Proc. Natl. Acad. Sci. USA
Vol. 92, pp. 7046–7050, July 1995
Applied Mathematics

Image processing via level set curvature flow

(image enhancement/image smoothing/geometric heat equation)

R. MALLADI AND J. A. SETHIAN

Lawrence Berkeley Laboratory and Department of Mathematics, University of California, Berkeley, CA 94720

Communicated by Alexandre J. Chorin, University of California, Berkeley, CA, February 27, 1995

ABSTRACT **We present a controlled image smoothing and enhancement method based on a curvature flow interpretation of the geometric heat equation. Compared to existing techniques, the model has several distinct advantages. (*i*) It contains just one enhancement parameter. (*ii*) The scheme naturally inherits a stopping criterion from the image; continued application of the scheme produces no further change. (*iii*) The method is one of the fastest possible schemes based on a curvature-controlled approach.**

The essential idea in image smoothing is to filter random noise present in the image signal without sacrificing the useful detail. In contrast, image enhancement focuses on preferentially highlighting certain image features. Together, they are precursors to many low-level vision procedures such as edge finding (1, 2), shape segmentation, and shape representation (3–6). In this paper, we present a method for image smoothing and enhancement that is a variant of the geometric heat equation that has several key advantages. (*i*) It contains just one enhancement parameter. (*ii*) The scheme naturally inherits a stopping criterion from the image; continued application of the scheme produces no further change. (*iii*) The method is one of the fastest possible schemes based on a curvature-controlled approach. The scheme is implemented using the level set curvature flow mechanism of Osher and Sethian (7, 8), which grew out of the earlier work in Sethian (9).

Traditionally, both one-dimensional and two-dimensional signals are smoothed by convolving them with a Gaussian kernel; the degree of blurring is controlled by the characteristic width of the Gaussian filter. Since the Gaussian kernel is an isotropic operator, it smooths across the region boundaries, thereby compromising their spatial position. As an alternative, Perona and Malik (10) have used an anisotropic diffusion process that performs intraregion smoothing in preference to interregion smoothing. A significant advancement was made by Alvarez, Lions, and Morel (ALM) (11), who presented a comprehensive model for image smoothing that includes the other models as special cases.

The ALM model consists of solving an equation of the form

$$I_t = g(|\nabla G * I|)\kappa|\nabla I|, \quad \text{with } I(x, y, t = 0) = I_0(x, y), \qquad [1]$$

where $G * I$ denotes the image convolved with a Gaussian filter. The geometric interpretation of the above diffusion equation is that the isointensity contours of the image move with speed $g(|\nabla G * I|)\kappa$, where $\kappa = \text{div } \nabla I/|\nabla I|$ is the local curvature. One variation of this scheme comes from replacing the curvature term with its affine invariant version [see Sapiro and Tannenbaum (12)]. By flowing the isointensity contours normal to themselves, smoothing is performed perpendicular to edges, thereby retaining edge definition. At the core of both numerical techniques is the Osher–Sethian level set algorithm for flowing the isointensity contours; this technique was also used in related work by Osher and Rudin (13).

In this work, we return to the original curvature flow equation and level set algorithm and build a numerical scheme for image enhancement based on an automatic switch function that controls the motion of the level sets in the following way. Diffusion is controlled by flowing under max(κ, 0) and min (κ, 0). The selection between these two types of flows is based on local gradient and curvature. The resulting technique is an automatic, extremely robust, computationally efficient, and straightforward scheme.

To motivate this approach, we begin by discussing curvature motion, namely,

$$I_t = F(\kappa)|\nabla I|. \qquad [2]$$

We then develop the complete model that includes image enhancement as well.

Curvature Flow

We first consider the problem of moving a closed nonintersecting curve in two dimensions along its gradient field with curvature-dependent speed. Following the arguments in Osher and Sethian (7), this curve can be embedded as the zero level set of a higher dimensional function ψ and its motion can be approximated by solving an equivalent equation of motion written for the function ψ. Specifically, the governing equation is given by

$$\psi_t = F(\kappa)|\nabla\psi|, \qquad [3]$$

where κ is the curvature and the function $\psi(x, y, t = 0)$ is set to the signed-distance function computed from the initial curve; distances are negative inside the curve and positive outside.

