Read Me

1 Download and install OpenCV & python IDLE before trying this project.

2 Do not modify the project code or change the video path.

3 Import the missing libraries numpy , math and cv2.

4 Before you start make sure you got imutils Python package

5 Enjoy! :)

LakeDiamond

Shape detection

14 - 12 - 2018

Table of Content

1

INTRODUCTION

Shape recognition

Process

01

The extraction of contours

Allow us to extract one by one the objects

02

The approximation of contours

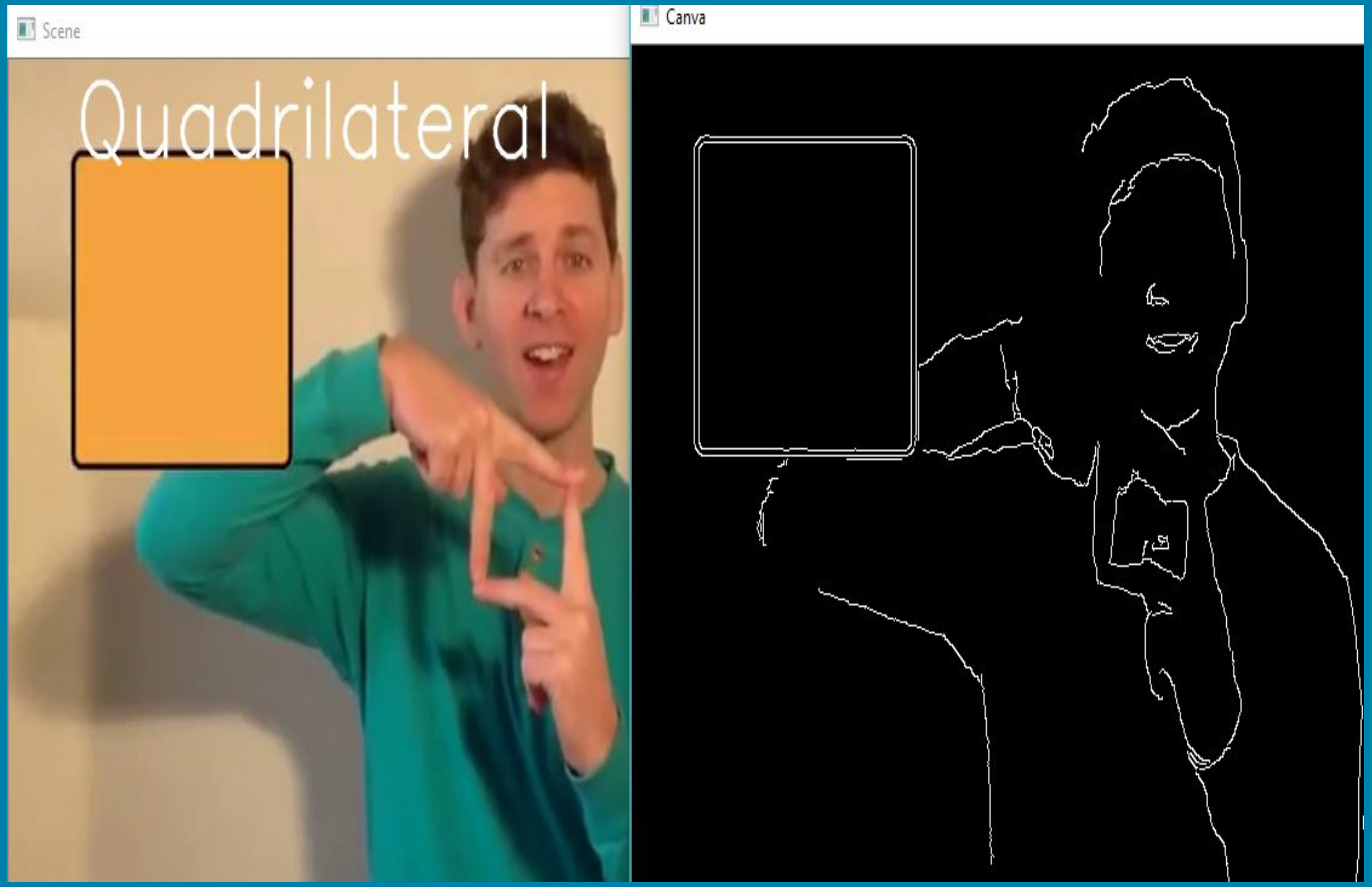
Using approxPolyDP function based on the Douglas-Peucker algorithm.

03

The classification of shapes

According to the number of lines forming their outline

Edge detection



Edge detection refers to the process of identifying and locating sharp discontinuities in an image.

