Adaptive Nonlocal Filtering: A Fast Alternative to Anisotropic Diffusion for Image Enhancement
ADAPTIVE NONLOCAL FILTERING: A FAST ALTERNATIVE TO ANISOTROPIC DIFFUSION FOR IMAGE ENHANCEMENT

Bruce Fischl and Eric Schwartz

November 1996

Technical Report CAS/CNS-96-033

Permission to copy without fee all or part of this material is granted provided that: 1. the copies are not made or distributed for direct commercial advantage, 2. the report title, author, document number, and release date appear, and notice is given that copying is by permission of the BOSTON UNIVERSITY CENTER FOR ADAPTIVE SYSTEMS AND DEPARTMENT OF COGNITIVE AND NEURAL SYSTEMS. To copy otherwise, or to republish, requires a fee and/or special permission.

Copyright © 1996

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

Keywords: Anisotropic Diffusion, Nonlinear Adaptive Filtering, Image Enhancement, Active and Real-Time Vision.

Bruce Fischl†
Dept. Cognitive and Neural Systems
Boston University
677 Beacon Street
Boston, MA 02215
e-mail: fischl@cns.bu.edu

Eric Schwartz‡
Dept. Cognitive and Neural Systems
Boston University
677 Beacon Street
Boston, MA 02215
e-mail: eric@thing4.bu.edu

† Author to whom reprint requests should be made.
‡ Supported in part by the Office of Naval Research (ONR N00014-95-1-0409).

Keywords: Anisotropic Diffusion, Nonlinear Adaptive Filtering, Image Enhancement, Active and Real-Time Vision.

Abstract - The goal of many early visual filtering processes is to remove noise while at the same time sharpening contrast. An historical succession of approaches to this problem, starting with the use of simple derivative and smoothing operators, and the subsequent realization of the relationship between scale-space and the isotropic diffusion equation, has recently resulted in the development of "geometry-driven" diffusion. Nonlinear and anisotropic diffusion methods, as well as image-driven nonlinear filtering, have provided improved performance relative to the older isotropic and linear diffusion techniques. These techniques, which either explicitly or implicitly make use of kernels whose shape and center are functions of local image structure, are too computationally expensive for use in real-time vision applications. In this paper, we show that results which are largely equivalent to those obtained from geometry driven diffusion can be achieved by a process which is conceptually separated into two very different functions. The first involves the construction of a vector-field of "offsets", defined on a subset of the original image, at which to apply a filter. The offsets are used to displace filters away from boundaries to prevent edge blurring and destruction. The second is the (straightforward) application of the filter itself. The former function is a kind of generalized image skeletonization; the latter is conventional image filtering. This formulation leads to results which are qualitatively similar to contemporary nonlinear diffusion methods, but at computation times that are roughly two orders of magnitude faster; allowing application of this technique to real-time imaging. An additional advantage of this formulation is that it allows existing filter hardware and software implementations to be applied with no modification, since the offset step reduces to an image pixel permutation, or look-up table operation, after the application of the filter.

1. Introduction.

Many early vision systems employ some type of filtering in order to reduce noise and/or enhance contrast in regions which 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, 1996a) or (ter Haar Romeny, 1994) for a more complete discussion of these issues).

An alternative branch of nonlinear diffusion research was developed by Nitzberg and Shiota (Nitzberg and Shiota, 1992). Instead of formulating the problem as a diffusion process, they constructed a nonlinear filter whose form and location, relative to the pixel being filtered, are functions of local image structure. The shape of their filter is narrow and elongated in the direction orthogonal to locally coherent gradients, and compact in regions of changing gradient direction in order to preserve corners and triple points.

The use of a nonlinear filter by itself addresses some of the issues of noise reduction. However, if the filter straddles an edge, it still results in blurring and hence either displacement or destruction of region boundaries. In order to alleviate this problem, 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 the offset field they proposed works well in many situations, we have found that it does not perform adequately in images which contain edges at different scales unless it is applied iteratively, a computationally prohibitive process. Since their nonlinear filter implicitly 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 functions of vector offset from image filtering, a much simpler and faster class of algorithms is provided. These algo-
rithms are shown to have better performance than the original Nitzberg-Shiota method, and to provide results comparable to nonlinear anisotropic diffusion methods, with a speed-up of roughly two orders of magnitude.

The use of offset vectors obviates the necessity of working with multi-scale filters, or of determining the correct "scale" to be applied at different locations in the image. Computationally, this is a critical issue. Nitzberg and Shiota were able to obtain results which are comparable to anisotropic diffusion by applying relatively large filters iteratively to an image (e.g. 4 applications of an 11x11 filter). However, as is the case for the nonlinear diffusion processes, the resulting procedure is far too time consuming for use in real-time vision systems. To alleviate this concern, we derive a method which provides anisotropic diffusion-quality image enhancement through a single application of a 3x3 filter coupled with an offset field. This is achievable because the offset vector field essentially introduces an adaptive scale parameter via the magnitude of the offset vectors.

