Image smoothing and enhancement via min/max curvature flow *

R. Malladi and J. A. Sethian

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

e-mail: {malladi,sethian}@csr.lbl.gov

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. Essentially, the method consists of moving the isointensity contours in a image under curvature dependent speed laws to achieve enhancement. 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.

Key Words: *Geometric Heat Equation, Level Sets, Curvature Flow, Image Smoothing, Image Enhancement, Mean Curvature*

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 are precursors to many low level vision procedures such as edge finding,[11,2] shape segmentation, and shape representation.[9,10,7] In this paper, we present a method for image smoothing and enhancement which is a variant of the geometric heat equation. This technique is based on a min/max switch which controls the form of the application of the geometric heat equation, selecting either flow by the positive part of the curvature or the negative part, based on a local decision. This approach has several key virtues. First, it contains only one enhancement parameter, which it most cases is automatically chosen. Second, the scheme automatically picks the stopping criteria; continued application of the scheme produces no further change. Third, the method is one

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.

0-8194-2040-9/96/$6.00

of the fastest possible schemes based on a curvature-controlled approach.

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[13] have used an anisotropic diffusion process which performs intraregion smoothing in preference to interregion smoothing. A significant advancement was made by Alvarez, Lions, and Morel (ALM),[1] who presented a comprehensive model for image smoothing.

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 $\kappa = \text{div}\frac{\nabla I}{|\nabla I|}$ is the local curvature. One variation of this scheme comes from replacing the curvature term with its affine invariant version (see Sapiro and Tannenbaum[15]). 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.[14]

In this work, we return to the original curvature flow equation, namely $I_t = F(\kappa) \mid \nabla I \mid$, and Osher-Sethian[12,17] 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 a straightforward scheme.

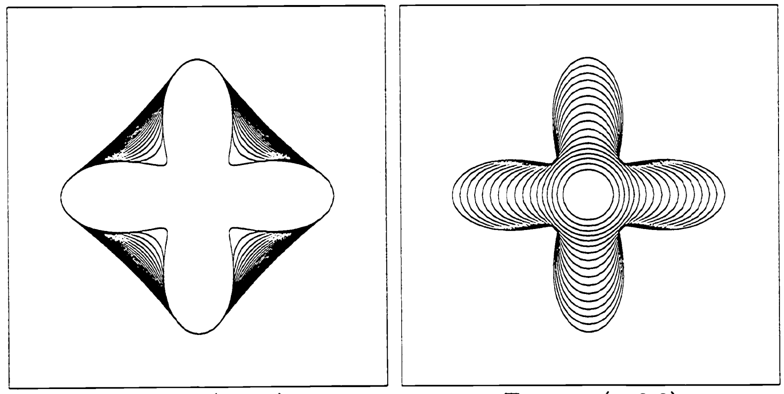
To motivate this approach, we begin by discussing curvature motion, and then develop the complete model which includes image enhancement as well. The crucial ideas on min/max flows upon which this paper is based have been reported earlier by the authors in[5]; more details and applications in textured and color image denoising may be found in Malladi and Sethian.[6] The outline of this paper is as follows. First, in Section II, we study the motion of a curve moving under its curvature, and develop an automatic stopping criteria. Next, in Section III, we apply this technique to enhancing binary and grey-scale images that are corrupted with various kinds of noise.

2 MOTION OF CURVES UNDER CURVATURE

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. For a specific speed function, namely $F(\kappa) = -\kappa$, it can be shown that an arbitrary closed curve (see Gage,[3] Grayson[4]) collapses to a single point.

2.1 The Min/Max flow

We now modify the above flow. Motivated by work on level set methods applied to grid generation[18] and shape recognition,[7] we consider two flows, namely $F(\kappa) = \min(\kappa, 0.0)$ and $F(\kappa) = \max(\kappa, 0.0)$. As shown in Figure 1, the effect of flow under $F(\kappa) = \min(\kappa, 0.0)$ is to 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



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

Figure 1: Motion of a curve under Min/Max flow

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.

