CHALLENGE 1 : CREATION OF EDGE DETECTOR AND COMPARING WITH LAPLACIAN AND LOG GAUSSIAN METHODS ON NOISE AND GRAY-SCALE IMAGES

Kevin SECRET-MORLAND

University of Condorcet, VIBOT, LE CREUSOT.

ABSTRACT

This challenge is first about the development of an algorithm for edge detection. I would like to extend this challenge to find other type of detectors like Laplace (1st and 2nd derivation) and Log-Gaussian to get a good comparison between the firth. In this paper, I will explain the working of this codes and demonstrate the advantages and inconvenient on different images like noisy and gray-scale.

***Index Terms*—** Challenge, Edge detection, Laplace, Log-Gaussian

1. CHALLENGE DESCRIPTION

The main challenge is to create a simple rectangle with size N x M. We have to describe exactly the contour of this rectangle. The description will be obtained through pixels having a Luminance different to 0 in a new image.The description can be inside or outside.

2. INTRODUCTION TO EDGE DETECTION

What is Edge detection ? Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision. The figure 1 shows the Canny edge detector. The algorithm of edge detection include four types of processing :

- Filtering : To improve the performance of an edge detector with respect to noise.

- Enhancement : To facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point

- Detection : To determine which points are edge points.

- Localisation : The location of the edge can be estimated with subpixel resolution if required for the application. The edge orientation can also be estimated.

Figure 1 : Canny filter example

3. MY EDGE DETECTOR

My algorithm is very simple to understand. The context is "if in a binarized image (0 or 1 pixel value), the neighbor pixel value is different, so it is considered as an edge. On a new image we describe the new edge pixel with the value of '1'.

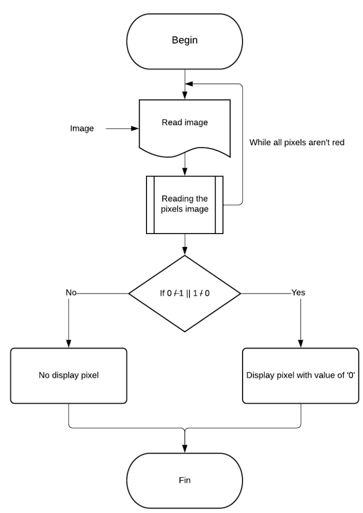


Figure 2 : my edge detector algorigram

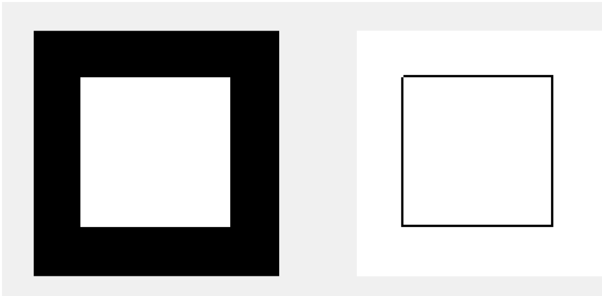


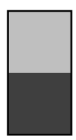
Figure 3 : Square edge detecting result
Right : Image Square / Left : detected square

4. 1ST ORDER DERIVATIVE OPERATOR

The 1st and 2nd derivations are two way to perform edge detection. there are very affected by the noise of an image because the larger is the noise, the stronger is the response and the pixels neighbours looks very different. So the best way to use an edge detector could be a smoothing of the image. Robert, Prewitt and Sobel are three 1st order derivative filters.

$\frac{\delta f}{\delta x} = f(x+1, y) - f(x, y)$

$\frac{\delta f}{\delta y} = f(x, y) - f(x, y+1)$



On the Figure 4 the result of the first derivative filter on a non noisy image. We can see a lot of details are erased but the contours are clear and the character is recognizable. In the top of the image there's artefacts corresponding to difference of shades in the original one.

