

WAVELET BASED IMAGE DENOISING BASED ON A MIXTURE OF LAPLACE DISTRIBUTIONS*

H. RABBANI AND M. VAFADOOST**

Dept. of Biomedical Engineering, Amirkabir University of Technology, Tehran I. R. of Iran
Email: vmansur@cic.aut.ac.ir

Abstract– The performance of various estimators, such as *maximum a posteriori* (MAP), strongly depends on correctness of the proposed model for distribution of noise-free data. Therefore, the selection of a proper model for the distribution of wavelet coefficients is very important in wavelet based image denoising. This paper presents a new image denoising algorithm based on the modeling of wavelet coefficients in each subband with a mixture of Laplace random variables. Indeed, we design a MAP estimator which relies on mixture distributions. Using this relatively new statistical model we are better able to capture the heavy-tailed nature of wavelet coefficients. The simulation results show that our proposed technique achieves better performance than several published methods, both visually and in terms of *root mean squared error* (RMSE).

Keywords– MAP estimator, mixture model, wavelet transforms

1. INTRODUCTION

Usually, noise reduction—the process of noise-free data estimation from noisy data observation—is an essential part of many image processing systems (Fig. 1). The main sources of noise arise from the imaging devices during image formation and channels during transmission [1]. A suitable noise reduction algorithm is used to reconstruct the main information of the image, so that the obtained image will have the greatest *peak signal-to-noise ratio* (PSNR) and least visual artifacts [1]. In recent years there has been a fair amount of research on wavelet-based image de-noising [2-8]. The motivation of denoising in the wavelet domain is that, while the wavelet transform is good at energy compaction, the small coefficients are more likely caused by noise, and the large coefficients caused by important signal features [5]. The small coefficients can be thresholded without affecting the significant features of the image [3]. Thresholding is a simple non-linear technique, which usually operates on one wavelet coefficient at a time [2]. In its most basic form, each coefficient is thresholded by comparing against the threshold: if the coefficient is smaller than the threshold, set to zero; otherwise it is kept or modified. Replacing the small noisy coefficients by zero and applying the inverse wavelet transform on the result may lead to reconstruction with the essential signal characteristics and with less noise [3].

Many of the wavelet based denoising algorithms have been developed based on soft thresholding proposed by Donoho [2], and examples of alternative approaches can be found in [5-11]. Generally, these methods lead to a threshold value that must be estimated correctly in order to obtain a good performance. Early methods, such as VisuShrink [3] use a universal threshold, while more recent ones, such as SureShrink [4] are subband adaptive algorithms and have better performance. BayesShrink [5], which is also a data-driven subband adaptive technique, outperforms both Visu-Shrink and SureShrink.

*Received by the editors May 23, 2005; final revised form July 9, 2006.

**Corresponding author

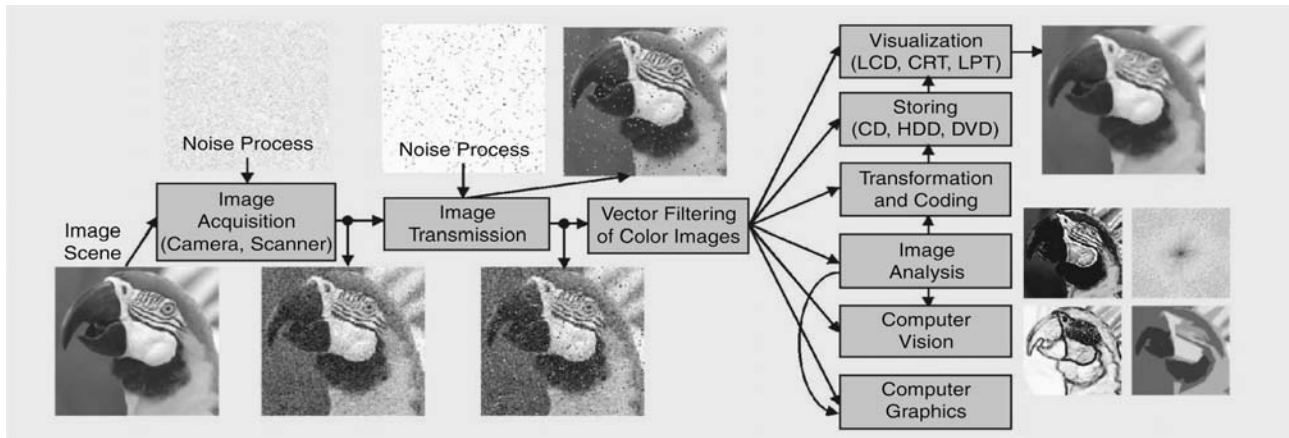


Fig. 1. Image processing chain [1]

The problem of wavelet based image denoising can be expressed as an estimation of clean coefficients from noisy data with Bayesian estimation techniques. If the MAP estimator is used for this problem, the solution requires *a priori* knowledge about the distribution of wavelet coefficients. Based on the distribution type, the corresponding estimator (shrinkage function) is obtained.

Various *probability density functions* (pdfs) such as Gaussian, Laplace, generalized Gaussian or other distributions were proposed for modeling noise-free wavelet coefficients [11-12]. For example, the classical soft threshold shrinkage function can be obtained by a Laplacian assumption. Bayesian methods for image denoising using other distributions have also been proposed [13-17].

In this paper we use a mixture of Laplace random variables to model the wavelet coefficients in each subband. Because the energy compactness property of the wavelet makes it reasonable to assume that essentially, only a few large coefficients contain information about the underlying image, the marginal distribution of wavelet coefficients is highly kurtotic, and can be described using suitable long-tailed distributions [11]. In [18], the wavelet-based *hidden Markov model* (HMM) is proposed for statistical signal processing and a mixture of Gaussian distributions is used for modeling this heavy-tailed property of wavelet coefficients. Our approach is similar to the method reported in [18], but we use Laplace components instead of Gaussian components. Because Laplace pdf has a large peak at zero and its tails fall significantly slower than a Gaussian pdf of the same variance, a mixture of Laplace pdfs can improve the modeling of wavelet coefficients distribution.

The rest of this paper is organized as follows. After a brief review on the basic idea of Bayesian denoising in Section 2, we describe how soft thresholding can be obtained using the Laplace pdf in Section 2a. To apply the soft threshold rule, we need to know some parameters. The estimation of these parameters is described in Section 2a.1. In Section 2b the theoretical base of denoising with the Laplacian mixture model is introduced. In order to be able to compare the ability of our model to capture the heavy-tailed property of wavelet coefficients in each subband with a simple model, both single Laplace and mixed Laplace are used for modeling the histograms of wavelet coefficients in Section 2a.2 and Section 2b.1 respectively. We obtain the shrinkage function derived from our Laplacian mixture model namely, LapMixShrink, in Section 2b.2. In Section 3 we use our model for wavelet-based denoising of several images corrupted with additive Gaussian noise at various noise levels. The simulation results in comparison with the VisuShrink, SureShrink, BayesShrink and *hidden Markov tree* (HMT) show that our algorithm achieves better performance, both visually and in terms of RMSE. Finally the concluding remarks are given in Section 4. To apply the LapMixShrink rule we need to implement the *Expectation Maximization* (EM) algorithm to determine the parameters of a mixture model. A simple description of the EM algorithm can be found in the Appendix.

2. BAYESIAN DENOISING

In this section, the denoising of an image corrupted by white Gaussian noise will be considered. We observe a noisy signal $g = x + n$ where n is independent, white, zero-mean Gaussian noise and we wish to estimate the noise-free signal x as accurately as possible according to some criteria [3]. In the wavelet domain, if we use an orthogonal wavelet transform, the problem can be formulated as $y = w + n$, where y is the noisy wavelet coefficient, w is the noise-free wavelet coefficient, and n is noise, which is independent white zero mean Gaussian [3].

If $p_W(w)$ denotes pdf of random variable W for $W = w$ and $p_{W|Y}(w|y)$ denotes the conditional pdf of random variable W for $W = w$ when given random variable Y for $Y = y$, the MAP estimator below will be used to estimate w from the noisy observation y [5]. This estimator is defined as

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} p_{w|y}(w|y) \quad (1)$$

Using Bayesian rule [5] we get

$$p_{w|y}(w|y) = \frac{p_{y|w}(y|w) p_w(w)}{p_y(y)}$$

Therefore, one gets

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} \frac{p_{y|w}(y|w) p_w(w)}{p_y(y)}$$

Because the term $p_y(y)$ does not depend on w , the value of w that maximizes the right hand side is not influenced by the denominator. Therefore the MAP estimate of w is given by

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} [p_{y|w}(y|w) p_w(w)]$$

Because y is the sum of w and n , a zero-mean Gaussian pdf, when w is a known constant, y will be a Gaussian pdf with mean w . Therefore, if the pdf of n is $p_n(n)$, then $p_{y|w}(y|w)$ will be $p_n(y-w)$.

Thus, Eq. (1) can be written as

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} [p_n(y-w) \cdot p_w(w)] \quad (2)$$

Equation (2) is also equivalent to

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} [\log(p_n(y-w)) + f(w)] \quad (3)$$

where $f(w) = \log(p_w(w))$.

We have assumed the noise is zero mean Gaussian with variance σ_n ,

$$p_n(n) = \text{Gaussian}(n, \sigma_n) := \frac{1}{\sigma_n \sqrt{2\pi}} \cdot \exp\left(-\frac{n^2}{2\sigma_n^2}\right) \quad (4)$$

Replacing (4) in (3) yields

$$\hat{w}(y) = \underset{w}{\operatorname{argmax}} \left[-\frac{(y-w)^2}{2\sigma_n^2} + f(w) \right]$$

Therefore we can obtain the MAP estimate of w by setting the derivative with respect to \hat{w} equal to zero. That gives the following equation to solve for \hat{w} .

$$\frac{y - \hat{w}}{\sigma_n^2} + f'(\hat{w}) = 0 \quad (5)$$

For example, if $p_w(w)$ is assumed to be *Gaussian*(w, σ), then $f(w) = -\log(\sqrt{2\pi}\sigma) - w^2/2\sigma^2$, and the estimator can be written as

$$\hat{w}(y) = \frac{\sigma^2}{\sigma^2 + \sigma_n^2} y \quad (6)$$

a) Soft thresholding

We now need to model $p_w(w)$ for the distribution of wavelet coefficients. The pdf for wavelet coefficients $p_w(w)$ is often modeled as a generalized (heavy-tailed) Gaussian [7],

$$p_w(w) = K(s, q). \exp\left(-\left|\frac{w}{s}\right|^q\right) \quad (7)$$

where s, q are the parameters for this model, and $K(s, q)$ is the normalization constant (which depends on s and q). Other pdf models have also been proposed [7-17].

