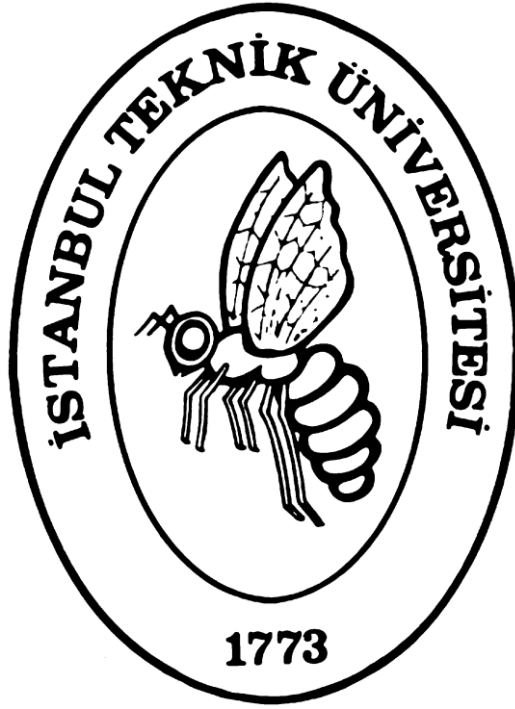


BLG477E



Multimedia Computing

Homework 01

STUDENT NAME : HÜSEYİN ERDOĞAN
STUDENT NO : 040100054

CRN : 22539

1 Environment

I used Microsoft Visual Studio 2013 and OpenCV 2.4.11 for the homework. I used Microsoft Visual Studio in Debug Mode for Win32. After running the program it wants an image's name from the user. This image must be in the "images" folder inside the project folder. After the program wants low and high Thresholding values from the user for Hysteresis Thresholding function. Then, program show 7 image to user which are listed at below.

- Original Image
- Greyscale Image
- Smoothing Image
- Gradient Image
- Non Maximum Suppression Image
- Hysteresis Thresholding Image
- Canny Image

2 Explanation of the Canny Edge Detection Function Steps

2.1 Converting to Grayscale

In this part, I controlled channel size of image, then if channel size is different then 1, I changed image to Greyscale. I use a formula that is shown at below, for Greyscale function.

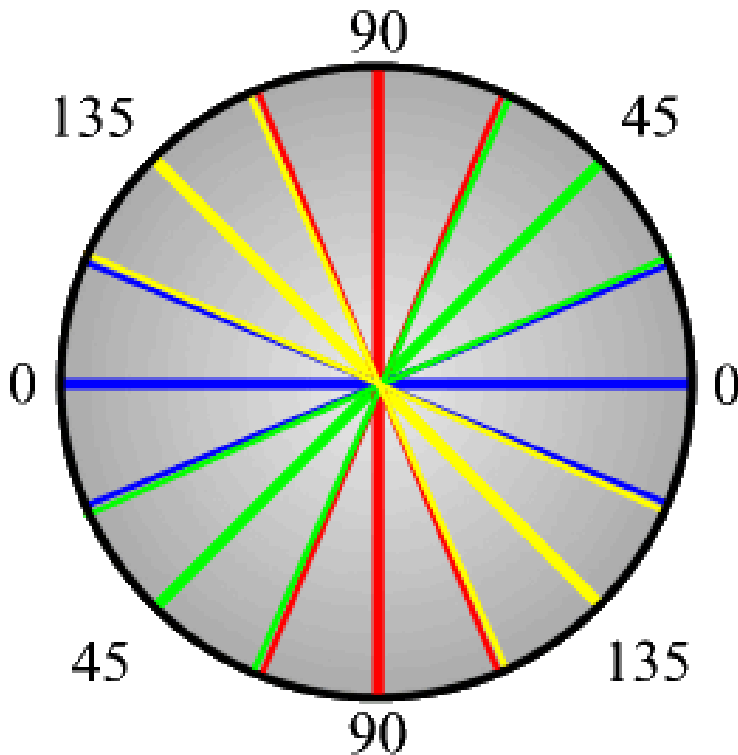
$$\text{Greyscale Pixel Value} = \text{Red Channel} * 0.299 + \text{Green Channel} * 0.587 + \text{Blue Channel} * 0.114 + 0.5$$

2.2 Smoothing

For smoothing, I used a 3*3 sized Gaussian filter. For the borders of the image I assign "0" to overflowing pixels.

