

# A NONLINEAR ADAPTIVE REGRESSION PROCESS FOR NOISE CORRUPT IMAGES

Nan Jiang<sup>†</sup>, Changchun Li<sup>†</sup>, Jennie Si<sup>†</sup>, and Glen P. Abousleman<sup>‡</sup>

<sup>†</sup>Arizona State University, Temp, AZ 85287

<sup>‡</sup>General Dynamics C4 Systems, Scottsdale, AZ 85257

## ABSTRACT

Most existing nonlinear regression filtering techniques for image denoising are claimed to be edge preserving without considering the pixel position information. This will cause speckling effects on the denoised image and inconsistent smoothing in the vicinity of texture-rich areas. This paper proposes a novel denoising method to address this problem. The proposed method removes the low to intermediate noise using edge-preserving range filtering, thereby removing short, false edges. The updated edge map is used for subsequent filtering in which pixel intensities are smoothed according to their minimum distance to the closest edge point. This procedure is carried out in an iterative scheme until the edge map stabilizes. We compare existing denoising algorithms with the proposed method. Experimental results validate the effectiveness and efficiency of the proposed method.

**Index Terms**— image denoising, wavelet shrinkage, partial differential function, bilateral filtering, local data adaptive

## 1. INTRODUCTION

Image denoising is widely used as the first step in image registration, image compression, and other related areas. In many applications, image denoising is used to produce good estimates of the original image from noisy observations. The restored image should contain less noise than the observations, while preserving detailed information such as edges.

Approaches for image denoising reside in three main categories: Wavelet-based thresholding such as wavelet shrinkage; Various spatial kernel regression approaches such as bilateral filtering; Methods with partial differential equations (PDE). Methods using wavelet shrinkage [1] appropriately modify (or suppress) small coefficients to remove noise from the reconstructed image. However, the assumption that small absolute value coefficients are likely dominated by noise is not always true when the energy of Gaussian additive noise is large. Bilateral filtering [2] combines closeness and similarity filtering. The similarity filtering is designated to preserve edges since the intensity closeness across edges is large. However, this scheme is deficient at handling speckle pixels

whose intensity values are either much larger or smaller than those of their neighboring pixels. The similarity smoothing effect is close to zero when the range gap is large. The idea behind fourth-order PDF [3] is to minimize a cost function proportional to the absolute value of the Laplacian of the image intensity function. Fourth-order PDF has the same problem as bilateral filtering at handling speckles in noisy images, since the cost function is designed to increase reciprocally with the absolute value of the Laplacian.

This paper proposes a novel denoising method that addresses the deficiencies of existing methods for speckle and heavy noises. The proposed method removes the low to intermediate noise using an edge-preserving range filter, thereby removing short false edges. The updated edge map is used for subsequent filtering in which pixel intensities are smoothed according to their minimum distance to the closest edge point. This procedure is carried out in an iterative scheme until the edge map stabilizes.

The remainder of this paper is organized as follows. In Section 2, our feature-preserving nonlinear regression method is presented. Section 3 presents the results of the proposed denoising method on test images with varying levels of additive white Gaussian noise. Finally, Section 4 presents a summary.

## 2. PROPOSED DENOISING ALGORITHMS

In this section we present the motivation for a more efficient denoising algorithm, and the formulation of our local-data-adaptive kernel and edge distance filtering algorithm.

### 2.1. Local Data Adaptive Kernel

As discussed in [4], many denoising schemes such as wavelet transformation and bilateral filtering can be categorized as certain type of kernel regression process, i.e., approximating the original signal with a linear combination of the convolutions with a selected kernel. Local-data-adaptive kernel regression considers both the sample location and the intensity properties of the local data. Therefore, the effective size and strength of the kernel is adaptive to the local image feature. The kernel for intensity value  $y$  at position  $\mathbf{x}$  and intensity values  $\{y_i\}$  in the neighborhood  $\{\mathbf{x}_i\}$  can be denoted as  $K_{adapt}(\mathbf{x}_i - \mathbf{x}, y_i - y)$ .

