Edge And Corner Detection

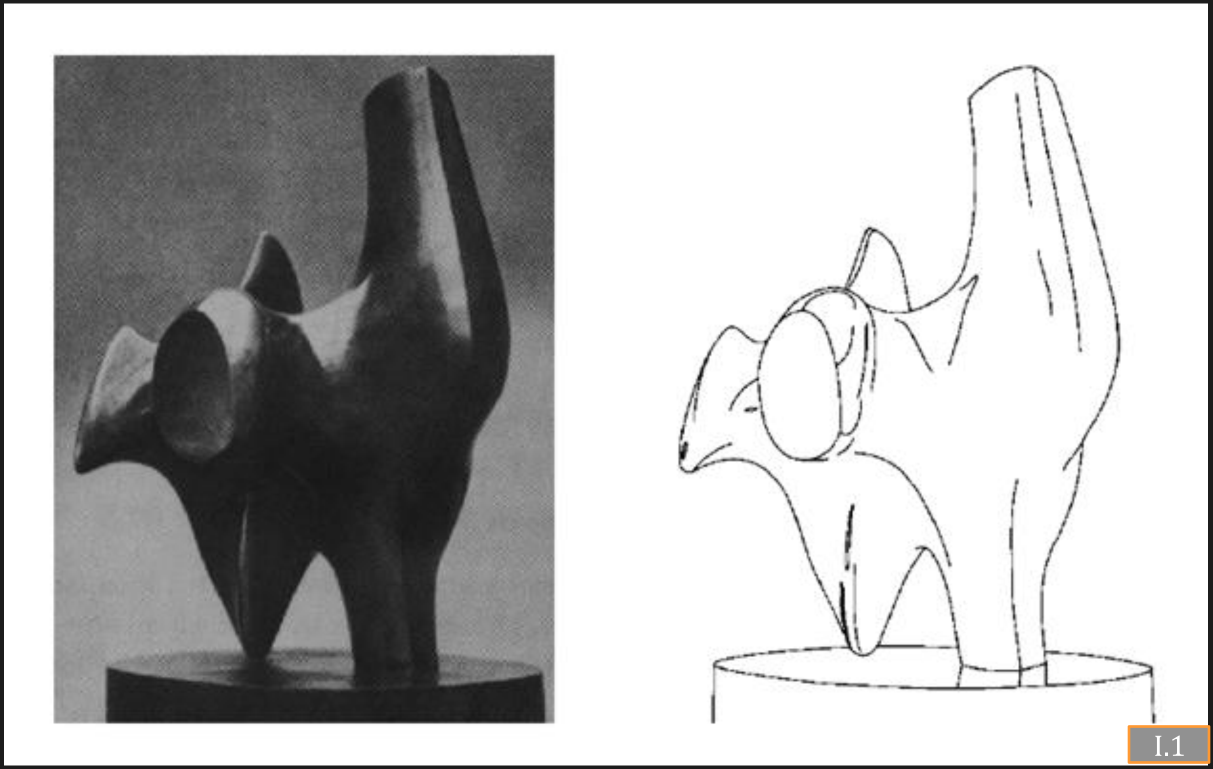
Computer Vision: CS 566

Computer Science

University of Wisconsin-Madison

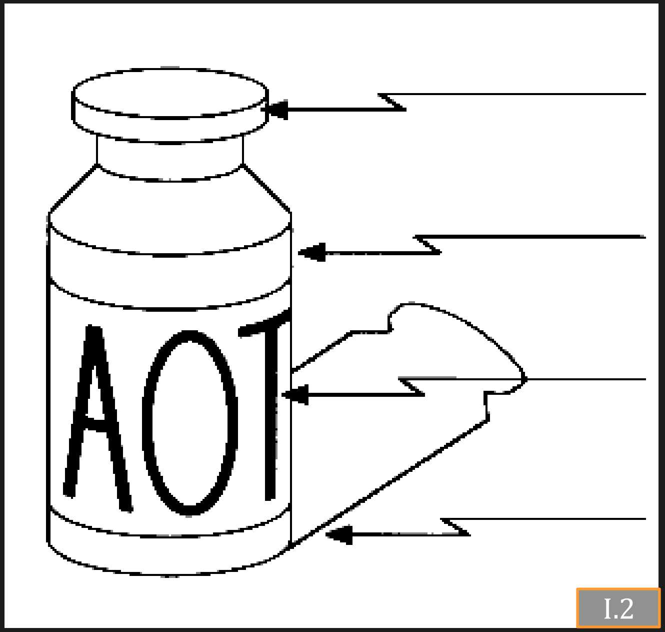
What Are Edges?

Rapid changes in image intensity within small region



Causes of Edges

Edges are caused by a variety of factors



Surface Normal Discontinuity

Depth Discontinuity

Surface Color Discontinuity

Illumination Discontinuity

Edge Detection

Convert a 2D Image into a Set of Curves

Topics:

(1) Theory of Edge Detection

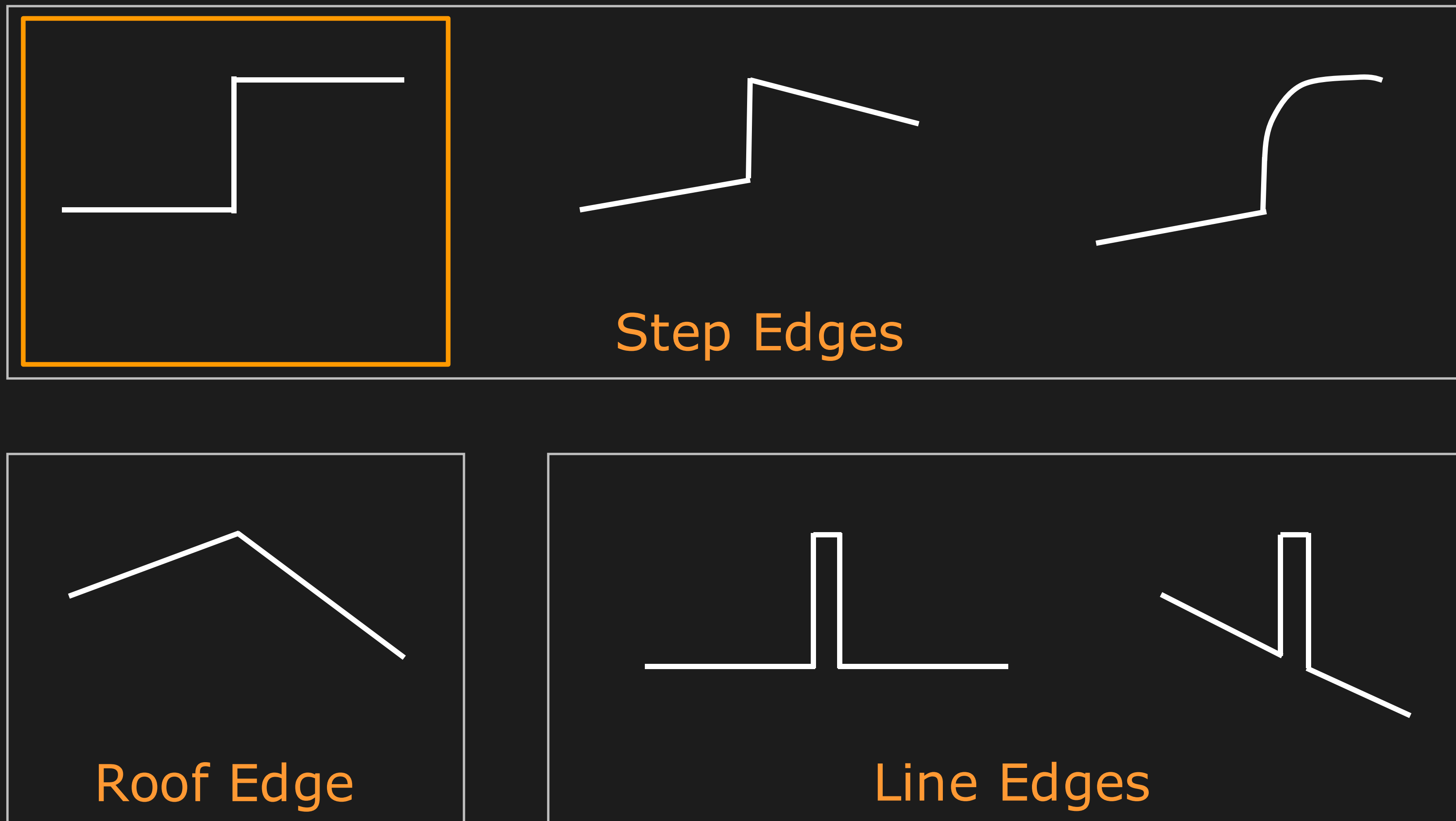
(2) Edge Detection Using Gradients

(3) Edge Detection Using Laplacian

(4) Canny Edge Detector

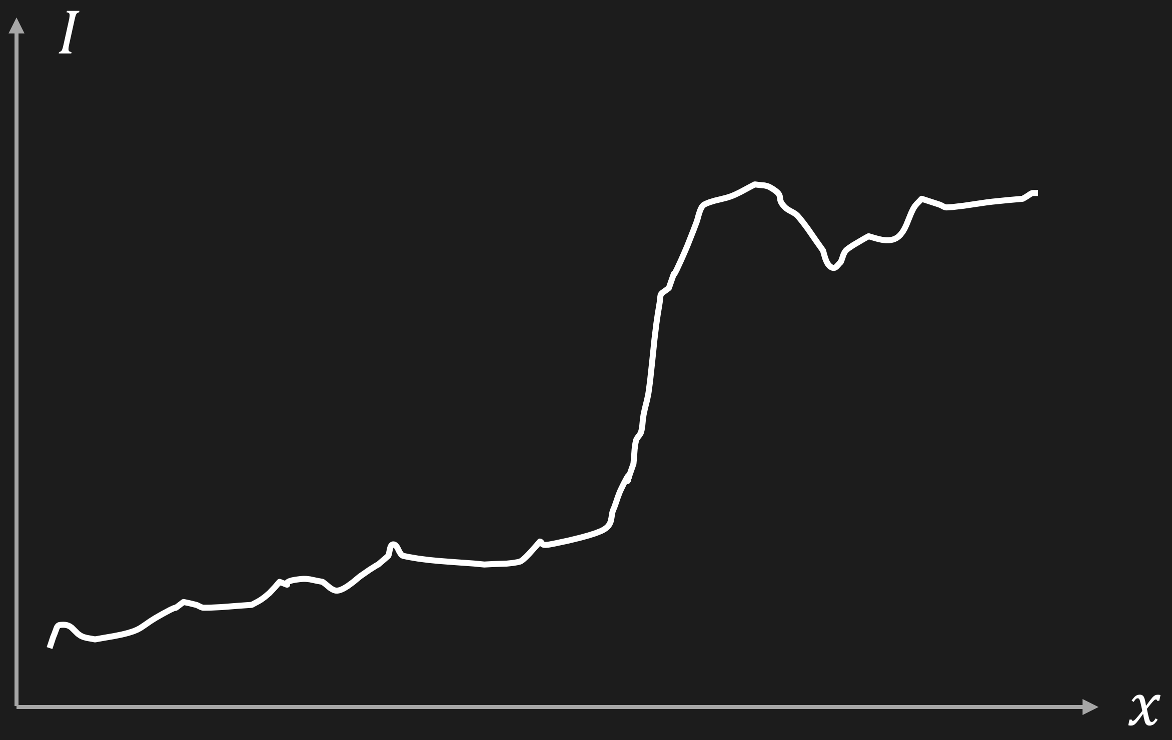
(5) Harris Corner Detector

Types of Edges



[Nalwa 1986]

