EE3731C – CA2 Project Report

Nguyen Phan Minh – A0074266B

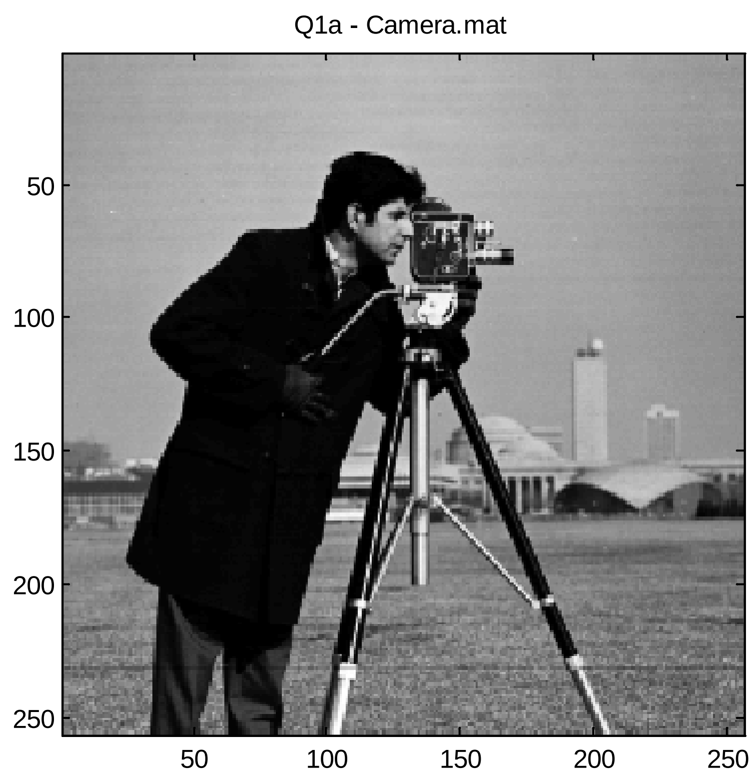
Semester I, AY 2013-14

Question 1: Image Denoising

MatLab files relevant to this question: haar_dec.m, ihaar.m, haar_rec.m, psnr.m, hist_image.m and question1.m. To run all simulations, execute question1.m.

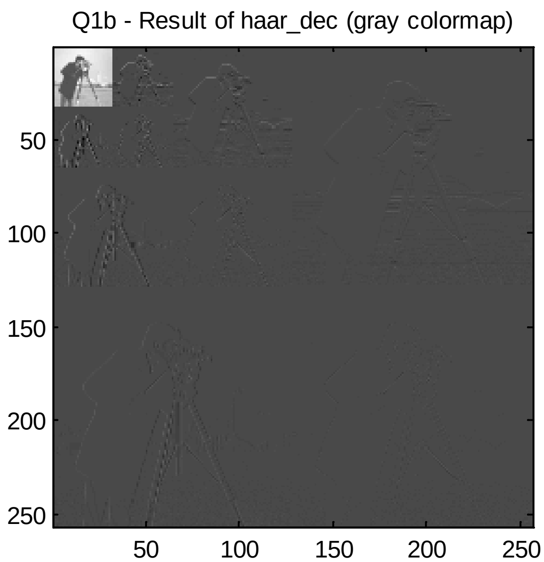
Question 1a:

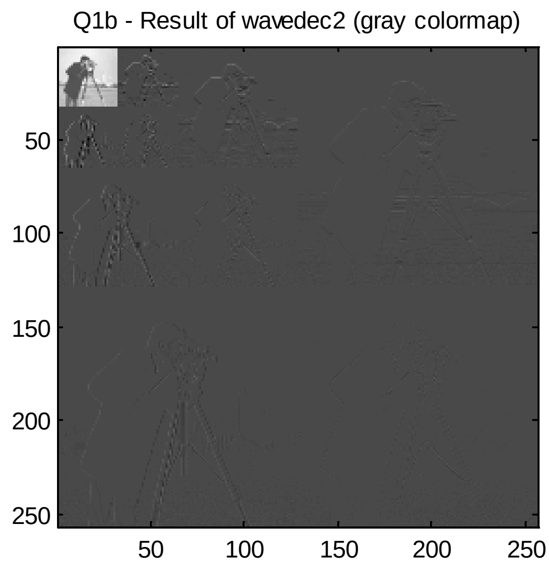
Display the **camera** image.

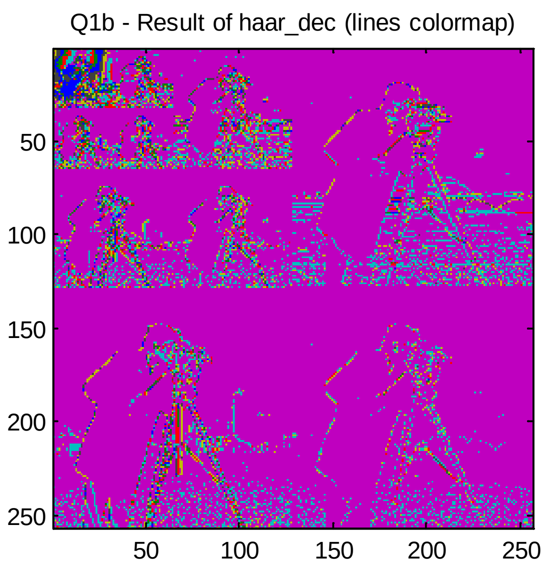


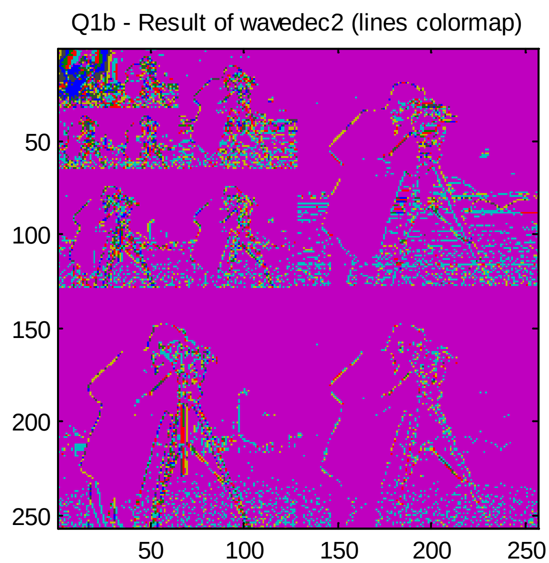
Question 1b:

The idea is simply to apply the Haar transform successively to the low-resolution component among all wavelet coefficients (i.e. the upper left one). Here we go with the understanding that before any transform is applied, the low-resolution component is the original image.



