Background

The use of lenses in camera leads to the fact that only objects in certain distance appear sharp in an image. Because of this the settings of the camera have to be changed almost every time when a picture is taken. In practice, the focus distance can be customized by changing the position of the lenses, which was earlier done manually by the photographer. However, today’s technology allows also very efficient automatic focusing, as we can later on observe.

The aim of this exercise is to implement the autofocus feature for a camera using MATLAB. Implementation is done for a web camera controlled by MATLAB Image Acquisition Toolbox http://www.mathworks.se/products/imaq/. This toolbox enables acquisition of images and video from a camera directly to MATLAB as well as changing the camera settings from MATLAB. The camera and the toolbox are installed on the PC of the lab TF304. The lab booking folder is located outside the room TC305. The instructor of this work is Hanna Silen (hanna.silen@tut.fi), room TF317.

Note: This exercise replaces the previous Autofocus exercise for Nokia N900 mobile phone. This is due to some compatibility problems with the lab Windows environment and Qt framework. However, as mentioned at the first lecture, if you have experience on programming for your own mobile device, you can write the implementation for that instead of MATLAB. In this case, contact the instructor beforehand.

Assignment

As said, the purpose of the work is to implement autofocus feature for a camera. The idea is to run the optics of the camera from end to end taking at the same time images with different settings and calculate a sharpness index for all of them. There are several different ways for calculating the sharpness index, here two of them are studied. The higher the index is, the more details the image contains, i.e., the image is better focused.

Start by familiarizing yourself with the MATLAB Image Acquisition Toolbox. An example code can be found at the lab PC, in C:\lightintensity\SGN16006_Ex6.m The MATLAB toolbox documentation is available at: http://www.mathworks.se/products/imaq/.

Our autofocus algorithm is based on taking images with a varying focus value (use e.g. 20 different levels). The optimal focus value is found by maximizing the sharpness index. The algorithm can be described step by step as follows:

1. The focus value range is [0 255]. Set the focus value to its minimum.
2. Take image with the settings you just set and convert it into a grayscale image. Convert the grayscale intensity values into double format.
3. Calculate the sharpness index for the image, now denoted by \( I \). The size of the image is \( N \times M \). Use at least the following two approaches to compute the sharpness index (alternative approaches can be found in the literature):

(a) Gradient-based sharpness index:
\[
f_1 = \sum_{n=1}^{N-1} \sum_{m=1}^{M} |I(n+1,m) - I(n,m)|.
\] (1)

(b) Variance-based sharpness index:
\[
f_2 = \sum_{n=1}^{N} \sum_{m=1}^{M} (I(n,m) - \mu)^2
\] (2)

where \( \mu \) is the mean of the intensity values.

4. Change the focus value and go back to step two. Go through the whole focus range from using at least 20 different, evenly spaced focus distances and calculate sharpness index for all of them.

5. When sharpness indices are calculated for the whole range of focus distances we can solve the optimal focus distance (which probably is not exactly any of the chosen (20 or so) values). For this we use (for simplicity) a second-degree curve \( f(x) = ax^2 + bx + c \) using three focus distances \( x_1, x_2 \) and \( x_3 \). Point \( x_2 \) refers to the point which produced the highest sharpness index, whereas \( x_1 \) and \( x_3 \) represent the points adjacent to \( x_2 \). This obviously gives us the following equations:

\[
\begin{align*}
y_1 &= ax_1^2 + bx_1 + c \\
y_2 &= ax_2^2 + bx_2 + c \\
y_3 &= ax_3^2 + bx_3 + c,
\end{align*}
\] (3)

where \( y_1, y_2 \) and \( y_3 \) refer to the sharpness indices calculated in points \( x_1, x_2 \) and \( x_3 \). Now we have three equations and three unknown variables \( a, b \) and \( c \). But to avoid implementing matrix inversion (or other tricks) in this exercise, we utilize an approximation instead of calculating the exact solution. This is done by defining:

\[
\begin{align*}
x_1 &= -1 \\
x_2 &= 0 \\
x_3 &= 1.
\end{align*}
\]

Substituting these to the equations (3) yields a much simplified result that is easy to implement. After this we need only to calculate the maximum point of that curve, i.e. the root of its derivative \( x_0 \). At this point we must remember that since we defined the points \( x_1, x_2 \) and \( x_3 \) separately from the other points, i.e. gave them values \(-1, 0 \) and \( 1 \), we need now to add \( x_0 \) \( \ast \) step to the focus distance which gave the highest sharpness index value. Finally, take the result image using the calculated optimal focus distance.
Repeat the described algorithm for at least two different targets. Include the sets of optimally/nonoptimally focused images in the report.

**Sending the report**

The report should at least include a description of the work and analysis of the results. MATLAB codes can be included in the report appendix. Write a full scientific report, either by using the document template available at the course webpage or your own template. Send the report in pdf format to hanna.silen@tut.fi by 11.5.2014. If needed, a free PDF software can be found at [http://www.primopdf.com/](http://www.primopdf.com/).