Sobel Filter for edge detection

Daniel Amirtharaj
damirtha@buffalo.edu

1 Objective

To detect edges in the given image using the Sobel operator.

2 Analysis

2.1 Sobel operator

The Sobel operator is a 3x3 matrix, that mimics the gradient of the intensity function of an image, and gives a result which highlights edges and contours in the input image.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Figure 1: Sobel X filter and the Y filter

There are 2 operators or filters for each of direction of the image intensity function, namely x and y directions, that give gradients in the corresponding direction. This gives 2 output images which give edges in each corresponding direction. These 2 images can then be combined to form one final image.

2.2 Convolution

This is an operation that is performed on the input image and the Sobel operator, to give the output. The Sobel operator is flipped on its x and y directions, and a dot product of the flipped Sobel matrix is performed with the corresponding window in the input image to give the value of the output image's pixel (on which the window is centered).

$$O(x,y) = \sum_{(x',y')} S(x',y') * I(x-x', y-y'),\ x'\epsilon(-k,k)\ and\ y'\epsilon(-k,k)$$

Here, O is the output image, S the Sobel operator and I the input image.

2.3 Padding

While convolving the input image with the Sobel operator, it can be observed that pixels on the edge do not completely overlap the operator, this leads to an uncertainty while calculating the value of the output pixel. This ambiguity is resolved by padding the input image on all 4 edges, by either adding zeroes or repeating the pixel values in the edges itself. In this project, zero padding has been used.

3 Method

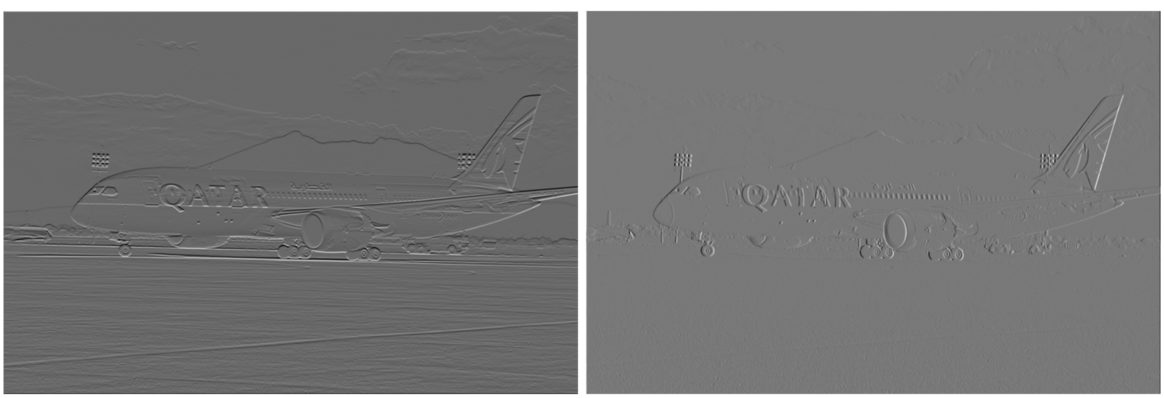
The following steps were used to obtain images using the Sobel operator.

1. Zero pad the input image (in grayscale) with one row/column of zeroes, on all 4 edges, such that the resulting resolution will be $(h{+}2)$x$(w{+}2)$, where hxw is the dimensions of the input image.
2. Convolve the x Sobel operator with the corresponding window of the input image, for each input pixel, to get the value of the output pixel for vertical edges.
3. Similarly convolve the y Sobel operator with the corresponding window of the input image, for each input pixel, to get the value of the output pixel for horizontal edges.
4. Normalize the pixel magnitudes to eliminate negative values, to obtain the gradient of the image in each direction.
5. Compute the magnitude of both horizontal and vertical edges, by taking the rms values of each pixel in the convolved image, to get edges detected in each direction.
6. Compute the combined magnitude of horizontal and vertical edges by taking the combined rms of the 2 convolved images, to give the final output image with edges detected in all directions.

4 Results

The following images, were obtained at various steps of the program.

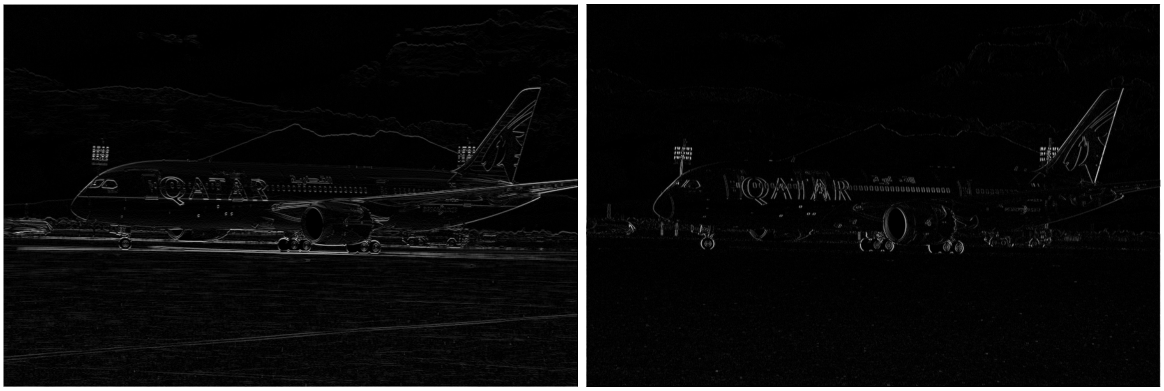
4.1 Normalized images



(a) Horizontal edges (y Gradient) (b) Vertical edges (x Gradient)

Figure 2: Normalized images after applying Sobel operators and eliminating zero values.

4.2 Edges in each direction



(a) Magnitude of Horizontal edges (from y Gradient)

(b) Magnitude of Vertical edges (from x Gradient)

Figure 3: Magnitude of edges

4.3 Final Image

Figure 4: Final image with edges in all directions after combining horizontal and vertical edge magnitudes.