Sobel Edge Detection

SOFTWARE ENGINEERING MODULE

By Prof Damir Dobric/ Prof. Adreas Pech

Name-Abhishek Sharma

Matriculation No.-1274291

Contents

Introduction

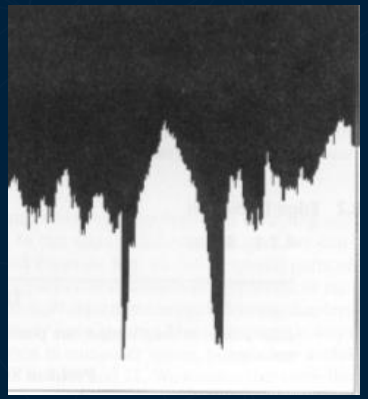


Fig: Intensity change

What is Edge Detection?

Edges in images are areas with strong intensity contrasts; a jump in intensity from one pixel to the next.

The process of edge detection significantly reduces the amount of data and filters out unneeded information, while preserving the important structural properties of an image.

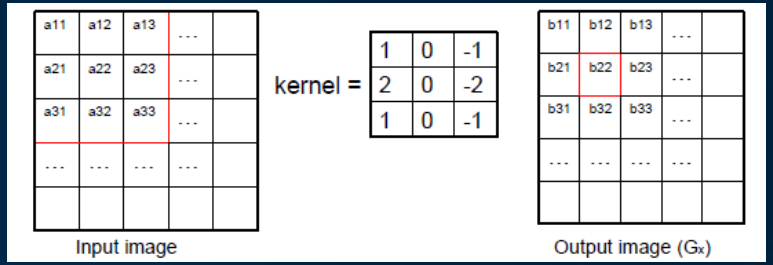
What is Sobel Edge Detection?

Sobel filter is used in image processing particularly within edge detection algorithms. Sobel operator is based on convolving the image with a small, separable and integer valued filter in horizontal and vertical direction.

Methods

The Sobel filter uses two 3 x 3 kernels. One for changes in the horizontal direction, and one for changes in the vertical direction. The two kernels are convolved with the original image to calculate the approximations of the derivatives. If we define Gx and Gy as two images that contain the horizontal and vertical derivative approximations respectively, the computations are:

$$G_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * A \quad \text{and} \quad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * A$$



Where A is the original source image.

For display purposes we can:
• map the gradient of zero onto a grey level.
– This makes negative gradients appear darker, and positive gradients appear brighter.
• Use the absolute values of the gradient map (stretched between 0 and 255).
This makes very negative and very positive gradients appear brighter

b22 = a13 - a11 + 2a23 - 2a21 + a33 - a31

The kernels contain positive and negative coefficients. This means the output image will contain positive and negative values.

The result shows how abruptly or smoothly image changes at each pixel, and therefore how likely it is that that pixel represents an edge

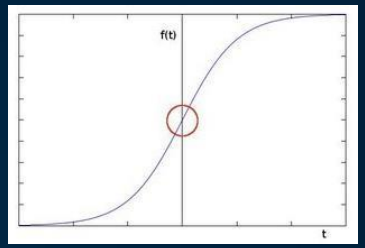
Sobel Filter Uses Gradient Method
I.e. First Derivative Approximation

$$f'(x) = \lim_{h->0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x) \quad (h=1)$$

At each pixel in the image, the gradient approximations given by Gx and Gy are combined to give the gradient magnitude and Direction :-

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$



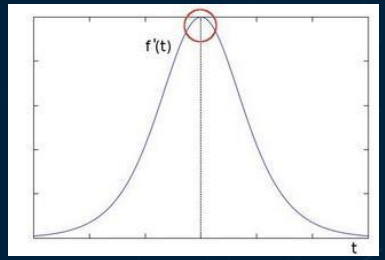
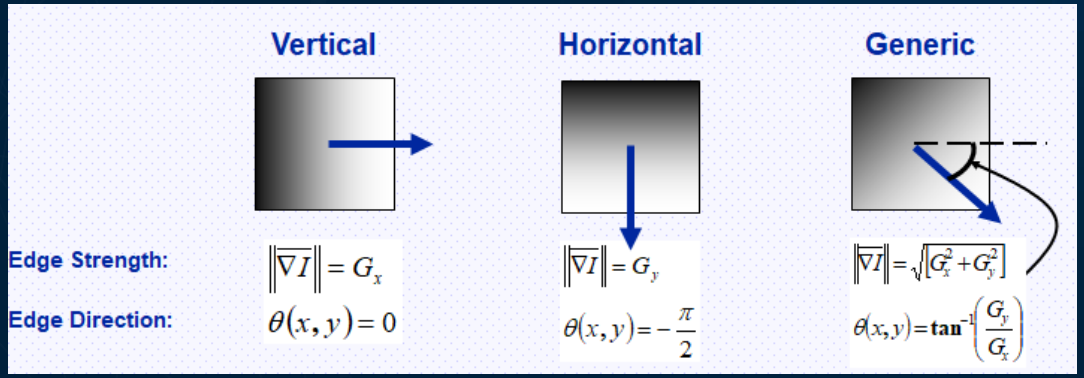