Different Methods of edge detection are available:

I. Robert's Cross operator

II. Sobel Operator

III. Prewitt Operator

IV. Laplacian of Gaussian Filter

V. Canny's Edge Detection Algorithm

2

Algorithms’ methodology

Algorithm methodology

1

Start

Convert image to gray-scale

2

Step 1 :

Apply adaptive histogram equalization

3

Step 2 :

Apply range filter

4

Step 3 :

Extract edge map via Canny edge detector algorithm

5

Finally :

Apply Ramer-Douglas-Peucker(RDP) algorithm

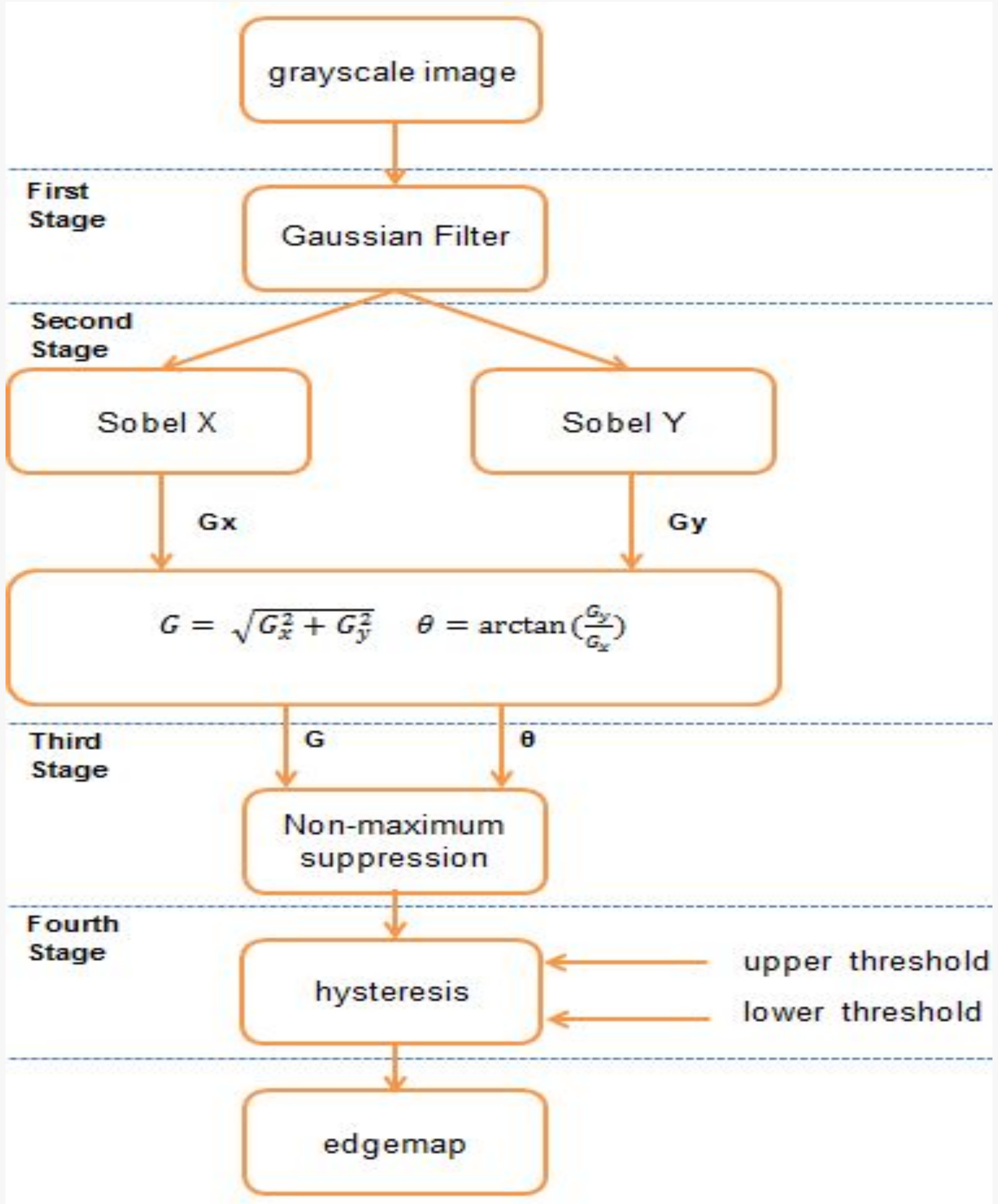
2

Contour detection methodology

Canny edge algorithm description

Definition

- A multi-stage edge detector.
- Uses a Gaussian filter in order to reduce the intensity of the gradients.



Canny edge algorithm

Process

5 steps

1 **Smoothing: Gaussian filter**

Apply Gaussian filter to smooth the image in order to remove the noise

2 **Edge detector operator**

Find the intensity gradients of the image

3 **The edge thinning**

Apply non-maximum suppression to get rid of filter defaults

4 **Potential edges**

Apply double threshold to determine potential edges

5 **Edge map**

Track edge by hysteresis:

Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Parameters Principles

3- Adjustable parameters

The size of Gaussian parameters

- Smaller filters cause less blurring, and allow detection of small, sharp lines.

