16720-B HW1

Jacob Miller

September 2020

2.1 Use the following trigonometry identity:

$$A \sin\theta + B\cos\theta = C\sin(\theta + \phi)$$
$$\text{where } C^2 = A^2 + B^2 \text{ and } \phi = \arctan\left(\frac{B}{A}\right)$$

Let $A = y$ and $B = x$, then

$$\rho = x\cos\theta + y\sin\theta$$
$$= \pm\sqrt{x^2 + y^2} * \sin(\theta + \phi) \text{ where } \phi = \arctan\left(\frac{x}{y}\right)$$

Based on the final sinusoidal equation, the amplitude is $\sqrt{x^2 + y^2}$ and the phase is $\arctan(\frac{x}{y})$.

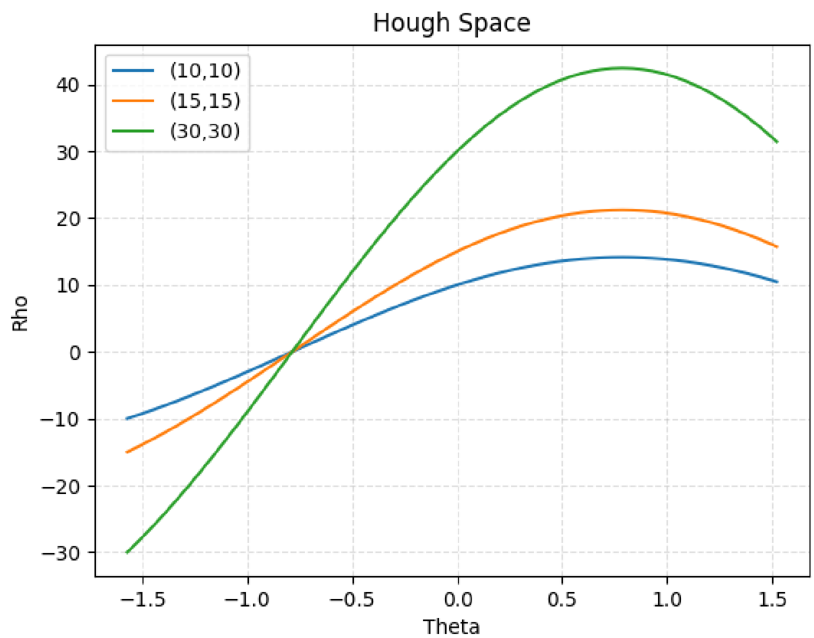
2.2 We parameterize the line using (ρ, θ) rather than (m, c) because the parameterization space is much more efficient. $m \in (-\infty, \infty)$ and the space of c is large, whereas $\theta \in [0, \pi]$ and $\rho \in [-\rho_{max}, \rho_{max}]$, resulting in a finite accumulator array size.

The following equations show how to express the slope and intercept of a line in terms of ρ and θ:

$$\rho = x\cos\theta + y\sin\theta$$
$$y\sin\theta = -x\cos\theta + \rho$$
$$y = -x\cot\theta + \frac{\rho}{\sin\theta}$$
$$Slope = -\cot\theta$$
$$Intercept = \frac{\rho}{\sin\theta}$$

2.3 By definition of Normal Form, we know that $|\rho| = \sqrt{x^2 + y^2}$. Given image points $x \in [1, W]$ and $y \in [1, H], |\rho| < \sqrt{W^2 + H^2}$. The range of θ must be equal to π due to the symmetry of Hough space, so it can be defined as $\theta \in [0, \pi]$.

2.4 As can be seen in the graph below, the sinusoidal waves intersect at $(\theta, \rho) = (-\frac{\pi}{4}, 0)$. Since $m = -\cot\theta$, substituting $-\frac{\pi}{4}$ for θ gives $m = 1$. Since $c = \frac{\rho}{\sin\theta}$, substituting 0 for ρ and $-\frac{\pi}{4}$ for θ gives $c = 0$.



2.5 As the dimension of the parameter space increases, both the time complexity and space complexity increases. Let D denote the number of dimensions in the parameter space. The time complexity will be $O(n^{D-1})$ and the space complexity will be $O(n^D)$. For example, the Hough Transform using (θ, ρ) has a 2D parameter space. Computing the Hough Transform requires looping over θ, making it $O(1)$. If θ were broken into n steps and ρ were broken into m steps, then the accumulator array would be $n \times m$, which is $O(n^2)$. If a third dimension were added to the parameters space, another loop would be added, increasing the time complexity, and another dimension would be added to the accumulator array/tensor.

3.2 To run 2D convolution with only one for loop and improve the runtime, I flatten the window of pixels from the image that will be multiplied with the filter. Each flattened window becomes a row in a new matrix. Once the matrix is filled, it is transposed so that each window becomes a column vector. Now the dot product of the flattened filter can be taken with each column vector to output the filtered image in vectorized form. This can be reshaped to the images original dimensions for visualization and processing further down the pipeline.

