

A Comprehensive Study on Wavelet Based Shrinkage Methods for Denoising Natural Images

S.SUTHA, E.JEBAMALAR LEAVLINE, D.ASIR ANTONY GNANA SINGH
Electrical and Electronics Engineering, Electronics and Communication Engineering, Computer
Science and Engineering
Anna University
Tiruchirappalli – 620 024
INDIA
suthapadmanabhan@gmail.com, jebi.lee@gmail.com, asirantony@gmail.com

Abstract: - Transmitting the information in the form of images has drawn much importance in the modern age. The images are often corrupted by various types of noises during acquisition and transmission. Such images have to be cleaned before using in any applications. Image denoising is a thurst area in image processing for decades. Wavelet transform has been an efficient tool for image representation for decades because of its simplicity, energy compaction and sparse representation. Ample of wavelet based thresholding techniques are proposed based on universal and adaptive thresholding techniques. Fixing an optimal threshold is a key factor to determine the performance of denoising algorithms. This optimal threshold shall be estimated from the image statistics for ensuring better performance of noise removal in terms of clarity (or quality of the) images. In this paper, an experimental study of the state of the art wavelet based thresholding methods is presented. The denoising performance of the wavelet based shrinkage methods are compared interms of mean square error, peak signal to noise ratio, image enhancement factor and the most recent measure namely multiscale structural similarity index.

Key-Words: - Image denoising, Wavelet transform, Threshold methods, Adaptive threshold, Wavelet subbands, Shrinkage methods.

1 Introduction

Image denoising plays a key role in the field of image processing. Denoising is usually employed as a pre-processing stage in Image processing areas like segmentation, analysis, feature extraction and object recognition. The noisy components present in an image reduce the clarity of the image by affecting the structural information and blur the edges. The undesired component present in the images defined as noise fall into two major categories namely additive and multiplicative noise. If $s(x, y)$ is the clean image and $n(x, y)$ denotes the noise, then corrupted image $w(x, y)$ in the presence of additive and multiplicative noise can be mathematically modeled as (1) and (2)

$$w(x, y) = s(x, y) + n(x, y) \quad (1)$$

$$w(x, y) = s(x, y) \times n(x, y) \quad (2)$$

Denoising is then reduced to a simple estimation problem, i.e. Estimation of $s(x, y)$ from $w(x, y)$. Most of the spatial and transform domain denoising techniques require the prior information about the type of noise present in the image. The probability density function (PDF) of the noise is modeled mathematically in different ways based on the

statistical properties of the noise. Some of the well known noise models are Gaussian, impulse, poisson, uniform, exponential, gamma or erlang noise who's PDFs, mean and variance of these PDFs [1-3].

Earlier, Fourier transform was used successfully for additive noise removal [4], yet Fourier transform cannot explore spatial and frequency information simultaneously. The discrete cosine transform (DCT) posses the characteristics of Decorrelation, Energy Compaction, Separability and Orthogonality for denoising [5], but it fails in the presence of singularities or edges. Because of the lack of sparsity, edges cannot be represented or restored effectively, and significant ringing artifacts arising from the Gibbs phenomenon become visible [6]. The Wavelet transform [7, 8] is a better choice for various image processing applications including image denoising.

In Wavelet based denoising, first the image is transformed with an orthogonal transform. Then, the transformed coefficients are thresholded by non linear shrinkage function [9]. Finally, the coefficients are reconstructed by the inverse orthogonal transform. This provides large

transformed coefficients of the image compared to the noise. Hence, the smaller coefficients are eliminated, and the image is reconstructed with the remaining coefficients to remove the noisy components present in the image. Several Wavelet based shrinkage functions are proposed in literature [9-14] following hard or soft threshold introduced by D. L. Donoho [15, 16]. The soft thresholding of wavelet coefficients is preferred over hard thresholding because of its visually appealing performance.

In this empirical study, a pragmatic investigation on the various wavelet based threshold methods for denoising the natural images. This paper is organized as follows. Section 2 deals with the methodology of Wavelet based denoising. Various threshold functions are discussed in section 3. Experimental results are presented in section 4 and conclusions are drawn in section 5.

2 Methodology

2.1 Wavelet Transform

Wavelet transform is a wonderful mathematical tool for signal and image processing due to its multi-resolution nature and computational efficiency. Wavelet schemes are especially suitable for applications where scalability and tolerable degradation are the important considerations. Wavelet transform decomposes a signal into a set of basis functions. Wavelets are derived from a single prototype wavelet $\psi(t)$ called mother wavelet by scaling and translation as in (3)

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \tag{3}$$

's' is the scale factor, τ - is the translation factor and the factor \sqrt{s} is for energy normalization across the different scales. The continuous wavelets represented by (3) are highly redundant. This problem is eliminated by discrete wavelets. Discrete wavelets are not continuously scalable and translatable but can only be scaled and translated in discrete steps hence piecewise continuous [17]. The discrete wavelet at level 'l' is represented in (4) can be extended easily to 2D case

$$\psi(2^j t) = \sum_k g_{l+1}(k) \varphi(2^{j+1} t - k) \tag{4}$$

where $\varphi(\cdot)$ is the scaling function and $g(\cdot)$ is the wavelet filter.

Unlike Fourier bases, Wavelet transform provides excellent time and frequency representation simultaneously. With the sub sampling property, the performance of the Wavelet transform can be realized using iterative filter bank structures. Every time the filter bank is iterated, the number of samples for the next stage is halved so that only one sample is left at the end. The iteration is halted at the point once the number of samples becomes smaller than the length of the scaling filter or the wavelet filter and length of the longest filter function determines the width of the spectrum of the scaling function [17].

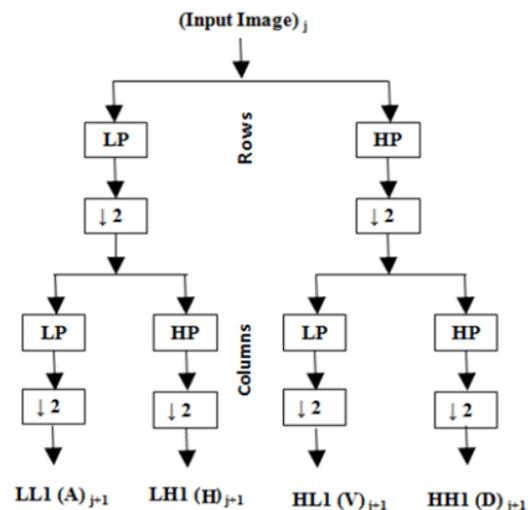


Fig 1: Wavelet decomposition (LP – Low pass filter, HP – High pass filter, A – Approximation coefficients, H, V, D – Horizontal, Vertical and Diagonal detail coefficients respectively)

The wavelet decomposition of an image is carried out as follows: In the first level of decomposition, the image is split into 4 subbands, namely the HH, HL, LH and LL subbands. The HH subband gives the diagonal details of the image; the HL and LH subbands give the horizontal and vertical features respectively.

