Image Processing: Flows under Min/Max Curvature and Mean Curvature*†

R. Malladi‡ and J. A. Sethian

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

December 17, 1999

Abstract

We present a class of PDE-based algorithms suitable for a wide range of image processing applications. The techniques are applicable to both salt-and-pepper grey-scale noise and full-image continuous noise present in black and white images, grey-scale images, texture images and color images. At the core, the techniques rely on a level set formulation of evolving curves and surfaces and the viscosity in profile evolution. Compared to existing techniques, our approach has several distinct advantages. First, it contains only one enhancement parameter, which in most cases is automatically chosen. Second, the scheme automatically stops smoothing at some optimal point; continued application of the scheme produces no further change. Third, the method is one of the fastest possible schemes based on a curvature-controlled approach.

1 Introduction

The essential idea in image smoothing is to filter noise present in the image signal without sacrificing the useful detail. In contrast, image enhancement focuses on preferentially highlighting certain image features. Together, they

*Correct citation: Malladi, R., and Sethian, J.A., Graphical Models and Image Processing, 58 , 2, pp. 127-141, 1996.

†Supported in part by the Applied Mathematics Subprogram of the Office of Energy Research under DE-AC03-76SF00098, and the National Science Foundation DARPA under grant DMS-8919074.

‡NSF Postdoctoral Fellow in Computational Science and Engineering

are precursors to many low level vision procedures such as edge finding [15, 5], shape segmentation, and shape representation [13, 14, 11]. In this paper, we present a method for image smoothing and enhancement which is a variant of the geometric heat equation which has several key advantages. First, it contains only one enhancement parameter, which it must cases is automatically chosen. Second, the scheme automatically stops smoothing at some optimal point; continued application of the scheme produces no further change. Third, the method is one of the fastest possible schemes based on a curvature-controlled approach.

The methods presented in this paper are derived from the Osher-Sethian [17] level set formulation of front propagation, which grew out of earlier by Sethian [23] on the mathematical formulation of curve and surface motion. The application of this level set perspective to image processing, and the design of a PDE-based approach to image enhancement and noise removal was introduced in two pivotal papers; the work of Alvarez, Lions and Morel [3] and the work of Osher-Rudin [16]. While the work presented here starts from the original curve evolution work [23] and level set formulation presented in [17, 24], it owes a significant debt to the ground-breaking work of Alvarez, Lions, and Morel and Osher-Rudin.

The fundamental idea in our approach is to return to the simplest possible problem, namely the evolution of a curve under its curvature. We design a technique in which the motion of this curve stops automatically at a desired point; this motion forms the core of our approach. Application of this scheme to both salt-and-pepper grey-scale noise and Gaussian noise removal in images is straightforward for binary, grey-scale, and color images. We then extend the technique to textured images, making use of both the curvature and the mean curvature of the underlying image when viewed as a graph.

The outline of this paper is as follows. First, in Section II, we give a very brief background. Next, in Section III, we study the motion of a curve moving under its curvature, and develop an automatic stopping criteria. In Section IV, we apply this technique to a binary images, and extend the technique to grey-scale images, textured images, and color images.

2 Background

Traditionally, both 1-D and 2-D 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 [18] have used an anisotropic diffusion process which performs intraregion smoothing in preference to inter-region smoothing. A significant advancement was made by Alvarez, Lions, and Morel (ALM) [3], who presented a comprehensive model for image smoothing which includes the other models as special cases.

The ALM model consists of solving an equation of the form

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

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 $\kappa = \operatorname{div}\frac{\nabla I}{|\nabla I|}$ is the local curvature. One variation of this scheme comes from replacing the curvature term with its affine invariant version (see Sapiro and Tannenbaum [20]). 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 Rudin, Osher and Fatemi [19].

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 a automatic switch function that controls the motion of the level sets in the following way. Diffusion is controlled by flowing under $\max(\kappa, 0)$ and $\min(\kappa, 0)$. The selection between these two types of flows is based on local intensity and gradient. The resulting technique is an automatic, extremely robust, computationally efficient, and straightforward scheme.

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

$$I_t = F(\kappa) \mid \nabla I \mid . \tag{2}$$

We then develop the complete model which includes image enhancement as well. The crucial ideas on min/max flows upon which this paper is based have been reported earlier by the authors in [10].

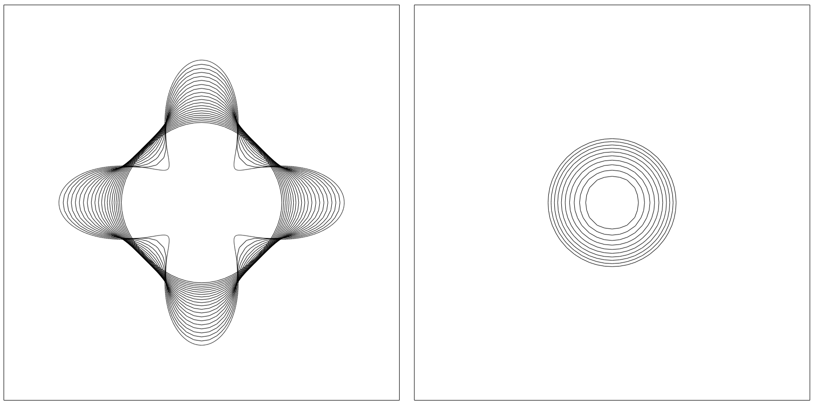
3 Motion of Curves under Curvature

3.1 Formulation

Consider a closed, nonintersecting curve in the plane moving with speed $F(\kappa)$ normal to itself. More precisely, Let $\gamma(0)$ be a smooth, closed initial curve in R^2, and let $\gamma(t)$ be the one-parameter family of curves generated by moving $\gamma(0)$ along its normal vector field with speed $F(\kappa)$. Here, $F(\kappa)$ is a given scalar function of the curvature κ. Thus, $n \cdot x_t = F(\kappa)$, where x is the position vector of the curve, t is time, and n is the unit normal to the curve.

Consider a speed function of the form $1 - \epsilon\kappa$, where ϵ is a constant. An evolution equation for the curvature κ, see [23], is given by

$$\kappa_t = \epsilon\kappa_{\alpha\alpha} + \epsilon\kappa^3 - \kappa^2 \tag{3}$$



$F = \kappa \quad (Begin)$ $\qquad$ $F = \kappa \quad (Cont.)$

Figure 1: Collapse of Star-Shaped Curve under Curvature

where we have taken the second derivative of the curvature κ with respect to arclength α. This is a reaction-diffusion equation; the drive toward singularities due to the reaction term $(\epsilon\kappa^3 - \kappa^2)$ is balanced by the smoothing effect of the diffusion term $(\epsilon\kappa_{\alpha\alpha})$. Indeed, with $\epsilon = 0$, we have a pure reaction equation $\kappa_t = -\kappa^2$. In this case, the solution is $\kappa(s,t) = \kappa(s,0)/(1 + t\kappa(s,0))$, which is singular in finite t if the initial curvature is anywhere negative. Thus, corners can form in the moving curve when $\epsilon = 0$.

For $\epsilon = 0$, the front develops a sharp corner in finite time as discussed above. In general, it is not clear how to construct the normal at the corner and continue the evolution, since the derivative is not defined there. One possibility is the "swallowtail" solution formed by letting the front pass through itself. However, from a geometrical argument it seems clear that the front at time t should consist of only the set of all points located a distance t from the initial curve. (This is known as the Huyghens principle construction, see [23]). Roughly speaking, we want to remove the "tail" from the "swallowtail". Another way to characterize this weak solution is through the following "entropy condition" posed by Sethian (see [23]): If the front is viewed as a burning flame, then *once a particle is burnt it stays burnt.* Careful adherence to this stipulation produces the Huyghens principle construction. Furthermore, this physically reasonable weak solution is the formal limit of the smooth solutions $\epsilon > 0$ as the curvature term vanishes, (see [23]). Extensive discussion of the role of shocks and rarefactions in propagating fronts may be found in [22].

