Question 1: Classifying Scenes

Texture-Based Image Segmentation: Your task is to write a program that segments an image into several regions based on the local texture. Fortunately, you can start by sample images in a folder called "textures" to test your algorithms before you use any other image if you prefer. There is a very nice website that will provide you with an unlimited amount of sample and test images: http://mosaic.utia.cas.cz/

Try to use the "grayscale" option first on sample images; then we move to use more color when you apply the code on the images. The sample 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. 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 8x8 or 16x16 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) Do not forget to save the images report the results

The top three results will each receive a $30 Amazon gift card each.