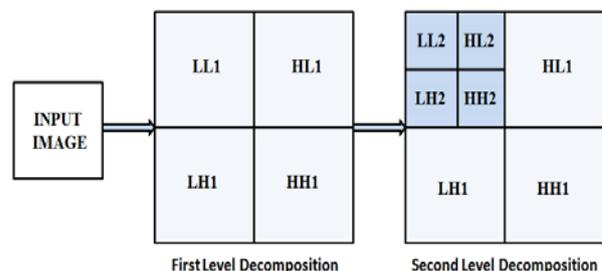
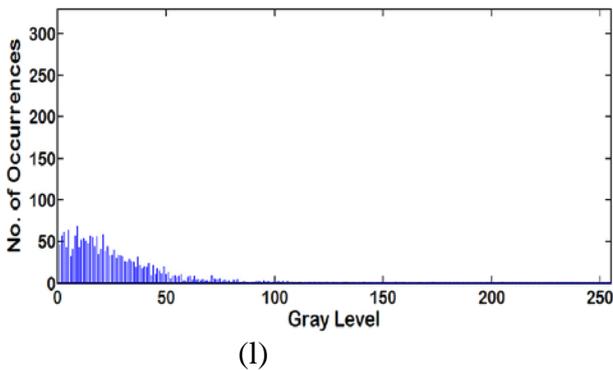
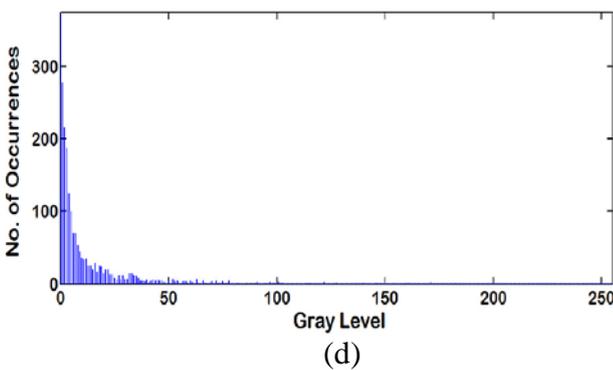
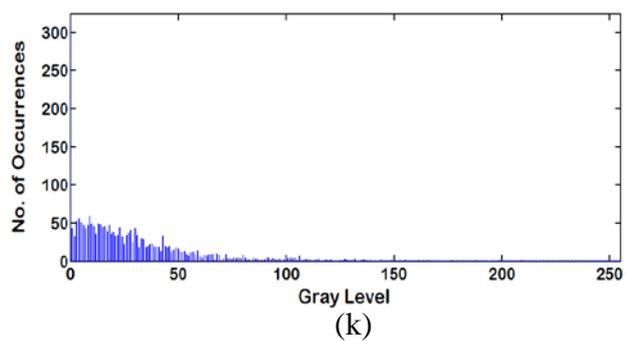
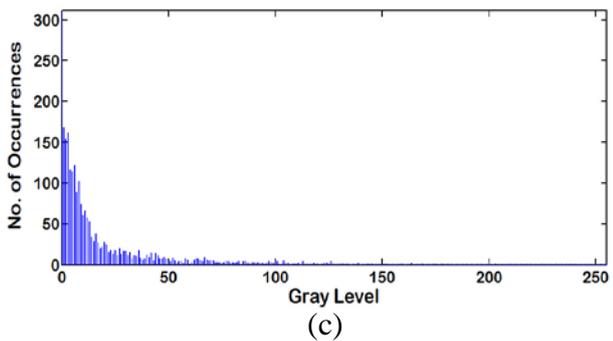
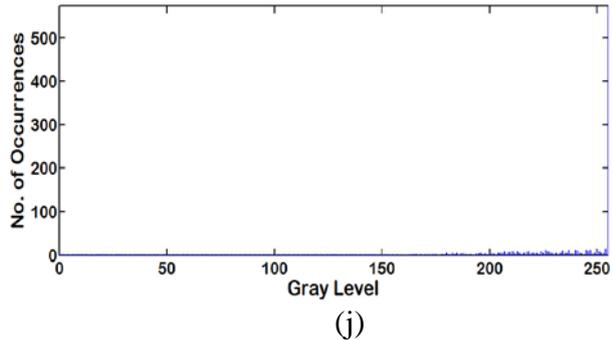
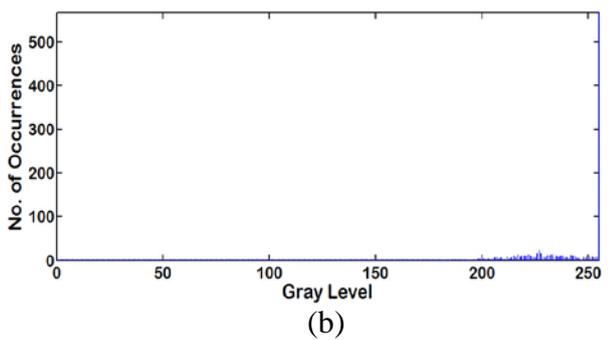
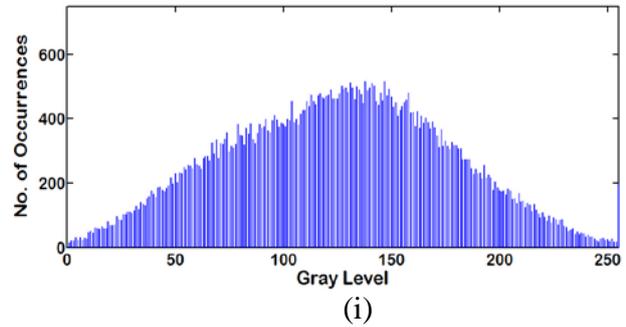
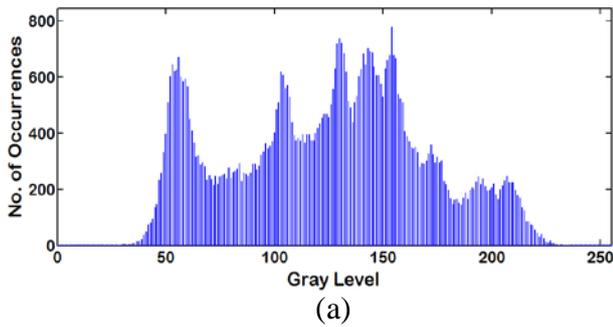


Fig 2: Subbands after two levels of wavelet decomposition

The LL subband is the low resolution residual consists of low frequency components and its subbands are further split at higher levels of decomposition [9, 17, 18]. This decomposition process is shown in Fig 1. Subbands after two levels of wavelet decomposition are shown in Figure 2.

Apart from efficient multiresolution representation and sub sampling, wavelets exhibit interesting characteristics such as sparsity and high energy compaction. These features are particularly useful in

image denoising and compression. The histograms of clean image, noisy image and their respective subband histograms up to two levels of wavelet decomposition are shown in Figure 3. From Figure 3(b) and 3(j) it is evident that, the wavelet coefficients that correspond to noisy pixels are significantly smaller than the image details. Hence, by simple threshold methods i.e. by shrinking the insignificant coefficients, effective denoising can be achieved



