MAEG 5720:
Computer Vision in Practice

Lecture 4:

Interest Point Detection

Dr. Terry Chang

2021-2022

Semester 1

香港中文大學
The Chinese University of Hong Kong

Department of Mechanical and Automation Engineering
機械與自動化工程學系

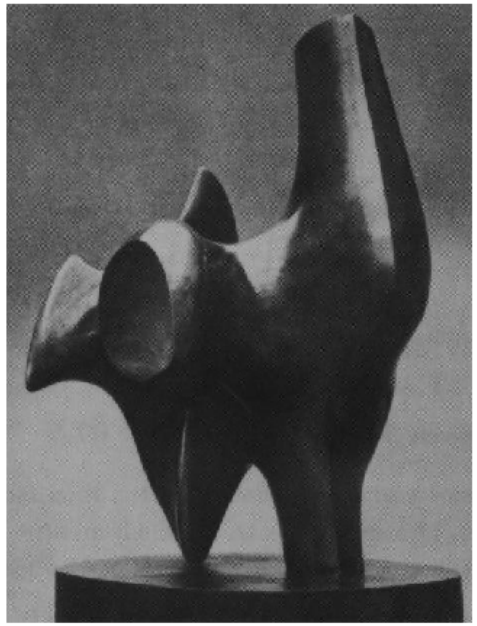
Last Lecture

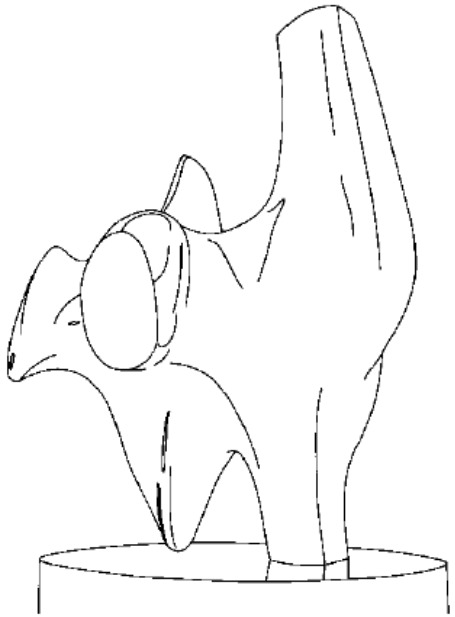
- Image as functions
- Filters in Spatial domain
- Filters in Frequency Domain
- Template Matching
- Image Pyramid

Today's Agenda

- Edge Detector
  - Gradient Detector (Roberts/Prewitt/Sobel)
  - Laplacian of Gaussian (Marr-Hildreth)
  - Gradient of Gaussian (Canny)
- Interest Point Detection
- Scale-invariant Region Selection
- Harris-Laplace Detector

Why Edges?





- Human are sensitive to edges!

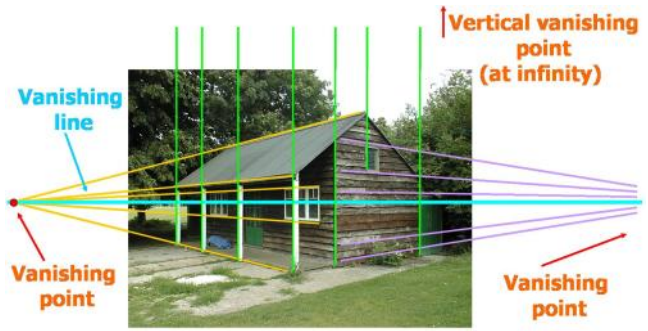
Edge Detection

- **Goal**: Identify sudden changes (***discontinuities***) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixel

Source: D. Lowe

Why do we care about edges?

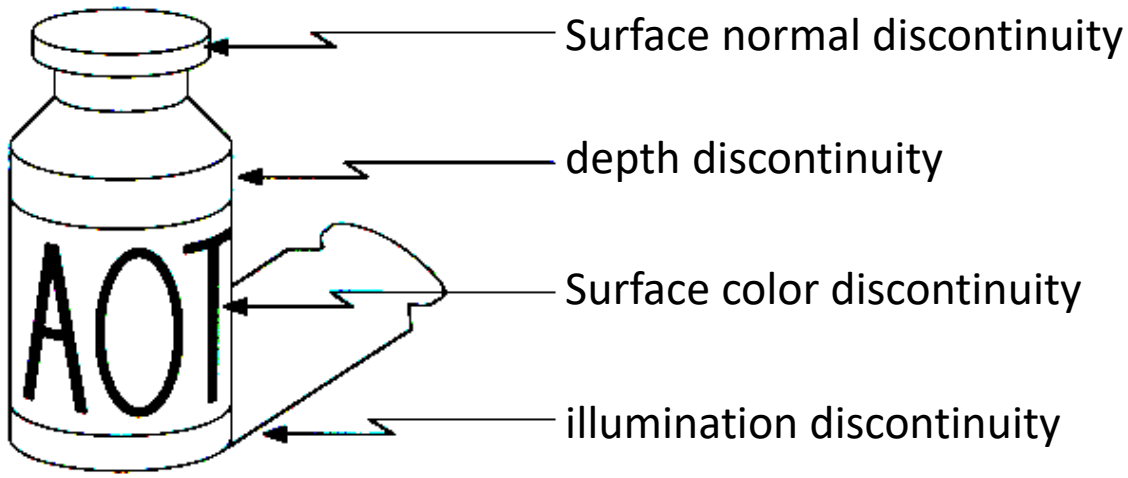
- Extract object information
- Recognize object
- Recover Geometry and viewpoint



Source: J. Hayes

Origins of edge

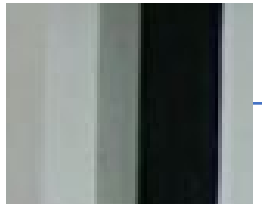
- Surface normal discontinuity
- Depth discontinuity
- Surface color discontinuity
- Illumination discontinuity



Closeup of edges

Depth discontinuity

Surface normal discontinuity



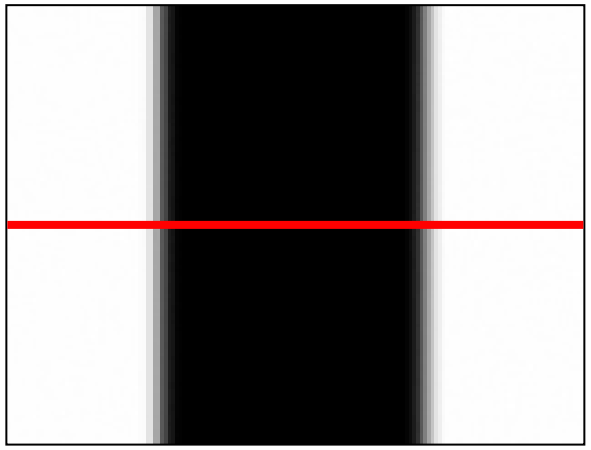
Surface color discontinuity

Source: D. Hoiem

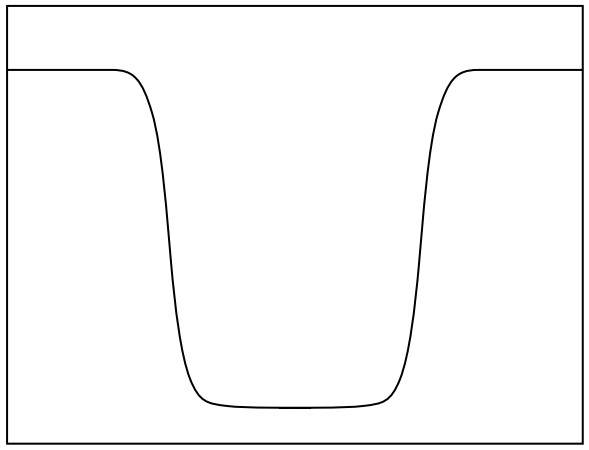
Characterizing edge

- An edge is a place of rapid change in the image intensity function

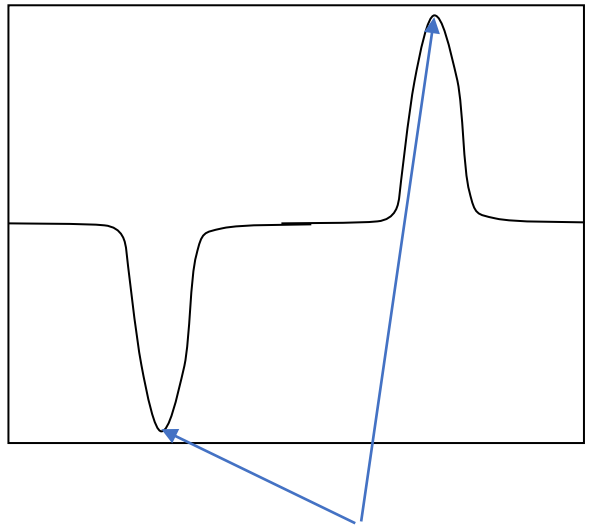
image



intensity function
(along horizontal scanline)



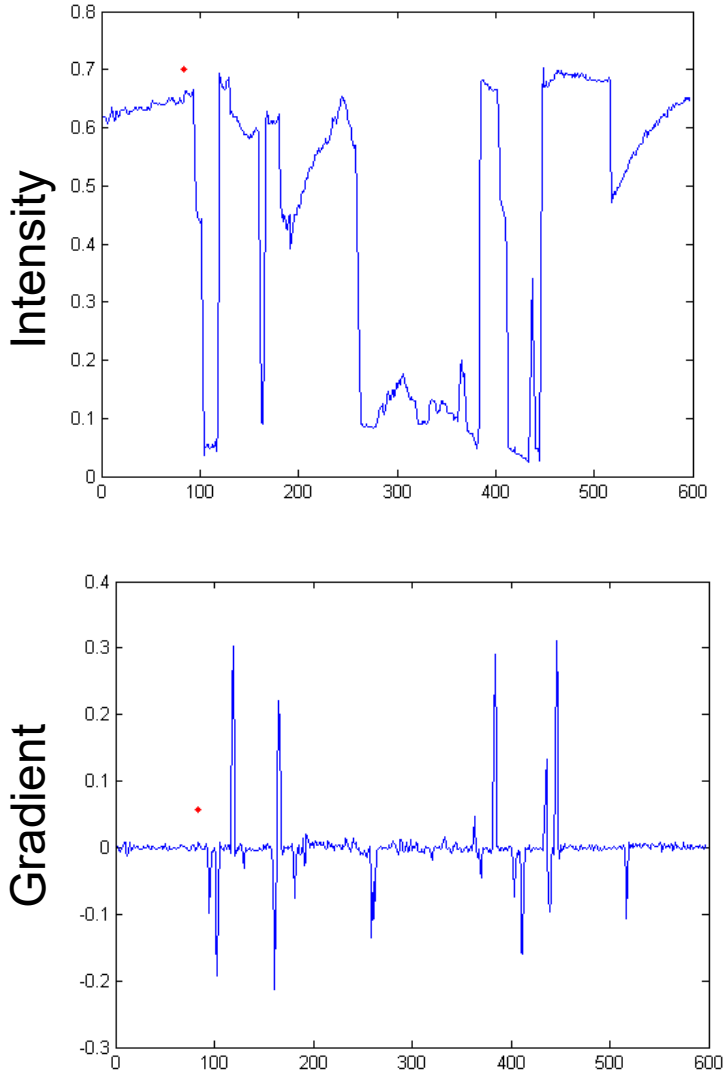
first derivative



Edges correspond to
extrema of derivative

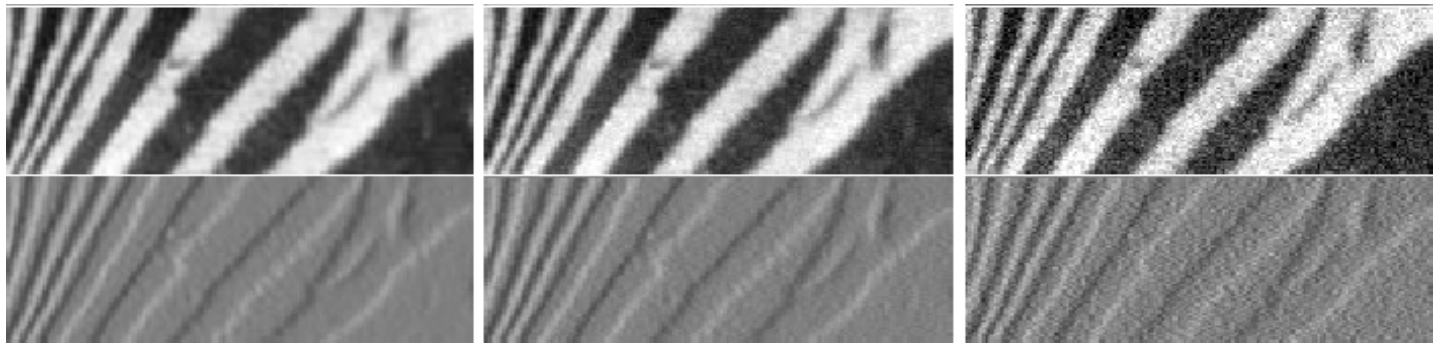
Credit: James Hays

Intensity profile



Source: D. Hoiem

Effect of noise



- Difference filters ***respond strongly*** to noise
  - Image noise results in pixels that look ***very different*** from their neighbors
  - Generally, the larger the noise, the stronger the response

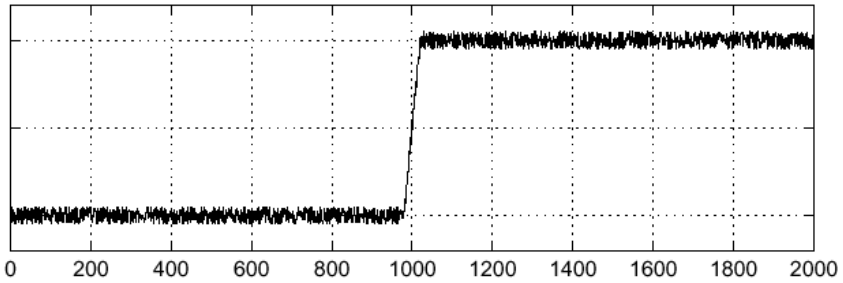
- What is to be done?
  - ***Smoothing*** the image should help, by forcing ***pixels*** different to their neighbors to look ***more like neighbors***

Source: D. Forsyth

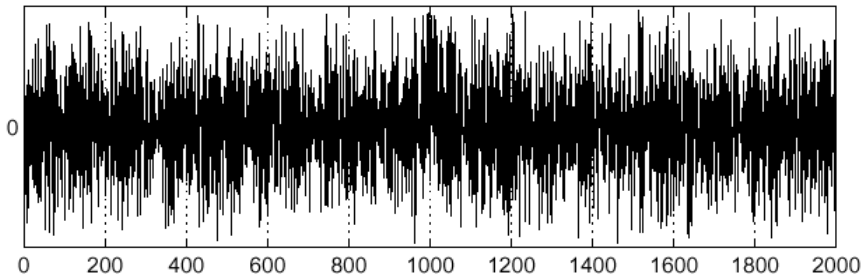
Effect of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$



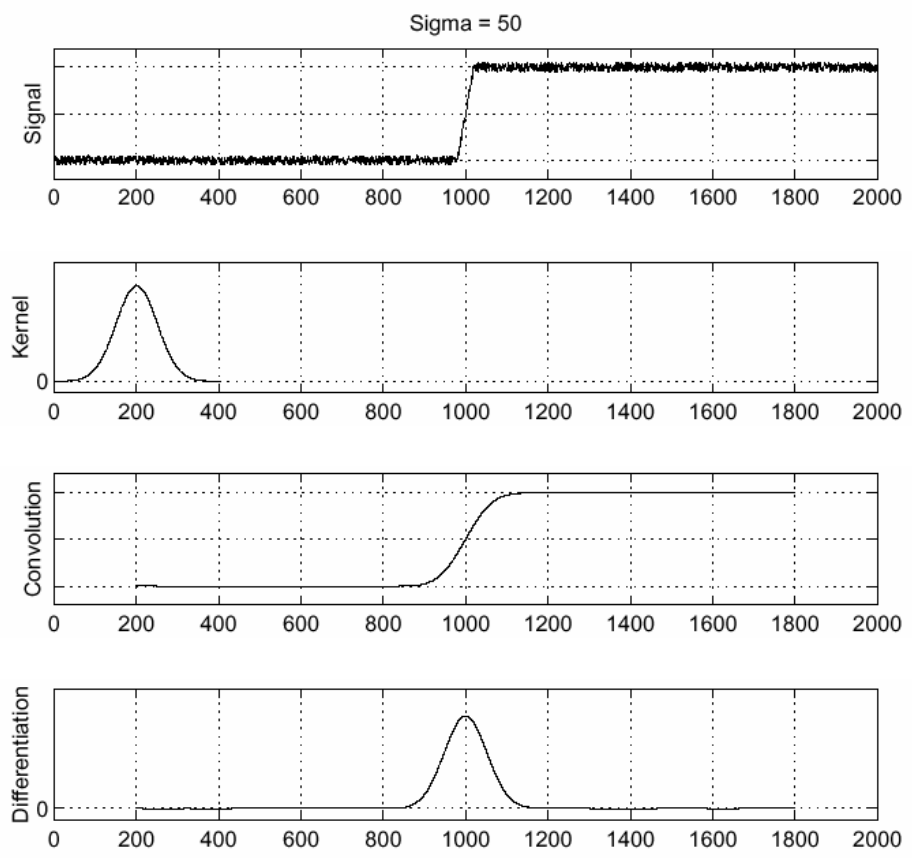
$\frac{d}{dx}f(x)$



Where is the edge?

Source: S. Seitz

Solution: Smoothing Filter



f

g

$f * g$

$\frac{d}{dx}(f * g)$

Edge

Gaussian

Gaussian Smoothed

Edge = maximum of derivative

- To find edges, look for *peaks* in $\frac{d}{dx}(f * g)$

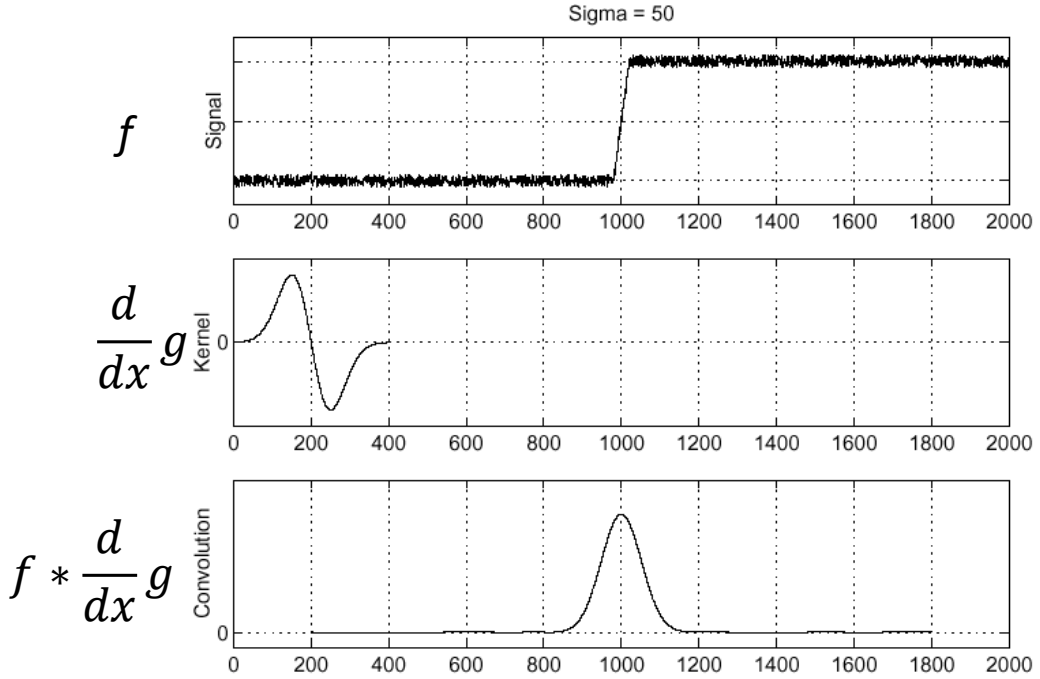
Source: S. Seitz

Derivative theorem of convolution

- This theorem gives up a very useful property

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

- This saves us one operation



Edge

Derivative of Gaussian

Edge = maximum of derivative

Source: S. Seitz

Common Edge Detectors

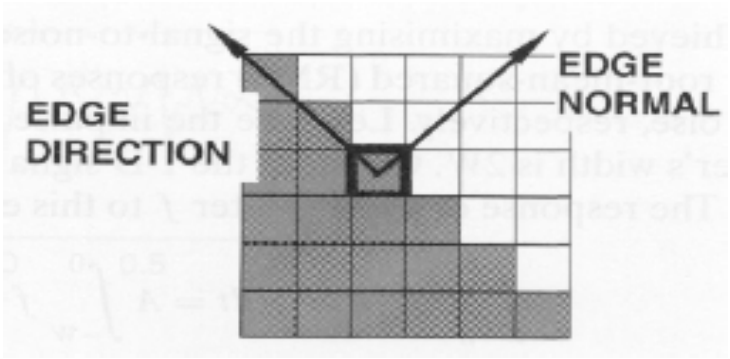
- Edge Detector
  - Gradient Detector
    - Roberts
    - Prewitt
    - Sobel
  - Gradient of Gaussian (Canny)
  - Laplacian of Gaussian (Marr-Hildreth)

Recall - Derivative in 2 Dimensions

- Given function $f(x, y)$
- Gradient Vector = $\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$
- Gradient Magnitude $|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$
- For Faster Computing, we approximate Magnitude by
  - $Mag(f(x, y) = |f_x| + |f_y|$
- Gradient Vector orientation $\theta(x, y) = \tan^{-1}(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}})$

Edge Detection by Gradient

- Properties of the gradient:
  - The ***magnitude*** of gradient provides information about the ***strength of the edge***
  - The ***direction*** of gradient is always ***perpendicular to the direction*** of the edge
- Main idea:
  - Compute derivatives in x and y directions
  - Find gradient magnitude
  - Threshold gradient magnitude



Slide Credit: S. Bedros

Image Derivative

- Derivative: Rate of change
  - Speed is the rate of change of a distance
  - Acceleration is the rate of change of speed

- $\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$

- Image is discrete function

  - $\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x) = f_x$

Discrete Derivative

- $\frac{df}{dx} = f(x) - f(x-1) = f'(x)$ backward difference
- $\frac{df}{dx} = f(x+1) - f(x) = f'(x)$ forward difference
- $\frac{df}{dx} = f(x-1) - f(x+1) = f'(x)$ centre difference

Example: Image derivative

- Suppose we have a 3x3 matrix of pixel
- We want to find the gradient magnitude at P5
- Backward Difference
  - $\frac{df}{dx} = P5 - P4, \quad \frac{df}{dy} = P5 - P2$
- Forward Difference
  - $\frac{df}{dx} = P6 - P5, \quad \frac{df}{dy} = P8 - P5$
- Centre Difference
  - $\frac{df}{dx} = P6 - P4, \quad \frac{df}{dy} = P8 - P2$

P1	P2	P3
P4	P5	P6
P7	P8	P9

Derivative Filter as Edge Detector

- Gradient operator
  - Robert's
  - Prewitt
  - Sobel

How it works?

- Compute derivatives in both x and y directions
- Find gradient magnitude
- Threshold gradient magnitude

Robert's Edge Detector

- The ***Robert's Edge Detector***

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

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

- This can be approximated by the following mask

+1	0
0	-1

Gx

0	+1
-1	0

Gy

The Gradient magnitude is given by:

$$|G| = \sqrt{{G_x}^2 + {G_y}^2}$$

Which is approximated by

$$|G| = |G_x| + |G_y|$$

Roberts Edge Detector

- A Simple approximation of the first image derivative
- Can locate the edge point but NOT the orientation of the edge.
- Work best with binary images.
- Sensitive to noise

Approximation of First Derivative

- Consider a 3x3 neighbour pixels at (i, j)

P1	P2	P3
P4	[i,j]	P6
P7	P8	P9

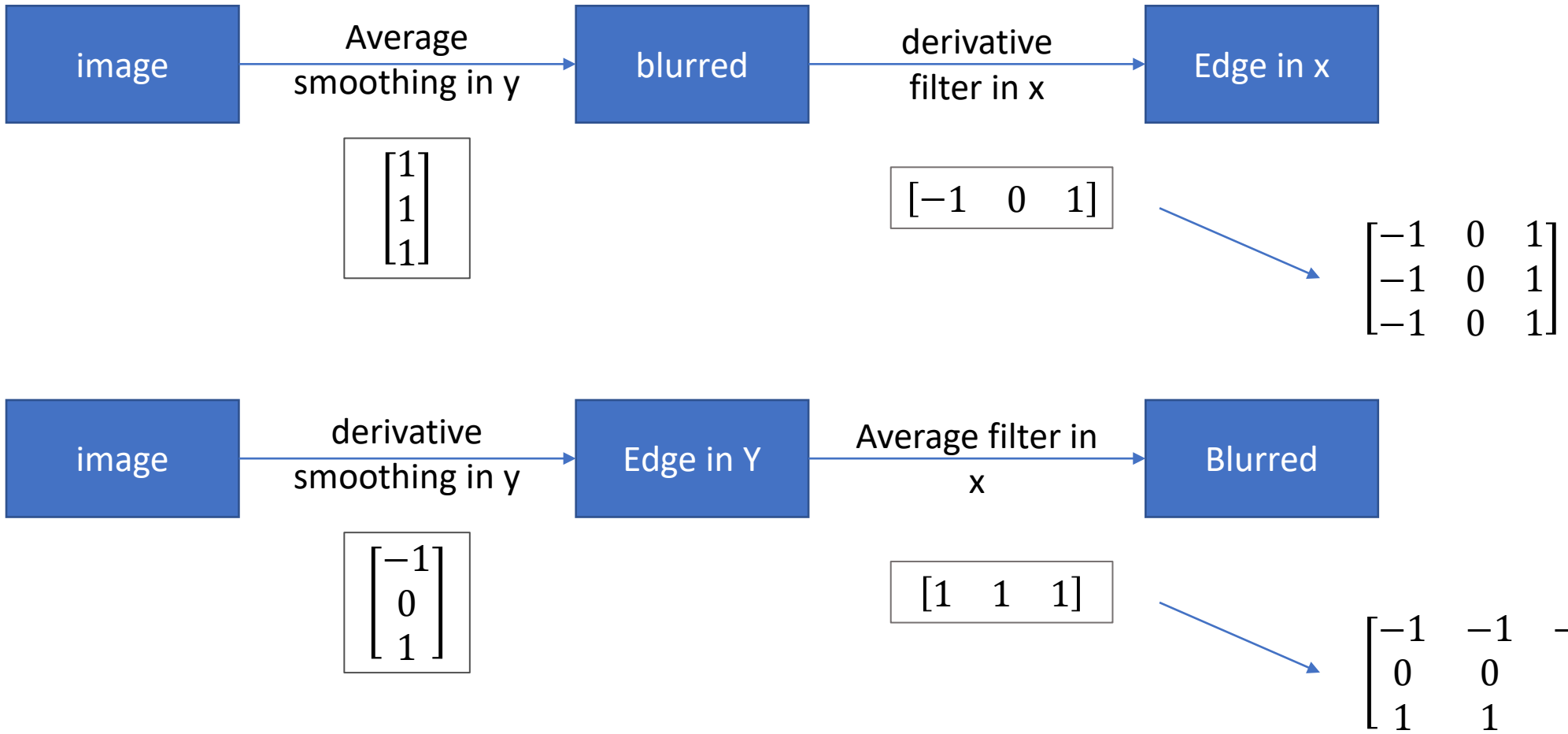
The partial derivative can be computed as follow

$$G_x = (P3 + CP6 + P9) - (P1 + CP4 + P7)$$
$$G_y = (P7 + CP8 + P9) - (P1 + CP2 + P3)$$

C is a constant

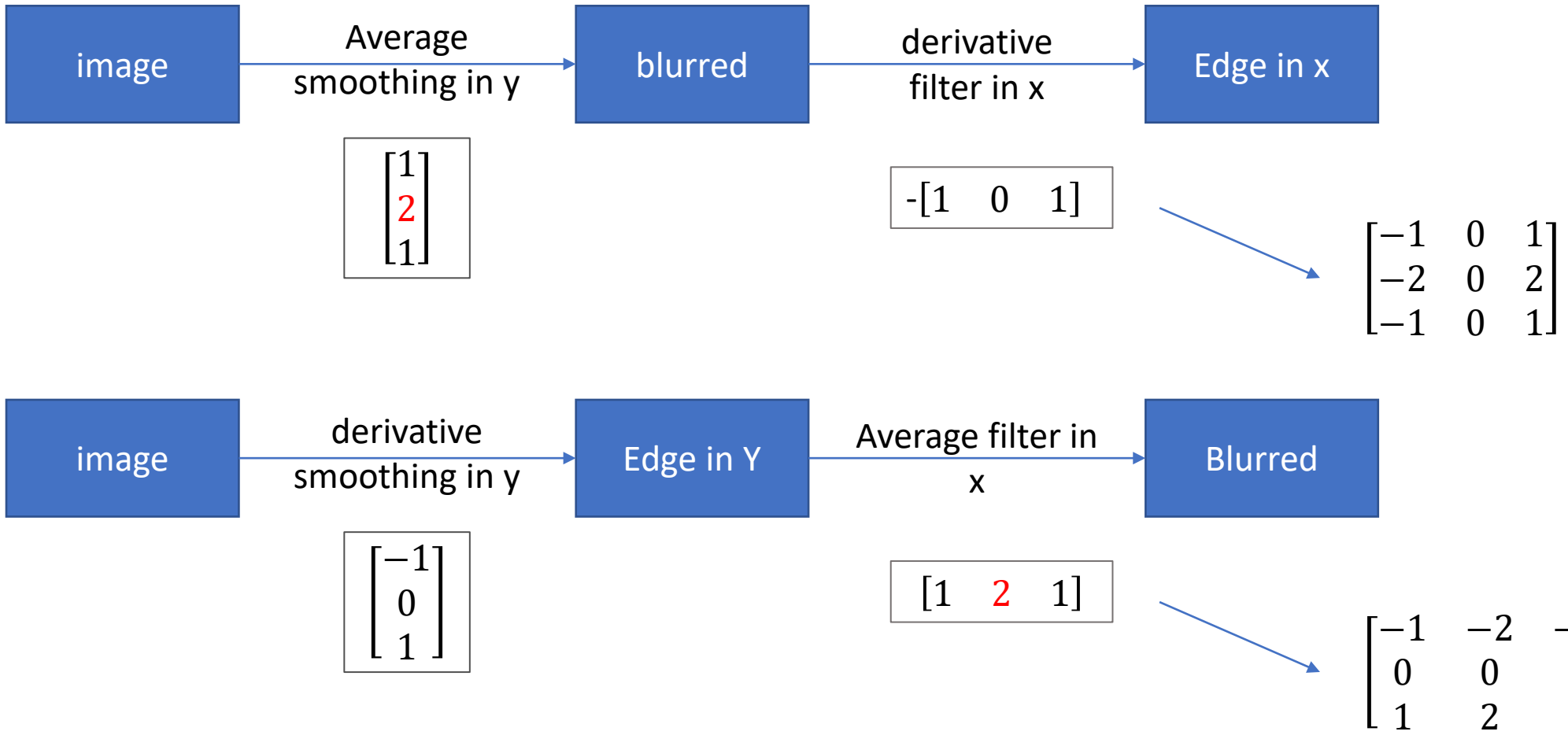
The constant C adjusts the weighting given to the pixel near the centre of the mask

