Matthew England
2/15/2016

Computer Vision CS8690/ECE8690
Assignment 1: Linear Hough Transform

Introduction:

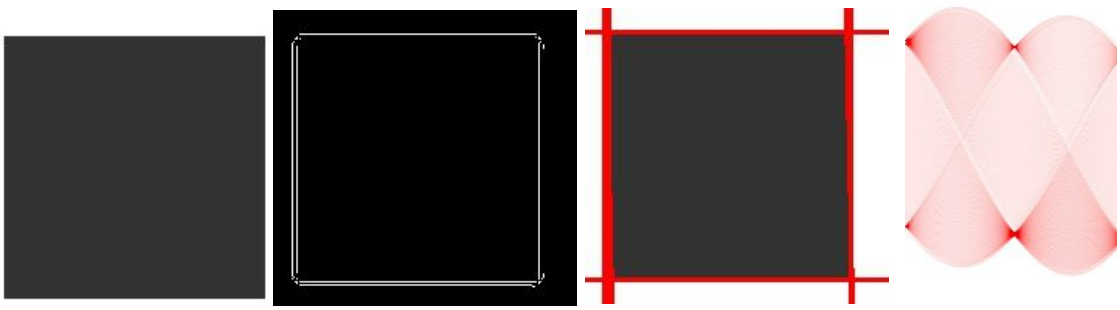
The hough transform allows researchers to find linear line segments in an image. The hough transform can be a feature extractor to help detect, roads, walls, and straight contours in images. A advantage of using hough transform is the ability to transform into a polar coordinate space (a.ka. Hough Space). This allows for the bucketing of overlapping polar coordinate lines to determine line segments.

Implementation:

I used the provided pseudo code look into other methods of searching the image space for lines within the confines of the hough transform. I convert the incoming image into grayscale to simply edge detection. I blur the image using a gaussian blur, then I apply the canny edge detector over the image. After that I begin to find points in the image. First I must convert every point into polar coordinates using the function: r = x * cos(theta) + y * sin(theta). Using the found polar coordinates, you store them into a grid histogram (i.e. accumulator matrix). After traversing the image, you then must find the local maximums of which a max window filtering over the accumulator matrix. I use a window search function of 9 x 9 pixels. If you find that your current point is not a local maximum, you must skip that point and move onto the next polar coordinate. After finding all the local maximums, I begin to add them together to create lines. I then convert the polar coordinates back into cartesian coordinates (2D image coordinates).

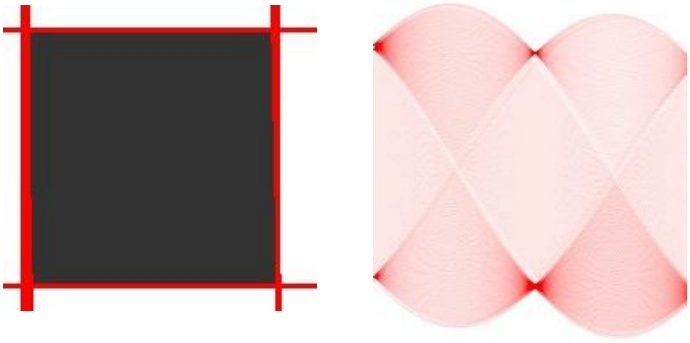
Results:

Square Image (Threshold = 25): Lines = 7

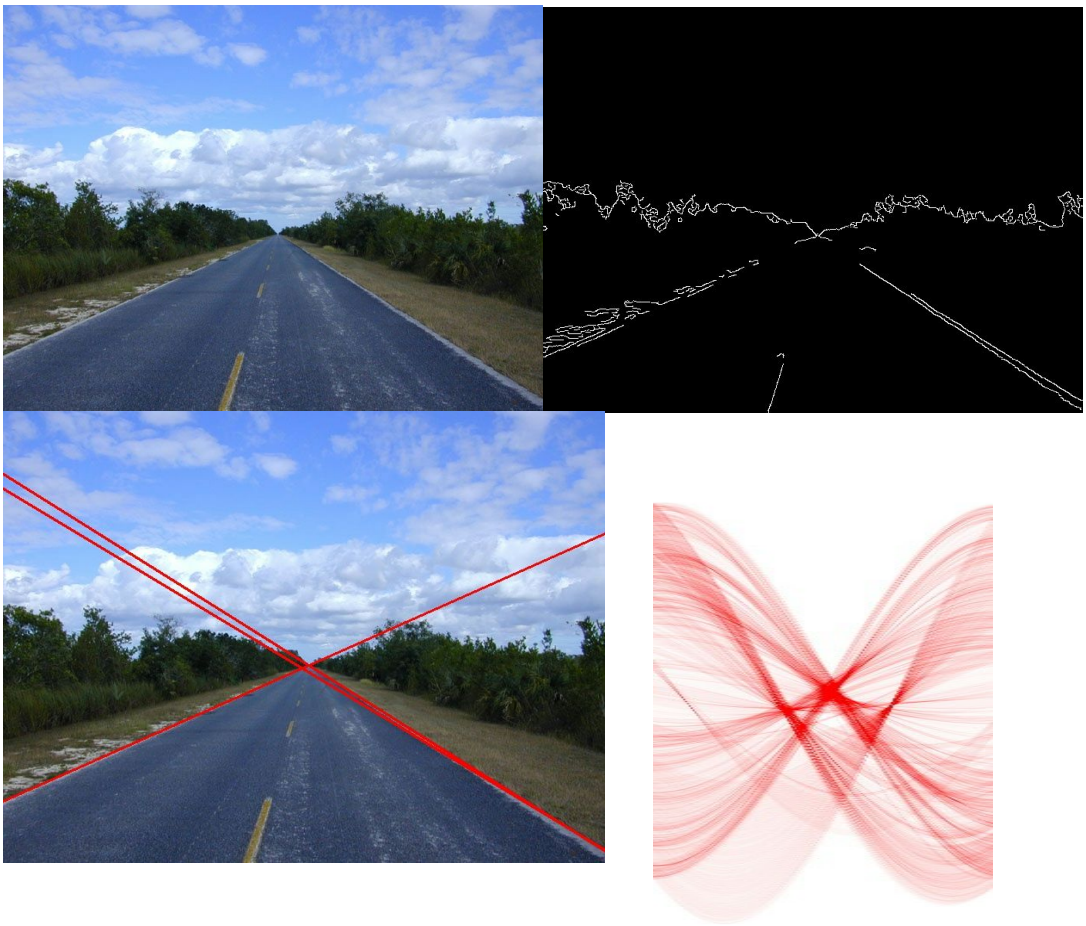


Matthew England
2/15/2016

Square Image (Threshold = 120): Lines = 5

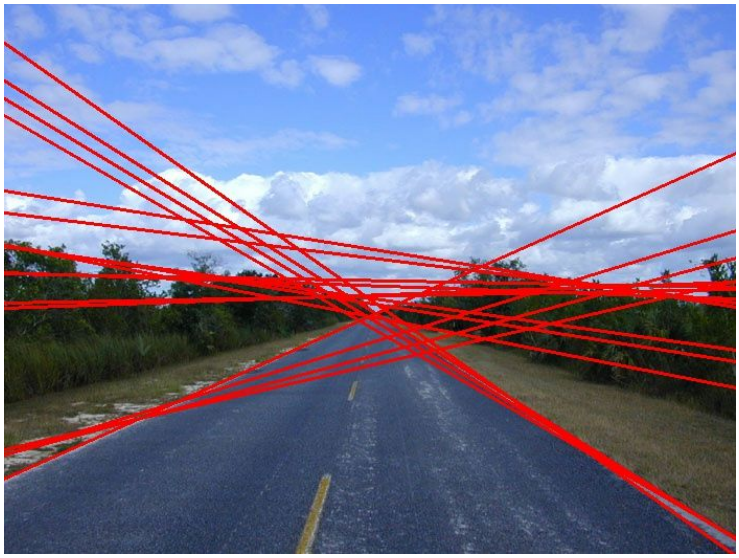


Road (Threshold =120): Lines 3

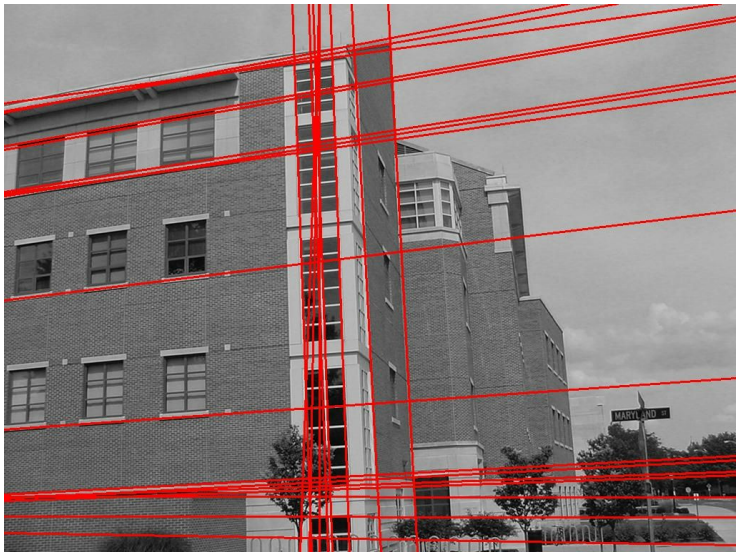


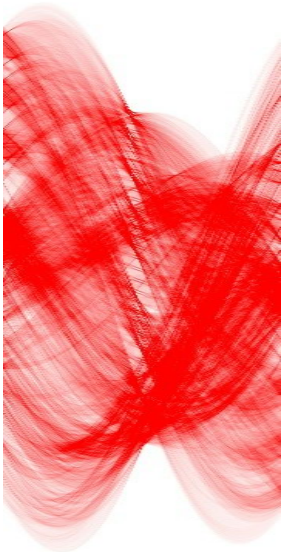
Matthew England
2/15/2016

Road (Threshold = 60): Lines 17



Building (threshold = 120): Lines 26

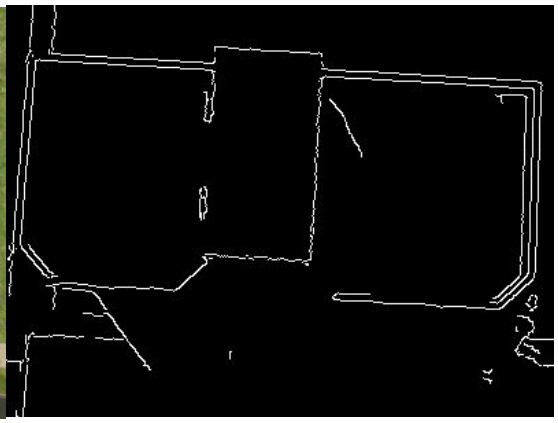


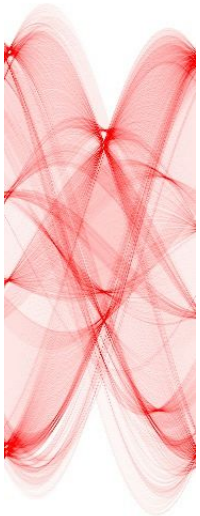


Matthew England
2/15/2016

Building (threshold = 175): Lines 7

Roof Top (threshold = 120): Lines 7

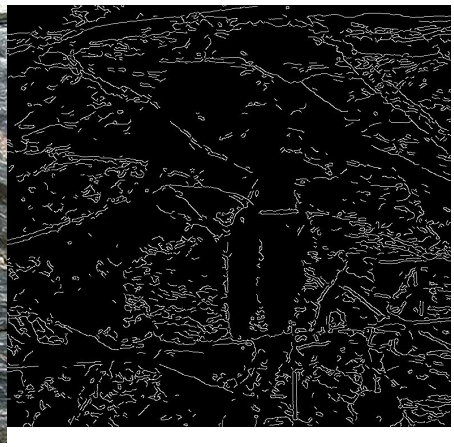


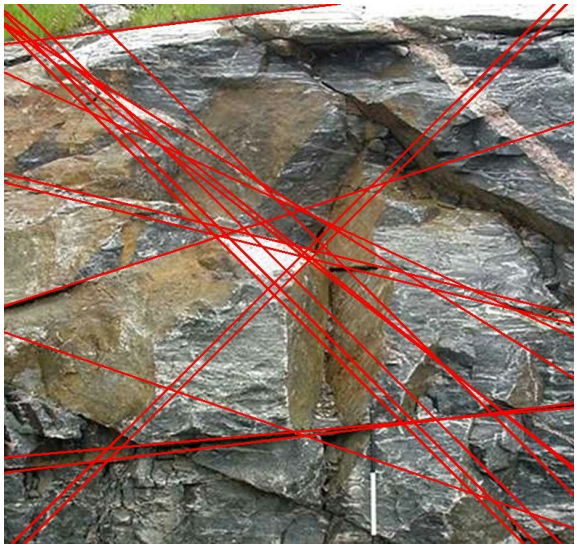


Matthew England
2/15/2016

Roof Top (threshold = 60): Lines 10

Rock (threshold = 125): Lines 17





Matthew England
2/15/2016

Conclusion:

A issue with hough transform you need to make sure you have only strong edges present from your edge detection. If you have a weak edge detection, detecting lines is difficult. As shown in the rock example. The texture of the rock provides a lot of edge points to use, thus the hough space is saturated. Another interesting discovery is as you decrease the threshold for the hough transform, you will always get an increase of found lines. However each image requires different thresholds for line detection such as the square requiring only 25 while the building requires 120 for good results. The best threshold for each image is shown above. As you increase the threshold, the hough transforms provides us with valuable information, which lines are the important in the image. If you need to find distinct lines, such as the shoulder of a road, then hough transform can provide valuable assistance. Using the hough space is critical into figuring out your important points. The darkest cluster of points in the hough space displays to us, that is a critical point in the image. Those critical points will be used to generate lines.

I did have some issues with creating the lines with most of the the images. But I improved my solution from last time, I found out I did not do enough decimal places for degree to radian conversion and I got caught on integer truncation on a polar coordinate to cartesian conversion. Lastly, I do have great results on road line detection, buildings, and square. As one can see I get the edges of the road until it becomes a vanishing point and my lines cross.