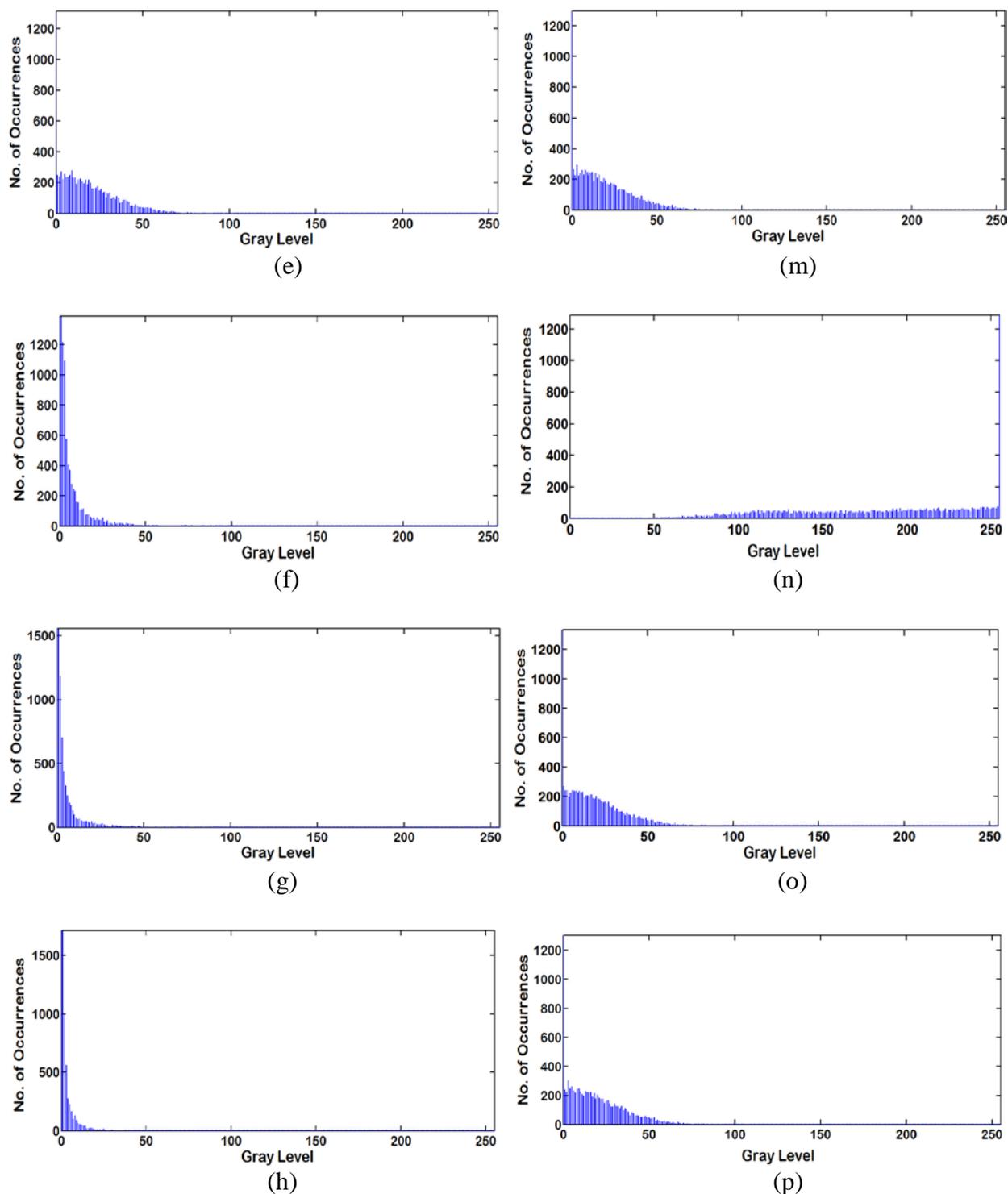


Fig 3: Histogram of clean image, noisy image and Wavelet subbands: Column 1: (a) Histogram of clean Lena image, (b) – (e) Histogram of (LL1, HL1, LH1 and HH1 -level 2 subbands) approximate, vertical, horizontal and diagonal coefficients respectively of clean image. (f)-(h) Histogram of (HL, LH and HH -level 1 subbands) vertical, horizontal and diagonal coefficients respectively of clean image. Column 2: (i) Histogram of Lena image corrupted with Gaussian noise of variance 0.01, (j) – (m) Histogram of (LL1, HL1, LH1 and H1 -level 2 subbands) approximate, vertical, horizontal and diagonal coefficients respectively of noisy image. (n)- (p) Histogram of (HL, LH and HH -level 1 subbands) vertical, Horizontal and diagonal coefficients respectively of noisy image.

2.2 Image denoising using Wavelet Transform:

Wavelet based denoising techniques follow the similar steps irrespective of the shrinkage function. A general framework for wavelet based denoising is shown in Figure 4. The algorithm of wavelet based image denoising is as follows.

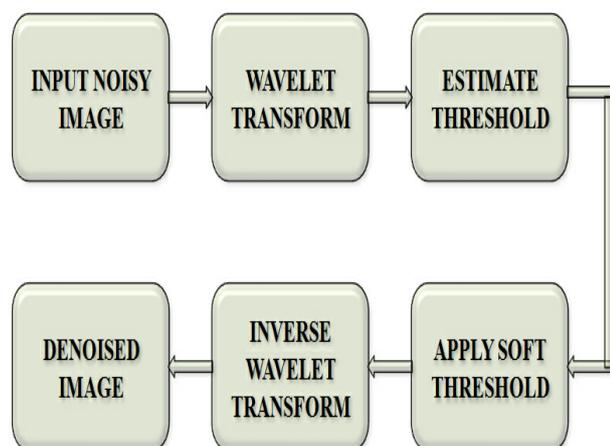


Fig 4. Wavelet Denoising Framework

- Step 1: Read the noisy image as input
- Step 2: Perform 2D Discrete Wavelet Transform and obtain Wavelet Coefficients (Subbands)
- Step 3: Estimate noise variance from the noisy image.
- Step 4: Calculate the threshold using suitable non-linear shrinkage function.
- Step 5: Apply soft thresholding.
- Step 6: Perform inverse 2D Discrete Wavelet Transform on the thresholded wavelet coefficients.
- Step 7: Obtain the denoised image
- Step 8: Evaluate the quality of the denoised image.

The performance of the denoising algorithm relies on the optimal value of threshold. Fixing an optimal threshold is not an easy task. The non linear threshold functions can be seen as two major categories namely fixed threshold and adaptive threshold. Fixed threshold methods apply same threshold value with hard/soft threshold on the complete set of wavelet coefficients. As shown in Figure 3, the ranges of magnitudes of all wavelet subbands are not similar. Hence, fixed threshold methods are likely to oversmooth image details, failing to preserve image details. On the other hand subband and scale adaptive threshold methods have been proposed to handle this. These methods use different threshold value for each subband at each scale so as to preserve image details.

3 Threshold Methods for Wavelet based Denoising

Selecting an optimal threshold is a crucial phase in denoising process. If the threshold is too large, noisy components may not be eliminated. On the other hand if the threshold is too small, it may remove the image details also resulting in overly smoothed images. The inefficient threshold may affect the edge details; this may degrade the visual quality. [9]. Hence, the threshold must to be selected carefully.

3.1 Universal Threshold

The universal threshold can be defined as in (5),

$$T_u = \sigma \sqrt{2 \log N} \quad (5)$$

N being the image size, σ being the noise variance is well known in wavelet literature as the Universal threshold. The universal threshold can give a better estimate of the image with the soft threshold [9, 15, 16]. However, the estimated threshold value depends on the image size. With a particular ' σ ', universal threshold yields larger threshold for big images and comparatively small threshold for smaller images, also it requires the prior knowledge about the noise distribution.

3.2 Visu Shrink

It follows the hard threshold rule. An estimate of the noise variance ' σ ' is defined based on the median absolute deviation which is a robust estimator in (6) and the threshold is calculated as in (7).

$$\hat{\sigma}^2 = \left[\frac{\text{median}(|X_{ij}|)}{0.675} \right]^2, X_{ij} \in HH1 \quad (6)$$

$$T_v = \hat{\sigma} \sqrt{2 \log N} \quad (7)$$

Visu Shrink does not deal with minimizing the mean squared error. Another disadvantage is that it cannot remove speckle noise. Yet, with additive gaussian noise assumption Visu Shrink exhibits better denoising performance than the universal threshold [9, 16].

3.3 Sure Shrink

A threshold chooser based on Stein's Unbiased Risk Estimator (SURE) was proposed by Donoho and Johnston and is called as Sure Shrink. It is a combination of the universal threshold and the SURE threshold. It has the distinct advantage of offering an analytic unbiased estimator. The goal of Sure Shrink is to minimize the mean squared error

of the estimate. Sure Shrink suppresses noise by thresholding the empirical wavelet coefficients [9]. Sureshrink is smoothness adaptive, which means that if the unknown function contains abrupt changes or boundaries in the image, the reconstructed image also has the same [14, 15, 20-21]. The risk for a particular threshold value 't' can be estimated. The optimal threshold can be selected by minimizing the risks in 't'. If {Xi: i =1,...,d} are the transformed coefficients in the 'j'th subband, the loss can be estimated as $\|\hat{X}-X\|^2$. For the soft threshold estimator, $\hat{X}_i = \eta_t(X_i)$,

$$SURE(t; X) = d - 2\#\{i : |X_i| \leq t\} + \sum_{i=1}^d \min(X_i, t)^2 \quad (8)$$

Then the threshold T_s is given by $T_s = \arg \min SURE(t; X)$. The SURE principle can be used to select a threshold that is applied to the image data, resulting in an estimate of the mean vector. This estimate is sparse and much less noisy than the raw image data [16]. The SURE principle just described has a serious draw-back in situations of extreme sparsity of the wavelet coefficients. In such cases the noise contributed to the SURE profile by many coordinates at which the signal is zero, swamps the information contributed to the SURE profile by the few coordinates where the signal is nonzero. Consequently, Sure Shrink uses a Hybrid scheme [22].

