BMI/CS 567 Medical Image Analysis
University of Wisconsin-Madison
Assignment #4

Instructor: Jeanette Mumford
Due April 23, 2019 by 2:30PM

(1) (6 points) Submit this problem as a .mlx file. Medium speedup for mean shift segmentation. In this problem you will use the `T1.mat` and `T2.mat` images that were distributed in the matlab code from the mean shift lecture. You will be editing the class mean shift code to create a faster algorithm. The speedup is often referred to as "Mean shift vector reutilization".

Mean shift requires taking each pixel's feature values and estimating the associated mode using an iterative algorithm. Working one pixel at a time is time intensive and the idea of this algorithm is to pick up other pixels along the way. The logic is that if another pixel is very close to the path the current pixel is moving through, it will likely end up at the same mode. Therefore, for the speedup, all points that are close to the path the current point of interest is traveling, get assigned to the same mode, as shown in the right panel of Figure 1. Recall that for mean shift you start by choosing a bandwidth, h. If x_0, a vector of length 2 for this example, is where you currently are, then the next step takes you to $x_1 = \frac{1}{n_x}\sum_{x_i \in S_h(x_0)} x_i$, the average feature within the circle of radius h centered at x_0. For the speedup, do the following

1. When identifying the pixels within the circle of radius h, also identify any points within the smaller circle of radius ϵh. ϵ will be between 0 and 1. The specific values to be used in this exercise are given below. Note, these points in the smaller circle do not impact or change the shift. You are simply recording that it landed close to the path of the pixel of interest.

2. Collect all points that fall within the smaller, ϵh radius at *any* iteration and when the algorithm converges, you simply assign all of these pixels, along with the original to the same mode. *These points will not go through the iterative algorithm and you will need to figure out how to code this.*

Code the Mean shift vector reutilization algorithm. Use a radius of $h = 1$, as we did in class, and try two settings for ϵ and compare the results of these two segmentations to the one obtained with standard mean shift (code from class). Specifically, include the multi-paneled plots illustrating your segmentation results, describe whether or not the segmentation results match and if not, why? Is the speedup worthwhile? Your code *must* skip over points that have already been assigned a mode because they fell into the path of a pixel that was processed earlier. You will be graded upon whether you implemented the algorithm properly and supplied all of images requested and answered the questions clearly.

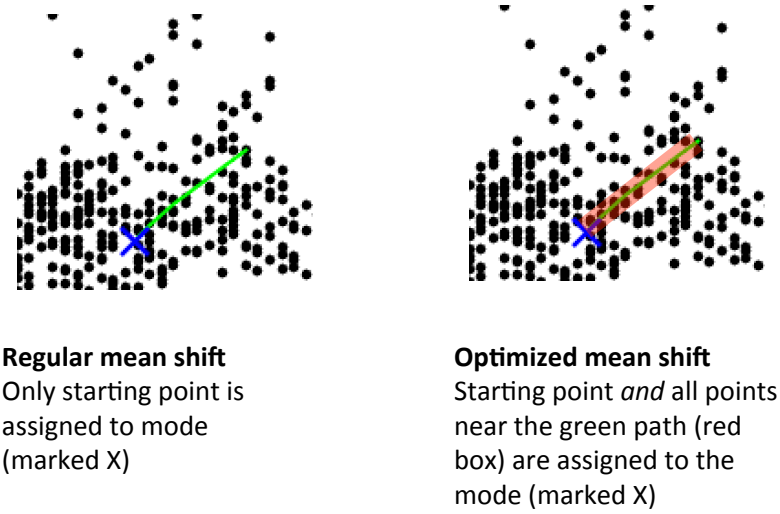

Figure 1: Regular mean shift (left) and mean shift vector reutilization (right).

(2) (10 points) Submit this problem as a separate .m file. The goal of this problem is to check that you're making progress on the final project and that you're able to write code that I can easily execute. Generate code that extracts 1 feature from each of the distributed retina images. Recall, this is a single value that summarizes something about each image. Write this code so it will estimate these features from the left out data set as well as the distributed data set. Please see the Final Project instructions for more details. In short, I should be able to change 2 paths in your code to the 2 sets of data (one I distributed and held out data). I will then run your code directly from the command line (akin to using the `run` command). Some of you may have written extra functions, so it is fine if I also need to change an `addpath` command. Just zip up all your MATLAB functions. This portion of the homework will *not* use the mlx/pdf option. You will submit MATLAB script(s) only. You will be graded on:

- Whether your code executes without errors.
- How well you describe what you're doing and your motivation for doing so in comments
- That you only use concepts that we have learned in this class. You cannot use ideas from other classes. For example, we have not covered texture analysis/Gabor filters, so they cannot be used on the final.
- Whether your code is efficient. It should be able to compute the features on all images within a minute or two, tops.
- The only output from the code should be any notes you include to verify the code is running and a scatter plot of the feature magnitude on the y-axis and the image number on the x-axis (one for the distributed data and the held out data). For the distributed data I ordered all of the healthy retinas first, so this will be an easy check to see if your feature has any potential. You should be able to see a noticeable difference in the feature magnitude for the first 18 vs second 18 values.