A Comparison of Some State of the Art Image Denoising Methods

Hae Jong Seo, Priyam Chatterjee, Hiroyuki Takeda, and Peyman Milanfar
Department of Electrical Engineering, University of California at Santa Cruz
{rokaf,priyam,htakeda,milanfar}@soe.ucsc.edu

***Abstract*— We briefly describe and compare some recent advances in image denoising. In particular, we discuss three leading denoising algorithms, and describe their similarities and differences in terms of both structure and performance. Following a summary of each of these methods, several examples with various images corrupted with simulated and real noise of different strengths are presented. With the help of these experiments, we are able to identify the strengths and weaknesses of these state of the art methods, as well as seek the way ahead towards a definitive solution to the long-standing problem of image denoising.**

I. Introduction

Denoising has been an important and long-standing problem in image processing for many decades. In the last few years, however, several strong contenders have emerged which produce stunning results across a wide range of image types, and for varied noise distributions, and strengths. The emergence of multiple very successful methods in a relatively short period of time is in itself interesting, in part because it points to the possibility that we may be approaching the limits of performance for this problem. At the same time, it is interesting to note that these methods share an underlying likeness in terms of their structure, which is based on nonlinear weighted averages of pixels, where the weights are computed from metric similarity of pixels, or neighborhoods of pixels. The said weights are computed by giving higher relevance to "nearby" pixels which are more spatially, and tonally similar to a given reference patch of interest. In this sense, as we will see below, they are all based on the idea of using a *kernel* function which controls the level of influence of similar and/or nearby pixels.

Overall, two important problems present themselves. First, what are the fundamental performance bounds in image denoising, and how close are we to them? And second, what makes these "kernel-based" methods so successful, can they be improved upon, and how? While we do not intend to address either of these questions in this paper, we do take a modest step in exposing the similarities, strengths, and weaknesses of these competing methods, paving the way for the resolution of the more fundamental questions in future work.

This work was supported in part by AFOSR Grant F49620-03-1-0387.

II. Nonparametric Kernel-Based Methods

In this section, we give descriptions of three algorithms. We discuss "Data-adaptive Kernel Regression" of Takeda *et al.* [1], "Non-local Means" of Buades *et al.* [2], and "Optimal Spatial Adaptation" of Kervrann, *et al.* [3].

A. Data-Adaptive Kernel Regression

The kernel regression framework defines its data model in 2-D as

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \cdots, P, \quad \mathbf{x}_i = [x_{1i}, x_{2i}]^T, \quad (1)$$

where y_i is a noisy sample at $\mathbf{x}_i$, $z(\cdot)$ is *the* (hitherto unspecified) *regression function* to be estimated, ε_i is an i.i.d zero mean noise, and P is the total number of samples in a neighborhood (window) of interest.

While the specific form of $z(\cdot)$ may remain unspecified, we can rely on a generic local expansion of the function about a sampling point $\mathbf{x}_i$. Specifically, if $\mathbf{x}$ is near the sample at $\mathbf{x}_i$, we have the N-th order Taylor series

$$\begin{aligned} z(\mathbf{x}_i) &\approx z(\mathbf{x}) + \{\nabla z(\mathbf{x})\}^T(\mathbf{x}_i - \mathbf{x}) \\ &\quad + \frac{1}{2}(\mathbf{x}_i - \mathbf{x})^T\{\mathcal{H} z(\mathbf{x})\}(\mathbf{x}_i - \mathbf{x}) + \cdots \quad (2) \\ &= \beta_0 + \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) + \boldsymbol{\beta}_2^T \operatorname{vech}\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\} + \cdots \quad (3) \end{aligned}$$

where ∇ and $\mathcal{H}$ are the gradient (2×1) and Hessian (2×2) operators, respectively, and $\operatorname{vech}(\cdot)$ is the half-vectorization operator which lexicographically orders the lower triangular portion of a symmetric matrix. Furthermore, β_0 is $z(\mathbf{x})$, which is the pixel value of interest.

