IMAGE DENOISING USING MODIFIED NEIGHSHRINK ALGORITHM

Indranil Sinharoy
Student ID# 21623541
Department of Electrical Engineering,
Southern Methodist University,
Dallas, Texas – 75205.
Email: isinhroy@smu.edu

ABSTRACT

Image de-noising, using wavelet techniques are very effective because of its ability to capture the energy of a signal in a few high-energy transform values, when the natural image is corrupted by Gaussian noise [3]. Additive random noise can easily be removed using simple Threshold methods. G.Y Chen, T.D. Bui and A. Krzyzak proposed a method called *NeighShrink* that use local information to shrink the wavelet coefficients [1]. However, the *NeighShrink* algorithm, as with most other de-noising algorithms, assumes prior knowledge of the noise variance. This paper investigates two aspects of the *NeighShrink* de-noising scheme – firstly a modified *NeighShrink* scheme is proposed that tries to preserve more of the edge information than the *NeighShrink* method, and secondly the influence of blind noise estimation on the performance of the modified *NeighShrink* method, called the *ModiNeighShrink*.

1. NeighShrink: INCORPORATING NEIGHBORING WAVELET COEFFICIENTS IN IMAGE DE-NOISING

Performing 1-level, 2D wavelet transform on an image, **f** decomposes the original image into four sub-images, having four different frequency sub-bands, namely, LL, LH, HL and HH [1]. These are also known by other names, for example, the sub-bands may be respectively called $\mathbf{a}^1$ or the first-averaged image (It is the lower resolution version of the image **f**), $\mathbf{h}^1$ called horizontal fluctuation, $\mathbf{v}^1$ called the vertical fluctuation, and $\mathbf{d}^1$ called the first diagonal fluctuation [3]. The sub-image $\mathbf{a}^1$, is formed by computing trends along rows of the image **f** followed by computing trends along its columns. The sub-image $\mathbf{h}^1$ is formed by computing trends along rows followed by computing fluctuations along columns. $\mathbf{v}^1$ is created by computing trends along columns followed by computing fluctuations along rows. The sub-image $\mathbf{d}^1$ is created by taking fluctuations along both rows and columns.

The next level of wavelet transform is applied to the low frequency sub band image LL ($\mathbf{a}^1$) only. This process is repeated until a pre-specified level is reached. The Gaussian noise will nearly be averaged out in the low frequency wavelet coefficients. Therefore only the wavelet coefficients in the high frequency levels need to be thresholded.

The wavelet de-noising scheme using neighboring wavelet coefficients is implemented in the following way:

Let $d_{j,k}$ denote the wavelet coefficients of interest
$B_{j,k}$ is a neighborhood window around $d_{j,k}$

$$\text{Also let } S^2_{j,k} = \sum_{(i,l)\, B_{j,k}} d^2_{i,l}$$

Then the wavelet coefficient to be thresholded is shrinked according to the formula:

$$d_{j,k} = d_{j,k}\, B_{j,k}$$

where, the shrinkage factor can be defined as:

$$B_{j,k} = (1 - T^2 / S^2_{j,k})_+$$

here, the + sign at the end of the formula means to keep the positive value while set it to zero when it is negative, and $T = \sigma*\text{sqrt}(2\log(n^2))$. This is a slightly modified version of Donoho's soft-threshold technique.

The de-noised image is then obtained by performing inverse 2D wavelet transform on the thresholded wavelet coefficients.

The neighborhood wavelet image denoising algorithm is as follows:

1. Perform forward 2D wavelet decomposition on the noisy image.
2. Apply the proposed shrinkage scheme to threshold the wavelet coefficients using a neighbourhood window $B_{j,k}$ and the universal threshold, $T = \sigma*\text{sqrt}(2\log(n^2))$.
3. Perform 2D wavelet transform on the thresholded wavelet coefficients.

2. PROPOSED MODIFICATIONS TO *NeighShrink* DENOISING ALGORITHM

During experimentation it was seen that when the noise content was high, the reconstructed image using *NeighShrink* contained "mat-like" aberrations. These aberrations could be removed considerably by Wiener filtering the reconstructed image at the last stage of IDWT. The cost of this additional filtering was slight reduction in sharpness of the reconstructed image. However there was a slight improvement in the PSNR of the reconstructed image using Wiener filtering. The following figures compare the effect of Wiener filtering:

The white oval shaped ring show in particular the areas of "mat" formations.