- A larger filter causes more blurring, smearing out the value of a given pixel over a larger area of the image.

=> The noisier the image, the greater the width

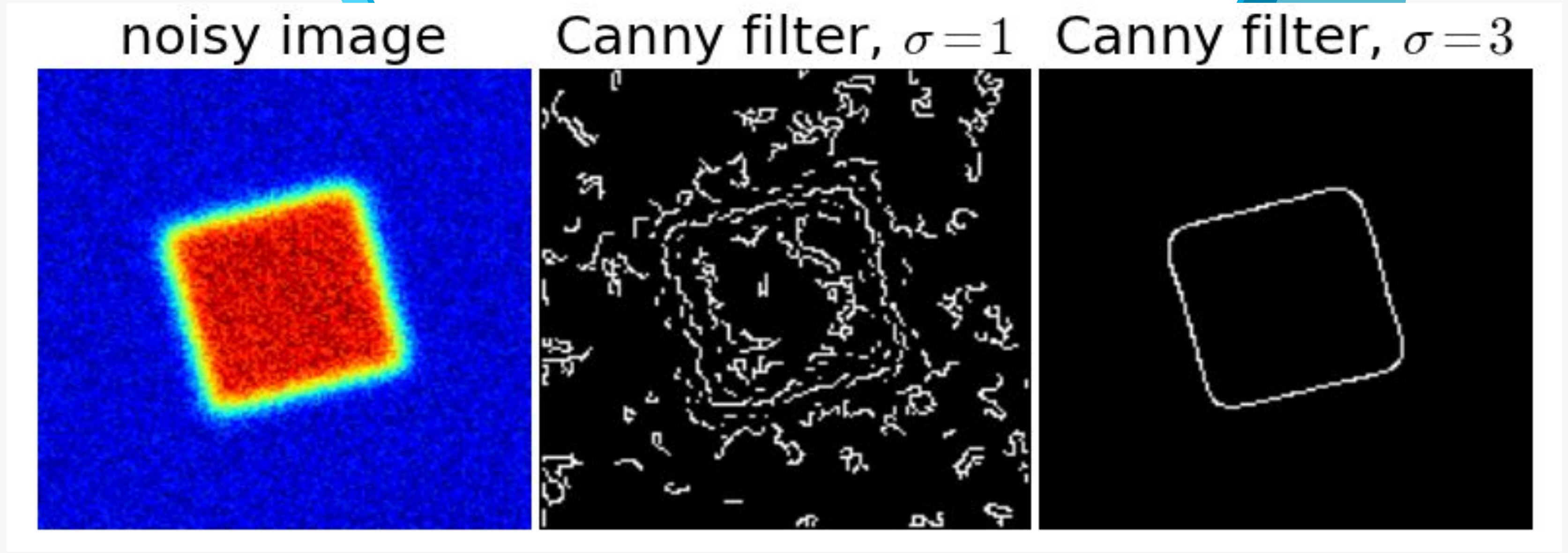
Thresholds

The use of two thresholds with hysteresis allows more flexibility:

The low and high threshold for the hysteresis thresholding.

Gaussian filter

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\left(x^2+y^2\right)/\left(2\sigma^2\right)}$$



1. The Gaussian reduces the effect of noise present in the image.
2. The potential edges are thinned down to 1-pixel curves by removing non-maximum pixels of the gradient magnitude.
3. The edge pixels are kept or removed using hysteresis thresholding on the gradient magnitude.

Edge detector operator

(G_x) , (G_y)

Intensity gradient

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x),$$

Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image.

The edge detection operator (Robert,Prewitt,Sobel) return:

A value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y).

From this the edge gradient and direction can be determined.

The edge thinning

Non-maximum suppression is applied to find "the largest" edge, after applying gradient calculation.

Step 1:

Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.

Step 2:

If the edge strength of the current pixel is the largest compared to the other pixels the value will be preserved. Otherwise, the value will be suppressed.