Here, we have evolved the curve using the Osher-Sethian level set method, see,[12] which grew out of earlier by Sethian[16] on the mathematical formulation of curve and surface motion. Briefly, this technique works as follows. Given a moving closed hypersurface $\Gamma(t)$, that is, $\Gamma(t = 0) : [0, \infty) \rightarrow R^N$, 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

$$\phi(x, t = 0) = \pm d \tag{2}$$

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 $\phi(x, t = 0) : R^N \rightarrow R$ with the property that

$$\Gamma(t = 0) = (x|\phi(x, t = 0) = 0) \tag{3}$$

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

$$\phi_t + F|\nabla\phi| = 0 \tag{4}$$

$$\phi(x, t = 0) \quad given \tag{5}$$

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.

Consider now the square with notches on each side shown in Figure 2a. We let the color black correspond to the "inside" where $\phi < 0$ and the white correspond to the "outside" where $\phi > 0$. 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 2b, we see the effect of curvature flow; the notches are removed, but the shape is fully diffused. In Figure 2c, 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

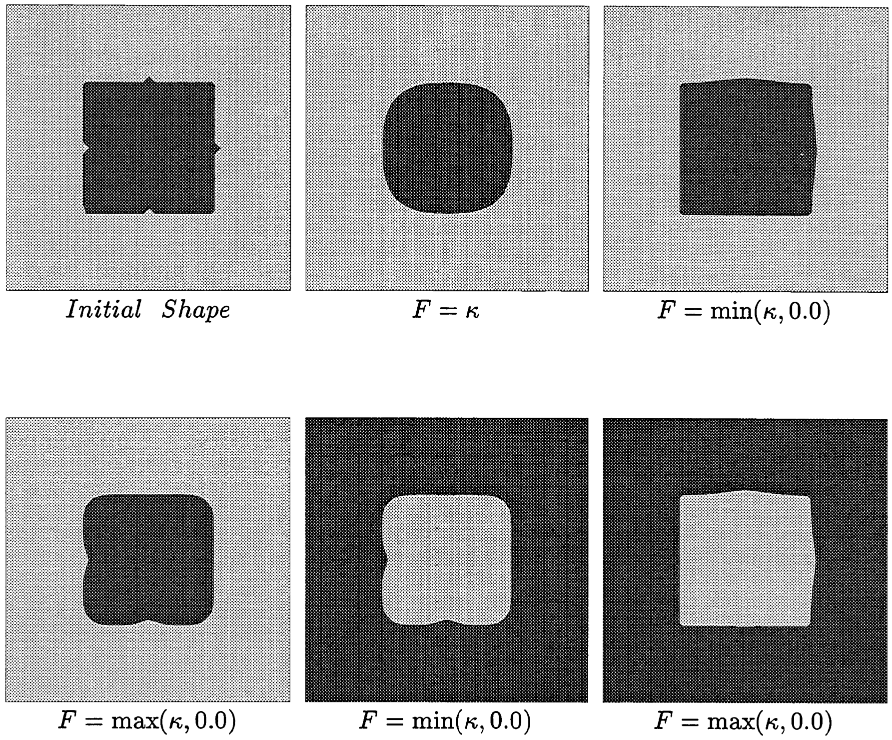


Figure 2: Motion of notched region under various flows

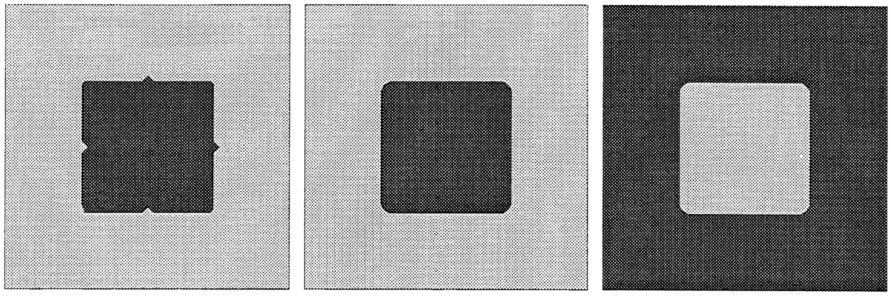
move under this flow. Conversely, as shown in Figure 2d, obtained with speed $F = \max(\kappa, 0.0)$, the inner notches stay fixed and the front moves in around them, while the outer notches are diffused. Continual application of this flow causes the shape to shrink and collapse. Finally, in Figure 2e and Figure 2f, 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.

