SpringerPlus
a SpringerOpen Journal

RESEARCH Open Access

# A Laplacian based image filtering using switching noise detector

Ali Ranjbaran*, Anwar Hasni Abu Hassan, Mahboobe Jafarpour and Bahar Ranjbaran

## Abstract

This paper presents a Laplacian-based image filtering method. Using a local noise estimator function in an energy functional minimizing scheme we show that Laplacian that has been known as an edge detection function can be used for noise removal applications. The algorithm can be implemented on a 3x3 window and easily tuned by number of iterations. Image denoising is simplified to the reduction of the pixels value with their related Laplacian value weighted by local noise estimator. The only parameter which controls smoothness is the number of iterations. Noise reduction quality of the introduced method is evaluated and compared with some classic algorithms like Wiener and Total Variation based filters for Gaussian noise. And also the method compared with the state-of-the-art method BM3D for some images. The algorithm appears to be easy, fast and comparable with many classic denoising algorithms for Gaussian noise.

**Keywords:** Local noise estimator; Denoising; Total variation; Energy functional; Laplacian

## Introduction

Denoising is one of the most important issues in image processing. The most popular noise removal methods are Adaptive Median Filtering (AMF), Total Variation (TV) based algorithms, Kernel based methods, Bilateral and Guided filtering and recently BM3D state-of-the-art in natural image denoising. In this work, we introduce a noise removal approach using a local noise estimator. We use the noise estimator in a minimization energy functional scheme. We obtain an iterative image denoising process using Laplacian. Denoising can be seen as adding values of pixels with their relative Laplacian weighted by local noise estimator.

Section 2 is related work. Section 3 represents our idea for denoising. We show how we can define a noise estimator using the sign of change in intensity of pixels in a 3x3 window. Using local noise estimator modified by Gaussian weight, we define an energy functional, drive the final equation and use it in an iterative denoising process. We show that although Laplacian is known as edge detector, it can be used for noise removal purposes. The algorithm is implemented in section 4. In section 5, results are shown and described. Moreover, figures of the denoising method are shown in comparison with Wiener, ROF and BM3D filters based on TV value and visual performance.

## Related work

A Total Variation based noise removal method (ROF) (Rudin et al. 1992) defines an energy functional that preserves edges of the image and smoothes Gaussian noisy area, based on the total variation norm minimizer. The TV regularization technique is a suitable method that can be extended to different noisy conditions such as Laplace and Poisson (Chan and Esedo Lu 2005; Li et al. 1994). The Split Bregman method (Goldstein and Osher 2008) is fast, reliable and extendable to different models of noise distribution. Split Bregman is a basic and effective tool in solving many functional-based problems such as Compressed Sensing (CS) (Candès et al. 2006). In recent years, the usage of kernel-based techniques in image denoising has developed the quality of noise removal results. The image used in kernel functioning is called the guidance image. One of the most popular approaches using the guidance image is bilateral filter (Petschnigg et al. 2004). Other important kernel-based methods are Data-adaptive kernel regression (Takeda et al. 2007), Non-Local Means (Buades et al. 2005) and Optimal Spatial Adaptation

* Correspondence: j.ranjbaran@yahoo.com
School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Seberang Perai Selatan, Pulau Pinang, Malaysia

(Kervrann et al. 2006). Another novel state of the art method was recently introduced as guided filter (He et al. 2010). The US patent 6229578 "Edge Detection Based Noise Removal Algorithm" (Acharya et al. 2001) is a denoising method based on using edge detector. This method removes noise by distinguishing between edge and non-edge pixels and applying a first noise removal technique to pixels classified as non-edge and a second noise removal technique to pixels classified as edge pixels. BM3D is a well-engineered algorithm which represents the current state-of-the-art method for denoising images corrupted by Additive White Gaussian Noise (AWGN) (Dabov et al. 2007; Dabov et al. 2006; Dabov et al. 2008; Dabov et al. 2009; Chen and Wu 2010). In another strategy, denoised image is considered as a linear combination of the original image and its average when the coefficients are determined by an edge detector (Ranjbaran et al. 2013).

## Methodology

Our methodology is based on using a local noise estimator in an energy functional minimizing scheme. Here we start with explaining our idea to define a local noise estimator. Consider Figure 1 where $u(x, y)$ is pixel intensity. A noise in x direction is estimated when the sign of the change of the image intensity for two adjacent pixels is in opposite direction. Taking two x direction gradient components $g_x$ and $g_{x+\Delta x}$ we write:

$$g_x = \frac{u(x, y) - u(x - \Delta x, y)}{\Delta x} \qquad g_{x+\Delta x} = \frac{u(x + \Delta x, y) - u(x, y)}{\Delta x} \tag{1}$$

A noise is detected when $g_x < 0$, $g_{x+\Delta x} > 0$ or $g_x > 0$, $g_{x+\Delta x} < 0$. Generally we have:

$$-g_x\, g_{x+\Delta x} > 0$$

Using Heaviside function $H$ the noise detector in x direction can be defined as:

$$SWN_x = H\left(-g_x\, g_{x+\Delta x}\right) \tag{2}$$

Noise appears in two conditions in an image, first as ideal noise, shown in Figure 1 and second as correlated noise (Figure 2). To distinguish these two cases we need to assign a weight to the detected location. Because $SWN_x$ acts as a switch its value is between zero and one. For ideal noisy pixels in the image plane, independent of the noise intensity, the value of weight should be one.

For the cases where noise is added to an edge as shown in Figure 2 the weight should be lower than one. To find a measure of the weight we consider that for ideal case $|u(x + \Delta x, y) - u(x - \Delta x, y)| = 0$ independent of noise intensity, but for non-ideal noise this difference is not zero. Then we can use a Gaussian weighting for the detected noise as the following equation:

$$WN_x = e^{-\left(g_x + g_{x+\Delta x}\right)^2} \tag{3}$$

Where:

$$g_x + g_{x+\Delta x} = \frac{u(x + \Delta x, y) - u(x - \Delta x, y)}{\Delta x} \tag{4}$$

As noise is added to the image in two directions we should drive the similar equations for y components. By using similar notations we find the final switching noise estimator as:

$$SWN = H\left(-g_x\, g_{x+\Delta x}\right) H\left(-g_y\, g_{y+\Delta y}\right) e^{-\left(g_x + g_{x+\Delta x}\right)^2} e^{-\left(g_y + g_{y+\Delta y}\right)^2} \tag{5}$$

To find a denoising way using $SWN$ we define a measure of noise intensity in the image plane. Since $SWN \geq 0$ the intensity of the noise in the noisy image can be defined as:

$$\iint H\left(-g_x\, g_{x+\Delta x}\right) H\left(-g_y\, g_{y+\Delta y}\right) e^{-\left(g_x + g_{x+\Delta x}\right)^2} e^{-\left(g_y + g_{y+\Delta y}\right)^2} \tag{6}$$

To reduce noise, we define the following energy functional and try to minimize it:

$$J = \iint (u - u_0)^2 + \lambda \iint H\left(-g_x\, g_{x+\Delta x}\right) H\left(-g_y\, g_{y+\Delta y}\right) e^{-\left(g_x + g_{x+\Delta x}\right)^2} e^{-\left(g_y + g_{y+\Delta y}\right)^2} \tag{7}$$

where $u$ and $u_0$ are denoised and noisy images respectively and the first part is regularization term. The energy

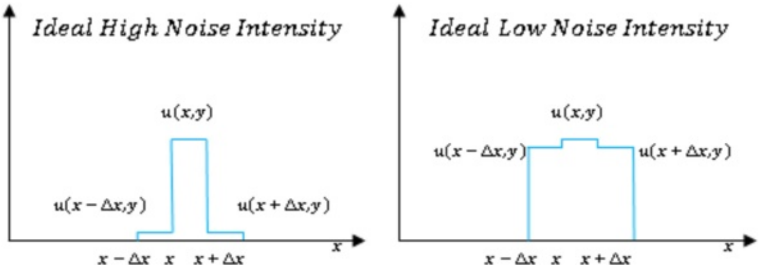


**Figure 1** Two ideal noisy pixels.

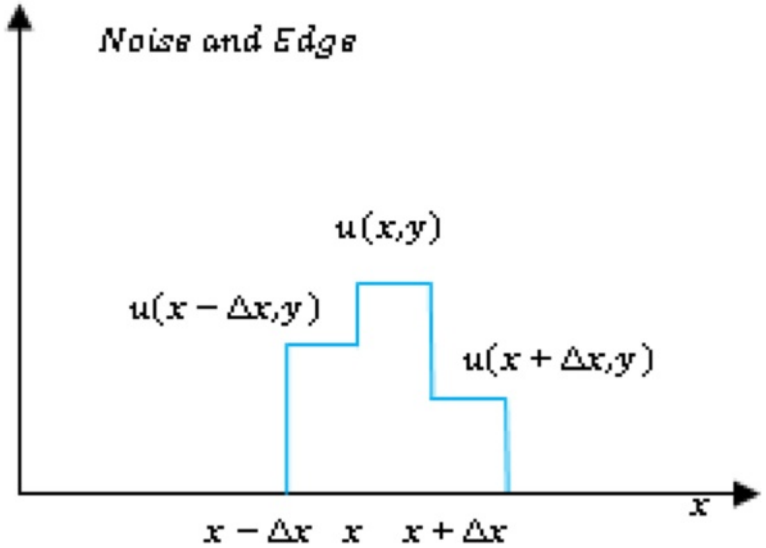


**Figure 2** Pixels include noise and edge.

functional is minimized in Appendix A. Although there are some techniques for computing $u$ by Equ. 47, this way is a bit sophisticated and difficult to implement. To find a more simple equation we add $\lambda\ f(u_0)$ to the two sides of the equation and write:

$$u_0 + \lambda f(u_0) = u + \lambda(f(u_0)–f(u)) \tag{8}$$

Based on $SWN$ operation and iteratively computing $f(u_0)$ and $f(u)$ in the locations classified as noise, we approximate $f(u_0) = f(u)$. Then the final equation we used in our implementation is:

$$u = u_0 + \lambda f(u_0) \tag{9}$$

This relation can be interpreted as follows: Because $u$ is disturbed by $f(u)$ and creates noisy image $u_0$ (Equ. 47), a similar process can restore $u$ from $u_0$ (Equ. 9). By decreasing noise after some cycles of iteration $f(u_0)$ goes to a small value. Equ. 47 presents a noise cancellation method based on using Laplacian value. The algorithm decreases the noise by adding the pixels value with Laplacian that weighted by $SWN$. Laplacian has been known as a common second-order edge detector but it has considerable value in noisy condition. Block diagram of the method is demonstrated in Figure 3.

