Detecting Edges Using Spatial Filtering.

Elika Ketabchi and Michael Liu

Introduction

Edges are one of the gray-level discontinuities in a digital image, which is an element of image segmentation. Segmentation breaks an image into its basic parts or elements and the level to which segmentation is carried depends on the problem being solved. That is, segmentation stops when the object that we are analyzing has been isolated.

In this presentation, we will explain:

- what an intensity(gray-scale) image is in MATLAB
- how edges are detected
- how first-and second-order derivatives are applied
- how these derivatives are based on the Gradient Operator
- how filtering is used to detect edges

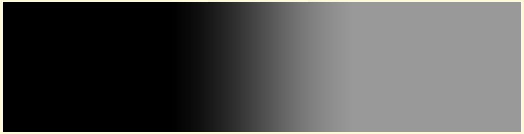
What's a Gray-scale Image?

- it's a data matrix in MATLAB
- each entry is a number ranging from 0 to 255
- 0 represents black, and 255 represents white
- everything else in between represents different shades of gray

Edge Detection

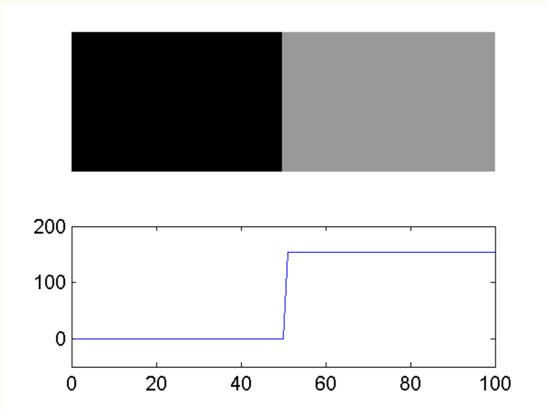
- A digital edge is made up of pixels which are connected and lie on the boundary between two regions.

This figure consists of a set of connected pixels with a sharp shift in the gray level.

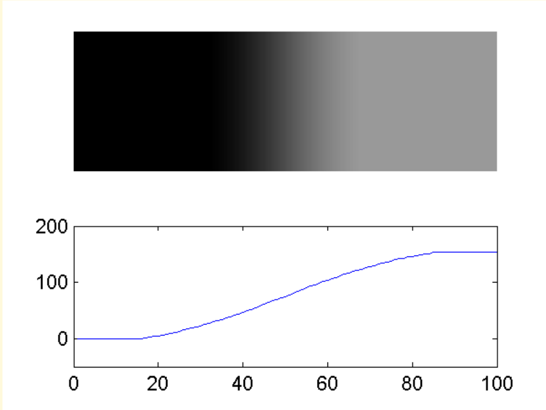


This figure shows a gradual transition from black to gray.

- By the use of MATLAB's *improfile* command we are able to plot the gray-level intensity of the following images.