As an example, consider the curve motion in Fig. 1. We show the motion of the same initial curve under the influence of different speed functions. In Fig. 1*A* the speed $F = \kappa$ and the curve position is plotted after every 0.001 sec. We continue this in Fig. 1*B*, where the curve position is displayed once every 0.0025 sec. Note that under plain curvature motion, the curve both smooths and shrinks and, hence, eventually disappears. Our goal is a diffusion process that retains the curve. One way to accomplish this is to use geometric smoothing without shrinkage, an approach described by Sapiro and Tannenbaum (14). They present schemes that perform area- and length-preserving smoothing of plane curves.

Consider now the modifications to straight curvature flow. Begin with a speed function $F = \max(\kappa, 0)$, which has been used in other applications such as grid generation (15) and shape recognition (6). This flow shrinks the curve as shown in Fig. 1*C*. On the other hand, consider the flow $F = \min(\kappa, 0)$. This motion diffuses the curve until it approaches the convex hull of the enclosed points (see Fig. 1*D*); the motion then halts. Note that if the signs in the signed-distance function are reversed (see Eq. 3), the curvature at every point also changes

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

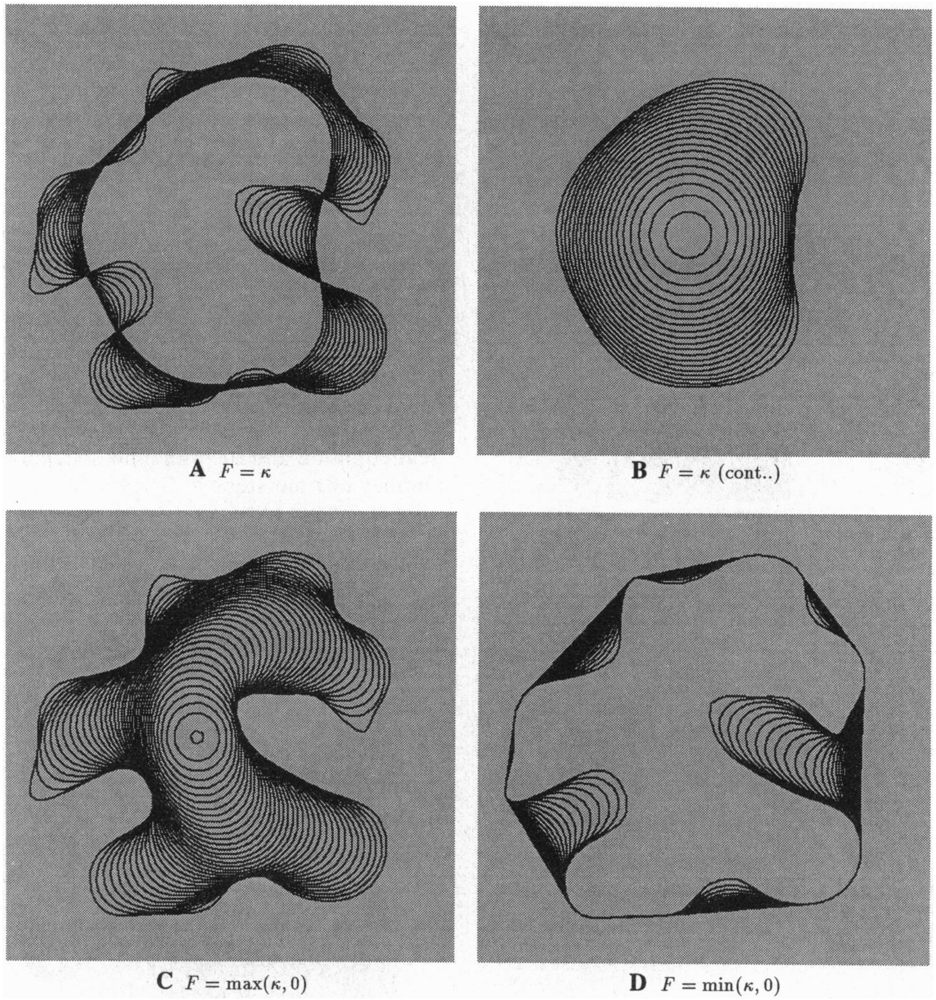

FIG. 1. Motion of a curve with curvature-dependent speed. (*A* and *B*) The initial curve moving under its curvature. (*C*) Its movement under max(κ, 0.0). (*D*) If a curve moves under min(κ, 0.0), diffusion stops at the convex hull of the initial curve.

sign, thereby reversing the effects produced by max(κ, 0) and min (κ, 0) flow rules.