Adding the intensity of pixels with the relative Laplacian is an averaging process. In an iterative process we can generally consider the evolution equation as:

$$u(t + \Delta t) = u(t) + \left(\lambda\ SWN\ \ \nabla^2 u\ \right)\Delta t \tag{10}$$

where $\Delta t$ is the evolution timing step. For the cases that $SWN = 1$ the evolution equation is:

$$u(t + \Delta t) = (1–\lambda)u(t) +\ \lambda\left(\frac{u(\mathrm{t}–\Delta t) + u(\mathrm{t} + \Delta t)}{2}\right)\ \ \Delta t \tag{11}$$

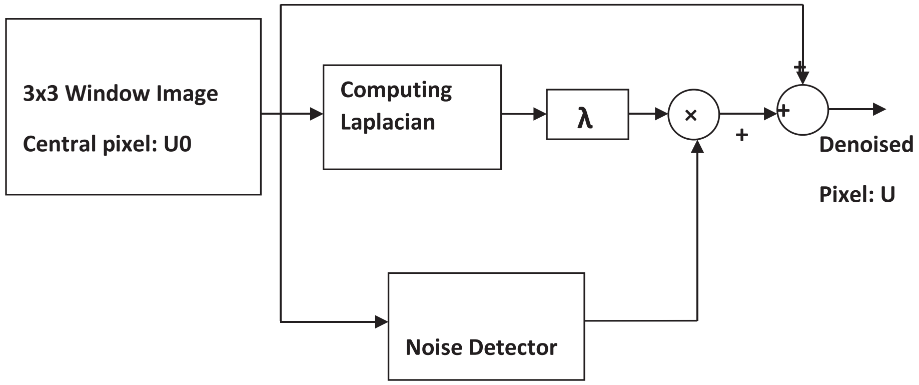


**Figure 3** Block diagram of the denoising method.

Then we have the following first order differential equation:

$$\dot{u} + \lambda u(t) = \lambda\, \bar{u} \tag{12}$$

When $\bar{u} = \frac{u(\mathrm{t}-\Delta t)+u(\mathrm{t}+\Delta t)}{2}$ is the average value of $u$. The timing response is:

$$u(t) = \bar{u} + (u_0 - \bar{u})\, e^{-\lambda t} \qquad t \geq 0 \tag{13}$$

where $\lambda$ is the time constant. For $\lambda \geq 0$ the denoised image $u$ exponentially approaches to it steady state value $\bar{u}$ in the locations where $SWN$ is high. To find the constraints on $\lambda$ we note that because $0 \leq u \leq 1$ and $-1 \leq f(u_0) \leq 1$, we have two constrains on $\lambda$ as:

$$-u_0 \leq \lambda \leq 1 - u_0 \qquad -(1-u_0) \leq \lambda \leq u_0 \tag{14}$$

As $0 \leq \lambda \leq 1 - u_0$ and we choose $\geq 0$, maximum value for $\lambda$ is:

$$\lambda_{max} = 1 \tag{15}$$

It is predictable that in implementing the algorithm by large number of iterations $\lambda$ must be a positive small value. In real condition $SWN$ is not constant and reduced during the evolution.

## Implementation

We have implemented our method in Matlab for Gaussian noise and evaluated it on different images. For implementation the Heaviside function is approximated by inverse tangent function:

$$H(-g_x\, g_{x+\Delta x}) H\left(-g_y\, g_{y+\Delta y}\right) \approx \frac{\frac{\pi}{2} + tan^{-1}\left(-C g_x g_{x+\Delta x}\right)}{\pi} \frac{\frac{\pi}{2} + tan^{-1}\left(-C g_y g_{y+\Delta y}\right)}{\pi} \tag{16}$$

According to Equ.12 $\lambda$ controls timing response and the value is between 0 and 1. So choosing the middle value can be suitable. Based on experimental results $C = 3$ is found as an appropriate value for noisy cases. We use a 3x3 window for simplicity and fast computing. The method is implemented by 10 numbers of iteration. The algorithm of implementation can be shown as the following steps:

1. Setting parameters : window size 3×3, $\lambda = 0.5$
2. Reading image $u_0$
3. Adding zero mean Gaussian Noise (imnoise code)
4. Computing $SWN$ for current pixel

$$SWN = \frac{\frac{\pi}{2} + tan^{-1}\left(-3 g_x g_{x+\Delta x}\right)}{\pi} \frac{\frac{\pi}{2} + tan^{-1}\left(-3 g_y g_{y+\Delta y}\right)}{\pi}\, e^{-\left(g_x + g_{x+\Delta x}\right)^2} e^{-\left(g_y + g_{y+\Delta y}\right)^2} \tag{17}$$

5. Computing Laplacian for current pixel

$$\nabla^2 u = \frac{u(x+\Delta x, y) + u(x-\Delta x, y) - 2u(x,y)}{4} + \frac{u(x, y+\Delta y) + u(x, y-\Delta y) - 2u(x,y)}{4} \tag{18}$$

6. Updating $u = u_0 + \lambda\ SWN\ \nabla^2 u_0$ for the current pixel
7. Going to step 4 and continuing
8. Finishing when the whole of the image is scanned for ten times.

## Results and discussions

We implemented our method in two noisy conditions (0.005 and 0.1 noise variance using imnoise matlab code for Gaussian noise) and compared it with ROF model using evolve2D code with 10 iterations and Wiener filter using wiener2 ('image', [3 3]) matlab code. $f(u_0)$ can be interpreted as a noise intensity pattern in the image plane. Similar to ROF model in which $\left\| div\left(\frac{\nabla u_0}{|\nabla u_0|}\right) \right\|$ can be used as a measure of noise variance (Dabov et al. 2007), $\|f(u_0)\|$ is related to noise intensity. An example is shown in Figure 4. The results including noisy and denoised images are demonstrated in Figure 5 for Boat, Figure 6 for Man and Figure 7 for House. Figures 8, 9 and 10 show the results for Cameraman, Lena and Barbara respectively. TV for noisy and denoised images is shown in Tables 1 and 2. TV of the denoised image is totally comparable with ROF model. The computation time of our model is equal to ROF method with 10 iterations. $f(u_0)$ has a decreasing behavior during the filtring time. At start, $f(u_0)$ is high in noisy pixels identified by $SWN$. Because image intensity is reduced by Laplacian value, $f(u_0)$ tends to zero after some iteration. An example of such response is shown if Figure 11. Number of iterations is the only parameter that controls smoothness and running time. Large number of iteration makes the image blurry. Since $f(u_0)$ goes to zero after some iteration, we can implement the algorithm adaptively until $f(u_0)$ reaches a small value. This value should be estimated experimentally to make the algorithm applicable for every noisy condition. We also compared our method with BM3D technique for Boat, House and Cameraman in two SNR 25 and 75. The results are shown in Figures 12, 13, 14, 15, 16, 17. The denoising technique used in this work can be generalized to other applications for reducing any

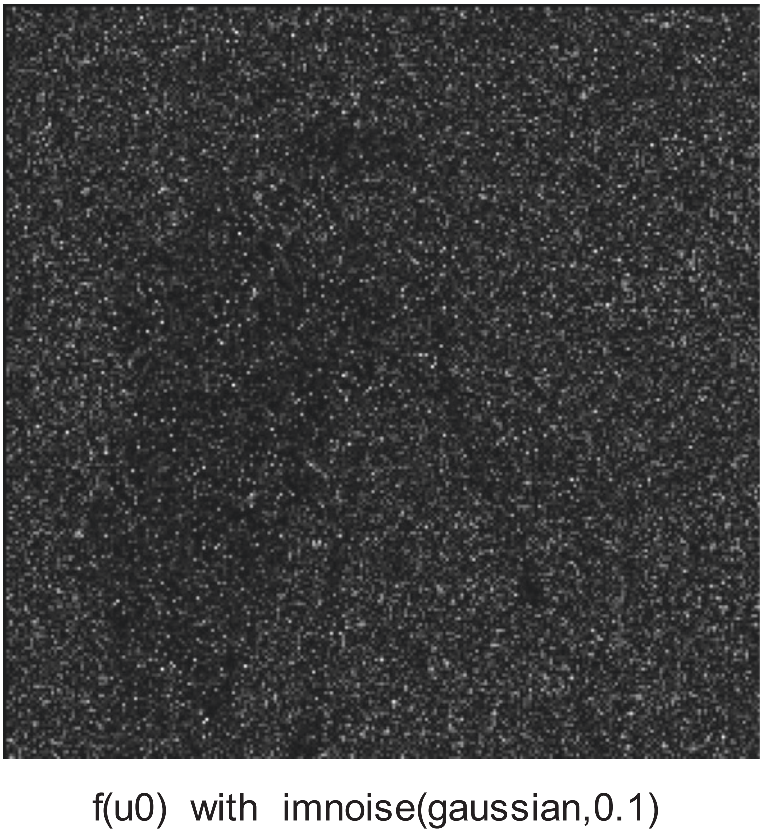


**Figure 4** ***f***(***u***$_0$) **for Cameraman with 0.1 noise variance.**

feature of the image ($F(u)$) in other image processing tasks such as deblurring. A proposing framework can be written as the following energy functional:

$$J = \iint (u - u_0)^2 + \lambda \iint Detector(F(u))\, Wiegth(F(u)) \tag{19}$$

Such that:

$$|Detector(F)| \le 1 \quad and \quad |Wiegth(F)| \le 1$$

and $\lambda$ is a control parameter determined experimentally. Minimizing the energy functional results the final relation for implementation.

$$\frac{\partial J}{\partial u} = \iint 2(u - u0) + \iint \lambda \; \frac{\partial (F)}{\partial (u)} \left[ \frac{\partial (Detector)}{\partial (F)} Wiegth(F) + \frac{\partial (Wiegth)}{\partial (F)} Detector(F) \right] = 0 \tag{20}$$

As an example, for deblurring, *Detector*($F$) and *Wiegth* ($F$) can be suggested as:

$$Detector(F) = H\left(g_x\, g_{x+\Delta x}\right) H\left(g_y\, g_{y+\Delta y}\right) \tag{21}$$

$$Wiegth(F) = e^{-\left(g_x - g_{x+\Delta x}\right)^2} e^{-\left(g_y - g_{y+\Delta y}\right)^2} \tag{22}$$

