OpenCV Filters

Prepared by  
Sergio Torres

Computer Vision  
CAP 4410

2/9/20

Table of Contents

Table of Contents 1

List of Figures 2

1. Implementing Filters 4

1.1 Box Filter 4

1.2 Sobel Filter 4

1.3 Gaussian Filter 7

1.4 Product Scope 7

2. Results 8

2.1 Dog Image Results 8

2.2 Bicycle Image Results 9

2.3 South\_L Image Results 10

2.4 Edge Image Results 11

References 12

# List of Figures

**Figure 1** Box Filter using OpenCV and own code……….……………..………………….................... 4

**Figure 2** Sobel Filter over X – Axis using own code…………............................................................... 5

**Figure 3** Sobel Filter over Y – Axis using own code…………............................................................... 5

**Figure 4** Sobel Filter over XY – Axis using own code............................................................................ 6

**Figure 5** Sobel Filter over XY – Axis using OpenCV …………............................................................ 6

**Figure 6** Gaussian Filters using OpenCV…………................................................................................ 7

**Figure 7** Dog Image using all Filters…………....................................................................................... 8

**Figure 8** Bicycle Image using all Filters………….................................................................................. 9

**Figure 9** South\_L Image using all Filters…………............................................................................... 10

**Figure 10** Edge Image using all Filters…………................................................................................... 11

# Implementing Filters

## Box Filter

The Box Filter had to be created using our own code and using the OpenCV function. The sliding window for both box filters starts at 1 and then increases by 2 every iteration for the whole image. I created functions to be called where the user has to input the Image that the filter will be placed on and the window that it will be shown on.

A screenshot of a cell phone

Description automatically generated

**Figure 1. Box Filter using OpenCV and own code**

## Sobel Filter

The Sobel Filters over the X, Y, and XY axis without the use of OpenCV are included in Figures 2, 3, 4. The Sobel Filter which does use OpenCV is shown in Figure 5. The sliding window that I use for each Sobel Filter has a width of 3 pixels and a height of 3 pixels. The filters are to be called as functions with the input image and the resulting window name as the parameters.

A screenshot of a computer

Description automatically generated

**Figure 2. Sobel Filter over X-Axis using own code**

A screenshot of a computer

Description automatically generated

**Figure 3. Sobel Filter over Y-Axis using own code**

A screenshot of a cell phone

Description automatically generated

**Figure 4. Sobel Filter over XY – Axis using own code**

A screenshot of a cell phone

Description automatically generated

**Figure 5. Sobel Filter over XY – Axis using OpenCV**

## Gaussian Filter

The gaussian filter was used with the Open CV function that comes with the built-in library. The code below (Figure 6) shows the Gaussian Filter being applied to the image. The window size starts off at 1 and then increments by 2 every iteration until it envelops the entire window. The standard deviation for both the x and y axis is set to 0 meaning that it will be calculated using the kernel size.

A screenshot of a cell phone

Description automatically generated

**Figure 6. Gaussian Filter using OpenCV**

## Product Scope

In this project, the goal is to implement a Box Filters, Sobel Filters and Gaussian Filters on the images provided. We are to create our own code for the box filter apply it and compare it with the built-in OpenCV function. The Sobel Filter is to be applied over the X, Y and XY axis with our own code and then apply the Sobel Filter over the XY axis using the built-in Sobel Filter function. The Gaussian Filter is to be applied using the built-in function only. We are to apply each Filter to every image that is being provided to us and label them appropriately.

# Results

## Dog Image Results

The results for the Dog Image with every filter that was required in the project is shown below side-by-side.

A dog looking at the camera

Description automatically generated

**Figure 7. Dog Image with all Filters**

## Bicycle Image Results

The results for the Bicycle Image with every filter that was required in the project is shown below side-by-side.

A close up of a bicycle

Description automatically generated

**Figure 8. Bicycle Image using all Filters**

## South\_L Image Results

The results for the South\_L Image with every filter that was required in the project is shown below side-by-side.

![A screenshot of a cell phone

Description automatically generated]()

**Figure 9. South\_L Image using all Filters**

## Edge Image Results

![A screenshot of a cell phone

Description automatically generated]()The results for the Edge Image with every filter that was required in the project is shown below side-by-side.

**Figure 10. Edge Image using all Filters**

References

[1] "User Interface — Opencv 2.4.13.7 Documentation". *Docs.Opencv.Org*, 2020, <https://docs.opencv.org/2.4/modules/highgui/doc/user_interface.html?highlight=namedwindow>.

[2] “Sobel Edge Detection Implementation Edit.” *Sobel Edge Detection Implementation - OpenCV Q&A Forum*, <https://answers.opencv.org/question/110726/sobel-edge-detection-implementation/>.

[3] *Edge Detection Tutorial*, [www.doc.gold.ac.uk/~mas02fl/MSC101/ImageProcess/edge.html](http://www.doc.gold.ac.uk/~mas02fl/MSC101/ImageProcess/edge.html)