Adaptive Nonlocal Filtering: A Fast Alternative to Anisotropic Diffusion for Image Enhancement

Fischl and Schwartz

July 1997

Technical Report CAS/CNS-97-011

Boston University Center for Adaptive Systems and
Department of Cognitive and Neural Systems
677 Beacon Street
Boston, MA 02215

Adaptive Nonlocal Filtering: A Fast Alternative to Anisotropic Diffusion for Image Enhancement.

Bruce Fischl†

Massachusetts General Hospital

NMR Center
Building 149 (2301)
13th Street
Charlestown, MA 02129

Eric L. Schwartz†*

Dept. Cognitive and Neural Systems
Boston University
677 Beacon Street
Boston, MA 02215
email: eric@thing4.bu.edu
(617) 353-6179

* Author to whom reprint requests should be made.

† Supported in part by the office of naval research (ONR N00014-95-1-0409).

Abstract - *Nonlinear anisotropic diffusion algorithms provide a significant improvement in image enhancement for segmentation, when compared to more traditional linear isotropic filters. However, the excessive computational cost of solving nonlinear PDE's (or applying large iterative filters) on images effectively precludes the use of these methods in real-time vision applications. In previous work, we showed that it is possible to adaptively estimate a "Greens Function Approximator", yielding an image enhancement technique which is about an order of magnitude faster, on a serial machine, than comparable nonlinear diffusion methods. Furthermore, the method is fully parallelizable, unlike the intrinsically serial diffusion methods. In the present paper, we show that an additional order of magnitude improvement is provided by a new image filtering paradigm, which we term nonlocal filtering. The basic idea is to adaptively determine a vector field (a form of skeletonization) of an image which specifies nonlocal application points for an image filter. We show one possible algorithm for determining this offset vector field which works well, and gives results which are comparable to, but two orders of magnitude faster than, nonlinear diffusion methods. Moreover, we show that applying an image filter guided by a vector field can be reduced to conventional local filtering acting on a pixel permutation of the original image. This allows the use of special purpose image filtering hardware and algorithms with no modification. The practical import of achieving high quality image enhancement for segmentation at real time rates, with no modification at all of conventional hardware and software filtering, is highly significant. From a theoretical point of view, the introduction of a geometry driven vector field, separate from the details of filter implementation, seems to provide a novel approach to image processing.*

1. Introduction.

Many early vision systems employ some type of filtering in order to reduce noise and/or enhance contrast in regions that correspond to borders between different objects within an image. The logical extreme of this process is the creation of a piecewise constant image with step discontinuities at region boundaries. This goal is unattainable using linear filtering techniques, as noise reduction blurs the locations of boundaries between regions, sometimes to the point of fusing them.

In order to address this problem, Perona and Malik (Perona and Malik, 1987; Perona and Malik, 1990) introduced a nonlinear version of the diffusion equation previously used by Koenderink and Hummel (Koenderink, 1984; Hummel, 1986) for early visual processing. In this formulation, image intensity is treated as a conserved quantity and allowed to diffuse over time, with the amount of diffusion at a point being inversely related to the magnitude of the intensity gradient at that location. This process produces visually impressive results in terms of the creation of sharp boundaries separating uniform regions within an image, but is computationally expensive (see (Fischl and Schwartz, 1997) or (ter Haar Romeny, 1994) for a more complete discussion of these issues).

Because of the extremely high computational cost of nonlinear diffusion approaches, we recently introduced an adaptive technique which provides an approximate "Greens Function" kernel, transforming the partial differential equation describing diffusion into an integral equation (Fischl and Schwartz, 1996; Fischl and Schwartz, 1997). The advantage of this approach is that the kernel estimator can be trained off-line, since the adaptive process generalized extremely well, and the run-time application of filtering with the approximated "Greens Function" can then be performed in a single-time step. This allows the intrinsically serial PDE diffusion approach to be replaced by a parallelizable integration, yielding roughly an order of magnitude improvement in performance even on a serial architecture.

An alternative approach to nonlinear image enhancement was developed by Nitzberg and Shiota (Nitzberg and Shiota, 1992), whose non-linear filter has excellent performance, comparable to non-linear diffusion methods, and whose theoretical basis includes the notion of "offset filtering" as developed in the present paper.

Specifically, Nitzberg and Shiota introduced an offset term which displaces kernel centers away from presumed edge locations, thus enhancing the contrast between adjacent regions without blurring their boundary. While their technique works well for many images, we have found that the particular offset field that they used does not perform adequately in images which contain edges at different scales, unless it is applied iteratively, a computationally expensive procedure. Since the nonlinear filter combines the "offset" and "filtering" functions in a single (8 parameter) expression, it is difficult to design a filter that performs adequately for a variety of images. Moreover, the resulting application requires large and complex kernels and is therefore still extremely slow, as was the case for the original nonlinear diffusion approaches. The key idea of the present paper is that by separating the estimation of an offset vector field from image filtering per se, we obtain a simpler, more robust and faster class of algorithms. We show in this paper that these "nonlocal" filters have better performance than the original Nitzberg-Shiota method, and provide results comparable to nonlinear anisotropic diffusion methods, with a speed-up of roughly two orders of magnitude. This enables the use of anisotropic diffusion methods, for the first time, in real time vision applications.

Of equal importance, from an implementation point of view, the nonlocal filtering is attractive as it can be carried out as a postprocessing procedure. This allows us to apply the desired filter (e.g. 3x3 median filter) to the original image, and then use the offset vector field, in the form of a look-up table (LUT), to produce the final result by a simple pixel permutation. This technique permits conventional hardware (e.g. fast 3x3 filtering) and/or existing code to be applied, unchanged, to produce results which appear comparable to the much more computationally expensive nonlinear diffusion methods. In addition, the modularization of this method in terms of a separate generalized skeletonization operation, coupled with a simple single scale (but nonlocally applied) image filter, should allow for efficient and easy development of hardware and further improved algorithmic aspects of the procedure.

The remainder of this paper is organized as follows. First, we will briefly summarize the basic idea of offset image filtering below, then outline the Nitzberg-Shiota algorithm, and show an example of a class of images for which their method fails to perform adequately. Finally, we compare the results of the offset filtering to isotropic diffusion, nonlinear anisotropic diffusion as well as the Nitzberg-Shiota filtering on a variety of different images, along with estimates of the runtime of each method.

2. Image Filtering and Displacement Vector Fields.

The purpose of filtering an image is to exchange the intensity value at each pixel for some linear or nonlinear function of its near neighbors, with the intent of producing a pixel value that is more representative of the region in which it lies. In image regions which correspond to the interior of an object this type of filtering produces desirable results. However, pixel values that lie on the border of two regions are not representative of either, but rather of some intermediate value. In this case, instead of calculating a new value for the border pixel using neighboring intensities, it is more effective to use a neighborhood which is offset in the interior direction from the edge, and thus more representative of the interior values. A useful metaphor for this procedure is to imagine that the boundary "repels" the filter, pushing it into the interior of a region.

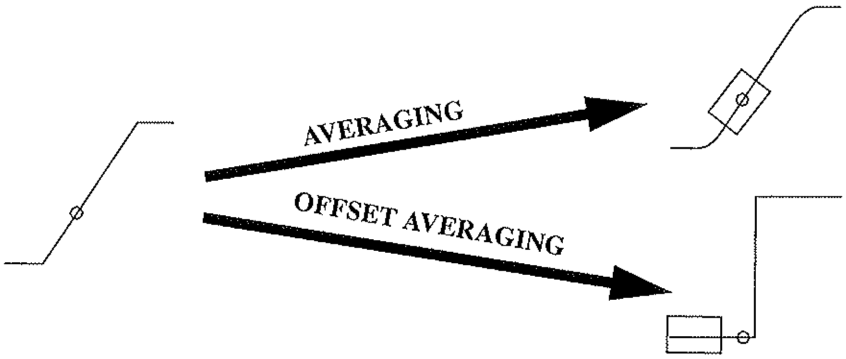


FIGURE 2.1. Example of the utility of a displacement vector for filtering. Left: a simple one dimensional ramp edge. Top right: Gaussian averaging centered on the edge leaves the ramp relatively unaffected. Bottom right: Gaussian averaging displaced values results in a sharp step edge. Boxes show locus of values which contribute to the marked point.

Offset filtering requires the generation of a vector field over the image domain which specifies an appropriate displacement at each point. Intuitively, the displacement direction should be either parallel or antiparallel to the dominant local gradient direction, based on which interior region the point is judged to be a member of. Second derivative (i.e. curvature based) information might be appropriate for this type of decision, but is in general too noisy to be useful when one of the goals is noise reduction. Another approach is the use of the first moment of gradient magnitude. In this formulation, the assumption is that the displacement should be towards the direction which has the smaller moment, and thus presumably the one in which the end of the edge is closer. This approach does not perform well, resulting in blurring of corners, as it ignores gradient direction information which is crucial to making a proper decision, as pointed out by (Nitzberg et al., 1993). Nitzberg and Shiota (Nitzberg and Shiota, 1992) proposed a method based on gradient direction as well as magnitude which performs well in these regions. However, their technique is problematic for images which contain edges at a number of scales, as we will show in the next section of this paper.

3. Nonlinear filtering vs. Anisotropic Diffusion.

The key idea for simplifying computationally intensive approaches to non-linear diffusion came from earlier work, in which we studied the structure of the intermediate kernels produced by the numerical integration of standard non-linear diffusion image segmentation algorithms (Fischl and Schwartz, 1997). In this work it became apparent that most of the detailed structure, produced at great computational cost, of the intermediate diffusion kernels, was not relevant to the final goal of enhancing and segmenting images. It appeared that a few relatively simple forms of kernel, when applied at an appropriate offset relative to image discontinuities, were doing the effective work of the algorithm, while the high-spatial frequency details of kernel structure was basically irrelevant. We exploited this observation by simplifying the final kernels via blurring and principal components analysis. We then employed a non-linear adaptive function approximator to "learn" how to produce an approximation to the final desired non-linear diffusion kernel, using 3x3 Sobel gradient estimates to initialize the learning algorithm. This approach, which obviously could be applied to a wide variety of non-linear diffusion and filtering applications, generalized well and provided about an order of magnitude speed-up. Also, by allowing us to estimate the final kernel, or "Greens Function Appoximator" as we termed it, we changed the iterative, sequential diffusion process into a potentially parallel, one step integration. This work (Fischl and Schwartz, 1997) is reviewed in the appendix to this paper, where the intermediate diffusion kernels, and their relative low-information content, is demonstrated. In the present paper, we show a further exploitation of this idea which has led both to a further order of magnitude increase in speed, and also to what we believe is an important insight into the structure of geometry driven image enhancement algorithms. This insight is that the diffusion process itself, as well as the detailed kernels (filters) it generates, is largely irrelevant to the results achieved. The key action of this class of algorithms appears to consist of two distinct components. The first is the construction of an "offset" vector field, which specifies the location, direction and magnitude at which a filter is to be applied. The second is the nature of the filter itself. We have found that the filter specification is relatively unimportant, from the point of view of image segmentation: Gaussian low pass filters, median filters, and band-pass filters of various characteristics all produce similar results, when applied at the correct vector field locations. The determination of the offset vector field is itself crucial, and we supply a simple and fast algorithm for determining this vector field, which is a form of skeletonization of the image with respect to its implicit edge structure.

The use of offset filters in the context of non-linear diffusion and segmentation, was first suggested by (Nitzberg and Shiota, 1992) This algorithm produces results which compare favorably to those produced by non-linear diffusion methods, but shares with them the problem of being too computationally intensive for use in real-time machine vision. The principle practical contribution of the present algorithm, which will now be presented, is simplification, improved performance, and roughly two orders of magnitude of speed increase. The principle theoretical contribution of the present algorithm is the combination of two different aspects of image processing, i.e. generalized skeletonization in the form of an offset vector field, and conventional image filtering, and the demonstration of what appears to be the core element of contemporary non-linear diffusion algorithms.