This work was supported by General Dynamics C4 Systems, and in part by NSF under grant ECS-0002098.

Therefore, the task of denoising becomes one of finding such a kernel that is adaptive to the local image feature. One of the most often used image features is the edge map. An edge is a sequential link of the first-directional-derivative's maxima and minima in the direction of the gradient. The common idea is that in areas near edges, the kernel orientation should be along the edge direction rather than across it, and the kernel size should be small. For the areas some distance from an edge, the kernel size can be relaxed for fast regression. Assume that the edge map is known for a given image, it is intuitive to employ a kernel whose radius corresponds to the closest edge pixel in the horizontal and vertical directions, respectively. That is,

$$K_{adapt}(\mathbf{x}_i - \mathbf{x}, y_i - y) = G(y_i - y, \sigma_s)K_G, \quad (1)$$

where

$$K_G = \rho(u_x(\mathbf{x}), u_y(\mathbf{x}))G(\mathbf{x}_i - \mathbf{x}, \sigma_x) \quad (2)$$

where  $u_x(\mathbf{x})$  represents the minimum horizontal distance from the kernel center,  $\mathbf{x}$ , to the neighboring edge pixel. Similarly  $u_y(\mathbf{x})$  represents the minimum distance from kernel center,  $\mathbf{x}$ , to the neighboring edge pixel in the vertical direction. Note that  $u_x(\mathbf{x})$  and  $u_y(\mathbf{x})$  correspond to the local gradient structure.  $\rho(\cdot)$  is a pillbox function centered at the kernel, that limits the kernel support to a certain radius such that

$$\rho(u_x(\mathbf{x}), u_y(\mathbf{x})) = \begin{cases} 0, & \text{if } \|\mathbf{x}_i - \mathbf{x}\| \geq a \\ a = \min(\sqrt{u_x(\mathbf{x}_i)^2 + u_y(\mathbf{x}_i)^2}, r_{max}) \\ 1, & \text{otherwise,} \end{cases} \quad (3a)$$

$$(3b)$$

where  $r_{max}$  is a predefined maximum contour radius that prevents the kernel from expanding too much in flat areas.

If we use a Gaussian kernel, the terms at the right side of Equation 1 and Equation 2 can be represented mathematically as shown below.

$$G(y_i - y, \sigma_s) = \exp\left(-\frac{(y_i - y)^2}{2\sigma_s^2}\right), \quad (4)$$

where  $\sigma_s$  controls the smoothing on intensity offsets.

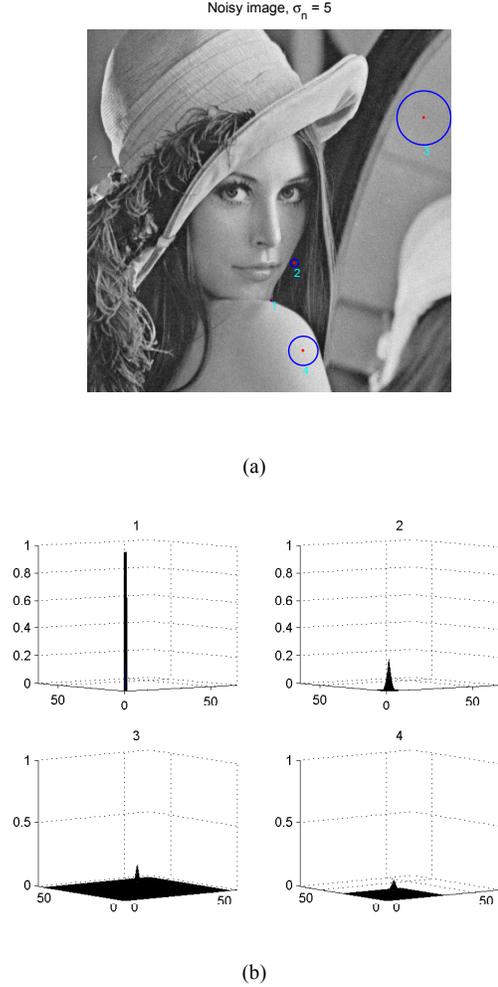
$$G(\mathbf{x}_i - \mathbf{x}, \sigma_x) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma_x^2}\right), \quad (5)$$