3.4 Bayes Shrink

Unlike universal threshold, Visu Shrink and Sureshrink, Bayes Shrink sets different thresholds for every subband. Also the noise distribution is assumed to be gaussian, and the relationship between the wavelet coefficients of the degraded image, uncorrupted image and generalised Gaussian noise with distribution $N(0, \sigma^2)$ (Y, X and V respectively), can be modeled as $Y = X+V$. Since all the above three factors are mutually independent, their variances satisfy the condition,

$$\sigma_y^2 = \sigma_x^2 + \sigma_v^2 \quad (9)$$

Since, the diagonal coefficients of first level wavelet decomposition (HH1) contains significant amount of information about the noise components, the noise variance 'σv' is calculated using the robust estimator in equation (6).

Variance of the corrupted image is estimated as

$$\hat{\sigma}_y^2 = \frac{1}{J} \sum_{j=1}^J W_j^2 \quad (10)$$

Where W_j are the wavelet coefficients in each scale 'j' and 'J' is the total number of wavelet coefficients. The threshold value using Bayesshrink is given by (11, 12)

$$T_b = \begin{cases} \frac{\hat{\sigma}_v^2}{\hat{\sigma}_x} \text{ if } \hat{\sigma}_v^2 < \hat{\sigma}_y^2 \\ \max\{|W_j|\}, \text{ otherwise} \end{cases} \quad (11)$$

$$\hat{\sigma}_x = \sqrt{\max(\hat{\sigma}_y^2 < \hat{\sigma}_v^2, 0)} \quad (12)$$

The estimation in equation (11) holds good for images corrupted by Gaussian noise. Nevertheless, it is less sensitive to the noise around edges [24, 25], but completely denoises the flat regions of the image. Modified bayes shrink overcomes this issue. The threshold is given by (13).

$$T_{mb} = \beta \frac{\hat{\sigma}_y^2}{\hat{\sigma}_x}, \quad \beta = \sqrt{\frac{\log J}{2j}} \quad (13)$$

'J' is the total of coefficients of wavelet. 'j' is the wavelet decomposition level present in the subband coefficients under consideration. The modified bayes shrink yields the best results for denoising and preserves edges better than bayes shrink [23].

3.5 Normalshrink

Normal shrink an adaptive threshold estimation method based on the generalized Gaussian distribution (GGD) modeling of subband coefficients. The threshold is computed by

$$T_n = \beta \frac{\hat{\sigma}_v^2}{\hat{\sigma}_y} \quad (14)$$

where σ_v and σ_y are the standard deviation of the noise and the subband data of noisy image respectively. β is the scale parameter, computed as

$$\beta = \sqrt{\log\left(\frac{L_j}{M}\right)} \quad (15)$$

L_j is the length of the subband at j^{th} level, M is the total number of decompositions, σ_v^2 is the estimated noise variance of HH1 subband and σ_y is the standard deviation of the image subband. This method is computationally more efficient and adaptive because the parameters required for estimating the threshold depend on subband data. Performance of normal shrink is similar to bayes

shrink. But normal shrink preserves as well as removes noise better than bayes shrink [24,25].

3.6 Minimax Threshold

The minimax principle was initially used in statistics to design estimators. Since the denoised signal can be assimilate to the estimator of the unknown regression function, the minimax estimator is the option that realizes the minimum, over a given set of functions, of the maximum mean square error[26].The Minimax threshold denoising algorithm was proposed in [15,27]. The optimal threshold is derived from minimising the constant term in an upper bound of the risk involved in the estimation. Two oracles namely diagonal linear projection (DLP) and the diagonal linear shrinker (DLS) are used as in equation (16,17). DLP tells when to “keep” or “kill” each wavelet coefficient, whereas DLS states how much shrinking is applied to each wavelet coefficient.

$$Risk_{DLP}(k) := \min(d^2, 1) \quad (16)$$

$$Risk_{DLS}(k) := \frac{d^2}{1 + d^2} \quad (17)$$

Minimax threshold does not give good visual quality, but it has the advantage of giving predictive performance [27, 28].

4 Experimental Results and Discussion

4.1 Experimental setup

The experiments were carried out using MATLAB 7.5.0(R2007b). In search of the best threshold method, all the threshold methods discussed in

section 3 are implemented and their performance was tested interms of mean square error (MSE), peak signal to noise ratio (PSNR), image enhancement factor (IEF) and multiscale structural similarity index (MSSIM). These tests are conducted on standard gray scale images corrupted by additive white gaussian noise at various noise densities.

4.2 Results and Discussion

Extensive experiments were carried out on a wide range of standard benchmark gray scale images. Table 1 – Table 4 shows the comparison of MSE, PSNR, IEF and MSSIM respectively for the benchmark Lena image at various noise densities. From the experimental results, it is evident that Bayes shrink and wavelet based minimax threshold produces better results than Universal threshold, Visu Shrink and Normal shrink. Sure shrink exhibits moderate denoising performance as evident from the results shown in Figure 5. The performance of the wavelet based shrinkage methods was tested on various Wavelet bases namely db1, db2, coif1, coif5, sym2, sym8, bior1.1, bior2.2, rbio1.1, rbio2.2 using the MATLAB Wavelet Tool Box function ‘*wfilters*’ and tabulated in Table 5. Almost all the wavelet filters perform in a much similar fashion. The denoised images resulting from various threshold methods compared are shown in Figure 6 and Figure 7. For smooth images like ‘Peppers’, Visu Shrink, Sure shrink, Bayes shrink and wavelet based minimax threshold are visually appealing. On the other hand, for images with more details (Barbara), Visu Shrink, Bayes shrink and minimax threshold are not able to preserve edges as in Figure 7. Sureshrink exhibits visually good results for images with more details.

Table 1: Comparison of MSE for Lena image at various noise levels

| MSE | | | | | | |
|----------------|-----------|-------------|------------|--------------|---------------|------------------|
| Noise Variance | Universal | Vishushrink | Sureshrink | Bayes Shrink | Normal Shrink | Minmax Threshold |
| 0.001 | 65.04797 | 65.04797 | 123.4624 | 34.68359 | 65.09261 | 33.87048 |
| 0.005 | 324.849 | 324.849 | 206.8091 | 103.4518 | 325.4177 | 109.4437 |
| 0.01 | 637.1436 | 637.1436 | 250.6609 | 160.6539 | 638.7855 | 168.7058 |
| 0.02 | 1238.726 | 1238.726 | 308.0724 | 234.8927 | 1243.426 | 255.0384 |
| 0.03 | 1806.012 | 1806.012 | 348.7247 | 290.374 | 1814.635 | 312.6099 |
| 0.04 | 2320.817 | 2320.817 | 388.3303 | 342.807 | 2333.93 | 368.3421 |

| | | | | | | |
|------|----------|----------|----------|----------|----------|----------|
| 0.05 | 2806.119 | 2806.119 | 417.9903 | 380.3319 | 2824.243 | 398.8822 |
| 0.06 | 3243.557 | 3243.557 | 460.5843 | 431.1035 | 3266.997 | 451.9094 |
| 0.07 | 3660.249 | 3660.249 | 486.4702 | 465.2139 | 3689.398 | 482.0533 |
| 0.08 | 4049.882 | 4049.882 | 516.9015 | 500.9522 | 4084.87 | 514.2099 |
| 0.09 | 4401.148 | 4401.148 | 555.5604 | 538.2613 | 4442.18 | 553.3394 |
| 0.1 | 4686.806 | 4686.806 | 577.8989 | 565.877 | 4733.887 | 571.0069 |
| 0.2 | 7122.454 | 7122.454 | 872.958 | 872.958 | 7238.49 | 872.958 |
| 0.3 | 8583.293 | 8583.293 | 1040.687 | 1040.687 | 8775.602 | 1040.687 |
| 0.4 | 9551.858 | 9551.858 | 1220.568 | 1220.568 | 9821.158 | 1220.568 |
| 0.5 | 10248.02 | 10248.02 | 1348.112 | 1348.112 | 10595.76 | 1348.112 |
| 0.6 | 10733.49 | 10733.49 | 1473.708 | 1473.708 | 11158.07 | 1473.708 |
| 0.7 | 11171.09 | 11171.09 | 1579.88 | 1579.88 | 11675.76 | 1579.88 |
| 0.8 | 11466.83 | 11466.83 | 1606.041 | 1606.041 | 12051.66 | 1606.041 |
| 0.9 | 11769.62 | 11769.62 | 1724.504 | 1724.504 | 12433.14 | 1724.504 |
| 1 | 11874.94 | 11874.94 | 1703.465 | 1703.465 | 12619.38 | 1703.465 |

