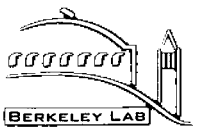
LBNL-42373



ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

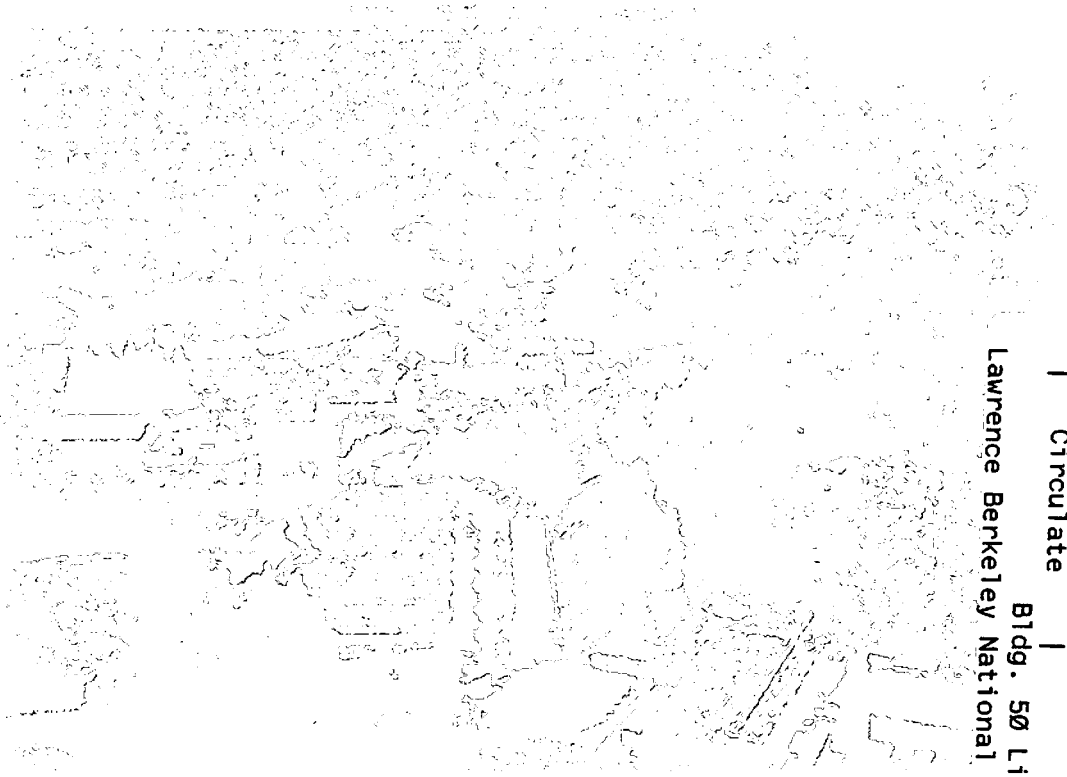
Edges are Image Discontinuities— Fast Edge Enhancement Based on Explicit-Jump Multiscale Analysis

Alessandro Sarti and Andreas Wiegmann

Computing Sciences Directorate
Mathematics Department

October 1998

To be submitted
for publication



DISCLAIMER

LBNL-42373

Edges are image discontinuities — fast edge enhancement based on explicit–jump multiscale analysis

Alessandro Sarti[1]
Andreas Wiegmann[1]
Department of Mathematics
Lawrence Berkeley National Laboratory
MS 50A-2152, 1 Cyclotron Rd
Berkeley, CA 94720

October 1998

[1]Supported by the Applied Mathematical Sciences Subprogram of the Office of Science under contract DE-AC03-76SF00098.

Abstract

We present a new idea for image filtering in computer vision. Thinking of edges as discontinuities in the image necessitates the use of specialized discretizations of partial differential operators that are consistent with the presence of such discontinuities. We discretize the partial differential operators based on upwinding on the gradient of an edge-indicator. Since this gradient points towards edges, no differences are computed across edges, explicitly introducing jumps in the image at the edges, while still allowing diffusion in the direction perpendicular to the edge. The explicit treatment of discontinuities leads to rapidly emerging edges and also allows the use of an absolute stopping criterion based only on the energy in smooth regions of the image.

1 Introduction

An image is given as a non-negative gray-level function $I_o : \Omega \rightarrow \mathbf{R}_{\geq 0}$, where the image domain Ω is a rectangle. The image is noisy so that ∇I_o need not exist at any point $(x, y) \in \Omega$. The imaging device creates images with a finite range of gray values, and since Ω is bounded this implies that $I_o \in L^2(\Omega)$.

The **edge enhancement problem** (EEP)[1] consists of computing a decomposition $\{\Omega_i\}$ of the domain $\Omega = \Omega_1 \cup \ldots \cup \Omega_N$ and computing an enhanced image I_e so that I_e "is similar to I_o", varies smoothly and slowly within each Ω_i and discontinuously on (part of) the boundary of the Ω_i. Boundaries of the homogeneous regions Ω_i that are not part of the boundary of Ω are called *edges*.

The **image segmentation problem** (ISP)[1] consists of computing a decomposition $\{\Omega_i\}$ of the domain $\Omega = \Omega_1 \cup \ldots \cup \Omega_N$ and computing a segmented image I_e so that I_e "is similar to I_o", constant within each Ω_i and discontinuous on the boundary of the Ω_i. A segmentation is a piecewise constant edge enhancement, but an edge enhancement is usually not a segmentation.

An enhanced image I_e may be viewed as the image of I_o under a map that takes functions without derivatives and maps them into "nearby" functions that have derivatives almost everywhere, i.e. from $L^2(\Omega) \rightarrow H^1(\Omega)$. The non-differentiability in the enhanced image occurs exactly across the edges. Our view is inspired by previous work on introducing discontinuities into solutions of elliptic boundary value problems in [13, 12], and similar to ideas presented in [3]. Liang and Wang [3] observed that a binary diffusion coefficient results in a method that "is simple to implement and analyze, and avoids difficulties and problems asscociated with nonlinear diffusion". We agree with this view, but wish to point out that the main difficulty with their, our as well as with all previous approaches is to decide where to suppress the diffusion. Once this is decided, [3] and our own work indicate that it is possible to avoid nonlinear diffusion and compute enhanced images much more quickly than previously possible. Our approach differs from [3] in how we decide the locations where to completely disable diffusion, and keeping linearized anisotropic diffusion in the sense of [1] in regions where we are not sure about the edge-detection. In particular, our diffusion coefficient is not binary as in [3], but does take on the value zero, different from [9]. Previous spatial difference operators, [9, 1, 7, 4, 5], implied the smoothness of the enhanced image across edges

[1] Definitions adapted from Mumford and Shah [6].

because differences were taken across the edges, which resulted in slow convergence to the desired goal of sharp edges. We will show by a simple example algorithm how it is possible to rapidly compute enhanced images with truly discontinuous edges by introducing explicit–jump differential operators. In particular, we will demonstrate on real examples that our method is 10 times faster than shock filtering [7] and how it helps with edge detection.

Besides the implied smoothness of the enhanced images, earlier methods also have the difficulty of determining when to stop the diffusion process [9, 1, 7]. In [4] a denoising method with a defined stop criterion has been proposed.

Our explicit treatment of discontinuities allows the use of an absolute stopping criterion that is independent of the amount of noise in the original image I_o. Our criterion is related to the total variation of the image; see e.g. Vogel and Oman, [11]. But instead of computing the total variation everywhere, by using our discretization we only compute the variation on smooth portions of the image.

In the next sections we define what we mean by an explicit–jump multiscale analysis, give the details of the implementation of the explicit jump differential operators and examples of enhanced images under a simple algorithm.