We can now apply these flows to the context of image smoothing. We solve the following anisotropic diffusion equation:

$$I_t = F(\kappa)|\nabla I|. \quad [4]$$

As mentioned previously, the geometric interpretation of this equation is the movement of isointensity contours in the normal direction with speed $F(\kappa)$. If diffusion is carried out with plain curvature motion—i.e., $F = \kappa$—then various isointensity contours shrink with time and disappear. This causes considerable loss in the image intensity detail.

Our approach is to use a combination of speed max(κ, 0) and min(κ, 0) in Eq. **4** to control the diffusion process. The idea is to select between the two types of flows based on local image properties. Specifically, let the speed be defined as:

$$F(\kappa) = \begin{cases} \max(\kappa, 0) & \text{if } a(x, y) < G(x, y) \\ \min(\kappa, 0) & \text{otherwise,} \end{cases} \quad [5]$$

where $a(x, y)$ is the average value of image intensity $I(x, y)$ in a small neighborhood around the point (x, y), and $G(x, y)$ is defined as the average intensity evaluated in the direction perpendicular to the gradient direction. Note that since the direction perpendicular to the gradient is the tangent direction to the isointensity contour through (x, y), the two points used to compute $G(x, y)$ lie on the same side of the region enclosed by it unless the point is an inflection point and the curvature vanishes. Choosing the speed according to the above equation controls the diffusion process. In addition, if we set a threshold on the local gradient magnitude by

$$F(\kappa) = \begin{cases} \kappa & \text{if } |\nabla I| < T \\ \text{min/max flow} & \text{otherwise,} \end{cases} \quad [6]$$

where T is some threshold value, enhancement of selected regions is performed. Thus, points at which the gradient magnitude is greater than T are preferred and are diffused using the min/max flow (Eq. **5**) and the remaining points are diffused using the plain curvature flow.

Experimental Results

In this section, we apply our scheme to smooth some synthesized and real images. First, consider the image shown in Fig. 2*A*, which consists of 16 cells, each with a shape drawn with a random intensity against a random intensity background. The image in Fig. 2*B* shows the original image diffused using isotropic heat equation $I_t = \nabla I$, which is the same as smoothing with a Gaussian kernel. We then apply our scheme to the image. The result of running our scheme to a comparable "scale," which in this case amounts to solving the equations for the same number of time steps, is shown in Fig. 2*C*. The original image is noiseless and consists of very clearly defined regions. Our scheme smooths some sharp corners and stops,

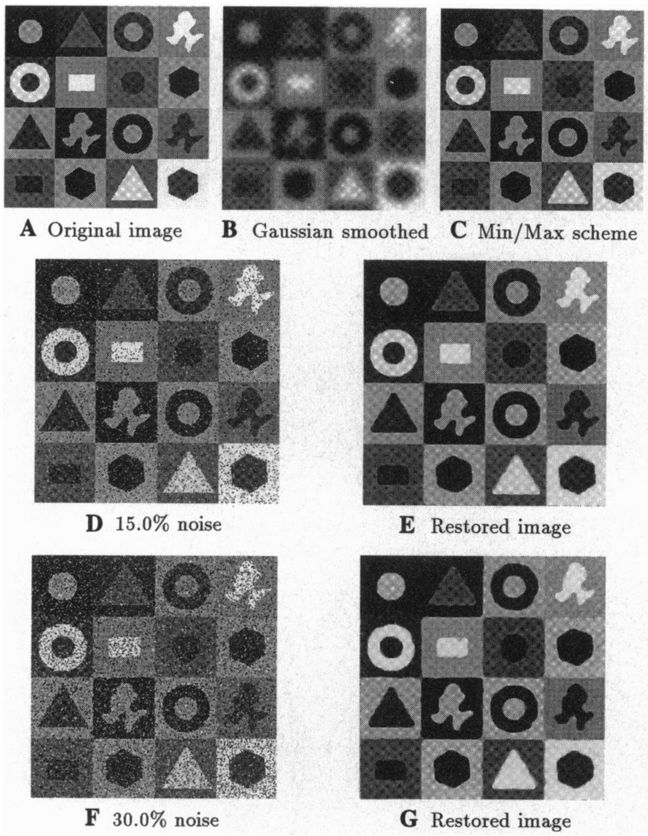

FIG. 2. min/max scheme as a method for image restoration. (*B* and *C*) Results of smoothing the original image (*A*) using the Gaussian filter and our scheme. (*D* and *F*) Original image corrupted with 15% and 30% noise. (*E* and *G*) The corresponding restored images obtained by our scheme.