Let us imagine now a very specific speed function, namely $F(\kappa) = -\kappa$. This case corresponds to a curve collapsing under its curvature. It can be shown that for an arbitrary smooth simple curve, (see Gage [8], Grayson [9]), such a curve collapses to a single point. In Figure 1a,b, we show a star-shaped region collapsing under this flow.

Here, we have evolved the curve using the Osher-Sethian level set method, see [17]. Briefly, this technique works as follows. Given a moving closed hypersurface $\Gamma(t)$, that is, $\Gamma(t=0) : [0, \infty) \to R^N$, we wish to produce an Eulerian formulation for the motion of the hypersurface propagating along its normal direction with speed F, where F can be a function of various arguments, including the curvature, normal direction, e.t.c. The main idea is to embed this propagating interface as the zero level set of a higher dimensional function ϕ. Let $\phi(x, t=0)$, where $x \in R^N$ is defined by

$$\phi(x, t=0) = \pm d \tag{4}$$

where d is the distance from x to $\Gamma(t=0)$, and the plus (minus) sign is chosen if the point x is outside (inside) the initial hypersurface $\Gamma(t=0)$. Thus, we have an initial function $\phi(x, t=0) : R^N \to R$ with the property that

$$\Gamma(t=0) = (x|\phi(x, t=0) = 0) \tag{5}$$

It can easily be shown that the equation of motion given by

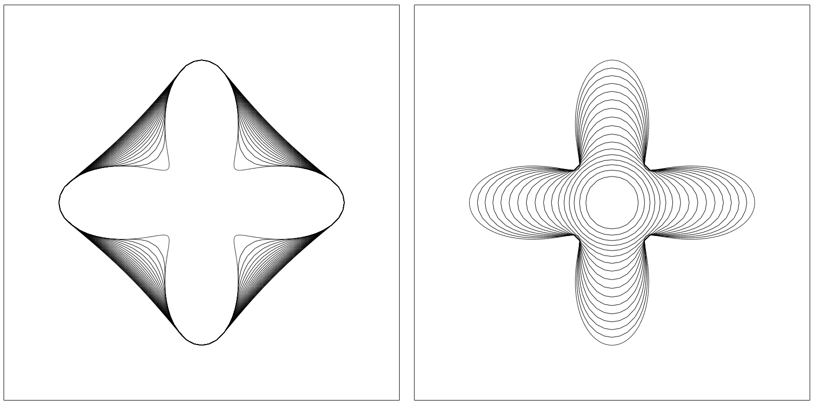
$$\phi_t + F|\nabla\phi| = 0 \tag{6}$$

$$\phi(x, t=0) \quad given \tag{7}$$

is such that the evolution of the zero level set of ϕ always corresponds to the motion of the initial hypersurface under the given speed function F. This evolution equation Eqn. 7 is solved by means of difference operators on a fixed Eulerian grid. Care must be taken in the case where the speed function F contains a hyperbolic component. For details, see [17, 24]. Since its introduction, this approach to front propagation has been used to model a wide variety of problems, including the generation of minimal surfaces [6], fast interface techniques [1], singularities and geodesics in moving curves and surfaces in [7], flame propagation [26, 27], shape reconstruction [13, 14, 12], shape representation and recognition [11], grid generation [25], and semiconductor manufacturing [2].

3.2 The Min/Max Flow

We now modify the above flow. In order to be careful about signs, we simply note that the boundary of a disk initialized so that the inside of the disk corresponds to a negative value for the signed distance function ϕ and a positive value for the signed distance function ϕ on the outside of the disk has a normal $\nabla\phi$ which points outwards away from the center of the disk, and a curvature defined as $\nabla \cdot \nabla\phi/|\nabla\phi|$ which is always positive on all the convex level contours. Thus, a flow under speed function $F = \kappa$ corresponds to the collapsing curvature flow, since the boundary moves in the direction of its normal with negative speed, and hence moves inwards.



$F = \min(\kappa, 0.0)$ $F = \max(\kappa, 0.0)$

Figure 2: Motion of Curve under Min/Max Flow

We need to be further careful about signs and amend a previous definition. We shall refer to a speed function F in the context of the level set equation

$$\phi_t = F|\nabla\phi| \tag{8}$$

thus, from now on, F will give the speed of the front in a direction opposite to its normal direction. Thus, a curve collapsing under its curvature will correspond to speed $F = \kappa$. This will be our convention for the remainder of this paper.

Now, motivated by work on level set methods applied to grid generation [25] and shape recognition [11], we consider two flows, namely

- $F(\kappa) = \min(\kappa, 0.0)$
- $F(\kappa) = \max(\kappa, 0.0)$

Here, we have chosen the negative of the signed distance in the interior, and the positive sign in the exterior region. As shown in Figure 2, the effect of flow under $F(\kappa) = \min(\kappa, 0.0)$ is allow the inward concave fingers to grow outwards, while suppressing the motion of the outward convex regions. Thus, the motion halts as soon as the convex hull is obtained. Conversely, the effect of flow under $F(\kappa) = \max(\kappa, 0.0)$ is to allow the outward regions to grow inwards while suppressing the motion of the inward concave regions. However, once the shape becomes fully convex, the curvature is always positive and hence the flow becomes the same as regular curvature flow; hence the shape collapses to a point.

We can summarize the above by saying that, for the above case, flow under $F = \min(\kappa, 0.0)$ preserves some of the structure of the curve, while flow under $F = \max(\kappa, 0.0)$ completely diffuses away all of the information.

Before proceeding, we examine a slightly more complex shape, which is instructive. In Figure 3, we show what happens to a double star-shaped region. We let the color black correspond to the "inside" where $\phi < 0$ and the white correspond to the "outside" where $\phi > 0$. First, in Figure 3a, we show evolution under plain curvature, that is $F = \kappa$. Eventually, the shape collapses completely. In Figure 3b, we show the same curve collapsing under $F = \min(\kappa, 0.0)$; here, the outer front moves to the convex hull, while the inner front collapses and disappears. The last shown state is stable. In Figure 3c, we show the same curve collapsing under $F = \max(\kappa, 0.0)$; here, the outer part of the front collapses, while the inner part expands to its convex hull. Eventually, the two meet, and the front disappears. Finally, in Figure 3d, we switch the roles of black and white; thus flow with speed $F = \max(\kappa, 0.0)$ corresponds to the same flow as in Figure 3d; changing the colors corresponds to changing between the maximum flow to the minimum flow.