Figure 2.1: Reconstruction without Wiener Filtering

This figure shows that the mat-like aberrations are removed at the cost of slight reduction in image sharpness. The black rings show the areas where the sharpness was lost.

Figure 2.2: Reconstruction with Wiener Filtering

It was also observed that although the *NeighShrink* algorithm gave better results in terms of PSNR, than other traditional methods, the reconstructed images were sometimes unacceptably blurred and lost some image details. One reason for this could be because the algorithm suppressed too many detail wavelet

coefficients. One way to circumvent this problem was to reduce the value of threshold itself. Experimenting on a set of images and on trial-and-error basis, it was found that reducing the threshold at each level by one-fourth, resulted in better image quality, both qualitatively (visual) and quantitatively (PSNR value). Figure 2.3(a) shows the detail coefficients at the first level of wavelet decomposition. Figure 2.3(b) shows the coefficients at the same level after thresholding using *NeighShrink* algorithm. Figure 2.3(c) shows the detail coefficients at the same level of wavelet decomposition using *modified NeighShrink* algorithm. That is by replacing T by ¾*T in the equation $B_{j,k} = (1- T^2/ S^2_{j,k})_+$

Thus the modified shrinkage factor becomes:

$$B_{j,k} = [1- (3/4)*T^2/ S^2_{j,k}]_+$$

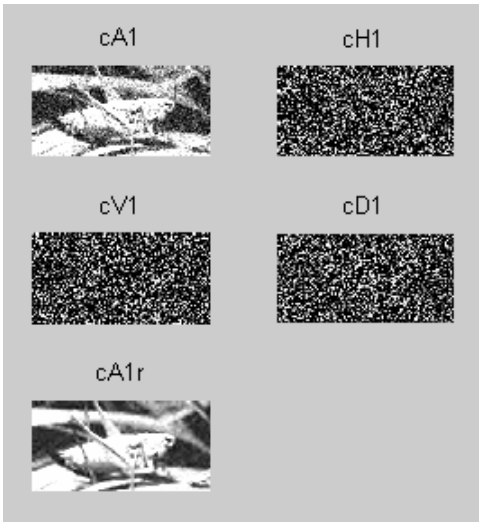

Figure 2.3(a): Approximation and detail coefficients at level 1.

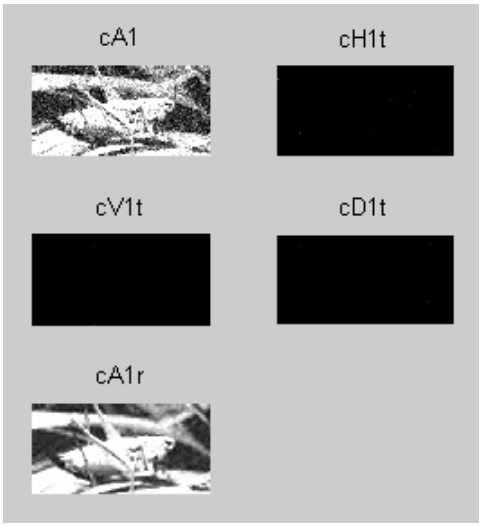

Figure 2.3(b): Approximation and detail coefficients at level 1, after thresholding using *NeighShrink* algorithm.

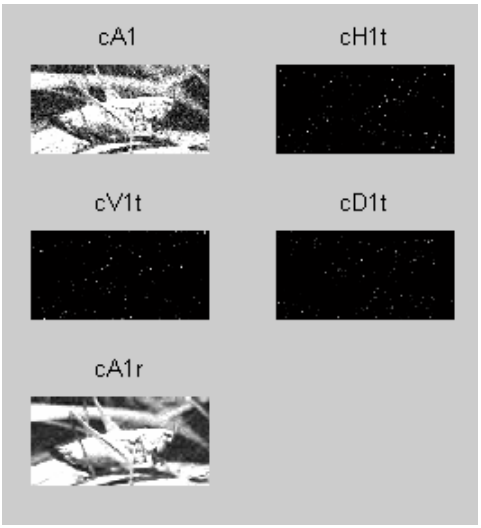

Figure 2.3(c): Approximation and detail coefficients at level 1, after thresholding using the modified *NeighShrink* algorithm. We can see that there are some residual wavelet coefficients left after thresholding. This happens at all the levels thus allowing some high frequency wavelet coefficients to remain after thresholding. These coefficients contain edge information in the original image.