Prewitt Edge Detector (When C=1)



Source: M. Shah

Sobel Edge Detector (when C=2)



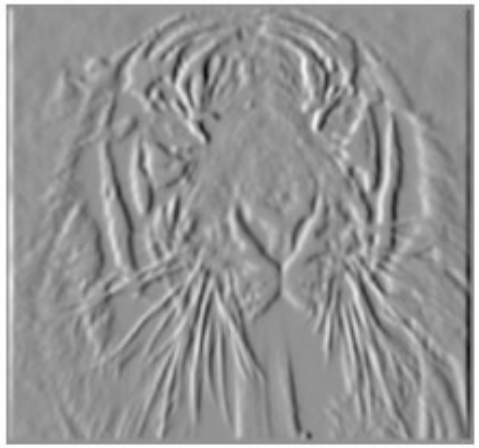
Source: M. Shah

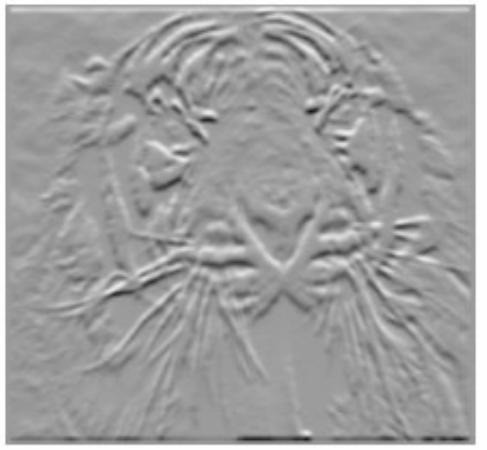
Example

Original Image

Gradient Magnitude

x-Direction





y-Direction

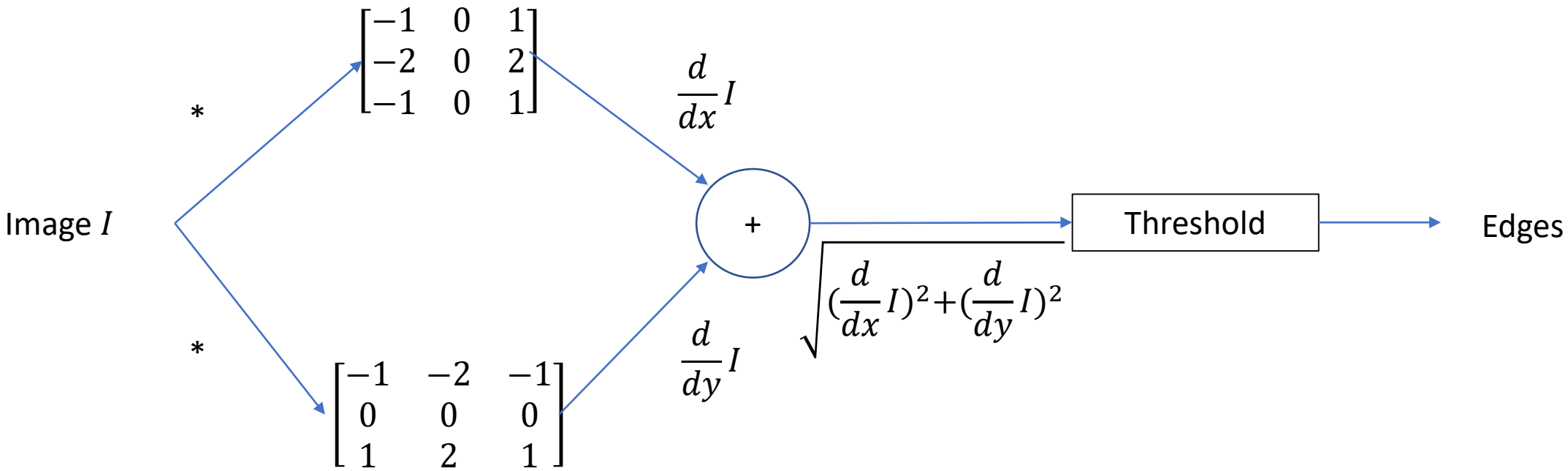
Image Credit: J.C. Niebles

Sobel and Prewitt Detectors

- Both detector approximates the gradient by a row and a column mask, which approximates the *first derivative* in both *horizontal* and *vertical* direction.

- The masks are then combined to form a single metric

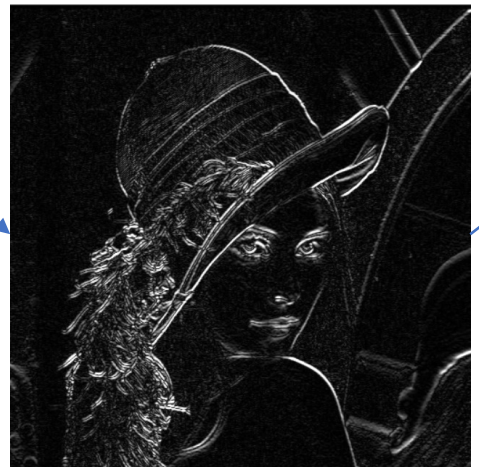
Sobel Edge Detector



Sobel Edge Detector

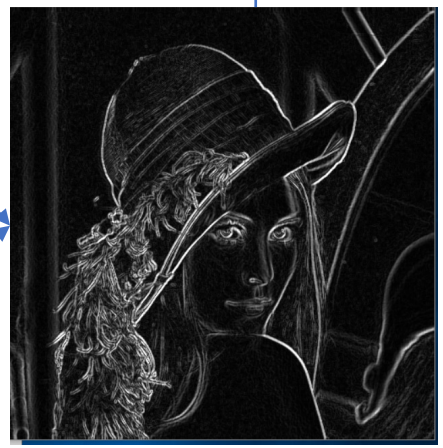
Image I

Derivative in X



Derivative in Y

Mag>threshold = 0



Sobel Edge Detector

Result of Prewitt and Sobel Detector

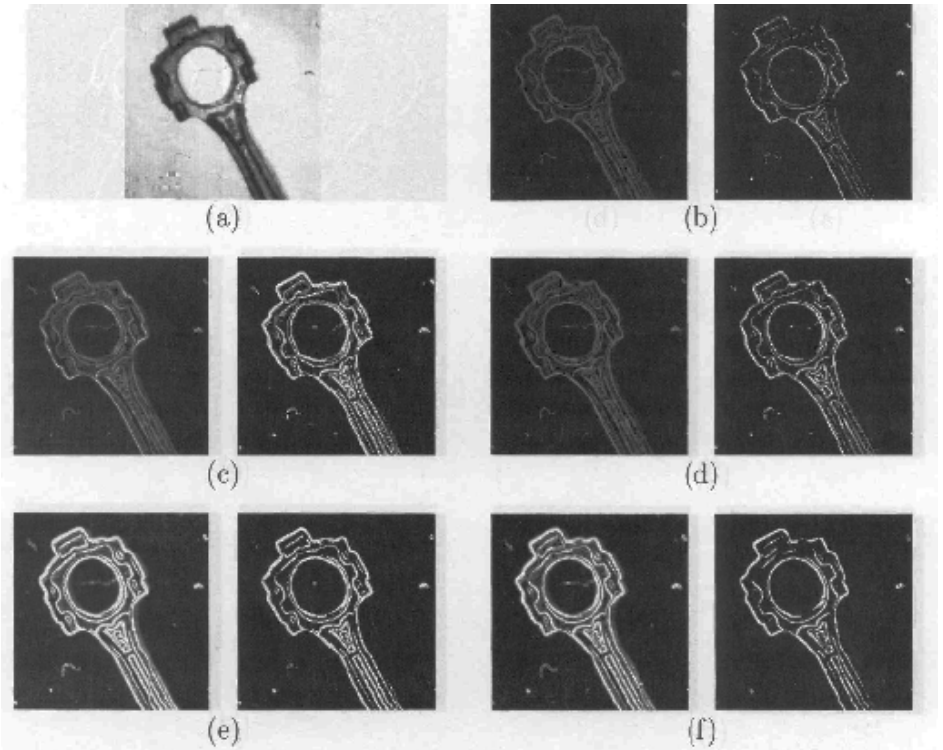


Figure 5.5: A comparison of various edge detectors without filtering. (a) Original image. (b) Simple gradient using 1×2 and 2×1 masks, $T = 64$. (c) Gradient using 2×2 masks, $T = 64$. (d) Roberts cross operator, $T = 64$. (e) Sobel operator, $T = 225$. (f) Prewitt operator, $T = 225$.

(without noise filtering)

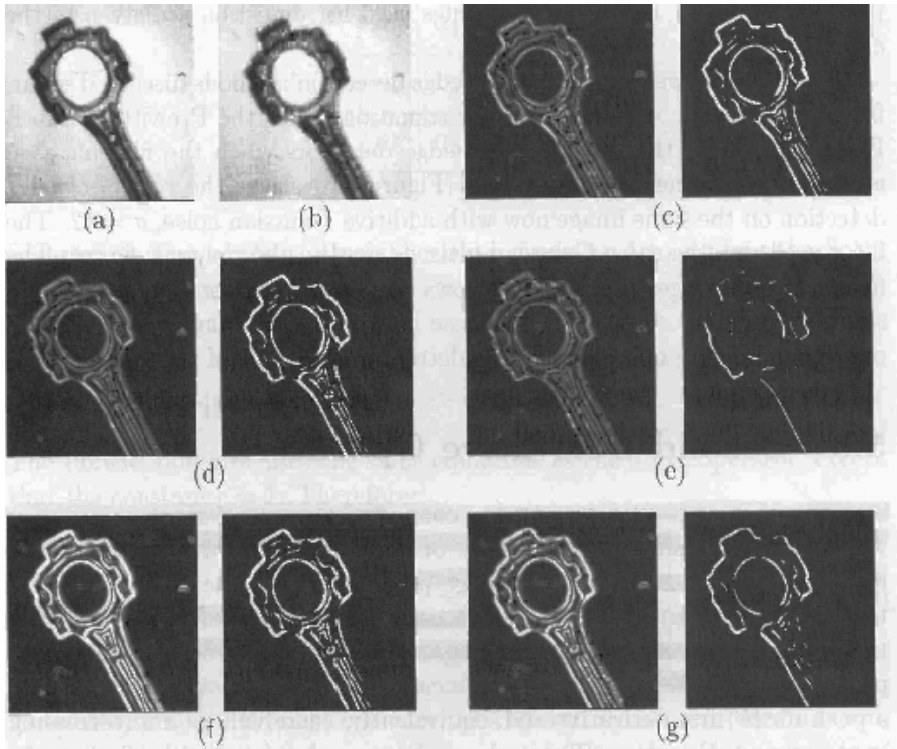


Figure 5.4: A comparison of various edge detectors. (a) Original image. (b) Filtered image. (c) Simple gradient using 1×2 and 2×1 masks, $T = 32$. (d) Gradient using 2×2 masks, $T = 64$. (e) Roberts cross operator, $T = 64$. (f) Sobel operator, $T = 225$. (g) Prewitt operator, $T = 225$.

(with noise filtering)

Pseudo Code For Prewitt / Sobel

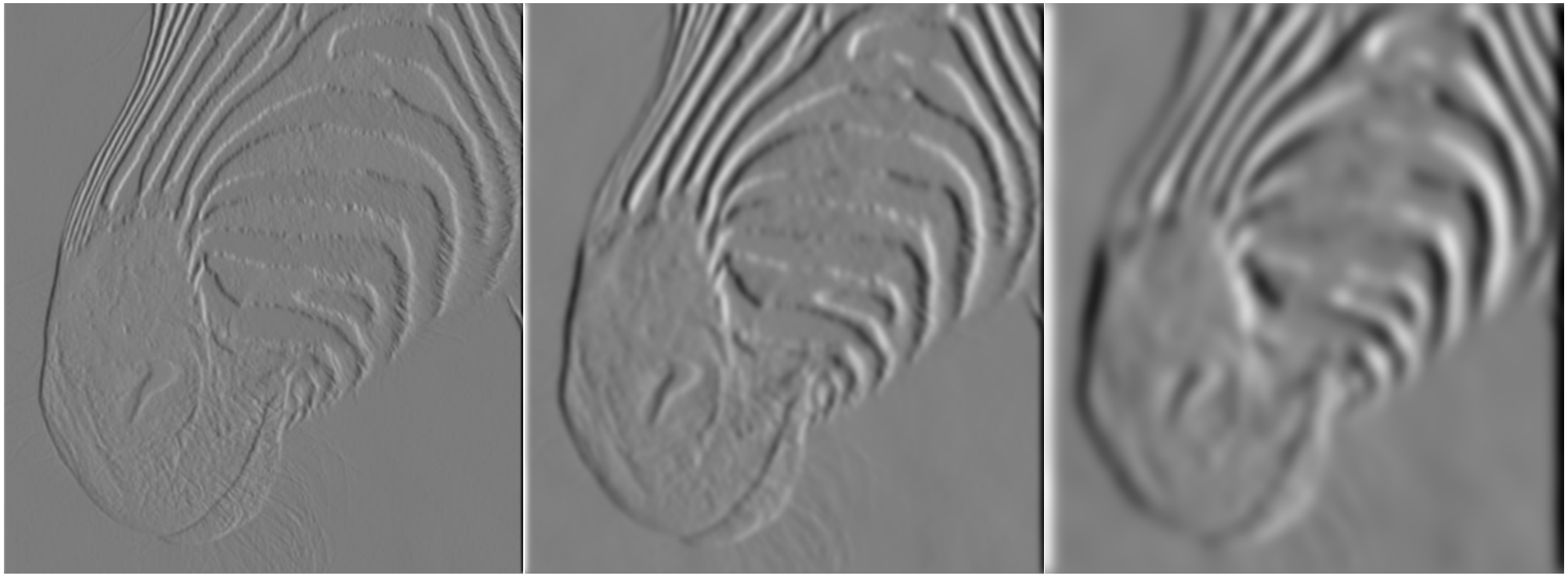
- 10 Load the image and convert to grey
- 20 Pad the image and convert to double
- 30 Define Mask Mx=[-1 0 1;-c 0 c;-1 0 1] C=1 Prewitt
  Mask My=[-1 -c -1;0 0 0; 1 c 1] C=2 Sobel

- 40 For every pixel, convolute with the Mask
  Gx=Mx*image; Gy=My*image
  Mag = sqrt(Gx^2+Gy^2)

50 Remove Padding

60 Convert the image back to uint8

Trade of between smoothing and localization



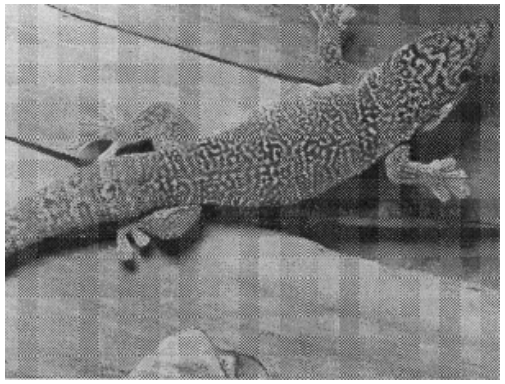
- Smoothed derivative removes noises, but blurs edge which affect the edge localization (i.e. the location of the “true” edge)

Source: D. Forsyth

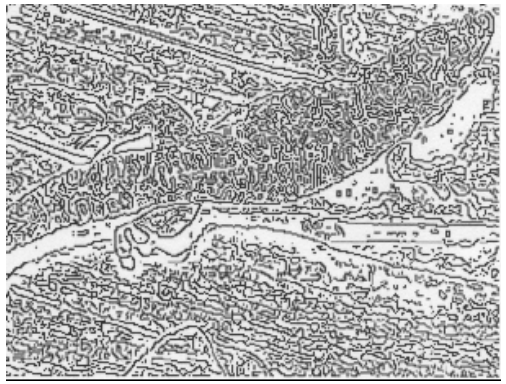
Noise Suppression vs Localization

- By using Gaussian filter with different σ, we can achieve different smoothing effect.

- Using larger the σ value reduces the noise but also affect the localization of the edge.



Smaller σ



Larger σ

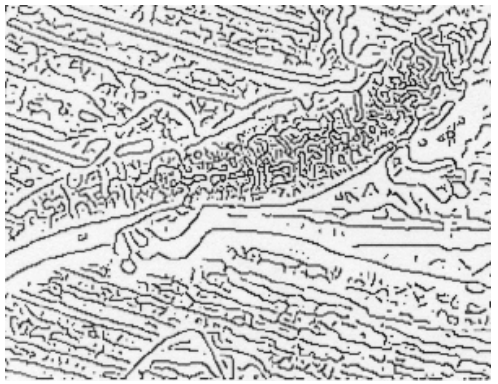


Photo Credit: S. Bedros

Comparing Edge Operators

Gradient: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

Roberts (2 x 2):

0	1
-1	0

1	0
0	-1

Sobel (3 x 3):

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

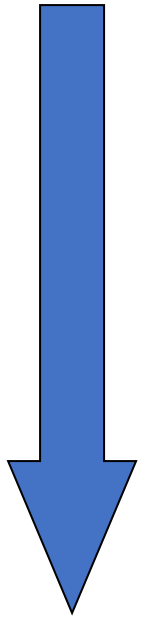
1	2	1
0	0	0
-1	-2	1

Sobel (5 x 5):

-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

Good Localization
Noise Sensitive
Poor Detection

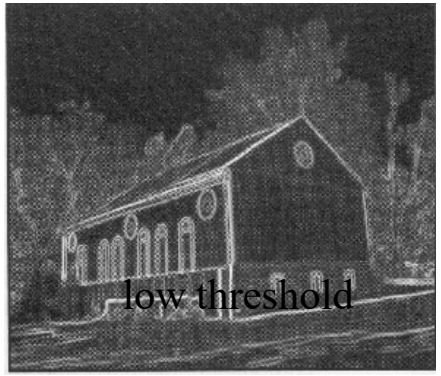


Poor Localization
Less Noise Sensitive
Good Detection

[Slide from Srinivasa Narasimhan]

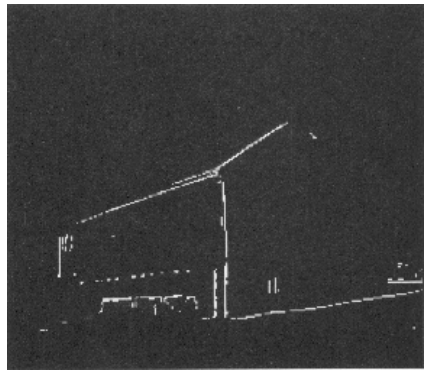
Effector of Threshold

Image



Gradient Mag

Low Threshold



High Threshold

Threshold value affects true edges and noise

Single Thresholding

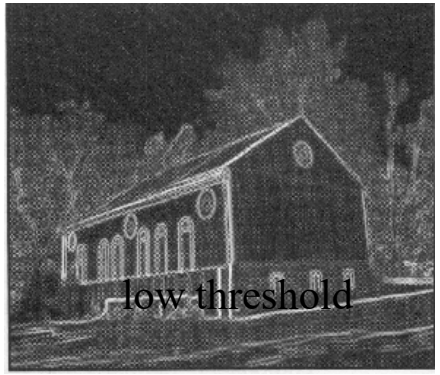
- Single Thresholding

$$E(x,y) = \begin{cases} 1 & \text{if } \|\nabla f(x,y)\| > T \text{ for some threshold } T \\ 0 & \text{otherwise} \end{cases}$$

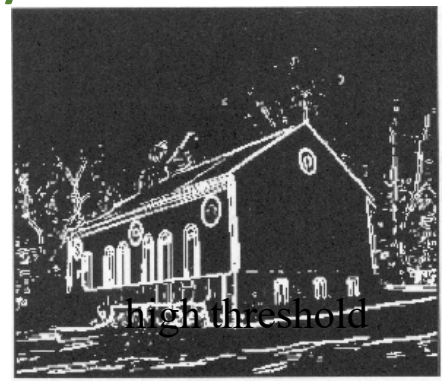
- Can only detect "Strong" edges.

- Does not guarantee "continuity"

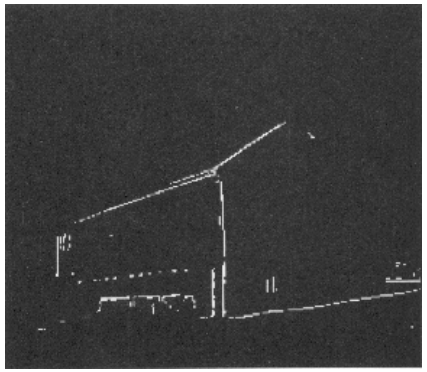
Original Image



Gradient Mag



Low Threshold



High Threshold

[Slide from S. Bedros

Hysteresis Thresholding

- Hysteresis thresholding applies TWO level of thresholds.
  - Low threshold t_l
  - High threshold t_h (usually, $t_h = 2t_l$)

$$\|\nabla f(x,y)\| \geq t_h \text{ This is a edge}$$
$$t_l \leq \|\nabla f(x,y)\| \leq t_h \text{ maybe an edge}$$
$$\|\nabla f(x,y)\| \leq t_l \text{ Not an edge}$$

When $t_l \leq \|\nabla f(x,y)\| \leq t_h$ is classified as an edge if the neighbouring pixel is a strong edge.

original image

edge magnitude

$T = \{5, 20\}$

$T = \{20, 40\}$

Higher Threshold gives less edges

Canny edge detector

- This is probably the most ***widely used edge detector*** in computer vision
  - Detection of edge with *low error rate*.
  - The edge point detected from the operator should accurately *localize* on the center of the edge.
  - A given edge in the image should only be marked once, and where possible, image noise should not create *false* edges.

