

highly-correlated predictive models, *predictive decorrelators do not guarantee compression*.

- 2) $H_r^{\text{inf}^4}(n, \rho)$ curves for different values of n and $\rho > 0.5$ converge to the straight line $H_r^{\text{inf}^4}(n, \rho) = 2(1 - \rho)$ (Fig. 4). It is interesting to observe that as n increases, the highest (best-case) value of $H_r^{\text{inf}^4}(n, 0)$ also decreases toward 1. Fig. 4 suggests for large n

$$H_r^{\text{inf}^4}(n, \rho) \approx \begin{cases} 1 & 0 \leq \rho \leq 0.5, \\ 2(1 - \rho) & 0.5 \leq \rho \leq 1 \end{cases}$$

with only a 5% numerical error for $n > 128$ in the $0 \leq \rho \leq 0.5$ interval, and an exact numerical match for $\rho > 0.5$. From here, as $\rho > 0.5$

$$C^{\text{sup}}(n, \rho) \geq C^{\text{sup}^4}(n, \rho) = \frac{1}{H_r^{\text{inf}^4}(n, \rho)} = \frac{1}{2(1 - \rho)} \geq 1. \quad (9)$$

This implies that, first, for large n and $\rho > 0.5$, predictive models are guaranteed to not increase the entropy for any inter-image correlation. Second, an elegant numerical result follows for the best-case predictive compression ratio. Since $C^{\text{sup}}(n, \rho_0) = C_0$ gives the lowest $\rho = \rho_0$ such that predictive models \hat{u} , ρ_0 —correlated to some images u in (1), can deliver C_0 effective compression (see Proposition 1), solving (9) for ρ yields the following.

Proposition 3: Effective compression ratio $C = C_0$ (2) cannot be achieved with (1) when model-to-image correlation

$$\rho(u, \hat{u}) < 1 - \frac{1}{2C_0}.$$

This holds true for *any* number of intensity levels n . In particular, any predictive compression is impossible for $\rho(u, \hat{u}) < 0.5$ ($C_0 = 1$); 2:1 compression is impossible for $\rho(u, \hat{u}) < 0.75$ ($C_0 = 2$), and 3:1 compression is impossible for $\rho(u, \hat{u}) < 5/6$ ($C_0 = 3$) (compare to Proposition 1 and Table I).

REFERENCES

- [1] N. Memon and X. Wu, "Recent developments in context-based predictive techniques for lossless image compression," *Comput. J.*, vol. 40, no. 2/3, 1997.
- [2] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," Computer Systems Laboratory, Hewlett-Packard, Palo Alto, CA, HPL-98-193, Nov. 1998.
- [3] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, Apr. 1997.
- [4] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*. New York: SPIE Opt. Eng. Press, 1991.
- [5] O. S. Pianykh, "Lossless predictive compression of correlated data," Ph.D. dissertation, Louisiana State Univ., Alexandria, May 1998.

Peer Group Image Enhancement

C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer

Abstract—Peer group image processing identifies a "peer group" for each pixel and then replaces the pixel intensity with the average over the peer group. Two parameters provide direct control over which image features are selectively enhanced: area (number of pixels in the feature) and window diameter (window size needed to enclose the feature). A discussion is given of how these parameters determine which features in the image are smoothed or preserved. We show that the Fisher discriminant can be used to automatically adjust the PGA parameters at each point in the image. This local parameter selection allows smoothing over uniform regions while preserving features like corners and edges. This adaptive procedure extends to multilevel and color forms of PGA. Comparisons are made with a variety of standard filtering techniques and an analysis is given of computational complexity and convergence issues.

Index Terms—Image enhancement, image smoothing, noise removal, nonlinear filtering.

I. INTRODUCTION

Noise removal and image smoothing are useful pre-processing steps in many image processing applications. A general objective in these applications is to suppress noise while preserving the edge information. Typically, additive Gaussian or impulse noise models are assumed.

Median filtering is a popular choice for impulse noise removal (see the tutorial paper [24] by Yin *et al.* which includes an extensive bibliography for this area). Median filtering forms an approximation u of an image g by passing a window of size $d \times d$ over g and taking the median intensity of the window at pixel location i as the value for u_i . The motivation for this type of filtering is that the median preserves edges (intensity discontinuities). Other related work can be found in [1], [10], [11]. Extensions to color and multidimensional signals include the vector median filter (VMF) [2], vector directional filtering (VDF) [22], and the directional distance filtering (DDF) [8]. The DDF is a combination of VMF and VDF. A common drawback of all these above methods is that they are typically implemented uniformly across the image and tend to modify pixels that are not corrupted by noise. In [3] a Teager-like operator is used to first detect the outliers so that only the noisy pixels are replaced. The detection is performed in each individual color component which may lead to errors in the overall color. For the case of mixed Gaussian and impulse noise, an adaptive nonlinear multivariate filtering method is proposed in [21]. It uses the mean value within a local neighborhood of pixels to estimate the original pixel value and hence may blur the edges and the details.

Other approaches to smoothing while preserving boundaries include variational methods and shock filtering. A standard variational approach [12, p. 24], [13], [14], [7] to segmenting and approximating an image g consists of finding an approximation u and a boundary set K that minimizes an objective functional that has two components, one corresponding to the error between the approximation and the original, and the other related to the length of the boundary. Functionals of this type are often referred to as the Mumford–Shah functionals.

Manuscript received March 31, 1999; revised July 6, 2000. This work was supported by the Office of Naval Research under ONR Grant N00014-96-1-0456. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Scott T. Acton.

C. Kenney, Y. Deng, and B. S. Manjunath are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: manj@ece.ucsb.edu).

G. Hewer is with the Naval Air Warfare Center, China Lake, CA 93555 USA. Publisher Item Identifier S 1057-7149(01)00100-2.

The weights associated with the two components play a significant role in the resulting enhancement quality, and an often encountered problem is the selection of these weights. It is not clear how to select good values for the weights associated with the smoothness and approximation terms.

In shock filtering [17]–[19], intensity values from the interior of regions move outward toward the region edges along gradient lines. The convexity of the intensity along the gradient direction determines the motion direction along the gradient and this direction assignment means that when two regions meet at an edge, the image intensity will experience a jump. Thus the edges of the image correspond to stationary shock fronts for this type of image processing. Note that in shock filtering the maximum values of the image intensity and the minimum values move outward from the interior of their regions to meet at the boundaries. This means that the contrast at the edges is maximized. This also means that shock filtering preserves the total variation of the original image. Shock filtering smoothes in the sense that each region assumes a constant value. However, shock filtering does not remove isolated noise such as salt-and-pepper noise, as discussed by Osher and Rudin in [17].

We propose a nonlinear algorithm for image smoothing and impulse noise removal that addresses the above mentioned drawbacks of the current methods. The proposed method is based on the idea that each pixel has a peer group of associated nearby pixels. The peer group is then used to modify the value of the pixel.