Table 2: Comparison of PSNR for Lena image at various noise levels

| PSNR | | | | | | |
|----------------|-----------|-------------|------------|--------------|---------------|------------------|
| Noise Variance | Universal | Vishushrink | Sureshrink | Bayes Shrink | Normal Shrink | Minmax Threshold |
| 0.001 | 30.006 | 29.20094 | 27.20758 | 32.7071 | 30.00303 | 32.78202 |
| 0.005 | 23.01399 | 25.82599 | 24.97511 | 27.98342 | 23.00639 | 27.7389 |
| 0.01 | 20.08843 | 24.60077 | 24.13994 | 26.07189 | 20.07725 | 25.8595 |
| 0.02 | 17.20105 | 23.45165 | 23.24428 | 24.42211 | 17.1846 | 24.06475 |
| 0.03 | 15.5636 | 22.80459 | 22.70598 | 23.50123 | 15.54291 | 23.18078 |
| 0.04 | 14.4744 | 22.30067 | 22.23879 | 22.78031 | 14.44993 | 22.46829 |
| 0.05 | 13.64974 | 21.93222 | 21.91914 | 22.32918 | 13.62178 | 22.12236 |
| 0.06 | 13.02059 | 21.5189 | 21.49771 | 21.78499 | 12.98932 | 21.58029 |
| 0.07 | 12.4957 | 21.26024 | 21.26024 | 21.45428 | 12.46125 | 21.29985 |
| 0.08 | 12.05638 | 21.00174 | 20.99673 | 21.13284 | 12.01902 | 21.0194 |
| 0.09 | 11.69514 | 20.6887 | 20.68349 | 20.82087 | 11.65484 | 20.70089 |
| 0.1 | 11.42203 | 20.51384 | 20.51228 | 20.60358 | 11.37863 | 20.56439 |
| 0.2 | 9.604507 | 18.72087 | 18.72087 | 18.72087 | 9.534324 | 18.72087 |
| 0.3 | 8.794264 | 17.9576 | 17.9576 | 17.9576 | 8.698035 | 17.9576 |
| 0.4 | 8.329925 | 17.26518 | 17.26518 | 17.26518 | 8.209177 | 17.26518 |
| 0.5 | 8.024405 | 16.83354 | 16.83354 | 16.83354 | 7.879482 | 16.83354 |
| 0.6 | 7.823396 | 16.44669 | 16.44669 | 16.44669 | 7.654912 | 16.44669 |
| 0.7 | 7.649847 | 16.14456 | 16.14456 | 16.14456 | 7.457951 | 16.14456 |
| 0.8 | 7.536368 | 16.07324 | 16.07324 | 16.07324 | 7.320335 | 16.07324 |
| 0.9 | 7.423179 | 15.76416 | 15.76416 | 15.76416 | 7.184994 | 15.76416 |
| 1 | 7.38449 | 15.81747 | 15.81747 | 15.81747 | 7.120424 | 15.81747 |

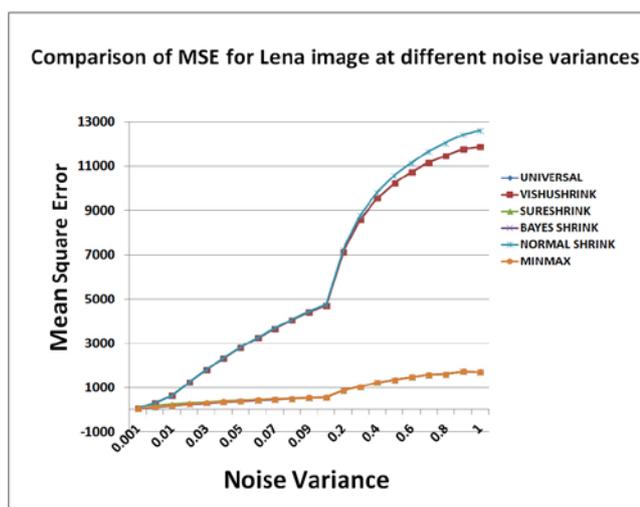
Table 3: Comparison of IEF for Lena image at various noise levels

| Noise Variance | IEF | | | | | |
|----------------|-----------|-------------|------------|--------------|---------------|------------------|
| | Universal | Vishushrink | Sureshrink | Bayes Shrink | Normal Shrink | Minmax Threshold |
| 0.001 | 0.456764 | 0.456764 | 0.2398 | 0.850749 | 0.456452 | 0.865553 |
| 0.005 | 0.222126 | 0.222126 | 0.348908 | 0.697498 | 0.221738 | 0.659311 |
| 0.01 | 0.135353 | 0.135353 | 0.344048 | 0.536802 | 0.135005 | 0.511182 |
| 0.02 | 0.080077 | 0.080077 | 0.321979 | 0.422291 | 0.079774 | 0.388933 |
| 0.03 | 0.057258 | 0.057258 | 0.296532 | 0.356121 | 0.056986 | 0.33079 |
| 0.04 | 0.045887 | 0.045887 | 0.274239 | 0.310656 | 0.045629 | 0.28912 |
| 0.05 | 0.038429 | 0.038429 | 0.257989 | 0.283534 | 0.038183 | 0.270348 |
| 0.06 | 0.033821 | 0.033821 | 0.238178 | 0.254466 | 0.033579 | 0.24275 |
| 0.07 | 0.030486 | 0.030486 | 0.229383 | 0.239863 | 0.030246 | 0.231484 |
| 0.08 | 0.028084 | 0.028084 | 0.220035 | 0.22704 | 0.027843 | 0.221187 |
| 0.09 | 0.025896 | 0.025896 | 0.205146 | 0.211739 | 0.025656 | 0.205969 |
| 0.1 | 0.024091 | 0.024091 | 0.195378 | 0.199529 | 0.023851 | 0.197737 |
| 0.2 | 0.01654 | 0.01654 | 0.134948 | 0.134948 | 0.016275 | 0.134948 |
| 0.3 | 0.014012 | 0.014012 | 0.115571 | 0.115571 | 0.013705 | 0.115571 |
| 0.4 | 0.012667 | 0.012667 | 0.099126 | 0.099126 | 0.012319 | 0.099126 |
| 0.5 | 0.011899 | 0.011899 | 0.090457 | 0.090457 | 0.011509 | 0.090457 |
| 0.6 | 0.011421 | 0.011421 | 0.083182 | 0.083182 | 0.010986 | 0.083182 |
| 0.7 | 0.010992 | 0.010992 | 0.077726 | 0.077726 | 0.010517 | 0.077726 |
| 0.8 | 0.010609 | 0.010609 | 0.075746 | 0.075746 | 0.010094 | 0.075746 |
| 0.9 | 0.010509 | 0.010509 | 0.071724 | 0.071724 | 0.009948 | 0.071724 |
| 1 | 0.010392 | 0.010392 | 0.072444 | 0.072444 | 0.009779 | 0.072444 |

