

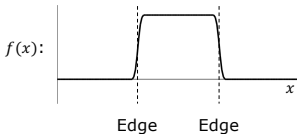
Edge Detection Using Gradients

Shree K. Nayar
Columbia University

Topic: Edge Detection, Module: Features
First Principles of Computer Vision

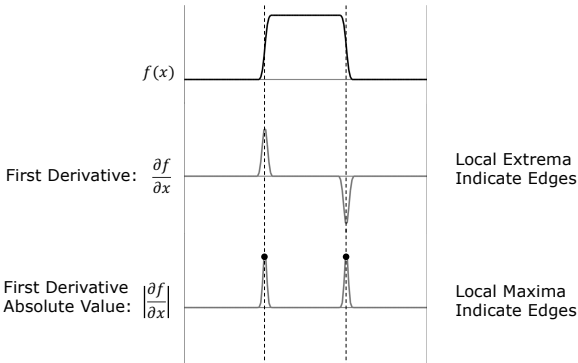
1D Edge Detection

Edge is a rapid change in image intensity in a small region.



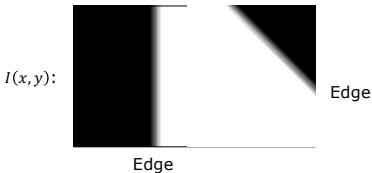
Basic Calculus: Derivative of a continuous function represents the amount of change in the function.

Edge Detection Using 1st Derivative



Provides Both Location and Strength of an Edge

2D Edge Detection



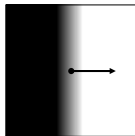
Basic Calculus: Partial Derivatives of a 2D continuous function represents the amount of change along each dimension.

Gradient (∇)

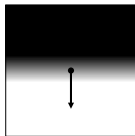
Gradient (Partial Derivatives) represents the direction of most rapid change in intensity

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

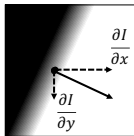
Pronounced as "Del I"



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0 \right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y} \right]$$

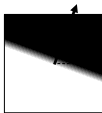


$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

Gradient (∇) as Edge Detector

Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

Gradient Orientation $\theta = \tan^{-1} \left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right)$

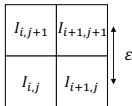


Discrete Gradient (∇) Operator

Finite difference approximations:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon} \left((I_{i+1,j+1} - I_{i,j+1}) + (I_{i+1,j} - I_{i,j}) \right)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon} \left((I_{i+1,j+1} - I_{i+1,j}) + (I_{i,j+1} - I_{i,j}) \right)$$



Can be implemented as Convolution!

$$\frac{\partial}{\partial x} \approx \frac{1}{2\varepsilon} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

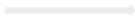
$$\frac{\partial}{\partial y} \approx \frac{1}{2\varepsilon} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Note: Convolution flips have been applied

Comparing Gradient (∇) Operators

Gradient	Roberts	Prewitt	Sobel (3x3)	Sobel (5x5)
$\frac{\partial I}{\partial x}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -3 & 0 & 3 & 2 \\ -3 & -5 & 0 & 5 & 3 \\ -2 & -3 & 0 & 3 & 2 \\ -1 & -2 & 0 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 2 & 3 & 5 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -3 & -5 & -3 & -2 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix}$
$\frac{\partial I}{\partial y}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	

Good Localization
Noise Sensitive
Poor Detection



Poor Localization
Less Noise Sensitive
Good Detection

Gradient (∇) Using Sobel Filter

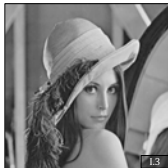
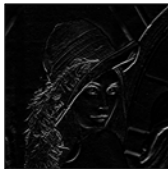


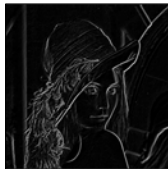
Image (I)



$\partial I / \partial x$



$\partial I / \partial y$



Gradient Magnitude 17

Edge Thresholding

Standard: (Single Threshold T)

$\|\nabla I(x, y)\| < T$ Definitely Not an Edge

$\|\nabla I(x, y)\| \geq T$ Definitely an Edge

Hysteresis Based: (Two Thresholds $T_0 < T_1$)

