Exercise I

Introduction to MATLAB Tools for Image Processing

Digital Photography
EPFL/IC/IVRG
2 sessions, starting 21 February 2014

1 Basic matrix operations

Manipulating matrices in MATLAB is easy as it is what it was created for. Images are matrices and most operations on images will be simple additions, multiplications, inversions. If you are not familiar with MATLAB, create small matrices and try some basic manipulations. See for example functions sum, size, inv, ones, zeros, find. See also basic operations like multiplication ($\star$) and power ($\hat{}$) and their element-wise equivalents $.\star$ and $.\hat{}$, respectively, and make sure to understand the difference. The functions max, min, mean are also useful to check if images are well normalized. The goal of this lab is to get familiar with the MATLAB Image Processing Toolbox. This exercise session articulates in two parts. The first one is a demo intended to guide you in a (short) IP tour. In the second part, you are asked to solve an exercise, to make you familiar with the most used set of functions in the IP Toolbox.

- Start MATLAB.
- Run exercice1_demo_IP.
- Read carefully the graph titles and the command line.
- Open the M-file "exercise1_demo_IP.m", and figure out what is going on.

Load the image "mandrill.jpg" and include the path into your M-file "exercise1_demo_IP.m".

2 MATLAB Exercise - Doing it Yourself

2.1 Matrices multiplication and applying linear transforms

1. Load the file "exercise1.mat", which contains eight 3×3 matrices. In this section, we work with six of them (A1 to A6).

2. Load the file "vega.jpg" and convert it into the double format.

3. For matrices A1, A2, and A3, can you already tell what the effect will be?

4. Apply each matrix (A1 to A6) to the image and display the original and six resulting images on one figure using the command subplot and add titles.

Hint: Images are $m \times n \times 3$ matrices and to apply a 3×3 transform, you must first reshape the image into a three-column matrix using the command "reshape".

2.2 Manipulating images

1. Load images "lavaux.jpg" and "polylan.jpg". Get the gray-scale version of both images.

2. Display the images using the function imshow.

3. Determine (using MATLAB coding) which one of the two images is brighter.

4. Change (using an automated form) both gray images so that they have approximately the same average brightness. Display them. Any comments?

   Merge the 2 images in the following ways (and display the result):

5. Top half of one image, bottom half of the other.

6. Interleave the lines.

7. Interleave the columns.

8. Interleave 15 pixels wide columns.

9. Add the images.

10. Add the images and normalize.

11. Average the images.

12. Load image "bc.bmp". Save the image in the TIFF format. What are the saving options?

13. Save the image in the JPEG format, with a quality factor of 25%. Load back the image. What differences do you see?

Hint: You may want to use a loop to solve 8.

2.3 Finding and replacing values in an image

1. Load the image "I_noise1.tif". This image contains Salt and Pepper noise, i.e. some pixels are clipped to values 0 and 255 (or 0 and 1 if the image is in the double format).

   We want to attenuate the noise:

2. Using the command find, retrieve the indices corresponding to the pixels of the image having the values 0 and 255.

3. Replace these pixels by the value 127. Can you still see the noise? We will now try to eliminate the noise using another noisy image with differently distributed noise.

4. Load the image "I_noise2.tif". Use this image to replace the noisy pixels in image "I_noise1.tif". Can you still see the noise? Is it better than the previous method?

Hint: Use the indices of the noisy pixels you found previously to get the values in image "I_noise2.tif".

2.4 A very brief overview of image filtering

We will see more of image filtering later in the class, but here is a short preview of what can be done.

1. Load image "bc.jpg" and convert it to a double format gray scale image.

2. The file "exercise1.mat" also contains two 3×3 matrices (filters) F1 and F2. Use the function imfilter to filter this image using F1 and display the resulting image. What effect does this filter have?

3. Use the function imfilter to filter this image using F2 and with its transpose and display the two resulting images. What effect does this filter have?

General hint: When in doubt, look it up in MATLAB help.
Note: Be careful of the figures in MATLAB when dealing with images. You may want to close the figure before displaying a new one.
The solution of this exercise will be posted on moodle on Friday, March 7th.