Project 2: Filtering and Edge Detection
ESE 358/CSE 327 Computer Vision, Fall 2000
SUNY at Stony Brook, Dimitris Samaras
Due in two weeks, Thu. 10/26/00

You will be given a program for edge detection based on thresholding the magnitude of the gradient as a template. Use this to complete the following programs.

Gray-scale images with 8 bits/pixel are provided in SUN raster format (lena.ras lena-noise.ras, and mface-gn13.ras, mface-gn8.ras, clown.ras). The images are in the course web-page.

Write a program to do the following.

1. (40 points) Implement the Gaussian smoothing filter. Apply the filter to the 4 input images (`lena.ras, lena-noise.ras`, and `mface-gn13.ras, mface-gn8.ras`), by selecting several different values of sigma (eg. 1, 2, 3 and 4) and the appropriate kernel sizes. Your program should accept as inputs an image and two numbers σ and n. It should output the filter kernel and the filtered image.

2. (10 points) For the two noisy images of the male face compare the results of your experiments regarding noise-removal and loss of detail. Which values of sigma give a better balance for each image? Submit you answer in a README file.

3. (40 points) Implement the Canny edge detector. Use your previous program for the gaussian filtering. Your program should accept a filtered image, the threshold levels required by the algorithm. You should output the horizontal edge image and the vertical edge image, G_x, G_y, the gradient magnitude image, the image after nonmaxima suppression and the thresholded image. Apply your algorithm to images `lena.ras` and `lena-noise.ras`, for $\sigma = 1, 2, 3$

4. (10 points) Show how the selection of sigma and threshold values affect the detail and quality (strength, smoothness) of the recovered edges. In order to demonstrate your algorithm, provide 4 or 5 different edge images using as input `clown.ras`. For each input provide the values used, and how they affected the edge outcome.

5. (20 points, extra credit). Implement the Sobel detector, and compare its performance to the Canny edge detector.

The input-output routines for basic input/output of gray-level Sun raster files `rasio.c, rasio.h` and the sample program of edge detection by using these subroutines and `edge2.c` can be found in the course web page:

`http://www.cs.sunysb.edu/~cse327`

You will need to submit the source code of the functions for gaussian kernel generation, gaussian filtering, Canny edge detection and the intermediate and output images as described earlier. Make sure that the Readme file contains the filename of each image and a short description (up to 2-3 lines) of your observations for the image (if required).

Submission will be done through the computer science transaction lab. The programs will have to compile in a Windows environment.