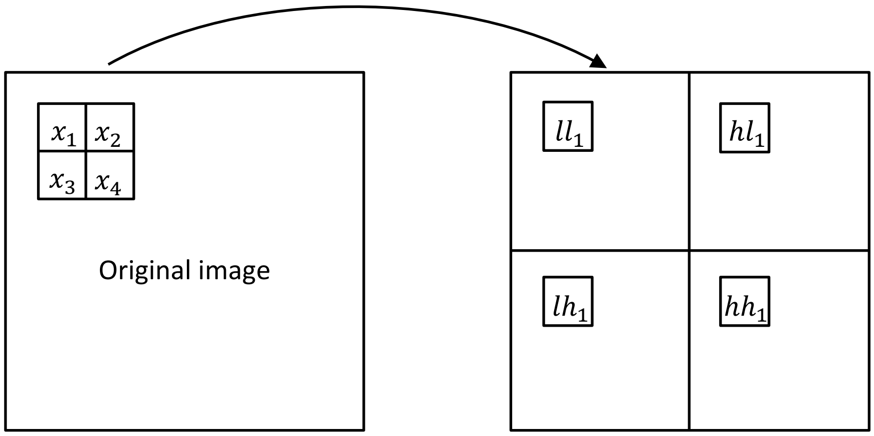






Question 1c:

The idea behind the Haar transform is as follows:



where:

$$ll_1 = \frac{1}{2}(x_1 + x_2 + x_3 + x_4) \qquad hl_1 = \frac{1}{2}(x_1 + x_2 - x_3 - x_4)$$

$$lh_1 = \frac{1}{2}(x_1 - x_2 + x_3 - x_4) \qquad hh_1 = \frac{1}{2}(x_1 - x_2 - x_3 + x_4)$$

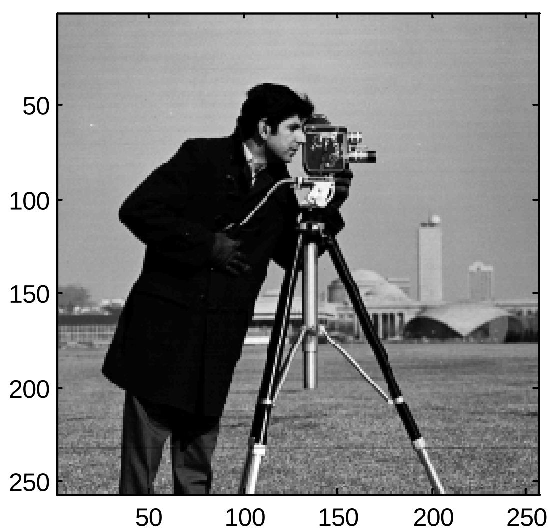
Then conversely:

$$x_1 = \frac{1}{2}(ll_1 + hl_1 + lh_1 + hh_1) \qquad x_2 = \frac{1}{2}(ll_1 + hl_1 - lh_1 - hh_1)$$

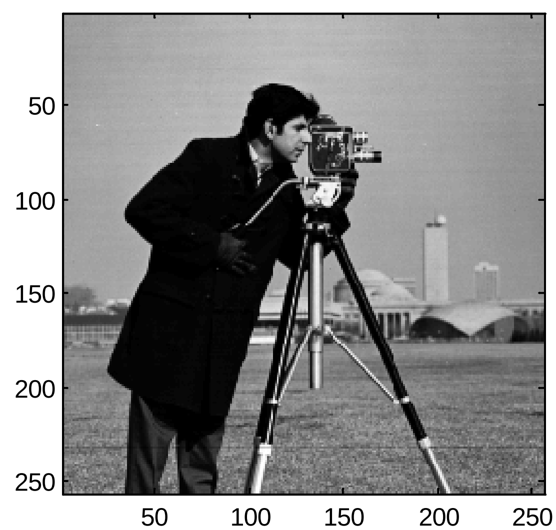
$$x_3 = \frac{1}{2}(ll_1 - hl_1 + lh_1 - hh_1) \qquad x_4 = \frac{1}{2}(ll_1 - hl_1 - lh_1 + hh_1)$$

Those equations define the inverse 1-level 2D Haar transform (implemented in ihaar.m). To perform the inverse J-level transform, again, we only need to perform the inverse 1-level transform successively.

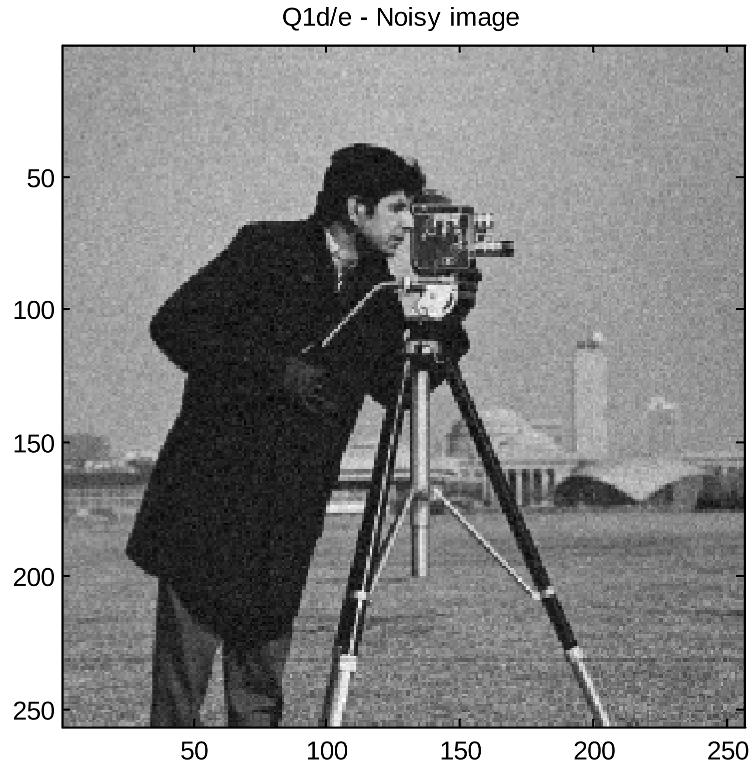
Q1c - Result of haar_rec



Q1c - Result of waverec2



Question 1d/e:



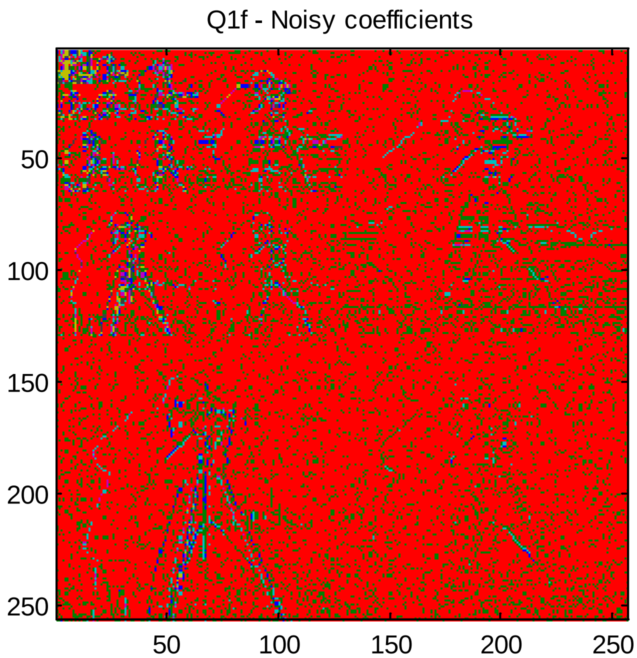
PSNR of the noisy image is about 28.1. The value is not strictly constant, since the noise is random.

