Zachary Rump

ECE 4655

Fall 2017

Assignment 3 Report – Canny Edge Detector

Objective

The objective for this assignment was to write a program to perform Canny Edge Detection, which is a multi-stage algorithm used to detect edges in images. The algorithm consists of three primary stages:

1. Filtered Gradient
2. Nonmaximum Supression
3. Hysteresis thresholding

Methods

Filtered Gradient

For this implementation I used Matlab/Octave for everything. First step is to load the image and convolve with a Gaussian filter. I chose to do this in multiple steps even though the assignment says you can convolve directly with derivative of Gaussian. I take the Gaussian filtered image and apply (convolute) the Sobel or Prewitt operators to find the X and Y components of the gradient at each point. Then, the magnitude and orientation of the gradient is stored so that nonmaximum suppression can be performed.

Nonmaximum Supression

In nonmaximum supression, we look at each pixel in the image and find the orientation in a set S = { 0, 45, 90, 135 } that is closest to the orientation of the gradient at that pixel (found in previous step). Once the closest orientation is chosen, the neighbors along the line at that orientation are inspected. E.g. if orientation is 45 we look at the NW and SE neighbor of the current pixel. If the gradient magnitude of the neighbors is greater than the value at the current pixel, we set current pixel intensity to 0 (black), otherwise we leave it as is. An intuitive way of thinking about this is just that we are looking at a local region of pixels (along the gradient), and anything that is not a local maxima in that region (brightest pixels) becomes a global minima (darkest pixels in image).

Hysteresis thresholding

The result of nonmaximum supression should be an image with less detail but with edges more clearly defined (higher contrast, large gradient magnitude). The next step is to clearly connect and define the edges by removing all other detail from image, and connecting edge segments together to form curves. This step takes a couple parameters T_h (threshold high) and T_l (threshold low), which are the pixel intensity values above and below which a pixel should be changed to white or black. For intensity values between T_h and T_l, the intensity is chosen based on whether or not there are local maxima (I(x,y) > T_l) as pixel neighbors.

Results

Figure 1.1 Original input

Figure 1.2 After applying edge detection algorithm. This result uses sigma = 2, T_low = 1, T_high = 50.