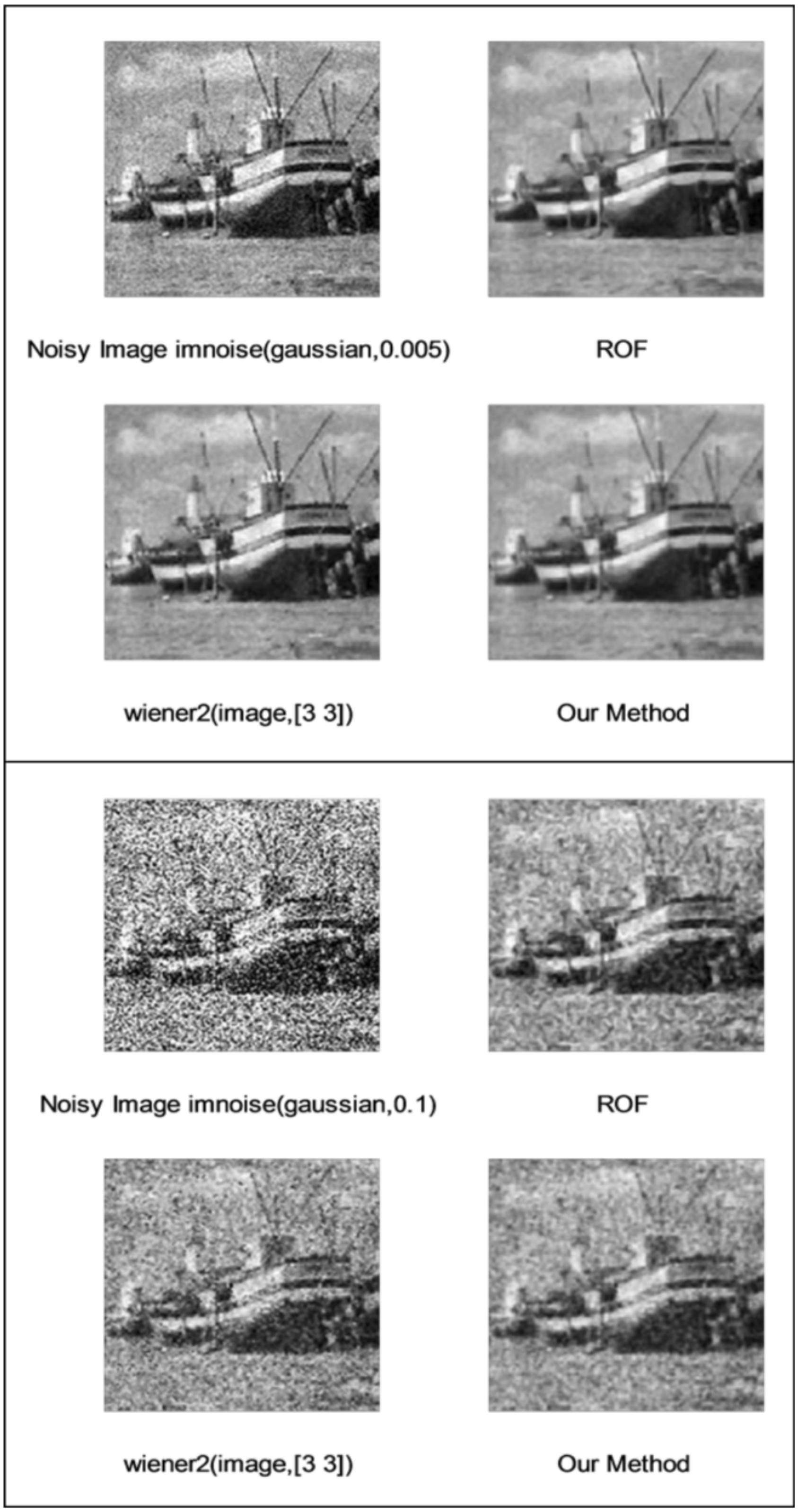
where the blurring locations identified when the sign of change in intensity of two adjacent pixels is equal ($g_x\, g_{x\,+\,\Delta x} > 0$ , $g_y\, g_{y\,+\,\Delta y} > 0$), and weighted by difference in their gradient components for x and y directions. Computing $\frac{\partial J}{\partial u} = 0$ the final relation is:

$$u = u_0 + \lambda \left(u_{0xxxx} + u_{0yyyy}\right) H\left(g_x\, g_{x+\Delta x}\right) H\left(g_y\, g_{y+\Delta y}\right) e^{-\left(g_x - g_{x+\Delta x}\right)^2} e^{-\left(g_y - g_{y+\Delta y}\right)^2} \tag{23}$$

This relation shows that in the blurred locations, identified by the blurring detector, $u$ is restored by the forth order of differentiation in the blurred areas.

## Conclusion

We presented a noise removal method using a local switching noise estimator in an energy functional minimizing process. We showed that in addition to using Laplacian in edge detection tasks, it can be used for noise removal applications. Smoothness can be easily controlled just by the number of iterations. We compared the



Noisy Image imnoise(gaussian,0.005)

ROF

wiener2(image,[3 3])

Our Method

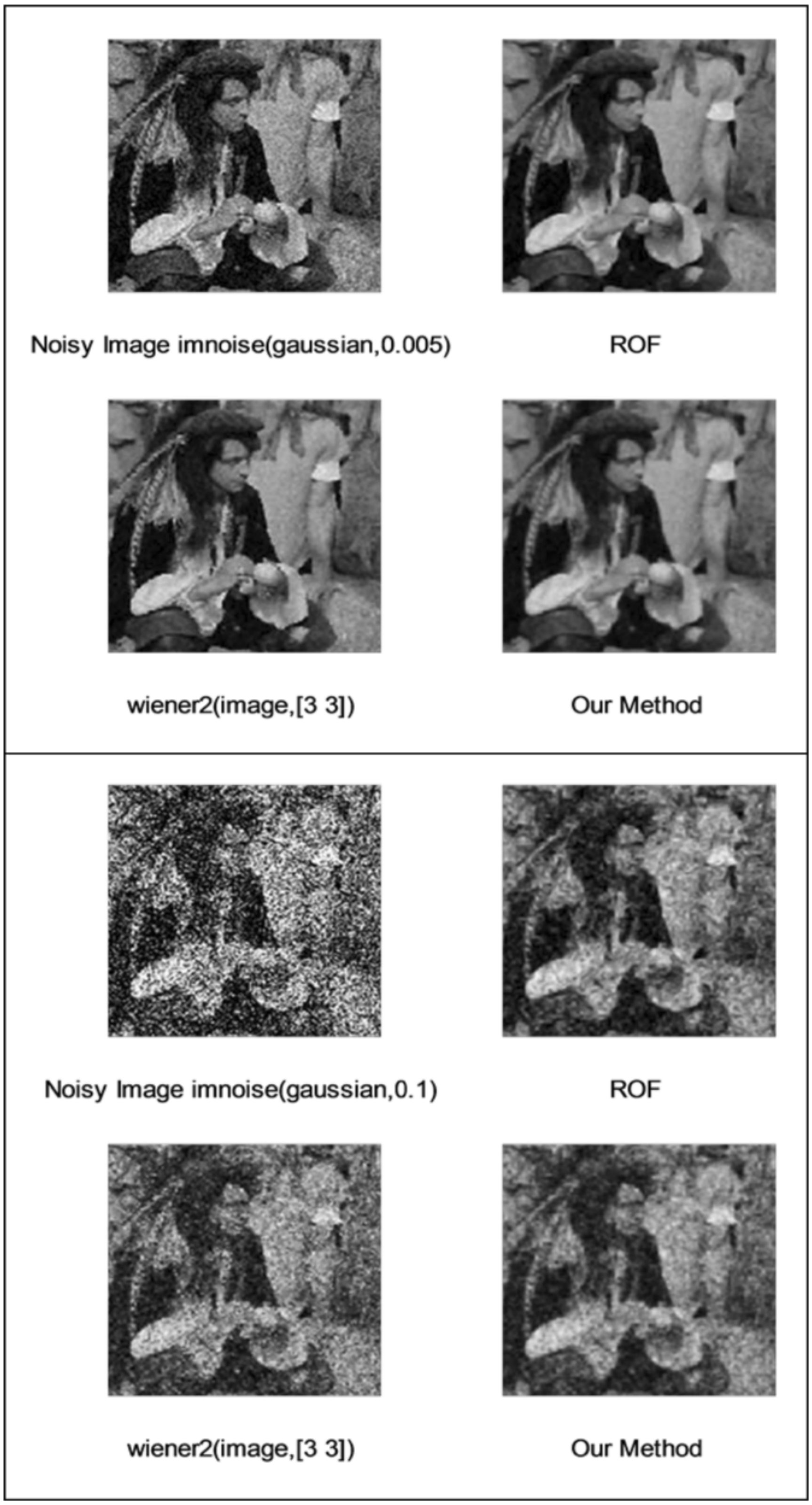
Noisy Image imnoise(gaussian,0.1)

ROF

wiener2(image,[3 3])

Our Method

**Figure 5** Results of ROF, wiener filter and our denoising method for Boat.



Noisy Image imnoise(gaussian,0.005)

ROF

wiener2(image,[3 3])

Our Method

Noisy Image imnoise(gaussian,0.1)

ROF

wiener2(image,[3 3])