2.2 The switch

In this section, we present the switch function to flow the above shape. Our construction of a switch is motivated by the idea of comparing the value of a function with its value in a ball around the 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

$$F_{min/max} = \begin{cases} \min(\kappa, 0) & \text{if } Average(x, y) < 0 \\ \max(\kappa, 0) & \text{otherwise} \end{cases} \tag{6}$$



Initial Shape $F = F_{min/max}$ $F = F_{min/max}$

Figure 3: Motion of notched region under Min/Max flow

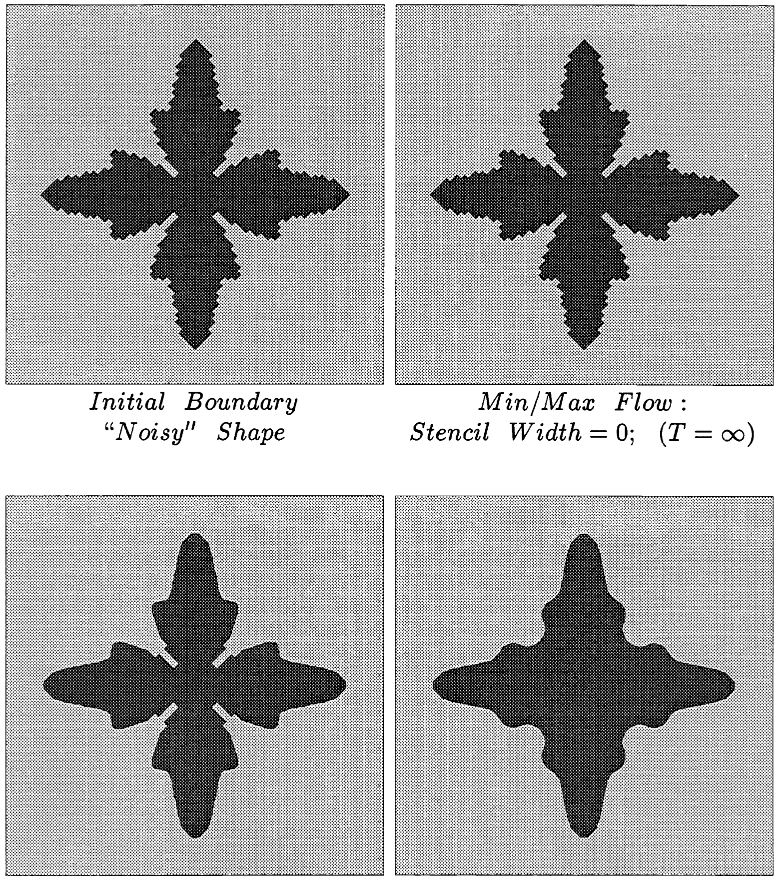
Here, we view 0 as the "threshold" value $T_{threshold}$; since it is halfway between the black value of -1 and the white value of 1. This flow can be seen to thus choose the "correct" flow between the min flow and the max flow. As a demonstration, in Figure 3a, we show the initial notched region. In Figure 3b, we show the results using the min/max given in Eqn. 6. 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 3c. 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.

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. Thus, the stencil size is the single parameter that determines the flow and hence the noise removal capabilities. We view this as a natural and automatic choice of the stencil, since it is given by the pixel refinement of the image. However, for a given pixel size, one can choose a larger stencil to exact noise removal on a larger scale; that is, we can choose to remove the next *larger* level of noise structures by increasing the size of our threshold stencil by computing 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 4; we start with an initial shape in Figure 4a which has "noise" in the boundary. We then perform the min/max flow until steady-state is achieved with stencil size zero in Figure 4b; 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; see Malladi and Sethian[6] for details. In Figure 4c, 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 4d. 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 roughly preserved up to the order of the smoothing.
5. The noise removal capabilities of the min/max flow is scale-dependent, and can be hierarchically adjusted.
6. The scheme requires only a nearest neighbor stencil evaluation.

