Image Processing - Lesson 9

Edge Detection

- Edge detection masks
- Gradient Detectors
- Compass Detectors
- Second Derivative - Laplace detectors

- Edge Linking
- Hough Transform

Image Processing - Computer Vision

Low Level

Image Processing
- representation, compression,transmission
- image enhancement
- edge/feature finding

Computer Vision
- image "understanding"

High Level

UFO - Unidentified Flying Object

Point Detection

Convolution with: $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Large Positive values = light point on dark surround
Large Negative values = dark point on light surround

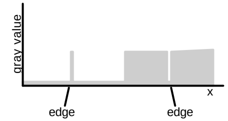
Example:

$$\begin{matrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 100 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{matrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

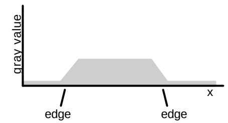
$$= \begin{matrix} 0 & 0 & -95 & -95 & -95 \\ 0 & 0 & -95 & 760 & -95 \\ 0 & 0 & -95 & -95 & -95 \end{matrix}$$

Edge Definition

Line Edge



Step Edge



Edge Detection

Line Edge Detectors

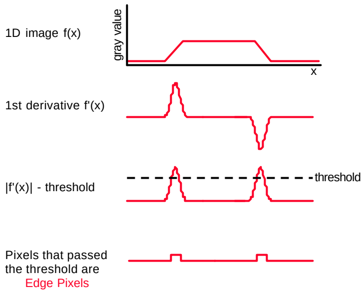
$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Step Edge Detectors

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Edge Detection by Differentiation

Step Edge detection by differentiation:

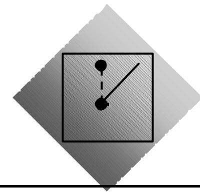


Gradient Edge Detection

Gradient $\nabla f((x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

Gradient Magnitude $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Gradient Direction $tg^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$



Differentiation in Digital Images

horizontal - differentiation approximation:

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

convolution with $[1 \ \ -1]$

vertical - differentiation approximation:

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

convolution with $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Gradient (F_A, F_B)

Magnitude $((F_A)^2 + (F_B)^2)^{1/2}$

Approx. Magnitude $|F_A| + |F_B|$

Roberts Edge Detector

$$F_A = f(x,y) - f(x-1,y-1)$$
$$F_B = f(x-1,y) - f(x,y-1)$$

$$A = \begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix}$$

Roberts and other 2x2 operators are sensitive to noise.

Prewitt Edge Detector

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothed operators

Sobel Edge Detector

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

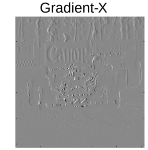
Isotropic Edge Detector

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & \boxed{0} & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & \boxed{0} & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

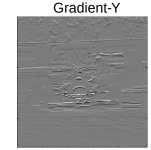
Example Edge

Original

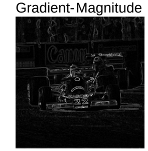
Gradient-X



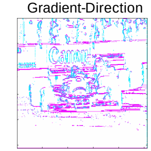
Gradient-Y



Gradient-Magnitude



Gradient-Direction



Compass Operators

$$\begin{bmatrix}1&1&1\\0&0&0\\-1&-1&-1\end{bmatrix}\ \begin{bmatrix}1&1&0\\1&0&-1\\0&-1&-1\end{bmatrix}\ \begin{bmatrix}1&0&-1\\1&0&-1\\1&0&-1\end{bmatrix}\ \begin{bmatrix}0&-1&-1\\1&0&-1\\1&1&0\end{bmatrix}$$

| N | NW | W | SW

$$\begin{bmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{bmatrix}\ \begin{bmatrix}-1&-1&0\\-1&0&1\\0&1&1\end{bmatrix}\ \begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix}\ \begin{bmatrix}0&1&1\\-1&0&1\\-1&-1&0\end{bmatrix}$$

| S | SE | E | NE

Given k operators, $g_k(x,y)$ is the image obtained by convolving f(x,y) with the k^{th} operator.

The gradient is defined as:

$$g(x,y) = \max_k g_k(x,y)$$

k defines the edge direction

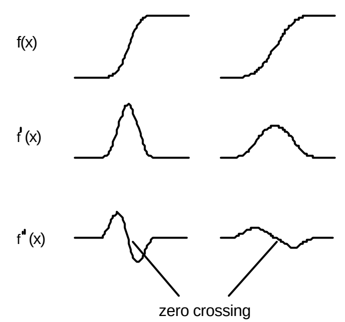
Various Compass Operators:

$$\begin{bmatrix}1&1&1\\1&\boxed{-2}&1\\-1&-1&-1\end{bmatrix}\quad \begin{bmatrix}1&1&1\\0&\boxed{0}&0\\-1&-1&-1\end{bmatrix}\quad \begin{bmatrix}1&2&1\\0&\boxed{0}&0\\-1&-2&-1\end{bmatrix}$$

