IMAGE RESTORATION USING A PDE-BASED APPROACH

E. Nadernejad, H. Hassanpour* and H. MiarNaimi

Department of Computer and Electrical Engineering, Noushirvani Institute of Technology
University of Mazandaran, P.O. Box 47144, Babol, Iran
h_miare@nit.ac.ir

*Corresponding Author

(**Received: May 13, 2007 – Accepted in Revised Form: November 22, 2007**)

Abstract Image restoration is an essential preprocessing step for many image analysis applications. In any image restoration techniques, keeping structure of the image unchanged is very important. Such structure in an image often corresponds to the region discontinuities and edges. The techniques based on partial differential equations, such as the heat equations, are receiving considerable attention in image restoration. In this paper, a new algorithm is proposed for image restoration using partial differential equations applied on neighbors longer than one pixel. To better preserve edges in this technique, slant edges are considered in addition to vertical and horizontal edges. A number of experiments have been preformed to evaluate performance of the proposed method and to compare its performance with the existing algorithms, wiener and median filters. The results express the considerable superiority of the proposed methods.

Keywords Digital Image Processing, Denoising, Edge, Partial Differential Equations

چکیده بازیابی و یا کاهش نویز تصویر یکی از مراحل اساسی در بسیاری از کاربردهای پردازش تصویر می باشد. در همه تکنیک های بازسازی تصویر حفظ ساختار تصویر، نظیر ناپیوستگی ها و لبه های موجود در آن بسیار حائز اهمیت می باشد. برای بازسازی تصاویر، اخیراً محققین توجه ویژه ای به روش های مبتنی بر معادلات با مشتقات جزیی مانند معادلات گرما نموده اند. در این مقاله الگوریتم جدیدی برای بازیابی تصاویر با استفاده از معادلات با مشتقات جزیی بر روی همسایگی با عمق بیش از یک پیکسل ارائه شده است. در این روش برای حفظ بهتر لبه ها علاوه بر لبه های افقی و عمودی، لبه های اریب نیز در نظر گرفته شده است. برای نشان دادن کارایی الگوریتم پیشنهادی و مقایسه آن با روش های موجود مانند فیلترهای وینر و میانه آزمایشات مختلفی انجام شده است. نتایج آزمایشات نشان می دهد که روش پیشنهادی برتری قابل توجهی دارد.

1. INTRODUCTION

Real images are often cluttered with noise. Noise reduction is usually the first process used in analysing digital images. In any image denoising application, it is very important that the deniosing process has no blurring effect on the image, and has no changes or relocation on image edges. In medical imaging, like X-ray imaging, the images often have noise which may prevent recognition of significant patterns such as a fracture[1,2].

In the literature, there are various methods for image denoising. Using simple filters, such as average filters, median filters and Gaussian filters, are some of the techniques employed for image denoising [3]. These filters reduce noise with the cost of smoothing the image and hence softening the edges. To overcome this problem, some other alternative methods have been presented in the literature. In [4-6], a denoising process is performed by thresholding the wavelet coefficients. In [7-9] partial differential equations (PDEs) are used as an alternate method for image denoising. These methods assume intensity of illumination on edges varies like geometric heat flow in which heat transforms from a warm environment to a cooler one until the temperature of the two environments reach to a balanced point. It was shown that these changes are in the form of Gaussian function [10]. As a result, sudden changes in edges might be due to the existance of noise. In fact, an image includes a

series of regions in which different regions might have different standard deviation. This issue is considered as a diffusion equation [11]. The diffusion equations offer a strong tool for image denoising.

The existing diffusion-based methods suggest the use of side neighbors [12]. This investigation shows that these methods may lead to smoothing the edges. In this paper, by considering longer side neighboring at different directions around a pixel, the problem of edge smoothing is considerably reduced [21]. In addition, results of this paper show that the proposed method has a better performance compared to the existing methods in image denoising.

This paper is organized as follows. Section 2 briefly describes exiting methods for image restoration. Then the proposed technique is described in Section 3. Several image quality measures are addressed in Section 4 to compare performance of the proposed approach with performance of the exist methods. Implementation and experimental results are provided in Section 5. The final section is the conclusion.

2. EXISTING IMAGE DENOISING TECHNIQUES

2.1. Median Filter The median filter is one of the most effective filters used for image denoising [25,26]. In this filter square windows of size $(2k+1)\times(2k+1)$ or in a cross form, are used and the central pixel is replaced with the median of the window pixels. Here the inner values of gray levels of each window are sorted then the middle value is the median:

$$g(m,n)=\text{median}\begin{pmatrix} f(m-k,n-k) & \cdots & f(m-k,m+k) \\ \vdots & f(m,n) & \vdots \\ f(m+k,n-k) & \cdots & f(m+k,n+k) \end{pmatrix} \quad (1)$$

One of the advantages of the median filter is its high capability in reducing impulse noise, but it is not true for Gaussian noise. The other advantage of this filter is that it does not generate a new gray level value as it may occur in mean filtering [1]. But, its main drawback is the relocation of the image edges as large as one or two pixels. In this approach, the level of denoising is improved by considering a larger window size, but it may cause more relocation on the image edges [3].

2.2. Wiener Filter Wiener filter-base approaches have a statistical view to image processing. In this filter it is assumed that image and noise have Gaussian distribution with zero mean. The Wiener filter aims to reduce the means square error, between the original image ($I_{orginal}$) and the approximated image (denoised image) ($I_{denoised}$):

$$|\varepsilon|^2=\left\|I_{orginal}-I_{denoised}\right\|^2 \quad (2)$$

The Wiener filter has a high capability in reducing Gaussian noises. As mentioned before this filter aims to minimize the total mean square error, therefore it may not consider the local features. Since the noise effect on the human vision system is dependent to the texture of the noisy region, so the steady luminous or higher luminous regions are reconstructed more efficiently than the other regions. Also if the damage in different regions of the images has different forms, it isn't possible to reconstruct the image by the Wiener filter [3].

2.3. Nonlinear Diffusion Filter Nonlinear diffusion filtering was introduced by Perona and Malik [11,13]. Although their approach in its original formulation has a deficiency, the genesis of the algorithm has encouraged many researchers to improve the technique [7,9,12]. In any diffusion-based approaches the variation of illumination on images is considered as geometric heat. As a result, a sudden change in an image is considered as a noise. In following subsections, initially, the basic idea of using diffusion equations for image nosing is briefly introduced. Then a numerical solution for the diffusion equation is outlined.