J. Canny, ***A Computaional Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

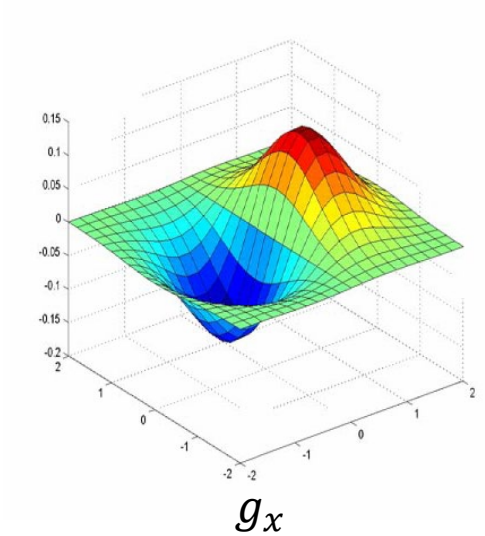
Steps of Canny Edge Detector

1. ***Smooth*** image with Gaussian filter
2. Compute ***derivative*** of filtered image
3. Find ***magnitude*** and ***orientation*** of gradient
4. Apply "***Non-maximum Suppression***"
5. Apply "***Hysteresis Threshold***"

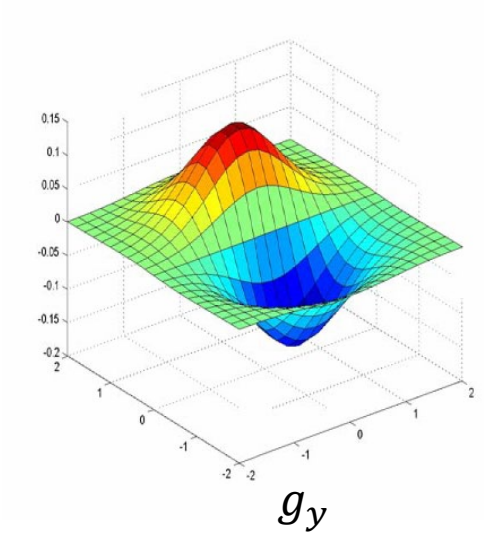
Canny Edge Detector First Two Steps

1. Smoothing
   - $S = I * g(x, y) = g(x, y) * I$

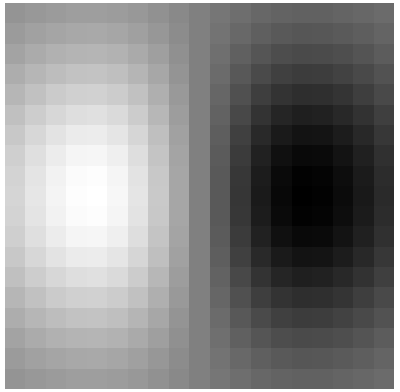
2. Derivative
   - $\nabla S = \nabla(g * I) = (\nabla g) * I$
   - $\nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix}$

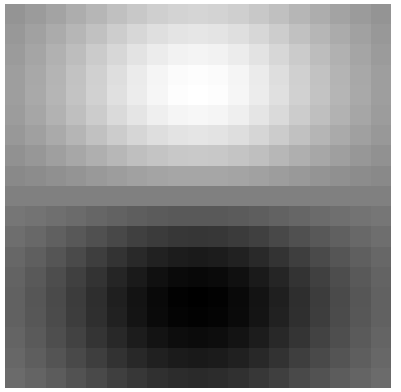


g_x



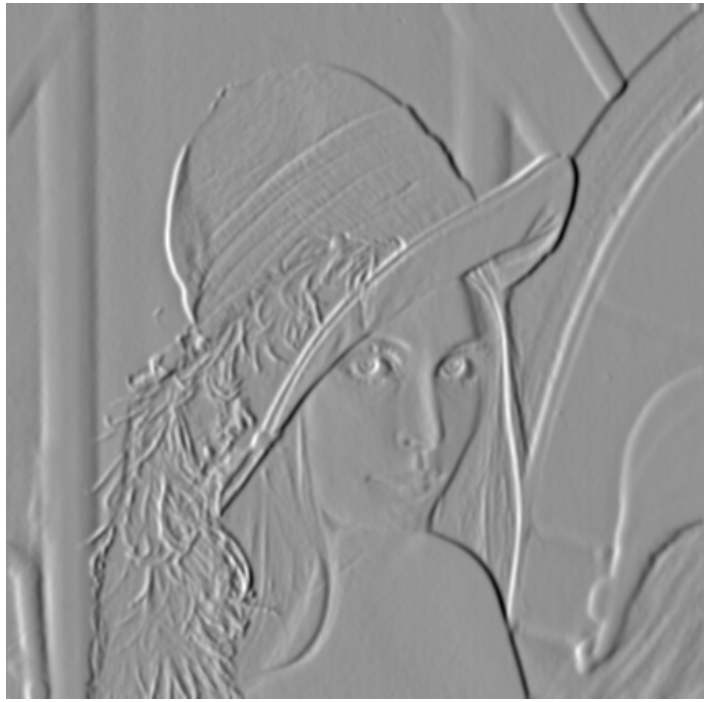
g_y





Derivative of Gaussian

Image



X-Derivative of Gaussian

Y-Derivative of Gaussian

Canny Edge Detector – Third Step

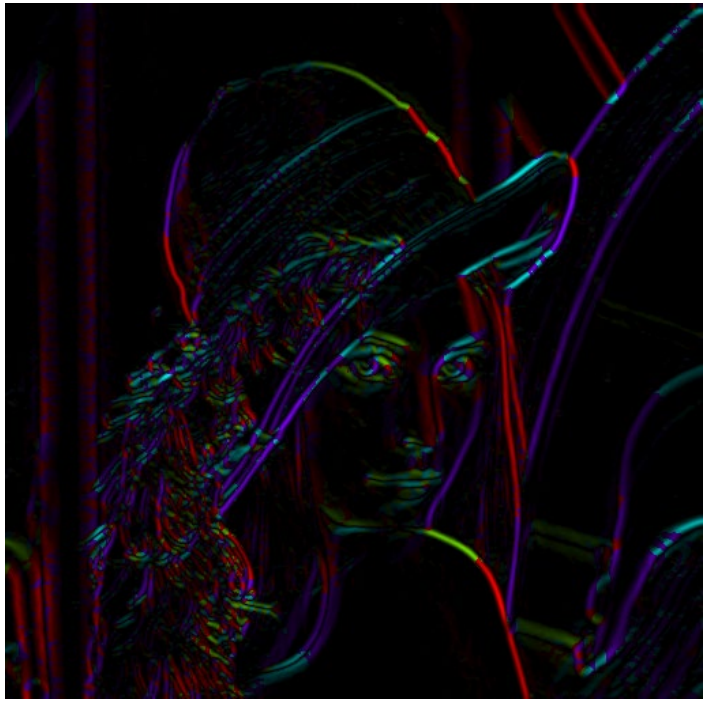
3. Gradient magnitude and gradient direction

Gradient vector (S_x, S_y)

Magnitude = $\sqrt{(S_x^2 + S_y^2)}$

Direction $\theta = \tan^{-1}\frac{S_y}{S_x}$

Gradient Magnitude



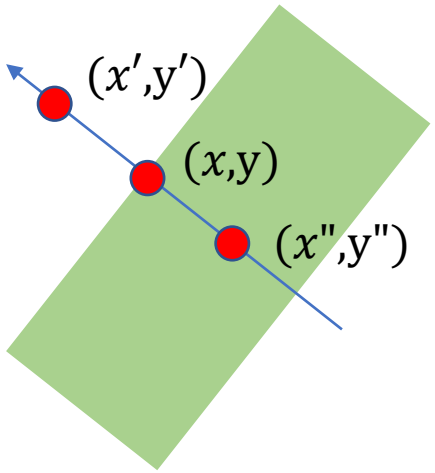
θ=atan(gy,gx)

Source: James Hays

Canny Edge Detector – Fourth Step

4. Non maximum suppression

- Suppress the pixels in $|\nabla G|$ which are **NOT local maximum**



$$M(x,y) = \begin{cases} |\nabla G|(x,y) & if\ |\nabla G|(x,y) > |\nabla G|(x',y')| \\ & \quad |\nabla G|(x,y) > G(x",y") \\ 0 & \text{Otherwise} \end{cases}$$

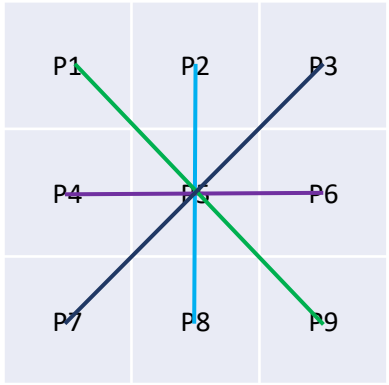
- In practice, quantize direction to horizontal and vertical and two diagonals

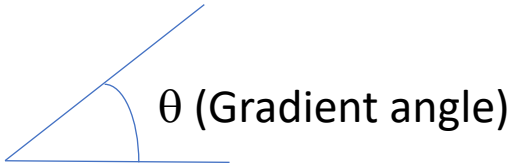
Quantization of Edge Directions

- In Practice, we quantize the gradient into 4 directions
- 1) Gradient angle = 0°

  P5 is max if (P5>P6 and P5>P4)
- 2) Gradient angle = 45°

  P5 is max if (P5>P3 and P5>P7)
- 3) Gradient angle = 90°
- P5 is max if (P5>P2 and P5>P8)
- 4) Gradient angle = 135°
- P5 is max if (P5>P1 and P5>P9)





Non-Maximum Suppression

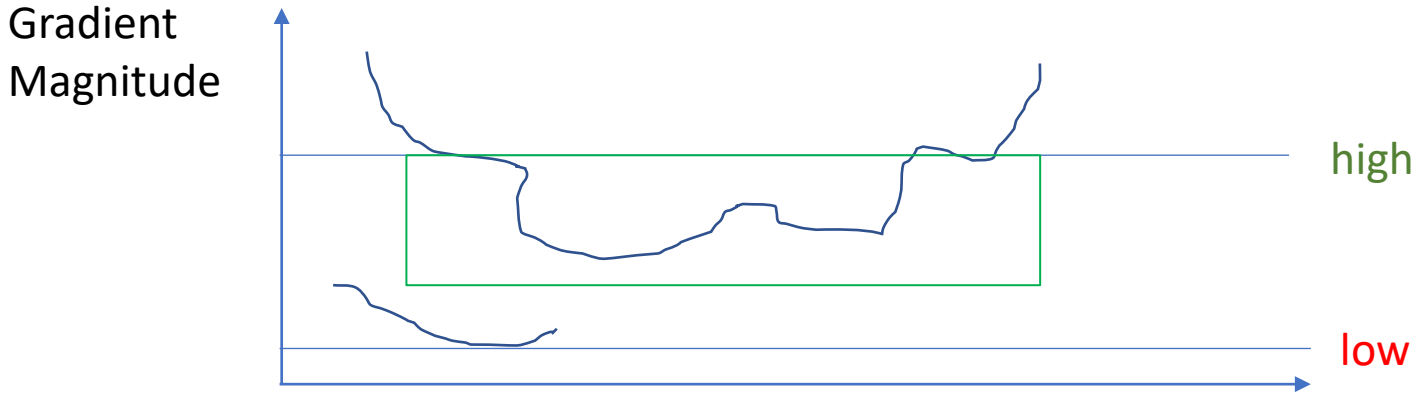
Before Non-Maximum Suppression

After Non-Maximum Suppression

Source: J. Hayes

Canny Edge Detector Hysteresis Thresholding

- If the gradient at a pixel is
- Above “High”, declare it as an ‘edge pixel’
- Below “Low”, declare it as an ‘non-edge pixel’
- Between “low” and “high”
  - Consider its neighbors iteratively then declare it an “edge pixel” if it is connected to an edge pixel directly or via pixel between “low” and “high”



Canny Edge Detector

After Non-Maximum Suppression

Canny Edges

Canny Edge Detector

Original Image

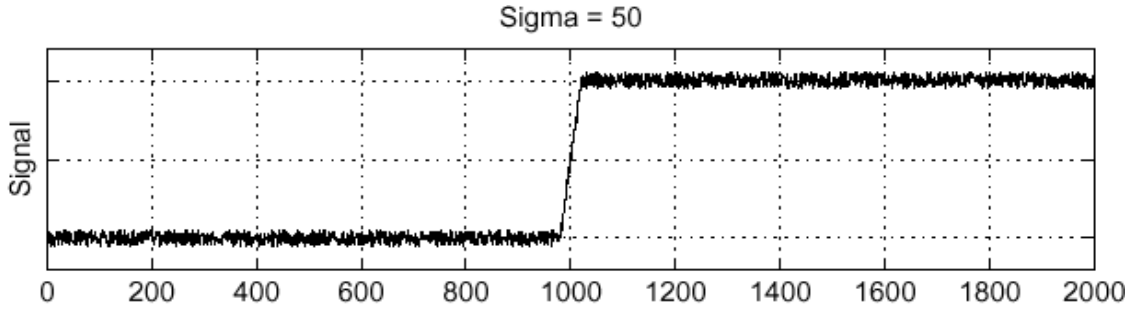
Canny Edges

Summary of Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin wide “ridges” down to single pixel width
4. Hysteresis thresholding
   - High and low thresholds
   - Use high threshold to start the edge curves and the low threshold to connect them

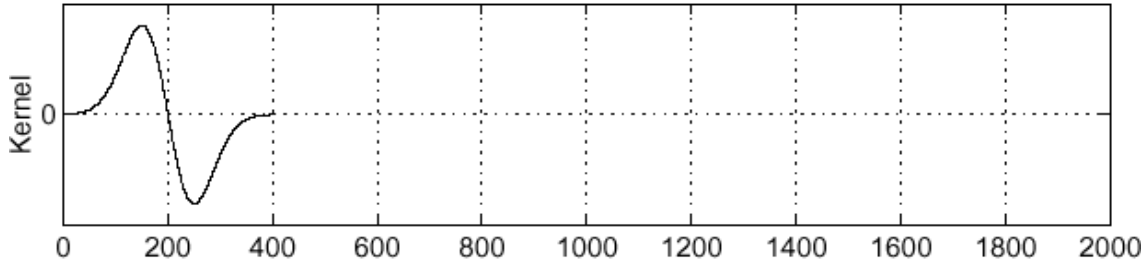
Recall Edge Detection using 1st Derivative

f



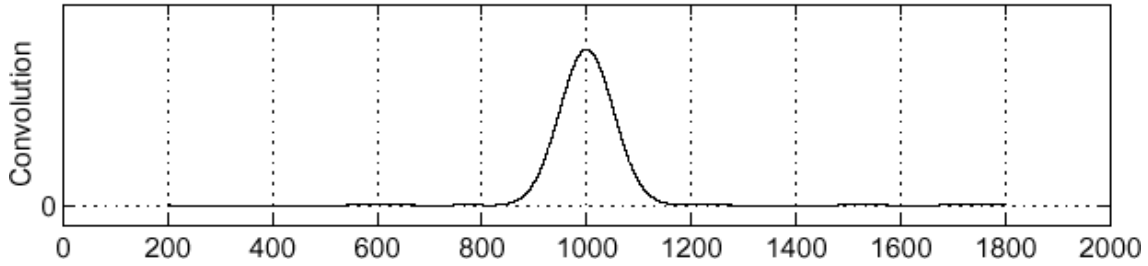
Edge

$\frac{d}{dx} g$



Derivative
of Gaussian

$f * \frac{d}{dx} g$

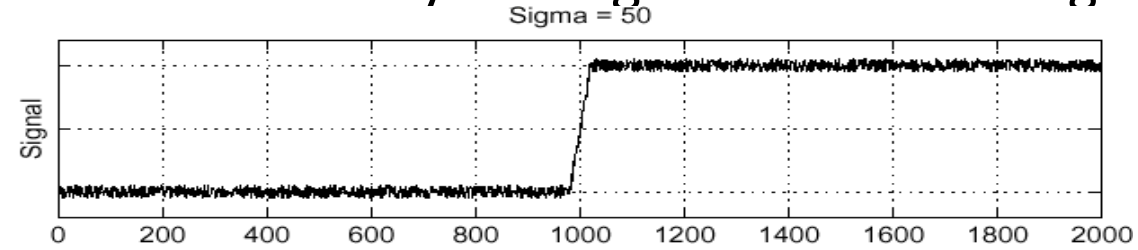


Edge = maximum
of derivative

Edge Detection Using the 2nd Derivative

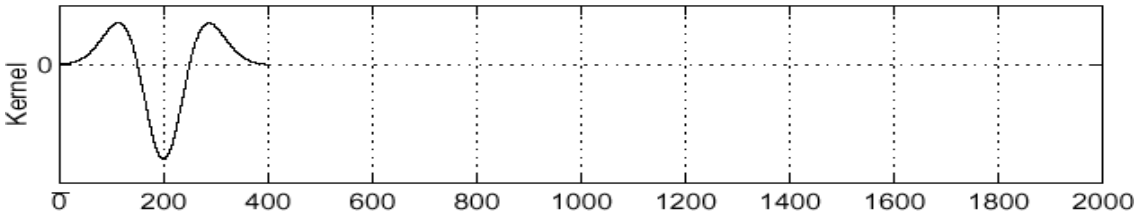
- Edge Points can be detected by finding the zero-crossing of the second derivative

f



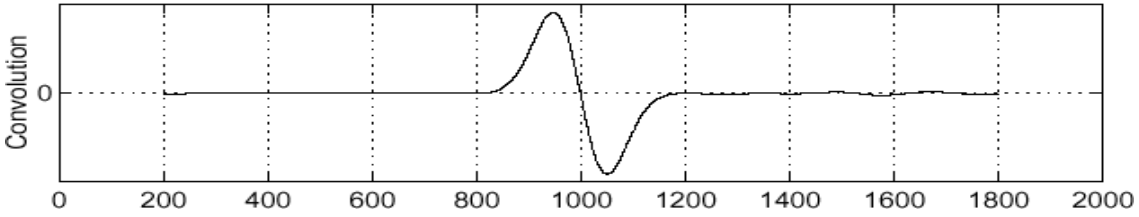
Edge

$\frac{d^2}{dx^2} g$



Second derivative of Gaussian (Laplacian)

$f * \frac{d^2}{dx^2} g$



Edge = zero crossing of second derivative

- Finding maxima/minima of gradient magnitude by locating zero-crossing

Edge Detection using 2nd Derivative

- Two Operators in 2D correspond to the second derivative:
  - Laplacian
  - Second Directional derivative

- The Laplacian is defined as

$$\nabla^2 = \nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Edge Detection using 2nd Derivative

- Applying to image
- $\nabla^2 f = \nabla \cdot \nabla f = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- This can be approximated by

$$\frac{\partial^2}{\partial x^2} = f(i, j+1) - 2(i, j) + f(i, j-1)$$

$$\frac{\partial^2}{\partial y^2} = f(i+1, j) - 2(i, j) + f(i-1, j)$$

- *Then*

$$\nabla^2 f = -4(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

Edge Detection using 2nd Derivative

- Example:

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

$$\nabla^2 f = -4z_5 + (z_2 + z_4 + z_6 + z_8)$$

  - The Laplacian can be implemented using the mask:

0	1	0
1	-4	1
0	1	0

- Example:

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

-	-	-	-	-	-
-	0	-5	-5	-5	-
-	-5	10	5	5	-
-	-5	10	0	0	-
-	0	-10	10	0	-
-	-	-	-	-	-

Slide from S. Bedros

Find Zero Crossings

- cases of zero-crossing
  - {+,-}, {-,+}, {+,0,-}, {-,0,+}

- Slope of zero-crossing {a,-b} is |a+b|

- To mark an edge and measure edge strength
  - compute slope of zero-crossing
  - apply a threshold to slope

Source: M. Shah

Marr-Hildreth Edge Detector

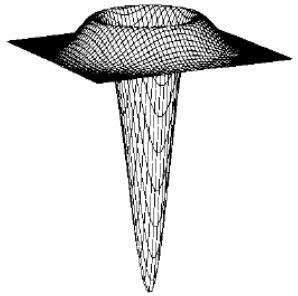
- Steps:

1. Marr Hildreth smooths the image by Gaussian filter
2. Apply Laplacian to S
3. Find Zero Crossings
4. Apply threshold to slope and mark edges.

$$\underbrace{\tilde{S}}_{Smoothed\ Image} = \underbrace{\tilde{g}}_{Gaussian\ Filter} * \underbrace{\tilde{I}}_{Image}$$

$$g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\Delta^2 S = \overbrace{\frac{\partial^2}{\partial x^2}S}^{Second\ order\ Derivative\ in\ x} + \overbrace{\frac{\partial^2}{\partial y^2}S}^{Second\ order\ Derivative\ in\ y}$$



Marr-Hildreth Edge Detector (LoG)

- Deriving the Laplacian of Gaussian (LoG)

$$\Delta^2 S = \overbrace{\frac{\partial^2}{\partial x^2} S}^{\substack{Second\ order \\ Derivative\ in\ x}} + \overbrace{\frac{\partial^2}{\partial y^2} S}^{\substack{Second\ order \\ Derivative\ in\ y}}$$

$$\Delta^2 S = \Delta^2 (g * I) = \left(\Delta^2 g\right) * I$$

$$\Delta^2 g = -\frac{1}{\sqrt{2\pi\sigma^3}} \left(2 - \frac{x^2+y^2}{\sigma^2}\right) e^{\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian Mask

$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-x^2}{2\sigma^2}}$$

x	-3	-2	-1	0	1	2	3
g(x)	0.011	0.135	0.607	1.000	0.607	0.135	0.011

2-D Gaussian Mask

$$g_x = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \left(-\frac{2x}{2\sigma^2}\right)$$

0	0	0	1	2	2	3	2	2	1	0	0	0
0	0	2	4	7	10	11	10	7	4	2	0	0
0	2	5	11	21	30	35	30	21	11	5	2	0
1	4	11	27	50	73	83	73	50	27	11	4	1
2	7	21	50	94	136	155	136	94	50	21	7	2
2	10	30	73	136	199	225	199	136	73	30	10	2
3	11	35	83	155	225	255	225	155	83	35	11	3
2	10	30	73	136	199	225	199	136	73	30	10	2
2	7	21	50	94	136	155	136	94	50	21	7	2
1	4	11	27	50	73	83	73	50	27	11	4	1
0	2	5	11	21	30	35	30	21	11	5	2	0
0	0	2	4	7	10	11	10	7	4	2	0	0
0	0	0	1	2	2	3	2	2	1	0	0	0

13x13 Gaussian Mask

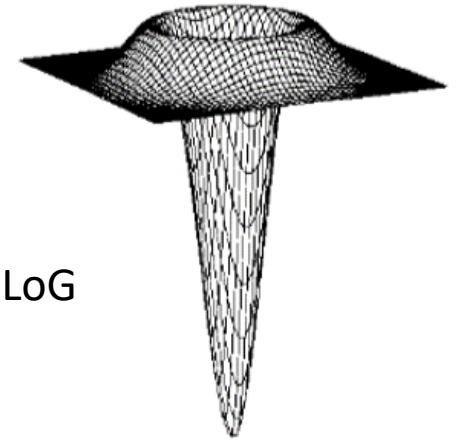
$\sigma = 2$

LoG Filter

$$\Delta^2 g = -\frac{1}{\sqrt{2\pi\sigma^3}}\left(2 - \frac{x^2+y^2}{\sigma^2}\right)e^{\frac{x^2+y^2}{2\sigma^2}}$$

0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.031	0.108	-0.242	-0.7979	-0.242	0.108	0.031
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008

LoG



Source: M. Shah

Separability of Gaussian

- Two-dimensional Gaussian can be separated into 2 one-dimensional Gaussian

$$h(x,y) = I(x,y) * g(x,y)$$

- Can be represented as two Gaussian

$$h(x,y) = \left(I(x,y) * g_1(x)\right) * g_2(y)$$

$$g(x) = e^{-(\frac{x^2}{2\sigma^2})} \qquad g(y) = e^{-(\frac{y^2}{2\sigma^2})}$$

$$g_1(x) = [.011 \quad .13 \quad .6 \quad 1 \quad .6 \quad .13 \quad .011] \qquad g_2(x) = \begin{bmatrix} .011 \\ .13 \\ .6 \\ 1 \\ .6 \\ .13 \\ .011 \end{bmatrix}$$

Separability of LoG

- $\Delta^2 S = \Delta^2 (g * I) = \left(\Delta^2 g\right) * I$

  Requires n^2 multiplications

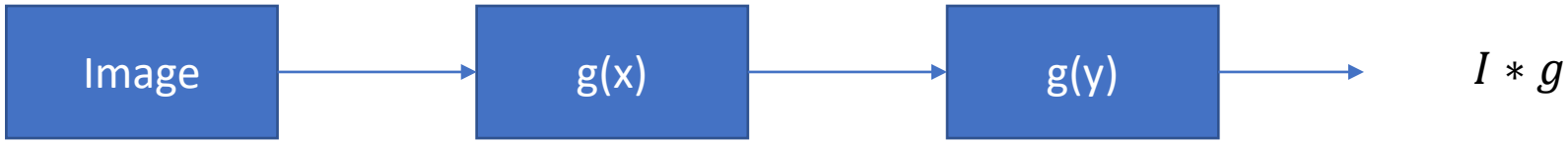
- $\Delta^2 S = \Delta^2 (I * g_{xx}(x)\ ) * g(y) + \left(I * g_{yy}(y)\right) * g(x)$

  Requires $4n$ multiplications

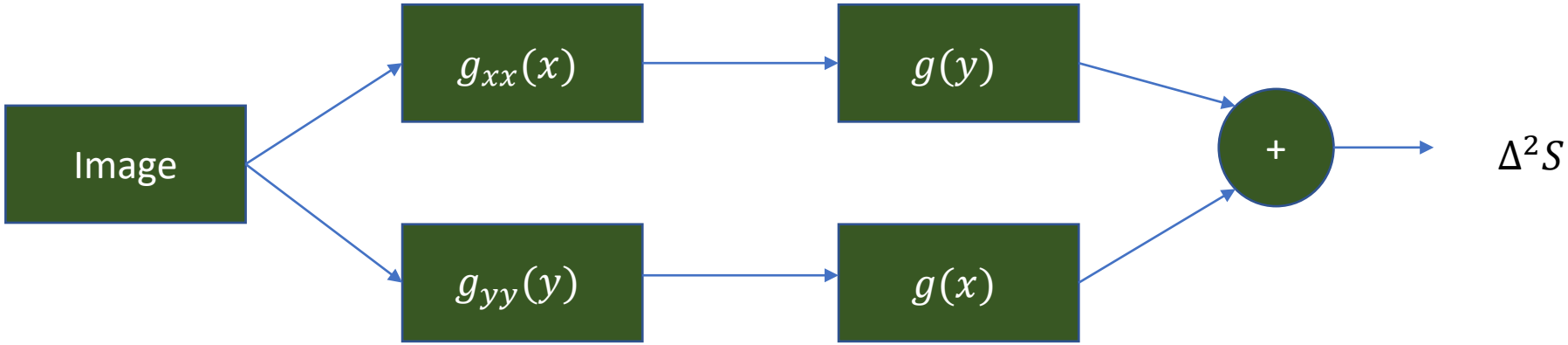
Source: M. Shah

Separability

- Gaussian Filtering



- LoG Filtering



Source: M. Shah

Example

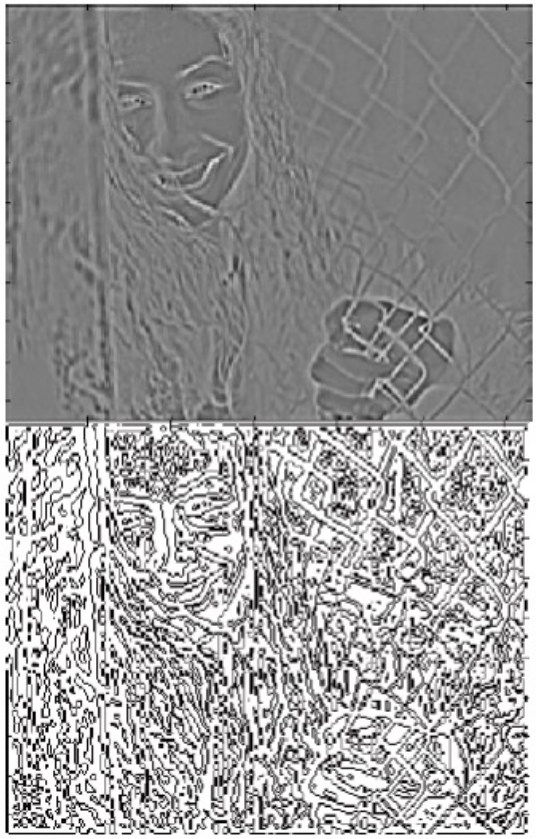
Image I

$I * (\Delta^2 g)$

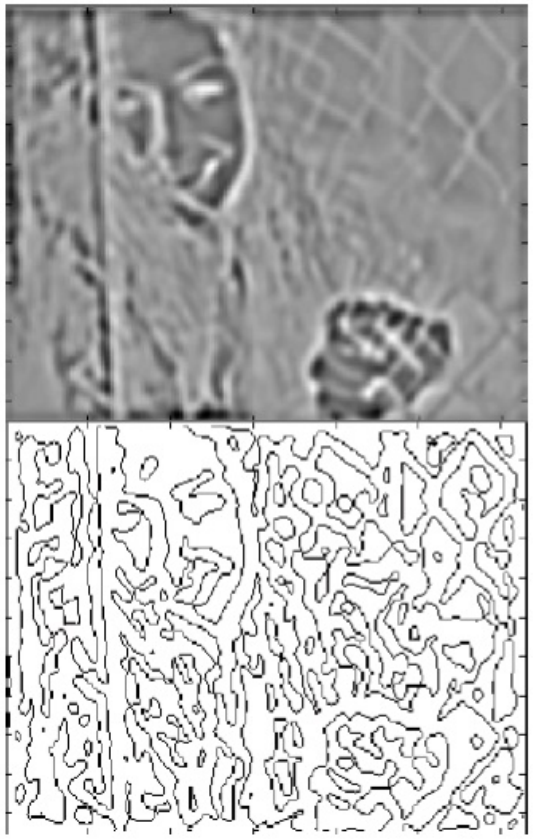
Zero Crossing
$I * (\Delta^2 g)$

Source: M. Shah

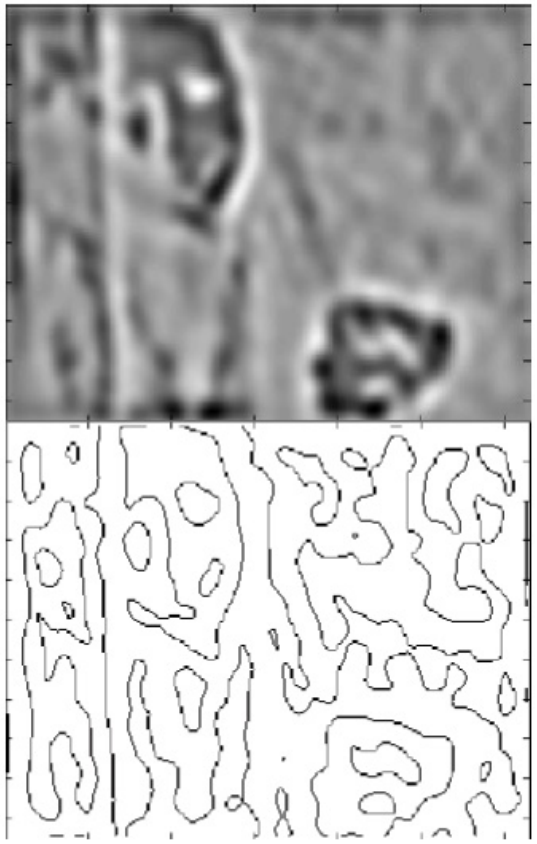
Effect with different sigma values



$\sigma = 1$



$\sigma = 3$



$\sigma = 6$

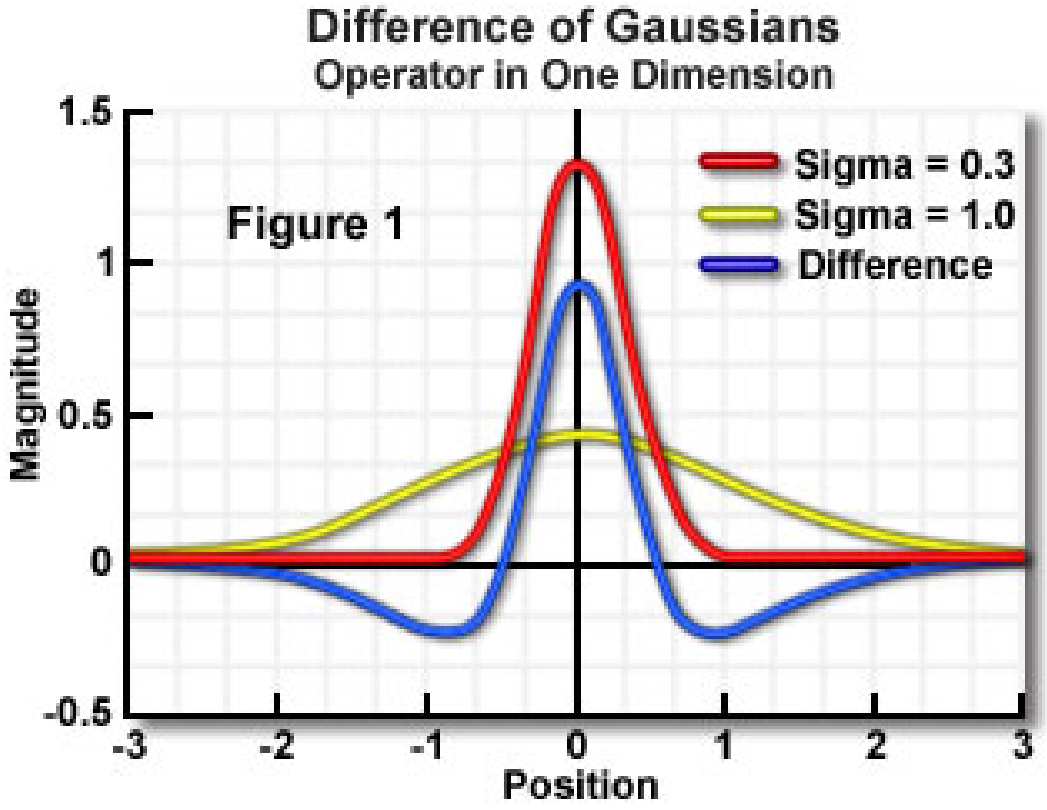
Source: R. Dockter

Summary of LoG Steps

1. Apply Log to the Image
   - Use 2D filters $\Delta^2 g(x, y)$
   - Use 4 x 1D filters $g(x), g_{xx}(x), g(y), g_{yy(y)}$
2. Find zero-crossing from each row and column
3. Find slope of zero-crossings
4. Apply threshold to slope and mark edges

Representing LoG as Difference of two Gaussians

- LoG can be represented as difference of two Gaussian functions with difference sigma values.



