Computer Vision
Assignment 3
German University in Cairo
Frederic-Gerald Morcos
4-1805
E13

Introduction

The Open Computer Vision library is used to get and set individual pixel values from and to the image as well as display it.

The command line interface to cv3 is as follows:

```
./cv3 <image-filename> [T1] [T2]
```

where if the given image is a gray-scale image, the program will apply the Canny Edge Detector algorithm to the image using T1 and T2 as thresholds (the higher will be used as T1 and the lower one as T2). If the given image is a binary image, the program will apply the connected component labeling algorithm to the image and output a file called **`output.txt`**.

NOTE: Source code contains comments.

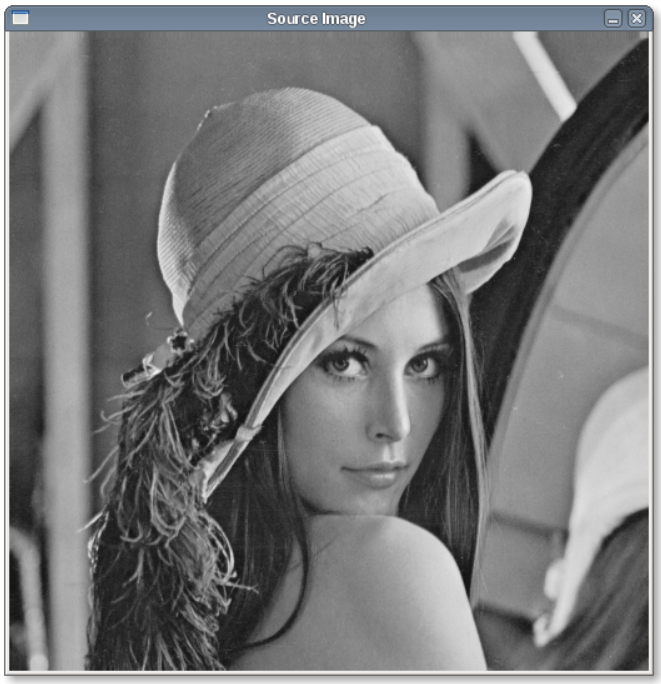
Question 1

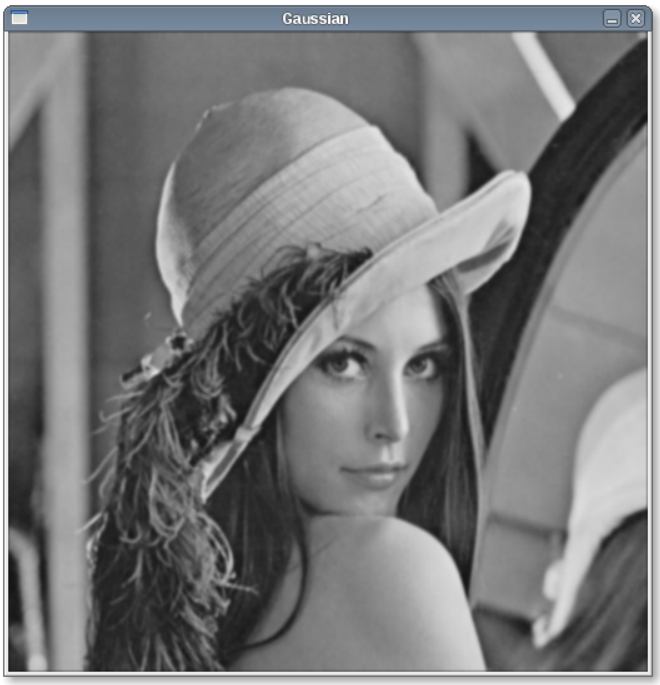
The Canny Edge Detection algorithm consists of 4 steps:

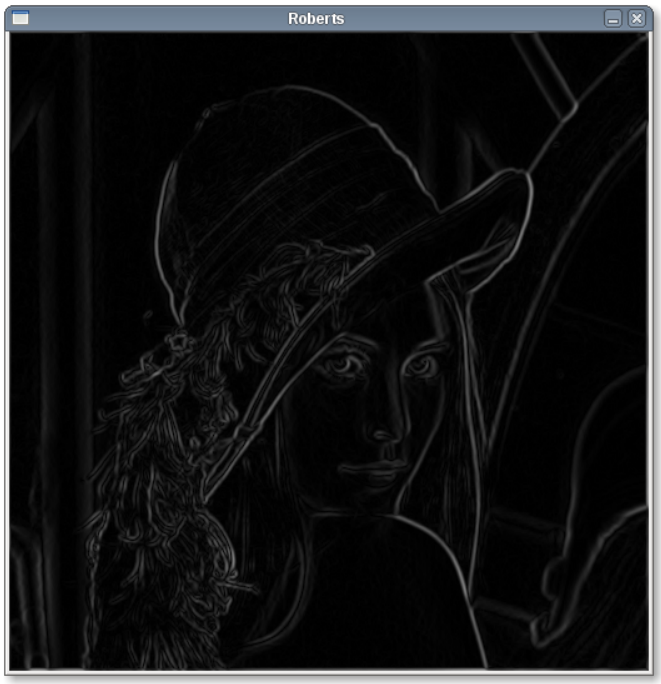
1. Apply Gaussian smoothing to reduce image noise.
2. Apply an edge detection algorithm (cv3 implements the Roberts and Sobel algorithms).
3. Apply non-maximal suppression to the gradient magnitude image.
4. Apply hysteresis thresholding to the output of non-maximal suppression (cv3 implements an interative approach and a recursive approach).

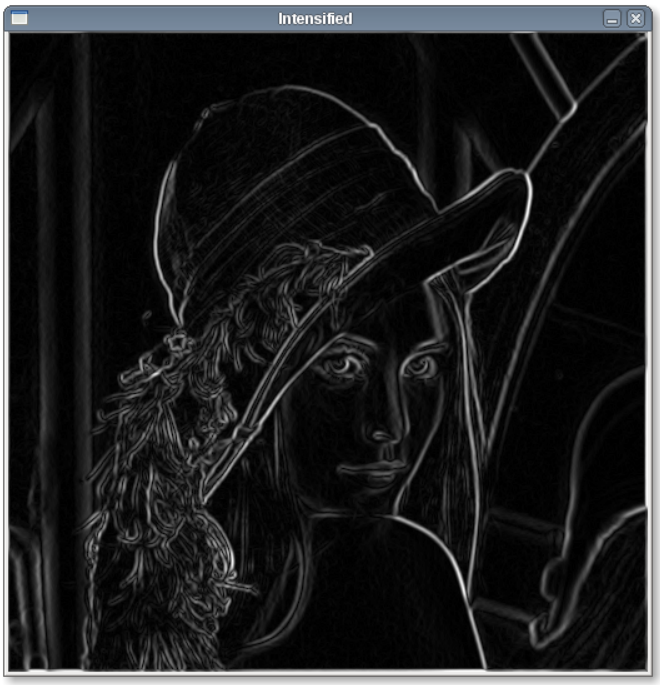
Selection of different algorithms for the same steps can be configured at compile-time by editing the **`config.h`** file.

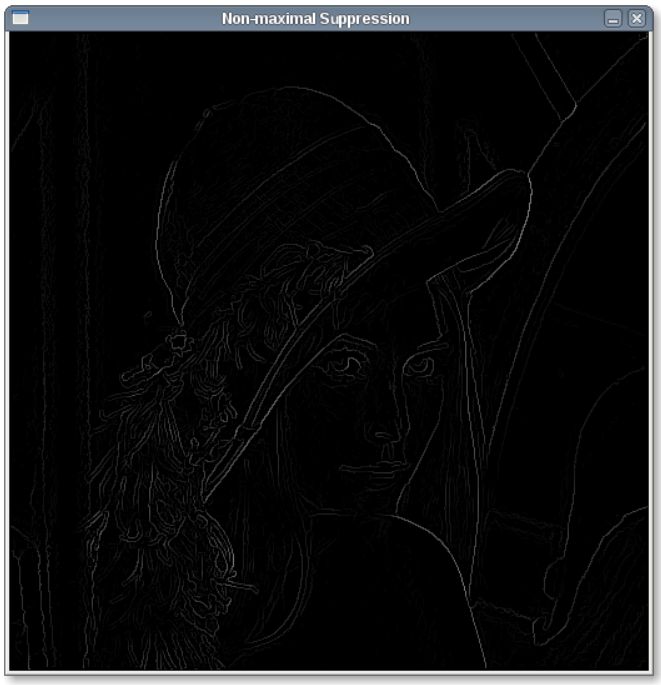
Using thresholds 10 and 20, Roberts and recursive hysteresis:

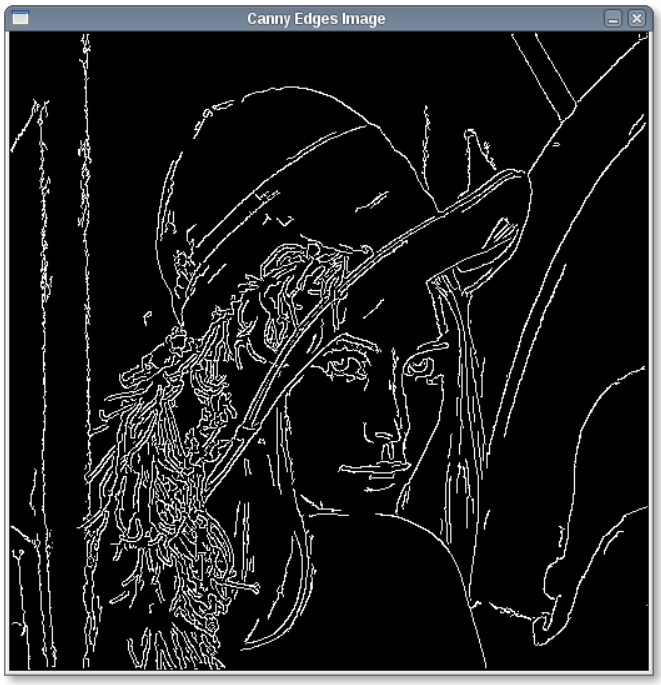

Source Image

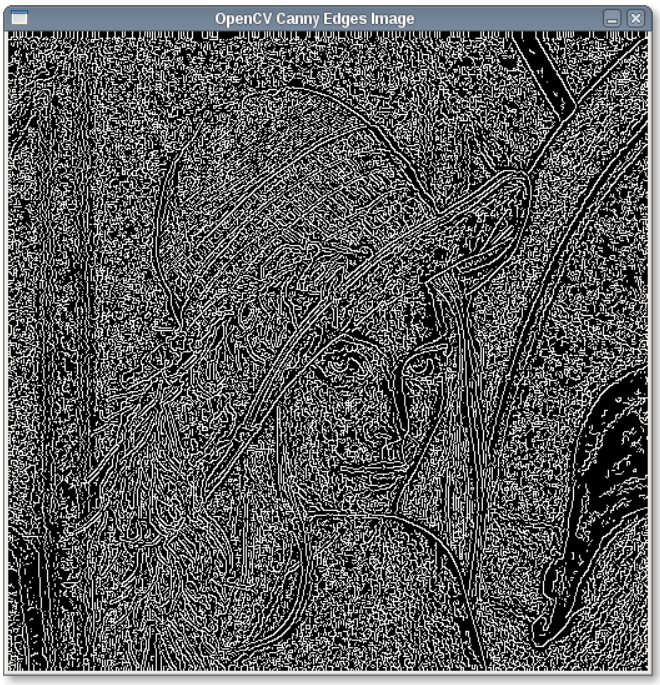

Gaussian


Roberts

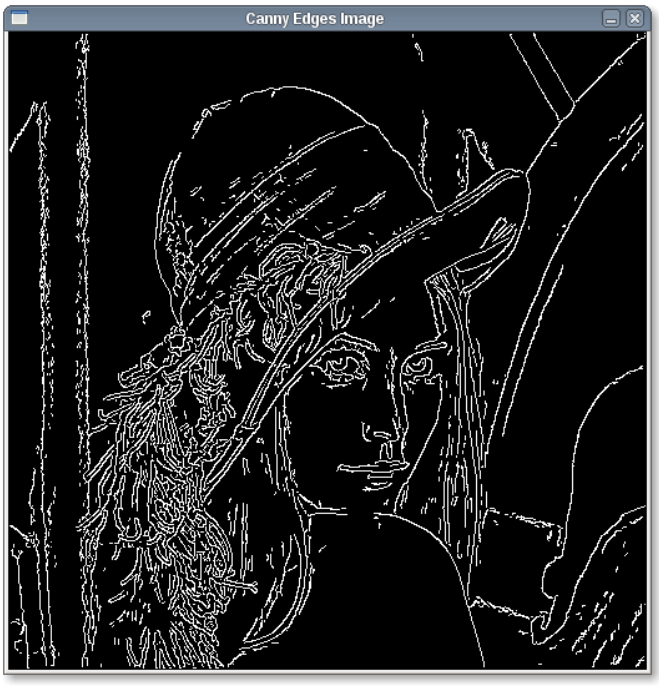

Intensified





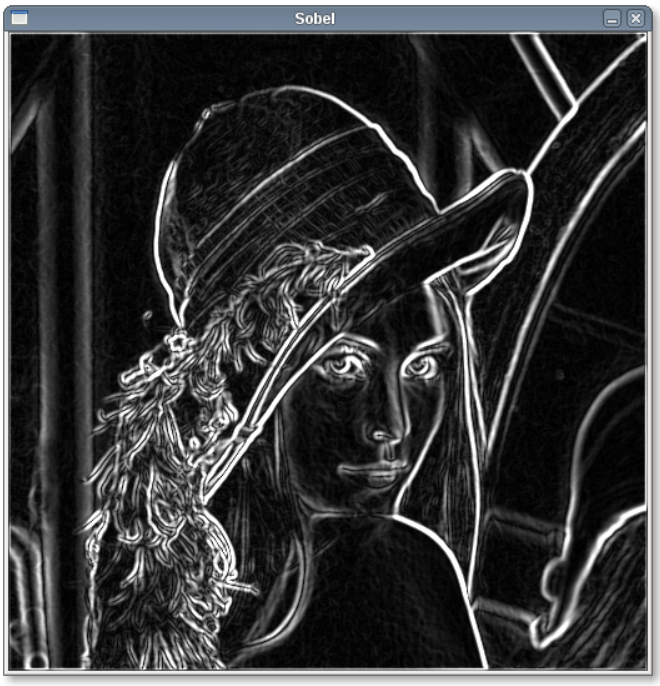


