University of Nebraska-Lincoln

Department of Computer Science and Engineering

CSCE 473/873: Computer Vision
Fall 2020
Assignment 3: Edge Detection

The goal of this assignment is to implement and compare the performance of some common edge detection algorithms.

You must complete this assignment with the partner assigned to you.

Homework

Use the filtering operator to implement the Sobel, Prewitt, Laplacian, and Marr-Hildreth edge operators. Examine their performance using the images provided from the Berkeley Segmentation Database.

- **Sobel**. It should take an image and a threshold value. It generates two images: (a) Edge magnitude image that shows the edges whose strength is greater than the threshold and (b) an edge orientation image.
- **Prewitt**. It should take an image and a threshold value. It generates two images: (a) Edge magnitude image that shows the edges whose strength is greater than the threshold and (b) an edge orientation image.
- **Laplacian**. It should take an image and a threshold value. It generates an edge magnitude image that shows the edges whose strength is greater than the threshold.
- **Marr-Hildreth**. It should take an image and a parameter to represent σ. The size of the filter ($N \times N$) is determined by σ and can be computed as follows:

$$N = \lceil \sigma \times 3 \rceil \times 2 + 1$$

Now implement the Canny edge detector and compare its performance with the other four implemented edge operators. For the Canny edge detection function, you must provide two threshold parameters (L, H) and a parameter to represent σ.

1. Compute the gradient using the derivative of Gaussian (DoG) filter.
2. Use non-maximum suppression to thin the edges. You may choose either 4 or 8 neighborhood for this. Make sure you justify your choice in the report.
3. Begin with the pixels whose magnitudes are above H and extend to include the neighboring pixels with magnitudes above L using hysteresis thresholding.
4. Create a new image that displays these edge-linked chains.

Compare your results with the corresponding edge operators in Matlab.

Potentially useful MATLAB functions: fspecial()and the operators in the **MATLAB tutorial** which make it efficient to cut out image subwindows and do the convolution (dot product) between them. padarray().
Forbidden functions you can use for testing, but not in your final code: edge().

Report

Summarize your work in a report. The report must be typewritten, formatted, and must be divided into sections corresponding to each task described above. In particular, you will describe your algorithm and any decisions you made to write your algorithm in a particular way. Then you will show and discuss the results of your algorithm. You must show the results of your edge detectors. Also, discuss anything extra you did. Feel free to add any other information you feel is relevant. **The report must be in pdf format and be no more than 4 pages.**

Handin

Put all your files in a folder called HW3 and submit the folder on canvas. This is very important, as you will lose points if you do not follow instructions. Every instruction you don't follow, you will lose 5 points. The folder you hand in must contain the following:

- README --- text file containing anything about the project that you want to tell the TA
- code/ --- directory containing **all** your code for this assignment
- report/ --- directory containing the report in pdf format
- images/ --- directory containing the images you used and the results.

Due Date

The assignment is due on October 1 and is worth 75 points.

Rubric

- +40 pts: Working implementation of the first four edge operators
- +20 pts: Canny edge detector
- +15 pts: Report (must have several examples)
- $-5 \times n$ pts: Lose 5 points for every time you do not follow the instructions for the assignment