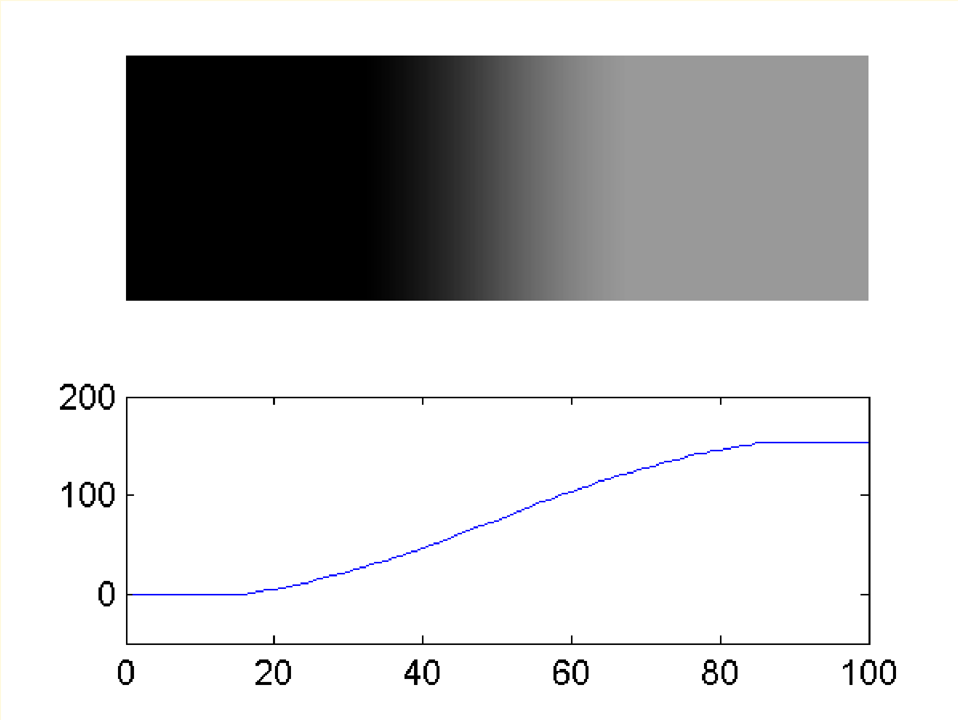


This figure consists of a set of connected pixels with a sharp shift in the gray level.



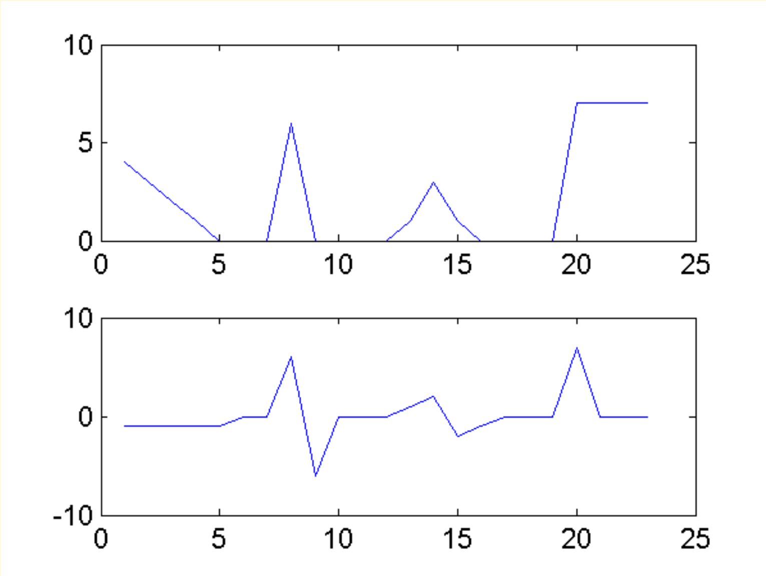
This figure shows a gradual transition from black to gray.

- The length of the ramp determines the thickness of the edge.
- Thus, fading edges have the tendency to be thick, and sharp edges have the tendency to be thin.



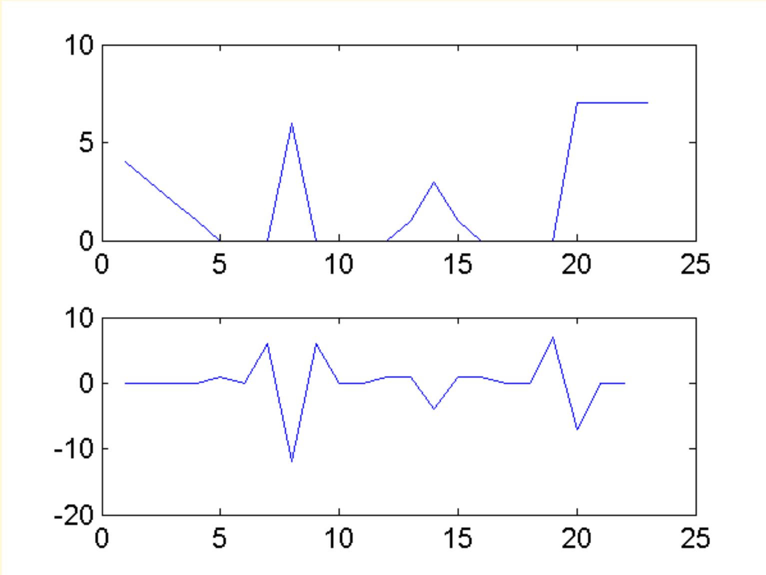
The First Derivative of the Gray-level Intensity

- must be zero in flat segments.
- must be nonzero at the start of a gray-level step or ramp.
- must be nonzero along the ramps.



