

An Adaptive Window Mechanism for Image Smoothing

Ardeshir Goshtasby and Martin Satter

Abstract

Image smoothing using adaptive windows whose shapes, sizes, and orientations vary with image structure is described. Window size is increased with decreasing gradient magnitude, and window shape and orientation are adjusted in such a way as to smooth most in the direction of least gradient. Rather than performing smoothing isotropically, smoothing is performed in preferred orientations to preserve region boundaries while reducing random noise within regions. Also, instead of performing smoothing uniformly, smoothing is performed more in homogeneous areas than in detailed areas. The proposed adaptive window mechanism is tested in the context of median, mean, and Gaussian filtering, and experimental results are presented using synthetic and real images and compared with a state-of-the-art method.

Keywords: Image smoothing; Adaptive smoothing; Nonlinear filtering; Median filtering; Mean filtering; Gaussian filtering

1 Introduction

Many computer vision tasks require smoothing as a preprocessing operation to reduce image noise. Noise is a function of time and to reduce it smoothing is required in the temporal domain. However, because often only one image is available, smoothing is performed in the spatial domain. To calculate the smoothed value at a pixel, often pixel

intensities in a square neighborhood centered at the pixel are averaged. This is despite the fact that image structure within a square window is usually non-symmetric. Even in so-called anisotropic smoothing square neighborhoods are used, though pixels within the window are given different weights. In this paper, the idea of adaptive windows in image smoothing is explored. Window shape, size, and orientation are adjusted to local image structure to smooth more along the region boundaries than across them.

An example of image smoothing is given in Fig. 1. Fig. 1a depicts one frame in a sequence of video frames of a static scene obtained by a stationary camera. The presence of random noise in the image is evident. By averaging 30 consecutive frames in the video, the image shown in Fig. 1b is obtained. This is smoothing in the temporal domain. Averaging or smoothing in the temporal domain reduces noise while preserving image structure. If we obtain the Fourier transforms of Figs. 1a and 1b, we see that the lower Fourier transform coefficients in both images are very similar. There are, however, considerable differences between higher Fourier transform coefficients in the two images as shown in Figs. 1c and 1d. It is important to note that some high spatial frequencies remain when performing smoothing in the temporal domain. Those high spatial frequencies represent sharp edges in Fig. 1b. Therefore, smoothing in the temporal domain preserves sharp edges. Isotropic smoothing in the spatial domain, on the other hand, eliminates high spatial frequencies that represent both noise and image structure. We would like to reduce high spatial frequencies that are due to noise but not due to image structure.

If we smooth Fig. 1a spatially with Gaussian filters of standard deviations 1 and 2 pixels, we obtain the images shown in Figs. 2a and 2b, respectively. Image structure in

Fig. 2a is preserved, but so is image noise. Noise has been considerably reduced in Fig. 2b, but critical image structure has also been lost. If we obtain the Fourier transforms of images 2a and 2b, we see that spatial smoothing eliminates critical high spatial frequencies that represent image structure. The Fourier transform coefficients in Fig. 2d no longer resemble those in Fig. 1d.

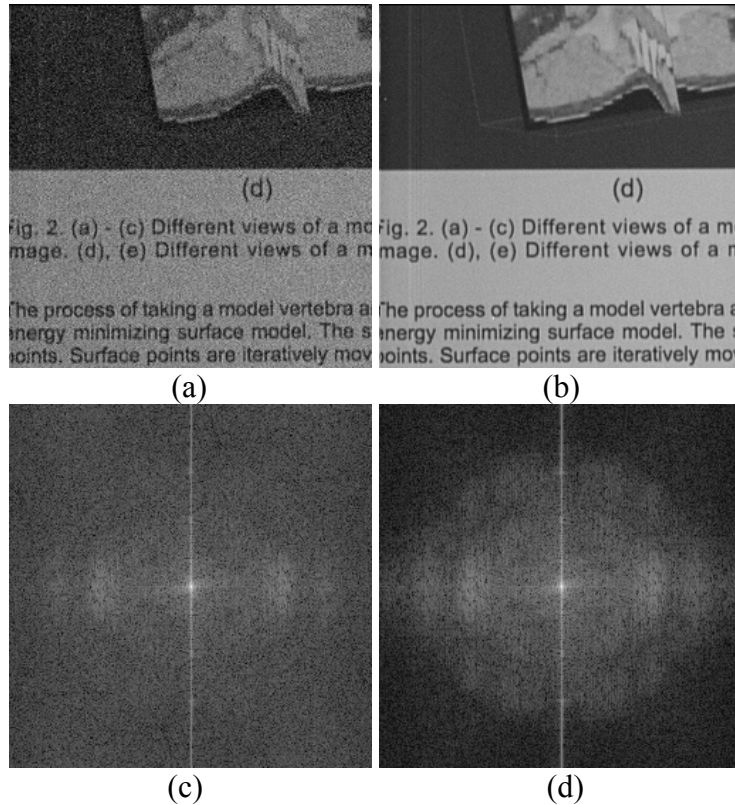


Fig. 1. (a) One frame of a static scene in a video sequence obtained by a stationary camera. (b) Smoothing in the temporal domain obtained by averaging intensities of 30 consecutive frames captured by the camera. (c), (d) Log Fourier transform magnitudes of (a) and (b), respectively.

To preserve high spatial frequencies from image structure while reducing random noise, instead of using uniform and isotropic smoothing, non-uniform and anisotropic smoothing is performed. To achieve this, the square window traditionally used is replaced with a rectangular window whose dimensions and orientation vary with local image structure. The proposed adaptive window mechanism will contain more pixels along

edges than across them. An image smoothed with the proposed window mechanism will retain its structure while smoothing random noise.

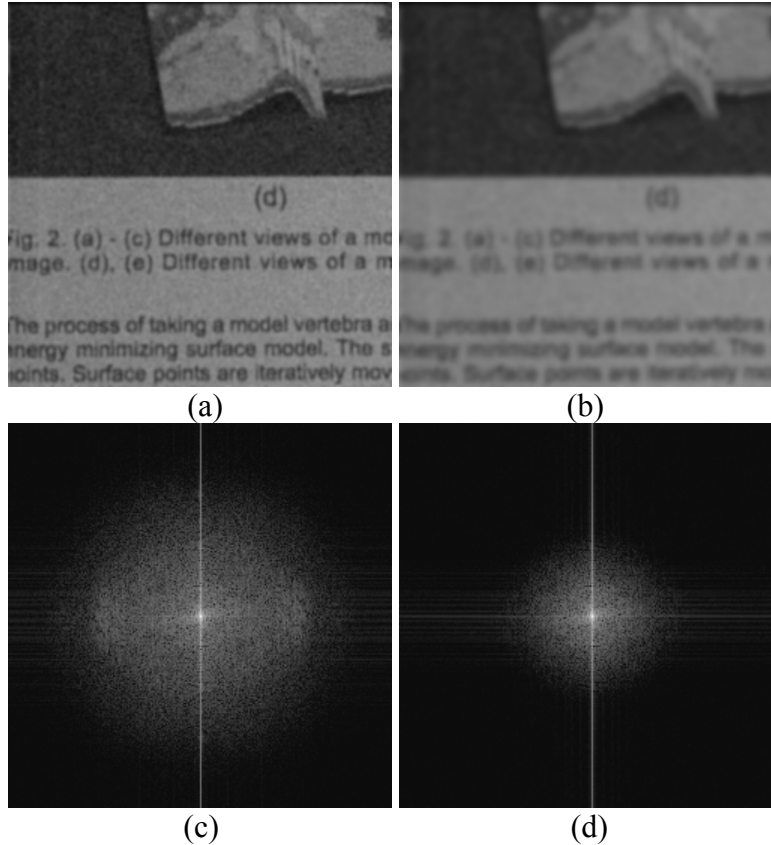


Fig. 2. (a), (b) Smoothing using Gaussians of standard deviations 1 and 2 pixels, respectively. (c), (d) Log Fourier transform magnitudes of (a) and (b), respectively.

An example of image smoothing by the proposed method is given in Fig. 3. Figs. 3a and 3b show Fig. 1a after being smoothed with the proposed adaptive window mechanism. The logarithm Fourier transform magnitudes of the smoothed images are shown in Figs. 3c and 3d. The presence of large high spatial frequency coefficients in the transformed images is evidence that some high spatial frequencies are retained after adaptive smoothing. These high spatial frequencies represent structural details preserved after smoothing. When comparing the Fourier transform coefficients of the image after

temporal smoothing with those after adaptive smoothing, we observe similarity between the two.

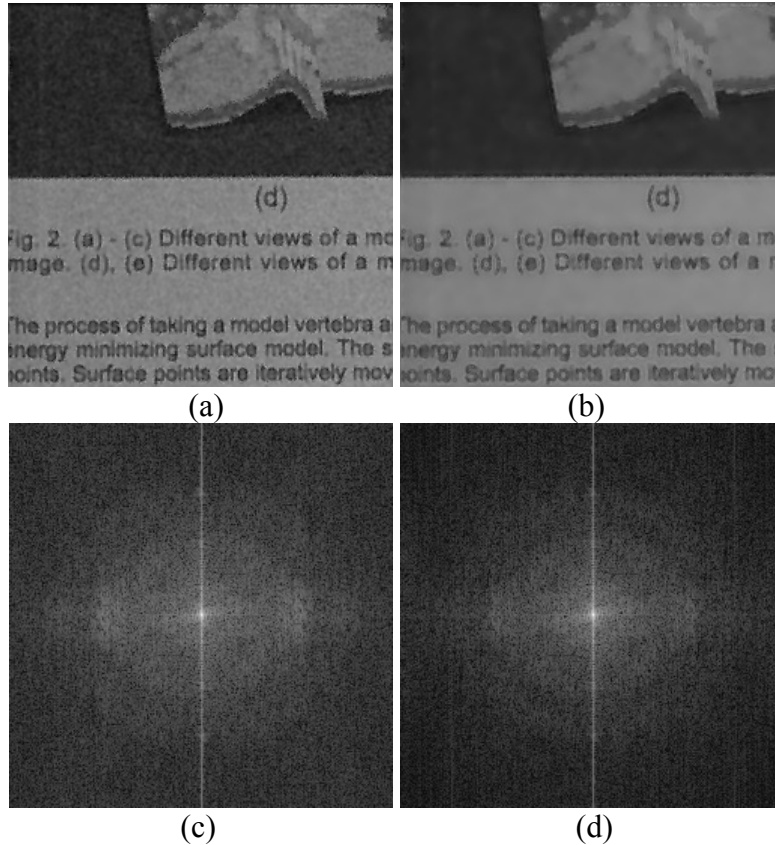


Fig. 3. (a), (b) Image smoothing using Gaussians of standard deviations 1 and 2 pixels in the context of the proposed adaptive window mechanism. (c), (d) Log Fourier transform magnitudes of the smoothed images.

