Exercise Digital Image Processing

SS 2008

Exercise 1
Submit by April, 28th, 10AM, for exercise on April, 30th

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 your 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.

1.1 (30 points)
Describe the principles of a CCD/CMOS camera. What are the differences between a color and a grayscale camera? Describe the three existing sensor models for color cameras.

1.2 (70 points)
Write a C/C++ program with OpenCV that simulates the output stages of a Bayer filter camera.
Therefore you need to do the following programming work:

a) Implement a Bayer filter. Your program should read a color image of your choice, filter it with the Bayer filter. The image that is obtained is called the Bayer Pattern image. It should be displayed. This image represents the raw output of a Bayer filter camera.

b) After the previous computation step two third of the color data is missing from each pixel. A demosaicing algorithm has to be used to interpolate the complete red, green and blue values for each pixel. Try two different interpolation filter:
   a. The filter implemented in OpenCV (function cvCvtColor).
   b. Implement you own linear interpolation filter, i.e. \( I(x,y) = 0.5[I(x-1,y)+I(x+1,y)] \) for a point to be reconstructed at coordinates \((x,y)\) in the x-direction. A similar derivation follows for the y-direction.
c) Your program should show both resulting reconstructed images in different windows. Compare both reconstructed images and compare the original image with your reconstructed images. Do you observe any differences? If yes which? Where do they result from?