2.3.1. The basic idea The diffusion equation is based on repetition. The main idea of using diffusion equations in image processing is the use of a two-dimensional Gaussian filter in which the

image I(x,y) is convolved with a window $K_\sigma(x,y)$:

$$K_\sigma(x,y)=\frac{1}{2\pi\sigma^2}\exp(-\frac{|x|^2+|y|^2}{2\sigma^2}) \quad (3)$$

where $K_\sigma(x,y)$ is a Gaussian filter, and σ represents the standard deviation of the filter's coefficients. By convolution operation, there would be the problem of edge smoothing. It has been previously shown that this problem can be resolved by treating intensity variations in an image as diffusion of the heat flow [13].

The diffusion equation for an image $I(x,y)$ would be as follow:

$$\frac{\partial I(x,y,t)}{\partial t}=\nabla^2 I(x,y,t)=\frac{\partial^2 I(x,y,t)}{\partial x^2}+\frac{\partial^2 I(x,y,t)}{\partial y^2} \quad (4)$$

where $I(x,y;0)=I_0(x,y)$ as initial image (t = 0), and I(x,y,t) is the image at $t = 0.5\sigma^2$.

The above equation can be rewritten as below:

$$\frac{\partial I(x,y,t)}{\partial t}=\nabla.(c(x,y,t)\nabla(x,y,t))\,I(x,y,0)=I_0(x,y) \quad (5)$$

Where ∇ is the gradient operator, c(x,y,t) is the diffusion factor, and ∇ is the divergence operator. If c has a constant value (independent to x,y,t), the obtained equation is called diffusion equation with isotropic diffusion factor. In this case, all points, even edges would be smoothed as there is no difference between a pixel on edge and other pixels. It is obvious that this is not an ideal condition. For resolving this deficiency, the diffusion factor could be considered as a function of x and y. Hence, the above equation is changed to a linear and anisotropic equation. If c is dependent to the image, the linear equation would be transformed to a nonlinear equation. This is the idea that was suggested in [11,13]. In these researches two different equations for the diffusion factor were suggested as below:

$$c(x,y,t)=1/(1+\frac{|\nabla I|^2}{k^2}) \quad (6)$$

$$c(x,y,t)=\exp(-|\nabla I|^2/2k^2) \quad (7)$$

In these equations the diffusion factor c changes at different point in the image. In those points that the gradient of the image is large, this factor has a small value (see Figure 1).

Consequently, the diffusion factor would be small around the edges. In (4) and (5) k is used to control the diffusion factor.

The major drawback of the above-mentioned method is that the gradient is computed from the noisy image. Hence, the place of edges in the image may not be clearly recognized [11]. In order to overcome this problem, it was suggested to find the gradient from the smoothed image [14], which means:

$$\nabla I_G=\nabla(G_\sigma * I(x,y,t)) \quad (8)$$

where G_σ is a Gaussian filter and * is the convolution operator.

There are a number of methods for solving diffusion equations. In any method, there are different parameters need to be adjusted. One of the most important parameters is to determining the diffusion factor. For the diffusion factor, in addition to 4 and 5, the following equations are also suggested [20].

$$c(x,y,t)=(1+\frac{|\nabla I_G|^2}{k^2})^{-1/2} \quad (9)$$

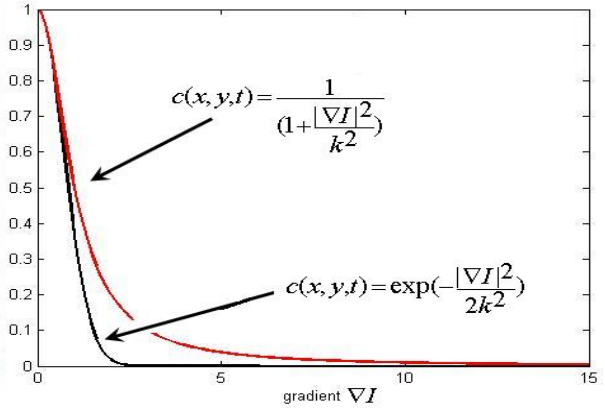

Figure 1. The diffusion factor introduced by Prona and Malik [11,13].

$$c(x,y,t)=\begin{cases}1 & \text{if } \left|\nabla I_G\right|=0\\ 1-\exp(-\dfrac{c_m}{(\nabla I_G^2/k^2)^m}) & \text{if } \left|\nabla I_G\right|>0\end{cases} \quad (10)$$

In 8, the parameter c_m is achieved from the solution of equation $\exp(-c_m)(1+2mc_m)=1$ [20]. The value of this parameter for m = 2,3,4 will be 2.33666, 2.9183, 3.31488, respectively. The edges are better retained for the larger value of m.

2.3.2. Solution for the diffusion equation As mentioned before, there are a number of methods for solving diffusion equations. In [15], the first and second derivatives and also the Laplacian of a current pixel are computed using the pixels in neighbors. Here, the most general numerical approach for a solution of these equations is presented [16,17]. To state the solution, by considering 6, Equation 4 is rewritten as:

$$\frac{\partial I(x,y,t)}{\partial t}=C(x,y,t)(\frac{\partial^2 I(x,y,t)}{\partial x^2}+\frac{\partial^2 I(x,y,t)}{\partial y^2})+ \frac{\partial I(x,y,t)}{\partial x}\times\frac{\partial C(x,y,t)}{\partial x}+\frac{\partial I(x,y,t)}{\partial y}\times\frac{\partial C(x,y,t)}{\partial y} \quad (11)$$

The approximate solution of 9 is:

$$I(x,y,t+\Delta t)=I(x,y,t)+ \Delta t(d_n c_n+d_s c_s+d_e c_e+d_w c_w), \quad (12)$$

where the parameters are defined as below:

$$d_n=I(x,y-1,t)-I(x,y,t) \qquad c_n=1/(1+(d_n/k)^2$$
$$d_s=I(x,y+1,t)-I(x,y,t) \qquad c_n=1/(1+(d_s/k)^2$$
$$d_e=I(x+1,y,t)-I(x,y,t) \qquad c_e=1/(1+(d_e/k)^2$$
$$d_w=I(x-1,y,t)-I(x,y,t) \qquad c_w=1/(1+(d_w/k)^2 \quad (13)$$

These parameters are the numerical equivalent for the derivation and diffusion factors for the four cardinal directions.

