in the discussion of high codimension motion in Section 2 of this chapter, i.e. we just set  $\psi_t = 0$  and choose  $\mathbf{v}$  as desired. See [12] for results and details.

#### 4.4 *The Level Set Method Links Active Contours, Mumford-Shah Segmentation and Total Variation Restoration*

This is based on work done jointly with Luminita Vese in [42]. There we proposed new models for active contours and image segmentation based on the Rudin-Osher-Fatemi [32] total variation (TV) minimization, done using a level set framework. We basically do TV restoration on functionals of level set functions. The new TV-based segmentation model has an additional scaling feature given by the magnitude of the jump. We are now minimizing a convex functional over a nonconvex set. The resulting method is free of some of the restrictive properties of the Mumford-Shah model. As in the Chan-Vese active contour segmentation model (see e.g. L. Vese's Chapter 7 in this book and the references therein) we see that interior contours are automatically detected as well as contours with or without steep gradients.

We begin by introducing our notation. Let  $\Omega \subset \mathbb{R}^N$  be an open and bounded domain, and  $u_0 : \Omega \rightarrow \mathbb{R}$  be a given observed image. We consider  $N = 1$  for signals,  $N = 2$  for plane images, and  $N = 3$  for volumetric images. We also use  $\mathcal{H}^{N-1}(\Gamma)$  to denote the  $(N-1)$ -dimensional Hausdorff measure of a hyper-surface  $\Gamma \subset \Omega$ .  $\mathcal{H}^0$  gives the counting measure,  $\mathcal{H}^1$  the length or the perimeter, and  $\mathcal{H}^2$  the area.

The Mumford and Shah model [24] has been proposed for image segmentation and partition. By this model, one tries to find piecewise-smooth optimal approximations  $u$  of the initial image  $u_0$ . The model is

$$\inf_{u, \Gamma} F^{MS}(u, \Gamma) = \int_{\Omega} (u - u_0)^2 dx + \nu \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \mu \mathcal{H}^{N-1}(\Gamma), \quad (1.26)$$

where  $\mu, \nu$  are positive (tuning) parameters. The first term in (1.1) insures that the segmented-reconstructed image  $u$  is an approximation of the observation  $u_0$ ; the second term insures that  $u$  is smooth outside the set  $\Gamma$ ; finally, the third term asks that the set  $\Gamma$  be smooth enough. This set will approximate the edges or contours of  $u$ .

Another important model proposed for image reconstruction is the total variation minimization of Rudin-Osher-Fatemi [32],[31]:

$$\inf_u G^{TV}(u) = \int_{\Omega} (u - u_0)^2 dx + \mu \int_{\Omega} |\nabla u|. \quad (1.27)$$

By this model, the initial noisy image  $u_0$  is well denoised, while preserving edges.

If we minimize the two functionals above restricted to the subset  $\{u = H(\phi) | \phi : \Omega \rightarrow \mathbb{R} \text{ is Lipschitz}\}$ , i.e. of characteristic functions, then the two functionals are the same:

$$\begin{aligned} F^{MS}(u = H(\phi), \Gamma = \{\phi = 0\}) &= G^{TV}(u = H(\phi)) \\ &= \int_{\Omega} (u_0 - 1)^2 H(\phi) dx + \int_{\Omega} (u_0 - 0)^2 (1 - H(\phi)) dx + \mu \int_{\Omega} |\nabla H(\phi)| \end{aligned}$$

The minimization problem:

$$\inf_{\phi} E(\phi) = \int_{\Omega} (u_0 - 1)^2 H(\phi) dx + \int_{\Omega} (u_0 - 0)^2 (1 - H(\phi)) dx + \mu \int_{\Omega} |\nabla H(\phi)|$$

has appeared in Osher-Kang [26] for denoising curves representing letters from fax transmissions, but this is also a particular case of the Chan-Vese model for active contours without edges based level sets and Mumford-Shah segmentation form [9],[10]. It also acts as a general method for denoising surfaces where we take  $u_0 = H(\varphi_0)$  and the initial surface to denoise is the zero level set of  $\varphi_0$ . See also [44]. If we restrict the energy (1.3) to the more general piecewise constant case

$$\{u(x) = c^+ H(\phi(x)) + c^- (1 - H(\phi(x))), c^+, c^- \in \mathbb{R}, \phi : \Omega \rightarrow \mathbb{R} \text{ Lipschitz}\},$$

we can write the above energy function of  $c^+, c^-, \phi$  as:

$$G(c^+, c^-, \phi) = \int_{\Omega} (u_0 - c^+)^2 H(\phi) dx + \int_{\Omega} (u_0 - c^-)^2 (1 - H(\phi)) dx + \mu |c^+ - c^-| \int_{\Omega} |\nabla H(\phi(x))|. \quad (1.28)$$

We consider now  $C^1(\mathbb{R})$  approximations and regularizations  $H_{\varepsilon}$  and  $\delta_{\varepsilon}$  of the Heaviside function and the Dirac Delta function  $\delta_0$ , as  $\varepsilon \rightarrow 0$ , such that  $H'_{\varepsilon} = \delta_{\varepsilon}$ .