Since this approach is based on *local* approximations and we wish to preserve image detail as much as possible, a logical step to take is to estimate the parameters $\{\boldsymbol{\beta}_n\}_{n=0}^N$ from all the samples $\{y_i\}_{i=1}^P$ while giving the nearby samples higher weights than samples farther away in spatial and radiometric terms. A formulation of the fitting problem capturing this idea is to solve the following optimization problem,

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_n\}_{n=0}^N} \sum_{i=1}^{P} \Big| & y_i - \beta_0 - \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) \\ & - \boldsymbol{\beta}_2^T \operatorname{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \cdots \Big|^q \\ & \cdot K_{\text{adapt}}(\mathbf{x}_i - \mathbf{x}, y_i - y) \quad (4) \end{aligned}$$

where q is the error norm parameter ($q = 2$ or 1 typically), N is the regression order ($N = 2$ typically), and $K_{\text{adapt}}(\mathbf{x}_i - \mathbf{x}, y_i - y)$ is the data-adaptive kernel function. Takeda *et al.* introduced steering kernel functions in [1]. This data-adaptive kernel is defined as

$$K_{\text{steer}}(\mathbf{x}_i - \mathbf{x}, y_i - y) = K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}), \tag{5}$$

where $\mathbf{H}_i$ is the (2×2) *steering matrix*, which contains four parameters. One is a global smoothing parameter which controls the smoothness of an entire resulting image. The other three are the scaling, elongation, and orientation angle parameters which capture local image structures. We estimate those three parameters by applying singular value decomposition (SVD) to a collection of estimated gradient vectors in a neighborhood around every sampling position of interest. With the steering matrix, the kernel contour is able to elongate along the local image orientation. In order to further enhance the performance of this methods, we apply orientation estimation followed by steering regression repeatedly to the outcome of the previous step. We call the overall process *iterative steering kernel regression* (ISKR).

Returning to the optimization problem (4), the minimization eventually provides a point-wise estimator of the regression function. For instance, for the zeroth regression order ($N = 0$) and $q = 2$, we have the estimator in the general form of:

$$\hat{z}(\mathbf{x}) = \frac{\sum_{i=1}^{P} K_{\text{adapt}}(\mathbf{x}_i - \mathbf{x}, y_i - y)\, y_i}{\sum_{i=1}^{P} K_{\text{adapt}}(\mathbf{x}_i - \mathbf{x}, y_i - y)}. \tag{6}$$

B. Non-Local Means

The Non-Local Means (NLM) method of denoising was introduced by Buades *et al.* [2] where the authors use a weighted averaging scheme to perform image denoising. They make use of the fact that in natural images a lot of structural similarities are present in different parts of the image. The authors argue that in the presence of uncorrelated zero mean Gaussian noise, these repetitive structures can be used to perform image restoration.

The estimator of the non-local means method is expressed as

$$\hat{z}_{\mathcal{NL}}(\mathbf{x}_j) = \frac{\sum_{i \neq j} K_{h_s,h_r}(\mathbf{y}_{w_j} - \mathbf{y}_{w_i})\, y_i}{\sum_{i \neq j} K_{h_s,h_r}(\mathbf{y}_{w_j} - \mathbf{y}_{w_i})}, \tag{7}$$

where $\mathbf{y}_{w_i}$ is a column-stacked vector that contains the given data in a patch w_i (the center of w_i being at $\mathbf{x}_i$):

$$\mathbf{y}_{w_i} = [\cdots\ y_\ell\ \cdots]^T, \quad y_\ell \in w_i. \tag{8}$$

The kernel function is defined as the weighted Gaussian kernel:

$$K_{h_s,h_r}(\mathbf{y}_{w_j} - \mathbf{y}_{w_i}) = \exp\left\{ -\frac{\|\mathbf{y}_{w_j} - \mathbf{y}_{w_i}\|^2_{\mathbf{W}_{h_s}}}{h^2} \right\}, \tag{9}$$

where the weight matrix $\mathbf{W}_{h_s}$ is given by

$$\mathbf{W}_{h_s} = \text{diag}\left\{ \cdots, K_{h_s}(\mathbf{x}_{j-1} - \mathbf{x}_j), K_{h_s}(\mathbf{0}), K_{h_s}(\mathbf{x}_{j+1} - \mathbf{x}_j), \cdots \right\}, \tag{10}$$

h_s and h_r are the parameters which control the degree of filtering by regulating the sensitivity to the neighborhood dissimilarities[1], and K_{h_s} is defined as the Gaussian kernel function. This essentially implies that the restored signal at position $\mathbf{x}_j$ is a linear combination (weighted mean) of all those given data which exhibit a largely similar (Gaussian-weighted) neighborhood.