Substitution of $q = 1$ in (7) simplifies the equation to a Laplace pdf,

$$p_w(w) = \text{Laplace}(w, \sigma) := \frac{1}{\sigma\sqrt{2}} \exp\left(-\frac{\sqrt{2}}{\sigma}|w|\right) \quad (8)$$

In this case

$$f(w) = -\log(\sigma\sqrt{2}) - \frac{\sqrt{2}}{\sigma}|w|$$

and so

$$f'(w) = -\frac{\sqrt{2}}{\sigma} \cdot \text{sign}(w)$$

therefore

$$y = \hat{w} + \frac{\sqrt{2}\sigma_n^2}{\sigma} \cdot \text{sign}(\hat{w})$$

thus

$$\hat{w} = \begin{cases} y + T, & y < -T \\ 0, & -T \leq y \leq T \\ y - T, & T < y \end{cases}$$

where $T = \frac{\sqrt{2}\sigma_n^2}{\sigma}$.

This is the soft threshold nonlinearity based on the MAP estimator. Figure 2 shows a graph of \hat{w} as a function of y . Other approaches give different formulas for choosing the threshold [11, 17]. The formula is often written in the following way

$$\hat{w}(y) = \text{sign}(y) \cdot \left(|y| - \frac{\sqrt{2}\sigma_n^2}{\sigma}\right)_+ \quad (9)$$

Here $(a)_+$ is defined as

$$(a)_+ = \begin{cases} 0 & \text{if } a < 0 \\ a & \text{otherwise} \end{cases}$$

Let's define the soft operator as

$$\text{soft}(g, \tau) := \text{sign}(g) \cdot (|g| - \tau)_+$$

The soft shrinkage function (9) can be written as

$$\hat{w}(y) = \text{soft}(y, \frac{\sqrt{2} \sigma_n^2}{\sigma}) \quad (10)$$

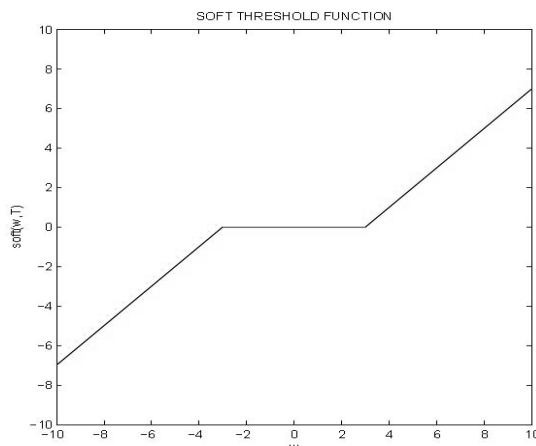


Fig. 2. Shrinkage function corresponding to the Laplace pdf (soft threshold)

The main idea in soft thresholding is to subtract the threshold value T from all coefficients larger than T and to set all other coefficients to zero. The threshold $T = \sqrt{2} \sigma_n^2 / \sigma$ has an intuitive appeal. The normalized threshold, $T / \sigma_n = \sqrt{2} \sigma_n / \sigma$, is inversely proportional to σ , the standard deviation of w , and proportional to σ_n , the noise standard deviation. When $\sigma_n / \sigma \ll 1$, the signal is much stronger than the noise. Therefore T / σ_n is chosen to be small in order to preserve most of the signal and remove some of the noise. Vice versa, when $\sigma_n / \sigma \gg 1$, the noise dominates. In this case, the normalized threshold is chosen to be large to remove the noise, which has overwhelmed the signal. Thus, this threshold is adapted to both the signal and noise characteristics, which are reflected in parameters σ_n and σ [5].

1. Parameters estimation: To apply the soft threshold rule we need to know σ_n and σ . Experiments show that σ is quite different from scale to scale. Figure 3 shows the standard deviation of each wavelet subband for the Lena image. Thus, we must estimate a different σ for each subband only from the noisy data. In fact, the results lead to a subband-dependent threshold.

Because the wavelet coefficients of the noise free image and the noise are independent, we have

$$VAR[y] = VAR[w] + VAR[n]$$

in which $VAR[x]$ is variance of the random variable x .

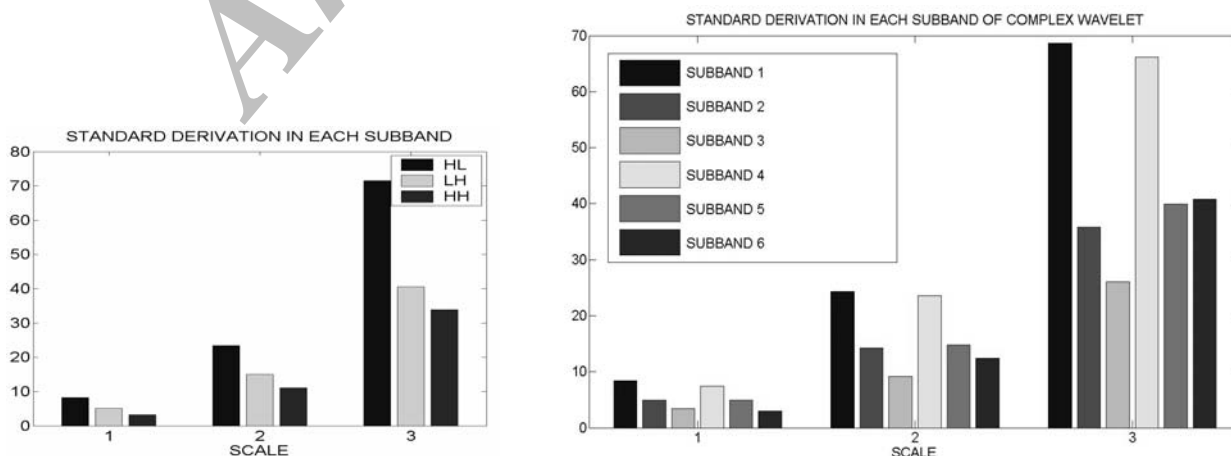


Fig. 3. Standard deviation of each wavelet subband for Lena image

As we assume that the variance of the noise is known, we write

$$\sigma^2 = \text{VAR}[y] - \sigma_n^2$$

The variance of y can be computed from each subband using the standard formula [3], where we assume all quantities are zero mean,

$$\text{VAR}[y] = \text{MEAN}[y^2]$$

where $\text{MEAN}[x]$ is the empirical mean [3] of x .

So we estimate σ as

$$\hat{\sigma} = \sqrt{\text{MEAN}[y^2] - \sigma_n^2}$$

In case we have a negative value under the square root (it is possible because these are estimates) we can use

$$\hat{\sigma} = \sqrt{\max(\text{MEAN}[y^2] - \sigma_n^2, 0)} \quad (11)$$

When σ_n is unknown, to estimate the noise variance from the noisy wavelet coefficients, a robust median estimator is used from the finest scale wavelet coefficients [3].

$$\sigma_n^2 = \frac{\text{median}(|y_i|)}{0.6745}, \quad (12)$$

$y_i \in \text{subband HH in finest scale}$

2. Modeling wavelet coefficients as a single Laplace distribution: In this section, a single Laplace pdf is used to model a histogram of a 512×512 Lena image in each subband. Figure 4 illustrates the histograms of the wavelet coefficients in the second scale and the best Laplace pdf is fitted to these histograms.

We use soft thresholding for noise reduction of the 512×512 Lena image. Zero mean white Gaussian noise is added to the original image ($\sigma_n = 10$). The RMSE between the original image and the processed image in a standard wavelet domain is calculated to be 4.97. The RMSE between the original image and the noisy image is simply σ_n , the standard deviation of the noise, for this example is set at 10. Therefore, the wavelet domain soft thresholding reduced this noise level by more than a factor of 2. Figure 5 shows a break down of the square error in each subband. Because a few large coefficients which correspond to coarser scales represent the main features of the signal, noise mostly affects small coefficients corresponding to the finer scales. Therefore, we can see in this figure that most of the square error occurs in the finer scales. The original image, the noisy image and the denoised image produced using the soft threshold, are illustrated in Fig. 6.

Due to the effectiveness and simplicity of soft thresholding, it is frequently used in the literature, but other shrinkage functions have also been proposed [13-17]. For example, Fig. 7 shows the differences between soft thresholding, hard thresholding [2-4] and garrot thresholding [8].

Both the hard and soft shrinkages have advantages and disadvantages. The soft shrinkage estimates tend to have a bigger bias, due to the shrinkage of large coefficients. Due to the discontinuities of the shrinkage function, the hard shrinkage estimates tend to have a larger variance and can be unstable—that is, sensitive to small changes in the data. Garrot shrinkage offers smaller RMSE than hard thresholding and a bigger RMSE than soft thresholding. Also, it generally has less sensitivity to small perturbations in the data than hard thresholding and a smaller bias than soft shrinkage [8].

Although soft thresholding usually has acceptable results for denoising, it does not have a good performance for other kinds of noise such as Poisson, which is signal-dependent. In this case, algorithms that use local variances have better results [17, 19].

