

HOMework 5 U1

EDGE DETECTION

Héctor Mauricio Mendoza Xicoténcatl

Abstract—In this report, we documented our coding process. This process involved different image operators and edge detectors. In addition, we wrote our results, thoughts and differences of each one of the image operators.

Index Terms—OpenCV, Prewitt Edge Detection, Sobel Operator, Contour Segmentation, Roberts Edge Detector.

1 PROBLEM TO SOLVE

Select two images, one with a landscape and a person or animal on it, and a portrait (photo of a person/animal) in first plane. Obtain:

- Roberts Edge Detector.
- Sobel Operator.
- Prewitt Edge Detection.

2 CODING PROCESS

Our coding process is divided into the different image operators.

2.1 ROBERTS EDGE DETECTOR

Similar to previous exercises, first we imported the libraries necessary for the program with the commands:

```
import numpy as np
import cv2
```

Then, for the Robert operator, we decided to create an array of the matrix values which correspond to the edge detection desired. These values were obtained through extensive documentation reading. The next step was the application of the Gaussian-Blur filter on the x and y arrays of our image.

```
roberts_x = np.array([[0,0,0], [0,1,0], [0,0,-1]])
```

```
roberts_y = np.array([[0,0,0], [0,0,1], [0,-1,0]])
```

The next step consisted of the addition of the squares of the x and y array as seen on the documentation.

```
out_rob = np.square(img_robertsx) +
np.square(img_robertsy)
```

In this last section, one step more is missing, the square root of the previous step, the reason is that OpenCV cannot make this operation in our figure due to inconsistencies in the output values of our operation.

2.2 SOBEL OPERATOR

Similar to the previous operator, the Gaussian-Blur filter is and will be useful for the operator in this document. For this operator, there is an already existing function that help us to apply it to our image. This is cv2.Sobel().

```
img_sobelx = cv2.Sobel(img_gaussian,cv2.CV_8U,
1,0,ksize=5)
img_sobely = cv2.Sobel(img_gaussian,cv2.CV_8U,0,
1,ksize=5)
out_sob = img_sobelx + img_sobely
```

2.3 PREWITT EDGE DETECTION

Similar to the Roberts Detection, the Prewitt Edge Detection needs a matrix array for x and

y. The difference between the two is that both have different values in their array values. Another important difference is that their final operation of values in an addition, instead of the square root of the addition of their squares (Roberts).

```
kernelx = np.array([[1,1,1],[0,0,0],[-1,-1,-1]])
kernely = np.array([[-1,0,1],[-1,0,1],[-1,0,1]])
out_pre = img_prewittx + img_prewitty
```

2.4 CANNY

We can consider as Canny as the easiest to use due to it consist of one function: `Cv2.Canny()`.

```
im_cann = cv2.Canny(img,100,200)
```

The main difference we were able to find is that with the canny filter we can only observe the edges of our figure. Those edges correspond to small lines on the black background. It is easier for us to see only the edges in comparison with other operators on this document.

2.5 LAPLACE OF GAUSSIAN

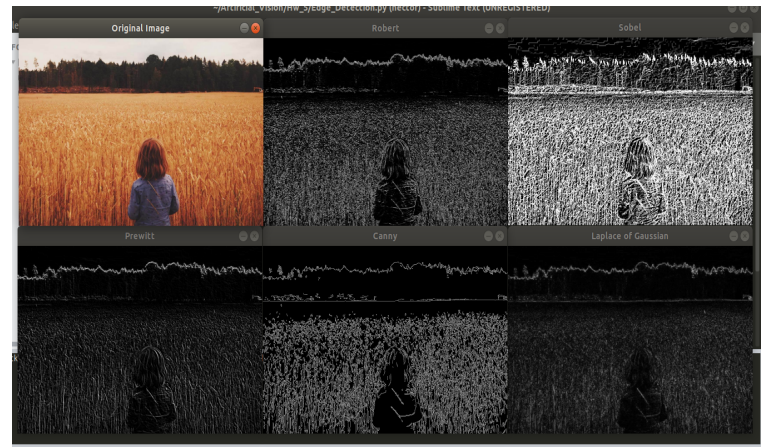
Laplace of Gaussian shares some similarities with the Canny operator. This is because it only needs two simple functions to work: `cv2.Laplacian()` and `cv2.convertScaleAbs()`.

The first one help us to convert our image with the Laplace operator. The second one converts our output image to a CV_8U image.

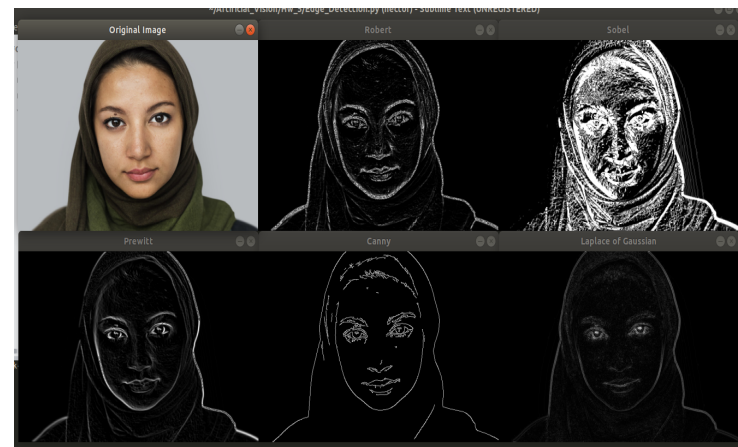
```
img_Log = cv2.Laplacian(img_gaussian,
cv2.CV_16S, ksize=3)
fin_Log = cv2.convertScaleAbs(img_Log)
```

Finally, we print our results.

3 RESULTS



Use of all operators in a landscape image



Use of all operators in photo of a person

4 CONCLUSIONS

I consider this activity was useful for us to distinguish the output differences of the operators. I could not say that there is objectively an operator better than other. Nevertheless, in my opinion I consider the Canny operator is better if we want to only get the edges of a certain figure. I also must say that certain operator work better with different types of images. We were able to see the differences of the operator working with a landscape figure and a photo of a person.

REFERENCES

- [1] "OpenCV: Laplace Operator", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/3.4/d5/db5/tutorial_laplace_operator.htm [Accessed: 30- Sep- 2020].

- [2] "OpenCV: Canny Edge Detector", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/3.4/da/d5c/tutorial_canny_detector.html. [Accessed: 30- Sep- 2020].
- [3] "OpenCV: Sobel Derivatives", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html. [Accessed: 30- Sep- 2020].



Héctor Mendoza Héctor Mauricio Mendoza Xicoténcatl is a Computational Robotics Junior student, currently studying at Universidad Politécnica de Yucatán.