In 10, the goal is to find a solution for the initial times before the heat flow. It is obvious that the temperature at different points changes toward balancing. Therefore, in the case of a gray level image, the value of pixels converges which leads to denoising. It should be noted that for an area containing an edge, a larger diffusion factor is chosen to preserve the edge.

Considering 11, one can claim that in 10 the rate of denoising is low for pixels with a large differentiation (d is large). In other words, for a large gradient value, which belongs to the edge, smoothing is not taking place. In fact, this is the way used in this technique to preserve edges. It is not difficult to find out that in 10 only the vertical and horizontal edges are considered. In addition, sudden changes are considered as edges.

3. THE PROPOSE METHOD

In order to eliminate the above restrictive hypothesis, a new method is proposed. The main idea in the proposed method is to change neighboring structure for preserving slant edges, in addition to vertical and horizontal edges, and also it assumes smooth changing for the pixel value on edges. For this reason the neighboring structure represented in Figure 2 is suggested.

In the proposed method the diffusion Equation 10 is updated as below:

$$I(x,y,t+\Delta t)=I(x,y,t)+\Delta t(d_n c_n+d_s c_s+d_e c_e+ d_w c_w+(d_{ne}c_{ne}+d_{se}c_{se}+d_{nw}c_{nw}+d_{sw}c_{sw}+ d_{nl}c_{sl}+d_{el}c_l+d_{el}c_{el}+d_{wl}c_{wl})\alpha+(d_{nel}c_{nel}+ d_{sel}c_{sel}+d_{nwl}c_{nwl}+d_{swl}c_{swl})\,\alpha/2 \quad (14)$$

Where α is a factor representing the significance (plentitude) of slant edges in the image. In this equation, the new parameters ($d_{ne}, d_{se}, \ldots$) are computed similar to the parameter in (11), (see Appendix I for the solution).

In Equation 14 the role of those neighbors that are closer to the center is more efficient compared to the other neighbors, hence the $\alpha/2$ is applied to

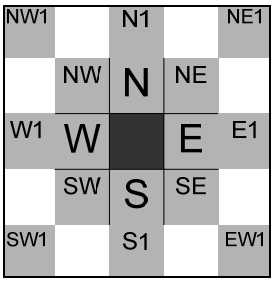

Figure 2. The neighboring structure for the numerical solution of diffusion equation in the proposed method.

the second part of the equation. It should be noted that in this equation the diffusion factors are updated in each iteration.

Results in this research indicate that the proposed technique, compared to the other existing methods, has a better performance in image denoising and preserving edges. In addition, this algorithm compared to other existing PDE-based methods, is converged with a lower number of iteration. Figure 3 shows the figure of merit (FOM) value in each iteration of using the proposed and the existing methods on the image of Lena. As can be seen, the proposed method is converged after 50 iterations whilst the other existing method is converged at 200 iterations. In using these techniques, the stop criterion is as follows:

$$\left|FOM(i_{th}) - FOM((i+1)_{th})\right| < \text{error} \quad \text{and} \quad FOM((i+1)_{th}) > FOM(i_{th}) \tag{15}$$

Results show that the proposed technique has also a lower blocking effect or artifacts on the enhanced image (see Figure 4).

In this paper, $\Delta = 0.25$ has been chosen, as it was suggested in [13]. To finding the optimum value for a and k, different experiments have been performed on different images (some of them were included in Appendix II). The results indicate that the optimum performance of the algorithm is achieved by choosing $0 < a < 1$, $5 < k < 100$.

In any PDE-based method the number of algorithm iterations are crucial. In other words, extra iteration may cause the pixel value to exceed the gray level value, relocating the edges, and blurring the image.

4. PERFORMANCE MEASURES

Four criteria were used in the image processing techniques, to assess the performance of the proposed method, and compare it with the performance of other existing methods. These criteria are briefly introduced below.

4.1. Figure of Merit The figure of merit is the edge preserving measure that is defined as below [22]:

$$FOM = \frac{1}{\max\left\{\hat{N}, N_{ideal}\right\}} \sum_{i=1}^{\hat{N}} \frac{1}{1+d_i^2\lambda} \tag{16}$$

In this equation $\hat{N}$ and N_{ideal} are the numbers of detected and original edge pixels, d_i is the Euclidean distance between the i^{th} detected edge pixel and the nearest original edge pixel, and λ is a constant typically set to 1/9 [23]. The dynamic range of FOM is between 0 and 1. FOM close to 1 indicates the edges were properly preserved. To find edges in all assessed techniques the Canny edge detector was used [3].

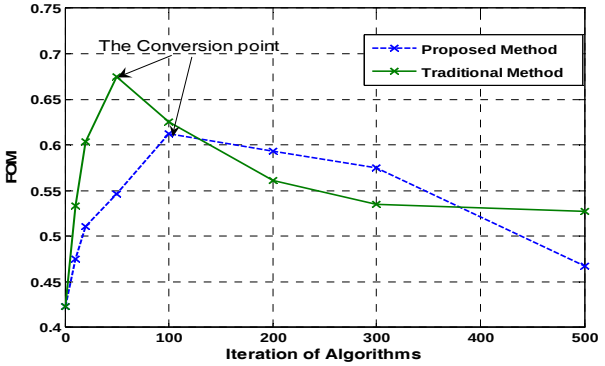

Figure 3. The FOM value in each iteration of using the proposed and the existing methods on Lena image.

(a) (b) (c)

Figure 4. The blocking effect of PDE-based method: (a) original image, (b) existing PDE-based method and (c) proposed method.