Our Method

**Figure 6** Results of ROF, wiener filter and our denoising method for Man.

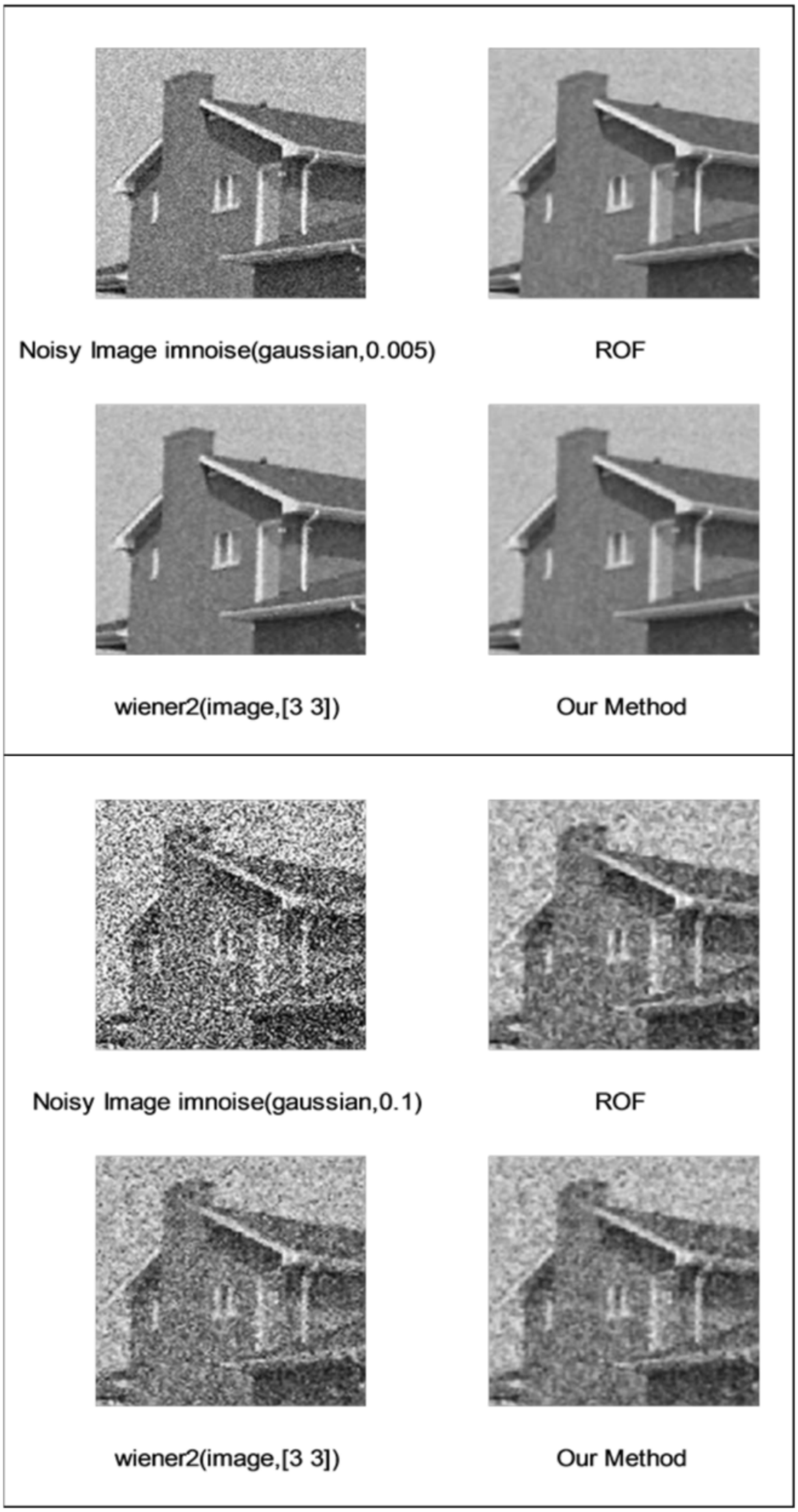
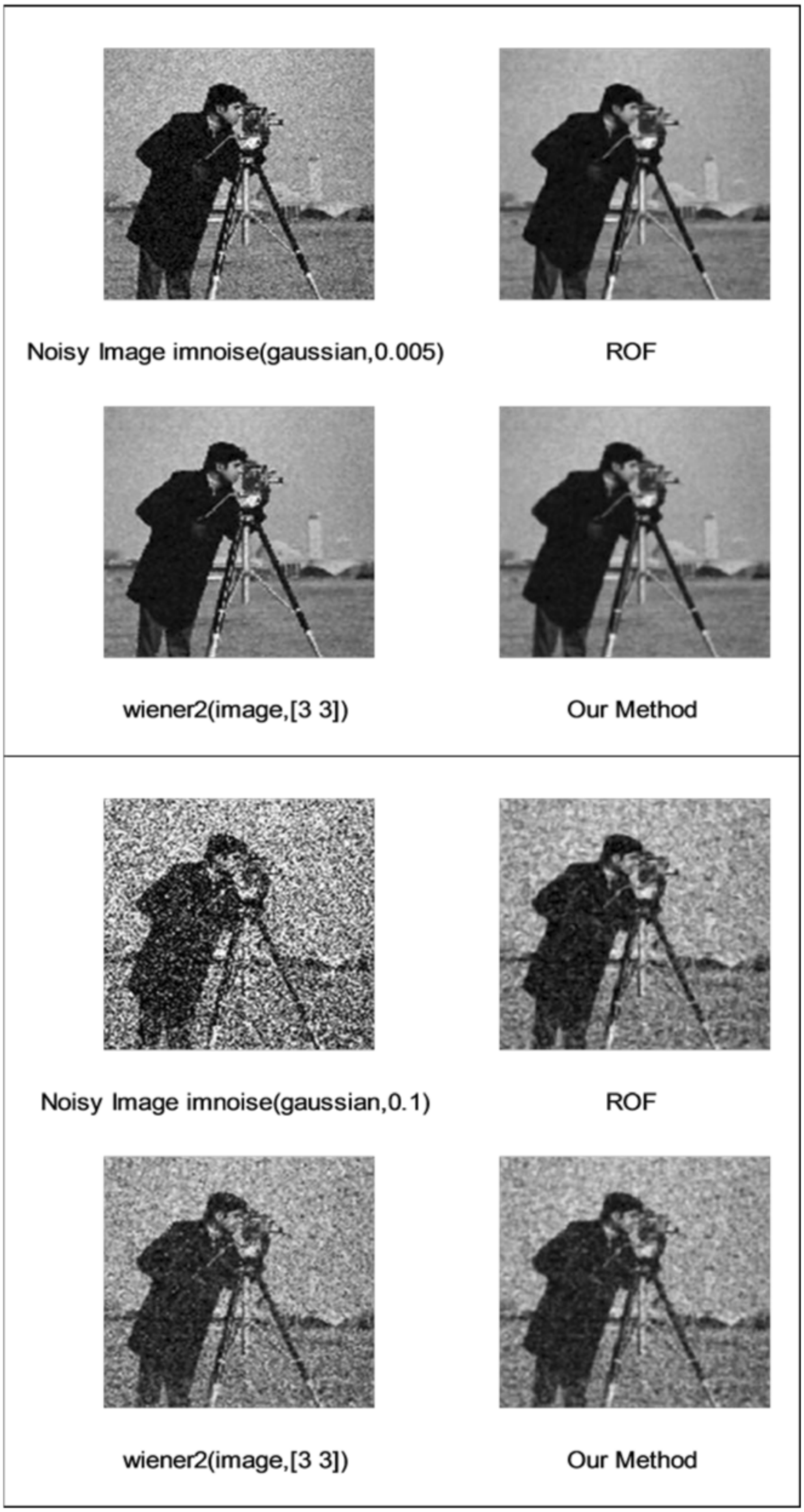

Noisy Image imnoise(gaussian,0.005)
ROF
wiener2(image,[3 3])
Our Method
Noisy Image imnoise(gaussian,0.1)
ROF
wiener2(image,[3 3])
Our Method

**Figure 7** Results of ROF, wiener filter and our denoising method for House.



Noisy Image imnoise(gaussian,0.005)

ROF

wiener2(image,[3 3])

Our Method

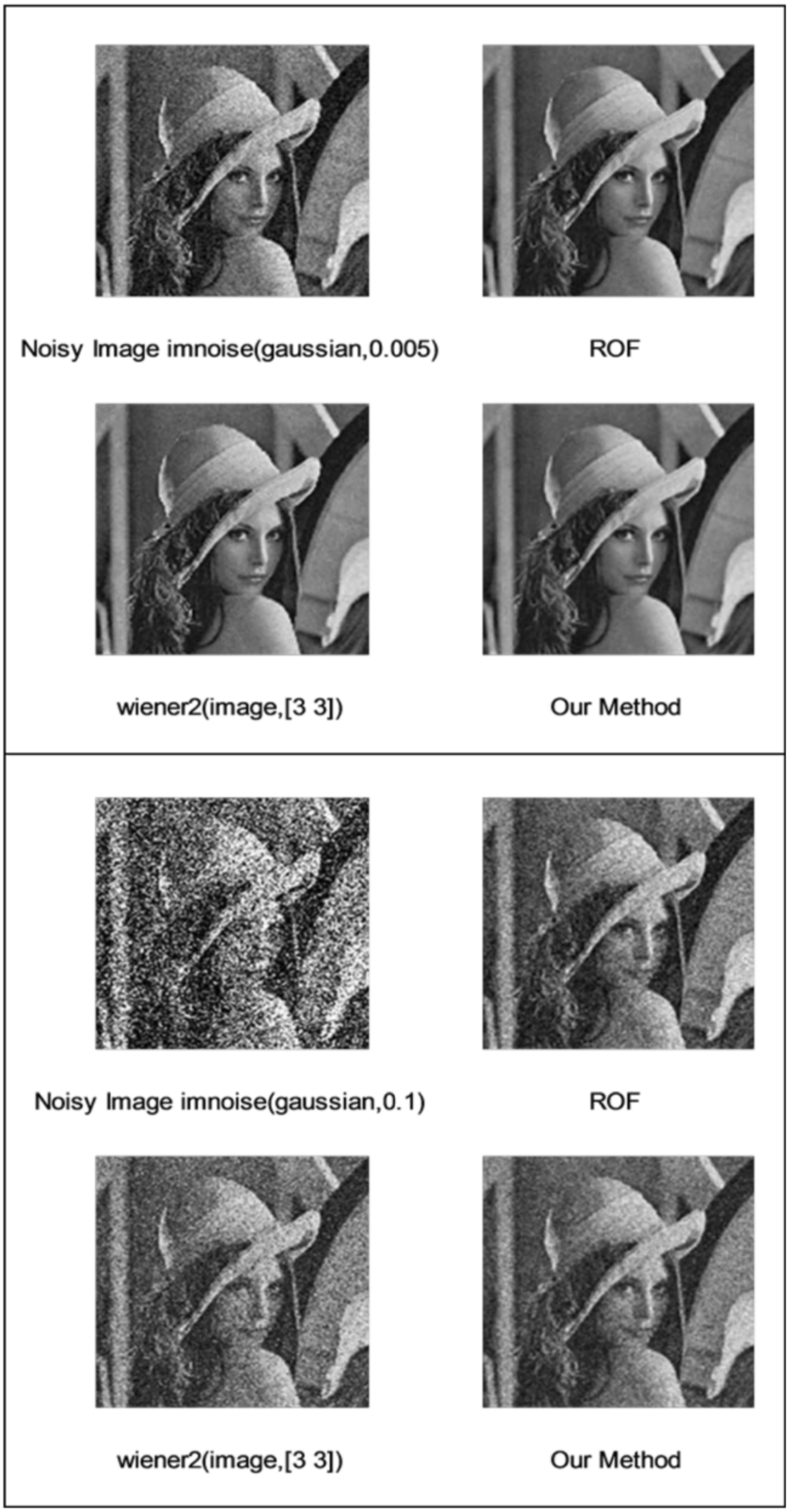
Noisy Image imnoise(gaussian,0.1)

ROF

wiener2(image,[3 3])

Our Method

**Figure 8** Results of ROF, wiener filter and our denoising method for Cameraman.



Noisy Image imnoise(gaussian,0.005)

ROF

wiener2(image,[3 3])