Difference of Gaussians (DoG)

$\sigma = 1$

$\sigma = 1.6$

DoG

LoG

Ratio (σ_1/σ_2) for best approximation is 1.6 or $\sqrt{2}$

Summary of Edge Detector

- Finite difference filters
  - respond strongly to noise
- Gradient of Gaussian filter- Canny Edge Detector
  - Detect edges with low error rate
  - Good localization
  - Reduce false detection.
- Laplacian of Gaussian (LOG) filter
  - apply Gaussian function for smoothing image and it perform better than finite difference filters.

Today's Agenda

- Edge Detection
  - Gradient Detector
  - Gradient of Gaussian (Canny)
  - Laplacian of Gaussian (Marr-Hildreth)
- Interest Point Detection
- Scale-invariant region selection
- SIFT Descriptor

Interest Point Detection

- What is ***Interest Point Detection*** in an Image?

- Goal: Find ***Same features*** between ***multiple images*** taken from different position or time

Motivation for Interest Points Detection

- Image Matching

Photo Source https://www.planetware.com

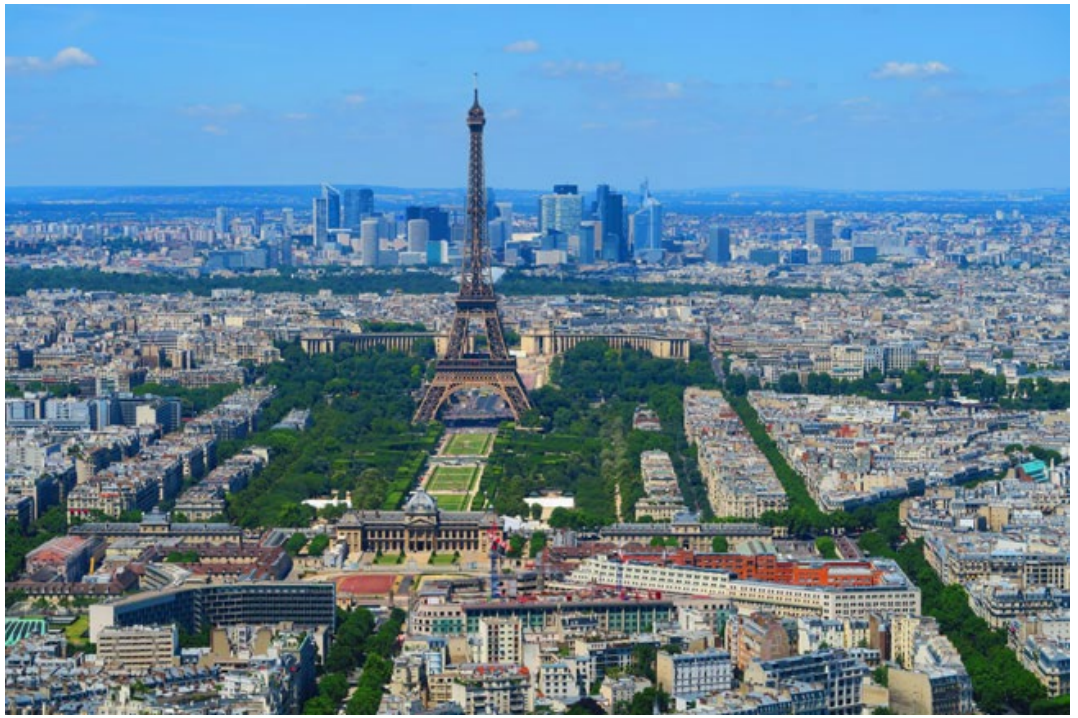


Photo Source https://xdaysiny.com

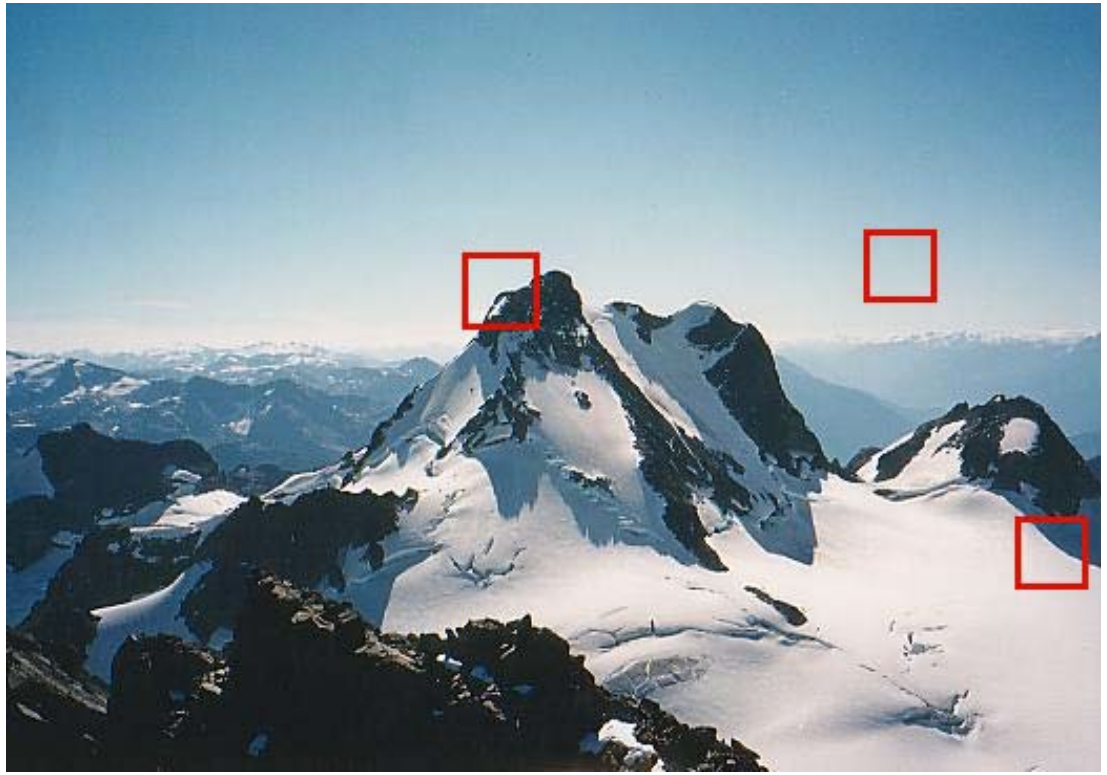
Motivation for Interest Points Detection

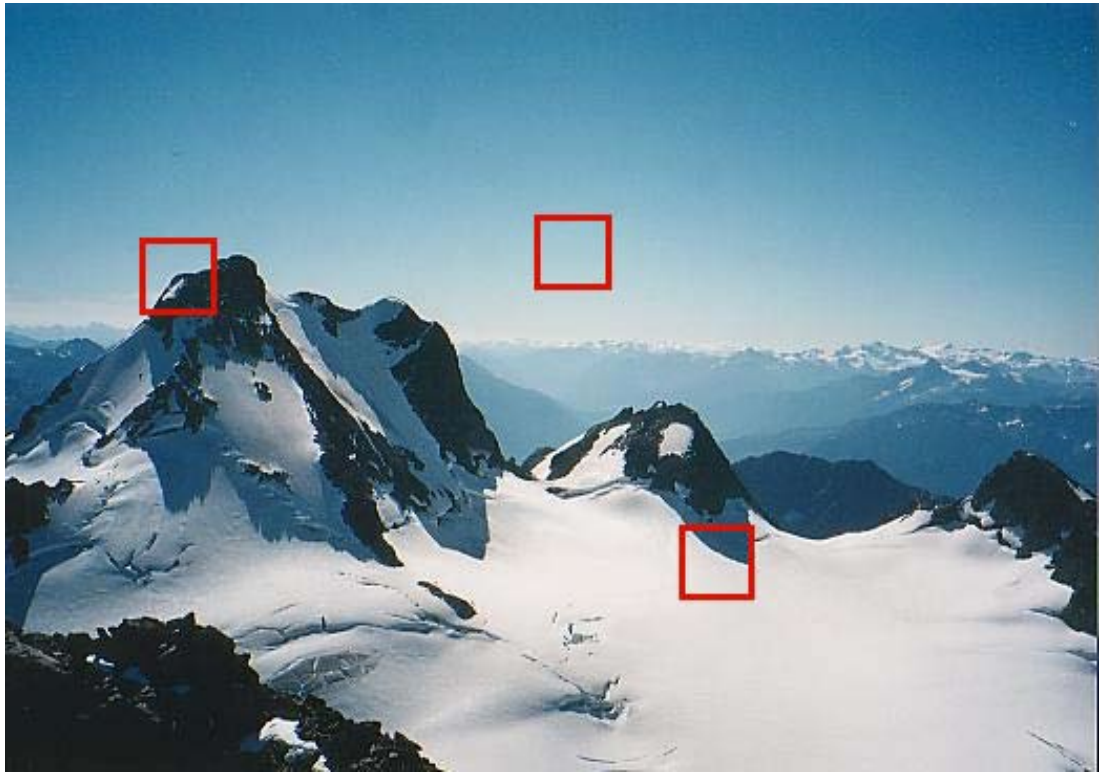
- Panorama Stitching

Source:R. Szeliski

Motivation for Interest Points Detection

- Panorama stitching (cont'd)





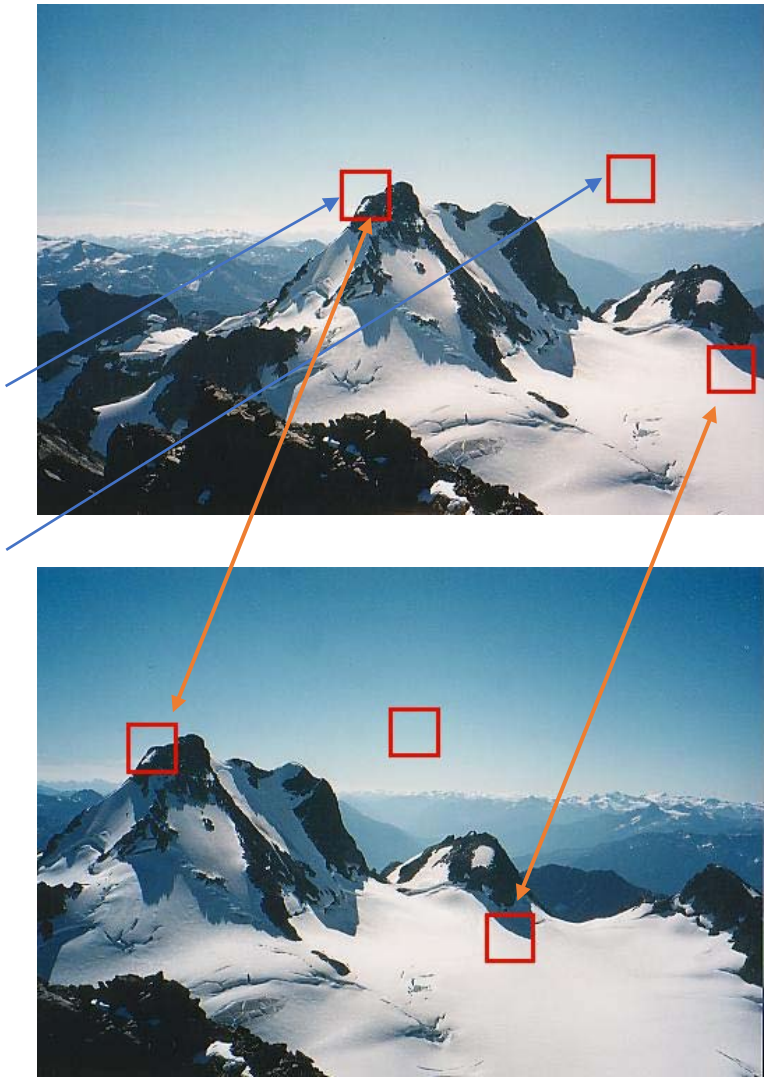
Source:R. Szeliski

Other applicants of Interest point detection

- Automate object tracking
- Point matching for computing disparity
- Stereo calibration
- Motion based segmentation
- Recognition
- 3D object reconstruction
- Robot navigation
- Image retrieval and indexing

General Approach

- Detection: Identify the interest points
- Define a region of each keypoint $X_1 = [x_1^1 \quad \dots \quad x_d^1]$
- Extract and normalize the feature contents $X_1 = [x_1^2 \quad \dots \quad x_d^2]$
- Compute the feature vector descriptor for each normalized region
- Match the feature vector descriptor



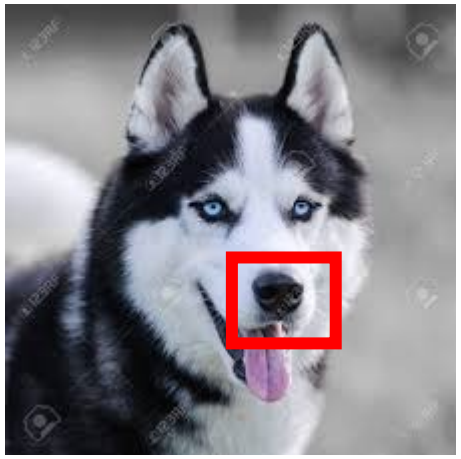
Why local features?

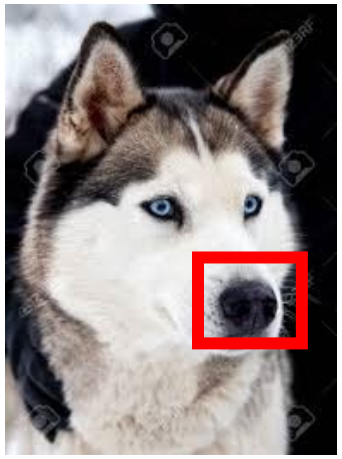
- Describe and match with local regions to increase robustness to

-Occlusions

- Articulations

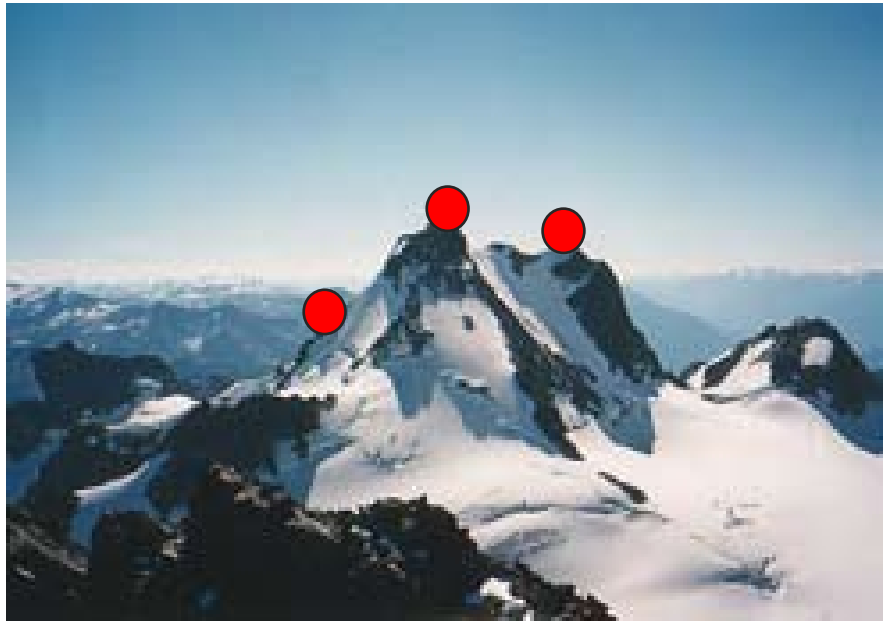
- Intra-category variations

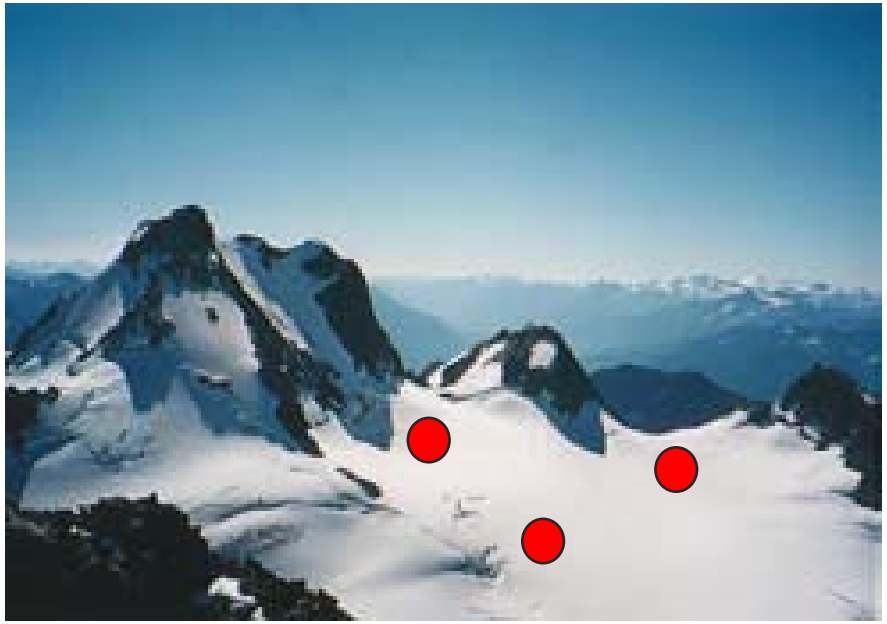




Common Requirements

- We want to detect the same point independently in both images



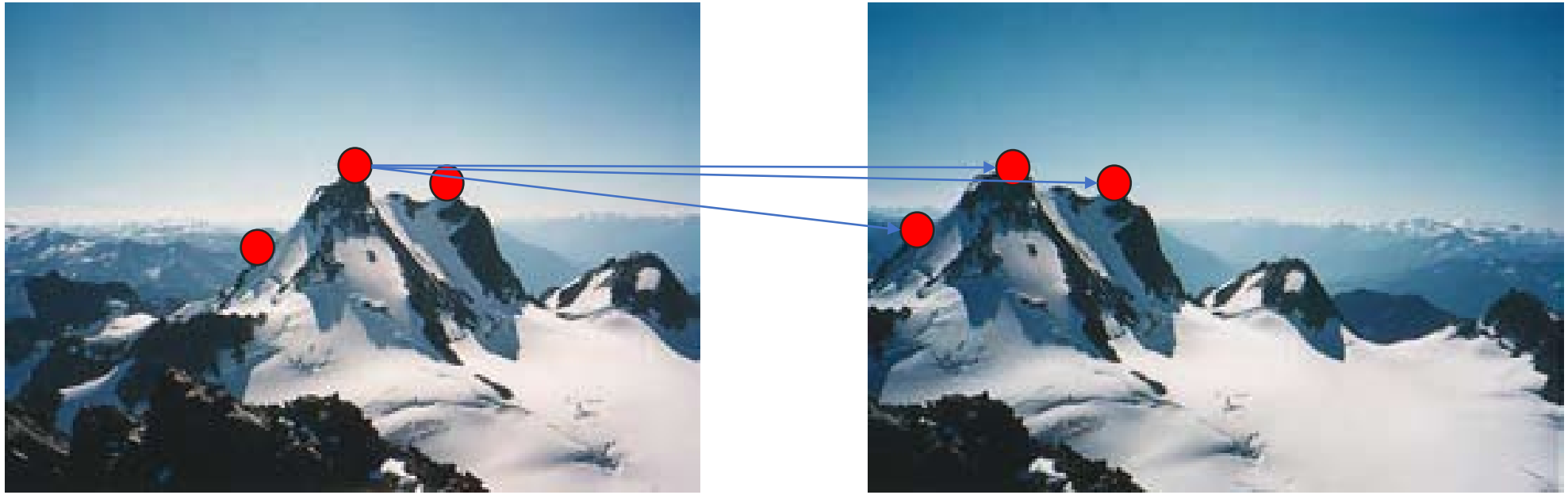


No Chance to match!

Need a repeatable detector!

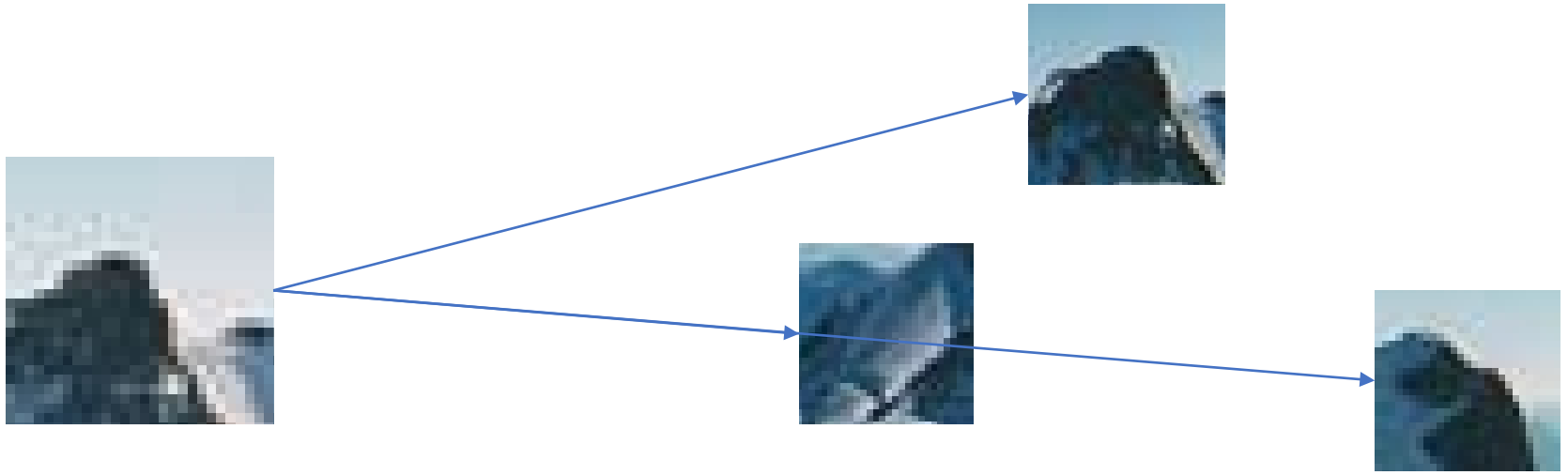
Common Requirements

- We want to be able to match which point goes with which.



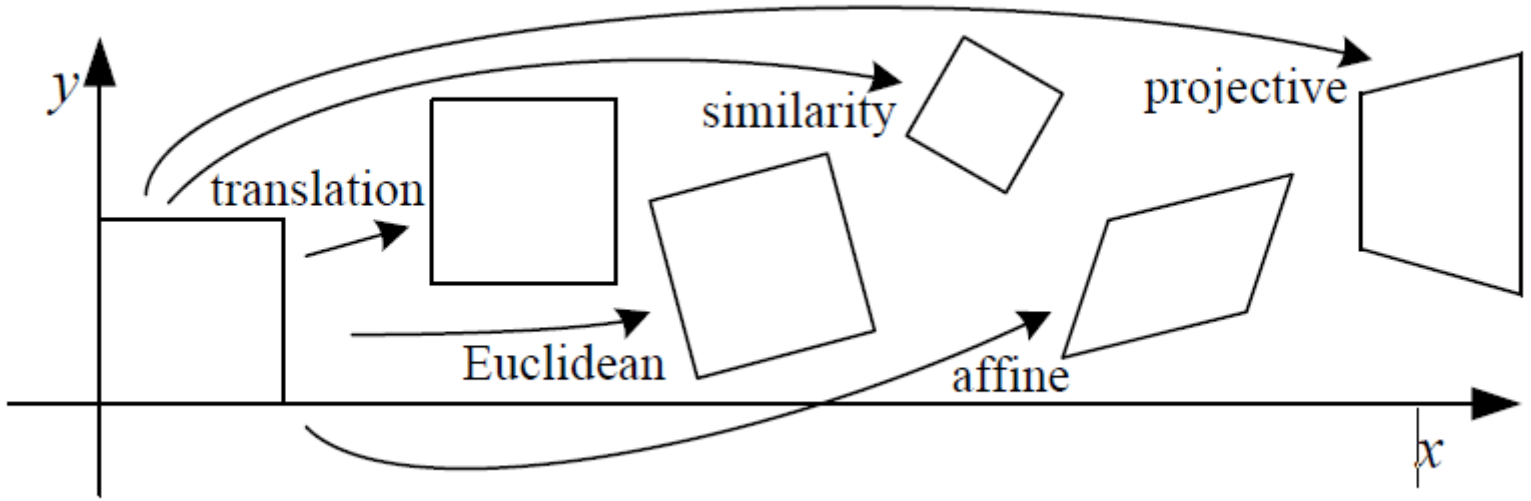
Common Requirements

- We want to be able to match which point goes with which



- Must be *invariance* to *geometric* transformations and *photometric* transformations between two views.

