

# An Efficient Implementation of Neighborhood based Wavelet Thresholding For Image Denoising

Sabahaldin A. Hussain

Electrical & Electronic Eng. Department  
University of Omdurman Sudan

Sami M. Gorashi

Electrical & Electronic Eng. Department  
University of Omdurman  
Sudan

## ABSTRACT

In this paper we propose computationally efficient denoising algorithm that thresholds the wavelet coefficient considering its neighbors in deciding whether it is noisy or noise free. The proposed algorithm select a suboptimal threshold and neighboring window size for every subband that minimized Mean Square Error(MSE) in the denoised image using Stein's Unbiased Risk Estimate(SURE). In this paper, we demonstrate the efficiency of the proposed denoising algorithm as compared with two other state-of-the art denoising algorithms.

## General Terms

Image processing.

## Keywords

Image denoising, Wavelet transform, Neighborhood.

## 1. INTRODUCTION

Digital images are generally affected by different types of noise. A noise is introduced in the transmission medium due to a noisy channel, imperfect instruments used in image processing, errors during the measurement process, noise due to degradation such in films, image compression and during quantization of the data for digital storage. Each element in the imaging chain such as lenses, film, digitizer, etc. contributes to the degradation. Thus, image denoising is a necessary and primary step in any further image processing tasks like segmentation, object recognition, computer vision, ...etc. To overcome image data corruption, we need to know something about the degradation process in order to develop a model for it. When we have a model for the degradation process, the inverse process can be applied to the image to restore it back to the original form. Noise modeling in images is greatly affected by capturing instruments, image quantization, data transmission channels, ...etc. Among several denoising algorithms, denoising that based on wavelet transform for cancelling white Gaussian noise finds wide range of applications since the pioneer work by Donoho and Johnstone[1]. VisuShrink[1], SureShrink[2], BayesShrink[3], SURE-LET[4], NUP/Wavelet transform[5], and Neural Network based thresholding[6] are examples of thresholding the wavelet detail coefficients term by term based on their individual magnitudes. Thereafter, several alternative denoising algorithms have been proposed by many authors that based on utilizing the information about the neighbor wavelet coefficients (block of coefficients) rather than individually. Including wavelet coefficient neighbors increases the estimation precision in deciding whether the coefficient is noisy or noise free. Cai and Silverman[7], proposed NeighCoeff for 1-D signals. Chen et al[8], proposed NeighShrink that extend NeighCoeff to image denoising. The main drawbacks of NeighShrink is that it uses a fixed large (due to its dependence on the number of image samples)

universal threshold value and identical neighboring window size in all wavelet subbands which may result in overly smooth images. To overcome NeighShrink weakness, Z. Dngwen, and C. Wengang[9] proposed SURENeighShrink that search for optimum threshold value and neighboring window size for every subband according to Stein's Unbiased Risk Estimate(SURE) method. It has been shown that the denoising performance of SURENeighShrink is considerably superior to NeighShrink and also outperforms SURE-LET[4] which is one of the best term-wise denoising algorithm that also based on SURE. The main drawback of the SURENeighShrink is its high computation cost when searching for optimum threshold value and neighboring window size for every wavelet subband. In this paper, we proposed an efficient implementation of SURENeighShrink that overcomes its computation complexity. The experimental results show that the proposed denoising algorithm achieves comparable denoising performance over the wide range of images and noise levels. Results also show that the proposed method outperforms NeighShrink considerably.

The rest of the paper is organized as follows. In section 2, we explain the proposed image denoising algorithm. The results of our proposed denoising algorithm will be compared with NeighShrink[8] and SURENeighShrink[9] in section 3. Finally, the concluding remarks are given in section 4.

## 2. PROPOSED ALGORITHM

The SURENeighShrink[9] denoising algorithm produces both higher PSNRs and better visual quality than some published best denoising algorithms. The main drawback of this denoising algorithm is its computation overhead when searching for optimum threshold value and neighboring window size for every wavelet subband . To overcome the computation overhead, we have experimented with SURENeighShrink denoising algorithm using various standard images over a wide range of noise levels including the effect of both wavelet name and number of decomposition levels. The results are reported which include the optimum threshold value and neighboring window size for the different standard images under test, different wavelet names, and different wavelet decomposition numbers. Results of extensive statistical work show that:-

1. The neighboring window size is directly proportional with the noise level. That is,  
$$L \propto \sigma_n \quad (1)$$

This means that larger neighboring window size is required to gather more information about the neighbors of the wavelet coefficient which in turn enables deciding whether the coefficient is noisy or noise free. This point explains the weakness of NeighShrink when denoising highly corrupted images because it uses fix window size for all noise levels.

