

Computer Science 664
Fall 2002

Assignment 1

Due Date: September 26.

Material: Handout on edge detection, lectures, paper by Wells in PAMI, mean shift paper.

In this assignment you will implement and compare some low-level feature extraction and segmentation algorithms

For this assignment you should submit a winzip or tar archive with: (1) a writeup answering questions in the assignment and explaining whatever code you wrote, (2) your source code, (3) a windows or linux executable.

In the writeup explain how to run each of your programs, showing sample output as required, and explaining whatever theoretical, design, or implementation issues that arose. Also answer questions that are given in the assignment below.

Hand in only listings of your source code, not listings of code that was provided to you such as the the library code.

The test images provided will be in pgm format, and output files should be in pbm format for binary images and pfm format for float images. There are libraries for windows that support manipulation of p*m files (these libraries also do filtering and edge detection, you may not use those routines in this assignment). The course web site will contain versions of the libraries and basic instructions on how to use them.

In all parts of the assignment, attention to efficiency issues, such as memory access order, is important.

1. Implement a program for approximate Gaussian filtering of an image using repeated box-filtering with one-dimensional box functions. This program should take input and output file names on the command line. It should input grey image file and output a float image file.

Your program should run in time independent of the size of the Gaussian (using dynamic programming as discussed in class).

The program should take an additional command line argument of what sigma to use, and should handle values of 1,2 and 4, giving an error for other values.

Be sure to include some example outputs and discuss how you chose the weights for the box filtering in your writeup.

2. Implement a program for computing the Canny gradient magnitude edge detector. This program should take an input float image (output by your box filtering pro-

gram) as input and write an output binary image. The program should take the two threshold levels, hi and lo, as command line arguments.

You should implement the complete Canny method as covered in class, using the non-maximum suppression (NMS) definition of an edge (which is a local maximum of the gradient magnitude, with respect to the gradient direction and its negation). Avoid computing square roots of the gradient in computing the magnitude and the direction. Also implement the two-level thresholding scheme (the “hysteresis mechanism”) which keeps weak edges that are connected to a strong edge (note the definition is recursive).

Be sure to include some example outputs and discuss any issues that came up in going from the description of the algorithm to your implementation.

3. This is a more open ended question. Lowpass filtering an image, as with a Gaussian, has the effect of blurring edges. A different approach is to try finding edges without first lowpass filtering. Simple finite-difference approximations to derivatives, such as used in the Canny edge detector, are not a good way of finding edges unless they have first been lowpass filtered. Other approaches to edge detection are based on finding regions in the image by some means other than lowpass filtering, and then using the boundaries of the resulting regions as edges.

One such approach is called the mean shift method, developed by Peter Meer’s group at Rutgers. This method will be discussed in class. There are also a paper and powerpoint available on the web:

- <http://www.caip.rutgers.edu/riul/research/papers/pdf/spatmsft.pdf>
- <http://www.caip.rutgers.edu/riul/research/slides/MeanShift/index.htm>

You can download a windows program that computes segmentations using this procedure (from <http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html>).

Discuss using the boundaries of mean shift segmentation regions versus using Canny edges, in terms of computation time, the appearance of the resulting edge images, and the relative strengths and weaknesses of the two methods of finding edges. Illustrate with example images where appropriate.

There is no code to write or hand in for this question.