2.3 Gradient Finding

In this part, I used Sobel filters. I assign Gradient values of image to a new image. I also store and group angles in a vector for next step. For grouping angles I used wheel that is shown at below.



2.4 Non-maximum Suppression

In this step, I suppress the non-maximum pixels along the norm of edge using the edge direction information. Edges become thinner after non-maximum suppression. Controls for pixel are shown below.

0	1	2
3	4	5
6	7	8

Indexes of Control Kernel

- If angle is 0, then 3 and 5 are controlled with 4.
- If angle is 45, then 2 and 6 are controlled with 4.
- If angle is 90, then 1 and 7 are controlled with 4.
- If angle is 135, then 0 and 8 are controlled with 4.

2.5 Hysteresis Thresholding

I combined Double Thresholding and Edge Tracking by Hysteresis steps in this single step. I used this pixel value controls which are shown at below, for discarding or keeping value of the pixel.

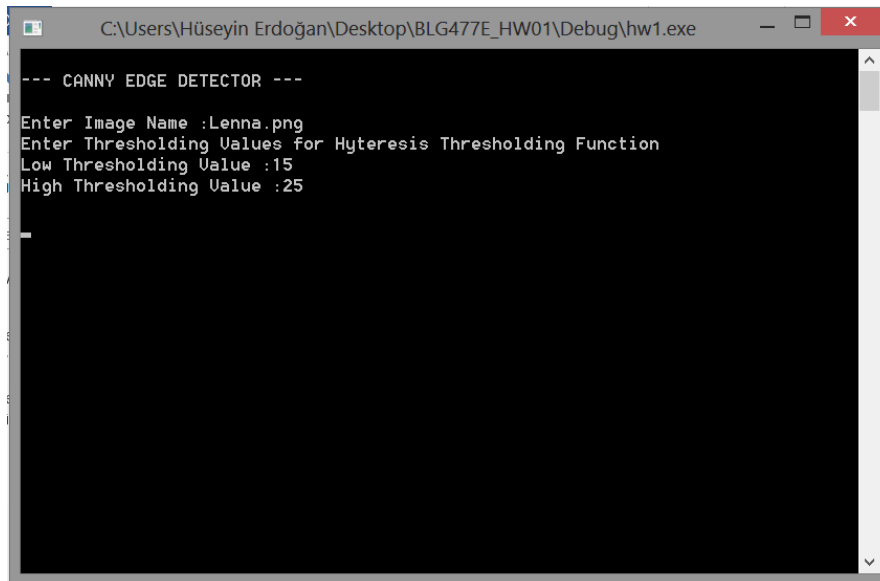
- If pixel (x, y) has gradient magnitude less than t_{low} discard the edge (write out black).
- If pixel (x, y) has gradient magnitude greater than t_{high} keep the edge (write out white).

- If pixel (x, y) has gradient magnitude between t_{low} and t_{high} and any of its neighbours in a 3×3 region around it have gradient magnitudes greater than t_{high} , keep the edge (write out white).
- If none of pixel (x, y) 's neighbours have high gradient magnitudes but at least one falls between t_{low} and t_{high} , search the 5×5 region to see if any of these pixels have a magnitude greater than t_{high} . If so, keep the edge (write out white).
- Else, discard the edge (write out black).