Our Method

Noisy Image imnoise(gaussian,0.1)

ROF

wiener2(image,[3 3])

Our Method

**Figure 9** Results of ROF, wiener filter and our denoising method for Lena.

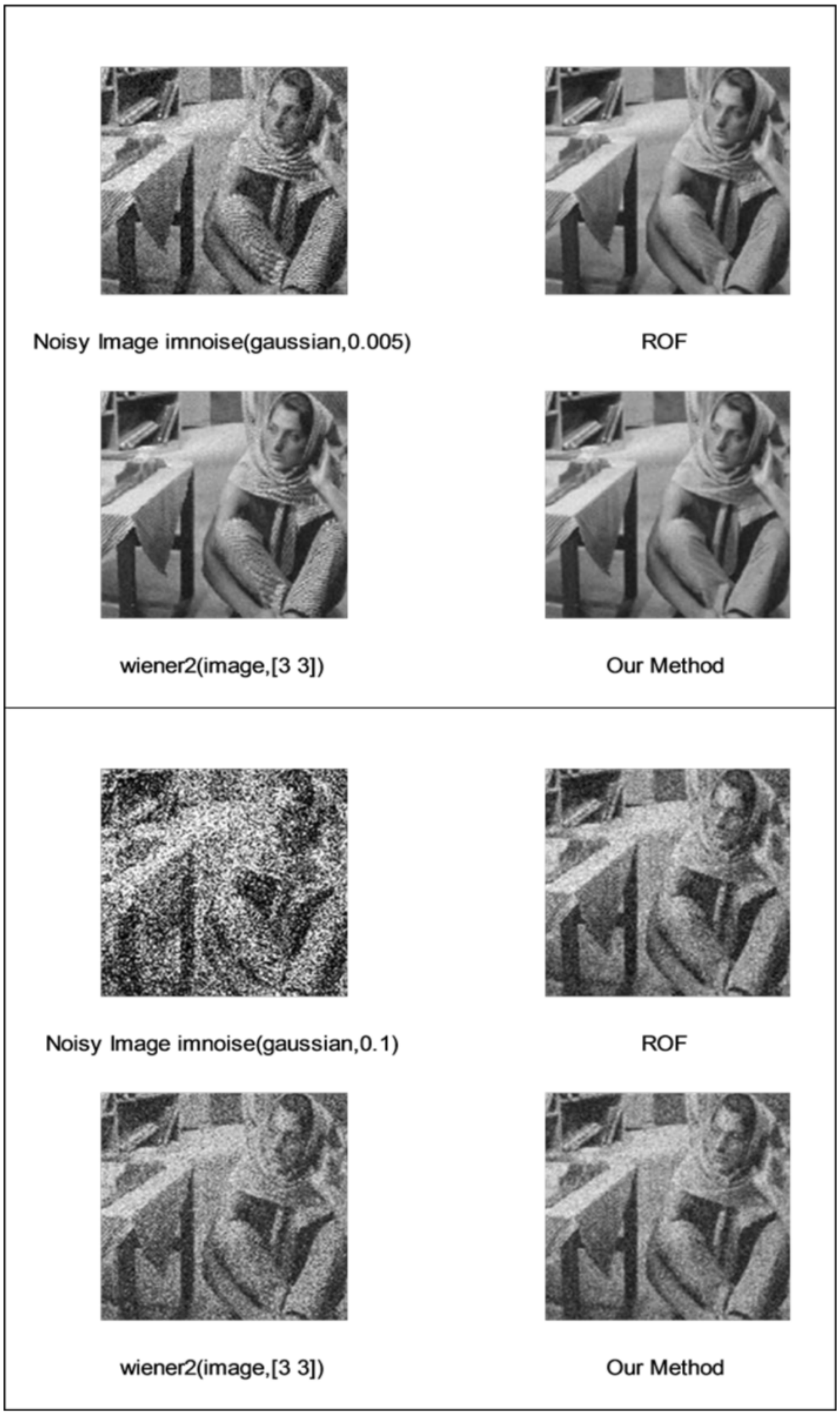

Noisy Image imnoise(gaussian,0.005)
ROF
wiener2(image,[3 3])
Our Method
Noisy Image imnoise(gaussian,0.1)
ROF
wiener2(image,[3 3])
Our Method

**Figure 10** **Results of ROF, wiener filter and our denoising method for Barbara.**

**Table 1 TV for 6 noisy and denoised images with noise variance 0.005**

| Noise variance (0.005) | TV (original mage) | TV (noisy image) | TV (ROF model) | TV (Wiener filter) | TV (our method) |
|---|---|---|---|---|---|
| Boat | 1.6 | 2.09 | 1.31 | 1.42 | 1.22 |
| Man | 1.76 | 2.20 | 1.32 | 1.51 | 1.22 |
| House | 1.3 | 1.81 | 1.21 | 1.27 | 1.14 |
| Cameraman | 1.67 | 2.13 | 1.33 | 1.53 | 1.20 |
| Lena | 1.26 | 1.74 | 1.16 | 1.24 | 1.12 |
| Barbara | 1.57 | 2.07 | 1.15 | 1.36 | 1.10 |

method with some classic methods like ROF and Wiener filters based on TV value and also with the state-of-the-art BM3D. The result of denoising is totally comparable with ROF model. The computation time is equal to ROF model with 10 iterations. Based on experimental results we concluded that in relation to classic filters like ROF the method appears to be easy, fast and applicable for many noisy images. We analyzed that the technique can be applied for other image processing applications like deblurring by defining the appropriate detector and weight functions. The main disadvantage of the method is its filtering action on the area of the image including low intensity edges. The method can be improved to represent better response by defining better noise detectors.

## Appendix A: driving denoising equation

Here we drive the denoising equation by minimizing the energy functional. First we use Taylor series and relate $u(x-\Delta x, y)$ to x component of the gradient of the image $u_x$. We have:

$$u(x-\Delta x, y) = u(x, y) - \Delta x\, u_x + \frac{1}{2}\Delta x^2\, u_{xx} \tag{24}$$

Then $g_x$ can be computed as:

$$g_x = u_x - \frac{1}{2}\Delta x\, u_{xx} \tag{25}$$

And for $g_{x+\Delta x}$ we can write:

$$g_{x+\Delta x} = u_x + \frac{1}{2}\Delta x\, u_{xx} \tag{26}$$

Using Taylor series similarly for y components, $g_y$ and $g_{y+\Delta y}$ are:

$$g_y = u_y - \frac{1}{2}\Delta y\, u_{yy} \tag{27}$$

$$g_{y+\Delta y} = u_y + \frac{1}{2}\Delta y\, u_{yy} \tag{28}$$

Based on the above relations, we write:

$$-g_x\, g_{x+\Delta x} = -\,{u_x}^2 + \frac{1}{4}\Delta x^2\, {u_{xx}}^2 \tag{29}$$

$$-g_y\, g_{y+\Delta y} = -\,{u_y}^2 + \frac{1}{4}\Delta y^2\, {u_{yy}}^2 \tag{30}$$

The minimizer $u$ is found by $\frac{\partial J}{\partial u} = 0$. Considering the following relations for simplicity:

$$H_1 = H\left(-g_x\, g_{x+\Delta x}\right) \quad , \quad H_2 = H\left(-g_y\, g_{y+\Delta y}\right) \quad , \\ Z_1 = e^{-\left(g_x + g_{x+\Delta x}\right)^2} \quad , \quad Z_2 = e^{-\left(g_y + g_{y+\Delta y}\right)^2} \tag{31}$$

We can write:

$$J = \iint (u - u_0)^2 + \lambda \iint H_1\, H_2\, Z_1\, Z_2 \tag{32}$$

**Table 2 TV for 6 noisy and denoised images with noise variance 0.1**

| Noise variance (0.1) | TV (original image) | TV(noisy image) | TV (ROF model) | TV (Wiener filter) | TV (our method) |
|---|---|---|---|---|---|
| Boat | 1.6 | 8.68 | 1.63 | 2.20 | 1.43 |
| Man | 1.76 | 8.28 | 1.62 | 2.26 | 1.42 |
| House | 1.3 | 8.72 | 1.59 | 2.12 | 1.39 |
| Cameraman | 1.67 | 8.44 | 1.61 | 2.22 | 1.41 |
| Lena | 1.26 | 8.11 | 1.53 | 2.12 | 1.34 |
| Barbara | 1.57 | 8.50 | 1.54 | 2.14 | 1.35 |

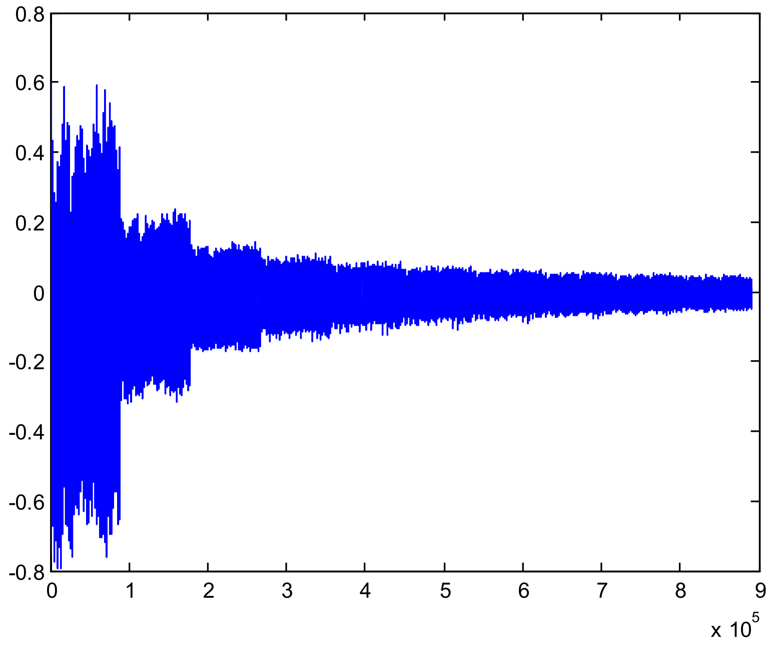


**Figure 11 Decreasing behavior of *f*($u_0$) during the denoising process.**

Differentiating $J$ with respect to $u$ we have:

$$\frac{\partial J}{\partial u}=\iint\Big[2(u-u_0)+\frac{\partial\lambda}{\partial u}(H_1\,H_2Z_1\,Z_2)$$
$$+\lambda\frac{\partial H(-g_x\,g_{x+\Delta x})}{\partial u}H_2Z_1\,Z_2$$
$$+\lambda\frac{\partial H\left(-g_y\,g_{y+\Delta y}\right)}{\partial u}H_1Z_1\,Z_2-2\lambda(g_x+g_{x+\Delta x})$$
$$\frac{\partial(g_x+g_{x+\Delta x})}{\partial u}H_1\,H_2Z_1\,Z_2$$
$$-2\lambda\left(g_y+g_{y+\Delta y}\right)\frac{\partial\left(g_y+g_{y+\Delta y}\right)}{\partial u}H_1\,H_2Z_1\,Z_2\Big]=0 \tag{33}$$

