State University of New York at Buffalo

CSE 473/573 Fall 2016 Homework Set #4

Date: Wednesday November 2, 2016; Due: ***<u>Wednesday November 16, 2016 at the start of Class</u>***

Name: ______________________________ Student Number: ________________________

Notice: Python programming for the following problems is preferred. However, solutions obtained via another programming language is also acceptable.

Problem (1) **(Edge Detection by Zero-crossing, DoG and LoG – Python Programming)** 50%

Zero-crossing of the second derivative has been considered as strong indications for edges in a given image. For practical implementations, robust computation of second derivative is usually preceded by the smoothing of an image. Two popular approaches to zero-crossing-based edge detection are Difference-of-Gaussian (DoG) and Laplacian-of-Gaussian (LoG). For the "UB Campus" test image, perform the following with Python programming:

(a) Obtain and display the DoG image by applying the following DoG mask to the test image

$$\begin{bmatrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ -1 & -3 & 5 & 16 & 5 & -3 & -1 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \end{bmatrix}$$

(b) Compute and display the zero-crossing of the DoG image obtained in (a)

(c) Compute and display the zero-crossing strong edges by removing weak edges that do not have first derivative support in (b)

(d) Compute and display the LoG zero-crossing edges by applying the following LoG mask to the test image

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

(e) Compare the results in (c) and (d) and explain why the edges obtained in (c) and (d) are different. Is there any way that we may obtain the same results? Please explain in detail.

Problem (2) **(Region Merging Segmentation – Python Programming)** 50%

Region merging is an effective scheme for region growing based segmentation. Region growing may begin with each pixel within an image in which a pixel represents a single region initially. Regions will be merged to satisfy the region segmentation conditions as defined by Equations (6.30) and (6.31). Perform the following algorithm to the "Mixed Vegetables" test image through Python programming:

(a) Perform Algorithm 6.18 – Region merging via boundary melting and display the results of the final region merging. Each student is required to set their own thresholds and also explain why such thresholds are selected.

Reference:

Algorithm 6.18: Region merging via boundary melting

1. Define a starting image segmentation into regions of constant gray-level. Construct a supergrid edge data structure in which to store the crack edge information.
2. Remove all weak crack edges from the edge data structure (using equation (6.32) and threshold T_1).
3. Recursively remove common boundaries of adjacent regions R_i, R_j, if

$$\frac{W}{\min(l_i, l_j)} \geq T_2 ,$$

   where W is the number of weak edges on the common boundary, l_i, l_j are the perimeter lengths of regions R_i, R_j, and T_2 is another preset threshold.
4. Recursively remove common boundaries of adjacent regions R_i, R_j if

$$\frac{W}{l} \geq T_3 \tag{6.33}$$

   or, using a weaker criterion [Ballard and Brown, 1982]

$$W \geq T_3 , \tag{6.34}$$

   where l is the length of the common boundary and T_3 is a third threshold.