Comp. 4900D: Assignment #2
Due: Thursday March 1, 2012

1) The goal of this question is to create a program that take as input two images that are related by a homography, and which "warps" the second image (piscine2.bmp) to align with the first image (piscine1.bmp). Both of these images are on the course web site. The output image that is created should have the original first image, along with the warped version of the second image. I have made the first image big enough to hold both the original first image along with the warped version of the second image. Your program needs to find some features in the two images, match these features to compute a homography, and use this homography to help create the final image which is a combination of the two input images. The simplest way to accomplish this is to use the code from the program findmatch.cpp in the C Samples directory (which you can find in the Windows OpenCV program menu) and modify it as necessary. Send me your source code and the final output image that you create which combines the two input images. In the area of overlap between the two images we have a choice of using pixels from the first image, or the second image. Describe some reasonable strategy for using pixels from both images to create a perceptually pleasing blend (you do not have to implement this strategy, just give a short description). **6 marks**
2) Two kinds of line detection are implemented in OpenCV, the Hough Transform and the probabilistic Hough transform. Assume that there are **n** feature points in an image and that it takes O(**k**) time to increment the accumulator array for each feature point when computing the normal (not probabilistic) Hough transform. What is the running time of the Hough transform in big O notation, that is O(…). Justify your answer. **1/2 mark**
3) The probabilistic Hough transform uses random sampling instead of an accumulator array. In this approach the number of random samples **r,** is not specified in the OpenCV call, but is an important hidden parameter. If there are **n** feature points in an image then what is the running time of the probabilistic Hough transform in big O notation. **1/2 mark**
4) If we wished to find an ellipse using the Hough Transform, which of the two approaches is most practical (Standard HT, or probabilistic HT). Justify your answer. **1 mark**
5) When we compute the Canny edges using different thresholds for the edge linking process do the number of edges change, and do the locations of these edges change? Give a yes or no answer to both of these questions and justify your answer in both cases. Similarly, if we take the input image and smooth it with a Gaussian of a significant size before computing the Canny edges does the number of edges change, and do the locations of these edges change? In this case by change we mean are they different from the result compared to the original unsmoothed input image? Give a yes or no answer to both of these questions and justify your answer in both cases. **2 marks**