In the remainder of this paper, first, related work in adaptive smoothing is reviewed. Then, the proposed adaptive window mechanism is described and implementation strategies are detailed. Next, experimental results on synthetic and real images are given and compared with a state-of-the-art method. Finally, concluding remarks are made. In this paper, smoothing and filtering are used as synonyms.

2 Related Work

Various adaptive image smoothing methods have been developed throughout years with early surveys provided by Nagao and Matsuyama [40] and Chin and Yeh [12]. Some adaptive smoothing methods are based on prior statistics or knowledge of the image to be smoothed. Lin et al. [34], Salembier [51], and Yin et al. [64] achieve adaptation by comparing the filter output with a reference image and minimizing the difference between the two. These methods assume that a representative sample of the image to be smoothed is available. Frost et al. [18], Lin and Wilson [35], and Restrepo and Bovik [47] collect image statistics and then determine the weights in a neighborhood for smoothing using the local statistics. Heuristics are used to collect the statistics. Roth and Black [48] describe a method for estimating image priors that capture the statistics of natural scenes and use the statistics in adaptive smoothing. A method described by Portilla et al. [45] decomposes an image into a large set of wavelets at different orientations and scales, modifies the wavelet coefficients based on their prior probabilities, and creates the smoothed image by the inverse wavelet transform.

A smoothing method that estimates and uses the probability density function from image data is known as mean-shift filtering and is based on the mean-shift pattern recognition technique of Fukunaga and Hostetler [19] and its application to mode detection in data [10]. Comaniciu and Meer [13] further develop and use the mean-shift idea in adaptive smoothing.

Another method, by Kang and Roh [29], also has its roots in pattern recognition. The method treats image smoothing as a Bayesian labeling problem with the solution obtained by finding the maximum a-posteriori probability of the true labeling. Markov

Random Field (MRF) theory is used to encode contextual image constraints into the prior probabilities. A parameter modification algorithm is introduced that can enhance the performance of the piecewise smooth model for color images in a discontinuity adaptive MRF modeling. Incorporation of edge information into the MRF modeling is meant to preserve object boundaries while smoothing random noise.

Energy-minimizing and optimization approaches to adaptive smoothing also exist in the literature. In an energy-minimizing model, Kervrann [30,31] keeps the window shape square but allows the window size to vary. At each pixel, a regression function is estimated by iteratively growing the window size and adaptively weighting the image intensities until a minimum is reached in an energy function that is defined in terms of bias and variance [20]. Mumford and Shah [38], Rudin et al. [49], and Chan et al. [9] describe other optimization-based smoothing methods.

Edge-preserving smoothing can be achieved via anisotropic diffusion also. By setting conduction coefficients within regions to 1 and conduction coefficients on the boundary between regions to 0, a diffusion process can be created that smoothes region interiors independently and without interaction with neighboring regions. However, since the boundary between regions is usually not known, instead of setting the conduction coefficients to 0 and 1, Perona and Malik [42] suggested setting the conduction coefficient at a pixel to a function of the intensity gradient magnitude at the pixel, thereby creating a process that smoothes more in low-gradient neighborhoods than in high-gradient neighborhoods.

Diffusion processes achieve smoothing by solving partial differential equations (PDEs) iteratively. You et al. [65] show that anisotropic diffusion is the steepest descent

method for solving a minimization problem. Weickert [60] describes various PDE-based iterative techniques for image smoothing and enhancement. Iterative methods are inherently unstable and their performances depend on the computational architecture used. Some drawbacks and limitations of the method of Perona and Malik have been noticed. The ill-posedness of the diffusion equation is shown by Catté et al. [8], while Whitaker and Pizer [61] notice a staircase effect of the process when the conductance parameter is not properly set. Li and Chen [33] propose varying and adapting the conductance parameter rather than keeping it fixed. Chen [11] provides a novel implementation of the nonlinear anisotropic fusion of Perona and Malik by introducing a contextual discontinuity measure. Alvarez et al. [1] and Aubert and Vese [3] introduce well-posed solutions to anisotropic smoothing, overcoming some of the difficulties of Perona and Malik's method. Black et al. [6] show that anisotropic diffusion can be considered a robust estimator for a piecewise smooth image from a noisy one. They develop new anisotropic fusion equations that produce improved edge continuity compared to the equations developed by Perona and Malik. Also related to diffusion smoothing are the coupled anisotropic diffusion of Tschumperlé and Deriche [58] and the method of Ashino et al. [2], which works in the Fourier domain. Gerig et al. [21] demonstrate the application of anisotropic diffusion in smoothing 2-D and 3-D spin-echo and gradient-echo magnetic resonance (MR) images.

A number of methods have been developed that achieve adaptive smoothing by appropriately weighting the intensities within square windows. Boulton et al. [7] introduce the "G-neighbors" idea where neighboring pixels that are similar enough to the center pixel are averaged to replace the value at the center pixel. Since only pixels adjacent to

the center pixel are considered in averaging, iterations are needed to include larger neighborhoods in smoothing. Methods to find the weighted average of intensities within sliding windows while changing the weights according to differential [22, 59] and statistical [32] measures have been proposed as well.

Also in the category of smoothing by weighted averaging is the method of Overton and Weymouth [41], which multiplies geometric and photometric terms with rational weights. The weights are functions of the intensity and geometric distances between the center pixel and the neighboring pixels. This method works well on images containing impulse noise, but it may not do particularly well when white noise is present as no particular intensity may be dominant in a neighborhood. Tomasi and Manduchi [57] follow the same idea but use Gaussians rather than rational weights and call the method *bilateral filtering*. Theoretical foundations of bilateral filtering are provided by Elad [15], showing that bilateral filtering can be interpreted as a Bayesian approach and as a single iteration of some well-known iterative algorithms. Use of bilateral filtering in edge detection in a spiral representation is explored by Hu et al. [26], where edge maps of an image and its successively smoothed versions are used to produce the final edge image. Pham and van Vliet [43] provided a separable fast implementation of bilateral filtering and Durand and Dorsey [14] proposed a piecewise linear approximation to the process, achieving two orders of magnitude in speed.

Barash [4] and Barash and Comaniciu [5] compared and related nonlinear diffusion, bilateral filtering, and mean-shift filtering. By extending neighborhoods, nonlinear diffusion in adaptive smoothing is achieved. The extended nonlinear diffusion process is converted to bilateral filtering using a choice of weights that depends on their

relative positions. Bilateral filtering is also related to mean-shift filtering in edge-preserving smoothing when using the local modes in the joint spatial-range domain. The main difference between bilateral filtering and mean-shift filtering is considered to be in the way local image information is used.

A class of smoothing methods that preserves image edges is based on surface estimation. Sinha and Schunck [53] described a discontinuity-preserving surface estimation that first cleans the data and then fits a discontinuity-preserving spline surface to the data. Terzopoulos [56] describes a multi-resolution iterative algorithm for surface reconstruction involving discontinuities. Qiu [46] estimates jump surfaces by local piecewise linear smoothing. Gijbels et al. [22] uses a similar method but adapts the neighborhoods based on image gradients. Other discontinuity-preserving smoothing methods that are based on surface estimation are described by Stevenson and Delp [55] and Yi and Chelberg [63].

Also related to surface fitting is the method of Meer et al. [37], which uses the facet model of Haralick and Watson [25]. This method finds all windows of a given size (say 3×3) that share a pixel, and a polynomial is fitted to intensities in each window. The smoothed value of the pixel is then taken to be the value of the polynomial obtained from the window producing the smallest residual variance when evaluated at the pixel. Computations are performed at three resolutions using 3×3 , 5×5 , and 7×7 windows, and at each pixel, the window most closely estimating a constant patch is selected to represent the smoothed value at the pixel. Related to this idea is the work of Polzehl and Spokoiny [44], which considers an image a combination of homogeneous regions and treats image smoothing as a regression problem. It determines the form of the neighborhood around a

pixel where the function can be closely approximated by a constant, thereby adapting smoothing to local image structure. Muneyasu et al. [39] propose windows with varying weights determined via a fuzzy inference model that adapts to local image gradients. Less smoothing is applied to one side of a window than to other sides by appropriately assigning the weights, thereby preserving the edges while smoothing image noise.

A method that changes the window shape but not its orientation is described by Saint-Marc et al. [50]. This method decomposes a 2-D Gaussian $G(x,y)$ into two 1-D Gaussians $G(x)$ and $G(y)$. The standard deviation of $G(x)$ is then set inversely proportional to the gradient magnitude in the x -direction, and the standard deviation of $G(y)$ is set inversely proportional to the gradient magnitude in the y -direction. As a result, a 2-D Gaussian is allowed to stretch horizontally or vertically depending on the relative horizontal and vertical gradient magnitudes. This smoothing works well in images of scenes containing horizontal and vertical edges; however, the process does not perform particularly well at and near edges that are diagonal.

A method that has some similarity with the method proposed here is that proposed by Gürelli and Onural [24]. This method considers anisotropic smoothing as the problem of estimating the direction of highest coherence at each pixel, from among eight discrete directions, and employing a 1-D window that approximates a line segment in that direction. Another method that has some similarity with the method proposed here is introduced by Xu et al. [62], which achieves adaptive smoothing using a 3×9 window oriented in the direction of least variance, from among sixteen possible directions. Also of some similarity to our method is the method of Schulze and Wu [52], which calculates a coefficient of variation at each pixel and adjusts the filter size based on the obtained

coefficients, smoothing areas with lower spatial variations more than areas with higher spatial variations.

3 Approach

To maintain image structure while reducing random noise, we propose the use of windows whose sizes, shapes, and orientations change with local image structure. This idea is depicted in Fig. 4. Instead of using square windows, we use rectangular windows whose dimensions and orientations adapt to local image details. Near a region boundary, the rectangular window becomes narrow and small and aligns with the boundary. In a homogeneous area, the window becomes square and large. An adaptive window contains fewer pixels when it is in a detailed area than in a homogeneous area, and the pixels used in smoothing lie more along region boundaries than across them.