The pixel that is pointing in the y-direction, it will be compared to the pixel above and below it in the vertical axis

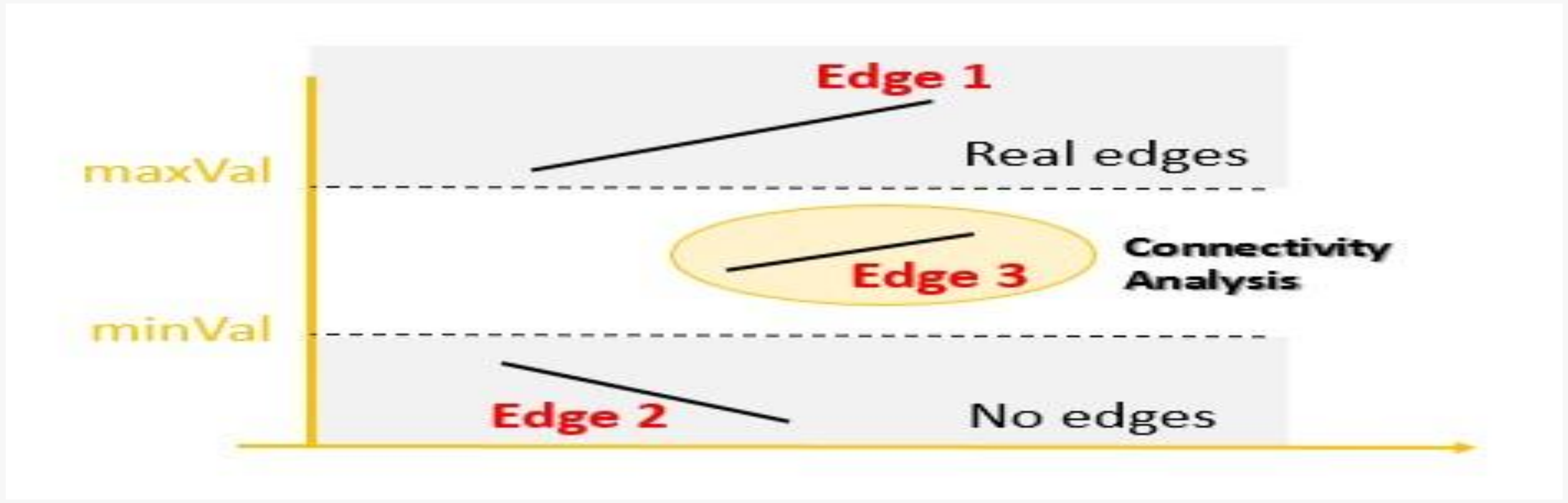
2

Approximation of contours

Potential edging

A double threshold filter to multibeam variables for the purpose of removing data (noise) within a specified range.

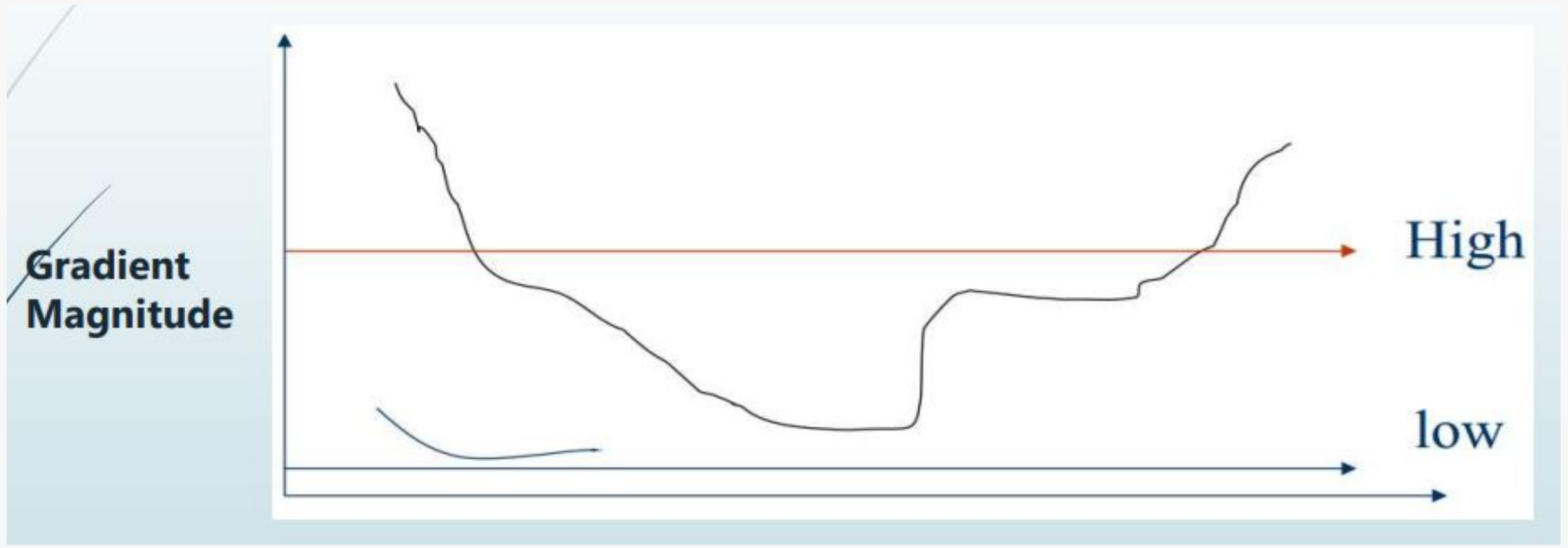
1. All samples with values below the **Lower threshold** are "thresholded out".
2. All samples with values above the **Upper threshold** are retained.
3. For samples with values between the thresholds:
   1. Values that are between the two thresholds and within the specified **Influence radius** ("close") from a data point that has been unchanged.
   2. Values that are between the two thresholds yet outside the specified **Influence radius** are set to a specified threshold value.