4.2. Structural Similarity The Structural Similarity (SSIM) factor can be used to measure the similarity between two images [24]. This factor consists of three different metrics. Let $x = \{ x_i \mid i = 1,2,3,...,N \}$ and $y = \{ y_i \mid i = 1,2,3,...,N \}$ be the original and the test images, respectively. The SSIM is defined as:

$$SSIM = Q = \frac{4\sigma_{XY}\overline{X}\,\overline{Y}}{(\sigma_X^2 + \sigma_Y^2)[(\overline{X})^2 + (\overline{Y})^2]} \tag{17}$$

where:

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i \ , \overline{Y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$
$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \overline{X})^2 \ , \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N} (y_i - \overline{Y})^2,$$
$$\sigma_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \overline{X})(y_i - \overline{Y}) \tag{18}$$

The dynamic range of SIMM is [-1,1]. The best value 1 is achieved if and only if $y_i = x_i$ for all i = 1,2,3,…,N. The lowest value of -1 occurs when $y_i = 2\overline{X} - x_i$ for all i = 1,2,3,…,N. The SIMM indicates any distortion as a combination of three different factors: loss of correlation, luminance distortion, and contrast distortion. In other words, Q in 17 can be rewritten as a product of three components:

$$SSIM = Q = Q_1Q_2Q_3 = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \times \frac{2\overline{X}\,\overline{Y}}{(\overline{X})^2 + (\overline{Y})^2} \times \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} = s(x,y)^{\alpha} \times l(x,y)^{\beta} \times c(x,y)^{\gamma} \tag{19}$$

The first component is the correlation coefficient between x and y, which represents the degree of linear correlation between x and y, and its dynamic range is between -1 and 1. The best value 1 is obtained when $y_i = ax_i + b$ for all i = 1,2,…,N, where a and b are constants and a > 0. Even if x and y are linearly related, there still might be relative distortions between them, which are evaluated in the second and third components. The second component, with a value range of [0,1], measures how much the x and y are close in luminance. It equals 1 if and only if $\overline{X} = \overline{Y}$. σ_X and σ_y can be considered as an estimate of the contrast in x and y. Eventually, the third component indicates how similar the contrasts of the images are. Its values range between 0 and 1, where the best value 1 is achieved if and only if $\sigma_X = \sigma_y$.

Parameters α, β, γ are used to adjust the significance of each of the three components.

In practice to use the above measure, the image is windowed equally, then for each window the SSIM is computed to find the average SSIM as follows:

$$\overline{\text{SSIM}}(X,Y)=\frac{1}{M}\sum_{j=1}^{M}\text{SSIM}(x_j,y_j) \quad (20)$$

where X and Y are the original and the denoised images respectively, M is the number of local windows in the image, x_j and y_j are the image contents at the j^{th} local window.

4.3. Mean Square Error The mean square error (MSE) is a criterion frequently used in signal and image processing, which is defined as below:

$$\text{MSE}=\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_{\text{orginal}}(i,j)-I_{\text{denoised}}(i,j))^2 \quad (21)$$

where $I_{original}$ is the original image and $I_{denoised}$ is the denoised image. In using this measure, the smaller the value of MSE, the better the denoising algorithm.

4.4. Signal to Noise Ratio The signal to noise ratio (SNR) is a well known measure which is defined as follows:

$$(\text{SNR})_{db}=10\log_{10}\left(\frac{\sigma^2_{\text{denised}}}{\sigma^2_{\text{orginal}}}\right) \quad (22)$$

Where $\sigma^2_{orginal}$ and $\sigma^2_{denised}$ are the standard deviations for the original and the denoised images. In using this measure, the larger the SNR value, the closer will be the denoised image to original one.

5. EXPERIMENTAL RESULTS

To evaluate the performance of the above-described algorithms, they all have been implemented using Matlab, and then applied on a data set of 100 standard images taken from references [18,19]. In these experiments, to make the images noisy the Gaussian noise and impulse noise have been used with 64 as the average value for both of the noises, and variance 400 for the Gaussian noise, respectively. These noises have been added to the images, once separately and then together. The results of the experiments are shown in Figure 5. These results have been extracted using the 20 images represented in [18,19].

As can be seen in the table, those algorithms that are based on heat diffusion equation, especially the proposed algorithm, are functioning better than the other methods. In addition, in terms of preserving the edge, once the depth of the neighbor used in diffusion equations increased, such as the one in the proposed neighboring, the edge of images are better preserved. As can be seen from Figure 5, the SSIM measure indicates that the proposed approach has often a better performance in preserving the structure of original images.

These results indicate that the proposed image denoising technique has a better performance compared to the other existing approach. To judge the act of this algorithm from the human eye's view, the results of applying different techniques on two of those images are shown in Figures 6 and 7.

6. CONCLUSIONS

In this paper, a new algorithm has been proposed for image denoising using a diffusion equation. In this method, the diffusion equation has been used in a way that it can consider slant edges in addition to vertical and horizontal edges. In other existing techniques changes in edges were assumed sudden and sharp. It has been indicated that this assumption may not be true for real images. The results of applying the proposed techniques on twenty different images show that it has a better performance in image denoising and preserving edges, compared to other existing approaches.