The method, as presented in theory, results in an extremely slow implementation due to the fact that a neighborhood around a pixel is compared to every other pixel neighborhood in the image in order to calculate the contributing weights. Thus for an image of size $M \times M$, the algorithm runs in $O(M^4)$. Such drawbacks have been addressed in some recent publications improving on the execution speed of the non-local means method [4], [5], while modestly compromising on the quality of the output.

In summary, the implementation of the work boils down to the pseudocode described in algorithm 1.

Algorithm 1 Non-Local Means algorithm

```
y ⇐ Noisy Image
z ⇐ Output Image
h_r, h_s ⇐ Filtering parameters

for every pixel y_j ∈ y do
    w_j ⇐ patch with y_j at the center
    W_j ⇐ search window for w_j
    for every w_i ∈ W_j and i ≠ j do
        K(i) ⇐ exp{ -||y_{w_i} - y_{w_y}||^2_{W_{h_s}} / h_r^2 }
        ẑ(x_j) ⇐ ẑ(x_j) + K(i) * y_i
    end for
    z(x_j) ⇐ ẑ(x_j) / Σ_i K(i)
end for
```

C. Optimal Spatial Adaptation

While the related NLM method is controlled by smoothing parameters h_r, h_s calibrated by hand, the method of Kervrann *et al.* [3] called "optimal spatial adaptation (OSA)" improves upon Non-Local Means method by adaptively choosing a local window size. The key idea behind this method is to iteratively grow the size of a local search window W_i starting with a small size at each pixel and to stop the iteration at an "optimal" window size. The dimensions of the search window grow as $(2^\ell+1) \times (2^\ell+1)$ where ℓ is the number of iterations.

To be more specific, suppose that $\hat{z}^{(0)}(\mathbf{x}_i)$ and $\hat{v}_i^{(0)}$ are the initial estimate of the pixel value and the local noise variance at $\mathbf{x}_i$, which are initialized as

$$\hat{z}^{(0)}(\mathbf{x}_i) = y_i, \quad \hat{v}_i^{(0)} = \hat{\sigma}^2, \tag{11}$$

[1] A large value of h_r results in a smoother image whereas too small a value results in inadequate denoising. The choice of this parameter is largely heuristic in nature.

Fig. 1. Examples of white Gaussian noise reduction: The columns left through right show the noisy image and the restored images by ISKR [1], NLM [2], OSA [3]. The rows from top to down are showing the experiments with different standard deviations ($\sigma = 15, 25, 50$). The corresponding PSNR values are shown in Table I.

where $\hat{\sigma}$ is an estimated standard deviation.

In each iteration, the estimation of each pixel is updated based on the previous iteration as follows.

$$\hat{z}^{(\ell+1)}(\mathbf{x}_i) = \frac{\sum_{\mathbf{x}_j \in W_i^{(\ell)}} K_{\mathbf{H}_i^{(\ell)}}(\hat{\mathbf{z}}_{w_i}^{(\ell)} - \hat{\mathbf{z}}_{w_j}^{(\ell)})\, y_j}{\sum_{\mathbf{x}_j \in W_i^{(\ell)}} K_{\mathbf{H}_i^{(\ell)}}(\hat{\mathbf{z}}_{w_i}^{(\ell)} - \hat{\mathbf{z}}_{w_j}^{(\ell)})} \tag{12}$$

where $\hat{\mathbf{z}}_{w_i}^{(\ell)}$ is a column stack vector that contains the pixels in an image patch w_i, $\mathbf{H}_j^{(\ell)} = h_r(\mathbf{V}_j^{(\ell)})^{-\frac{1}{2}}$, and h_r is the smoothing parameter. The matrix $\hat{\mathbf{V}}_j^{(\ell)}$ contains the harmonic means of estimated local noise variances:

$$\hat{\mathbf{V}}_j^{(\ell)} = \frac{1}{2}\mathrm{diag}\left[\cdots, \frac{(\hat{v}_{i-1}^{(\ell)})^2(\hat{v}_{j-1}^{(\ell)})^2}{(\hat{v}_{i-1}^{(\ell)})^2 + (\hat{v}_{j-1}^{(\ell)})^2}, \frac{(\hat{v}_i^{(\ell)})^2(\hat{v}_j^{(\ell)})^2}{(\hat{v}_i^{(\ell)})^2 + (\hat{v}_j^{(\ell)})^2}, \cdots\right] \tag{13}$$

and K is defined as the Gaussian kernel function:

$$K_{\mathbf{H}_i^{(\ell)}}(\hat{\mathbf{z}}_{w_i}^{(\ell)} - \hat{\mathbf{z}}_{w_j}^{(\ell)}) = \exp\left\{-\frac{(\hat{\mathbf{z}}_{w_i}^{(\ell)} - \hat{\mathbf{z}}_{w_j}^{(\ell)})^T(\hat{\mathbf{V}}_j^{(\ell)})^{-1}(\hat{\mathbf{z}}_{w_i}^{(\ell)} - \hat{\mathbf{z}}_{w_j}^{(\ell)})}{h_r^2}\right\}. \tag{14}$$

A patch size p is considered to be able to take care of the local geometry and texture in the image and is fixed (e.g 9×9 or 7×7) while the size of a local search window W_i is grows iteratively, determined by a point-wise statistically-based stopping rule.

The optimal window size is determined by minimization of the local mean square error (MSE) estimate at each pixel with respect to the search window size. In the absence of ground truth, this is approximated as the upper bound of the MSE obtained by estimating the bias and the variance separately. This estimation process is presented in detail in [3].

III. Experiments

In this section, we will compare the denoising performance of the methods introduced in the previous section by using synthetic and real noisy images. For all the experiments, we chose $q = 2$ and $N = 2$ for ISKR.

The first denoising experiment is shown in Fig. 1. For this experiment, using the Lena image, we added white