Real Edges



Problems: Noisy Images and Discrete Images

Edge Detector

We want an Edge Operator that produces:

- Edge Position
- Edge Magnitude (Strength)
- Edge Orientation (Direction)

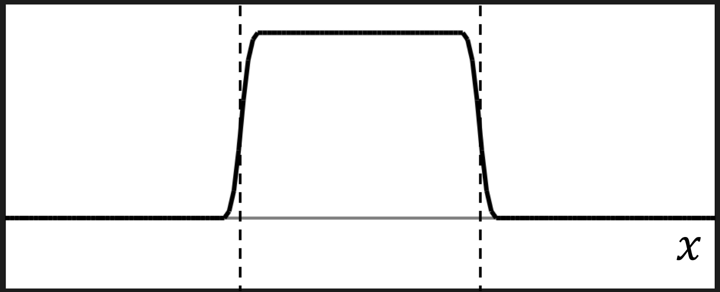
Crucial Requirements:

- High Detection Rate
- Good Localization
- Low Noise Sensitivity

1D Edge Detection

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

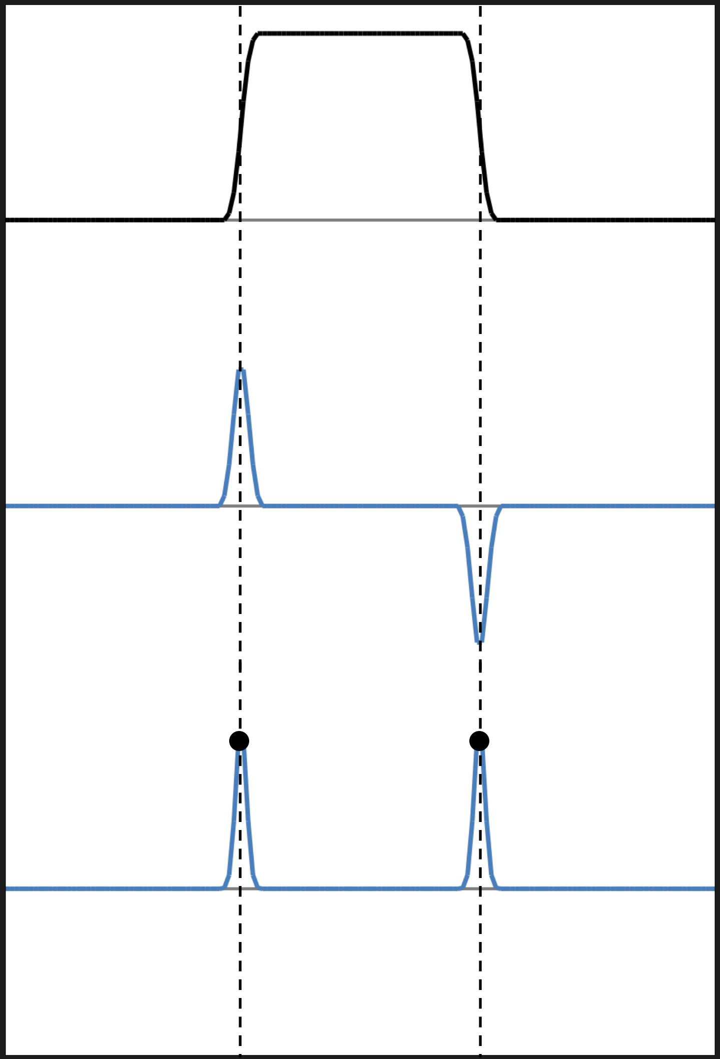
$f(x)$:



Edge Edge

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

Edge Detection Using 1st Derivative



$f(x)$

First Derivative: $\frac{\partial f}{\partial x}$

First Derivative Absolute Value: $\left|\frac{\partial f}{\partial x}\right|$

Local Extrema Indicate Edges

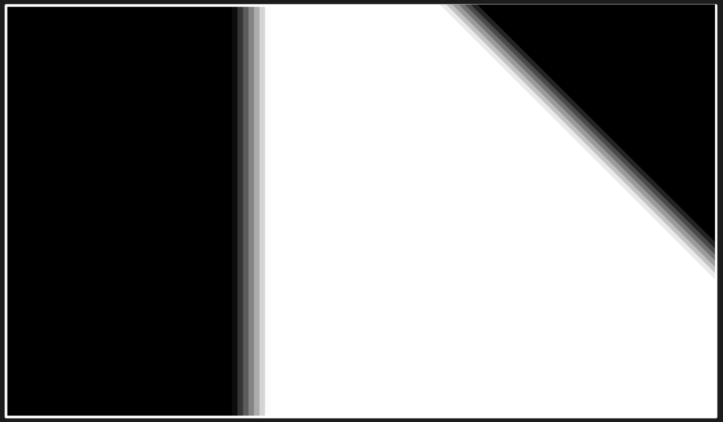
Local Maxima Indicate Edges

Provides Both Location and Strength of an Edge

2D Edge Detection

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

$I(x, y)$:



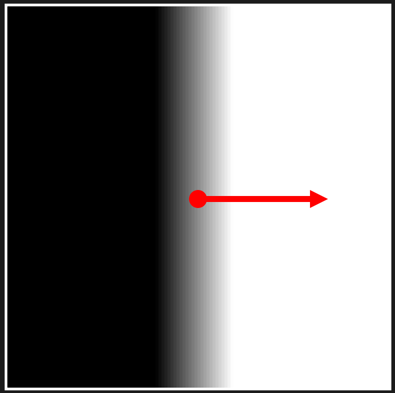
Basic Calculus: Partial Derivatives of a 2D continuous function represents the amount of change along each dimension.

Gradient (∇)

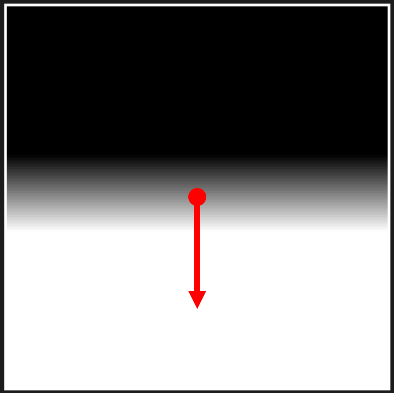
Gradient (Partial Derivative) Represents the Direction of Most Rapid Change in Intensity

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

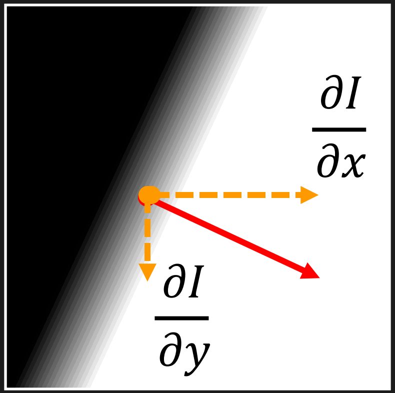
Pronounced as "Del I"



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



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

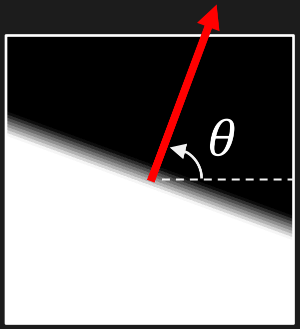


$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

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:

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

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

$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i,j}$	$I_{i+1,j}$

ε

Can be implemented as Convolution!

$$\frac{\partial}{\partial x} \approx \frac{1}{2\varepsilon}\begin{array}{|c|c|}\hline -1 & 1 \\ \hline -1 & 1 \\ \hline\end{array} \qquad \frac{\partial}{\partial y} \approx \frac{1}{2\varepsilon}\begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline\end{array}$$

Comparing Gradient (∇) Operators

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

Good Localization
Noise Sensitive
Poor Detection

$\longrightarrow$

Poor Localization
Less Noise Sensitive
Good Detection

Gradient (∇) Using Sobel Filter

Image (I)

$\partial I / \partial x$

$\partial I / \partial y$

Gradient Magnitude

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 if Definitely an Edge

Sobel Edge Detector

Image (I)

$\partial I / \partial x$

$\partial I / \partial y$

Gradient Magnitude

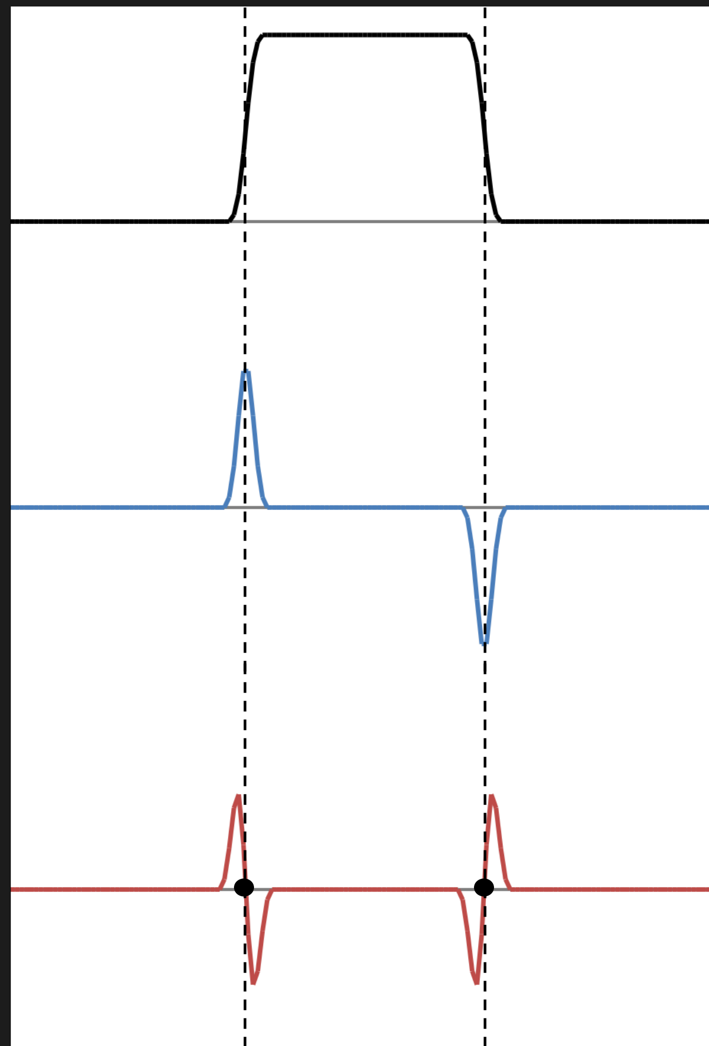
Thresholded Edge

Edge Detection Using 2nd Derivative

$f(x)$

First Derivative: $\frac{\partial f}{\partial x}$

Second Derivative: $\frac{\partial^2 f}{\partial x^2}$



Local Extrema
Indicate Edges

Zero-Crossings
Indicate Edges

Provides Only the Location of an Edge

Laplacian (∇^2) as Edge Detector

Laplacian: Sum of Second Derivatives

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

Pronounced as "Del Square I"