where  $\sigma_x$  is the scale factor that determines the smoothing. It is defined as

$$\sigma_x = \frac{g + \alpha}{\alpha} \sigma_c, \quad g = \sqrt{\hat{I}_x^2 + \hat{I}_y^2}, \quad (6)$$

where  $\alpha > 1$  is a regularization constant that prevents  $\sigma_x$  from becoming extremely large,  $\hat{I}_x$  and  $\hat{I}_y$  denote the smoothed first derivative at  $\mathbf{x}_i$  in the  $x$  and  $y$  directions, respectively. It is easy to see that  $g$  is the magnitude of the gradient at  $\mathbf{x}_i$ .

Figure 1 shows an example of how  $K_G$  adapts to various local data structures of the Lena image. Note that in flat areas, it is spread out to suppress the noise while depending on the underlying features. Also note that the scale factor,  $\sigma_x$ , is relatively large to remove speckle pixels in the flat areas, whereas in texture-rich areas, the contour size is small to preserve the details. For the special case edge pixels, even



**Fig. 1:** Examples of kernel adaptive to the local data structure. The image is degraded by adding additive White noise with  $\sigma_n = 5$ . Circles show the areas covered by the kernels.

though they generally have high gradient magnitude, which results in a relatively high  $\sigma_x$  value in  $G(\mathbf{x}_i - \mathbf{x}, \sigma_x)$ , it is penalized by  $\rho(u_x(\mathbf{x}), u_y(\mathbf{x}))$ , which equals to zero when  $u_x(\mathbf{x})=u_y(\mathbf{x})=0$ . Therefore, the information at edge pixels is preserved.

The potential pitfall of the combination term,  $K_G$ , is that it assumes a reliable edge map is available. However, this is not always true in practice, particularly if a noise disturbance exists. If some pixels are falsely categorized as edge pixels

due to noise, the above combination term is not able to remove them. Observations indicate that the false edges from noisy pixels are generally of smaller intensity offset from the neighboring pixels if compared to true edge pixels. As a result, it is possible to remove them by the kernel term in Equation 4. Accordingly, the steps of thinning and hysteresis in the Canny edge detector [5] remove the spurious oscillations effectively. Edge maps generated by the Canny method contain structural information that is more stable than single feature points.

## 2.2. Iterative Regression

Our proposed denoising kernel is adaptive to the combination of true image information and the noise information. Consequently, it is more effective in an iterative scheme in that the output image of each iteration is closer to the underlying genuine data feature, which in turn optimizes the proposed kernel in next iteration. The proposed iterative algorithm is presented below:

1. Initialize a counter  $n = 0$  for original image  $I^0$ .
2. Get an edge map  $E^n$  from image  $I^n$  using Canny edge detector, followed by Euclidean Distance Transform.
3. Apply similarity filtering defined in Equation 4.
4. Apply filtering defined in Equations 1 and 5, the output image is  $I^{n+1}$ .
5. Perform Canny edge detection on  $I^{n+1}$  again to obtain edge map  $E^{n+1}$ , and then  $E^{n+1} = E^{n+1} \wedge E^n$ .
6. Exit loop if the following stop criterion is satisfied, otherwise increment the counter  $n = n + 1$  and go back to step 2.

$$\exists i : \psi(E^{n+1}, E^n) < \epsilon_E \wedge MSE(I^{n+1}, I^n) < \epsilon_I, \quad (7)$$

where

$$\psi(E^{n+1}, E^n) = \frac{m(E^n) - m(E^{n+1})}{m(E^n)}, \quad (8)$$

where  $m(E^n)$  is the number of edge pixels in image  $E^n$ ,  $MSE(\cdot)$  is defined in Equation 14.  $0 < \epsilon_E < 1$ ,  $\epsilon_I > 0$  is desirable change amount of edge map and intensity values, respectively.