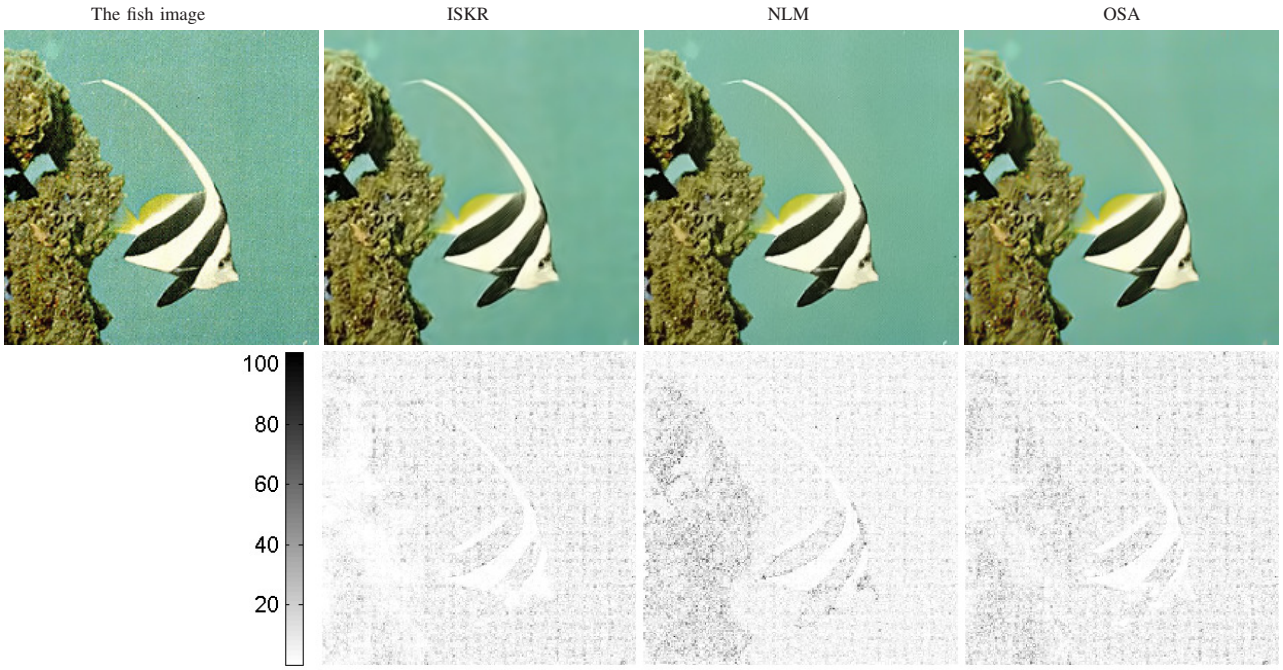

Fig. 2. Fish denoising examples: The images in the first row from left to right illustrate the noisy image, the estimated images by ISKR, NLM, and OSA method, respectively, and the second row illustrate absolute residual images in the luminance channel.

TABLE I
THE PSNR VALUES OF THE EXAMPLES OF WHITE GAUSSIAN NOISE REDUCTION (FIG. 1).

STD (σ)	Noisy	SKR	NLM	OSA
15	24.60	33.69	32.40	33.71
25	20.22	31.70	29.59	31.73
50	14.60	28.28	25.55	28.46

Gaussian noise with three different standard deviations ($\sigma = 15, 25, 50$). The synthetic noisy images are in the first column of Fig. 1, and the denoised images by ISKR, NLM, and OSA are shown in the second, third, and fourth columns, respectively. The corresponding PSNR[2] values are shown in Table I. For ISKR and NLM, we chose the parameters to produce the best PSNR values. The OSA method automatically chose its smoothing parameter.

Next, we applied the three method to some real noisy images: Fish and JFK images. The noise statistics are unknown for all the images. Applying ISKR, NLM, and OSA in $\mathbf{Y}\ \mathbf{C_b}\ \mathbf{C_r}$ channels individually, the restored images are illustrated in the first rows of Figs. 2 and 3, respectively. To compare the performances of the denoising methods, we take the absolute residuals in the luminance channel, which are shown below the corresponding denoising results of each method.

[2]Peak Signal to Noise Ratio $= 10 \log_{10}\left(\frac{255^2}{\text{Mean Square Error}}\right)$ [dB]

IV. CONCLUSION

While the present study is modest in its scope, several interesting but preliminary conclusions do emerge. First, we consider the relative performance of the considered methods. While very popular recently, the NLM method's performance, measured both qualitatively and quantitatively, is inferior to the other two methods. This is a bit surprising given the relatively recent surge of activity in this direction. The computational complexity of the NLM method is also very high, but as we mentioned earlier, this is a problem that has recently been addressed [4], [5].

The other two methods (ISKR and OSA) are very close in performance, with OSA having a slight edge in terms of PSNR. However, as the authors have also stated in their paper [3], this method tends to do less well when there is excessive texture present in the image. The ISKR algorithm suffers from a similar, but somewhat milder version of the same problem. A good comparison of these effects can be seen in Fig. 2.

The OSA method's performance depends strongly on the initial estimate of the noise variance, which can be badly biased if the assumptions of Gaussian noise statistics are violated. Indeed, if the estimated variance is much higher than the correct noise variance, this method can perform rather poorly. As such, it is worth pointing out that in the *real* experiments reported in this paper (Figs. 2 and 3) the automatically estimated noise variance led to rather poor results for OSA. Therefore, we adjusted this value by hand until the most visually appealing result was obtained. To be fair, we followed the same line of thinking and chose the