Minimizing the above energy, and embedding the gradient descent into a dynamic scheme, we obtain:

$$\begin{aligned} c^+(t) &= \frac{\int_{\Omega} u_0(x) H_{\varepsilon}(\phi(t, x)) dx}{\int_{\Omega} H_{\varepsilon}(\phi(t, x)) dx} \\ &\quad - \left( \frac{c^+(t) - c^-(t)}{|c^+(t) - c^-(t)|} \right) \left( \frac{\mu \int_{\Omega} \delta_{\varepsilon}(\phi(t, x)) |\nabla \phi(t, x)| dx}{2 \int_{\Omega} H_{\varepsilon}(\phi(t, x)) dx} \right), \\ c^-(t) &= \frac{\int_{\Omega} (x)(1 - H_{\varepsilon}(\phi(t, x))) dx}{\int_{\Omega} (1 - H_{\varepsilon}(\phi(t, x))) dx} \\ &\quad - \left( \frac{c^-(t) - c^+(t)}{|c^-(t) - c^+(t)|} \right) \left( \frac{\mu \int_{\Omega} \delta_{\varepsilon}(\phi(t, x)) |\nabla \phi(t, x)| dx}{2 \int_{\Omega} (1 - H_{\varepsilon}(\phi(t, x))) dx} \right), \\ \frac{\partial \phi}{\partial t} &= \delta_{\varepsilon}(\phi) \left[ \mu |c^+ - c^-| \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - (u_0 - c^+)^2 + (u_0 - c^-)^2 \right]. \end{aligned}$$

If we compare the equations from this TV-based model and the Chan-Vese model from [9],[10] now the values  $c^+$  and  $c^-$  are no longer the averages of the image  $u_0$  on the corresponding regions. In addition, there is an extra factor  $|c^+ - c^-|$  in the regularization term, multiplied by  $\mu$ . As we see in the numerical results of [42], this extra factor, the jump has the following role: if we apply both models to the same image and with the same regularizing parameter  $\mu$ , the TV-based model has a stronger constraint

on the total length, therefore less noise will be kept by the new TV-based model, comparing with the Chan-Vese model based on Mumford-Shah.

There are many extensions possible. In particular  $u(x)$  can be taken to be of the form:

$$u(x) = u^+(x)H(\varphi(x)) + u^-(x)(I - H(\varphi(x))) \text{ for } u^+, u^- \text{ both } C^1$$

functions or defined in their domains of definition. Typically, a space of polynomials of a fixed degree might be appropriate.

for numerical results and more details see [42].

#### 4.5 Modeling Textures with Total Variation Minimization and Oscillating Patterns

In many problems of image analysis, we have an observed image  $f$ , representing a real scene. The image  $f$  may contain noise (some random pattern of zero mean for instance) and/or texture (some repeated pattern of small scale details). The image processing task is to extract the most meaningful information from  $f$ . This is usually formulated as an inverse problem: given  $f$ , find another image  $u$ , “close” to  $f$ , such that  $u$  is a cartoon or simplification of  $f$ . In general,  $u$  is an image formed by homogeneous regions and with sharp boundaries. Most models assume the following relation between  $f$  and  $u$ :  $f = u + v$ , where  $v$  is noise or small scale repeated detail (texture), and extract only the  $u$  component from  $f$ . Usually, the component  $v$  is not kept, assuming that this models the noise. A successful algorithm of this type is due to Rudin-Osher-Fatemi [32],[31], see equation (1.27).

In some cases, the  $v$  component is important, especially if it represents texture. Texture can be defined as a repeated pattern of small scale details. The noise is also a pattern of small scale details, but of random, uncorrelated values. Both types of patterns (additive noise or texture) can be modeled by oscillatory functions taking both positive and negative values, and of zero mean [23].