2 The method

Following earlier work [1] **(add more references here!)**, we use

$$g(\xi) = \frac{1}{1 + (\xi/\beta)^2}, \quad I_s = (G_\sigma * I_o), \text{ and } \quad \mathcal{G}(x, y) = g(\|\nabla I_s\|).$$

Here β is a parameter that selects the contrast, σ is the variance of the Gaussian

$$G_\sigma = \frac{\exp(-(x^2 + y^2)/\sigma^2)}{\sigma\sqrt{\pi}}$$

(σ is also called the scale), I_s a smoothed version of I_o (computed via heat-flow, an idea usually attributed to Koenderink [2]; see also next section) and $\mathcal{G}$ an edge indicator for I_o. $\mathcal{G}$ has small positive values for large gradients and values close to 1 for small gradients. This behavior is reversed from other edge indicators as for example in [15]. The negative gradient of our edge indicator, $-\nabla\mathcal{G}$, points toward edges.

Suppose that K is the set of edges determined[2] by $-\nabla\mathcal{G}$ and let

$$f_{K,\mathcal{G}}(I) = \int_{\Omega\setminus K} \mathcal{G}\|\nabla I\|^p dA. \tag{1}$$

We call $(f_{K,\mathcal{G}}(I))^{1/p}$ the p-energy of I with respect to K and $\mathcal{G}$. Usually, we consider $p = 1$ or $p = 2$. An image with reduced energy, i.e. $f_{K,\mathcal{G}}(I) < f_{K,\mathcal{G}}(I_o)$ is called edge enhanced, an image with zero energy, i.e. $f_{K,\mathcal{G}}(I) = 0$ is called a segmentation[3] of I_o. In this paper, we focus on (EEP), while [14] deals with (ISP). Edge enhanced images may be discontinuous on the edge set K, independent of p.

We call the reduction of (1), starting from I_o and preserving the integral[4] of the image on each Ω_i, an *explicit–jump multiscale analysis.*

We use the energy $f_{K,1}(I)$ as an absolute stopping criterion for the explicit–jump multiscale analysis. It takes into account only the variation of an image in smooth regions, and we will give an example showing that this can be explored to devise a method that automatically adjusts to the amount of noise in an image. The following results show that decreasing $f_{K,\mathcal{G}}$ also reduces the energy $f_{K,1}$.

Lemma 1 *For any $\mathcal{G} > 0$, the only local minimizers of (1) are the global minimizers of (1). These global minimizers are the edge-conforming segmentations I_e, i.e. images that are constant on each of the Ω_i.*

[2]We make this determination precise for the discretization of the problem in the next section.

[3]K induces a decomposition $\Omega = \Omega_1 \cup \ldots \cup \Omega_N$, and $f_{K,\mathcal{G}}(I) = 0$ implies $\|\nabla I\| = 0$ on each of the Ω_i.

[4]This property is often referred to as conservation. It is necessary for *causality* of the method.

Proof. The edge-conforming segmentations are global minimizers, because $f_{K,\mathcal{G}}(I) \geq 0$ for any $K, \mathcal{G}$ and I, and $f_{K,\mathcal{G}}(I_e) = 0$ if I_e is a segmentation that conforms with the edge set K.

Suppose that I is a minimizer of $f_{K,\mathcal{G}}$, but not constant on at least one of the Ω_i. Then

$$\int_{\Omega \setminus K} \mathcal{G} \|\nabla(I - \epsilon I)\|^p dA = (1-\epsilon)^p \int_{\Omega \setminus K} \mathcal{G} \|\nabla I\|^p dA < \int_{\Omega \setminus K} \mathcal{G} \|\nabla I\|^p dA.$$

Since we can choose ϵ arbitrarily small, in any neighborhood of I we can find an image $(I - \epsilon I)$ that has a smaller energy, which contradicts the minimality of I. □

Corollary 1 *To reduce the energy $f_{K,1}$ below ϵ, it suffices to reduce $f_{K,\mathcal{G}}$ below $\epsilon\tilde{\mathcal{G}}$, where $\tilde{\mathcal{G}}$ is a lower, positive bound for $\mathcal{G}$.*

Corollary 2 *The limit of any explicit–jump multiscale analysis is the unique segmentation that preserves the integral on each of the Ω_i.*

A strength of our method lies in preserving edges in **two ways**. Wherever we have detected an edge and placed it in K, the enhanced image will be discontinuous. But even if we have missed to put an edge in K, if $\mathcal{G}$ is small near this edge, the enhanced image will be fairly sharp because the minimization favors images with more smoothing in areas where $\mathcal{G}$ is large than in areas where $\mathcal{G}$ is small. This latter effect is called anisotropic diffusion and was introduced in the case $p = 2$ by Perona and Malik; see for example [9]. For $p = 1$ it was later used as "image selective smoothing" by Alvarez, Lions and Morel, [1].

The explicit–jump multiscale analysis behaves very similarly for $p = 1$ and $p = 2$ (images are smooth inside Ω_i, but not across K). This stems directly from our explicit introduction of discontinuous edges. Without discontinuous edges the behavior is very different as outlined in [8]. The benefit is that for $p = 2$ the Euler-Lagrangian of (1) is a linear equation. Fast methods for that particular case will be described elsewhere, in the case of segmentations in [14].

3 Discretization

We introduce a variable for every pixel in the $(m \times n)$ pixel image I_o. The variables are ordered from left to right then top to bottom into the vector $I_h = \{I_{h,l}\}_{l=1}^{mn}$ where $l = (i-1)n + j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. See also Figure 16 a) for an example.

3.1 The edge indicator

The computation of the edge indicator $\mathcal{G}$ requires a smoothed image I_s. We find I_s as follows:

$$I_s = (\mathcal{I} + dt\Delta)^k I_o,$$

where $\mathcal{I}$ is the identity, dt is the time step, k the number of time steps, Δ is a centered difference approximation[5] of the Laplacian with zero Neumann boundary conditions (see Appendix A), I_o is the image with pixels ordered in the above sense and $\sigma = 2\sqrt{kdt}$. Then

$$\mathcal{G}_l = g\left(\sqrt{(D_l^x I_s)^2 + (D_l^y I_s)^2}\right),$$

where D^x and D^y are centered differences with zero Neumann boundary conditions in the x and y directions respectively. Subscript l indicates the pixel where a variable lives or where an operator is to be evaluated, for a matrix this means the row.

3.2 Upwind discretization of the gradient

We discretize ∇I in (1) by upwinding (borrowed from level set methods and ultimately from hyperbolic conservation laws, see [10]) on $-\nabla\mathcal{G} = -(\mathcal{G}_x, \mathcal{G}_y)$, which is found by applying centered differences. Let D^{-x}, D^{-y} be backward differences in the x and y directions with zero Neumann boundary conditions,

[5] All difference approximations of differential operators need to comply with our pixel ordering. As linear operators on vectors, they can be realized as sparse matrices of dimension $mn \times mn$.

respectively, and D^{+x}, D^{+y} forward differences in the x and y directions with zero Neumann boundary conditions. Define the upwind gradient

$$\nabla^{up} I_h = (D^{xup} I_h, D^{yup} I_h)$$

where[6]

$$D^{xup} = \mathrm{diag}(\chi_{(-\infty,0]}(\mathcal{G}_x))D^{-x} + \mathrm{diag}(\chi_{(0,\infty)}(\mathcal{G}_x))D^{+x},$$
$$D^{yup} = \mathrm{diag}(\chi_{(-\infty,0]}(\mathcal{G}_y))D^{-y} + \mathrm{diag}(\chi_{(0,\infty)}(\mathcal{G}_y))D^{+y}.$$

Here $\mathrm{diag}(V)$ is the $mn \times mn$ matrix with the elements of the mn vector V on its diagonal and χ_Ω is a characteristic function, i.e.

$$\chi_\Omega(\zeta) = \begin{cases} 1 & \text{for } \zeta \in \Omega, \\ 0 & \text{for } \zeta \notin \Omega. \end{cases}$$

The above definitions introduce an arbitrary preference for backward differences where G is constant. This could be avoided by using centered differences in that case, but centered differences have the drawback of widening the stencil and decoupling the center point.