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.



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 4: Motion of a StarShaped region with noise under Min/Max flow at various stencil levels

- 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.

Interested reader is referred to Malladi and Sethian[6] for a detailed explanation of the above issues.

3 APPLICATIONS

3.1 Application of Min/Max flows to binary images

We now apply our scheme given by Eqn. 6 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 6, 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

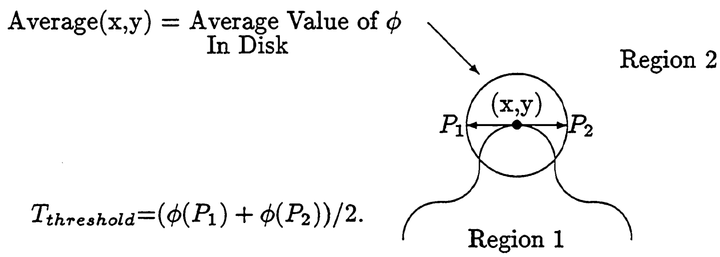


Figure 5: Threshold test for Min/Max flow

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%, and 80% noise.

3.2 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 5). 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,

$$F_{min/max} = \begin{cases} \max(\kappa, 0) & \text{if } Average(x, y) < T_{threshold} \\ \min(\kappa, 0) & \text{otherwise} \end{cases} \tag{7}$$

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.

Now 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 25% noise in Figure 6g. We first use the min/max flow from Eqn.7 until a steady-state is reached (Figure 6h). This removes most of the noise. We then continue with a larger stencil for the threshold to remove further noise (Figure 6i). For the larger stencil, we compute the average $Average(x, y)$ over a larger disk, and compute the threshold value $T_{threshold}$ using a correspondingly longer tangent vector.

3.3 Selective smoothing of 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

$$M = \frac{(1 + I_{xx})I_y^2 - 2I_x I_y I_{xy} + (1 + I_{yy})I_x^2}{(1 + I_x^2 + I_y^2)^{3/2}} \tag{8}$$

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

$$I_t = M(1 + I_x^2 + I_y^2)^{1/2} \tag{9}$$

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:

$$F_{min/max/smoothing} = \begin{cases} M & \text{if } \mid \nabla I \mid < V_{gradient} \\ \text{min/max flow} & \text{otherwise} \end{cases} \tag{10}$$

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. This is an empirical statement rather than one based on a strict proof.

In Figure 7, we show results of this scheme (Eqn.6) applied to a digital subtraction angiogram (DSA). In Figure 7a, we show the original image. In Figure 7b, we show the steady-state min/max flow image. In Figure 7c, we show the steady-state obtained with min/max flow coupled to mean curvature flow in the lower gradient range.

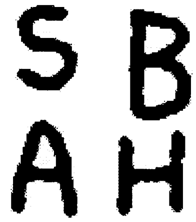
3.4 Additional examples

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 8; here, no noise is artificially added, and instead our goal is to enhance certain features within the given images and make them aminable to further processing like shape finding.[9,10,8]

Next, we study the effect of our min/max scheme on multiplicative noise added to a grey-scale image. In Figure 9a & b, we show the reconstruction of an image with 15% multiplicative noise.

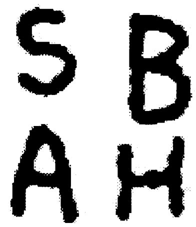
Next, we add 100% Gaussian grey-scale noise; that is, a random component drawn from a Gaussian distribution