2. For low noise levels,  $\sigma_n \leq 10$  , single wavelet decomposition is quite enough. The window size ranges from 3 to 11.
3. Larger window size is required for the diagonal subbands mostly for the first and second wavelet decomposition.
4. The optimum window size for the third and fourth decomposition levels ranges from 1 to 3.
5. Over a wide range of images sizes corrupted with high noise levels, the optimal window size rarely exceeds 13.
6. Extensive experiments have been carried out to study, statistically, the relation between the optimal threshold value and the corresponding optimal window size for every wavelet subband. Results show that, there is a direct relationship between them. That is,

$$T \propto L \quad (2)$$

From this statistical study, we create a set of arrays that contain the optimal window size for each decomposition level and for each orientation in each subband namely HL, LH, and HH (Horizontal, Vertical, and Diagonal). These arrays cover the noise levels ranges from 10 to 100. We named each array as window size kernel. Table[1], shows these suggested window kernels for different orientation and noise levels. The selection mechanism for neighboring window size in each orientation for every wavelet subband can be state

$$\text{Select } L \text{ from } \begin{cases} \text{Window Kernel}_1 & \text{if } 0 < \hat{\sigma}_n \leq 10 \\ \text{Window Kernel}_2 & \text{if } 10 < \hat{\sigma}_n \leq 20 \\ \text{Window Kernel}_3 & \text{if } 20 < \hat{\sigma}_n \leq 30 \\ \text{Window Kernel}_4 & \text{if } 30 < \hat{\sigma}_n \leq 50 \\ \text{Window Kernel}_5 & \text{if } 50 < \hat{\sigma}_n \leq 75 \\ \text{Window Kernel}_6 & \text{if } 75 < \hat{\sigma}_n \leq 100 \end{cases} \quad (3)$$

Regarding threshold value in each subband, we suggest to use the following empirical formula,

$$T = 2 \times L + 1 \quad (4)$$

7. The wavelet coefficient is shrunk or suppressed according to

$$\hat{v}_{i,j} = v_{i,j} \times \max \left\{ 1 - \left( \frac{T^2}{\sum_{x,y \in B_{i,j}} v_{x,y}^2} \right), 0 \right\} \quad (5)$$

Where B is a window of size  $L \times L$  in subband S centered at  $(i, j)$ ,  $v_{i,j}$  is the wavelet coefficient to be shrunk or suppressed, T is a threshold value for subband S, and  $\hat{v}_{i,j}$  is the estimated noise free wavelet coefficient. According to the above steps, the proposed denoising algorithm, select from a predefined data base the suboptimal neighboring window size and threshold value for each orientation in each subband depending upon the level of estimated noise. Figure(1), shows the flow chart that describes the implementation steps of the newly proposed denoising algorithm.