3.1. Nitzberg-Shiota Nonlinear Filter.

Nitzberg and Shiota derived a nonlinear image filter (Nitzberg and Shiota, 1992) and an equivalent integro-differential equation formulation (Nitzberg et al., 1993) which yields impressive results. It smooths noise while retaining and enhancing edge and corner information. The nonlinear filter is of the form

$$I(x, y) = \frac{1}{Z(x, y)} \iint_D I(x', y') \, k(x, y, x', y') \, dx' dy' , \qquad (3.1)$$

where *I(x,y)* is the intensity image, *D* is the image domain, *Z(x,y)* is a normalizing term, and *k(x,y,x',y')* is a variable Gaussian kernel which is shaped by the local differential structure of the image. In local coordinates, the kernel function is given by

$$k(x, y) = p_1(x, y)\, e^{-\frac{(Ex^2 + Fxy + Gy^2)}{\sigma^2}} \quad , \tag{3.2}$$

where $p_1(x,y)$ is a Gaussian window, and E, F, and G (which are closely related to the components of the first fundamental form of differential geometry, with the image considered to be a surface embedded in R^3), are defined as

$$\begin{aligned} E &= \iint_U I_x(x', y')^2 p_2(x - x', y - y')\, dx'dy' \quad , \\ F &= \iint_U I_x(x', y')\, I_y(x, y)\, p_2(x - x', y - y')\, dx'dy' \\ G &= \iint_U I_y(x', y')^2 p_2(x - x', y - y')\, dx'dy' \end{aligned} \tag{3.3}$$

with $p_2(x,y)$ a Gaussian window function, I_x and I_y are partial derivatives in the x and y direction respectively, and U is a neighborhood of x,y. Nitzberg and Shiota observe that the shape of the kernel $k(x,y)$ should have the following properties:

- Broad, where the intensity gradient is small, smoothing uniform regions.
- Narrow, in the dominant gradient direction in regions of large coherent gradient to smooth along an edge, but not across it.
- Small and concentrated, where the gradient is large in different directions to preserve corners and triple points.

Figure (3.1) illustrates the form of the Nitzberg-Shiota filter on the edge of a blurred square, with the local image intensities mapped onto the surface of the filter. Note that while the variation of the filter shape minimizes the amount of averaging across the edge, some still occurs, resulting in boundary dislocation and possible degradation of the final edge map.

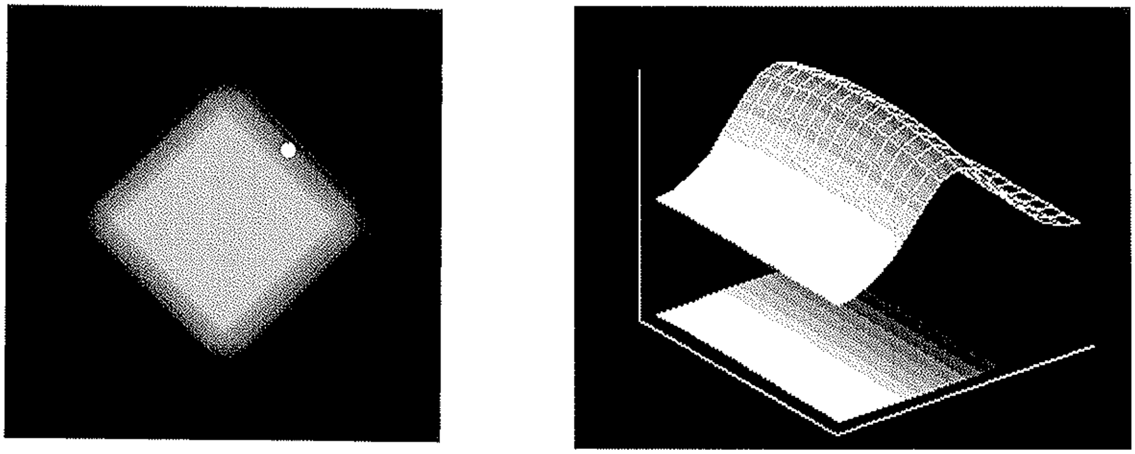


FIGURE 3.1. Example of the form of the Nitzberg-Shiota filter. Left: a blurred square. The dot represents the center of the filter. Right: the form that the filter takes, together with the local neighborhood used to construct it.

4. Offset Vector Field Computation.

The modulation of filter shape aids in the filtering process. However, if the kernel is symmetric around the central pixel some averaging of edge values must occur as shown in figure (3.1). In order to alleviate this problem Nitzberg and Shiota proposed an offset term which "pushes" the center of the kernel away from the point being filtered. The purpose of generating this type of offset vector field is to displace filters away from border areas. We have found that a reasonable means of accomplishing this is to displace in the direction normal to the boundary of a region, as this is the direction with no component along the edge. In order to compute this type of vector field, three issues must be addressed, each of which depends on an estimation of the position and orientation of the local edge, if one exists.

The first is the determination of the normal vector itself. Since the gradient is normal to the level sets of an image, using the gradient direction as an estimation of the normal is a reasonable approach. Once the normal vector has been computed, it must be assigned a sign. That is, a determination must be made as to whether the displacement should be in the direction of increasing or decreasing gradient. This choice reflects a decision as to which region the point in question has been assigned - the region at the "top" of the gradient, or the region at the "bottom". A reasonable criterion for making this choice is to attempt to displace away from the midpoint of the presumed edge location. Finally, once the normal vector and its sign have been fixed, the magnitude of the displacement must be determined. This is a critical decision, as the magnitude must be sufficient to displace the kernel entirely out of the border area, but small enough to avoid displacing it out of small regions representing fine-scale image structure.

The computation of an appropriate offset vector field *v(z)* can therefore be separated into the calculation of three separate quantities, each of which is computed in terms of smoothed image gradients as calculated by a Sobel operator[1].

- Offset *orientation.*
- Offset *direction.*
- Offset *magnitude.*

Symbolically this can be written as[2]

$$v(z) = m(z)\,d(z)\,\frac{O(z)}{|O(z)|}, \quad z = \begin{bmatrix} x & y \end{bmatrix}^T . \tag{4.1}$$

O(z) in equation (4.1) is the offset orientation, and refers to the choice of the normal vector. Offset direction, denoted *d(z)*, is a binary value (1 or -1) corresponding to the choice of a sign for the normal vector. The offset direction term determines whether the offset vector is in the orientation direction or the opposite one (i.e. orientation+π). Finally, offset magnitude *m(z)* encodes the length of the offset vector. In this section we will outline a simple procedure for computing each

[1] We typically use a Gaussian blurring kernel (σ=2) before computing the offset vector field. This results in a vector field which is noise-tolerant and smooth.

[2] Each of these quantities is also a function of the image intensity gradient. We suppress this functional dependence to avoid unnecessary notational clutter.

of these quantities, then show their improved performance relative to the corresponding calculations proposed by Nitzberg and Shiota.

4.1. Offset Orientation.

The orientation of the offset vector at each point in an image should reflect the estimated orientation of the local edge, if one exists. Specifically, we wish the offset orientation to be orthogonal to that of the local edge, and hence normal to the boundary. A simple means for accomplishing this in a manner insensitive to noise is to use the gradient of the smoothed image. Denoting the smoothed image by I, the offset orientation at the point is then given by

$$O(z) = \nabla I(z) . \tag{4.2}$$

4.2. Offset Direction.

Once the offset orientation has been fixed through equation (4.2), the direction of the offset field must be computed, corresponding to the choice of sign of the normal vector. The choice of direction is therefore a binary one, and can be seen as a preliminary, local segmentation decision, reflecting whether the current point has been assigned to the region at the top of the local intensity gradient or to the region at the bottom. Note that this decision is made on a per pixel basis, so it amounts to a form of evidence gathering, and is, in our experience, quite robust over a wide range of images, as illustrated by results presented later in this paper.

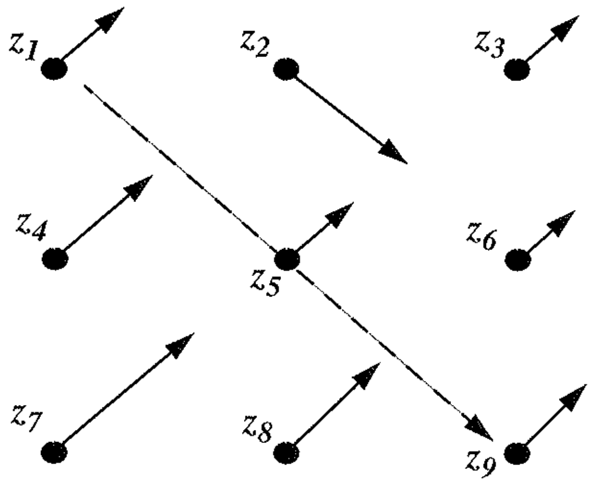


FIGURE 4.1. This figure shows a 3x3 neighborhood of the pixel z_5.The arrows show the gradient computed at each of the neighboring pixels of z_5. The dashed vector shows the estimate of edge orientation at z_5, i.e. the normal to the gradient at z_5, which we will assume is the "correct" edge orientation in that neighborhood. The gradient at pixel z_2 is shown to "disagree" with the support of its neighborhood, perhaps because of a fluctuation due to noise, or a "corner" at that location. The reason that equations (4.2) through (4.7) provide a correct heuristic for computing the offset vector field is then seen in this particular example because: 1.) The gradient at z_2 does not contribute to the neighborhood sum because it is parallel to the estimated local edge orientation and is therefore discounted by the first term in equation (4.3). 2.) Similarly, the gradients at z_1 and z_9 do not contribute because, while they are orthogonal to the local edge, the vectors z_5-z_9 and z_5-z_1 are parallel to the local edge direction and hence do not contribute any information regarding which side of the edge the point z lies on (as measured by the second term in (4.3))). In contrast, each of the gradients at z_4, z_7 and z_8 has an appropriate orientation and contributes evidence that the offset direction should be up and to the right, as the point z_7 lies down and to the left of the point z_5

The direction calculation is intended to determine which side of the center of the local edge the current point is on, as computed using two criteria. The first is that we only wish gradients which are similar to the offset orientation to contribute significantly to the choice of direction. This discounts nearby corner and/or noise-induced gradients which do not contain information about the local point. The second criterion is that gradients of the proper orientation on one side of a point with respect to the offset orientation should contribute towards a displacement in the opposite direction (see figure (4.1)). Combining these two constraints yields the following expression for the calculation of offset direction.

$$d(z) = -\mathrm{sgn}\left(\int_W (O(z) \cdot \nabla I(z+z'))\,(O(z) \cdot z')\,dz'\right), \tag{4.3}$$

where *sgn* is the sign function. The first term in (4.3) reflects the first criterion given above: it weights gradients in the offset direction more strongly than those at other orientations. As noted previously, this term is important as it discounts noise and corner-induced gradients. The second term represents the second constraint, and is similar to computing a first moment. It weights the contribution of a point based on its distance in the direction orthogonal to the estimated local edge. Points with an appropriate gradient (as computed via the first term) that are distant from the central point indicate the presence of an edge in that direction, and hence push the offset in the opposite direction. If *d(z)* is positive, the displacement direction is in the orientation direction, otherwise it is in the opposite direction.

