Exercises for
Introduction to Pattern Recognition (IntroPR)
Lennart Husvogt
Work sheet 12, 15.01.2016

Wavelets: A Compression Example

Exercise 27 **Programming Task:** In this exercise, we simulate an image compression scenario. Let us assume that we have a (hypothetical) compression algorithm where it costs nothing to encode a 0. All other values are encoded in one byte, regardless of their size.

With this approximation of a true compression algorithm, we are going to quantify the benefits of wavelets for image compression. In many applications, e.g., image compression, enhancement and analysis, Haar wavelets are frequently used because of their simplicity. Thus, the goal of this exercise is to implement the discrete wavelet transform using Haar wavelets.

In your code, you will need these (short) components:

(a) A method to transfer a grayscale image into the discrete Haar wavelet domain, i.e., to decompose the image into low-low, low-high, high-low and high-high coefficients.

(b) A method for the inverse transform, i.e. to obtain a grayscale image from the Haar wavelet coefficients.

(c) A loop to perform these two methods on multiple scales, i.e., to repeatedly encode the low-low coefficients of the previous step. Conversely, the inverse discrete wavelet transform should also be able to operate on multiple scales.

Now, for our compression application: A straightforward compression algorithm is to determine a threshold θ. All wavelet coefficients that are lower than θ are set to 0. Experiment with different values for θ: when does the image quality become too poor? How many 0's do you introduce into the image? The amount of 0's is your compression ratio. Try to save a decent amount of storage with aggressive thresholding — but have also an eye on how the image quality deteriorates.

For your experiments, you can use the black/white Lena image from our web page, or use your own images.