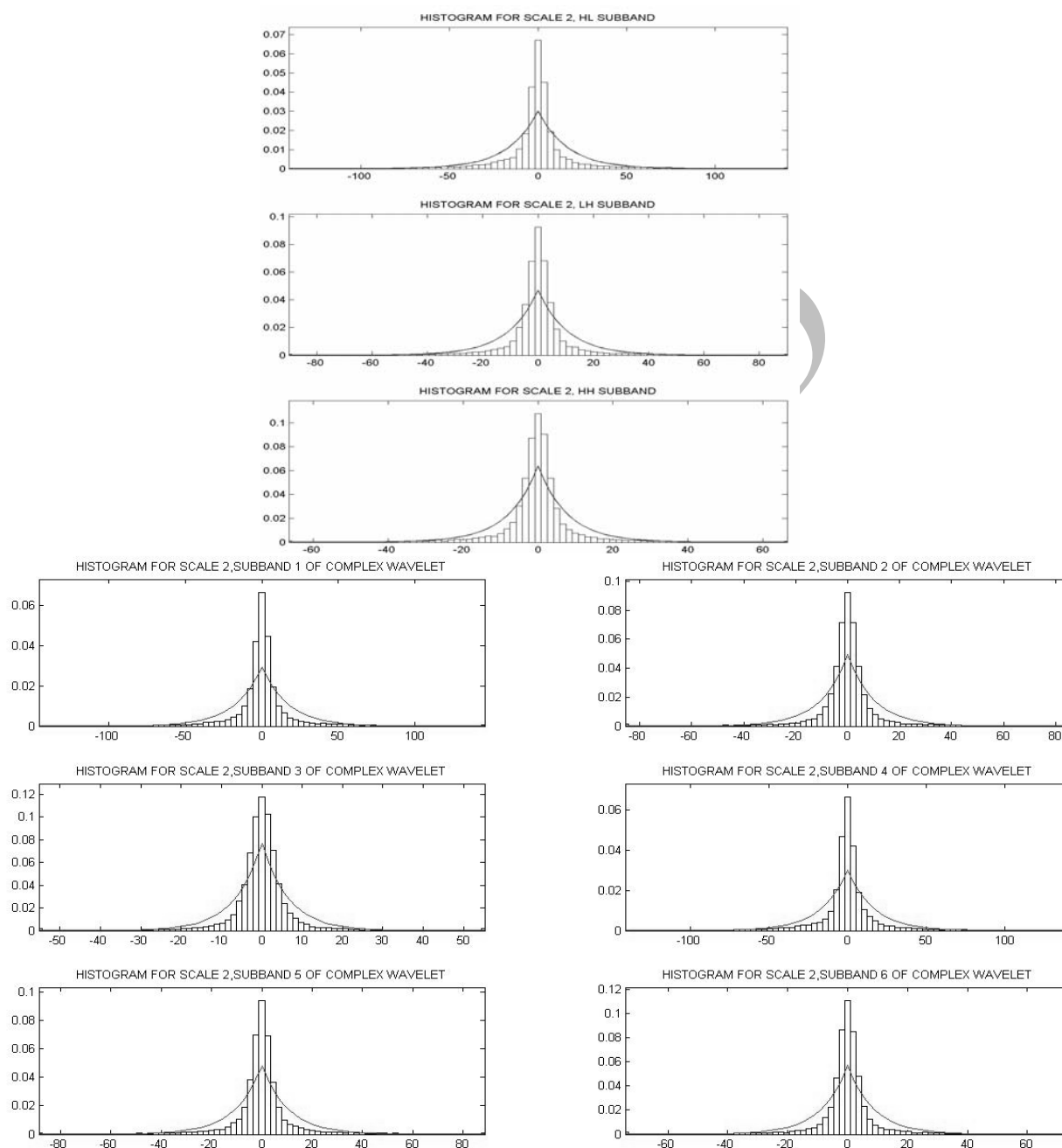


Fig. 4. Histograms of the wavelet coefficients and the best fitted Laplace model in second scale of Lena image

b) Denoising based on mixture models

A mixture model for a random variable has a pdf that is the sum of two simpler pdfs,

$$p(w) = ap_1(w) + (1 - a)p_2(w) \quad (13)$$

When $p_1(w)$ and $p_2(w)$ are two nonnegative functions that integrate into 1, then $p(w)$ will be a valid pdf.

If each $p_1(w)$ and $p_2(w)$ has one parameter such as their variance, then $p(w)$ will have 3 parameters. The mixture model has more parameters than either $p_1(w)$ or $p_2(w)$ has alone, so it is more flexible for

matching the histogram of a given dataset. If there are too many parameters, then it will be difficult to estimate those parameters accurately from the data. There is a trade off between the number of parameters and the ability to estimate them.

In the following sections we will use a mixture of two Laplace pdfs:

$$\begin{aligned}
 p_w(w) &= a \cdot \text{Laplace}(w, \sigma_1) + (1-a) \cdot \text{Laplace}(w, \sigma_2) \\
 &= a \frac{1}{\sigma_1 \sqrt{2}} \exp\left(-\frac{\sqrt{2}}{\sigma_1} |w|\right) + (1-a) \frac{1}{\sigma_2 \sqrt{2}} \exp\left(-\frac{\sqrt{2}}{\sigma_2} |w|\right)
 \end{aligned}
 \tag{14}$$

to model the distribution of wavelet coefficients of images. It will be necessary to estimate the three parameters σ_1, σ_2 and a from the data. While σ_1 and σ_2 represent the standard deviation of the individual components, they are not very easily related to the standard deviation of the random variable w , for example

$$\text{VAR}[w] \neq a^2 \sigma_1^2 + (1-a^2) \sigma_2^2$$

Nor are other simple relations available. The estimation of the three parameters is more difficult than it is for a single component model. For a mixture model, an iterative numerical algorithm is required to estimate the parameters. The most frequently used algorithm to determine the parameters of a mixture model is the EM algorithm. A simple description of the EM algorithm can be found in the Appendix.

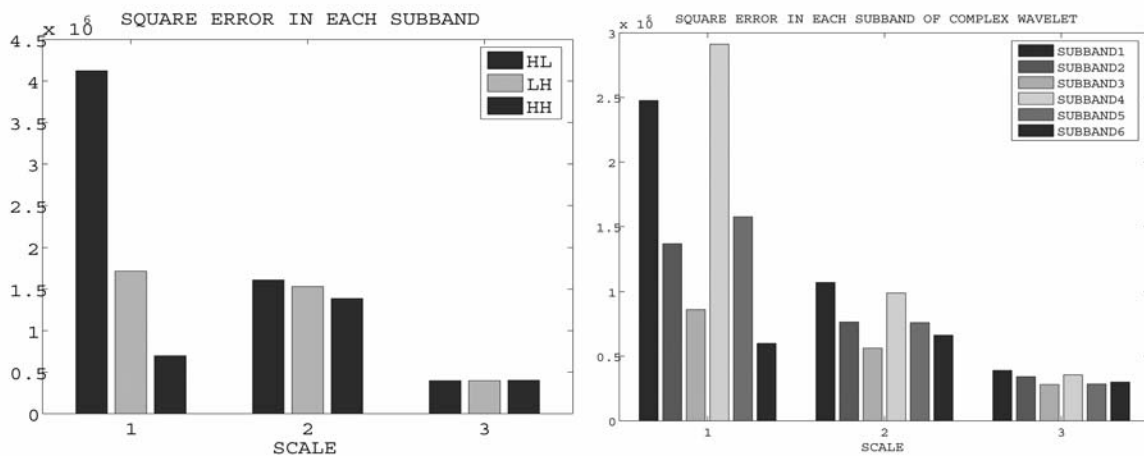


Fig. 5. Square error in each subband of Lena image after soft thresholding



Fig. 6. Soft thresholding of Lena image corrupted with additive Gaussian noise with standard deviation 10

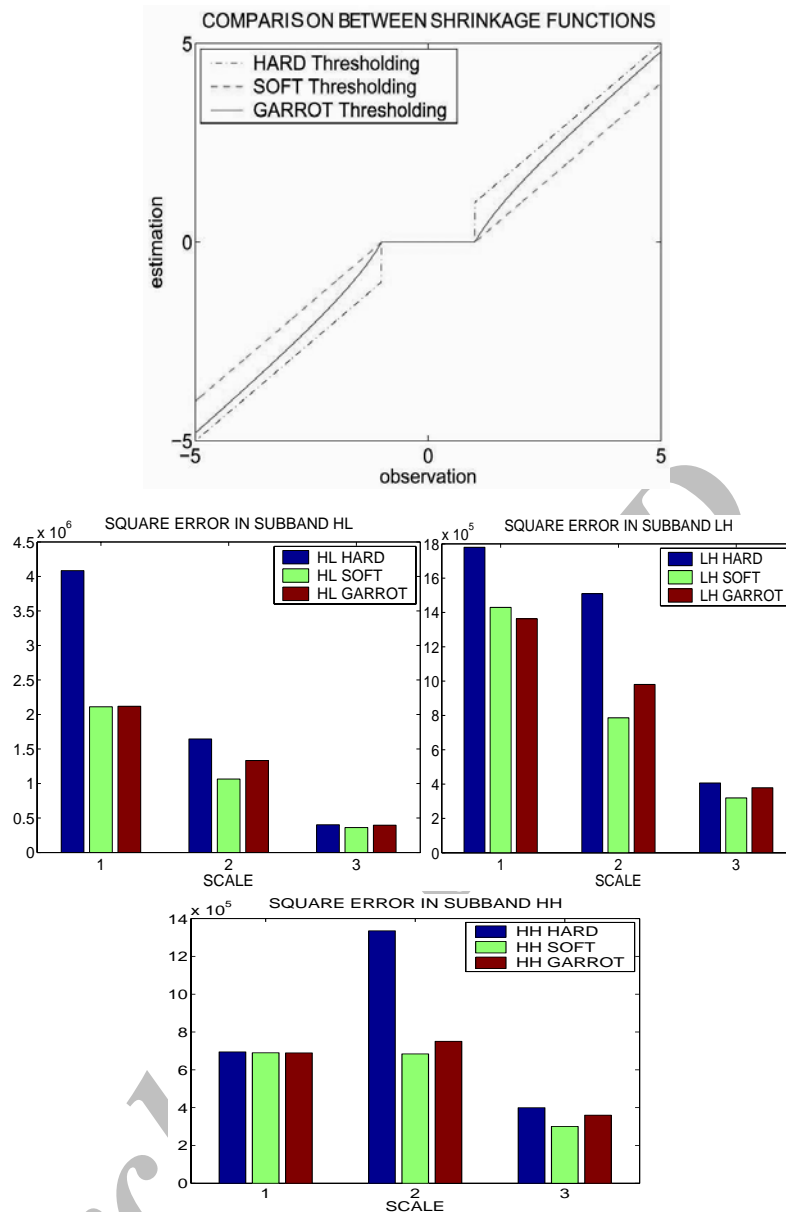


Fig. 7. Difference between soft, hard and garrot thresholding and RMSE in each subband of Lena image after denoising with these methods

Note that the random variable w in (13) is not the result of adding two random variables. If that were the case, then $p(w)$ would be a convolution of $p_1(w)$ and $p_2(w)$. Instead, w can be generated using a two step procedure. First, generate a binary random variable v according to

$$p(v=1) = a, \quad p(v=2) = 1-a$$

The value of v will be either 1 or 2. For $v=1$, p_1 is used to generate w , while for $v=2$, p_2 is used to generate w . Because this procedure produces a random variable w with the pdf in Eq. (13), w can be considered as being generated by either p_1 or by p_2 (even if that is not how w is physically produced).