Table 2.1 compares the gain (in dB) of the de-noised images using *NeighShrink* scheme with that obtained using Modified *NeighShrink* scheme. The gain is defined to be the difference in PSNR of the de-noised image and the noisy image. It is evident that the modified scheme produced de-noised image with higher PSNR. Since no image processing algorithm testing is complete without subjective tests, figure 2.4 compares de-noised images obtained by the two methods.

Table 2.1

Image	**Noise Variance added to image**	**(PSNR of reconstructed image – PSNR of noisy image) in dB**	
		NeighShrink	**Modified *NeighShrink***
Insect	0.02	7.7993	**8. 5364**
Insect	0.04	8.4918	**8. 9809**
Insect	0.06	8.7364	**9. 2108**
Insect	0.08	8.8008	**9. 1448**
Lena	0.02	10.3316	**11. 2563**
Lena	0.04	11.4527	**12. 0812**
CT of Head	0.01	10.3917	**11. 3368**
CT of Head	0.02	11.5002	**12. 2766**
Galaxy	0.02	13.1097	**13. 1965**

Figure 2.4(a): Original, Noisy and De-noised image of an MRI of spine using Modified *NeighShrink* algorithm. Value of Gaussian Noise variance added is 0.04. The gain in dB of the de-noised image was 11.186 dB. Sym16 was the analyzing wavelet.

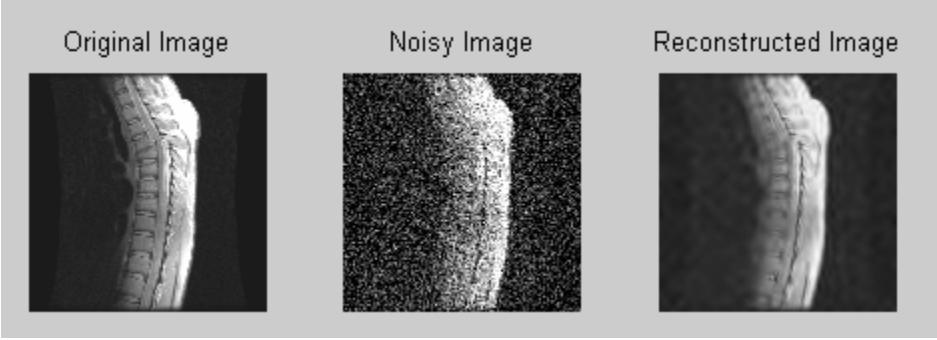

Figure 2.4(b): Original, Noisy and De-noised image of an MRI of spine using *NeighShrink* algorithm. Value of Gaussian Noise variance added is 0.04. The gain in dB of the de-noised image was 10.8748 dB. Sym16 was the analyzing wavelet.

It is quite evident from the above figure that the de-noised image produced by the modified *NeighShrink* scheme, *ModiNeighShrink*, retained edge information better than the *NeighShrink* scheme. It was found that *ModiNeighShrink* always produced higher PSNR in the de-noised image than NeighShrink.

3. CHOICE OF PRIMARY RESOLUTION, NUMBER OF DECOMPOSITION LEVELS AND ANALYZING WAVELET

3.1 CHOICE OF ANALYZING WAVELET

The success of shrinkage procedures depend largely upon the choice of primary resolution (the scale level at which to begin thresholding) and choice of analyzing wavelet [2]. David Donoho and Ian Johnstone had proved mathematically that if a certain kind of orthogonal basis existed, then it would do the best possible job of extracting a signal from white noise. For a large class of signals (images) an orthogonal wavelet transform compresses the "energy" of the signal in a relatively small number of large coefficients. Thus the orthogonal wavelet transform compacts the energy of the signal into a small space in transform domain. But the energy of the white noise remains dispersed throughout, resulting in relatively small coefficients. These coefficients can be removed using a threshold. In the words of Barbara Bruke Hubbard, while noise masks the signal in "position space", the two becomes disentangled in "wavelet space" [4].

Table 3.1 is part of a more detailed tabulation obtained by performing wavelet de-noising by different analyzing wavelets on a set of images of various classes. Three levels of wavelet decomposition were used. The results reflect that the sym16 analyzing wavelet performed better than the rest in most occasions. One more important point to observe is that the de-noising algorithm performed particularly well for class of images that did not contain too many edges (high frequency components) spread all over the image area for example the gain in dB is much low for the *MRI knee* image and the *crowd* image than that of the *lena* and *insect* image. The *lena* and *insect* images are typical examples of images that have details mostly in a particular area in the image space.