"Zero-Crossings" in Laplacian of an image represent edges

[Marr 1980]

Discrete Laplacian(∇^2) Operator

Finite difference approximations:

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

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

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

$I_{i-1,j+1}$	$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i-1,j}$	$I_{i,j}$	$I_{i+1,j}$
$I_{i-1,j-1}$	$I_{i,j-1}$	$I_{i+1,j-1}$

ε

Convolution Mask:

$$\nabla^2 \approx \frac{1}{\varepsilon^2}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

OR

$$\nabla^2 \approx \frac{1}{6\varepsilon^2}\begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

(More Accurate)

Laplacian Edge Detector

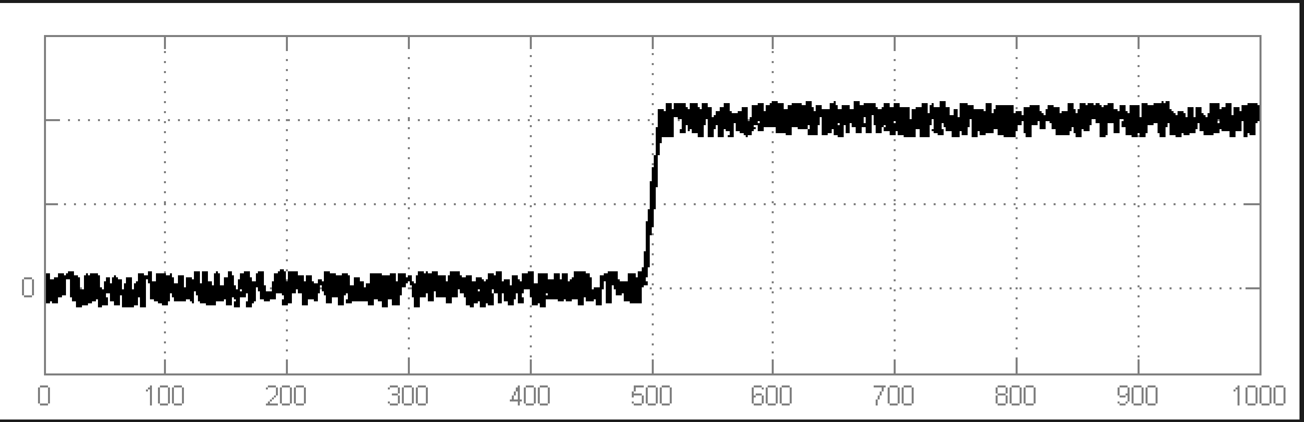
Image (I)

Laplacian
(0 maps to 128)

Laplacian
"Zero Crossings"

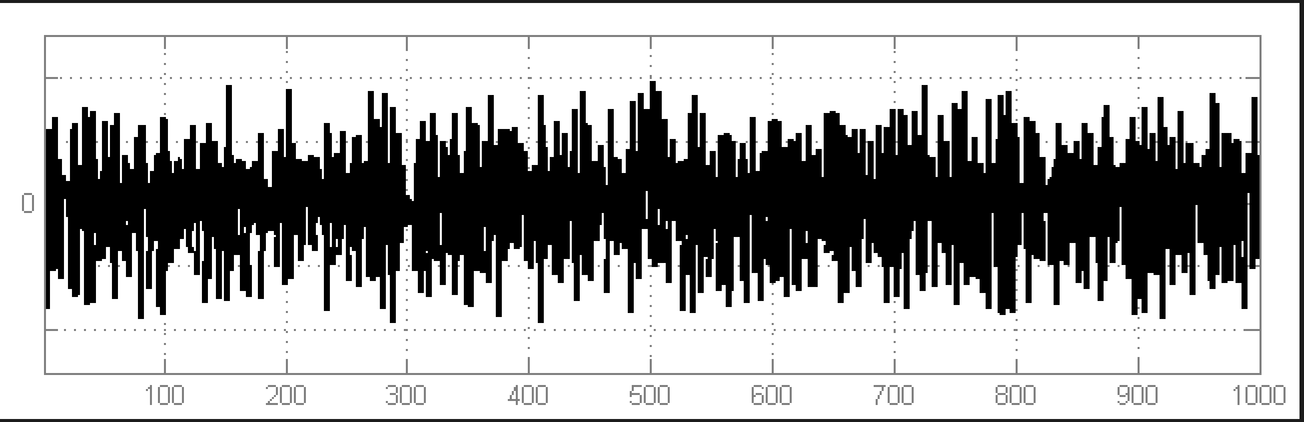
Effects of Noise

$f(x)$



$\nabla f(x)$

(Gradient)



Where is the edge??