First we compute $\frac{\partial g_x}{\partial u}$:

$$\frac{\partial g_x}{\partial u}=\left[\frac{u_{xx}}{u_x}+\frac{u_{xy}}{u_y}\right]-\left[\frac{\Delta x}{2}\left(\frac{u_{xxx}}{u_x}+\frac{u_{xxy}}{u_y}\right)\right]=\gamma_x-\beta_x \tag{34}$$

Similarly for $g_{x+\Delta x}$ we have:

$$\frac{\partial g_{x+\Delta x}}{\partial u}=\gamma_x+\beta_x \tag{35}$$

Then we can write:

$$\frac{\partial(g_x+g_{x+\Delta x})}{\partial u}=2\gamma_x=2\left[\frac{u_{xx}}{u_x}+\frac{u_{xy}}{u_y}\right] \tag{36}$$

And for y direction:

$$\frac{\partial\left(g_y+g_{y+\Delta y}\right)}{\partial u}=2\gamma_y=2\left[\frac{u_{yy}}{u_y}+\frac{u_{yx}}{u_x}\right] \tag{37}$$

Second we have:

$$\frac{\partial\left(-g_x\,g_{x+\Delta x}\right)}{\partial u}=-2u_{xx}-2u_{xy}\frac{u_x}{u_y}+\frac{\Delta x^2}{2}\left(\frac{u_{xxx}}{u_x}+\frac{u_{xxy}}{u_y}\right)u_{xx} \tag{38}$$

$$\frac{\partial\left(-g_y\,g_{y+\Delta y}\right)}{\partial u}=-2u_{yy}-2u_{yx}\frac{u_y}{u_x}+\frac{\Delta y^2}{2}\left(\frac{u_{yyy}}{u_y}+\frac{u_{yyx}}{u_x}\right)u_{yy} \tag{39}$$

Derivative of Heaviside Function appears the impulse function:

$$\frac{\partial H(.)}{\partial u}=\frac{\partial H(.)}{\partial(.)}\,\frac{\partial(.)}{\partial u}=\delta(.)\,\frac{\partial(.)}{\partial u} \tag{40}$$

where $\delta(.)$ is the impulse function. We Ignore its effect in the denoising process and assume $\lambda$ is constant during the implementation. Therefore we have:

$$\frac{\partial H(-g_x\,g_{x+\Delta x})}{\partial u}\approx 0\quad,\quad\frac{\partial H\left(-g_y\,g_{y+\Delta y}\right)}{\partial u}\approx 0\quad,\quad\frac{\partial\lambda}{\partial u}\approx 0 \tag{41}$$