3 Results of the Canny Function on Lenna.png Image

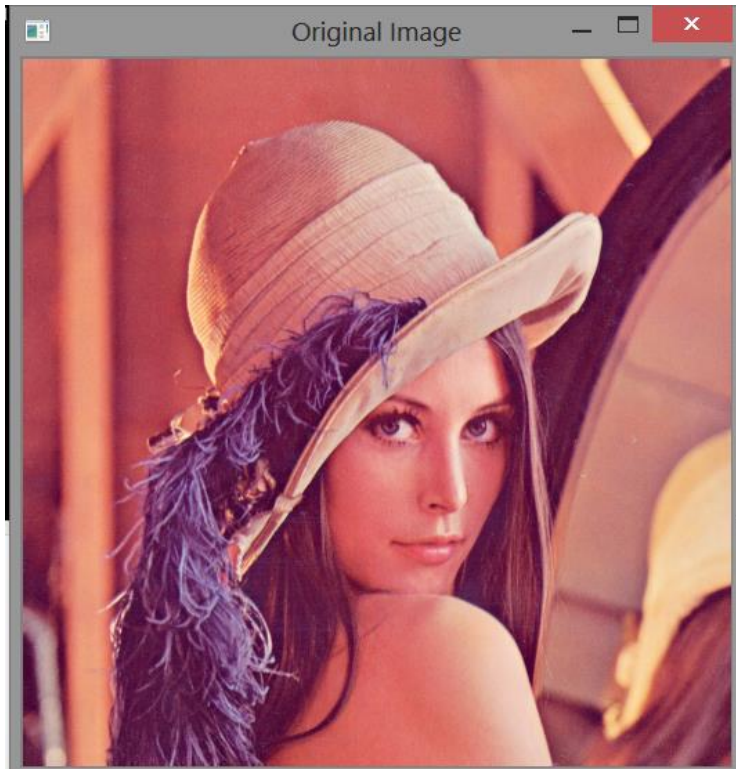
In this example, I used 15 for low Thresholding value and 25 for high Thresholding value.

- Terminal Screen

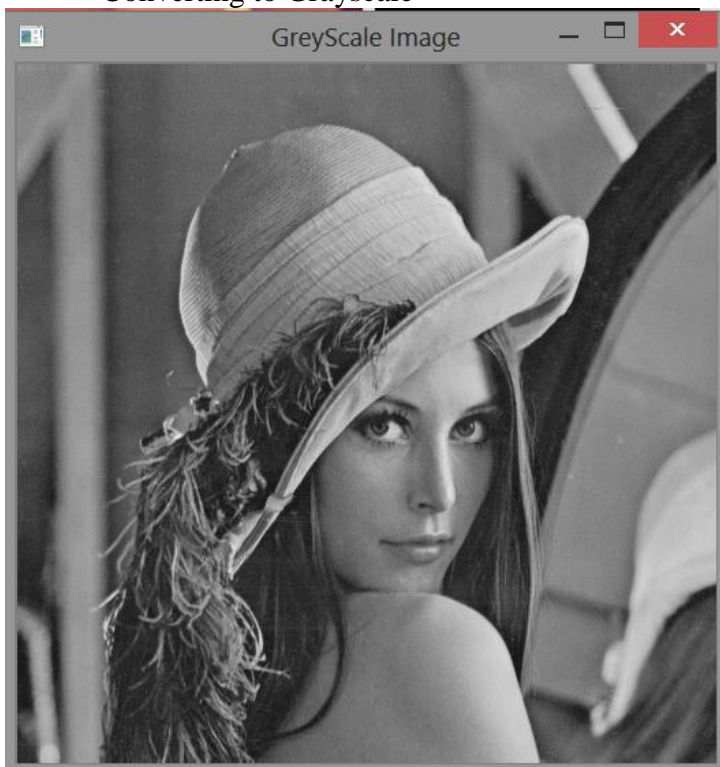


```
C:\Users\Hüseyin Erdoğan\Desktop\BLG477E_HW01\Debug\hw1.exe
--- CANNY EDGE DETECTOR ---
Enter Image Name :Lenna.png
Enter Thresholding Values for Hyteresis Thresholding Function
Low Thresholding Value :15
High Thresholding Value :25
```

