MP5 - Canny Edge Detection

Logan Boswell

Overview

This MP is focused on implementing Canny edge detection. In order to implement this technique, multiple python functions were written to apply Gaussian smoothing, calculating the image gradient, finding thresholds for edge processing, suppressing non-maxima, and applying thresholds and linking edges. Also one function was written that implements Canny edge detection by using the functions listed above.

Gaussian Smoothing

The first step taken in Canny edge detection is to perform Gaussian smoothing on an image. In my implementation, an image is first taken in and converted to gray. Then, a mesh grid is generated using np.meshgrid() which is used for finding the normalized kernel. After this is determined, the kernel is used to apply the smoothing to the image by using the convolve2d() function from the scipy.signal module.

Image Gradient

After I was able to smooth images with my Gaussian function, I started working on my image gradient function. In my function, the gradient in the x and y directions are calculated using the cv2.Sobel() function, and then these gradients were used to find the magnitude and angle at each pixel.

Finding Thresholds

In order to find thresholds for edge detection, the magnitude array calculated from the ImageGradient() function was converted to a histogram and the cdf was determined. To find the high threshold, I found the first value where the cdf is greater than the percentage of non-edges, which is an input to my function. Once the high threshold was determined, the low threshold was set to half of the high threshold.

Non-maxima Suppression

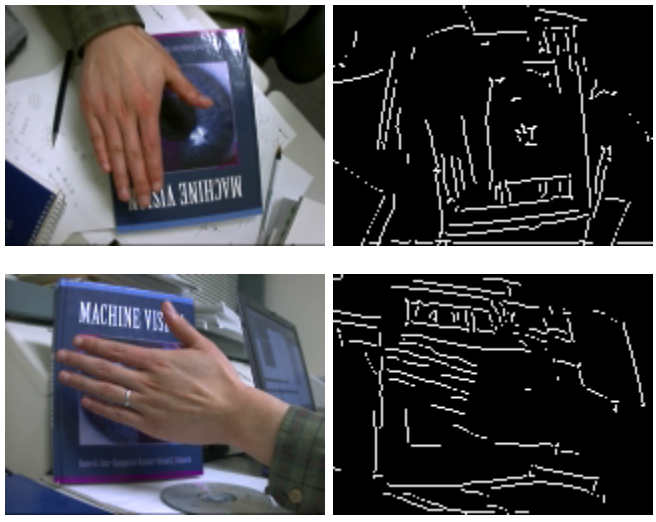
After the image gradient was found, non-maxima values for the magnitude array were suppressed by looping through each element in the magnitude array and comparing with values of certain neighbor elements. The neighbor elements were determined by looking at the angle value corresponding to the current magnitude element. There are 4 cases based on the angle that determine which neighboring elements will be compared to the current element. Only elements that are local maxima are preserved after this operation, and all other elements are set to 0.

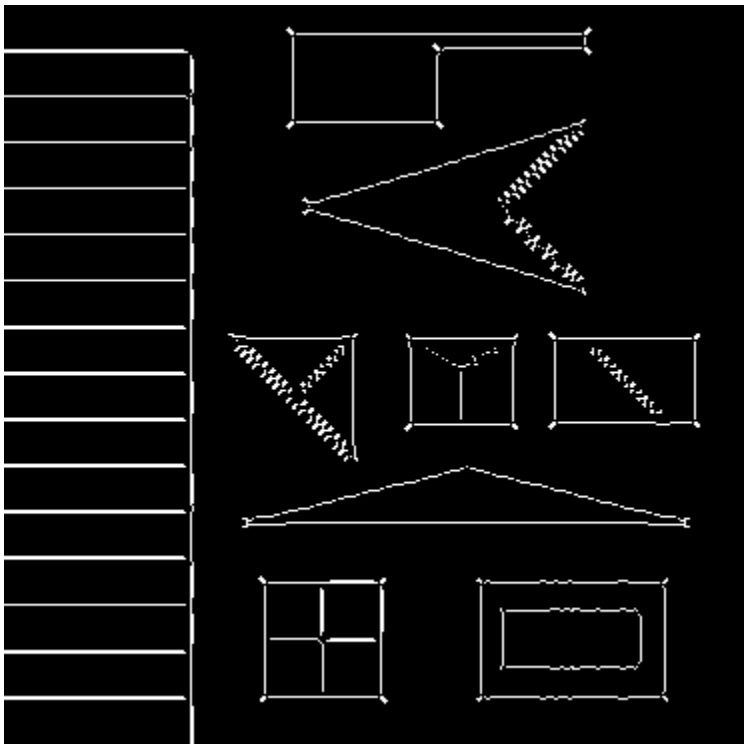
Edge Linking

With the non-maxima values suppressed, the thresholds can be applied to the magnitude array from the image gradient. Two new arrays are determined, one with magnitudes that are greater than the high threshold value (representing strong edges) and one with magnitudes that are greater than the low threshold value (representing weaker edges). From here, the edges are linked recursively, starting from the strong edges.

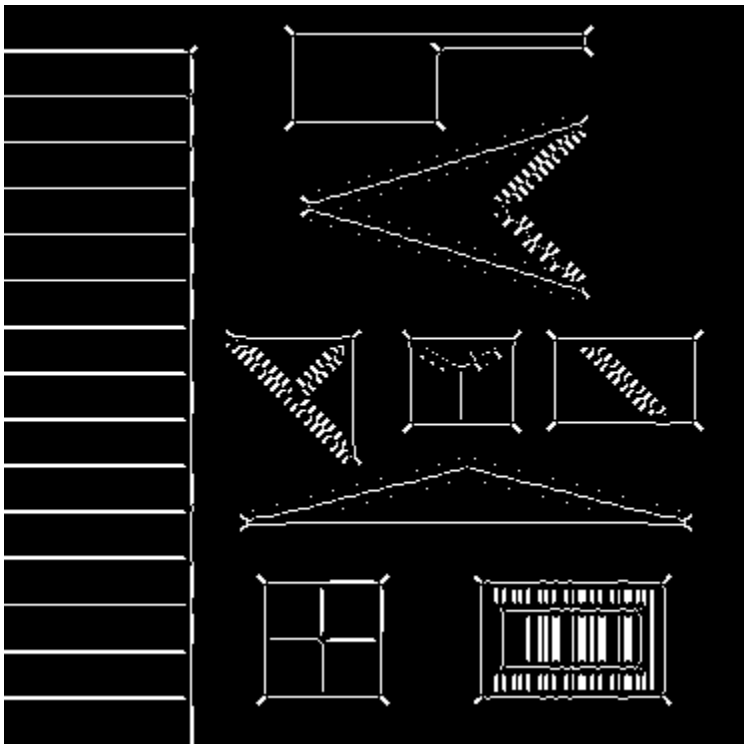
Results

Here are the testing results with the given .bmp images and N=5, sigma=1, percentageOfNonEdge=.85





Here are the testing results with the given .bmp images and N=5, sigma=1, percentageOfNonEdge=.72



Here are the testing results with the given .bmp images and N=9, sigma=2, percentageOfNonEdge=.8