Table 3.1

Image	Noise variance	Gain* in dB for different choice of Analyzing wavelet							
		db4	db8	db16	sym4	sym8	sym16	coif4	coif5
lena	0.02	11.24	11.26	11.05	11.25	11.29	11.32	11.40	**11.40**
lena	0.04	12.05	12.07	12.02	12.08	12.15	12.15	12.18	**12.23**
lena	0.06	12.32	12.27	12.36	12.35	12.38	**12.51**	12.43	12.49
lena	0.08	12.42	12.50	12.46	12.51	12.52	**12.56**	12.53	12.50
insect	0.02	8.54	8.58	8.60	8.61	8.71	**8.77**	8.68	8.74
insect	0.04	9.08	9.08	9.10	9.02	9.20	**9.20**	9.11	9.19
insect	0.06	9.15	9.21	9.19	9.17	9.22	**9.29**	9.31	9.24
insect	0.08	9.15	9.20	9.29	9.14	9.27	9.20	9.26	**9.27**
crowd	0.02	7.23	7.25	7.18	7.25	7.37	**7.39**	7.31	7.38
crowd	0.04	7.88	7.90	7.87	7.88	7.97	**8.02**	8.03	7.98
crowd	0.06	8.12	8.19	8.10	8.10	8.23	**8.25**	8.22	8.22
crowd	0.08	8.15	8.23	8.21	8.21	8.29	**8.31**	8.27	8.27
MRI knee	0.02	4.92	4.98	4.88	4.94	5.03	5.02	**5.05**	5.01
MRI knee	0.06	5.67	5.70	5.63	5.64	5.75	**5.75**	5.69	5.71

* The gain in dB is the difference in the PSNR of the de-noised image and the PSNR of the noisy image.

3.2 CHOICE OF PRIMARY RESOLUTION

In all the experiments, threshold was applied to all the details coefficients (horizontal, vertical and diagonal) at all the decomposed levels. No threshold was applied to the approximation coefficients at any level.

3.3 NUMBER OF DECOMPOSITION LEVELS

Several levels of decompositions were carried out for de-noising. It was found that 3-level decomposition and 4-level decomposition gave optimum results. 5-level decomposition was almost same as 4-level decomposition. However the 5-level decomposition most of the times blurred the de-noised image to an unacceptable degree. Higher levels of decomposition resulted in more blurring. Figure 3.1 shows de-noising of *plant* image using sym16 analyzing wavelet and 0.03 value of noise variance. It can be seen from the figure below that wavelet de-noising using four levels of decomposition produced optimum result.