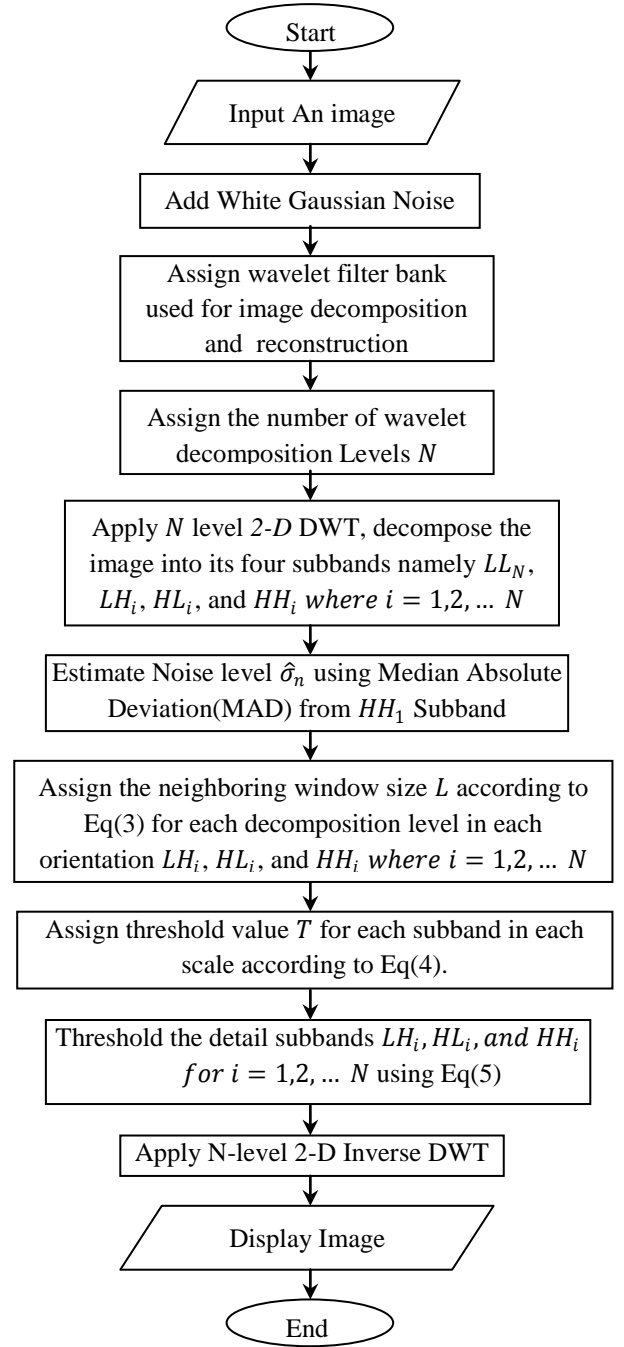


Fig 1: The Flowchart of The DBNeighShrink

**Table 1. Window size kernels for different noise levels**

Window Size Kernel <sub>1</sub> ( $\sigma_n = 10$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	3	1	1	1
LH	3	1	1	1
HH	5	1	1	1
Window Size Kernel <sub>2</sub> ( $\sigma_n = 20$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	5	3	1	1
LH	5	1	1	1
HH	7	3	1	1
Window Size Kernel <sub>3</sub> ( $\sigma_n = 30$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	9	3	1	1
LH	7	3	1	1
HH	9	5	1	1
Window Size Kernel <sub>4</sub> ( $\sigma_n = 50$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	11	5	1	1
LH	7	3	1	1
HH	13	9	1	1
Window Size Kernel <sub>5</sub> ( $\sigma_n = 75$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	11	5	3	1
LH	9	3	1	1
HH	13	9	1	1
Window Size Kernel <sub>6</sub> ( $\sigma_n = 100$ )				
Decomposition Level \ Subband Orientation	1	2	3	4
HL	11	5	3	1
LH	7	3	3	1
HH	13	13	1	1

**4. RESULTS AND DISCUSSIONS**

An experiment was conducted to assess the performance of the proposed Data Based NeighShrink (we shall call it DBNeighShrink) denoising algorithm for denoising images corrupted with Gaussian noise with zero mean and standard deviations 10, 20, 30, 50, 75, and 100. The wavelet transform employs Daubechies least asymmetric compactly supported wavelet with eight vanishing moments at four scales of decomposition. The noise standard deviation is estimated using robust Median Absolute Deviation (MAD) using the highest level wavelet coefficients. We shall use the Peak Signal to Noise Ratio(PSNR) as our measure of the relative denoising algorithms performance.

In this experiment, we have compared our proposed denoising algorithm with the conventional NeighShrink[8], and SURENeighShrink[9]. NeighShrink uses the universal threshold  $T = \sigma_n \sqrt{2 \log_2(M)}$  where M is the image size with fixed  $3 \times 3$  neighboring window size for all subbands as recommended by [8]. SURENeighShrink uses an optimal threshold and neighboring window size for every subband that minimize MSE in denoised image using Stein’s Unbiased Risk Estimate (SURE).

The PSNR from various denoising algorithms are compared in Table(2). The data are collected from an average of ten runs. The best denoising algorithm among others in terms of PSNR value is highlighted in bold font for each test image.

