Islamic University of Technology
Department of Computer Science

Canny Edge Detection Algorithm

Samnun Azfar
20041130
CSE4733 Digital Image Processing

March 25, 2025

Contents

Chapter 1

Canny Edge Detection

1.1 Motivation for Single Pixel Thick Edge Response

One of the main objectives in developing the canny edge detection algorithm was to generate single-pixel thick edges that were closer to the center of the actual edge. Some of the following reasons motivated the necessity of a single pixel thick edge response:

- **Accurate Edge Localization:** Ensures that the detected boundary aligns perfectly with the actual edge.
- **Reduction of False positives:** A single pixel thick edge response for one edge ensures that there isn't multiple responses for a single edge in the original image.

1.2 Objectives of Canny

Some of the objectives [1] for the development of the Canny Edge Detector are:

- Less Error Rate
- Localized Edge Points
- A single point for each actual edge

1.3 Algorithm Description

The steps of the algorithm are as follows[2][1]:

1. The canny algorithm begins by performing smoothing by a Gaussian kernel:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.1}$$

with the original image through convolution:

$$f(x,y) = G(x,y) * I(x,y) \tag{1.2}$$

2. After the blurring is performed, *Gradient Angles* and *Gradient Magnitudes* of the blurred image is found. Mathematically this operation is defined as:

$$Grad_x(x,y) = \frac{\delta}{\delta x} f(x,y) \tag{1.3}$$

and:

$$Grad_y(x,y) = \frac{\delta}{\delta y} f(x,y) \tag{1.4}$$

After which the gradient magnitudes and the angles were calculated:

$$Magnitude = \sqrt{Grad_x^2 + Grad_y^2} \tag{1.5}$$

$$Angle = \arctan \frac{Grad_y}{Grad_x} \tag{1.6}$$

3. Note that the Magnitude and Angle Arrays contain the Gradient Magnitudes and the Gradient Angles for all the points of the image. On these two arrays, the next operation known as the *Non Maxima Suppression* is performed. The gradient response of an image creates a ridge like region in the local maxima. The surrounding region of the maxima have high gradient response even though they are not the actual edge. Thus to narrow down the edge *Non Maxima Suppression is to be performed*. The steps of *Non Maxima Suppression* are:

For each pixel in $Angle(x,y)$**:**

(a) Categorize the gradient angle response of one pixel to one of 4 categories according to 1.1.

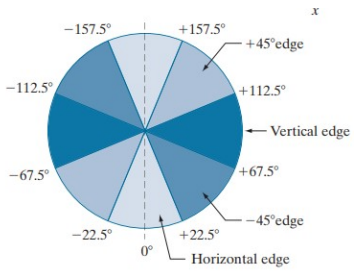


Figure 1.1: Angle division For Non Maxima Suppresion

(b) After categorization is done (horizontal,vertical, 45 degree and -45 degree), go to the current **pixel** and denote its gradient magnitude as K. Now the algorithm finds the neighbor of the **current pixel** along the direction of the categorization. For example if my current pixels gradient angle is categorized as *horizontal* then the neighbors would be $(x+1, y)$ and $(x-1, y)$ considering x as the horizontal axis. Now if K is less than any of the gradient magnitude of its neighbors along the direction then set the result of *Non Maxima Suppression* as 0 , else set it to $Magnitude(x, y)$.

4. After Non Maxima Suppression is done, we will now have to do thresholding. The thresholding process generates two boolean arrays, G_L and G_h with the thresholds T_l and T_h.

$$G_l(x,y) = \begin{cases} True & NonMaximaResult(x,y) > Tl \\ False & otherwise \end{cases} \tag{1.7}$$

$$G_h(x,y) = \begin{cases} True & NonMaximaResult(x,y) > Th \\ False & otherwise \end{cases} \tag{1.8}$$

$$G_l(x,y) = G_l(x,y) - G_h(x,y) \tag{1.9}$$

5. Now we use the two thresholding arrays to generate the final output. To generate the final output we do:

   For each pixel in G_h:

   - Include the center pixel in the result if True in G_h.
   - Include any of the 4/8 connected neighbor if that neighbor if that neighbor is True in G_l

1.4 Intermediate Steps Visually

:

Original Image

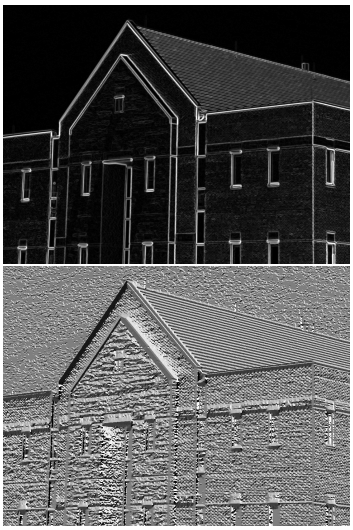
This is the original image to apply canny edge detection algorithm.

Smoothed Image

A gaussian kernel with size $(7, 7)$ and $\sigma = 1.5$ is applied on this image.

Gradient magnitude and angles

The gradient magnitudes(top figure) and angles(bottom figure) of the smoothed image is found. The magnitudes and angles exist for each of the pixels.

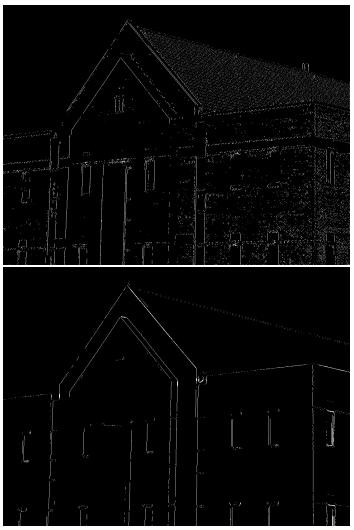


Non Maxima Supression

Using the gradient magnitudes and the angles the non maximum points were suppressed.

Thresholding operation

The thresholding on the suppressed image results in two arrays G_h(bottom) generated with a threshold $T_h = 150$ and G_l(top) generated with a threshold of $T_l = 40$.



Final Output

The final output is computed using G_h and G_l through the algorithm described in 1.3

Figure 1.2: Result of Canny Edge Detection Algorithm

The images were generated using the code from [3] and the original image used here was collected from [1].

1.5 Analysis and Comparison

In this section we compare the output of the canny edge detection algorithm with some of the other existing edge detection methods.

- The gradient computation provides edge response for pixels which arent the boundary of the object i.e. the roof of the house in the image.
- The Marr-Hildreth algorithm gives a clearer response than the gradient, but the canny algorithm is superior in generating one pixel thick edges for the objects present.
- The G_l image in the canny algorithm intermediate step contains a lot of egde responses on the roof of the house. But the thresholding operation successfully removes all the pixels on the roof utilising G_h. This process is known as hysteresis[2].

Figure 1.3: Canny

Figure 1.4: Gradient

Figure 1.5: Marr-Hildreth

Bibliography

[1] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing, Global Edition*. en. 4th ed. London, England: Pearson Education, Sept. 2017 (pages 1, 2, 5).

[2] *Canny edge detector*. URL: https://en.wikipedia.org/wiki/Canny_edge_detector (pages 2, 6).

[3] Samnun Azfar. *CSE-4734 DIP - Canny Assignment*. https://github.com/azfarus/CSE-4734-DIP/tree/main/Canny_Assignment. GitHub repository. 2024. URL: https://github.com/azfarus/CSE-4734-DIP/tree/main/Canny_Assignment (page 5).