3.3 The Goal

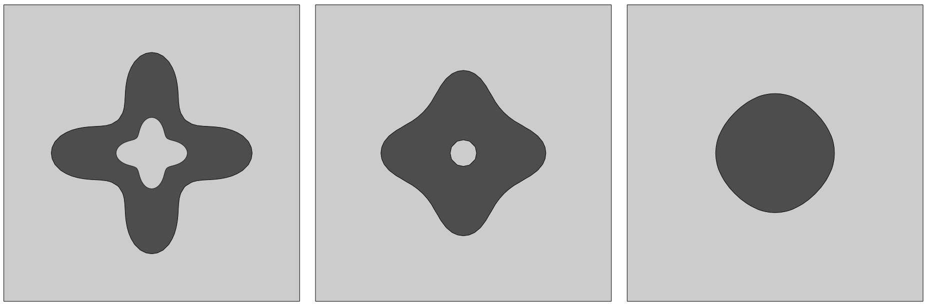
Consider now the square with notches on each side shown in Figure 4a. We imagine that the notches are one unit wide, where a unit most typically will correspond to a pixel width. Our goal is to use the above flow to somehow remove the notches which protrude out from the sides. In Figure 4b, we see the effect of curvature flow; the notches are removed, but the shape is fully diffused. In Figure 4c, we see the effect of flow with speed $F = \min(\kappa, 0.0)$; here, one set of notches are removed, but the other set have been replaced by their convex hull. If we run this flow forever, the figure will not change since the convex hull has been obtained, which does not move under this flow. Conversely, as shown in Figure 4d, obtained with speed $F = \max(\kappa, 0.0)$, the inner notches stay fixed and the front moves in around them, while the outer notches are diffused. Continual application of this flow causes the shape to shrink and collapse. Finally, in Figure 4e and Figure 4f, we reverse the roles of black and white, showing the effects of the min and max flows are now reversed.

The problem is that in some places, the notch is "outwards", and in others, the notch is "inwards". Our goal is a flow which somehow chooses the correct choice of flows between $F = \max(\kappa, 0.0)$ and $F = \min(\kappa, 0.0)$. The solution lies in a switch function which determines the nature of the notch.

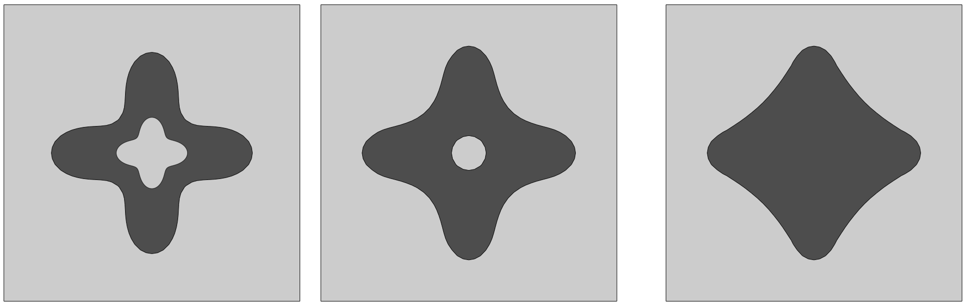
3.4 The Switch

In this section, we present the switch function to flow the above shape. Our choice is somewhat mysterious at first; rather than present the reasoning behind the switch, we shall first describe it, and then in the next section explain why it works.

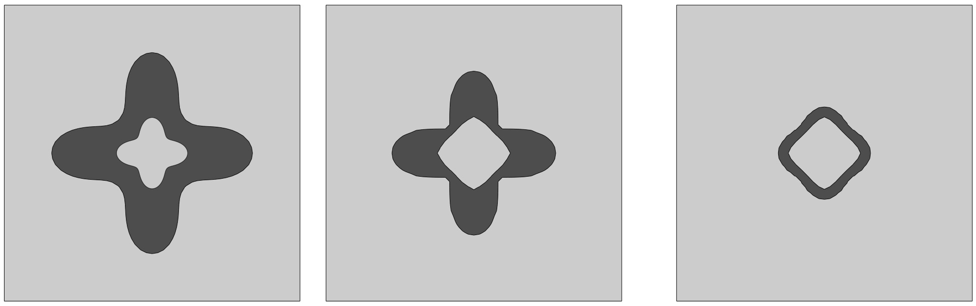
Our construction of a switch is motivated by the mean value theorem for harmonic functions, which states that the value of a harmonic function at any



$F = \kappa \quad (T = 0)$ $F = \kappa \quad (T = 1)$ $F = \kappa \quad (Continued)$



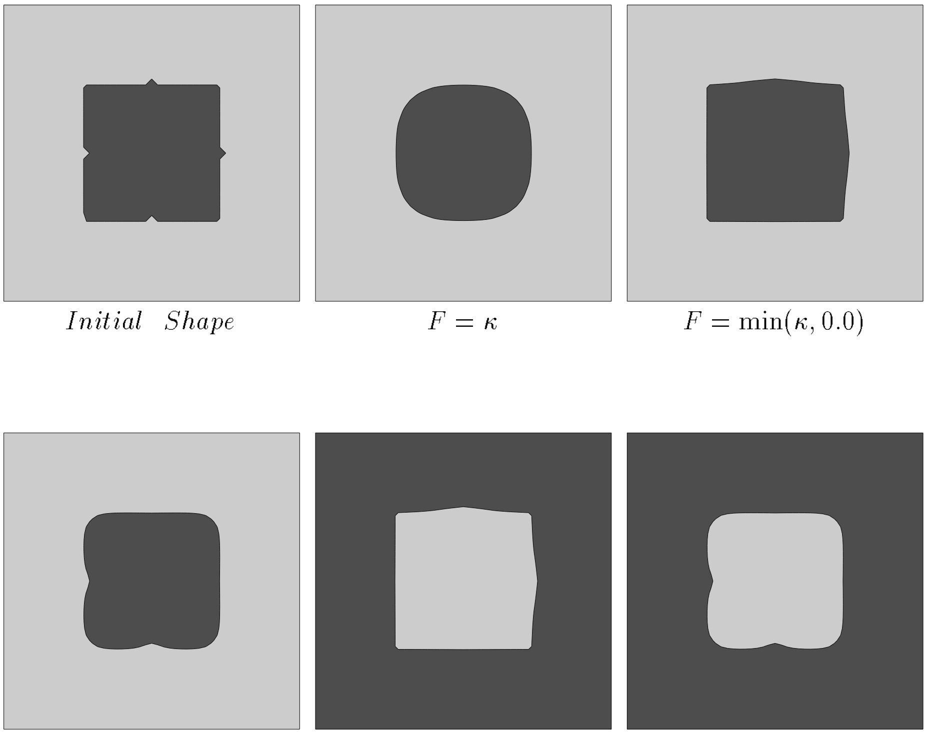
$F = \min(\kappa, 0.0) \quad (T = 0)$ $F = \min(\kappa, 0.0) \quad (T = 1)$ $F = \min(\kappa, 0.0) \quad (Continued)$



$F = \max(\kappa, 0.0) \quad (T = 0)$ $F = \max(\kappa, 0.0) \quad (T = 1)$ $F = \max(\kappa, 0.0) \quad (Continued)$

$F = \max(\kappa, 0.0) \quad (T = 0)$ $F = \max(\kappa, 0.0) \quad (T = 1)$ $F = \max(\kappa, 0.0) \quad (Continued)$

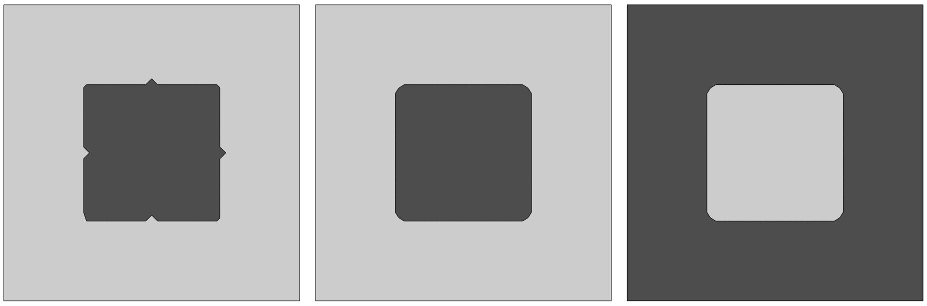
Figure 3: Motion of Complex Region under Various Flows



