CS 294-2 Grouping and Recognition

Jitendra Malik, CS Division, Dept. of EECS

Fall 1999

- Implement a brightness image segmenter based on modeling the image as piecewise constant and using k-means. Try this on both synthetic and real images. Now generalize the model to a mixture of Gaussians (each patch has brighness values drawn from a normal distribution with paramaters μ_i, σ_i) and use EM. You can use the k-means solution to initialize the EM algorithm.

- Consider the Markov Chain for the drunk on a line of length 10 (reflecting boundary conditions as in the example in class). Just to make it a bit more fun, make rightward transitions twice as likely as leftward transitions. You can compute the stationary distribution of this Markov Chain by finding the first eigenvector. Or you can find it by simulation (run the chain for say a 1000 steps and compute the empirical frequency of each of the 10 states). Compare.

- Write functions for 1D and 2D filtering with Gaussian derivatives of first and second orders. Test them on synthetic and real images. You may find the Matlab functions *conv*, *conv2*, *filter*, *filter2* relevant. Use 6 equally spaced orientations and 3 scales (σ spaced half-octave apart) and an aspect ratio of 3:1. This should give you 36 filters.

- Use k-means to find textons for some real images.

- Implement an edge detector based on finding local maxima of oriented energy. You can implement this for a given scale by finding the dominant orientation at a pixel and moving perpendicular to it to find if it is a local maximum. Take the union of the edges at different scales to get the overall edge map.

- Implement segmentation for point patterns based on proximity using normalized cuts. Assume an exponential decay with distance.