# EE3731C Signal Processing Methods: Project

Nguyen Anh Tuan, Matric No. A0074465Y

October 30, 2013

## 1 Image Denoising

### 1.1 Haar Wavelet Transform

In this section, we will demonstrate the simplest form of wavelet transform, namely, Haar wavelet. For each layer of transformation, a square image $N \times N$ is decomposed into 4 blocks LL, LH, HL and HH, which contain the low-frequency (differential) and high-frequency (average) components along the horizontal and vertical directions respectively. Fig.1a demonstrate the transformation where each components are computed as:

$$\begin{bmatrix} ll \\ hl \\ lh \\ hh \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Subsequently, the inverse transform could be realized as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} ll \\ hl \\ lh \\ hh \end{bmatrix}$$

The low-frequency component LL is further decomposed in subsequent levels to obtain $\{LL_2, LH_2, HL_2, HH_2\}$, ..., $\{LL_J, LH_J, HL_J, HH_J\}$, where $J$ is the number of transform levels. Wavelet coefficients are stored in the standard pattern as shown in Fig.1b.

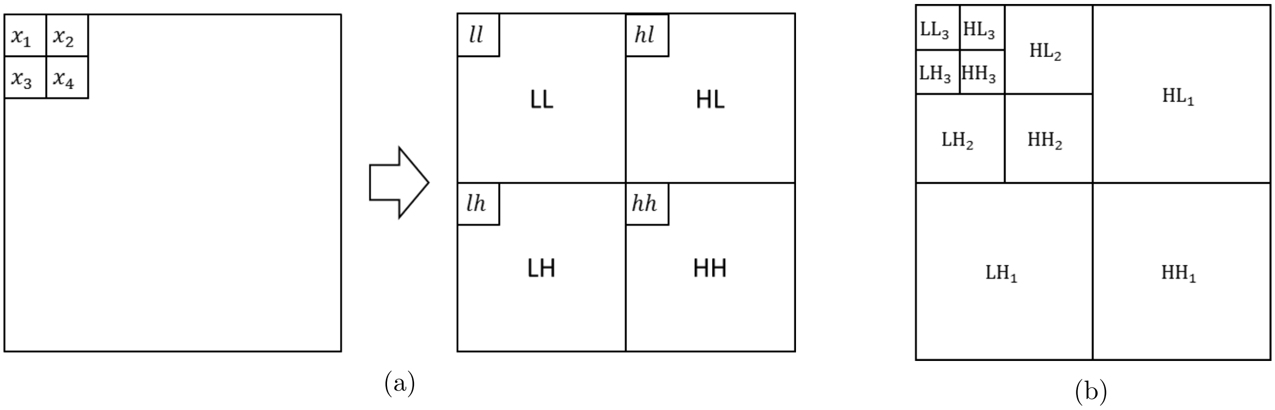
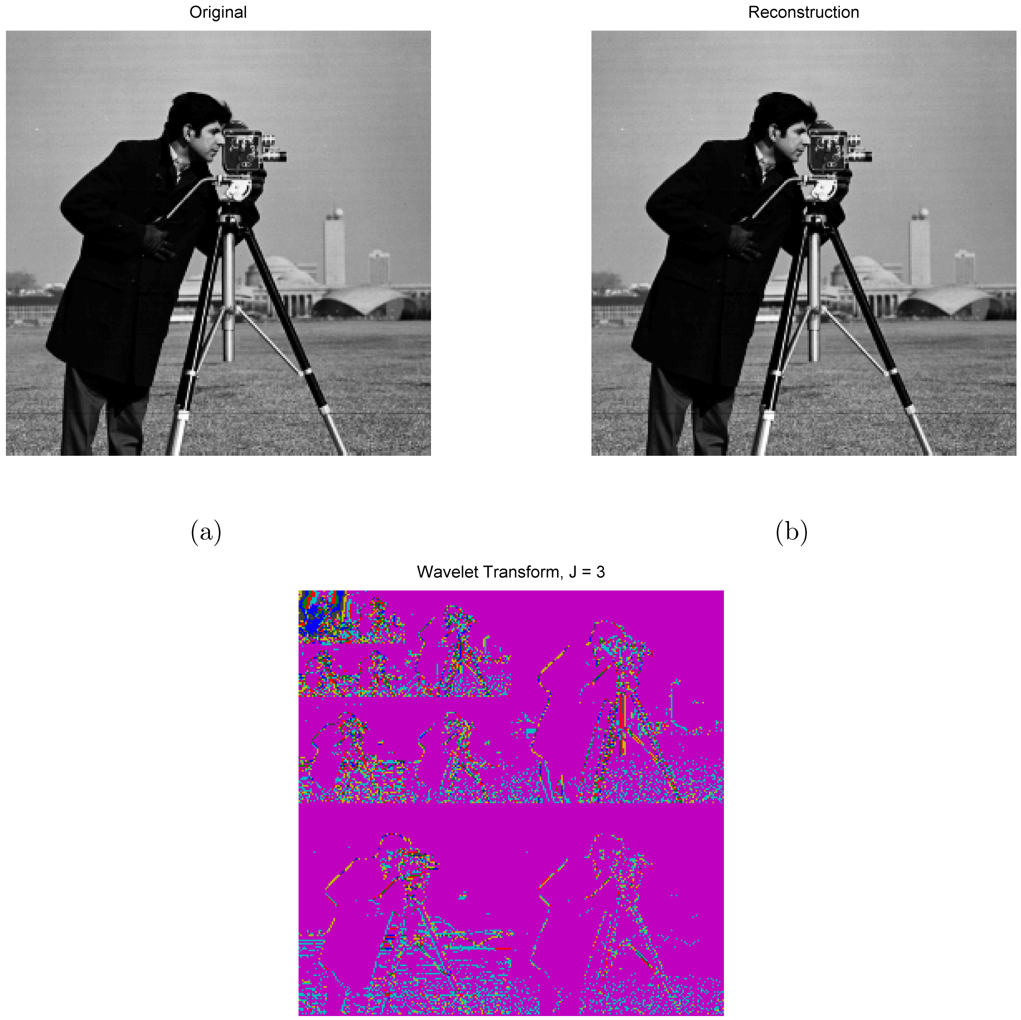