1. Modeling wavelet coefficients as a mixture of Laplace pdfs: In [18] a mixture of two Gaussian pdfs is proposed for modeling wavelet coefficients distribution

$$p_w(w) = a \cdot \text{Gaussian}(w; \sigma_1) + (1-a) \cdot \text{Gaussian}(w; \sigma_2) \quad (15)$$

Because the Laplace pdf has a large peak at zero and tails that fall significantly slower than a Gaussian pdf of the same variance, a mixture of Laplace pdfs can improve the modeling of wavelet coefficients distribution.

In this section, a mixture of two Laplace distributions is used to model the wavelet coefficients of a 512×512 Lena image in each subband. Figure 8 shows the mixture pdf obtained using the EM algorithm, together with the histogram for the second scale. We see that the mixture of two Laplace pdfs follows the histogram much more closely than both the Gaussian mixture model and a single Laplace pdf (Fig. 9).

The nonlinear soft threshold rule for wavelet-based image denoising is the MAP estimator of the coefficients in Gaussian noise when the noise free coefficients are distributed according to the Laplace distribution. However, the plots in this section show that the Laplace distribution is not always an accurate model for the distribution of the noise-free coefficients, therefore, an alternative non-linearity derived using the mixture model may work more effectively than the soft threshold rule. In the next section we will illustrate the nonlinear shrinkage rule derived from the Laplacian mixture model, and later compare its performance with the soft threshold rule.

2. Threshold functions derived from the mixture models: This section describes a non-linear shrinkage function for wavelet-based denoising derived by assuming that noise-free wavelet coefficients follow a mixture model. Specifically, we assume that the noise-free wavelet coefficients are modeled as a mixture of two Laplace random variables.

If w follows the mixture pdf, $p(w) = ap_1(w) + bp_2(w)$ where $a + b = 1$ and p_1 and p_2 are valid pdfs individually, then how can we estimate w from a noisy observation $y = w + n$, where n is an independent zero-mean Gaussian random variable with standard deviation σ_n ? Because the estimate of w depends on y , it is denoted by $\hat{w}(y)$.

One way to obtain an estimate is by the following rule:

$$\hat{w}(y) = p_a(y)\hat{w}_1(y) + p_b(y)\hat{w}_2(y) \quad (16)$$

where $p_a(y)$ is the probability that w was generated by p_1 , and where similarly, $p_b(y)$ is the probability that w was generated by p_2 . The expression $\hat{w}_1(y)$ is an estimate of w based on the assumption that w was generated by p_1 , and that similarly $\hat{w}_2(y)$ is an estimate of w based on the assumption that w was generated by p_2 . If p_1 and p_2 are Laplace pdfs with parameters σ_1 and σ_2 respectively, then the soft threshold function can be used to get $\hat{w}_1(y)$ and $\hat{w}_2(y)$. We would have

$$\hat{w}(y) = p_a(y)\text{soft}(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_1}) + p_b(y)\text{soft}(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_2}),$$

but we still need to determine $p_a(y)$ and $p_b(y)$. For these values we can use the formulas based on Bayes theorem [5] as follows:

$$p_a(y) = \frac{ag_1(y)}{ag_1(y) + bg_2(y)} \quad (17)$$

$$p_b(y) = \frac{bg_2(y)}{ag_1(y) + bg_2(y)} \quad (18)$$

where $g_1(y)$ is the pdf of y under the assumption that w was generated by p_1 , and similarly, $g_2(y)$ is the pdf of y under the assumption that w was generated by p_2 . So we have

$$\hat{w}(y) = \frac{ag_1(y)}{ag_1(y) + bg_2(y)} \text{soft}\left(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_1}\right) + \frac{bg_2(y)}{ag_1(y) + bg_2(y)} \text{soft}\left(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_2}\right) \quad (19)$$

Because y is the sum of w and independent Gaussian noise, the pdf of y is the convolution of the pdf of w and the Gaussian pdf,

$$\begin{aligned} p_y(y) &= (ap_1(y) + bp_2(y)) * p_n(y) \\ &= ap_1(y) * p_n(y) + bp_2(y) * p_n(y) \\ &= ag_1(y) + bg_2(y) \end{aligned}$$

If w follows the Laplacian mixture model, according to Eq. (14) we have

$$g_1(y) = \text{Laplace}(y, \sigma_1) * \text{Gaussian}(y, \sigma_n) = \frac{1}{\sqrt{2}\sigma_1} \exp\left(-\frac{\sqrt{2}}{\sigma_1}|y|\right) * \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{y^2}{2\sigma_n^2}\right) \quad (20)$$

and

$$g_2(y) = \text{Laplace}(y, \sigma_2) * \text{Gaussian}(y, \sigma_n) = \frac{1}{\sqrt{2}\sigma_2} \exp\left(-\frac{\sqrt{2}}{\sigma_2}|y|\right) * \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{y^2}{2\sigma_n^2}\right) \quad (21)$$

$g_1(y)$ and $g_2(y)$ are not one of the standard pdfs that are commonly known. Figure 10 shows the pdf of the sum of a Laplace and a Gaussian random variable. A formula for pdf of y that is the sum of a Laplace random variable with standard deviation σ_i and a zero-mean Gaussian random variable with variance σ_n is given by [20]

$$g_i(y) = \frac{1}{2\sqrt{2}\sigma_i} \exp\left(-\frac{y^2}{2\sigma_n^2}\right) \left[\text{erfcx}\left(\frac{\sigma_n}{\sigma_i} - \frac{y}{\sqrt{2}\sigma_n}\right) + \text{erfcx}\left(\frac{\sigma_n}{\sigma_i} + \frac{y}{\sqrt{2}\sigma_n}\right) \right], \quad i=1,2 \quad (22)$$

where

$$\begin{aligned} \text{erfcx}(x) &= \exp(x^2) \text{erfc}(x^2) \\ \text{erfc}(x) &= 1 - \text{erf}(x) \\ \text{erf}(x) &= \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \end{aligned}$$

If we use the notation $\text{LapGauss}(y, \sigma_x, \sigma_n)$ for the pdf in (22), then we will have

$$\begin{aligned} g_1(y) &:= \text{LapGauss}(y, \sigma_1, \sigma_n) \\ g_2(y) &:= \text{LapGauss}(y, \sigma_2, \sigma_n) \end{aligned}$$

and so y will be a mixture of two LapGauss pdf with the following pdf:

$$p_y(y) = ag_1(y) + bg_2(y) = a.\text{LapGauss}(y, \sigma_1, \sigma_n) + b.\text{LapGauss}(y, \sigma_2, \sigma_n) \quad (23)$$

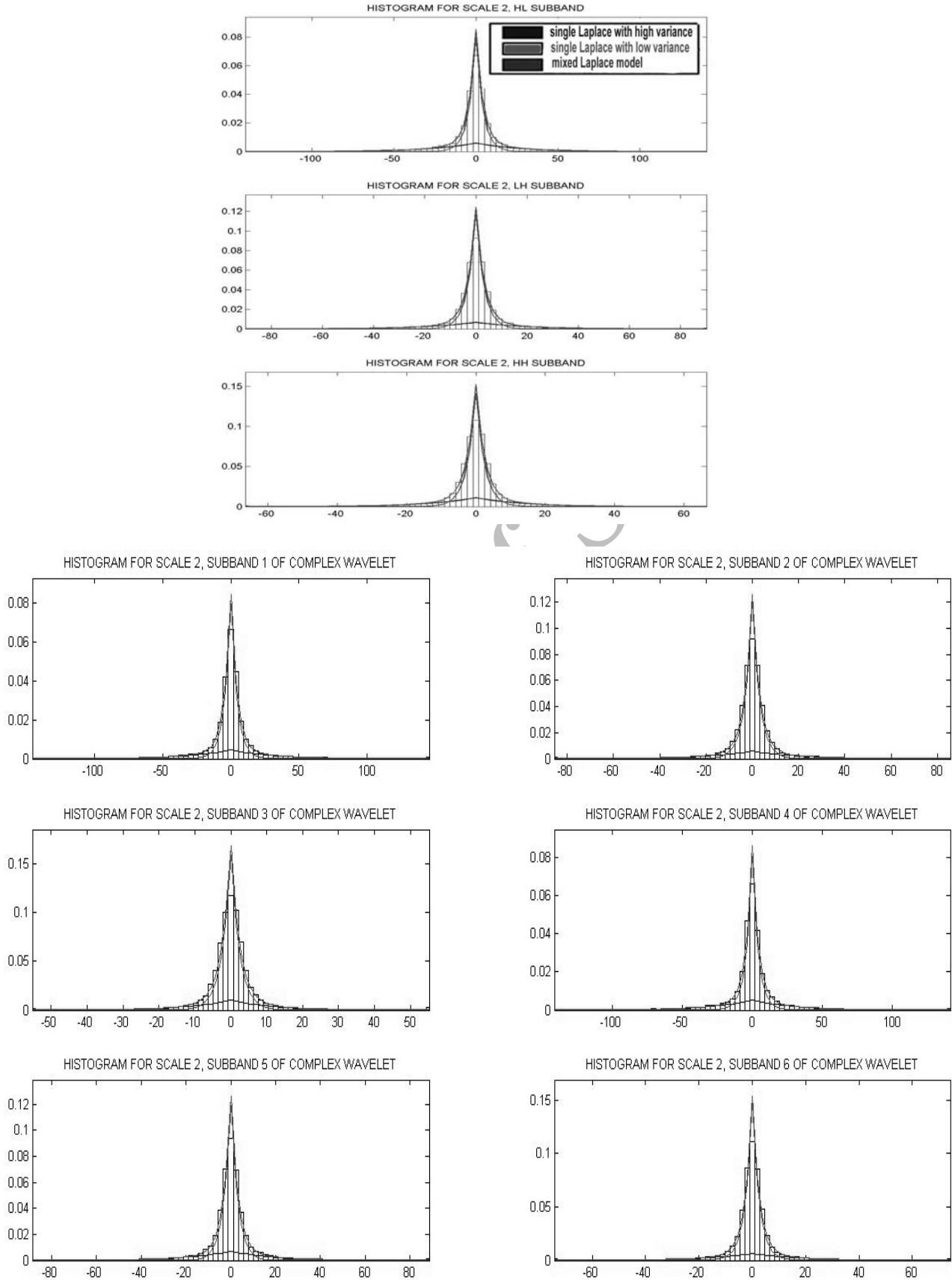


Fig. 8. Histograms of the wavelet coefficients and the best fitted Laplacian mixture model in second scale of Lena image