4.3. Offset magnitude.

Once the offset direction and orientation have been fixed, all that remains to be determined is how far the displacement should be in the selected direction. This is a critical decision, as the displacement must be large in broad edges, but small in regions which contain small-scale image structure. In order to satisfy both these constraints we use a one dimensional search mechanism which allows the offset magnitude to grow based on information outside the window *W*. Specifically, we search in the offset direction for a zero crossing of the vector field in that direction (i.e. until the dot product of the offset at the central point with a point in the offset direction is non-positive). This indicates that the vector field has either vanished, signifying the interior of a region, or has changed orientation by at least 90^{0}, possibly indicating the presence of the far edge of the region. The dot product therefore provides a barrier which prevents the offset vector from extending across additional edges in the offset direction.

The offset calculation is therefore a two step procedure. First, an initial offset field is computed using equations (4.2)-(4.3) via

$$v_i(z) = d(z)\,O(z)\,. \tag{4.4}$$

Next, we search in the offset direction for the first point *z*' such that the dot product of the initial offset at *z*' with the initial offset at the central point *z* is non-positive.

$$m(z) = min\ \alpha : v_i(z + \alpha v_i(z)) \cdot v_i(z) \le 0 \tag{4.5}$$

Finally, we form the final vector field using *m(z)* as the magnitude of the initial vector field

$$v(z) = m(z)\frac{v_i(z)}{|v_i(z)|} \tag{4.6}$$

It is worth noting that the adaptive offset magnitude embeds a different notion of scale into offset filtering than is usually used in contemporary applications of diffusion or scale-space architectures. The diffusion formalism grew out of linear filtering techniques such as those of Burt (Burt and Adelson, 1983), Witkin (Witkin, 1983) and Marr (Marr and Hildreth, 1980). In these approaches, the scale of a feature is defined by the size of the kernel required to detect it. In the anisotropic extension of the diffusion paradigm, scale is associated with integration time modulated by local gradient magnitude, and by extension with the distance across which intensity values diffuse to arrive at a given location. Regions of high gradient inhibit the amount of diffusion, and are thus associated with a smaller scale than smoother image areas. The integration of the anisotropic diffusion equation therefore results in intensity values near edges being replaced with smoothed versions of interior intensity values from the direction away from the local edge. In our approach, the relationship between scale and distance is made explicit via the magnitude of the displacement vector at a given location. Larger scale (i.e. more blurred) edges result in longer displacement vectors, but no change in filter size. Conversely, the presence of small scale image features constrains the length of the displacement vectors, preserving the features in question. The smoothing associated with diffusion can then be accomplished using any of a variety of standard fixed-size filters, which are applied nonlocally at the offset location.

4.4. Comparison with Nitzberg-Shiota Kernel Displacement.

Nitzberg and Shiota employ a different technique to calculate an offset vector field which suffers from a number of drawbacks. They use an offset vector of the form $\alpha(x,y)=\phi(v)$ where v is the initial offset vector and ϕ is a vector valued compressive nonlinearity of the form

$$\phi(v) = c\frac{v}{\sqrt{\mu^2 + \|v\|^2}} , \tag{4.7}$$

with c and μ constant. The function ϕ limits the length of the displacement vector, with c specifying the maximum length, and μ modulating the slope of the compression. The initial offset vector v is given in terms of vector offsets $z=(x,y)^T$ and $z'=(x',y')^T$ by

$$v(z) = \int_W \frac{1}{|z'|}(\nabla I(z+z') \cdot z')\nabla I(z+z')p_3(z')dz', \tag{4.8}$$

where p_3 is another Gaussian window. Equation (4.8) bundles the calculation of offset orientation, direction, and magnitude into a single smooth function. This is undesirable for a number of reasons. First, since (4.8) is smooth and changes sign at the midpoint of the edge, the magnitude of v will be small in a neighborhood of the midpoint. Thus, the interior regions of broad edges will have small offset magnitudes, the opposite of what is desired. Furthermore, the calculation of orientation and direction together renders the orientation more sensitive to noise than would otherwise be the case. This is again due to the smoothness of equation (4.8). The zero-crossing of offset magnitude at the midpoint of an edge results in small vectors in that region whose orientation therefore tends to be overly sensitive to noise. This is illustrated in figure (4.1) which depicts a typical situation in the middle of a broad noisy edge. Conflicting direction information causes the

Nitzberg-Shiota algorithm to compute an inappropriate offset orientation, while the average gradient direction given by equation (4.2) yields an appropriately horizontal orientation.

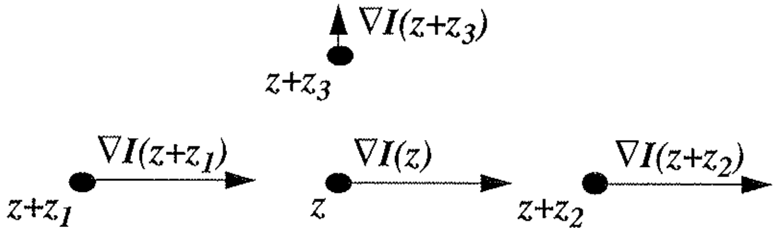


FIGURE 4.2. Example of the superiority of computing offset orientation separately from offset direction. In the Nitzberg-Shiota approach the above situation yields a downward pointing displacement vector. This is due to the simultaneous calculation of orientation and direction. The gradients at $z+z_1$ and $z+z_2$ cancel in the Nitzberg-Shiota approach, while the gradient at z does not contribute at all. Thus, the entire offset vector is a result of the small, possibly noise-induced gradient at $z+z_3$. In contrast, the offset orientation computed from smoothed gradient direction as specified by (4.2) will be only slightly deflected from horizontal.

4.4.1. Offset Magnitude Comparison.

The Nitzberg-Shiota displacement calculation involves similar gradient orientation in opposite directions adding destructively (via the dot product term). This is intended to yield an appropriate offset direction, regardless of direction of contrast, as well as a measure of edge strength in the directions parallel and antiparallel to the local gradient. While the form of their displacement vector computation accomplishes these goals, it also results in regions of broad coherent gradient giving rise to small offset magnitudes. As noted above, this is exactly the opposite of what is desired, as it produces displacement lengths which are insufficient to shift kernels out of this type of edge region.

Figure (4.3) and (4.4) illustrate this effect. Figure (4.3) depicts a synthetic test image in which a small hollow rectangle and a larger circle have been blurred with Gaussian kernels of different standard deviations. The broad shallow gradients at the border of the circle result in Nitzberg-Shiota displacement vectors whose length is close to zero throughout much of the border region, due to the cancellation noted above. Amplifying the lengths of these vectors such that they displace the kernels out of the edge region requires setting the constant c in equation (4.7) to such a large value that the regions in the interior of the rectangle at the right are displaced entirely out of

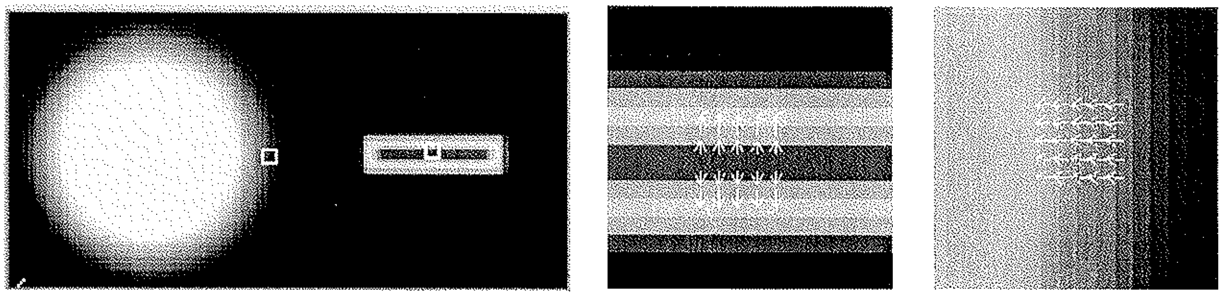


FIGURE 4.3. Test image (left), and blow-up of region in the center of the rectangle (center) and on the right edge of the circle. The arrows represent gradient direction and relative magnitude.

the rectangle. This results in the destruction of the rectangle, while the circle is still only mildly enhanced.

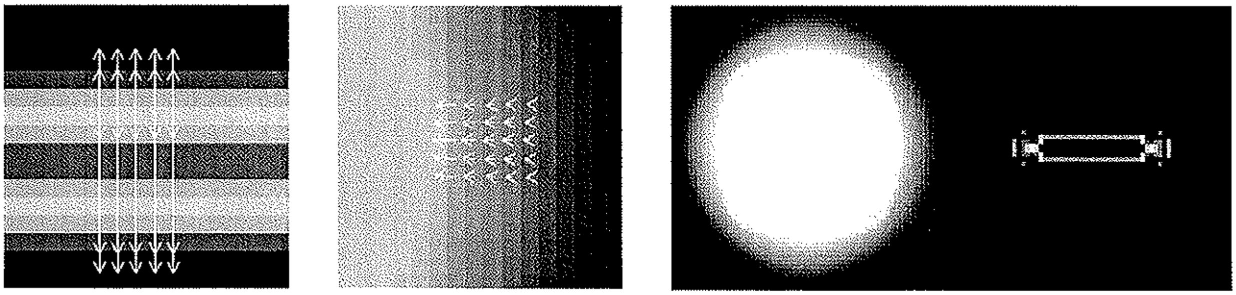


FIGURE 4.4. Nitzberg-Shiota displacement field in same regions of the rectangle (left) and the circle (center). Right: result of filtering using Nitzberg-Shiota displacement vectors and a median filter.

This problem arises due to the relationship between offset vector magnitude and gradient coherence specified by equation (4.8). Similar gradient directions in opposite hemifields (i.e. on opposite sides of the central point with respect to the estimated local edge orientation) should effect the *direction* of the resulting offset, but should not diminish the *magnitude* of such vectors. The length computation in (4.8) is the opposite of what is desired in these cases, producing small offsets near the center of an edge, and larger offsets as the border of the edge is approached. This problem is remedied by the search mechanism described in section (4.3), as it bases the magnitude of the displacement vectors on the spatial scale of the local edge, as opposed to its strength.

The adaptive search technique that we developed thus has a number of significant advantages. The presence of additional edges in the offset direction exert a repulsive force on the vector fields, limiting their lengths, and preserving small image features such as the interior of the rectangle. If no such edges exist, the vector lengths can grow to the extent required to push the filter into a relatively uniform region, as is necessary within broad edges such as the border of the circle. Regions of noise do not have coherent gradient directions, and thus tend to cancel in the offset computation.

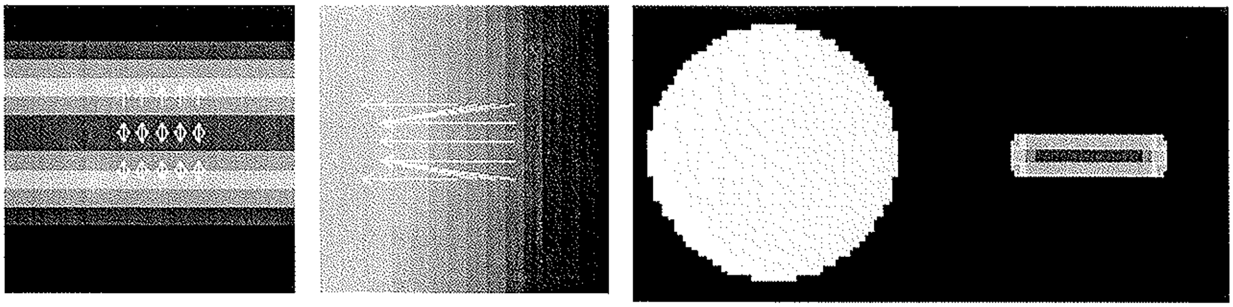


FIGURE 4.5. Vector field computed as described in section (4.1)-(4.3) on the circle (left) and the rectangle (center). Using this vector field in conjunction with a median filter sharpens the circle while retaining the rectangle (right).

