Name: Soo Ah Lee
Andrew ID: sooahl

HOMEWORK ASSIGNMENT 1
Image Filtering and Hough Transform

DUE : Mon September 23, 2019 11:59 PM

1. Theory questions

1.1 Hough Transform Line Parametrization

Q1. Show that it you use the line equation $\rho = x\cos\theta + y\sin\theta$, each image point (x,y) results in a sinusoid in (ρ, θ) Hough space. Relate the amplitude and phase of the sinusoid to the point (x,y)

If we start from $\frac{x}{a} + \frac{y}{b} = 1$, we can define a as $\frac{\rho}{\cos\theta}$ and b as $\frac{\rho}{\sin\theta}$, in this case ρ is the distance between line and origin. Using this formula, $\cos\theta$ is $\frac{\rho}{a}$ and $\sin\theta$ is $\frac{\rho}{b}$. To summarize, the equation is $x\cos\theta + y\sin\theta = \rho$. The shape of sinusoid is defined according to the position of the point regarding the origin of the image from which the parameters are computed. If the point (x, y) is far from the origin, the curve is going to have a wider amplitude. On the other hand, the closer the point gets to the origin the smaller become the amplitude. The position of the origin is an important aspect. Indeed if we let the origin at (0,0) points really far are going to have a very large amplitude, and we can have a bigger Hough space. And the phase is related to θ. So the slope of line depend on phase.

Q2. Why do we parametrize the line in terms (ρ, θ) instead of the slope and intercept (m, c)? Express the slope and intercept in term of (ρ, θ).

The line representation in terms of the slope and intercept (m, c) is not used in practice. Because for vertical lines the slope is infinite, that is, $m = \infty$. For this reason, we use the line in terms (ρ, θ). It does not exhibit singularities and provides a natural linear quantization for its parameters, the angle θ and the rho ρ. If we use in terms of the slope and intercept (m, c), the equation is $y = mx + c$. Then, we can define m using phase, that is $\tan\theta$. The equation becomes $y = \tan\theta x + c$, and multiply $\cos\theta$ at both side.

Q3. Assuming that the image points (x, y) are in an image of width W and height H, that is, x in [1, W], y in [1, H]. What is the maximum absolute value of ρ, and what is the range for θ?

The maximum absolute value of ρ is diagonal line of image, that is, $\sqrt{W^2 + H^2}$. The range of θ is depended on the theta resolution. But the overall line segment range is $[0, 2\pi]$. So the real range is $[0, 2\pi]$.

Q4. For point (10,10) and points (20,20) and (30,30) in the image, plot the corresponding sinusoid waves in Hough space, and visualize how their intersection point defines the line. What is (m, c) for this line? Please use Matlab to plot the curves and report the result in your write-up.

For the point (10, 10), plot the corresponding sinusoid waves in Hough space is

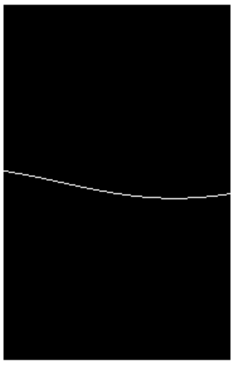


Figure 1. Plot the point (10,10) in Hough Space

For the point (20, 20), plot the corresponding sinusoid waves in Hough space is

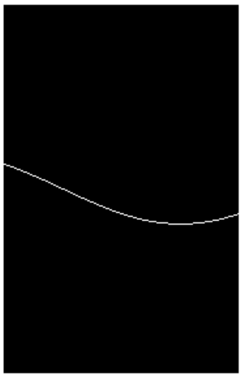


Figure 2. Plot the point (20,20) in Hough Space

The cross point of (10,10) and (20,20) in Hough space is

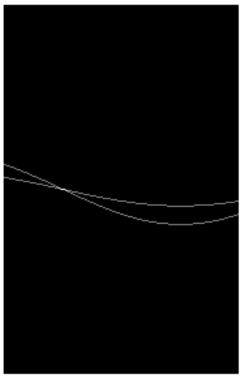


Figure 3. Cross point of (10,10) and (20,20)