Levels of Geometric Invariance

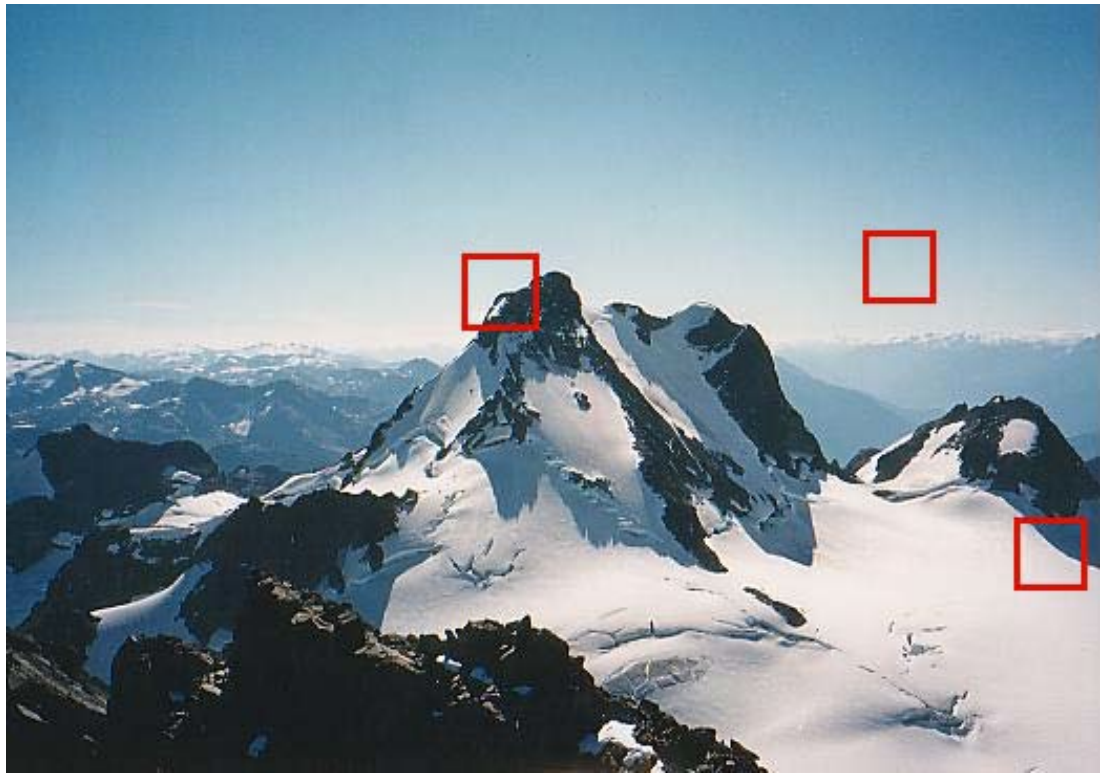


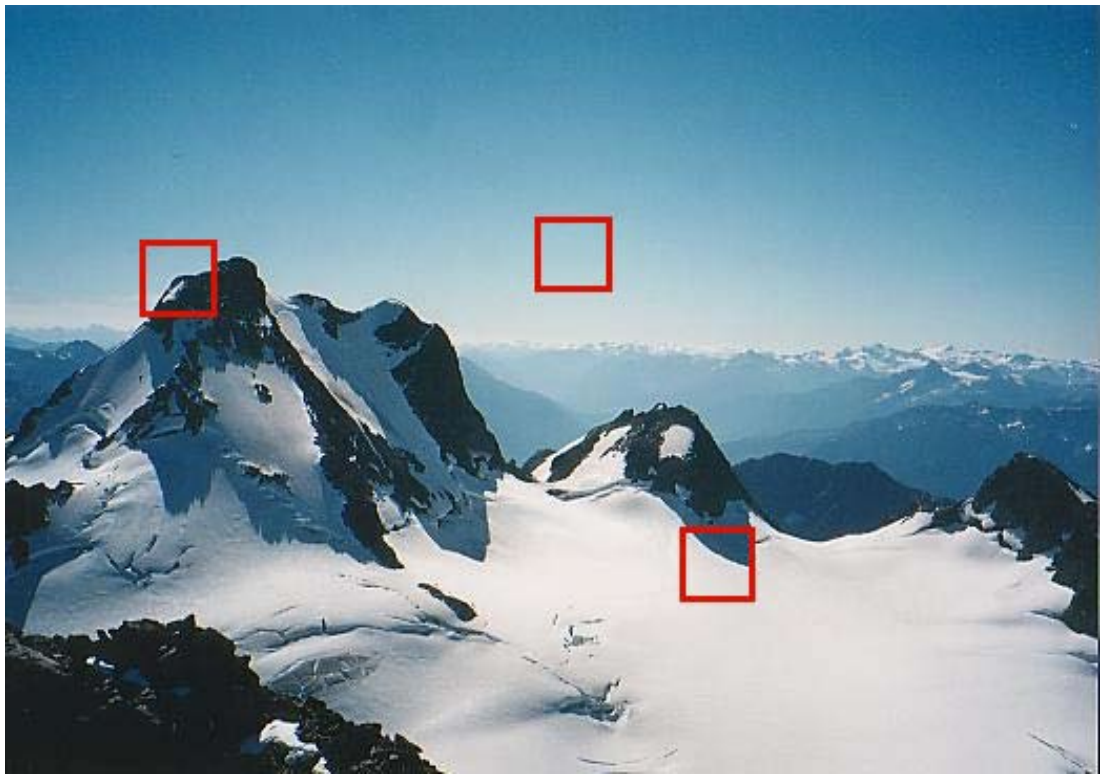
Photometric Transformations

- Often modeled as a linear transformation:
  - Scaling + Offset

Slide credit: Tinne Tuytelaars

What features should we choose?



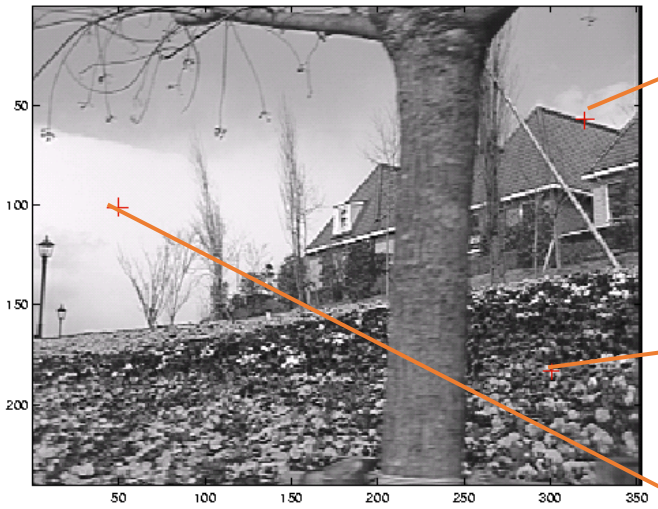


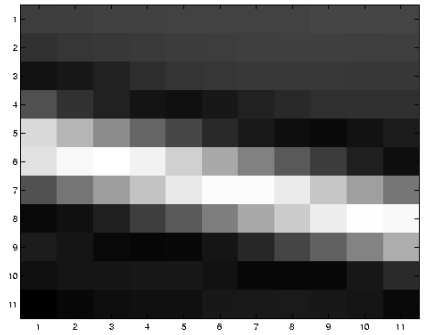
Use patches with gradient in at least two different orientations!

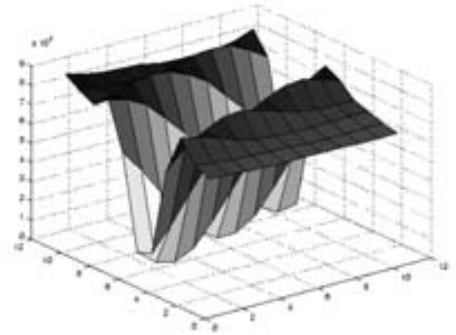
Source:R. Szeliski

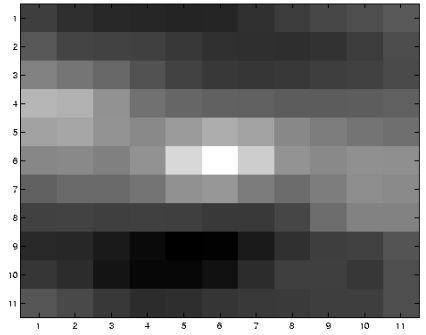
What features should we choose?

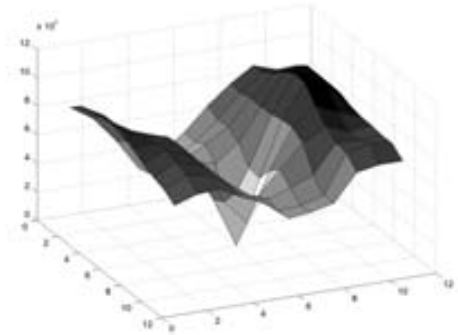
Auto-correlation

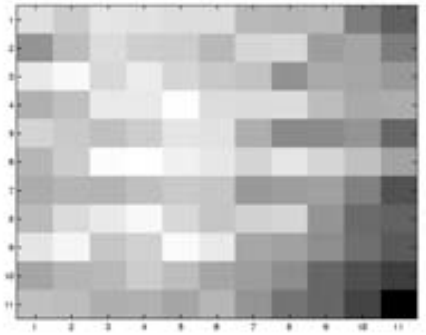


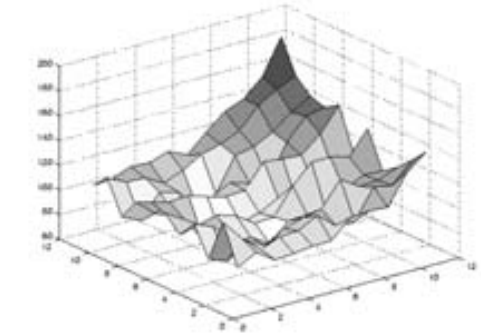












Correlation VS SSD

To match two images, we find i , j minimize the Sum of Squares Difference

$$SSD = \sum_k \sum_l (f(k,l) - h(i+k, j+l))^2$$

$$SSD = \sum_k \sum_l (f(k,l)^2 - 2h(i+k, j+l) f(k,l) + h(i+k, j+l)^2)$$

Minimize

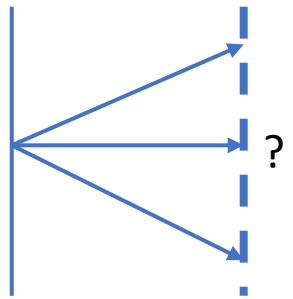
$$\sum_k \sum_l (-2h(i+k, j+l) f(k,l))$$

Maximize

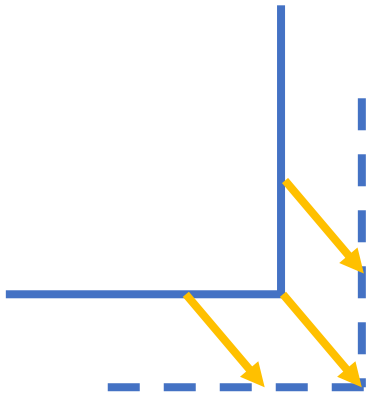
$$\sum_k \sum_l (h(i+k, j+l) f(k,l)) = Correlation$$

What kind of feature to match? Edge VS Corner

- Matching with edges are ambiguous
  - Only normal direction across edge can be identified
  - Tangential direction cannot be identified
- Matching with Corners are unambiguous



ambiguity



unambiguity

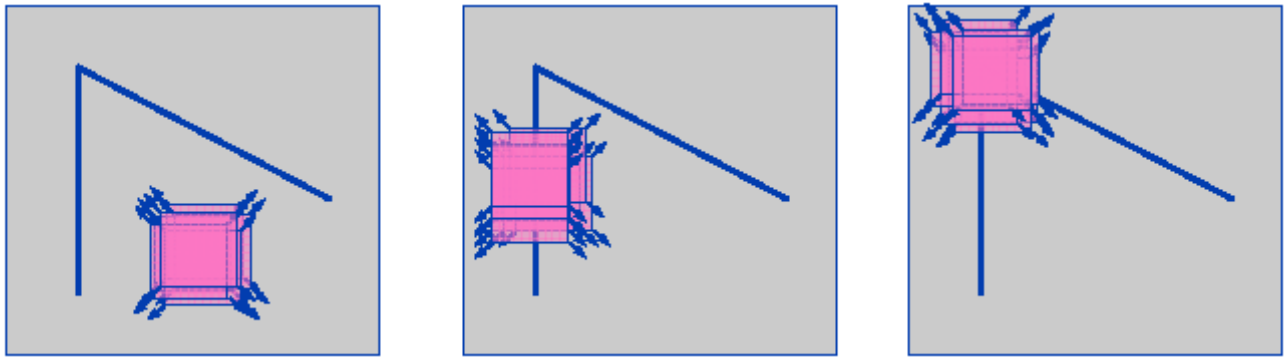
Match with Corner!

Interest Point Detector

- Properties
  - Detect all (or most) true interest points
  - No false interest points
  - Well localized
  - Robust with respect to noise
  - Efficient detection

Possible Approaches to corner detection

- Image gradient has two or more dominant directions near a corner
- Shifting a windows in any direction should give a large change in intensity



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Source: R. Collins

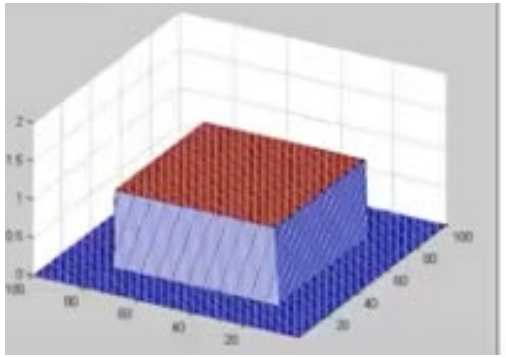
Mathematics of Interest Point Detector

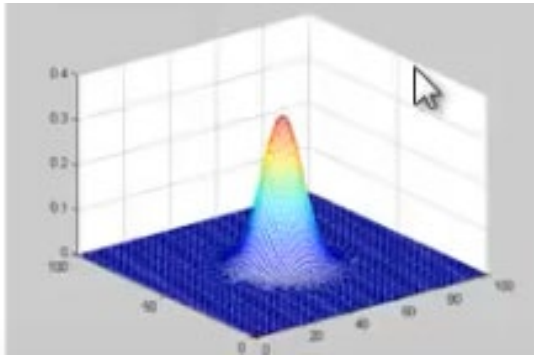
- Change of intensity for the shift (u,v)

- $E(u,v) = \sum \overbrace{w(x,y)}^{window\ function} \left[\overbrace{I(x+u,y+v)}^{shifted\ intensity} - \overbrace{I(x,y)}^{intensity}\right]^2$

- Window Function $w(x,y)$

Square Window





Gaussian Window

Mathematics of Harris Corner Detector

- *Intensity Changes*
  - $E(u,v) = \sum_x \sum_y \overbrace{w(x,y)}^{window\ function} \left[\overbrace{I(x+u,y+v)}^{shifted\ intensity} - \overbrace{I(x,y)}^{intensity}\right]^2$

- By first-order Taylor series approximation
  - $f(x-\Delta x, y-\Delta y) \approx f(x,y) + \left[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right] \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}$
  - $I(x+u,y+v) \approx I(x,y) + \left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right] \begin{bmatrix}u \\ v\end{bmatrix}$
  - $E(u,v) = \sum_x \sum_y w(x,y) \left(\left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right] \begin{bmatrix}u \\ v\end{bmatrix}\right)^2$

Mathematics of Harris Corner Detector

- $E(u,v) = \sum_x \sum_y w(x,y)\left(\left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right]\begin{bmatrix}u\\v\end{bmatrix}\right)^2$

- $E(u,v) = \sum_x \sum_y w(x,y)\left[\begin{bmatrix}I_x & I_y\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}\right]^2$

Mathematics of Harris Detector (Cont'd)

- $E(u,v) = \sum_x \sum_y w(x,y) \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$

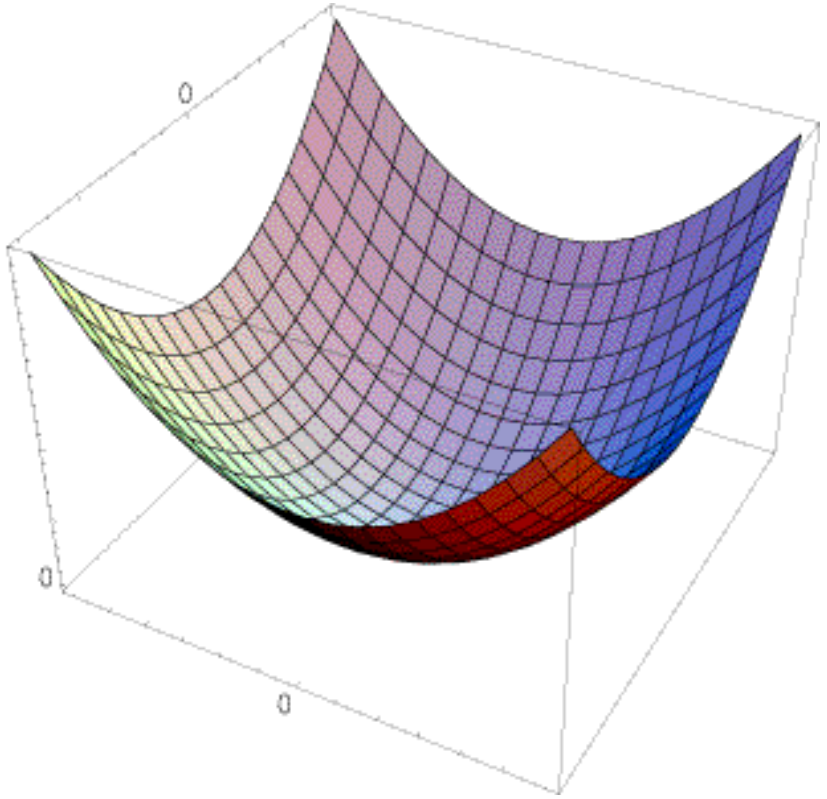
- Since $u^2 = u^T u \Rightarrow \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right]^2 = \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right]^T \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right]$
  - $E(u,v) = \sum_x \sum_y w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$
  - $E(u,v) = \begin{bmatrix} u & v \end{bmatrix} \left[\sum_x \sum_y w(x,y) \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \right] \begin{bmatrix} u \\ v \end{bmatrix}$
  - $E(u,v) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$ $\quad M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$ M is Second Moment Matrix

Interpreting the second moment matrix

- The surface $E(u, v)$ is locally approximated by a quadratic form.

- $E(u, v) = [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$

- $M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$



Review: Eigenvalues and eigenvectors

- Definition of ***Eigenvalue***/***Eigenvector***
  - Given a square matrix $\mathrm{A}_{m\times m}$ and vector $\mathbf{x}$ represented below

$$\mathrm{A}\mathbf{x} = \boldsymbol{\lambda}\mathbf{x}$$

where

$\boldsymbol{\lambda}$ is called the ***eigenvalue*** of A with the corresponding ***eigenvector*** $\mathbf{x}$

Review: eigenvalues and eigenvectors

Let

$$\mathrm{A}\mathbf{X} = \boldsymbol{\lambda}\mathbf{X}$$

we have

$$\mathrm{A}\mathbf{X} - \boldsymbol{\lambda}\mathbf{X} = \mathbf{0}$$

$$(\mathrm{A} - \boldsymbol{\lambda}\boldsymbol{I})\mathbf{X} = \mathbf{0}$$

Solving

$$\det(\mathrm{A} - \boldsymbol{\lambda}\boldsymbol{I}) = 0$$

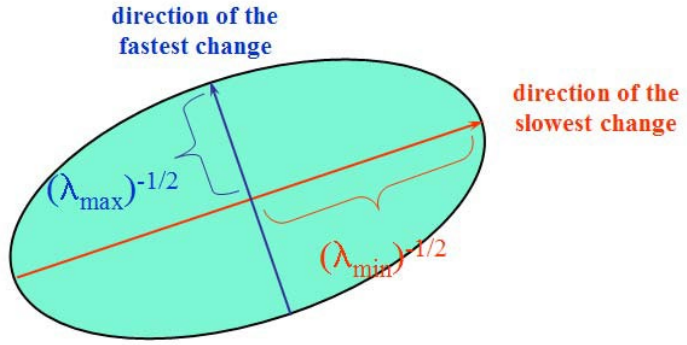
Mathematics of Harris Detector (Cont'd)

- $E(u,v) = [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

- Since E(u,v) is symmetric, let λ_1 and λ_2 be ***eigenvalues*** of M, we have

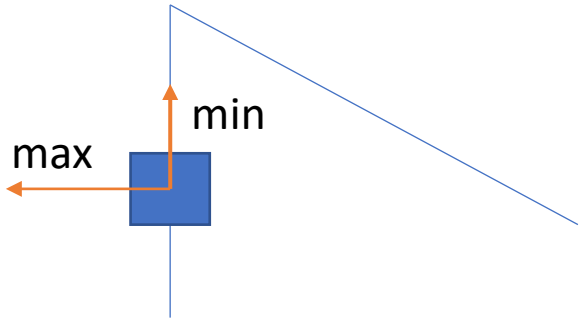
- $M = Q^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathrm{Q}$

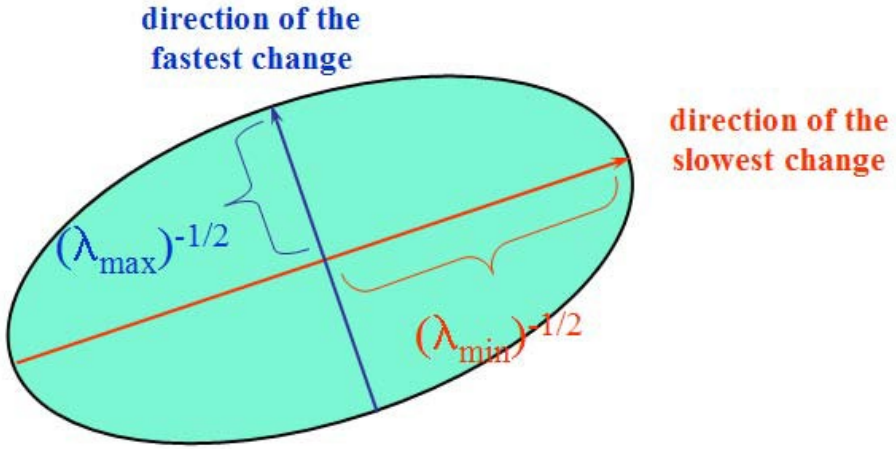


- We can visualize M as an ellipse with axis lengths and orientation determined by eignenvalues and eigenvectors repectively

Mathematics of Harris Detector (Cont'd)

- The eigenvectors of M indicates the direction of intensity changes
- The eigenvalues of M encode the magnitude of intensity changes



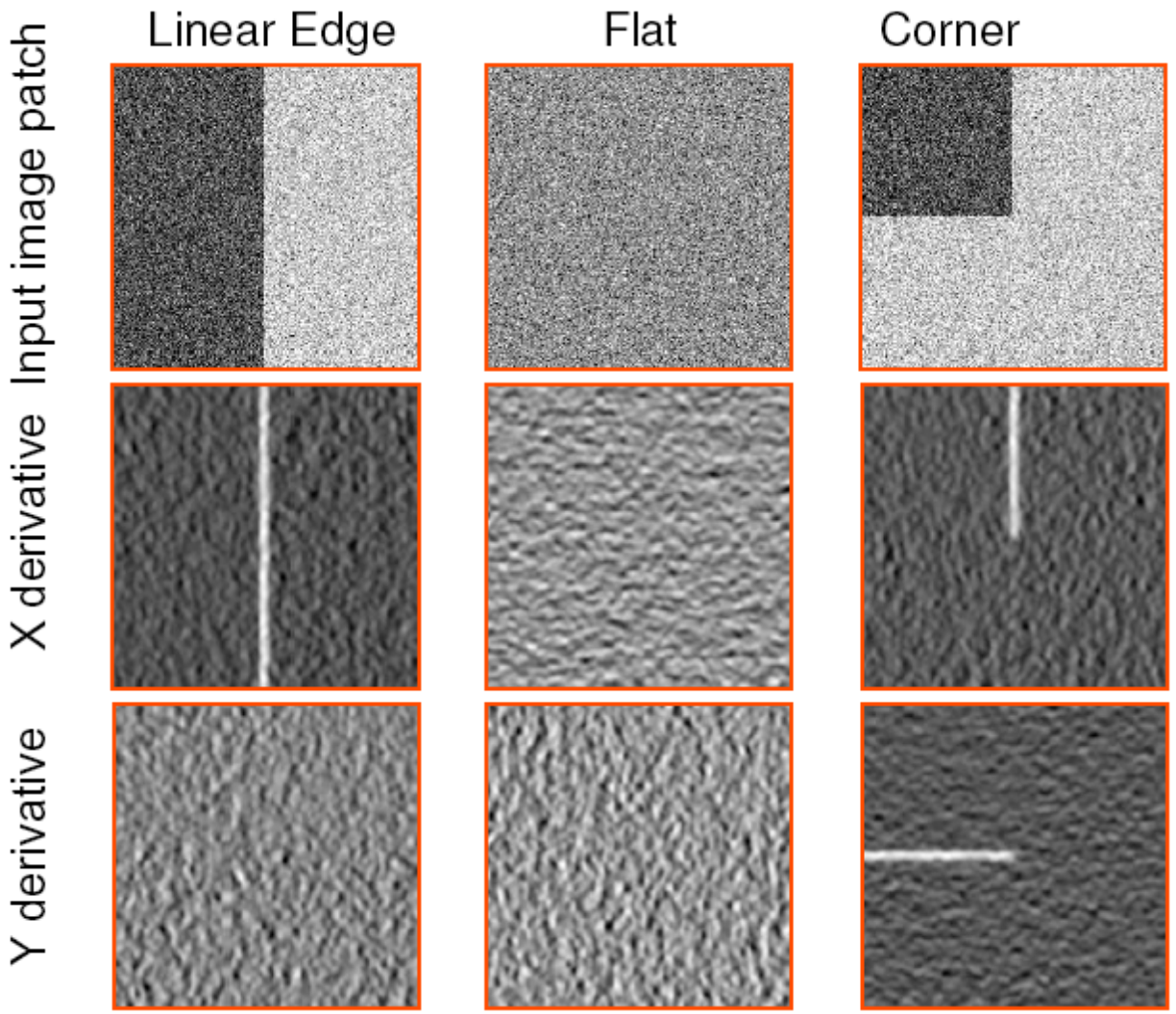


More intuitive way to understand second moment matrix

- Treat gradient vectors as a set of (Ix, Iy) points with center of mass defined at (0,0)
- Fit an ellipse to that set of points via scatter matrix
- Analyse ellipse parameters for varying cases