$\|\nabla I(x, y)\| < T_0$ Definitely Not an Edge

$\|\nabla I(x, y)\| \geq T_1$ Definitely an Edge

$T_0 \leq \|\nabla I(x, y)\| < T_1$ Is an Edge if a Neighboring Pixel
is Definitely an Edge

Sobel Edge Detector

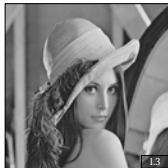
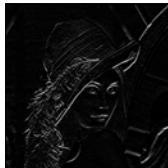


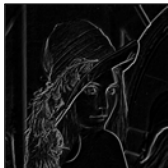
Image (I)



$\partial I / \partial x$



$\partial I / \partial y$



Gradient Magnitude



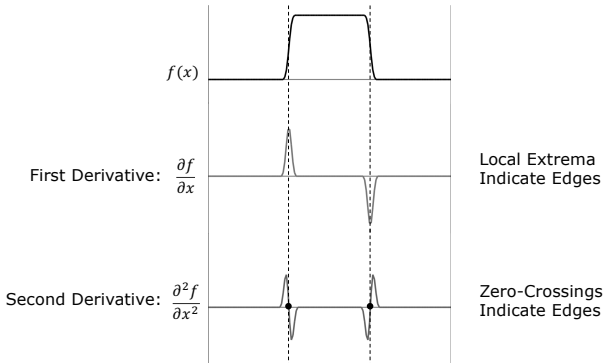
Thresholded Edge

Edge Detection Using Laplacian

Shree K. Nayar
Columbia University

Topic: Edge Detection, Module: Features
First Principles of Computer Vision

Edge Detection Using 2nd Derivative



Laplacian (∇^2) as Edge Detector

Laplacian: Sum of Pure Second Derivatives

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Pronounced as "Del Square I "

- Edges are "zero-crossings" in Laplacian of image
- Laplacian does not provide directions of edges

Discrete Laplacian(∇^2) Operator

Finite difference approximations:

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{1}{\varepsilon^2} (I_{i-1,j} - 2I_{i,j} + I_{i+1,j})$$

$$\frac{\partial^2 I}{\partial y^2} \approx \frac{1}{\varepsilon^2} (I_{i,j-1} - 2I_{i,j} + I_{i,j+1})$$

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

$I_{i-1,j+1}$	$I_{i,j+1}$	$I_{i+1,j+1}$	$\updownarrow \varepsilon$
$I_{i-1,j}$	$I_{i,j}$	$I_{i+1,j}$	
$I_{i-1,j-1}$	$I_{i,j-1}$	$I_{i+1,j-1}$	

Convolution Mask:

$$\nabla^2 \approx \frac{1}{\varepsilon^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

OR

$$\nabla^2 \approx \frac{1}{6\varepsilon^2} \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} \quad \text{(More Accurate)}$$

Laplacian Edge Detector



Image (I)



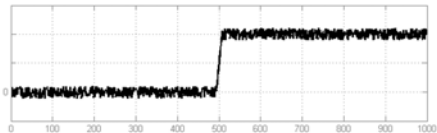
Laplacian
(0 maps to 128)



Laplacian
"Zero Crossings"

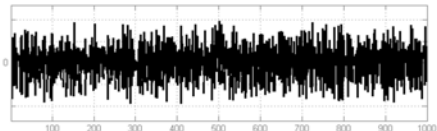
Effects of Noise

$f(x)$



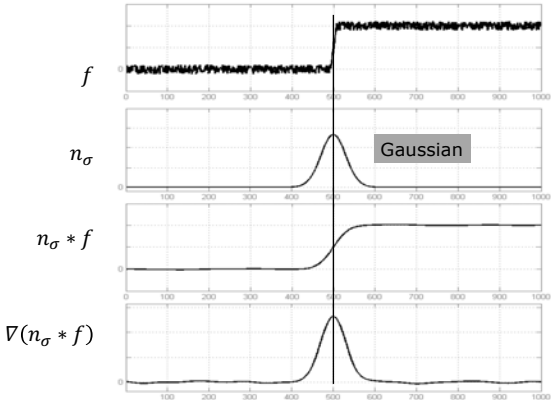
$\nabla f(x)$

(Gradient)