7. APPENDIX I

7.1. The Approximate Solution to the Diffusion Equation, and the Proposed Method There are a number of methods in

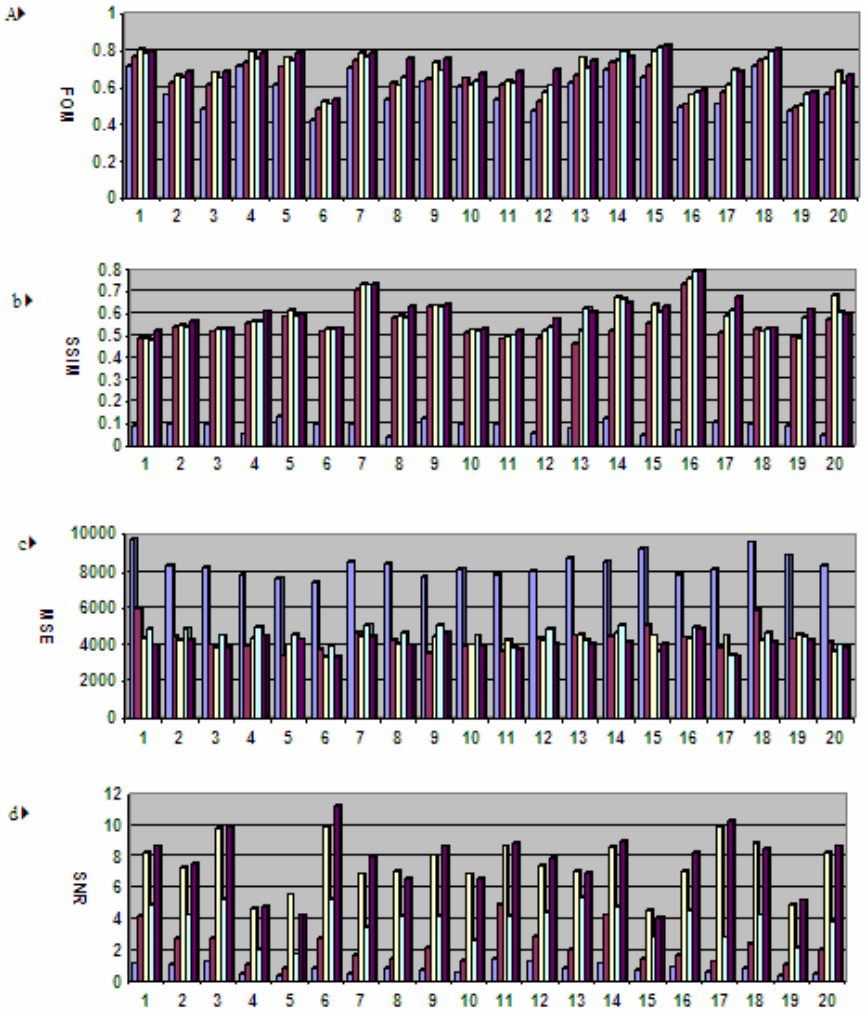

Figure 5. Results of measuring performance of the compared algorithms on 20 different images shown in Appendix I using: (a) FOM, (b) SSIM, (c) MSE and (d) SNR.

literature for solving diffusion equations. A numerical approach can be used for the application of digital image denoising. The proposed approach in [27] to the diffusion equation computes the differential and Laplacian of a pixel using the neighboring pixels. Here the general solution of the anisotropic diffusion equation has been described:

$$\frac{\partial I(x,y;t)}{\partial t} = \text{div}[c(x,y;t)\nabla I(x,y;t)] \qquad (I1)$$

where $I(x,y;t)$ and $c(x,y;t)$ are the image under analysis and the diffusion coefficient, and ∇ is the gradient operator. Equation I1 is equivalent to

$$\text{div}[c(x,y;t)\nabla I(x,y;t)] = \frac{\partial}{\partial x}(c(x,y;t)\times \frac{\partial}{\partial x}I(x,y;t)) + \frac{\partial}{\partial y}(c(x,y;t)\times\frac{\partial}{\partial y}I(x,y;t)) \qquad (I2)$$

The above equation can be implemented using a finite difference method. Consider the difference approximation of :

$$\frac{\partial}{\partial x}(c(x,y;t)\times\frac{\partial}{\partial x}I(x,y;t)) \qquad (I3)$$

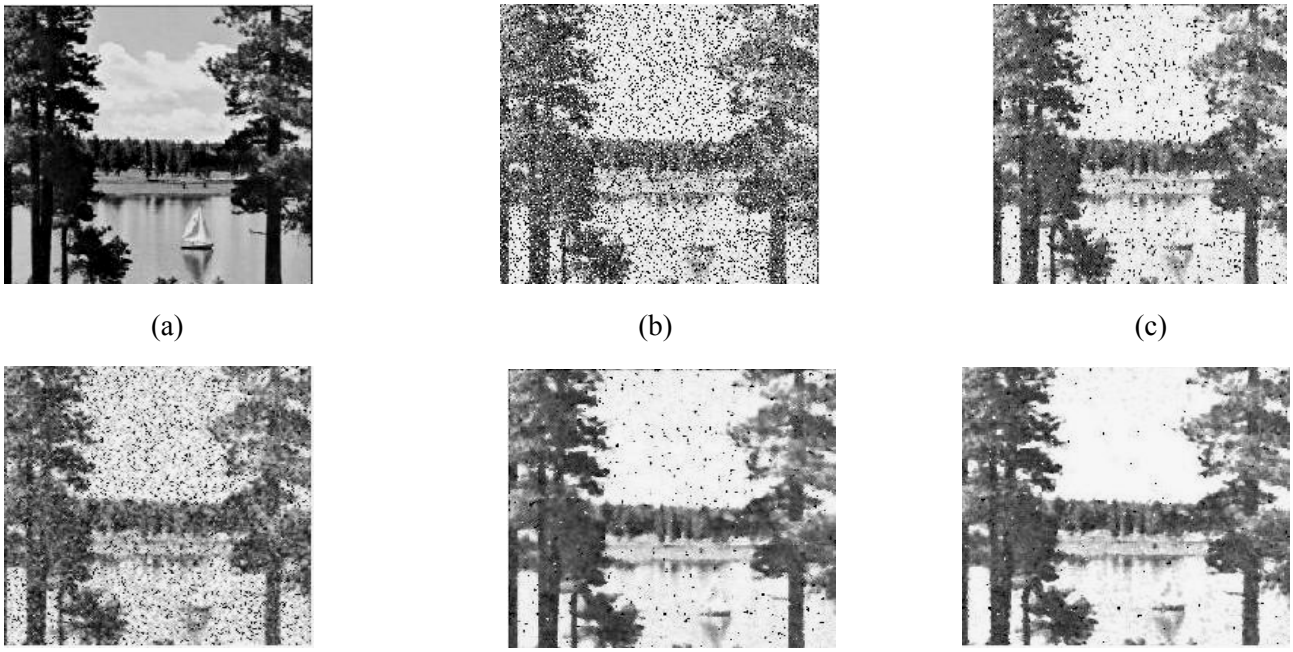

Figure 6. Output of different methods on image No. 7 shown in Appendix II:
(a) original image (b) noisy image; output of: (c) the existing diffusion-based method,
(d) Wiener filter, (e) Median filter and (f) the proposed method.