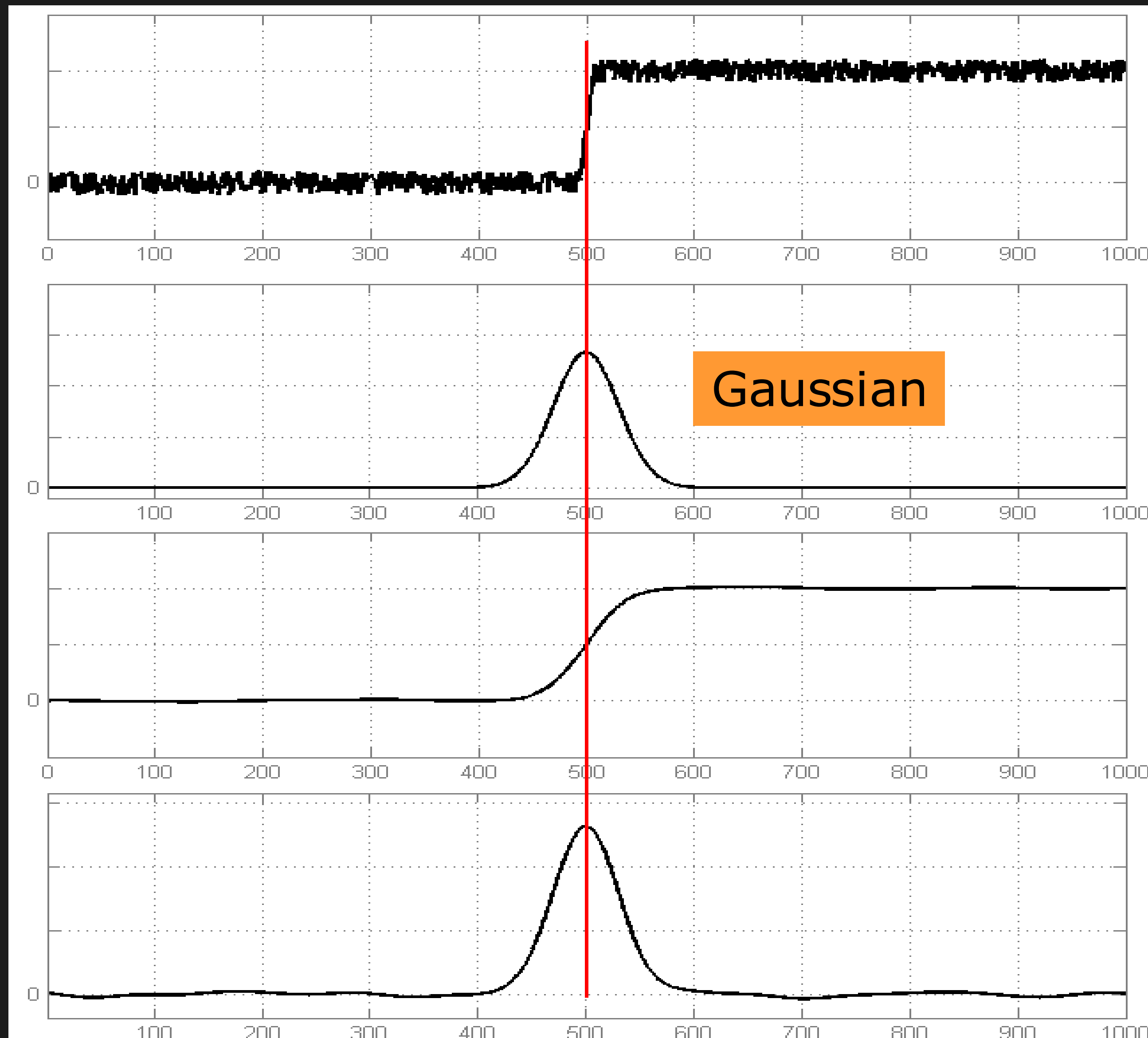
Solution: Gaussian Smooth First

f

n_σ

$n_\sigma * f$

$\nabla(n_\sigma * f)$



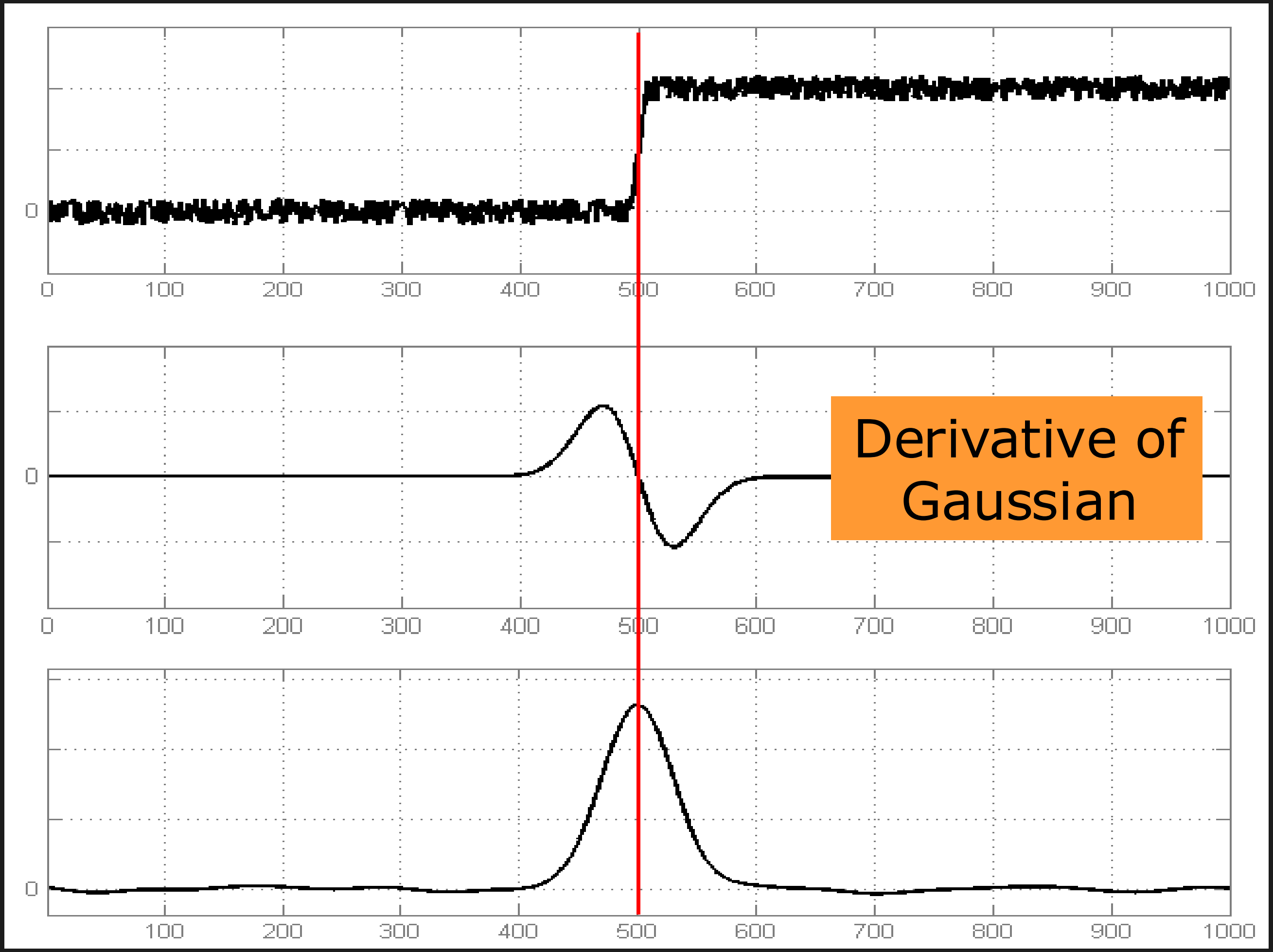
Derivative of Gaussian ($\nabla(n_\sigma)$)

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

f

$\nabla(n_\sigma)$

$\nabla(n_\sigma) * f$



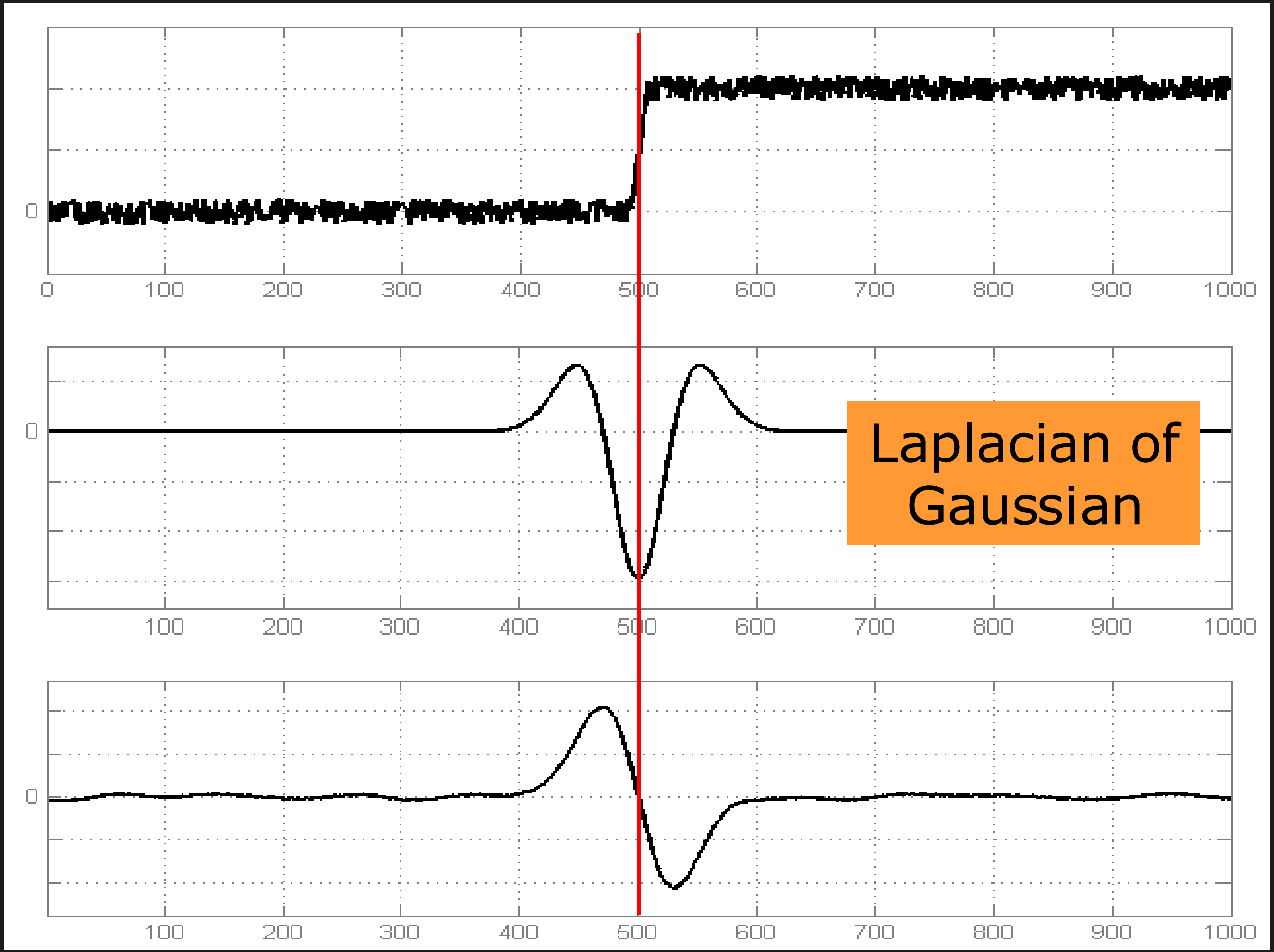
Laplacian of Gaussian ($\nabla^2 n_\sigma$ or $\nabla^2 G$)

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

f

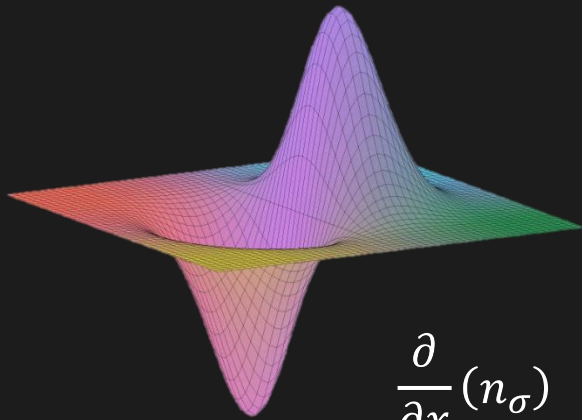
$\nabla^2(n_\sigma)$

$\nabla^2(n_\sigma) * f$

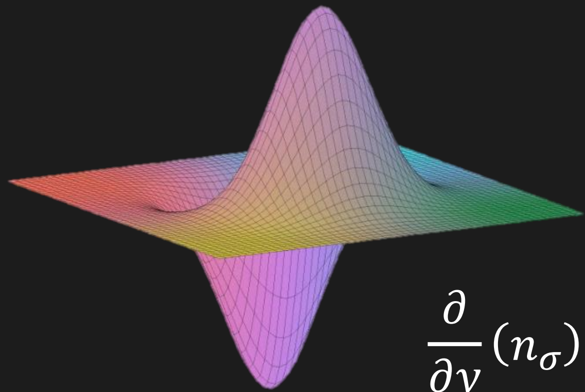


Gradient vs. Laplacian

Derivative of Gaussian (∇G)

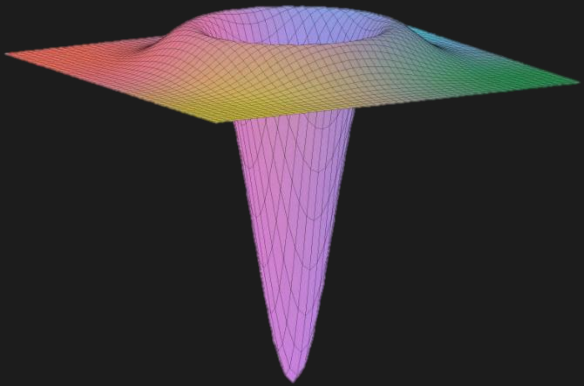


$$\frac{\partial}{\partial x}(n_\sigma)$$



$$\frac{\partial}{\partial y}(n_\sigma)$$

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



Inverted "Sombrero"
(Mexican Hat)

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

Gradient vs. Laplacian

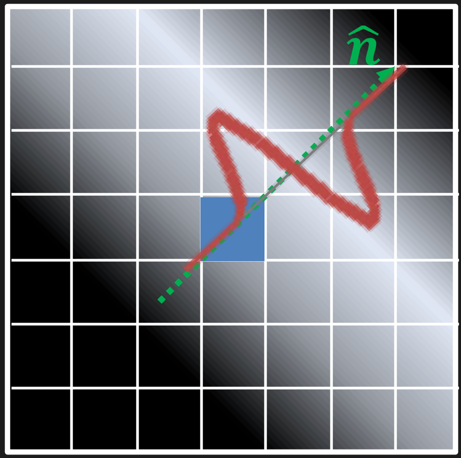
Gradient	Laplacian
Provides location, magnitude and direction of the edge.	Provides only location of the edge.
Detection using Maxima Thresholding.	Detection based on Zero-Crossing.
Non-linear operation. Requires two convolutions.	Linear Operation. Requires only one convolution.

An operator that has the best of both?

