

Anisotropic Filtering with Multi-resolution Edge Analysis and Connectivity Analysis

Bo Jiang

Abstract—Generally, an edge-high frequency information in an image would be filtered or suppressed after image smoothing. This results in noise attenuation, but the image also loses sharpness. This loss of sharpness can impact the usefulness of the processed image for further tasks. Our new anisotropic filtering performs multi-resolution edge analysis and connectivity analysis to make sure that only isolated edge information that represents noise gets filtered out, and the averaging process is within smooth regions not across edges, hence preserving the overall edge structure of the original image. Experimental results obtained from a suite of images but with different signal-noise-ratios (SNR), show that this method is robust to levels of noise and correctly preserves the edges, even for very extremely noisy condition.

Index Terms—Anisotropic filtering, connectivity analysis, multi-resolution, noise reduction.

I. INTRODUCTION

The impact of noise on image quality in scenes acquired under poor visibility conditions is quite significant. Additionally, the presence of noise also significantly restricts how the image can be used for computer vision and pattern recognition applications. For this reason, noise reduction in digital images has been an active topic of research in recent years. Several different approaches have been used for noise reduction, and they encompass a wide variety of processing methods. Linear filtering methods such as low-pass (smoothing) filter techniques assume that noise (mostly) is comprised of high spatial-frequency components in the spatial-frequency representation of an image. However, the reduction of noise using such techniques also often leads to the suppression of significant edge features. Since the presence of edges is perceptually related to sharpness and, hence, contrast [1], these approaches impact image quality significantly. There are smoothing methods that can also preserve edges, and Winkler *et al.* [2] provide a good synopsis of such methods.

Other researchers have used edge-preserving, non-linear filters like the median filter [3]. While such techniques are quite useful, they have significant shortcomings in the presence of a high level of noise due to the static nature of the filter extent. For this reason, several adaptive methods for noise reduction that preserve edges have also been developed

[4], [5]. Hamza and Kim [6] use robust estimation techniques to derive non-linear filters that can be used to denoise both impulse and Gaussian noise. Witkin [7] proposed scale-space filtering by Gaussian filters for noise reduction and feature preservation. To protect edges during smoothing, Perona and Malik [8] extended scale-space filtering by anisotropic diffusion. Donoho and Johnstone [9] developed a noise-reduction algorithm based on wavelet. Van De Ville *et al.* [10] use the idea of fuzzy derivatives and fuzzy smoothing to reduce the impact of noise.

More attention is being paid to edge feature analysis and, based on this, in trying to design new and effective noise reduction methods. Several researchers have shown the importance of using edge primitives as a basis for recognition in visual perception [11], [12]. This edge pattern analysis can be used for both automatic assessment of spatially variable noise and as a foundation for new noise reduction methods [13]. Edges can be divided into basic categories [14], [15] such as ramp, step, stair, and pulse: different types have different shapes. These edges can be filtered with a Gaussian to estimate their slope [16], [17]. Because noise can generally be assumed to be independent of signal, have little regional connectivity, and have random orientation, its estimate would be small under a Gaussian filter. Furthermore, it has been shown that the Gaussian is close to the optimal operator for different edges [18]. This kind of image analysis can be used as the basis of a promising solution for image denoising. If we divide an edge image into signal and noise pixels based upon local edge analysis, then we can apply different filters to the signal pixels and the noise pixels, reducing the overall impact of noise on image quality [19].

In this paper, we propose a new anisotropic noise reduction algorithm based on edge analysis to protect boundaries sharp and keep piecewise smoothing during filtering. The most common way of dealing with additive white Gaussian noise is to apply a low-pass filter to the noisy image. But, once the intensity image has been filtered with a Gaussian filter, the high frequency information-edges and noise-are attenuated together. Therefore, an important consideration is the scale of the Gaussian filter: small scales let more noise and edges through, while larger scales suppress both noise and edges. This trade-off between noise reduction and sharpness retention impacts post-processing tasks. To address this problem, multi-resolution edge analysis using a combination of Gaussian filters with different scales is applied to the edge detection process, which provides good edge-detection capabilities in the presence of different SNR.

After edge detection process, we attempts to preserve edges in the images by using connectivity analysis to classify edge or noise pixels. The idea of connectivity analysis relies on the characteristics of noise and features. Specifically, noise does

Manuscript received September 30, 2014; revised August 26, 2015. This work is supported by the Ministry of Finance “Life Science Instrumentation Development Program”, managed by Chinese Academy of Sciences with grant No. ZDYZ2012-3.

Bo Jiang is with the Guangzhou Institutes of Biomedicine and Health, Chinese Academy of Sciences, Guangzhou 510530, China (e-mail: jiang.bo@gibh.ac.cn).

not have regional connectivity and also has random orientation. Using the idea of regional connectivity, pixels on edges are classified as “noise”, which have length smaller than a given threshold, and replace them with an average of their neighbors, hence reducing the impact of noise at that location. Our new algorithm performs connectivity analysis on edge-data to make sure that only edge information unsatisfied with prescribed length that represents noise gets filtered out, hence preserving the overall edge structure of the original image. The basic steps of the new algorithm are as follows:

- 1) Find edges in the noisy image by multi-resolution edge analysis.
- 2) Use connectivity analysis to apply an anisotropic mean filter to regions around pixels classified as noise, and preserve pixels classified as edges.

Experimental results show that the proposed algorithm can improve image quality by removing noise in images corrupted by white Gaussian noise, even for very low SNR values.