3.3 Discretization of the energy

The integral in (1) is discretized with:

$$F(I_h) = \delta A \sum_l f_l(I_h) = \delta A \sum_l \mathcal{G}_l \|\nabla_l^{up} I_h\|^p. \tag{2}$$

Here δA is the area of a single pixel. We ususally assume $\delta A = 1$.

For a descent method for (2) we need the gradient with respect to the discrete variables

$$\nabla_I F = \sum_l \nabla_I f_l, \tag{3}$$

where

$$\nabla_I = \left\{ \frac{\partial}{\partial (I_h)_l} \right\}_{l=1}^{mn}.$$

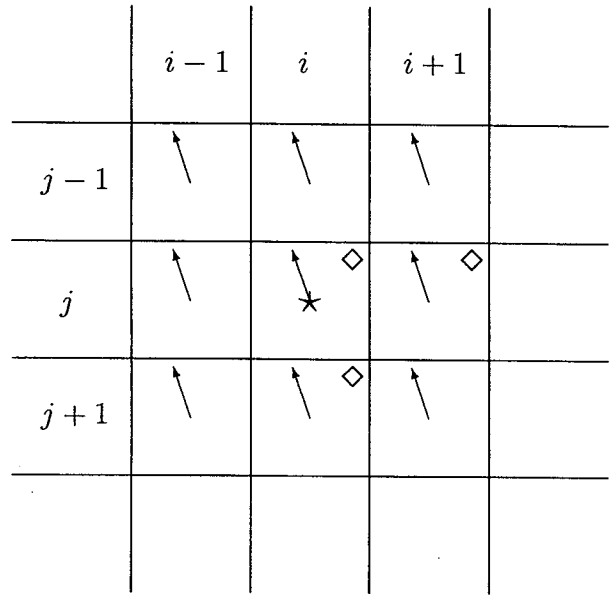


Figure 1: Upwind scheme for the function f_l. Arrows represent the vector field $-\nabla\mathcal{G}$, $\star$ indicates the location where f_l lives and $\diamond$ marks the stencil of f_l.

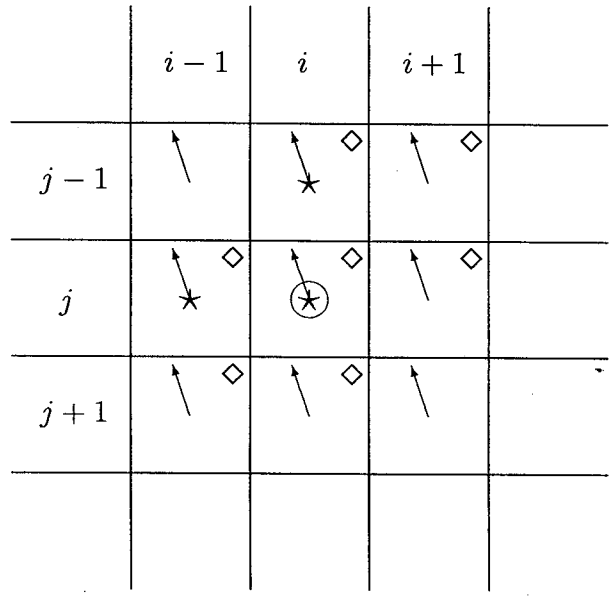


Figure 2: Stencil of the discrete gradient (4) away from edges. $\bigcirc$ is the point of computation of the gradient, $\star$ is the location of the f_l that depend on the value of I at $\bigcirc$. $\diamond$ indicate the stencils of the f_l. The stencil of the gradient is the union of $\diamond$.

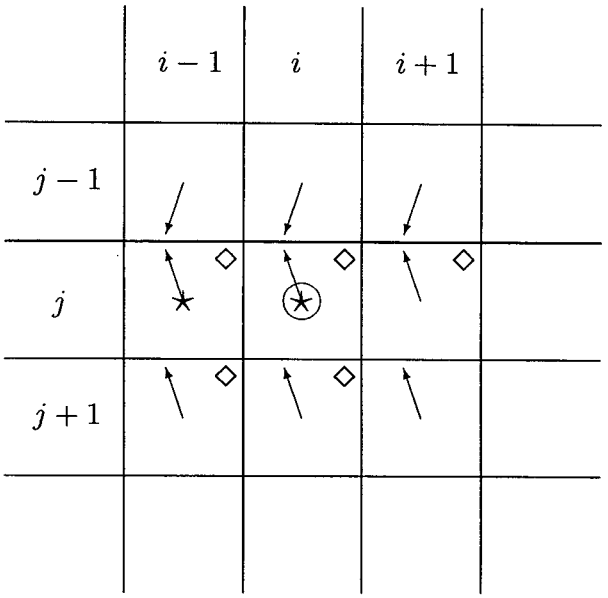


Figure 3: Stencil of the discrete gradient close to an edge (bold line). The notation is the same as in Figure 2.

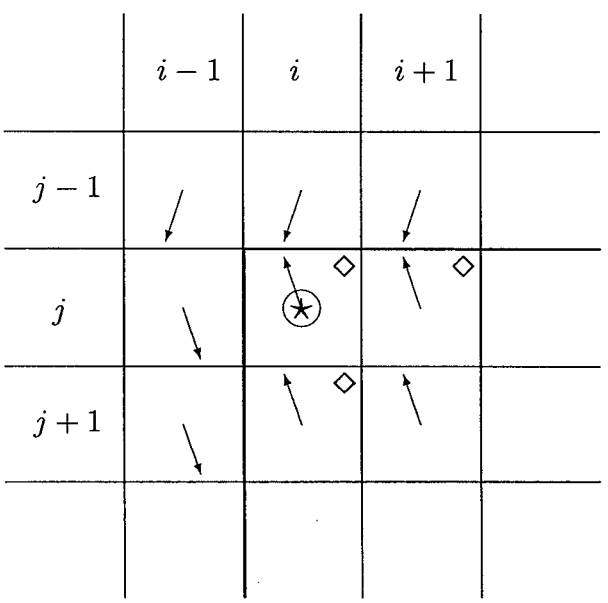


Figure 4: Stencil of the discrete gradient close to a corner (bold line). The notation is the same as in Figure 2.

Each nonlinear discrete function f_l depends only on the value of the lth pixel and its 4 spatial neighbors, that allows to rewrite (3) as local conditions

$$\nabla_{I,ij} F = \frac{\partial f_{i+1,j}}{\partial I_{i,j}} + \frac{\partial f_{i-1,j}}{\partial I_{i,j}} + \frac{\partial f_{i,j+1}}{\partial I_{i,j}} + \frac{\partial f_{i,j-1}}{\partial I_{i,j}} + \frac{\partial f_{i,j}}{\partial I_{i,j}} \tag{4}$$

for $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$ where we have written the formulas in 2D pixel coordinates for clarity. Each derivative is evaluated analytically (see Appendix A) or approximated by finite differences as

$$\frac{\partial f_*}{\partial I_{i,j}} \approx \frac{f_*(I_{i,j} + \epsilon) - f_*(I_{i,j})}{\epsilon}.$$

The approximation has the benefit of being fast and automatic, independently of the choice of p in (1). The loss of accuracy using the above approximation compared to the analytic derivatives in Appendix B was negligible.