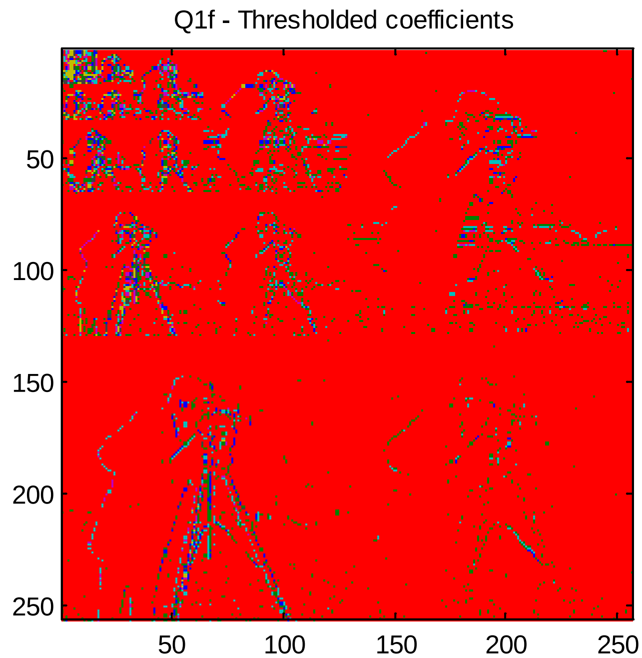
Note that the PSNR is computed using the following:

$$\mathrm{PSNR} = 10\log_{10}\left\{\frac{\mathrm{MAX}^2}{\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}[L(i,j)-K(i,j)]^2}\right\}$$

where $N \times N$ is the size of the image, L and K are the original and denoised image matrices respectively, and $\mathrm{MAX} = \max_{(i,j)} L(i,j)$. This choice of MAX is because the format of the given image is rather unspecified. Even if the image is 256-level gray scale, in which $\mathrm{MAX} = 255$, such choice usually leads to only negligible difference.

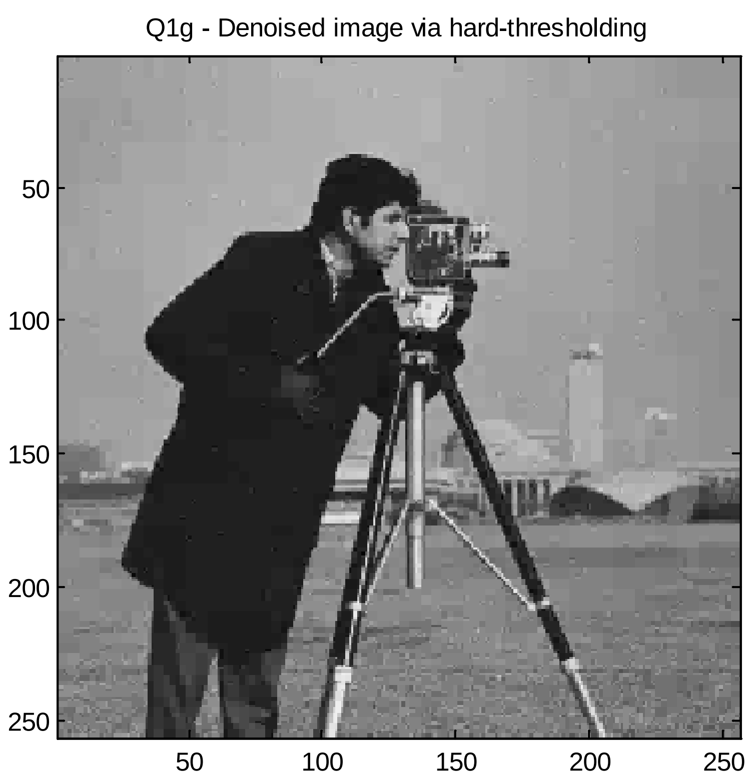
Question 1f:





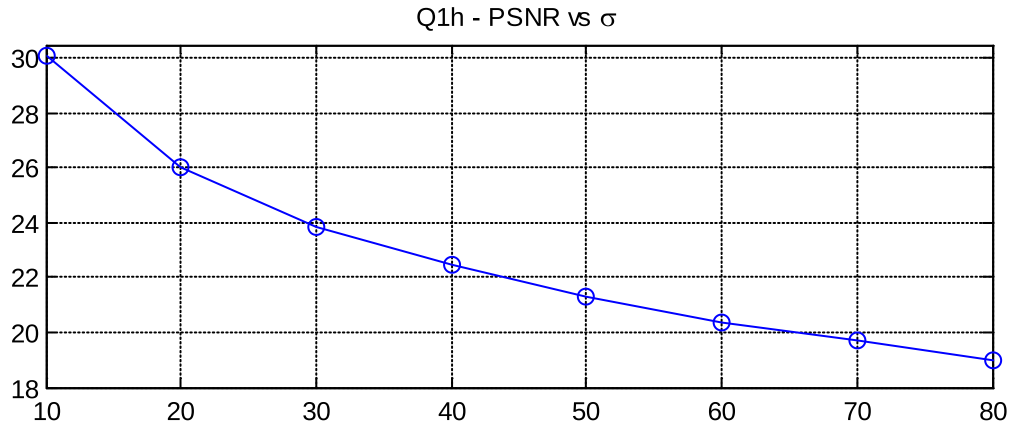
The picture above is under the colormap option of “lines” for visibility. It can be observed that the noisy background of the noisy wavelet coefficients is almost removed after applying the hard-thresholding scheme. The remaining coefficients are either the original ones with high magnitude, or the noisy ones also with high magnitude.

Question 1g: PSNR of the denoised image via hard-thresholding is about 30.04.



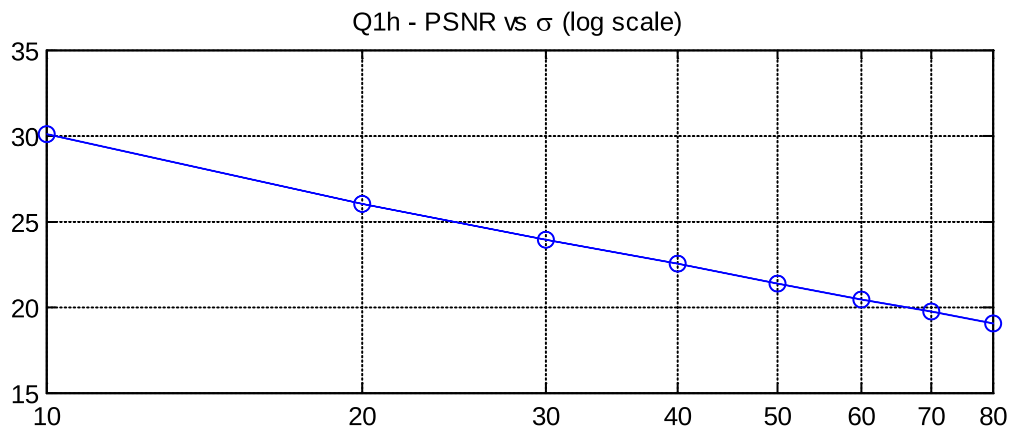
Question 1h:

The following shows the plot of PSNR of the denoised image against σ.