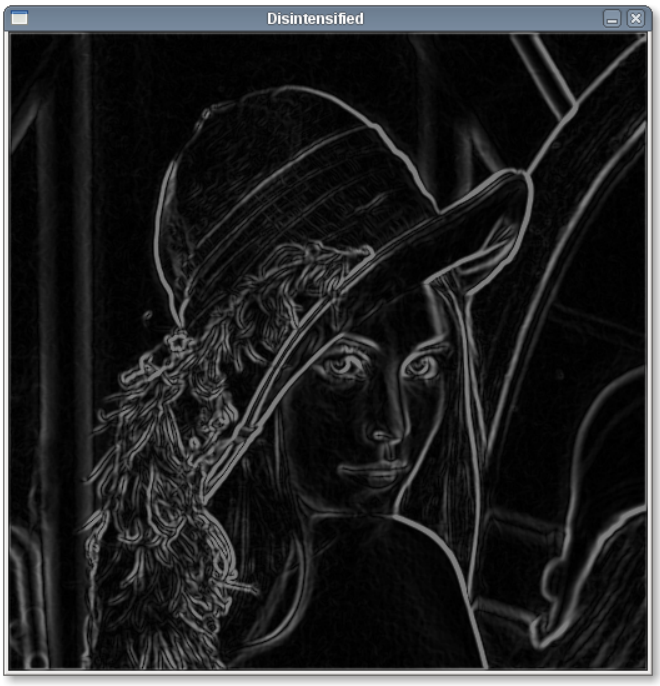
As can be seen, the OpenCV built-in cvCanny function detects a lot more edges with the same thresholds used. Using the iterative hysteresis thresholding algorithm instead of the recursive one yields:

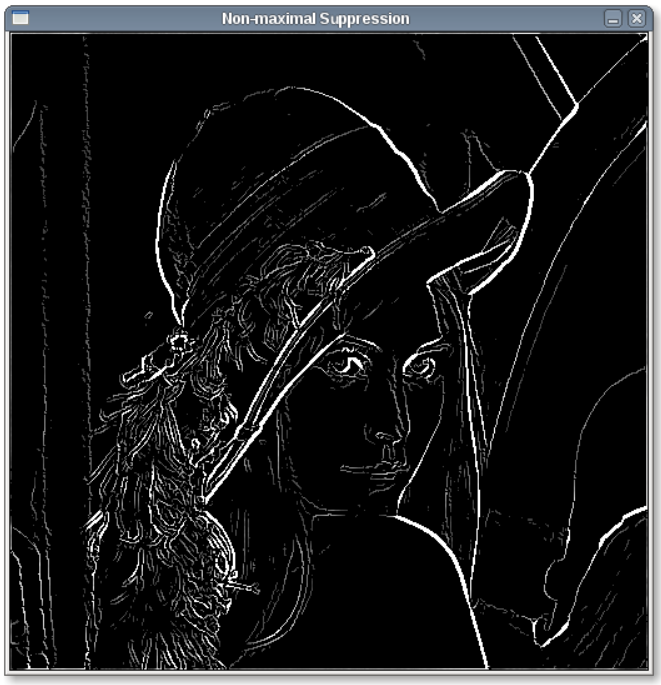


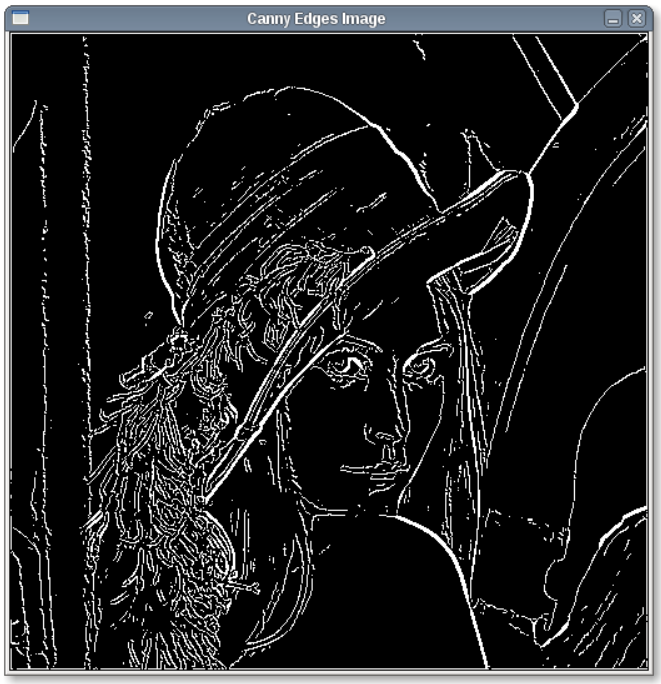
This detects more noise but also more edges.

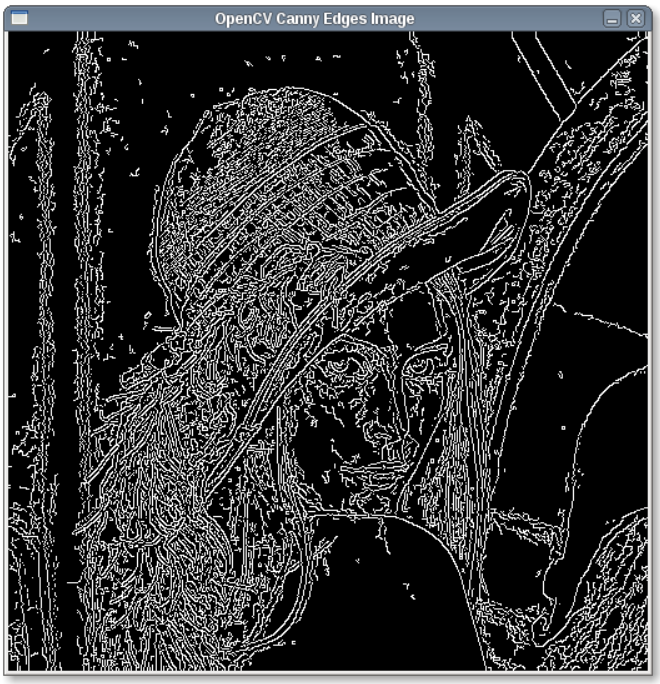
Using the Sobel edge detection algorithm and recursive hysteresis thresholding with thresholds 30 and 60:





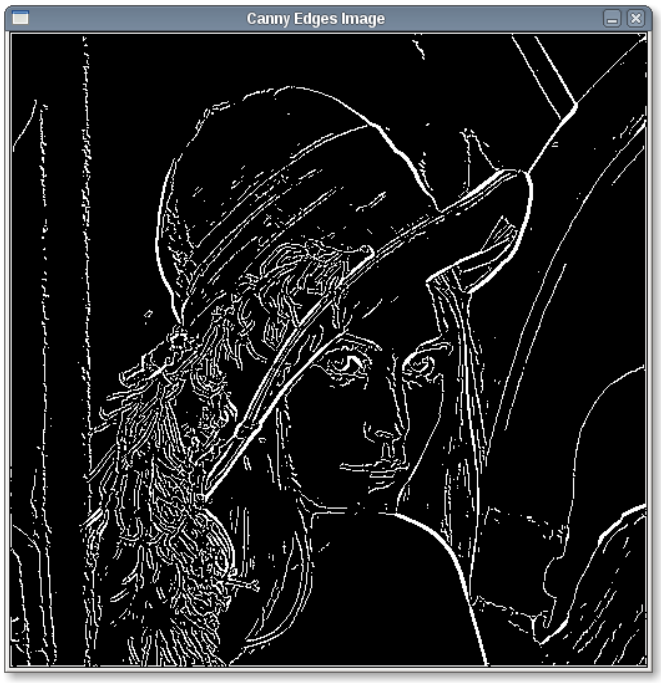






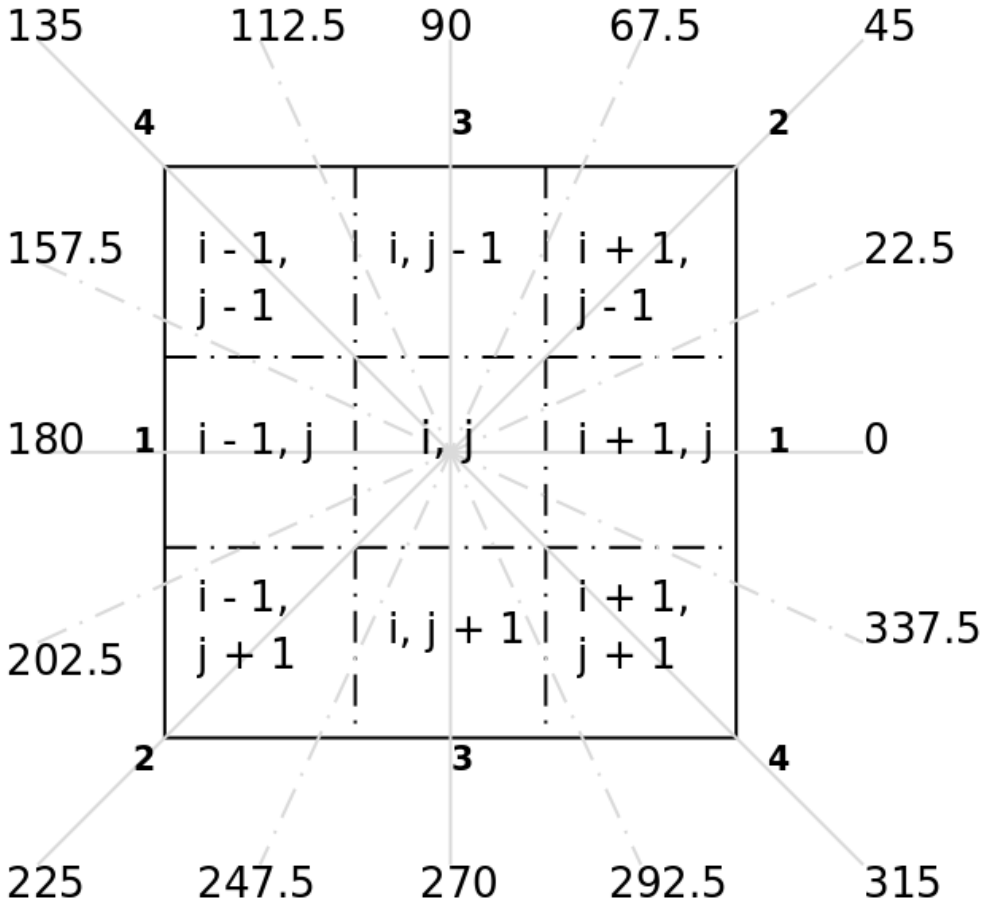
As can be seen, the Sobel algorithm outputs thick edges without gradient magnitude. This prevents these thick edges from being detected by the non-maximal suppression algorithm.