Initial Shape | $F = \kappa$ | $F = \min(\kappa, 0.0)$

$F = \max(\kappa, 0.0)$ | $F = \max(\kappa, 0.0)$ | $F = \min(\kappa, 0.0)$

Figure 4: Motion of Notched Region under Various Flows



$Initial \quad Shape$ $\qquad$ $F = F_{min/max}$ $\qquad$ $F = F_{min/max}$

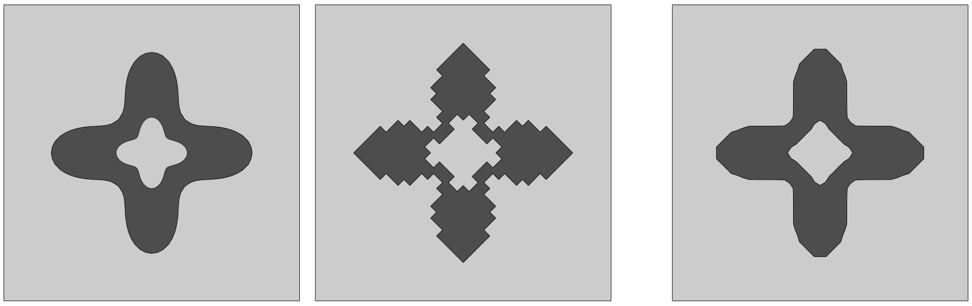
Figure 5: Motion of Notched Region under Min/Max Flow

point is equal to the average of its boundary values. Very roughly speaking, we might think of curvature flow as somewhat related to heading towards a harmonic function. Thus, imagine the simplest case, namely that of a black and white image, in which black is given the value $\phi = -1$ and white given the value $\phi = 1$. We select between the two flows based on the sign of the deviation from the mean value theorem. Define $Average(x, y)$ as the average value of the image intensity $I(x, y)$ in a square centered around the point (x, y) with sidelength $(2. * StencilWidth + 1)$, where, for now $StencilWidth = 0$. Then, at any point (x, y), define the flow by

$$F_{min/max} = \begin{cases} \min(\kappa, 0) & \text{if } Average(x, y) < 0 \\ \max(\kappa, 0) & \text{otherwise} \end{cases} \tag{9}$$

Here, we view 0 as the "threshold" value $T_{threshold}$; since it is halfway between the black value of -1 and the white value of 1. This flow can be seen to thus choose the "correct" flow between the min flow and the max flow. As demonstration, in Figure 5a, we show the initial notched region. In Figure 5b, we show the results using the min/max given in Eqn. 9. To verify that our scheme is independent of the positioning of the colors, we reverse the initial colors and show the results of the *same* min/max flow in Figure 5c. The small amount of rounding that is seen at the corners is due to the coarseness of the calculation; the simulation is performed on a 49×49 grid, and the contour plotter which locates the zero level set rounds the edges.

As a further test, we return to our double-starshaped region, and again run the min/max flow. We show the initial shape in Figure 6a. In Figure 6b, we add an oscillation on the size of the grid, that is, we alternately switch grid values along the boundary between the two regions. In Figure 6c, we then show the results of the min/max flow. What happens is that the small-scale "noise" is removed; once this happens, the boundary achieves a final state which does not change and preserves structures larger than the one-pixel wide noise. Thus, the large arms are *not* destroyed by the flow. If we were to exchange the roles of



Initial Shape — *Noise Added* — $F = F_{min/max}$ *Final* $(T = \infty)$

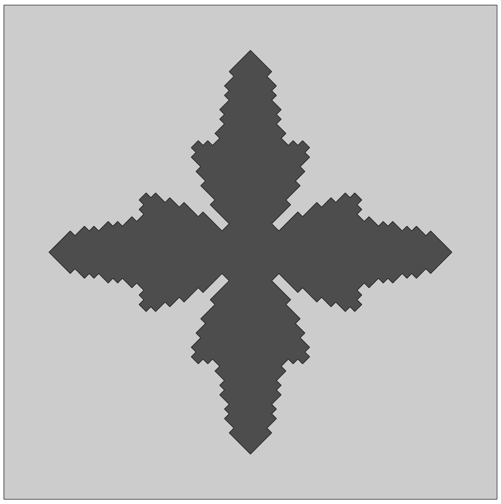
Figure 6: Motion of Double-StarShaped Region under Min/Max Flow

black and white, the same final shape would be obtained.

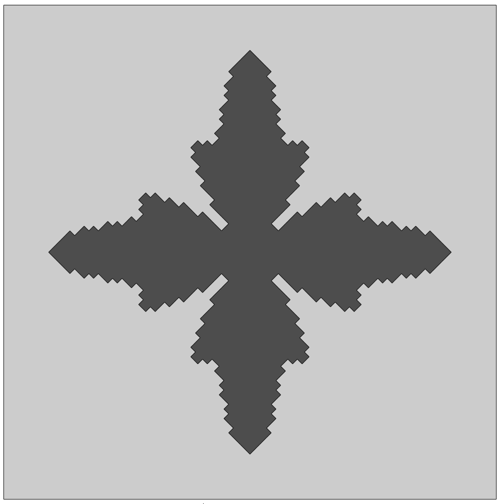
We note that the level of noise removed is a function of the size of the stencil used in computing the switch in the min/max speed. What remains are structures than are not detected by our threshold stencil. We can choose to remove the next *larger* level of noise structures by increasing the size of our threshold stencil; we compute the average $Average(x, y)$ over a large square. We then use this larger stencil and continue the process by running the min/max flow. We have done this in Figure 7; we start with an initial shape in Figure 7a which has "noise" in the boundary. We then perform the min/max flow until steady-state is achieved with stencil size zero in Figure 7b; that is, the "average" consists only of the value of ϕ at the point (x, y) itself. We note that when we choose a stencil size of zero, nothing happens; this is explained in detail in the next section. In Figure 7c, we perform the min/max flow until steady-state is achieved with stencil size of 1, and the continue min/max flow with a larger stencil until steady-state is again achieved in Figure 7d. As the stencil size is increased, larger and larger structures are removed.

We can summarize our results as follows:

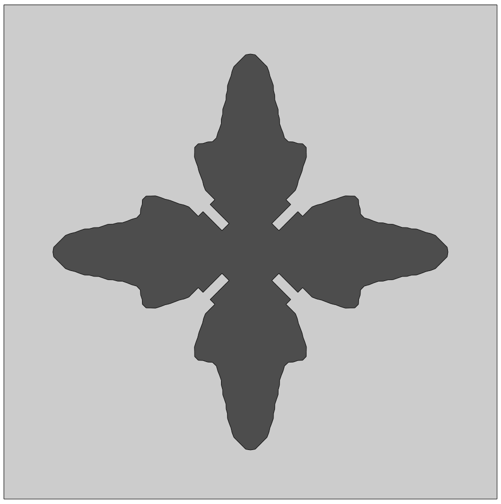
1. The single min/max flow selects the correct motion to diffuse the small-scale pixel notches into the boundary.
2. The larger, global properties of the shape is maintained.
3. Furthermore, and equally importantly, the flow stops once these notches are diffused into the main structure.
4. Edge definition is maintained, and, in some global sense, the area inside the boundary is preserved.
5. The noise removal capabilities of the min/max flow is scale-dependent, and can hierarchically adjusted.