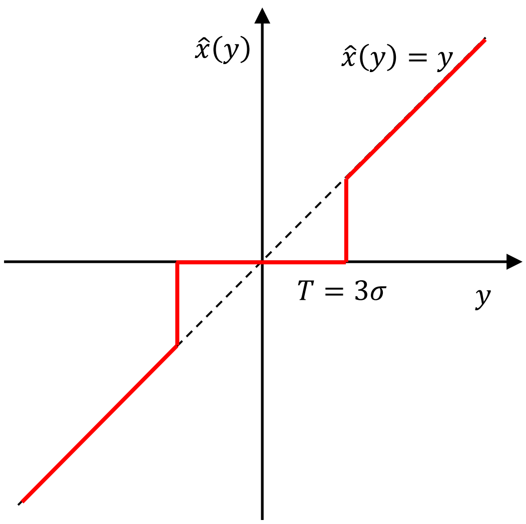
PSNR decreases as σ increases, which is intuitive since higher σ implies worse noise and thus the image is corrupted more severely.

However the decreasing rate seems to slow down. This might be explained as follows: the mean-square error MSE is possibly proportional to σ^2, while $\mathrm{PSNR} \sim \log \mathrm{MSE}$, and therefore, $\mathrm{PSNR} \sim \log \sigma$ (i.e. $\mathrm{PSNR} = \alpha + \beta \log \sigma$ for some constant α and β). This explains the trend observed, and further reinforced by the following plot PSNR vs $\log \sigma$.



Question 1i:

There are two common thresholding schemes in the literature: hard-thresholding and soft-thresholding. Suppose that the observed value of a wavelet coefficient x is y, then: $y = x + n$, where $n \sim \mathcal{N}(0, \sigma^2)$ is the noise. Let $\hat{x}(y)$ be the extrapolated value of x based on the observation y. The two thresholding schemes are then as follows:



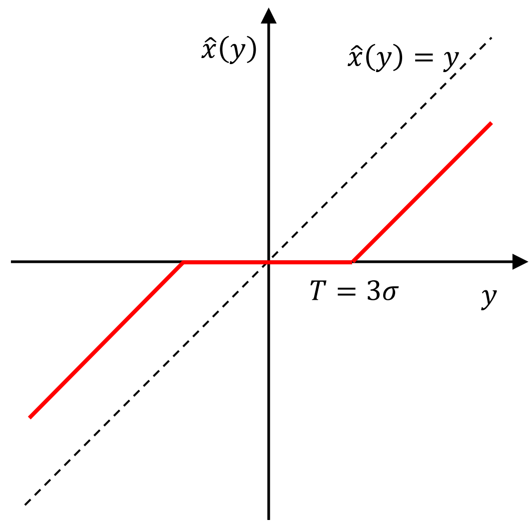


Figure: Hard-thresholding and soft-thresholding schemes respectively.

Soft-thresholding scheme is based on the intuition that since noise also partially corrupts high-magnitude coefficients, one should offset those coefficients. The offset chosen is equal to the threshold $T = 3\sigma$. That is, for soft-thresholding:

$$\hat{x}(y) = \begin{cases} 0, & |y| < T \\ y - T, & y \geq T \\ y + T, & y \leq -T \end{cases}$$

A natural question is why this offset has to be so. Note that, for detail coefficients, the magnitude can be either high or close to 0. When the magnitude is close to 0, regardless of whether the noise perturbs it slightly further away from 0, it is appropriate for the threshold T to be chosen such that it <u>suppresses</u> that low-magnitude coefficient completely to 0. However, the implication of the offset is different; it is not meant to suppress anything, but rather to remove the effect of noise while retaining the high-magnitude coefficient. Note that we can do all these because details of the image are contained more in the high-magnitude detail coefficients.

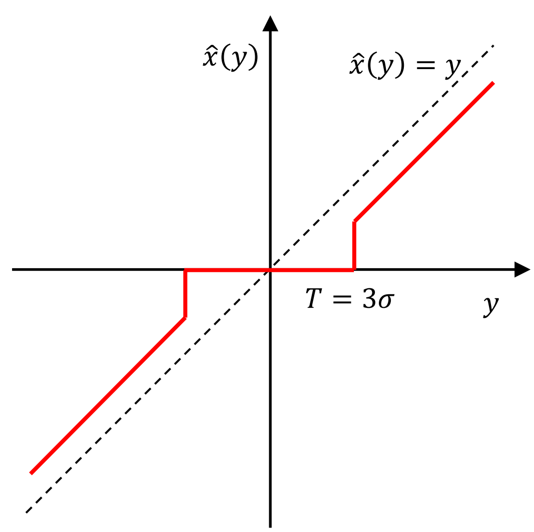
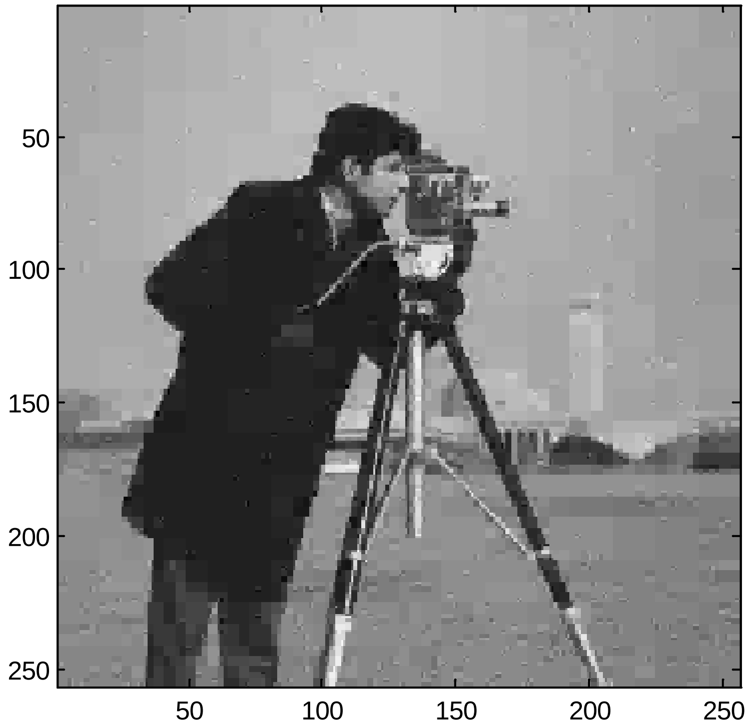