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, to produce the final result by a simple pixel permutation. This technique permits conventional hardware (e.g. fast 3x3 filtering) and existing code to be applied, unchanged, to produce results which appear comparable to the much more computationally expensive nonlinear diffusion methods. From a conceptual point of view, we provide a different interpretation of the role of "scale" in image processing. This is the case as there is no explicit scale-space structure in this method, yet the method performs well on multi-scale image features due to the adaptive scale in the form of the offset vector magnitude noted above. In addition, the modularization of this method in terms of a separate generalized skeletonization operation, coupled with a very simple single scale (but nonlocally applied) image filter, should allow for efficient and easy development of hardware and further improved algorithmic aspects of this procedure.

We will briefly summarize the basic idea of offset image filtering below, then outline and compare our technique to the Nitzberg-Shiota algorithm. Next, an example of a class of images for which their method fails to perform adequately is given. Finally, we compare the results of the offset filtering to nonlinear anisotropic diffusion.

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 which 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 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 is judged to be "closer". Nitzberg and Shiota (Nitzberg and Shiota, 1992) proposed a method based on gradient direction as well as magnitude which performs well in many cases, but fails for images which contain edges at a number of scales, as we will show below. First, however, we briefly outline anisotropic diffusion approaches to these problems, and its relationship to the proposed nonlinear filtering technique.


An alternative to nonlinear filtering is to selectively blur an image, modulating the amount of blurring based on local gradient information. 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. For example, Nitzberg and Shiota have derived an integro-differential equation which is equivalent to the application of their adaptive filter (Nitzberg et al., 1993).

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, 1996a; Fischl and Schwartz, 1996c). In this work, we obtained an approximation of the nonlinear analog of a "Greens Function" for the original partial differential diffusion equation. This resulted in the replacement of 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". That is, we were able 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". It is instructive at this point to summarize some findings from our earlier study, since this motivated 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. 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.

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 (3.1). 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!

Typically, the diffusion kernels had the appearance of delta functions located exactly on edge features in the image, and gradually transformed into rotationally asymmetric truncated Gaussians as one moved away from a strong edge. Figure (3.1) 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 in this figure, 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. It is clear from this observation 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. The only remaining implementation issue following this observation is the calculation of an appropriate offset vector field.

4. Offset Vector Field Computation.

Nitzberg and Shiota employed an adaptive filter, using local image structure to module the shape of the filter to average along an edge, but not across it. However, if the kernel is symmetric around the central pixel, some averaging of edge values must occur (see figure (4.1)). 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 the kernel 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 offset 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 issue 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 estimate 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. The magnitude should 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 generated by applying a Sobel operator to a Gaussian smoothed (\( \sigma = 2 \)) image.

- Offset orientation \( O(z) \).
- Offset direction \( d(z) \).

FIGURE 3.1. Typical diffusion kernels. Left: original image with diffusion kernels shown together with their location. Right: image after applying kernels to original.

![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.](image)
Offset magnitude, \( m(z) \).

Symbolically the offset calculation can be written in terms of these quantities as

\[
v(z) = m(z) d(z) \begin{bmatrix} O(z) \\ O(z) \end{bmatrix}, \quad z = \begin{bmatrix} x \\ y \end{bmatrix}^T \quad (4.1)
\]

Where \( O(z) \) is the offset orientation, and refers to the choice of the normal vector. Since the sign of the vector has yet to be determined, we constrain its angle to be in the range \([0, \pi]\). Offset direction \( d(z) \) is a binary value \((1 \text{ or } -1)\) 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+\( \pi \)). Finally, offset magnitude \( m(z) \) encodes the length of the offset vector.

The orientation and direction are computed first via

\[
O(z) = \int_W s(\nabla I(z+z')) \nabla I(z+z') \, dz',
\]

\[
s(z) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}
\]

\[
d(z) = \text{sgn} \left( \int_W \begin{vmatrix} O(z) \cdot \nabla I(z+z') \end{vmatrix} (O(z) \cdot z') \, dz' \right) \quad (4.2)
\]

\[
W \text{ is a window around the point } z=(x, y)^T, \text{ typically } 3x3, \text{ and } z'=(x', y')^T \text{ is the vector from the central point } z \text{ to each point in the window. The orientation is basically a vector sum of gradients in the window } W \text{ with the function } s \text{ used to limit the orientation to the range } [0, \pi]. \text{ The direction calculation is made up of two terms. The first discounts gradients which are not in the orientation direction, and are therefore either caused by noise or a corner; while the second simply causes edges in one direction to push the offset vector in the opposite direction.}
\]

