ECE 512 / CS 532 - Computer Vision
Prof. Sen
Homework #3
Due: Wednesday, April 9, 2007

This homework assignment consists of only programming assignments. You need to submit it by email before midnight on the due date. Two of them will require you to develop programs in C or C++ using VisualStudio. The last will require a MATLAB program. I have provided two different image data sets, one for problems 1 and 2 and the second for problem 3. Remember, you are to work independently! I will give you a zero if it seems that you have collaborated with others without permission. Submit your .m files and zip up your VisualStudio projects for submission.

1. In this problem, you will learn how to represent the data structures for matrix-based segmentation algorithms we discussed in class. You should do your programming in C++ using Visual Studio and start with the project template given in ImageProc.zip. You have to create a specialized matrix data structure because you cannot represent the data in a continuous array of data for obvious reasons: a 512×512 image will require an affinity matrix of size $512^4 = 68.7 \times 10^9$. If you store floats at every position, it would take 275 GB of memory (while 32-bit Windows only supports a maximum of 4GB of addressable memory)! Therefore, it is important to know how to implement a sparse matrix data structure, which is what you will do in this assignment.

   (a) First, you will have to implement a sparse matrix data structure class to represent the affinity matrix. Provide the following operators: addition, subtraction, and multiplication (with a matrix, a vector, and a scalar). You can assume that the affinity matrix will be symmetric (element $M_{ij} = M_{ji}$) to save space even further.

   (b) Once you have written your sparse matrix class, use it to compute the affinity matrix for an image in the test data set. Combine intensity and distance to come up with the affinity measure, using the formulas shown in Section 14.5.3 of the book. Set the distance and quantization to go to zero at a 15 pixel distance. For intensity, have it fade to zero with an intensity change of 10%. Run your algorithm on the images in SegmentationTest.zip.

   (c) Extra credit (worth 100% of HW3 grade): Write a solver for the eigenvalues/eigenvectors for your sparse matrix. I suggest the Lanczos algorithm. Use this to segment the image in SegmentationTest.zip as described in the normalized cuts paper.

2. In this problem, you will implement the *k*-means algorithm to segment intensity images. It is probably best to write this in C++ using VisualStudio.

   (a) Implement the *k*-means algorithm so that you can automatically segment an intensity image into a user-specified number of clusters.

   (b) Run your algorithm on the images of the data set in SegmentationTest.zip and show the segmentation results with a varying number of clusters.

3. In this problem, you will write a series of programs to automatically extract the vanishing point in images, which is similar to long problem #2 of the midterm. You are to do this for all the test images in VanishingPoint.zip. This problem is best done in MATLAB.

(a) Use your Canny Edge Detector from homework #2 to identify the pixels lying on object edges in the image.

(b) Apply the Hough transform to these pixels to identify line segments that go through each of those edge points. Use the results of the Hough transfrom to find the equations of the most likely lines in the image.

(c) Now draw out these lines in image space and see where they intersect most often. The idea is that the vanishing point will be the point where the largest number of lines in the image intersect. You might have to do some averaging to find the vanishing point accurately. Mark the point on the image.