Fig: Showing Edge Jump

A angle (gradient's direction) value of 0 would indicate a vertical edge that is darker on the left side.



The direction of the edge at location ***(x,y)*** is perpendicular to the gradient vector at that point

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x-1, y)$$
$$\frac{\partial f}{\partial y} = f(x, y+1) - f(x, y-1)$$

$$g_x = h_x * f(x, y)$$
$$g_y = h_y * f(x, y)$$

RESULTS (In C- sharp)

1. Sobel Edge Algorithm is implemented in C sharp using .net core inaccordance with the architecture of Learning Api.
2. Two Projects were created one for writing the Algorithm (class library project)and other one for testing(Unit test Project).Coding methodology is test driven as per Learning Api.

Load Input Image from Input image folder(Using Bitmap)=>From unit test
Convert image into 3-D double[,,] Array data.(using conversion helper class)=> In Unit test
Run the Algorithm(using learning api method=>api.Run(from Unit test) ->image convolution +gray scale conversion happens in class library project
Convert again double[,,] data into Image(using conversion helper class)=>in unit test
Save the image in output image folder(inside bin directory)=>In unit test

Fig: Flow of program

Fig: Input Image 1

Fig: Output Image 1

More Sobel Edge Results

Fig: Input image 2

Fig: Input Gray-scale image

Fig: Output of Gray scale

Fig: output image 2

Fig: Input image 3

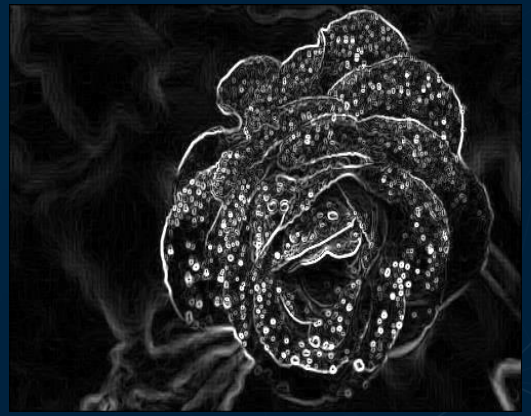


Fig: Output image 3

Discussion

Importance and Advantages of Sobel edge detection over other edge detection

1. **Edge Orientation**: The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges.

2. **Noise Environment**: Edge detection is difficult in noisy images, since both the noise and the edges contain high-frequency content. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges.

3. **Edge Structure**: Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator is chosen to be responsive to such a gradual change in those cases. Newer wavelet-based techniques actually characterize the nature of the transition for each edge in order to distinguish, for example, edges associated with hair from edges associated with a face.

Edges play quite an important role in many applications of image processing, in particular for machine vision systems that analyze scenes of man-made objects under controlled illumination conditions. Detecting edges of an image represents significantly reduction the amount of data and filters out useless information, while preserving the important structural properties in an image.

Conclusion

1. The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point I of an input grayscale image. The Sobel edge detector uses a pair of 3 x 3 convolution masks, one estimating gradient in the x direction and the other estimating gradient in Y–direction.
2. It is easy to implement than the other operators. Transferring a 2-D pixel array into statistically uncorrelated data set enhances the removal of redundant data, as a result, reduction of the amount of data required to represent a digital image. Considering data communication especially the internet, massive data transfer causes serious problems for interactive network users. Edge detection helps in optimizing network bandwidth and it is needed to keep track of data flowing in and out of the network. It helps to extract useful features for pattern recognition.
3. Sobel operator effectively highlights noise found in real world pictures as edges though, the detected edges could be thick. The Canny edge detector and similar algorithm solved these problems by first blurring the image slightly then applying an algorithm that effectively thins the edges to one-pixel. This may constitute a much slower process, hence, Sobel operator is highly recommended in massive data communication found in image data transfer.
4. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image.

Thank You