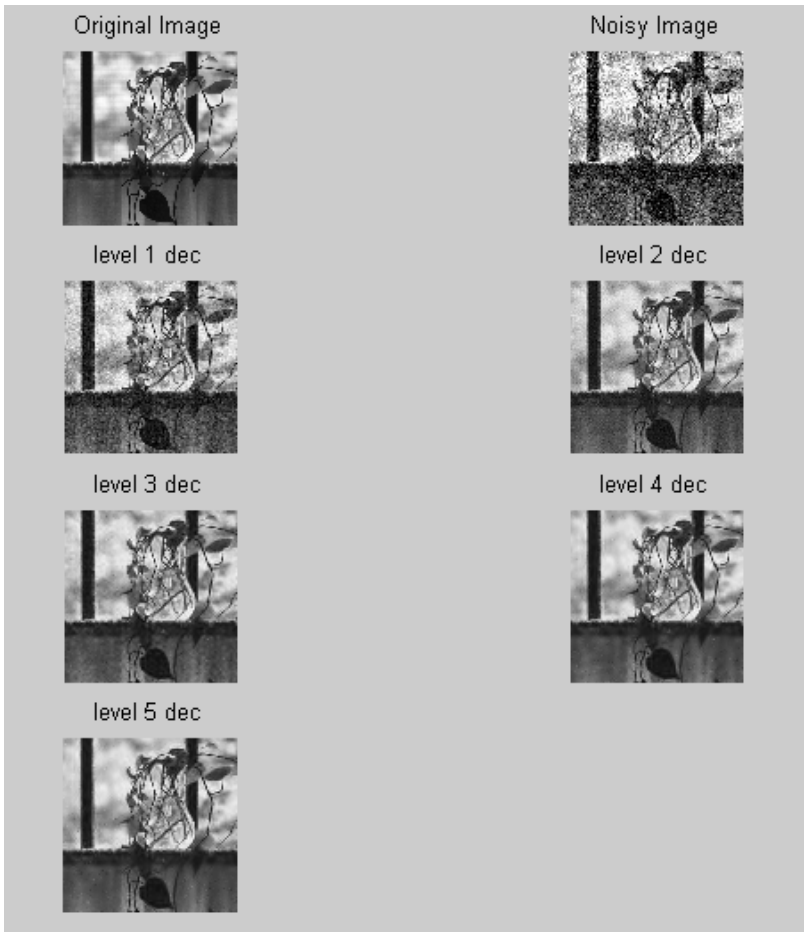

Figure 3.1: Wavelet de-noising using *ModiNeighShrink* scheme using various decomposition levels. Value of Gaussian noise variance added was 0.03. Analyzing wavelet was dB4.

4. EXPERIMENTAL RESULTS

The implementation was done on an image *lena*. The program was written in Matlab. There was one main program called the *modineighshrink.m* and four additional functions called *lden3.m*, *psnr.m, mean_var.m* and *var_noisyblock.* The function *lden3.m* performed the thresholding of the coefficients. The function *psnr.m* calculated the Peak Signal to Noise Ratio (PSNR) as shown in Table 4.1 for de-noised image *lena* for various Gaussian white noise levels. The PSNR is defined as

$$PSNR = -10*\log_{10}\{[\text{sum}_{i,j}(B(i,j) - A(i,j))^2]/[n^2*\max_{i,j}A(i,j)^2]$$

Table 4.1 also compares the PSNR obtained by different methods of de-noising of images, for example by Weiner filtering and Averaging filter. The results obtained clearly indicate that the wavelet based de-

noising using neighborhood coefficients performed best. Function *mean_var* calculated the mean and variance of a given image and the function *var_noisyblock* calculated the variance of a sub-block of an image.

The experiments were done using a window size of 3x3. The neighborhood window sizes of 3x3 and 5x5 are good choices for the algorithm *NeighShrink* [1]. Increasing the window size to a larger dimension diminishes the de-noising ability of the algorithm and decreasing the window size, like a term-by-term thresholding doesn't increase the de-noising ability either.

Four levels of wavelet decompositions were used. Sym16 was used as analyzing wavelets. It was seen that decomposing to higher levels blurred the image [1].

The following results were obtained using the modified *NeighShrink* algorithm described in the above described paper. Value of noise variance (normalized) added is 0.03.

Figure 4.1: The figure shows the original image, the noisy image and the reconstructed image using wavelet thresholding of coefficients using neighborhood information. (Noise variance was known).

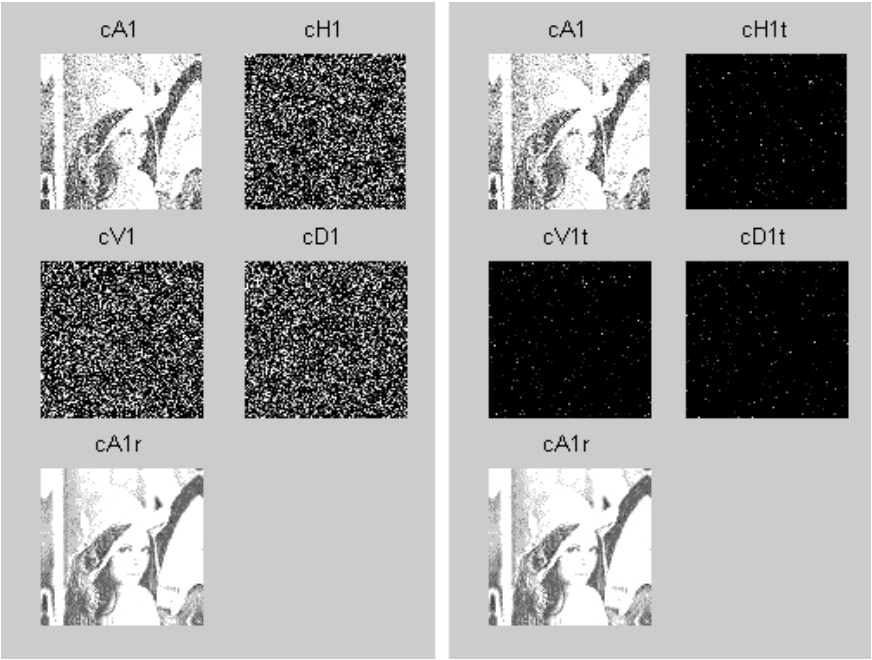

Figure 4.2: The figure shows the first decomposition level of the noisy image and the reconstructed approximation coefficient cA1r from the thresholded detail coefficients. cA1 is the first level approximation coefficient, cH1 is the first level horizontal detail coefficient, cV1 is the first level vertical detail coefficient and cD1 is the first level diagonal detail coefficient. cH1t, cV1t, cD1t are the respective thresholded coefficients.