As can be seen from these tables, the average output PSNR of SURENeighShrink and DBNeighShrink are close to each other for almost all images under test. Thus, Both SURENeighShrink and DBNeighShrink are robust, they achieve nearly equivalent performance in terms of average PSNR over all noise levels for various images under test.

The NeighShrink isn’t robust in terms of both average PSNR value and visual quality as compared with others. Its results become worse when the noise level is increased. This denotes that it isn’t appropriate for high noise level for NeighShrink to use the fixed window size and single universal threshold in all wavelet subbands. It produces overly smoothed images with no sufficient edge preservation which in turn makes the image get blurry and loses much of its resolution.

The proposed DBNeighShrink outperforms NeighShrink over all noise levels and for all test images. This is due to the fact that DBNeighShrink overcomes the drawbacks of NeighShrink which uses fix universal threshold value and neighboring window size for all subbands. While, DBNeighShrink uses different neighboring window size and threshold value for each subband in each scale. As an example for Lena Image, DBNeighShrink, achieves an average PSNR gain of 1.476 dB as compared with NeighShrink. While, SURENeighShrink achieves an average PSNR gain of just 0.112 dB as compared with DBNeighShrink. However, SURENeighShrink requires much processing time compared with DBNeighShrink. This is due to the fact that SURENeighShrink search for optimal window size and threshold value for every wavelet subband by minimizing Stein’s unbiased risk estimate while DBNeighShrink uses a predefined data base regarding suboptimal window size and threshold value as a function of noise intensity level which in turn makes it computationally more efficient as compared with SURENeighShrink.

As an example, we compare the computational complexity of NeighShrink, SURENeighShrink, and DBNeighShrink using CPU run time. Although CPU time is not an exact measure, it gives a rough estimation of complexity. All the experiments were done with an Intel Dual-Core processor at 2.6GHz with 2GB of RAM, using MATLAB R2008b environment under Microsoft Windows XP SP2 operating system. The average execution time of ten runs for each denoising algorithm was recorded for Lena image of size  $512 \times 512$ . Results show that, the average execution time of SURENeighShrink is about 27.352 seconds while DBNeighShrink requires just 2.761 seconds. On the other hand, NeighShrink, requires about 2.364 seconds. Thus, by taking the computational complexity into our considerations, DBNeighShrink exhibits both good performance and low computational complexity as compared with NeighShrink and SURENeighShrink.

Finally, it is important to examine some denoised images visually. Figure(2), shows that for low noise level degradation ( $\sigma_n \leq 10$ ), almost all denoising algorithms achieve nearly equivalent performance although NeighShrink exhibits less PSNR value. Figure(3) and figure(4), show the effect of denoising for moderate noise level. Noticeably, the images in figure(3-b) and figure(4-b) are little over smoothed which is the nature of NeighShrink as discussed before. While, SURENeighShrink and DBNeighShrink denoising algorithms provide better edges preservation and exhibit nearly equivalent visual quality. Figure(5) and figure(6), show the effect of denoising for high level of noise degradation. Clearly, NeighShrink is the worse denoising algorithm compared with others. It is overly smooth the image and destroying most of denoised image details as can be seen in figure(5-b) and more clearly in the zoom-in of the window edge in figure(6-b). Therefore NeighShrink is not suitable especially for high noise levels. While, all other denoising algorithms achieve better denoising and edge preservation in terms of both PSNR value and visual perception.

To summarize, referring to the results in Table(2), we can see that SURENeighShrink outperforms other denoising algorithms most of the time but at the expense of extremely high processing demands. SURENeighShrink and DBNeighShrink achieve comparable denoising performance in terms of PSNR value as well as in terms of visual quality.