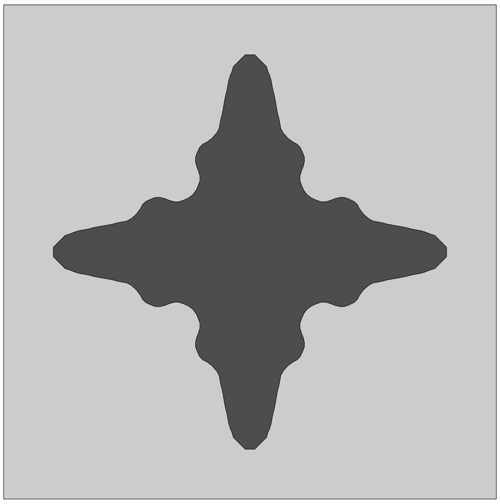
Initial Boundary
“Noisy” Shape



Min/Max Flow:
Stencil Width $= 0$; $(T = \infty)$



Min/Max Flow:
Stencil Width $= 1$; $(T = \infty)$



Continued Min/MaxFlow:
Stencil Width $= 2$; $(T = \infty)$

Figure 7: Motion of StarShaped Region with Noise under Min/Max Flow at Various Stencil Levels

6. The scheme requires only a nearest neighbor stencil evaluation.

4 The Switch Function in More Depth: Min/Max Flow and the Role of Masks in Level Sets Flow

The above min/max flow switch is, in fact, remarkably subtle in what it does. It works because of three reasons:

- First, the embedding of a front as a level set allows us to use information about neighboring level sets to determine whether to use the min flow or the max flow
- Second, the level set method allows the construction of barrier masks to thwart motion of the level sets
- Third, the discretization of the problem onto a grid allows one to select a natural scale to the problem

In this section, we examine our choice of a min/max switch in more detail, and show why scales below a given range are removed.

4.1 Min/Max Flow on Zero Order Structures

Consider first the case of black disk on a white background, built in the following way. We imagine initialization with the signed distance function, with ϕ chosen as negative on the inside and positive on the outside, and for display purposes, map all negative ϕ values into black and all positive ϕ values into white. Then the normal $\vec{n}$ points from the black disk outwards towards the white exterior. As defined by $\nabla \cdot \nabla\phi/|\nabla\phi|$, the curvature of the boundary is positive. Obviously then if we move this boundary under curvature flow, i.e., $F = \kappa$, the boundary will move inwards and the disk will disappear. If we move the boundary under the flow $F = \max(\kappa, 0)$, the same motion occurs, since the curvature of all the level curves is positive. If we move this boundary under the flow $F = \min(\kappa, 0)$, the shape will remain fixed.

Let us define a flow, called $F^{Stencil=0}_{\min/\max}$, as follows:

$$F^{Stencil=0}_{\min/\max} = \begin{cases} \min(\kappa, 0) & \text{if } \phi(x,y) < 0 \\ \max(\kappa, 0) & \text{if } \phi(x,y) \geq 0 \end{cases} \tag{10}$$

Our choice of the superscript "Stencil=0" means that we only use information about the level set function ϕ in a radius 0 around the point (x, y). We now examine the effect of the above flow. All the level curves in the black region corresponding to $\phi < 0$ have positive curvature, hence they do not move, since the min function is used and the speed function evaluates to zero at such points.

Conversely, all the level curves in the white region, corresponding to $\phi > 0$ must move, since the curvature is positive there and the speed function in the white region evaluates to the curvature. Thus we see that in the above flow, the black region acts a barrier to the attempts of the white exterior to move the zero level set. Thus, the level curves "pile up" around the boundary corresponding to the zero level set, but it does not move.

Conversely, suppose we exchange the roles of black and white, and consider a white circle on a black background. We shall show that the same thing occurs. This case corresponds to $\phi > 0$ in the interior and $\phi < 0$ in the exterior. The normal points from out to the inside, and the curvature (as defined by $\nabla \cdot \nabla\phi / |\nabla\phi|$) evaluates to negative. Under our flow $F^{Stencil=0}_{\min / \max}$, we again examine the two regions. The level curves inside the white region corresponding to $\phi > 0$ all have negative curvature, hence under $\max(\kappa, 0)$ they cannot move and act as a barrier; similarly, the level curves in black region corresponding to $\phi < 0$, all of whom have negative curvature, attempt to move inwards, and are stopped by the frozen white mask. Thus, the zero level set itself does not move.

Thus, under the flow $F^{Stencil=0}_{\min / \max}$, convex shapes do not move. We now show that nonconvex shapes cannot move either. Consider the four-pointed star-shaped region as given in Figure 1a, where the inside is black and the outside is white. Inside the star-shaped region, the level set function is negative, and hence the flow $\min(\kappa, 0)$ is selected and attempts to form a convex hull *as seen from the black region.* Outside, the level set function is positive, and hence the max flow is selected; this attempts to form the convex hull *as seen from the white region.* The net effect is for the white region to act as a barrier to the black region and vice-versa, hence there is no motion of the zero level set itself.

We summarize as follows; the flow given by $F^{Stencil=0}_{\min / \max}$ allows no motion of the boundary. This seems like a tremendous amount of work to go through to achieve $F = 0$, but we now show that a speed function based on a larger stencil, i.e., one which includes more points than just the value of ϕ at (x, y), produces motion.

4.2 Min/Max Flow on First Order Structures

Consider the level set flow defined on a grid of size h, and imagine that we have a shape with a perturbations on the boundary of size one cell. As an example, we return to the black square on a white background with notches given in Figure 4a. We have just seen that the flow $F^{Stencil=0}_{\min / \max}$ produces no motion. Consider now a flow of Stencil=1, in which we use the average value of ϕ in a neighborhood of radius one unit h around the point (x, y) to tell us which flow to select;

$$F^{Stencil=1}_{\min / \max} = \begin{cases} \min(\kappa, 0) & \text{if } Ave^{R=h}_{\phi(x,y)} < 0 \\ \max(\kappa, 0) & \text{if } Ave^{R=h}_{\phi(x,y)} \geq 0 \end{cases} \tag{11}$$

Here, $Ave^{R=h}_{\phi(x,y)}$ is defined as the average value of ϕ in a neighborhood of radius h around the point (x, y). Similar to the above, each region attempts to "convexify" itself and hence resists the other; only in the notches is there some confusion as to which "side" the notch belongs. In the notches, however, the sign of the average value of ϕ is the opposite of the sign of the value at the notch. Hence, the notches do not act as barriers, and the "external" flow (either black or white), is allowed to flow through the notch, until the perturbation is removed. Once the perturbation (i.e. notch) is removed, there are no remaining first order structures; that is, nowhere are "convex" bumps (as seen as either the black or white side) allowed to move. Another way to characterize this is to say that the switch function evaluates to zero when it is impossible to further convexify one region without "de-convexifying" the other.

4.3 Min/Max Flow on kth Order Structures

The above discussion explains why the min/max flow works; furthermore, it provides a nesting of flows designed to remove successively larger scales. By way of notation, we can now write that

$$F_{\min\,/\,\max}^{Stencil=k} = \begin{cases} \min(\kappa, 0) & \text{if } Ave_{\phi(x,y)}^{R=kh} < 0 \\ \max(\kappa, 0) & \text{if } Ave_{\phi(x,y)}^{R=kh} \geq 0 \end{cases} \tag{12}$$

Thus, a stencil of size k computes the average over a disk of radius kh, where h is the discretization size. Roughly speaking, we can see that this flow will attempt to remove structures of width kh. As an example, let $k = \infty$, and return to the case of the disk. Since the average will compute to a value close to the background color, on this scale all structures are insignificant and the max flow will be chosen everywhere, forcing the boundary to disappear.

We thus see that the min/max flow as defined in Eqn. 12 provides a hierarchy of scales that can be removed from a shape. In the next section, we apply this flow to a variety of images.

