

Comp. 4900D: Assignment #2  
Due: Friday March 1, 2013 at 9:00 AM

- 1) Write a program that takes the image tracks.jpg and finds Canny edges, and then finds lines in these edges using the Hough transform. The easiest way to do this is to take the programs edge.c and houghlines.c in the C samples directory of OpenCV, and then combine them. First find edges (there are three parameters you can adjust to create different edges; one way to try different combinations is to change them in real-time using trackbars). Then find the lines in the resulting edges using the Hough transform and draw the lines over the edges in the image. You should adjust the Canny parameters so that the results should look similar to the image hough-lines.jpg. In particular, there should be between two and five lines (no more and no less). You should submit your software along with the final output image. **3 marks**
- 2) Use the program called harris-shell.c and add some code to find the corners in the image checkers.jpg. You should say that a pixel in the image is a corner if it passes the given threshold for Harris Corners. You do not need to thin the corners using non-maxima suppression. You should draw a small circle or dot on these corner pixels, and then save the resulting image. The final image should look similar to Corner-out.pgn. You need to look up the formula to find the eigenvalues of a real, symmetric, two by two matrix. This is described in <http://people.math.gatech.edu/~klounici6/2605/Lectures%20notes%20Carlen/cha p3.pdf> You should submit your source code along with the final output image. **4 marks**
- 3) 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  $O(\dots)$ . Justify your answer. **1/2 mark**
- 4) 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**
- 5) 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**
- 6) 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. **1 marks**