Lab 1, DIP1

The following lab will introduce basic commands to get started doing Image Processing using MatLab. You'll implement different methods used for image scaling (image enlargement).

1. Load the image *cameraman.tif* (you'll find it on the course homepage) and import the image in MatLab using the command *imread().* Determine the image dimensions. Find the max and min pixel value in the image.

2. Show the image on the screen using the command *imshow().*

3. Show the part of the image that consists of the rows from 100 to 150 and columns from 100 to 150.

4. Use the histogram function *hist()* on the image *cameraman.tif* (you'll need to cast the pixel values of type uint8, using the command *double()*)

5. Generate a new image by setting the pixel values below the mean value in the original image to 0 and the values above the mean value are set to 255 (Hint used the command: *find()*). Check that the new binary image only contains the values 0 and 255.

6. Enlarging the image 3 times using *nearest neighbor interpolation* (GW p. 66).

On page 66 in GW the method *bilinear interpolation* is described. Bilinear interpolation interpolates the pixel values inside the rectangle with vertices (x_0, y_0), (x_0, y_1), (x_1, y_0) and (x_1, y_1). If the interpolated value is denoted $f(x_0 + \Delta x, y_0 + \Delta y)$ the values is calculated according to the following formula:

$$\begin{aligned} f(x_0+\Delta x, y_0+\Delta y) = f(x_0,y_0) & \\ &+[f(x_1,y_0)-f(x_0,y_0)]\cdot\Delta x \\ &+[f(x_0,y_1)-f(x_0,y_0)]\cdot\Delta y \\ &+[f(x_1,y_1)+f(x_0,y_0)-f(x_0,y_1)-f(x_1,y_0)]\cdot\Delta x\cdot\Delta y \end{aligned} \tag{1.1}$$

, where $0 \le \Delta x \le x_1 - x_0$ and $0 \le \Delta y \le y_1 - y_0$.

7. Check (1.1) for $\Delta x \in \{0,1\}$ and $\Delta y \in \{0,1\}$. Implement a MatLab program that uses the bilinear method to enlarge the image 3 times. It is possible to save some calculations using the formula below (check that (1.1) leads to (1.2) below):

$$\begin{aligned} p &= f(x_0,y_0)+[f(x_1,y_0)-f(x_0,y_0)]\cdot\Delta x \\ q &= f(x_0,y_1)+[f(x_1,y_1)-f(x_0,y_1)]\cdot\Delta x \\ f(x_0+\Delta x, y_0+\Delta y) &= p+(q-p)\cdot\Delta y \end{aligned} \tag{1.2}$$

Checkout the differences between the images calculated by the 2 methods: Nearest neighbor interpolation and Bilinear interpolation (show the image difference).

MatLab ImageProcissing Toolboxen has a function called *imresize().* It can be used to change the size of an image. Try this function out, using the options *bilinear* or *bicubic.*

7. If there is more time, maybe you have you own method for enlargments of images?
   Why is bilinear interpolation useful when you what to rotate an image?