thereby retaining all the detail. Note that we could run the scheme for another 100 time steps without any change. On the other hand, in Figs. 2*D*, 2*F*, and 3*A* we add different amounts of noise to the image. This is done by randomly choosing an image location and setting the intensity value at that location to a random value in the range [0..255]. These corrupted images can be restored by using our min/max scheme as shown in Figs. 2*E*, 2*G*, and 3*B*. These images have been produced by running our scheme for 100, 200, and 350 time steps with an enhancement factor of 1.0, 5.0, and 7.5, respectively. The image size is 256 × 256. We would like to emphasize that our scheme eliminates noise without sacrificing edge definition.

For some images, it is possible to exploit the prior knowledge of the foreground and background intensity to globally fix the value of the function $G(x, y)$ instead of computing it locally. One such example is the set of images in Fig. 4. Here the original image consists of black characters written on white background. The left column shows the original image corrupted with different amounts of noise and the corresponding restored images are shown in the right column. In all cases the min/max scheme has been used with the G function set to 128, which is the average of black (0) and white (255) pixel values.

Next, we apply our scheme to some real images. First, consider the noisy digital subtraction angiogram in Fig. 5*A*. Different levels of enhancement can be achieved by changing the value of T in Eq. **6**. Fig. 5 *B–F* show the results of solving the min/max scheme for 100 time steps but with different values of T. In the next set of figures we compare our image smoothing scheme with that of the Gaussian filtering method. In order to make the comparison, the images in Fig. 6 have been generated by solving the isotropic heat equation and the min/max scheme for exactly the same number of time steps.

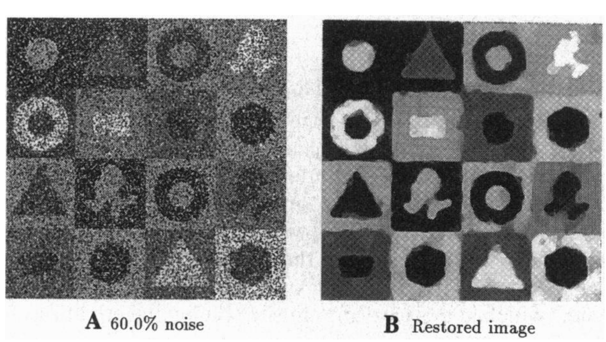

FIG. 3. Image restoration continued. (*A*) Image corrupted with 60% noise. (*B*) Restored image.

