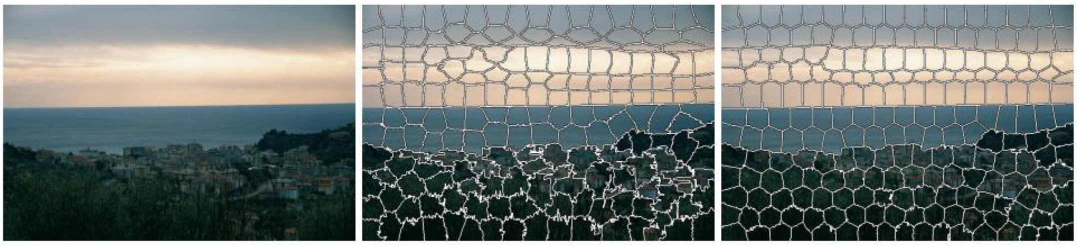
1. (PCA) There are n p-dimensional data points and we can stack them into a data matrix: $X = \{x_i\}_{i=1}^{n}, x_i \in R^{p\times 1}$, where $X \in R^{p\times n}$. The covariance matrix of X is $C = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$, where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ (actually, it is the mean of all the data points). Based on the discussion in class, we know that if α_1 is the eigen-vector associated with the largest eigen-value of the covariance matrix C, the data projections along α_1 will have the largest variance. Please prove that the eigen vector α_2 associated with the second largest eigen-value of C has the following two properties: (1) it is orthogonal to α_1; and (2) among all the orientations orthogonal to α_1, the variance of the data projects along α_2 is the largest one.

2. Segmentation is one important problem in the field of computer vision. A common problem in segmentation algorithms is that it is usually hard to find a fixed set of parameters that work well for all images in a data set. The same set of parameters may provide an oversegmentation for some images while they may cause undersegmentation for the others. Next, we will try to obtain a good segmentation by, first, oversegmenting theimages, and then, combining the regions that

have common color and texture characteristics using clustering. We will compare this segmentation to a pixel-based clustering approach.

(1) The first step is to obtain an oversegmentation for each image. Use the code at http://ivrl.epfl.ch/research/superpixels to obtain superpixels for each image in our data set. There are two versions (SLIC and SLIC0). You can use either version.



(a) Input image (b) Output of SLIC (c) Output of SLIC0

Figure 1 (a) original input image and the results of two different versions of SLIC.

(2) Compute Gabor texture features for each image. You can use the code at http://www.peterkovesi.com/matlabfns/PhaseCongruency/gaborconvolve.m. (Use Matlab's rgb2gray function to convert the RGB image to grayscale)

(3) Cluster the superpixels to get larger homogeneous regions. First, for each region (using only the pixels in that region), compute:

A. a color feature vector containing the average of the color features;

B. a texture feature vector containing the average of the Gabor texture features (e.g., (1) 3 values (average R, average G, average B); (2) 16 values for 4 scales and 4 orientations));

C. a color-texture feature vector that contains the color feature vector and the texture feature vector appended together. Note that you may have to normalize each feature component to the [0,1].

Now, each superpixel is represented as a point in the feature space. You can use any clustering algorithm to group the superpixels into regions. The output at this step is a matrix that contains a new integer label for each pixel where each label indicates the cluster id of that pixel’s superpixel. You have to perform this step separately using the feature vectors defined in **A**, **B**, and **C** above.

(4) The fourth step is to perform clustering using color information for each individual pixel. Instead of the feature vectors computed from superpixels, apply the clustering algorithm used in Part 3 to individual pixels. The goal is to compare these clustering results with the clustering obtained from superpixels.

Submit:

1. Description of the parameters used for superpixel segmentation and Gabor texture feature extraction.
2. Gabor texture feature examples (output of the gaborconvolve function) for several scales and orientations for at least five different images.
3. Clustering/segmentation results in Part 3 and Part 4. You must show the obtained regions by overlaying segment boundaries on the image as in Figure 1.
4. Citation for any external code used
5. Detailed discussion of the results with respect to different features, different superpixel extraction parameters, different clustering parameters, and clustering using superpixel based and pixel-based information.