The efficacy of this method is illustrated by the right hand image in figure (4.5) which shows the result of median filtering the image in figure (4.3) using a displacement vector field calculated via equations (4.1)-(4.6). The exterior of the circle is sharp and clearly defined, while the interior of the rectangle is preserved.

4.4.1. Offset Orientation/Direction Comparison.

In this section we compare the Nitzberg-Shiota orientation/direction calculation to the one defined in section (4.1). In order to facilitate the comparison, we combine the Nitzberg-Shiota calculation with the magnitude calculation given in section (4.3)[1]. As illustrated by figure (4.1), in the central parts of broad noisy edges the Nitzberg-Shiota calculation becomes sensitive to noise, and their vector field loses coherence. This situation is again depicted in figure (4.5), which displays a noisy, blurred circle at the left, as well as the Nitzberg-Shiota and the proposed offset fields in the central and right-hand images respectively. Each of the offset fields is superimposed over the result of filtering the region specified by the box in the left-hand image with a Gaussian filter applied at locations specified by the given vector field. Note the breaking down of coherence in the Nitzberg-Shiota vector field depicted in the central image, resulting in a jagged border in the filtered image. In contrast, the right-hand image illustrates the robustness of the separate orientation/direction/magnitude vector field calculation given in sections (4.1)-(4.3), which retains coherence even at this noise level, and results in a smooth border in the filtered image.

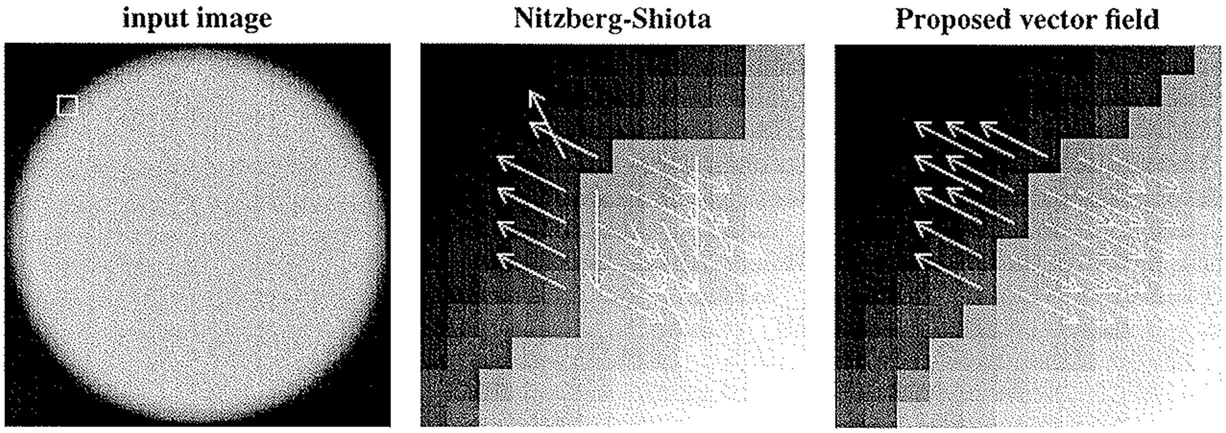


FIGURE 4.6. Comparison of offset orientation/direction calculation on the border of a noisy, blurred circle. Left: input image. The center and right-hand images show the Nitzberg-Shiota offset field and the proposed offset field in the region specified by the box in the left-hand image. Each of the vector fields is superimposed over the result of applying a Gaussian filter at the offset locations to the specified region. Note the loss of coherence in the Nitzberg-Shiota field in the central image which leads to distortions in the border of the circle, while the proposed vector field in the rightmost image retains its coherence at this noise level, yielding a smooth border.

5. Postprocessing with displacement vector fields.

The most straightforward implementation of offset filtering is to apply the displaced filter directly to the (remote) pixel neighborhood. However, filtering with a displacement vector field can be formulated in a different way which greatly simplifies both software development, and potential hardware implementation of this process. The offset filter process outlined in section (4) is identical to filtering an image without a displacement vector field, then using the displacement

[1] This is necessary as without this procedure the Nitzberg-Shiota vector field would be vanishingly small in the entire edge region.

vectors to shuffle the positions of the image intensity values.[1] In this way, the value at each point in the filtered image is replaced with the value at the location specified by the displacement vector field. This process is illustrated in figures (5.1) and (5.2). In figure (5.1) a point I_0 and its neighboring intensity values I_i, i=1...8 (upper left) are transformed into a new set of intensity values J_0...J_8 (upper right) via a filter function f, which could be any standard filter such as the mean or median. After the filtering is complete, the offset vector field at the lower left is used to direct a permutation of the filtered intensity values, resulting in the final neighborhood at the lower right. An example of this process on a real-world image is given in figure (5.2).

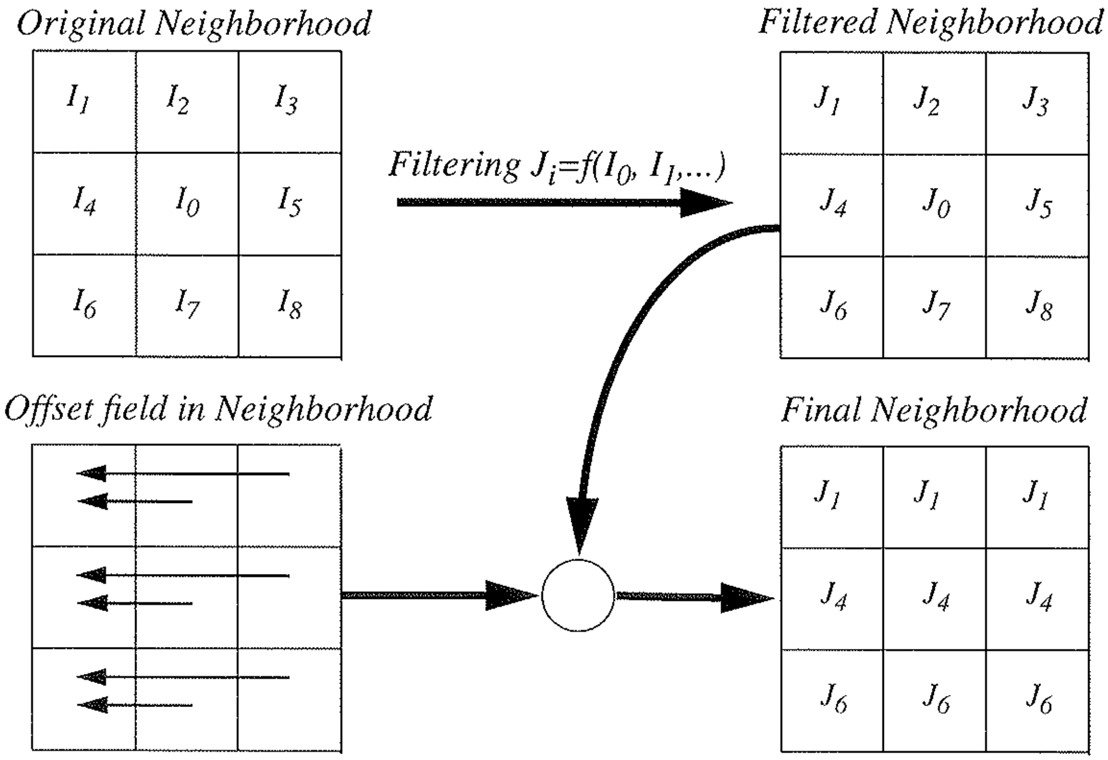


FIGURE 5.1. Schematic illustration of postprocessing with an offset vector field. Top left: original intensity values in a neighborhood of the point I_0. Top right: the neighborhood after applying a filter function f. Bottom left: offset field in the neighborhood. Bottom right: final intensity values in the neighborhood after using the offset field to permute the filtered intensity values.

The transformation of the offset filtering into a postprocessing procedure has a number of notable advantages. Most importantly, it allows efficient implementations of offset filtering using existing algorithms and fast hardware. The median filter is an excellent example of this process, since efficient implementations exist which make use of the overlap of neighboring windows to speed up the median computation (Huang et al., 1979; Danielsson, 1981). Straightforward use of displaced windows renders this method inapplicable. However, applying the displacement vectors after the application of a standard median filter enables the use of this type of optimization. From

[1] This is true for image-independent filters such as the mean or median. For the Nitzberg-Shiota type nonlinear filter this technique uses the filter shape at the remote location as opposed to the shape at the current pixel. This is probably advantageous as it is the structure of the neighborhood around the filter we are concerned with, not that around the pixel being assigned a new value.

an implementation standpoint, the post-processing procedure obviates the need to modify each individual filter to employ a displacement field. Furthermore, postprocessing permits the offset computation to be carried out on the smoother filtered image. The post-processing approach is obviously advantageous with respect to both hardware development, as the pixel permutation procedure has a straightforward hardware implementation, as well as software implementations, because the filters can be developed independently from the application of the offset field.

Finally, we have typically found that only a relatively small percentage of image values are used in this technique (30% or less depending on the noise and small scale structure present in the image). This implies that the filter computation can be limited to these sites, providing an increase in speed by a factor of 3 or more. Examining the distribution of these locations, as depicted in figure (5.2), verifies the behavior of the algorithm. Intensities in the central image indicate the number of locations mapped to each pixel[1]. The most used locations tend to be clustered close to the center of each region, giving the image the appearance, for some image structures, of a medial axis transform. This occurs due to the search in the offset direction, which proceeds until the vector field reverses direction due to the influence of either an edge in the search direction, or a relatively homogenous region which gives rise to small offset vectors. In some cases, as we have shown, this is close to the medial axis of the region, as it contains intensity values which are more representative of the typical intensity than are intensities close to the border. Nevertheless, it must be emphasized that the medial axis *per se* occurs in this particular example because of a favorable relation between the window size (i.e. the size of the smoothing kernel) and the scale of the image features. For image structures which are large relative to the window size, the offset locations will occur on an interior border of the region, which may be far from the medial axis. A more correct statement of the nature of the offset vector field is that although in some cases it is medial axis in appearance, in general it is a form of adaptive skeletonization that is more general than the medial axis.

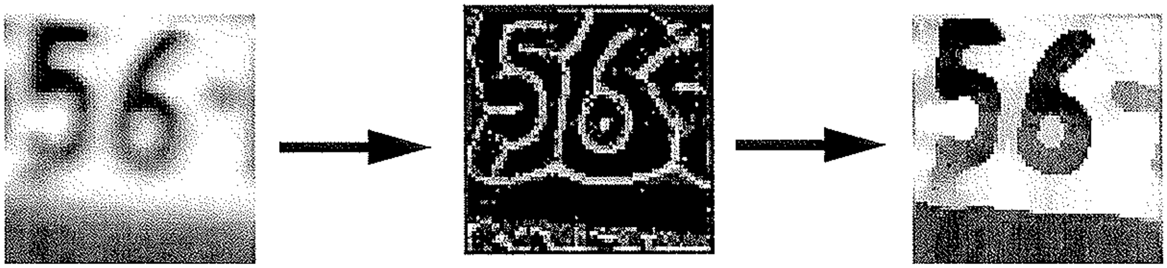


FIGURE 5.2. Postprocessing with an offset vector field. Left: image filtered with standard (i.e. non-offset) median filter. Center: offset locations computed from median filtered image. Pixel intensity indicates the number of image locations mapped to each location. Light spots indicate positions whose intensity values are used in final image (right), while dark regions are not used. This indicates the generalized "skeletonization" associated with the offset vector field. In this particular case, the offset field is similar to a medial axis transform.

These observations are of interest as there is recent physiological (Lee, 1995; Lee et al., 1996; Lamme, 1995) as well as psychophysical (Frome, 1972; Kovács and Julesz, 1994) evidence for the importance of structures which vaguely resemble the medial axis in primate visual systems.

