MSc Computer Vision 2014-2015
Advanced Image Analysis
Wavelet Decomposition and Filter Bank

Marwan Osman
marwan.aosman@gmail.com
MSCV 5

This report will show the results of implementing a simple wavelet transform using the Daubechies D4 filter, and its application to image denoising.

Implementation

The required wavelet transform is implemented in the function `fwt_2d_marwan.m`. This function computes the `j-level` wavelet transform of an input NxN image. The function has the following inputs:

- **mode**: Indicates whether to compute the forward or inverse wavelet transform. `0` indicates forward, `1` indicates inverse.
- **input**: This is the input NxN image to be transformed. N is assumed to be a power of 2.
- **nlevel**: The required number of levels to compute for the wavelet transform.
- **h**: The Analysis low pass filter used for the transform.

And the **output** of the function is the wavelet transform coefficients, if the transform was forward, or the reconstructed image, if the transform was inverse.

The function is implemented as a recursive function, since for each level we need to compute the transform for half of the coefficients for next level. First, we construct the high pass filter from given low pass filter. Depending on the transform mode, we begin computing the transform.

Forward wavelet transform is applied on the input image on each row first. Then it is applied again on the temporary output of the previous operation but this time on each column. If we still have more levels to compute, the function is recursively called again and given the first quadrant of coefficients which is relevant to the transform.

Inverse wavelet transform is similar to the forward one, except that the inverse transform is called recursively on the first quadrant of coefficients, then applied on each column then on each row. It follows the inverse steps of the forward implementation.

Results

For analysing the application of wavelet transform on denoising images, we start by adding Gaussian white noise on the test image. Then we compute the transform coefficients.

There are two methods of thresholding noise: Soft Threshold and Hard Threshold. The difference between the results of the two methods is shown in Fig.3. We can notice that the Soft Thresholding adds blurring to the image and looses edge information much more than Hard Thresholding. This is expected as Soft Thresholding is not preserving the original wavelet coefficients values, but rather keeping the difference between the coefficients and the thresholding parameter.

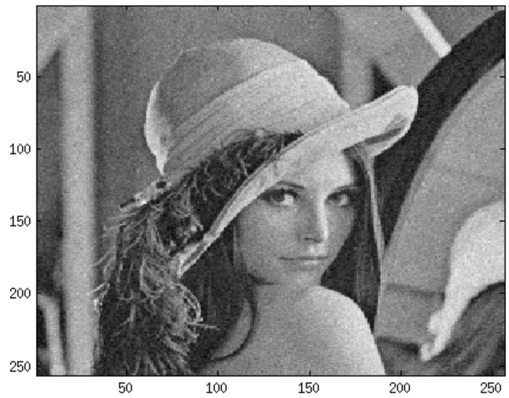

Figure 1: Testing Noisy Image Input

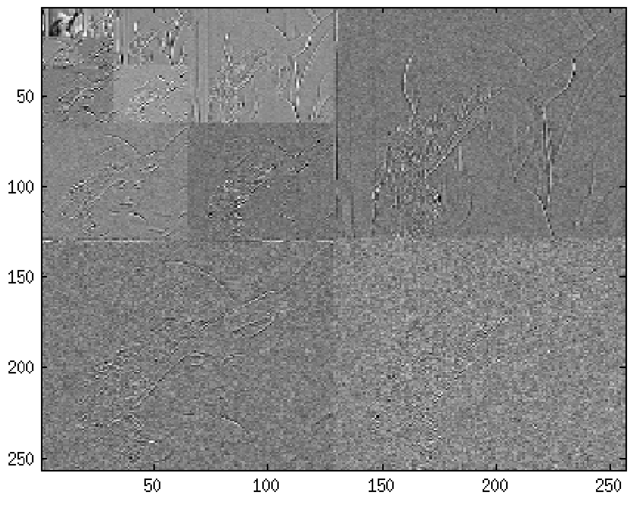

Figure 2: Wavelet Transform for Noisy Lena Image

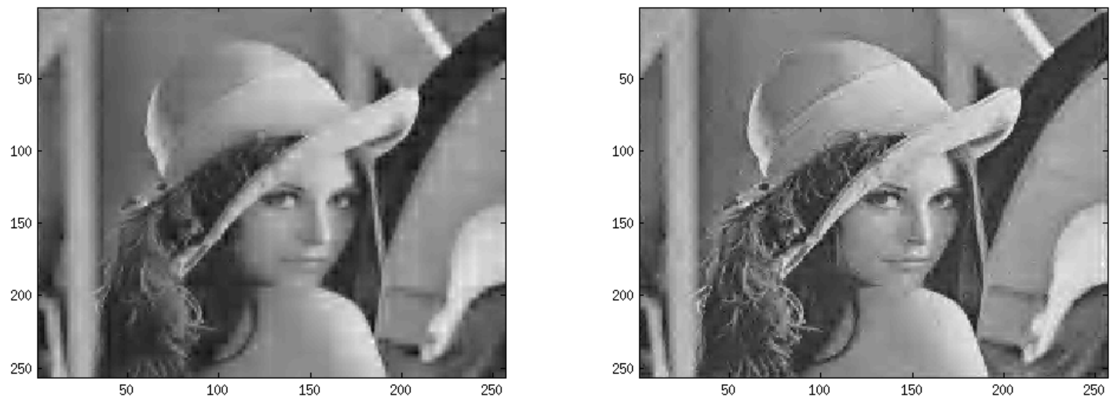

Figure 3: The difference between soft and hard thresholding the coefficients of the wavelet transform.