EECS 332 Introduction to Computer Vision

Machine Problem 5: Canny Edge Detector

Name: Runlong Lin   ID:3075255

1. Introduction

Edge detection is a very important technique in computer vision. There are a lot of algorithms and detectors for edge detection. Typically, canny edge detection is the most popular methods with good performance. It's developed by John F.Canny in 1986. Canny detector consists of several smart algorithms, including nonmaxima suppressing and edge linking, which are the most essential and brilliant ideas among them. In Machine Problem 5, we will implement canny edge detector from scratch.

2. Algorithms description

Canny edge detector is mainly developed by the following steps: Gauss Smoothing, Image Gradient calculation, Nonmaxima Supressing, Thresholding and edge linking.

- **Gaussian Smoothing**

Here is the Matlab function for Gauss Smoothing:

```
 function[img_out]=GaussSmoothing(N,sigma,img_input)
function [T] = GaussKernel(N, sigma)
```

where N and sigma is the parameter for Gaussain filter, T is the Gaussian operator.

Gauss Smoothing is also called Gauss Blur, which means that it can blur the image with Gaussian function. It's widely used in image processing for reducing the noise. Figure 1. is the result of Gaussian Smoothing.

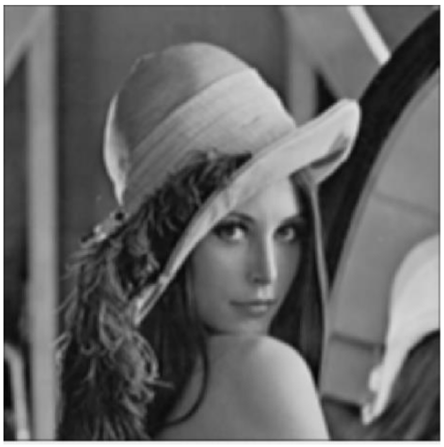

Fig.1. Gaussian Smoothing image

- **Image Gradient Calculation**

In order to implement edge detection, we should find the "change" between pixels. We can achieve the goal by calculating the image gradient. There are lots of operators for calculating the image gradient. In MP5, we use Sobel operator to calculate the magnitude and direction of the edge map. Here is the function:

```
function[mag,theta]=ImageGradient(img_input)
```

where mag is the magnitude matrix for the edge map and theta is the direction matrix.

Figure 2. is the magnitude image of Gaussian version of original image.

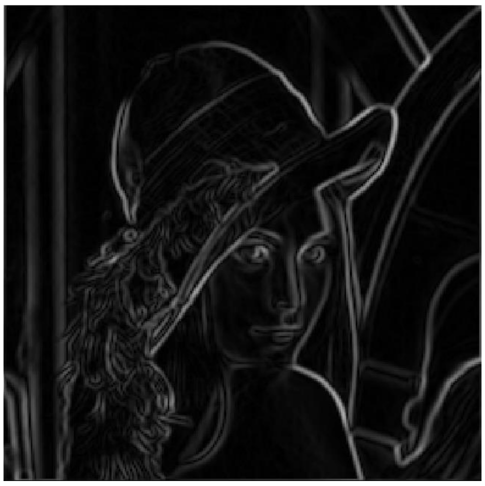

Fig.2. Gradient image

As we obtain the magnitude image, we can roughly see the edge of the original image. But we need further adjustment.

- **Non-maximum Suppressing**

As we want to detect the edge pixel precisely, we can find the local maximum in of image gradient. This is the core idea of Non-maximum suppressing. Here is the function:

```
function[mag_out]=NonMaximaSupress(mag, theta)
```

where mag is the magnitude matrix for the edge map, theta is the direction matrix and mag_out is the image gradient after non-maximum suppressing.

In this function, we divided the direction into four kinds based on their theta value. Then we scan the pixels. If the pixel is not the local maximum, we set it as zero. Figure 3 is the result of non-maximum suppressing. After this operation, we can obtain quite thin edges.