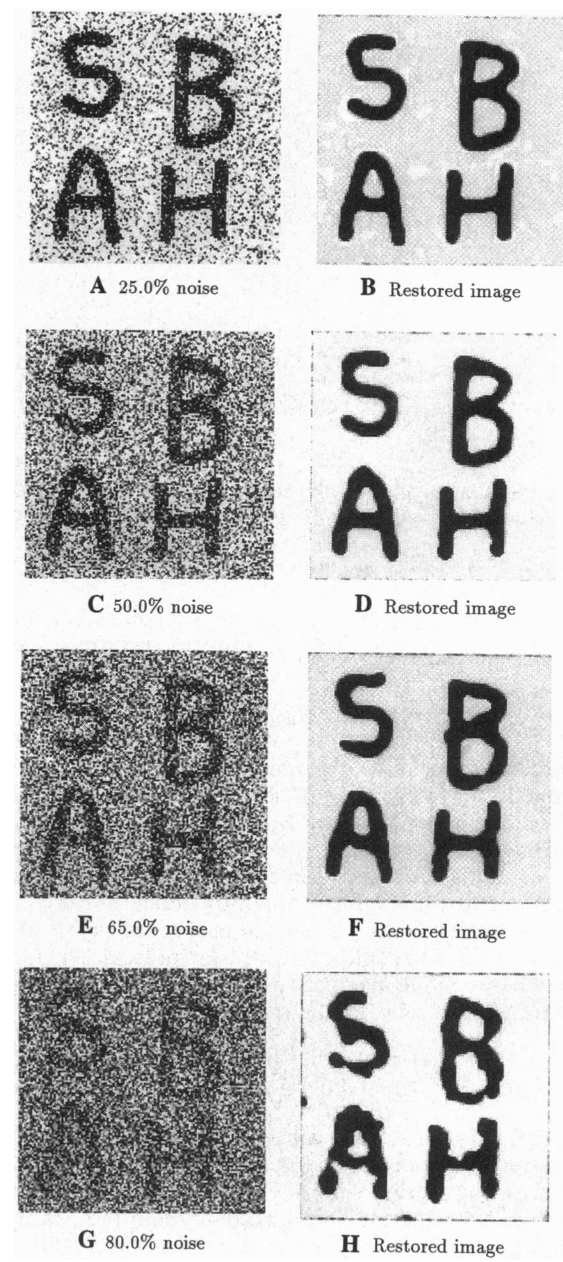

FIG. 4. Image restoration by setting $G(x, y)$ value globally.

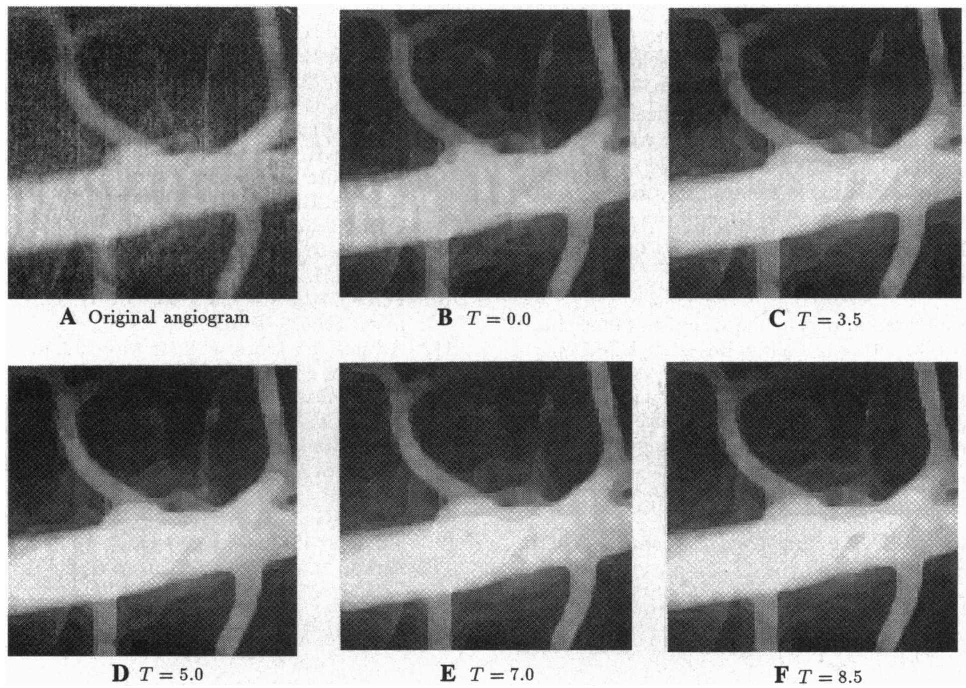

FIG. 5. Results of applying the min/max scheme with different values of enhancement threshold T to a digital subtraction angiogram.

