CSci 4968 and 6270
Computational Vision,
Fall Semester, 2011-2012
Homework 3
Due: Tuesday, October 18, 2011, 5 pm

Submission Guidelines

This homework is somewhat different. There are four questions, but only one — the first one — involves programming. Also, depending on how we progress in class, the due date for the fourth question may be delayed by a week.

Please submit your solutions via email to `cvstewart@gmail.com`. Submit a *pdf file* containing your write-ups and, for problem 1, your image results and source code files. You may hand-write your solutions, but either scan them or take a well-lit picture that you submit electronically.

Problems

1. (**30 points**) In this problem, you will explore k-means clustering in the context of **color** image segmentation. There is a k-means clustering implementation in Matlab and in most existing computer vision software systems.

   (a) Select a series of images on which to apply k-means clustering. For some of these, you should be able to obtain nice results, while for others the results will likely be quite poor. See if you can predict when. As part of your submission, show me these images.

   (b) Start with the basic k-means implementation:

      i. Take the pixels of your image and convert the RGB values to HSV values (Matlab function `rgb2hsv`) to form 3-component data vectors, and run k-means on them. Make sure your image is not too large — you may need to downsample the image. Think about and experiment with a good size to work with.

      ii. Use the result to produce a segmentation of your image as follows: Each pixel will have a resulting label. Each connected component of pixels all having the same label is a segment. Show an image where each pixel in a segment has the same color and pixels in different segments have different colors. make sure the results clearly show the resulting segmentation.

   (c) Experiment with different sizes of k ranging from very small (just 2 or 3) to very large. Illustrate and explain your results. How does the computation time change with increasing k?

   (d) Add pixel location information to the data vector, making it a 5-component vector. You may need to scale the pixels values up or down to get a good result so that the pixels and the colors are roughly comparable. Compare the results here to the results without pixel locations.

2. (**20 points**) In our discussion of image processing we looked at an example of applying some techniques to identifying changed regions (moving regions) between two images. In particular, we computed image differences, thresholded the result, applied

morphological operations, and then used connected components labeling to extract the regions of change/motion. Now consider readdressing the same problem using graph-cuts:

(a) How might you formulate your data term, R_p? How might you handle regions of the images that are homogeneous (roughly all one intensity or a very slowly changing intensity)?

(b) How might you formulate the neighborhood term?

In each case, argue briefly and coherently for the ideas you choose.

3. **(15 points)** Give an algebraic proof that a straight line in the world projects onto a straight line in the image. Hint: write the parametric equation of a line, project it to the image, and remove the parameter t, yielding the implicit form of the line.

4. **(25 points)** This problem explores the mathemetics of implementing the SIFT descriptor voting algorithm without having to write the code.

   A keypoint has been detected at image location (x_0, y_0), and the gradient histogram technique has computed an orientation θ_0, where $\theta_0 \in [0, 2\pi)$. We want to compute the vote for a nearby pixel location (x, y) with gradient direction angle θ (computed from the partial derivatives of L) and magnitude g. All of the foregoing measurements are in the image coordinate system at the scale at which the keypoint is detected. Let w be the width of the sample regions used in forming the descriptor. The descriptor is computed in a new coordinate system, which we call the "keypoint coordinate system", centered at (x_0, y_0), with its x axis oriented along angle θ_0 with respect to the image's x axis. Votes are determined in this coordinate system.

   (a) Give the locations of the **centers** of the $4 \times 4 = 16$ sample regions (width w), both in the image coordinate system and in the keypoint coordinate system. Hint: this is relatively simple in the keypoint coordinate system, so start there and then apply a similarity transformation to take them into the image coordinate system.

   (b) What are the position and orientation of (x, y) and θ in the keypoint coordinate system? For this you need to apply the similarity transformation in reverse.

   (c) Now consider the voting that forms the descriptor. Define the representation (in software) needed to compute and store the votes.

   (d) What sample regions does (x, y) contribute votes to (at most 4)?

   (e) What orientations does θ contribute votes for (at most 2)?

   (f) Give the formula for the weighted vote from image coordinate (x, y), with gradient orientation θ and magnitude g, for one combination of sample region and orientation from the previous two answers. This is effectively the trilinear interpolation step discussed in the paper.