Figure 1: (a) Illustration of 2D Haar wavelet transform. (b) Standard 2D wavelet decomposition.

Haar wavelet transform and inverse transform is applied on image *camera* with 3-level. The results are shown in Fig.2. As all high-frequency components are used for reconstruction, no differences between the original and the reconstructed images could be observed.



(a)

(b)

(c)

Figure 2: Applying Haar wavelet transform on image *camera* with 3-level. (a) Original, and (b) reconstructed image. (c) Haar wavelet coefficient (lines colormap).

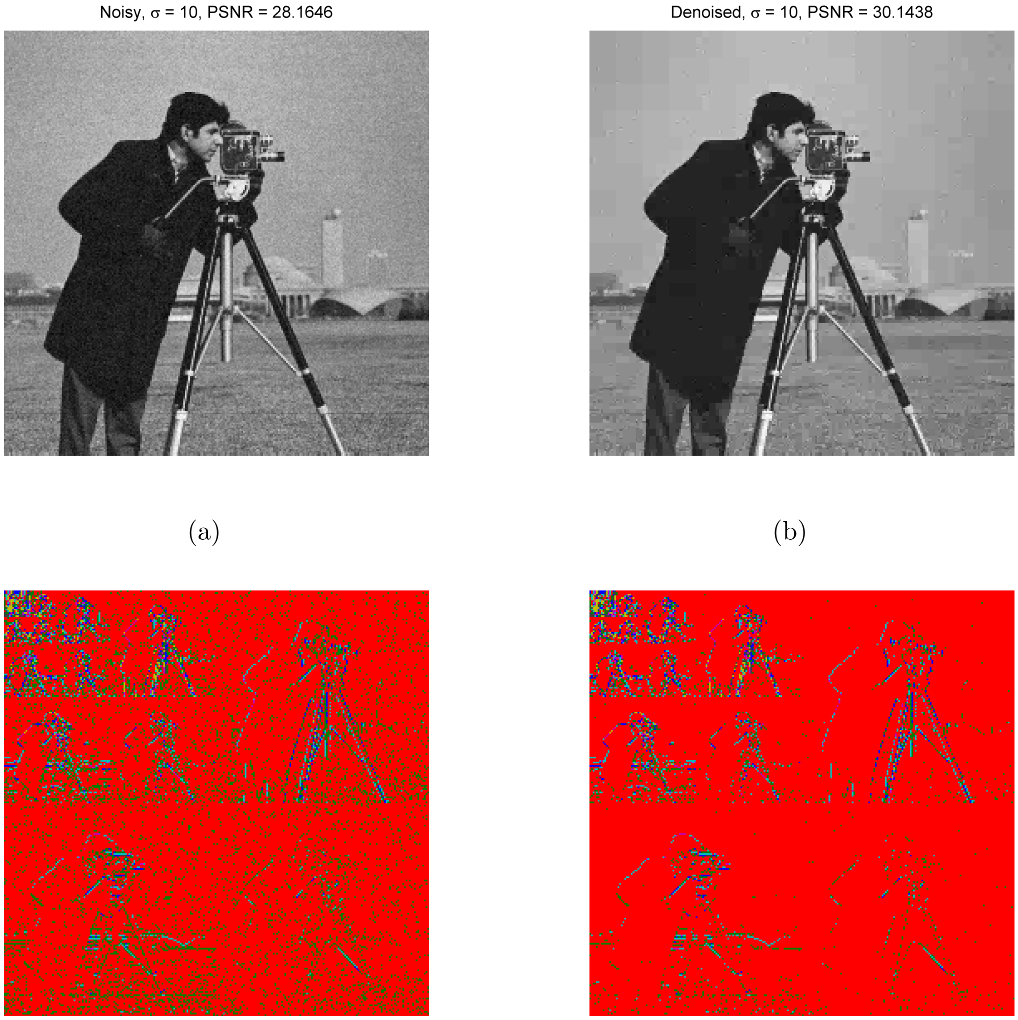
### 1.2 Image Denoising

A small amount of Gaussian white noise with variance $\sigma^2$ is added to the image. The amount of distortion is measure in term of Peak-Signal-to-Noise-Ratio (PSNR). The PSNR between 2 original image I and noisy image K is given as:

$$PSNR = 20\log_{10}(MAX) - 10\log_{10}\left(\frac{1}{N^2}\Sigma_{i=0}^{N}\Sigma_{j=0}^{N}[I(i,j) - K(i,j)]^2\right)$$

Where $MAX = 255$ is the maximum value of each pixel. The noisy *camera* image with $\sigma = 10$ and $PSNR = 28.16dB$ is shown in Fig.3a.

To denoise the image, Haar wavelet transform with 4-level, following hard-thresholding scheme is applied. All detail coefficients smaller than threshold $\lambda = 3\sigma$ is set to zero. As shown in Fig.3c and Fig.3d, thresholding effectively remove most of the white noise presented