Figure: New scheme

Then the following new scheme is proposed as a result:

$$\hat{x}(y) = \begin{cases} 0, & |y| < T \\ y - \lambda, & y \geq T \\ y + \lambda, & y \leq -T \end{cases}$$

Q1i - Denoised image via new thresholding scheme



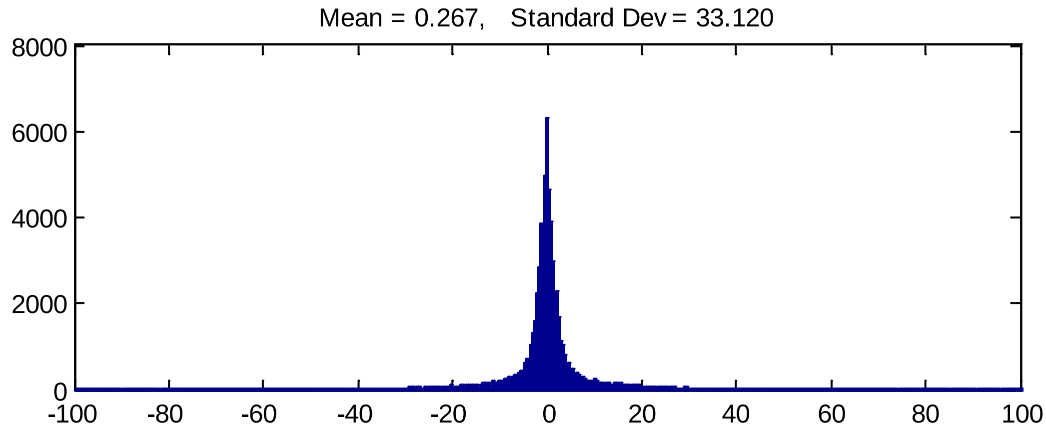
This result is for $\sigma = 20$, $J = 4$, $T = 3\sigma$ and $\lambda = 1.18\sigma$. PSNR of the denoised image via the new scheme is 26.2, while the PSNR of the noisy image is 22.0.

Question 1 – Extra 1:

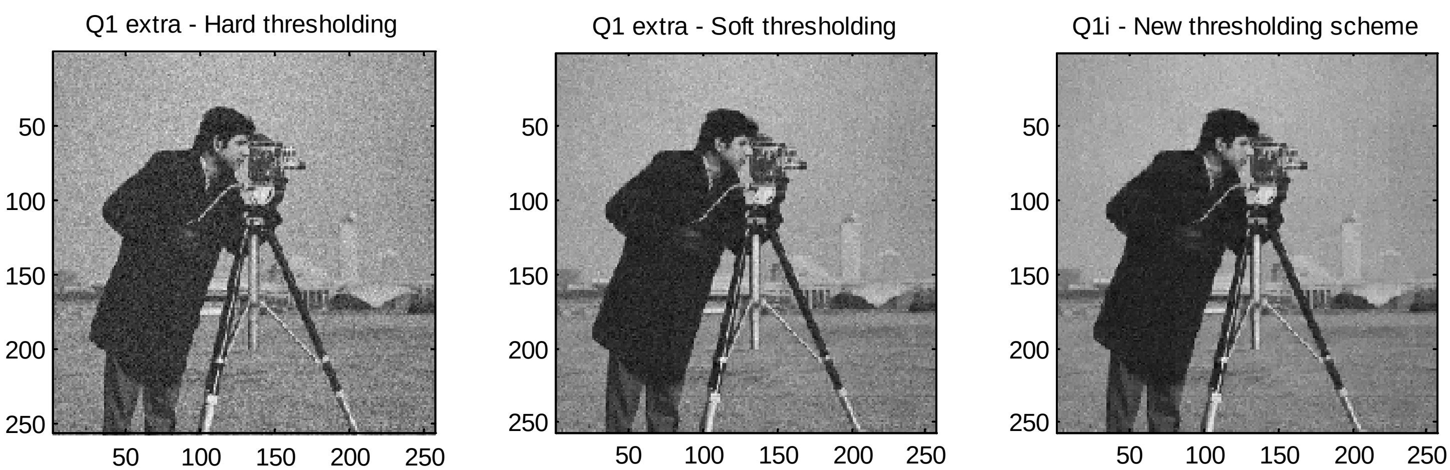
Now to show that the new thresholding scheme gains advantages over the existing schemes, we do the following experiment: we compare the results of 3 schemes under the same threshold T. With $T = 3\sigma$, the difference is marginal, so we would wish to do the comparison under another threshold value. It has been shown[1] that under Bayesian detection framework, a near-optimal choice for soft-thresholding is $T = \sigma^2/\sigma_X$, where σ_X is the standard deviation of the wavelet coefficients of the original image, given that the coefficients follow a Gaussian distribution. We shall use this choice of T in the following experiment.

Implementing the function hist_image.m, we obtain the following histogram for **camera** image:

[1] S.G. Chang, Y. Bin, M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol.9, no.9, pp.1532-1546, Sep 2000.



The distribution is approximately Gaussian. Hence, our choice would be $T = \sigma^2/33.1$.



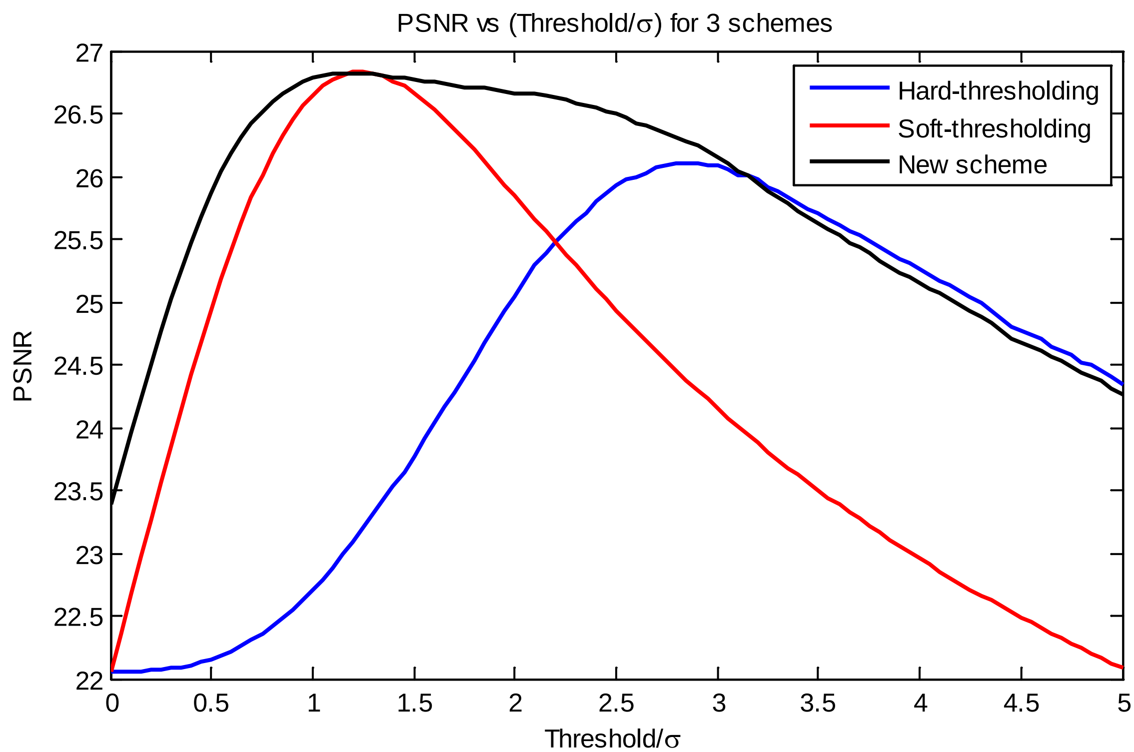
These results are for $\sigma = 20$, $J = 4$, $T = \sigma^2/33.1$ and $\lambda = 1.18\sigma$. The PSNR values are as follows:

- PSNR of the noisy image = 22.0
- PSNR of the hard-thresholding scheme = 22.2
- PSNR of the soft-thresholding scheme = 25.4
- PSNR of the new scheme = 26.2

Visually the new scheme one looks less noisy in the background as compared to the others. The improvement is thus obvious, both visually and in terms of PSNR.

Question 1 – Extra 2:

Along the same line, we would wish to expand our investigation for a range of threshold value T. The performance of the schemes would then be clearer. In the following, we plot the PSNR vs T/σ for the three schemes, keeping $\sigma = 20$, $J = 4$ and $\lambda = 1.18\sigma$.



It can be observed that the optimal points of soft-thresholding and the new scheme are the same, and they outperform the optimal point of hard-thresholding. Interestingly, the performance of the new scheme dominates the other two schemes for most T/σ values, with negligible loss of performance compared to hard-thresholding for large T/σ. The new scheme is thus more **robust** to variations in either T or σ, which makes this scheme more practical[2].

---

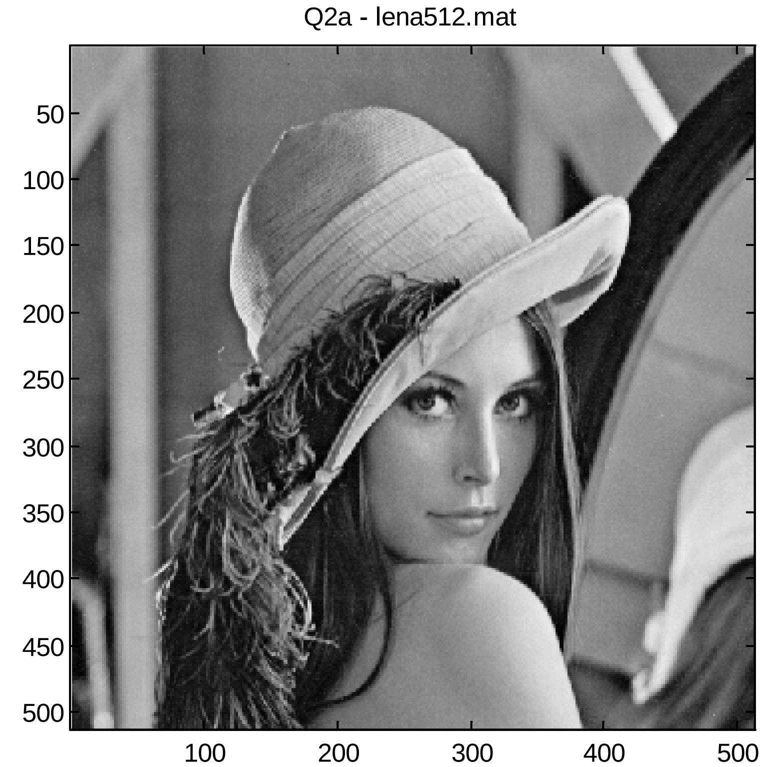
[2] Intuitively, the relative performance of the schemes remains the same when σ changes and the threshold T is adjusted accordingly. As such, it is the ratio T/σ that matters to the relative performance.
In practice, noise can vary (i.e. the noise model is colored noise, not white noise). However, the system may not adjust T by itself, and thus, the ratio T/σ varies. As shown in the plots, the new scheme is most robust against such variation in T/σ. We can conclude that its relative performance is best and most practical among the three schemes.

Question 2: Image Compression

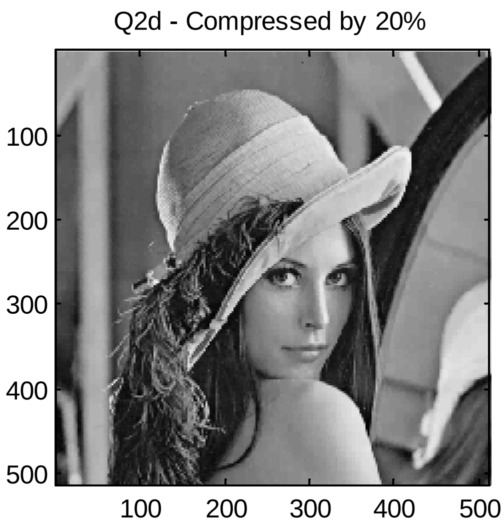
MatLab files relevant to this question: question2.m. To run all simulations, execute question2.m.

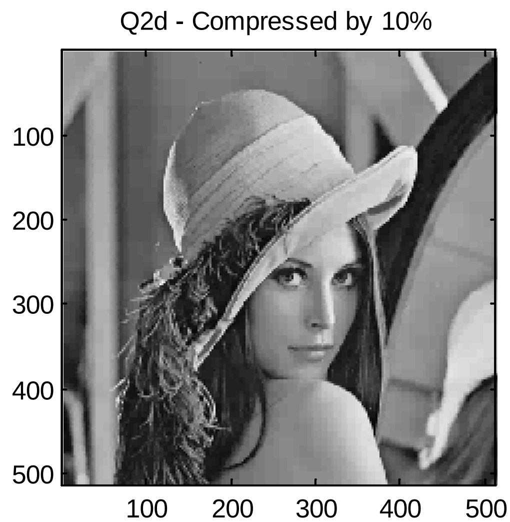
Question 2a:

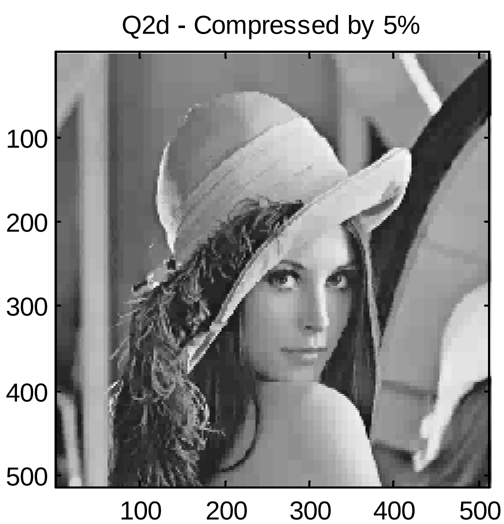
Display the **Lena** image.