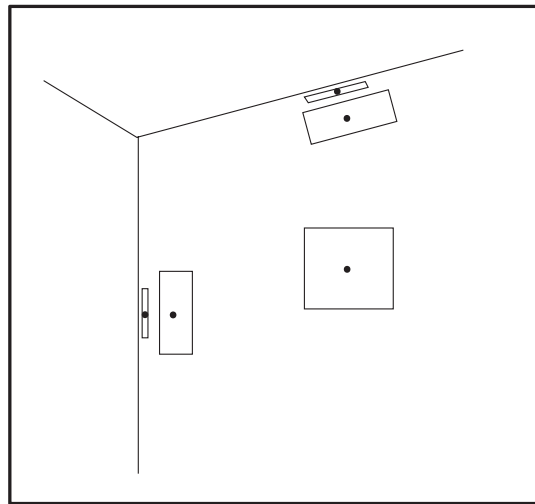


Fig. 4. The adaptive window mechanism concept: The window used to smooth an image at a pixel is oriented in such a way that its long side (width) is in the direction of minimum gradient and its dimensions are inversely proportional to the minimum and maximum gradient magnitudes at the pixel.

When median or mean filtering is needed, the width (W) and height (H) of the window are related to the minimum and maximum gradient magnitudes at a pixel by

$$W = a/(g_n+1), \quad (1)$$

$$H = a/(g_m+1), \quad (2)$$

where g_n and g_m are the minimum and the maximum gradient magnitudes at the pixel. The addition of 1 in the denominator is to avoid division by zero. Parameter a is the proportionality term. When $g_n = g_m = 0$, a can be considered the length of the side of the square window needed to smooth homogeneous areas in an image. It is obvious that a larger a should be selected for a noisier image than for a less noisy image.

Near an object boundary, although gradient magnitude in the direction normal to the boundary may be high, gradient magnitude in the direction along the boundary will be low, producing a window whose long side aligns with the boundary and whose short side becomes normal to the boundary. As the window gets closer to the boundary, it gets narrower and smaller. This mechanism prevents information on the two sides of the boundary from merging and, consequently, keeps the object boundaries sharp.

When Gaussian filtering is needed, knowing that a 2-D Gaussian can be decomposed into two 1-D Gaussians,

$$G(x,y) = G(x) \times G(y), \quad (3)$$

we let the standard deviations of the 1-D Gaussians at a pixel be inversely proportional to the minimum and maximum gradient magnitudes at the pixel. If σ_x and σ_y represent the standard deviations of $G(x)$ and $G(y)$, respectively, we let

$$\sigma_x = a/2(g_n+1), \quad (4)$$

$$\sigma_y = a/2(g_m+1), \quad (5)$$

and will orient the 1-D Gaussians in such a way that the wider Gaussian will be in the direction of minimum gradient and the narrower Gaussian will be in the direction of

maximum gradient. Equations (4) and (5) relate parameters of an anisotropic 2-D Gaussian to local image gradients. The Gaussian gets smaller as local gradient increases, and it gets narrower near an edge since gradient magnitude across the edge is larger than gradient magnitude along the edge. The 2-D Gaussian will be oriented in such a way that when decomposed, the wider of the two 1-D Gaussians aligns with the direction of minimum gradient and the narrower 1-D Gaussian will align with the direction of maximum gradient. The global parameter a , which is controlled by the user or an external process, scales the automatically determined parameters σ_x and σ_y . Parameter a allows the user to increase or decrease the amount of smoothing. There may be a need to increase global smoothing due to the presence of a high level of noise in an image, or reduce smoothing globally if the image is known to contain very little noise.

Next, the implementation details of the proposed window mechanism are provided.

4 Implementation

If the direction of maximum gradient at a pixel with coordinates (x_i, y_i) is θ_i and the direction normal to it is α_i , and if window dimensions at the pixel calculated by (1) and (2) are W and H , we center a rectangular window at the origin with its width in the X direction and its height in the Y direction. Next, we rotate the window by α_i , translate it by (x_i, y_i) , multiply its entries with corresponding image intensities, and save the result at (x_i, y_i) in the smoothed image. This process is graphically depicted in Fig. 5.

In Fig. 5, assuming coordinates of points in the rightmost coordinate system are denoted by (x, y) and coordinates of points in the leftmost coordinate system are denoted by (X, Y) , the relation between points in the two coordinate systems can be written as

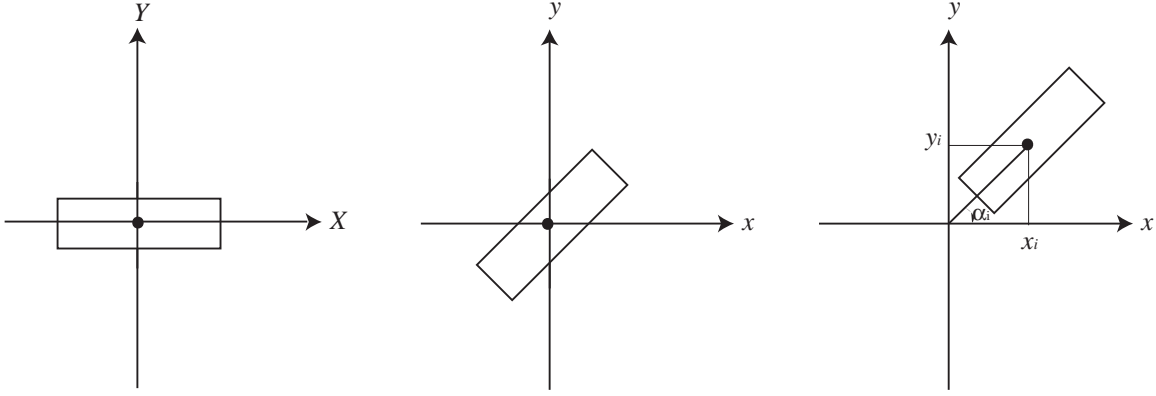


Fig. 5. Implementation of the adaptive window mechanism: Consider a window with desired dimensions at the origin. Rotate the window by $\alpha_i = \theta_i + \pi/2$, where θ_i is the direction of maximum gradient at (x_i, y_i) . Then, translate the window to (x_i, y_i) , multiply window entries with the corresponding image intensities, and add the values to obtain the intensity at (x_i, y_i) in the smoothed image.

$$x = X\cos(\alpha_i) - Y\sin(\alpha_i) + x_i, \quad (6)$$

$$y = X\sin(\alpha_i) + Y\cos(\alpha_i) + y_i. \quad (7)$$

Equations (6) and (7) make it possible to determine, for each entry (X, Y) in the window, the corresponding pixel in the image. Corresponding image and window values are multiplied and added together to obtain the smoothed image intensity. Although (X, Y) are integers, (x, y) will be floating point numbers. To determine the correspondence between image pixels and window entries, either (x, y) are rounded or image intensity at (x, y) is computed from the bilinear interpolation of intensities of the four pixels surrounding (x, y) .

In median filtering, all that is needed is to determine pixels in the image that belong to the four corners of the window and find the median of pixels on and inside the rectangular window defined by the four corners. In mean filtering, the average of pixels on and inside the window is computed. Therefore, in both median and mean filtering, there is no need to actually find the correspondence between image pixels and window

entries. It is sufficient to locate window corners in the image and use pixels on or inside the window defined by the four corners to perform the smoothing. This is limited to windows whose entries are treated similarly though. In Gaussian smoothing, since weights in a window vary with their distances to the window center, correspondence between window entries and image pixels as defined by equations (6) and (7) need to be determined.

In adaptive Gaussian smoothing, window coordinates and image coordinates are related by

$$X = (x-x_i)\cos(\alpha_i) + (y-y_i)\sin(\alpha_i), \quad (8)$$

$$Y = -(x-x_i)\sin(\alpha_i) + (y-y_i)\cos(\alpha_i). \quad (9)$$

A 2-D Gaussian can be written by a combination of two 1-D Gaussians:

$$G(X, Y) = \exp\left\{-\frac{X^2}{2\sigma_x^2}\right\} \exp\left\{-\frac{Y^2}{2\sigma_y^2}\right\}. \quad (10)$$

Since a Gaussian extends from $-\infty$ to ∞ , to compute the smoothed value at (x_i, y_i) , for each pixel (x, y) in the image, the corresponding window entry (X, Y) is determined from (8) and (9). Coordinates (X, Y) correspond to coordinates (x, y) after being transformed such that (x_i, y_i) lies at the origin and the X -axis aligns with the direction of least gradient. Intensity at (x, y) is then multiplied with the Gaussian weight at (X, Y) . The process is repeated for all pixels (x, y) in the image and the results are added to obtain the smoothed value at (x_i, y_i) . Therefore, we can either multiply the Gaussian defined by (10) with the transformed image according to (8) and (9) or transform the Gaussian according to (6) and (7) and then multiply its values with corresponding image values. Theoretically, this has to be repeated for every pixel in the image.

In practice, however, Gaussians approach zero exponentially and considering the digital nature of images, it is sufficient to let a window's width and height be equal to $6\sigma_x$ and $6\sigma_y$, respectively. Use of larger windows will not affect 99% of the smoothed values due to the digital nature of the image intensities, and the 1% of the intensities that change will change very little. Using equations (4) and (5), we relate the window dimensions for adaptive Gaussian smoothing to image gradients and parameter a by

$$W = 6\sigma_x = 3a/(g_n+1), \quad (11)$$

$$H = 6\sigma_y = 3a/(g_m+1). \quad (12)$$

W and H are integers after rounding the right-hand sides of (11) and (12). In case either W or H evaluates to an even number, the odd number closest to it is selected instead. This is done to make the windows symmetric with respect to their axes and avoid favoring one side of the window against its opposite side.

In the computation of adaptive Gaussian smoothing, horizontal window entries centered at the origin as shown by the leftmost coordinate system in Fig. 5 are mapped to the image coordinates according to (6) and (7), and smoothing is performed by multiplying filter values with image intensities and adding the results. To speed up the computations, the Gaussians can be precalculated and saved and reused as needed. Since windows have discrete dimensions, only a small number of such windows need to be precalculated and saved.

