COMP/BMME 576, Spring 2018 Pizer Assignment 3, in **new** teams of 2 (pick a different partner than in the previous assignment).

Laplacian zero crossings

Friday 16 February 2018 Due Thursday 1 March 2018 at class time

1) Find online a jpg image of something with some object edges with reasonable contrast. Read that image via your MATLAB program. Select a 128 × 128 square subimage I(j,k). This subimage and a variant on the 128 × 128 diagonal edge image you produced in the derivative assignment will be the two images on which you operate in this assignment. Output both of these images into the grading program.

   The variant of the diagonal edge image you produced should be different in that on the diagonal the value should be 500, whereas on side of the diagonal pixels have a value of 1000 and on the other side it should have a value of 0. Thus the edge in this image ideally will be precisely on the diagonal, which is different from what you had formerly, where the edge should be offset from the diagonal by 0.5 pixel.

The AmpPhaseDFT and ReconfromAmpPhase MATLAB functions provided on SAKAI with assignment 3 will be used in this assignment. The amplitude and phase arrays produced by AmpPhaseDFT and those used as input to ReconfromAmpPhase are each 65 × 128 arrays. In these arrays the j,kth element corresponds to frequency (j-1)/128 in the vertical direction. The horizontal direction is more complicated. For k between 1 and 65 inclusive, the frequency is (k-1)/128. For k between 66 and 128 inclusive, the frequency is the negative number (k-129)/128.

2) The objective of this assignment is to see how the Laplacian of Gaussian operator produces zero crossings at edges in an image and to see how the width (standard deviation σ) of the Gaussian affects the edges that are found. In this part you must compute the value of the filter at each discrete frequency that appears in the amplitude and phase outputs of AmpPhaseDFT. Write down the formula for the filter value for index j,k and for Gaussian width σ; use this formula while ignoring the minus sign and any multipliers constant in frequency (e.g., the 2σ multiplier in the expression for the FT of the Laplacian), and either use the exponential formula for the FT of the Gaussian or, equivalently, evaluate the Gaussian in the FT of the Gaussian and ignore the $2\pi\sigma^2$ multiplier in front of the Gaussian in the FT of the Gaussian and the constant multiplying the exponential in the formula for the Gaussian. Compute this filter array in j,k. Display and turn in the part of this array as a 65 × 65 image with j and k running from 1 to 65.

3) For each of your two input images and for two values of σ discussed below, apply AmpPhaseDFT to your input image, and apply your filter on the magnitude output of the result. Output to the grader the matlab line where you evaluate the filter. You will have produced the magnitude output of the frequency domain result, namely the Laplacian of Gaussian applied to your input. Because the kernel is real and symmetric in both x and y, its application in the frequency domain does not change the phase. The σ values you should use, in two different trials, are 2 pixels (σ =2) and 5 pixels (σ =5).

4) For each of the four results in the frequency domain (two values of $\sigma \times$ two input images) apply ReconfromAmpPhase to the pair of arrays whose magnitude is the frequency domain result you computed and whose phase is the same as that produced by AmpPhaseDFT. Display and submit these four images in the following order: (image 1, σ =2) result, (image 1, σ =5) result, (image 2, σ =2) result, (image 2, σ =5) result.

5) For each of the four images from part 4 you want to display an image showing those pixels that correspond to a zero crossing. To do that, for each pixel in that image first test if that pixel's intensity is adequately far away from 0.0 (I tested whether its magnitude was less than 10E-7). If not, set the zero crossing result image pixel's value to (or leave it at) zero. For the remaining pixels, you should test whether it has any (horizontal, vertical, or diagonal) neighbor pixels with intensity with opposite sign (not zero) from that of the pixel in question. If it has such neighbors set the intensity to 1000 in the zero-crossing image, and if it does not, set the intensity to 0 in the zero crossing image. Display and turn in the four zero-crossing images in the same order as for part 4.

6) Consider what you have seen as to how well do the zero crossings characterize the edges in the original image. In particular, submit an answer to each of the following questions in the form "Y" or N" for yes or no. A) Are there breaks in the edge? B) Are the edges displaced from where they should be? C) Do these properties change with the value of σ? Try to explain to yourself any of the behavior you have seen – but there is nothing to be passed in with regard to this explanation.