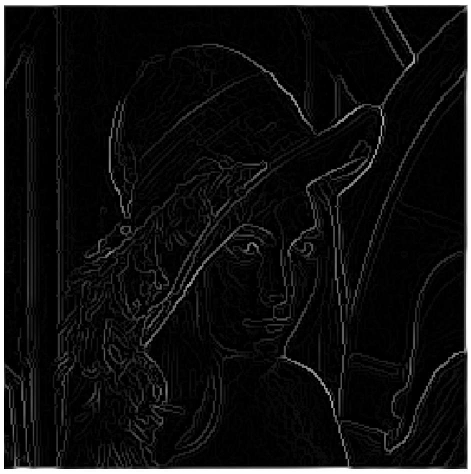

Fig.3. Non-maximum suppressing image

- **Find Threshold**

To get rid of the non-edge pixel in figure x, we can use thresholding method. Meanwhile we will use histogram method to help us find the threshold. Here is the prototype of the function.

```
function[th_low,th_high]=FindThreshold(mag, percentageOfNonEdge)
```

where th_low and th_high is the low threshold and high threshold. Here we simply set the low threshold as the half of the high one. Meanwhile, we use the percentage of the non-edge pixel as an input parameter to help us find the threshold. If we set this value higher, we will get less edge pixel. But we may also ignore the real edge pixel if the value is too high. Setting the percentage of non-edge as 0.88, we can get two thresholds. Figure 4 is the image gradient with low threshold filtering and Figure 5 is the high threshold version.

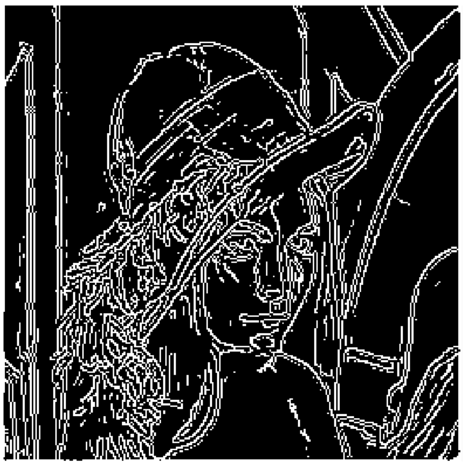
Fig.4. weak edge image

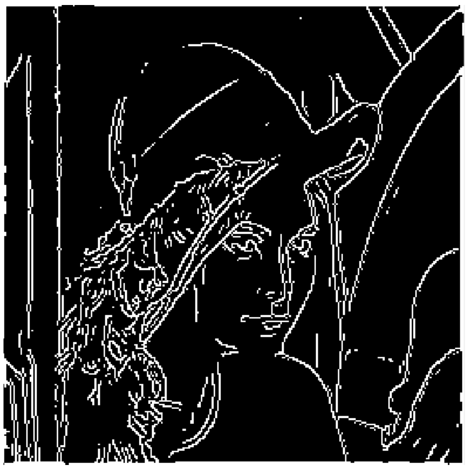
Fig.5. strong edge image

As the figures show, we can see more details in the image with low threshold. But there are also more pixels which are not belonging to edge pixels.

- **Edge Linking**

We have already obtained the strong edge and weak edge in the last step. In the weak edge image, there are lots of non-edge pixel. We can use edge linking to get rid of this pixel. Here we choose the pixels between high and low threshold. Then we check the neighbors of these pixels. If there is a neighbor pixel belongs to the strong edge, then we can assign this pixel as the strong edge. By doing so, we can connect the strong edge and edge pixels in weak edge image. Figure 6 is the result of edge linking, which is also our final result of canny edge detection.

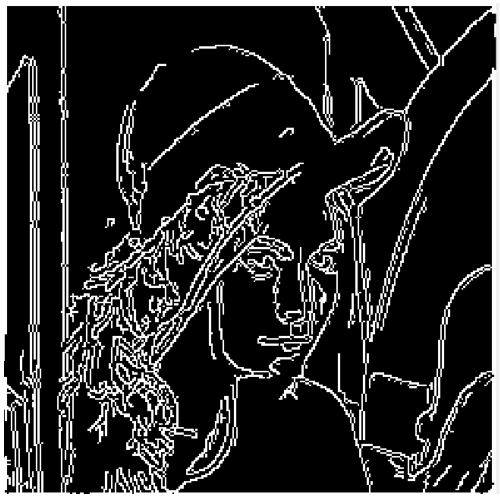
Fig.6. edge linking image

3. Result Analysis

In the 2nd part of this report, we also shown the part of the result image of canny edge detector. In this part, we will focus on the comparison of result using different parameters and different detectors.