Isotropic and uniform Gaussian smoothing is very efficient because a 2-D Gaussian can be decomposed into two 1-D Gaussians and smoothing can be achieved using the same 1-D Gaussian horizontally and vertically. In adaptive smoothing, not only the size of the 1-D Gaussian varies across an image, the orientation of the Gaussian

varies. This requires reorienting a Gaussian using formulas (6) and (7) before multiplying its entries with corresponding image entries. The cost to pay for this improvement in smoothing is this additional computation time, which slows down the process by a factor of 2 to 3 compared to isotropic smoothing.

5 Results

In order to determine the behavior of the proposed adaptive smoothing, experiments were carried out using synthetic and real images. Fig. 6a shows a synthetically generated image containing four homogenous regions with intensities 50, 100, 150, and 200. Fig. 6b shows a synthetic image with intensities of the background and the three circular regions equal to 250, 150, 100, and 0. There is a small square (2×2 pixels) at the center of the smallest circle with intensity 127. Zero-mean uniformly distributed noise was added to the images and the noise-corrupted images were smoothed by the proposed adaptive smoothing in the context of median, mean, and Gaussian smoothing. The root-mean-squared (RMS) differences between an original image and its smoothed versions under different magnitudes of noise were calculated and tabulated.

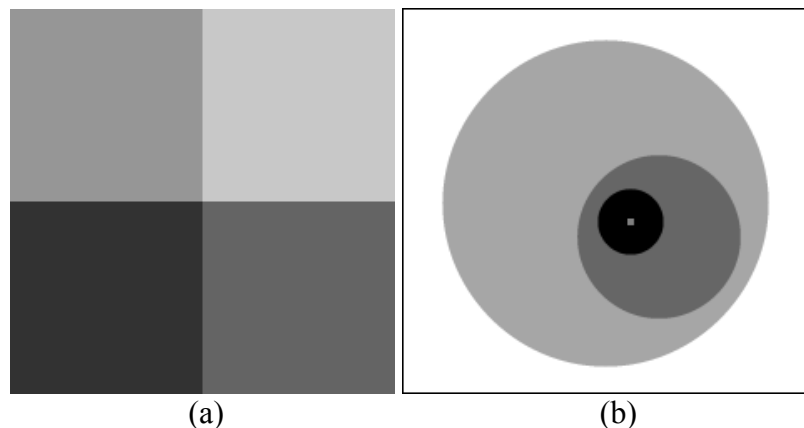


Fig. 6. (a), (b) The synthetically generated images of known intensities used to evaluate the performance of the proposed adaptive smoothing.

In a homogeneous area, we want window dimensions in both adaptive smoothing and isotropic smoothing be the same. Assuming a square window of side a pixels is used in isotropic smoothing, the same window size is obtained in adaptive smoothing when maximum and minimum gradients are both zero. In adaptive median and mean filtering, $a \times a$ will represent the largest window size used when smoothing in homogeneous areas. In adaptive Gaussian filtering, the largest window size will be $3a \times 3a$ pixels according to formulas (11) and (12). Windows used in Gaussian smoothing are larger than those used in median and mean filtering to allow the Gaussians to become sufficiently small at window borders.

5.1 Median Filtering

Median filtering is effective when an image contains impulse noise. To add impulse noise to an image, zero-mean uniformly distributed numbers in the range $[-50, 50]$ were generated and added to the intensities of 10% of randomly selected pixels in Figs. 6a and 6b to obtain Figs. 7a and 7c. Figs. 7b and 7d show images obtained after adding similar noise to 20% of randomly selected pixels in the images.

To determine the minimum and maximum gradient magnitudes at a pixel, an image is first convolved with a Gaussian to reduce the effect of noise. If some information about the noise level in an image is available, that information should be used to select the standard deviation of the Gaussian smoother. For the test images used in this section, the standard deviation of the Gaussian smoother was set either to 1 or 1.5 pixels depending on parameter a . If $a > 7$ pixels in equations (1) and (2), the standard deviation of the Gaussian smoother was set to 1.5; otherwise, it was set to 1 pixel. Note that this smoothing is performed only to calculate the gradient magnitude and direction at

a pixel to adjust the size and orientation of the smoothing window. Therefore, once the size and orientation of a window are selected, smoothing is performed on the original image and not on its smoothed version.

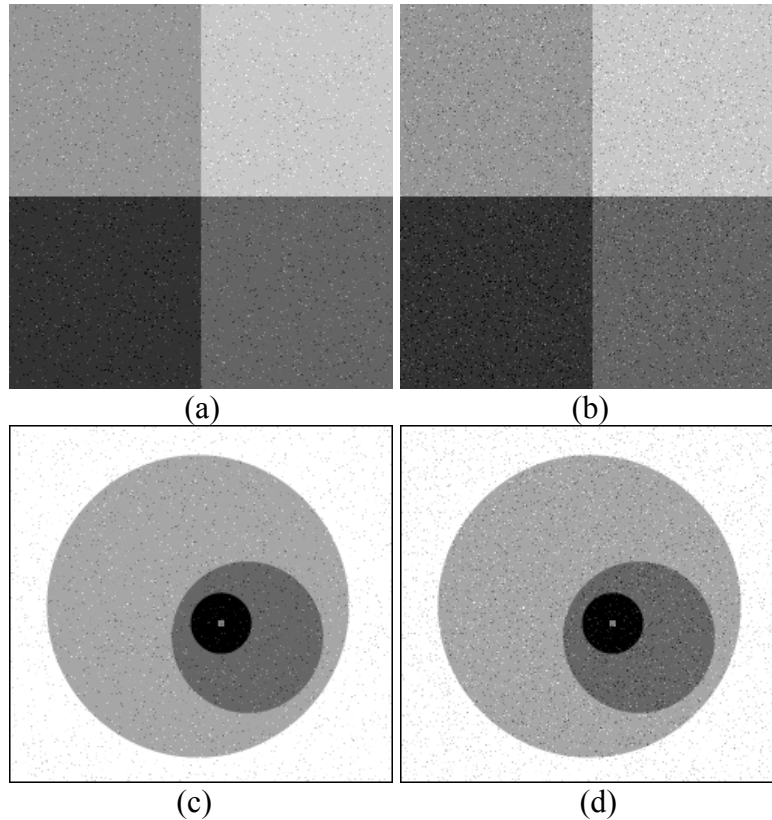


Fig. 7. (a), (b) Uniformly distributed noise in the range $[-50, 50]$ corrupting 10% and 20% of randomly selected pixels in Fig. 6a. (c), (d) Similar noise corrupting Fig. 6b.

When smoothing in homogeneous areas in an image, windows used in adaptive smoothing become large and square. Near an edge, although gradient magnitude in the direction normal to the edge may be high, gradient magnitude in the direction parallel to the edge will be low, producing windows that are wide in the direction along the edge but narrow in the direction normal to the edge.

After finding the window dimensions from minimum and maximum image gradients at a pixel using formulas (1) and (2), the median of pixels in the window is

found and saved at the location of the window center in the smoothed image. This is repeated for each pixel in an image and the root-mean-squared (RMS) difference between the original noise-free image and the smoothed noisy version of the image is determined. This is repeated for various values of parameter a and the RMS differences are tabulated in Table 1.

Table 1. The RMS difference between noise-free images 6a and 6b and noise-corrupted images 7a–7d after adaptive median filtering using rectangular windows of dimensions computed from (1) and (2). The bold numbers indicate optimal a for images 7c and 7d.

Noisy Image	$a = 3$	$a = 5$	$a = 7$	$a = 9$	$a = 11$	$a = 15$	$a = 21$
Fig. 6a	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Fig. 7a	2.56	1.94	1.93	1.91	1.74	1.36	0.24
Fig. 7b	3.65	2.92	2.83	2.67	2.51	1.72	0.76
Fig. 6b	0.0	0.0	0.0	0.03	0.07	0.08	2.12
Fig. 7c	2.59	2.47	2.11	1.62	1.34	1.35	2.49
Fig. 7d	3.79	3.57	2.84	2.20	1.88	2.09	2.84

An ideal adaptive smoother should not change image structure if noise is not present in the image. The proposed adaptive smoother provides this property relatively well as evidenced by the results in Table 1 for Figs. 6a and 6b. When parameter a becomes very large, it starts to smooth very small image structures and errors start to emerge in Fig. 6b. However, for parameter a up to 15 pixels, the errors produced by adaptive smoothing are either immeasurable or very small.

Errors are larger for Fig. 7c than for Fig. 7d because Fig. 6b contains curved boundaries, and very large rectangular windows cannot align well with curved boundaries. We can also see that error increases with an increase in noise level. In addition, it is observed that there is an optimal a , which depends primarily on image structure and secondarily on noise level. The bold numbers in Table 1 identify the optimal parameter a for image 6b under different levels of noise. For image 6a, the

optimal a appears to be larger than 21 pixels because smaller errors are obtained as a is increased up to 21 pixels.

The smoothing results for $a = 21$ pixels are shown in Fig. 8. Figs. 8a–c are visually indistinguishable from Fig. 6a. The difference between Figs. 8d–f and Fig. 6b is in the small square area at the center of the smallest circle. Because the area is smaller than the width of the smallest smoothing window, the small square area is treated as noise and smoothed. Except for this small square area, other areas in these images are indistinguishable from image 6b.