## 2.3. Edge Distance Filtering (EDF)

Edge Distance Filtering is the linear combination of the weighted average of pixel values in the neighborhood in which the weights are determined by the *Edge Distance difference* between the pixel at the kernel center,  $\mathbf{x}$ , and its neighboring pixels,  $\mathbf{x}_i$ , in addition to *geometric distance*:

$$K_{EDF} = G(u(\mathbf{x}_i, \mathbf{x}), \sigma_u)G(\mathbf{x}_i - \mathbf{x}, \sigma_x), \quad (9)$$

where

$$u(\mathbf{x}_i, \mathbf{x}) = \|u(\mathbf{x}_i) - u(\mathbf{x})\|. \quad (10)$$

$u(\mathbf{x}_i)$  is the Euclidean distance from  $\mathbf{x}_i$  to its closest edge pixel.  $\sigma_u$  denotes the smoothing scale factor.  $g(\mathbf{x}_i, \mathbf{x})$  is defined in Equation 5. If we use a Gaussian kernel, then

$$G(u(\mathbf{x}_i, \mathbf{x}), \sigma_u) = \exp\left(-\frac{u(\mathbf{x}_i, \mathbf{x})^2}{2\sigma_u^2}\right). \quad (11)$$

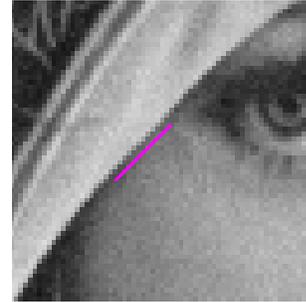
If we use  $\sigma_u \rightarrow 0$ , such that

$$\begin{aligned} G(u(\mathbf{x}_i, \mathbf{x}), \sigma_u) &= 1, \quad \text{if } u(\mathbf{x}_i) = u(\mathbf{x}) \\ G(u(\mathbf{x}_i, \mathbf{x}), \sigma_u) &\rightarrow 0, \quad \text{otherwise} \end{aligned} \quad (12)$$

We further enforce the condition such that:

$$\begin{aligned} S &= \{\mathbf{x}_i : u(\mathbf{x}_i) = u(\mathbf{x}), i = 1, 2, 3, \dots, k\} \\ \forall \mathbf{x}_i : \mathbf{x}_i &\in \{N_j \mid \forall \mathbf{x}_j \in S\}, i \neq j, i, j = 1, 2, \dots, k. \end{aligned} \quad (13)$$

Noisy image,  $\sigma_n = 5$



**Fig. 2:** Example of 1-D EDF filter contour for Lena with  $\sigma_n = 5$ . The EDF filter centers at [230, 279] with a radius of 10 pixels.

where  $N_j$  is the neighborhood of  $\mathbf{x}_j$ .  $\mathbf{x}_i$  in  $N_j$  means  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are neighbors.  $k$  is the number of pixels with the same edge distance as the kernel center,  $\mathbf{x}$ , in the window. The reason we enforce a connected set is that the edge is the boundary of two areas with contrast intensities, and separated sets with the same edge distance might actually reside across the edge. Additionally smoothing over the edges is undesirable in image denoising. The contours of the EDF kernel will be one-dimensional curves only along local edges. This curve-shaped filter is perpendicular to the local orientation and it is slightly bent according to the local edges to better fit the underlying image structure. Figure 2 shows the 1-D EDF filter superimposed on the Lena image. The combination of the 1-D EDF kernel and the local data adaptive kernel compensates for the drawbacks of either one. That is, the 1-D EDF kernel does not filter flat regions well, and the local data adaptive kernel is ineffective with removing speckles in the vicinity of local edges.