in the detail components. By setting the threshold to $\lambda = 3\sigma$, we expect 99% of the white noise power is eliminated. However, some small details of the original image are also inevitably removed. As a result, certain amount of artifacts due to denoising could be seen in the reconstructed image, which leads to $PSNR = 30.14dB$


Noisy, σ = 10, PSNR = 28.1646
Denoised, σ = 10, PSNR = 30.1438

(a)

(b)

(c)

(d)

Figure 3: Denoising *camera* image using wavelet transform and hard-thresholding $\lambda = 3\sigma$

Different level of noise from $\sigma = 10$ to $\sigma = 80$ is applied to evaluate the performance of denoising algorithm. PSNR before and after denoising is shown in Fig.4. PSNR before denoising decrease exponentially as the noise power increase because

$$PSNR \sim -\log_{10}\left(\frac{1}{N^2}\Sigma_{i=0}^{N}\Sigma_{j=0}^{N}[I(i,j) - K(i,j)]^2\right) \sim -\log_{10}(\textit{Noise Power})$$

Moreover, PSNR after denoising is also decrease. It is because by design, hard-thresholding with $\lambda = 3\sigma$ totally eliminate the noise power; as the noise power increase, the algorithm also remove more details of the original image. More effective methods of thresholding will be discussed in the next session.

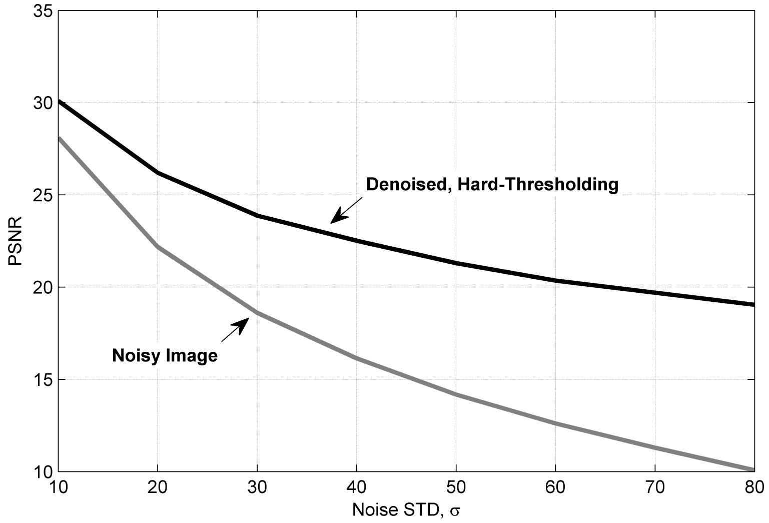


Figure 4: PSNR before and after denoising with hard-thresholding method, $\lambda = 3\sigma$.