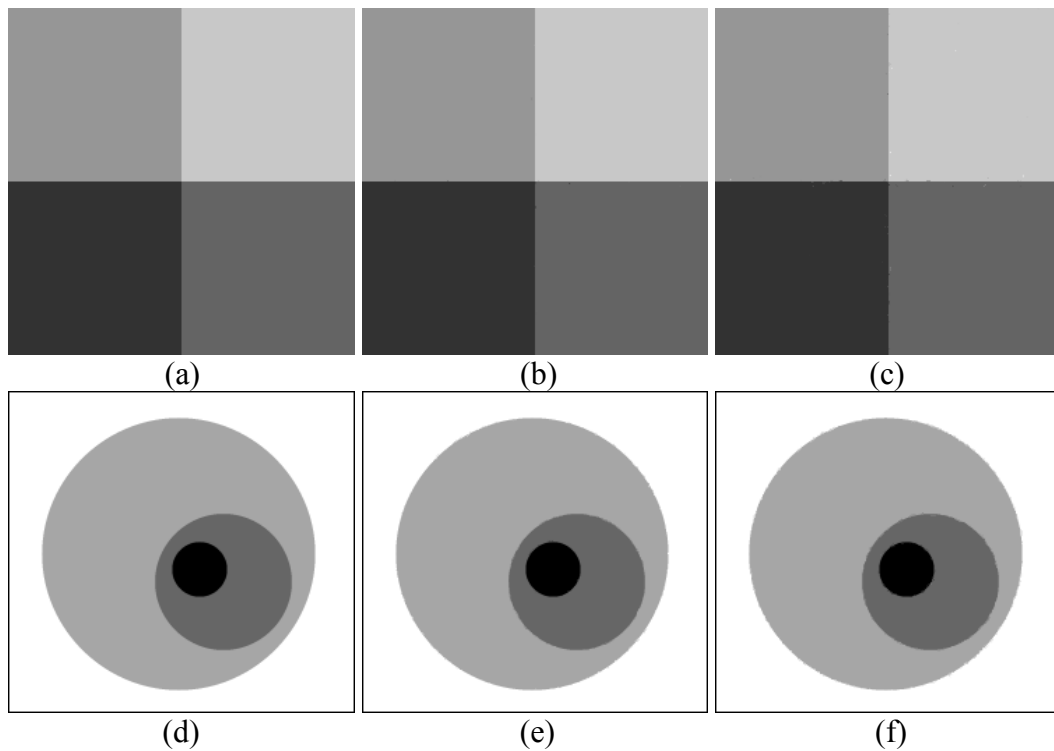


Fig. 8. (a)–(c) Figs. 6a, 7a, and 7b after median filtering using adaptive windows with $a = 21$ pixels. (d)–(f) Figs. 6b, 7c, and 7d after median filtering using adaptive windows with $a = 21$ pixels.

5.2 Mean Filtering

In mean filtering, windows similar to those used in median filtering are used, but instead of finding the median of the values, the mean of the values is computed. In this

experiment, images shown in Figs. 6a and 6b were again used. This time, however, uniformly distributed numbers in the range $[-20, 20]$ were generated and added to all pixel intensities in Fig. 6a to obtain Fig. 9a. Fig. 9b shows the same but using random numbers in the range $[-50, 50]$. This was repeated for Fig. 6b to obtain Figs. 9c and 9d.

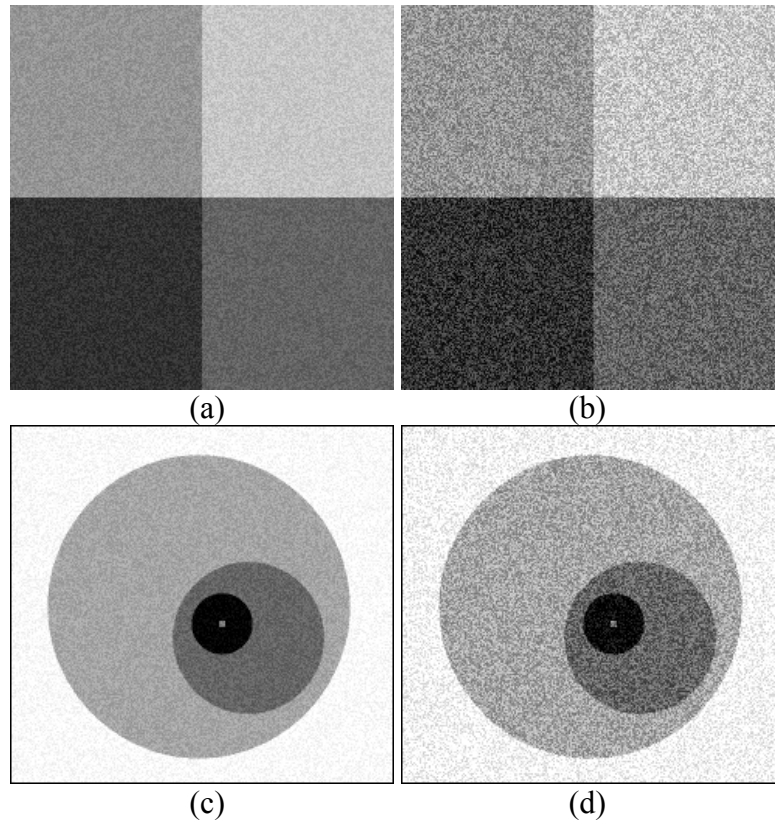


Fig. 9. (a), (b) Images obtained by adding zero-mean uniformly distributed noise of amplitudes 20 and 50 to all pixels in image 6a. (c), (d) Images obtained by adding similar noise to image 6b.

By performing adaptive mean filtering on the noise-corrupted images shown in Fig. 9, the errors shown in Table 2 were obtained. Errors in mean filtering are larger than those in median filtering. Optimal window size found in mean filtering is slightly larger than that in median filtering. This can be attributed to the fact that only 10% or 20% of pixels in median filtering were corrupted by noise, while all pixels in mean filtering were corrupted by noise. The optimal window sizes for mean and median filtering are close,

however. This confirms that optimal parameter a depends primarily on image structure and secondarily on image noise.

Table 2. The RMS difference between ideal images 6a and 6b and noisy images 9a–9d after adaptive mean filtering using rectangular windows of dimensions inversely proportional to the maximum and minimum gradient magnitudes. The bold numbers indicate optimal a in smoothing for different images.

Noisy Image	$a = 3$	$a = 5$	$a = 7$	$a = 9$	$a = 11$	$a = 15$	$a = 21$
Fig. 6a	0.0	0.0	0.0	0.22	0.22	0.89	0.92
Fig. 9a	5.65	5.59	5.59	5.55	5.51	3.83	4.00
Fig. 9b	12.82	9.87	8.95	8.75	6.43	6.38	6.36
Fig. 6b	0.0	0.0	0.0	1.08	2.31	2.95	3.60
Fig. 9c	6.33	6.29	5.43	5.39	5.78	5.58	6.15
Fig. 9d	14.81	12.92	12.21	11.21	11.18	10.88	11.09

Adaptive mean filtering also hardly changes an image’s structure when the image is not noisy. Some errors are obtained when windows become exceedingly large, as very large windows cannot lie well along curved boundaries. Mean filtering results for $a = 21$ pixels are shown in Fig. 10. Due to the very narrow (less than 1 pixel) height of windows in the direction of maximum gradient, pixels at region boundaries are not smoothed.

5.3 Gaussian Filtering

Letting $W = 3a/(g_n+1)$, $H = 3a/(g_m+1)$, $\sigma_x = W/6$, and $\sigma_y = H/6$ according to formulas (11) and (12), filling a window’s entries with Gaussian weights according to (3), orienting the window in the direction of least gradient at a pixel, multiplying corresponding window and image values, and adding them, the smoothed value at the pixel is obtained. Repeating this calculation at each pixel in the test images, the results shown in Table 3 are obtained.

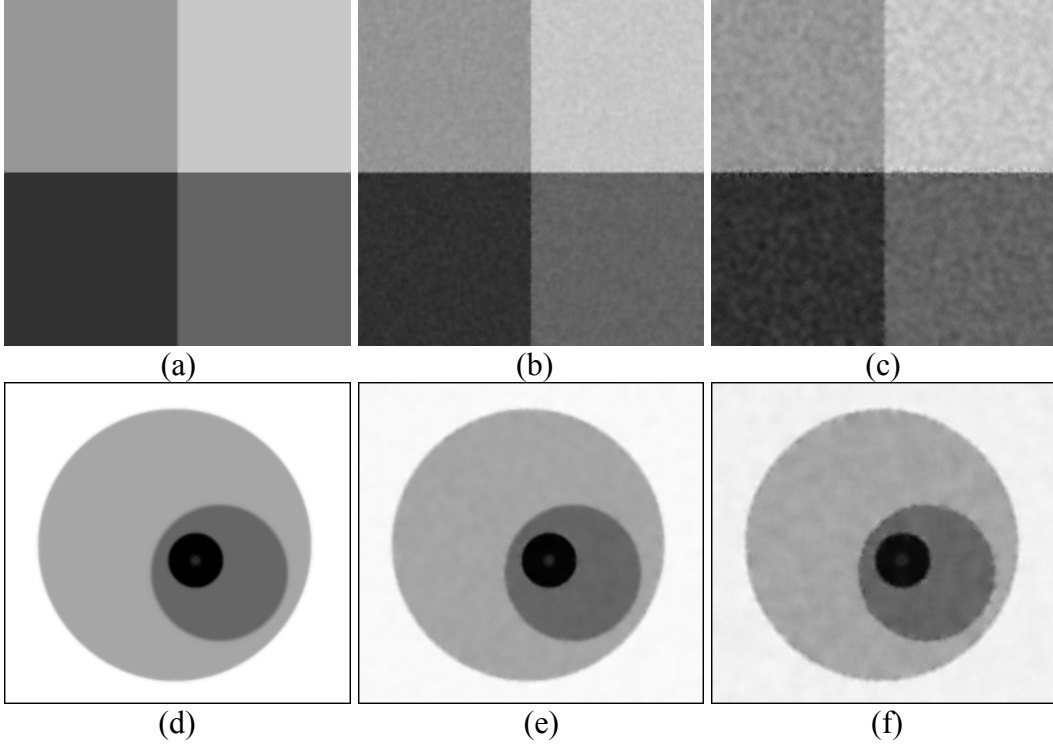


Fig. 10. (a)–(c) Mean filtering using adaptive windows with $a = 21$ pixels on images 6a, 7a, and 7b. (d)–(f) Mean filtering using adaptive windows with $a = 21$ pixels on images 6b, 7c, and 7d.

Table 3. The RMS difference between original images 6a and 6b and adaptive Gaussian smoothed noisy versions of the images using $\sigma_x = a/2(g_n+1)$ and $\sigma_y = a/2(g_m+1)$. Bold numbers indicate the optimal a for different images.