The rest of the paper is organized as follows. The details of the new algorithm are given in Section II. Experimental analysis and results of combining the edge-detection analysis with the connectivity analysis are shown in section III. In section IV, the conclusions are presented.

II. ALGORITHM

A. Multi-resolution Edge Detection

The multi-resolution edge-detection algorithm, derived from our former algorithm [20], uses the idea of the difference-of-Gaussian operator in a different way. Instead of using the difference between two representations of an image obtained by filtering with a Gaussian of different scales (widths), the idea here is to use the logical AND operation on the image at one or more resolution to find the edge. This process can be described using the following steps:

- 1) Generate the multi-resolution image representations:

$$G_j(m, n) = G(m, n) * F_j(m, n), j = 1, \dots, 6, \quad (1)$$

$$F_j(m, n) = \exp\left(-\frac{m^2 + n^2}{\sigma_j^2}\right), \sigma_j^2 = 2^{j-1} \sigma_1^2, \quad (2)$$

where $G(m, n)$ are the pixels from the input image, G_j represents one image in the resolution pyramid of level j . F_j denotes the Gaussian filter with standard deviation of σ_j . σ_1 is the standard deviation of $F_1(m, n)$ and can be varied depending on the image, G .

- 2) Each image $G_j, j = 1, \dots, 6$ has associated modulus M_j and phase P_j images. M_j and P_j are computed as:

$$DG_j^x = G_j D_x, D_x = [-1 \ 1], \quad (3)$$

$$DG_j^y = G_j D_y, D_y = [-1 \ 1]^T, \quad (4)$$

$$M_j = \sqrt{(DG_j^x)^2 + (DG_j^y)^2}, \quad (5)$$

$$P_j = \tan^{-1}\left(\frac{DG_j^y}{DG_j^x}\right), \quad (6)$$

where $[\]^T$ indicates vector transposition.

- 3) The M_j and P_j images are used to obtain the edge using a top-down search algorithm. For a pixel to be classified as an edge, it must satisfy the condition:

$$(M_j(m, n) > \tau_m) \wedge (|P_j(m, n) - \Phi(m, n)| < \tau_p), \quad (7)$$

where \wedge is the logical AND operator, τ_m and τ_p are magnitude and phase thresholds, respectively, and $\Phi(m, n) = P_1(m, n)$. τ_m and τ_p are predefined global values. When Eq. 7 is satisfied, the pixel at (m, n) is classified as a potential edge candidate.

- 4) As pointed out by Clark [21], [22], the edges after Gaussian filtering that correspond with zero-crossings can be divided into two groups: the authentic edge and the phantom edge. To further accurately localize the edge, an edge candidate has to satisfy two other conditions:

- a) There is a zero-crossing point of second derivative between the edge candidate and its neighbor pixel in at least one orientation ϕ from four predefined orientations:

$$\begin{aligned} G_j''(m, n)G_j''(m, n+1) &< 0, & 0^\circ, \\ G_j''(m, n)G_j''(m+1, n-1) &< 0, & 45^\circ, \\ G_j''(m, n)G_j''(m+1, n) &< 0, & 90^\circ, \\ G_j''(m, n)G_j''(m+1, n+1) &< 0, & 135^\circ. \end{aligned} \quad (8)$$

- b) If the first condition is satisfied, then, the product of first derivative and third derivative has to be negative in the same orientation ϕ :

$$G_j'(m, n)G_j'''(m, n) < 0. \quad (9)$$

If all the conditions stated above are satisfied, then the pixel at (m, n) in the image would be judged as an edge. Once all the edge pixels have been found, an edge map is generated. This edge-map is then used with connectivity analysis to eliminate the edges due to noise.

Using images from all 6 resolutions typically results in very thick edges because of the heavy blurring associated with large values of j in Eq. 2. Experimental results led to the employment of a 3-of-6 rule in which we use a combination of 3 out of the 6 possible resolutions to generate the output image. Different results can be obtained depending upon which resolutions are used. Small scales give finer edges but are more prone to letting noise through. Larger scales eliminate the noise but produce thick edges. Once the three optimal resolutions have been determined, a second variation results in producing better edge results. In this variation, we accept a pixel to be an edge pixel if it complies with all conditions specified by equations stated above at that location in 2-of-3 resolutions.

B. Connectivity Analysis

For each non-zero pixel in the edge image, we perform

connectivity analysis which dictates that we search for edges of a prescribed length, EL , in the image. The rationale for this is that edges due to noise will usually be isolated and will form either singular points or very short edges in the edge image. This idea expands on the ideas presented by Rahman and Jobson [19] and Jiang and Rahman [20] by looking at larger connectivities to differentiate between pixels due to noise and those due to signal. As an example, consider the case when $EL = 2$. In this case the minimum length of a signal feature would be 2, and all isolated single point edge pixels would be eliminated. Similar procedures apply for $EL > 2$, where longer and longer edge features are classified as signal and shorter ones eliminated as noise. EL is application dependent, which is determined by defining what constitutes a feature edge. The algorithm can be described by the following steps:

- 1) Assume the pixel being processed is $p(i_0, j_0)$, where (i_0, j_0) is its coordinate in the image. Search the 3×3 neighborhood area. If another edge pixel is found, and $EL > 1$, go to Step 2. If an edge pixel is found but $EL = 1$, then, the pixel $p(i_0, j_0)$ would be classified as an edge. If there is no edge pixel, the pixel $p(i_0, j_0)$ would be classified as noise.
- 2) Move to the new 3×3 area, the center pixel of which is the edge pixel found in Step 1. Not all the pixels in the new 3×3 neighborhood would be examined because some of these pixels have already been tested for connectivity in the previous step, and there is a chance that the edge features would loop back unless such pixels are eliminated from the search space. For this reason only pixels in front of the current edge-pixel are tested (Fig. 1). This corresponds to examining only those pixels whose distance from the pixel being classified is larger than 1 in the Euclidean sense. The rule specified in Step 1 is used to classify the edge pixel as either noise or signal. In Fig. 1 'o' denotes classified edges, '?' denotes potential edge pixels, and 'x' denotes pixels that are not examined since they can cause the search to loop. Only the pixels marked as '?' are tested in each analysis step. If a connecting pixel is found, such as that shown in Fig. 1 (middle), the area of search is shifted to that pixel and centered on it. Using this new center, the new 3×3 search area is examined, except for the pixels marked with the 'x' symbol. This process is repeated until the requisite EL has been achieved. The requisite EL can be reiterated as the condition that should also be satisfied

$$|i_0 - i| \geq EL \vee |j_0 - j| \geq EL, \quad (10)$$

where \vee is the logical OR operator, (i_0, j_0) is the location of the pixel under consideration, and (i, j) is the current edge location. This condition places a limit on how short a Manhattan distance is allowed to satisfy the EL constraint. If the condition of the requisite EL is achieved, then, the pixel $p(i_0, j_0)$ would be classified as an edge. Otherwise, the pixel $p(i_0, j_0)$ would be classified as noise.

After connectivity analysis has been performed, pixels that are classified as non-edges are used to replace the pixel in the corresponding location in the original image by the average value of non-edges in their associative neighbor areas. We call this process an anisotropic filtering: local image variation

is measured at every non-edge, and its pixel value is averaged from neighborhoods whose size and shape depend on local variation. The default size of the neighborhood depends on the level of SNR and varies from 3×3 , 5×5 , to 7×7 . During smoothing, the intra-region smoothing should occur preferentially over inter-region smoothing. Therefore, only those pixels lying at the same region as the center pixel are used to compute the average value of the neighborhood. Otherwise, after smoothing, the pixels closing the boundaries would have mixed information from both intra-region and inter-region pixels.

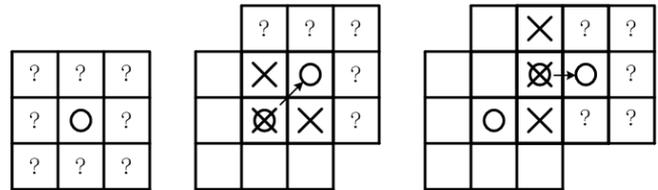


Fig. 1. Edge front: only pixels in front of the current pixel are examined for classification.

III. RESULTS

We compare the performance of the proposed algorithm with several other edge detectors and filters: the Sobel edge operator [3], the lateral-inhibition edge operator (LIH) [23], the Canny edge operator [18], the mean filter [3], the median filter [3], the adaptive median filter (AMF) [4], the adaptive threshold median filter (ATMF) [5], and the Bilateral filter [24]. For the edge detectors, they are used to detect edges instead of the multi-resolution edge detection algorithm before filtering. For the filters, they are simply applied to do full noise-reduction.

A. Experimental Evaluation

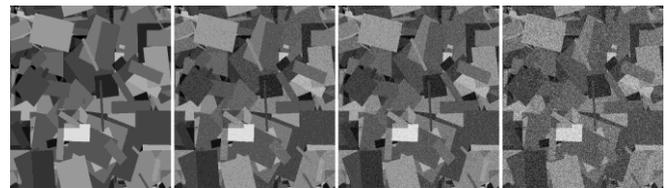


Fig. 2. (left) Original image, G ; (second) noisy image $G_{SNR} = 10$; (third) noisy image, $G_{SNR} = 5$; and (right) noisy image, $G_{SNR} = 1$.

We have tested our algorithm on color and grayscale images with root-mean-square (RMS) SNRs varying from 1 to 10. For color images, the algorithm is applied to each channel individually. The baseline images were computer generated so they are noise free, allowing us to control the $\{snr\}$ of the images precisely by adding white Gaussian noise of requisite amplitude to the data. This provides a controlled environment in which the performance of the algorithms can be evaluated as a function of the different system parameters which can be modified. Also, this allows us to compare the effectiveness of the noise-reduction algorithms since we can access the original noise-free image and compare the denoised directly with the original image.

Fig. 2 shows an original, noise-free, image G and three noisy images, $G_{SNR} = 10$, $G_{SNR} = 5$, and $G_{SNR} = 1$. In general, except when the SNR close to 1, traditional edge detection methods can detect the edges that have been impacted by

noise. Fig. 3 shows the multi-resolution images G_j for the original noise-free image, G , and Fig. 4 shows the multi-resolution images G_j for $G_{SNR} = 1$ ($\sigma_{noise} = 68$). The F_j (Eq. 2) used for these images were generated using ($\sigma_1 = 2$). This value of σ_j was deemed to be optimal after conducting a series of experiments based on different images, which are noise-free or corrupted by white Gaussian noise with various SNR from 1 to 10. This scale makes the resolution differences between each neighbor scales large enough to differentiate details and noise but not so overwhelming so as to lead to too much blurring.