For the point (30, 30), plot the corresponding sinusoid wave in Hough space is

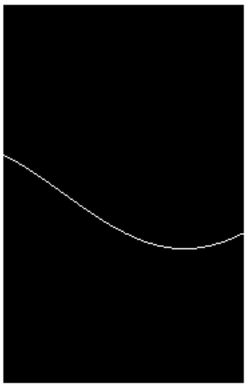


Figure 4. Plot the point (30,30) in Hough Space

The cross point of (10,10), (20,20) and (30,30) in Hough space is

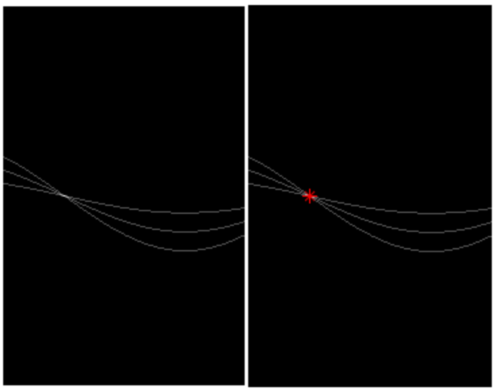


Figure 5. Cross point of (10,10), (20,20) and (30,30)

We can display this point at Image space. That point represents rho and theta of the line. So the line will be y = x as like below figure.

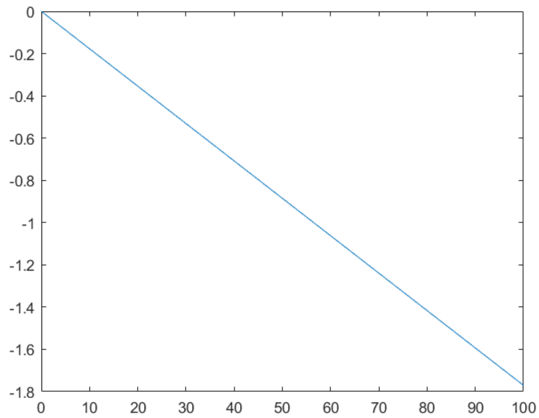


Figure 6. Plot the line corresponding point

2. Implementation

Q2.1 Convolution

I use sobel filter for x axis gradient, that is, [1 0 -1 ; 2 0 -2; 1 0 -1]. The result of my convolution is below.

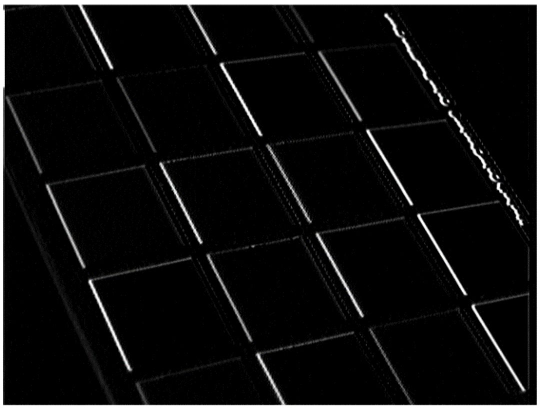


Figure 7. My convolution result

Q2.2 Edge detection

When I use my edge function, the result is below.

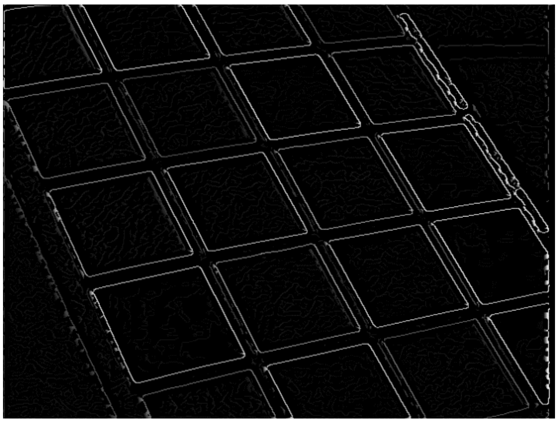


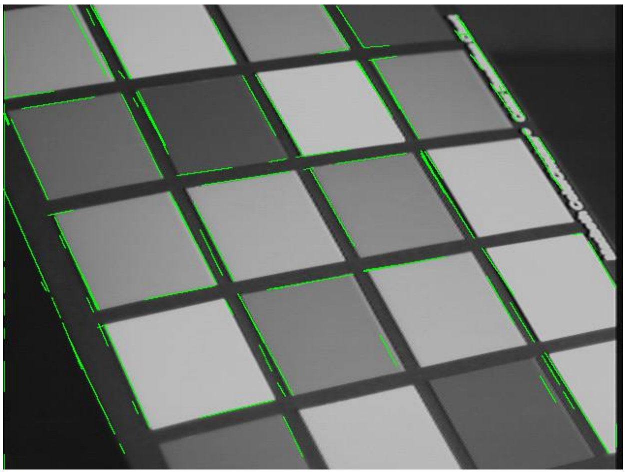
Figure 8. my Edge function result

Q2.3 The Hough transform

The result of my Hough transform is below.