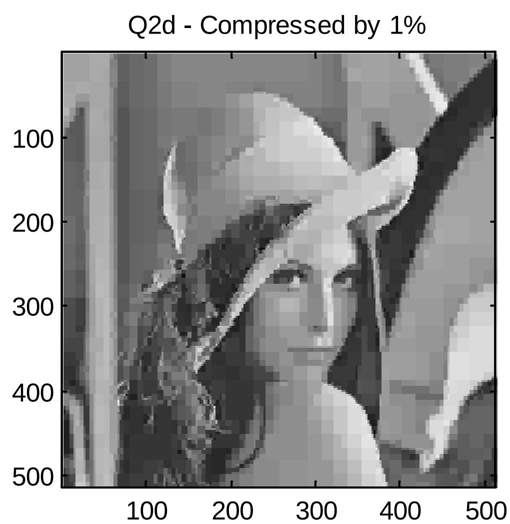


Question 2d:





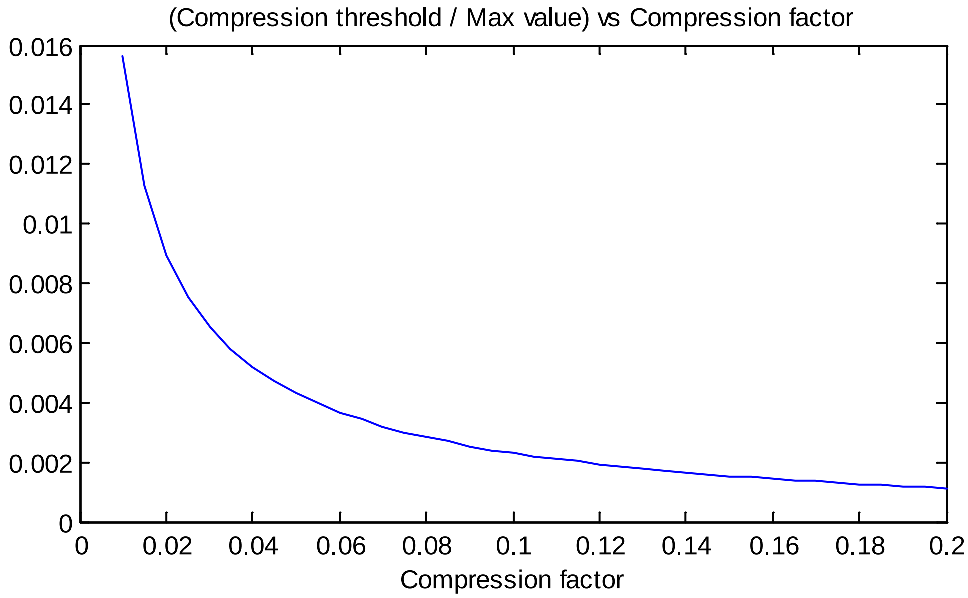




PSNR values:

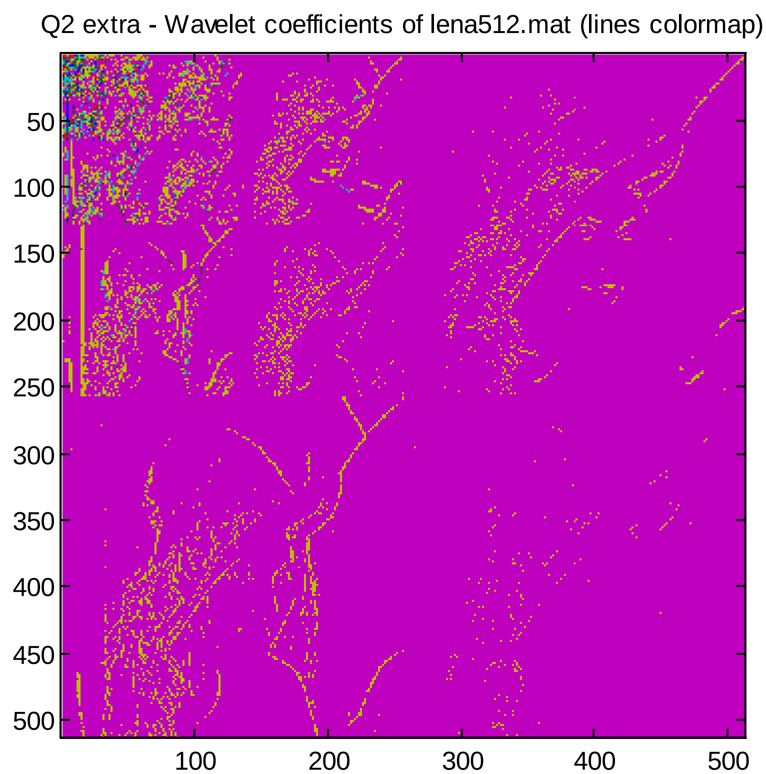
- PSNR = 39.3 (Compression factor = 20%)
- PSNR = 35.0 (Compression factor = 10%)
- PSNR = 31.6 (Compression factor = 5%)
- PSNR = 25.9 (Compression factor = 1%)

The visual quality is hardly compromised for 20% and 10% compressions, and amazingly is not severely degraded even for 5% and 1% compressions. This can be explained by the following graph:



Here the compression threshold is the value below which the coefficient is compressed to 0 and max value is the largest coefficient. Observe that even for 1% compression, the compression threshold is still very small compared to the max value, which means those coefficients that are compressed to 0 do not carry much information. This explains the good visual quality observed in the compressed images.

This is not surprising, since the wavelet coefficients are generally sparse.