Noisy Image	$a = 3$	$a = 5$	$a = 7$	$a = 9$	$a = 11$	$a = 15$	$a = 21$
Fig. 6a	0.0	0.85	0.98	0.88	0.90	0.83	1.62
Fig. 9a	4.95	4.04	3.67	3.38	3.19	2.90	2.96
Fig. 9b	9.99	6.10	5.20	4.76	4.54	4.46	4.62
Fig. 6b	0.19	0.89	0.42	1.20	1.96	4.09	6.57
Fig. 9c	5.81	5.30	5.01	4.94	5.29	6.45	8.46
Fig. 9d	13.45	11.40	10.81	10.78	10.10	11.68	12.94

Again, the smoothing effect on a noise-free image has been minimal and visually indistinguishable from the original image with small values of a . For larger values of a , the smoothing effect starts to appear in the image containing the circular regions due to the presence of curved boundaries. The table again confirms that optimal a is influenced

more by image structure than by image noise. For Fig. 6a and its noisy versions, the same optimal a is obtained independent of the magnitude of noise, and for Fig. 6b, as noise is increased, the optimal a increases only slightly.

Results of Gaussian smoothing using adaptive windows with $a = 21$ pixels are shown in Fig. 11. The proposed adaptive smoothing has been able to reduce noise considerably without changing image structure significantly.

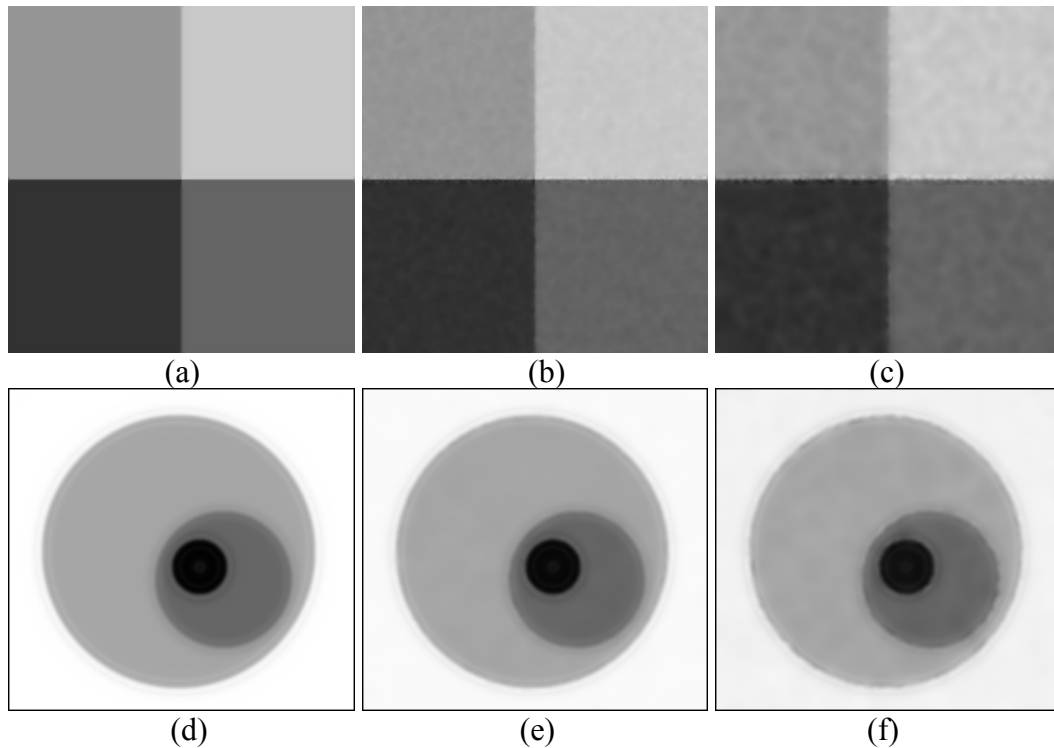


Fig. 11. (a)–(c) Adaptive Gaussian filtering of Figs. 6a, 9a, and 9b with $a = 21$ pixels. (d)–(f) Adaptive Gaussian filtering of Figs. 6b, 9c, and 9d with $a = 21$ pixels.

5.4 Gaussian Smoothing Using Real Images

Figs. 12–14 demonstrate the quality of adaptive Gaussian smoothing when applied to real images. Since the original images 12a and 13a contained very little noise, zero-mean uniformly distributed noise in the range $[-20, 20]$ was generated and added to the images to obtain the images shown. Using adaptive Gaussians with $\sigma_x = a/2(g_n+1)$ and

$\sigma_y = a/2(g_m+1)$, and letting $a = 6$ pixels, the results depicted in Figs. 12b and 13b were obtained. The proposed adaptive smoothing has been able to reduce the noise significantly without blurring the edges.



Fig. 12. (a) Zero-mean uniformly distributed noise corrupting the image of a plane. (b) Adaptive Gaussian filtering with $\sigma_x = a/2(g_n+1)$ and $\sigma_y = a/2(g_m+1)$, where g_n and g_m are the minimum and maximum gradient magnitudes at a pixel and $a = 6$ pixels.

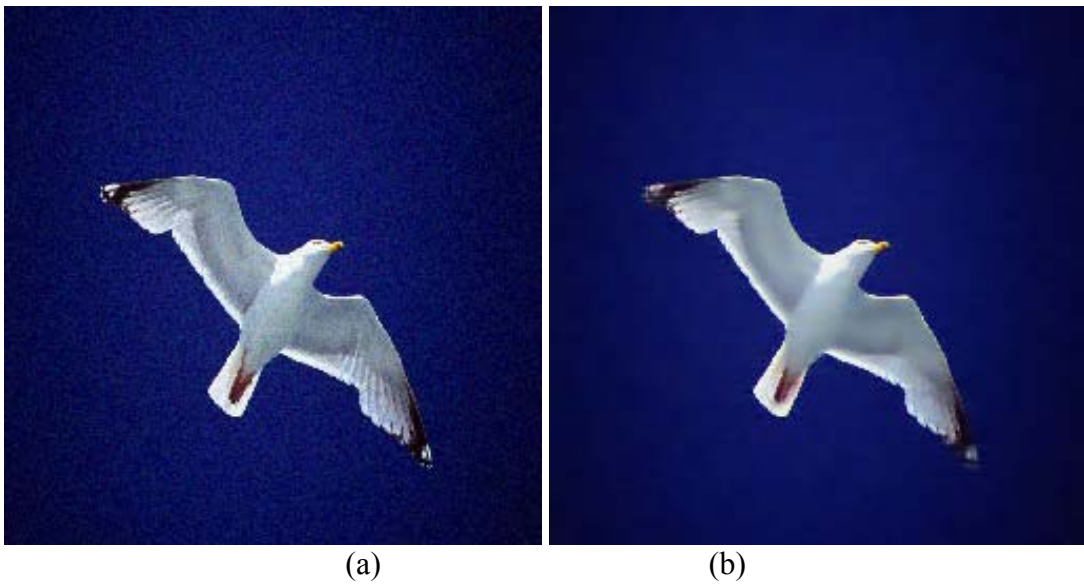


Fig. 13. (a) Zero-mean uniformly distributed noise corrupting a color image of a seagull. (b) Adaptive Gaussian filtering with $\sigma_x = a/2(g_n+1)$ and $\sigma_y = a/2(g_m+1)$, where g_n and g_m are the minimum and maximum gradient magnitudes at a pixel and $a = 6$ pixels.

Examples of images that already contain noise are given in Figs. 14 and 15. Fig. 14a shows a synthetic aperture radar (SAR) image. After adaptive Gaussian filtering with $a = 3$ pixels, the result shown in Fig. 14b was obtained. Noise has been considerably reduced while preserving very narrow road structures. To preserve the small details, a very small parameter a is used. Larger parameter values would smooth the very narrow road structures in the image.

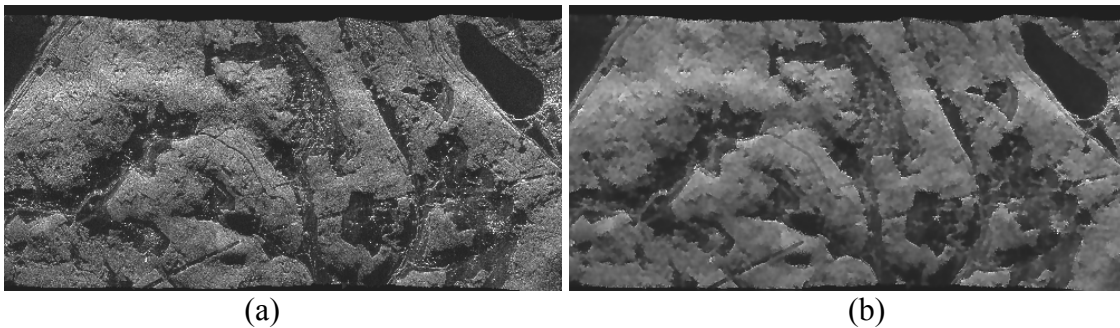


Fig. 14. (a) A SAR image from the Shuttle Radar Topography Mission. (b) Adaptive Gaussian filtering of the image with $a = 3$ pixels.

Fig. 15a shows a cardiac angiogram. Smoothing the image by adaptive Gaussian filtering with $a = 6$ pixels, the image shown in Fig. 15b is obtained. The structure of blood vessels has been preserved while reducing random noise throughout the image.

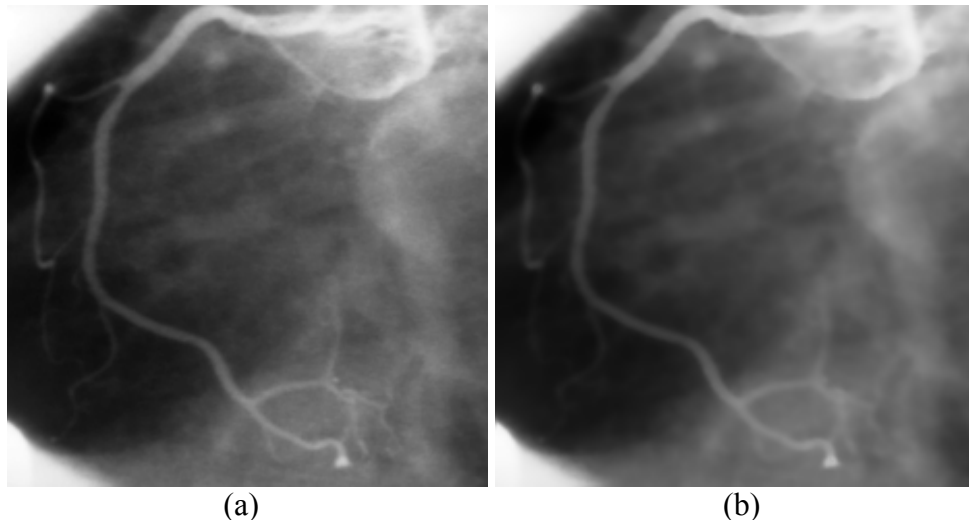


Fig. 15. (a) A cardiac angiogram. (b) Adaptive Gaussian filtering of the angiogram when $a = 6$ pixels.