There are many ways to select the peer group for a given pixel. For example, see the earlier work by Yaroslavsky [23] presenting an abstract formulation of the group idea. In general, peer group members should share some common values. For a single image, the peer group may be nearby pixels with similar intensity values. For a sequence of images used in determining optical flow fields, the peer group can be nearby pixels (in time and space) with similar intensity values and similar velocity values. In another context, texture values may be assigned to each pixel and the peer group determined by nearness in texture space. In this paper we will use the following peer group definition.

Definition: For an image g , the peer group $P(n, d)$ associated with a pixel i consists of the n pixels in a $d \times d$ window centered at i that are nearest in intensity to $g(i)$. Peer group averaging (PGA) is the process of replacing $g(i)$ with the average over its peer group $P(n, d)$.

In the next section we discuss the stability and convergence of the PGA method. The interplay between the window size (the parameter d in the above definition), peer group size, and the characteristics of the image objects is discussed in Section III. This is followed by a procedure for automatically selecting the peer group size using the Fisher discriminant in Section IV. In Section V we extend the PGA to color images. Section VI presents a multilevel approach to PGA. We conclude in Section VII with discussions.

II. CONVERGENCE OF PGA

PGA is stable in the sense that the new pixel value at i must lie between the maximum and minimum intensities in the window of size $d \times d$ centered at i . In practice we find that PGA converges quickly and that after two or three iterations little additional change occurs. Although the nonlinear aspects of PGA make a general convergence analysis difficult, we establish convergence for a modified form of PGA in which the peer group membership is fixed after the first few iterations.

A. One-Dimensional Monotone Signals

We begin by considering the behavior of the PGA scheme on monotone signals. The analysis shows that PGA for peer groups of size two and windows of size three converges to a piecewise function with the breakpoints at the zeros of the second derivative of the original signal.

This is the 1-D version of the idea that the zeros of the Laplacian serve as boundary points for images. While the peer group size two is too small to be of practical value, the primary motivation for the following analysis is that it establishes a connection between PGA and shock filtering (see Section III-A).

Assume that g is a monotonically increasing 1-D signal. The peer group for pixel i is determined by nearness of intensity values to g_i ; in cases of ambiguity caused by equal intensity differences, we select the peer group to minimize the distance of the peer group members from pixel i , with preference to the right if necessary.

Lemma 1: Let g be monotonically increasing over $[1, n]$ with a change in convexity at k : $g_{i-1} - 2g_i + g_{i+1} > 0$ for $i \leq k$, and $g_{i-1} - 2g_i + g_{i+1} < 0$ for $i > k$. Then the PGA algorithm with peer groups of size 2 and windows of size 3 applied to g converges to a bimodal piecewise constant function u . The point where g changes convexity ($x = k$) is also the boundary point for the two constant regions of u : $u_i = (g_1 + g_2)/2$ for $i \leq k$, and $u_i = (g_{n-1} + g_n)/2$ for $i > k$.

The proof is outlined in Appendix A. For general signals, if g is convex or concave in the interval I defined by $k_0 \leq i \leq k_1$ then the PGA algorithm with peer group size 2 and window size 3 converges to a constant value c in the interval I . Thus the PGA algorithm converges to a piecewise constant function with the regions of constancy determined by the convexity breakpoints (i.e. those points for which $g_{i-1} - 2g_i + g_{i+1}$ changes sign) of the original function g . This is illustrated in Example 2 which compares PGA and median filtering for a Gaussian hump.

B. Convergence for Fixed PGA

In the course of testing numerical examples under the PGA algorithm we observed that the peer groups can change significantly during the first few iterations. This is then followed by a period during which there is little or no change in the peer group structure. The nonlinearity of the PGA algorithm is tied solely to the selection of the peer group members. For ease of analysis we consider a modified version of the PGA algorithm in which membership in the peer groups is fixed after a certain number of regular PGA steps. The following gives a convergence result for this “fixed” PGA approach and applies to both 1-D and 2-D signals (images).

Lemma 2: Let the sequence u_k be generated using the PGA algorithm, starting from $u_0 = g$, with fixed peer groups for each pixel after iteration k_0 . Then the sequence u_k converges to a limiting image u .

Proof: After the peer groups are fixed the PGA iteration can be written as $x_{k+1} = Ax_k$ where the vector x_k is formed by stacking the columns of the image u_k and the matrix A does the peer group averaging. (Thus PGA is linear after the peer groups are fixed. This permits the following analysis.) The entries of the rows of A sum to 1 and consist of either 0 or $1/n$, where n is the peer group number. We say that a subset S of pixels is *strongly connected* or *irreducible* if 1) $i \in S$ implies that the peer group for i is also in S , and 2) for any i and j in S there is a path in S from i to j and a path FROM j to i . That is, there are pixels i_1, \dots, i_m in S such that i_p is in the peer group of i_{p+1} for $p = 1$ to $p = m - 1$ with $i_1 = i$ and $i_m = j$, and vice versa. The image may contain one or more irreducible subsets. For reasons that will become clear below we refer to these subsets as the *primary* irreducible subsets of the image.

For any irreducible subset S the iteration $x_{k+1} = Ax_k$ becomes $\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k$ where \tilde{x} are the pixel intensities in S and \tilde{A} is an irreducible nonnegative matrix with row sums equal to 1. Applying the Perron–Frobenius Theorem [16], we see that \tilde{A} has a simple eigenvalue of largest modulus and all other eigenvalues are of smaller modulus. (Note that we have used the fact that the main diagonal entries of \tilde{A} are positive to establish that the dominant eigenspace has dimension

one; see [16, Ths. 6.1.4 and 6.1.5 p. 219].) Because the row sums are 1 we have $\lambda_{\max}(\hat{A}) = 1$ with associated eigenvector $v = (1, \dots, 1)^T$. Thus $\hat{x}_{m+k_0} = \hat{A}^m \hat{x}_{k_0}$, which is just the power method [20]. Since the dominant eigenspace has dimension one, the power method converges and \hat{x}_k converges to $\hat{x} = cv$ where c is the projection of x_{k_0} onto v ; i.e., c is the average of the entries of x_{k_0} .

Now repeat this for each irreducible subset of pixels. From this analysis we see that the limiting image u under the PGA algorithm is constant on each primary irreducible region of the image.

The remaining pixel intensities satisfy a recursion of the form $\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{b}_k$ where \hat{b}_k represents the contribution to the averaging of the irreducible sets of pixels. Again determine the irreducible subsets of pixels with respect to \hat{A} . These subsets are the *secondary* irreducible subsets of the image. The PGA iteration on these subsets has the form $\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{b}_k$ where \bar{A} is irreducible with positive main diagonal entries equal to $1/n$ where n is the peer group number. Since at least one row of \bar{A} sums to less than 1 (this is true of any row with a nonzero entry in \bar{b}_k) the spectral radius of \bar{A} is less than 1 by the Perron–Frobenius Theorem (see [16, Theorems 6.1.4 and 6.1.5 p. 219]). Let $\bar{x} = (I - \bar{A})^{-1}\bar{b}$. Then $\bar{x}_{k+1} - \bar{x} = \bar{A}(\bar{x}_k - \bar{x})$. Since $\rho(\bar{A}) < 1$ the difference $\bar{x}_k - \bar{x}$ must converge to zero, i.e., \bar{x}_k converges to \bar{x} .