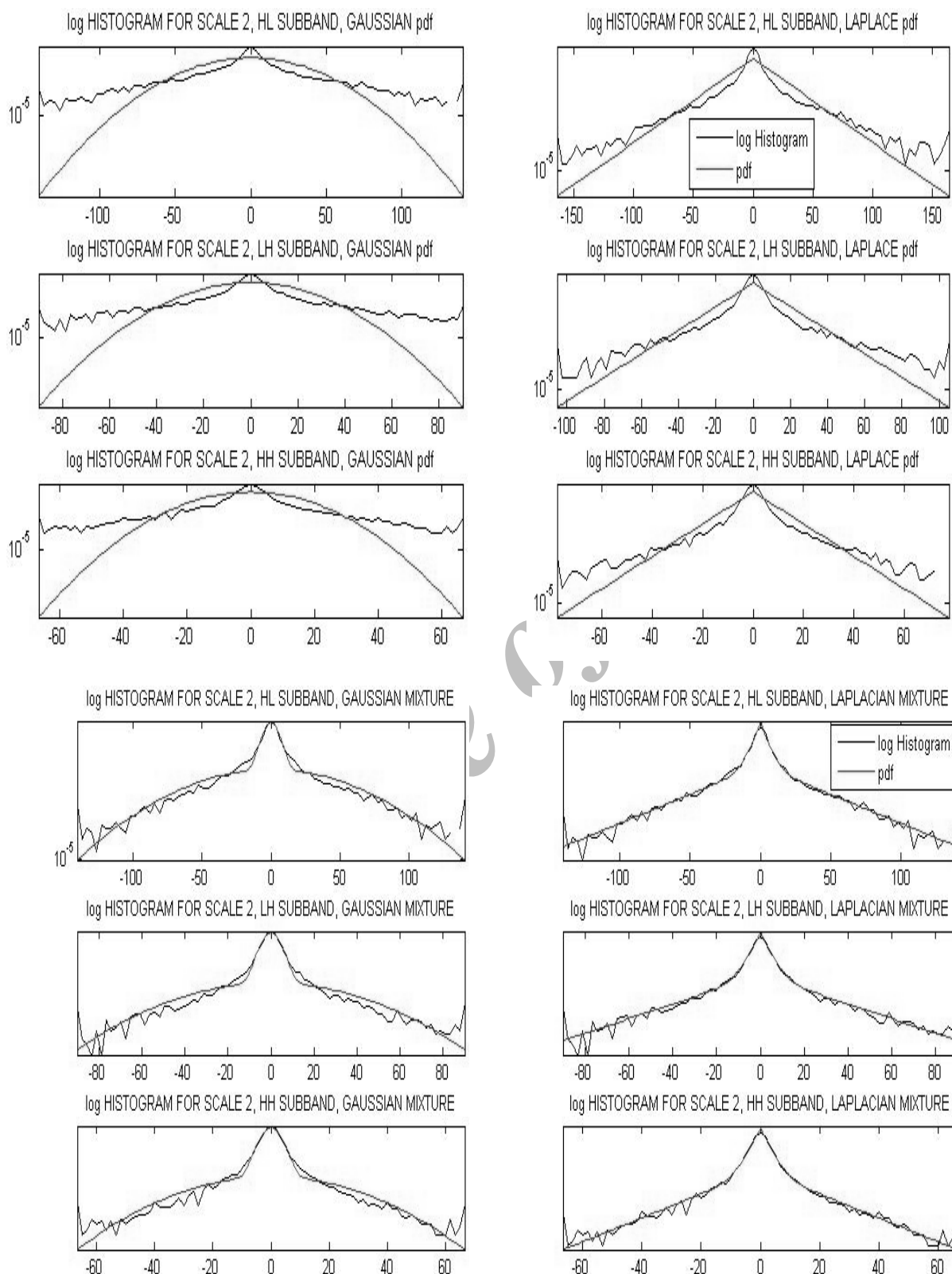


Fig. 9. Histograms of the wavelet coefficients and the best fitted pdf in second scale of Lena image in the log domain. From top left, clockwise: Gaussian pdf, Laplace pdf, a mixture of Laplace pdfs and a mixture of Gaussian pdfs

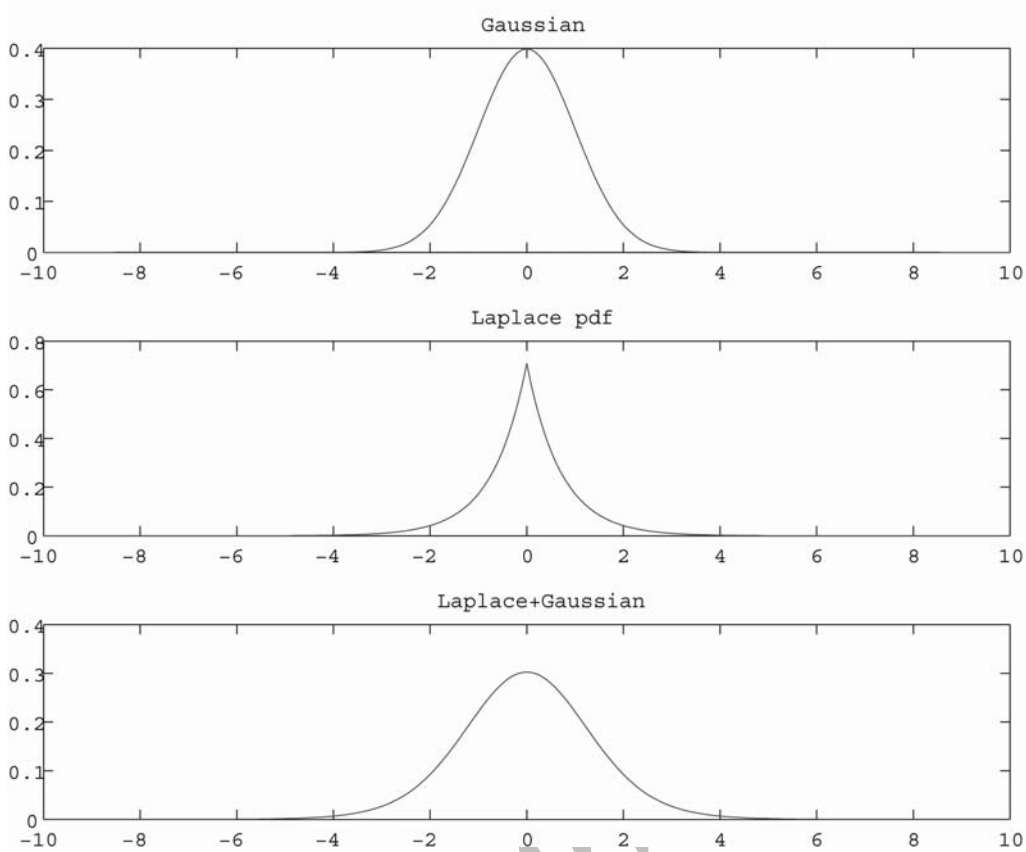


Fig. 10. pdf of the sum of a Laplace and a Gaussian random variable

Figure 11 shows the histogram of the noise-free 512×512 Lena image and the mixture of two Laplace pdfs fitted to it, as well as the histogram of the noisy image corrupted with additive Gaussian noise with $\sigma_n = 10$ and the mixture of two *LapGauss* pdfs fitted to it.

To find the shrinkage function, the *LapGauss* pdf is not needed directly, but only as it appears in the following expression

$$p_a(y) = \frac{ag_1(y)}{ag_1(y) + bg_2(y)} = \frac{aLapGauss(y, \sigma_1, \sigma_n)}{aLapGauss(y, \sigma_1, \sigma_n) + bLapGauss(y, \sigma_2, \sigma_n)}$$

After canceling some common terms and rearranging we get

$$p_a(y) = \frac{1}{1 + R}$$

where

$$R = \frac{\frac{b}{\sigma_2} [erfcx(\frac{\sigma_n}{\sigma_2} - \frac{y}{\sqrt{2}\sigma_n}) + erfcx(\frac{\sigma_n}{\sigma_2} + \frac{y}{\sqrt{2}\sigma_n})]}{\frac{a}{\sigma_1} [erfcx(\frac{\sigma_n}{\sigma_1} - \frac{y}{\sqrt{2}\sigma_n}) + erfcx(\frac{\sigma_n}{\sigma_1} + \frac{y}{\sqrt{2}\sigma_n})]}$$

As $p_a(y) + p_b(y) = 1$ we get

$$p_b(y) = \frac{R}{1 + R}$$

so

$$\hat{w}(y) = \frac{1}{1+R} \text{soft}\left(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_1}\right) + \frac{R}{1+R} \text{soft}\left(y, \frac{\sqrt{2}\sigma_n^2}{\sigma_2}\right) \quad (24)$$

where R is given by the formula above.