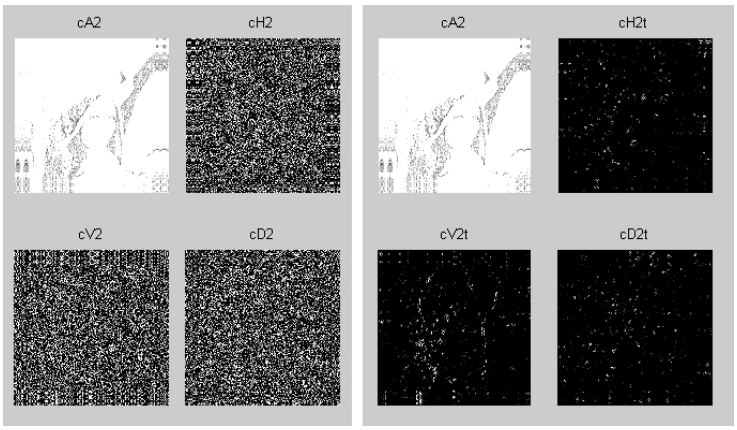

Figure 4.3: The figure shows the coefficients at second level of wavelet decomposition and the same coefficients after applying the threshold.

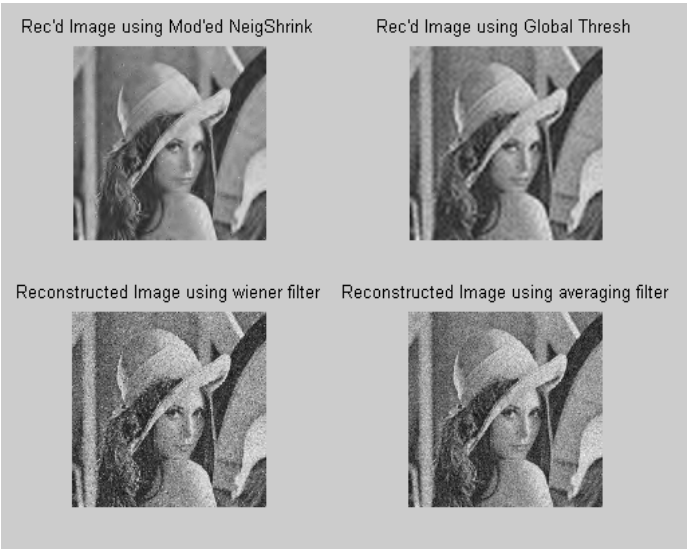

Figure 4.4: This figure compares the modified *NeighShrink* de-noising result with other de-noising schemes. It can be seen that the modified *NeighShrink* method performed better than the rest. The result can be verified by comparing the PSNR for each method in table 4.1.

Table 4.1: Comparison of PSNR (dB) and gain in dB of the various Image de-noising methods:

Norm. Var	Noisy Image (PSNR)	Mod' NeighShrink		Global Thresh		Wiener Filt		Averaging Filt	
		PSNR	Gain	PSNR	Gain	PSNR	Gain	PSNR	Gain
0.02	71.020	**82.533**	**11.513**	79.547	8.527	78.191	7.171	79.710	8.690
0.04	68.301	**80.617**	**12.316**	78.757	10.456	75.515	7.214	77.238	8.937
0.06	66.843	**79.463**	**12.620**	78.229	11.386	74.132	7.290	75.809	8.967
0.08	65.873	**78.517**	**12.644**	77.705	11.831	73.221	7.348	74.815	8.942
0.10	65.193	**77.695**	**12.502**	77.196	12.003	72.544	7.351	74.035	8.842

5. ESTIMATION OF NOISE VARIANCE

The noise variance plays an important role in defining constraints used in some of the de-noising algorithms [5]. Estimation of noise variance in a given image is a challenging problem in the area of image de-noising. A de-noising algorithm performs ideally if *a priori* knowledge about the noise is known. However, in most practical situations, one may not have the required information about the variance of the noise or the noise model. Thus, most algorithms assume known variance of the noise and the noise model to compare the performance with different algorithms. Gaussian noise with different variance values is added in the natural images to test the performance of the algorithm [2]. All the experiments performed in the previous sections assumed that noise variance was known. If we assume that the observation noise is a zero-mean, white random process that is uncorrelated with the image, then the noise field is completely characterized by its variance, which is commonly estimated by the sample variance computed over a low contrast local region of the observed image [5].