Thus the limiting image u on the secondary irreducible subsets of the image is determined by the constant values on the primary irreducible subsets. Repeating this process for tertiary and higher irreducible sets accounts for all the pixels and completes the proof.

III. PROPERTIES OF PGA AND PARAMETER SELECTION

For PGA there are two parameters: window size $d \times d$ and peer group number n . We can make some general points about selecting the peer group size by considering a particular example. First suppose that in a window of size d , the central pixel is part of group of N pixels of the same intensity. Let us refer to this group of pixels as O (for object) and define $n(O) = N$ as the number of pixels in O . For example O might be a line or corner or even a disconnected group of pixels in the window. If the rest of the pixels in the window have intensities that differ from the common intensity in O then the result of peer group averaging depends critically on whether the peer group number n is greater than $n(O)$ or not. If $n \leq n(O)$ then the peer group for the central pixel will consist of pixels in O and the average over the peer group is equal to the common intensity value over O . However if $n > n(O)$ then the peer group must include some pixels outside of O . Potentially the average might still equal the common value over O if high and low values cancel but in the generic case the average over the peer group will differ from the common value over O . Thus a necessary condition for preserving the intensity of the central pixel is $n \leq n(O)$.

To illustrate, in a 3×3 window, if O is a straight line of width one passing through the central pixel then $n(O) = 3$ and taking the peer group number $n \leq 3$ preserves the intensity value for the central pixel under one step of PGA. On the other hand taking $n > 3$ introduces some blurring to the central pixel. For this example, taking $n \leq 3$ only guarantees invariance for the central pixel for one PGA step since the other pixels on the object may suffer from a window occlusion effect and thus change their value. For example if the line terminates at some point then in a 3×3 window with the end of the line at the window's center, the line has only 2 pixels in the window. In this case the $n(O)$ has shifted to 2 and we would need $n \leq 2$ to preserve the central pixel intensity.

If we consider the effect of peer group size relative to the window diameter we see that for the smallest window size $n = 1$ the peer group is just the central pixel and thus the average over the peer group produces no change. The largest peer group size is $n = d^2$ in which case

the entire window is the peer group. Averaging over the entire window blurs the image. Informally we can say then that as the peer group size increases so does the smoothing effect under PGA. This leads us to the rule of thumb: to obtain maximal smoothing while preserving an object of size $n(O)$ use a peer group of size $n = n(O)$. For more details, we refer to [5].

We now consider some examples illustrating the performance of PGA on 1-D and 2-D signals.

Example 1: This is a signal consisting of two steps of different heights and widths; see Fig. 1(a). Fig. 1(b) shows the original signal plus $N(0, 3)$ noise. The result of using PGA on the noisy data with $n = 9$ and $d = 17$ is shown in Fig. 1(c). Since the peer group number is less than the object number (the intervals are of length 10 and 20 respectively) the steps are preserved. However, when we set the peer group number to $n = 11$ which is larger than the shorter of the step intervals then as expected we lose the small step, as seen in Fig. 1(d). Here we used $d = 2n - 1 = 21$ for the window diameter.

Fig. 1(e) and (f) shows the results of applying adaptive PGA (described in Section IV) for windows of sizes $d = 11$ and $d = 21$ respectively with the best peer group number selected at each pixel from the range $n_{lower} = (d + 1)/2$ to $n_{upper} = d - 1$. These choices on the lower range have the effect of eliminating objects of size less than n_{lower} and hence when $n_{lower} = 6$ as in Fig. 1(e) the steps are preserved. However when $n_{lower} = 11$ as in Fig. 1(f) then the steps of size 10 are lost.

We also note that a similar example has been studied by Oman [15] using a variety of approximation methods including Sobolev H^1 reconstruction, total variation approximation, low pass Fourier reconstruction, and wavelet methods (in which de-noising in the manner of Donoho and Johnstone [6] was used for Harr and Daubechies wavelets). The PGA results for $n = 3$ are superior to (or approximately the same in the case of the total variation method) the results reported by Oman. It can be further noted that the PGA results do not use any estimate of the variance of the noise in the data.

The next example compares median filtering with PGA for a Gaussian hump.

Example 2: Let g be the Gaussian function $g(x) = e^{-x^2/2\sigma^2}$. Fig. 2(a) shows g for $\sigma = 10$. Fig. 2(b) shows the PGA approximation u for $n = 2$ and $d = 3$. The break points in u occur at $x = \pm\sigma$ which are also the zero-crossings of the Laplacian of g . In contrast the median filter results do not depend on the variance of the Gaussian; instead the breakpoints are determined entirely by the window diameter as seen in Fig. 2(c) for a window of size $d = 11$ and Fig. 2(d) for a window of size $d = 51$.

Example 3: Fig. 3(a) shows a lace image from the Brodatz texture album. This image gives a nice illustration of how to preserve lines while smoothing interior regions. The thin stems in the image are 2 pixels wide. Using $n = wd$ to preserve lines of width w in a $d \times d$ window we selected $n = 6$ for a 3×3 window (using 5 PGA iterations). The results are seen in Fig. 3(b). Note that the interiors of the leaf regions are smoothed without loss of boundary definition and the stems are preserved. However where the stems neck-down to lines that are just one pixel wide, PGA with $n = 6$ produces a break in the stem. This indicates the need for an adaptive scheme to select the peer group size at each pixel (see Section IV). For this image we restricted the peer group size to lie between 3 pixels (since this preserves lines of one pixel width in a 3×3 window) and 8 (for smoothing the interior leaf regions); as can be seen the stems now have no breaks.

A. PGA and Shock Filtering

In its simplest form for signals, shock filtering uses the original signal g as initial data for a nonlinear convection equation: $u_t = -\text{sgn}(u_{xx})u_x$ with $u(x, 0) = g(x)$. In this formulation we

