Exercise Digital Image Processing

SS 2008

Exercise 8
Submit by June, 23rd, 10:00AM, for exercise on June, 25th

MULTIMEDIA COMPUTING LAB
PROF. DR. RAINE R LIENHART
DIPL.-ING. EVA HÖRSTER

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.

8.1 (50 points)
In the lecture the algorithm for online learning of arbitrary feature distributions was explained by means of OpenCV source code (See slides “Code: Online Distribution Learning”). Describe in detail the algorithm presented. Clearly state the principles on which the algorithm is based. What are the key components which are needed? Draw one or more figures which explain how the algorithm works. Draw as many figures as you need to explain the algorithm.
8.2 (50 points)
a) Explain in words the task and the problem of the operation of the Hough transform. What problem does it solve? How does it attack the problem?

b) Assume you are supposed to detect unfilled circles as shown in the image below.

Write pseudo C/C++ code that detects circular structures in images by means of using the hough transform.

HINT: Partition your task in at least three functions. Do not try to write one complete function. Use printf() or std::cout to write out the results.

The top level function prototype is given by

```c
void houghCircle (IplImage *src)
```

Fill in the function body and define additional sub-functions. Do not use a different function prototype. Assume that the source image (i.e. src) is a grayscale image. The background is white, while the object pixels have some shade of gray. In order to detect points lying on a circle, you may use the function cvCircle

```c
cvCircle( CvArr* img, CvPoint center, int radius, CvScalar color);
```

to print a circle on a properly initialized image img.