

DIP HW4 Report

R07922036 邱能賢

May 15, 2019

Problem 1: Hough Transform for Line Detection

- (a) Perform edge detection on I_1 and output the resultant edge map as E .

I perform edge detection with Canny edge detector, the same method as in HW#2.



I_1 , sample1.raw

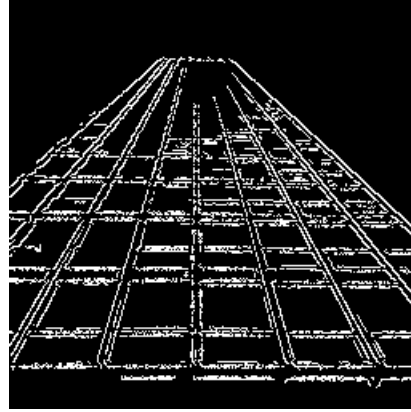


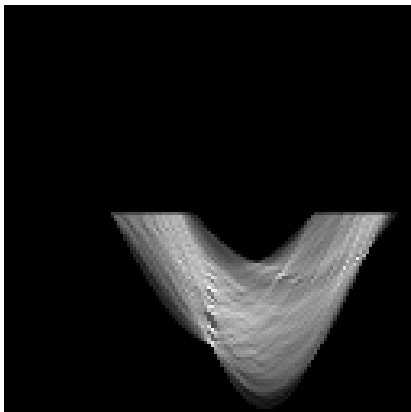
image E - edge map

- (b) Perform Hough transform on E and output the accumulator array as a new image, H_1 , where the horizontal axis and vertical axis represent theta and rho values.

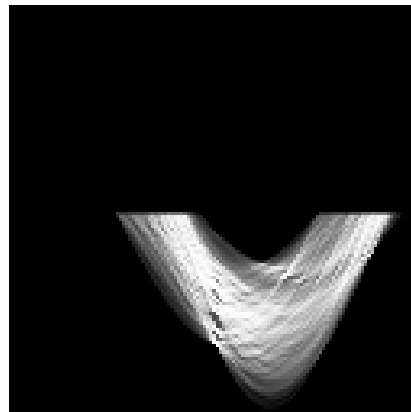
I set the accumulator array with a size of 127×127 , that is, quantize both ρ and θ to 127 bins.

- (c) Perform contrast adjustment on H_1 and output the result as H_2 for better visualization.

I perform global histogram equalization to do the contrast adjustment on H_1 , as in HW#1.



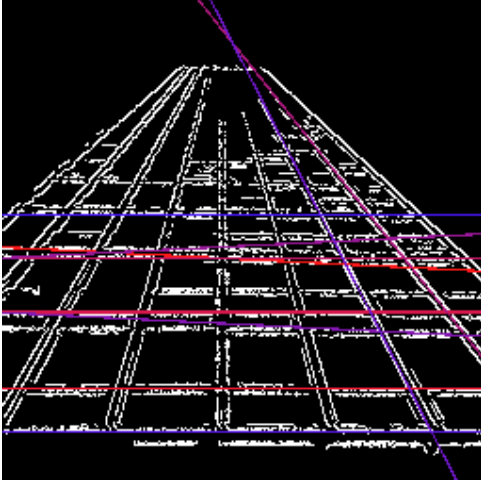
H_1



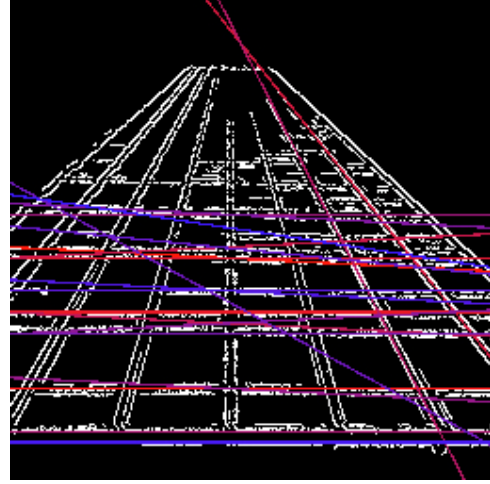
H_2

- (d) By utilizing the accumulator array, draw the top 10 and top 20 significant lines with different colors on the edge map E and output the resultant images as D_1 and D_2 .

I use an C++ `multimap`, which is implemented with a binary tree, to sort the accumulator array values. Then, find the top 10 or 20 (ρ, θ) coordinate. Each (ρ, θ) coordinate corresponds to a line on the image coordinate. The color of each line represents the order of their accumulated values. The red one is the most significant line, and the color turns blue as the accumulated value decrease.



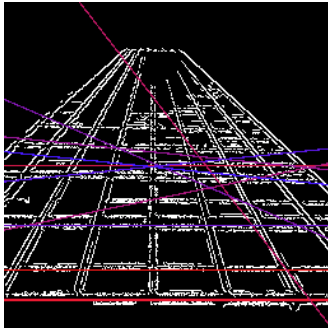
D_1



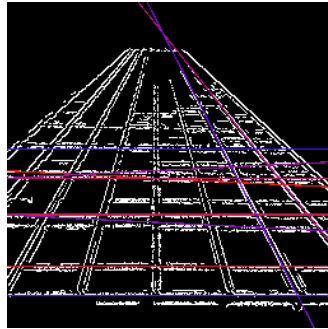
D_2

Most lines did align with the edges, but because the edge map is not very clean, some lines are overlaying each other with only slight difference in angle. In D_2 , there are even some less significant lines that doesn't align with the edges.

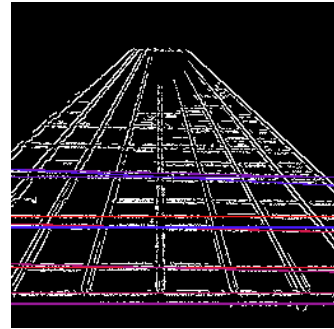
I've also tried different Hough array size, and here are the result of image D_1 :



size = 63



size = 127



size = 255

When the array size is too small, many edge map pixel will be accumulated at the same bin. Without enough “resolution”, the lines doesn't match well with lines on the edge map. On the other hand, if the array size is too large, due to the unclean edge map, one noisy line may be quantized into multiple pixels. So in image D_1 we can see that one edge will be split into 2 many overlaying lines.

Hough transform is a very good method to detect lines or circles in an image, even if there are some discontinuity of the edges or background noise. But, as the discussion above showed, the result is highly affected by the quantization array size, so it must be chosen very carefully. Also, Hough transform is dependent on the quality of the edge map. The result will be much cleaner and accurate if the method of edge detection is improved.