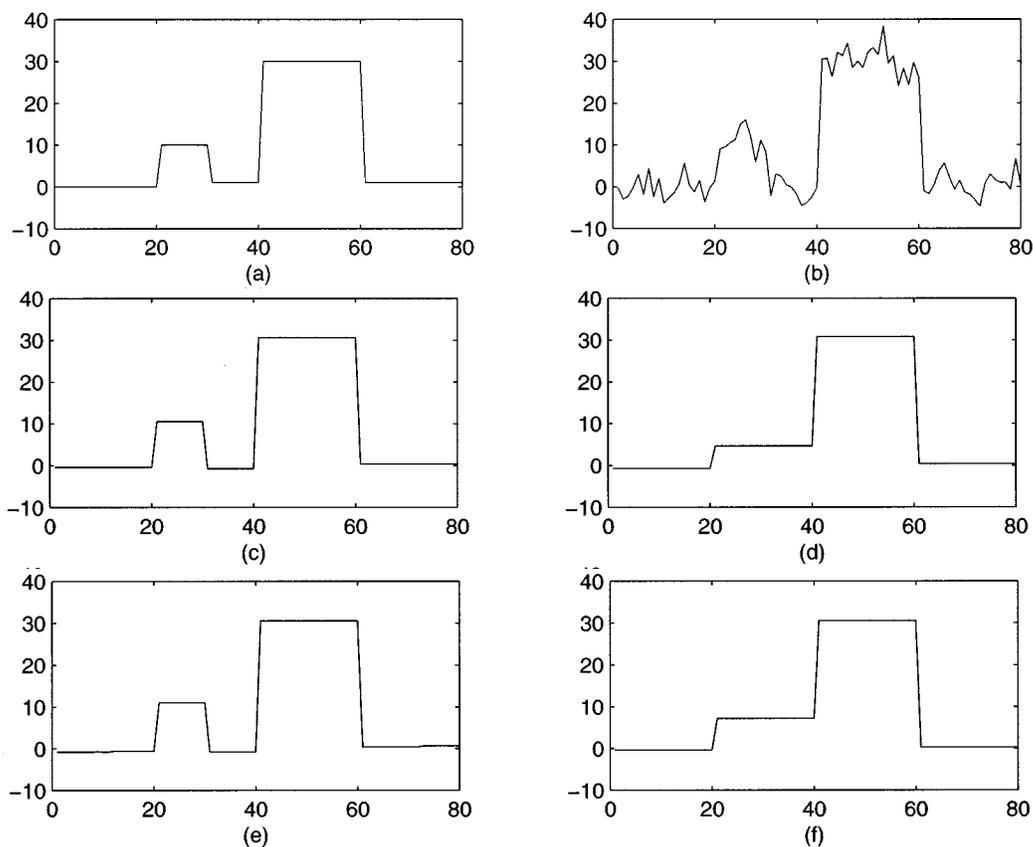


Fig. 1. PGA parameter choice effects for a noisy step function. The window size is $d = 2n - 1$ (see text for details) where n is the peer group size. (a) Original signal with two steps, (b) signal with additive Gaussian noise (zero mean and standard deviation three), (c) PGA result with $n = 9$, (d) PGA result for $n = 11$, and (e)–(f) the results with automatic parameter selection for window sizes $d = 11$ and $d = 21$, respectively.

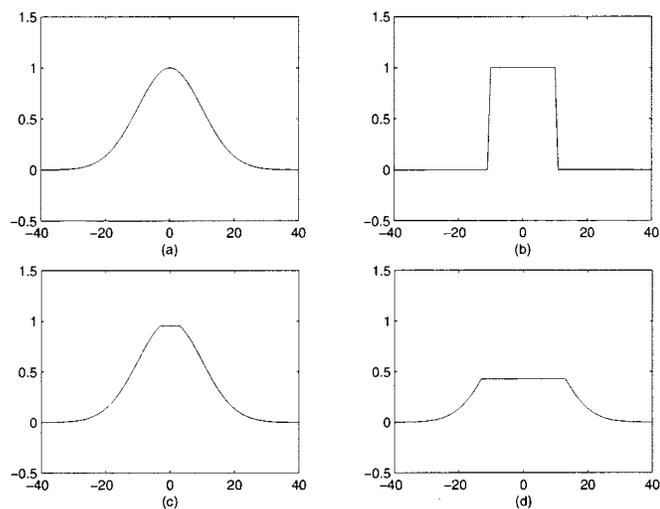


Fig. 2. Comparing PGA with median filtering. (a) Gaussian hump with $\sigma = 10$, (b) PGA result with $n = 2$ and $d = 3$. The break points occur at $x = \pm\sigma$. In contrast, the break points for median filtering are determined entirely by the window size d , as shown in (c) for $d = 11$, and (d) for $d = 51$.

must be careful to form derivative approximations from the appropriate direction. Thus if intensity information is to move from right to left, then we want u_x to represent the right hand derivative and we use a forward difference to approximate u_x . Similarly we use a backward difference if we want intensity information to move from left to right.

Consider a simple Euler update scheme for the shock filter equation: let h be the time step and set $u_i^{new} = u_i + hu_t$. If u is monotone increasing at i and $u_{xx} < 0$ in the sense that $u_{i+1} - 2u_i + u_{i-1} < 0$ then the choice $h = 1/2$ leads to $u_i^{new} = (u_i + u_{i+1})/2$. This is the same result we would get with PGA for a peer group of size $n = 2$ because the convexity condition $u_{i+1} - 2u_i + u_{i-1} < 0$ is the same as $|u_{i+1} - u_i| < |u_i - u_{i-1}|$ when u is monotone increasing. Similarly, if $u_{xx} > 0$ the choice $h = 1/2$ in the shock filter Euler update leads to the same result as the PGA update: $u_i^{new} = (u_{i-1} + u_i)/2$.

This intersection of shock filtering and PGA for particular parameter choices means that results for one method apply immediately to the other. For example, PGA with $n = 2$ for signals is total variation preserving because the same is true for shock filtering. However, the two methods are not the same for other choices of parameters. In particular PGA with larger peer group sizes automatically incorporates smoothing over the peer group and is able to handle problems such as the isolated intensity spikes of salt and pepper noise.

IV. AUTOMATIC PARAMETER SELECTION

Although the preceding observations make it possible to predict in a general way how the peer group size affects the smoothing under PGA, it is still the case that in most images we want to vary the peer group size from point to point in order to enhance some features and smooth others. For example, if we use a 3×3 window then a peer group of size 6 preserves straight edges but not corners. If we lower the peer group number to size 4, then corners are also preserved but we do not achieve the smoothing that we see with $n = 6$.

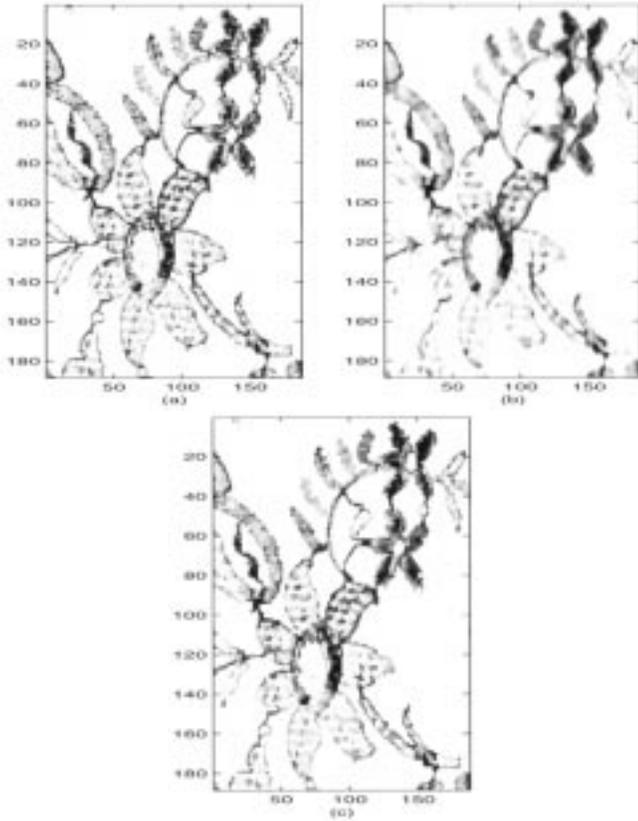


Fig. 3. PGA parameter choice effects for a Brodatz lace image. (a) Original image, (b) PGA smoothing with $n = 6$, and (c) PGA smoothing with adaptive parameter selection preserves the edges better.

