CS 675 – Computer Vision – Spring 2018

Instructor: Marc Pomplun

Assignment #4

Posted on April 19 – due by April 26, 2pm
Question 1 due by May 2, 9pm to enter contest

Question 1: Texture-Based Image Segmentation

Your task is to write a program that segments an image into several regions based on the local texture. Fortunately, there is a very nice website that will provide you with an unlimited amount of sample and test images:

http://mosaic.utia.cas.cz/

You should use the "grayscale" option; we will not consider color here. The images you get will be randomly "stitched together" from several textures, and the image segmentation task for your program is to find out where the different pieces are located. You will also receive corresponding images that show where the boundaries between the different textures are located.

The basic principle that you should use is the following: First, derive local feature vectors that describe the texture at given coordinates (x, y). These features could be obtained through Law's filters, co-occurrence matrices, any other techniques that you know or develop, or combinations of them. I will provide code for the methods we discussed in class. The resulting data could be stored in an array of tables, or you could introduce a new data structure to hold the data.

Whichever techniques you choose, the basic idea is that you should get a high-dimensional vector for each location in the image. More similar vectors should indicate more similar textures, i.e., local areas that are more likely to belong to the same texture. Ideally, for each of the textures in the image, the corresponding image locations should form a cluster in this high-dimensional space. Therefore, your next step is to apply some form of cluster analysis on these vectors in order to determine groups of similar vectors. This may work better if you compute average vectors for small neighborhoods (maybe 8×8 or 16×16 pixels) instead of individual pixel locations. Furthermore, you could also include the spatial coordinates in the feature vectors, because neighboring pieces of the image are more likely to belong to the same texture than are distant pieces (but it may be difficult to decide how to weight this information relative to the other features). You are allowed to "tell" your algorithm how many different groups (segments) you would like to generate.

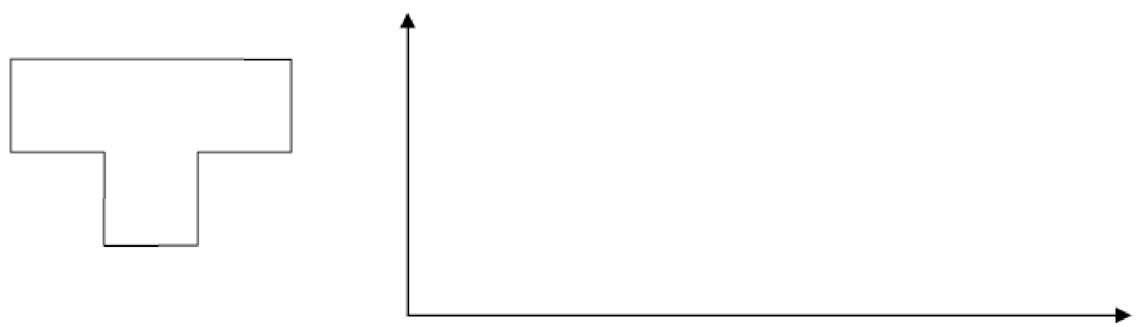
Finally, in the input image, mark the areas belonging to different groups in different colors, for example, by setting either the R, G, or B values (or combinations of them) of all relevant pixels to 0 or 255. Alternatively, you could simply overwrite the original gray values with solid color. Let your program write the result to the hard drive, and provide five sample segmentations. Your code should include a function segmentTexture for generating the segmented images with the following signature:

Image segmentTexture(Image inputImg, int segments)

I will run your program on a new set of 4 texture images containing 3, 4, 5, and 6 textures, and present the results (anonymously) in class. Then you will vote for the best-performing segmentation algorithm. The winner will, in good tradition, get a BestBuy gift certificate for $50, some extra course points, and the honor to explain to everyone the winning algorithm.

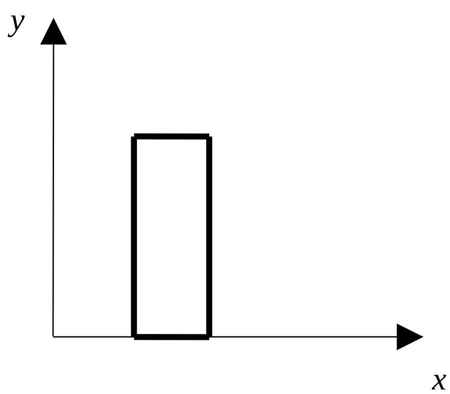
Question 2: Signature, Please!

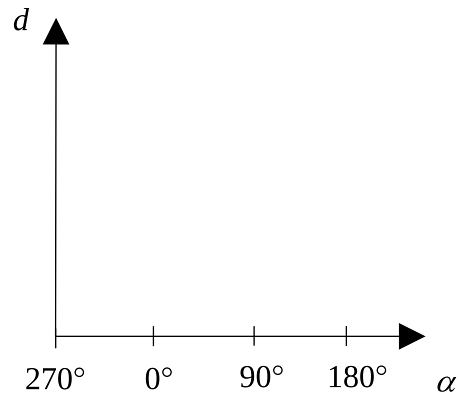
Use the diagram on the right to sketch the signature of the contour on the left. It does not have to be perfectly scaled.



Question 3: A Rectangle in Hough Space

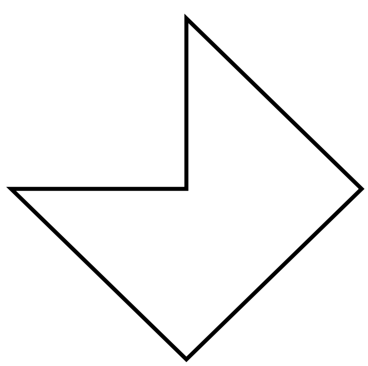
Let us perform a Hough transform for the detection of straight lines as we discussed in class. Therefore, we are transforming the *x*-*y* image space into an α-*d* space, representing the orientation and length of the normal. If our input image is the perfect rectangle with horizontal and vertical edges shown in the left figure, approximately what pattern of maxima would we expect in the transformed space (right figure)? Please describe how you computed your result.





Question 4: Slope Density Functions

(a) Draw the slope density function of the following contour:



(b) Describe how this function changes if you rotate the contour clockwise by 90 degrees. You can describe it in words or draw another diagram.