### 1.3 New Thresholding Schemes

Soft-thresholding is another scheme suggested by Donoho and Johnstone [1, 2] for image denoising besides hard-thresholding method. The expression of hard-thresholding is given as:

$$f_{hard,\lambda}(x) = \begin{cases} 0 & ,\forall|x| \leq \lambda \\ x & ,\forall|x| > \lambda \end{cases}$$

While soft-thresholding expression is:

$$f_{soft,\lambda}(x) = \begin{cases} 0 & ,\forall|x| \leq \lambda \\ sign(x)(|x| - \lambda) & ,\forall|x| > \lambda \end{cases}$$

Where $\lambda$ is the threshold. 2 thresholding functions with the same $\lambda$ are shown in comparison in Fig.5. While hard-thresholding simply keeps or removes the coefficients, soft- method shrinks all coefficients towards 0 by the amount of $\lambda$. However, it has been shown that soft-thresholding does not necessary gain advantage over hard-thresholding. Soft- scheme usually performs better for "smooth" images. Another issue is choosing the thresholding values. If the threshold is too small, not much noise is removed. On the other hand, bias is introduced if the threshold is too large, causing quality degeneration.

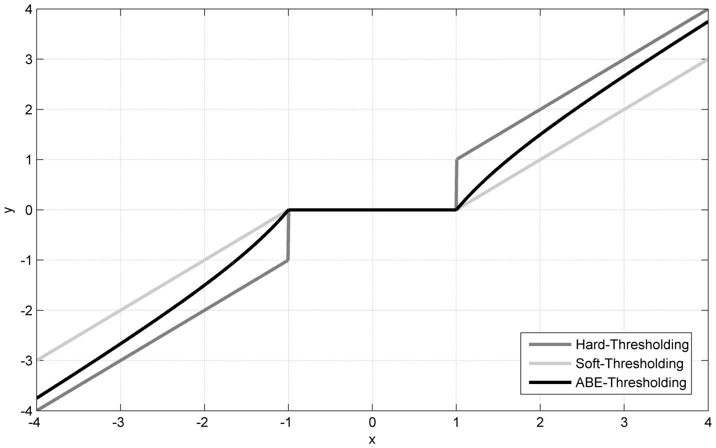


Figure 5: Thresholding function for 3 schemes with the same value of $\lambda$

Fig.6 shows the PSNR using soft- and hard-thresholding scheme with $\sigma = 20$ and different threshold value $\lambda$. With $\lambda = 3\sigma$, hard-threshold outperforms soft- scheme. In fact, $\lambda = 3\sigma$ is also the optimum threshold for hard-scheme. With optimum choice of $\lambda$, it is possible for soft-thresholding to achieve higher improvement than hard-scheme. However, it is often difficult to estimate this optimum value of $\lambda$ for different images.

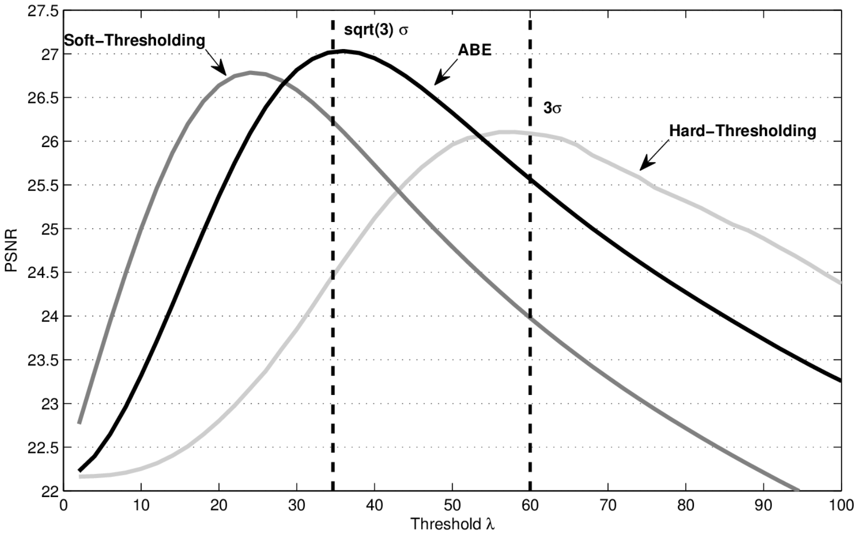
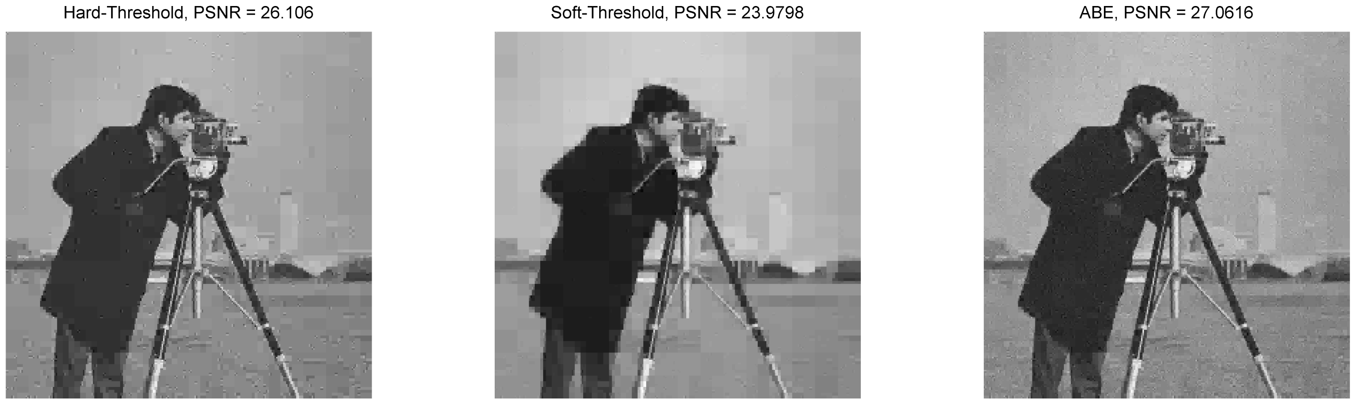