To get around this problem, in [4] we introduced the idea of using the Fisher discriminant to select the peer group for each pixel. That is for a particular pixel let g_1, g_2, \dots, g_m be the intensity values over the window with g_c the intensity of the central pixel. Form the intensity differences $d_i = |g_i - g_c|$. Use the Fisher discriminant to separate these differences into two groups. That is, maximize the objective functional

$$F(k) = \frac{|a_1 - a_2|^2}{v_1 + v_2}$$

over the peer group k , where

$$a_1 = \sum_{i=1}^k d_i/k, \quad a_2 = \sum_{i=k+1}^m d_i/(m-k+1)$$

$$v_1 = \sum_{i=1}^k (d_i - a_1)^2, \quad v_2 = \sum_{i=k+1}^m (d_i - a_2)^2.$$

Fig. 1(e) and (f) show the results of PGA using automatic parameter selection for two choices of the window size d . It is observed that automatic parameter selection is not sensitive to the specific choice of window size within reasonable limits (see example 1). Fig. 3(c) shows the result of using the Fisher discriminant on the lace picture from the Brodatz album. Notice the improvement over Fig. 3(b) in preserving the image details.

A complexity analysis of the PGA (including the automatic parameter selection) is given in Appendix B.

Example 4: Fig. 4(a) shows a noisy step function. Using PGA with a 3×3 window and peer group size $n = 6$ provides excellent smoothing but degrades the corner of the step [Fig. 4(b)]. A smaller peer group size $n = 3$ preserves the corner but is not as good in its smoothing,

as shown in Fig. 4(c). To remedy this we use PGA with the Fisher discriminant maximized over the peer group size $4 \leq n \leq 8$ since $n = 4$ preserves corners. The result is seen in Fig. 4(d). The corner is now preserved and the noise is well damped. Fig. 5 shows the values of the Fisher discriminant objective function for the corner pixel with the peer group number varying between 2 and 8. The clear maximum at $n = 4$ indicates the presence of a corner.

V. COLOR IMAGE PROCESSING

The definition of peer group can be easily generalized for color images and multi-spectral images. For color pixels, 3-D color vectors are used instead of the intensity values in gray-scale images. Color similarity between two color vectors can be measured by their Euclidean distance. Color similarity is used to determine the peer group. We adapt the PGA method presented above to color images with two slight modifications: first, the differences in the distances d_i are used to identify potential *noisy* pixels. These noisy pixels are not used in estimating the peer group size. Second, instead of the simple average of the peer group members, we use a weighted average where the weights decrease exponentially as the distance of the peer group member increases from the center pixel. This is modeled using a standard Gaussian (variance = 1).

The presence of impulse noise may affect the PGA performance on color images. To address this, we first calculate the differences of the distances d_i before the peer group classification using the Fisher discriminant method. Let $f_i = d_{i+1} - d_i$. The first and last few M values in this ordered sequence are tested to see if $f_i \leq \alpha$, where α is set to a high value for images with a low signal-to-noise ratio. Those pixels that fail the test are not used in the Fisher discriminant method for adaptively finding the peer group members. In the experiments we use $M = d/2$ where $d \times d$ is the size of the window used.

If the purpose is to remove impulse noise and not to smooth the image, the center pixel in the window is first checked to see if it is a possible noisy pixel. Only the noisy pixels are replaced. The peer group in this special case has only one member which is the vector median [2] of the local window. This approach is similar to the SDRM method proposed in [1].

Example 5: To test the effectiveness of PGA for impulse noise removal (no smoothing), the pixels of “baboon” and “pepper” images are corrupted by randomly generated impulse noise. Different percentages of the total number of pixels are corrupted. The PGA method compared with the Vector Median Filter (VMF) [2] and the Teager-operator method (TEA) [3]. The window size used is 3×3 and the color space is RGB for all the methods. The α parameter for the PGA and the TEA is tuned to obtain the best results for each case. The results are tabulated in Tables I and II. The “none” column indicates the SNR without any noise removal. In both the cases, the PGA method performs better than the other two methods. Fig. 6 illustrates the effects of peer group processing compared to other methods for removal of impulse noise in color images. Shown in (a) is a small area in the “baboon” image, and (b) shows the same area after corruption. Fig. 6(c) shows the result using the VMF, (d) shows the result using the TEA method, and (e) shows the result of peer group processing. It can be seen that VMF removes the noise but also changes the color of other pixels while TEA fails to replace the noise with a similar color to the original one. PGA gives the best approximation to the original image.

Example 6: Fig. 7 illustrates the use of PGA for color image smoothing. A part of the “baboon” image is shown in Fig. 7(a). The result of PGA is shown in (b) and the result of Gaussian filtering is shown in (c) for comparison. Window size is 5×5 . It can be seen that the peer group processing approach smooths the color image without blurring the details compared to Gaussian filtering.