4 Testing the Gaussian Filters:
I performed two tests to verify the performance of my Gaussian filter functions. First, I visually compared the images before and after applying the Gaussian filter. Figure 1 shows the effects of applying the Gaussian filter. The edges are clearly less crisp, creating the desired blurring effect. The blurred image is also brighter because it was not normalized after applying the filter.

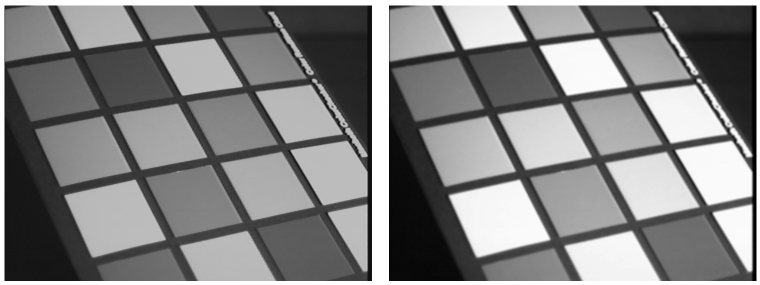


Figure 1: The original image (left) and the same image after applying the Gaussian filter (right).

The second test was examining the runtime for the filter with two for loop and the improved filter. The original image filter took approximately 1.39 seconds to run on the first image, while the improved filter took only 0.5 seconds. The tests were performed on the same CPU and the same image. The improved filter had 64% better performance.

Testing the Edge Filter and Non-Maximum Suppression:
Testing the edge filter and non-maximum suppression involved examining the edge magnitude image with the raw edge detections and after applying non-maximum suppression. Figure 2 shows the edge magnitude image before and after non-maximum suppression. The raw edge magnitude image looks as expected, with the edges of each square easily discernible and not much noise. After non-maximum suppression, the edges are slimmed down to one pixel wide, making the white lines look thinner and more crisp. I noticed better performance when comparing each pixel with the two neighbors on both sides of the gradient direction rather than just one neighbor per side.

Testing the Edge Threshold:
Testing the edge threshold involved observing the edge threshold image using different thresholds. Figure 3 shows an edge magnitude image after using a threshold of 10.0. This threshold performed well because it captured most of the edges and didn't include extraneous lines or much noise.

Testing the Hough Transform:
Testing the Hough transform involved checking the image representing the accumulator matrix. Figure 4 shows a representation of that accumulator matrix for

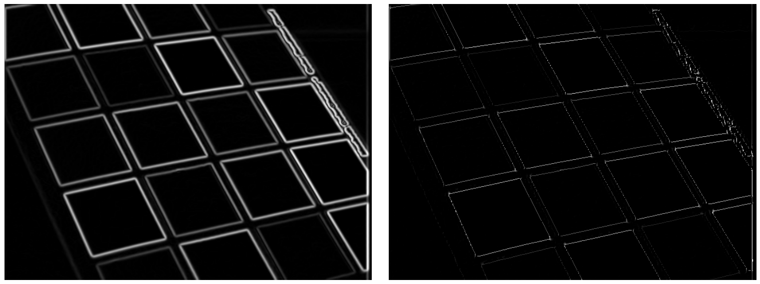


Figure 2: The edge magnitude image (left) and the same image after applying the non-maximum suppression (right).