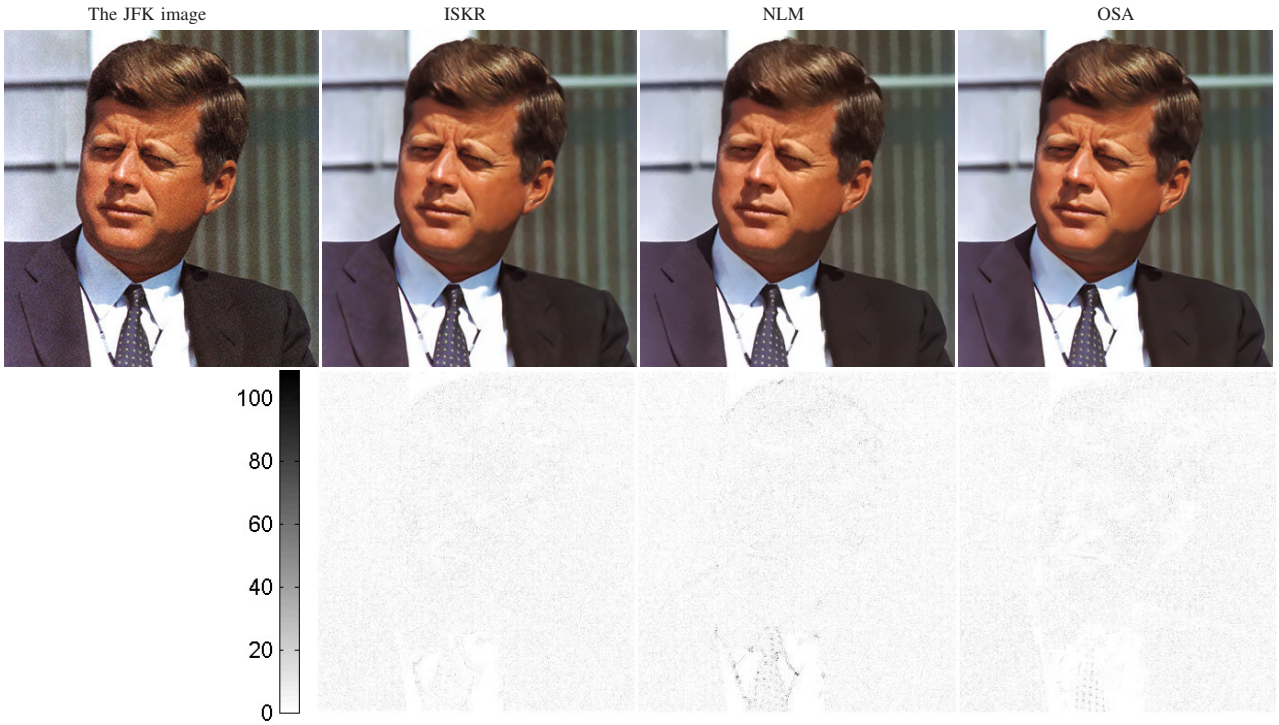

Fig. 3. JFK denoising examples: The images in the first row from left to right illustrate the noisy image, the estimated images by ISKR, NLM, and Kevrann's method, respectively, and the second row illustrate absolute residual images in the luminance channel.

parameters for ISKR and NLM as well to yield the best visual results.

While the ISKR does not depend on an explicit estimate or knowledge of the underlying noise variance (or distribution), several parameters such as window size, and the number of iterations, must be set by hand. Regarding the latter, if the iterations are continued, the image becomes increasingly blurry and MSE rises. Also, the ISKR is computationally very intensive, and efforts must be made in order to improve this aspect of the algorithm.

In terms of possible improvements, for all considered methods, there is room for growth and further innovation. In terms of both NLM, and OSA, it is worth noting that the weights produced by these methods for local pixel processing are always restricted to be non-negative numbers. This is an inherent limitation which can be overcome, and should lead to improved performance. For the ISKR, the choice of novel iteration methods; a proper stopping rule (limiting the number of iterations) based on the analysis of residuals of the estimation process; and reduction of computational complexity are all important issues for future research.

REFERENCES

[1] 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.

[2] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, vol. 4, no. 2, pp. 490–530, 2005.

[3] C. Kervrann and J. Bourlanger, "Optimal spatial adapation for patch-based image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 10, October 2006.

[4] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *Signal Processing Letters, IEEE*, vol. 12, no. 12, pp. 839–842, 2005.

[5] M. V. Radu Ciprian Bilcu, "Fast nonlocal means for image de-noising," in *Proceedings of IS&T/SPIE Symposium on Electronic Imaging, Digital Photography III Conference*, vol. 6502, San Jose, California USA, January-February 2007.