Edge map

Canny operator uses a method called "hysteresis"

This stage decides which are the edges and which are not. For this, we need two threshold values, minVal and maxVal.

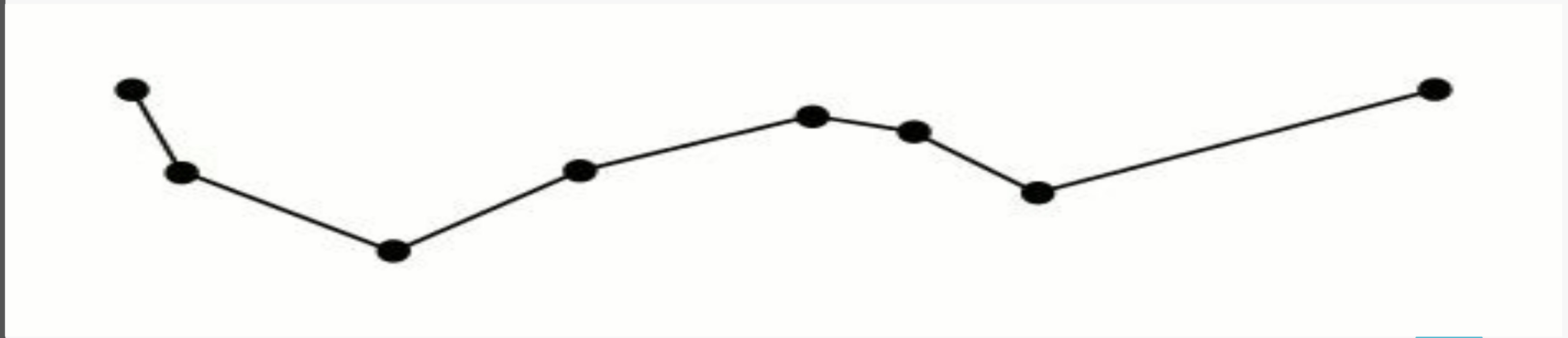


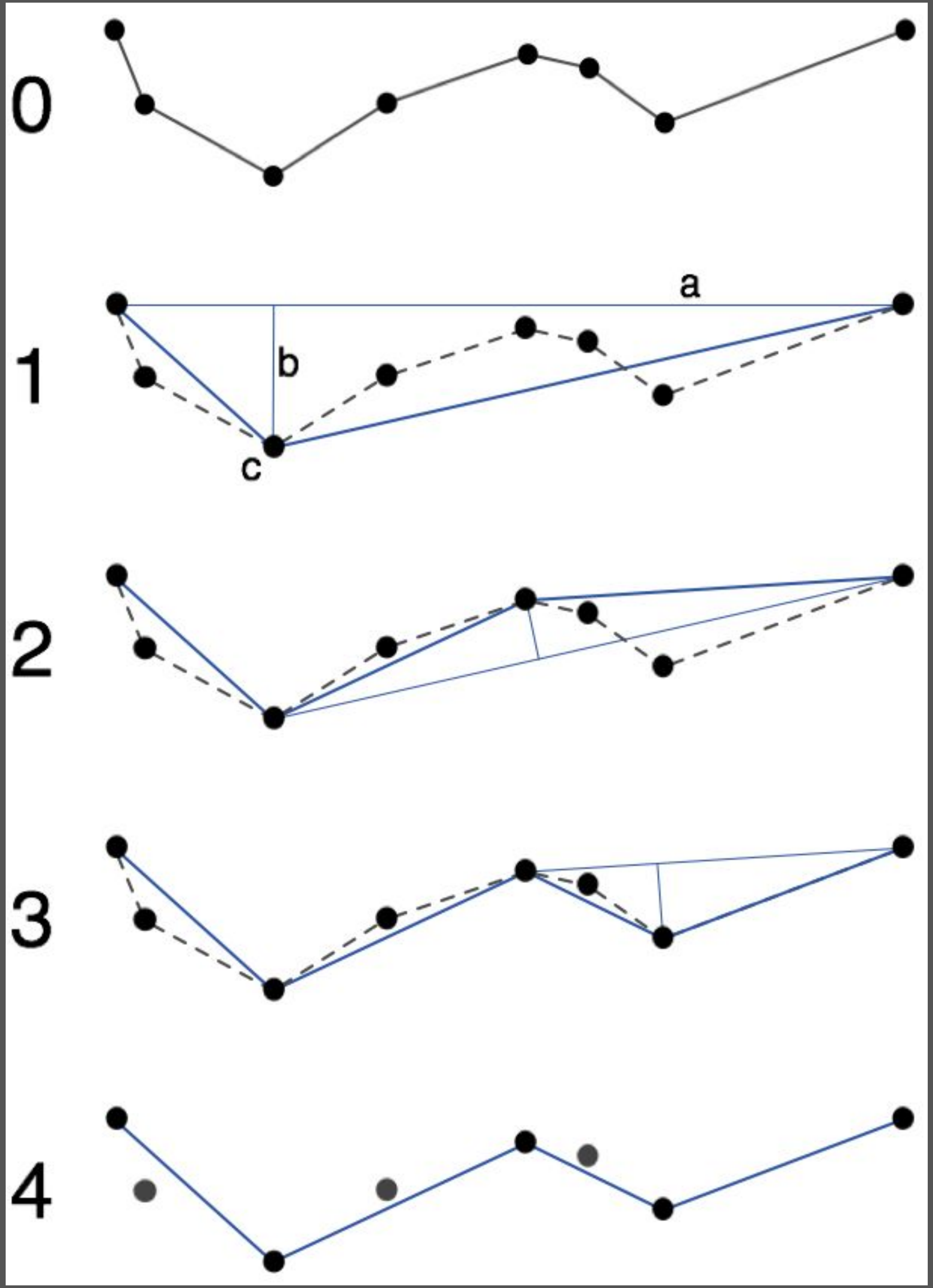
2

Shape trace of contours

Ramer-Douglas-Peucker Algorithm

Python/NumPy implementation, **serves to reduce the number of nodes and describe the contours by a set of segments.**
Used for reducing the number of points in a curve that is approximated by a series of points for 2D and 3D data.
The split and merge algorithm for a better code and resource





The algorithm works recursively by the "divide and conquer" method.

1. At each step, all the nodes are traversed between the terminals and the node furthest from the segment