As can be seen from Fig. 3 and Fig. 4, while noise suppression due to the Gaussian filters is small for $j = 1, 2, 3$, i.e., $\Delta(G) = |G_j - G_{SNR=1,j}|$ is large, $\Delta(G)$ is relatively small for $j = 4, 5, 6$, i.e., noise suppression is large. From our experiments we determined that it is not necessary to use all six images with different resolutions to obtain an edge image: in general, three resolutions are enough. The question, then, is: which three resolutions should be used? For the noise-free image or good SNR, i.e., SNR=10, since we do not want to attenuate too much original information, smaller scales of the Gaussian filters, i.e., $j=1, 2, 3$, would be used to create the three resolution images. The selection is not so obvious for the extremely noisy condition, such as SNR=1. We will discuss this low SNR condition in detail below.

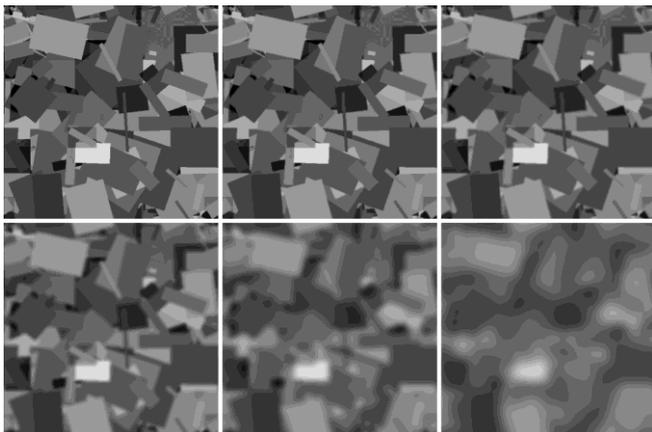


Fig. 3. Multi-resolution images: noise-free image and $\sigma_1 = 2$ (Eq. 2).

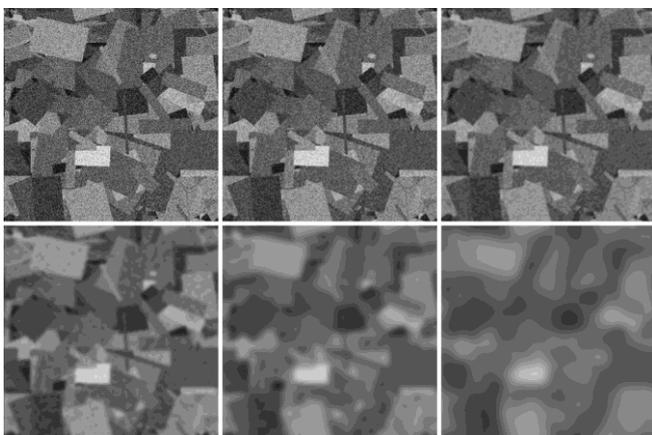


Fig. 4. Multi-resolution noisy images: SNR = 1 and $\sigma_1 = 2$ (Eq. 2).

From Fig. 4, we see that smaller σ_j keep the image sharp and noisy and vice versa. Using just the smallest or largest σ_j does not provide good performance. Experiments show that the larger σ_j , such as the one shown in Fig. 4 (bottom-right) lose

almost all the high-frequency information and hardly give any help for edge detection. In Fig. 5, different combinations of resolutions are shown. In each case, three neighboring scaled images were selected. We found that different combinations of resolutions required different values for τ_m . For every combination, increasing τ_m removes more noise but also loses more edges. However, the impact of τ_p was not critical, since conditions stated in Eq. 8 and Eq. 9 utilize all four orientations in searching.

Using the same σ_1 , another new scheme was introduced. A pixel at (m, n) was considered to be an edge pixel if it complies with all conditions specified in section IIA for two out of the three (2-of-3) resolutions, rather than for all three (3-of-3) resolutions (Fig. 5). Results for this scheme are shown in Fig. 6. While this new scheme leads to more connected edges, it also allows more noise pixels to be classified as edges. However, the visual impact is better than the 3-of-3 scheme because the edge structures are better preserved. The value for τ_m changes slightly for optimal results as that used in Fig. 5.

For the 3-of-3 and 2-of-3 schemes, using different Gaussian filters to suppress the noise would make the edges thicker as σ_j increases. Thinner edges can be produced only at the cost of classifying more edge pixels as noise pixels, especially for very low SNRs. We have determined experimentally that if SNR equals to or larger than 10, then the proposed algorithms can produce edges that are as thin as those produced by traditional algorithms.

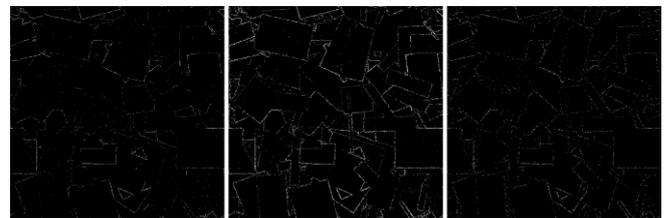


Fig. 5. The results of combining 3 different layers on an image (SNR = 1): (left) layers 1, 2, and 3; (center) layers 2, 3, and 4; and (right) layers 3, 4, and 5.

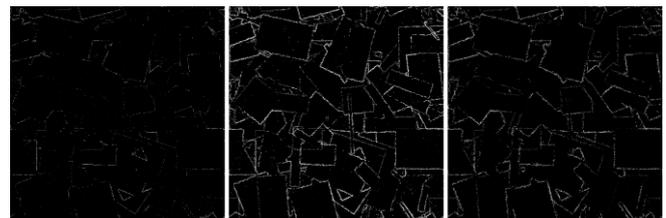


Fig. 6. The results of using the 2-of-3 different layers on an image (SNR = 1): (left) layers 2, 3, and 4; (center) layers 3, 4, and 5; and (right) layers 4, 5, and 6.