5. 2ND ORDER DERIVATIVE OPERATOR

5.1. Laplacian operator

The second derivative of a smoothed step edge is a function that crosses zero at the location of the edge. The Laplacian operator is the two-dimensional equivalent of the second derivative. The formula for the Laplacian of a function $f(x, y)$ is :

$$\frac{\delta^2 f}{\delta x^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

$$\frac{\delta^2 f}{\delta x^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

the 2nd order derivative operator is stronger than the 1st order and give more details on edges. The Laplacian operator signals the presence of an edge when the output of the operator makes a transition through zero. We can zero crossing the filtered image to better render the edges. In principle, the zero crossing location can be estimated to subpixel resolution using linear interpolation, but the result may be inaccurate due to noise.

Figure 4 : Lena filtered with 1st order derivative filter

Figure 5 : Lena filtered with 2nd order Laplacian derivative filter

6. LAPLACIAN OF GAUSSIAN (MARR-HILDRETH OPERATOR)

The edge points are detected by finding the zero crossings of the second derivative of the image intensity are very sensitive to noise. Therefore, it is desirable to filter out the noise before edge enhancement. To do this, the Laplacian of Gaussian (LoG), due to Marr and Hildreth, combines Gaussian filtering with the Laplacian for edge detection.
The characterstics of LoG edge detector are:

- A Gaussian smoothing.
- The enhancement step is the second derivative Laplacian operator.

- The detection criterion is the presence of a zero crossing in the second derivative (Figure 7).
- The edge location can be estimated with subpixel resolution using linear interpolation.

We can manipulate the filter (Figure 6) by changing the value of the standard deviation variable σ where the function is to the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1-\frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Here is the influence of the sigma value on our Lena image:

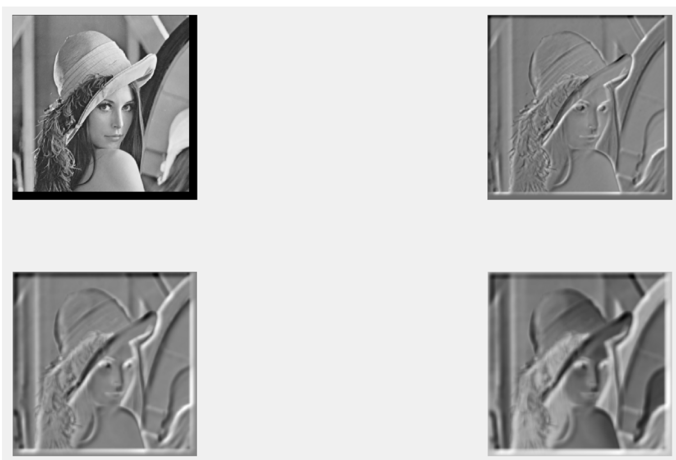


Figure 6 : Lena filtered with 2nd order LoG filter / no zero crossing

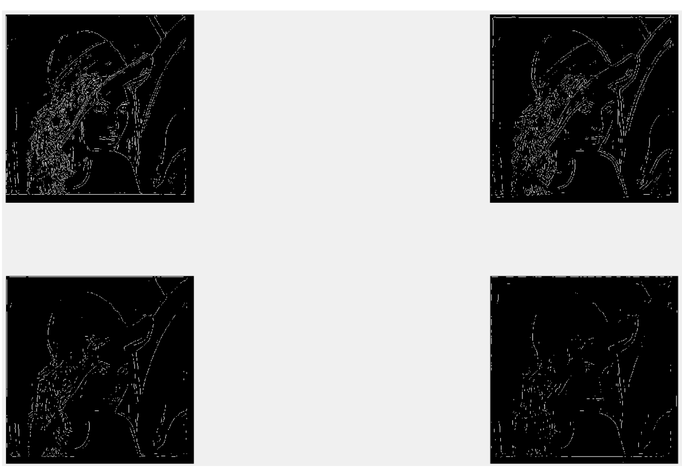


Figure 7 : Lena filtered with 2nd order LoG filter / with zero crossing

7. NOISE COMPARISION

Now, after presenting this filters, we can compare them-self with 3 types of noise : Gaussian, Poisson and SaltPepper, Figure 8 to 12. The value of Sigma on the LoG algorithm is fixed to 1.