[1] This image is actually the log of the mapping frequency plus one. The compression is necessary for display purposes.

These findings have given rise to a number of computational models which make use of the medial axis as a shape descriptor (Blum, 1967; Blum, 1973; Blum and Nagel, 1978; Burbeck and Pizer, 1995) or for image enhancement (Osher and Sethian, 1988; Malladi and Sethian, 1995). In our approach, the skeletonization is generated as a by-product of contrast enhancement and noise reduction. The image skeletonization is performed not as an end, or for shape description, but merely as the locus of points at which offset filters will be located to provide image contour enhancement.

6. Subsampling with offset vectors.

Offset vector fields can also be used for image scaling. This is illustrated in figure (6.1). The image at the left is the 256x256 pixel original. The central image is downsampled by 2 in each dimension using bicubic interpolation to generate each pixel value, while the rightmost image is subsampled using a displacement vector field and a median filter. That is, a displacement vector field is generated at each of the locations to be sampled in the original, then the smaller image is constructed by assigning the median intensity value in a neighborhood of the displaced location to the destination pixel. As is the case for straightforward filtering, the use of displaced sampling prevents filtering across edge boundaries, while retaining small-scale image features.

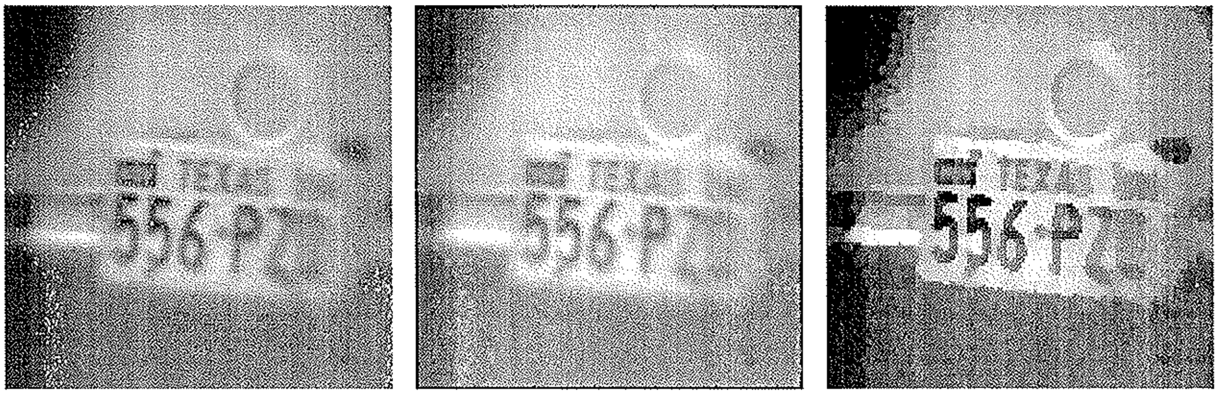


FIGURE 6.1. Subsampling with displacement vectors. Left: an image subsampled with bicubic interpolation. Right: the same image subsampled using the offset vector field.

The same technique can be used for nonuniform sampling. For example, the complex log transformation which approximates the mammalian retino-cortical mapping is also widely used as a means for constructing space-variant vision systems[1] (Rojer and Schwartz, 1990; Weiman, 1988; Sandini and Dario, 1989, Sandini et al., 1989; Messner and Szu, 1986; Schenker et al., 1981; Bonmassar and Schwartz, 1994; Bonmassar and Schwartz, 1995; Yamamoto et al., 1996). In this approach, pixel sampling is performed at exponentially increasing intervals. In order to avoid aliasing, each range pixel is typically constructed by averaging a region whose size in the domain of the mapping grows exponentially with increasing eccentricity (distance

[1] In this context it is worth noting that in other work we have shown that the structure of the complex log mapping allows the diffusion PDE to be integrated at rates which are exponential functions of eccentricity, yielding an order of magnitude speed increase over and above the pixel compression supplied by the transformation itself (Fischl et al., 1996).

from the center of the image). This tends to dampen high frequency information such as edges in the periphery. The use of a displacement vector field in the subsampling can alleviate this problem. This is illustrated in figure (6.2) which shows the complex log transformation of the image in figure (6.1) using these two techniques. The figure at the left is constructed by averaging domain pixels, while the one at the right is a subsampling using the displacement vector field to specify which domain pixel neighborhood to median filter. The image non-uniformly sampled with the displacement field is both sharper and less noisy than its counterpart.

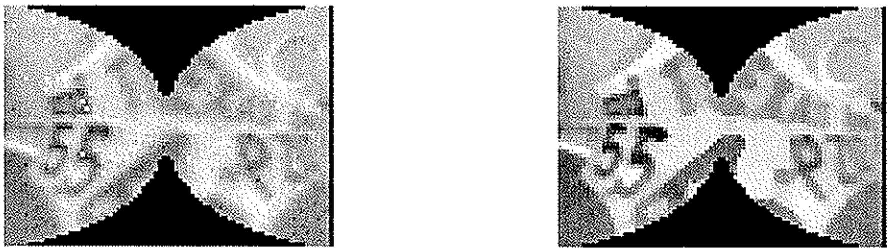


FIGURE 6.2. Nonuniform sampling using pixel averaging (left) versus sub-sampling with a displacement vector field (right).

7. Results.

In this section we present a comparison of an offset median filter with a standard median filter as well as the result of using anisotropic diffusion for image filtering. We use the median as it is a nonlinear filter with good noise suppression capabilities at relatively low computational cost. The Perona-Malik technique for nonlinear diffusion is not noise-tolerant (Whitaker and Pizer, 1991; El-Fallah and Ford, 1994), and is hence inappropriate for comparison purposes. For that reason, the images presented in this section are generated using the mean curvature based diffusion algorithm of El-Fallah and Ford (El-Fallah and Ford, 1994) which has good noise-suppression qualities. For comparison purposes we also present images filtered with a standard Gaussian, an offset Gaussian, as well as the nonlinear filter of Nitzberg and Shiota outlined in section (3.1). This highlights an additional advantage of nonlocal filtering: the choice of filter can be made independently from the use of the offset vector field, perhaps on the basis of estimated image statistics.

The offset computation slows down the median and Gaussian filters by between about a factor of six or seven, but is still approximately an order of magnitude faster than our earlier Greens Function approximation to nonlinear diffusion (Fischl and Schwartz, 1997; Fischl and Schwartz, 1996), which was itself roughly an order of magnitude faster than the nonlinear diffusion process[1]. Running on a 50 MHz Sparc-10, a 3x3 median filter applied to a 64x64 pixel image requires approximately 0.06 seconds. Using the displacement vector field increases the time to 0.35 seconds, while the 50 time steps used to integrate the anisotropic diffusion necessitate between 15 and 16 seconds. These results together with timing for the other filters presented below are summarized in figure (7.1). The Nitzberg-Shiota algorithm was found to perform well using an 11x11 filter in conjunction with a 7x7 window for their vector field calculation, resulting in a computational expense comparable to that of anisotropic diffusion.

[1] All these algorithms have the same computational complexity dependence on the number of pixels in the image (assuming a fixed number of iterations for the PDE integration). We use 64x64 pixel images as an example, but the relative computational times are invariant to the number of pixels in the image.

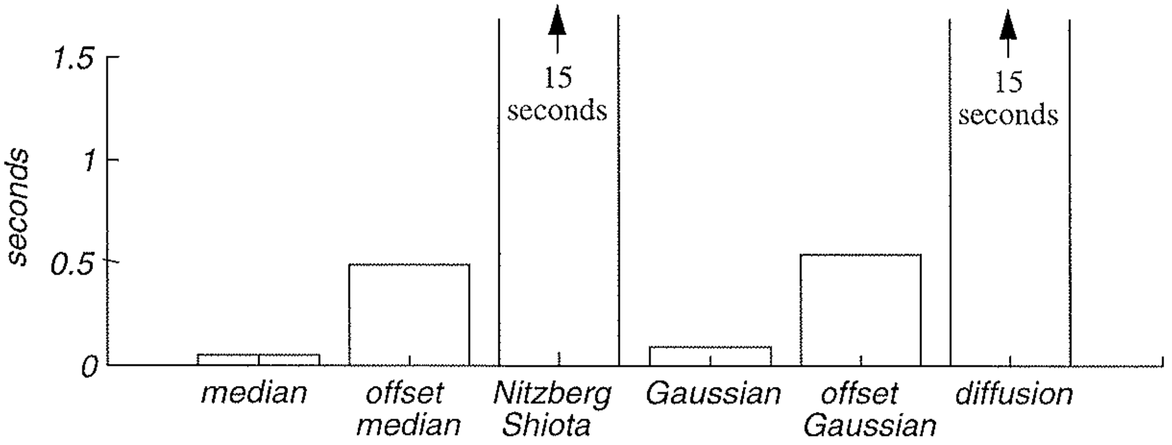


FIGURE 7.1. Plot of the cost of the different filtering methods shown in table (7.1). The timing is based on filtering a 64x64 pixel image on a Sparc-10. The anisotropic diffusion requires between 15 and 16 seconds to complete 50 iterations on this size image, and extends off the top of the graph. The Nitzberg-Shiota filtering was applied iteratively (3 applications) using an 11x11 filter and a 7x7 window size for the vector field calculation, resulting in a processing time comparable to that of the anisotropic diffusion. The other filters all use a 3x3 window size.

Table (7.1) contains the results of the six different types of filtering noted above. Every second row shows edge maps generated using a Canny edge detector (Canny, 1986) on the output of each filter. These maps are useful as a means of qualitatively comparing the different filtering techniques, since they are directly dependent on the preservation of the differential structure of the image. As can be seen from these results, the median filter is adequate for noise reduction, but

Table 7.1: Comparison of different types of filtering

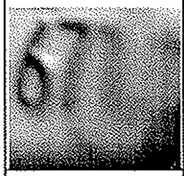
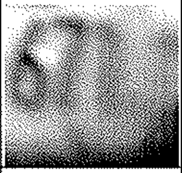
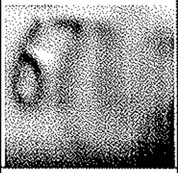
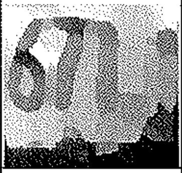
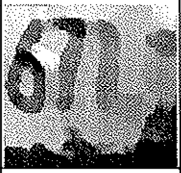
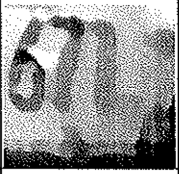
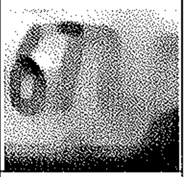
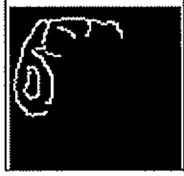
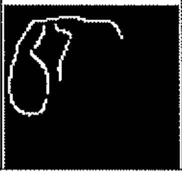
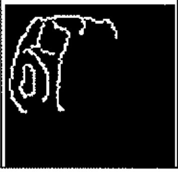
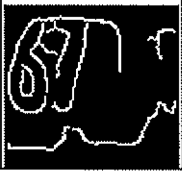
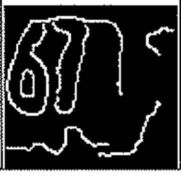
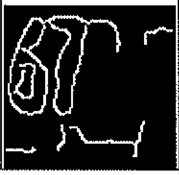
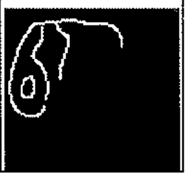
original	Gaussian filter	median filter	offset Gaussian	offset median filter	Nitzberg-Shiota filter	Anisotropic diffusion
						
						

