

Hough Transform Circle Detector

Duration: 2 1/2 weeks (27 July 2016 – 15 August 2016)

Handin: Monday August 15, 5PM (via Vula)

Problem statement:

Write a simple Hough feature detector that can detect Circles of arbitrary size in the input images.

The program must support the following functionality:

Simple edge filter A grey-scale image will need to be processed by a simple edge filter (see notes) followed by thresholding, to determine a binary image which contains background and edge pixels only;

Highlighting the extracted feature(s) Once a feature has been detected, a circle corresponding to the feature must be plotted over the image to show the result. This image can be saved and then viewed.

Multiple features the procedure to determine accumulator maxima will need to search for multiple local maxima; each of these will be a new feature (circle).

Some sample test images with circles are available at

<http://www.cs.uct.ac.za/~patrick/hons/images>

Note: the code needs to **compile/run** on nightmare.cs and the zip should contain everything required to build/run on nightmare.cs. If you supply a binary, it should be statically linked.

You should include a concise text report that explains how you tackled the problem, issues you found, and how to run the program on input files.

Most of the marks will be for the output of the algorithm (whether it correctly finds circles) but some will be allocated to well structured/commented code, optimizations to make the code run faster and so on. A GUI is not necessary, but will get you some additional marks. At the very least your program, should read an image and produce an output image, as described above. You can use Java or C++ as long as I can run the resultant program (please make sure to include all libraries I might need if it is C++). I do not have Eclipse etc so Java would need to work with just the regular JRE installed (or packaged as a complete .jar file). **You are permitted to use image libraries for Image I/O – all other code must be your own.**