Upwinding deals with the edge set K in an automatic way. Since no difference is taken across K, we do not require any smoothness across K; jumps are explicitly allowed and discontinuous edges are possible. Figure 1 shows the stencil for f_l at the pixel $\star$. Recall that $f_l = \|\nabla_l^{up} I_h\|^p$ and note that the $\diamond$ symbols indicate the stencil for the upwind gradient and hence for f_l. In Figure 2, the $\bigcirc$ symbol marks the location where we compute $\nabla_{I,l} F$. The $\star$ symbols mark the pixels such that f_l at these pixels depends on the value of I_h at $\bigcirc$ and the $\diamond$ symbols indicate the union of the stencils for these f_l, i.e. the stencil for $\nabla_{I,l} F$ in (4) at the point $\bigcirc$. In Figure 3 and Figure 4, the notation is the same. The value of $\nabla_{I,l} F$ depends on fewer pixels, because fewer f_l depend on the $\bigcirc$ pixel. **Discontinuous edges are possible because by construction the computation of $\nabla_{I,l} F$ does not involve pixels on the other side of an edge.**

[6]The upwind direction needs to be computed slightly differently at the left and upper image boundaries.

3.4 Numerical Scheme

As the simplest example of an explicit–jump multiscale analysis of (1) we flow (4), which correponds to the steepest descent method for (2)

$$\frac{\partial I_{i,j}}{\partial t} = -\left(\frac{\partial f_{i+1,j}}{\partial I_{i,j}} + \frac{\partial f_{i-1,j}}{\partial I_{i,j}} + \frac{\partial f_{i,j+1}}{\partial I_{i,j}} + \frac{\partial f_{i,j-1}}{\partial I_{i,j}} + \frac{\partial f_{i,j}}{\partial I_{i,j}}\right). \tag{5}$$

The stencil of (5) is space-variant (but not time-variant) and depends on the upwinding direction $-\nabla\mathcal{G}$, as explained before.

The formulation of (1) as a discrete minimization problem allows flexibility and many choices of optimization method different from (5).

4 Examples

All examples are computed based on (5) with $p = 1$ unless otherwise noted. Δt and k in the calculation of the smoothed image I_s vary between examples.

4.1 Deblurring

Figure 5 a) shows a piecewise constant gray-scale image after strong blurring. Figure 5 b) shows the effect of 15 iterations of explicit–jump multiscale filtering. Two effects are noticable. Edges are mostly sharp and pixelized, but a little more blurred where edges intersect. The first effect is created by the fact that edges exist between pixels and hence are always horizontal or vertical. The second effect demonstrates that even when we "miss" an edge, the small values of $\mathcal{G}$ lead to "slow leaking" (using anisotropic difusion) across the edge. Figure 6 shows the edge–indicating vector field computed from a section of the image in Figure 5 a), but shown on the original (unblurred) image. Arrows correctly point towards edges in the original.

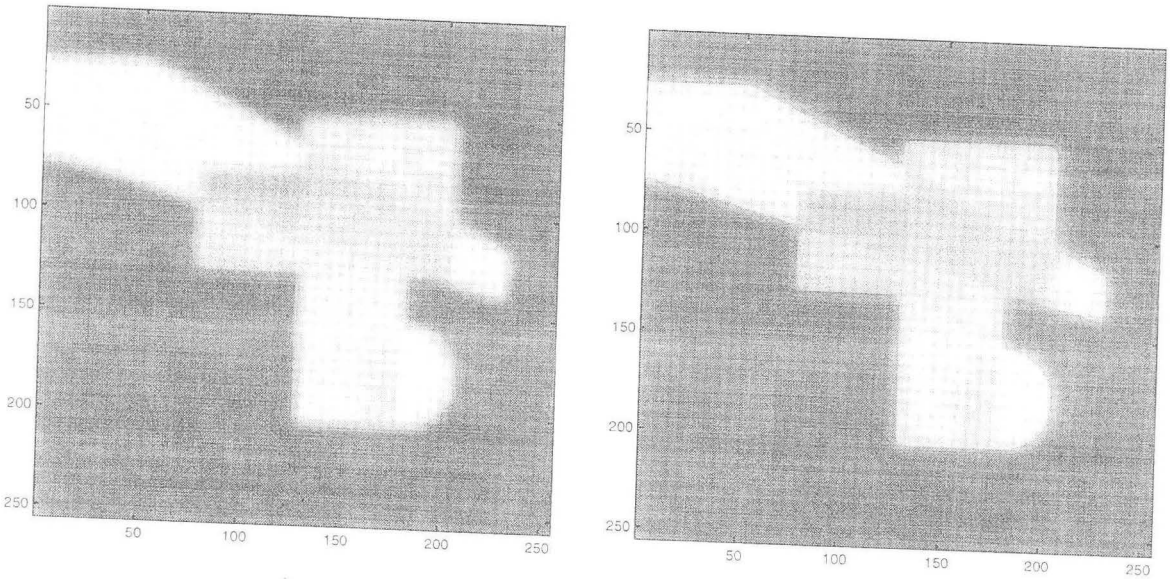


Figure 5: *a) A pieccwise constant image after strong blurring. b) The image from a) after 15 iterations explicit jump multiscale filtering.*

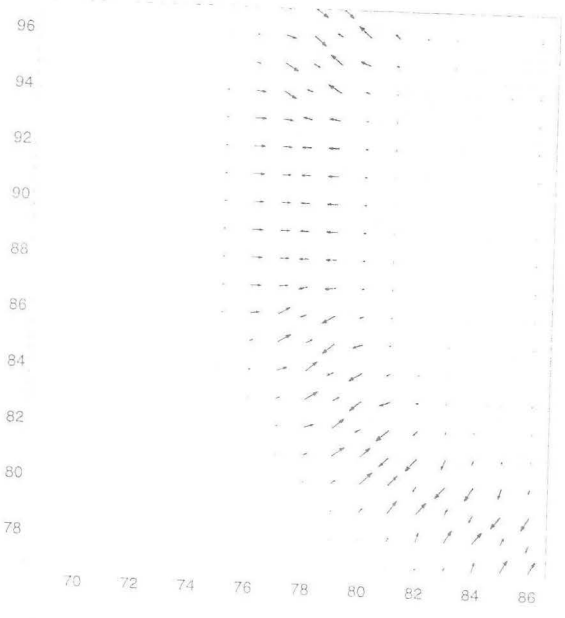
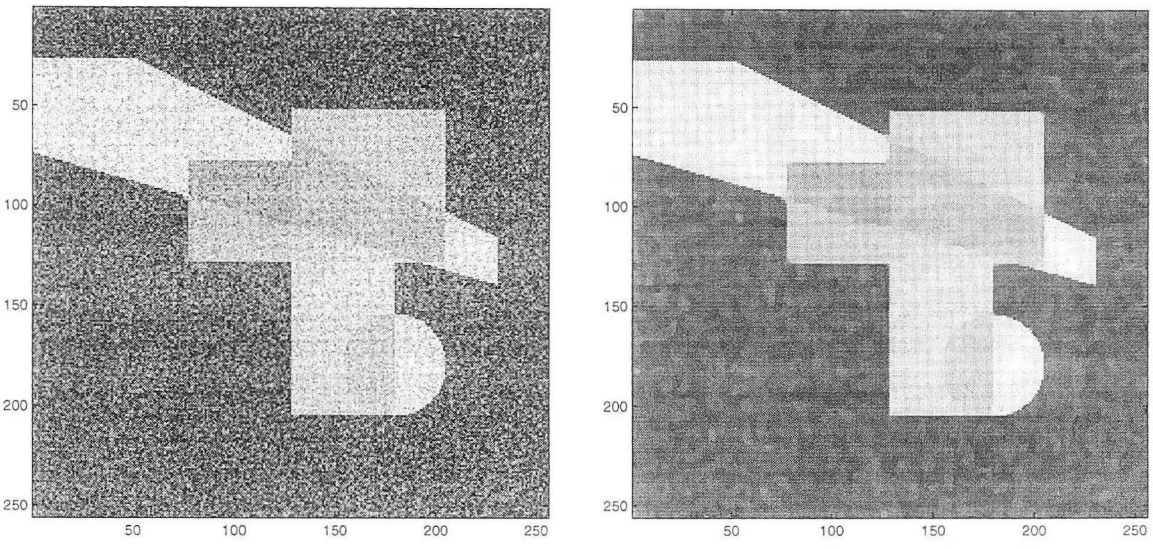