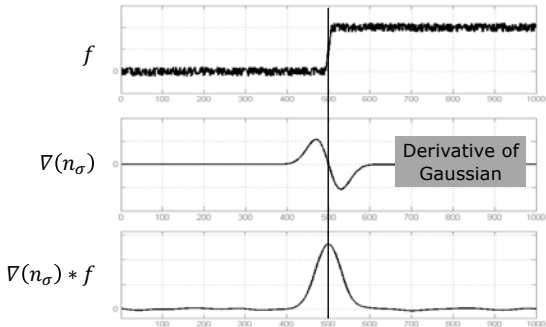
Where is the edge??

Solution: Gaussian Smooth First



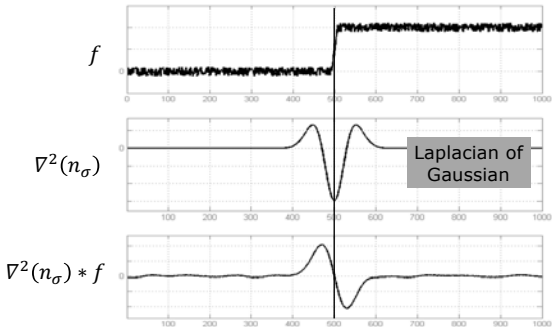
Derivative of Gaussian ($\nabla(n_\sigma)$)

$\nabla(n_\sigma * f) = \nabla(n_\sigma) * f$...saves us one operation.



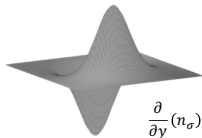
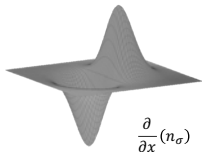
Laplacian of Gaussian ($\nabla^2 n_\sigma$ or $\nabla^2 G$)

$\nabla^2(n_\sigma * f) = \nabla^2(n_\sigma) * f$...saves us one operation.

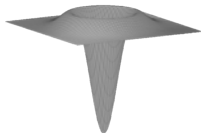


Gradient vs. Laplacian

Derivative of Gaussian (∇G)



Laplacian of Gaussian ($\nabla^2 G$)



$$\frac{\partial^2}{\partial x^2}(n_\sigma) + \frac{\partial^2}{\partial y^2}(n_\sigma)$$

Inverted "Sombrero"
(Mexican Hat)

Gradient vs. Laplacian

Provides location, magnitude and direction of the edge.	Provides only location of the edge.
Detection using Maxima Thresholding.	Detection based on Zero-Crossing.
Non-linear operation. Requires two convolutions.	Linear Operation. Requires only one convolution.

An operator that has the best of both?

Corner Detection

Shree K. Nayar
Columbia University

Topic: Edge Detection, Module: Features
First Principles of Computer Vision

Corners

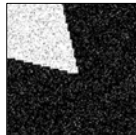
Corner: Point where Two Edges Meet. i.e., Rapid Changes of Image Intensity in Two Directions within a Small Region



"Flat" Region



"Edge" Region



"Corner" Region

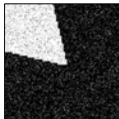
Image Gradients

Flat Region

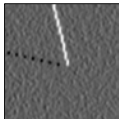
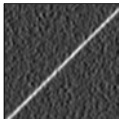
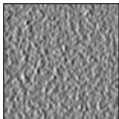
Edge Region

Corner Region

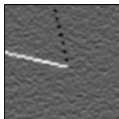
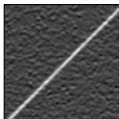
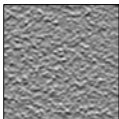
I



$I_x = \frac{\partial I}{\partial x}$



$I_y = \frac{\partial I}{\partial y}$



Distribution of Image Gradients

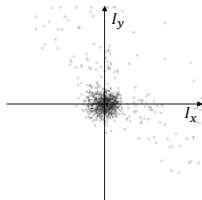
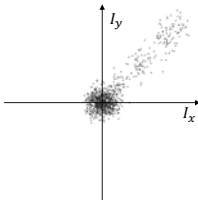
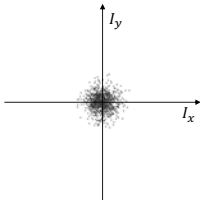
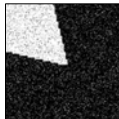
Flat Region



Edge Region



Corner Region



Distribution of I_x and I_y is different for all three regions.