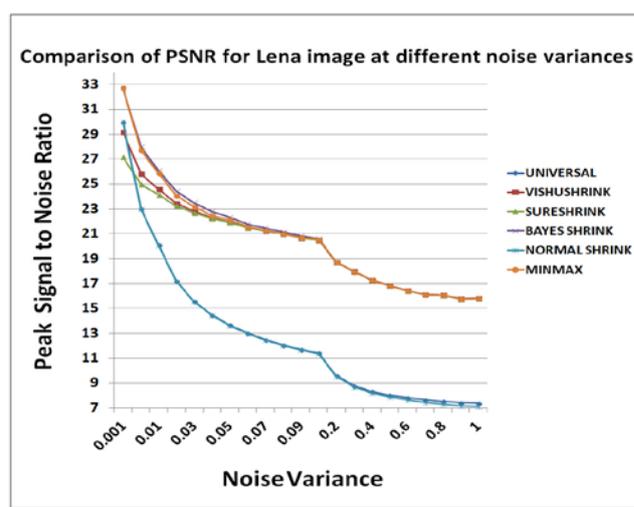
Table 4: Comparison of MSSIM for Lena image at various noise levels

| Noise Variance | MSSIM | | | | | |
|----------------|-----------|-------------|------------|--------------|---------------|------------------|
| | Universal | Vishushrink | Sureshrink | Bayes Shrink | Normal Shrink | Minmax Threshold |
| 0.001 | 0.969465 | 0.969465 | 0.961489 | 0.980391 | 0.969451 | 0.98231 |
| 0.005 | 0.901003 | 0.901003 | 0.922124 | 0.941498 | 0.900924 | 0.944488 |
| 0.01 | 0.849203 | 0.849203 | 0.891466 | 0.909213 | 0.849064 | 0.91029 |
| 0.02 | 0.783583 | 0.783583 | 0.847653 | 0.863001 | 0.783351 | 0.859649 |
| 0.03 | 0.741924 | 0.741924 | 0.818245 | 0.830021 | 0.741612 | 0.826176 |
| 0.04 | 0.708474 | 0.708474 | 0.792984 | 0.80374 | 0.708092 | 0.797163 |
| 0.05 | 0.68496 | 0.68496 | 0.77639 | 0.785539 | 0.684505 | 0.780942 |
| 0.06 | 0.658536 | 0.658536 | 0.754734 | 0.761634 | 0.658027 | 0.757305 |
| 0.07 | 0.641516 | 0.641516 | 0.74417 | 0.749587 | 0.640928 | 0.745526 |
| 0.08 | 0.620337 | 0.620337 | 0.722502 | 0.726334 | 0.619721 | 0.723376 |
| 0.09 | 0.604886 | 0.604886 | 0.708318 | 0.712633 | 0.604217 | 0.708979 |
| 0.1 | 0.59393 | 0.59393 | 0.699466 | 0.702528 | 0.593214 | 0.701294 |

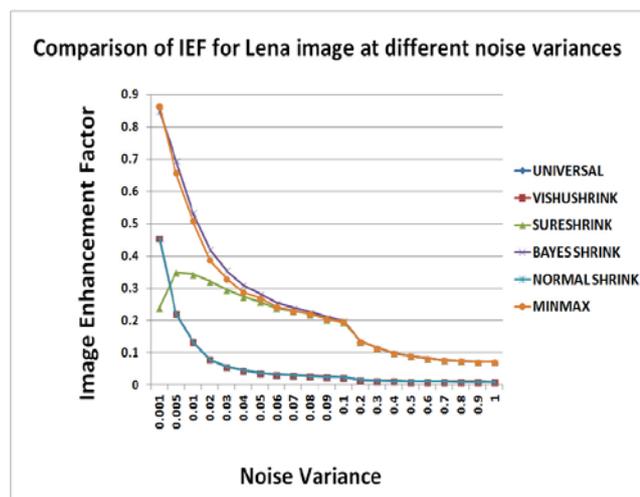
| | | | | | | |
|-----|----------|----------|----------|----------|----------|----------|
| 0.2 | 0.490857 | 0.490857 | 0.602508 | 0.602508 | 0.489728 | 0.602508 |
| 0.3 | 0.436399 | 0.436399 | 0.552589 | 0.552589 | 0.434903 | 0.552589 |
| 0.4 | 0.384161 | 0.384161 | 0.494463 | 0.494463 | 0.382402 | 0.494463 |
| 0.5 | 0.363174 | 0.363174 | 0.470306 | 0.470306 | 0.361147 | 0.470306 |
| 0.6 | 0.337012 | 0.337012 | 0.439264 | 0.439264 | 0.334746 | 0.439264 |
| 0.7 | 0.337012 | 0.337012 | 0.439264 | 0.439264 | 0.334746 | 0.439264 |
| 0.8 | 0.303117 | 0.303117 | 0.396626 | 0.396626 | 0.300529 | 0.396626 |
| 0.9 | 0.28282 | 0.28282 | 0.378427 | 0.378427 | 0.280007 | 0.378427 |
| 1 | 0.282146 | 0.282146 | 0.378552 | 0.378552 | 0.279066 | 0.378552 |



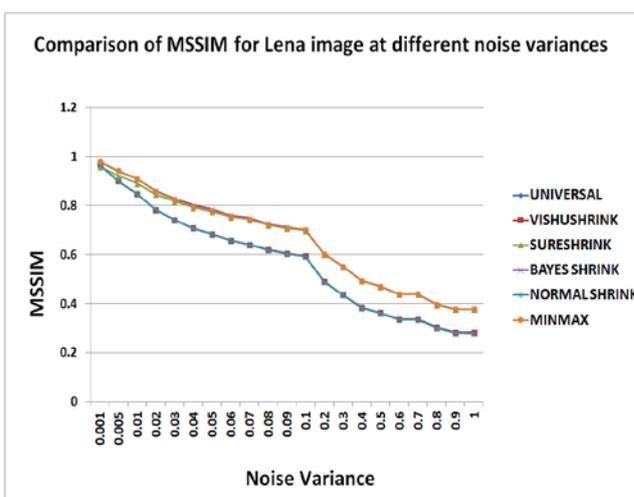
(a)



(b)



(c)



(d)

Figure 5: a) to d) – Comparison of MSE, PSNR, IEF and MSSIM for Lena image at various noise levels

Table 5: Comparison of MSE, PSNR, IEF and MSSIM for Barbara and Cameraman images (with additive white gaussian noise of density 0.01) with various wavelet filters namely db1,db2,coif1,coif5,sym2, sym8, bior1.1, bior2.2, rbio1.1, rbio2.2.(I- Universal Threshold, II – Vishushrink, III – Sureshshrink, IV - Bayes Shrink, V - Normal Shrink, VI - Minimax Threshold)