5.5 Comparison with a State-of-the-Art Method

To determine the performance of the proposed adaptive smoothing with respect to state-of-the-art methods, our method was compared with the optimization method of Kervrann [30, 31]. Extensive results in [31] show that Kervrann’s method produces close to or better results than several state-of-the-art methods reported earlier. These methods include the total variation minimizing of Rudin et al. [49], the bilateral filtering of Tomasi and Manduchi [57], the anisotropic diffusion of Perona and Malik [42], the adaptive weights smoothing of Polzehl and Spokoiny [44], and Wiener filtering [27].

Comparison of the proposed adaptive smoothing with the optimization method of Kervrann on different images and under different noise levels reveals that for very detailed images where the signal-to-noise ratio is high, the method of Kervrann restores image structures better than the proposed method, but when the signal-to-noise ratio is low and the scene is not highly textured, the proposed method restores image structures better than the method of Kervrann.

Examples comparing smoothing results by the proposed and Kervrann methods on the *Lena* image are given in Fig. 16. Zero-mean Gaussian noise of standard deviations 10, 20, and 50 were added to the *Lena* image as shown in Figs. 16b–d. Smoothing results by the method of Kervrann are shown in Figs. 16e–g. Results obtained by the proposed method with $a = 6$ pixels are shown in Figs. 16h–j. Comparing the images visually, we see that in very detailed image areas such as the hair and the cap details, the method of Kervrann restores image structures better, while the proposed method restores larger image structures such as the shoulder, face, and hat boundaries better.

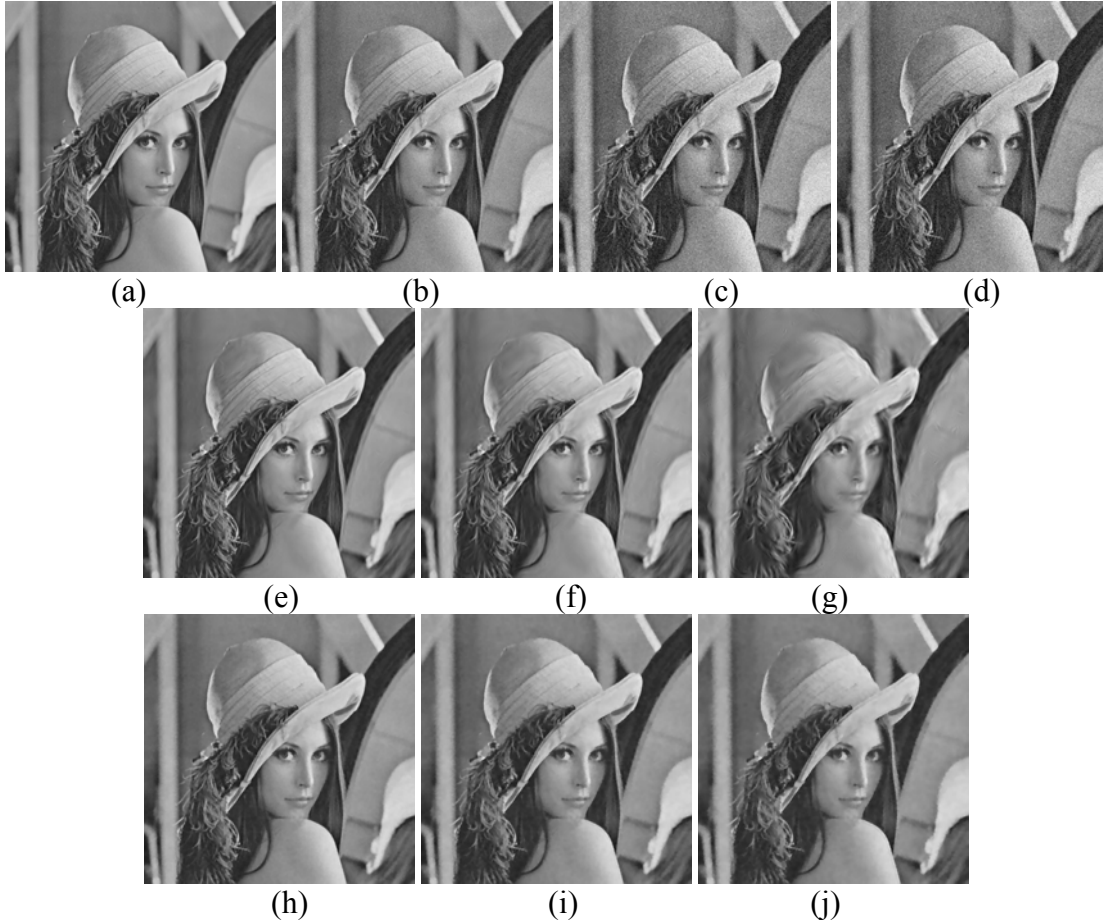


Fig. 16. (a) The Lena test image. (b)–(d) Zero-mean Gaussian noise of standard deviations 10, 20, and 50, respectively, added to the Lena image. (e)–(g) Smoothing of noise-corrupted Lena images by the adaptive smoothing of Kervrann [30,31]. (h)–(j) The smoothing results obtained by the proposed method.

Quantitatively, the RMS difference between the original Lena image and the smoothed noisy images of Lena by the proposed and Kervrann methods are included in Table 4. For small amounts of noise, the method of Kervrann produces a smaller RMS error than the proposed method. However, as the noise level is increased, the proposed method produces a smaller RMS error than the method of Kervrann. Parameter $a = 6$ pixels appeared the best when smoothing the noisy Lena images. A smaller parameter a would retain image noise while a larger a would smooth critical image structure.

Table 4. RMS difference between the original Lena image and the smoothed noisy versions of the image by the Kervrann and proposed methods.

Noise Level σ	Kervrann's Smoothing	Proposed Smoothing
10	4.8	5.2
20	5.9	6.0
50	9.4	7.3

6 Discussion and Conclusions

A camera focusing on an object creates an image that appears sharp in the object area and blurred in areas representing objects in front of or behind the focus plane. This blurring normally increases as the distance of an object to the focus plane increases. In blurred areas in an image, noise is of particular concern because intensities vary slowly and noise can produce false edges. In sharp areas, intensity variation from image structure is high and the effect of noise will not be as high. Therefore, a larger neighborhood is needed to smooth a blurred area compared to a sharp area in an image. The adaptive window mechanism introduced in this paper automatically adjusts the window size to local image details.

The proposed window mechanism smoothes an image more in the direction of least gradient than in other directions. This selective smoothing preserves image structure while reducing random noise. Because details in an image can vary from neighborhood to neighborhood, to reduce noise in an image it is required to use windows that adapt to local image details. The proposed window mechanism adapts the size, shape, and orientation of a rectangular window to local image details. Window dimensions at a pixel are calculated using the minimum and maximum gradient magnitudes at the pixel, and window orientation is selected so that its long side aligns with the direction of minimum

gradient. In this manner, region boundaries are preserved while smoothing region interiors.

In addition to automatically adjusting the window size based on local image information, the method provides a global parameter a that the user can control to increase or decrease the automatically determined smoothing. For instance, if noise level in an image is known to be high, a proportionately higher a can be selected.

In this paper, window size was determined using image gradients. Alternatively, window size can be selected using local scale in an image. To estimate local scale, Jeong and Kim [28] used a number of ad-hoc criteria to formulate an energy function that would satisfy the criteria. Then by iteratively minimizing the energy function, the optimal local scale in an image is found. Lindeberg [36] used normalized images where coordinates of pixels were multiplied by their (unknown) local scales. A local scale was then determined while locally maximizing the gradients. Elder and Zucker [16] considered the scale at a pixel to be the smallest scale where the gradient exceeded a critical value. Fdez-Valdivia et al. [17] suggested measuring spatial information using multiple detectors, each sensitive to a range of frequencies. They assumed that scale depends on the response of only relevant detectors and measured relative sharpness of a sensor's response across scales using a bank of Gabor filters. They defined the natural scale in a local neighborhood as the scale that locally maximized a sharpness measure. In the presence of noise, as with our method, these methods may not produce reliable local scales because noise influences the image intensities that these methods use to determine local scales.

Experimental results on synthetic and real images show that the proposed adaptive smoothing can compete with state-of-the-art methods in the literature. A comparison with the optimization-based method of Kervrann [30,31] shows that although under low levels of noise the proposed method may not do as well as the method of Kervrann, under moderate to high levels of noise, the proposed method preserves image structure better than the method of Kervrann, which was found to produce close to or better results than state-of-the-art methods before it.

The proposed method works best on images containing abundant straight and slowly bending edges. It does not do as well on highly textured images where sharply bending edges are abundant. This can be attributed to the rectangular nature of the windows. As the smoothness parameter a is increased to smooth more noise, windows increase in size, including pixels from adjacent regions when the regions are narrow and highly curved. For images containing narrow and curved regions, parameter a should be set sufficiently low to avoid inclusion of pixels in neighboring regions in the same window.

The adaptive window mechanism introduced in this paper may be used in conjunction with other window-based adaptive smoothing methods. Weighted averaging methods that use square windows [7,22,32,41,57,59] can be revised to use rectangular windows that are oriented in the direction of least gradient without changing anything else. If done so, it is believed that image structure will be preserved better than the methods when using square windows. This is because properly oriented rectangular windows can select pixels on one side of an edge in smoothing compared to square windows that have to use pixels on both sides of an edge even though the pixels are

properly weighted. The process that associates weights to pixels in a square window can be made to associate weights to rectangular windows.

The adaptive smoothing method described in this paper is a few times slower than isotropic smoothing. Implementation of the proposed method is very simple. The computations involve only the incorporation of the mapping defined by formulas (6) and (7) into the isotropic smoothing process.

