Project 3: Sobel Edge Detection

# Introduction

The purpose of this assignment is to experiment with Sobel edge detection. The Sobel edge detection works by taking a derivative of a column or row of pixels of a single channel of color, and using that information to find edges and the direction of the edge. It achieves this by using a matrix (known as a mask) that is multiplied against a window of pixels surrounding a target pixel, then summing up each cell in the resulting matrix. The matrix has to be designed in a way such that small changes in color going in one direction produce a small number, while large changes in color in the same direction will produce a large number. Two matrices were used for this experiment (the variable name for the matrix is ):

A 5x5 matrix:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -5 | -4 | 0 | 4 | 5 |
| -8 | -10 | 0 | 10 | 8 |
| -10 | -20 | 0 | 20 | 10 |
| -8 | -10 | 0 | 10 | 8 |
| -5 | -4 | 0 | 4 | 5 |

A 3x3 matrix:

|  |  |  |
| --- | --- | --- |
| -10 | 0 | 10 |
| -20 | 0 | 20 |
| -10 | 0 | 10 |

The user of the program determines which matrix to use. When one of these matrices is applied to a window (of the same size as the selected matrix), and the resulting matrix is summed and that data is stored in a variable called .

Then the same matrix that was used to produce is transposed, is multiplied against the same window and it will produce .

The gradient value is produced by these two numbers:

The reason the formula includes a division by 10 is because the mask will produce numbers too large to fit in a single byte.

The gradient value is used to generate the resulting image. Below you can see a region of the image has the edge detection filter applied to it. The values in that region correspond to the values of .



To clean up the image the user can provide a threshold value that will turn the image into a binary image (see below).



For the direction of the edge the following formula is used:

The user of the program can choose a direction to filter the image so that only edges that have the chosen direction are shown. For example the same image as the previous examples has a degrees direction applied to it:



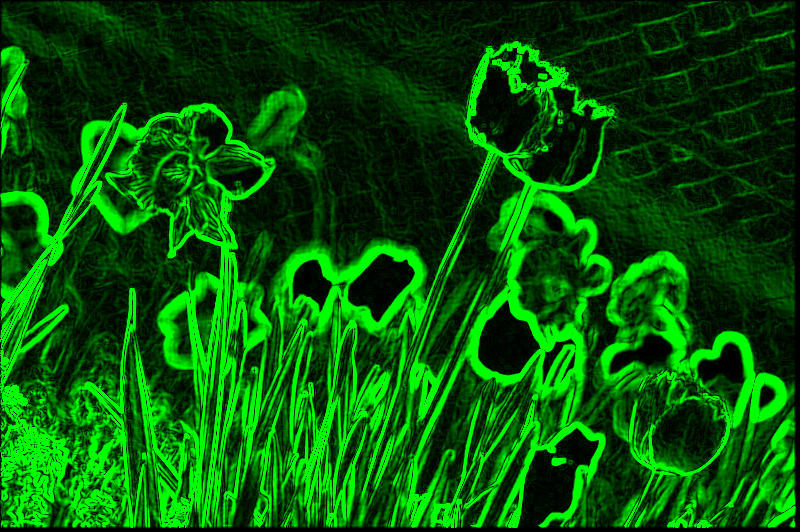
As you can see the number of edges was reduced greatly and you can also see that the continuous lines are all horizontal.

# Sobel with RGB

Sobel was designed to work on gray images, but the concept can be applied to each channel of an RGB image. Below is an example of the Sobel effect applied to each RGB channel (the original image is on top).







These images only show edges from the selected color channel, so if the edge does not exists with that color, then no edge will be shown in the resulting image. For example, the yellow flower’s center on the left side of the image does not appear in the blue edge photo. The combined images produces the following image:



# Sobel with HSI

Just like with RGB the Sobel filter can be applied to each channel in an HSI image. The following images were made using the Sobel edge detection for each HSI channel (the order is H, S, I):





Below is the Sobel filter applied to all HSI channels.