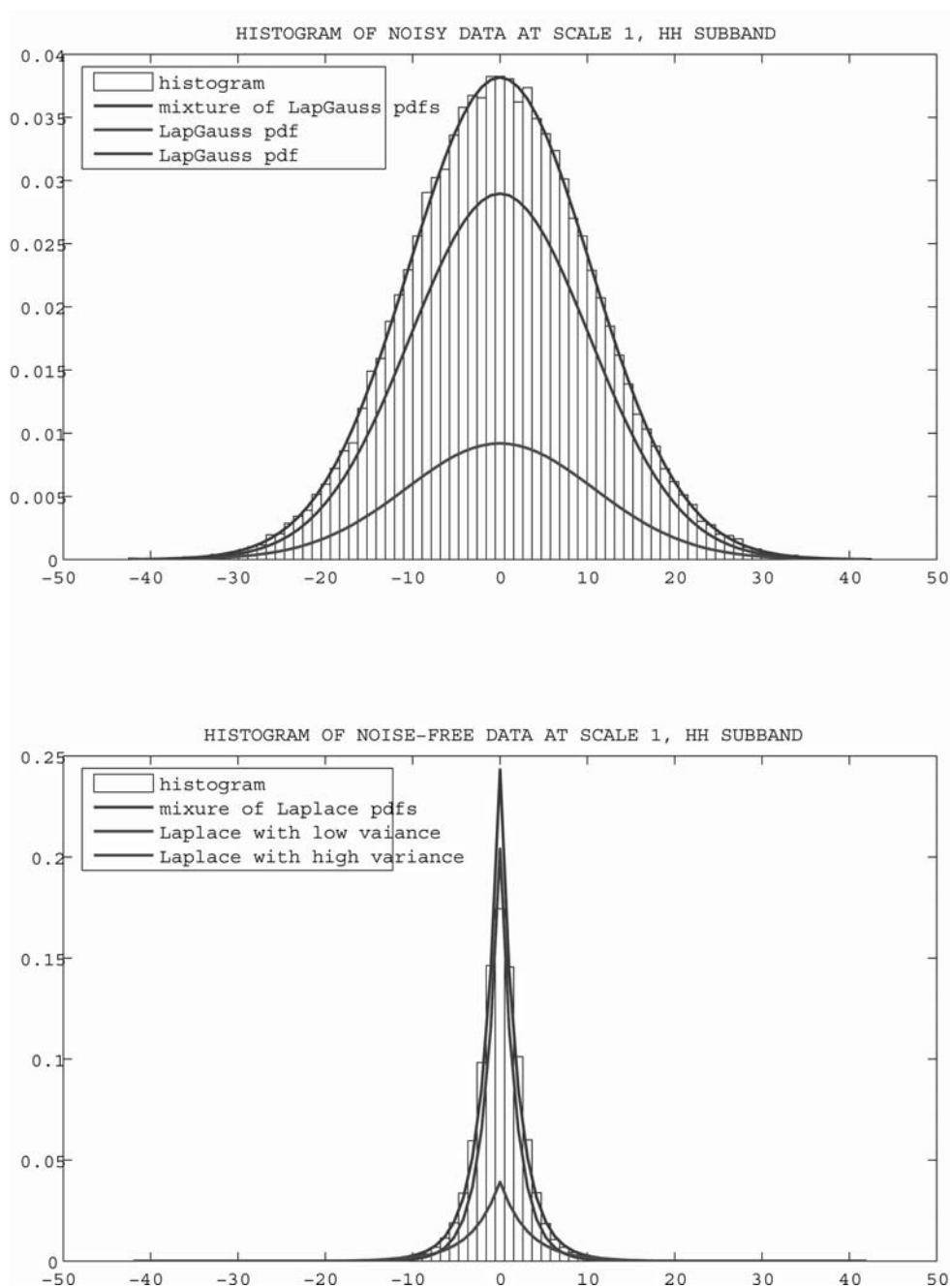


Fig. 11. Histogram of the noise-free and noisy Lena image and the best fitted mixture model

Like the soft threshold function, this nonlinear shrinkage function that we call LapMixShrink reduces (or shrinks) the value of y to estimate w . For several different values of the model parameters, some of the shrinkage functions are given in Fig. 12. The nonlinear function does not shrink large values of y as much as the soft threshold function does.

Note that in order to use this type of shrinkage function, it is necessary to estimate the parameters of the mixture model (a, σ_1 and σ_2) from the noisy data. In practice, we have only noisy data $y = w + n$ as a

mixture of two *LapGauss* components, and assuming the noise variance σ_n is known, we can estimate the model parameters a, σ_1 and σ_2 using the EM algorithm. The threshold functions for each subband of the 512×512 Lena image corrupted with additive Gaussian noise with $\sigma_n = 10$ after estimating parameters is illustrated in Fig. 13.

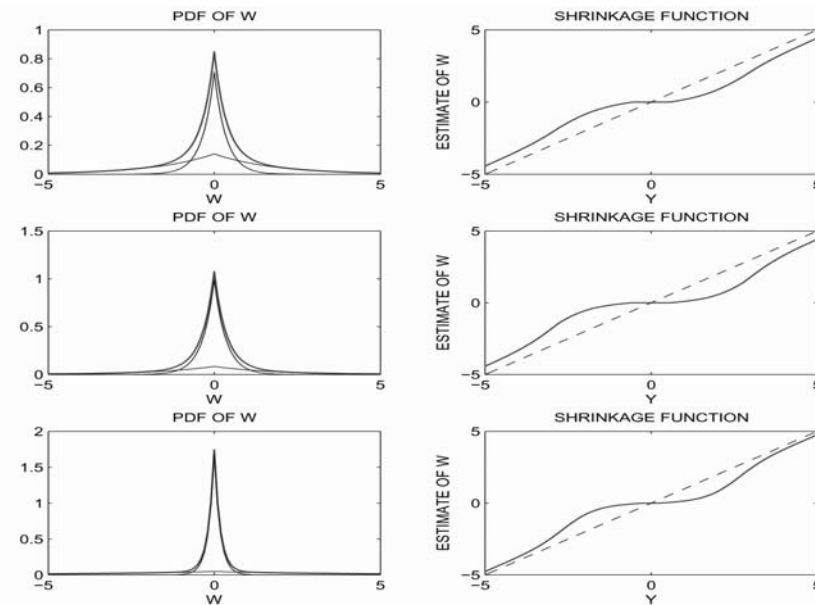


Fig. 12. *LapMixShrink* function produced from a mixture of two Laplace pdf

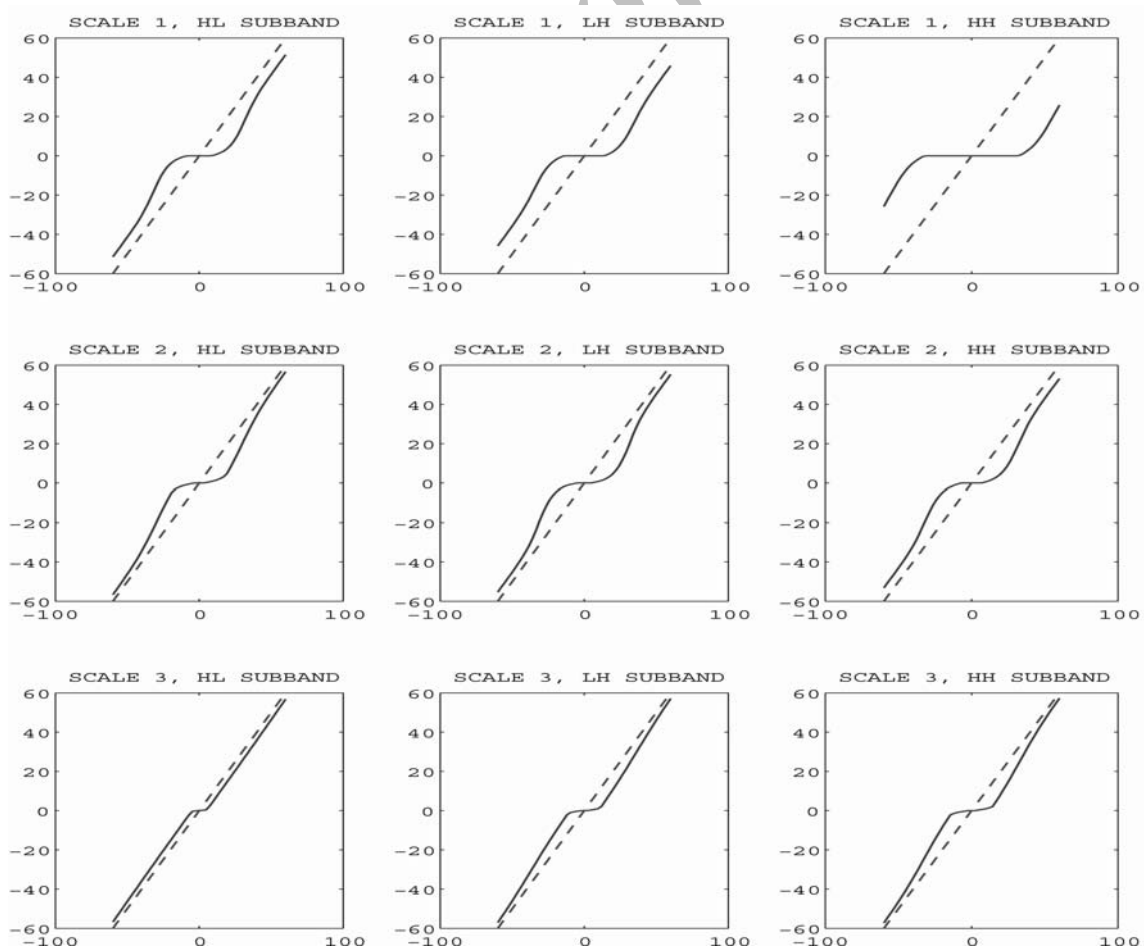


Fig. 13. The threshold functions *LapMixShrink* for Lena image in each subband

Instead of LapMixShrik, other nonlinear shrinkage functions can be found with other mixture models. For example, if w follows the Gaussian mixture model (15), p_1 and p_2 are Gaussian pdfs with parameters σ_1 and σ_2 respectively. Therefore, Eq. (6) can be used to get $\hat{w}_1(y)$ and $\hat{w}_2(y)$ in Eq. (16),

$$\hat{w}_1(y) = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_n^2} y, \quad \hat{w}_2(y) = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_n^2} y$$

In this case, $g_1(y)$ and $g_2(y)$ in Eqs. (17) and (18) can be written as

$$\begin{aligned} g_1(y) &= \text{Gaussian}(y, \sigma_1) * \text{Gaussian}(y, \sigma_n) \\ &= \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) \\ g_2(y) &= \text{Gaussian}(y, \sigma_2) * \text{Gaussian}(y, \sigma_n) \\ &= \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2}) \end{aligned}$$

Thus, Eqs. (17) and (18) can be written as

$$\begin{aligned} p_a(y) &= \frac{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2})}{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) + b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2})} \\ p_b(y) &= \frac{b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2})}{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) + b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2})} \end{aligned}$$

Therefore, we can write Eq. (16) as

$$\begin{aligned} \hat{w}(y) &= \frac{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) \frac{\sigma_1^2}{\sigma_1^2 + \sigma_n^2} y}{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) + b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2})} \\ &+ \frac{b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2}) \frac{\sigma_2^2}{\sigma_2^2 + \sigma_n^2} y}{a \text{Gaussian}(y, \sqrt{\sigma_1^2 + \sigma_n^2}) + b \text{Gaussian}(y, \sqrt{\sigma_2^2 + \sigma_n^2})} \end{aligned} \quad (25)$$

Figure 14 shows this shrinkage function, that we have named the GaussMixShrink nonlinearity, for the same model parameters according to Fig. 13.

