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
email: fischl@cns.bu.edu

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 significant improvement in image enhancement as compared to linear filters. However, the excessive computational cost of solving nonlinear PDEs precludes their use in real-time vision applications. In the present paper we show that two orders of magnitude speed improvement is provided by a new image filtering paradigm in which an adaptively determined vector field specifies nonlocal application points for an image filter.*

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 their 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 spatial 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.

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.

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. Nitzberg and Shiota (Nitzberg and Shiota, 1992) proposed a method based on gradient direction as well as magnitude which performs well, but fails in regions which contain edges at a number of scales, an issue we resolve in the computation of our offset field.

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 kernels, 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 were basically irrelevant[1]. We exploited this observation by simplifying the final kernels via blurring and principal

1. The enhanced image which results from applying the diffusion kernels is actually quite insensitive to the spatial structure of the kernels. Blurring and/or thresholding of the kernels results in images which are visually indistinguishable from the image generated by the full diffusion process.

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. In the present paper we show a further exploitation of the relatively low information content of the intermediate diffusion kernels that has led both to an additional 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.

4. Offset Vector Field Computation.

Filtering with a kernel that is symmetric around the central pixel results in averaging of edge values, and therefore blurring of the edge. 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”. The 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 offset vector field $\boldsymbol{v}(\boldsymbol{z})$ can be written as[1]

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

$\boldsymbol{O}(\boldsymbol{z})$ in equation (4.1) is the offset orientation, and refers to the choice of the normal vector. Offset direction, denoted $d(\boldsymbol{z})$, 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 $m(\boldsymbol{z})$ encodes the

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

length of the offset vector. In this section we will outline a simple procedure for computing each of these quantities.

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

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

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.

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. Combining these two constraints yields the following expression for the calculation of offset direction.

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

where *sgn* is the sign function and $\boldsymbol{z}'$ refers to (x,y) coordinates relative to the filter center. 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 $d(\boldsymbol{z})$ 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^{o}, 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 thus a two step procedure. First, an initial offset field is computed using equations (4.2)-(4.3) via

$$\boldsymbol{v}_i(\boldsymbol{z}) = d(\boldsymbol{z})\,\boldsymbol{O}(\boldsymbol{z}) \ . \tag{4.4}$$

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

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

Finally, we form the final vector field using $m(\boldsymbol{z})$ as the magnitude of the initial vector field

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

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.

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 vectors to shuffle the positions of the image intensity values.[1] In this way, the value at each point

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.

in the filtered image is replaced with the value at the location specified by the displacement vector field.

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 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 (5.1), verifies the behavior of the algorithm. Intensities in the central image indicate the num-

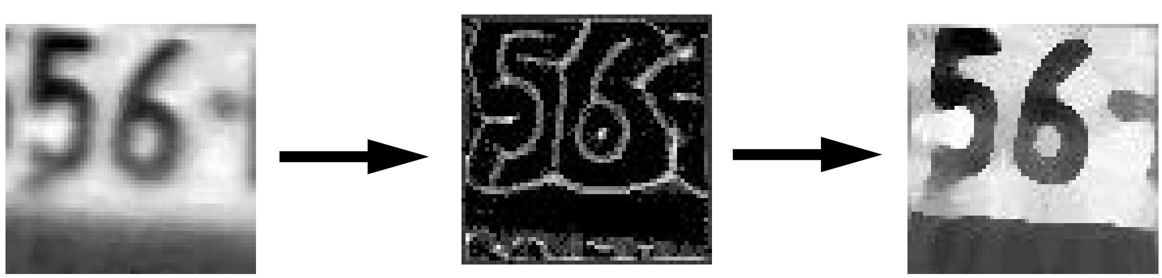


FIGURE 5.1. Postprocessing with an offset vector field. Left: image filtered with standard (i.e. non-offset) median filter. Center: offset locations computed from median filtered image. Pixel intensity indicates the number of image locations mapped to each location. Light spots indicate positions whose 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.

ber 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 form of skeletonization similar to a medial axis transform (Blum, 1967). 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.

