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 curvature flow equation is given by, $\frac{\partial C(s,t)}{\partial t} = \kappa N$ where $\kappa$ is the curvature of the curve $C(s,t)$ and $N$ 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 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 (when it reaches the iteration limit for instance).

2. A popular method to represent a textured region is via the use of what is known as a structure tensor descriptor. Given an image, $I(x, y)$, the structure tensor at each pixel is a $(2, 2)$ matrix formed by, $W(x, y) * \nabla I(x, y) \nabla I(x, y)^T$, where $W(x, y)$ is a known Kernel for e.g., a Gaussian Kernel. The structure tensor captures neighborhood information at each pixel for instance, the dominant directional characteristics within the neighborhood. Once you have computed the structure tensor at each pixel, you then have a tensor field. Our goal then is to segment this tensor field using the Chan-Vese active contour model given by the following variational principle.

$$E(C, T_1, T_2) = \int_{R} d^2(T(x), T_1) dx + \int_{R^c} d^2(T(x), T_2) dx + \alpha |C|$$  \hspace{1cm} (1)

Where the curve $C$ is the boundary of the desired unknown segmentation, $R$ is the region enclosed by $C$ and $R^c$ is the region outside $C$, $T_1$ and $T_2$ are the mean values of the tensor field in region $R$ and $R^c$ respectively, $|C|$ is the arclength of the curve $C$, $\alpha$ is a regularization parameter, $d(., .)$ is the measure of distance between two structure tensors. The minimizer of the above variational principle is obtained using standard EL-equations. The corresponding active contour flow equation is given by,

$$\frac{\partial C}{\partial t} = - \left[ \alpha k - d^2(T, T_1(t)) + d^2(T, T_2(t)) \right] N$$

where $T_1 = M(T, R)$, $T_2 = M(T, R^c)$  \hspace{1cm} (2)

The curve evolution equation (2) can be easily implemented in a level set framework. The corresponding level set formulation is given by:

$$\frac{\partial \phi}{\partial t} = \left[ \alpha \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - d^2(T, T_1) + d^2(T, T_2) \right] |\nabla \phi|$$  \hspace{1cm} (3)

The above model can be viewed as a modification of the active contour model without edges for scalar valued images by Chan and Vese described in class. For this assignment, you will develop code for the level-set form given above for two distance functions $d(., .)$ used in the level-set form above. (i) The Euclidean distance between the vectorized forms of the structure tensor matrices and, (ii) the square root of the J-divergence between the two normal distributions whose covariance matrices are the two given structure tensors. This “distance” is expressed mathematically as,

$$d(T_1, T_2) = \sqrt{\frac{1}{2} tr(T_1^{-1}T_2 + T_2^{-1}T_1) - n}$$  \hspace{1cm} (4)

where $tr(\cdot)$ is the matrix trace operator, $n$ is the size of the square matrix $T_1$ and $T_2$. Using this distance, the mean value of a tensor field contained in a given region $R$ is obtained by the following minimization problem:

$$M(T, R) = \min_{M \in SPD(n)} \int_{R} d^2[M, T(x)] dx$$  \hspace{1cm} (5)
and is given by,

\[ M = \sqrt{B^{-1}} \left[ \sqrt{\sqrt{BA} \sqrt{B}} \right] \sqrt{B^{-1}} \]  

(6)

where \( A = \int_R T(x) dx \), \( B = \int_R T^{-1}(x) dx \) and \( SPD(n) \) denotes the set of symmetric positive definite matrices of size \( n \) (the structure tensors in our case).

For this question, you will use the test image provided to you on the class website to test your code on. **Note that you are allowed to use any existing Chan-Vesse implementations on the web as your base for this assignment but, you must modify it to use the distance functions specified above in order to make it work for this problem.** For implementation of the level-set form of Chan-Vesse model, you can use the finite difference method to discretize the partial differential equation given above. A gentle introduction to the finite difference implementation of the level-set method is explained in this note on the web, [https://profs.etsmtl.ca/hlombaert/levelset/](https://profs.etsmtl.ca/hlombaert/levelset/).

3. Write a program to implement the K-means clustering algorithm to segment the texture images in problem (2). You will use the same two distance functions specified in problem (2). Then compare and contrast the results of segmentation you achieve using the K-means clustering and the Chan-Vesse method in problem (2). You will write a succinct report to this end.

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.