Signal and Image Processing - Assignment 4

Assigned May 29^{th} **Due June 23^{rd} 23:55**

Time you spent on this assignment: ____ hours

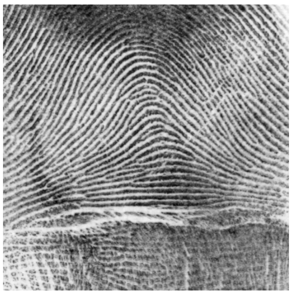
You need to submit a PDF report and one Python ($\leq$ v.3.6) script for each task, all in the same folder. Make sure that each file is named according to the task. You can also choose to upload a Jupyter Notebook containing both the functions and the report. Please write (at least) your name and student ID into each file you submit. Feel free to define any other auxiliary functions.

The maximum amount of points awarded for this assignment is **30 points**.

The purpose of this lab is to get a feel for a wavelet transform. You will transform an image into wavelet space and manipulate it there. After that you will reconstruct it into the spatial domain to understand what just happened. You can use any implementation of wavelet transform as long as it is restricted to **1D** processing.

For testing, you can use one of the following images (`http://vda.univie.ac.at/Teaching/SIP/19s/Lab4/test_images.zip`) or a suitable image of your choice as long as it has at least 512x512 pixels.





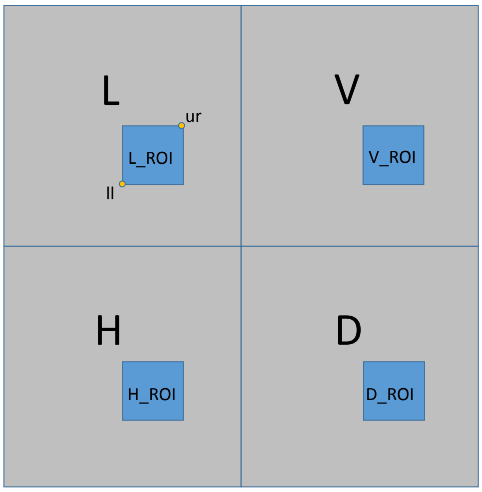

1 2D Wavelet Transform [15 Points]

(a) (8 points) *Encoding.* Implement a 2D wavelet and inverse 2D wavelet transform using 1D wavelet transforms. It should have the following interface:
`y2d = wt2d( x2d, Lo_D, Hi_D, nlevels )`
`x2d = iwt2d( y2d, Lo_R, Hi_R, nlevels ),`
where `x2d` is the input image, `Lo_D` and `Lo_R` are the 1D low-pass approximation filters while `Hi_D` and `Hi_R` are the 1D high-pass detail filters for decomposition and reconstruction, respectively. The number of desired scales is specified by `nlevels`. The encoded signal `y2d` should have the standard layout (Fig. 6.30 in the textbook), having approximation at the top-left block and the detail coefficients around it.

(b) (1 point) *Energy preservation.* Apply a 5-level Haar wavelet transform to the image. Check that the Haar wavelet transform is orthogonal, which means that the energy (`norm`) of the coefficients is the same as the energy of the image.

(c) (2 point) *Perfect reconstruction.* Apply a 5-level Haar wavelet to the image and reconstruct it back. Show that a perfect (up to numerical precision) reconstruction is achieved by computing a mean squared error (MSE) between the original and the reconstructed image.

(d) (4 point) *Spatial localization.* To show how wavelet transform is localized in space, unlike Fourier transform, perform a reconstruction (`iwt2d`) of **only** a specific region of interest (ROI) in the image. See the figure below for clarification.



Implement the following function:
`x2d_ROI = iwt2d_ROI( y2d, Lo_R, Hi_R, nlevels, ll_x, ll_y, ur_x, ur_y ),`

The four additinal arguments should specify the bounding box (defined by lower-left corner (`ll_x,ll_y`) and an upper right corner (`ur_x, ur_y`)), were the `x` and `y` coordinates are in the range [0,1] relative to the image width and height, respectively.

Hand in:

- Your `wt2d`, `iwt2d`, and `iwt2d_ROI` functions as specified.
- A script `task1`, computing the required steps in order and outputing all intermediate results as **captioned** figures. Use comments to mark each subtask.

2 Wavelet-based Edge Enhancement [5 points]

(a) (3 points) Perform wavelet-based edge detection by setting the approximation component of the wavelet transform to zero, analogous to the `Example 6.23` in the Gonzalez, Woods 4ed textbook.

(b) (2 points) Use the detected edges to enhance (*sharpen*) the original image by making its edges more pronounced.

Hand in:

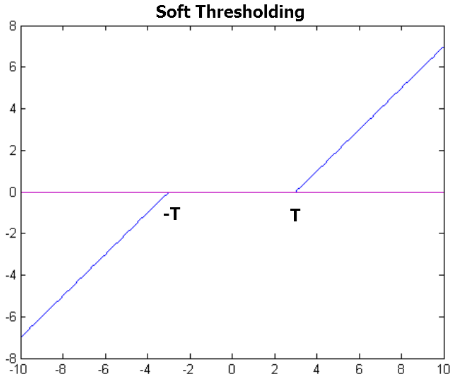
- A script `task2`, showcasing your work, analogous to the previous task.

3 Wavelet-based Denoising [10 Points]

(a) (1 point) Add Gaussian noise (e.g., σ of 10% of max image value) to the original image. Visualize both images (the original and its noisy version) side by side. Compute peak signal to noise ratio (PSNR) between the two images.

(b) (3 points) Perform *a three scale* wavelet decomposition, threshold the detail coefficients and reconstruct the image. Implement soft thresholding scheme (*wavelet shrinkage*) defined as:

$$\Theta_T(x) = \max\left(0, 1 - \frac{T}{|x|}\right) x$$



Draw a graph of PSNR (y-axis) vs. threshold T (x-axis). Thresholds should range from 0 to maximum detail coefficient value.

(c) (6 points) Implement a subband adaptive threshold known as *NormalShrink* described in this paper: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.9103`.
Perform *a three scale* decomposition and reconstruct the image after the *NormalShrink* thresholding. Display the encoded image in the wavelet domain before and after thresholding. Report the obtained PSNR. How does it compare to the best value obtained in the previous subtask?

Hand in:

- A script `task3`, showcasing your work, analogous to the previous tasks.