Here are the results with canny detectors (with Gaussian parameters **N=3, sigma=3 and percentageOfNonEdge = 0.88**):

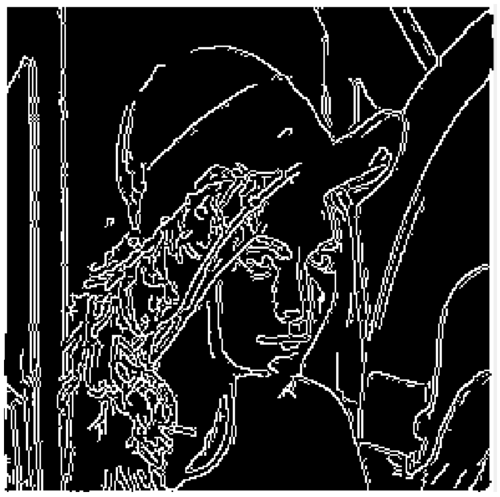
Fig.7. lena.bmp

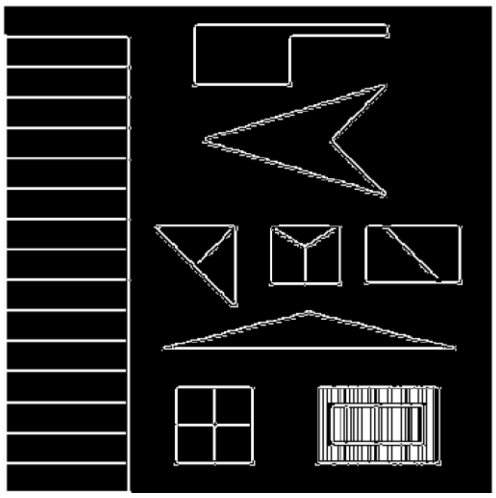
Fig.8. test.bmp

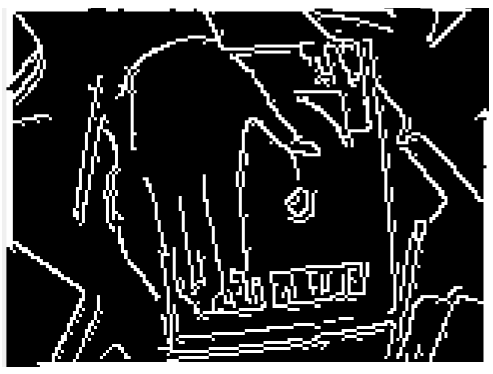
Fig.9. joy.bmp

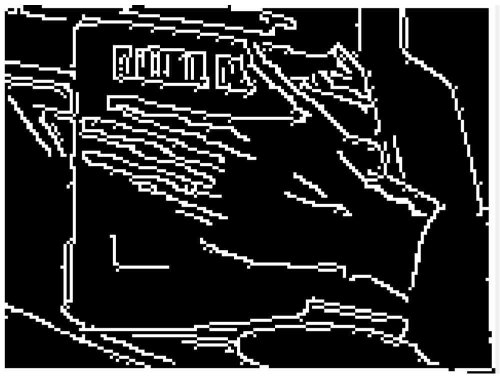
Fig.10. pointer.bmp

We also use another picture for testing:

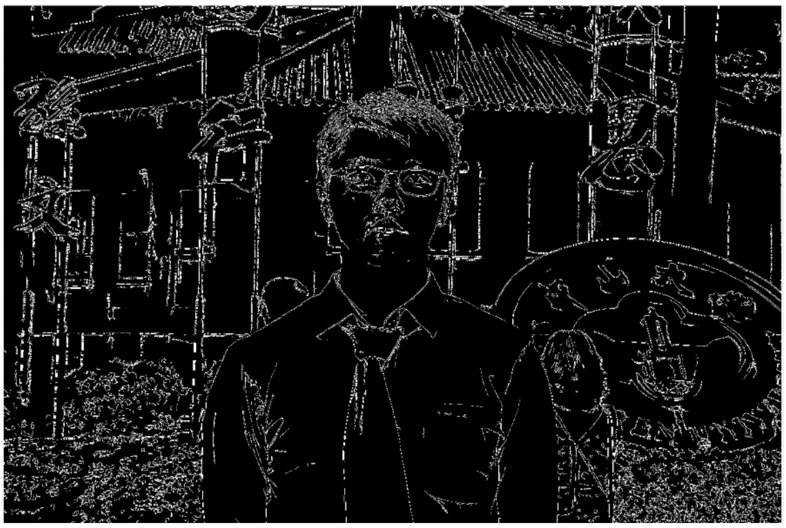

Fig.11. tesing image

Analysis: We can obtain well performanced result with canny detector.

- **Comparison with different percentageOfNonEdge (percentageOfNonEdge=0.80,0.85,0.88,0.90)**

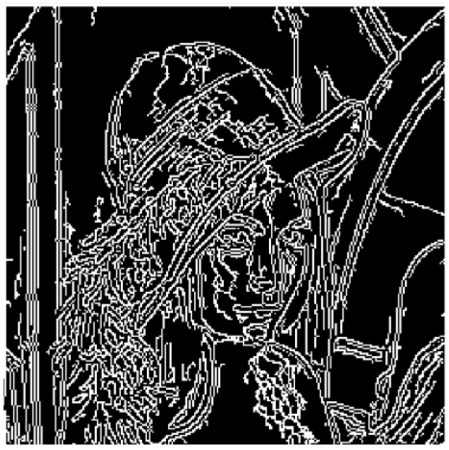

Fig.12. p=0.80

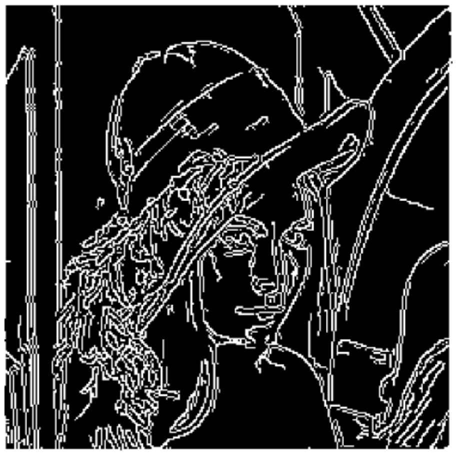

Fig.13. p=0.85

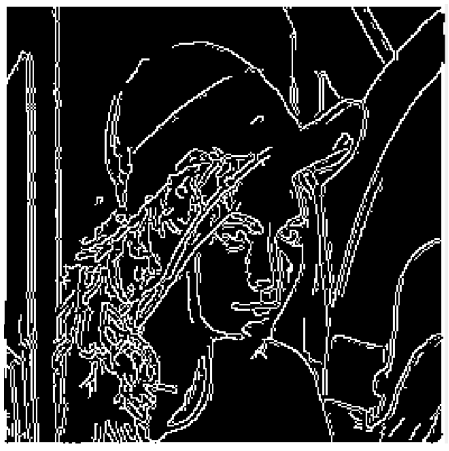

Fig.14. p=0.88

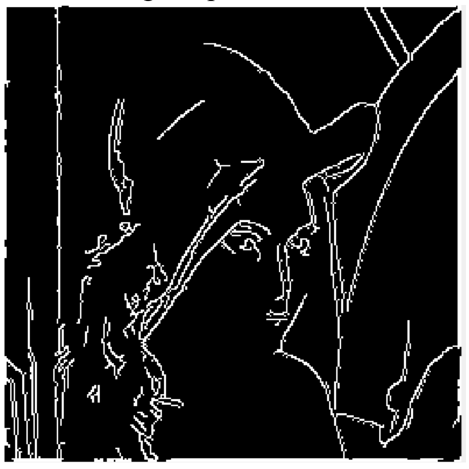

Fig.15. p=0.90

Analysis: With higher percentage of non-edge, we can get rid of some non-edge pixel and obtain a clear edge. But we may also lose some detailed edge pixels.