B. Comparisons

In Fig. 7, the results of edge detectors are shown. Both 3-of-3, and 2-of-3 methods produce similar results, though 2-of-3 classifies more pixels as edges, and is, hence, noisier. But, based on connectivity analysis used to differentiate noise or edges, a little more noise and edges would be better for the next processing step. The Sobel operator is badly affected by noise, as all noise causes an edge transition. LIH seems to lose connectivity since it is primarily a point detector. The Canny operator is good at suppressing noise and detecting edges because it has a filtering step. However, since it only

utilizes one scale Gaussian filter, a larger scale has to be employed for very low SNRs. Therefore, a number of small curves, distortions, and projections are found, which impact the denoising element of our proposed algorithm.

Fig. 8 shows the edge images obtained with the methods used to generate Fig. 7, but for the noise-free case. In the noise-free image, all the methods get excellent results, with few differences in performance. Both the 3-of-3 and the 2-of-3 methods give results similar to the SNR = 1 case ($\sigma_{\text{noise}} = 68$), but the edges for the 3-of-3 method are finer than those produced by the 2-of-3 method. This is because fewer pixels are eliminated using Eq. 7 for the 2-of-3 method since the comparison take place over fewer resolutions. The Sobel operator correctly finds the edges in the image. However, the edges it produces are thicker than those produced by other operators. LIH marks the location of the edges precisely and the edges are thin. The Canny operator also performs well and produces thin edges.

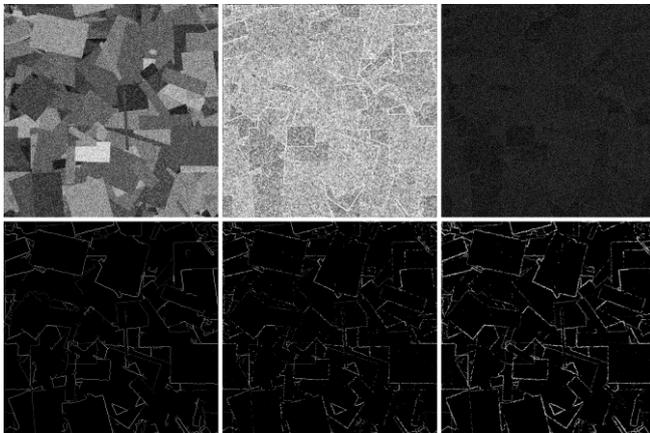


Fig. 7. Edge-detection on an image (SNR = 1): (top-row-left) noisy image; (top-row-center) Sobel; (top-row-right) LIH; (bottom-row-left) Canny; (bottom-row-center) 3-of-3 (EL=1); and (bottom-row-right) 2-of-3 (EL=1).

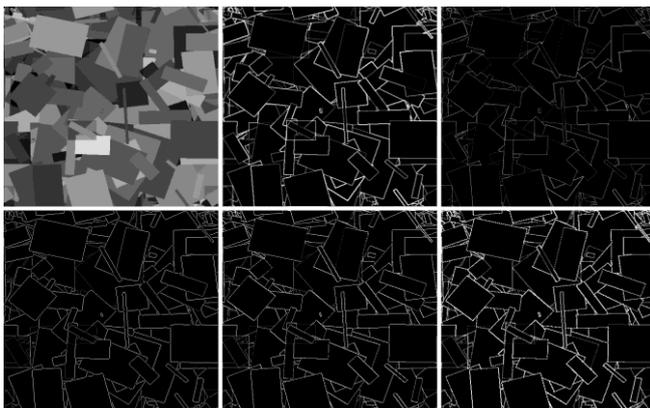


Fig. 8. Edge-detection on a noise-free image: (top-row-left) noise-free image; (top-row-center) Sobel; (top-row-right) LIH; (bottom-row-left) Canny; (bottom-row-center) 3-of-3 (EL=1); and (bottom-row-right) 2-of-3 (EL=1).

Fig. 9 shows the effect of the connectivity analysis. The original edge image is generated by using the 2-of-3 algorithm. The condition $EL = 1$ represents the case where all the edges are passed through, i.e., no connectivity analysis is performed. For $EL > 1$, the difference is obvious: as EL increases, the small edges appearing frequently in the center of blocks of smooth regions away from the authentic boundaries are increasingly removed. However, the algorithm takes longer to process the image since multiple

neighborhoods need to be examined at each pixel location. In our experiments, the size of the image is 512×512 . When $EL < 5$, the computation time is less than 1 second in C/C++ compiler on 2.40GHz Intel Core2 CPU. However, when EL larger than 5, the connectivity analysis has to takes more than 5 seconds. Thus, by help of connectivity analysis, the edges with prescribed lengths are classified as signal and kept, while those with lengths shorter than EL are classified as noise and eliminated using smoothing.

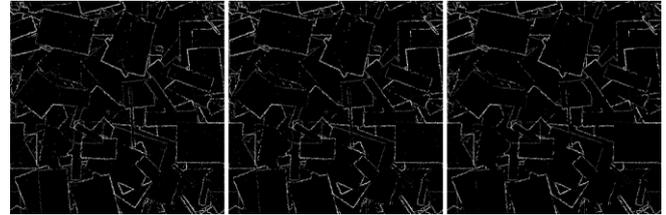


Fig. 9. The effect of connectivity analysis: (left) No connectivity analysis; (center) $EL = 3$; and (right) $EL = 5$.

