Edge Detection

Jose Luciano

April 24, 2022

Abstract

My goal in this lab is to implement an edge detection algorithm. The algorithm was implemented by applying Sobel's filters to the greyscale image to calculate the magnitude of the gradient. The algorithm was able to successfully detect edges.

Technical Discussion

The function find_edges is able to take a greyscale image as an input and return a binary image consisting of intensity values 255 and 0. The function calls upon two other functions named gradient_magnitude and spatial_filter.

The function gradient_magnitude calls upon the function spatial_filter and uses Sobel's filters as the input for spatial filters. Sobel's filters are 3x3 matrices that are used to determine the edges of an image and have the following values:

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

The function spatial_filter iterates through each pixel of the given image and applies Sobel's filters to each pixel. The values of the matrix to be returned were calculated using the following formula:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x+s, y+t)$$

where w is the spatial filter and f is the image. The zero-padding technique was used to handle the edges of the image (matrix form). The zero-padding technique is a method that enlarges the matrix by adding a layer of zeros around the matrix, and its purpose is to make the image's size a power of two. That is to say, our nxn image will result in a (n+2z) x (n+2z) image where n is the original size of the image and 2z is the number of rows and columns to be added. The matrices that function spatial_filter returns are used in the function gradient_magnitude to calculate the magnitude of the gradient by applying the following formula to each pixel :

$$|G| = \sqrt{G_x^2 + G_y^2}$$

where G_x and G_y are the matrices returned by the function spatial_filter.

Results

The first threshold value that was used was 100 and the result can be seen in Figure 2. When comparing Figure 2 to Figure 3 there is a noticeable difference in the number of edges detected. Figure 2 has more edges detected than Figure 3. This is because the threshold value of Figure 3 is 200 which is significantly larger than that of Figure 2, so when the pixel values are being compared in function find_edges, more pixels will be assigned a pixel intensity value of 0 (black). Figure 4 is the result of using the built-in function edge, and when both Figures 2 and 3 are compared, it is evident that Figure 4 does not detect as many edges as our algorithm.

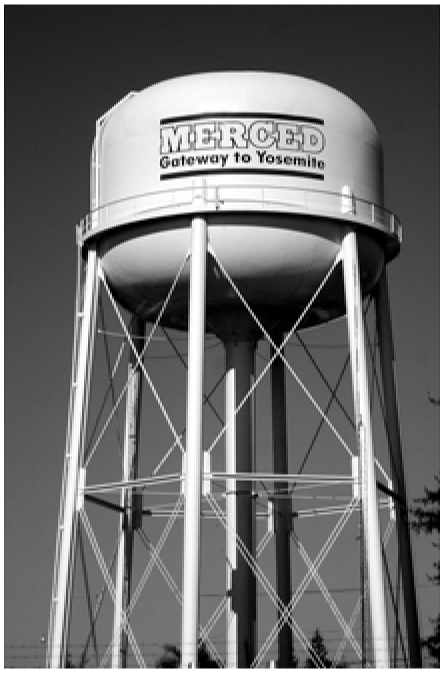

Figure 1. Original Greyscale Image

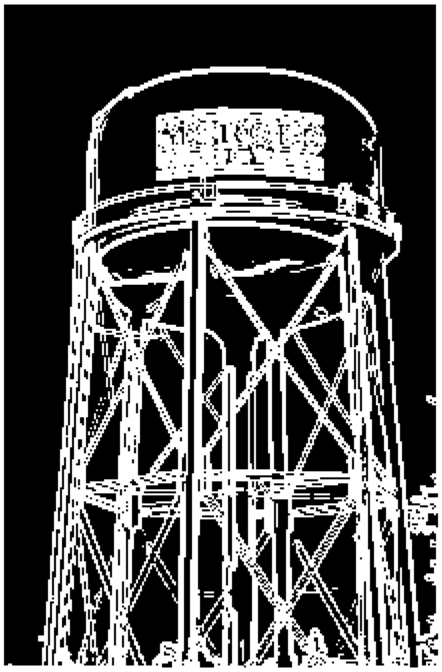

Figure 2. Edge Detected Image (Threshold Value: 100)

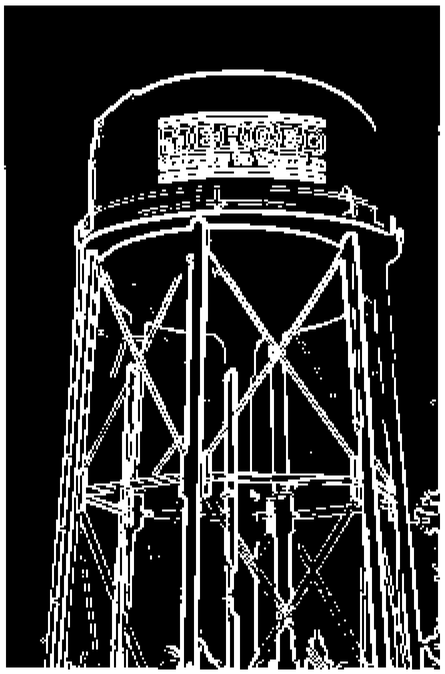
Figure 3. Edge Detected Image (Threshold Value: 200)

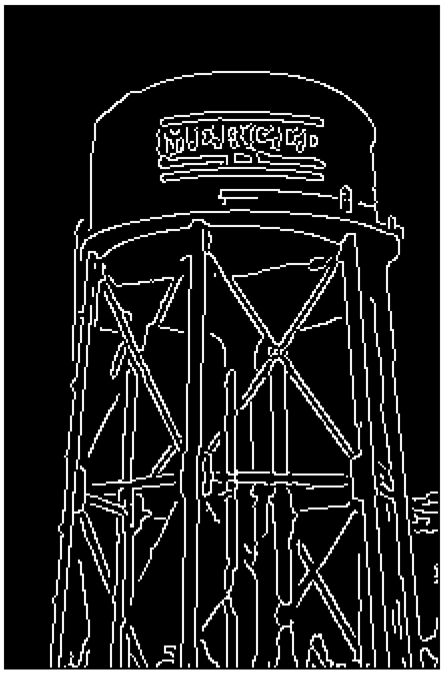
Figure 4. Canny Edge Detection Method