5 Min/Max Flows for Binary, Grey-Scale, Texture, and Color Images

5.1 Application of Min/Max Flows to Binary Images

We now apply our scheme given by Eqn. 9 to the problem of binary images with noise. Since we are looking at black and white images, where 0 corresponds to black and 255 to white, the threshold value $T_{threshold}$ is taken as 127.5 rather than 0. In Figure 8, we add noise to a black and white image of a hand-written character. The noise is added as follows; 10% noise means that at 10% of the pixels, we replace the given value with a number chosen with uniform distribution between 0 and 255. Thus, a full spectrum of gray noise is added to the original binary image, The left column give the original figure with the corresponding percentage of noise; the right column are reconstructed values. We stress once again that the figures on the right are converged; they stop automatically, and continued application of the scheme yields no change in the results. Results are reconstructed from 25%, 50%, 60%, and 80% noise.

5.2 Grey-Scale Images: Min/Max Flows and Scale-Dependent Noise Removal

Imagine a grey-scale image; for example, two concentric rings of differing grey values. Choosing a threshold value of 127.5 is clearly inappropriate, since the value "between" the two rings may not straddle the value of 127.5, as it would it an original binary image. Instead, our goal is to locally construct an appropriate thresholding value. We follow the philosophy of the algorithm for binary images.

Imagine a grey scale image, such as the two concentric rings, in which the inner ring is slightly darker then the exterior ring; here, we interpret this as ϕ being more negative in the interior ring than the exterior. Furthermore, imagine a slight notch protruding outwards into the lighter ring, (see Figure 9). Our goal is decide whether the area within the notch belongs to the lighter region, that is, whether it is a perturbation that should be suppressed and "reabsorbed" in to the appropriate background color. We determine this by first computing the average value of the intensity ϕ in the neighborhood around the point. We then must determine a comparison value which indicates the "background" value. We do so by computing a threshold $T_{threshold}$, defined as the average value of the intensity obtained in the direction perpendicular to the gradient direction. Note that since the direction perpendicular to the gradient is tangent to the isointensity contour through (x, y), the two points used to compute are either in the same region, or the point (x, y) is an inflection point, in which the curvature is in fact zero and the min/max flow will always yield zero.

Formally then,

$$F_{min/max} = \begin{cases} \max(\kappa, 0) & \text{if } Average(x,y) < T_{threshold} \\ \min(\kappa, 0) & \text{otherwise} \end{cases} \tag{13}$$

This has the following effect. Imagine again our case of a grey disk on a lighter grey background, where the darker grey corresponds to a smaller value of ϕ than the lighter grey. When the threshold is larger than the average, the max is selected, and the level curves move in. However, as soon as the average becomes larger, the min switch takes over, and the flow stops. The arguments are similar to the ones given in the binary case.

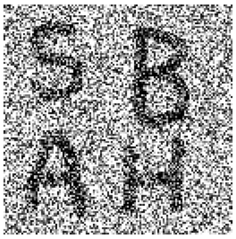
In Figure 10, we use this scheme to remove salt-and-pepper gray-scale noise from a grey-scale image. Once again, we add noise to the figure by replacing $X\%$ of the pixels with a new value, chosen from a uniform random distribution between 0 and 255, Our results are obtained as follows. We begin with two levels

(a) 25.0% noise

(b) Restored

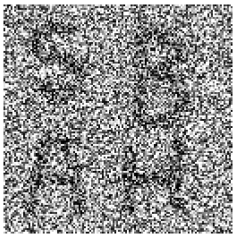
(c) 50.0% noise

(d) Restored



(e) 65.0% noise

(f) Restored



(g) 80.0% noise

(h) Restored

Figure 8: Image restoration of Binary Images with Grey-Scale Salt-and-Pepper Noise Using Min/Max Flow

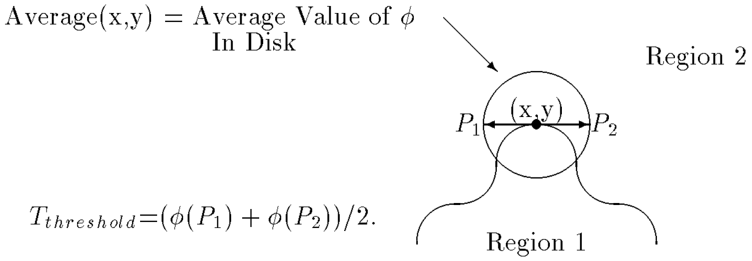


Figure 9: Threshold Test for Min/Max Flow

of noise; 25% noise in Figure 10a and 50% noise in Figure 10d. We first use the min/max flow from Eqn.13 until a steady-state is reached in each case, (Figure 10b and Figure 10e). This removes most of the noise. We then continue with a larger stencil for the threshold to remove further noise (Figure 10c and Figure 10f). For the larger stencil, we computer the average $Average(x, y)$ over a larger disk, and compute the threshold value $T_{threshold}$ using a correspondingly longer tangent vector.

5.3 Selective Smoothing Coupled to Min/Max Flows: Medical Images

In certain cases, one may want to remove some level of detail in an image; for example, in medical imaging, in which a low level of noise or image gradient is undesired, and the goal is enhancement of features delineated by large gradients. In this case, a simple modification of our min/max flow can achieve good results. We begin by defining the mean curvature of the image when viewed as a graph; that is, let

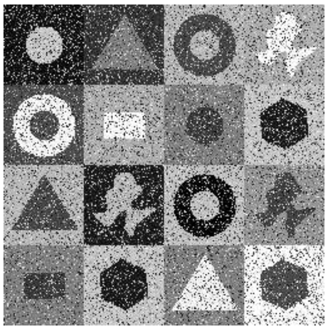
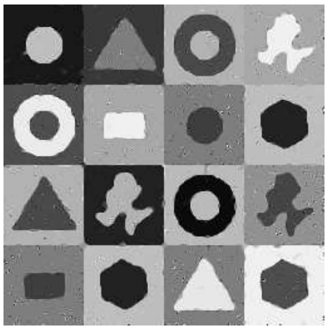
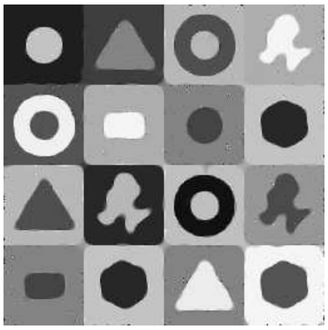
$$M = \frac{(1+I_{xx})I_y^2 - 2I_xI_yI_{xy} + (1+I_{yy})I_x^2}{(1+I_x^2+I_y^2)^{3/2}} \tag{14}$$

be the mean curvature. If we flow the image according to its mean curvature, i.e.,

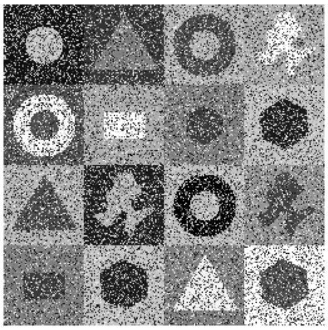
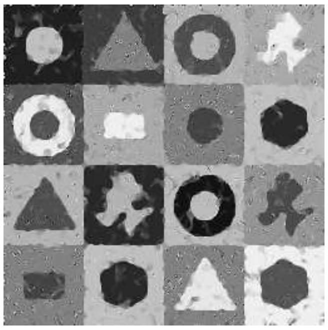
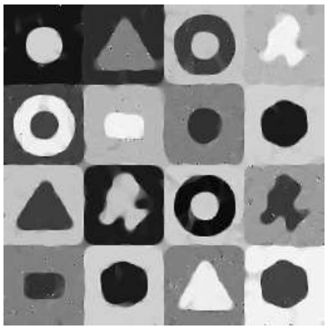
$$I_t = M(1+I_x^2+I_y^2)^{1/2} \tag{15}$$

this will smooth the image. Thus, given a user-defined threshold $V_{gradient}$ based on the local gradient magnitude, we use the following flow to selectively smooth the image:

$$F_{min/max/smoothing} = \begin{cases} M & \text{if } \mid \nabla I \mid < V_{gradient} \\ \text{min/max flow} & \text{otherwise} \end{cases} \tag{16}$$

(a) 25% Noise (b) Min/Max Flow (c) Cont.: Larger Stencil

(d) 40% Noise (e) Min/Max Flow (f) Cont.: Larger Stencil

Figure 10: Min/Max Flow. The left column is the original with noise, the center column is the steady-state of min/max flow, the right column is the continuation of the min/max flow using a larger stencil

(a) Original (b) Min/Max Flow (c) Min/Max + Mean Curvature Flow

Figure 11: Min/Max Flow with Selective Smoothing. The left image is the original. The center image is the steady-state of min/max flow. The right image is the steady-state of the min/max flow together with mean curvature flow for selective smoothing.