Canny Edge Detector

- Smooth Image with 2D Gaussian: $n_\sigma * I$
- Compute Image Gradient using Sobel Operator: $\nabla n_\sigma * I$
- Find Gradient Magnitude at each pixel: $\|\nabla n_\sigma * I\|$
- Find Gradient Orientation at each Pixel:

$$\hat{\boldsymbol{n}} = \frac{\nabla n_\sigma * I}{\|\nabla n_\sigma * I\|}$$



$\|\nabla n_\sigma * I\|$

- Compute Laplacian along the Gradient Direction $\hat{\boldsymbol{n}}$ at each pixel

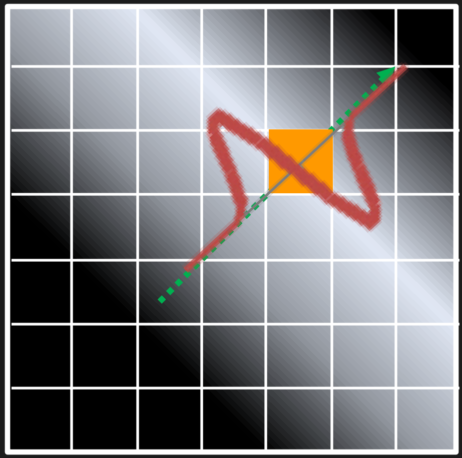
$$\frac{\partial^2 (n_\sigma * I)}{\partial \hat{\boldsymbol{n}}^2}$$

[Canny 1986]

Canny Edge Detector

- Smooth Image with 2D Gaussian: $n_\sigma * I$
- Compute Image Gradient using Sobel Operator: $\nabla n_\sigma * I$
- Find Gradient Magnitude at each pixel: $\|\nabla n_\sigma * I\|$
- Find Gradient Orientation at each Pixel:

$$\hat{\boldsymbol{n}} = \frac{\nabla n_\sigma * I}{\|\nabla n_\sigma * I\|}$$



$\|\nabla n_\sigma * I\|$

- Compute Laplacian along the Gradient Direction $\hat{\boldsymbol{n}}$ at each pixel

$$\frac{\partial^2 (n_\sigma * I)}{\partial \hat{\boldsymbol{n}}^2}$$

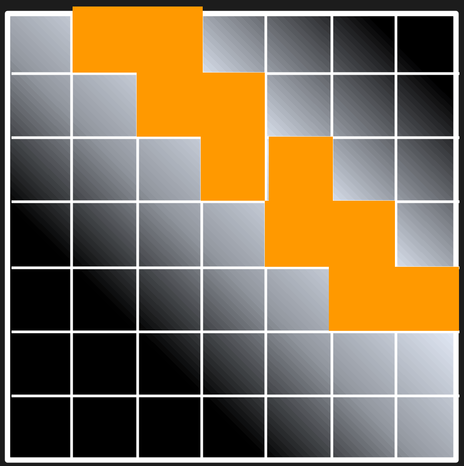
- Find Zero Crossings in Laplacian to find the edge location

[Canny 1986]

Canny Edge Detector

- Smooth Image with 2D Gaussian: $n_\sigma * I$
- Compute Image Gradient using Sobel Operator: $\nabla n_\sigma * I$
- Find Gradient Magnitude at each pixel: $\|\nabla n_\sigma * I\|$
- Find Gradient Orientation at each Pixel:

$$\hat{\boldsymbol{n}} = \frac{\nabla n_\sigma * I}{\|\nabla n_\sigma * I\|}$$



$|\nabla n_\sigma * I|$

- Compute Laplacian along the Gradient Direction $\hat{\boldsymbol{n}}$ at each pixel

$$\frac{\partial^2 (n_\sigma * I)}{\partial \hat{\boldsymbol{n}}^2}$$

- Find Zero Crossings in Laplacian to find the edge location

[Canny 1986]