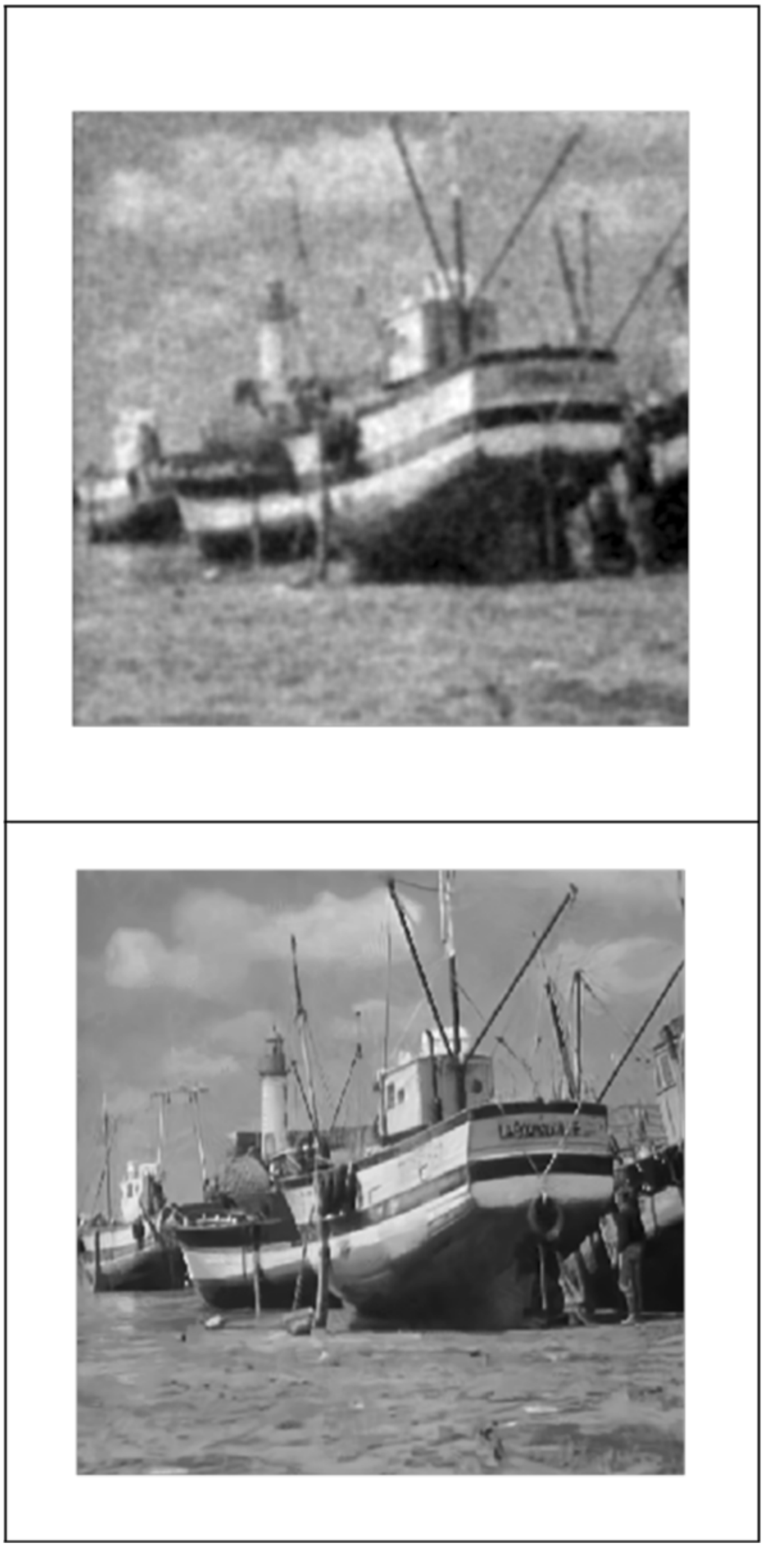


**Figure 12** Results of our method (up) and BM3D technique (down) with SNR = 25 for Boat.

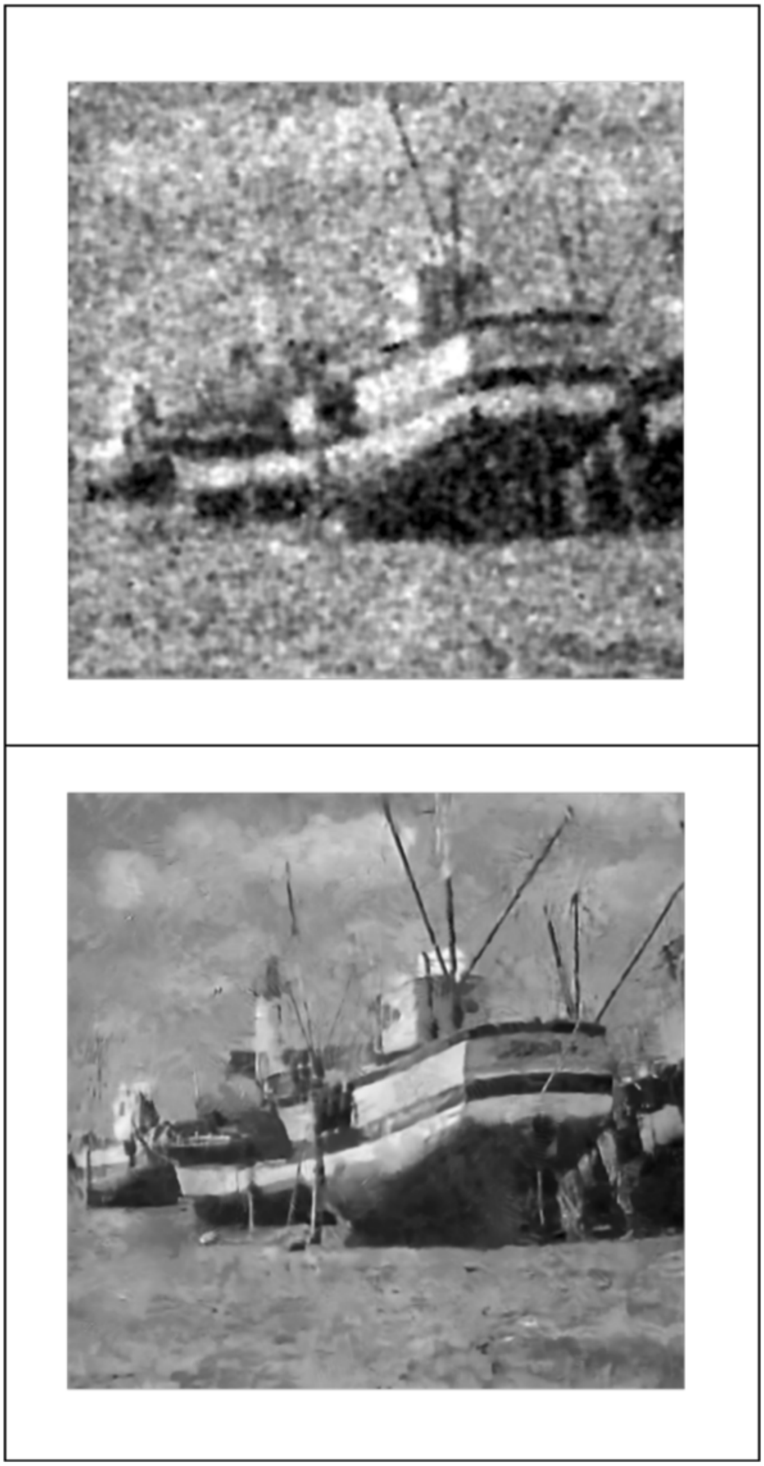


**Figure 13** Results of our method (up) and BM3D technique (down) with SNR = 75 for Boat.

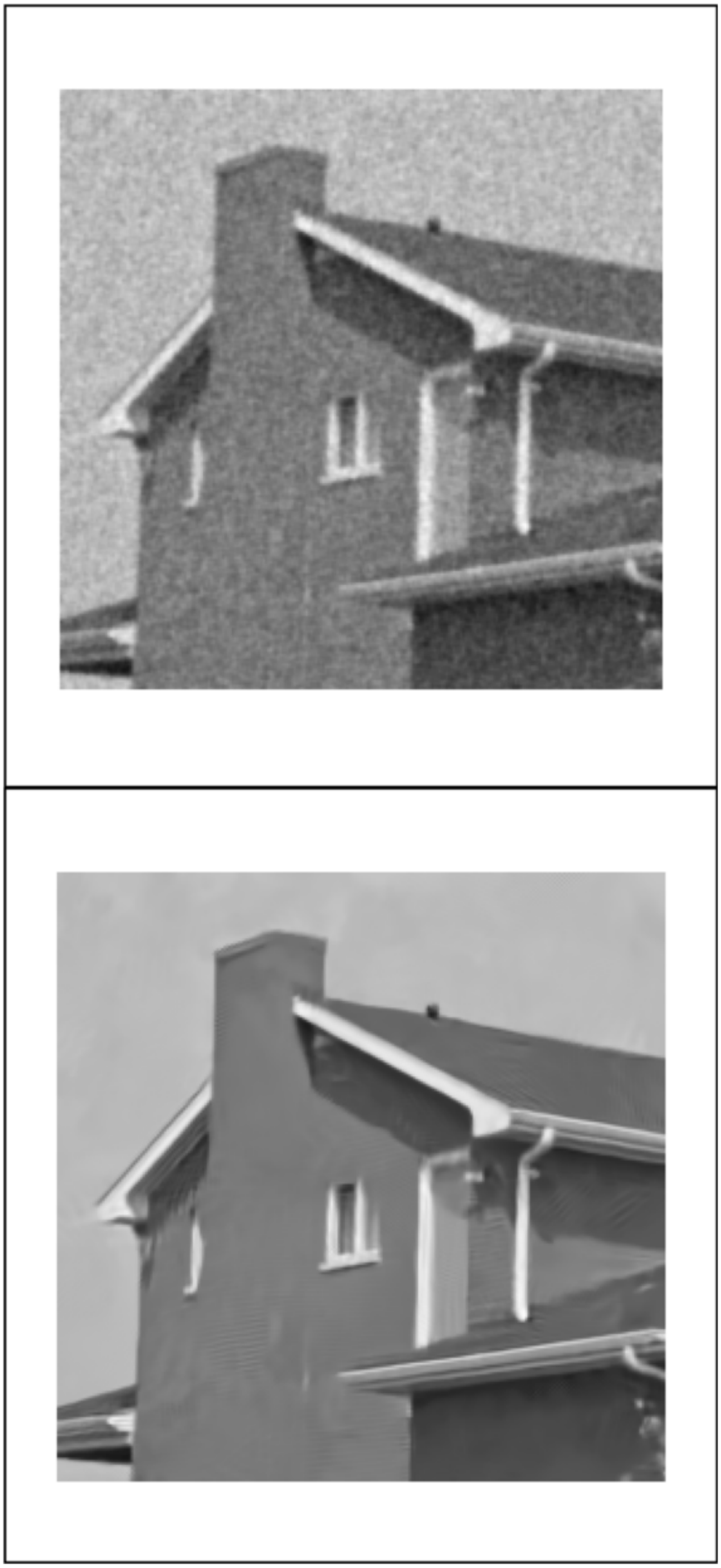


**Figure 14** Results of our method (up) and BM3D technique (down) with SNR = 25 for House.

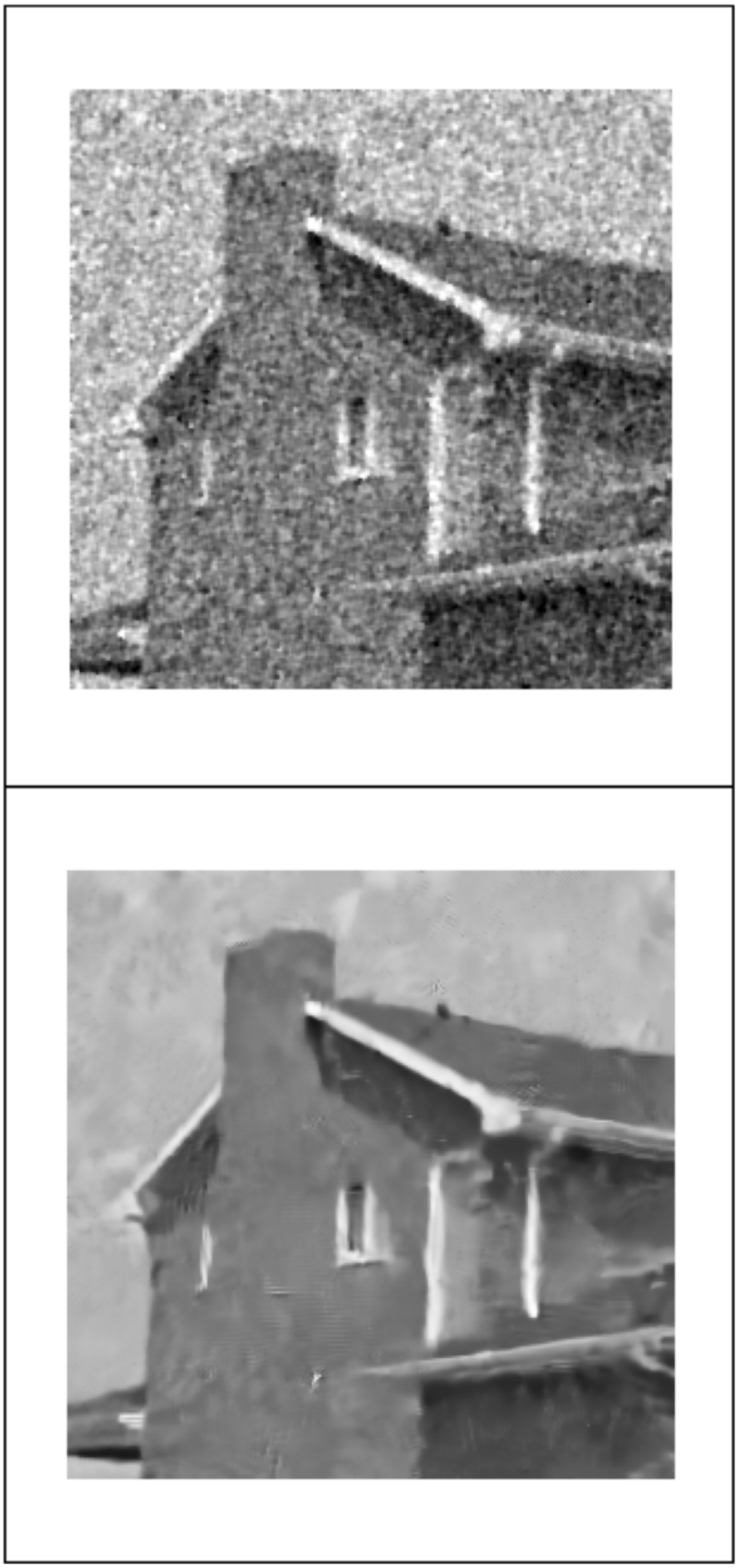


**Figure 15** **Results of our method (up) and BM3D technique (down) with SNR = 75 for House.**

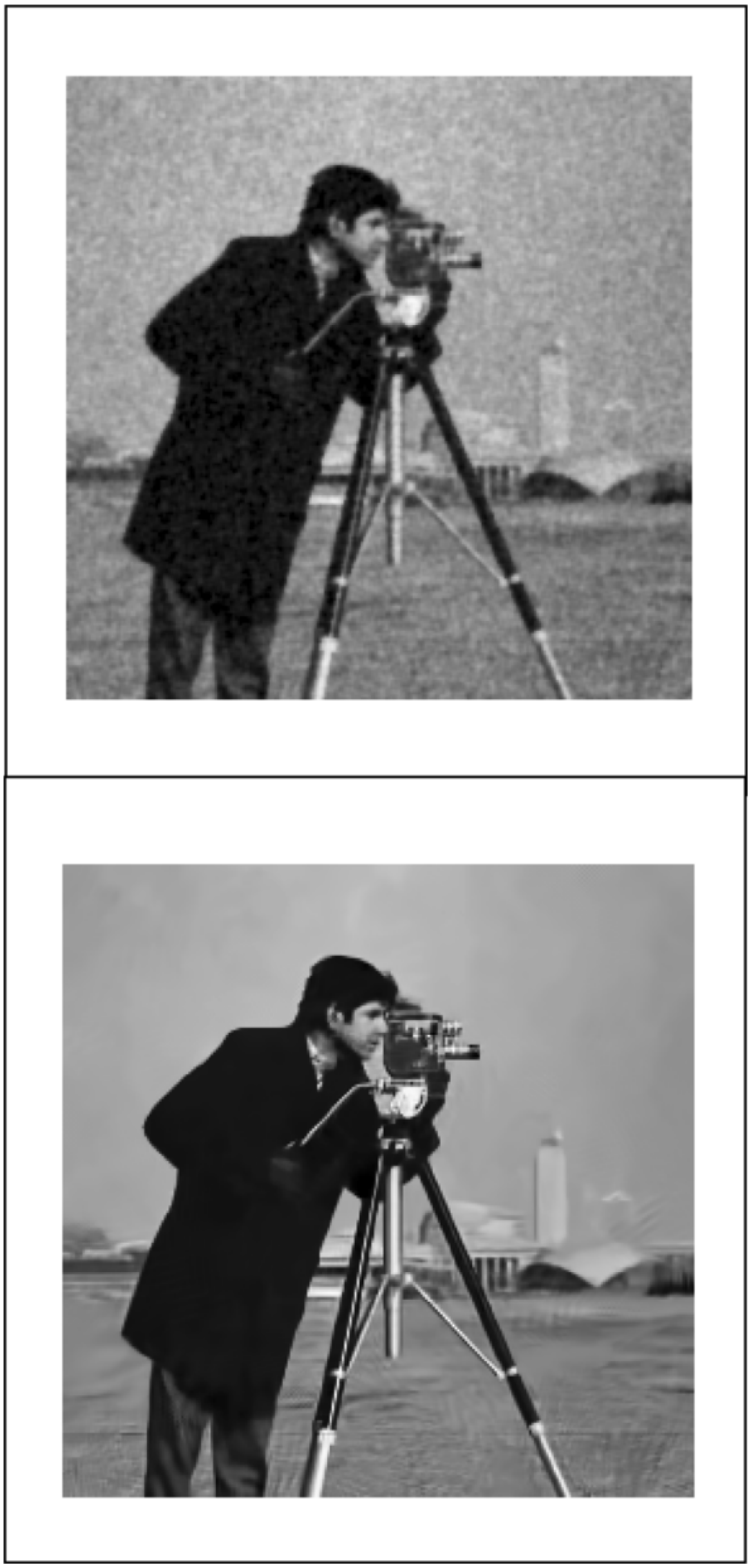


**Figure 16** Results of our method (up) and BM3D technique (down) with SNR = 25 for Cameraman.

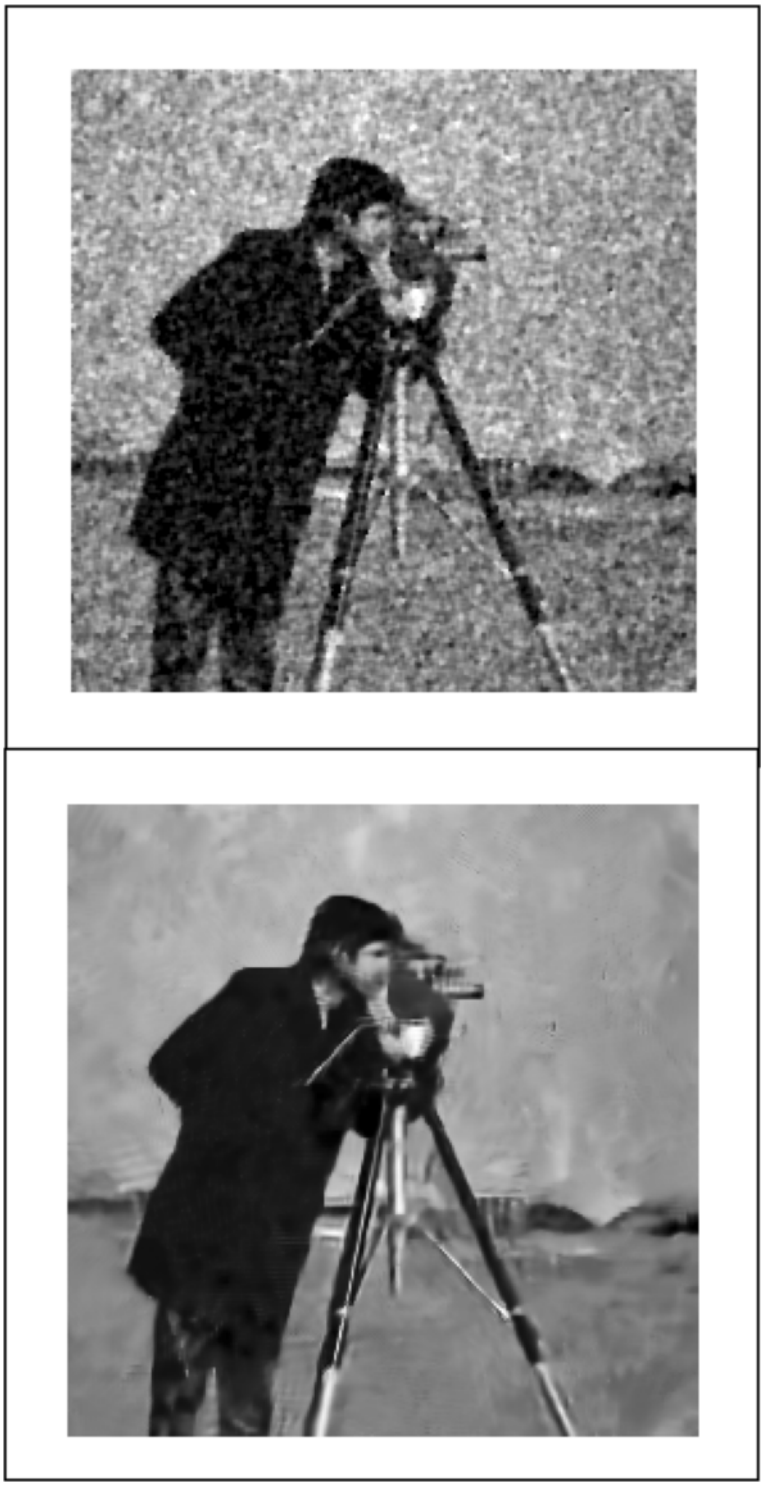


**Figure 17** Results of our method (up) and BM3D technique (down) with SNR = 75 for Cameraman.

Then $\frac{\partial J}{\partial u}$ can be simplified to:

$$\frac{\partial J}{\partial u} = \iint 2(u{-}u0){-}4\lambda \iint H_1\ H_2 Z_1\ Z_2 \left[ \left(g_x + g_{x+\Delta x}\right) \left( \frac{u_{xx}}{u_x} + \frac{u_{xy}}{u_y} \right) + \left(g_y + g_{y+\Delta y}\right) \left( \frac{u_{yy}}{u_y} + \frac{u_{yx}}{u_x} ] \right] = 0 \tag{42}$$

After simplification, the result is:

$$\begin{aligned} u &= u_0 + 4\lambda\, SWN \left( \nabla u_x\, \gamma_x + \nabla u_y\, \gamma_y \right) \\ &= u_0 + 4\lambda\, SWN \left( \nabla^2 u\ + \nabla u^2 \frac{u_{xy}}{u_x u_y} \right) \end{aligned} \tag{43}$$

Because $u_x u_y$ in the second part of the equation may create zero in the denominator, we write:

$$\nabla u^2 \frac{u_{xy}}{u_x u_y} = {u_x}^2 \frac{u_{xy}}{u_x u_y} + {u_y}^2 \frac{u_{xy}}{u_x u_y} \tag{44}$$

Using partial differential relations we have:

$${u_x}^2 \frac{u_{xy}}{u_x u_y} = \frac{\left(\frac{\partial u}{\partial x}\right)^2}{\frac{\partial u\ \partial u}{\partial x \partial y}}\ \frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial x^2} = u_{xx} \tag{45}$$

Similarly for y component:

$${u_y}^2 \frac{u_{xy}}{u_x u_y} = \frac{\left(\frac{\partial u}{\partial y}\right)^2}{\frac{\partial u\ \partial u}{\partial x \partial y}}\ \frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial y^2} = u_{yy} \tag{46}$$

The final equation is:

$$u = u_0 + 2\lambda\, SWN\ \ \nabla^2 u\ = u_0 + \lambda\, f(u) \tag{47}$$

### Competing interests

Half of this work is supported by USM graduate assistance grant of School of Electrical and Electronic Engineering, Universiti Sains Malaysia.

### Authors' contributions

AR, AHAH, MJ and BR carried out the image denoising studies, participated in the sequence alignment, edited the paper and drafted the manuscript. All authors read and approved the final manuscript.