Figure 6: *The edge-indicating vector field $-\nabla \mathcal{G}$ (arrows) is computed from $\mathcal{G}$ based on the blurred image, but shown on the original (unblurred) image. Arrows correctly point towards edges in the original.*



a) b)

Figure 7: *a) The same piecewise constant image as in Figure 5 a), but in the presence of zero-mean Gaussian noise instead of blurring. The image from a) after explicit–jump multiscale filtering.*

4.2 Noise removal

Figure 7 a) shows the same piecewise constant gray-scale image with additive zero-mean Gaussian noise. Explicit–jump multiscale filtering produces Figure 7 b). The image is very similar to the one obtained from the blurred image. This is quite intuitive, since the filtering smoothes the noisy image with a heat equation ("blurs") first, before computing the edge map, so that the edge maps are similar except for edges induced by noise in region interiors.

Figure 8 shows how the p-energy with respect to K and $\mathcal{G}$ is decreased during the filtering from Figure 7 a) to Figure 7 b). We see a typical steepest descent behavior, with vast initial improvements and slow convergence later.

4.3 An absolute stopping criterion

The explicit jump discretization allows to separately measure the energy in smooth regions and the energy contained on the edges. As an example,

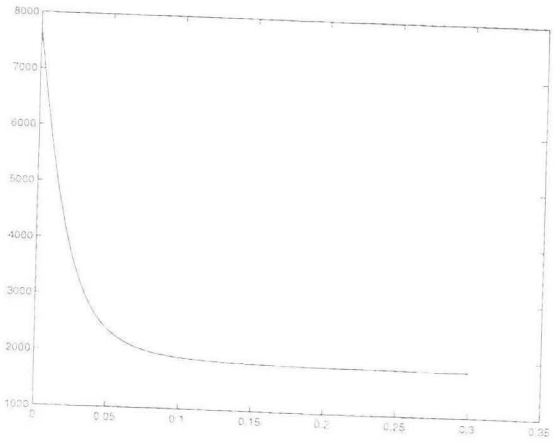


Figure 8: *The decrease of the* 1*-energy with respect to* K *and* $\mathcal{G}$ *for the 15 steps of explicit-jump multiscale filtering from Figure 7a) to b).*

consider

$$I(x,y) = \begin{cases} 0.5(x-0.5) + 0.8(y-0.5) + 0.5 & \|(x-0.5, y-0.5)\|_2 < 0.3 \\ 0 & \text{otherwise} \end{cases},$$

defined in the unit square $(x,y) \in [0,1] \times [0,1]$ as shown in Figure 9. The image has edge energy $0.3\pi \approx 0.9425$ and interior $\Omega \setminus K$-energy $\sqrt{0.25 + 0.64}$ $\pi 0.09 \approx 0.2667$ (on $\|(x-0.5, y-0.5)\|_2 \neq 0.3$). We discretize the unit square with 128 pixels in each direction, so the width and height of each pixel is $h = 1/128$. For this discretization, the calculation of $\int_{\Omega\setminus K} \|\nabla I\| dA$ via $\sum_{i,j} h^2 \| \left(\nabla^{up} I\right)_{ij} \|_2$ yields 0.2670, which is correct to three digits.

Next, we show how this may be used to determine a stopping criterion in the presence of noise, without a priori knowledge. Figure 10 a) and b) show two noise-free images with different geometric but same texture properties. The $\Omega \setminus K$-energies are approximately 4.58 (a) and 5.29 (b).

Figure 11 a) and Figure 12 a) show noisy images where the clean image has the same statistical properties as the previous clean images. The difference is the variance of the zero-mean Gaussian noise (0.1 versus 0.2), which means that the peak noise to peak signal ratio

$$\frac{\max(\text{ noise }) - \min(\text{ noise })}{\max(\text{ signal }) - \min(\text{ signal })}$$

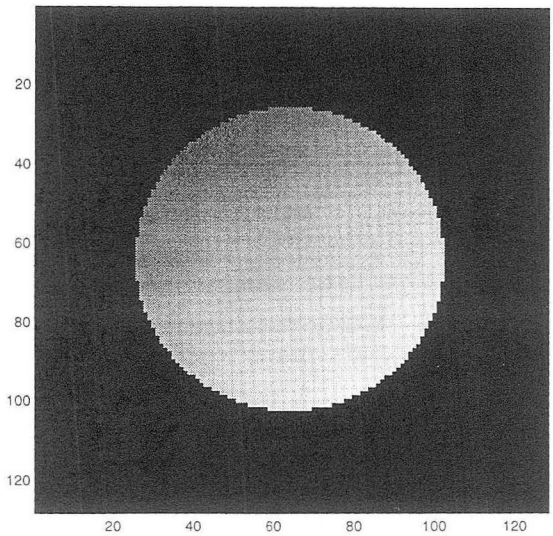
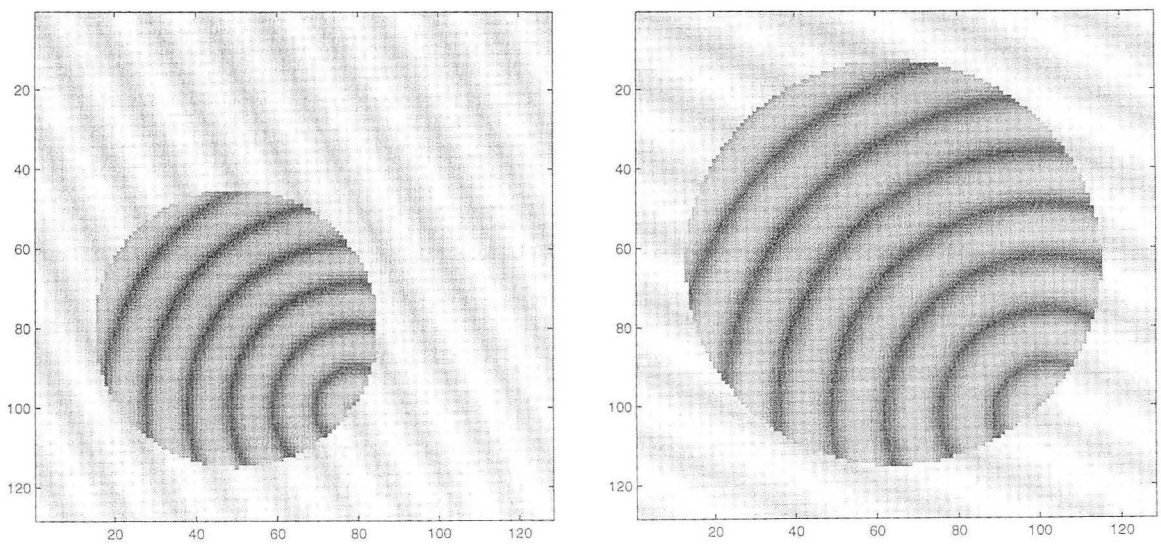
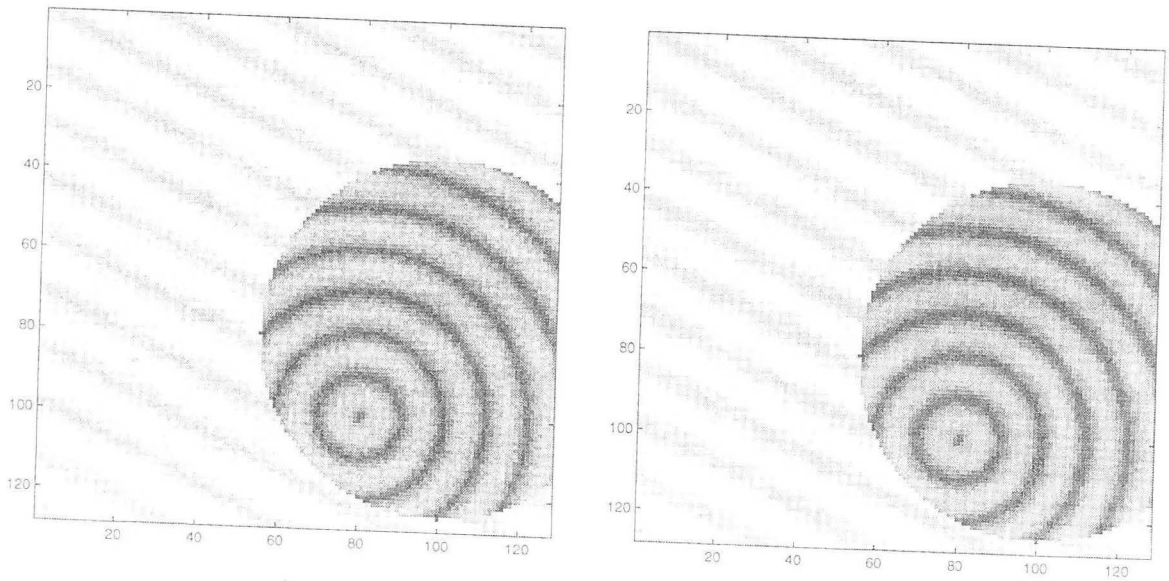