Table 7.1: Comparison of different types of filtering

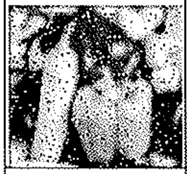
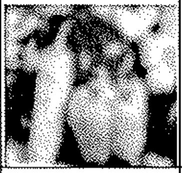
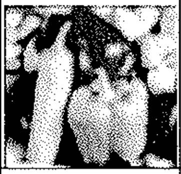
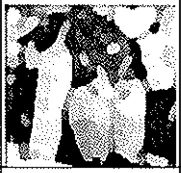
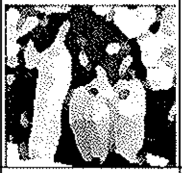
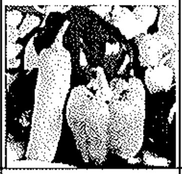
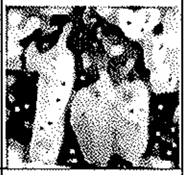
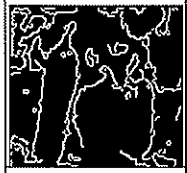
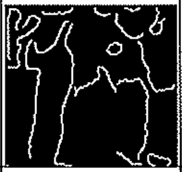
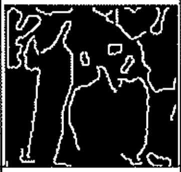
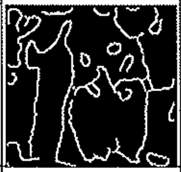
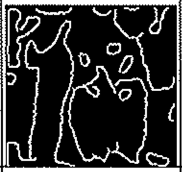
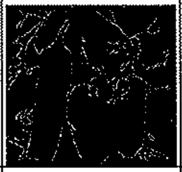
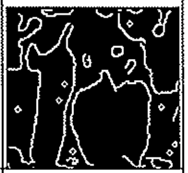
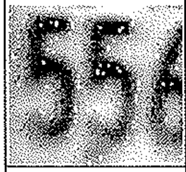
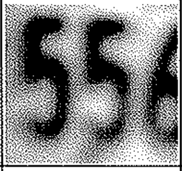
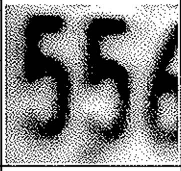
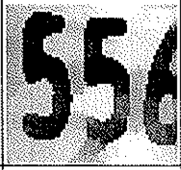
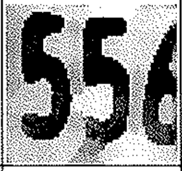
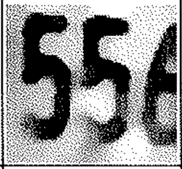
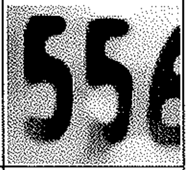
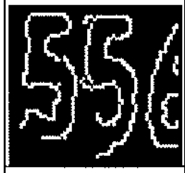
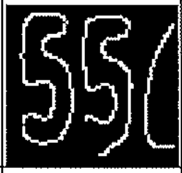
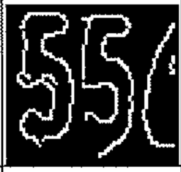
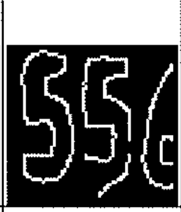
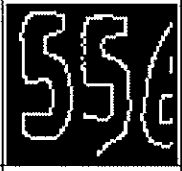
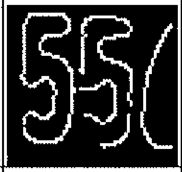
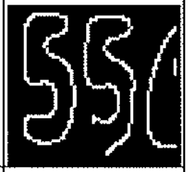
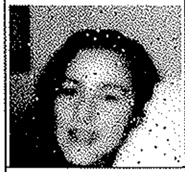
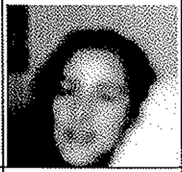
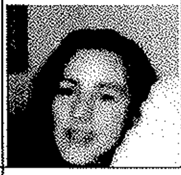
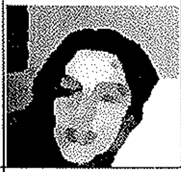
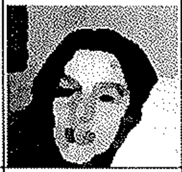
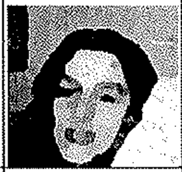
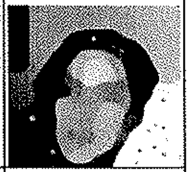
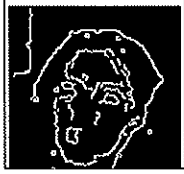
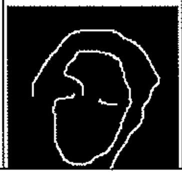
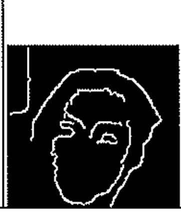
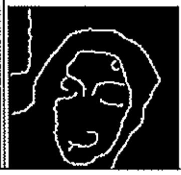
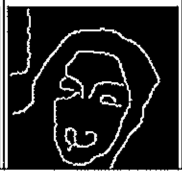
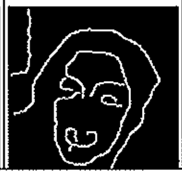
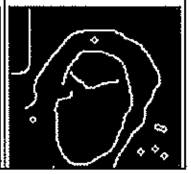
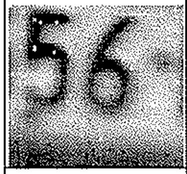
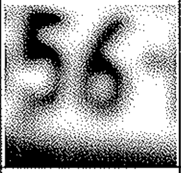
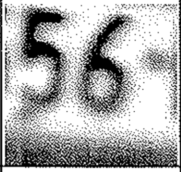
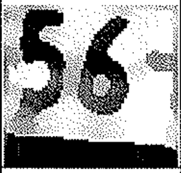
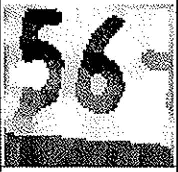
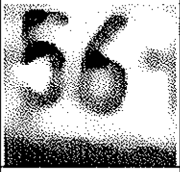
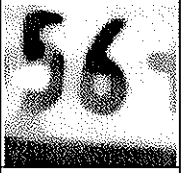
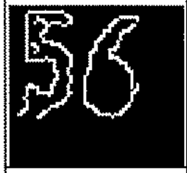
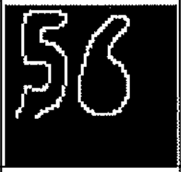
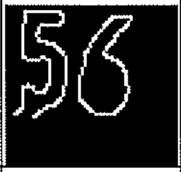
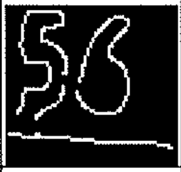
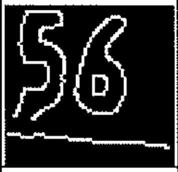
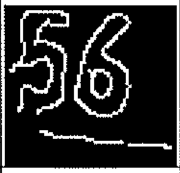
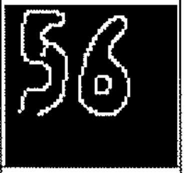
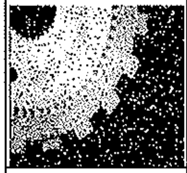
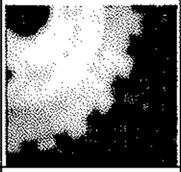
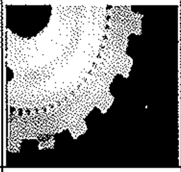
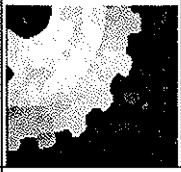
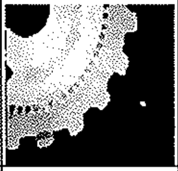
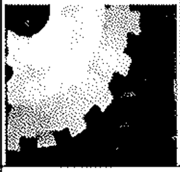
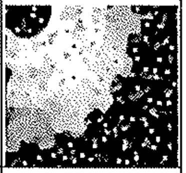
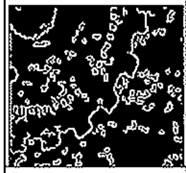
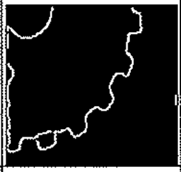
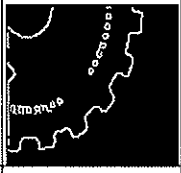
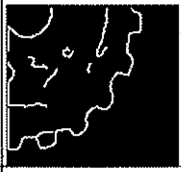
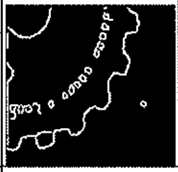
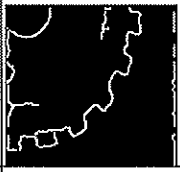
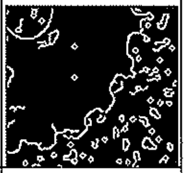
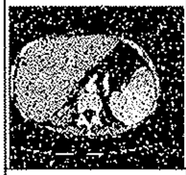
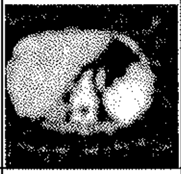
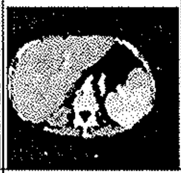
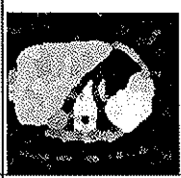
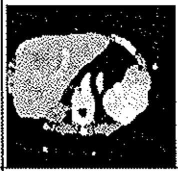
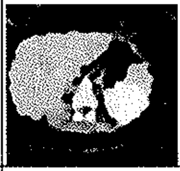
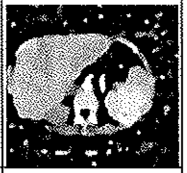
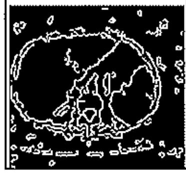
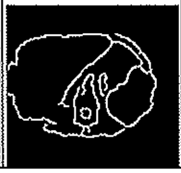
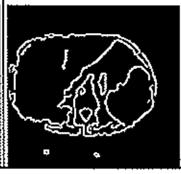
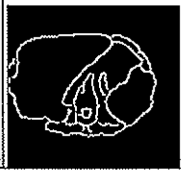
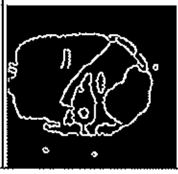
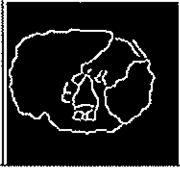
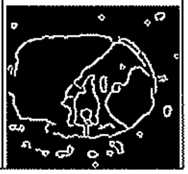
original	Gaussian filter	median filter	offset Gaussian	offset median filter	Nitzberg-Shiota filter	Anisotropic diffusion
						
						
						
						
						
						

Table 7.1: Comparison of different types of filtering

original	Gaussian filter	median filter	offset Gaussian	offset median filter	Nitzberg-Shiota filter	Anisotropic diffusion
						
						
						
						
						
						

does not increase contrast, and therefore is unable to recover edge information which is not of sufficient strength in the initial image to be recognized by the Canny detector. In contrast, the anisotropic diffusion increases contrast, and has reasonable noise reduction properties, although it tends to lose high curvature points such as the corners of the characters or the license plate. The application of a displacement vector to the median filter retains its noise reduction qualities, but also enhances contrast, bringing out edge information that the Canny detector does not identify in the initial image.The Gaussian filter used in conjunction with the adaptive length displacement vector field gives comparable results, with the median probably yielding the best noise-suppression for the 3x3 size.