Figure 2.1 Original image of building

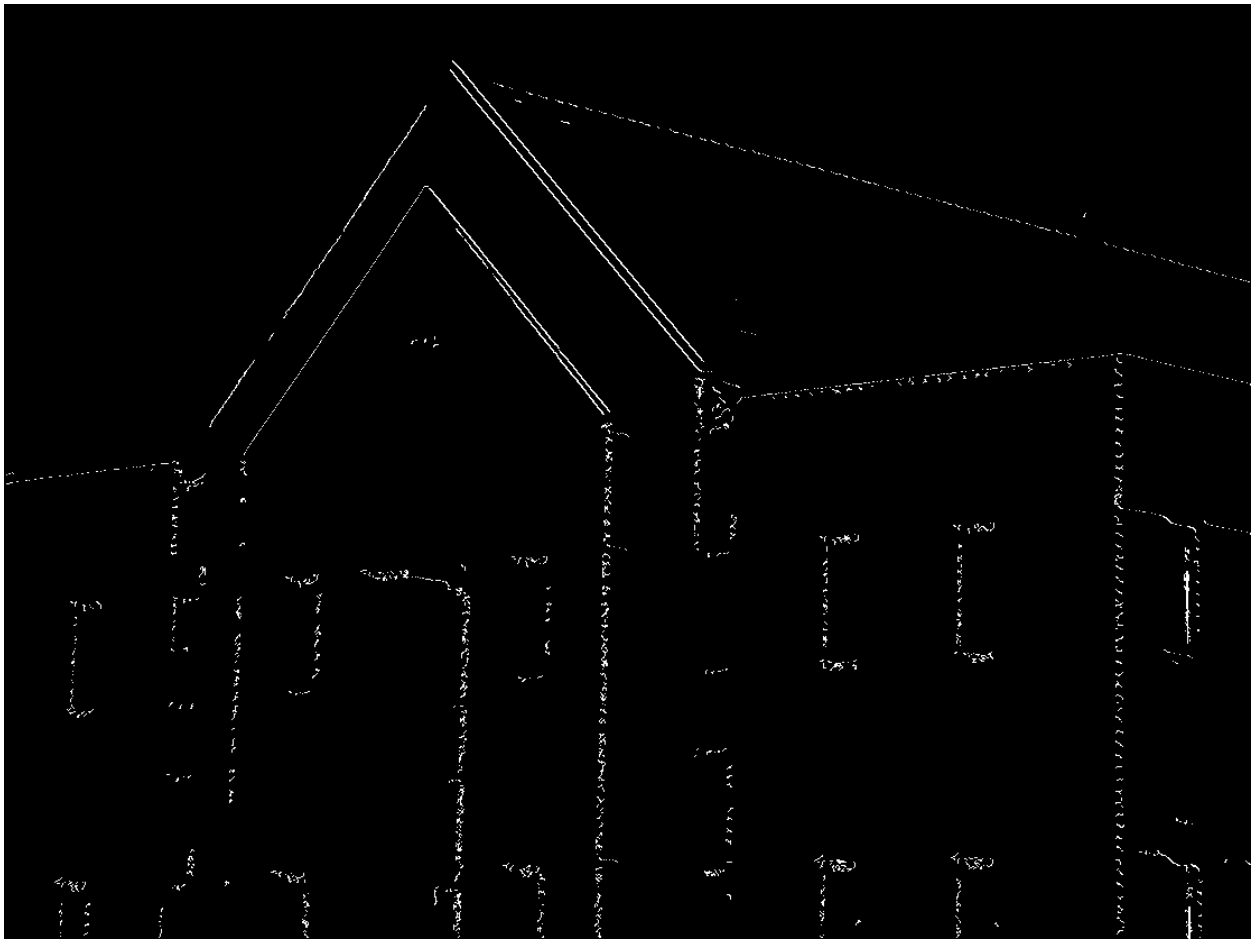