Figure 6: PSNR of denoised image using 3 scheme of thresholding.

Another more robust thresholding scheme has been suggested by Figueiredo and Nowak [3], namely, Amplitude-scale-invariant Bayes Estimation (ABE). The simplified ABE-thresholding function is given as:

$$f_{ABE,\lambda}(x) = \begin{cases} 0 & ,\forall |x| \leq \lambda \\ x - \frac{\lambda^2}{x} & ,\forall |x| > \lambda \end{cases}$$

Where $\lambda = \sqrt{3}\sigma$. ABE function is shown in Fig.5 in comparison with hard- and soft-thresholding with the same $\lambda$. ABE function has nonlinear expression and approachs hard- curve when $x$ becomes larger. It is designed to take advantages of soft- scheme for small coefficients while reduce the bias introduced for large coefficients. Comparing PSNR improvement in Fig.6, 7, 8, it is evident that at designed threshold $\lambda = \sqrt{3}\sigma$, ABE- scheme not only achieve near-optimum performance, but also surpasses the maximum of both hard- and soft- scheme.


Hard-Threshold, PSNR = 26.106

Soft-Threshold, PSNR = 23.9798

ABE, PSNR = 27.0616

Figure 7: Image denoising with $\sigma = 20$ using 3 schemes of thresholding.

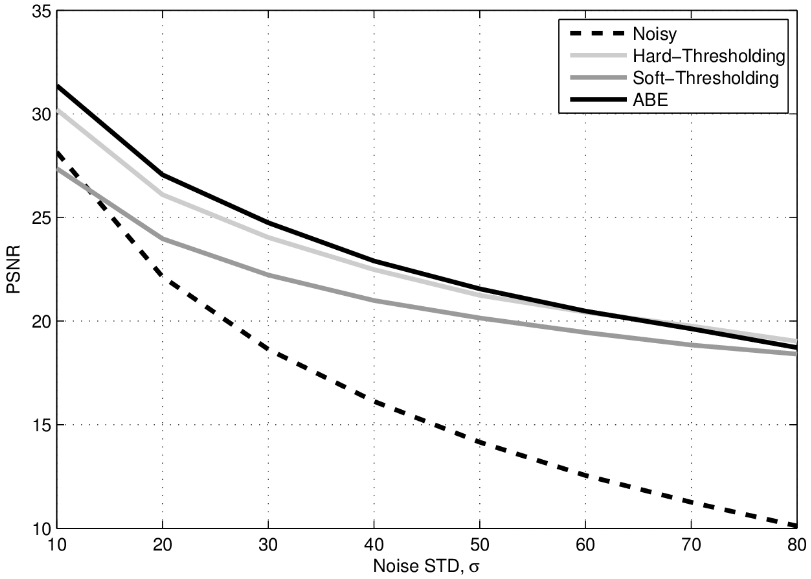


Figure 8: PSNR of reconstructed image with 3 schemes of thresholding.

## 2 Image Compression

This section demonstrate a type of lossy image compression using wavelet transform. Fig.9 presents the original *lena* image, while the compressed images with different compression ratio is shown in Fig.10. The PSNR values decrease with higher compression ratio (Fig.12). Nevertheless, the visual quality degeneration is hardly recognizable for 20% and 10% compression. The 5% compression image has lighter color, but the detail remain relatively sharp. Visible distortions could only be seen in the 1% compression version. Fig.11 presents normalized distribution of wavelet coefficients. Wavelet transform effectively compress the most significant information in a few large value coefficients. The majority of the coefficients has small values, hence carry little information. As a result, they could be filtered without significant loss of image quality.