Thus, below a prescribed level based on the gradient, we smooth the image using flow by mean curvature; above that level, we use our standard min/max flow. Other choices for the smoothing flow include isotropic diffusion and curvature flow. We have had the most success with mean curvature flow; isotropic diffusion is too sensitive to variations in the threshold value $V_{gradient}$, since edges just below that value are diffused away, while edges are preserved in mean curvature flow. Our choice of mean curvature flow over standard curvature flow is because mean curvature flow seems to perform smoothing in the selected region somewhat faster.

In Figure 11, we show results of this scheme (Eqn.9) applied to a digital subtraction angiogram (DSA). In Figure 11a, we show the original image. In Figure 11b, we show the steady-state min/max flow image. In Figure 11c, we show the steady-state obtained with min/max flow coupled to mean curvature flow in the lower gradient range.

5.4 Preserving Fine Detail: Min/Max Flows for Textured Images

In some cases, one wants to remove noise from an image which contains considerable fine grain structure, for example, in textured images. If information about the texture is known *a priori*, then it is natural to invent schemes which use such information to preserve the texturing features. However, we are concerned here with schemes that have no previous information about the nature of the texture in the image, and determine all noise removal processes on the basis of only the image itself.

Consider now an image, which view as described by an intensity function $I(x, y)$. Our previous schemes have focussed on viewing the image as a collection of level curves of the intensity function. However, viewed as a graph, we may also evaluate the mean curvature (which is the average of the two principle curvatures) and the Gaussian curvature (the product of the two principle curvatures) of the image when viewed as a surface, as discussed above in the section on selective smoothing.

If we view a texture as a collection of ridges, valleys, grains, e.t.c., then we note that isolated spots of noise (peaks and troughs) will have large absolute values of mean curvature, while the fundamental texture forms will have, relatively speaking, lower values of the mean curvature. This is because the ridge/valley lines correspond to lines with no curvature. Of course this is not an absolute statement, but it suggests the following scheme for textured images. We compute the mean curvature of an image when viewed as a graph, and then establish a lower value (M_1) and an upper value (M_2) for the mean curvature. In the lowest range we turn off all flows; in the middle range we perform min/max flow, and in the upper range, we perform mean curvature flow as described above. That is, we have:

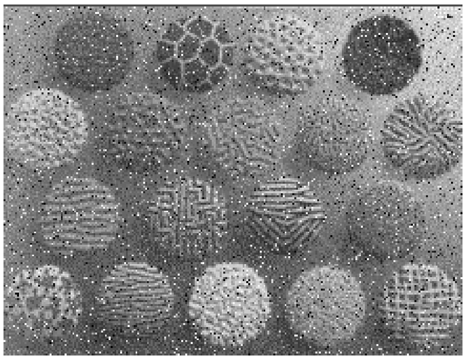
$$F_{min/max/texture} = \begin{cases} 0 & \text{if } \mid M \mid < M_1 \\ \text{min/max flow} & \text{if } M_1 < \mid M \mid < M_2 \\ \text{M} & \text{if } M_2 < \mid M \mid \end{cases} \tag{17}$$

We apply the results of this scheme (Eqn.17) for processing textured images with noise in Figure 12. Again, we add salt-and-pepper grey-scale noise to the original.

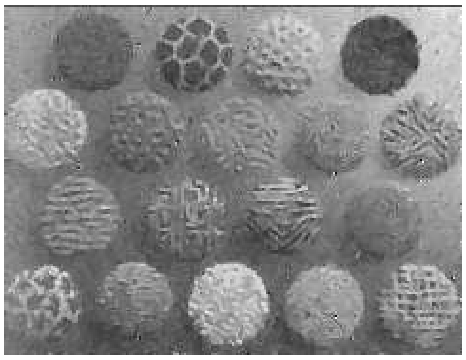
5.5 Color Images

The extension of the techniques presented above to color images is straightforward. We decompose the signal into three components (either HSV, RGB, or some other suitable framework), and process each channel separately according to the techniques developed above; the full result is then assembled.

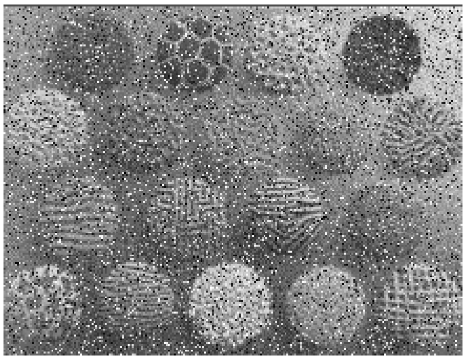
There is a distinct problem with this approach. In each channel, we solve the min/max partial differential equation flow. Because each channel is treated separately, occasionally intermediate values during the flow in each channel can superposition to create a new, unexpected color. This is in fact a well-known problem in color interpolation theory; given any two colors, it is not clear how to smoothly interpolate from one to the other so that the intermediate colors are always perceived as an appropriate blend between the two. While possible solutions are possible if one confines the two colors to restricted regions of color space (such as two colors with the same hue), there is no general solution to this phenomenon. Our algorithm makes no attempt to find an optimal interpolation



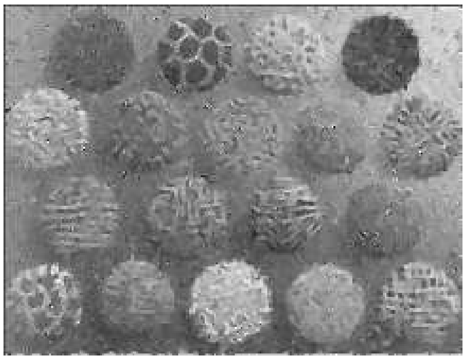
(a) 10.0% noise



(b) Restored



(c) 20.0% noise



(d) Restored

Figure 12: Textured Images: Min/Max Flow with Flow Range Determined by Mean Curvature

(a) 15.0% noise

(b) Restored

Figure 13: Color Images: Reconstruction of Color Images using Min/Max Flow (This is a 24-bit color image; quality is degraded if grey-scale reproduced.)

path through color space; another way to say this is that we do not couple the three distinct partial differential equations.

In Figure 13 we remove noise from a color, textured image of a rug. Noise is added in all three channels; 15% noise means that at 15% of the pixels, the values in each channel are discarded, and each is replaced by a random value drawn from a uniform distribution between 0 and 255.