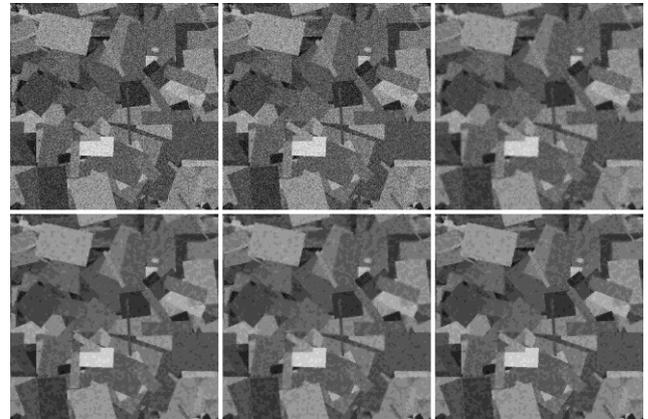


Fig. 10. Results of anisotropic filtering (7x7) with various edge detectors and connectivity analysis ($EL = 5$): (top-row-left) noisy image, SNR = 1 ($\sigma_{\text{noise}} = 68$); (top-row-center) Sobel; (top-row-right) LIH; (bottom-row-left) Canny; (bottom-row-center) 3-of-3; and (bottom-row-right) 2-of-3.

So far we have evaluated the impact of varying the number of resolutions and the parameters σ_1 , τ_m and τ_p on the performance of the edge detectors and the connectivity analysis. Recall that the overall noise reduction process relies on the edge-detection mechanism followed by noise suppression at those locations where the pixels are not classified as edge pixels. We also introduced the idea of edge connectivity analysis to determine which edge pixels were actually produced due to noise. To determine the effectiveness of the proposed methods, we performed two different group tests. In Fig. 10, the results of noise reduction using anisotropic filtering by help of edge detectors are shown. Fig. 11 shows the results of the noise reduction filters.

After edge detection, we can either apply:

- 1) A smoothing filter to every pixel that is not an edge pixel without performing edge connectivity analysis to determine if it were a noise pixel or not, or
- 2) Edge analysis with different edge length requirements and further reduce the total number of edge pixels by eliminating those that were classified as noise.

Experiments show that using connectivity analysis eliminates noise along the boundaries of regions with intensity transitions and produces an overall less noisy image. Increasing EL would, additionally, remove isolated small

edges, further reducing the noise along the boundaries of regions. Larger connectivity neighborhoods produce smoother results, yet, the impact on sharpness and contrast is minimal. The 3-of-3 and 2-of-3 methods are able to eliminate more noise than Sobel and LIH because of better edge detection and localization. The performance of the Canny operator is similar to those of the proposed methods.

Compared with our proposed methods, the traditional methods are less effective in very low SNR environments. The AMF does not produce good results for the additive white Gaussian noise. Similarly, the ATMF does not perform well even though it is good at reducing multi-layer impulse noise [5]. While the mean filter (7×7) produces good results in terms of eliminating visible noise distortion, it blurs the edges severely depending on the SNR-limiting the utility of the processed image for further operations. The overall brightness and contrast are reduced. The median filter (7×7) has better performance than the mean filter. But, since it utilizes a global filtering scheme, it impaired the magnitude of the overall edge structure. On the other hand, as seen in the result of the mean filter, the areas lying on the border of different objects are also blurred and mixed together in the result of the median filter. The bilateral filter performs good-quality edge-preserving smoothing by combining the geometric closeness and photometric similarity. However, due to very low SNR, the bilateral filter has to increase the distance of the geometric closeness and the range of the photometric similarity to balance the noise and the features. Therefore, it generates a cleaner image, but loses most high-frequency information.

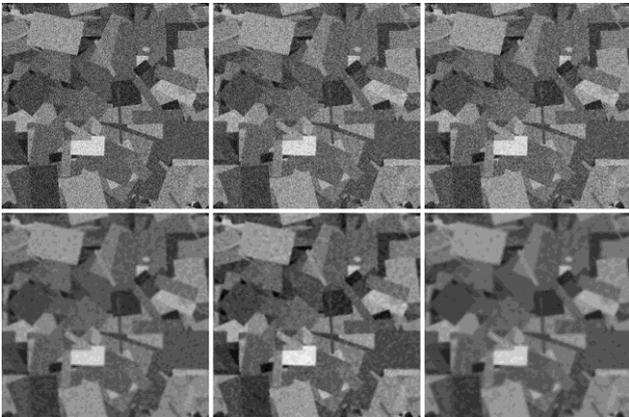


Fig. 11. (top-row-left) noisy image, SNR = 1, ($\sigma_{noise} = 68$); (top-row-center) AMF (7×7); (top-row-right) ATMF (7×7); (bottom-row-left) mean filter (7×7); (bottom-row-center) median filter (7×7); and (bottom-row-right) bilateral filter.

In all, the experimental results show that both the proposed methods and the Canny operator are more effective in preserving edges and removing noise. With the help of the connectivity analysis, both the proposed methods are better than the Canny operator. Furthermore, due to the anisotropic filtering directed by detected edges, another defect-the blurring and mixture in the boundary areas often happening in global filtering methods are largely reduced.