Canny Edge Detector Results

Image

$\sigma = 1$

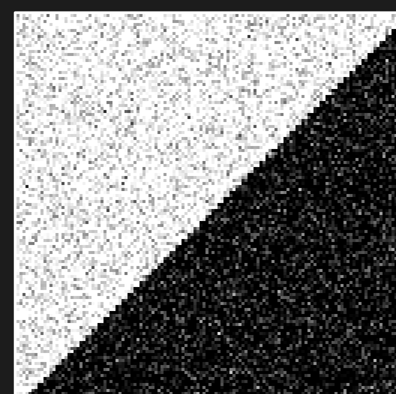
$\sigma = 2$

$\sigma = 4$

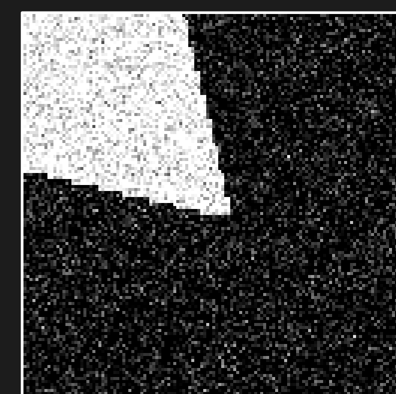
Corners

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

"Flat" Region

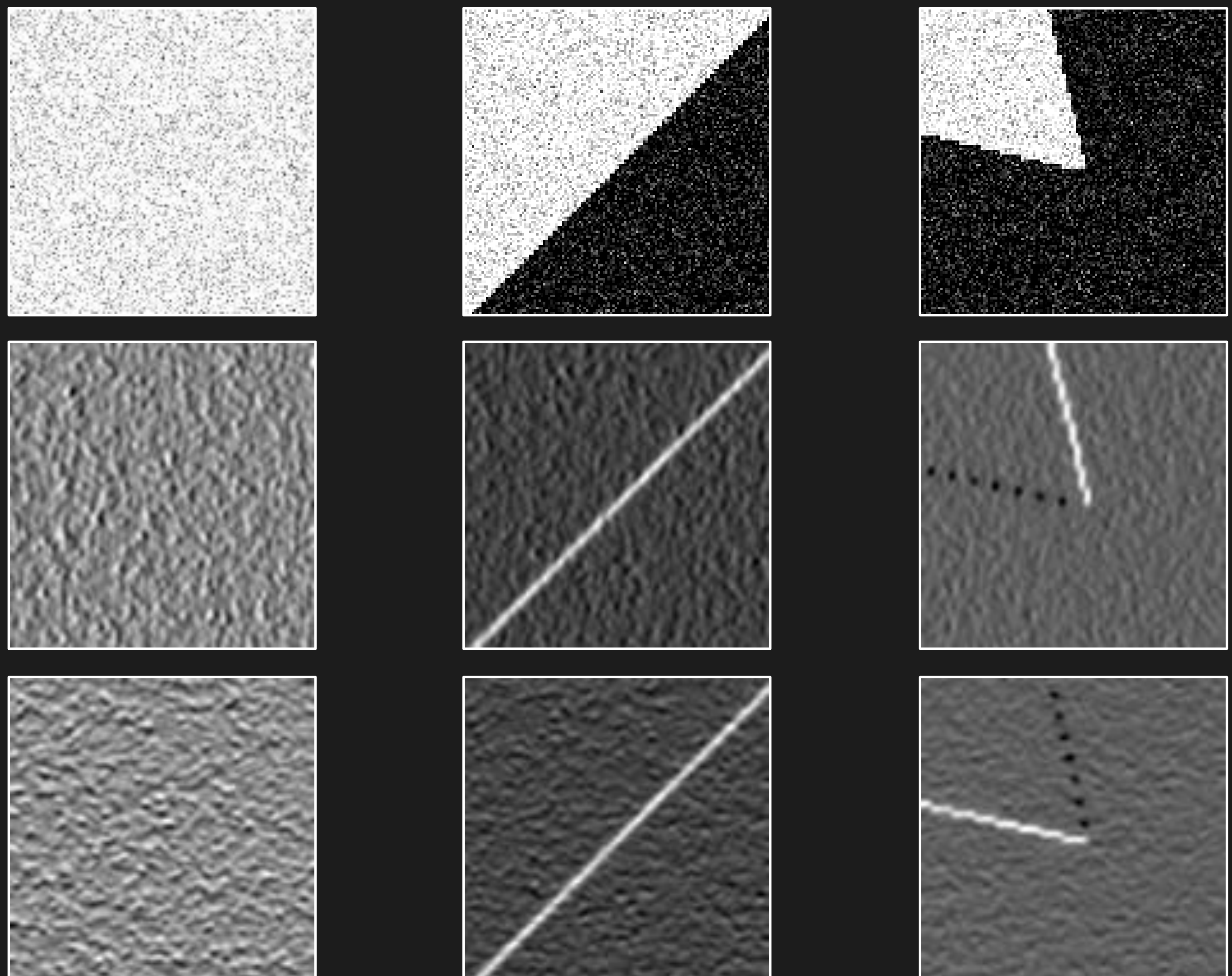


"Edge" Region



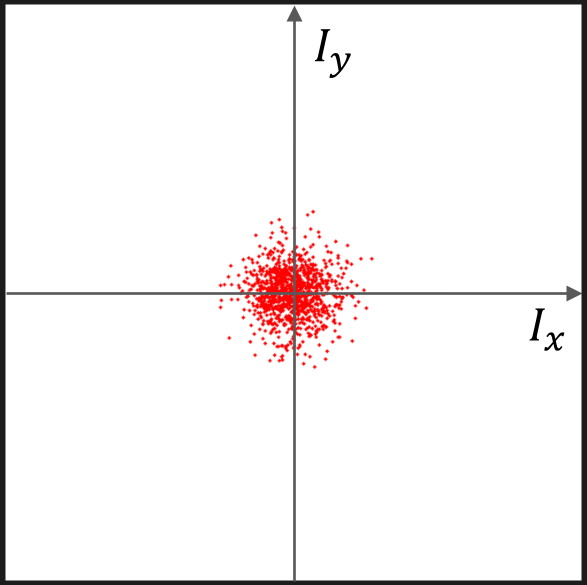
"Corner" Region

Image Gradients

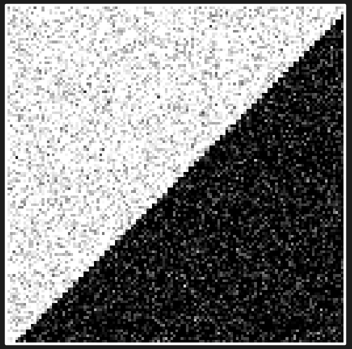


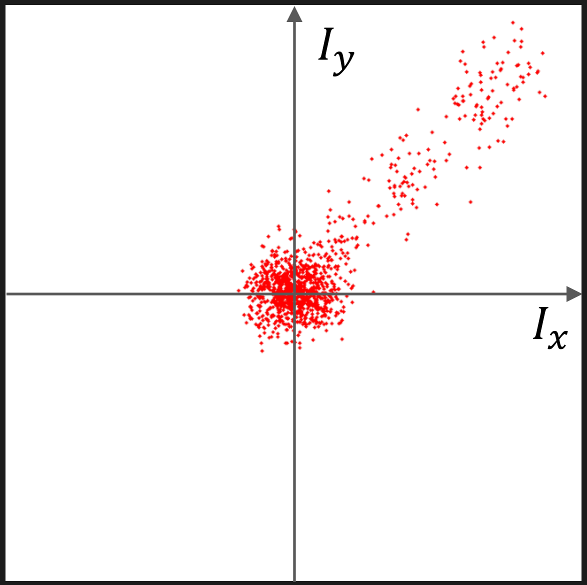
Distribution of Image Gradients

Flat Region

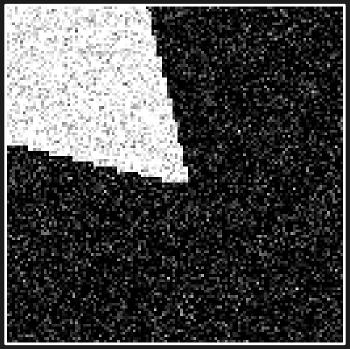


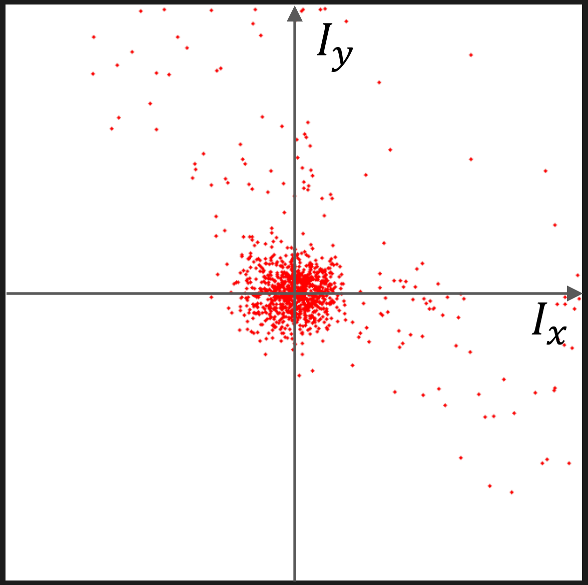
Edge Region





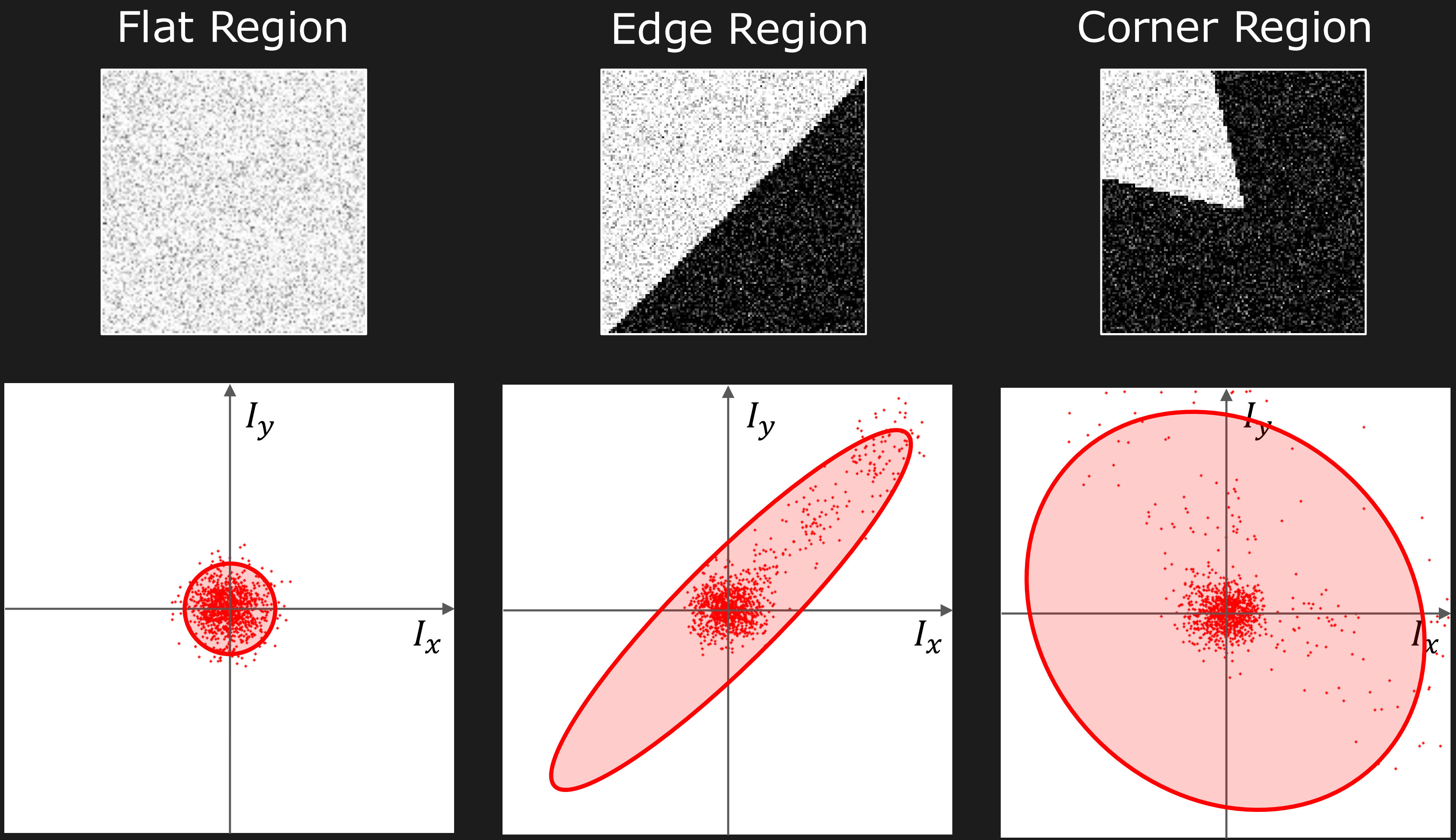
Corner Region





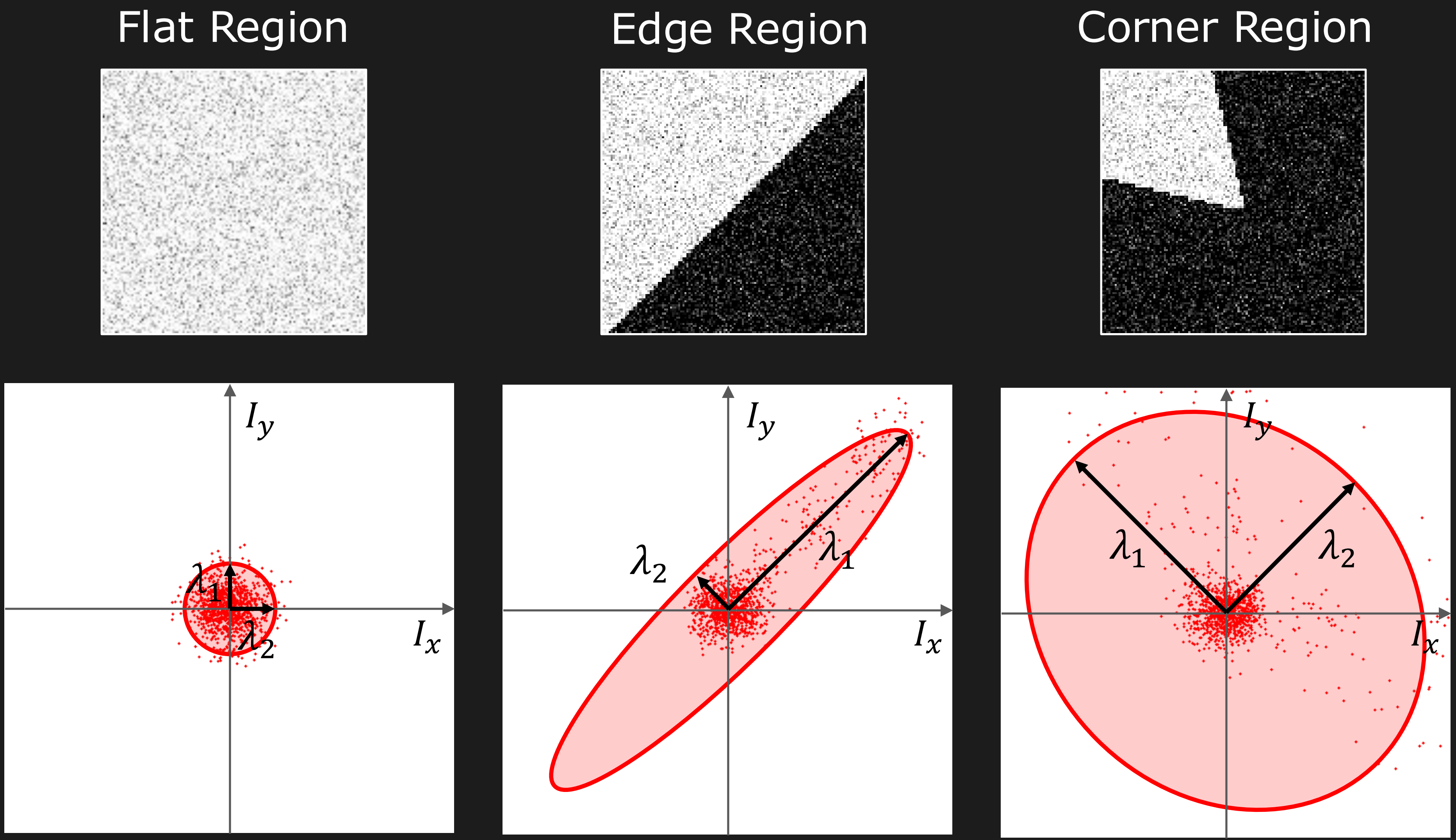
Distribution of I_x and I_y is different for all three regions.