$$\begin{bmatrix}5&5&5\\-3&\boxed{0}&-3\\-3&-3&-3\end{bmatrix}$$

Kirsch Edge Detector

Derivatives



Laplacian Operators

Approximation of second derivative (horizontal):

$$\frac{\partial f^2(x,y)}{\partial^2 x} = f''(x,y) = f'(x+1,y) - f'(x,y) =$$
$$= [f(x+1,y) - f(x,y)] - [f(x,y) - f(x-1,y)]$$
$$= f(x+1,y) - 2 f(x,y) + f(x-1,y)$$

convolution with: [1 -2 1]

Approximation of second derivative (vertical):

convolution with: $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

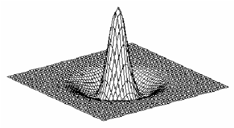
Laplacian Operator $\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$

convolution with: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

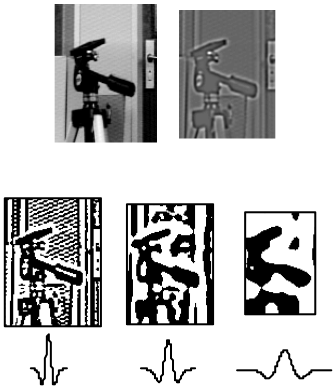
Variations on Laplace Operators:

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

All are approximations of:

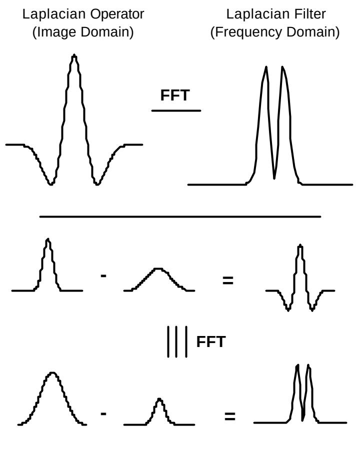


Example of Laplacian Edge Detection

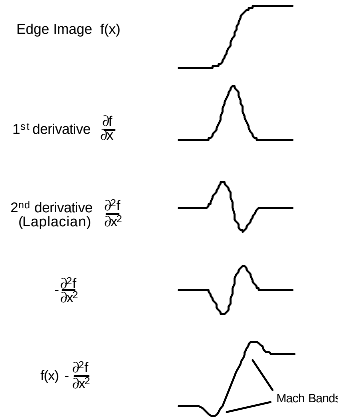


Laplacian ~ Difference of Gaussians

DOG = Difference of Gaussians



Enhancement Using the Laplacian Operator



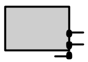
Edge Linking

(x,y) is an edge pixel.
Search for neighboring edge pixels that are "similar".

Similarity:

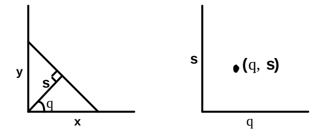
Similarity in Edge Orientation
Similarity in Edge strength (Gradient Amplitude)

Perform Edge Following along similar edge pixels.
(as in Contour Following in binary images).

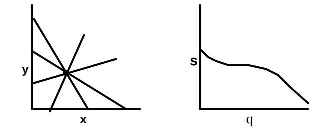


Hough Transform

Image Domain | **Hough Domain**

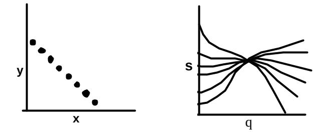


straight line | Hough Transform



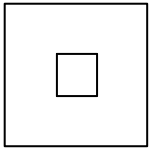
single point = many possible lines

Hough transform Method for finding line segments

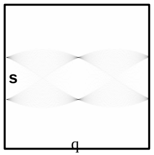


many points on a line =
many lines in the Hough transform space which intersect at 1 point.

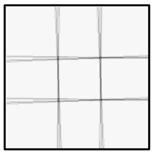
Hough Transform Example



Original square image



Hough Transform (s,θ) space

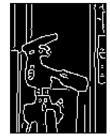


Reconstructed line segments

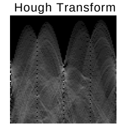
Hough Transform Example

Original

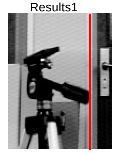
Edges



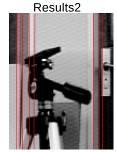
Hough Transform



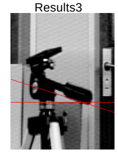
Results1



Results2

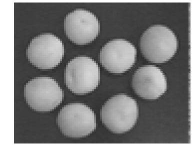


Results3

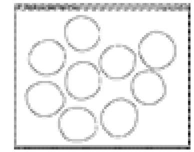


Hough Transform Example

Original



Edges



Result