Cody Nichoson

ECE 332

November 4, 2021

MP#6 - Hough Transform Line Detection

Description of Aglorithm

The purpose of this algorithm is to identify prominent lines in a given image. The process undergone is known as the Hough transform. The algorithm uses edge detection (Canny in this case) and a voting procedure in order to determine the most likely candidates of lines in an image. An important aspect of the Hough transform is the parameter space, which is an alternate way of expressing a line and is where the voting procedure takes place.

The Hough transform first begins with edge detection. I used the Canny edge detector in my algorithm due to its prevalence and simplicity in implementation. With clear edges identified, all non-zero points in the image (those detected as "edges" by the Canny detector) are transformed to the parameter space, which may be referred to as (m,c). This transform converts these (x,y) points to corresponding lines in the parameter space. A series of points in the (x,y) plane are considered to be linear when they have the same slope (m) and y-intercept (c). Conversely, this must also mean that a series of lines in the (m,c) must be linear in the (x,y) plane when they intersect at a common (m,c) point. Knowing this, it is logical that a large number of parameter space lines intersecting at a common point must mean that point represents a well-defined line in the (x,y) plane. This process can be viewed as a "voting" system, where each line intersection in the parameter space (m,c) is a "candidate" and those candidates with a large number of lines "voting" for them likely represent well-defined lines in the (x,y) plane. Once the

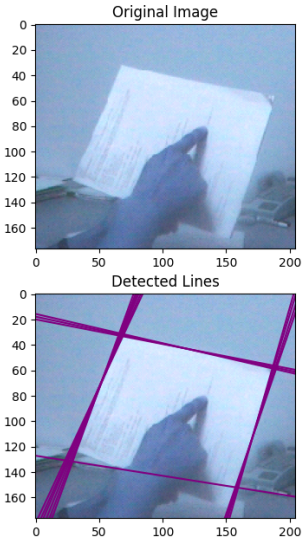
lines with the most votes are determined, they can be converted back to the (x,y) plane and plotted on the original image to confirm the line detection.

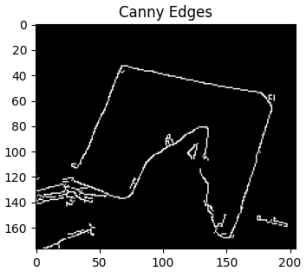
Results and Analysis

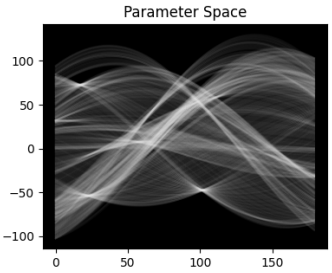
Testing the three images provided required the adjusting of several key parameters in order to achieve desirable results. Namely, the threshold parameter used in the Hough transform played a large role in the results' quality. The simple test images, test.bmp and test2.bmp, were less prone to error when changing the threshold, but the more complex image, input.bmp, required more precise tuning of the threshold. A threshold that was too high would not detect any lines, while a threshold that was too low would produce a large number of lines in inaccurate locations.

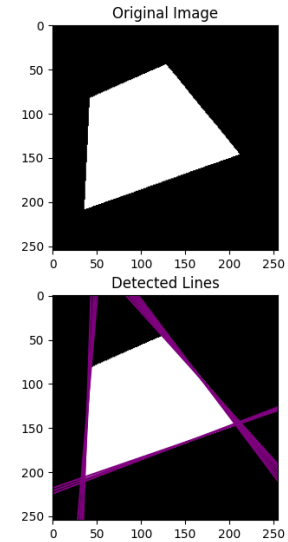
Additionally, the low and high thresholds of the Canny edge detector also played a role in the quality of the Hough transform. Similar to the Hough threshold, the simple test images were not as sensitive to Canny threshold changes, while the more complex input.bmp image required more tuning to achieve a desirable result.

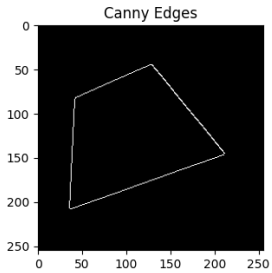
In general, the Hough transform performed as desired and displayed the expected lines for each test image. The one area that could be improved, however, is the number of lines detected for each actual line. In my results, each line in the image results in several detected lines at slightly different angles imposed over one another. In a perfect line detector, only a single line would be detected for each actual line in the image.

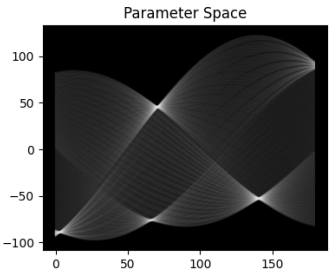

Original Image
Detected Lines

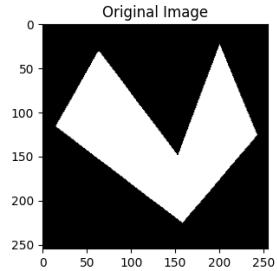

Canny Edges

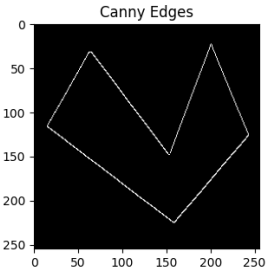

Parameter Space

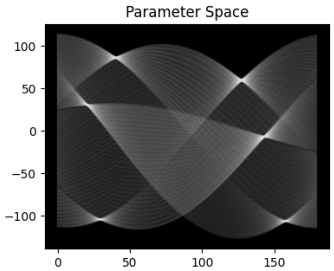

Original Image
Detected Lines

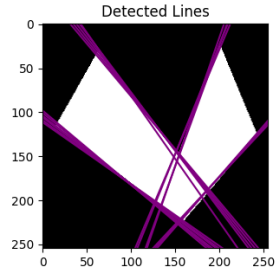

Canny Edges


Parameter Space


Original Image
0
50
100
150
200
250
0
50
100
150
200
250


Canny Edges
0
50
100
150
200
250
0
50
100
150
200
250


Parameter Space
100
50
0
−50
−100
0
50
100
150


Detected Lines
0
50
100
150
200
250
0
50
100
150
200
250