Computer Vision CS543/ECE549
Homework 2

Due Date: 10 March 2011

General instructions: Do not use built-in code or code from the Internet for the Hough transform, K-means, or mixture of Gaussians. Explain your algorithms in paragraph form, with equations where they are helpful.

1 Circle Detection using Hough Transform (30%)

For the given car image, compute gradients and detect circles using the Hough transform. Submit an image showing detected circles overlayed on the top of original image and a pseudo code for the algorithm. Explain your algorithm in words and equations (not code), including the parameterization of your Hough space, how to generate votes, the grid resolution in Hough space, how to compute local maxima, etc.

Useful hints:

- Ignore very small circles because they will be unduly influenced by a few pixels.
- It will make life easier to use a small bin size (e.g., 1 pixel for the radius and center position).
- Notice that, with a larger radius, more pixels have the opportunity to vote for it. You may want to account for that by weighting the votes.
- It's helpful to weight votes according to edge orientation.
- Don't worry if you have several extra circles or miss a couple that you'd expect to get.

2 Texture Matching using a Bag-of-Words model (30%)

In this problem, you will generate a bag-of-words representation of images to encode the overall texture properties of each image. This works by using k-means to group filter responses from each pixel into a discrete set of clusters and using the image level statistics of the clustering to describe the overall texture.

By replacing the filterbank responses with other local descriptors such as SIFT, this representation can be used in a wide variety of applications including object recognition and image retrieval. It should be noted that the bag-of-*words* representation originated in NLP to represent documents by the occurrences of characteristic words.

a) **Codebook generation:** For each image in the "train" directory, compute the vector $\mathbf{x}_i \in \mathbb{R}^{48}$ of the 48 Leung-Malik filterbank responses for each pixel i (5 pts). Implement k-means to cluster the responses from a subset of pixels (as many as time/memory permits) from the training images $\{\mathbf{x}\}$, cluster the descriptors into k clusters (12 pts).

Repeat with three random initializations, and use the trial that gives the lowest average euclidean distance to the assigned cluster centers.

b) **Feature generation:** (7 pts) Using the k clusters from a), quantize each pixel to one of k values by assigning them to the index of the nearest cluster center: $v_i = \text{argmin}_k ||\mathbf{x}_i - \mathbf{c}_k||_2$. Then, for each image, compute a histogram that counts the number of pixels that are quantized to each cluster center. Normalize the histogram so that it sums to 1. This histogram provides a summary of each image and can be used to compare the texture similarity between images.

c) **Image matching:** (6 pts) For each image in the "test" directory, find the training example with the smallest euclidean distance. Report the percentage of test examples that are correctly matched to a training example of the same type (e.g. class 12 has two training examples: D12_1.png, D12_2.png, and one test example: D12.png). For several images that are misclassified, find and display (in order) the top three training images with the smallest euclidean distances.

What is the smallest number of cluster centers (k) you can use before accuracy degrades significantly? - Explore this on a subset of the harder texture classes. What types of image transformation (e.g. scaling, rotation, etc.) is this representation invariant to?

Code for computing the LM filterbank can be found here: `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html`.

3 EM - Mixture of Multinomials (15%)

Probabilistic mixture models are useful in a variety of applications, such as gaussian mixture models for segmentation (see problem 4). Multinomial distributions are another useful distribution for mixture models, and can be used to model the bag-of-words representation seen in the previous problem: for a given texture i, codeword j occurs with probability θ_{ij}.

A mixture of multinomials would allow modeling images that are composed of multiple textures, each defined by Θ_i, where texture i occurs with probability τ_i. More sophisticated methods such as pLSA and LDA replace τ with a per-image distribution over textures classes, which must be inferred. Again, note that this was originally introduced for representing documents composed of multiple "topics."

Derive the EM algorithm for the following multinomial mixture model for n examples $\{x_i\}$:

$$P(\mathbf{x}|\{\Theta_i\},\{\pi_i\}) = \sum_i \pi_i P(\mathbf{x}|\Theta_i), \text{ s.t. } \sum_i \pi_i = 1,\ 0 \le \pi_i \le 1$$

$$P(\mathbf{x}|\Theta_i) = \frac{n!}{\prod_j x_j!} \prod_j \theta_{ij}^{x_j}, \text{ s.t. } \sum_j \theta_{ij} = 1,\ 0 \le \theta_{ij} \le 1$$

Show the Expectation step (7 pts) and give the EM update formulae for π_i (3 pts) and Θ_i (5 pts). Show all steps including application of Bayes rule and computation of derivatives. Lagrange multipliers can be helpful for keeping π_i and Θ_i on the probability simplex.

4 Segmentation (25%)

a) (15 pts) Implement a Gaussian mixture model with diagonal covariance to perform foreground/background segmentation on the butterfly RGB image provided. Initialize the foreground by the pixels inside the box shown in red and background by the rest. The top left and bottom right corners of this box are $[(29; 104)(475; 248)]$. Generate seperate GMMs for foreground and background pixels ($P(x|\Theta_{fg})$, $P(x|\Theta_{bg})$ respectively). Use an appropriate number of clusters and estimate the model parameters using EM. You may use K-means for initializing EM. Show the probability that each pixel belongs to the foreground with an intensity image and the final pixel segmentation. Explain your choice of the number of mixture components for foreground and background and the choice of the threshold for generating the final segmentation.

b) (10 pts) Refine the segmentation using graph cuts. Use the estimated FG and BG log probabilities for the unary term and a contrast sensitive model (e.g. $\exp(\frac{-||c(x)-c(y)||^2}{2\sigma^2}$, where $c(x)$ indicates a pixel's color) for the pairwise term.

Code for computing the graph cuts can be found in the included GCmex1.5/ directory. OS-X users must first run compile_gc.m. Please contact Ian if there are any issues executing these functions.