| PM | WF | Barbara | | | | | | Peppers | | | | | |
|---------|---------|---------|--------|--------|--------|--------|--------|---------|--------|--------|--------|--------|--------|
| | | I | II | III | IV | V | VI | I | II | III | IV | V | VI |
| MSE | db1 | 626.00 | 626.00 | 364.35 | 222.71 | 627.64 | 215.51 | 611.42 | 611.42 | 183.59 | 124.59 | 613.15 | 122.20 |
| | db2 | 629.07 | 629.07 | 317.97 | 190.39 | 630.76 | 175.08 | 611.22 | 611.22 | 137.47 | 106.19 | 612.99 | 100.68 |
| | coif1 | 630.67 | 630.67 | 316.92 | 189.00 | 632.37 | 173.10 | 606.57 | 606.57 | 130.90 | 103.77 | 608.33 | 98.35 |
| | coif5 | 626.40 | 626.40 | 297.84 | 162.06 | 628.12 | 139.71 | 611.12 | 611.12 | 116.79 | 95.51 | 612.89 | 91.31 |
| | sym2 | 628.66 | 628.66 | 318.38 | 190.89 | 630.35 | 176.73 | 610.76 | 610.76 | 137.14 | 105.49 | 612.53 | 99.66 |
| | sym8 | 625.41 | 625.41 | 297.70 | 163.79 | 627.13 | 142.35 | 610.96 | 610.96 | 118.26 | 98.24 | 612.73 | 91.71 |
| | bior1.1 | 629.87 | 629.87 | 363.34 | 223.09 | 631.52 | 215.08 | 611.32 | 611.32 | 184.90 | 125.64 | 613.06 | 124.28 |
| | bior2.2 | 626.94 | 626.94 | 303.94 | 281.78 | 628.68 | 179.24 | 609.85 | 609.85 | 125.32 | 228.72 | 611.64 | 117.07 |
| | rbio1.1 | 630.84 | 630.84 | 362.98 | 221.18 | 632.49 | 214.97 | 610.21 | 610.21 | 185.18 | 128.37 | 611.95 | 124.77 |
| rbio2.2 | 626.46 | 626.46 | 358.14 | 364.52 | 627.99 | 217.97 | 611.08 | 611.08 | 170.28 | 187.99 | 612.70 | 133.49 | |
| PSNR | db1 | 20.165 | 22.693 | 22.516 | 24.653 | 20.154 | 24.796 | 20.267 | 26.138 | 25.492 | 27.176 | 20.255 | 27.260 |
| | db2 | 20.144 | 23.294 | 23.107 | 25.334 | 20.132 | 25.698 | 20.269 | 27.088 | 26.749 | 27.870 | 20.256 | 28.101 |
| | coif1 | 20.133 | 23.333 | 23.121 | 25.366 | 20.121 | 25.748 | 20.302 | 27.258 | 26.961 | 27.970 | 20.289 | 28.203 |
| | coif5 | 20.162 | 23.758 | 23.391 | 26.034 | 20.150 | 26.678 | 20.270 | 27.693 | 27.457 | 28.330 | 20.257 | 28.525 |
| | sym2 | 20.147 | 23.280 | 23.101 | 25.323 | 20.135 | 25.658 | 20.272 | 27.093 | 26.759 | 27.900 | 20.260 | 28.145 |
| | sym8 | 20.169 | 23.736 | 23.393 | 25.988 | 20.157 | 26.597 | 20.271 | 27.666 | 27.402 | 28.208 | 20.258 | 28.506 |
| | bior1.1 | 20.138 | 22.709 | 22.528 | 24.646 | 20.127 | 24.805 | 20.268 | 26.082 | 25.461 | 27.139 | 20.256 | 27.187 |
| | bior2.2 | 20.159 | 23.823 | 23.303 | 23.632 | 20.146 | 25.596 | 20.279 | 27.191 | 27.151 | 24.538 | 20.266 | 27.446 |
| | rbio1.1 | 20.132 | 22.726 | 22.532 | 24.683 | 20.120 | 24.807 | 20.276 | 26.081 | 25.455 | 27.046 | 20.264 | 27.170 |
| rbio2.2 | 20.162 | 22.513 | 22.590 | 22.514 | 20.151 | 24.747 | 20.270 | 25.667 | 25.819 | 25.389 | 20.258 | 26.876 | |
| IEF | db1 | 0.138 | 0.138 | 0.238 | 0.389 | 0.138 | 0.402 | 0.142 | 0.142 | 0.474 | 0.698 | 0.142 | 0.712 |
| | db2 | 0.137 | 0.137 | 0.271 | 0.452 | 0.137 | 0.490 | 0.142 | 0.142 | 0.632 | 0.818 | 0.142 | 0.863 |
| | coif1 | 0.138 | 0.138 | 0.275 | 0.461 | 0.138 | 0.503 | 0.142 | 0.142 | 0.657 | 0.829 | 0.141 | 0.875 |
| | coif5 | 0.138 | 0.138 | 0.290 | 0.533 | 0.137 | 0.618 | 0.142 | 0.142 | 0.744 | 0.909 | 0.142 | 0.951 |
| | sym2 | 0.137 | 0.137 | 0.271 | 0.451 | 0.137 | 0.487 | 0.142 | 0.142 | 0.633 | 0.823 | 0.142 | 0.871 |
| | sym8 | 0.138 | 0.138 | 0.290 | 0.528 | 0.138 | 0.607 | 0.142 | 0.142 | 0.735 | 0.885 | 0.142 | 0.948 |
| | bior1.1 | 0.138 | 0.138 | 0.239 | 0.389 | 0.137 | 0.403 | 0.141 | 0.141 | 0.466 | 0.686 | 0.141 | 0.693 |
| | bior2.2 | 0.137 | 0.137 | 0.283 | 0.305 | 0.137 | 0.480 | 0.142 | 0.142 | 0.691 | 0.379 | 0.142 | 0.739 |
| | rbio1.1 | 0.137 | 0.137 | 0.239 | 0.392 | 0.137 | 0.403 | 0.141 | 0.141 | 0.466 | 0.672 | 0.141 | 0.691 |
| rbio2.2 | 0.138 | 0.138 | 0.242 | 0.237 | 0.138 | 0.397 | 0.141 | 0.141 | 0.507 | 0.459 | 0.141 | 0.647 | |
| MSSIM | db1 | 0.840 | 0.840 | 0.853 | 0.889 | 0.840 | 0.889 | 0.786 | 0.786 | 0.875 | 0.873 | 0.786 | 0.881 |
| | db2 | 0.839 | 0.839 | 0.873 | 0.905 | 0.839 | 0.910 | 0.787 | 0.787 | 0.901 | 0.903 | 0.787 | 0.907 |
| | coif1 | 0.840 | 0.840 | 0.871 | 0.903 | 0.840 | 0.908 | 0.787 | 0.787 | 0.900 | 0.898 | 0.787 | 0.905 |
| | coif5 | 0.839 | 0.839 | 0.875 | 0.907 | 0.839 | 0.916 | 0.786 | 0.786 | 0.899 | 0.898 | 0.786 | 0.902 |
| | sym2 | 0.840 | 0.840 | 0.872 | 0.906 | 0.840 | 0.909 | 0.787 | 0.787 | 0.902 | 0.903 | 0.786 | 0.907 |
| | sym8 | 0.841 | 0.841 | 0.877 | 0.909 | 0.841 | 0.917 | 0.786 | 0.786 | 0.900 | 0.897 | 0.786 | 0.904 |
| | bior1.1 | 0.839 | 0.839 | 0.852 | 0.888 | 0.839 | 0.888 | 0.787 | 0.787 | 0.875 | 0.874 | 0.787 | 0.882 |
| | bior2.2 | 0.840 | 0.840 | 0.869 | 0.872 | 0.840 | 0.900 | 0.788 | 0.788 | 0.893 | 0.840 | 0.788 | 0.881 |
| | rbio1.1 | 0.839 | 0.839 | 0.853 | 0.888 | 0.839 | 0.889 | 0.787 | 0.787 | 0.875 | 0.875 | 0.787 | 0.882 |
| rbio2.2 | 0.840 | 0.840 | 0.862 | 0.861 | 0.840 | 0.898 | 0.787 | 0.787 | 0.891 | 0.890 | 0.787 | 0.896 | |