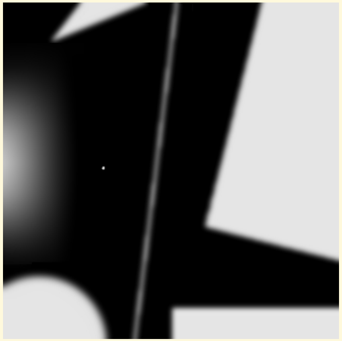
The Second Derivative of the Gray-level Intensity

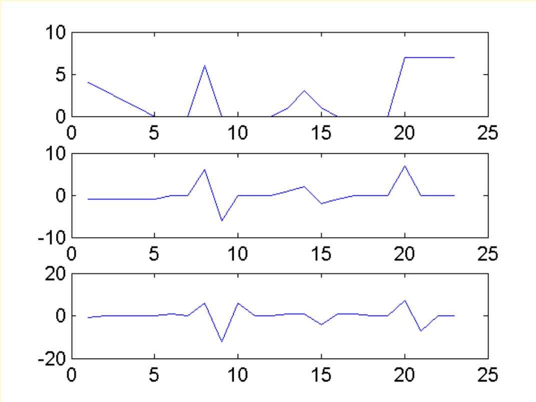
- must be zero in flat areas.
- must be nonzero at the start or end of a gray-level step or ramp.
- must be zero along ramps of constant slope.



In Summary

- First-order derivative detects "thick" edges and has a strong response to a gray-level step.
- Second-order derivative detects fine detail.
- In general, second derivative is better for image enhancement, because of its ability to detect fine detail. However, for our purposes, we will concentrate on the first derivative because of it's applicability and simplicity.





Gradient Operator

First-order derivatives of an image are based on the gradient, since an image is two dimensional. The gradient of an image $f(x, y)$ at location (x, y) is defined as the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is

$$\|\nabla f\| = [G_x^2 + G_y^2]^{1/2}.$$

As you have might guessed, the magnitude of the gradient vector will produce our edges, but the question is how?

What is Filtering?

Filtering works by multiplying each pixel in a neighborhood of an image with a corresponding coefficient, and summing all the results to get the response at each point $f(x, y)$. If the neighborhood is the size of $m \times n$, mn coefficients are necessary. The coefficients are arranged in a matrix called a filter mask.

$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
$f(x-1, y)$	$f(x, y)$	$f(x+1, y)$
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

This matrix shows the pixels of the image that are under the mask.

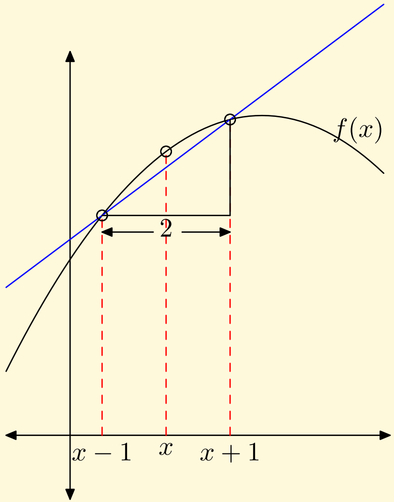
$w(-1, -1)$	$w(0, -1)$	$w(1, -1)$
$w(-1, 0)$	$w(0, 0)$	$w(1, 0)$
$w(-1, 1)$	$w(0, 1)$	$w(1, 1)$

Mask coefficients

Prewritt's Mask

Remember that a basic definition of the first derivative is

$$f'(x) = \frac{f(x+1) - f(x-1)}{2}.$$



We will ignore the the scalar value 2, because it has no significance. Now, when we take the function $f(x, y)$ and take the partial derivative with respect to x, we get

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

and when we take the partial derivative with respect to y, we get

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

Remember that the pixels of the image under a filter mask is represented by

$f(x-1,y-1)$	$f(x,y-1)$	$f(x+1,y-1)$
-1 $f(x-1,y)$	0 $f(x,y)$	1 $f(x+1,y)$
$f(x-1,y+1)$	$f(x,y+1)$	$f(x+1,y+1)$

The pixels represent where the partial derivative with respect to x will be taken. This is how the filter mask is created. The mask is essentially taking the partial derivative with respect to x when we filter the image. We will soon explain how.

$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
-1	0	1
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

Also, the rest of the entries in the first and third column of the filter mask will be replaced with -1 and 1 respectively, because those entries are also locations of where the partial derivative with respect to x will be taken.

-1	0	1
-1	0	1
-1	0	1

Finally, we will create another filter mask to determine the partial derivative with respect to y.

-1	-1	-1
0	0	0
1	1	1

We will run each mask on separate, but identical images. We will then square each entry in each of the two resulting filtered images, add them together, take the square root of the resulting matrix, and we will get

$$\|\nabla f\| = [G_x^2 + G_y^2]^{1/2}.$$

Now each entry in our resulting image (data matrix) represents the magnitude of the gradient vector.

Filtering with Prewitt's Mask

We will demonstrate in MATLAB how edge detection works with linear spatial filtering.