

# Edge Detection

Michael Holusa, Jan Gaura

2020-02-17

Image segmentation is an important task in image analysis. This process extracts objects from an image background. Since the objects often consists of another color (brightness) than the background, the image segmentation is usually based on the detection of object boundaries. These boundaries contains edges - a location in the image where the color (brightness) function significantly changes. Therefore, the location of edge can be determined using the derivative of the function (see Figure 1). In this exercise, you will implement two basic edge detection methods.

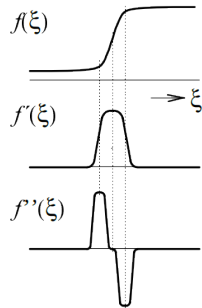


Figure 1: The brightness function and its first and second derivative.

## First Derivative

As it is seen in Figure 1, the absolute value of first derivative is high at the image point where the edge is located. Therefore, the edge is determined using derivatives  $\partial f/\partial x$  and  $\partial f/\partial y$  describing changes in the brightness function in the  $x$  and  $y$  directions. Practically, we work with the images in the discrete form, therefore, the derivatives are substituted by the differences

$$f_x(x, y) = f(x + 1, y) - f(x, y) \quad (1)$$

$$f_y(x, y) = f(x, y + 1) - f(x, y) \quad (2)$$

The size of edge is computed as

$$e(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}. \quad (3)$$

This value is computed for all image pixels. If  $e(x, y)$  is higher than a chosen threshold, it is considered as a real edge. The result of this method (in the normalized form) can be seen in Figure 4.

### Second Derivative

In Figure 1, it is shown that the edge can be detected using second derivative. The brightness function reaches the positive and negative extrema, the zero value between these two extrema determines the edge. For this purpose, the Laplacian operator can be used. Similarly as before, the brightness function is analyzed in the  $x$  and  $y$  directions, i.e.  $f_{xx} = \partial^2 f(x, y) / \partial x^2$ ,  $f_{yy} = \partial^2 f(x, y) / \partial y^2$ . The Laplacian operator is defined as

$$\nabla^2 f(x, y) = f_{xx}(x, y) + f_{yy}(x, y). \quad (4)$$

Similarly as before, due to the discrete image domain, the derivatives are replaced by the differences in the form

$$f_{xx}(x, y) = f(x-1, y) - 2f(x, y) + f(x+1, y) \quad (5)$$

$$f_{yy}(x, y) = f(x, y-1) - 2f(x, y) + f(x, y+1). \quad (6)$$

Using the Laplacian operator, the equation is in the form

$$\nabla^2 f(x, y) = f(x-1, y) + f(x+1, y) + f(x, y-1) + f(x, y+1) - 4f(x, y). \quad (7)$$

The result of this method (in the normalized form) can be seen in Figure 4.

### Sobel Operator

This operator is also based on the differences between the pixel values in the  $x$  and  $y$  directions. For computing the size of edge at  $(x, y)$ , the Sobel operator uses values of all the neighboring pixels. Let us use the labels of the neighboring pixels as it is shown in Figure 2.

$A$	$B$	$C$
$D$	$f(x, y)$	$F$
$G$	$H$	$I$

Figure 2: The labeling of image pixels.

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

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

Figure 3: The Sobel masks for the  $x$  and  $y$  directions.

The value of  $f(x, y)$  is computed as

$$f_x(x, y) = (C - A) + 2(F - D) + (I - G) \quad (8)$$

$$f_y(x, y) = (A - G) + 2(B - H) + (C - I) \quad (9)$$

These formulas can be represented as  $3 \times 3$  convolution masks as it is shown in Figure 3. Therefore, you can compute the output image using image convolution. The result of the Sobel operator can be seen in Figure 4.

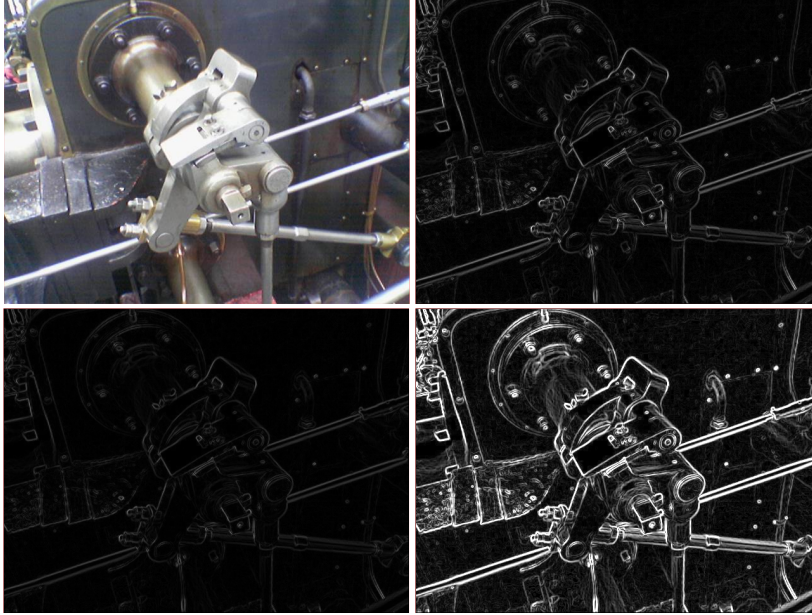


Figure 4: The input image (top left), the edges detected by the first derivative (top right), the second derivative (bottom left), and by the Sobel operator (bottom right).