CS 216/295: Image Understanding

Fall 2008

Homework 2 Due at beginning of class, 10/30

Review: your class notes and Forsyth and Ponce Chapter 9
Suggested reading: Forsyth and Ponce 4 (radiometry), 6.1-6.4 (color)
Submit your writeups for the following problems electronically via the EEE website "hwk2" dropbox. If you are not registered for the class, you can send me the assignment via email.

1. Color quantization: Although k-means clustering on color alone doesn't provide very satisfying segmentations, it can be used to reduce the color palette of an image. Write a script which takes a color image and a value for k and returns a new version of the image which uses only k distinct colors. Your code should cluster the pixel values using k-means and then replace each pixel with the mean color of the cluster to which it belongs. Demonstrate your code on a (colorful) image for different values of $k = 2, 5, 10$ What happens if you scale one of the feature coordinates, say the R value by a factor of 10?

2. Color spaces: Images are typically represented in terms of RGB values. However, one can represent color using other coordinate systems which are non-linear functions of the RGB values. One such space is HSV. Modify your algorithm from the previous problem to quantize the image in HSV coordinates rather than RGB coordinates (you'll find the MATLAB functions rgb2hsv and hsv2rgb handy.) Compare the quantized images with those quantized in RGB with the same setting for k. Which looks perceptually closer to the original image? Why does the choice of color space matter here? How could one go about determining an "optimal" color space to perform image quantization in? (Bonus: figure out the conversion to CIE LAB space and repeat the experiment there. Make sure and scale the coordinates in a reasonable manner. Is there further improvement?)

3. Filterbank: In order to analyze texture, we would like to describe the image in terms of distribution of filter outputs. Implement a filter bank function which takes an image as input and filters the image with Gaussian derivatives in the horizontal and vertical directions at 3 different scales $\sigma = 1, 2, 4$. Also create an additional center surround

filter by taking the different of two isotropic Gaussian functions at two different scales, i.e. $G_2(x,y) - G_1(x,y)$ and $G_4(x,y) - G_2(x,y)$. Feel free to use the various tricks we've discussed for making this process fast (e.g. separability). Your filterbank function should take as input one grayscale image and return 8 filter response images. Submit your code and an image which shows your 8 filter kernels.

4. Filter Distributions: Now convince yourself that the distribution of filter outputs really captures something about the texture. Use the image:

   `http://www.ics.uci.edu/~fowlkes/class/cs216/hwk/zebra_small.jpg`

   Select a image patch (say 40x40 pixels) over the zebra's neck and compute the mean absolute response of each of the 8 filter responses in that region [1]. Compute the mean absolute responses of the filterbank channels for similar regions in the tree leaves above the zebra's back and also on the grass in front of the zebra. Print out the three mean response vectors for these regions and explain the differences you see in terms of your filters.

5. Textons: A nice way to represent the joint distribution of filter responses is to quantize them. Use your same k-means procedure from the first problem to quantize your filter bank output (now your feature vectors will be of dimension 8 instead of 3) using $k = 20$. Rather than output the cluster centers, create an image where each pixel is assigned its cluster label $1 \dots 20$. Submit the resulting texton image (use `colormap jet`).

6. Segmentation: Now that we have a good way to describe texture, lets try to segment the zebra image. For each pixel in the image, compute a histogram of the texton labels which appear in a 5x5 neighborhood around the patch. This will give a feature vector of length 20 associated with each pixel since there are 20 different types of textons. Now run k-means on these resulting feature vectors and display the resulting segmentation. We expect that pixels which all lie on the zebra's neck will have similar texton histograms and hence will end up in the same cluster. Compare the results of your clustering to simply doing

[1] If you don't take an absolute value then the positive and negative components will tend to cancel out and give you mean 0

k-means on the color values (as in problem 1). Submit your MATLAB code and your favorite segmentation result.

7. Segmentation Experiments:

   Please do at least one of the following additional experiments:

   (a) Segment using both texture and color features. You will need to figure out how to scale the two relative to each other so that k-means does something reasonable. Also consider adding in the x,y coordinates of the pixel as additional features (again appropriately scaled).
   (b) Experiment with different numbers of textons, filterbank scales, window sizes and number of clusters in the final algorithm. What gives you the best results (comes closest to segmenting out the zebra) ?
   (c) Rather than using the filterbank, try quantizing image patches. For each pixel, build a feature vector which is the intensity values of the 8 neighboring pixels relative to the center pixel. Vector quantize these and visualize the texton map. Can you get as good a segmentation as with the filterbank?
   (d) Replace the final clustering of texton histograms with your favorite clustering algorithm. Can you come up with a more principled approach for clustering histograms, e.g. use the L1 distance or something fancier?

8. Project Proposal:

   Put together a short description of a proposed project for the class. I will post a set of ideas and papers online but feel free to suggest something else which interests you.