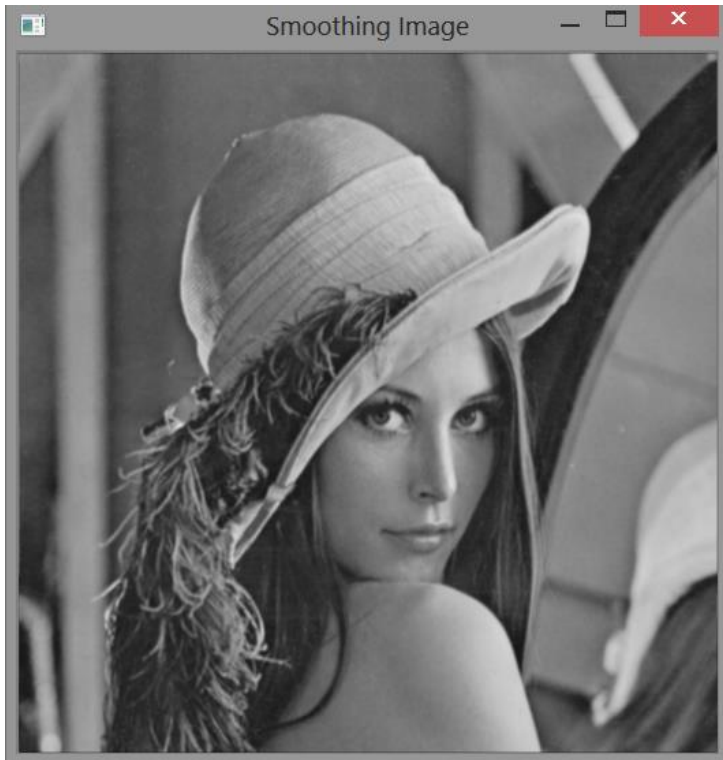
- Original Image



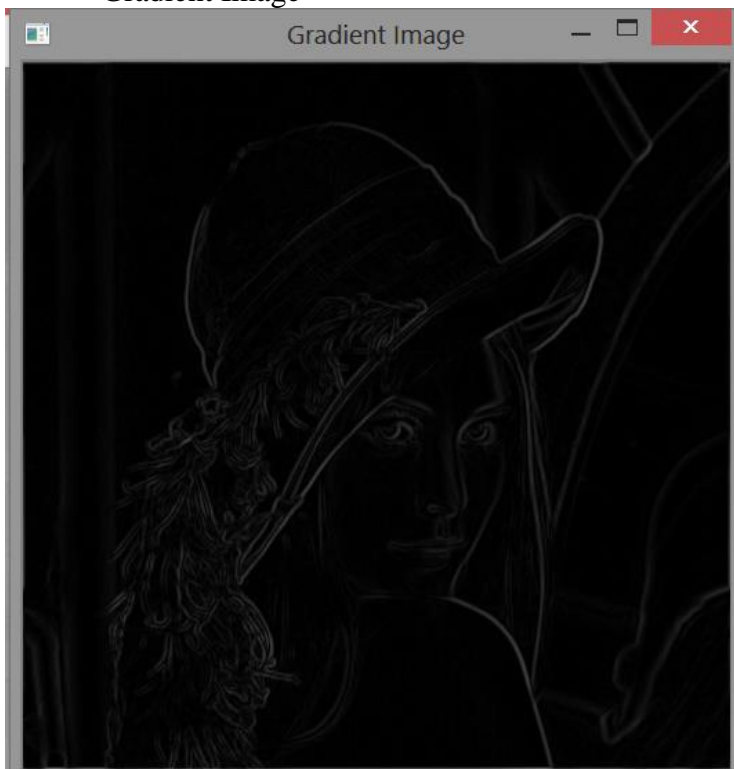
- Converting to Grayscale



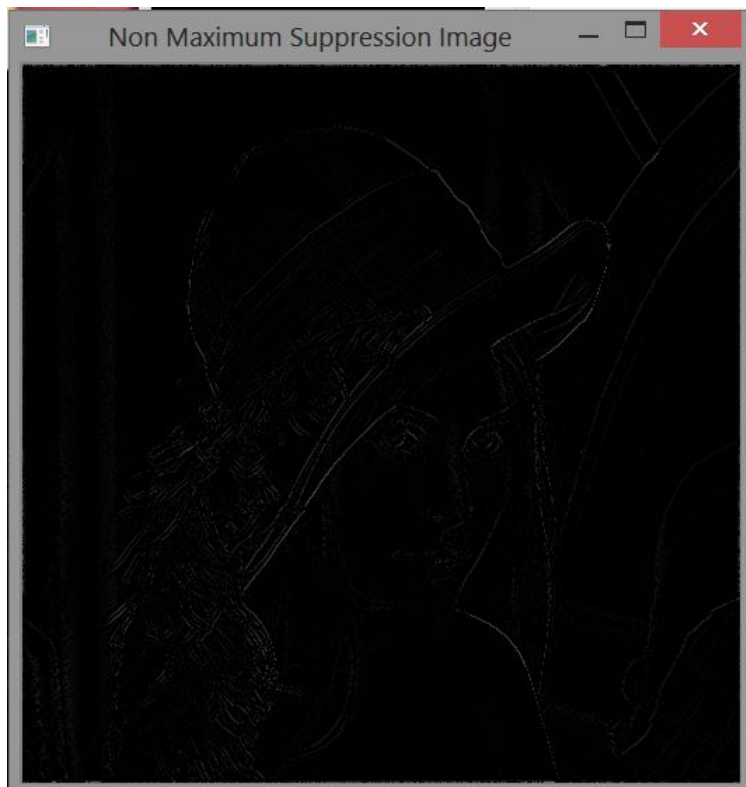