- **Comparison with different N (N=3,5,7)**

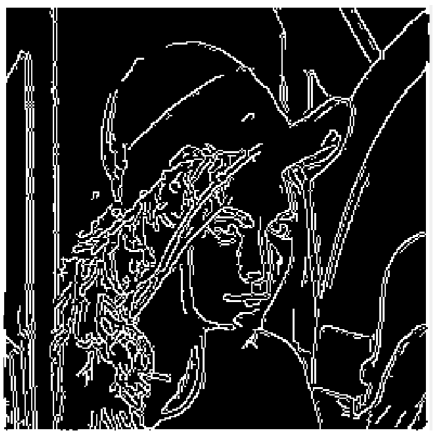
Fig.16. N=3

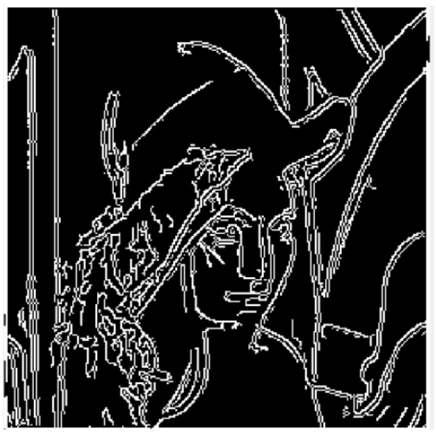
Fig.17. N=5

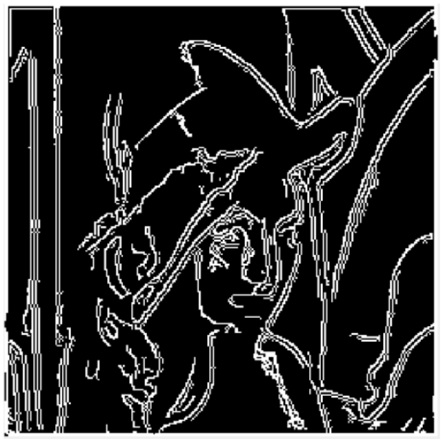
Fig.19. N=7

Analysis: With larger sizes of Gaussian operator, our image becomes more obscure. This is determined by the Gaussian function itself. With a larger size filter, more pixels will be considered into the calculation of the mean value.

