CAP5416: Programming Assignment #2, Due Date:Nov. 14th 2016

You should turn in well documented MATLAB code for the following problems as well as a report containing a discussion of your findings and display of your results. Code without documentation will NOT be graded. Programs and the report needs to be turned in by midnight on the due date, through Canvas.

1. Implement the curvature flow equation for a closed planar curve. The curavature flow equation is given by, $\frac{\partial C(s,t)}{\partial t} = \kappa \mathbf{N}$ where κ is the curvature of the curve $C(s,t)$ and $\mathbf{N}$ is the curve normal. The above equation needs to be discretized using the finite difference method.

   Your program should accept as input, a curve represented by an ordered set of pairs (x_i, y_i), use a large number of points, say 2,000. At each iteration, compute the displacement of each point according to the flow equation. Check your code for different time step sizes. Use three different curves as the input and at least three different time step sizes. You will use any existing simple MATLAB GUI to visualize the curve evolution on your screen. Submit the plots of the curves created by the curvature flow at the initialization, two intermediate stages and the final stage of the evolution process.

2. Develop code (using the derivations of the flow shown in class) to evolve a contour that can align itself to a smooth gradient vector field in the image. You should add a curvature flow term to it so that the resulting contour remains smooth (this was not done in the class notes). Thus, there will be two terms in the curve evolution, (i) the curvature term, (ii) the edge alignment term. For the edge alignment term, you will,

   (a) write a procedure that will first smooth the image gradient field exactly as in the GVF paper.

   (b) Take the above smoothed vector field as the input vector field in the edge alignment term.

   Test your curve evolution code on, (1) the "pacman" image and (2) your favorite image with just a single object in it. In both cases, make sure that you do not initialize the contour too far from the desired edges to which you want to align the contour to. Your report should contain the equations you used in your algorithm for evolving the curve and to achieve the smoothing of the vector field, their discretization and the final equation used in your implementation and any other implementation issues.

   (c) Compare your results with the GVF segmentation solution by downloading the GVF code from the web. This is a visual comparison. Hence, you will present snap shots of the contour initialization, an intermediate stage of the evolution of the contour and the final solution. There should be three snapshots depicting various stages of evolution. One which is in the first one third of the evolution process and the second roughly in the middle of the evolution process and then the final converged solution. You will show this for both the methods in order to facilitate the visual comparison.

Implementation of the code for all the problems MUST be original to you excepting the last problem, where you can download the GVF code use it as is. Originality will be checked using tools such as "turnitin" . You may use any GUI-builder you wish, to display the evolution of the contour and display your results. You should turn in a report that has figures depicting the results from each problem and *a succinct discussion* of the results. Your report should contain, snap shots of the initialized active contour, two intermediate stages of the contour and then the final achieved result. Each of these active contour snap shots should be superposed on the original image depicting the position of the contour in the image. Use a bright color (say red) to show the active contour. Interpolate the active contour points to display a smooth contour in all the problems.