C. Error Metrics

Three of the error metrics are used to compare the restoration results, which are the root-mean-square error

(RMSE), the peak signal-to-noise ratio (PSNR), and the Fidelity. The RMSE is the cumulative root-squared error between two images I_1 and I_2 , where I_1 is the original noise-free image, and I_2 is variously produced by different image restoration methods. The mathematical formula of the RMSE is given by

$$RMSE = \sqrt{\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_1(m,n) - I_2(m,n))^2}, \quad (11)$$

The PSNR is a measure of the peak error by decibels (dB), which is defined as:

$$PSNR = 20 \log_{10} \left(\frac{\max |I_2(m,n)|}{RMSE} \right). \quad (12)$$

A commonly used metric of similarity between two images I_1 and I_2 is the Fidelity, $F(I_1, I_2)$ defined as

$$F(I_1, I_2) = 1 - \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_1(m,n) - I_2(m,n))^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_1(m,n)^2}, \quad (13)$$

where $F = 1$ when $I_1 = I_2$. The fidelity metric corresponds fairly closely with visual judgment for comparing images.

TABLE I: COMPARATIVE RESTORATION RESULTS BY EDGE DETECTORS IN RMSE AND PSNR

	Noisy Image (SNR=1)	Sobel + filter	LIH + filter	Canny + filter	3-of-3 + filter	2-of-3 + filter
RMSE	57.93	56.42	29.21	19.98	19.97	19.66
PSNR	12.87	13.10	18.82	22.10	22.12	22.26

The results of three error metrics are tabulated in Tables I, II, and III. We generated 40 noise-free images for both color and gray images. They were corrupted by Gaussian white noise with different SNR from 1 to 10. Then, we applied various filtering methods stated above to those images and examined their performance. Due to limit space, all Tables only shows average results generated from all tested images in the SNR = 1 case ($\sigma_{noise} = 68$), since traditional methods cannot work well in this extremely noisy condition.

A lower value for RMSE means lesser error, and as seen from the inverse relation between the RMSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of signal-to-noise is higher. Through using RMSE and PSNR, both proposed 3-of-3 and 2-of-3 algorithms own better performance than others.

Using F for assessing the performance, it is clear that the connectivity analysis method indeed improves noise reduction for LIH and our proposed methods. Furthermore, increasing the size of the connectivity neighborhood can improve F by as much as 7% for LIH. However, there is no obvious impact when the Sobel operator and the Canny operator are used. Also, F for AMF and ATMF are not good: both produce denoised images with a contrast that is poorer

than that of the original image. The mean filter (7×7) performs about as well, in terms of F , as the proposed noise reduction. But, since it utilizes a global filtering scheme, it impaired the magnitude of the overall edge structure. The overall brightness and contrast are reduced. The median filter (7×7) has better performance than the mean filter in term of F . In summary, the experimental results show that both the proposed methods and the Canny operator are more effective in preserving edges and removing noise.

TABLE II: COMPARATIVE RESTORATION RESULTS BY RESTORATION ALGORITHMS IN RMSE AND PSNR

	Noisy Image (SNR=1)	AMF 7×7	ATMF 7×7	Mean Filter 7×7	Median Filter 7×7	Bilateral Filter
RMSE	57.93	42.74	40.65	22.86	19.99	21.76
PSNR	12.87	15.52	15.95	20.60	22.10	20.85

TABLE III: FIDELITY IMPROVEMENT WITH NOISE REDUCTION

Noisy Image (SNR=1)	Fidelity		
	Without CA	With CA	
		EL = 3	EL = 5
Noisy Image (SNR=1)	0.775		
Sobel + Anisotropic filter	0.776	0.781	0.781
LIH + Anisotropic filter	0.866	0.923	0.934
Canny + Anisotropic filter	0.970	0.970	0.970
3-of-3 + Anisotropic filter	0.971	0.972	0.972
2-of-3 + Anisotropic filter	0.972	0.972	0.973
AMF 7×7	0.884		
ATMF 7×7	0.873		
Mean Filter 7×7	0.963		
Median Filter 7×7	0.970		
Bilateral Filter	0.966		

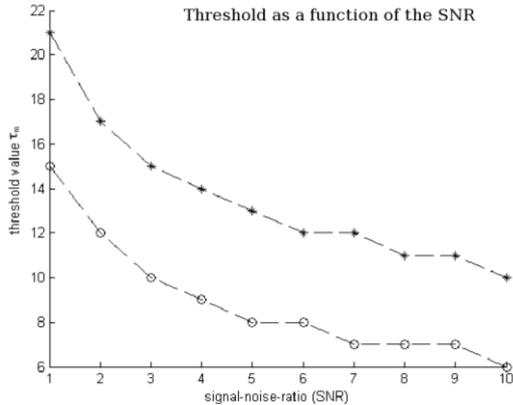


Fig. 12. Magnitude threshold τ_m as a function of the SNR: * color data; o grayscale data.

During our experiments we found that the value of the parameters in Section IIA needed to be changed depending upon the value of the SNR. From observation we see that to find a thinner edge σ_1 should be decreased and τ_m should be increased; otherwise, more edges and more noise would be produced. At the same time, due to small σ_1 , the residual noise would be higher and affect the edges, making it harder to differentiate between edges and noise. Under these conditions, though the edge might be thinner, the noise would be worse not just around the true edges but also spread across non-edge areas. However, since we do not use all the resolutions, reducing σ_1 is equivalent to using a different set of resolutions that correspond to the same σ_1 value, so the overall impact of changing σ_1 is in selecting different resolutions that do not impact overall performance. The same cannot be said,

however, about τ_m . Fig. 12 shows that the SNR and τ_m are inversely related: as SNR increases, τ_m should decrease. This regularity was found by statistical analysis across 400 different color and gray images corrupted by Gaussian white noise with different SNR from 1 to 10. After that, the proposed algorithm is applied to those images and examined the regularity of τ_m . This regularity makes intuitive sense since less noise in the image (higher SNR) corresponds to fewer false edges in the image corresponding to noise and, hence, does not require a larger threshold to eliminate such edges.