Using iterative hysteresis thresholding with the Sobel algorithm yields:



Which doesn't differ much from the recursive algorithm.

Angles output from the Sobel or Roberts algorithms are used to detect directions of edges and are used in the non-maximal suppression and hysteresis thresholding algorithms. These are labeled as follows and pixels are selected in the code as shown below:



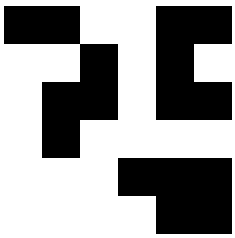
Question 2:

The Connected Component Labeling algorithm consists of 2 steps:

1. Labeling pixels based on 8-way connectivity and taking “notes” of similar labels.
2. Using the notes to unify similar labels.

Then, the new labels are used to detect shapes in the image. This is done by detecting the X-Min, Y-Min, X-Max and Y-Max for each label number. This way we can get the positions and sizes of the shapes. Then, scanning the frame around each shape (from [X-Min, Y-Min] to [X-Max, Y-Max]) and checking, if there are more than 1 label included in the frame window, then it is a circle, else then it is either a square or a rectangle. The background is discarded by detecting the largest shape in the image (with the largest area).

An example image:



Original:

0	0	255	255	0	0
255	255	0	255	0	255
255	0	0	255	0	0
255	0	255	255	255	255
255	255	255	0	0	0
255	255	255	255	0	0

After Pass 1:

1	1	2	2	3	3
4	4	1	2	3	5
4	1	1	2	3	3
4	1	2	2	2	2
4	4	4	6	6	6
4	4	4	4	6	6

Notes from Pass 1:

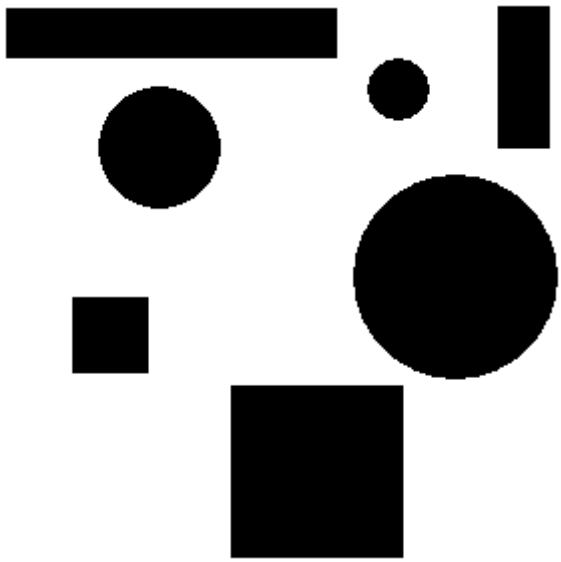
2 = 4

After Pass 2:

1	1	2	2	3	3
2	2	1	2	3	5
2	1	1	2	3	3
2	1	2	2	2	2
2	2	2	6	6	6
2	2	2	2	6	6

A textual output wouldn't be very useful with such an image so the provided image is used for the next example.

Another Example:



Notes:

4 = 5
5 = 6
7 = 8
8 = 9
9 = 10
10 = 11
11 = 12
1 = 13
13 = 14
1 = 14
15 = 16
16 = 17
17 = 18
18 = 19
19 = 20
20 = 21
21 = 22
22 = 23
1 = 24
24 = 25
25 = 26
26 = 27
27 = 28
1 = 28
1 = 30
30 = 31
31 = 32
32 = 33
33 = 34
34 = 35
35 = 36
36 = 37
1 = 37

Textual Output:

```
Rectangle at point [255, 14], width 25 and height 70.
Rectangle at point [10, 15], width 164 and height 24.
Circle with center [205, 55] and radius 15.
Circle with center [86, 84] and radius 30.
Circle with center [233, 148] and radius 50.
Square at point [43, 159] and size 37.
Square at point [122, 203] and size 85.
```