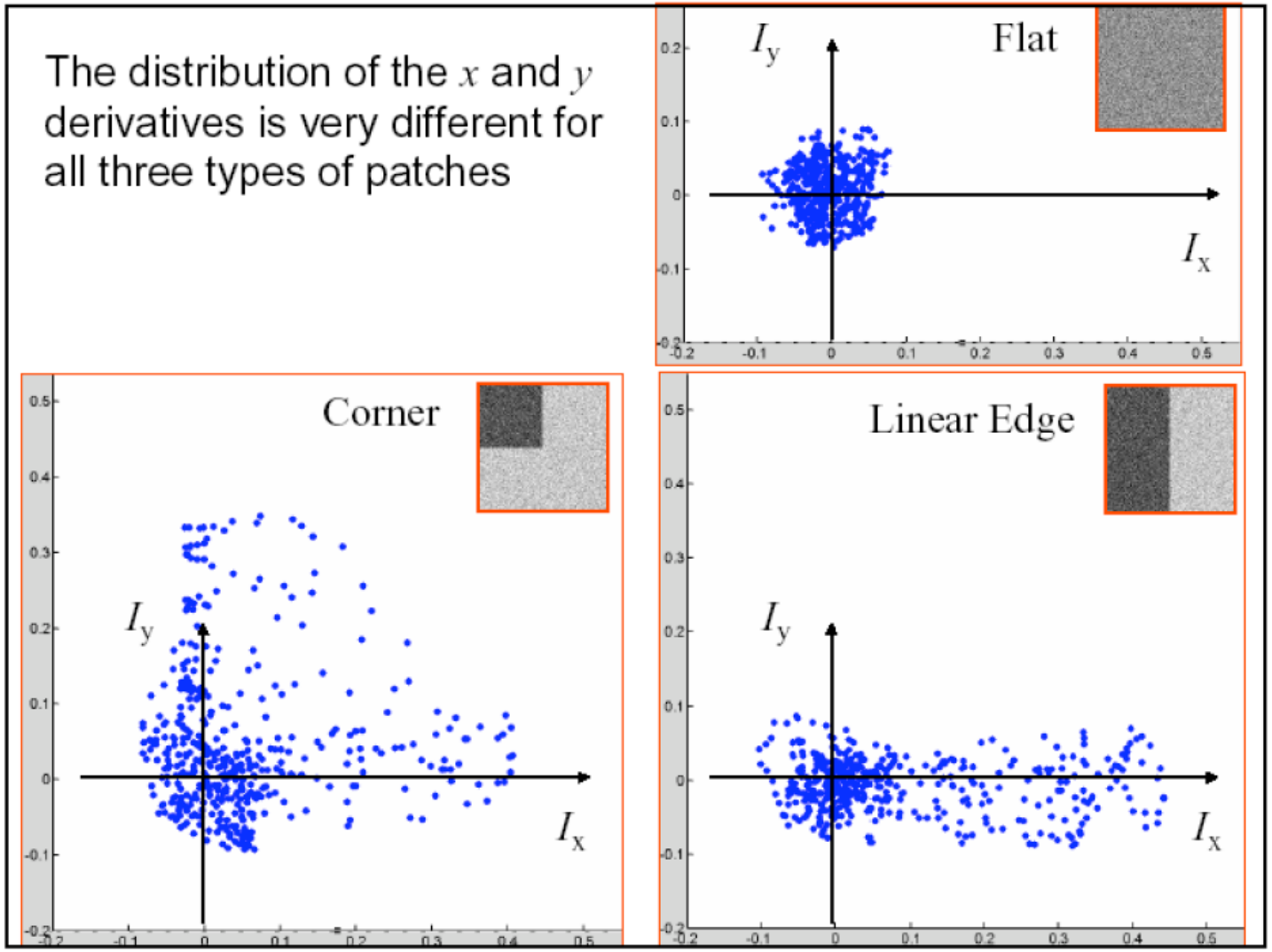
Slide credit: Robert Collins CSE4865

Edge and Derivatives



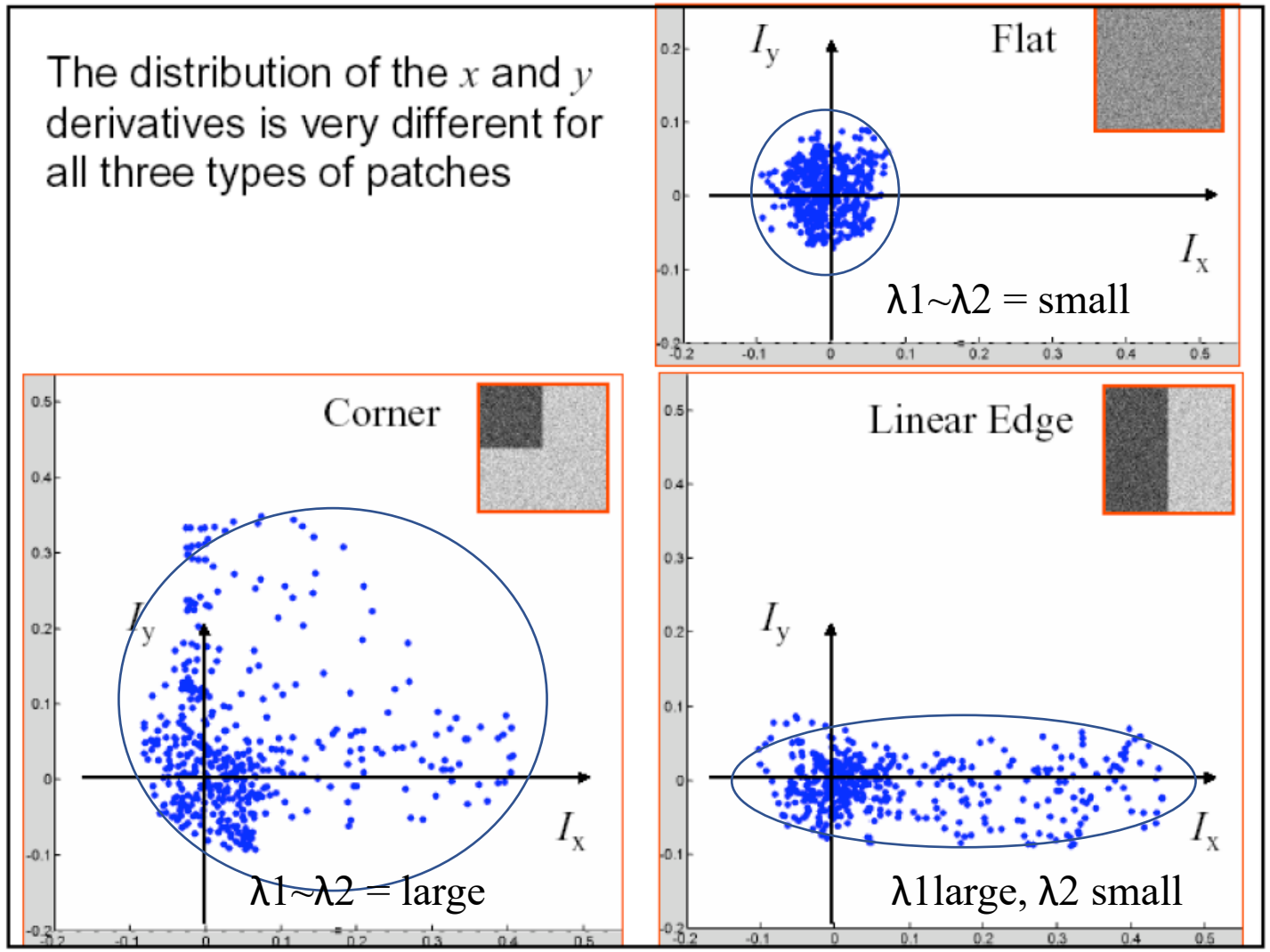
Slide credit: Robert Collins CSE4865

Distribution of I_x and I_y as 2D points



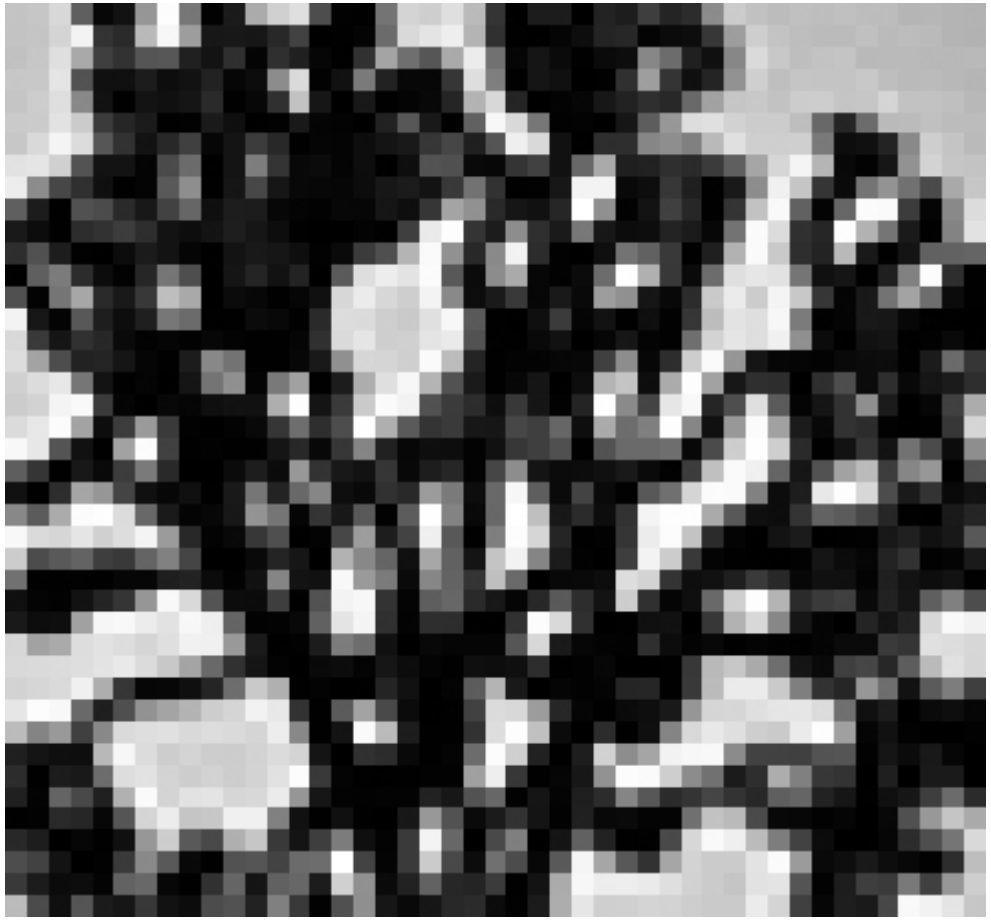
Slide credit: Robert Collins CSE4865

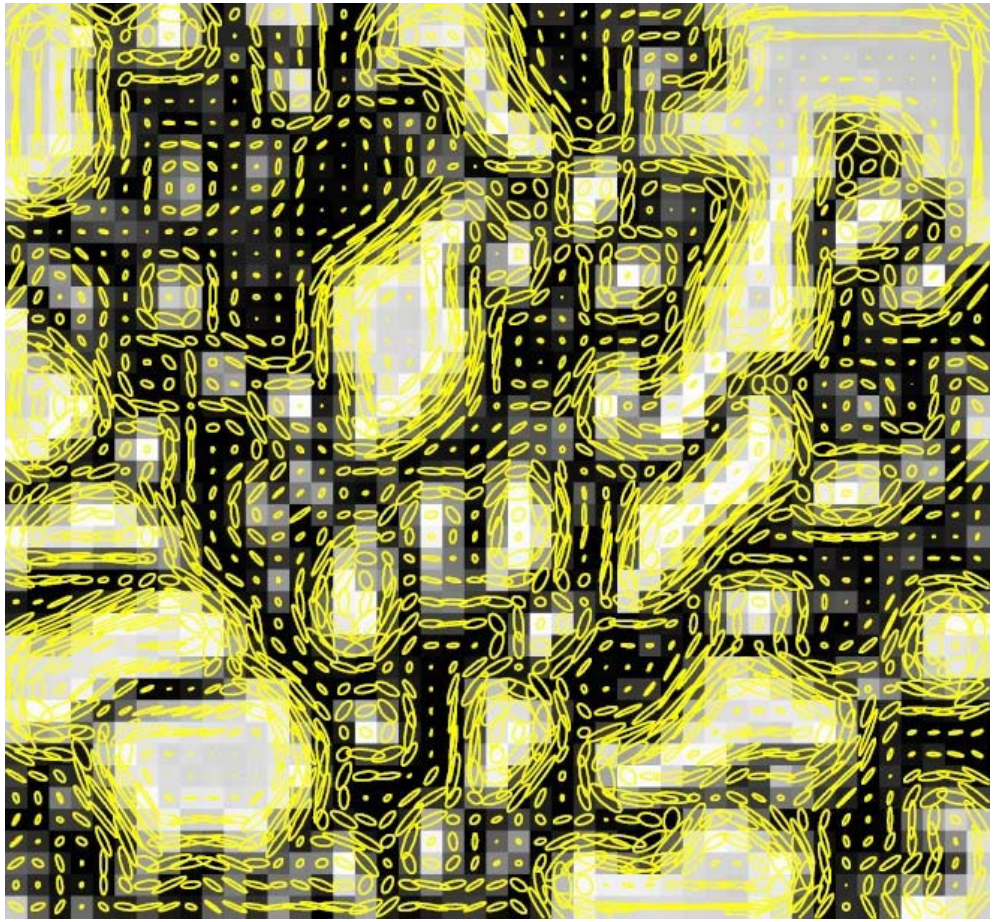
Fit an ellipse on I_x and I_y



Slide credit: Robert Collins CSE4865

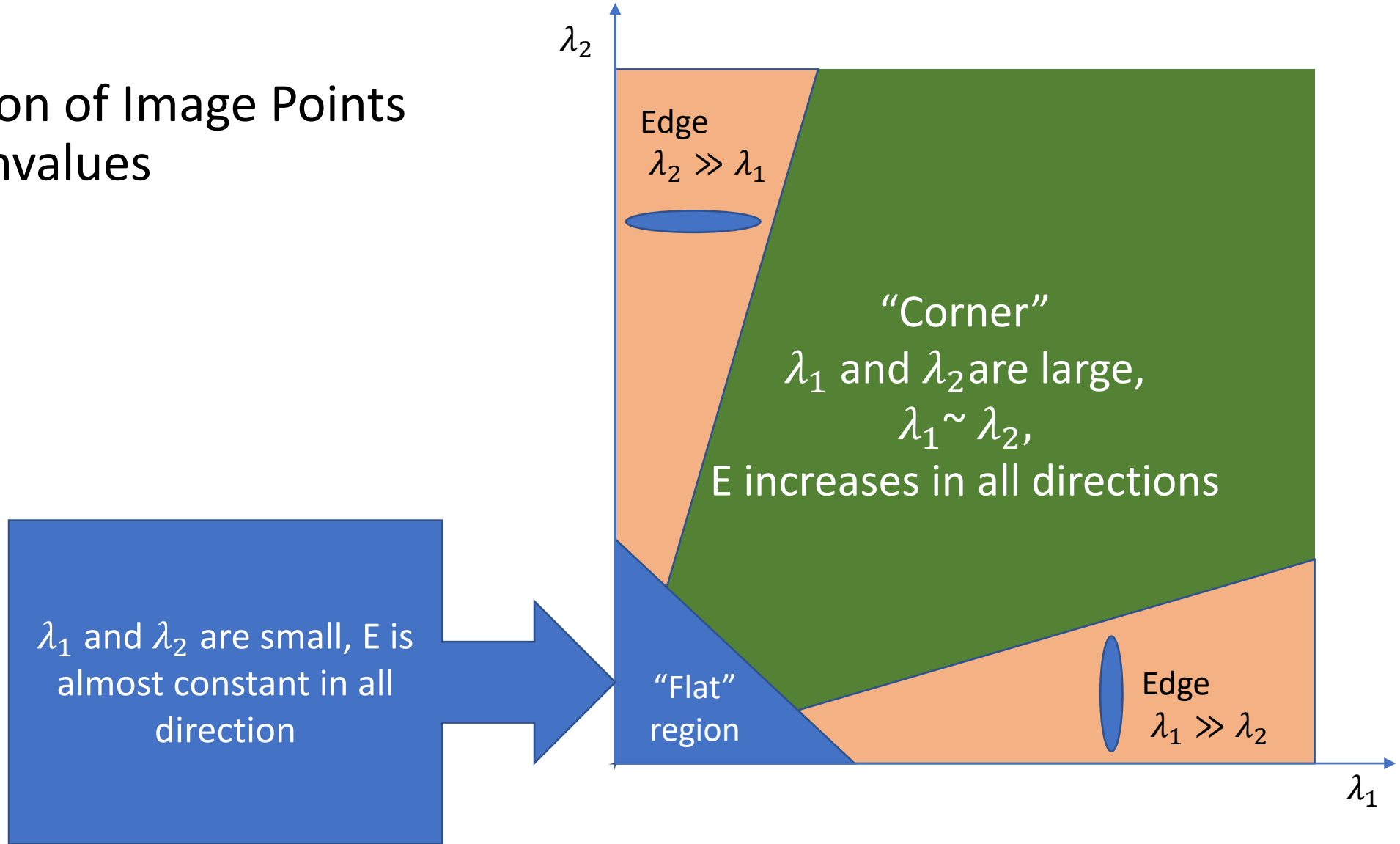
Visualization of second moment matrices





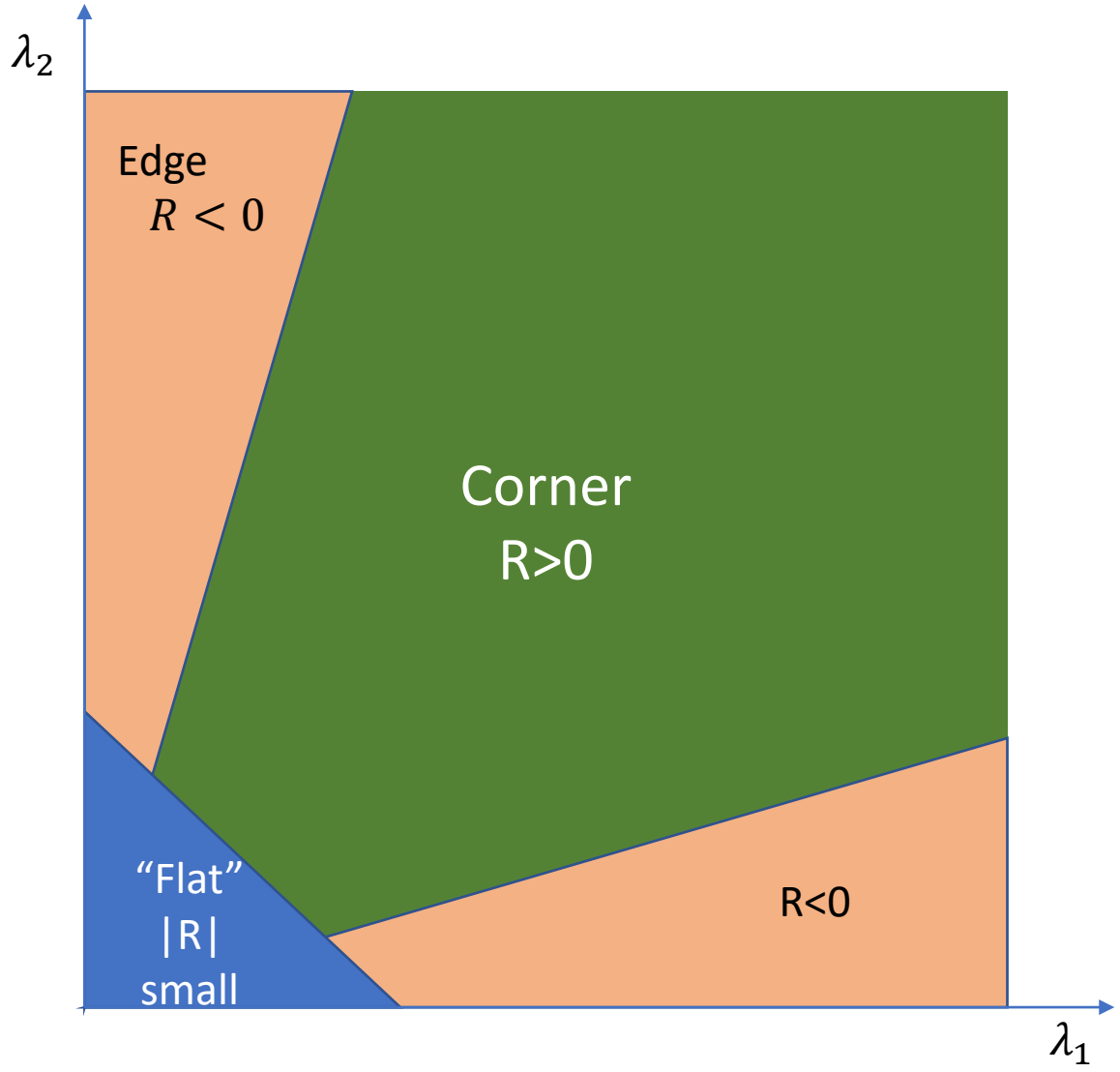
Classification of Image Points by eignvalues

- Classification of Image Points using eigenvalues

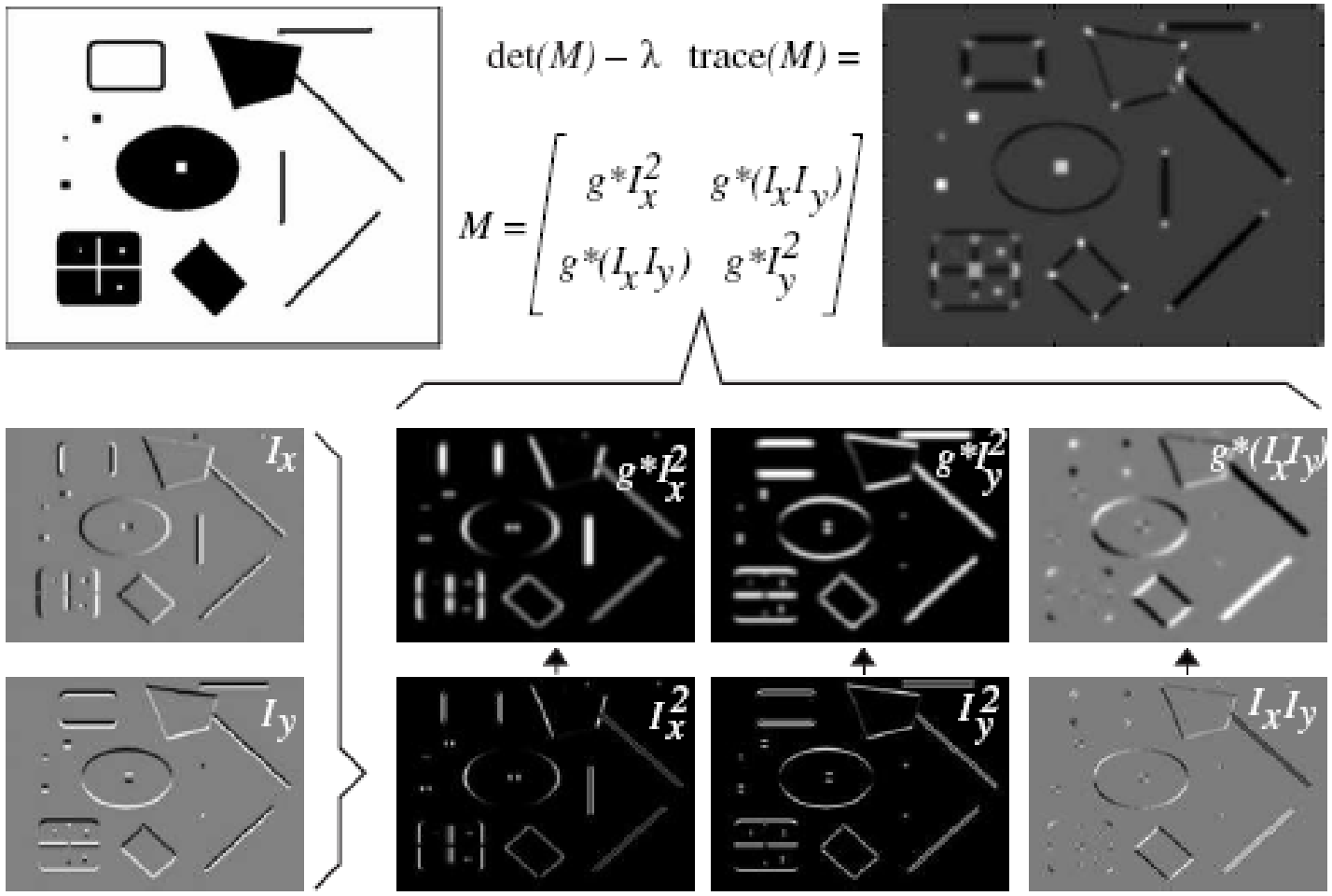


Mathematics of Harris Detector

- Measure of corner response
  - $R = \det(M) - k\left(trace(M)\right)^2$
  - $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$
    - k: constant (0.04~0.06)
- R depends of λ_1 and λ_2
  - Large R is corner
  - R is negative with large magnitude is edge
  - |R| small is flat region