Figure 9. My Hough transform result

Q2.4 Finding lines
Q2.5 Fitting line segments for visualization



The result of 2.4 and 2.5 is shown together.

3. Experiments

Q3.1 I use img03 for this experiment.

i) Use default parameters. (sigma = 2, threshold = 0.03, rhoRes = 2, thetaRes = pi/90, nLines = 50)

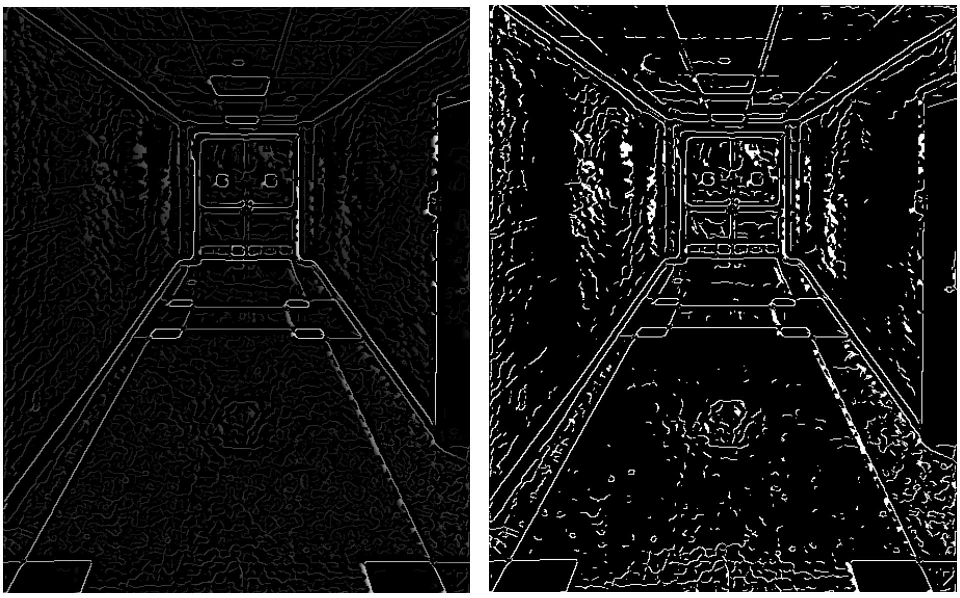
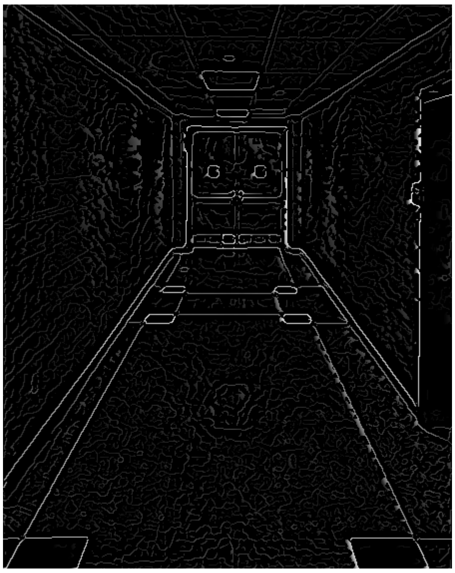


Figure 10. Edge result (left) and applying threshold result (right)

Figure 11. Hough transform result (left) and Find lines result (right)

ii) Change threshold (sigma = 2, threshold = 0.1, rhoRes = 2, thetaRes = pi/90, nLines = 50)



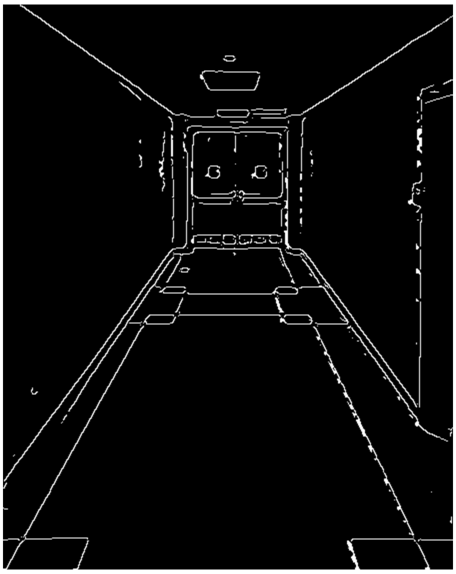


Figure 12. The result of threshold 0.1 (right)

Figure 13. Hough transform of threshold 0.1 (left) and line segmentation (right)

