**Name: Pravin Reddy Sereddy**

**Email: spravin1@binghmaton.edu**

**Purpose of the Project:**

The main purpose of the project is getting the knowledge of how the image data is stored in a computer and to get the knowledge of image enhancement. By image enhancement we can get the clear images from the bad images with the noise data. So, we can get reliable good images for different types of methods.

We have used the unsharp masking, Sobel operator, Laplacian of gaussian operator, and scale space filters used for the edge detection and enhancement.

And all these methods, implementations and results are explained in the below section

**Programs required environment setup and Running the Programs:**

There is 1 program file in this zipped file

**Project\_2.cpp** is for the given grey scale image we apply the unsharp masking for image enhancement and Sobel operator for the edge detection, and Laplacian of gaussian for the better edge detection of the image.

**Required Environment setup:**

* Opencv 3.4
* Linux based system with g++ compiler for compiling the c++ programs

**Running the program:**

Select the above program required and compile the program in the terminal with above environment

**For compiling:**

g++ -ggdb filename.cpp -o filename ‘pkg-config –cflags –libs opencv’

**For running:**

./filename image1 and image2

**Programming Design**

**1. unsharp masking method to enhance the edges of the images**

* First apply the convolution of gaussian smoothing filter 3x3 mask to the given image
* The gaussian filter is

1 2 1

2 4 2

1 2 1

* Then subtract the smoothened image pixel values from the original image pixel value at each position and we get the small detail image
* Now we add the detail image pixel values to the original image pixel values to get the final unsharp image with better edge enhanced image

**2. Sobel edge detection method for the given grey scale image**

* Take the given image and apply the convolution mask S(x) to the image
* Store all the values in the separate matrix sobelX
* And, do the same with the S(y) and store in a different matrix sobelY

-1 0 1 -1 -2 -1

S(x) = -2 0 2 S(y) = 0 0 0

-1 0 1 1 2 1

* Now take all the values from the sobelX and sobelY and do and store these values in the new image pixel values.
* Some of these values are greater than the grey scales values. So, we normalize these values to 255 fi the values are greater than 255
* Print the new image we get the Sobel edge operator image with the edge detected

**3. Laplacian of gaussian mask generation method and apply on the images**

* For implementing the mask 7x7 and the mask 11x11 we have applied this formula

****

* Substitute the sigma value and the x and y values in the formula we get the LOG mask values
* Ge the generated masks and convolute it on the images given and these new images that are generated helps in edge detection.

**Results:**

**1.**

A picture containing animal, insect, indoor, sky

Description generated with high confidence

**A statue of a person

Description generated with high confidence**

2.

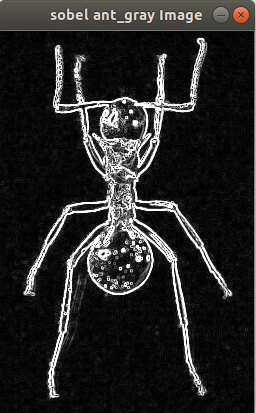
A picture containing animal, insect

Description generated with very high confidence

**A statue of a person

Description generated with very high confidence**

3.



A picture containing building, outdoor, photo

Description generated with very high confidence

4.

A picture containing device, photo, road

Description generated with very high confidence

A screenshot of a computer

Description generated with high confidence

5.

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a computer

Description generated with high confidence

5.

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a computer

Description generated with high confidence

**5. Bug Report**

While apply the LOG 11x11 mask for the image I am getting the black image. I could not get the edge detected image

**6. References:**

* [**http://academic.mu.edu/phys/matthysd/web226/Lab02.htm**](http://academic.mu.edu/phys/matthysd/web226/Lab02.htm)