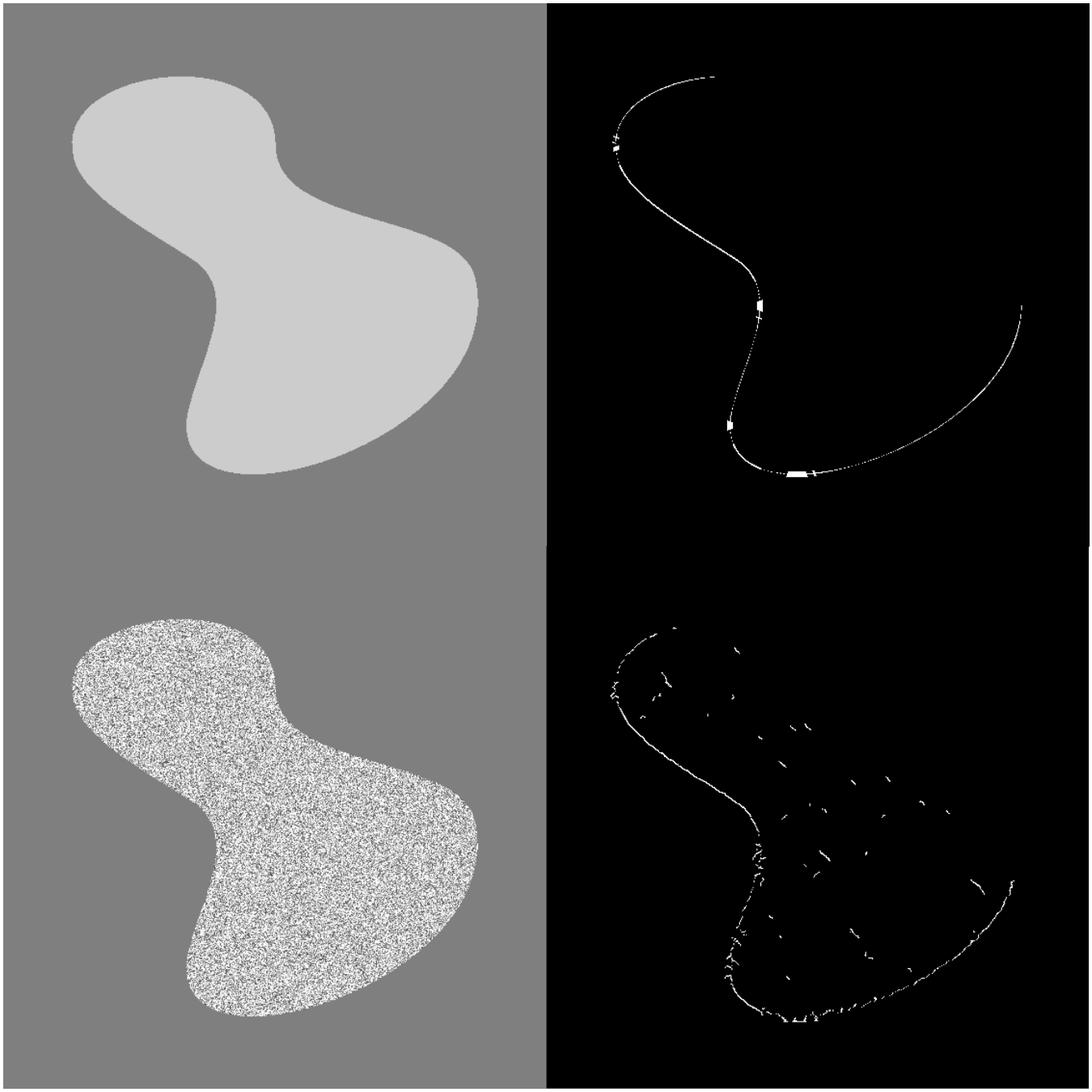
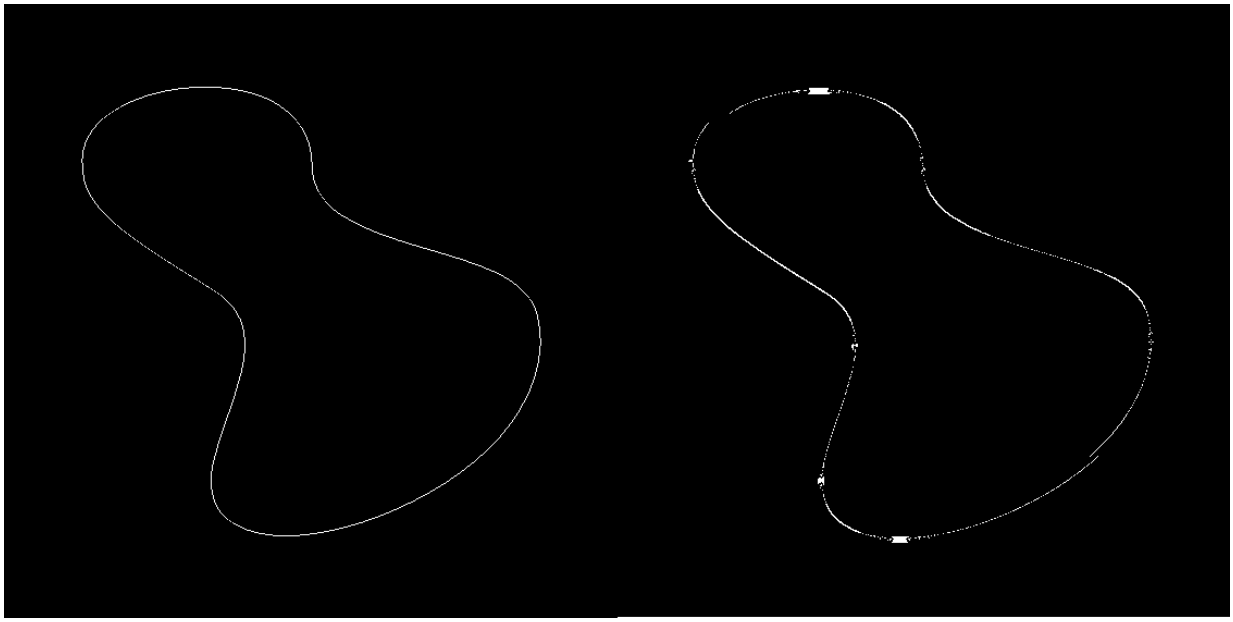