Fitting Elliptical Disk to Distribution



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

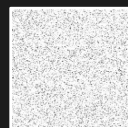
Fitting Elliptical Disk to Distribution

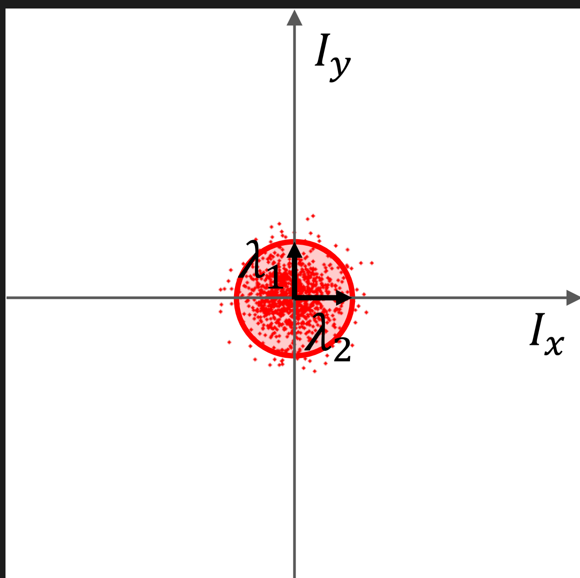


λ_1: Length of Semi-Major Axis λ_2: Length of Semi-Minor Axis

Interpretation of λ_1 and λ_2

Flat Region

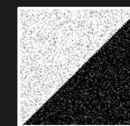


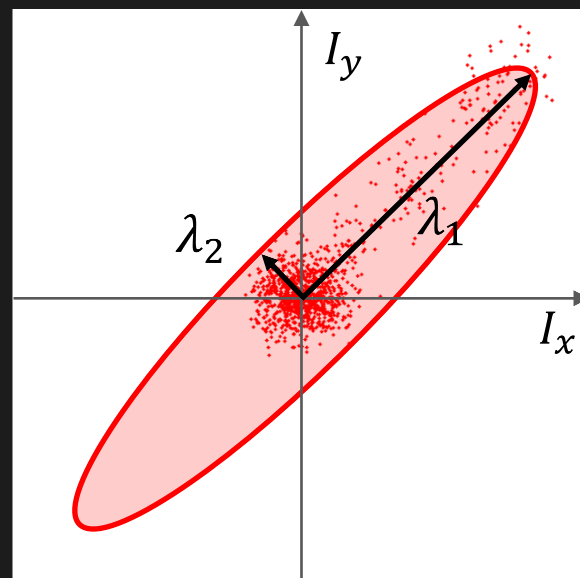


$\lambda_1 \sim \lambda_2$

Both are Small

Edge Region



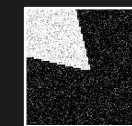


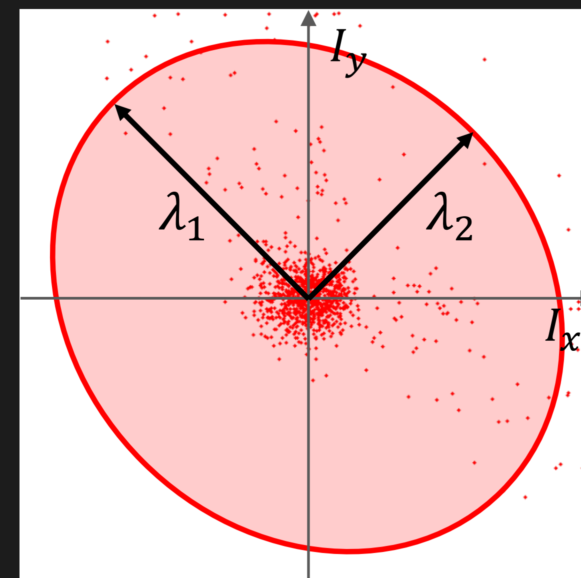
$\lambda_1 \gg \lambda_2$

λ_1 is Large

λ_2 is Small

Corner Region

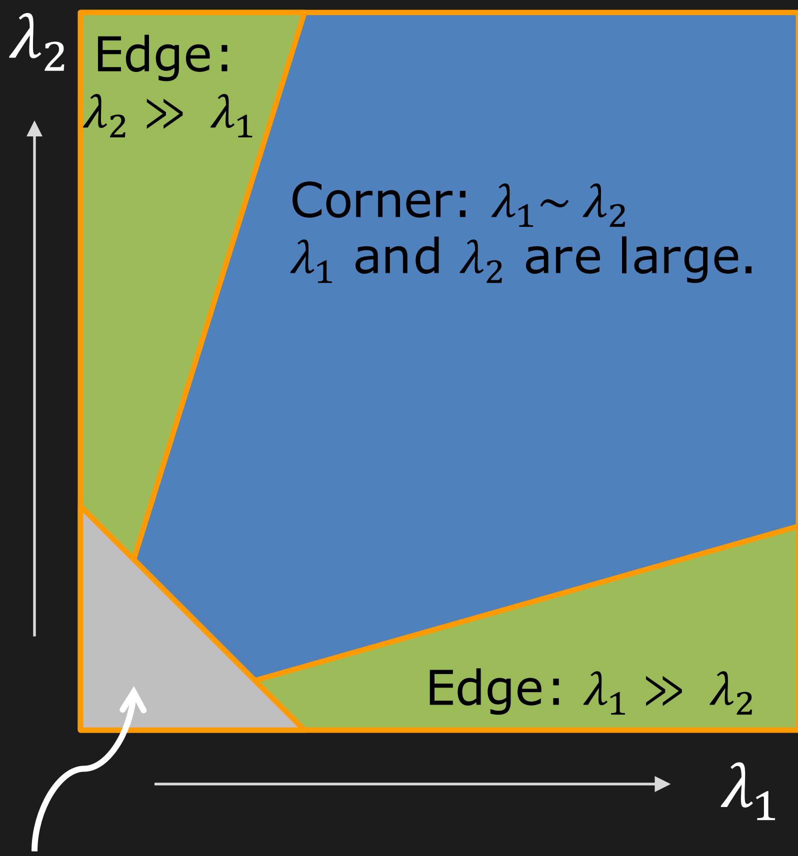




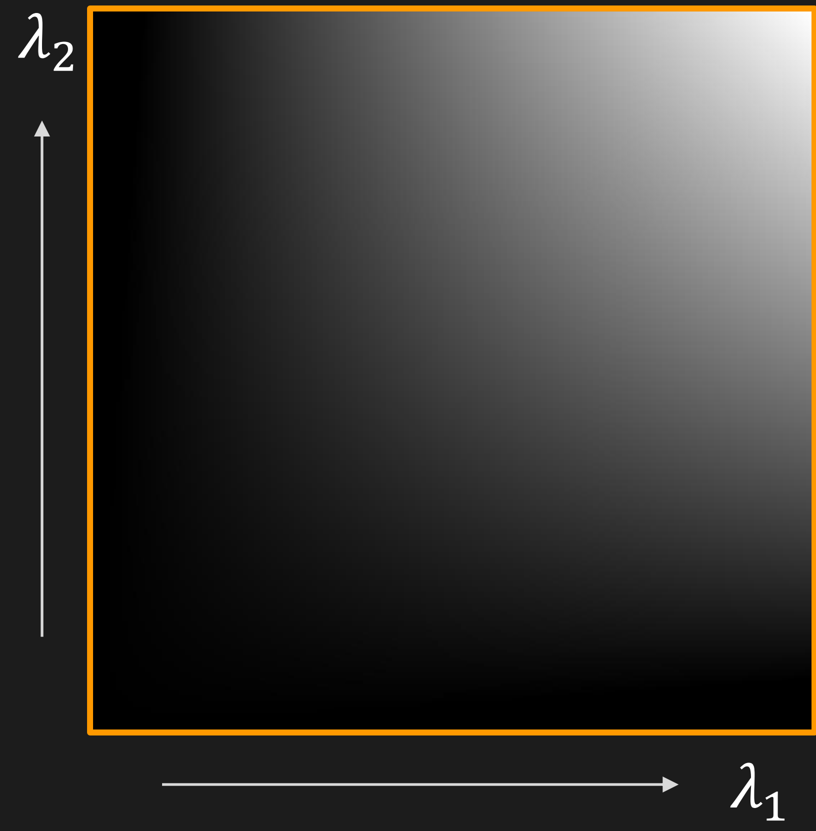
$\lambda_1 \sim \lambda_2$

Both are Large

Harris Corner Response Function



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



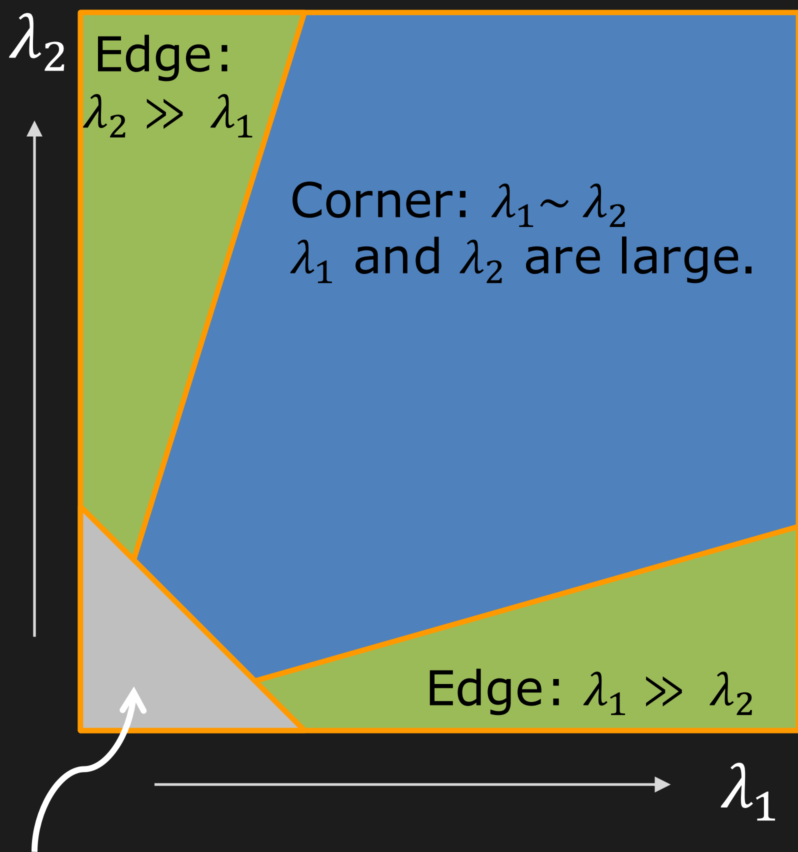
$$\boxed{R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2}$$

where: $0.04 \leq k \leq 0.06$

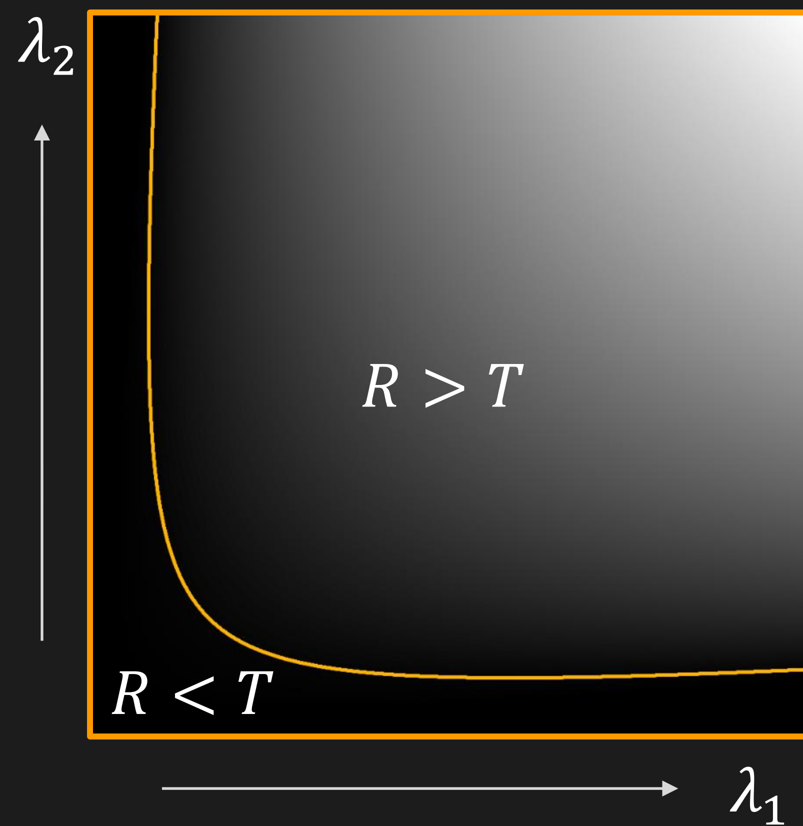
(Designed Empirically)