IV. CONCLUSION

In this paper, instead of using global noise-reduction mechanisms such as a smoothing filter, e.g., the mean filter, or the median filter, we made use of edge analysis and connectivity analysis both for noise assessment and as foundations for an anisotropic filtering algorithm. Starting with the premises that edges in the image should be preserved, and that the overall impact of noise is to reduce resolution, edge detection algorithm that is based on multi-resolution analysis and connectivity analysis is applied to reduce noise by robustly extracting authentic edges.

The significance of this method is to leave the resolution of all connected edges intact while reducing resolution in an area where noise is detected. Since the resolution of an image is reduced by the overall impact of noise, reducing resolution in noisy areas by smoothing within smooth regions eliminates the appearance of noise while not affecting overall image quality. Experimental results and the image comparison metrics show that this new anisotropic filtering method is effective even for very low SNR values.

Future work includes applying the proposed method in real images from diverse areas to assess its performance.

ACKNOWLEDGMENT

The author wishes to thank the Ministry of Finance “Life Science Instrumentation Development Program” managed by Chinese Academy of Sciences with grant No. ZDYZ2012-3 for the funding, which made this work possible.

REFERENCES

- [1] W. F. Schreiber and R. A. Haddad, “Fundamentals of electronic imaging systems,” *Applied Optics*, vol. 29, no. 19, 1990.
- [2] V. Aurich, G. Winkler, K. Hahn, A. Martin, and K. Rodenacker, “Noise reduction in images: Some recent edge-preserving methods,” *Pattern Recognition and Image Analysis*, vol. 9, no. 4, pp. 749-766, 1999.
- [3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall, 2008.
- [4] H. Hwang and R. A. Haddad, “Adaptive median filters: new algorithms and results,” *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499-502, 1995.
- [5] B. Jiang and W. Huang, “Adaptive threshold median filter for multiple-impulse noise,” *Journal of Electronic Science and Technology of China*, vol. 5, no. 1, pp. 70-74, 2007.
- [6] A. B. Hamza and H. Krim, “Image denoising: A nonlinear robust statistical approach,” *IEEE Transactions on Signal Processing*, vol. 49, no. 12, p. 3045, 2001.
- [7] A. Witkin, “Scale-space filtering: A new approach to multi-scale description,” in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, 1984, pp. 150-153.
- [8] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

- [9] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, 1994.
- [10] D. Ville, M. Nachtgael, D. Weken, E. Kerre, W. Philips, and I. Lemahieu, "Noise reduction by fuzzy image filtering," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 429-436, 2003.
- [11] D. Walters, "Selection of image primitives for general-purpose visual Processing," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 2, pp. 261-298, 1987.
- [12] D. J. Jobson, "Spatial vision processes: From the optical image to the symbolic structures of contour information," *Tech. Rep. 2838*, NASA, Washington, DC, November 1988.
- [13] D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "Automatic assessment and reduction of noise using edge pattern analysis in non-linear image enhancement," *Proceedings of SPIE*, vol. 5438, p. 141, 2004.
- [14] D. J. Williams and M. Shah, "Edge characterization using normalized edge detector," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 4, pp. 311-318, 1993.
- [15] J. R. Beltran, J. Garcia-Lucia, and J. Navarro, "Edge detection and classification using Mallat's wavelet," in *Proc. IEEE International Conference Image Processing*, pp. 293-297, 1994.
- [16] J. R. Beltran, F. Beltran, and A. Estopanan, "Multiresolution edge detection and classification: Noise characterization," in *Proc. International Conference on Systems, Man, and Cybernetics*, 1998, pp. 4476-4481.
- [17] G. Palacios and J. R. Beltran, "Cell nuclei segmentation combining multiresolution analysis, clustering methods and colour spaces," *Proceedings of Machine Vision and Image Processing*, pp. 91-97, 2007.
- [18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
- [19] Z. Rahman and D. J. Jobson, "Noise, edge extraction, and visibility of features," in *Proc. SPIE*, 2005, vol. 5817, p. 200.
- [20] B. Jiang and Z. Rahman, "Noise reduction using multi-resolution edge detection," in *Proc. SPIE*, 2009, vol. 7245, p. 724507.
- [21] J. J. Clark, "Authenticating edges produced by zero crossing algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 43-57, 1989.
- [22] B. Jiang, "Real-time multi-resolution edge detection with pattern analysis on graphics processing unit," *Journal of Real-Time Image Processing*, 2014.
- [23] F. O. Huck, C. L. Fales, and Z. Rahman, *Visual Communication: An Information Theory Approach*, Kluwer Academic Publishers, 1997.
- [24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE International Conference on Computer Vision*, 1998, pp. 839-846.



Bo Jiang received a B.S. degree in information engineering from Chengdu University of Technology, China, in 2004, a M.S. degree in communication and information engineering from the University of Electronic Science and Technology of China, China, in 2007, and a Ph.D. degree in electrical and computer engineering from Old Dominion University, USA, in 2010.

He is currently an associate investigator of the Division of Applied Stem Cell Automation at Guangzhou Institutes of Biomedicine and Health, Chinese Academy of Sciences, Guangzhou, Guangdong, China.

His research interests include bioinformatics, generic pattern recognition, and real-time, non-linear image processing.