(a) 25.0% noise

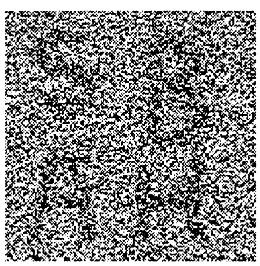


(b) Restored

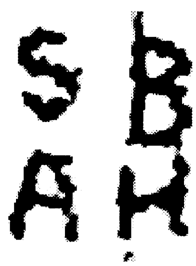
(c) 50.0% noise



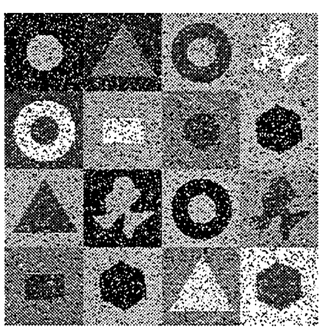
(d) Restored



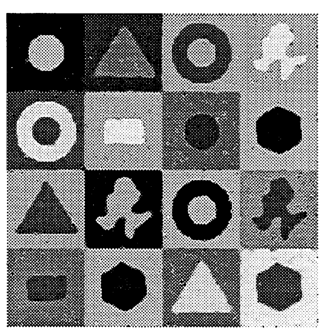
(e) 80.0% noise



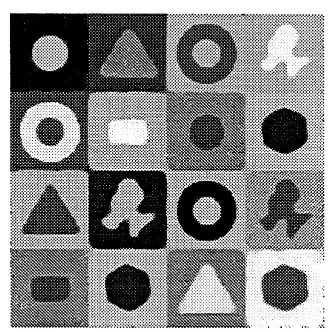
(f) Restored



(g) 25% Noise



(h) Min/Max Flow



(i) Cont.: Larger Stencil

Figure 6: Image restoration using min/max flow of binary and grey-scale images corrupted with grey-scale salt-and-pepper noise

(a) Original

(b) Min/Max Flow

(c) Min/Max + Mean Curvature Flow

Figure 7: 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.

with mean zero is added to each (every) pixel. In Figure 9c & d, we show the original with noise together with the reconstructed image with the min/max flow.

4 REFERENCES

[1] 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.

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

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

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

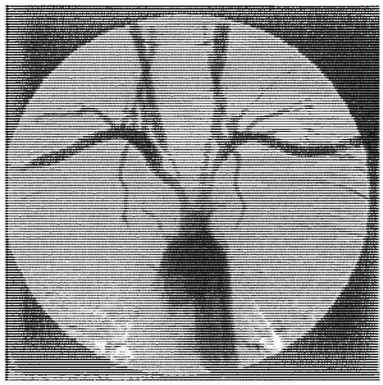
[5] R. Malladi and J. A. Sethian, "Image processing via level set curvature flow," *Proc. Natl. Acad. of Sci., USA*, Vol. 92, pp. 7046–7050, July 1995.

[6] R. Malladi and J. A. Sethian, "Image processing: Flows under min/max curvature and mean curvature," to appear in *Graphical Models and Image Processing*, March 1996.

[7] 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.

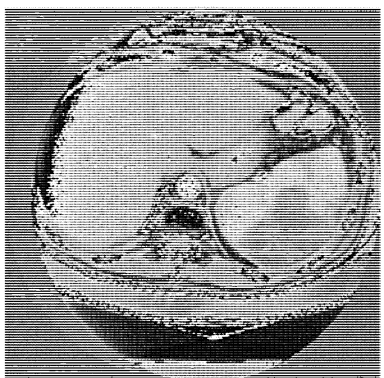
[8] R. Malladi, D. Adalsteinsson, and J. A. Sethian, "Fast method for 3D shape recovery using level sets," submitted.

[9] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery," in *Proceedings of Third European Conference on Computer Vision*, LNCS Vol. 800, pp. 3–13, Stockholm, Sweden, May 1994.