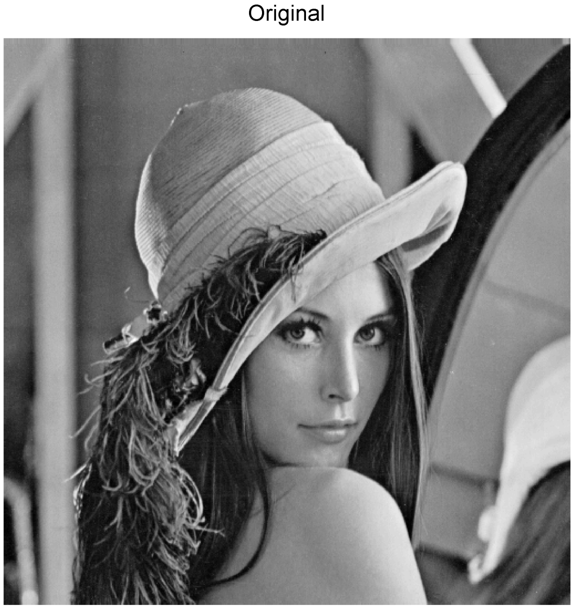
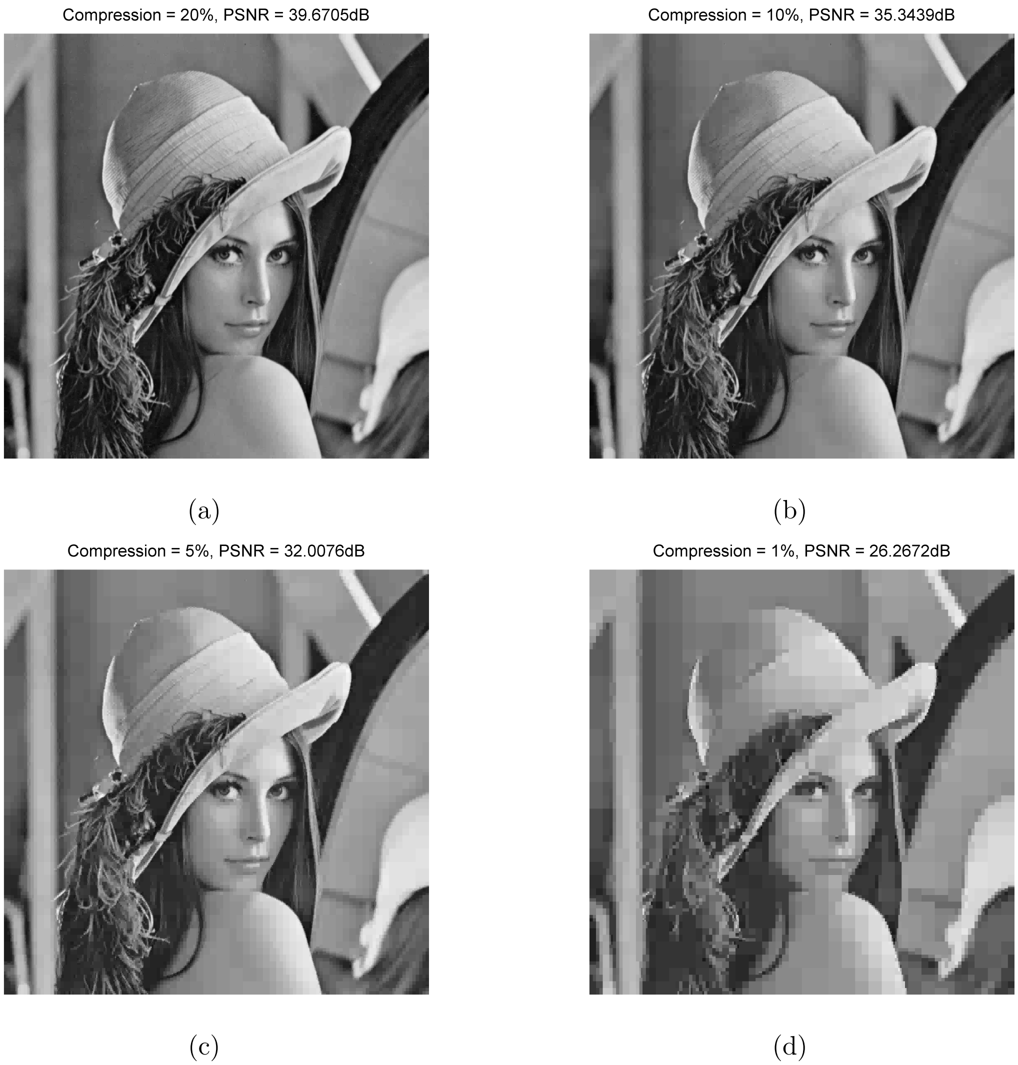


Figure 9: Original *lena* image.