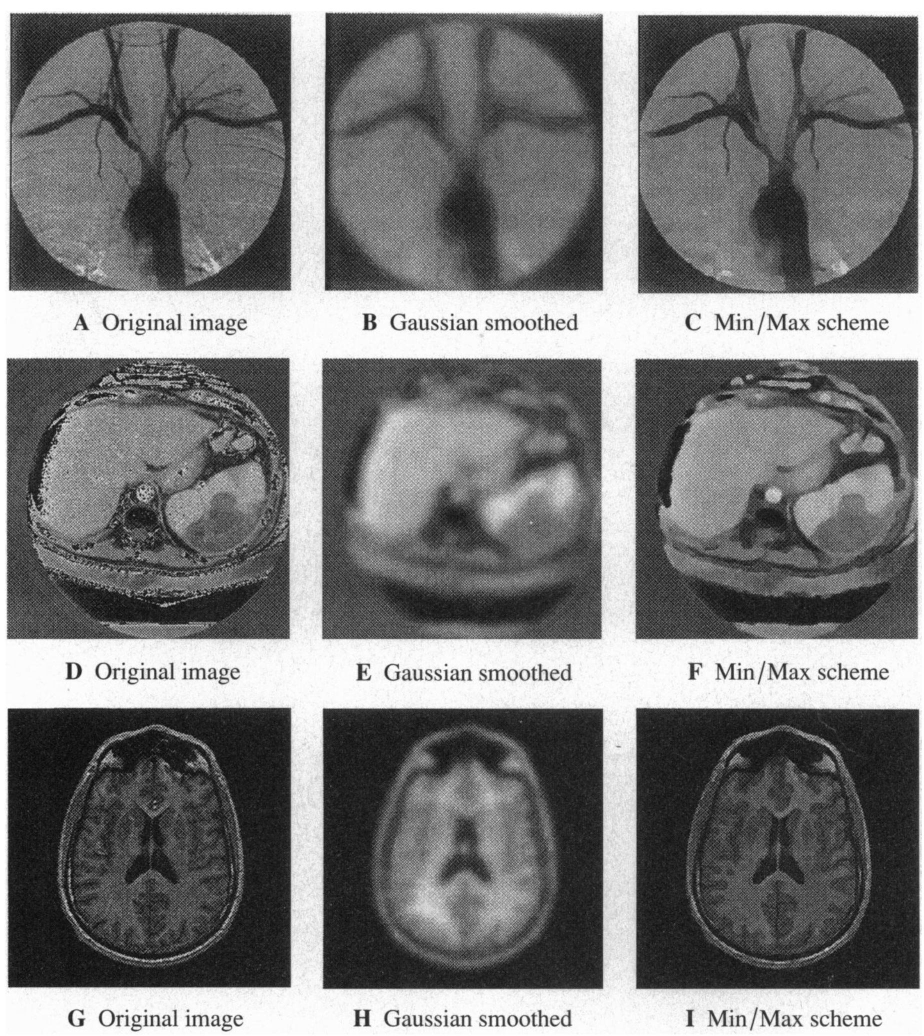

FIG. 6. Comparison of min/max scheme for image smoothing with that of Gaussian smoothing.

Conclusion

In this paper, we present an image-processing scheme that is based on a variant of the geometric heat equation. The scheme relies on applying appropriate speed laws on the isointensity contours of an image. The scheme is automatic with just one enhancement parameter and does not require a stopping criterion. The scheme is extremely fast. The curvature κ in Eqs. **5** and **6** can be replaced by $\kappa^{1/3}$, its affine invariant version (12). An extension to color image enhancement is straightforward (to be presented elsewhere).

This research was supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract DE-AC03-76SD00098 and by the National Science Foundation Advanced Research Projects Agency under Grant DMS-8919074. R.M. is a National Science Foundation Postdoctoral Fellow in Computational Science and Engineering.

1. Marr, D. & Hildreth, E. (1980) *Proc. R. Soc. London B* **207**, 187–217.
2. Canny, J. (1986) *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-8**, 679–698.
3. Kass, M., Witkin, A. & Terzopoulos, D. (1988) *Int. J. Comput. Vision*, 321–331.
4. Malladi, R., Sethian, J. A. & Vemuri, B. C. (1994) *Lect. Notes Comput. Sci.* **800,** 3–13.
5. Malladi, R., Sethian, J. A. & Vemuri, B. C. (1995) *IEEE Trans. Pattern Anal. Machine Intell.* **17**, 158–175.
6. Malladi, R. & Sethian, J. A. (1994) *CVGIP: Image Understanding*, (Lawrence Berkeley Laboratory, Univ. of California, Berkeley), Report LBL-36069.
7. Osher, S. & Sethian, J. A. 91988) *J. Comput. Phys.* **79**, 12–49.
8. Sethian, J. A. (1990) *J. Differential Geom.* **31**, 131–161.
9. Sethian, J. A. (1985) *Commun. Math. Phys.* **101**, 487–499.
10. Perona, P. & Malik, J. (1990) *IEEE Trans. Pattern Anal. Machine Intell.* **12**, 629–639.
11. Alvarez, L., Lions, P. L. & Morel, J. M. (1992) *SIAM J. Num. Anal.* **29**, 845–866.
12. Sapiro, G. & Tannenbaum, A. (1993) *Proceedings of the Conference on Information Sciences and Systems* (Johns Hopkins University, Baltimore).
13. Osher, S. & Rudin, L. I. (1990) *SIAM J. Num. Anal.* **27**, 919–940.
14. Sapiro, G. & Tannenbaum, A. (1994) *Lect. Notes Comput. Sci.* **801,** 449–458.
15. Sethian, J. A. (1994) *J. Comput. Phys.* **115**, 440–454.