Project 2

Image Processing

Division of Electronics and Information Engineering

September 29, 2020

Purpose: To practise edge detections

Specifications:

1. Write edge detection routines for Sobel Edge, Prewitt Edge and Canny Edge operators.

1) Apply Sobel Kernel $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ for Sobel Edges with a reasonable threshold.

2) Apply Prewitt Kernel $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ for Prewitt Edges with a resonable threshold.

3) Canny Edge Detector requires longer multiple steps discussed in class.

a) Create a 1D Gaussian mask G to convolve with your image (I). The standard deviations(s) of this Gaussian is a parameter to the edge detector (call it σ>0).

b) Create a 1D mask for the first derivative of the Gaussian in the x and y directions; call these G_x and G_y. The same σ>0 value is used as in step a)

c) Convolve your image I with G along the rows to give the x component image (I_x) and down the columns to give the y component image (I_y).

d) Convolve I_x with Gx to give I'_x, the x component of I convolve with the derivative of the Gaussian, and convolve I_y with G_y to give I'_y, y component of I convolved with the derivative of the Gaussian.

e) Compute the magnitude of the edge response by combining the x and y components. The magnitude of the result can be computed at each pixel (x, y) as: $M(x,y) = \sqrt{I_x^{'}(x,y)^2 + I_y^{'}(x,y)^2}$.

f) Implement non-maximum suppression algorithm (the center pixel must have a larger gradient magnitude than its neighbors in the gradient direction) that we discussed in the lecture. Pixles that are not local maxima should

be removed with this method. In other words, not all the pixels indicating strong magnitude are edges in fact. We need to remove false-positive edge locations from the image.

g) Finally apply Hysteresis thresholding to obtain the final edge-map.

2. You need to generate intermediate images for each step to check how your algorithm works.
3. Try images uploaded to class page and your favorite images. For more images, refer https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html. You may calculate precision and recall for Berkeley data.
4. Report your experiments with analytic document.

Discussions:

1. Analyze result images from three different implementation
2. What are effects of σ to images?
3. Does Canny Edge Detector achieve the three goals?

Hand in : Turn in your completed document to eiprof@naver.com. Your document include ①Problem description, ②Source codes with full of comments, ③Results (screen capture), ④Analysis report, and ⑤Others (such as references) in order. The email title should be 'IP Your_Name HakBeon(student id number) Proj_number'

Due date : October 6, 2020