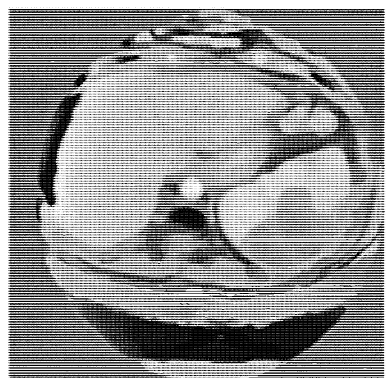


(a) Original image

(b) Min/Max:Final



(c) Original image

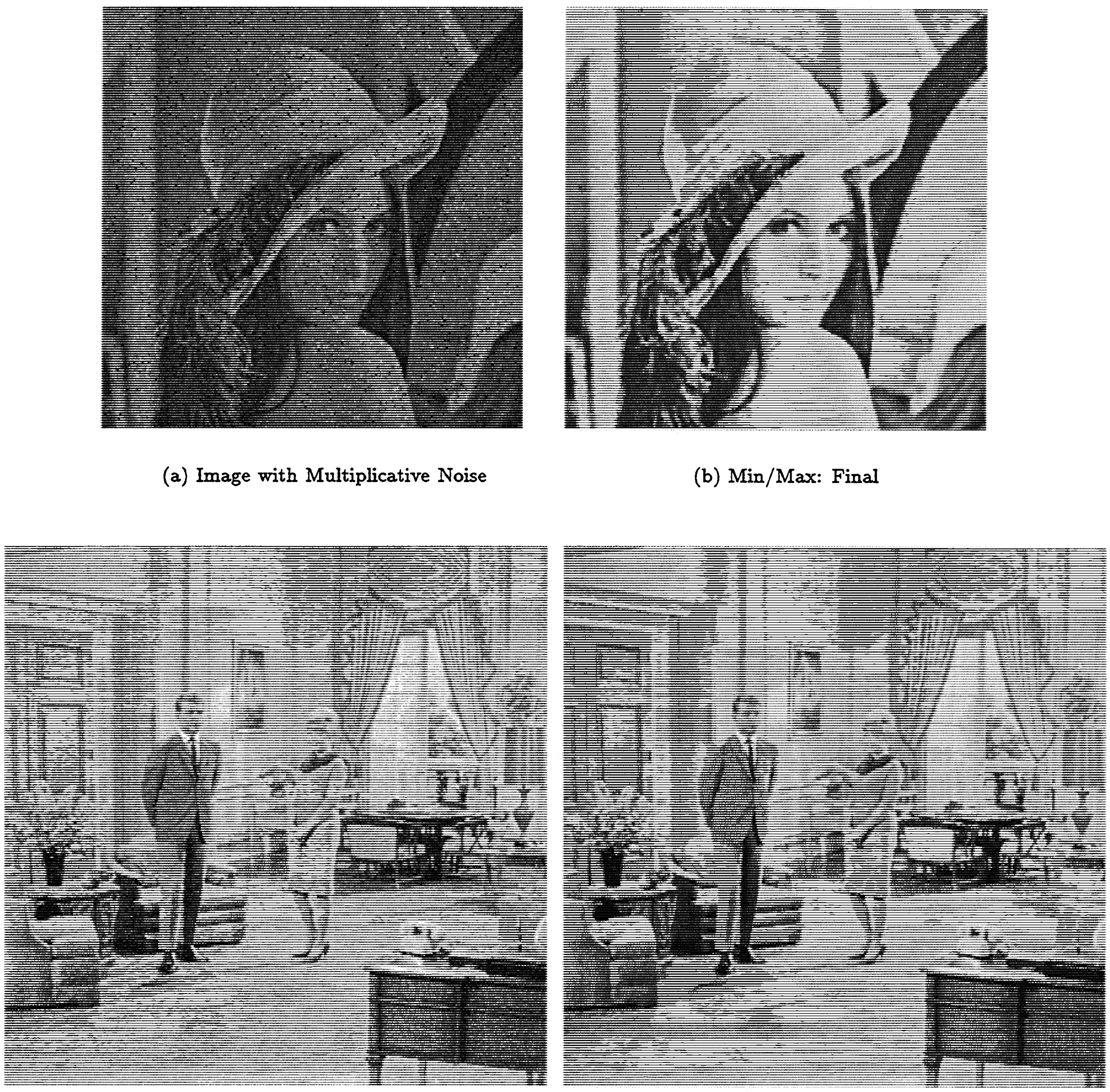


(d) Min/Max:Final

(e) Original image

(f) Min/Max:Final

Figure 8: Min/Max flow with selective smoothing



(a) Image with Multiplicative Noise

(b) Min/Max: Final

(c) Image with Gaussian Noise

(d) Min/Max: Final

Figure 9: Min/Max flow applied to multiplicative and Gaussian noise

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

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

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