8. Conclusion.

Linear filtering can be used to efficiently reduce noise in images at the cost of blurring and possibly fusing region boundaries. Nonlinear techniques are useful in this context, resulting in both contrast enhancement as well as noise reduction. The general goal of the various approaches that have been developed is to avoid "smoothing" across edge structure in the image, while smoothing along the edge structure. Anisotropic diffusion equation based methods achieve this by modifying the diffusion constant adaptively so that more diffusion occurs along, as opposed to across edges (Perona and Malik, 1987). Neural network approaches achieve similar goals by emulating this behavior with detailed networks of model neurons (Cohen and Grossberg, 1984; Grossberg and Mingolla, 1985). However, the computational cost of these algorithms prohibits their use in real-time or quasi real-time vision applications.

In this paper we have presented an alternative technique, which modifies the use of standard image filters such as the mean or median, to make use of displacement vector fields. The displacement vectors push kernels away from edge regions, preventing edge blurring and destruction, while achieving results which appear to be qualitatively similar to diffusion based approaches, but with considerable computational savings. The motivation for this idea came from a detailed study that we made in previous work in which we examined the effective kernels produced by several different anisotropic diffusion methods. It was clear from this work that the diffusion equation was overfitting the final image to the fine-grained "noise" in the image, and could be replaced by the offset vector field method outlined in this paper. We have found this method to hold up well for a wide variety of images, and in all cases to provide significant improvements in speed of computation. Combining the nonlocal filtering with a space-variant vision representations (e.g. Rojer and Schwartz, 1990) we have achieved frame-rate enhancement[1]. In principle, the conjunction of these two techniques can provide 3-5 orders of magnitude of speedup over conventional anisotropic diffusion on space-invariant image architectures (two orders of magnitude in the diffusion stage, and one to three orders of magnitude from the space-variant pixel compression).

In summary, we have outlined a new approach to image filtering which achieves results that are comparable to nonlinear diffusion, but with a much simpler and faster implementation. This work has the following practical advantages over other methods with similar goals:

- *Speed*: the offset vector filter is approximately two orders of magnitude faster than nonlinear diffusion, and roughly one order of magnitude faster than the Greens Function approximator (Fischl and Schwartz, 1997; Fischl and Schwartz, 1996) to nonlinear diffusion.
- *Hardware application*: By using the image permutation form of the offset vector filter, it is possible to use existing, or future, fast filter hardware, and a simple LUT or image permutation, to implement the nonlocal filtering.

[1] We have run the nonlocal filtering at over 30 Hz on a 180 MHz dual P6 using a relatively large (80x64 pixel) logmap.

- *Algorithm design*: By separating the process into a generalized skeletonization (i.e. determing the location, direction and magnitude of the offset vector field), and a simple single scale filter, the design of new versions of this class of algorithm is greatly simplified.

Finally, from a theoretical point of view, the following insights are provided by this work:

- The desirable aspects of scale-space methods are retained without the need to explicitly introduce scale, which is represented in this method by the magnitude of the offset vector field.
- The desirable performance of nonlinear diffusion is retained without reference to any underlying diffusive, or intrinsically serial, process.
- Nonlocal filter operators, implicit in the work of Nitzberg and Shiota, are explicitly developed in this paper.
- The combination of two very different aspects of image processing (i.e. generalized skeletonization, as represented by the determination of the offset vector field locus, direction and magnitude) with conventional image filtering, seem to offer a fertile area for future development.

9. Bibliography.

Blum, H. (1967). A new model of global brain function. *Perspectives in Biology and Medicine*, 10:381–407.

Blum, H. (1973). Biological shape and visual science: Part I. *Journal of Theoretical Biology*, 38:205–287.

Blum, H. and Nagel, R. N. (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180.

Bonmassar, G. and Schwartz, E. (1995). Space-variant fourier analysis: the exponential chirp transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted.

Bonmassar, G. and Schwartz, E. L. (1994). Geometric invariance in space-variant vision systems: the exponential chirp transform. In *ICPR Proceedings*, ICPR-12, pages 204–207. International Conference on Pattern Recognition.

Burbeck, C. A. and Pizer, S. M. (1995). Object representation by cores: Identifying and representing primitive spatial regions. *Vision Research*, 35(13):1917–1930.

Burt, P. and Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 9(4):532–540.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698.

Cohen, M. A. and Grossberg, S. (1984). Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance. *Perception and Psychophysics*, 36:428–456.

Danielsson, P.-E. (1981). Getting the median faster. *Computer Graphics, and Image Processing*, 17:71–78.

El-Fallah, A. I. and Ford, G. E. (1994). Nonlinear adaptive image filtering based on inhomogeneous diffusion and differential geometry. *SPIE Image and Video Processing II*, 2182:49–63.

Fischl, B., Cohen, M. A., and Schwartz, E. L. (1996). Real-time anisotropic diffusion using space-variant vision. Technical Report CAS/CNS-TR-95-026, Boston University, Department of Cognitive and Neural Systems, 677 Beacon St., Boston MA.

Fischl, B. and Schwartz, E. (1997). Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):342–352.

Fischl, B. and Schwartz, E. L. (1996). Learned adaptive nonlinear filtering for anisotropic diffusion approximation in image processing. In *International Conference on Pattern Recognition*, Vienna, Austria.

Frome, F. S. (1972). A psychophysical study of shape. Technical Report TR-198, University of Maryland, Computer Science Center, College Park, Maryland.

Grossberg, S. and Mingolla, E. (1985). Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading. *Psychological Review*, 92(2):173–211.

Huang, T., Yang, G., and Tang, G. (1979). A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18.

Hummel, A. (1986). Representations based on zero-crossings in scale-space. In Fischler, M. and Firschein., O., editors, *Readings in Computer Vision: Issues, Problems, Principles and Paradigms*. Morgan Kaufmann, Los Angeles.

Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, 50:363–370.

Kovács, I. and Julesz, B. (1994). Perceptual sensitivity maps within globally defined visual shapes. *Nature*, 370:644–646.

Lamme, V. A. F. (1995). The neurophysiology of figure-ground segregation in primary visual cortex. *The Journal of Neuroscience*, 15(2):1605–1615.

Lee, T. S. (1995). Neurophysiological evidence for image segmentation and medial axis computation in primate V1. In Bower, J., editor, *Advances in Computational Neuroscience. Proceedings of the fourth annual computational neuroscience conference*. Plenum Press.

Lee, T. S., Lamme, V. A. F., and Mumford, D. (1996). The role of V1 in scene segmentation and shape representation. *Nature*.

Malladi, R. and Sethian, J. A. (1995). Image processing via level set curvature flow. *Proceedings of the National Academy of Sciences*, 92:7046–7050.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London*, B 207:187–217.

Messner, R. and Szu, H. (1986). An image processing architecture for real time generation of scale and rotation invariant patterns. *Computer Vision, Graphics and Image Processing*, 31:50–66.

Nitzberg, M., Mumford, D., and Shiota, T. (1993). *Filtering, Segmentation and Depth.* Springer-Verlag, Berlin.

Nitzberg, M. and Shiota, T. (1992). Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):826–833.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: Algorithms based on Hamilton Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49.

Perona, P. and Malik, J. (1987). Scale space and edge detection using anisotropic diffusion. In *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pages 16–27, Miami, FL.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Rojer, A. S. and Schwartz, E. L. (1990). Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *10th International Conference on Pattern Recognition*, volume 2, pages 278–285.

Sandini, G., Bosero, F., Bontino, F., and Ceccherini, A. (1989). The use of an anthropomorphic visual sensor for motion estimation and object tracking. *Proceedings of OSA optical meeting on image understanding and machine vision.*

Sandini, G. and Dario, P. (1989). Active vision based on space-variant sensing. *International symposium on robotics research.*

Schenker, P., Cande, E., Wong, K., and Patternson, W. (1981). New sensor geometries for image processing: computer vision in the polar exponential grid. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing.*, pages 1144–1148.

ter Haar Romeny, B. M. (1994). *Geometry Driven Diffusion in Computer Vision.* Kluwer, Monterey, CA.

Weiman, C. (1988). 3-d sensing with polar exponential sensory arrays. *SPIE Conference on digital and optical shape representation and pattern recognition.*

Whitaker, R. T. and Pizer, S. M. (1991). A multi-scale approach to nonuniform diffusion. *Computer Vision, Graphics and Image Processing*, 57:99–110.

Witkin, A. (1983). Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1021, Karlsruhe, West Germany.

Yamamoto, H., Yeshurun, Y., and Levine, M. D. (1996). An active foveated vision system: attentional mechanisms and scan path convergence measures. *Computer vision and image understanding: CVIU*, 63(1):50.

10. Appendix.

An alternative to nonlinear filtering is to selectively blur an image, modulating the amount of blurring based on local gradient information (Perona and Malik, 1987). This approach has basically the same goals as the filtering described above: noise reduction and contrast enhancement. While the two formulations seem quite different, in fact they have much in common. In this section we will discuss the relationship between the anisotropic diffusion approach and nonlinear filtering using two filtering techniques as examples.

10.1. Green's Function Approximation (GFA) Filtering.

In earlier work, we approximated the solution to a nonlinear diffusion process by transforming the numerical integration of the partial differential equation (PDE) describing the diffusion into an adaptive filtering procedure (Fischl and Schwartz, 1997; Fischl and Schwartz, 1996). In this work, we obtained an approximation of the nonlinear analog of a "Greens Function" for the original partial differential diffusion equation. Our goal was to replace the PDE formulation, which is computationally expensive because it requires serial integration over time, with a single spatial integration or filtering step using the approximated "Greens Function". This enabled us to construct an approximate solution of the diffusion problem in a single time-step by filtering the initial image data with the adaptively estimated "Greens Function". By studying the intermediate computational results of several nonlinear diffusion methods we were able to determine where the computational costs of integrating a diffusion equation are encountered, thus motivating the idea of the "offset" vector field discussed in the present paper.

Our strategy was to monitor the numerical integration of a nonlinear anisotropic diffusion equation, and to save the paths through which intensity values diffused at each integration step. For a given image and evolution time, we therefore computed a set of space-variant kernels, called diffusion kernels, that *exactly* mirrored the integration of the nonlinear diffusion equation for that time. The construction of these kernels allowed us to view the nonlinear filter implicitly encoded by the numerical implementation of the diffusion process. Visually examining the diffusion kernels obtained in this way yielded insight into the inefficiency of the diffusion approach. By studying the entire set of kernels with principal components analysis we found (perhaps not surprisingly!) that the kernels obtained over the full integration regime were accounted for by only a small number of different basic kernel shapes. Furthermore, the kernels occurred with specific offsets from the edges in the image. With minimal smoothing, approximately 90% of the variance in the kernels was accounted for by the first 5 or 10 principal components (see figure (10.1)).

Visual examination of the diffusion kernels also elucidated where the inefficiency of the full diffusion equation solution was encountered: although the resultant kernels appeared to be simple in gross outline, they actually consisted of very detailed (high spatial frequency) structure, which can be seen in figure (10.2). Omitting this detailed structure, via simplification through either principal components analysis or lowpass filtering, made little difference to the final solution. Clearly the detailed PDE solution was "overfitting" the local image structure in a way which was computationally inefficient, since the details of the process were not meaningfully coupled to the final output!

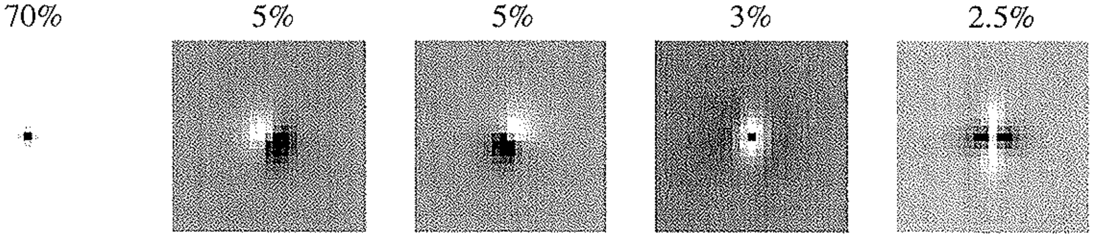


FIGURE 10.1. First five principal components of blurred kernels generated by iterating mean curvature-based diffusion equation (El-Fallah and Ford, 1994) using the image in figure (10.2) as initial conditions. The percentages above the figures indicate the amount of variance accounted for by each component.