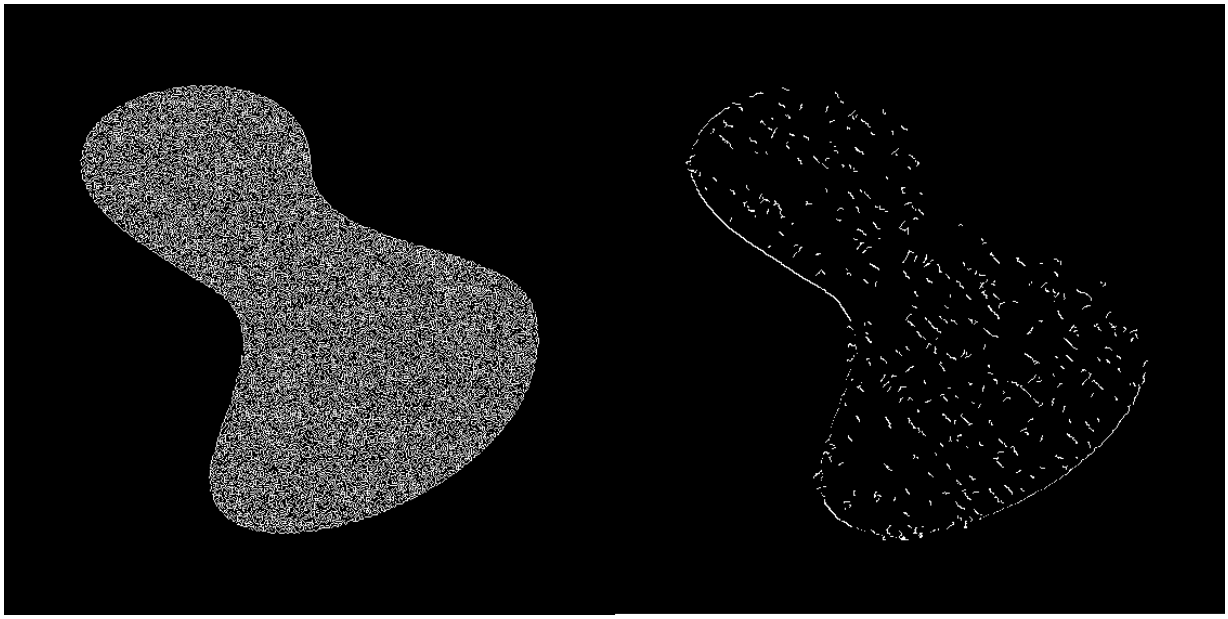
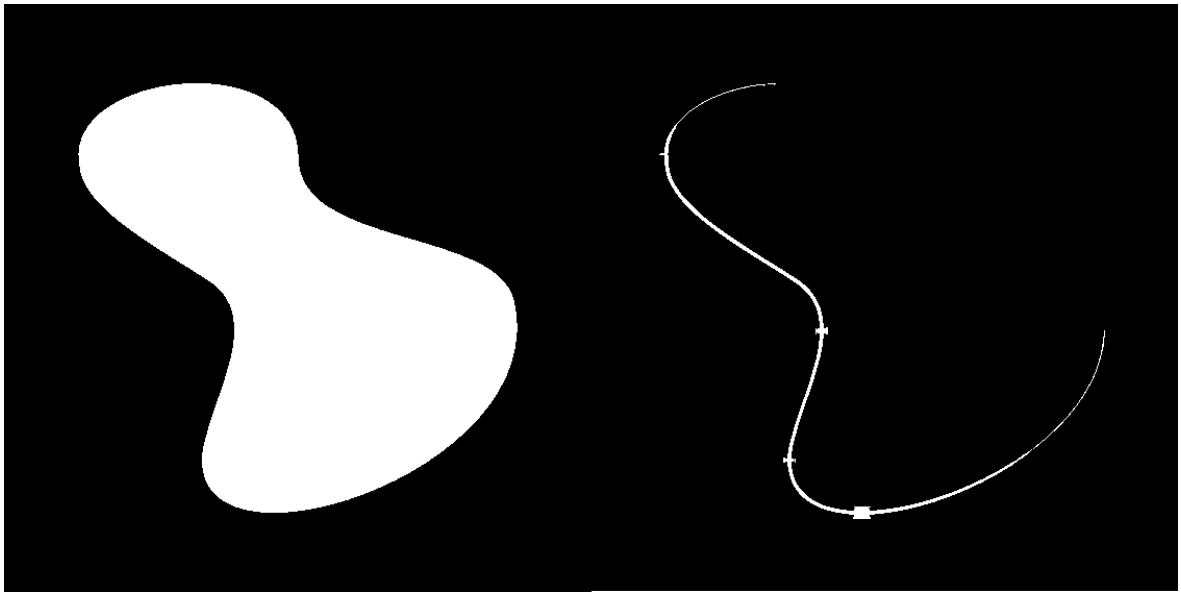