2. if there is no node between the terminals the algorithm ends,
3. The algorithm is called recursively on two sub-parts of the polyline: from the first terminal to the remote node, and from the remote node to the final terminal.

Ramer-Douglas-Peucker Algorithm complexity

Closing cause:

the algorithm ends necessarily because in the worst case the polyline (or the polygon) is not simplifiable and each branch formed by the recursion will end when the terminals will not be separated by a node

since at each iteration the number of recursion branches is multiplied by two and all nodes are visited.

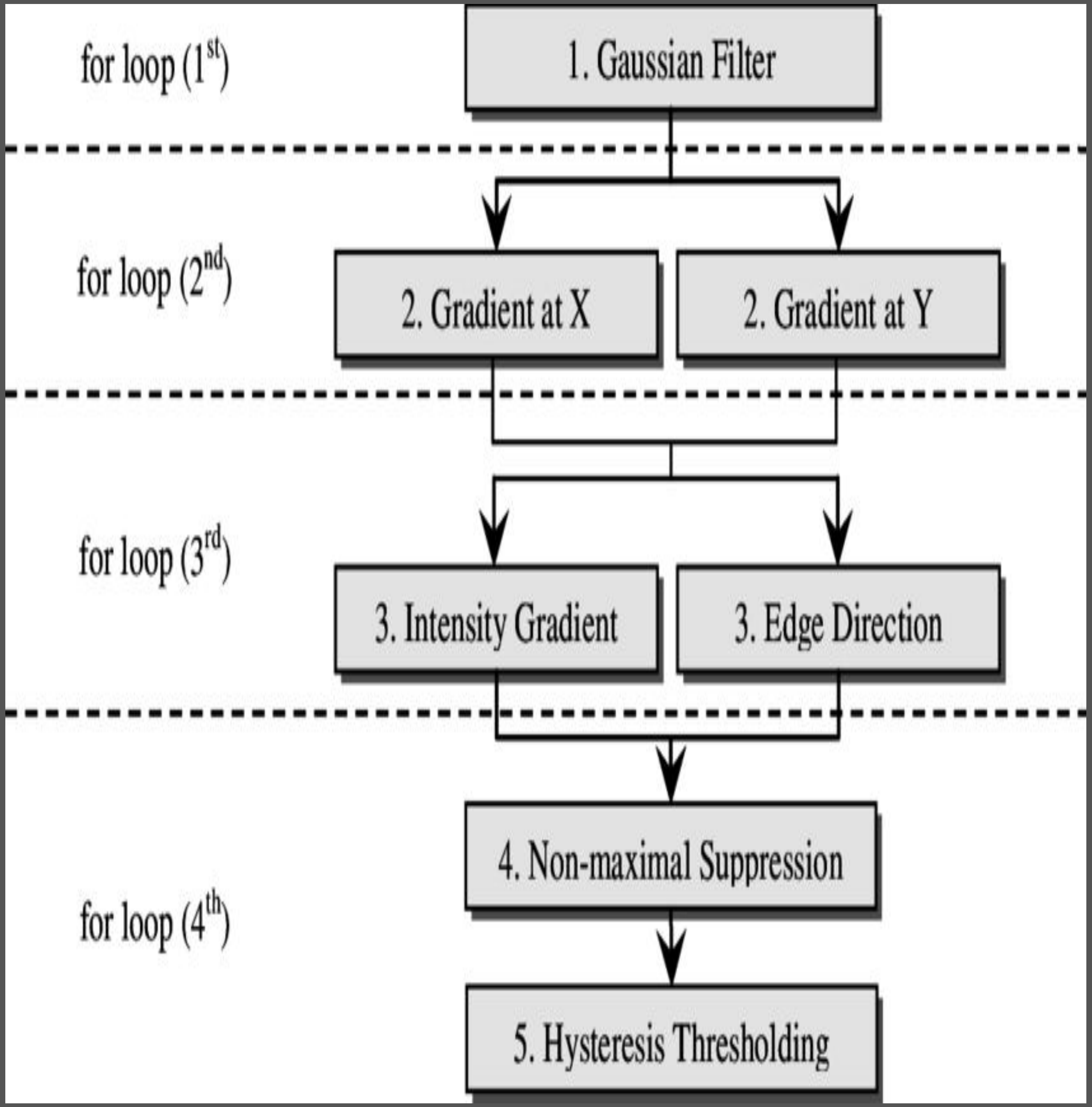
the complexity is: $\mathcal{O}(n \log n)$