With Luminata Vese in [43], we showed how we can extract from  $f$  both components  $u$  and  $v$ , in a simple total variation minimization framework of Rudin-Osher-Fatemi [32]. The obtained decomposition can then be useful for segmentation of textured images and texture discrimination, among other possible applications. The textured component  $v$  is completely represented using only two functions  $(g_1, g_2)$ . This is much simpler and much more efficient than the classical techniques for textures, such as wavelet decomposition, or more specifically Gabor transform. These two techniques use a large number of channels to represent a textured image.

Let  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  be a given image (we assume that the image initially defined on a rectangle in  $\mathbb{R}^2$ , has been extended by reflection to the entire space). We assume that  $f \in L^2(\mathbb{R}^2)$ . In real applications, the observed image  $f$  is just a noisy version of a true image  $u$ , or it is a textured image,



and  $u$  would be a simple sketchy approximation or a cartoon image of  $f$ . In the presence of additive noise, the relation between  $u$  and  $f$  can be expressed by the linear model, introducing another function  $v$ , such that

$$f(x, y) = u(x, y) + v(x, y).$$

In the Rudin-Osher-Fatemi restoration model [32],  $v$  represents noise or small scale repeated details, while  $u$  is an image formed by homogeneous regions, and with sharp edges. Given  $f$ , both  $u$  and  $v$  are unknown (if  $v$  is noise, we may know some statistics of  $v$ , such that it is of zero mean and given variance). In [32], the problem of reconstructing  $u$  from  $f$  is posed as a minimization problem in the space of functions of bounded variation  $BV(\mathbb{R}^2)$ , this space allowing for edges or discontinuities along curves. Their model, very efficient for denoising images while keeping sharp edges, is

$$\inf_{u \in L^2} F(u) = \int |\nabla u| + \lambda \int |f - u|^2 dx dy. \quad (1.29)$$

Formally minimizing  $F(u)$  yields the associated Euler-Lagrange equation

$$u = f + \frac{1}{2\lambda} \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right).$$

In practice, to avoid division by zero, the curvature term  $\operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right)$  is approximated by  $\operatorname{div} \left( \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}} \right)$ , but this not computed explicitly. Only the component  $u$  is kept in the ROF model.

Note that the curvature of the level contours of the final processed image is proportional to the noise in this model.

Note that the above function  $v$  in the ROF model can be formally written as:  $v = \operatorname{div} g$ , where  $g = (g_1, g_2)$  and  $g_1 = -\frac{1}{2\lambda} \frac{u_x}{|\nabla u|}$ ,  $g_2 = -\frac{1}{2\lambda} \frac{u_y}{|\nabla u|}$ . We have that  $g_1(x, y) + g_2^2(x, y) = \frac{1}{2\lambda}$  for all  $(x, y)$ , so that  $\|\sqrt{g_1^2 + g_2^2}\|_{L^\infty} = \frac{1}{2\lambda}$  (later we will use the notation  $|g| = \sqrt{g_1^2 + g_2^2}$ ).

In [23], Yves Meyer proves that the ROF model will remove the texture, if  $\lambda$  is small enough. In order to extract both the  $u$  component in  $BV$  and the  $v$  component as an oscillating function (texture or noise) from  $f$ , Meyer [23] proposes the use of a space of functions, which is in some sense the dual of the  $BV$  space. He introduces the following definition.

**Definition 1.** Let  $G$  denote the Banach space consisting of all generalized functions  $f(x)$  which can be written as

$$f(x) = \partial_x g_1(x, y) + \partial_y g_2(x, y), \quad g_1, g_2 \in L^\infty(\mathbb{R}^2), \quad (1.30)$$

induced by the norm  $\|f\|_*$  defined as the lower bound of all  $L^\infty$  norms of the functions  $|g|$  where  $g = (g_1, g_2)$ ,  $|g(x, y)| = \sqrt{g_1(x, y)^2 + g_2(x, y)^2}$  and where the infimum is computed over all decompositions (1.4) of  $f$ .

Meyer proposes a new image restoration model

$$\inf_u E(u) = \{|\nabla\varphi| + \lambda\|v\|_*, f = u + v\}. \quad (1.31)$$

A procedure for computing the minimum of this functional based on the Euler-Lagrange equation is not easily obtained. Instead in [43] we proposed a variant of this model.

We are motivated by the following approximation to the  $L^\infty$  norm of  $|g| = \sqrt{g_1^2 + g_2^2}$ , for  $g_1, g_2 \in L^\infty(\mathbb{R}^2)$ :

$$\left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^\infty} = \lim_{p \rightarrow \infty} \left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^p}.$$

Then, we propose the following minimization problem, inspired by (1.31):

$$\begin{aligned} \inf_{u, g_1, g_2} \{G_p(u, g_1, g_2) &= \int |\nabla u| + \lambda \int |f - u - \partial_x g_1 - \partial_y g_2|^2 dx dy \\ &+ \mu \left[ \int \left( \sqrt{g_1^2 + g_2^2} \right)^p dx dy \right]^{\frac{1}{p}} \}, \end{aligned} \quad (1.32)$$

where  $\lambda, \mu > 0$  are tuning parameters, and  $p \rightarrow \infty$ .