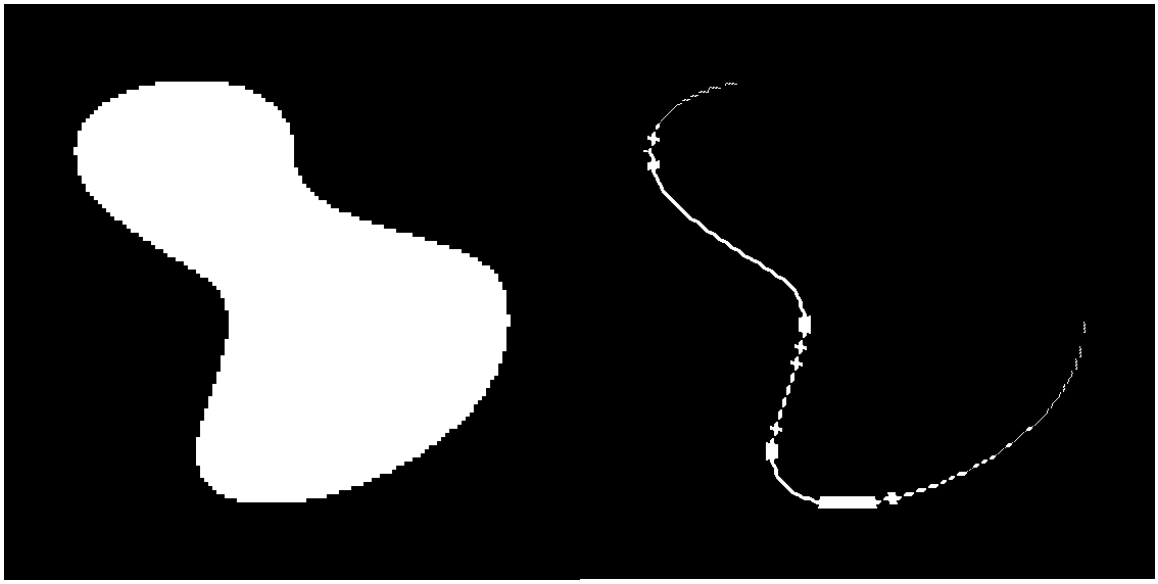
Figure 2.2 Shows building image after edge detection. This result uses sigma=2, T_low = 50, and T_high = 81. This removes most of the detail from the image e.g. the lines of the roof tile.











Figures 3.x Figure 1001 shown in the images above. Originals are on the left and the edge detection is on the right. All of these images used sigma = 2.5;
T_h = 50;
T_l = 2;

Remaining Issues / Potential Solutions

- I’m using the Sobel operator in the nonmaximum suppression step to find the gradient. Could use the Prewitt operator instead and/or add a parameter to the function to decide which one to use.
- Could improve the hysteresis thresholding step by finding a better way to find connected regions (local maxima)