The procedure that was adopted for the particular experiment is described as follows:

I. Find the global variance of the whole image, which is of say MxN.
II. Divide the image into sub-blocks of size mxn (m<M & n<N).
III. Find the local variance of each of the sub-blocks.
IV. If the local variance of a particular sub-block is greater than the global variance, one can presume that, that block contains some edge information. One can then discard that particular value. Ideally, variance of a sub-image that originally had constant intensity level will reflect true estimate of noise variance, if the noise is an additive Gaussian white noise.
V. Find the average of the retained sub-block variances.
VI. Repeat the above procedure for sub-blocks of larger size.
VII. Once the averages of the sub-block variance are available, find the average of all these averages. This will give an estimate of the noise variance in the image.

The procedure mentioned above is certainly not the best way of estimating noise variance of an image. Figure 5.1 shows the Mean absolute error in the variance estimation for various values of variances and block sizes. A set of nineteen images belonging to different classes were used to find the mean absolute error. One may observe that the error in general increases with the size of sub-block.

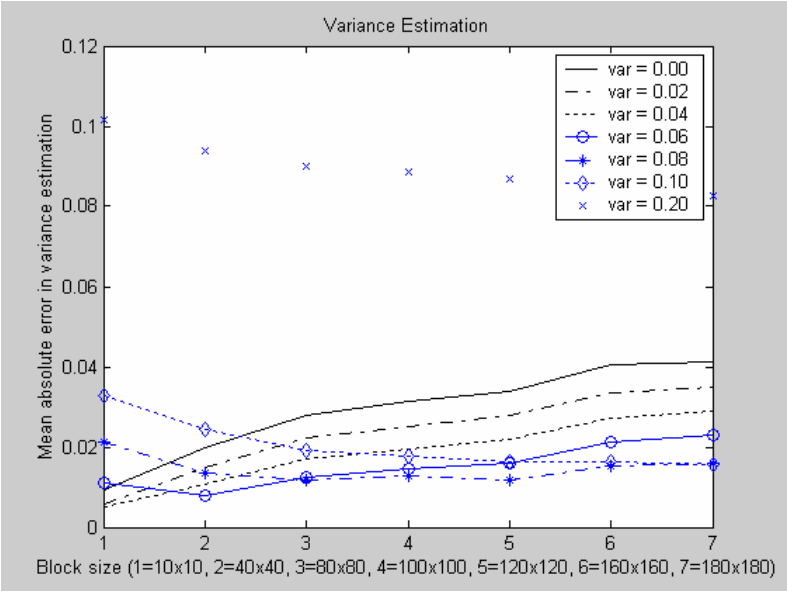

Figure 5.1: Mean absolute error in the noise estimation procedure mentioned above.

For the purpose of evaluation of the modified *NeighShrink* de-noising algorithm, when *a priori* knowledge of noise variance is not known, sub-blocks of sizes 10x10, 40x40 and 80x80 were considered. The estimated variance was calculated by averaging the three variance estimates calculated using the three different sub-block sizes.

Figure 5.2 shows the result obtained by *ModiNeighShrink* algorithm with no prior information about the noise. For comparison with the previous results (when noise variance was assumed to be known), Gaussian noise with same variance value (0.03) was added to *lena* image. Sym16 was chosen as the analyzing wavelet. One can see that the de-noising capability of the algorithm didn't diminish much as compared to when the exact noise variance was known. The modified *NeighShrink* algorithm still performed better than the other algorithms.

Figure 5.2: The original image, noisy image (variance of Gaussian noise added: 0.03) and the reconstructed image by modified *NeighShrink* algorithm. The estimated noise variance is 0.0388

Figures 5.3, 5.4 and 5.5 show the original, noisy and de-noised images for different image classes. It can be seen that the modified *NeighShrink* algorithm performs well, even if the noise estimate is not accurate for different image classes.