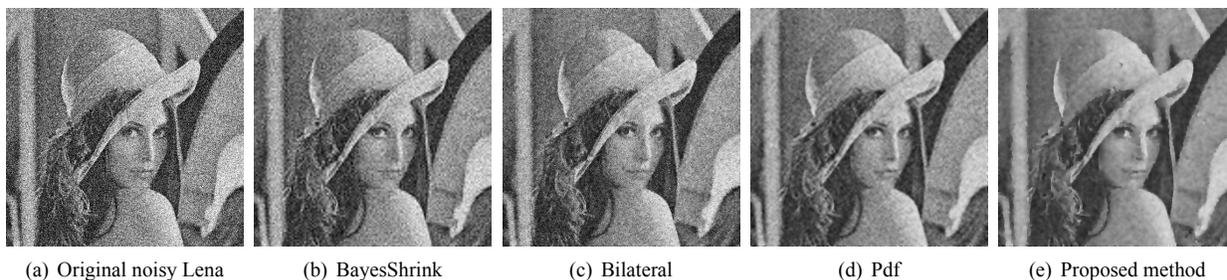


Fig. 3: Visual comparison of various denoising methods on Lena when  $\sigma_n = 60$ .

### 3. EXPERIMENTAL RESULTS

Figure 3 illustrates the visual comparison of the denoising methods discussed in this paper. Observations reveal that our proposed method performs comparably to traditional denoising methods with light noise, and substantially outperforms the others with intermediate to heavy noise.

The mean squared error (MSE) is an averaged pixelwise intensity difference between the ground truth image and the denoised image:

$$MSE(\tilde{I}, \hat{I}) = \frac{1}{N} \sum (\tilde{x} - \hat{x})^2, \quad (14)$$

where  $\tilde{x}$  is the true value of the  $i$ th pixel belonging to the original image,  $\tilde{I}$ , and  $\hat{x}$  is its estimated value from the noisy data belonging to denoised image,  $\hat{I}$ , and  $N$  is the number of pixels in  $\tilde{I}$  and  $\hat{I}$ . The log scale for MSE results can be represented as PSNR:

$$PSNR = -10 \log_{10} \left( \frac{MSE}{255^2} \right) \quad (15)$$

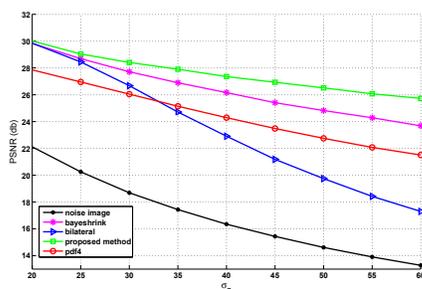


Fig. 4: Comparison of denoising methods using MSE at various noise variance levels. Fourth order pdf iteration times =  $\sigma + 40$ ; Bilateral Filtering  $\sigma_c = 6$ ,  $\sigma_s = 0.2$ ; Bayes Shrink with db4 at level 4; Proposed method  $\sigma_c = 10$ ,  $\sigma_s = 0.5$ , mask size =  $3 \times 3$ .

Figure 4 shows that the proposed algorithm has the highest PSNR values at various noise levels where  $\sigma_n$  is from 20 to 60, which means the reconstructed image after denoising is

closest to the ground truth. Note the PSNR values of bilateral filtering are merely good unless  $\sigma_n > 30$ . This is due to the failure of similarity filtering on highly distorted pixel values in flat areas.

### 4. DISCUSSION AND CONCLUSION

In this paper, a novel local-feature-adaptive denoising method is proposed. We show that the method is an iterative, non-linear, edge-preserving process. In contrast to traditional nonlinear filtering, which defines weighting coefficients without considering the location information, the hysteresis step of the edge detection in our method eliminates speckles effects. Unlike other isotropic kernel regression processes, our method uses EDM-based smoothing to compensate for the disturbance of noise in the vicinity of texture-rich areas, thereby making the entire denoising effect more consistent.

### 5. REFERENCES

- [1] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. on Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.
- [2] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *In Proceedings of International Conference on Computer Vision*, 1998, p. 839.
- [3] M. Lysaker, A. Lundervold, and Xue-Cheng Tai, “Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time,” *IEEE Transaction on Image Processing*, vol. 12, no. 12, pp. 1579–1590, Decemeber 2003.
- [4] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007.
- [5] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, November 1986.