Instruction for Image Compression using PCA

©Signals and Systems Group
Uppsala University

August, 2005

1 Introduction

In this laboratory work you will study principal component analysis(PCA) using the $Matlab^{TM}$ environment. Feature extraction and data compression are closely related problems that can be attacked using PCA. The task here is to perform feature extraction based on a real picture, the famous "Lena" which has been used for evaluation of many image processing and image compression methods since the 1970's.

After loading the image into matlab (`load lena`), it can be presented in the graphical window by means of the command `imagesc(lena)`. You should also change the colors using `colormap(gray)`.

2 Extraction of non-overlapping 8x8 blocks

The image compression approach considered here is based on the idea that subimages of Lena can be described using relatively few features. Instead of storing the original image, the values of the features in the subimages are stored (and then used for reconstruction).

Therefore you should first use the m-file `extract.m` which scans the image "Lena" and extracts non-overlapping $8 \times 8 = 64$ pixel blocks. After extraction, each block is converted into a column vector[1]. 1024 blocks from the image are extracted resulting in a 64×1024 matrix.

3 PCA Analysis of Extracted Blocks

The following subtasks should be completed.

1. Use the MATLAB function `cov.m` and `eig.m`[2] to perform a PCA analysis of the covariance matrix of the data set and determine the 10 largest

[1] In Matlab, a matrix A is converted into a column vector a by means of the command `a=A(:)`. You can reconstruct A from a by means of `reshape.m`

[2] You can use `svd.m` instead of `eig.m`.

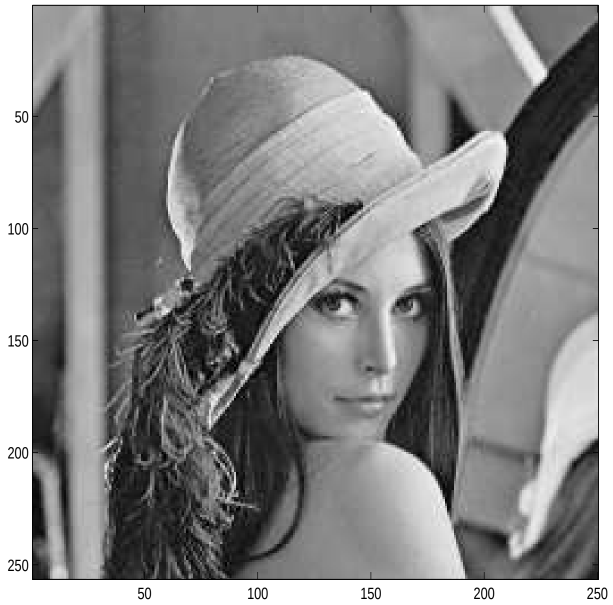

Figure 1: The famous test image "Lena"

eigenvalues and the corresponding eigenvectors. Which are the 10 largest eigenvalues? Which is the mean value (mean image)? Use the function `subplot.m` and plot the eigenvectors (eigenimages) which correspond to the six largest eigenvalues in the same window.

2. Use the mean vector and the eigenvectors correponding to the 10 largest eigenvalues to compress the subimages.

3. Try to reconstruct each subimage by means of its low-dimensional representation. Don't forget to include the mean value!

4. Use the m-file `reconstruct.m` to convert the reconstructed subimages into a 256 × 256 pixel image again. You can use `imagesc.m` to present it in the graphical window. Make a presentation where the original and the reconstruction are shown together.