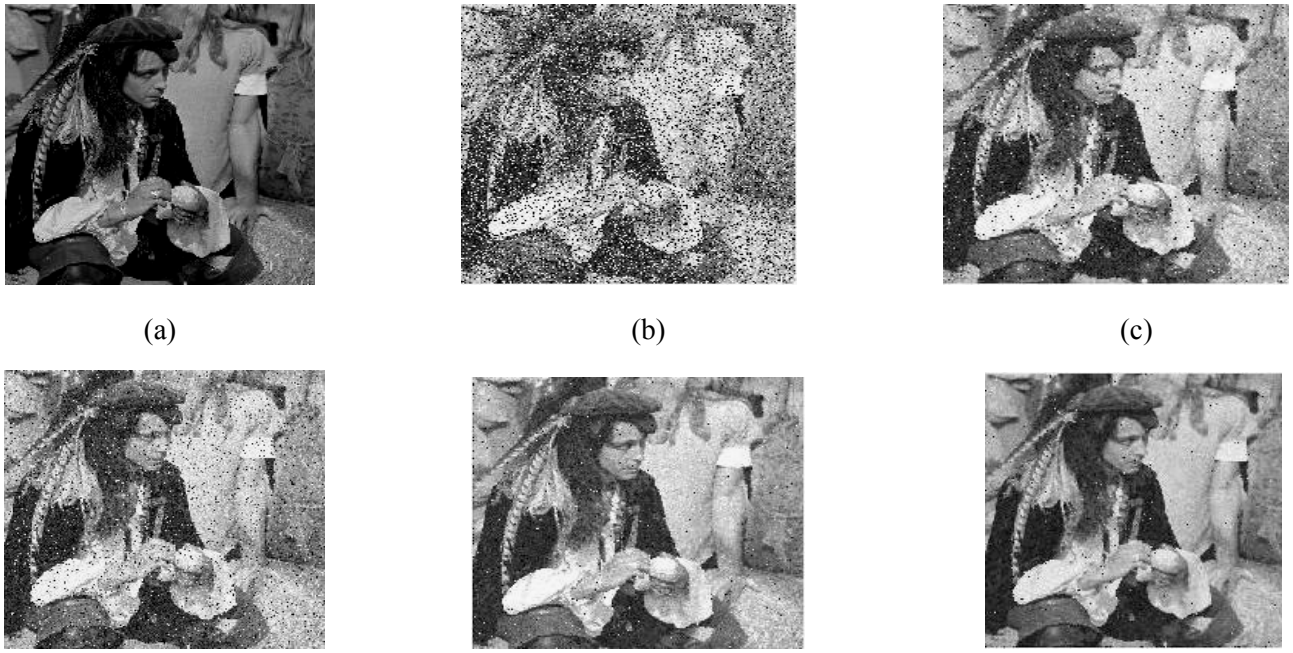

Figure 7. Output of different methods on image No. 14 shown in Appendix II:
(a) original image (b) noisy image; output of: (c) the existing diffusion-based method,
(d) Wiener filter, (e) Median filter and (f) the proposed method.

which is expressed as:

$$\approx \frac{\partial}{\partial x}\left\{c(x,y;t)\frac{1}{\Delta x}[I(x+\Delta x,y;t)-I(x,y;t)]\right\}\approx \frac{1}{\Delta x^2}$$
$$\{c(x,y;t)[\,I(x+\Delta x,y;\,t)-I(x,y;t)-I(x,y;t)+$$
$$I(x-\Delta x,y;t)]+[c(x+\Delta x,y;t)-c(x,y)]$$
$$[I(x+\Delta x,y;t)-I(x,y;t)]\,\}\equiv\frac{1}{\Delta x^2}\{c(x+\Delta x,y;t)$$
$$[I(x+\Delta x,y;t)-I(x,y;t)]+c(x,y;t)[I(x-\Delta x,y;t)$$
$$-I(x,y;t)]\}=\frac{1}{\Delta x^2}\{c(x+\Delta x,y;t)[I(x+\Delta x,y;t)-$$
$$I(x,y;t)]+c(x,y;t)[I(x-\Delta x,y;t)-I(x,y;t)]\} \qquad (I4)$$

Similarly, there is

$$\frac{\partial}{\partial y}\left(c(x,y;t)\times\frac{\partial}{\partial y}I(x,y;t)\right)$$
$$\approx\frac{1}{\Delta y^2}\{c(x,y+\Delta y;t)[I(x,y+\Delta y;t)- \qquad (I5)$$
$$I(x,y;t)]+c(x,y;t)[I(x,y-\Delta y;t)-I(x,y;t)]\}$$

this approach can be followed to obtain the other differential operators.

By inserting Equations I4, I5 into I2 and letting $\Delta x=1,\ \Delta y=1$, the difference approximation of $\delta I(x,y;t)/\delta t$ can be obtained which is expressed as

$$\frac{\partial I(x,y;t)}{\partial t}\approx c(x+1,y;t)[I(x+1,y;t)-I(x,y;t)]+$$
$$c(x-1,y;t)[I(x-1,y;t)-I(x,y;t)]+c(x,y+1;t)$$
$$[I(x,y+1;t)-I(x,y;t)]+c(x,y-1;t)[I(x,y-1;t)-$$
$$I(x,y;t)]+\alpha\{c(x+1,y+1;t)[I(x+1,y+1;t)-I(x,y;t)]$$
$$+c(x+1,y-1;t)[I(x+1,y-1;t)-I(x,y;t)]+$$
$$c(x-1,y+1;t)[I(x-1,y+1;t)-I(x,y;t)]+$$
$$c(x-1,y-1;t)[I(x-1,y-1;t)-I(x,y;t)]+$$
$$c(x+2,y;t)[I(x+2,y;t)-I(x,y;t)]+c(x-2,y;t)$$
$$[I(x-2,y;t)-I(x,y;t)]+c(x,y+2;t)[I(x,y+2;t)$$
$$-I(x,y;t)]+c(x,y-2;t)[I(x,y-2;t)-I(x,y;t)]\}+$$
$$\frac{\alpha}{2}\{c(x+2,y+2;t)[I(x+2,y+2;t)-I(x,y;t)]+$$
$$c(x+2,y-2;t)[I(x+2,y-2;t)-I(x,y;t)]+$$
$$c(x-2,y+2;t)[I(x-2,y+2;t)-I(x,y;t)]+$$
$$c(x-2,y;t)[I(x-2,y;t)-I(x,y;t)]\} \qquad (I6)$$