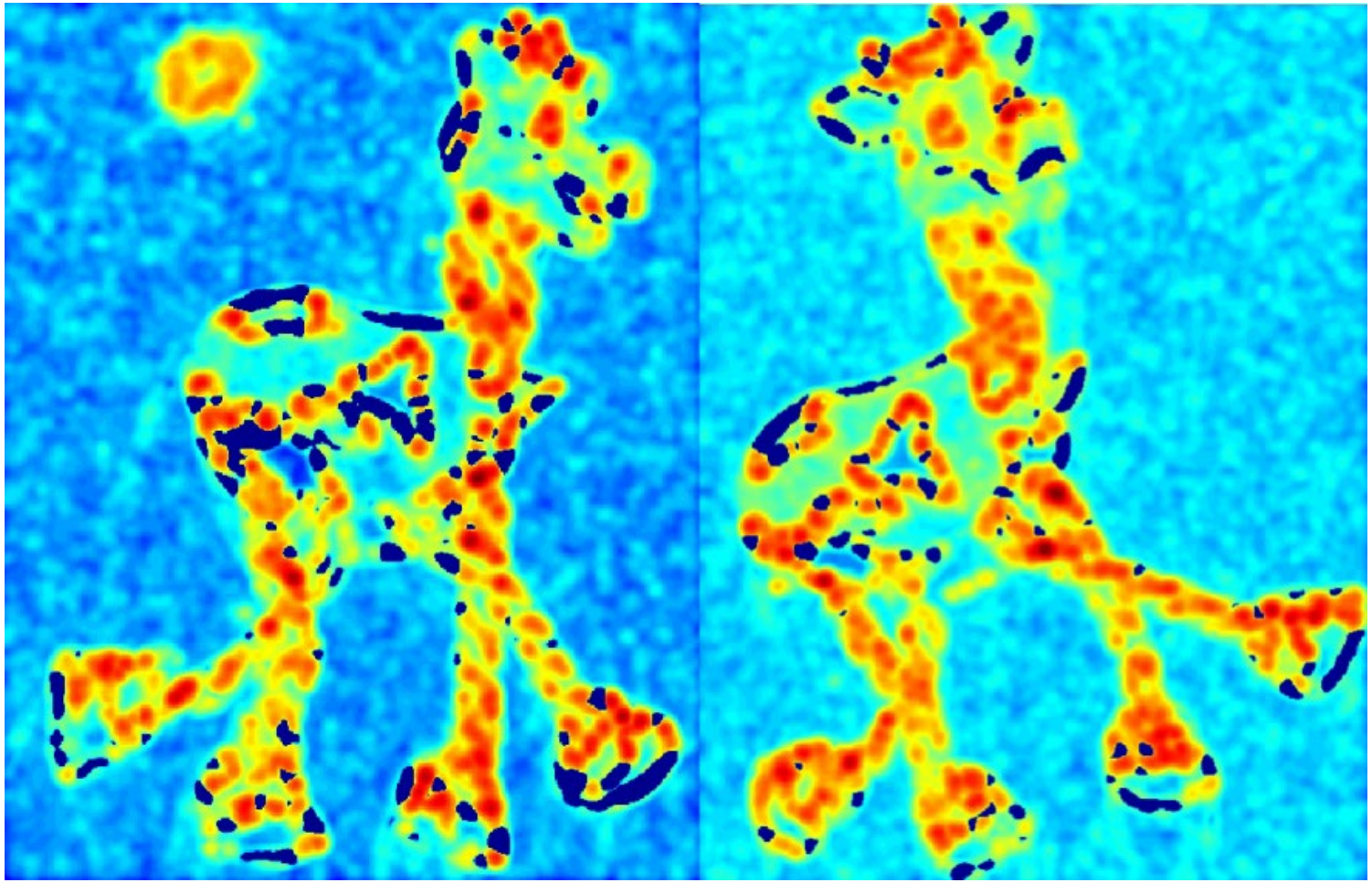


Harris Detector - Example



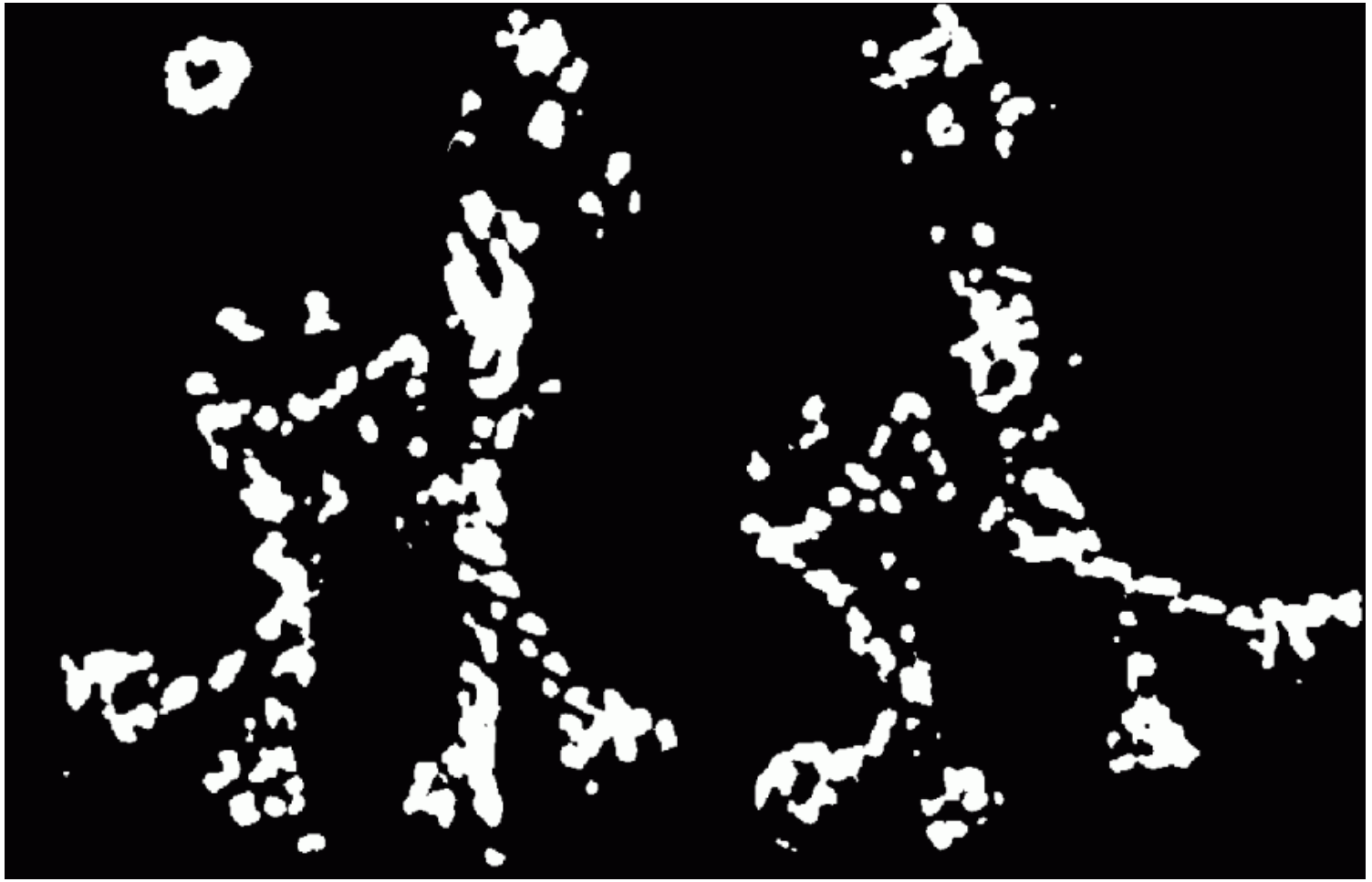
Harris Detector - Example

Harris Detector - Example



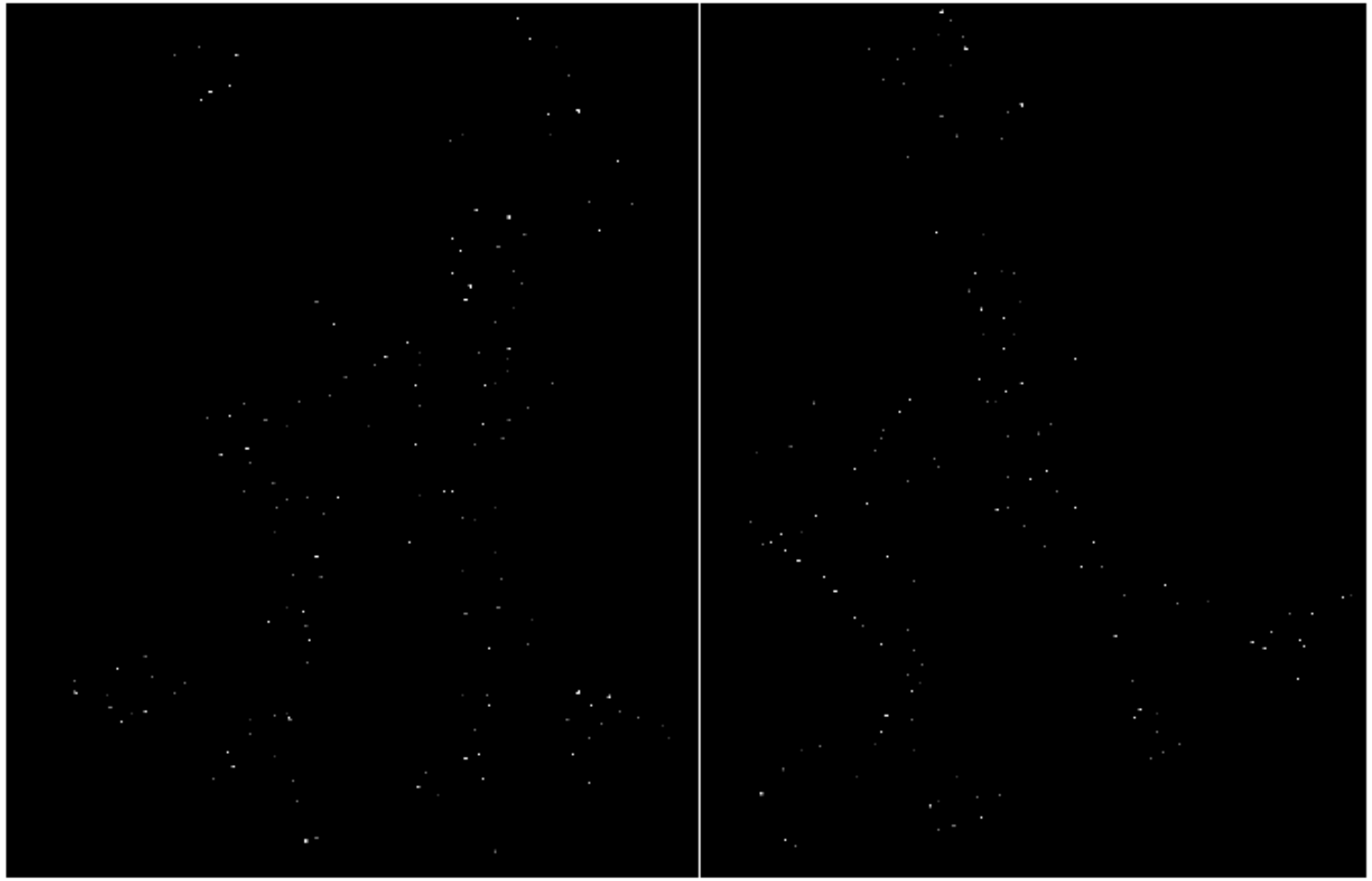
Corner Response R

Harris Detector - Example



Corner Response R> threshold

Harris Detector - Example



Take only points of local maxima of R

Harris Detector - Example

Map Corners on the original image

Algorithm Summary

- 1) Compute horizontal and vertical derivative of image I_x and I_y
- 2) Compute the local structure of matrix

$$\mathrm{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

- 3) Evaluate the corner response R

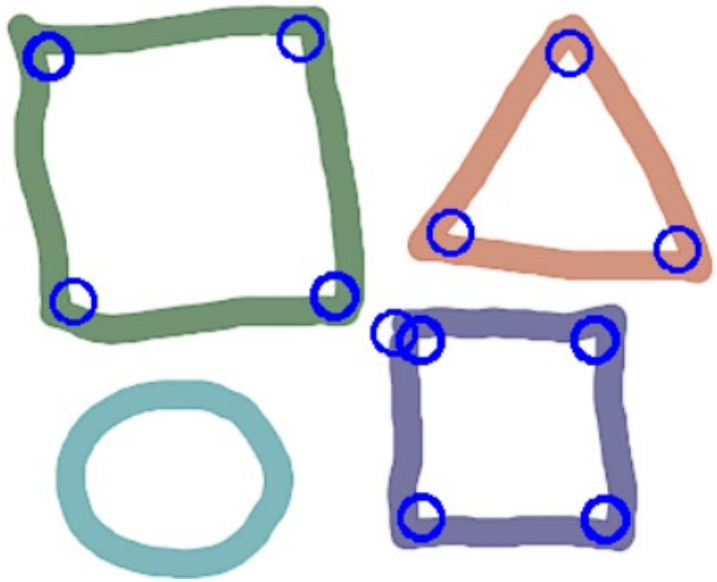
$$R = \det(M) - k\big(trace(M)\big)^2$$ *k*- empirical constant, k=0.04~0.06

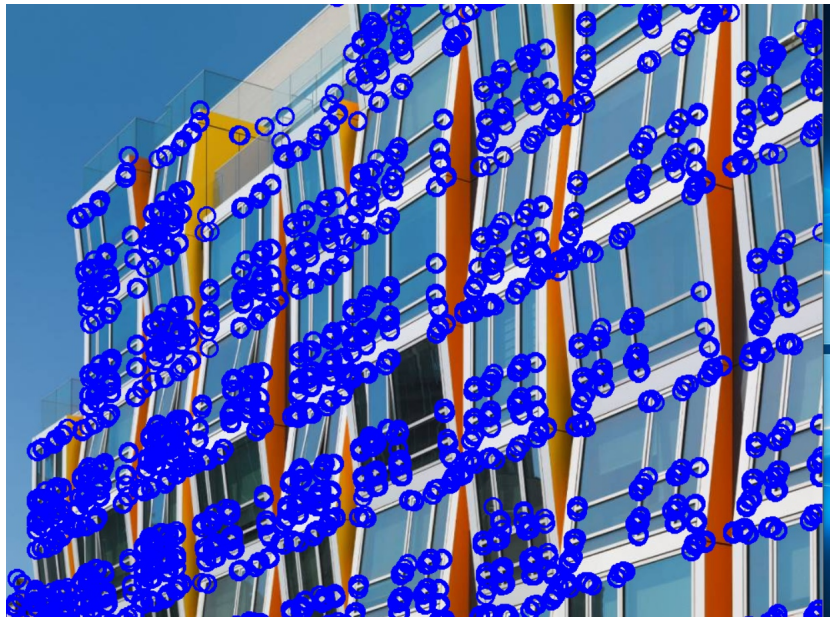
- 4) Choose the best corner by selecting a threshold on the response R>threshold

5) Take the point of locally maximum R as the detected feature points by perform non-maximal suppression

(i.e.) pixel where R is bigger than all the 4 or 8 neighbors.

Examples of Harris Detector



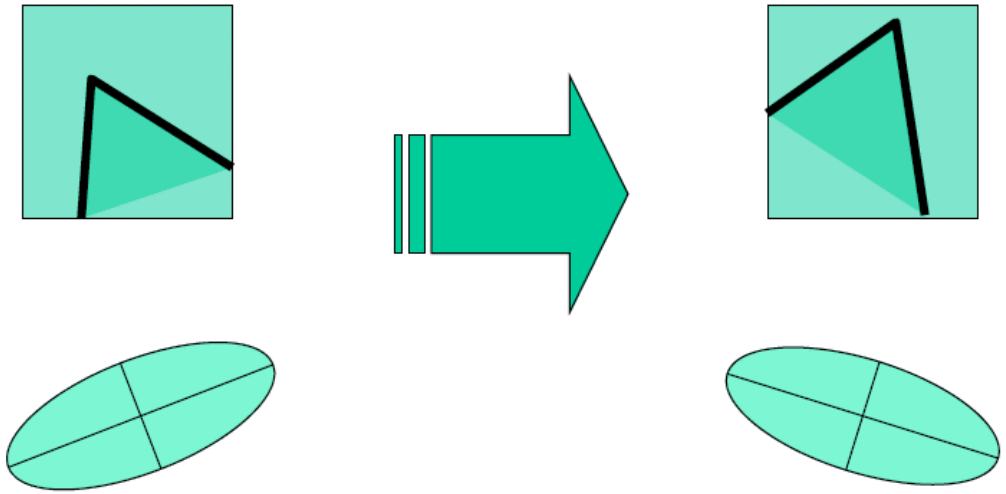


Examples of Harris Detector

Results are good for finding stereo correspondences

Harris Detector Properties

- Translation & Rotation invariance?



- Ellipse rotates but its shape (i.e. eigenvalues) remains the same

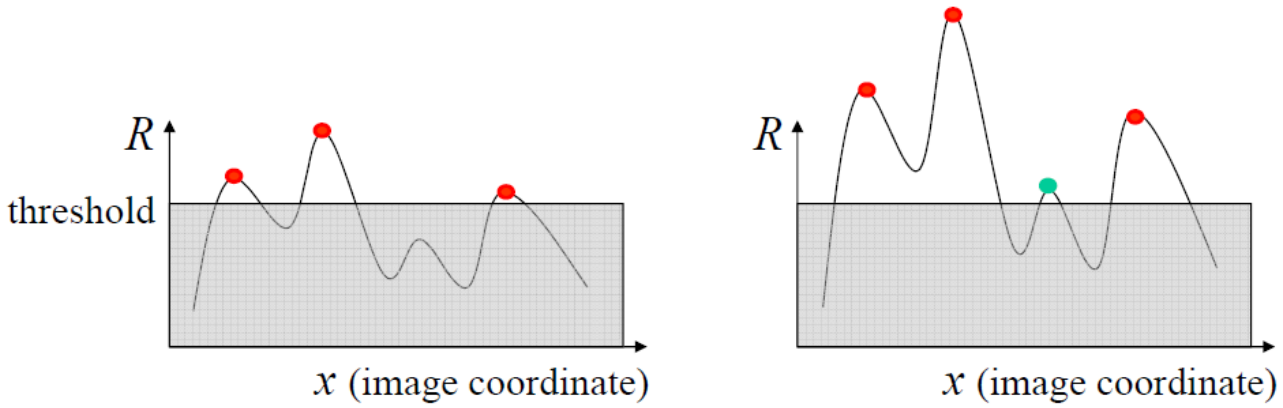
Corner response R is invariant to image rotation!

Harris Detector Properties

- Translation & Rotation invariance?

Harris Detector Properties

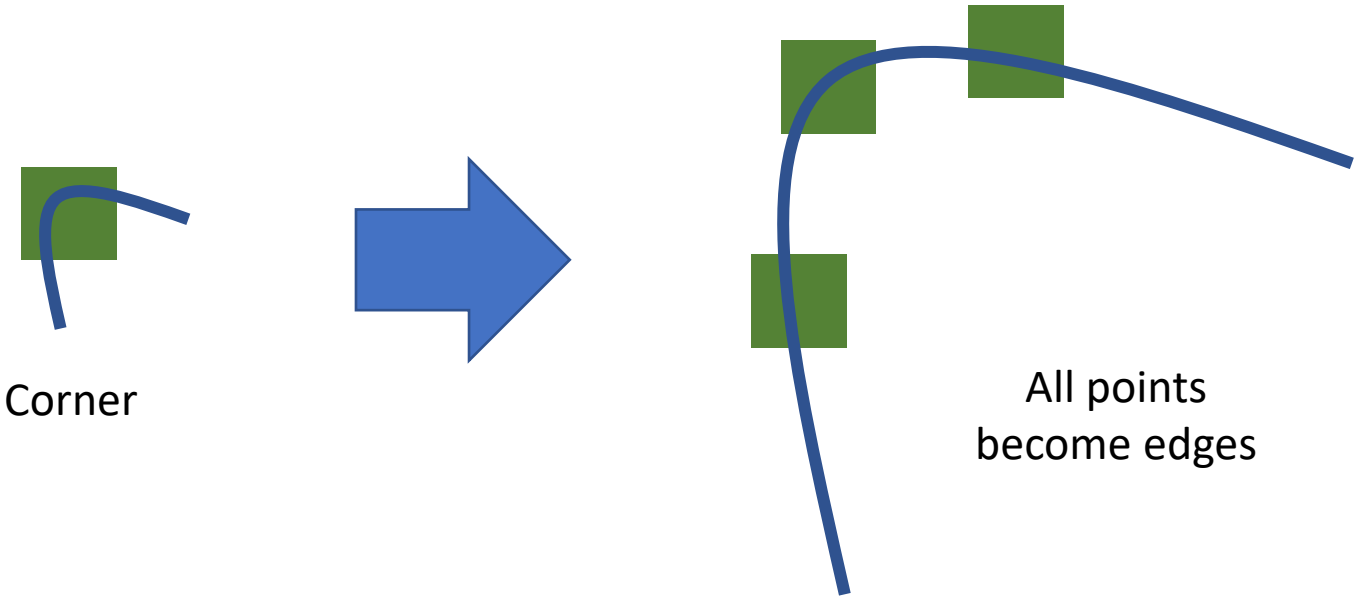
- Photometric Invariance?
- Affine intensity change?
- Since only derivatives are used, it is invariance to intensity shift $I(x, y) \rightarrow I(x, y) + b$
- In



Partially invariant to affine intensity change

Harris Detector: Properties

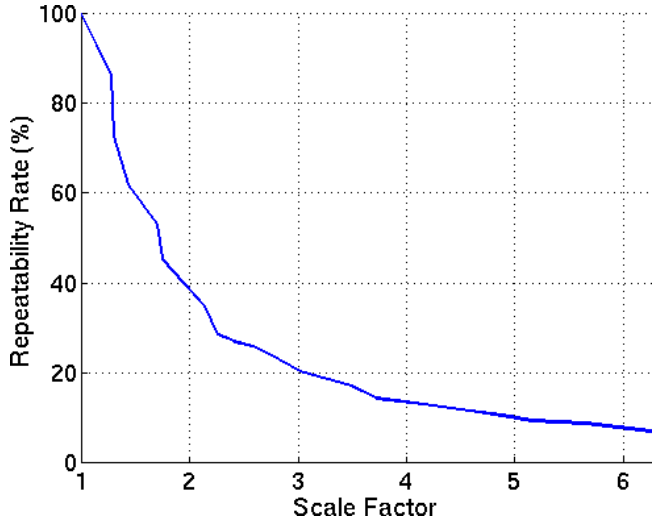
- Translation & Rotation invariance?
- Scale Invariance?

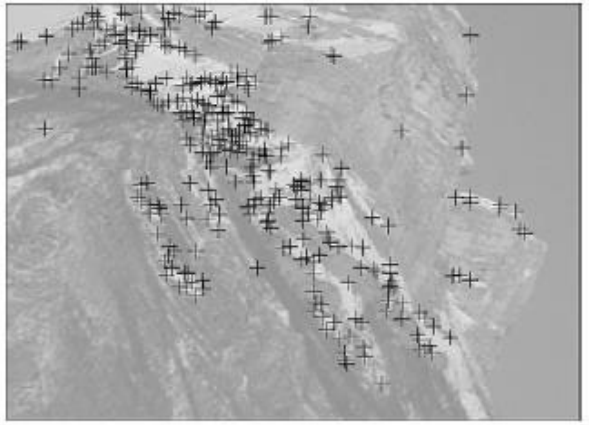


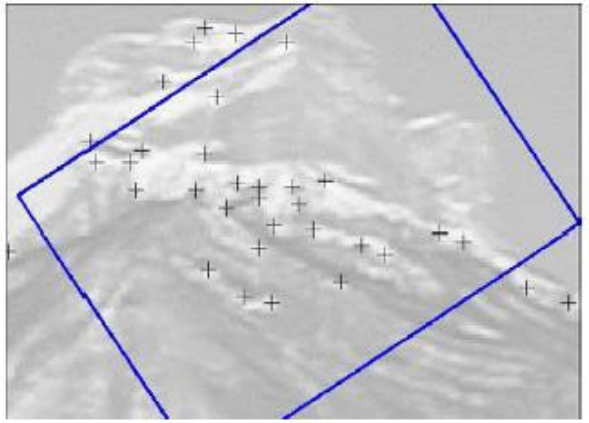
Corner response R is NOT invariant to image scale

Harris Detector

- Sensitive to:
  - Scale change
  - Significant viewpoint change
  - Significant contrast change







Slide credit: R. Dockter

Today's Agenda

- Edge Detection
  - Gradient Detector
  - Laplacian of Gaussian (Marr-Hildreth)
  - Gradient of Gaussian (Canny)
- Interest Point Detection
  - Harris Corner Detection
- Scale-invariant Region Selection

Scale Invariant Region Selection

- Consider regions of difference sizes around a point

- Regions of corresponding sizes will look the same in both images
- How to select the scale for detection?

From Points to Region

- Harris Detector define interest points
- Precise localizations
- High repeatability

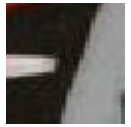
- In order to compare those points, we need to compute a descriptor over region.
- How can we define a region in a scale invariant manner?

- i.e. how can we detect scale invariant interest regions?

Source: Bastian Leibe

First Trial - Exhaustive Search

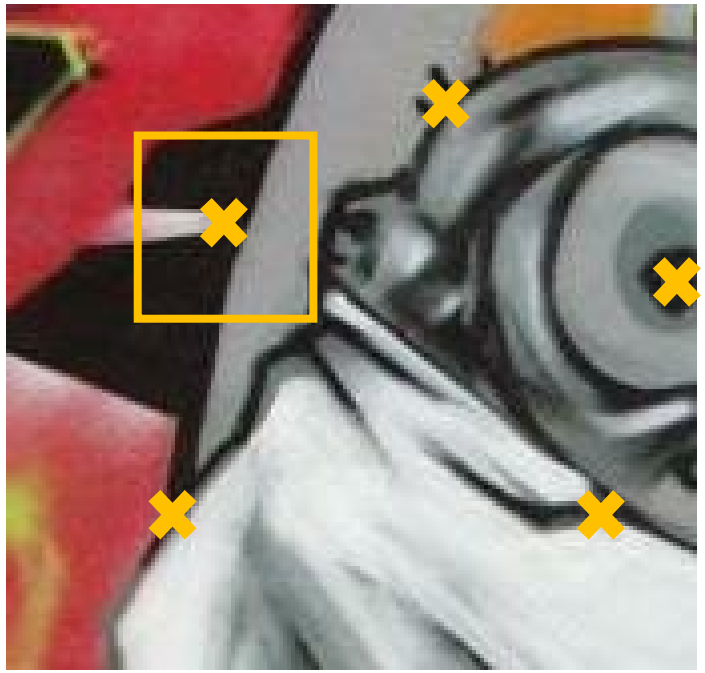
- Multi-Scale Approach – compare descriptors while varying the patch size



Slide Source:T Tuytelaars

First Trial - Exhaustive Search

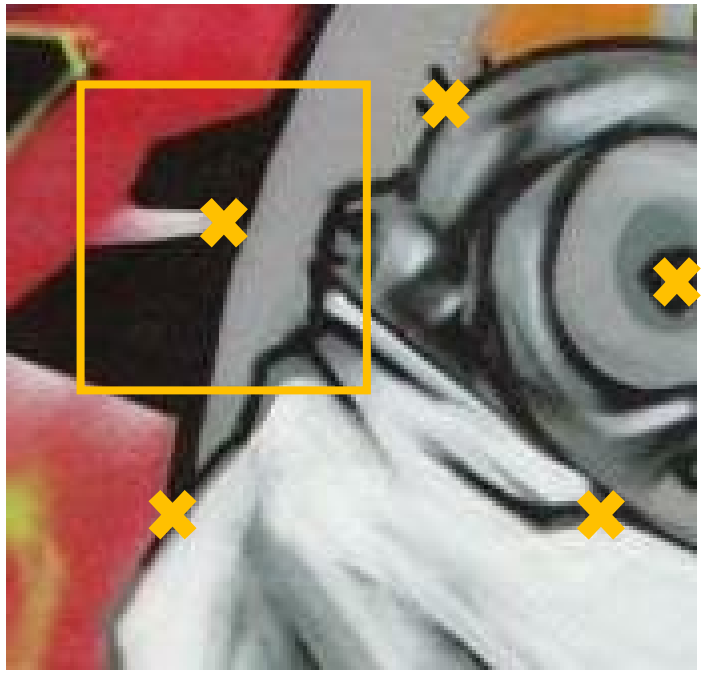
- Multi-Scale Approach – compare descriptors while varying the patch size



Slide Source:T Tuytelaars

First Trial - Exhaustive Search

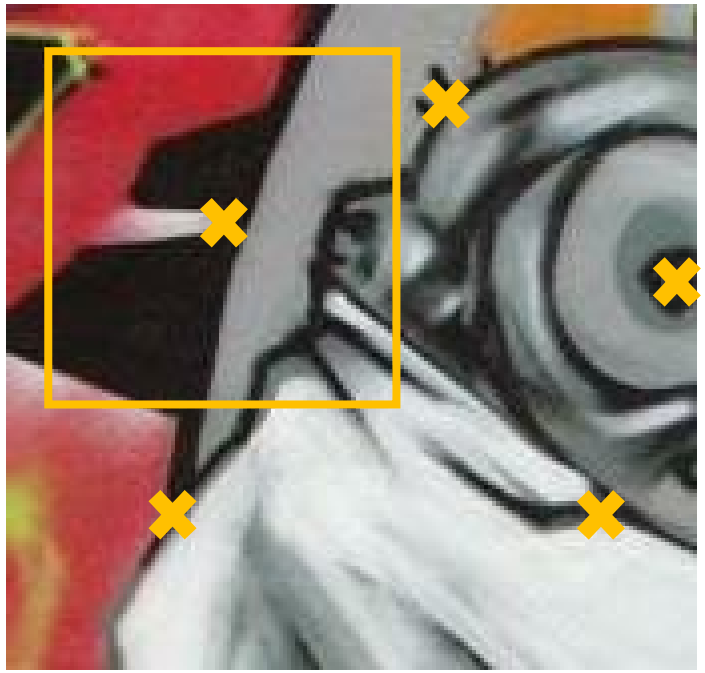
- Multi-Scale Approach – compare descriptors while varying the patch size



Slide Source:T Tuytelaars

First Trial - Exhaustive Search

- Multi-Scale Approach – compare descriptors while varying the patch size



Slide Source:T Tuytelaars

First Trial - Exhaustive Search

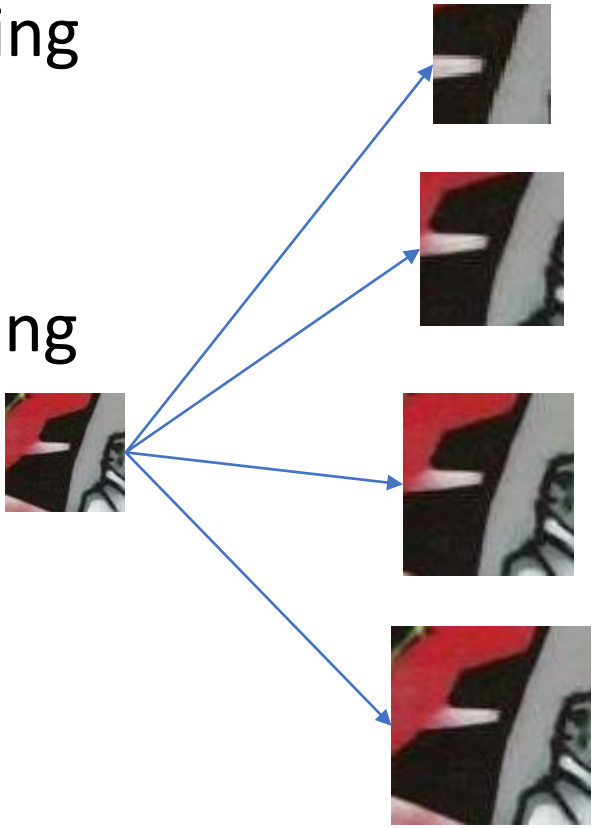
- Comparing descriptors while varying the patch size