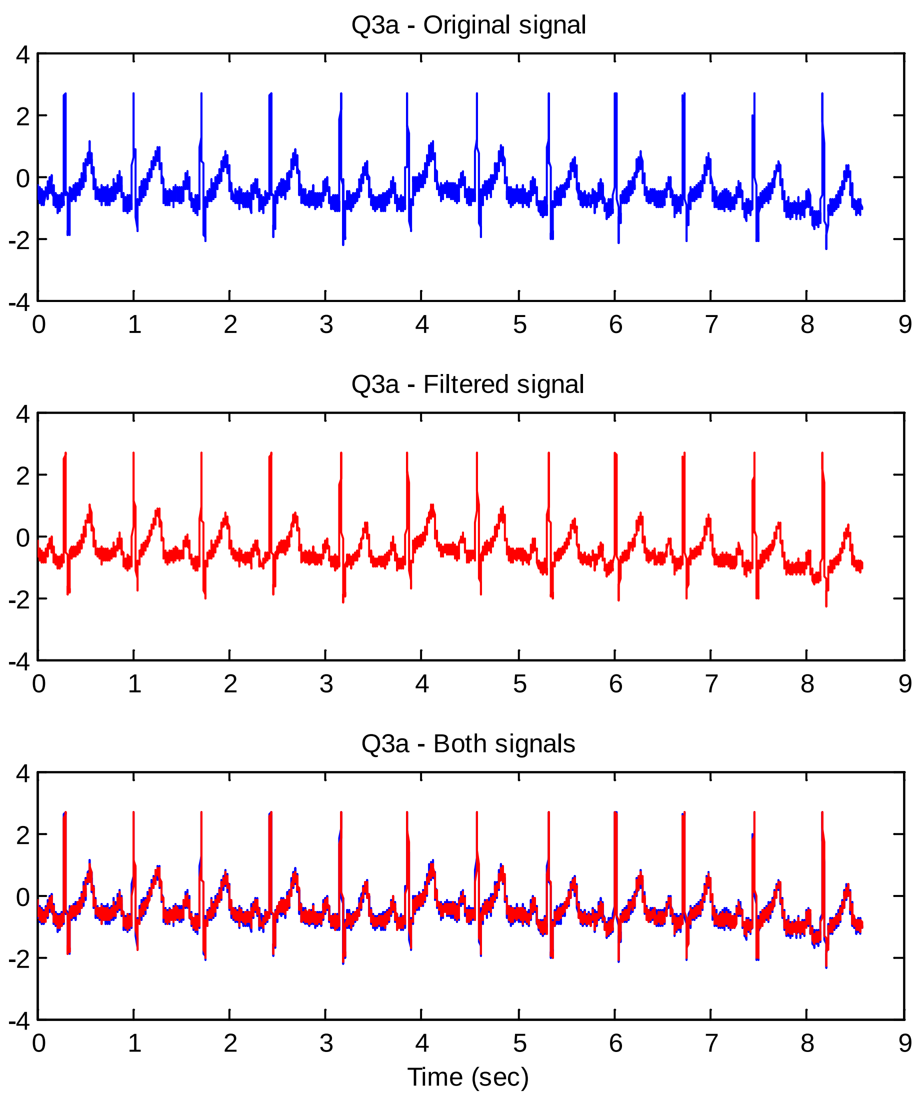
Question 3: Signal Filtering

MatLab files relevant to this question: question3.m. To run all simulations, execute question3.m.

Question 3a:

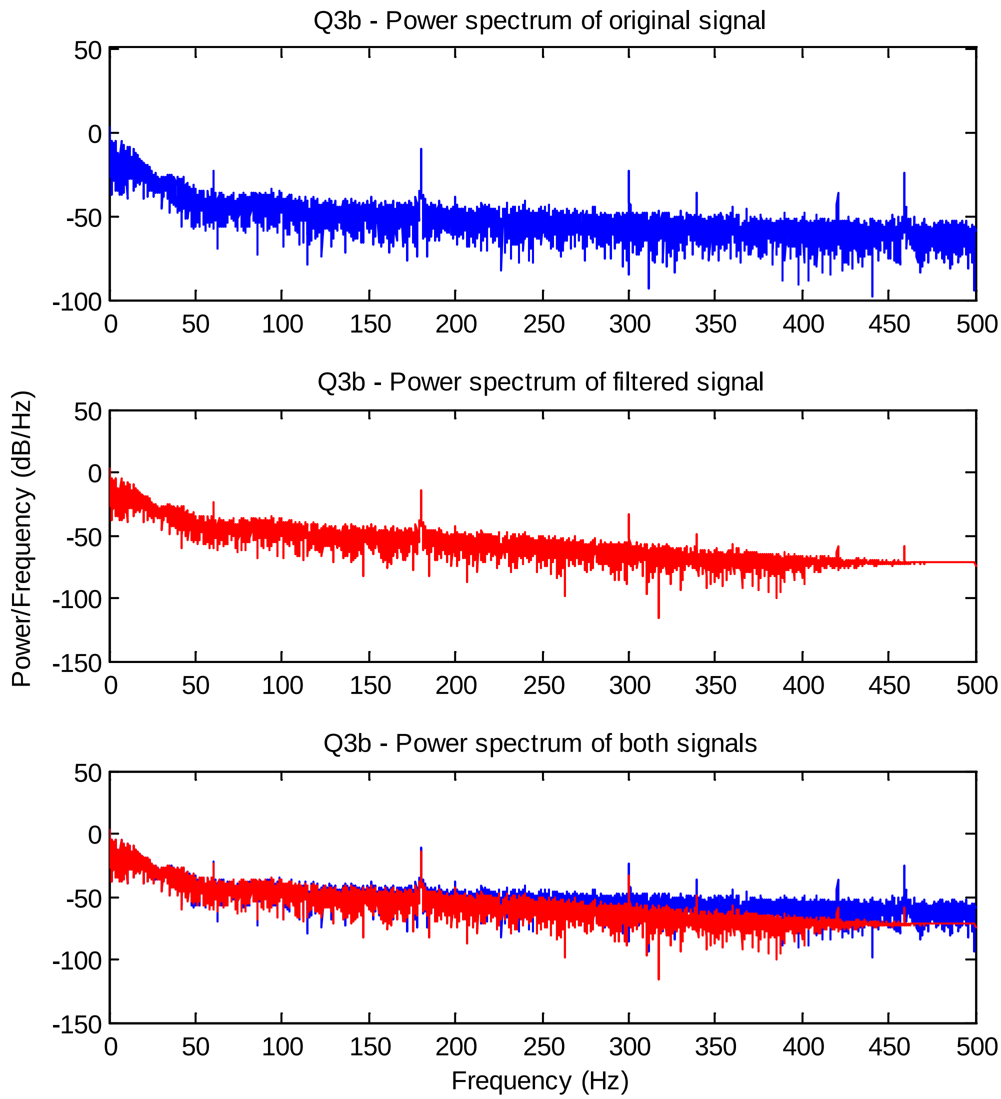
By inspection, one can see that $a = [1]$ and $b = [0.25 \quad 0.5 \quad 0.25]$. This is because:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b(1) + b(2)z^{-1} + \ldots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \ldots + a(nb+1)z^{-na}}$$



Question 3b:

The spectrum is plotted using the periodogram method.



The spectrum shrinks as the frequency increases to $f_s/2$ (where f_s is the sampling rate). The Hanning filter seems to remove high-frequency components. This is because of the particular shape of $\left|H\left(e^{j\omega}\right)\right|$:

$$\left|H\left(e^{j\omega}\right)\right| = \left|\frac{1}{4} + \frac{1}{2}e^{-j\omega} + \frac{1}{4}e^{-j2\omega}\right|$$
$$= \frac{1}{2}(1 + \cos\omega)$$