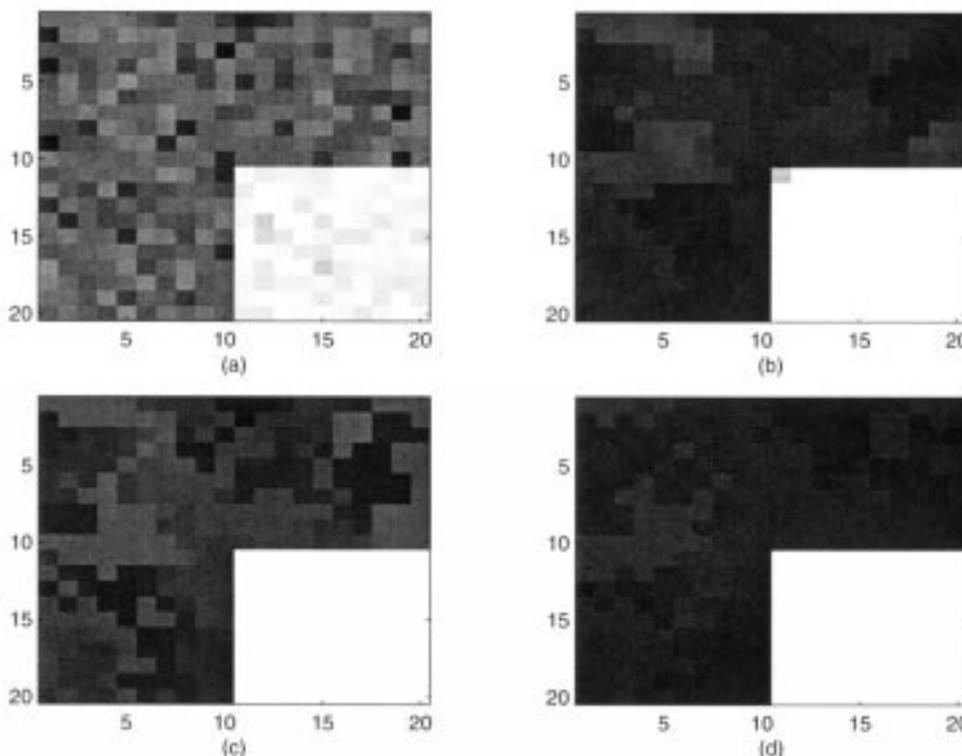


Fig. 4. Noisy step example. (a) Noisy step, (b) PGA result with $n = 6$ blurs the corner, (c) PGA with $n = 3$ preserves the corner but the smoothing is not good, and (d) the adaptive PGA method preserves the corner as well as smooths the noisy step.

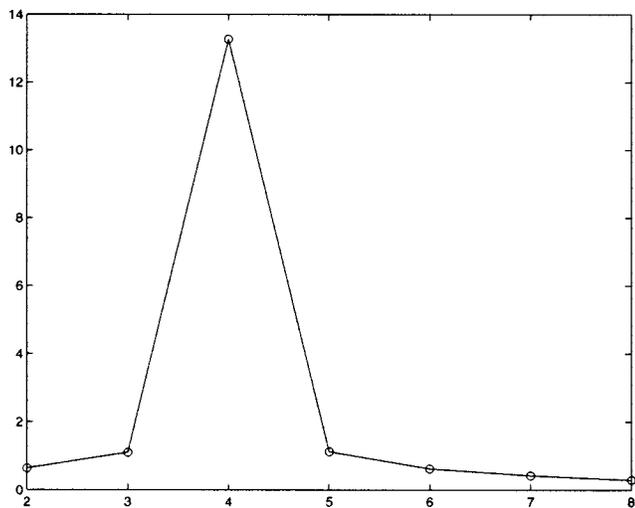


Fig. 5. Fisher discriminant values for peer group sizes between two and eight for the example shown in Fig. 4. Note that the maximum at $n = 4$ corresponds to the presence of the corner.

TABLE I
SNR (IN DECIBELS) FOR “BABOON” IMAGE

| Noise | None | VMF | TEA | PGF |
|-------|------|------|------|------|
| 1 % | 24.1 | 18.1 | 30.2 | 32.5 |
| 5 % | 17.0 | 17.9 | 24.4 | 26.4 |
| 10 % | 14.0 | 17.6 | 21.8 | 23.6 |
| 20 % | 11.0 | 17.1 | 18.7 | 20.6 |

TABLE II
SNR (dB) FOR “PEPPER” IMAGE

| Noise | None | VMF | TEA | PGF |
|-------|------|------|------|------|
| 1 % | 23.4 | 30.9 | 38.9 | 41.2 |
| 5 % | 16.3 | 28.8 | 31.8 | 34.6 |
| 10 % | 13.3 | 27.0 | 28.0 | 31.2 |
| 20 % | 10.3 | 24.7 | 24.2 | 27.4 |

VI. MULTISCALE PGA

One problem with PGA is the limitation to small windows for computational speed. In particular it would be nice to be able to obtain uniform smoothing over large regions without having to use large windows and peer groups. To achieve this we have developed a multilevel PGA procedure similar in spirit to multi-grid methods for solving large systems of linear equations. We work on several levels by defining windows with skips between pixels. At the first level is the usual window with a distance of 1 between pixels; the next level has a distance of 2 between pixels etc. Let W_k be the window at (i, j) with a skip of k between pixels. For example for a 3×3 window, $W_k = \{(i - k, j - k), (i - k, j), (i - k, j + k), (i, j - k), (i, j), (i, j + k), (i + k, j - k), (i + k, j), (i + k, j + k)\}$.

Define the PGA iteration at level k as the usual PGA iteration where the peer group is selected from the window W_k rather than W_1 . Thus the only difference in the PGA iteration at different levels is the window from which we select the peer group; as a consequence the computational effort of doing one PGA iteration at level k is the same as the effort of doing one iteration at level $k = 1$. This is the procedure that we use in the example below; as an alternative one can obtain additional speed by only doing the PGA iteration at level k at every k th pixel in

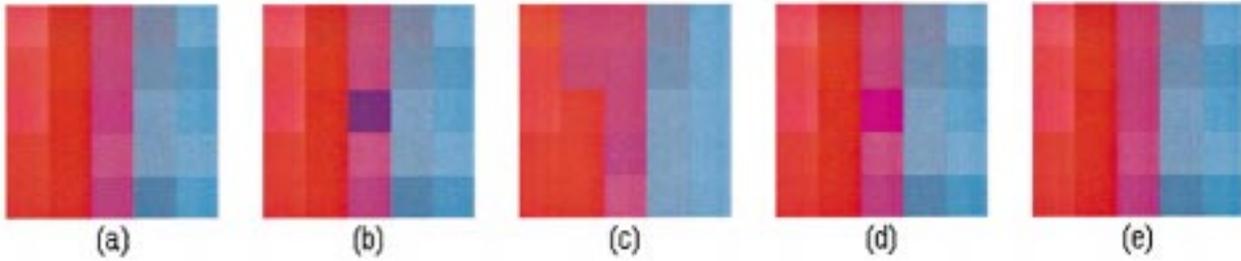


Fig. 6. (a) Small area of the original “baboon” image, (b) same area of the corrupted image, (c) result of the vector median filtering, (d) result of Teager-operator method, and (e) result of peer group processing.

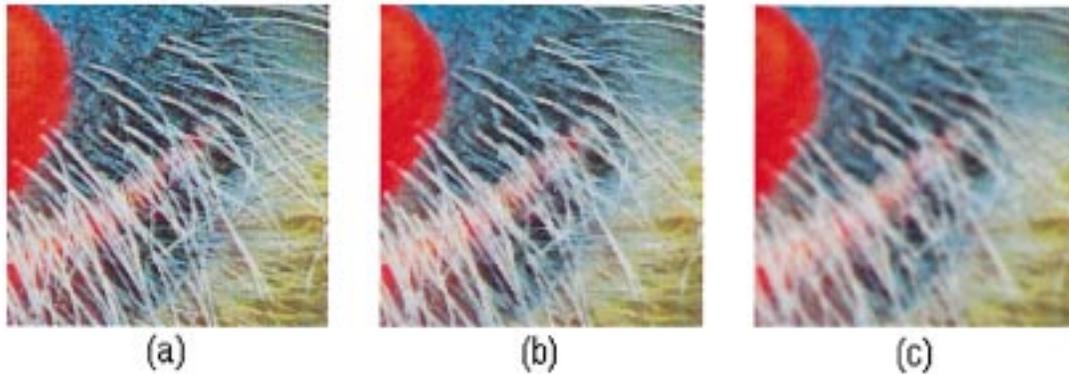


Fig. 7. (a) Part of the original “baboon” image, (b) result of PGA, and (c) result of Gaussian filtering.

i and j ; this would reduce the computational cost by a factor of $1/k^2$ at the k th level while producing nearly the same smoothing effect over large distances when iterating over several levels.