Compression = 20%, PSNR = 39.6705dB
Compression = 10%, PSNR = 35.3439dB
Compression = 5%, PSNR = 32.0076dB
Compression = 1%, PSNR = 26.2672dB

(a) (b)

(c) (d)

Figure 10: Lossy image compression using wavelet transform.

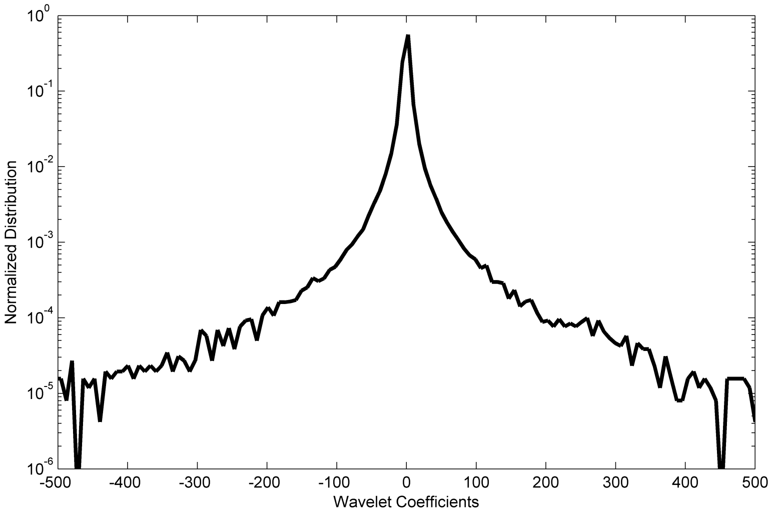


Figure 11: Normalized distribution (histogram) of wavelet coefficients.

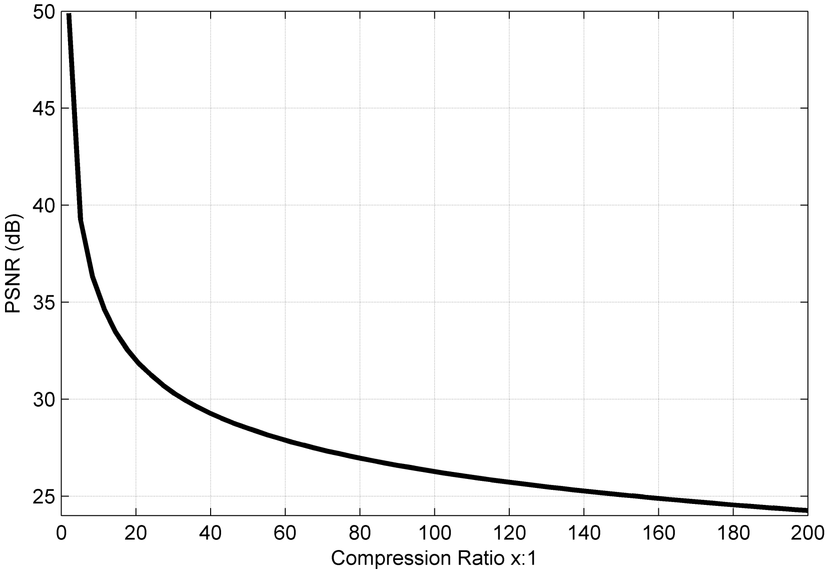


Figure 12: PSNR vs. compression ratio.

# 3 Signal Filtering

Comparing the transfer function expression:

$$
\begin{aligned}
H(z) &= \frac{b(1) + b(2)z^{-1} + ... + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + ... + a(nb+1)z^{-na}} \\
&= \frac{1}{4}[1 + 2z^{-1} + z^{-2}]
\end{aligned}
$$

It implies $a = [1], b = [0.25, 0.5, 0.25]$ for the **filter** command in MATLAB. Fig.13 presents ECG waveform before and after filtering. It appears that some high-frequency components has been filtered out. However, the effect could be hardly observed.