6 Coupling of Min/Max Flow to Previous Work

Previous work on partial differential equation/level set schemes for image processing may be easily incorporated into our framework; in short, the curvature flow is replaced by our min/max flow. We point out three couplings in particular:

- Image sharpening algorithms based on shock filters, as in [16, 4] may be coupled to our min/max flow.
- The rate of smoothing is sometimes made to vary inversely with the gradient magnitude [19]. If such an effect is desired in our setting, one merely replaces the curvature κ in our flow $\min/\max(\kappa, 0)$ by $\kappa/|\nabla\phi|$.

- The affine invariant flow $\kappa^{1/3}$ [20] can also be modified by using the $\min / \max(\kappa^{1/3}, 0)$.

All of the above couplings are completely straightforward, and we do not discuss them in detail here.

7 Additional Examples

In this section, we present further images which are enhanced by means of our min/max flows. We begin with a series of medical images in Figure 14; here, no noise is artificially added, and instead our goal is to enhance certain features within the given images.

Next, we study the effect of our min/max scheme on multiplicative noise added to a grey-scale image. In Figure 15 we show the reconstruction of an image with 15% multiplicative noise.

Next, we add 100% Gaussian grey-scale noise; that is, a random component drawn from a Gaussian distribution with mean zero is added to each (every) pixel. In Figure 16 we show the original with noise together with the reconstructed min/max flow image. Here we use the coupling $\min / \max(\kappa/|\nabla\phi|, 0)$.

Finally, we apply our scheme to a color image with 100% Gaussian color noise added; that is, Gaussian noise added to all three channels. In Figure 17 we show the original with noise together with the reconstructed min/max flow image.

Acknowledgements: All calculations were performed at the University of California at Berkeley and the Lawrence Berkeley National Laboratory. We would like to thank Guillermo Sapiro for helpful discussions.

References

[1] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *J. Comp. Phys.*, Vol. 118(2), pp. 269–277, May 1995.

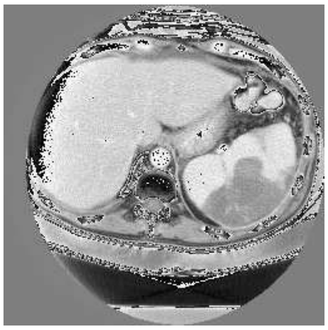
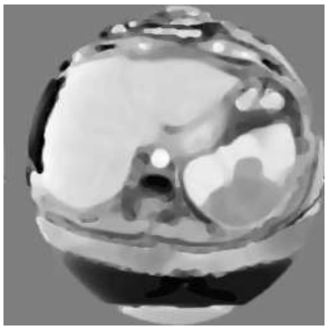
[2] D. Adalsteinsson and J. A. Sethian, "A unified level set approach to etching, deposition and lithography: I, II, and III." I, II: to appear, *J. Comp. Phys.*, 1995; III: under preparation.

[3] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. II," *SIAM Journal on Numerical Analysis*, Vol. 29(3), pp. 845–866, 1992.

[4] L. Alvarez and L. Mazorra, "Signal and image restoration using shock filters and anisotropic diffusion," *SIAM J. Numer. Anal.*, Vol. 31, No. 2, pp. 590–605, April 1994.

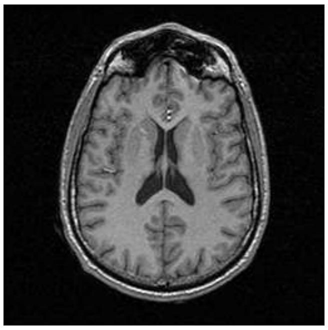
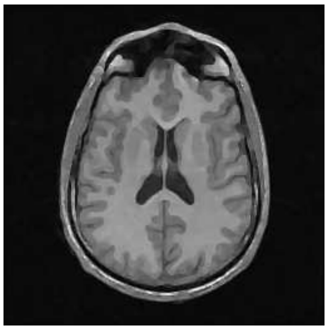

(a) Original image

(b) Min/Max:Final

(c) Original image

(d) Min/Max:Final

(e) Original image

(f) Min/Max:Final

Figure 14: Min/Max Flow with Selective Smoothing

(a) Image with Noise (b) Min/Max:Final

Figure 15: Min/Max Flow Applied to Multiplicative Noise

[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp. 679–698, 1986.

[6] D. L. Chopp, "Computing minimal surfaces via level set curvature flow," *Journal of Computational Physics*, Vol. 106, pp. 77–91, 1993.

[7] D. L. Chopp and J. A. Sethian, "Flow under curvature: singularity formation, minimal surfaces, and geodesics," *Jour. Exper. Math.*, Vol. 2(4), pp. 235–255, 1993.

[8] M. Gage, "Curve shortening makes convex curves circular," *Inventiones Mathematica*, Vol. 76, pp. 357, 1984.

[9] M. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Diff. Geom.*, Vol. 26, pp. 285–314, 1987.

[10] R. Malladi and J. A. Sethian, "Image processing via level set curvature flow," to appear in *Proc. Natl. Acad. of Sci., USA*, 1995.

[11] R. Malladi and J. A. Sethian, "A unified approach for shape segmentation, representation, and recognition," Report LBL-36069, Lawrence Berkeley Laboratory, University of California, Berkeley, August 1994.

[12] R. Malladi, D. Adalsteinsson, and J. A. Sethian, "Fast method for 3D shape recovery using level sets," under preparation.

[13] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery," in *Proceedings of*

(a) Original image with Gaussian Noise

(b) Reconstructed Min/Max Flow

Figure 16: Continuous Gaussian Noise added to Image

(a) Original image with Gaussian Noise

(b) Reconstructed Min/Max Flow

Figure 17: Continuous Gaussian Noise added to Color Image (This is a 24-bit color image; quality is degraded if grey-scale reproduced.)

Third European Conference on Computer Vision, LNCS Vol. 800, pp. 3–13, Stockholm, Sweden, May 1994.

[14] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17(2), pp. 158–175, Feb. 1995.

[15] D. Marr and E. Hildreth, "A theory of edge detection," *Proc. of Royal Soc. (London)*, Vol. B207, pp. 187–217, 1980.

[16] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Num. Anal.*, Vol. 27, pp. 919–940, 1990.

[17] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," *Journal of Computational Physics*, Vol. 79, pp. 12-49, 1988.

[18] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12(7), pp. 629–639, July 1990.

[19] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Modelisations Matematiques pour le traitement d'images*, INRIA, pp. 149–179, 1992.

[20] G. Sapiro and A. Tannenbaum, "Image smoothing based on affine invariant flow," *Proc. of the Conference on Information Sciences and Systems*, Johns Hopkins University, March 1993.

[21] G. Sapiro and A. Tannenbaum, "Area and length preserving geometric invariant scale-spaces," *Proc. of Third European Conference on Computer Vision*, LNCS Vol. 801, pp. 449–458, Stockholm, Sweden, May 1994.

[22] J. A. Sethian, "An analysis of flame propagation," Ph.D. Dissertation, Mathematics, University of California, Berkeley, 1982.

[23] J. A. Sethian, "Curvature and the evolution of fronts," *Commun. in Mathematical Physics*, Vol. 101, pp. 487–499, 1985.

[24] J. A. Sethian, "Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws," *Journal of Differential Geometry*, Vol. 31, pp. 131–161, 1990.

[25] J. A. Sethian, "Curvature flow and entropy conditions applied to grid generation," *Journal of Computational Physics*, Vol. 115, No. 2, pp. 440–454, 1994.

[26] C. Rhee, L. Talbot, and J. A. Sethian, "Dynamical study of a premixed V flame," to appear, *Jour. Fluid Mech.*, 1995.

[27] J. Zhu and J. A. Sethian, "Projection methods coupled to level set interface techniques," *J. Comp. Phys.*, Vol. 102, pp.128-138, 1992.