3. EXPERIMENTAL RESULTS

In the previous section, we have proposed a new statistical model for wavelet coefficients and obtained a MAP estimator for this model. This section presents image denoising examples to show the efficiency of this new model and compare it with other methods in the literature. We test our algorithm in a standard and complex wavelet transform domain. Complex wavelet transform is an over-complete wavelet transform featuring near shift invariance and has improved directional selectivity compared to the standard wavelet transform [16, 21].

Figure 15 shows part of the original image, noisy image and denoised image obtained using our shrinkage functions illustrated in Fig. 13. Denoised images with GaussMixShrink and LapMixShrink in a standard and complex wavelet domain are illustrated in Fig. 16. We also have compared our shrinkage function (24) with the classical soft thresholding estimator given in (10) for image denoising. The 512×512 Lena image is used for this purpose. Zero mean white Gaussian is added to the original image.

Part of the denoised image using the soft threshold and the denoised image using the LapMixShrink function are illustrated in Fig. 17 and the LH subband of the denoised images in third scale are shown in Fig. 18. The denoised image using the soft threshold has a RMSE of 4.97, while the denoised image obtained using our shrinkage function has a RMSE of 4.83. A comparison between the RMSE in each subband after denoising with soft thresholding and the LapMixShrink method is illustrated in Fig. 19.

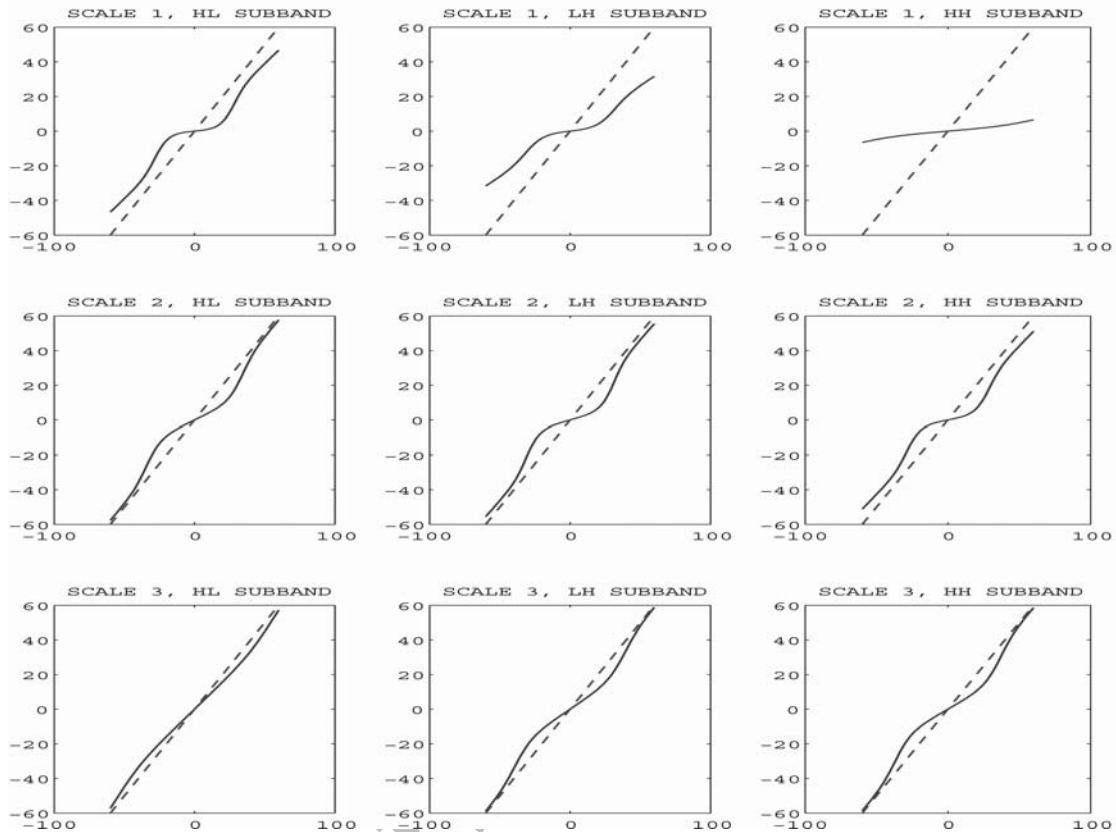


Fig. 14. The threshold functions *GaussMixShrink* for Lena image in each subband



Fig. 15. Denoising with *LapMixShrink* for Lena image corrupted with additive Gaussian noise with $\sigma_n = 10$



Fig. 16. Denoising with *GaussMixShrink* and *LapMixShrink* for Lena image corrupted with additive Gaussian noise with $\sigma_n = 20$

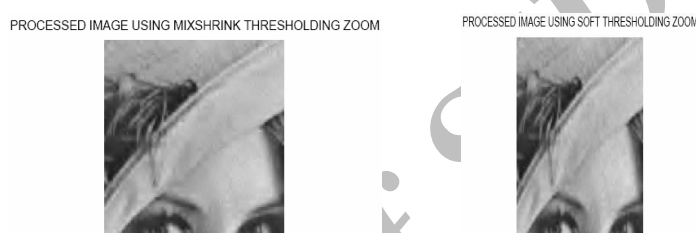


Fig. 17. Denoising with soft thresholding and *LapMixShrink* for Lena image corrupted additive Gaussian noise with $\sigma_n = 10$

We also tested our algorithm for a mixture of two and three Laplace pdfs in a standard and complex wavelet domain using different additive Gaussian noise levels $\sigma_n = 10, 20, 30$ to three 512×512 grayscale images, namely, Lena, Barbara and Boat, and compared them with VisuShrink, SureShrink, BayesShrink and HMT. Performance analysis is done using the PSNR measure. The results can be seen in Table 1. Each PSNR value in the table is averaged over ten runs. In this table, the highest PSNR value is bolded and the best PSNR in the standard wavelet domain is underlined. As seen from the results, our algorithm mostly outperforms the others.

Table 1. Average PSNR values of denoised images over ten runs for different test images and noise levels of noisy, Visushrink, Sureshrink, Bayesshrink, HMT system and our Model 1 (two Laplace components in wavelet domain), our Model 2 (three Laplace components in wavelet domain), our Model 3 (three Laplace components in complex wavelet domain)

	Noisy	VisuShrink	SureShrink	BayesShrink	HMT	Our model 1	Our model 2	Our model3
Lena								
$\sigma_n = 10$	28.18	28.76	33.28	33.32	<u>33.84</u>	33.60	33.63	34.83
$\sigma_n = 20$	22.14	26.46	30.22	30.17	30.39	30.41	<u>30.42</u>	31.72
$\sigma_n = 30$	18.62	25.14	28.38	28.48	28.35	28.67	<u>28.75</u>	29.91
Boat								
$\sigma_n = 10$	28.16	26.49	31.19	31.80	<u>32.28</u>	31.94	31.99	33.00
$\sigma_n = 20$	22.15	24.43	28.14	28.48	28.54	28.59	<u>28.63</u>	29.58
$\sigma_n = 30$	18.62	23.33	26.52	26.60	26.83	26.74	<u>26.84</u>	27.64
Barbara								
$\sigma_n = 10$	28.16	24.81	30.21	30.86	31.36	31.40	<u>31.43</u>	33.09
$\sigma_n = 20$	22.14	22.81	25.91	27.13	<u>27.80</u>	27.25	27.30	28.88
$\sigma_n = 30$	18.62	22.00	24.33	25.16	25.11	25.14	<u>25.18</u>	26.54

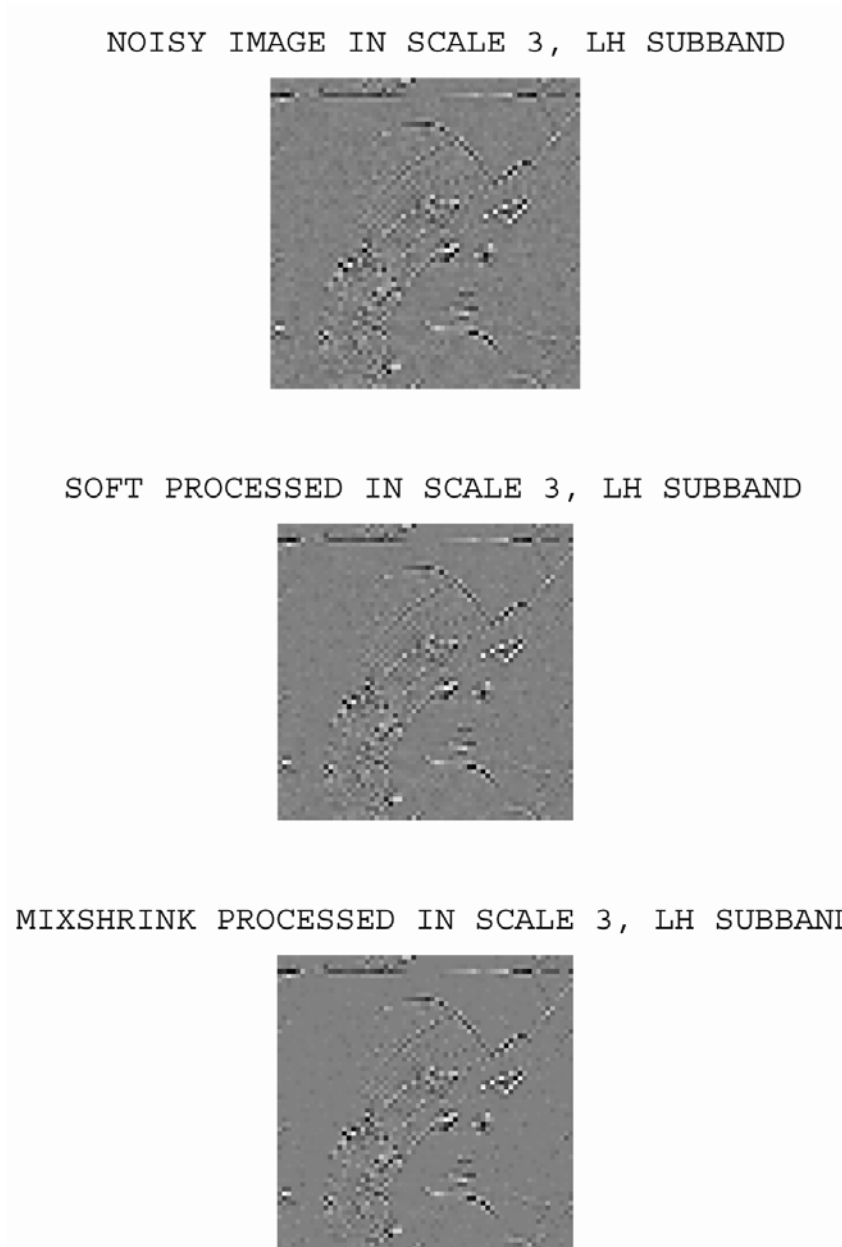


Fig. 18. Comparison between denoising with soft thresholding and *LapMixShrink* in scale 3, LH subband