When I change threshold, the line segmentation result is more clear. Because, higher threshold refine the edge results, so not definite lines are deleted. So hough transform result is more clear and the lines are clearly. Especially, the line segmentation result is connected.

iii) Change rho resolution to 1(sigma = 2, threshold = 0.03, rhoRes = 1, thetaRes = pi/90, nLines = 50)

Figure 14. The result of rho resolution 1 (left) and line segmentation (right)

The line segmentation result of rho resolution 1 is more clear but the result is disconnected yet.

iv) Change theta resolution to pi/180 (sigma = 2, threshold = 0.03, rhoRes = 2, thetaRes = pi/180, nLines = 50)

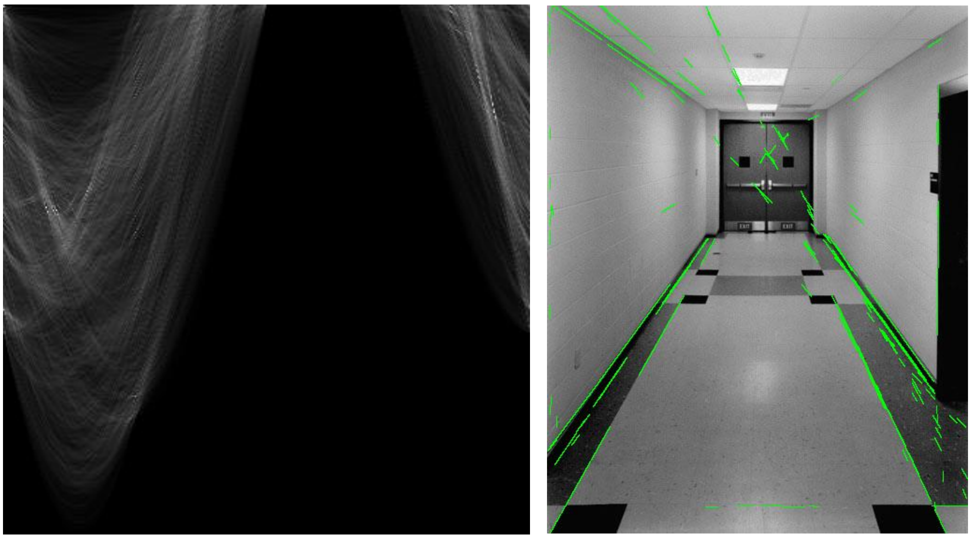


Figure 15. Hough transform of theta resolution pi/180 (left) and line segmentation result (right)

When we use theta resolution of pi/180, the space is more larger. And the difference of line segmentation result is that the discontinuous small segmentation is reduced.

v) Change nLines to 10 (sigma = 2, threshold = 0.03, rhoRes = 2, thetaRes = pi/90, nLines = 10)

Figure 16. Line segmentation result of nLines 10

This result just shows main lines. If we reduce nLines, we just select more important lines on image. As you see in image, the result is clearer and there are more less discontinuous slices.

vi) Change sigma to 3, threshold to 0.1, nLines to 30 (sigma = 3, threshold = 0.1, rhoRes = 2, thetaRes = pi/90, nLines = 30)

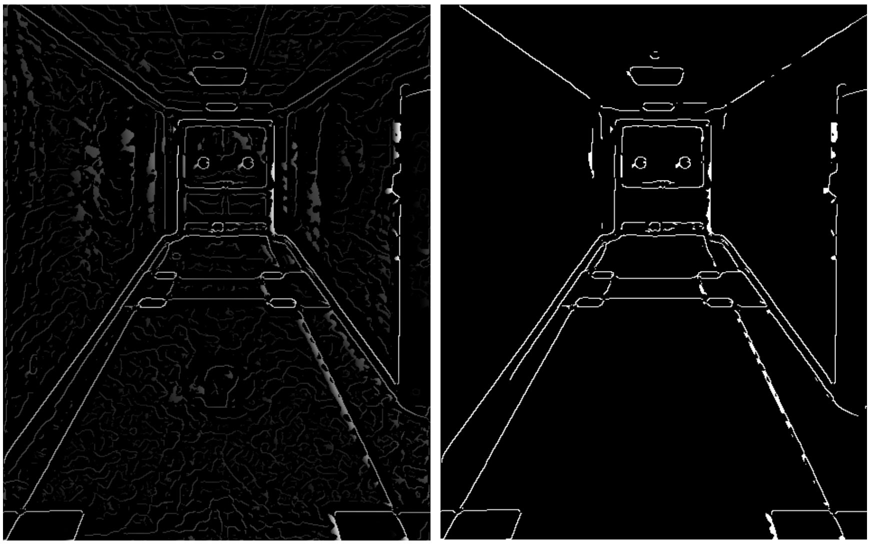


Figure 17. Edge detection result (left) and threshold result (right)