Acknowledgement

This work was supported by the Air Force Research Laboratories (Award No. FA8650-05-1-1914). Images in Figs. 12 and 13 are from Kodak, image of Fig. 14 is from the U.S. Geological Survey, image of Fig. 15 is from Rush-Presbyterian-St. Luke's Medical Center, and the image of Lena in Fig. 16a as well as images in Figs. 16e–g are courtesy of Charles Kervrann, INRIA. This paper is dedicated to Prof. Richard Dubes and Prof. Carl Page, Computer Science Department, Michigan State University, the late professors of the first author, for their inspiring guidance during the author's graduate studies.

References

1. L. Alvarez, P-L Lions, and J-M Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numerical Analysis*, **29**(3), 1992, 845–866.
2. R. Ashino, S. J. Desjardins, A. A. Kolyshkin, and R. Vaillancourt, Noise smoothing in the Fourier domain by a multi-directional diffusion, *Technical Report CRM-2934*, Division of Mathematical Sciences, Osaka Kyoiku University, Sept. 2003.
3. G. Aubert and L. Vese, A variational method in image recovery, *SIAM J. Numer. Anal.*, **34**(5), 1997, 1948–1979.
4. D. Barash, A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(6), 2002, 844–847.
5. D. Barash and D. Comaniciu, A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering, and mean shift, *Image and Vision Computing*, **22**(1), 2004, 73–81.

6. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, Robust anisotropic diffusion, *IEEE Trans. Image Processing*, **7**(3), 1998, 421–432.
7. T. Boulton, R. A. Melter, F. Skorina, and I. Stojmenovic, G-neighbors, *Proc. SPIE Conf. Vision Geometry II*, 1993, 96–109.
8. F. Catté, P. L. Lions, J. M. Morel, and T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Numerical Analysis*, **29**(1), 1992, 182–193.
9. T. F. Chan, S. Osher, and J. Shen, The digital TV filter and nonlinear denoising, *IEEE Trans. Image Processing*, **10**(2), 2001, 231–241.
10. Y. Cheng, Mean shift, mode seeking, and clustering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**(8), 1995, 790–799.
11. K. Chen, Adaptive smoothing via contextual and local discontinuities, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **27**(10), 2005, 1552–1567.
12. R. T. Chin and C. L. Yeh, Quantitative evaluation of some edge-preserving noise-smoothing techniques, *Computer Vision, Graphics, and Image Processing*, **23**, 1983, 67–91.
13. D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(5), 2002, 603–619.
14. F. Durand and J. Dorsey, Fast bilateral filtering for the display of high-dynamic-range images, *ACM Trans. Graphics*, **21**(3), 2002, 257–266.
15. M. Elad, On the origin of the bilateral filter and ways to improve it, *IEEE Trans. Image Processing*, **11**(10), 2002, 1141–1151.
16. J. H. Elder and S. W. Zucker, Local scale control for edge detection and blur estimation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(7), 1998, 699–716.
17. J. Fdez-Valdivia, J. A. Garcia, J. Martinez-Baena, and X. R. Fdez-Vidal, The selection of natural scales in 2-D images using adaptive Gabor filters, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(5), 1998, 458–469.
18. V. S. Frost, J. A. Stiles, K. S. Shanmugam, J. C. Holtzman, and S. A. Smith, An adaptive filter for smoothing noisy radar images, *Proceedings of the IEEE*, **69**(1), 1981, 133–135.
19. K. Fukunaga and L. D. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Trans. Information Theory*, **21**, 1975, 32–40.

20. T. Gasser, L. Sroka, and C. J. Steinmetz, Residual variance and residual pattern in nonlinear regression, *Biometrika*, **73**, 1986, 625–633.
21. G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz, Nonlinear anisotropic filtering of MRI data, *IEEE Trans. Medical Imaging*, **11**(2), 1992, 221–232.
22. I. Gijbels, A. Lambert, and P. Qiu, Edge-preserving image denoising and estimation of discontinuous surfaces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **28**(7), 2006, 1075–1087.
23. R. E. Graham, Snow-removal: A noise-stripping process for picture signals, *TRE Trans.*, **8**, 1961, 129–144.
24. M. I. Gürelli and L. Onural, A class of adaptive directional image smoothing filters, *Pattern Recognition*, **29**(12), 1996, 1995–2004.
25. R. M. Haralick and L. Watson, A facet model for image data, *Computer Graphics and Image Processing*, **15**, 1981, 113–129.
26. Q. Hu, X. He, and J. Zhou, Multi-scale edge detection with bilateral filtering in spiral architecture, *Conference in Research and Practice in Information Technology Series*, **100**, 2004, 29–32.
27. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, 1989.
28. H. Jeong and C. I. Kim, Adaptive determination of filter scales for edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **14**(5), 1992, 579–585.
29. D. J. Kang and K. S. Roh, A discontinuity adaptive Markov model for color image smoothing, *Image and Vision Computing*, **19**, 2000, 369–379.
30. C. Kervrann, An adaptive window approach for image smoothing and structure preserving, *Proc. European Conf. Computer Vision (ECCV)*, Prague, Czech Republic, March 2004.
31. C. Kervrann and J. Boulanger, Local adaptivity to variable smoothness for exemplar-based image denoising and representation, *INRIA Research Report No. 5624*, July 2005.
32. J. S. Lee, Digital image enhancement and noise filtering by use of local statistics, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **2**(2), 1980, 165–168.
33. X. Li and T. Chen, Nonlinear diffusion with multiple edginess thresholds, *Pattern Recognition*, **27**(8), 1994, 1029–1037.
34. J. H. Lin, T. M. Sellke, and E. J. Coyle, Adaptive stack filtering under the mean absolute error criterion, *IEEE Trans. ASSP*, **38**(6), 1990, 938–954.

35. H. M. Lin and A. N. Wilson, Jr., Medial filters with adaptive length, *IEEE Trans. Circuits and Systems*, **35**, 1988, 675–690.
36. T. Lindeberg, On scale selection for differential operators, *Proc. 8th Scandinavian Conf. Image Analysis*, May 1993, 857–866.
37. P. Meer, R. H. Park, and K. Cho, Multiresolution adaptive image smoothing, *CVGIP: Graphical Models and Image Processing*, **56**(2), 1994, 140–148.
38. D. Mumford and J. Shah, Optimal approximations by piecewise smooth functions and variational problems, *Comm. Pure and Appl. Math.*, **42**(5), 1989, 577–685.
39. M. Muneyasu, Y. Wada, and T. Hinamoto, Realization of adaptive edge preserving smoothing filters, *Electronics and Communications in Japan, Part 3*, **80**(10), 1997, 19–27.
40. M. Nagao and T. Matsuyama, Edge preserving smoothing, *Computer Graphics and Image Processing*, **9**, 1979, 394–407.
41. K. J. Overton and T. E. Weymouth, A noise reducing preprocessing algorithm, *Proc. IEEE Computer Science Conf. Pattern Recognition and Image Processing*, 1979, 498–507.
42. P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **12**(7), 1990, 629–639.
43. T. Q. Pham and L. J. van Vliet, Separable bilateral filtering for fast video preprocessing, *Proc. Int'l Conf. Multimedia and Expo (ICME)*, Amsterdam, Netherlands, June 2005.
44. J. Polzehl and V. G. Spokoiny, Adaptive weights smoothing with applications to image restoration, *Journal of Royal Statistical Society, Series B*, **62**, 2000, 335–354.
45. J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Trans. Image Processing*, **12**(11), 2003, 1338–1351.
46. P. Qiu, The local piecewisely linear kernel smoothing procedure for fitting jump regression surfaces, *Technometrics*, **46**, 2004, 87–98.
47. A. Restrepo and A. C. Bovik, Adaptive trimmed mean filters for image restoration, *IEEE Trans. ASSP*, **36**(8), 1988, 1326–1337.
48. S. Roth and M. J. Black, Field of experts: A framework for learning image priors, *Proc. Computer Vision and Pattern Recognition*, **2**, 2005, 860–867.
49. L. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, **60**, 1992, 259–268.

50. P. Saint-Marc, J. S. Chen, and G. Medioni, Adaptive smoothing: A general tool for early vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **13**(6), 1991, 514–529.
51. P. Salembier, Adaptive rank order based filters, *Signal Processing*, **27**, 1992, 1–25.
52. M. A. Schulze and Q. X. Wu, Nonlinear edge-preserving smoothing of synthetic aperture radar images, *Proc. New Zealand Image and Vision Computing*, Aug. 1995, 65–70.
53. S. S. Sinha and B. G. Schunck, Discontinuity preserving surface reconstruction, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, 1989, 229–234.
54. V. Spokoiny, Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice, *Annals of Statistics*, **26**, 1988, 1356–1378.
55. R. L. Stevenson and E. J. Delp, Viewpoint invariant recovery of visual surfaces from sparse data, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **14**, 1992, 897–909.
56. D. Terzopoulos, The computation of visible-surface reconstruction, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **10**(4), 1988, 417–438.
57. C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, *Proc. Int'l Conf. Computer Vision*, Bombay, India, 1998, 839–846.
58. D. Tschumperlé and R. Deriche, Regularization of orthonormal vector sets using coupled PDE's, *Proc. IEEE Workshop Variational and Level Set Methods in Computer Vision*, 2001, 3–10.
59. C. C. Wang, A. H. Vagnucci, and C. C. Li, A gradient inverse weighted smoothing scheme and the evaluation of its performance, *Computer Vision, Graphics, and Image Processing*, **15**, 1981, 167–181.
60. J. Weickert, *Anisotropic Diffusion in Image Processing*, ECMI Series, Teubner-Verlag, Germany, 1998.
61. R. T. Whitaker and S. M. Pizer, A multi-scale approach to non-uniform diffusion, *CVGIP: Image Understanding*, **57**(1), 1993, 99–110.
62. C. Xu, H. Wang, J. Wang, and L. Ge, Adaptive filter in SAR interferometry derived DEM, *APSG Workshop on Geodynamics and Natural Hazards*, Hong Kong, June 2005.
63. J. H. Yi and D. M. Chelberg, Discontinuity-preserving and viewpoint invariant reconstruction of visible surfaces using a first-order regularization, **17**, 1995, 624–629.

64. L. Yin, J. T. Astola, and Y. A. Neuvo, Adaptive stack filtering with application to image processing, *IEEE Trans. Signal Processing*, **41**(1), 1993, 162–184.
65. Y-L You, W. Xu, A. Tennenbaum, and M. Kaveh, Behavioral analysis of anisotropic diffusion in image processing, *IEEE Tran. Image Processing*, **5**(11), 1996, 1539–1553.