Figure 9: *A piecewise linear gray-level function with edge-energy* 0.9425 *and* $\Omega \setminus K$ *energy* 0.2667, *see text for details.*



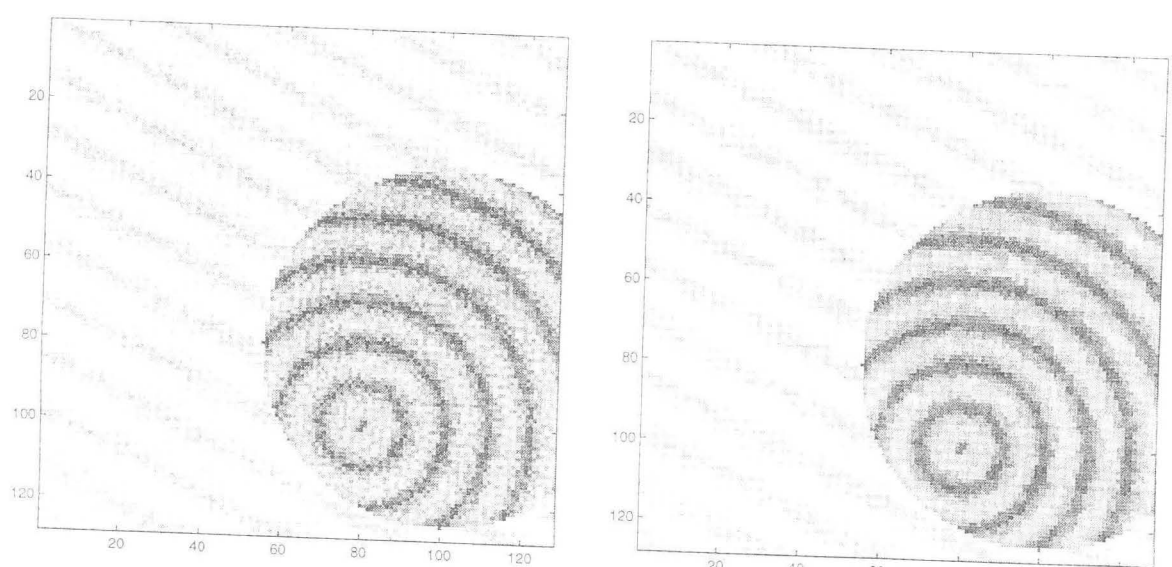
a) b)

Figure 10: *Noise-free images with different geometric but same texture properties. The average of their* $\Omega \setminus K$ *energies is used as an absolute stopping criterion for the examples in Figure 11 a) and Figure 12 a)*



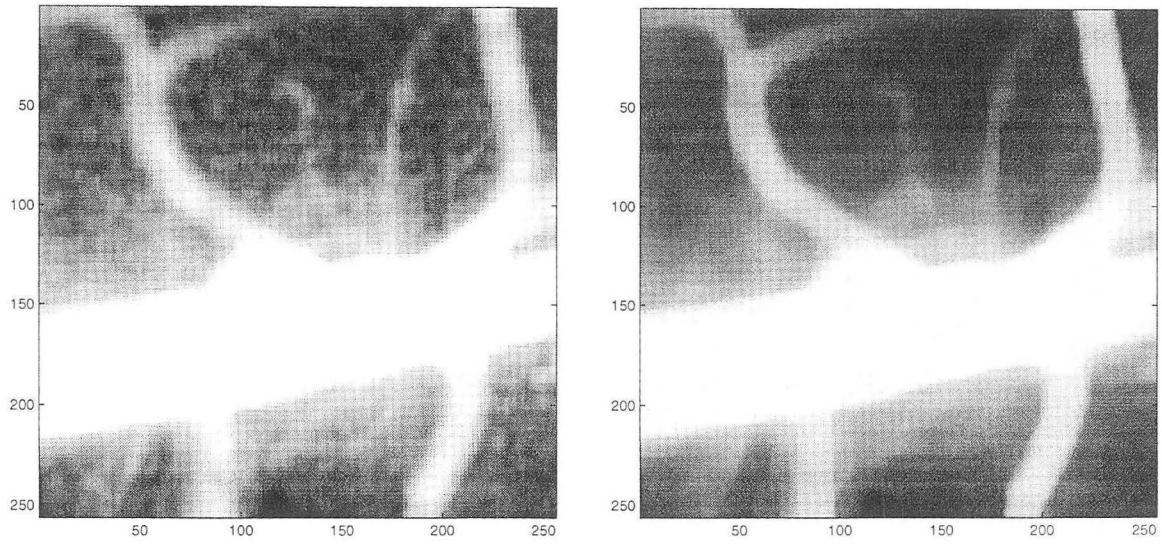
a) b)

Figure 11: *a) Noisy image with 30% peak-noise-to-peak-signal ratio. b) Image after explicit-jump multiscale filtering, stopping when the $\Omega \setminus K$ energy is close to the energy of the clean images in Figure 10.*



a) b)

Figure 12: *a) Noisy image with 60% peak-noise-to-peak-signal ratio. b) Image after explicit-jump multiscale filtering, stopping when the $\Omega \setminus K$ energy is close to the energy of the clean images in Figure 10.*



a) b)

Figure 13: *a) High quality angiography (an artery). b) The image in a) after 500 iterations of shock filtering.*