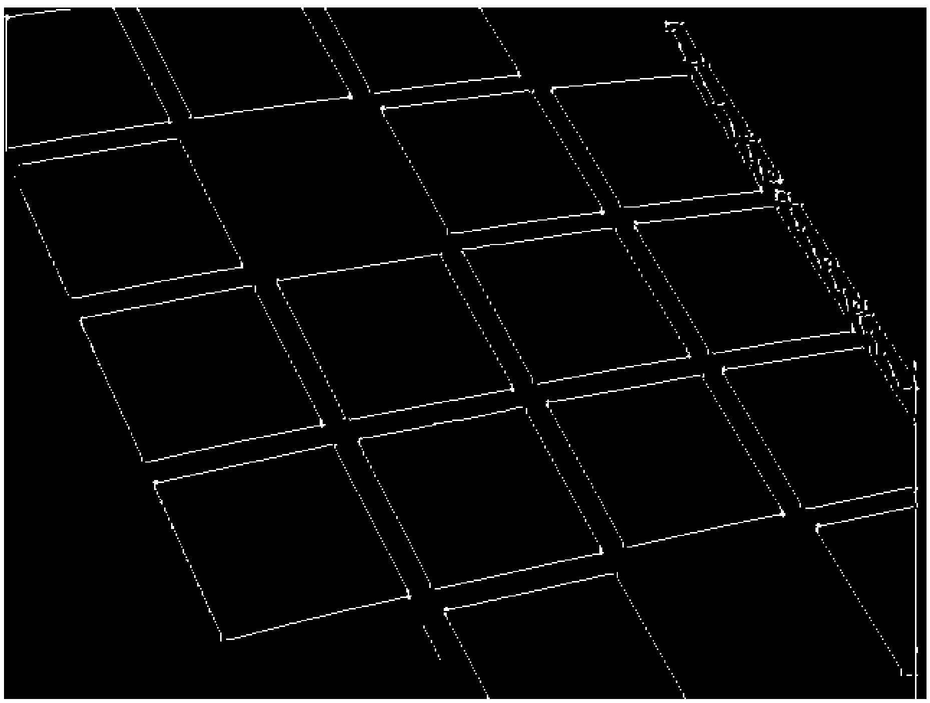


Figure 3: The edge magnitude image after thresholding.

the first test image. The lines in Hough space and perpendicular to their normal representation because of the order in which I indexed the matrix. Sinusoidal lines can be seen in this image, along with distinct peaks, which indicate where lines appaer in the original image.

Testing Hough Lines and Line Segments:
Testing the Hough lines and line segments involved visualizing the lines on each

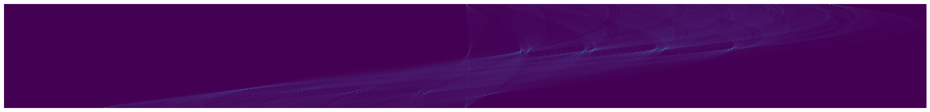


Figure 4: The Hough transform accumulator matrix for the first test image.

image and analyzing how accurately they overlap with the edges on the image. Figure 5 shows the final results of visualizing the lines on each test image using the Hough transform.

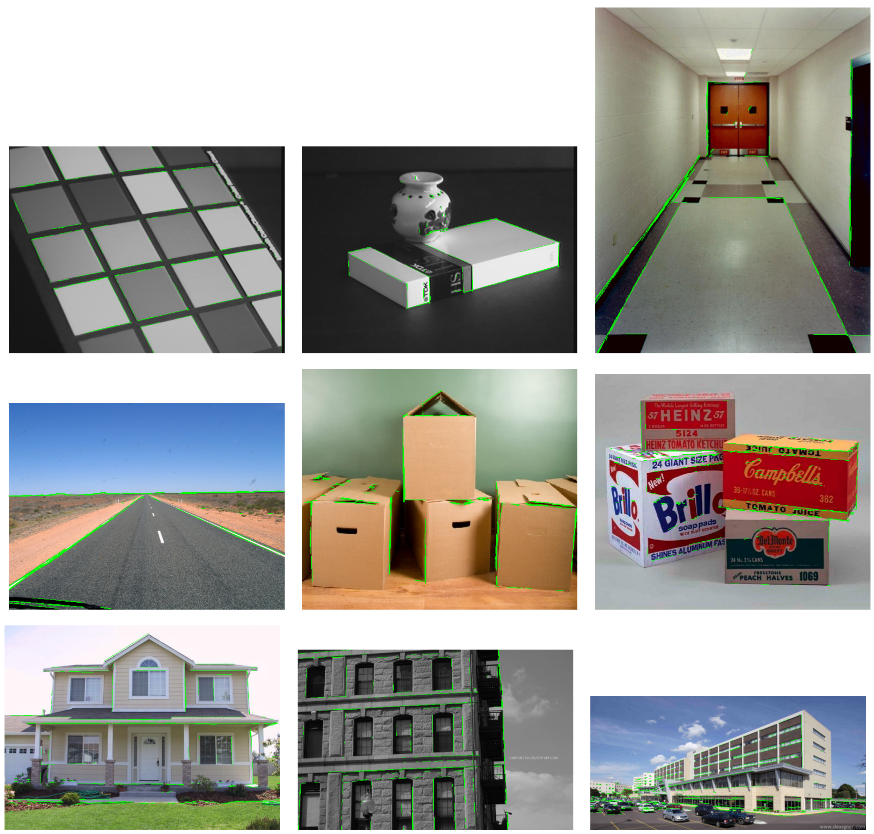


Figure 5: Test images 1-9 with Hough line segments drawn on.

Determining Hough line segments was the most troublesome. Since the Hough transform used in this homework is only meant to detect straight lines, it can be difficult to draw lines on objects where the edges are rounded or more chaotic. I found that requiring a minimum line segment length improved performance by removing some extraneous lines.

Since most of the parameters were fixed for this assignment, the main parameters I experimented with was the edge magnitude threshold. I found that a threshold of 10.0 work adequately for all of the test images. Some of the images, such as the image with the house, had better edge detection performance with a lower threshold because the gradients were not as steep.