Alternating the PGA iteration between levels results in speeding the passage of intensity information within regions. As a side note we observe that there is a simple way to implement the PGA interaction at level k without modifying the original PGA program. For example, to do a PGA iteration with a distance of 2 between pixels in each window, one simply has to subsample the image skipping every other pixel and then run regular PGA on the subsampled image. Subsampling this way transforms a large image into four smaller images denoted by g_{11} , g_{12} , g_{21} and g_{22} depending on whether the first pixel in the subsample is $(1, 1)$, $(1, 2)$, $(2, 1)$ or $(2, 2)$. After doing one PGA iteration on each of the smaller images they are then recombined into a larger image. Similarly for level k we can break g into k^2 smaller images, run PGA on each of the smaller images and then recombine these smaller images into one larger image.

There are a variety of ways to implement a multilevel PGA scheme. In our experiments we used the following procedure: do one PGA iteration at level 1, then one iteration each at levels 2, 4, \dots , 2^m . This is referred to as a PGA cycle through level 2^m . Repeat starting at level 1. The selection of the final level 2^m is determined by the amount of overall smoothing that we want. Extended objects in the image are preserved and smaller objects are eliminated. The extent to which this occurs can be partially analyzed by noting that an object O of radius r will necessarily change in intensity (in the generic case) when doing PGA at level $k > r$ because the peer group for any pixel in O will include pixels outside O .

In practice we found that this multilevel PGA method converged rather quickly and that little additional change occurred after three to five cycles for $m = 4$. More details and applications can be found in [5] and [9].

Example 7: Fig. 8(a) shows a satellite image of an agricultural area. In Fig. 8(b) we see the result of regular PGA with a 3×3 window and peer group $n = 6$. Fig. 8(c) shows the result of using multilevel PGA with 3×3 windows and peer group size $n = 6$ with $m = 4$. Note the enhanced smoothing within regions and the elimination of the small sensor artifacts which show up as a series of white blobs running down the center of the original image. Since these blobs are rather small $r < 2^m$ they are attenuated by the averaging process under subsampling. This suggests that multilevel PGA could be used in an object detection top-hat procedure.

VII. CONCLUSIONS

A peer group image processing method has been presented that has several natural advantages over competing methods. Automatic local parameter selection allows the adaptive form of PGA to preserve edges and corners while obtaining significant smoothing over uniform regions in the image. The computational effort of adaptive PGA is many times less than that of PDE based procedures (such as variational methods) that achieve similar results. PGA also has the advantage of ease of implementation and extension to multilevel and vector applications. The simplicity of the underlying idea allows analytic results to be obtained concerning the convergence of the PGA iteration, relations to other methods such as shock filtering and median filtering, and how the choice of parameters affects the resulting approximation. The parameters of the PGA method are directly related to the characteristics of the image features that are to be enhanced. This is in contrast to most image enhancement methods in which it is unclear how to select the method parameters or weights to achieve a desired result.

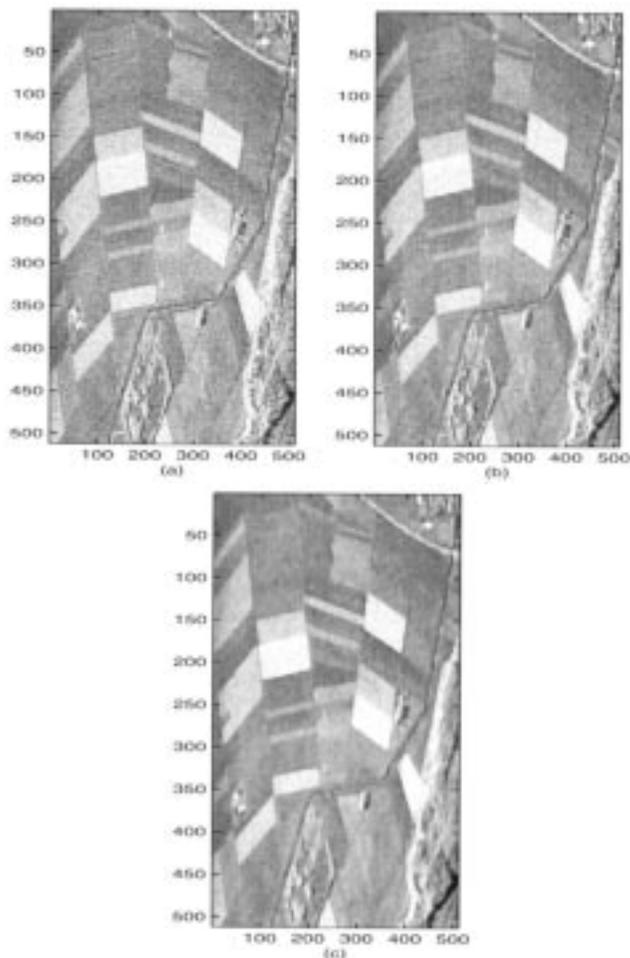


Fig. 8. Comparing (a) the original image, (b) after PGA, and (c) after multilevel PGA.

APPENDIX A PROOF OF LEMMA 1

For $i \leq k$, the convexity assumption can be written as $g_{i+1} - g_i > g_i - g_{i-1}$. Since g is monotonically increasing, both sides of this inequality are nonnegative. Consequently, for $i \leq k$, the value g_i is closer to g_{i-1} than to g_{i+1} . This means that for $2 \leq i \leq k$, the peer group for pixel i consists of pixels $i-1$ and i ; i.e., we are averaging to the left. For $i=1$, the peer group consists of pixels 1 and 2. It is easy to show that both the monotonicity and convexity of the function are preserved and the peer groups remained fixed. In particular, the peer group for pixels 1 and 2 are the same. Thus, the intensity values for these pixels converge in one step to their average and then remain constant. Since the pixels $2 < i \leq k$ are left averaging, they converge to the common value of pixels 1 and 2; i.e. to $(g_1 + g_2)/2$. Similar arguments show that the function values of pixels $k < i \leq n$ converge to the common value $(g_{n-1} + g_n)/2$ which completes the proof.