to the case where each segment is forced to be cutted: $O(n^2)$

.

2

Classification of edges’ algorithm



Canny edge algorithm complexity

Using the FFT, it's possible to implement convolutions in time O(n log n).

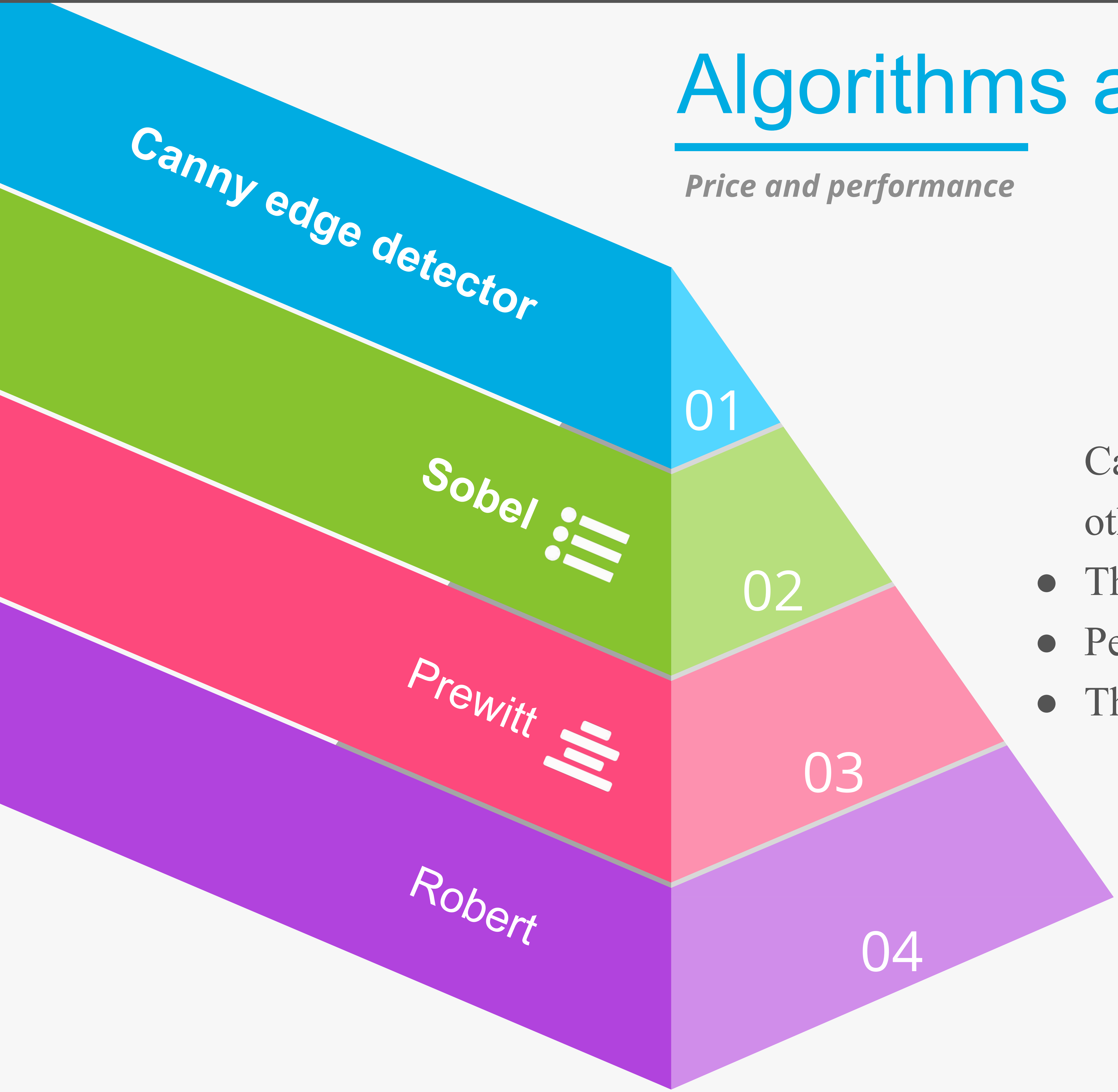
If the image has dimensions m × n, the time complexity will be O((m*n) log (m*n)) .