8. RESULT

With this different results, we can say first all edge detectors are sensible to noise except the LoG which uses directly a Gaussian smoothing. But Salt and pepper is difficult to use for edge detection, even with 2nd derivative and LoG.
Our algorithm is surprising the less affecting by Salt Pepper noise whereas it's simply coded. We can see a huge difference of sensibility from the 1st order to the 2d order derivative operator. This experience prove the importance to use Gaussian blur before using edge detectors. The best between the fourth is the LoG because it uses the 2nd derivative operator (Laplacian) added with Gaussian smoothing.

Figure 8 : Lena noised : First : orignial / Second : Poisson / Third : Gaussian / Fourth : Salt Pepper

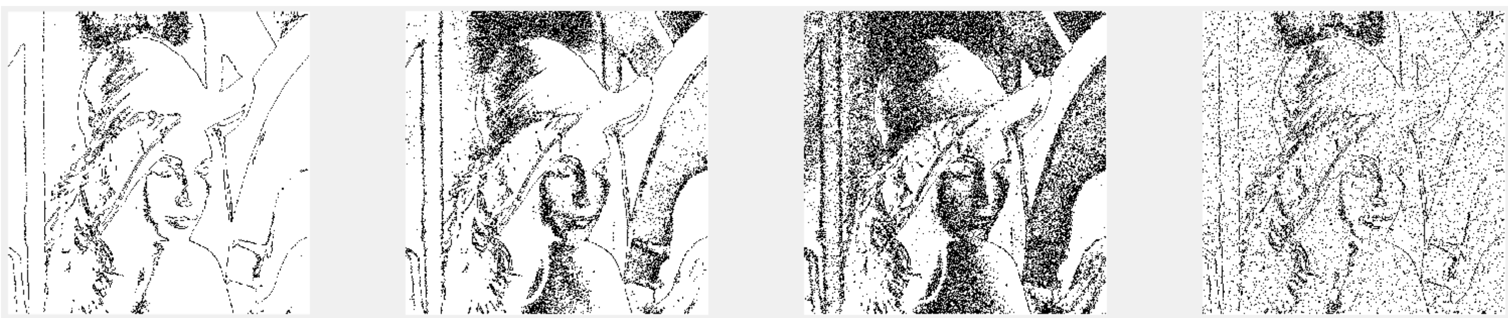


Figure 9 : Lena edge detection with challenge algorithm : First : orignial / Second : Poisson / Third : Gaussian / Fourth : Salt Pepper

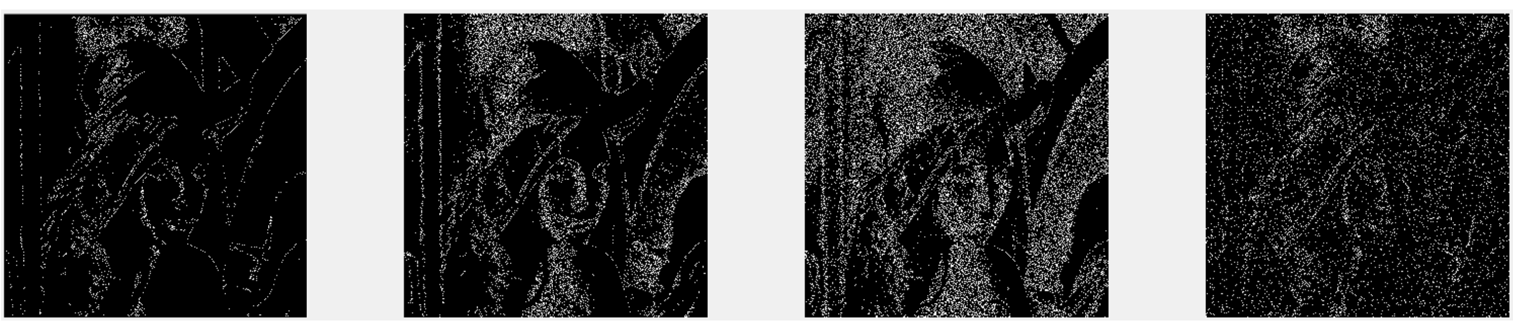


Figure 10 : Lena edge detection with 1st order derivative : First : orignial / Second : Poisson / Third : Gaussian / Fourth : Salt Pepper