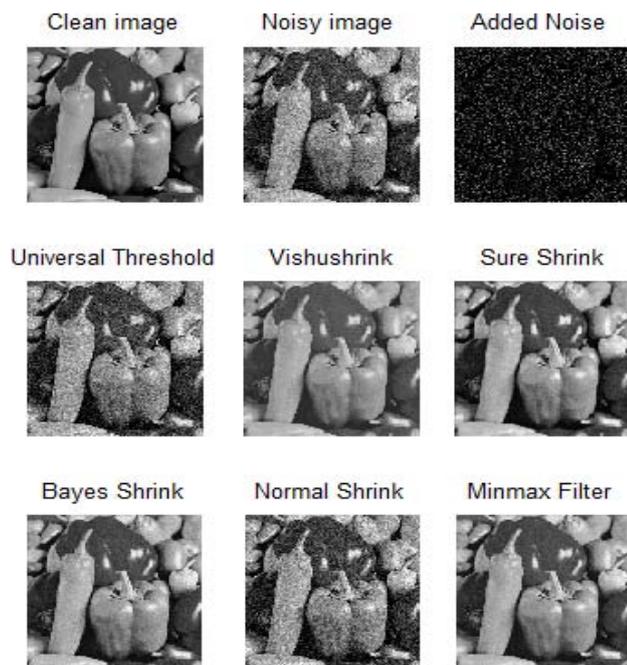


Fig 6: Denoised images (Peppers, AWGAN with variance 0.01, wname -db1)

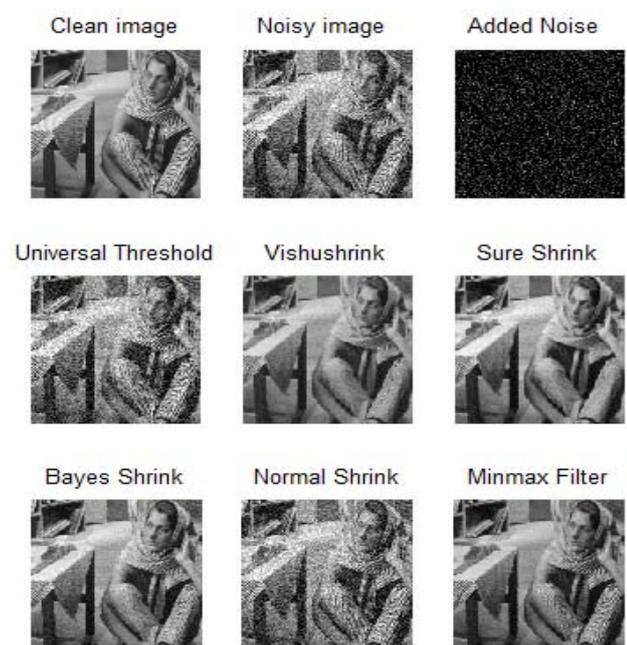


Fig 7: Denoised images (Barbara, AWGAN with variance 0.01, wname -db1)

5 Conclusion

An empirical study of wavelet based thresholding methods for image denoising is presented in this paper. It is identified that, Wavelet transform is an

efficient tool for image denoising and the optimum threshold value determines the goodness of the denoising algorithm. The experimental result shows that Sure shrink performs well in terms of improving visual quality for both smooth and detailed images among the shrinkage methods. Bayes shrink performs considerably better in improving visual quality.

However, the images denoised by wavelet based denoising are prone to checkerboard artifacts due to the limited directional selectivity of wavelets. This effect is resolved with the use of highly directional representations to improve denoising performance.

Acknowledgment

The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, that have improved the presentation.

References:

- [1] Rafael C.Gonzalez, Richard E.Woods (2002), "Digital Image Processing", Second Edition, Pearson Education Asia.
- [2] Jain, A. K. (1989). *Fundamentals of digital image processing* (Vol. 569). Englewood Cliffs, NJ:: Prentice hall.
- [3] Vaseghi, S. V. (2008). *Advanced digital signal processing and noise reduction*. Wiley.
- [4] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm>
- [5] Yu, Guoshen, and Guillermo Sapiro. "DCT image denoising: a simple and effective image denoising algorithm." *Image Processing On Line* 2011 (2011). <http://dx.doi.org/10.5201/ipol.2011.yg-dct>
- [6] Foi, A., K. Dabov, V. Katkovnik, and K. Egiazarian, "Shape-Adaptive DCT for Denoising and Image Reconstruction", *Proc. SPIE Electronic Imaging 2006, Image Processing: Algorithms and Systems V*, 6064A-18, San Jose, 2006.
- [7] Stephane Mallat, "A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way", 3rd edn, Academic Press ©2008 ISBN:0123743702 9780123743701
- [8] Stephane Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. II, NO. 7. JULY 1989

- [9] E.Jebamalar Leavline, S.Sutha and D.Asir Antony Gnana Singh. Article: *Wavelet Domain Shrinkage Methods for Noise Removal in Images: A Compendium*. International Journal of Computer Applications 33(10):28-32, November 2011. Published by Foundation of Computer Science, New York, USA
- [10] S.Grace Chang, Bin Yu, Martin Vetterli 2000, "Adaptive wavelet thresholding for denoising and compression", IEEE Trans. On Image processing, Vol.9, No.9, PP. 1532-1546.
- [11] Nasri M, Nezamabadi-pour H: *Image denoising in the wavelet domain using a new adaptive thresholding function*. Neurocomputing 2009, 72 (4-6) 1012-1025.
- [12] Akansu AN, Serdijn WA, Selesnick IW: *Emerging applications of wavelets: A review*. Physical Communication 2010, 3 (1) 1–18.
- [13] Cho D, Bui TD, Chen G: *Image denoising based on wavelet shrinkage using neighbor and level dependency*. Int. J. Wavelets Multiresolut. Inf. Process 2009, 7 (3) 299-311.
- [14] Abdourrahmane A, Dominique P, Mercier G: *Wavelet shrinkage: unification of basic thresholding functions and thresholds*. Signal, Image and Video Processing 2010, 5 (1) 11-28
- [15] Donoho, D. L., & Johnston, J. M. (1994). *Ideal spatial adaptation by wavelet shrinkage*. Biometrika, 81(3), 425-455.
- [16] Donoho, D. L. (1995). *De-noising by soft-thresholding*. Information Theory, IEEE Transactions on, 41(3), 613-627.
- [17] Valens, C. (1999). *A really friendly guide to wavelets*. URL <http://perso.orange.fr/polyvalens/clemens/wavelets/wavelets.html>, 11.
- [18] Vetterli, Martin, and Jelena Kovačević. *Wavelets and subband coding*. Upper Saddle River, NJ: Prentice Hall PTR, 1995.
- [19] Donoho, D. L., & Johnston, I. M. (1994, November). *Threshold selection for wavelet shrinkage of noisy data*. In Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE (pp. A24-A25). IEEE.
- [20] Yan, F., Cheng, L., & Peng, S. (2008). *A new interscale and intrascale orthonormal wavelet thresholding for SURE-based image denoising*. Signal Processing Letters, IEEE, 15, 139-142.
- [21] Thierry Blu, and Florian Luisier "The SURE-LET Approach to Image Denoising". IEEE Transactions On Image Processing, VOL. 16, NO. 11, pp 2778 – 2786 , NOV 2007.
- [22] Raphan, M., & Simoncelli, E. P. (2008). *Optimal denoising in redundant representations*. Image Processing, IEEE Transactions on, 17(8), 1342-1352.
- [23] S.Grace Chang, Bin Yu, Martin Vetterli 2000, "Adaptive wavelet thresholding for denoising and compression", IEEE Trans. On Image processing, Vol.9, No.9, PP. 1532-1546.
- [24] Iman Elyasi, and Sadegh Zarmehi 2009," Elimination Noise by Adaptive Wavelet Threshold" World Academy of Science, Engineering and Technology.
- [25] Lakhwinder Kaur, Savita Gupta and R.C. Chauhan, "Image Denoising using Wavelet Thresholding".
- [26] Rosas-Orea, M. C. E., Hernandez-Diaz, M., Alarcon-Aquino, V., & Guerrero-Ojeda, L. G. (2005, February). *A comparative simulation study of wavelet based denoising algorithms*. In Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Proceedings. 15th International Conference on (pp. 125-130). IEEE.
- [27] Sardy, S. (2000). *Minimax threshold for denoising complex signals with waveshrink*. Signal Processing, IEEE Transactions on, 48(4), 1023-1028.
- [28] Donoho, D. L., & Johnston, I. M. (1998). *Minimax estimation via wavelet shrinkage*. The Annals of Statistics, 26(3), 879-921.