The first term insures that  $u \in BV(\mathbb{R}^2)$ , the second term insures that  $f \approx u + \text{div } g$ . Clearly, if  $\lambda \rightarrow \infty$  and  $p \rightarrow \infty$ , this model is formally an approximation of the model (1.31) originally proposed by Y. Meyer.

Formally minimizing the above energy with respect to  $u_1, g_1, g_2$ , yields the following Euler-Lagrange equations:

$$u = f - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \text{div} \left( \frac{\nabla u}{|\nabla u|} \right), \quad (1.33)$$

$$\begin{aligned} \mu \left( \|\sqrt{g_1^2 + g_2^2}\|_p \right)^{1-p} \left( \sqrt{g_1^2 + g_2^2} \right)^{p-2} g_1 \\ = 2\lambda \left[ \frac{\partial}{\partial x} (u - f) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right], \end{aligned} \quad (1.34)$$

$$\begin{aligned} \mu \left( \|\sqrt{g_1^2 + g_2^2}\|_p \right)^{1-p} g_2 \\ = 2\lambda \left[ \frac{\partial}{\partial y} (u - f) + \frac{\partial^2}{\partial_x \partial_y} g_1 + \frac{\partial^2}{\partial_y^2} g_2 \right]. \end{aligned} \quad (1.35)$$

In our numerical calculations, we have tested the model for different values of  $p$ , with  $1 \leq p \leq 10$ . The obtained results are very similar. The case  $p = 1$  yields faster calculations per iteration, so we give here the form of the Euler-Lagrange equations in this case ( $p = 1$ ):

$$u = f - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \text{div} \left( \frac{\nabla u}{|\nabla u|} \right), \quad (1.36)$$

$$\mu \frac{g_1}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[ \frac{\partial}{\partial x}(u - f) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right], \quad (1.37)$$

$$\mu \frac{g_2}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[ \frac{\partial}{\partial y}(u - f) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right]. \quad (1.38)$$

If the domain is finite, with exterior normal to the boundary denoted by  $(n_x, n_y)$ , the associated boundary conditions are:

$$\begin{aligned} \frac{\nabla u}{|\nabla u|}(n_x, n_y) &= 0, \\ (f - u - \partial_x g_1 - \partial_y g_2)n_x &= 0, \\ (f - u - \partial_x g_1 - \partial_y g_2)n_y &= 0. \end{aligned}$$

We have seen in the numerical results of [43], the proposed minimization model (1.32) allows to extract from a given real textured image  $f$  the components  $u$  and  $v$ , such that  $u$  is a sketchy (cartoon) approximation of  $f$ , and  $v = \operatorname{div}(g_1, g_2)$  represents the texture or the noise. In addition, the minimizer obtained for  $g = (g_1, g_2)$  allows us to discriminate two textures, by looking at the functions  $|g| = \sqrt{g_1^2 + g_2^2}$ ,  $|g_1|$  or  $|g_2|$ .

Our model is of the form

$$f = u + v + w$$

where  $w$  is the residual after the TV and texture parts are removed. According to equation (1.33) it is once again proportional to the level sets of  $u$ .

See [43] for details, and analytic justification and very promising numerical results.

## 5 Conclusion

The level set method and related PDE based techniques have helped create a paradigm shift in imaging science. Older books in image processing had the word “digital” in their title. Now it appears that imaging science has developed an important “continuum” point of view. Of course this shift began before the invention of the level set method but it is fair to say that the method and the associated numerical technology have greatly impacted research in this area.

## 6 REFERENCES

- [1] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, 118:269–277, 1995.