## 5. CONCLUSIONS

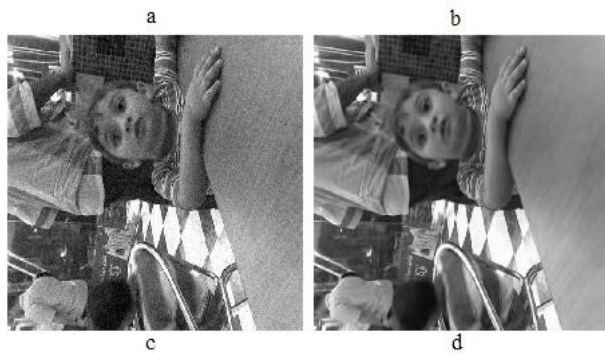
The NeighShrink[8] uses a fixed suboptimal universal threshold value and identical neighboring window size in all wavelet subbands which results in overly smooth images. In this paper, we improve NeighShrink through selecting different suboptimal threshold and neighboring window size for every subband that minimized Stein's Unbiased Risk Estimate(SURE). Experimental results indicate that the proposed DBNeighShrink outperforms NeighShrink over all noise levels and for all test images. On the other hand, SURENeighShrink and DBNeighShrink achieve comparable denoising performance in terms of PSNR value as well as in terms of visual quality. Thus, we conclude that the proposed DBNeighShrink exhibits both good performance and low computational complexity as compared with NeighShrink and SURENeighShrink.

## 6. REFERENCES

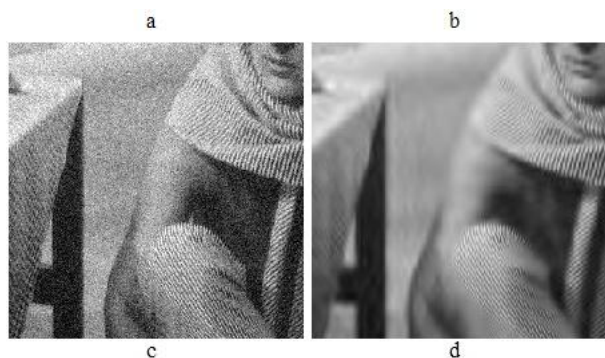
- [1] D. L. Donoho and Iain M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, 81, pp. 425-455, 1994.
- [2] D. L. Donoho, Iain M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage", *Journal of American Statistical Association*, 90(432), pp.1200-1224, 1995.
- [3] S. G. Chang, B. Yu and M. Vetterli, "Adaptive Wavelet Thresholding for Image Denoising and Compression", *IEEE Transactions on Image Processing*, Vol. 9, No. 9, pp. 1532-1546, 2000.
- [4] F. Luisier, T. Blu, M. Unser, "A New SURE Approach to Image Denoising: Interscale Orthonormal Wavelet Thresholding", *IEEE Transactions on Image Processing*, Vol. 16, No. 3, pp. 593-606, 2007.
- [5] R. Song, Y. Li, Q. Zhang, and Z. Zhao, "Image denoising method based on non-uniform partition and wavelet transform", *International Congress on Image and Signal Processing(CISP)*, pp.703–706, 2010.
- [6] Y. Norouzzadeh and M. Rashidi, "Image denoising in wavelet domain using a new thresholding function", *International Conference on Information Science and Technology*, pp.721–724, 2011.
- [7] T. T. Cai and B. W. Silverman, "Incorporating information on neighboring coefficients into wavelet estimation", *Sankhya A* 63 pp.127–148, 2001.
- [8] G.Y. Chen, T.D. Bui, A. Krzyzak, "Image denoising using neighbouring wavelet coefficients", *IEEE Transactions on signal Processing*, Vol. 10, No. 10, pp. 917-920, 2004.
- [9] Z. Dengwen, C. Wengang, "Image denoising with an optimal threshold and neighboring window", *Pattern Recognition Letters*, 29, pp.1694-1697, 2008.