### Authors' information

**Ali Ranjbaran** received the B.Tech. degree in Electronic Engineering from Sistan Baloochestan University, Iran, in 1989, and M.Sc. degree in Electronic Engineering from Shiraz University, Iran, in 2000 and Ph.D. degree in Robotic Vision from Universiti Sains Malaysia, Penang, Malaysia in 2013. His research interests include image and signal processing.

**Anwar Hasni Abu Hassan** received the B.S. degree in Quality Control and Instrumentation from Universiti Sains Malaysia, Penang, Malaysia, in 1993, and M.Sc. and Ph.D. degree in Mechatronics and Electronic Engineering from University of Hull, United Kingdom, in 1995 and 2002. Currently a lecturer of Electrical and Electronic Engineering, USM, his research interests include mobile robot navigation and vision-based techniques.

**Mahboobe Jafarpour** received the B.Tech. degree in Computer Engineering from Sharif University of technology, Tehran, Iran, in 1994. Her research interest is machine vision.

**Bahar Ranjbaran** is a university student studying in the field of Electronic Engineering. Her research interests include robotics, mobile robot navigation and vision-based techniques.

Received: 23 December 2014 Accepted: 22 January 2015
Published online: 08 March 2015

### References

Acharya T, Tsai PS (2001) Edge-Detection Based Noise Removal Algorithm, US Patent 6,229,578

Buades A, Coll B, Morel JM (2005) A review of image denoising algorithms, with a new one. Multiscale Model Simul 4(2):490–530

Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, Information Theory. IEEE Trans Image Process 52:489–509

Chan TF, Esedo Lu S (2005) Aspects of total variation regularized L1 function approximation. SIAM J Appl Math 65(5):1817–1837

Chen Q, Wu D (2010) Image denoising by bounded block matching and 3D filtering. Signal Process 90(9):2778–2783

Dabov K, Foi A, Katkovnik V, & Egiazarian K (2006). Image denoising with block-matching and 3D filtering. In Electronic Imaging 2006. International Society for Optics and Photonics.

Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095

Dabov K, Foi A, Katkovnik V, Egiazarian K (2008) A Nonlocal and Shape-Adaptive Transform-Domain Collaborative Filtering. In: Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA

Dabov K, Foi A, Katkovnik V, Egiazarian K (2009) BM3D Image Denoising With Shape-Adaptive Principal Component Analysis. In: SPARS'09-Signal Processing with Adaptive Sparse Structured Representations

Goldstein T, Osher S (2008) "The split Bregman Method for L1 Regularized Problems," UCLA CAM Report, vol. 2., p 4

He K, Sun J, Tang X (2010) "Guided Image Filtering," The 11th European Conference on Computer Vision, Part I., pp 1–14

Kervrann C, Bourlanger J, Bourlanger J (2006) Optimal spatial adaptation for patchbased image denoising. IEEE Trans Image Process 15:10

Li Y, Santosa F, Center CT (1994) "An Affine Scaling Algorithm For Minimizing Total Variation In Image Enhancement" Vol. 201:Citeseer

Petschnigg G, Agrawala M, Hoppe H, Szeliski R, Cohen M, Toyama MK (2004) Digital photography with flash and no-flash image pairs. ACM Trans Graph 23:664–672

Ranjbaran A, Hassan AHA, Kheng ES, Ranjbaran B (2013) An edge detection based noise removal algorithm. Espec Issue Image Video Process 40:769–784

Rudin LI, Osher S, Fatemi E (1992) "Nonlinear Total Variation Based Noise Removal Algorithms," Physica D: Nonlinear Phenomena, vol. 60., pp 259–268

Takeda H, Farsiu S, Milanfar P (2007) Kernel regression for image processing and reconstruction, IEEE Trans Image Process 16(2):349–366