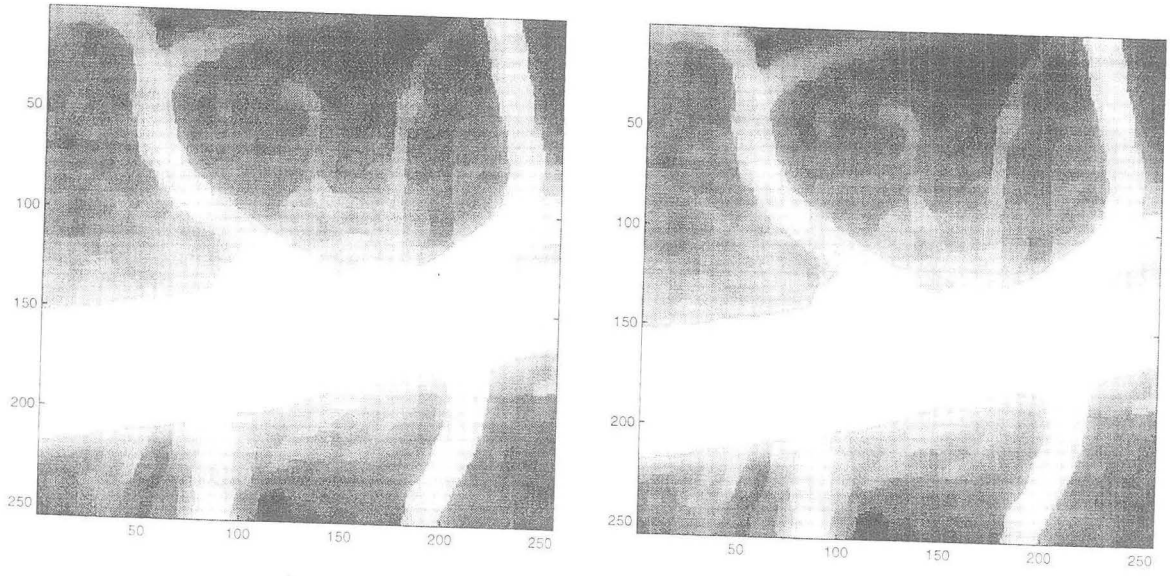
is 30% and 60%, respectively.

The noise increases the $\Omega \setminus K$ energy, where of course $\mathcal{G}$ and hence K is computed based on the noisy image. We reduce the energy to the average of the energies of the clean images in Figure 10 and recover the images in Figure 11 b) and Figure 12 b), respectively.

Independent of the amount of noise in the two examples we use the same, absolute, stopping criterion, which only depends on "statistical" properties, i.e. a priori information that we have deduced from similar, clean images.

4.4 Shock filtering, L^1 and L^2 explicit–jump multiscale filtering.

Figure 13 a) shows a high quality angiography, Figure 13 b) the results of 500 iterations of shock filtering ([1]) with largest stable time-step. Figure 14 a) and b) show the results of 30 iterations of explicit–jump multiscale filtering in L^1 ($p = 1$) and L^2 ($p = 2$). The edges are even sharper and found much more quickly than for the shock filtering, while the results are fairly independent of p.



a) b)

Figure 14: *a) The image from Figure 13 a) after 30 iterations of explicit-jump multiscale filtering with $p = 1$. b) The image from Figure 13 a) after 30 iterations of explicit-jump multiscale filtering with $p = 2$.*

4.5 Future Improvements

The reason slightly blurry regions occur for example in Figure 5 b) lies in the unreliability of $\nabla \mathcal{G}$ near edge crossings, usually referred to as T-junctions. The vector field near a T-junction is shown as arrows in Figure 15 and is seen to be pointing correctly towards the edges away from the T-junction, but is not exact at the T-junction. Possible approaches to improve this may lie in using gradient vector flow ([15]) or subpixel resolution to improve the quality of the vector field.

Another improvement lies in using the magnitude and not just the direction of $-\nabla \mathcal{G}$. For example in Figure 10 a) and b), edges are detected not just on the boundary of the disk, but also between stripes. The magnitude of $-\nabla \mathcal{G}$ could be used in this case to exclude the edges.

Based on $p = 2$, an extremely fast segmentation algorithm is in preparation [14].

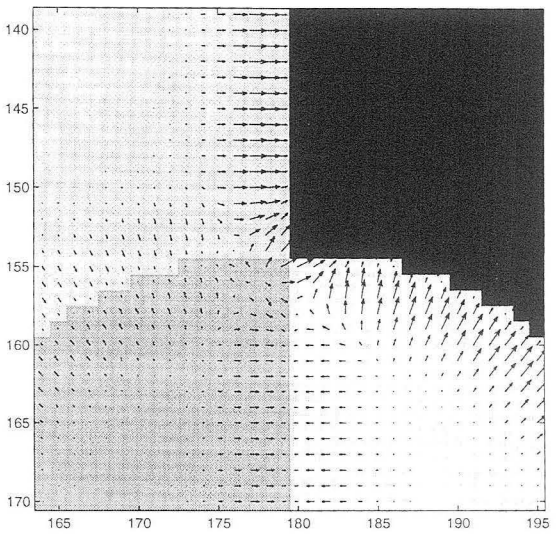


Figure 15: *The edge-indicating vector field $-\nabla \mathcal{G}$ (arrows) near the intersection of two edges. Arrows correctly point towards edges, except close to the intersection.*

5 Conclusions

We introduced a method to perform image filtering by considering edges as image discontinuities. Spatially varying difference operators explicitly allow jumps in the image. This yields enhanced images with extremely sharp edges after very few iterations. The separation of energy into edges and smooth regions allows the use of an absolute stopping criterion, independent of the amount of noise in the original image.

We believe that explicit jumps can be built into many image processing schemes and may be important in the future in devising fast pde based methods.

References

[1] L. Alvarez, P. L. Lions, and J. M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, pages 845—866, 1992.

[2] J.J. Koenderink. The structure of images. *Biol. Cybern.*, pages 363—370, 1984.

[3] P. Liang and Y. F. Wang. Local Scale Controlled Anisotropic Diffusion with Local Noise Estimate for Image Smoothing and Edge Detection. In *International Conference on Computer Vision, Bombay, India*, January 1998.

[4] R. Malladi and J. A. Sethian. Image processing: Flows under Min/Max curvature and mean curvature. *Graphical Models and Image Processing*, 58:127—141, 1996.

[5] R. Malladi and J. A. Sethian. 'Level set methods for curvature flow, image enchancement, and shape recovery in medical images. In H. C. Hege and K. Polthier, editors, *Visualization and Mathematics: Experiments, Simulations, and Environments*, pages 329—345. Springer Verlag, 1997.

[6] D. Mumford and J. Shah. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics*, 42(4):577—685, 1989.

[7] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Num. Anal.*, 27:919—940, 1990.

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

[9] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. PAMI*, 12(7):629—639, July 1990.

[10] J. A. Sethian. *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and material science.* Cambridge University Press, 1996.

[11] C.R. Vogel and M.E. Oman. Iterative methods for Total Variation Denoising. *SIAM J.Sci.Comput.*, 17(1):227—238, 1996.

[12] A. Wiegmann. *The Explicit–Jump Immersed Interface Method and Interface Problems for Differential Equations.* PhD thesis, University of Washington, 1998.