APPENDIX B OPERATION COUNTS FOR PGA AND MULTILEVEL PGA

In order to find the PGA replacement value for pixel i from a window of size d (note: we use d to denote both 1-D and 2-D data. For 2-D size d implies a window of $d \times d$ pixels centered at i), and a peer group of size n , we first must form the differences $d_{ij} = |g(i) - g(j)|$. Here, i and j index the pixel locations. We assume that the intensity may be a vector

of dimension D_I . For example for color PGA, $D_I = 3$. The computational cost per difference is then equal to the intensity dimension D_I . The number of differences that must be computed per window is equal to the number of pixels in the window. Let n_d denote the number of pixels in the window of size d . The total cost for computing the differences over the window is given by $C_{diff} = D_I n_d$.

The next step is to find the n smallest differences from the set of n_d differences. This gives us the peer group for pixel i . This sorting operation can be accomplished in $C_{sort} = n n_d$ comparisons. Finally we average over the n intensities in the peer group to get the PGA replacement value for pixel i . For intensities of dimension D_I this has a cost of $C_{average} = n D_I$.

Thus, the total computational cost C_{total} per pixel per PGA iteration for peer groups of size n , windows of size d , and intensities of dimension D_I is $C_{total} = C_{diff} + C_{sort} + C_{average}$. If we count additions the same as comparisons we obtain $C_{total} = D_I n_d + n n_d + n D_I$. Thus, for gray scale images we have $C_{total} = (n+1)d^2 + n$ and for color images (three-color space) $C_{total} = (3+n)d^2 + 3n$.

A. Operation Counts for Adaptive PGA

PGA based on the Fisher discriminant can be analyzed in a similar way. We first compute the differences $d_{ij} = |g(i) - g(j)|$ as in regular PGA. Again the cost is the same, $C_{diff} = D_I n_d$. Next, we will need to assess the value of the Fisher discriminant over the desired range of peer groups. For simplicity we assume that we will consider all possible peer group sizes n in the range $1 \leq n \leq n_d$ where as above n_d is the number of pixels in the window of size d . This requires that the differences be sorted from smallest to largest. Let $r_1 \leq r_2 \leq \dots \leq r_{n_d}$ be the sorted differences. The cost of this is given by $C_{sort} = O(n_d \log n_d)$.

For a given peer group size n , the Fisher discriminant value is

$$F(n) = \frac{|A_1(n) - A_2(n)|}{V_1(n) + V_2(n)}$$

where $A_1(n)$ and $V_1(n)$ are respectively the average and variance over r_1, \dots, r_n ; $A_2(n)$ and $V_2(n)$ are respectively the average and variance over r_{n+1}, \dots, r_{n_d} . However we can exploit update formulas for the averages and variances as n changes to $n+1$ to avoid computational redundancy. In this case we find that the total cost of computing $F(1), F(2), \dots, F(n_d)$ is

$$C_{Fisher} = 11 n_d.$$

Selecting the peer group number with the maximal Fisher value requires another n_d comparisons. The average for this group has already been calculated so we find the total cost per pixel of one adaptive PGA iteration is $C_{total} = C_{diff} + C_{sort} + C_{Fisher} + n_d$

$$C_{total} = D_I n_d + O(n_d \log n_d) + 12 n_d.$$

ACKNOWLEDGMENT

C. Kenney would like to thank Prof. S. Osher for pointing out the connection between PGA and shock filtering. The authors also thank the reviewers for their constructive criticisms and Prof. S. Chandrasekaran for many fruitful discussions.

REFERENCES

- [1] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. Image Processing*, vol. 5, pp. 1012–1025, 1996.
- [2] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, pp. 678–89, Apr. 1990.
- [3] F. A. Cheikh, R. Hamila, M. Gabbouj, and J. Astola, "Impulse noise removal in highly corrupted color images," in *Proc. Int. Conf. Image Processing*, vol. 1, 1996, pp. 997–1000.

- [4] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, "Peer group filtering and perceptual color image quantization," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 1999, pp. 21–24.
- [5] Y. Deng, G. Hewer, C. Kenney, and B. Manjunath, "Peer group image processing," Univ. Calif., Santa Barbara, ECE Tech. Rep., Nov. 1999.
- [6] D. L. Donoho and I. M. Johnstone, "Ideal de-noising in a basis chosen from a library of orthogonal bases," *Comptes Rendus Acad. Sci. Paris A*, vol. 319, pp. 1317–1322, 1994.
- [7] G. Hewer, C. Kenney, and B. Manjunath, "Variational image segmentation using boundary functions," *IEEE Trans. Image Processing*, vol. 7, pp. 1269–1282, Sept. 1998.
- [8] D. G. Krakos and P. E. Trahanias, "Combining vector median and vector directional filters: The directional distance filters," in *Proc. of ICIP*, vol. 1, 1995, pp. 171–174.
- [9] C. Kenney and G. Hewer, "Partial differential equations for multiscale image and video processing," in *Applications of Mathematical Signal Processing Techniques to Mission Systems*, Nov. 1999, RTO Lecture Series 216, pp. 6-1–6-13.
- [10] H. G. Longbotham and D. Eberly, "Statistical properties, fixed points, and decomposition with WMMR filters," *J. Math. Imag. Vis.*, vol. 2, pp. 99–116, 1992.
- [11] H. G. Longbotham and D. Eberly, "WMMR filters: A class of robust edge enhancers," *IEEE Trans. Signal Processing*, vol. 41, pp. 1680–1684, 1992.
- [12] J. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Boston, MA: Birkhauser, 1995.
- [13] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Conf. Computer Vision Pattern Recognition*, San Francisco, CA, 1985, pp. 22–26.
- [14] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. XLII, no. 4, 1989.
- [15] M. Oman, "Study of variational methods applied to noisy step data," 1996.
- [16] J. M. Ortega, *Matrix Theory A Second Course*. New York: Plenum, 1987.
- [17] S. Osher and L. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numer. Anal.*, vol. 27, pp. 919–940, 1990.
- [18] S. Osher and L. Rudin, "Shocks and other nonlinear filtering applied to image processing," in *Proc. SPIE Appl. Dig. Image Proc. XIV*, vol. 1567, 1991, pp. 414–430.
- [19] L. Rudin, "Images, numerical analysis of singularities, and shock filters," Ph.D. dissertation, Computer Science Dept., Caltech, Pasadena, CA, 1987.
- [20] G. W. Stewart, *Introduction to Matrix Computations*. New York: Academic, 1973.
- [21] K. Tang, J. Astola, and Y. Neuvo, "Adaptive nonlinear multivariate image filtering for mixed noise removal," in *Proc. ISCAS*, vol. 1, 1993, pp. 427–430.
- [22] P. E. Trahanias and A. N. Venetsanopoulos, "Vector directional filters—A new class of multichannel image process filters," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 528–534, 1993.
- [23] L. P. Yaroslavsky, "Linear and rank adaptive filters for picture processing," in *Digital Image Processing and Computer Graphics: Theory and Applications*, L. Dimitrov, E. Wenger, W. Muenchen, and R. Oldenburg, Eds., 1991, p. 374.
- [24] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 157–192, 1996.