Thus, the discrete realization of anisotropic diffusion for image can be obtained from Equation I6 as follows

$$I(x,y;t+\Delta t)=I(x,y;t)+\Delta t[c(x+1,y;t)[I(x+1,y;t)$$
$$-I(x,y)]+c(x-1,y;t)[I(x-1,y;t)-I(x,y)]+$$
$$c(x,y+1;t)[I(x,y+1;t)-I(x,y)]+c(x,y-1;t)$$
$$[I(x,y-1;t)-I(x,y)]+\alpha\times\{c(x+1,y+1;t)$$
$$[I(x+1,y+1;t)-I(x,y)]+c(x+1,y-1;t)$$
$$[I(x+1,y-1;t)-I(x,y)]+c(x-1,y+1;t)$$
$$[I(x-1,y+1;t)-I\,(x,y)]+c(x-1,y-1;t)$$
$$[I(x-1,y-1;t)-I(x,y)]+c(x+2,y;t)$$
$$[I(x+2,y;t)-I(x,y)]+c(x-2,y;t)[I(x-2,y;t)$$
$$-I(x,y)]\}+\frac{\alpha}{2}\{c(x+2,y+2;t)[I(x+2,y+2;t)-$$
$$I(x,y)]+c(x+2,y-2;t)[I(x+2,y-2;t)-I(x,y)]$$
$$+c(x-2,y+2;t)[I(x-2,y+2;t)-I(x,y)]+$$
$$c(x-2,y-2;t)[I(x-2,y-2;t)-I(x,y)]] \qquad (I7)$$

From Equation I7, it is found that the key problem in anisotropic diffusion is the choice of c(x,y;t). Similar to 2D anisotropic diffusion, c(x,y;t) can be chosen as follows:

$$c(x,y;t)=\exp(-[\frac{\nabla I(x,y;t)}{k}]^2),$$
or (I8)
$$c(x,y;t)=\frac{1}{1+[\frac{\nabla I(x,y;t)}{k}]^2}$$

Therefore it can be written for two dimensional image I(x,y;t):

$$\partial I(x,y;t)/\partial t=\mathrm{div}[c(x,y;t)\nabla I(x,y;t)]$$
$$I(x,y,0=I_0(x,y),(\partial I(x,y;t)/\partial\vec{k})\big|\partial\Theta=0 \qquad (I9)$$

Where $\partial\Theta$ indicates the border of Θ, $\vec{k}$ is the outer normal to the $\partial\Theta$ and the diffusion coefficient c(x,y;t) is given by I8.

7.2. Numerical Implementation of the Algorithm The proposed algorithm Equation I9 can be solved numerically. Let the time step be Δt and the spatial step be h in x, y directions, then the

time and space coordinates can be discretized as:

$t = n\Delta$, $n = 0,1,2,\ldots$;

$x = ih$, $y = jh$,

$i = 1,2,3,\ldots M-1$,

$j = 0,1,2,\ldots N-1$,

where $Mh \times Nh$ is the size of the image. Let $I^n_{i,j} = I(ih, jh, n\Delta t)$ then the final image can be obtained using the four-stage approach described below:

7.2.1. Stage I Computing the derivative approximations and the Laplacian approximation

$$\nabla^2 I^n_{i,j} = (I^n_{(i+1,j)} + I^n_{(i-1,j)} + I^n_{(i,j+1)} + I^n_{(i,j-1)} - 4 \times I^n_{(i,j)}) \alpha [I^n_{(i+1,j+1)} + I^n_{(i+1,j-1)} + I^n_{(i-1,j+1)} + I^n_{(i-1,j-1)} + I^n_{(i+2,j)} + I^n_{(i-2,j)} + I^n_{(i,j+2)} + I^n_{(i,j-2)}] - 8 \times \alpha \times I^n_{(i,j)} + \frac{\alpha}{2}[I^n_{(i+3,j+3)} + I^n_{(i+3,j-3)} + I^n_{(i-31,j+3)} + I^n_{(i-3,j-3)}] - 4 \times \alpha \times I^n_{(i,j)}) / h^2 \quad \text{(I10)}$$

The symmetric boundary conditions are used

$$I^n_{-1,j} = I^n_{0,j}, I^n_{M,j} = I^n_{M-1,j}, j = 0,1,2,\ldots,N-1,$$
$$I^n_{i,-1} = I^n_{i,0}, I^n_{i,N} = I^n_{i,N-1}, i = 0,1,2,\ldots,M-1, \quad \text{(I11)}$$

7.2.2. Stage II Computing the diffusion coefficient c(x,y;t) for example:

$$c^n_{i,j} = \exp(-[\frac{\nabla^2 I^n_{i,j}}{k^2}]) \quad \text{(I12)}$$

The other diffusion coefficients can be obtained similarly.

7.2.3. Stage III Computing the divergence of $c(.)\nabla I$

$$d^n_{i,j} = \frac{1}{h^2}[c^n_{i+1,j}(I^n_{i+1,y} - I^n_{x,y}) + c^n_{i-1,j}(I^n_{i-1,y} - I^n_{x,y}) + c^n_{i,j+1}(I^n_{i,y+1} - I^n_{x,y}) + c^n_{i,j-1}(I^n_{i,y-1} - I^n_{x,y})] + \frac{\alpha}{h^2}[c^n_{i+1,j+1}(I^n_{i+1,y+1} - I^n_{x,y}) + c^n_{i+1,j-1}(I^n_{i+1,y-1} - I^n_{x,y}) + c^n_{i-1,j+1}(I^n_{i-1,y+1} - I^n_{x,y}) + c^n_{i-1,j-1}(I^n_{i-1,y-1} - I^n_{x,y}) + c^n_{i+2,j}(I^n_{i,+2,j} - I^n_{x,y}) + c^n_{i,j+2}(I^n_{i,y+2} - I^n_{x,y}) + c^n_{i-2,j}(I^n_{i-2,j} - I^n_{x,y}) + c^n_{i,j-21}(I^n_{i,y-2} - I^n_{x,y})] + \frac{\alpha}{2h^2}[c^n_{i+3,j-3}(I^n_{i+3,y-3} - I^n_{x,y}) + c^n_{i+3,j+3}(I^n_{i+3,y+3} - I^n_{x,y}) + c^n_{i-3,j-3}(I^n_{i-3,y-3} - I^n_{x,y}) + c^n_{i-3,j+3}(I^n_{i-3,y+3} - I^n_{x,y})] \quad \text{(I13)}$$

With symmetric boundary conditions:

$$d^n_{-1,j} = d^n_{0,j}, d^n_{M,j} = d^n_{M-1,j}, j = 0,1,2,\ldots,N-1,$$
$$d^n_{i,-1} = d^n_{i,0}, d^n_{i,N} = d^n_{i,N-1}, i = 0,1,2,\ldots,M-1, \quad \text{(I14)}$$

7.2.4. Stage IV The numerical approximation to the differential equation is given by

$$I^{n+1}_{i,j} = I^n_{i,j} + \frac{\Delta t}{16} d^n_{i,j} \quad \text{(I15)}$$

This equation is equivalent to 13 (in this paper: $\frac{\Delta t}{16} \approx 0.25$).