- [2] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.
- [3] M. Bardi and S. Osher. The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations. *SIAM Anal.*, 22(2):344–351, 1991.
- [4] G. Barles. *Solutions de Viscosité des Equations de Hamilton-Jacobi*. Springer-Verlag, Berlin, 1996.
- [5] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro. Variational problems and PDE's on implicit surfaces. the framework and examples in image processing and pattern formation. *J. Comput. Phys.*, 174:759–780, 2001.
- [6] M. Bertalmio, G. Sapiro, L.-T. Cheng, and S. Osher. Variational problems and PDE's on implicit surfaces. In *1st IEEE Workshop on Variational and LSM in Computer Vision, ICCV Vancouver*, pages 186–193.
- [7] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM J. Numer. Anal.*, 36(3):667–695, 1999.
- [8] P. Burchard, L.-T. Cheng, B. Merriman, and S. Osher. Motion of curves in three spatial dimensions using a level set approach. *J. Comput. Phys.*, 170:720–741, 2001.
- [9] T. Chan and L. Vese. An active contour model without edges, scale-space theories in computer vision. *Lec. Notes in Comput. Sci.*, 1682:141–151, 1999.
- [10] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. in *IEEE/Comput. Soc. Proc. of the 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 161–168, 2001.
- [11] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving Stefan problems. *J. Comput. Phys.*, 135:8–29, 1997.
- [12] L.-T. Cheng, P. Burchard, B. Merriman, and S. Osher. Motion of curves and constrained on surfaces using a level set approach. *J. Comput. Phys.*, 175:604–644, 2002.
- [13] M. Crandall, L.C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton Jacobi equations. *Trans. Amer. Math Soc.*, 282:485–502, 1984.

- [14] M. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277:1–42, 1983.
- [15] M.G. Crandall, H. Ishii, and P.-L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Am. Math. Soc. Bull.*, 27:1–67, 1992.
- [16] F. Durand. *3D Visibility: Analysis, Study and Application*. PhD thesis, MIT, 1999. PhD Thesis.
- [17] L.C. Evans, H.M. Soner, and P.E. Souganidis. Phase transitions and generalized motion by mean curvature. *Commun. Pure Appl. Math*, 65:1097, 1992.
- [18] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys*, 152:457–492, 1999.
- [19] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes iii. *J. Comput. Phys*, 71:231–303, 1987.
- [20] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.
- [21] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys*, 126:202–228, 1996.
- [22] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys*, 126:202–212, 1996.
- [23] Y. Meyer. Oscillating patterns in image processing and nonlinear evolution equations. *University Lecture Series*, 22, AMS, 2002.
- [24] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure. Appl. Math.*, 42:577–685, 1989.
- [25] S. Osher, L.-T. Cheng, M. Kang, H. Shim, and Y.-H. Tsai. Geometric optics in a phase space and eulerian framework. *J. Comput. Physics*, LSS Inc. #LS-01-01, 2001. to appear.
- [26] S. Osher and M. Kang. Private communication.
- [27] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput Physics*, 79:12–49, 1988.

- [28] S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *J. Comput Physics*, 28:902–921, 1991.
- [29] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155:410–438, 1999.
- [30] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.*, 29:867–884, 1992.
- [31] L. Rudin and S. Osher. Total variational based image restoration with free local constraints. In *Proc. IEEE Int. Conf. Image Proc. Austin, TX*, pages 31–35. IEEE Press, Piscataway, NJ, 1994.
- [32] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal. *Physica D*, 60:259–268, 1992.
- [33] J. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1999.
- [34] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [35] C.W. Shu and S. Osher. Implementation of essentially non-oscillatory shock capturing schemes ii. *J. Comput. Phys.*, 83:32–78, 1989.
- [36] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- [37] R. Tsai, P. Burchard, L.-T. Cheng, S. Osher, and G. Sapiro. Dynamic visibility in an implicit framework. *UCLA CAM Report*, 02-06, 2002. submitted to Trans. of ACM.
- [38] Y.R. Tsai. Rapid and accurate computation of the distance function using grids. *J. Comput. Phys.*, 178:175–195, 2002.
- [39] Y.R. Tsai, L.-T. Cheng, S. Osher, and H.K. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *UCLA CAM Report*, 01-27, 2001. submitted to SINUM.
- [40] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories, December 1994. Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, FL.
- [41] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.

- [42] L. Vese and S. Osher. The level set method links active contours, Mumford-Shah segmentation and total variation restoration. *UCLA CAM Report*, 02-05, 2002.
- [43] L. Vese and S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *UCLA CAM Report*, 02-19, 2002. to appear, *J. Scientific Computing*.
- [44] R. Whitaker. A level set approach to 3d reconstruction from range data. *Int. J. of Comput. Vision*, 1997.
- [45] H. Zhao. Fast sweeping method for eikonal equations I: Distance function. *submitted to SINUM*, 2002.
- [46] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.
- [47] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method, 2001. 1st IEEE Workshop on Variational and Level Set Methods.
- [48] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method. *Comput. Vision and Image Understanding*, 80:295–314, 2000.