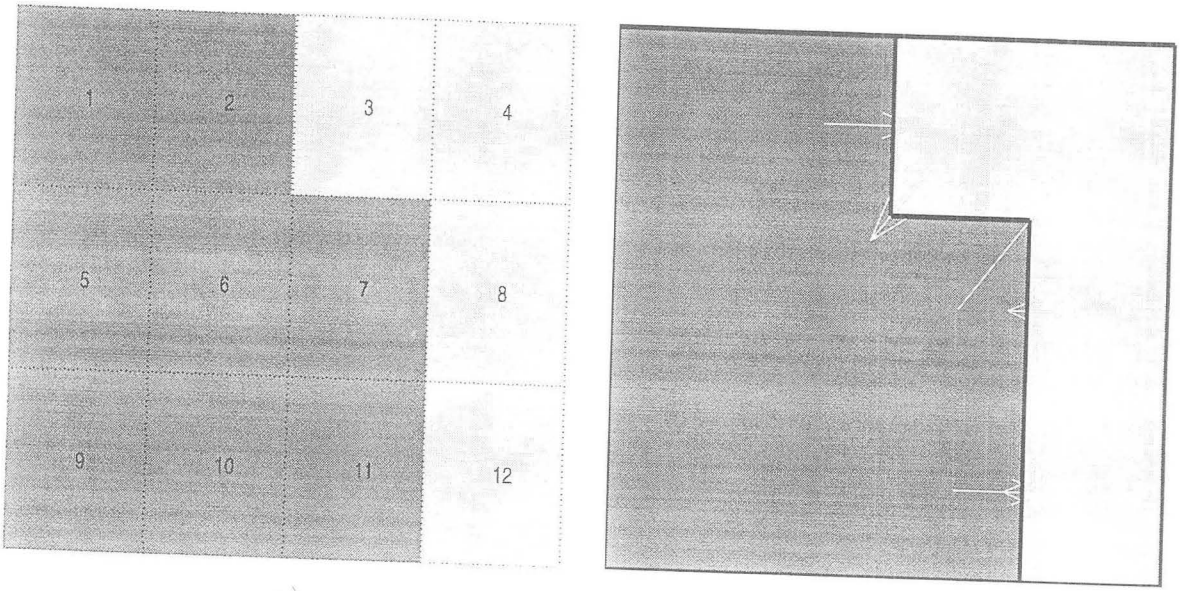
[13] A. Wiegmann and K. P. Bube. The Explicit–Jump Immersed Interface Method: Finite Difference Methods for PDE with piecewise smooth solutions. To appear in *SIAM J. Num. Anal. and appeared as preprint in: Seminarberichte aus dem Fachbereich Mathematik, Vol. 62, Part 2, FernUniversität Gesamthochschule Hagen*, October 1997.

[14] A. Wiegmann and A. Sarti. Fast image segmentation. Work in progress, 1998.

[15] C. Xu and J.L. Prince. Snakes, Shapes and Gradient Vector Flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

A Finite differences as sparse matrices

We implement a finite difference approximation of a differential operator acting on an image as the application of a matrix to a vector. In addition to the stencil, we need to know the ordering of the entries of the image into the vector and the boundary conditions. For example, consider the image from Figure 16. The entries are ordered in the convention of the C programming language. Let h denote the mesh spacing, usually 1.



a) b)

Figure 16: *a) 3×4 gray-value image with pixel ordering. b) Edges (black lines) and non-vanishing portions of $-\nabla \mathcal{G}$ (white arrows) computed with almost no smoothing for the image in a).*

a) Implement $\partial I / \partial x$ by forward differences, with Neumann boundary conditions affecting the discretization at the right boundary:

$$B = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad D^{+x} = \frac{1}{h} \begin{pmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{pmatrix}.$$

In general, for an $m \times n$ image B is an $n \times n$ matrix with -1 on the diagonal, 1 above the diagonal and zero last row. D^{+x} is an $mn \times mn$ matrix and has m blocks B on the diagonal.

b) Implement $\partial I / \partial y$ by backward differences, with Neumann boundary conditions affecting the discretization at the top boundary:

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, D^{-y} = \frac{1}{h} \begin{pmatrix} 0 & 0 & 0 \\ -B_1 & B_1 & 0 \\ 0 & -B_1 & B_1 \end{pmatrix}.$$

In general, for an $m \times n$ image B_1 is the n-dimensional identity matrix. D^{-y} is an $mn \times mn$ matrix with n rows of zeros, $m-1$ blocks B_1 on the diagonal and $m-1$ blocks $-B_1$ below the diagonal.

Backward differences in x and forward differences in y are analoguous. These differences can be combined into higher order differences, for example $(D^{-x})^T D^{-x} + (D^{-y})^T D^{-y} = \Delta$ (all difference operators discretize differential operators as well as first order zero Neumann boundary conditions).

The discretization of the upwind operators for the simple image in Figure 16 a), based on $\mathcal{G}$ obtained with minimal smoothing and $-(\mathcal{G}_x, \mathcal{G}_y)$ as shown as white arrows[7] in Figure 16 b) is given below. Some features are notable: Rows 1,5 and 9 of D^{xup} (rows 1,2 and 4 of D^{yup}) use forward differences instead of backward differences even though $\mathcal{G}_x = 0$ ($\mathcal{G}_y = 0$). This is due to the modification of the upwind operator at the boundaries. At the 8 and 12 pixels, no difference in x is taken. No difference in y is taken at the 3 pixel.

$$D^{xup} = \begin{pmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

[7]We use zero Neumann conditions in the calculation of $-\nabla\mathcal{G}$.

$$D^{yup} = \begin{pmatrix}
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

B Analytic derivatives

The analytic derivatives below can be used quite efficiently using sparse matrix techniques.

B.1 The gradient for $p = 2$

$$f_{K,\mathcal{G}}(I) = \int_{\Omega \setminus K} \mathcal{G} \|\nabla I\|^2 dA,$$

so

$$F(I) = \sum_l \mathcal{G}_l \left\{ \left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2 \right\}.$$

Hence

$$\begin{aligned}
\frac{\partial F}{\partial I_k}(I) &= \sum_l \mathcal{G}_l \frac{\partial}{\partial I_k} \left\{ \left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2 \right\} \\
&= 2 \sum_l \mathcal{G}_l \left\{ \left(D_l^{xup} I\right) D_{lk}^{xup} + \left(D_l^{yup} I\right) D_{lk}^{yup} \right\},
\end{aligned}$$

so

$$\nabla_I F(I) = 2A_n(I)^T \mathcal{G}, \tag{6}$$

with

$$A_n(I) = \mathrm{diag}(D^{xup} I) D^{xup} + \mathrm{diag}(D^{yup} I) D^{yup}.$$

Suppose there are no edges and we set $\mathcal{G} = 1$. Then $\mathcal{G}_x = \mathcal{G}_y = 0$ and $D^{xup} = D^{-x}$, $D^{yup} = D^{-y}$. We have $\mathrm{diag}(D^{-x} I)^T \mathcal{G} = \mathrm{diag}(D^{-x} I)\mathcal{G} = D^{-x} I$, $\mathrm{diag}(D^{-y} I)^T \mathcal{G} = \mathrm{diag}(D^{-y} I)\mathcal{G} = D^{-y} I$ and so $2A_n(I)^T \mathcal{G} = 2\left((D^{-x})^T D^{-x} + (D^{-y})^T I\right.$ which is a discretization of $-2\Delta I$. In that case, our method reduces to image processing via the heat equation, as expected from (1).

B.2 The gradient for $p = 1$

$$f_{K,\mathcal{G}}(I) = \int_{\Omega \setminus K} \mathcal{G} \|\nabla I\|^1 dA,$$

so

$$F(I) = \sum_l \mathcal{G}_l \sqrt{\left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2}.$$

Hence

$$\begin{aligned} \frac{\partial F}{\partial I_k}(I) &= \sum_l \mathcal{G}_l \frac{\partial}{\partial I_k} \sqrt{\left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2} \\ &= \sum_l \mathcal{G}_l \frac{\left(D_l^{xup} I\right) D_{lk}^{xup} + \left(D_l^{yup} I\right) D_{lk}^{yup}}{\sqrt{\left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2}}, \end{aligned}$$

so

$$\nabla_I F(I) = A_n(I)^T \left(\left(\mathrm{diag}(A_d(I))\right)^{-1} \mathcal{G} \right), \tag{7}$$

where

$$A_n(I) = \mathrm{diag}(D^{xup} I) D^{xup} + \mathrm{diag}(D^{yup} I) D^{yup}$$

as in the $p = 2$ case and

$$A_d(I) = \left\{ \sqrt{\left(D_l^{xup} I\right)^2 + \left(D_l^{yup} I\right)^2} \right\}_{l=1}^{mn} = \|\nabla^{up} I\|.$$

We deal separately with the case when the denominator A_d is zero.

Ernest Orlando Lawrence Berkeley National Laboratory
One Cyclotron Road | Berkeley, California 94720

Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098