- **Comparison of different sigma (sigma=3,5,7 N=3 percentageOfNonEdge=0.88)**

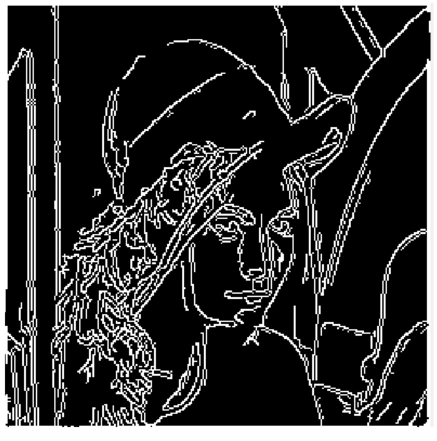
Fig.20. sigma=3

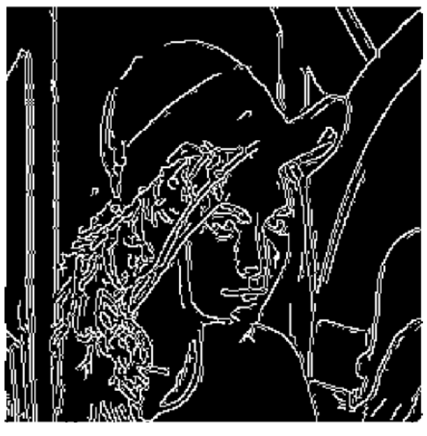
Fig.21. sigma=5

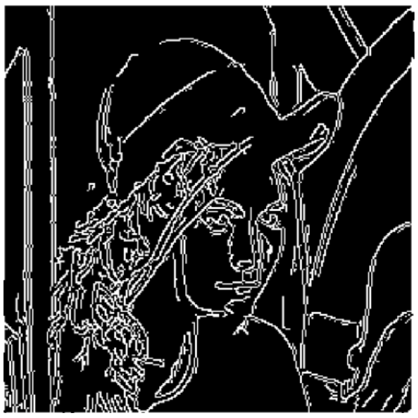

Fig.22. sigma=7

Analysis: It seems that different has no obvious effects on the image.

- **Comparison of different detectors (detectors=sobel, Roberts, zerocross, canny)**

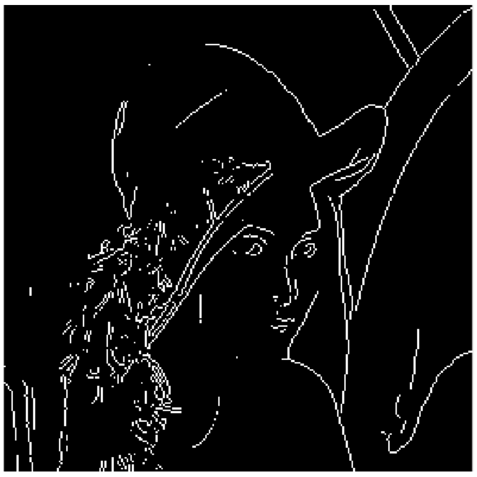

Fig.23. sobel detector

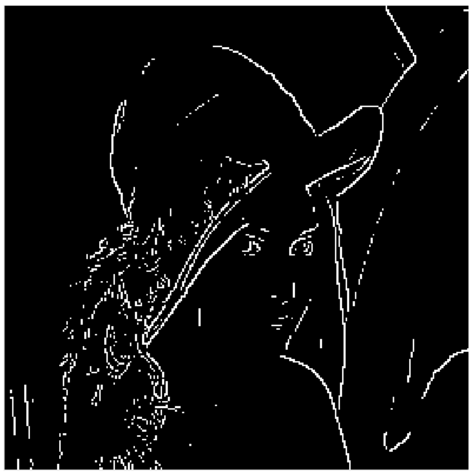

Fig.24. Robert detector

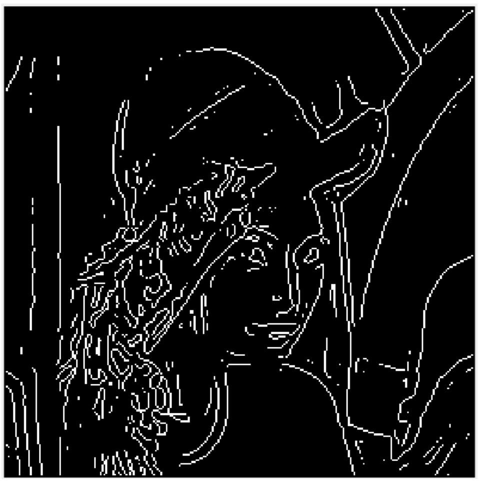

Fig.25. zerocross detector

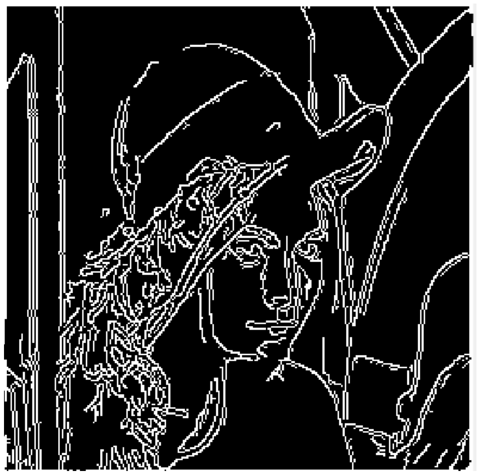

Fig.26. canny detector