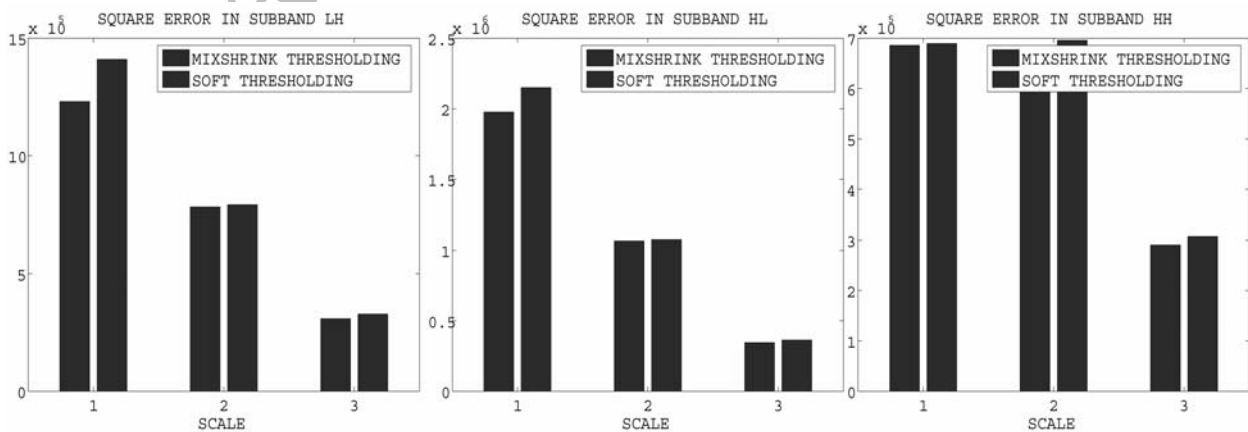


Fig. 19. RMSE in each subband after denoising with soft thresholding and *LapMixShrink*

4. CONCLUSION AND FUTURE WORKS

In this paper we use a LapMixShrink function based on a mixture of Laplace pdfs for modeling wavelet coefficients in each subband. Experiments show that our model has better visual results than other methods such as soft thresholding. In order to show the effectiveness of the new estimator, we compared the LapMixShrink method with effective techniques in the literature and we see that our denoising algorithm mostly outperforms the others. The performance of this subband-adaptive data-driven system is also demonstrated on the complex wavelet domain.

Instead of this shrinkage function, other nonlinear shrinkage functions can be used. For example, instead of using a Laplace pdf we can use generalized Gaussian distribution or, instead of using the MAP estimator for a mixture of Laplace random variables in Gaussian noise, we can use the *minimum mean squared error* (MMSE) estimator. These new pdfs and estimators may lead to better results. Also, instead of processing each wavelet coefficient individually, better denoising results can be achieved by processing groups of wavelet coefficients together [11, 15-17]. Thus, if we can use a model for wavelet coefficients that not only is a mixture but is also bivariate, such as, bivariate Gaussian mixture, bivariate Laplacian mixture, Cauchy mixture or circular symmetric Laplacian mixture, the performance of the denoising algorithm will be improved. Because the state-of-the-art algorithms [17, 19] generally use local adaptive methods, using these methods in combination with the LapMixShrink function, such as LapMixShrink with local parameters, may further improve the denoising results.

Acknowledgements: The authors wish to acknowledge with grateful thanks the contributions of Professor Ivan Selesnick.

REFERENCES

1. Lukac, R., Smolka, B., Martin, K., Plataniotis, K. N. & Venetsanopoulos, A. N. (2005). Vector Filtering for Color Imaging. *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, 22(1), 74-86.
2. Donoho, D. L. (1995). Denoising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41, 613-627.
3. Donoho, D. L. & Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3), 425-455.
4. Donoho, D. L. & Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.*, 90(432), 1200-1224.
5. Chang, S., Yu, B. & Vetterli, M. (2000). Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9, 1532-1546.
6. Coifman, R. & Donoho, D. (1995). *Time-invariant wavelet denoising*. in Wavelet and Statistics, A. Antoniadis and G. Oppenheim, Eds. New York: Springer-Verlag, 103, Lecture Notes in Statistics, 125-150.
7. Figueiredo, M. A. T. & Nowak, R. D. (2001). Wavelet-based image estimation: An empirical Bayes approach using Jeffrey's noninformative prior. *IEEE Trans. on Image Processing*, 10, 1322-1331.
8. Gao, H. (1998). Wavelet shrinkage denoising using the non-negative garrot. *J. Comput. Graph. Stat.*, 7, 469-488.
9. Hyvarinen, A. (1999). Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Comput.*, 11, 1739-1768.
10. Hyvarinen, A., Oja, E. & Hoyer, P. (2001). *Image denoising by sparse code shrinkage*. S. Haykin and B. Kosko, Eds. Piscataway, NJ: IEEE.

11. Simoncelli, E. P. (1999). Bayesian denoising of visual images in the wavelet domain. Bayesian Inference in Wavelet Based Models, P. Müller and B. Vidakovic Eds. New York: Springer-Verlag, 141, Lecture Notes in Statistics.
12. Vidakovic, B. (1999). *Statistical Modeling by Wavelets*, New York: Wiley.
13. Jansen, M. (2001). Noise Reduction by Wavelet Thresholding. New York: Springer-Verlag, 161, Lecture Notes in Statistics.
14. Abramovich, F., Sapatinas, T. & Silverman, B. (1998). Wavelet thresholding via a Bayesian approach. *J. R. Stat.*, 60, 725–749.
15. Sendur, L. & Selesnick, I. W. (2002). Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans. Signal Processing*, 50(11), 2744–2756.
16. Achim, A. & Kuruoglu, E. E. (2005). Image denoising using bivariate alpha-stable distributions in the complex wavelet domain. *IEEE Signal Processing Lett.*, 12(1), 17-20.
17. Strela, V., Portilla, J. & Simoncelli, E. (2000). Image denoising using a local Gaussian scale mixture model in the wavelet domain. in *Proc. SPIE 45th Annu. Meet, San Jose, CA (United States)*, 363-371.
18. Crouse, M. S., Nowak, R. D. & Baraniuk, R. G. (1998). Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Signal Processing*, 46(4), 886-902.
19. Mihcak, M. K., Kozintsev, I., Ramchandran, K. & Moulin, P. (1999). Low complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Lett.*, 6, 300–303.
20. Hansen, M. & Yu, B. (2000). Wavelet Thresholding via MDL for Natural Images. *IEEE Trans. Inform. Theory*, 46(5).
21. Abdelnour, A. F. & Selesnick, I. W. (2001). Nearly symmetric orthogonal wavelet bases. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP), Salt Lake City (United States)*, 3693-3696.

APPENDIX

EM algorithm:

The Expectation-Maximization algorithm is an iterative numerical algorithm that can be used to estimate the parameters of a mixture model. Each iteration consists of an E-step and an M-step. Here we give only a simple description of the EM algorithm. The mixture model is

$$p(x) = ap_1(x) + bp_2(x)$$

where $a + b = 1$. The data is x_n for $n = 1, 2, \dots, N$. From the data we want to estimate the 3 parameters a, σ_1 and σ_2 .

The EM algorithm works by introducing an auxiliary variable that represents, for each data point, how likely that the data point was produced by one or the other of the two components $p_1(x)$ and $p_2(x)$. This auxiliary variable is denoted by $r_1(n)$ and $r_2(n)$. $r_1(n)$ represents how responsible $p_1(x)$ is for generating data point x_n , while $r_2(n)$ represents how responsible $p_2(x)$ is for generating data point x_n .

The EM algorithm starts by initializing a, b, σ_1 and σ_2 , and then proceeds with a sequence of E-M steps until the parameters satisfy some convergence condition. The initial values for a and b should satisfy $a + b = 1$.

The E-step calculates the responsibility factors

$$r_1(n) \leftarrow \frac{ap_1(x_n)}{ap_1(x_n) + bp_2(x_n)}$$

$$r_2(n) \leftarrow \frac{bp_2(x_n)}{ap_1(x_n) + bp_2(x_n)}$$

Note that the responsibility factors are between 0 and 1 and that $r_1(n) + r_2(n) = 1$.

The M-step updates the parameters a, b, σ_1 and σ_2 . The mixture parameters a and b are computed by

$$a \leftarrow \frac{1}{N} \sum_{n=1}^N r_1(n), \quad b \leftarrow \frac{1}{N} \sum_{n=1}^N r_2(n)$$

It is easy to verify that $a + b = 1$ is guaranteed. One way to update σ_1 and σ_2 is to modify the basic formula for the sample variance. Instead of estimating the variance as the mean of the squares of the data values using the usual formula, $\hat{\sigma}_1^2 \leftarrow (1/N) \sum_{n=1}^N x_n^2$, we can estimate σ_1^2 as a weighted sum of the data values, where the weight for x_n is the responsibility of $p_1(x)$ for the data point x_n . That gives the following formulas for the Laplacian case.

$$\sigma_1 \leftarrow \sqrt{2} \frac{\sum_{n=1}^N r_1(n) |x_n|}{\sum_{n=1}^N r_1(n)}, \quad \sigma_2 \leftarrow \sqrt{2} \frac{\sum_{n=1}^N r_2(n) |x_n|}{\sum_{n=1}^N r_2(n)}$$

For many mixture models such as a mixture of *LapGauss* pdfs, a closed form for computing σ_1 and σ_2 does not exist. In these cases, the following formula produced from a mixture of Gaussian pdfs can be used to approximate σ_1 and σ_2 .

$$\sigma_1^2 \leftarrow \frac{\sum_{n=1}^N r_1(n) x_n^2}{\sum_{n=1}^N r_1(n)}, \quad \sigma_2^2 \leftarrow \frac{\sum_{n=1}^N r_2(n) x_n^2}{\sum_{n=1}^N r_2(n)}$$

Archive of SID