	NeighShrink	SURENeighShrink	DBNeighShrink
<b>Lena Image</b>			
$\sigma_n = 10$	32.822	34.626	<b>34.645</b>
$\sigma_n = 20$	29.648	<b>31.461</b>	31.416
$\sigma_n = 30$	27.741	<b>29.599</b>	29.492
$\sigma_n = 50$	25.300	<b>27.072</b>	26.898
$\sigma_n = 75$	23.284	<b>24.601</b>	24.449
$\sigma_n = 100$	21.752	<b>22.716</b>	22.506
Average PSNR	26.758	<b>28.346</b>	28.234
<b>Barbara Image</b>			
$\sigma_n = 10$	30.436	32.783	<b>32.789</b>
$\sigma_n = 20$	26.685	<b>29.033</b>	28.955
$\sigma_n = 30$	24.692	<b>26.978</b>	26.920
$\sigma_n = 50$	22.431	<b>24.452</b>	24.381
$\sigma_n = 75$	20.919	<b>22.273</b>	22.228
$\sigma_n = 100$	19.727	<b>20.640</b>	20.564
Average PSNR	24.148	<b>26.027</b>	25.973
<b>Goldhill Image</b>			
$\sigma_n = 10$	30.206	32.606	<b>32.638</b>
$\sigma_n = 20$	27.588	<b>29.547</b>	29.541
$\sigma_n = 30$	26.241	<b>27.924</b>	27.866
$\sigma_n = 50$	24.548	<b>25.761</b>	25.723
$\sigma_n = 75$	22.982	<b>23.678</b>	23.504
$\sigma_n = 100$	21.498	<b>22.043</b>	21.832
Average PSNR	25.511	<b>26.927</b>	26.834

**Table 2. PSNR Results for Denoising Lena, Barbara, and Goldhill**



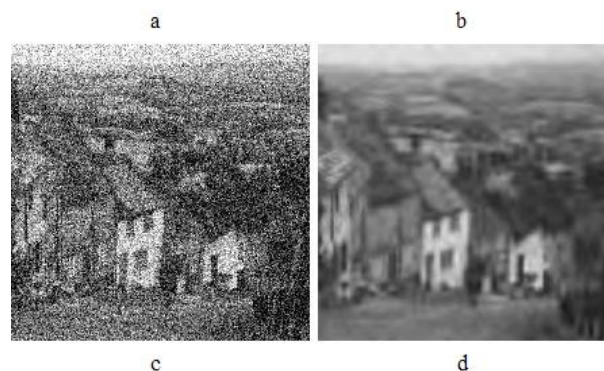
**Fig 2: Denoising results for Child:** (a) Noisy image PSNR=28.196dB ( $\sigma_n=10$ ); (b) Denoised image using NeighShrink, PSNR=32.029dB; (c) Denoised image Using SURENeighShrink PSNR=34.059; (d) Denoised image using DBNeighShrink, PSNR=33.594.



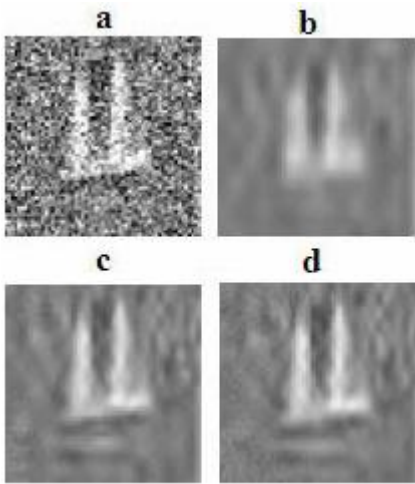
**Fig 3: Denoising results for Barbara:** (a) Noisy image PSNR=22.176 ( $\sigma_n=20$ ); (b) Denoised image using NeighShrink, PSNR=26.662; (c) Denoised image using SURENeighShrink, PSNR=29.022; (d) Denoised image using DBNeighShrink, PSNR=28.875.



**Fig 4: Denoising results for Lena:** (a) Noisy Image, PSNR=18.711dB( $\sigma_n=30$ ); (b) Denoised image Using NeighShrink, PSNR=27.720 dB; (c) Denoised image Using SURENeighShrink, PSNR=29.596 dB; (d) Denoised image Using DBNeighShrink, PSNR=29.485 dB.



**Fig 5: Denoising results for Goldhill:** (a) Noisy Image, PSNR=14.622 dB ( $\sigma_n=50$ ); (b) Denoised image Using NeighShrink, PSNR=24.586 dB; (c) Denoised image Using SURENeighShrink, PSNR=25.766 dB; (d) Denoised image Using DBNeighShrink, PSNR=25.747 dB.



**Fig 6: Denoising results for House: (a) Noisy Image, PSNR=14.576 dB ( $\sigma_n=50$ ); (b) Denoised image Using NeighShrink, PSNR=25.163 dB; (c) Denoised image Using SURENeighShrink, PSNR=26.340 dB; (d) Denoised image Using DBNeighShrink, PSNR=26.312 dB.**