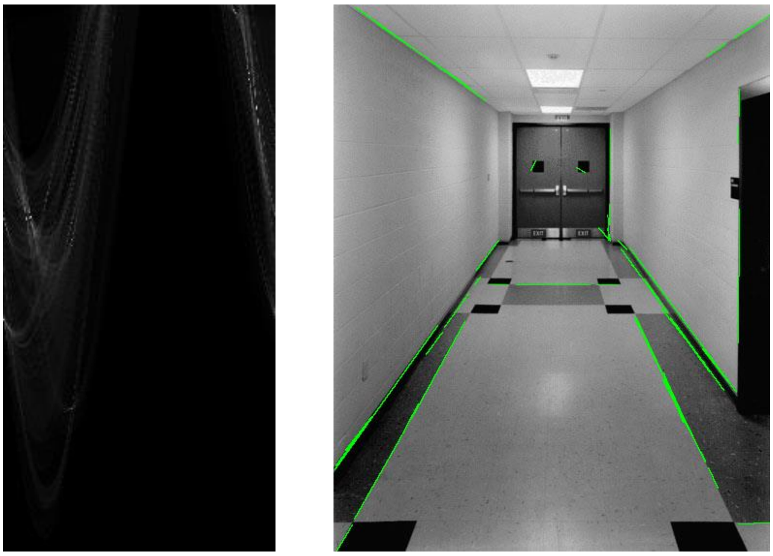


Figure 18. Hough transform result (left) and line segmentation result (right)

If we use higher sigma, the smoothing effect is more. So we can delete more noise. And threshold 0.3 just capture more distinct lines. At last we choose just 30 lines, that can reduce discontinuous short slices.

As you see above, my algorithm work well in various set of parameters. And in my case, set of (sigma = 2, threshold = 0.1, rhoRes = 2, thetaRes = pi/90, nLines = 30) is the optimal set. In our case, we just use 4 set of gradients. So the lines are not fitted to images. There are various set of lines in image. So if we want to more fitted result, we need to use various angles. So the Hough transform step causes most problems. If I want to my code more powerful, as I say before, I need to use more various gradient angles and do more pre-processing before transformation.

4. Try your own images!

Q4.1x Implement houghlines yourself

I use this image which is below.

Figure 19. own image

For this image, I choose parameter set as (sigma = 2, threshold = 0.3, rhoRes = 2, thetaRes = pi/90, nLines = 50). They are chosen empirically.

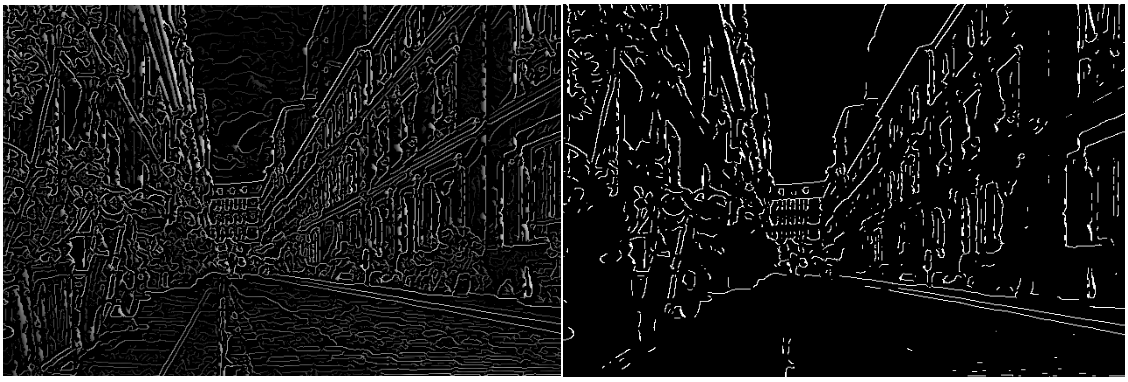


Figure 20. Edge detection result (left) and threshold result (right)

Figure 21. Hough transform result (left) and line segmentation result (right)

As you see above, the edge detection result is so noisy. There are so many features and obstacles. To reduce these, I apply high threshold to image to catch only important features. After hough transform, the result does not come together in one spot. I try solving it by adjusting parameters. The result of line segment is not clear. But it can catch main detections. There are some short slices. To solve it, we need to use various angle of gradients.