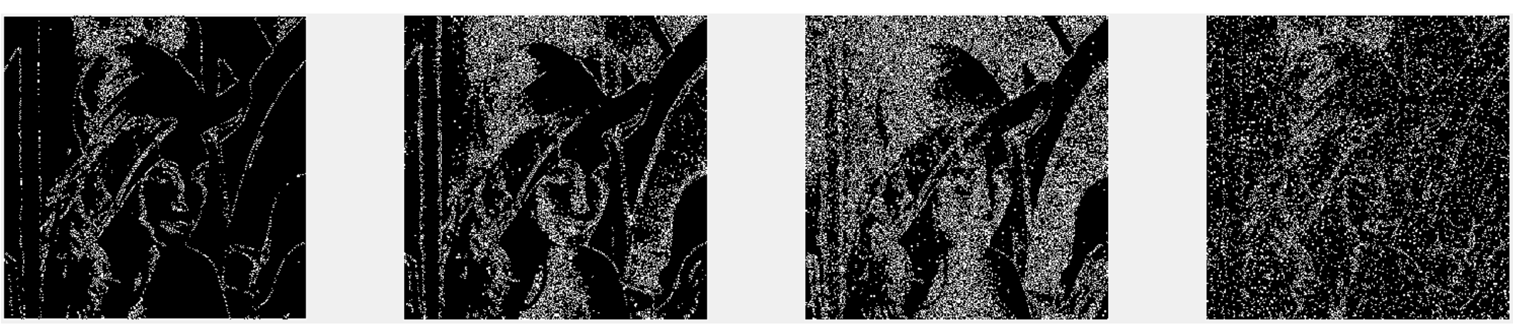


Figure 11 : Lena edge detection with 2nd order derivation (Laplacian) : First : original / Second : Poisson / Third : Gaussian / Fourth : Salt Pepper

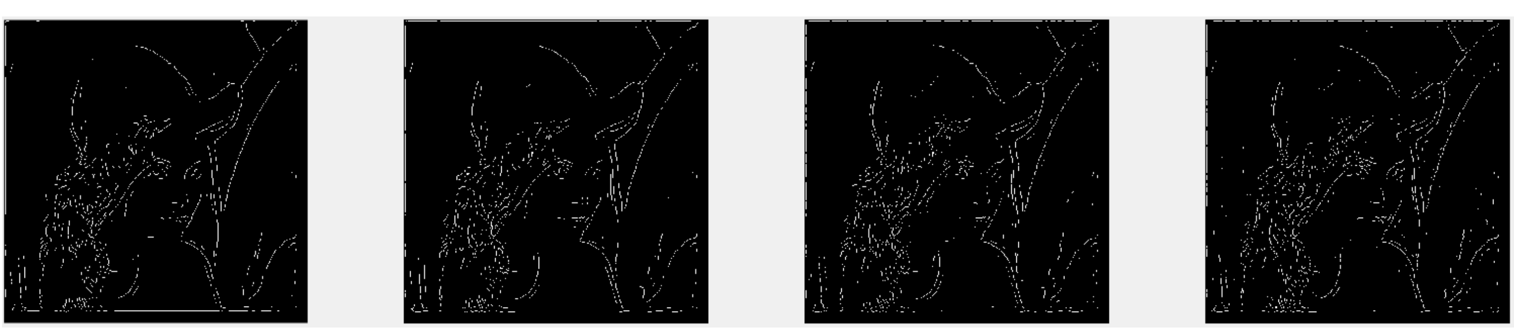


Figure 12 : Lena edge detection with LoG : First : orignial / Second : Poisson / Third : Gaussian / Fourth : Salt Pepper

9. REFERENCES

[1] Chapter 5 : Edge Detection
http://www.cse.usf.edu/ r1k/MachineVisionBook/MachineVision.files/MachineVision$_{C}hapter5.pdf$

[2] Differents type of edge detection, mostly 1st and 2nd derivation
https://web.fe.up.pt/ campilho/PDI/NOTES/EdgeDetection.pdf

[3] Chapter 7 : Edge detection course – Saad J Bedros
http://www.me.umn.edu/courses/me5286/vision/VisionNotes/2017/ME5286-Lecture7-2017-EdgeDetection2.pdf