8. REFERENCES

1. Zhong, J., Ning, R. and Conover, D., "Image denoising based on multiscale singularity detection for cone beam CT breast imaging", ***IEEE Trans Med Imaging***., Vol, 23, (2004), 696-703.

2. Mielikäinen, T. and Ravantti, J., "Sinogram Denoising of Cryo-Electron Microscopy Images", ***International Conference on Computational Science and its Applications***, Singapore, (2005), 9-12.
3. Gonzalez, R. C. and Woods, R. E., "Digital Image Processing", Prentice Hall, (2004).
4. Fodor, I. K. and Kamath, C., "On denoising images using wavelet-based statistical techniques", Technical report UCRL-JC-142357, Lawrence Livermore National Laboratory, (2001).
5. Fodor, I. K. and Kamath, C., "Denoising through wavelet shrinkage: An empirical study", ***Journal of Electronic Imaging***, Vol. 12, (2003), 151-160.
6. Zhai, J. and Zhang, S., "Image denoising via wavelet threshold: single wavelet and multiple wavelets transform", ***Int. Conf. on Mach. Learning and Cyb***., Vol. 5, (2005), 3232-3236.
7. Xiaogang, D., and Pollak, I., "Multiscale segmentation with vector-valued nonlinear diffusions on arbitrary graphs", ***IEEE Transactions on Image Processing***, Vol. 15, (2006), 1993-2005.
8. Lee, S. H., Kang, M. G. and Park, T. K., "CCD noise filtering based on 3-dimensional nonlinear partial differential equation", ***IEEE Transactions on Consumer Electronics***, Vol. 44, (1998), 1086-1090.
9. Pollak, I., Willsky, A. S., and Krim, H., "Image segmentation and edge enhancement with stabilized inverse diffusion equations", ***IEEE Transactions on Image Processing***, Vol. 9, (2000), 256-266.
10. Weickert, J., Romeny, B. M. and Viergever, M., "Efficient and reliable schemes for nonlinear diffusion filtering", ***IEEE Trans. on Image Processing***, Vol. 7, (1998), 398-410.
11. Perona, P. and Malik, J., "Scale-space and edge detection using anisotropic diffusion", ***IEEE Computer Society Workshop on Computer Vision***, (1987), 16-27.
12. Gilboa, G., Sochen, N. and Zeevi, Y. Y., "Image enhancement and denoising by complex diffusion processes", ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, Vol. 26, (2004), 1020-1036.
13. Perona, P., and Malik, J. "Scale-space and edge detection using anisotropic diffusion***" IEEE Trans. on Pattern Analysis and Machine Intel***., Vol. 127, (1990), 629-639.
14. Weeratunga, S. K. and Kamath, C., "PDE-based non-linear diffusion techniques for denoising scientific/industrial images: An empirical study", ***Proceedings, Image Processing: Algorithms and Systems, SPIE Electronic Imaging***, San Jose, (January 2002), 279-290.
15. Handloviová, A. and Mikula, K. and Sgallari, F., "Variational Numerical Methods for Solving Nonlinear Diffusion Equations Arising in Image Processing", ***Journal of Visual Communication and Image Representation***, Vol. 13, (2002), 217-237.
16. Weeratunga, S. K. and Kamath, C., "A comparison of PDE-based non-linear anisotropic diffusion techniques for image denoising", SPIE Electronic Imaging, (2003).
17. Krivá, Z. and Mikula, K., "An Adaptive Finite Volume Scheme for Solving Nonlinear Diffusion Equations in Image Processing", ***Journal of Visual Communication and Image Representation***, Vol. 13, (2002), 22-35.
18. Medical Information Finder, http://www.netmedicine.com., retrieved on, (October 2006).
19. Gonzalez, R. C. and Woods, R. E., "Image Databases, http://www.imageprocessingplace.com", retrieved on, (October 2006).
20. Weickert, J. and Scharr, H., "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance", ***Journal of Visual Communication and Image Representation***, Vol. 13, (2002), 103-118.
21. Hassanpour, H., Nadernejad, E. and Miar, H., "Image Enhancement Using Diffusion Equations", ***International Symposium on Signal Processing and its Applications***, Sharjah, UAE, (2007).
22. Pratt, W. K., "Digital Image Processing", Wiley, New York, NY, USA, (1978).
23. Yang, Z. and Fox, M. D., "Speckle Reduction and Structure Enhancement by Multichannel Median Boosted Anisotropic Diffusion", ***EURASIP Journal on Applied Signal Processing***, Vol. 16, (2004), 2492-2502.
24. Wang, Z. and Bovik, A. C., "A universal image quality index", ***IEEE Signal Processing letters***, Vol. 9, (2002), 81-84.
25. Senel, H. G., Peteres, R. A. and Dawant, B., "Topological median filters", ***Image Processing, IEEE Transactions***, Vol. 11, (2002), 89-104.
26. Koschan, A. F. and Abidi, M. A., "A Comparison of Median Filter Techniques for Noise Removal in Color Images", ***Proc. 7th German Workshop on Color Image Processing***, Erlangen, Germany, (October 2001), 69-79.
27. Fisch, B. and Schowart, E. L., "learning an Integral Equation Approximation to Nonlinear Anisotropic Diffusion in Image processing", ***IEEE Trans on Pattern Analysis and Machine Intelligence***, Vol. 19, No. 4, (1997), 342-352.