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. 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. 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;

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

Catté et al., 1992; Catte 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 (50 iterations of diffusion, with *A*=500). For comparison purposes we also present images filtered with a standard Gaussian (σ=2), an offset Gaussian, as well as the nonlinear filter of Nitzberg and Shiota outlined in section (3.2). 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 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].[2] 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 (6.1). The Nitzberg-Shiota algorithm was found to perform well using an 11x11 filter (filter parameter σ=0.02, $p_1(x,y)=p_2(x,y)=p_3(x,y)=N(0,2))$ in conjunction with a 7x7 window for their vector field calculation (μ=0.05, *c*=3), 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.
2. Recent work using implicit integration schemes has sped up the integration of the nonlinear diffusion PDE by an order of magnitude (Weickert, 1997; Weickert et al., 1997; Weickert et al., 1998)

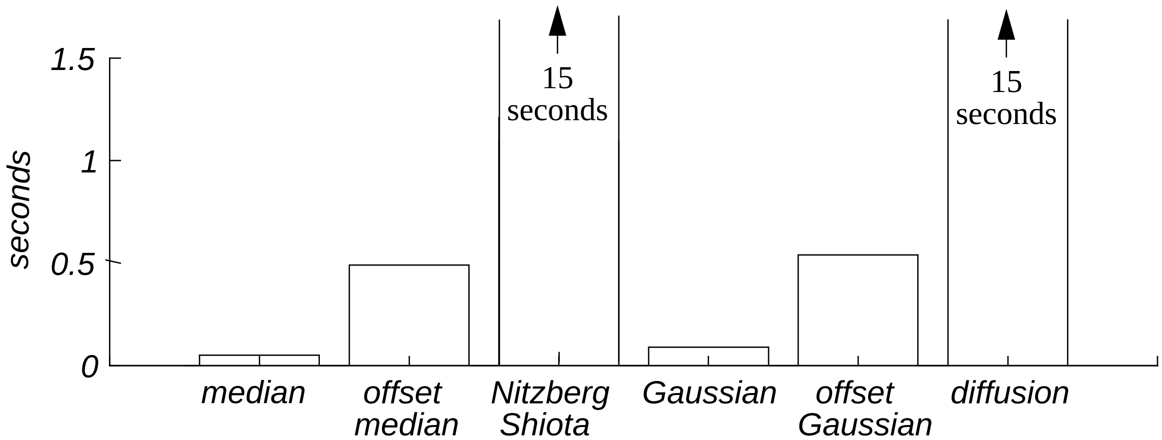


FIGURE 6.1. Plot of the cost of the different filtering methods shown in table (6.1). The timing is based on filtering a 64x64 pixel image on a Sparc-10. The anisotropic diffusion requires between 15 and 16 seconds to complete 50 iterations on this size image, and extends off the top of the graph. The 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 (6.1) 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 6.1: Comparison of different types of filtering

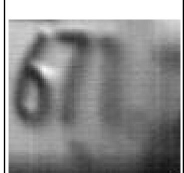
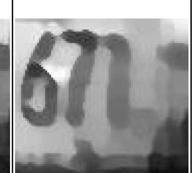
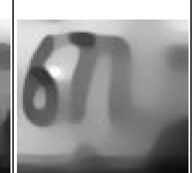
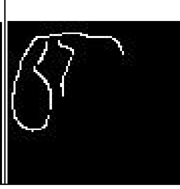
original	Gaussian filter	median filter	offset Gaussian	offset median filter	Nitzberg-Shiota filter	Anisotropic diffusion
						
						

Table 6.1: 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. 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.

7. 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 spatial LUT or image permutation, to implement the nonlocal filtering.
- *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.

---

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

8. 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.

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.

Catté, F., Lions, P.-L., Morel, J.-M., and Coll, T. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Numerical Analysis*, 29(1):182–193.

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. 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.

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.

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

Weickert, J. (1997). Recursive separable schemes for nonlinear diffusion filters. In ter Haar Romeny, B., Florack, L., Koenderink, J., and Viergever, M., editors, *Lecture Notes in Computer Science, volume 1252*, pages 260–271. Springer.

Weickert, J., ter Haar Romeny, B., and Viergever, M. (1998). Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, page to appear.

Weickert, J., Zuiderveld, K., ter Haar Romeny, B., and Niessen, W. (1997). Parallel implementations of AOS' schemes: A fast way of nonlinear diffusion filtering. In *Proceedings 1997 IEEE International Conference on Image Processing ICIP-97*, Santa Barbara, CA. Springer.

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.