- Smoothing Image



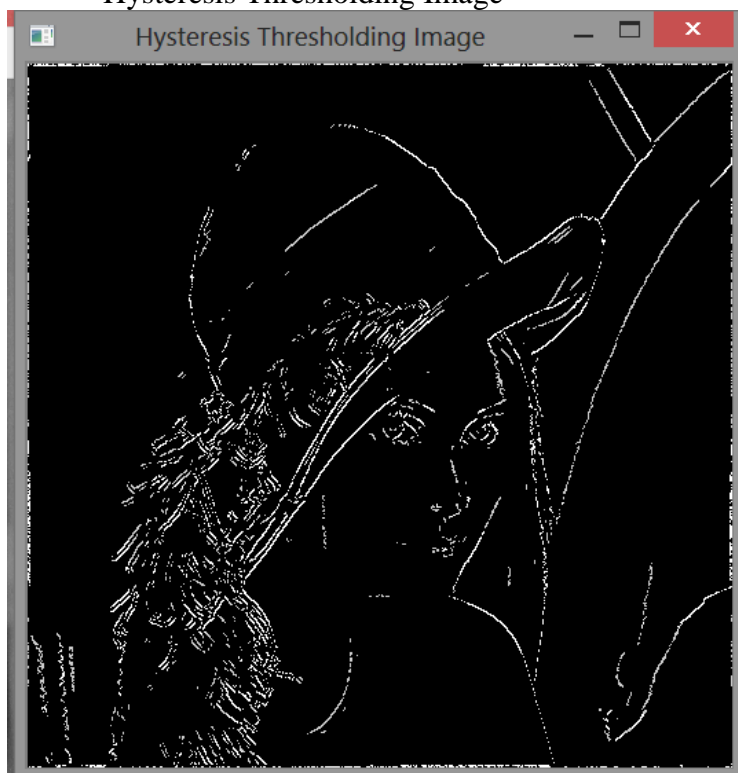
- Gradient Image



- Non-maximum Suppression Image



- Hysteresis Thresholding Image



- Canny Image