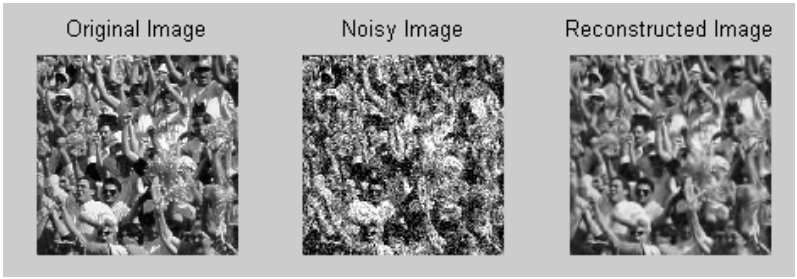

Figure 5.3: The original image, noisy image (variance of Gaussian noise added: 0.03) and the reconstructed image by modified NeighShrink algorithm. The estimated noise variance is 0.0563

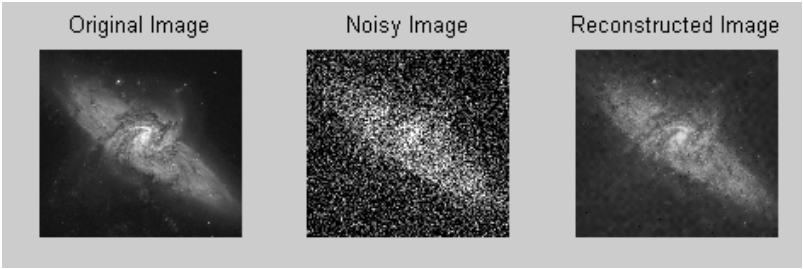

Figure 5.4: The original image, noisy image (variance of Gaussian noise added: 0.08) and the reconstructed image by modified NeighShrink algorithm. The estimated noise variance is 0.0451

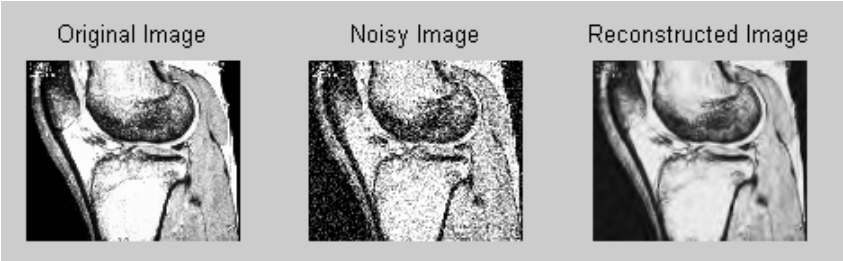

Figure 5.5: The original image, noisy image (variance of Gaussian noise added: 0.04) and the reconstructed image by modified NeighShrink algorithm. The estimated noise variance is 0.0606

This is actually not very surprising. While the traditional methods require that one knows or assumes something about the signals one wants to extract from noise, for the wavelet methods its enough to know that a signal belongs to a much larger family, which includes most classes that one encounters in the field of de-noising of natural images. Then, without knowing anything more, Donoho says, "you do as well as someone who makes correct assumptions, and much better than someone who makes wrong assumptions" [4]. This is because wavelet transforms has the ability to represent the essential information in a few large coefficients.

6. CONCLUSION AND FUTURE WORK

In this paper we studied image de-noising by modifying the *NeighShrink* algorithm. Experimental results show that the modified *NeighShrink* algorithm gives better results than *NeighShrink*, Weiner filter, averaging filter and also wavelet de-noising technique using global threshold, even when an accurate estimate of noise variance was not known. We also saw that the wavelet transform based de-noising schemes can be applied to a wide class of images that are generally encountered in de-noising field using a very general procedure. In future, one can expect better performance from the modified *NeighShrink* algorithm by incorporating better estimation techniques like Image recovery using the EM algorithm in the *NeighShrink* algorithm.

7. REFERENCES

1) G.Y. Chen, T.D. Bui and A. Krzyzak, "Image denoising using neighbouring wavelet coefficients," ICASSP 2004.
2) Mukesh C. Motwani, Mukesh C. Gadiya, Radhi C. Motwani and Frederick C. Harris, Jr., "Survey of Image Denoising Techniques".
3) James S. Waker,"A primer on Wavelets and their Scientific Applications," Chapman & Hall/CRC. 1999.
4) Burbara Burke Hubbard, "The world according to wavelets". Universities Press (India) Private Limited. 1998.
5) Vijay K. Madisetti, Douglas B. Williams and et al., "Digital Signal Processing Handbook," CRC Press with IEEE Press. 1998.
6) Michel Misiti, Yuevs Misiti, Georges Oppenheim and Jean-Michel Poggi, "Wavelet Toolbox for use with MatLab."