By removing all the high and low values, then dropping all other pixels that aren't near other pixels that can be done in time O(m*n):

Overall time complexity is O((m*n) log(m*n))

Algorithms analysis

Price and performance



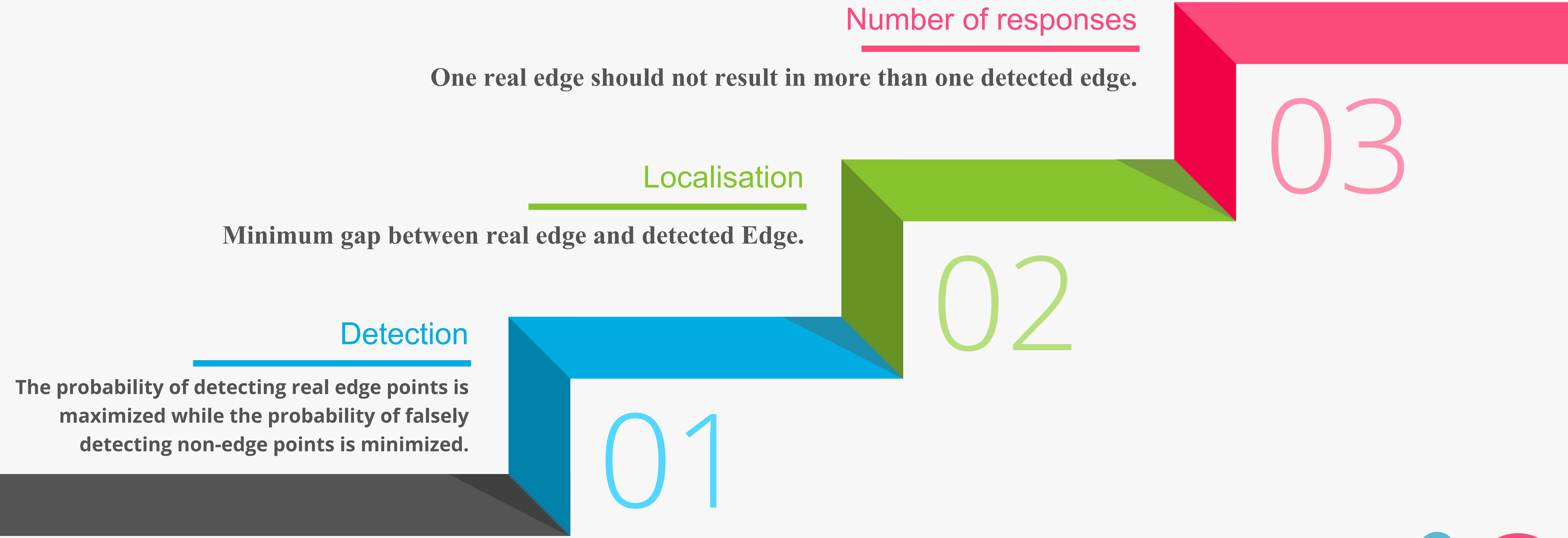
Canny edge detector algorithm between all the other solutions is:

- The most expensive,
- Performs better than all these operators.
- The least optimal filter for ressources

2

Conclusion

Why canny edge algorithm?



Detection

The probability of detecting real edge points is maximized while the probability of falsely detecting non-edge points is minimized.

Localisation

Minimum gap between real edge and detected Edge.

Number of responses

One real edge should not result in more than one detected edge.

Documentation

The split-and-merge algorithm: Ramer-Douglas-Peucker algorithm, RDP.

Edge detection algorithm: Canny edge detector algorithm

• • •

Thank You

• • •

Contact

Dhaouadi Zahra

zohra.dhaouadi@esprit.tn

+21626612978

We find a way or we make one..!