Exercise Digital Image Processing
SS 2008

Exercise 7
Submit by June, 16th, 10:00AM, for exercise on June, 18th

Notes:

- You have a choice: Solutions to text exercises that do not involve programming can be in English or German, at your choice. Submission to text exercises can be made on paper or by email (scanned documents, PDFs, or Word/OpenOffice) to Eva.Hoerster@informatik.uni-augsburg.de before the above due date.
- Solutions to programming exercises must be submitted by email to eva.hoerster@informatik.uni-augsburg.de before the above due date. Only submit you source code (*.h and *.cpp files). Do not submit any executables, binary or object files, project or solution files, nor any other input data that can be downloaded from the course website (i.e. image or video data provided as part of the assignment). DO NOT COMPRESS YOUR SOURCE CODE FILES (.rar, .zip, etc. is not allowed)! Your code must compile and run; if your code fails to compile, you will receive zero points for the exercise.

7.1 (50 points)
Write a C/C++ program with OpenCV that estimates foreground and background in each image of a video. Therefore you need to do the following programming work:

a) Install the Xvid codec (can be downloaded from http://www.xvid.org/Downloads.15.0.html ) and download the example video from the course website.

b) Read the video from your downloaded file (use functions cvCaptureFromFile and cvQueryFrame in highgui of OpenCV).

c) Use the first 100 frames for your video to calculate a model for the background. These frames do not contain any foreground objects. Convert the color images in gray-scale images and model each pixel separately by a Gaussian distribution, i.e. mean and variance. Therefore calculate the mean and variance of each Gaussian distribution from your data (gray-scale values of each pixel in the 100 images).

d) Now use your model to estimate the background and foreground parts in each frame of the remaining video. This is done by comparing the actual value of each pixel in your image to its model. If the pixel value lies within an interval of two times the standard deviation away from your calculated mean it is considered to belong to the background, otherwise it is considered to be foreground. Show for each image of the remaining video an image of the same size, where background pixels are marked black and foreground pixels are marked white.
e) Evaluate your results: Does the estimation of moving foreground/background parts work properly with the simple model? If not, name reasons. How could the model be improved?

7.2 (50 points)
Implement and test lightning correction for images.

a) Collect 10 different images of faces under varying lightning conditions. You may even want to use different light sources to control the shadows.

b) Cut out the faces in each image. Resize the faces so that all images have the same size, e.g. 40 x 40 pixels.

c) Implement the lightning correction in C/C++ using OpenCV. You should implement and use an oval mask such as shown on slide “Example: Lightning Correction” in the lecture before your actual estimation of the best fit linear function. Your code should show the original image, the best fit linear function and the lightning corrected image.
   Hint: An example code regarding the lightning correction has been given in the lecture.

d) Test your implementation with your collected faces and evaluate the results.