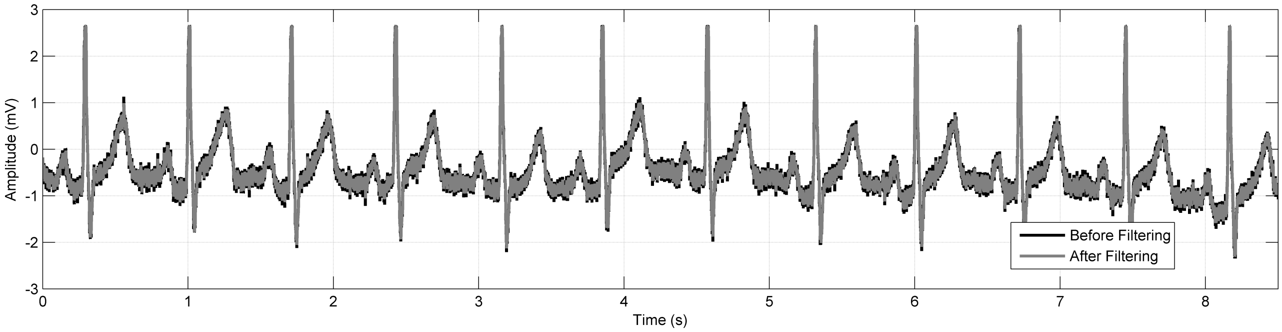


Figure 13: ECG signal before and after filtering.

The signals's Power Spectrum Density (PSD) is shown in Fig.14. Welch's method with window size = 1024 is used to obtain a less "noisy" estimation of PSD. The plot clearly indicate that high-frequency components of the original signal has been attenuated by the filter.

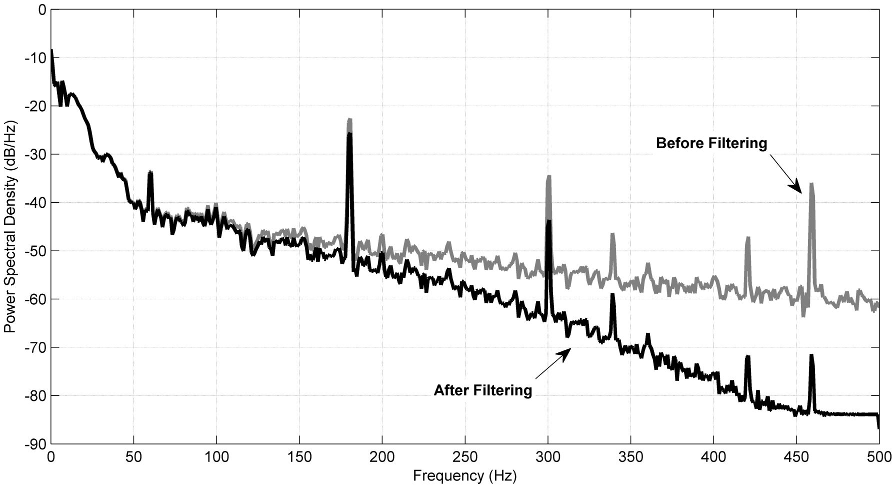


Figure 14: ECG signal Power Spectral Density.

This could also be realized by analyzing the filter's frequency response:

$$\begin{aligned}|H(e^{j\omega})| &= \frac{1}{4}|1+2e^{-j\omega}+e^{-2j\omega}| \\ &= \frac{1}{2}(1+\cos\omega)\end{aligned}$$

Where $\omega$ is the normalized frequency. For $f$ from 0 to Nyquist frequency, Hanning filter acts as a low-pass filter (Fig.15).

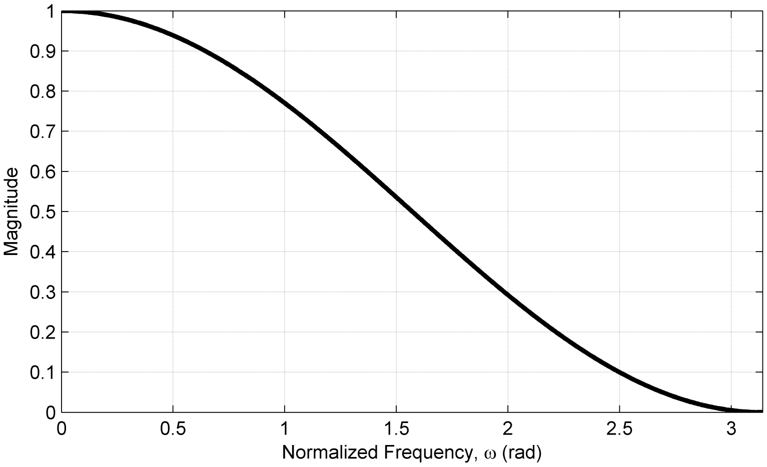


Figure 15: Hanning filter frequency response.

## References

[1] Donoho, D.; and Johnstone, I., Ideal adaptation via wavelet shrinkage, *Biometrika*, vol. 81, pp. 425455, 1994.

[2] Donoho, D.; and Johnstone, I., Adapting to unknown smoothness via wavelet shrinkage, *J. Amer. Statist. Assoc.*, vol. 90, pp. 12001224, 1995

[3] Figueiredo, M. A. T.; Nowak, R. D., "Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior," *Image Processing, IEEE Transactions on* , vol.10, no.9, pp.1322,1331, Sep 2001.