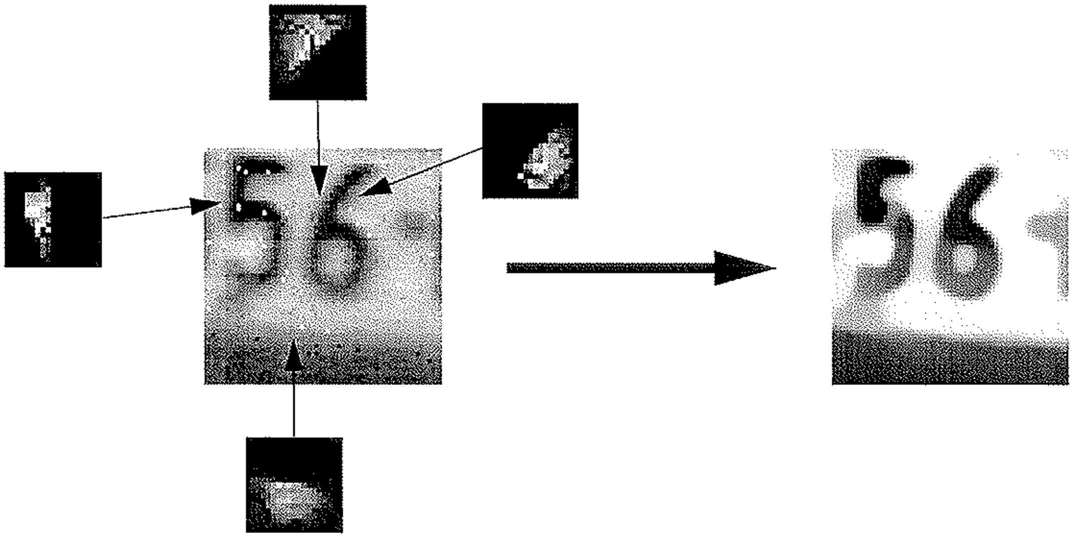


FIGURE 10.2. Typical diffusion kernels. Left: original image with diffusion kernels shown together with their location. Right: image after applying the full set of kernels of to original.

Visual examination of the diffusion kernels also yielded insight into the manner in which the diffusion equation achieves image enhancement. Typically, the kernels had the appearance of delta functions located exactly on edge features in the image, gradually transforming into rotationally asymmetric truncated Gaussians as one moved away from a strong edge. Figure (10.2) illustrates this effect. The large left hand image shows a region of a license plate before undergoing diffusion, while the right hand image shows the result of iterating the Perona and Malik type diffusion process proposed by El-Fallah and Ford (El-Fallah and Ford, 1994) for 100 time steps. The smaller images surrounding the initial license plate are the diffusion kernels at each of the four specified locations. As can be seen, the shape of each kernel has the same orientation as the dominant local edge direction. Furthermore, the kernels from locations slightly offset from the edge are themselves offset so that they do not straddle the edge. Thus, in regions near an edge, the diffusion equation implements a nonlinear filter which averages intensity values from one side of the edge or the other, but not both. It is clear from these observations that replacing the computationally expensive temporal integration of the diffusion equation with a simple filter, locating the filter centers at the correct offset from each pixel, could greatly speed up the solution.

In our previous work we designed a procedure to adaptively learn the shape and position of the diffusion kernels directly, employing a nonlinear function approximator (a multi-layer perceptron trained with back-propagation) to estimate the principal component coefficients as a function of local image structure. A surprising result of this study was the discovery that a single, relatively small (64x64 pixels) image was sufficient for training a filter which achieved good results on a wide variety of images. This finding, in conjunction with the principal components analysis noted above, suggests that diffusive-type image enhancement filtering of gray-scale images can be based upon a small number of local image features, *but the filters must be applied at image-driven offsets.*

In order to confirm this conjecture we subjected the diffusion kernels to a variety of transformations and observed the results of filtering with the transformed kernels. If the location of the kernel centers is the critical computational component of the nonlinear diffusion, then transformations which modify the shape of the kernels, but do not dramatically alter the location of the filtering should have little effect on the enhancement process. The results of these experiments are given in figures (10.3)-(10.5). Each of these figures shows the original image at the left together with four diffusion kernels surrounding it, as well as the result of filtering with the transformed kernels on the right. In figure (10.3) the kernels have been smoothed with a Gaussian smoothing kernel (σ=1). In figure (10.2) the kernels have been thresholded (T=0.005) and normalized, while in figure (10.5) the kernels have been smoothed (σ=1), then thresholded (T=0.005) and normalized.

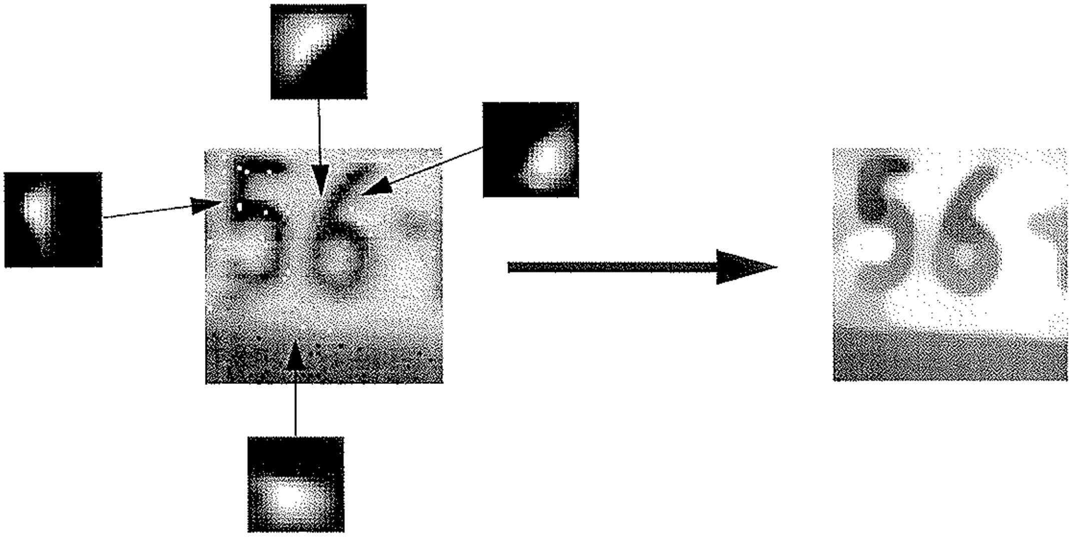


FIGURE 10.3. Result of using blurred (Gaussian blurring, σ=1) diffusion kernels. Left: original image with diffusion kernels shown together with their locations. Right: image after applying kernels to original.

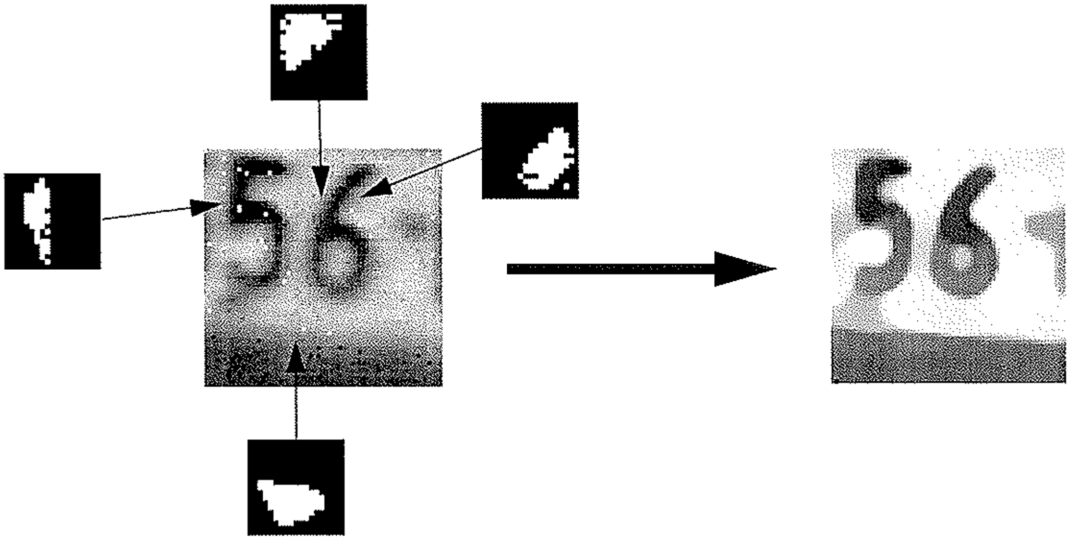


FIGURE 10.4. Result of using thresholded (*T*=.005) kernels. Left: original image with diffusion kernels shown together with their locations. Right: image after applying kernels to original.

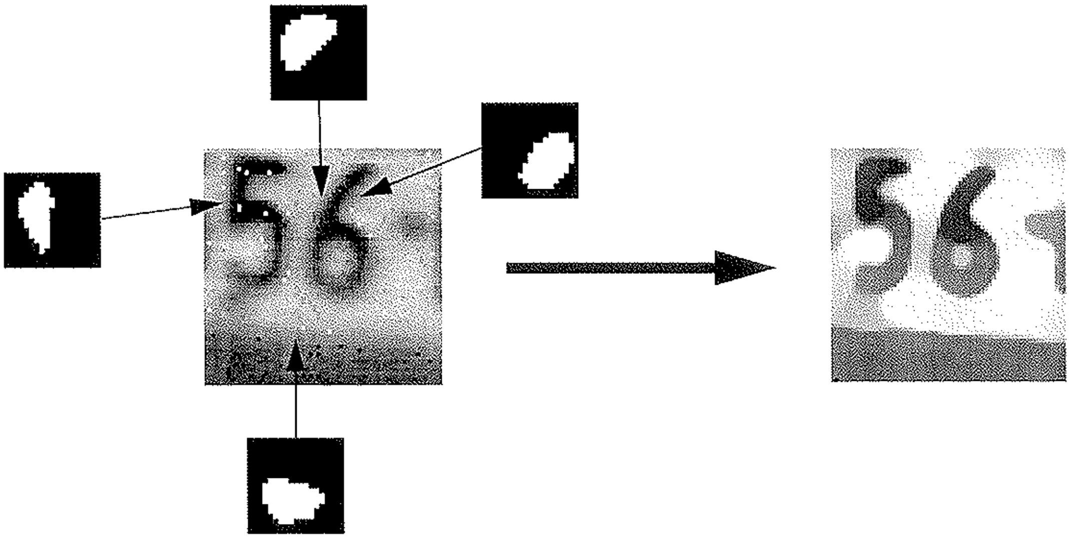


FIGURE 10.5. Result of using blurred (Gaussian, σ=1) and then thresholded (*T*=0.005) diffusion kernels. Left: original image with diffusion kernels shown together with their locations. Right: image after applying kernels to original.

Examining the right-hand images in these figures it is clear that these modifications have little or no visible effect on the result of filtering, despite the fact that the diffusion kernels in figure (10.5) have more than an order of magnitude less information than the original kernels.[1] This indicates that the costly fitting process performed by the nonlinear diffusion is unnecessary for the image enhancement results.

[1] We employ the standard information theoretic measure, using intensity value frequency histograms to estimate the probability distributions and summing *-p log(p)*.

It is therefore apparent that most of the computational time of the nonlinear diffusion solution is spent in an inefficient "random-walk" which is extremely expensive to perform, but which appears in the end to be equivalent to the application of simple filters at image-driven offsets.