Once the orientation and direction of the vector field have been determined, the magnitude is calculated via a one dimensional 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 less than or equal to zero). This indicates that the vector field has either vanished, signifying the interior of a region, or has changed orientation by at least 90°, 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

\[
v_i(z) = d(z) O(z) \quad (4.4)
\]

Then, 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.

\[
m(z) = \min \alpha : v_i(z+\alpha v_i(z)) \cdot v_i(z) \leq 0 \quad (4.5)
\]

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

\[
v(z) = m(z) \frac{v_i(z)}{|v_i(z)|} \quad (4.6)
\]

In contrast, Nitzberg and Shiota bundle the generation of their offset vector field into a single calculation given by

\[
v(z) = \int_W \frac{1}{|z|} (\nabla I(z+z') \cdot z') \nabla I(z+z') \, dz' \quad (4.7)
\]

\[
\phi(v) = \frac{c v}{\sqrt{\mu^2 + \|v\|^2}} \quad (4.8)
\]

Where \( p \) is a Gaussian and \( \phi \) is a vector valued compressive nonlinearity, used to limit the length of the displacement vectors, with \( c \) specifying the maximum length, and \( \mu \) modulating the slope of the compression. The resulting offset vector field is smooth with a zero-crossing in magnitude around the estimated center of an edge, so that the vector field reverses direction as a point changes from “interior” to “exterior”. However, the smoothness of their vector field is a serious drawback. It causes their vector field to have small magnitude throughout the interior of broad edges, resulting in offsets which are insufficient to displace a filter out of the edge region.

In order to see the efficacy of the search mechanism for adaptively determining the appropriate offset magnitude,

---

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

---

**FIGURE 4.2** Test image which contains both large scale edges (the circle) as well as small-scale image structure (the rectangle).
vis-a-vis the Nitzberg-Shiota approach, consider figures (4.2)-(4.4). Figure (4.2) depicts an image which contains both large and small-scale image features in the form of a circle and a rectangle respectively. Figure (4.3) illustrates the form of the Nitzberg-Shiota vector field in these regions. In order to generate substantial displacements on the border of the circle (top right), it is necessary to set the parameter $c$ in equation (4.8) to such a large value, that the vectors in the vicinity of the rectangle are magnified to extend entirely out of the rectangle. Subsequent filtering using this offset field (bottom) results in destruction of the rectangle while the circle is only mildly enhanced.

![FIGURE 4.3. Top: Nitzberg-Shiota offset field in the center of the rectangle (left) and on the right edge of the circle (right). Bottom: the result of a 3x3 offset median filter.](image)

In contrast, the search mechanism we employ enables the offset magnitude to be determined adaptively by local image structure in the offset direction. This permits the vectors on the border of the circle to grow to the length necessary to extend entirely out of the border region (figure (4.4), right), while the offsets around the rectangle are constrained by the small scale image structure in that region (figure (4.4), left). The subsequent filtering results in well-defined boundaries for both the circle and the rectangle (figure (4.4), bottom).

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), Witken (Witken, 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.

![FIGURE 4.4. Top: adaptive length offset vector field computed via equations (4.1)-(4.6) in the center of the rectangle (left) and on the right edge of the circle (right). Bottom: the result of a 3x3 offset median filter.](image)

5. Postprocessing with displacement vector fields.

The most straightforward implementation of offset filtering is to apply the displaced filter directly to the (offset) 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. 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.
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.

6. Results.

In this section we present a comparison of an offset median filter with 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, although other filters such as a simple Gaussian also work well with the offset field. 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 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.

The offset computation slows down the median filter by 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, 1996a; Fischl and Schwartz, 1996c), which was itself roughly an order of magnitude faster than the nonlinear diffusion process. Running on a 50 MHz Sparc-10, a 3x3 median filter applied to a 256x256 pixel image requires approximately 0.75 seconds. Using the displacement vector field increases the time to roughly 5.5 seconds, while the 100 time steps used to integrate the anisotropic diffusion necessitate about 450 seconds. In addition, the offset median frequently outperforms the anisotropic diffusion in terms of image quality. This is illustrated in figure (6.1). As can be seen in this image, the anisotropic diffusion results in both too much blurring of the state name and the rightmost characters, as well as the retention of speckle noise in the periphery. In contrast, the offset median filtered image in the middle has retained small-scale image structure in the form of the state name as well as providing substantial noise-reduction and sharpening of large-scale features such as the license plate itself.


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. In fact, we have found this method to hold up well for a wide variety of images, and in all cases to provide very significant improvements in speed of computation. When combined with space-variant vision representations (e.g. Rojer and Schwartz, 1990), it is possible to achieve frame-rate enhancement, providing 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, 1996a; Fischl and Schwartz, 1996c) 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.
- **Algorithm design**: By separating the process into a generalized skeletonization (i.e. determine the orientation, 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, i.e. 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 orientation, direction and magnitude) with conventional image filtering, seem to offer a fertile area for future development.

8. Bibliography.