[Harris 1988]

Harris Corner Response Function



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



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

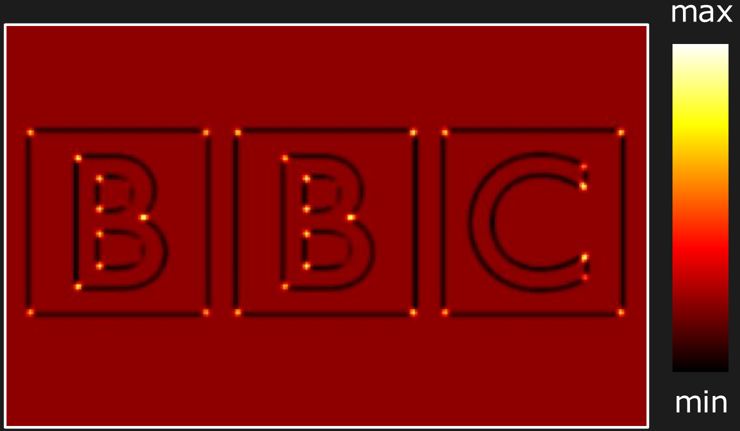
where: $0.04 \leq k \leq 0.06$

(Designed Empirically)

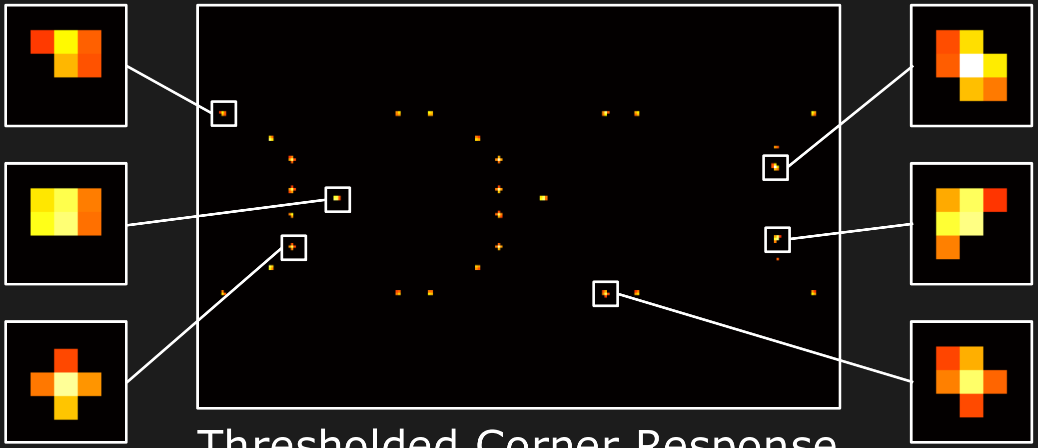
[Harris 1988]

Harris Corner Detection Example

Image



Corner Response R



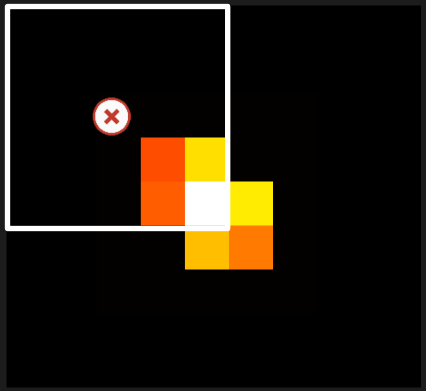
Thresholded Corner Response
$R > T$

How to determine the actual corner pixel?

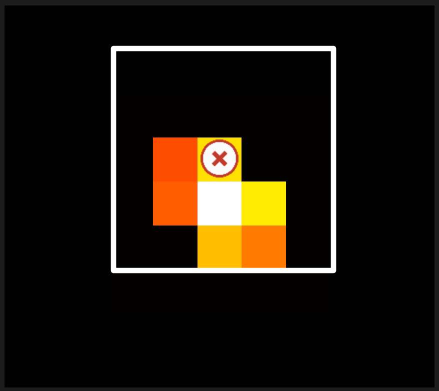
Non-Maximal Suppression

1. Slide a window of size k over the image.

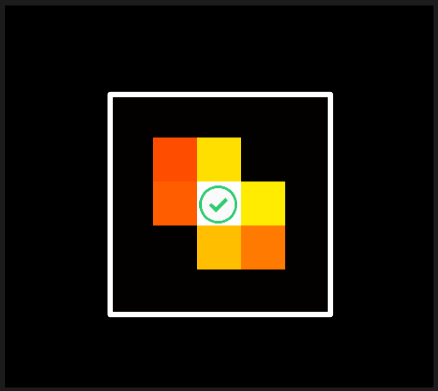
2. At each position, if the pixel at the center is the maximum value within the window, label it as positive (retain it). Else label it as negative (suppress it).



Suppress



Suppress

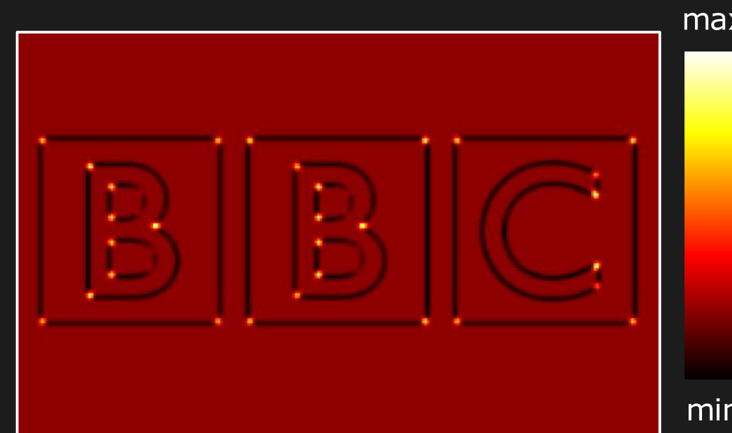


Retain

Used for finding Local Extrema (Maxima/Minima)

Harris Corner Detection Example

Image

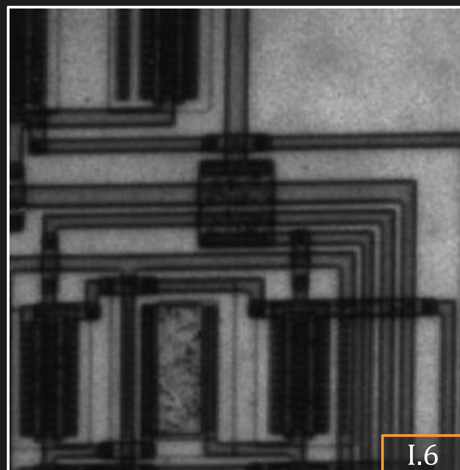


Corner Response R

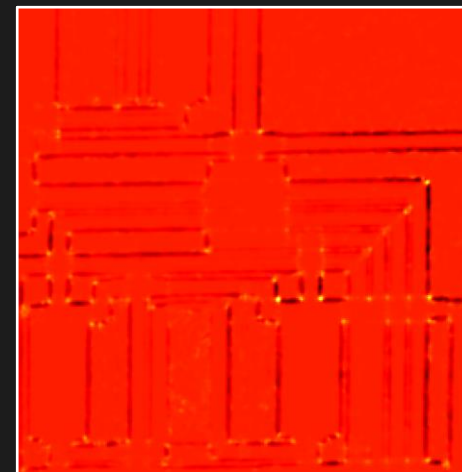
Thresholded Corner Response
$R > T$ $(T = 5.1 \times 10^7)$

Detected Corners

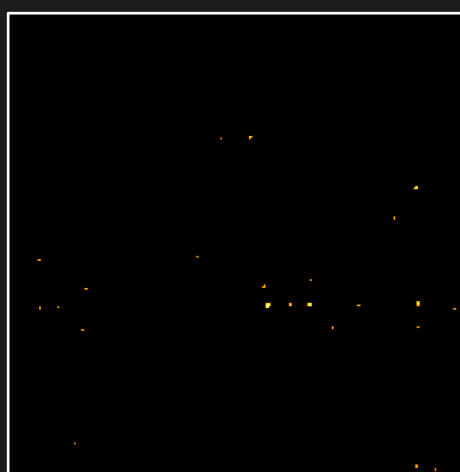
Harris Corner Detection Example



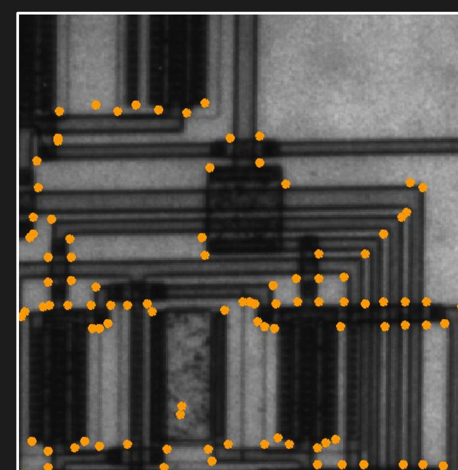
Image



Corner Response R



Thresholded Corner Response
$R > T \ (T = 5.1 \times 10^7)$



Detected Corners

References: Textbooks

A Guided Tour of Computer Vision (Chapter 3)
Nalwa, V., Addison-Wesley Pub

Computer Vision: A Modern Approach (Chapter 8)
Forsyth, D and Ponce, J., Prentice Hall

Digital Image Processing (Chapter 3)
González, R and Woods, R., Prentice Hall

Robot Vision (Chapter 8)
Horn, B. K. P., MIT Press

References: Papers

[Canny 1986] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[Harris 1988] Harris, C. and Stephens, M., A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference. pp. 147–151.

[Marr 1980] Marr, D. and Hildreth, E., "Theory of Edge Detection," Proc. R. Soc. London,B 207, 187-217, 1980.

[Nalwa 1986] Nalwa, V. S. and Binford, T. O., "On detecting edges," IEEE Trans. Pattern Analysis and Machine Intelligence, 1986.

Image Credits

I.1	Adapted from Fig 3.1, Nalwa, V., A Guided Tour of Computer Vision.
I.2	Adapted from Fig 3.3, Nalwa, V., A Guided Tour of Computer Vision.
I.3	Matlab Demo Image
I.4	http://en.wikipedia.org/wiki/File:Caf%C3%A9_wall.svg
I.5	http://www.michaelbach.de/ot/geom_KitaokaBulge/index.html
I.6	Matlab Demo Image