Fitting Elliptical Disk to Distribution

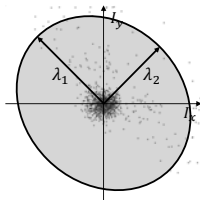
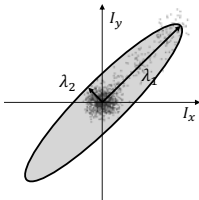
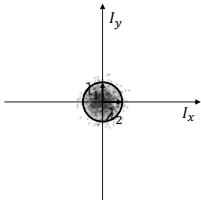
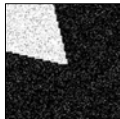
Flat Region



Edge Region



Corner Region



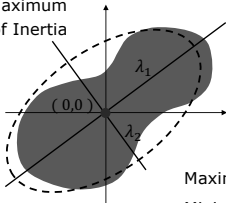
λ_1 : Length of Semi-Major Axis

λ_2 : Length of Semi-Minor Axis

Fitting an Elliptical Disk

Axis of Maximum
Moment of Inertia

Axis of Minimum
Moment of Inertia



Maximum Moment of Inertia = E_{max}

Minimum Moment of Inertia = E_{min}

(See lecture on Binary Images)

Length of Semi-Major Axis = $\lambda_1 = E_{max}$

Length of Semi-Minor Axis = $\lambda_2 = E_{min}$

Fitting an Elliptical Disk

Second Moments for a Region:

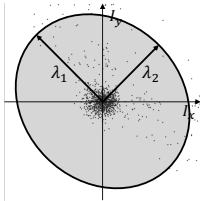
$$a = \sum_{i \in W} (I_{x_i})^2 \quad b = 2 \sum_{i \in W} (I_{x_i} I_{y_i})$$
$$c = \sum_{i \in W} (I_{y_i})^2 \quad W: \text{Window centered at pixel}$$

(See lecture on Binary Images)

Ellipse Axes Lengths:

$$\lambda_1 = E_{max} = \frac{1}{2} \left[a + c + \sqrt{b^2 + (a - c)^2} \right]$$

$$\lambda_2 = E_{min} = \frac{1}{2} \left[a + c - \sqrt{b^2 + (a - c)^2} \right]$$



Interpretation of λ_1 and λ_2

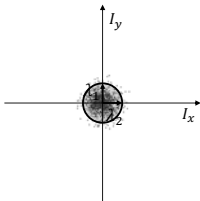
Flat Region



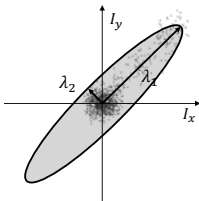
Edge Region



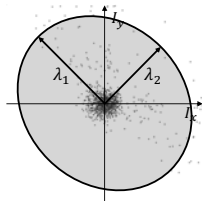
Corner Region



$\lambda_1 \sim \lambda_2$
Both are Small

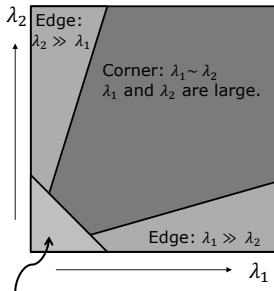


$\lambda_1 \gg \lambda_2$
 λ_1 is Large
 λ_2 is Small

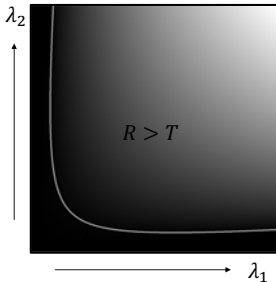


$\lambda_1 \sim \lambda_2$
Both are Large

Harris Corner Response Function



Flat: $\lambda_1 \sim \lambda_2$
 λ_1 and λ_2 are small.



$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

where: $0.04 \leq k \leq 0.06$
(Designed Empirically)
[Harris 1988]