 - Computationally inefficient

 - Inefficient but possible for matching

 - Prohibitive for retrieval in large databases

 - Prohibitive for recognition



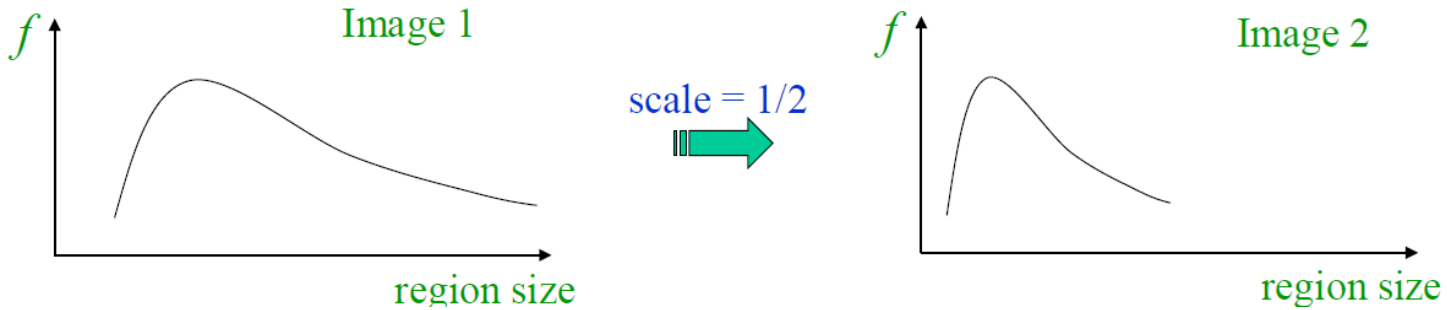
Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Solution
  - Design a function on the region, which is "scale invariant" the same for

Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

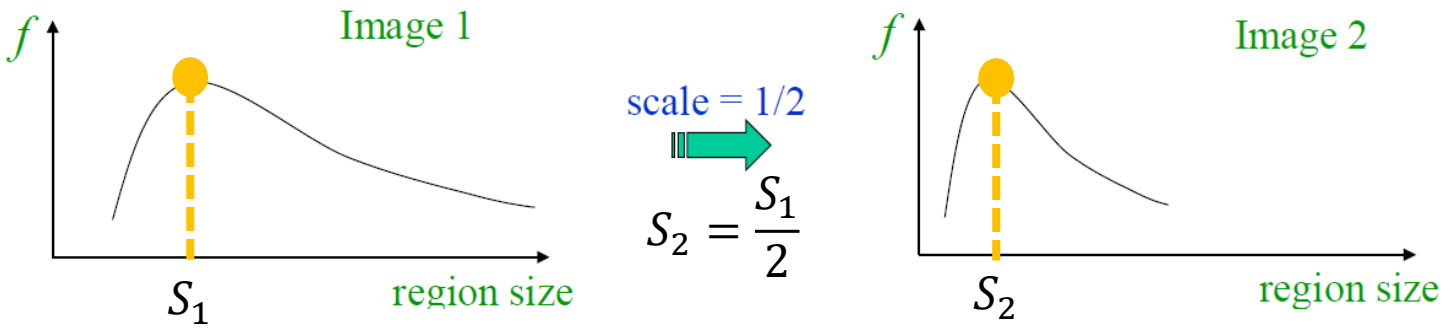
  - For a point in one image, we can consider it as a function of region size (patch



Slide Credit: Kristen Grauman

Automatic Scale Selection

- Common approach
  - Take a local maximum of this function
  - Observation: region size for which the maximum is achieved should be *invariant* to image scale.
  - For a point in one image, we can consider it as a function of region size (patch width)

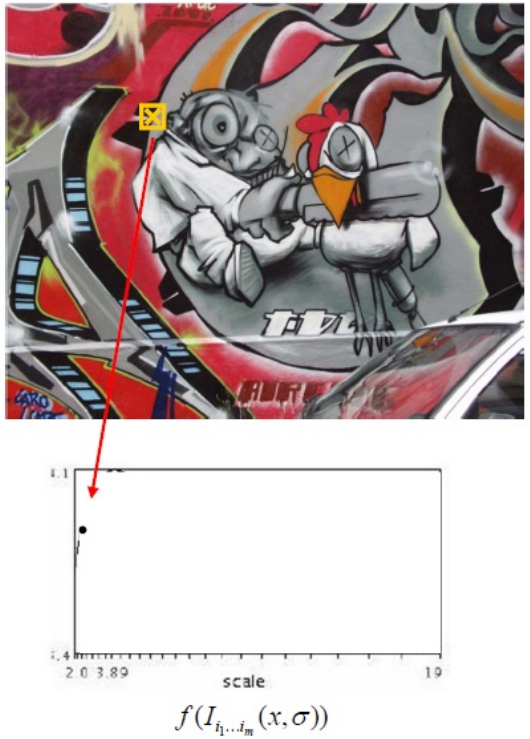


Important: this scale invariant region size is found in each image independently!

Slide Credit: Kristen Grauman

Automatic Scale Selection

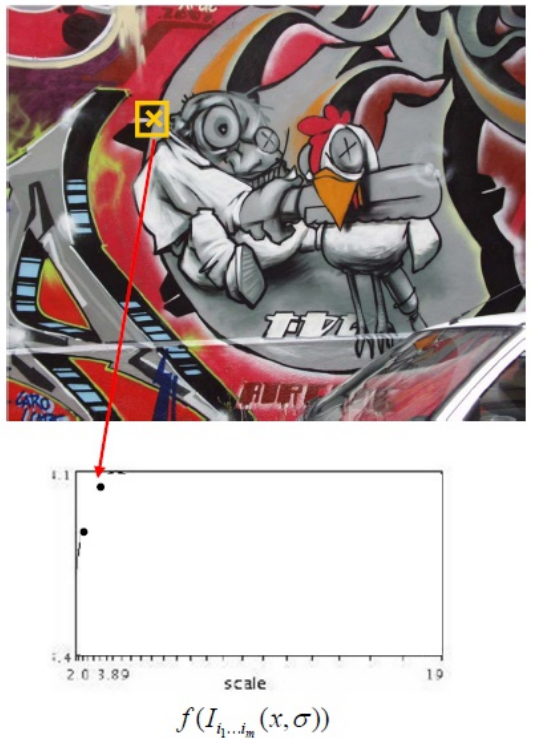
- Function responses for increasing scale (scale signature)



Slide Credit: Kristen Grauman

Automatic Scale Selection

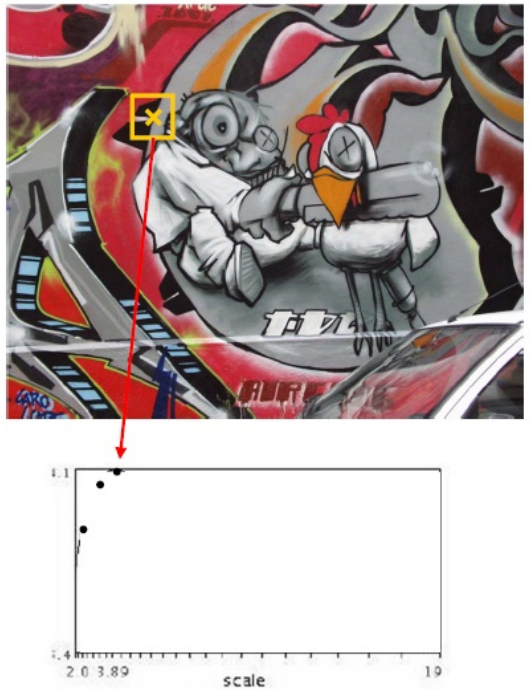
- Function responses for increasing scale (scale signature)



Slide Credit: Kristen Grauman

Automatic Scale Selection

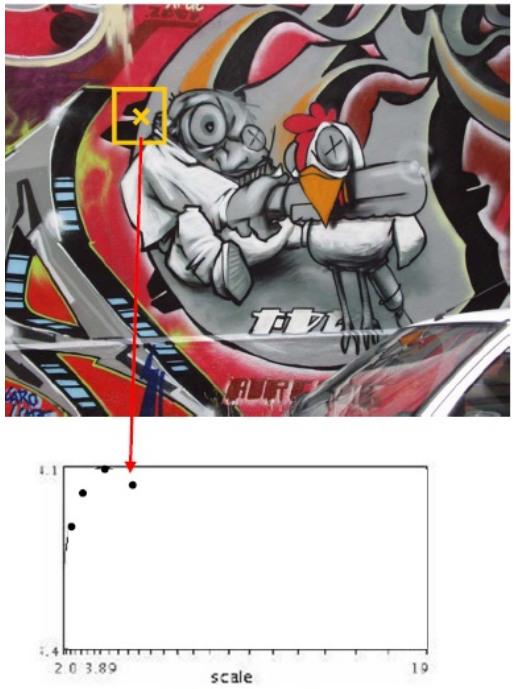
- Function responses for increasing scale (scale signature)



Slide Credit: Kristen Grauman

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



Slide Credit: Kristen Grauman

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

$f(I_{i_1 \ldots i_m}(x,\sigma))$

$f(I_{i_1 \ldots i_m}(x',\sigma'))$

Slide Credit: Kristen Grauman

Automatic scale selection

- Normalize: rescale to fixed size

How should we choose the function for the scale selection?

How should we choose the function?

- Several derivative based functions F can be used to compute a scale representation of an image.
- Criteria:
  - Rotation invariant
  - Illumination invariant is less critical as we are looking for extrema.

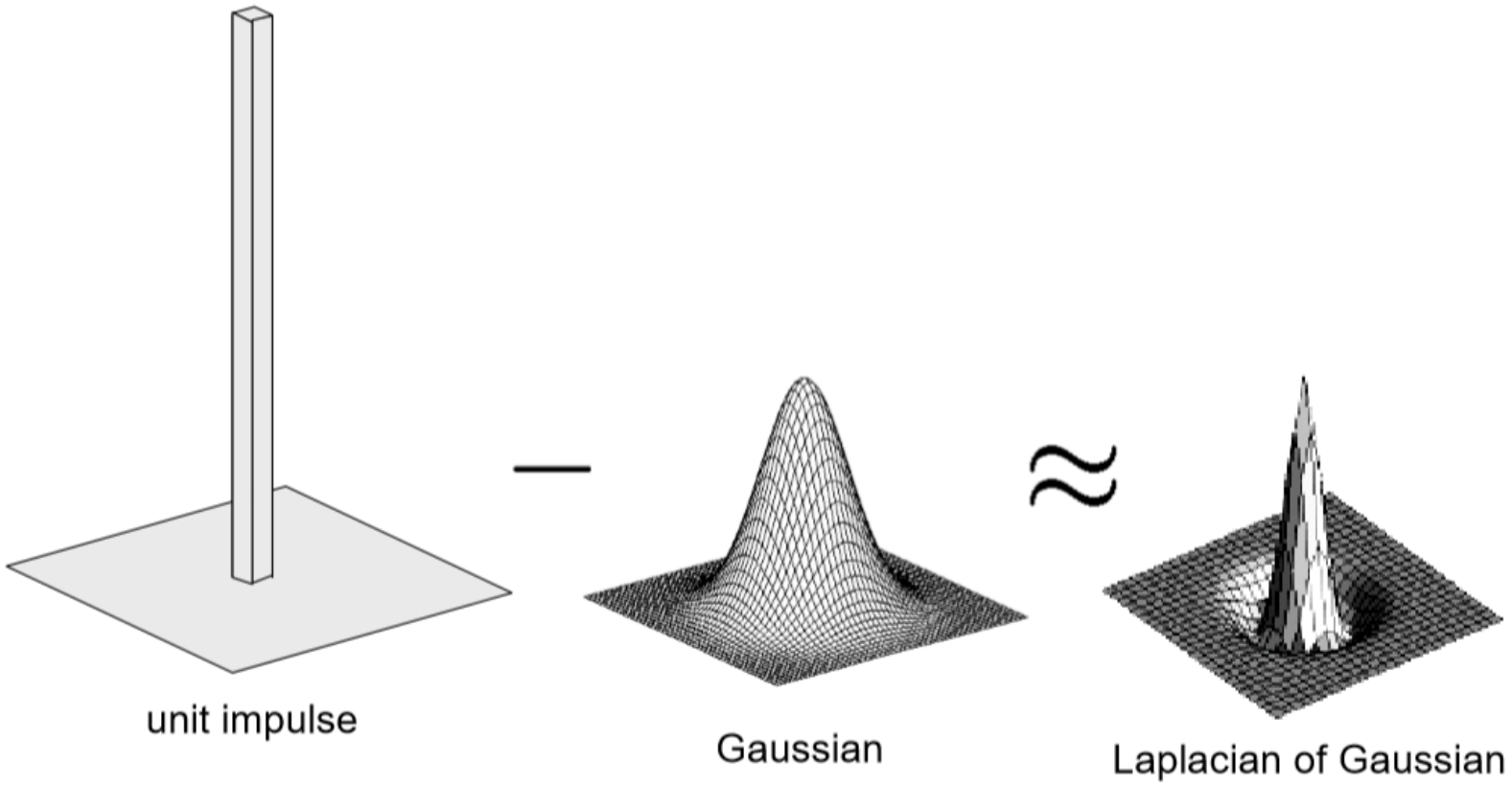
- 1) Square Gradient $s^2\left(L_x^2(\mathbf{x},s)+L_y^2(\mathbf{x},s)\right)$
- 2) Laplacian $\left|s^2(L_{xx}(\mathbf{x},s)+L_{yy}(\mathbf{x},s))\right|$
- 3) Different of Gaussian $|I(\mathbf{x})*G(s_{n-1})-I(\mathbf{x})*G(s_n)|$
- 4) Harris Function $\det(\mathbf{C})-\alpha\mathrm{trace}^2(\mathbf{C})$

Best Result

Second Best Result

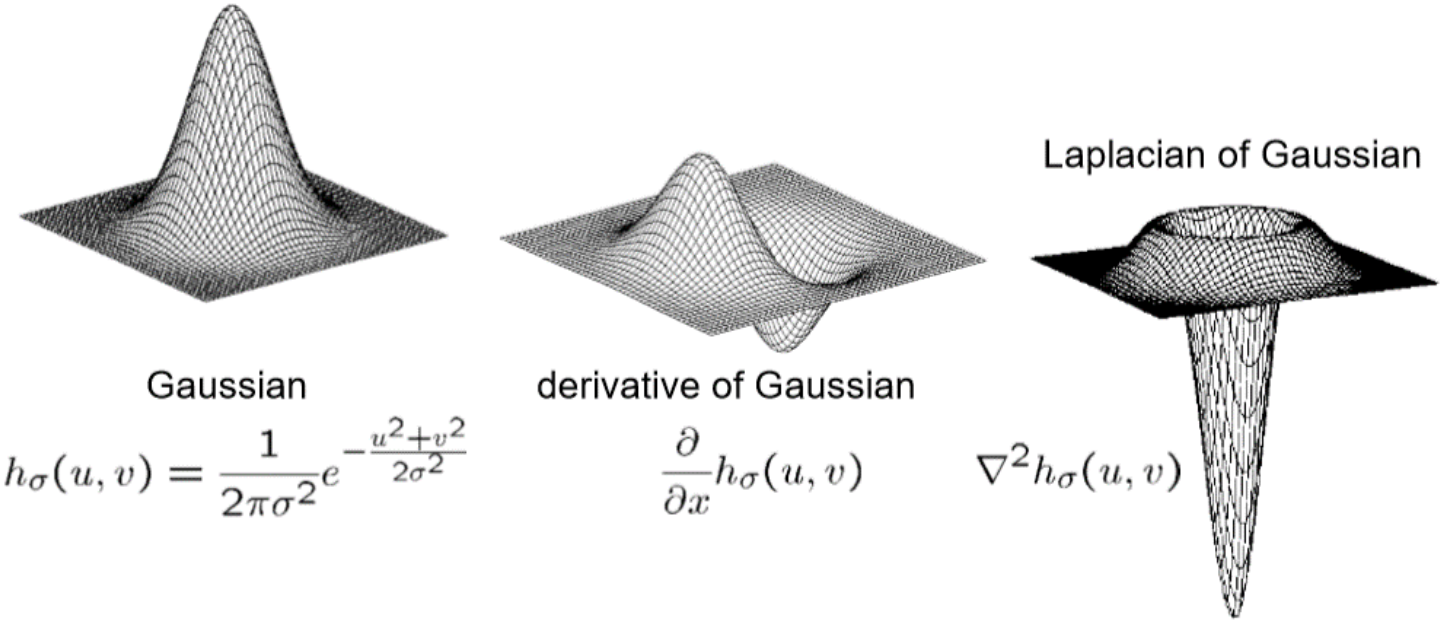
C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors", *International Journal of Computer Vision,* 37(2), pp. 151-172, 2000.

Laplacian of Gaussian



Source: Lazebnik

Recall - 2D Edge Detection Filters



Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

$$\nabla^2 h_\sigma(u,v)$$

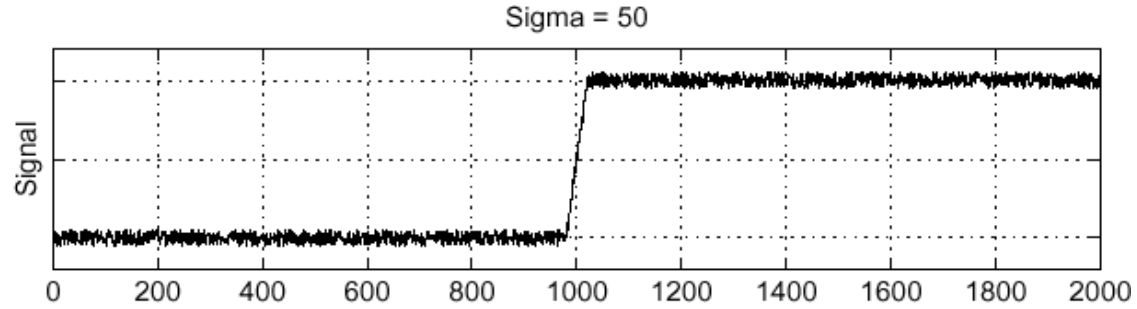
∇^2 is the **Laplacian** operator:

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

How LoG responds to blobs

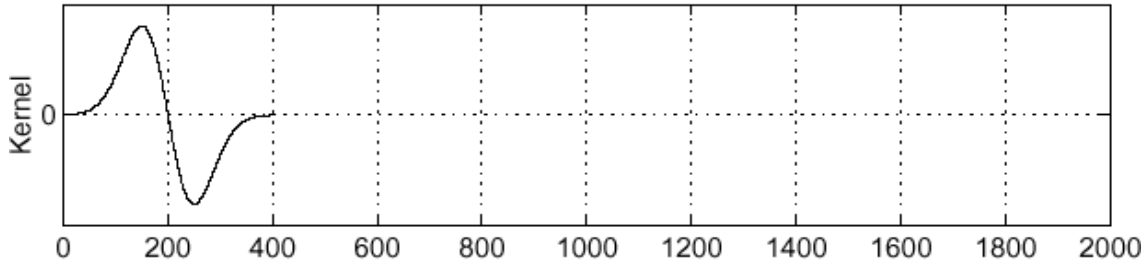
- Recall – Edge detection using 1st derivative

f



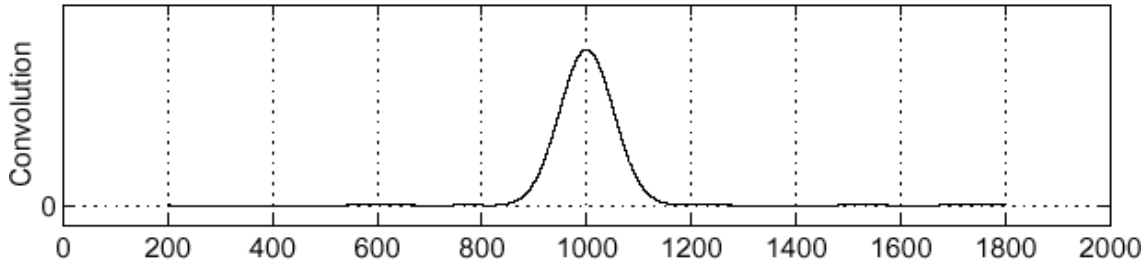
Edge

$\frac{d}{dx} g$



Derivative of Gaussian

$f * \frac{d}{dx} g$

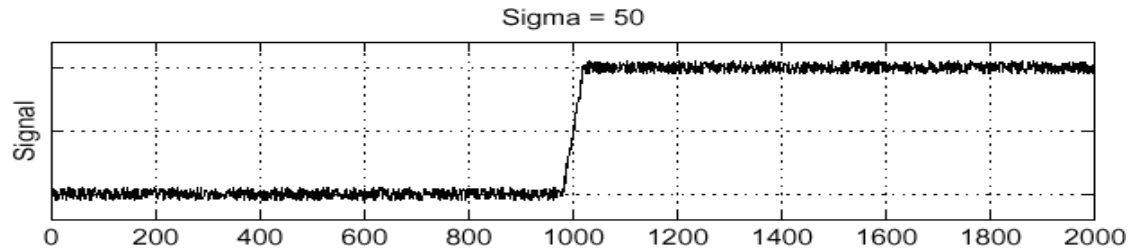


Edge = maximum of derivative

How LoG responds to blobs

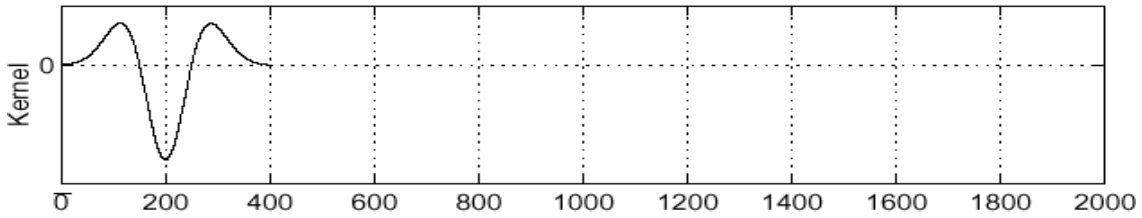
- Edge detection using 2nd derivative

f



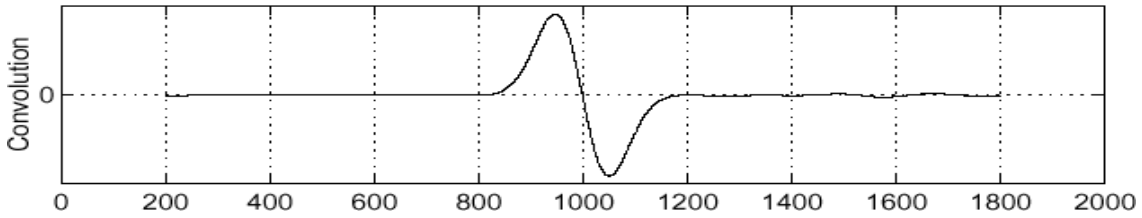
Edge

$\frac{d^2}{dx^2} g$



Second derivative of Gaussian (Laplacian)

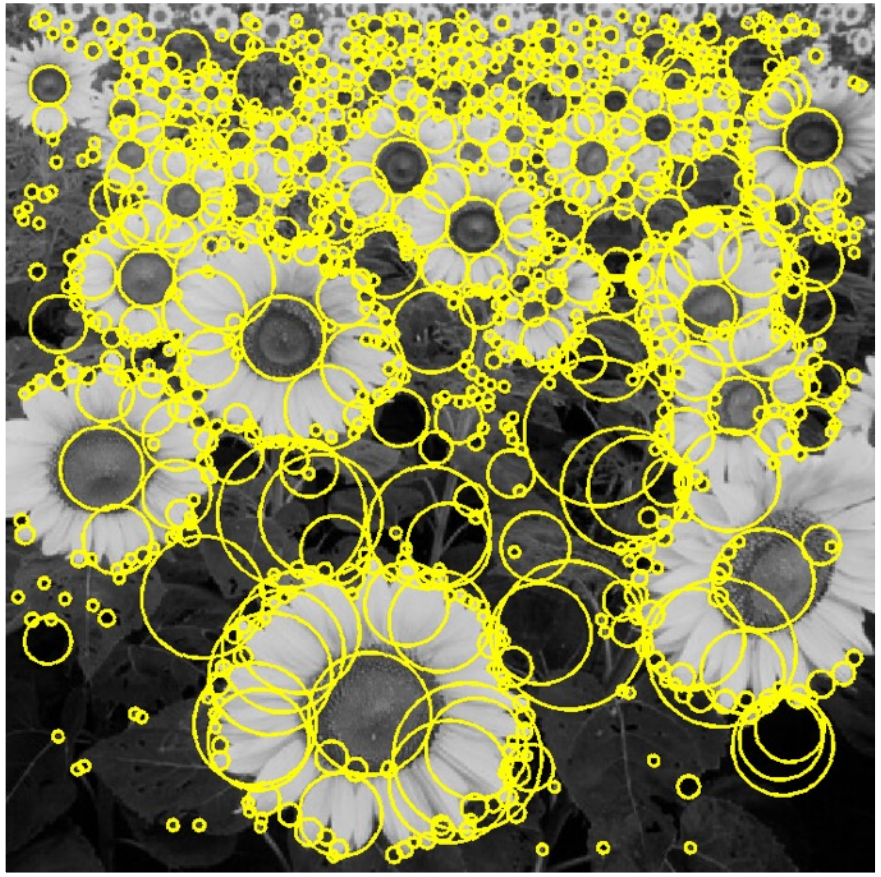
$f * \frac{d^2}{dx^2} g$



Edge = zero crossing of second derivative

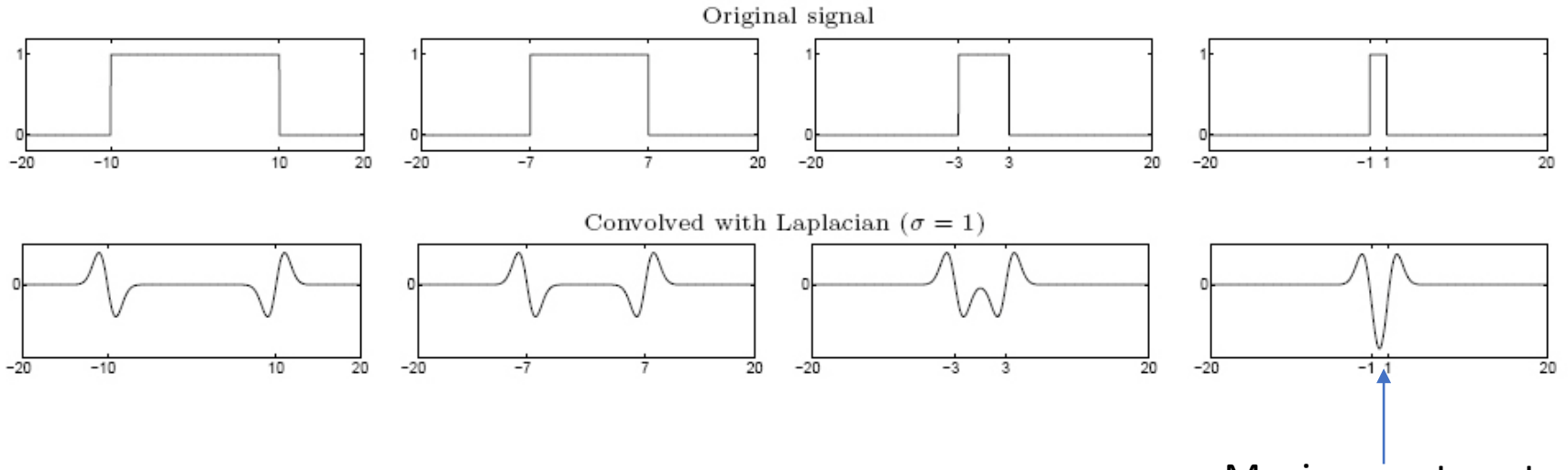
Blob Detection

- “Peak” or “Valley” in intensity



Blob consists of two edges

- Blob Detection



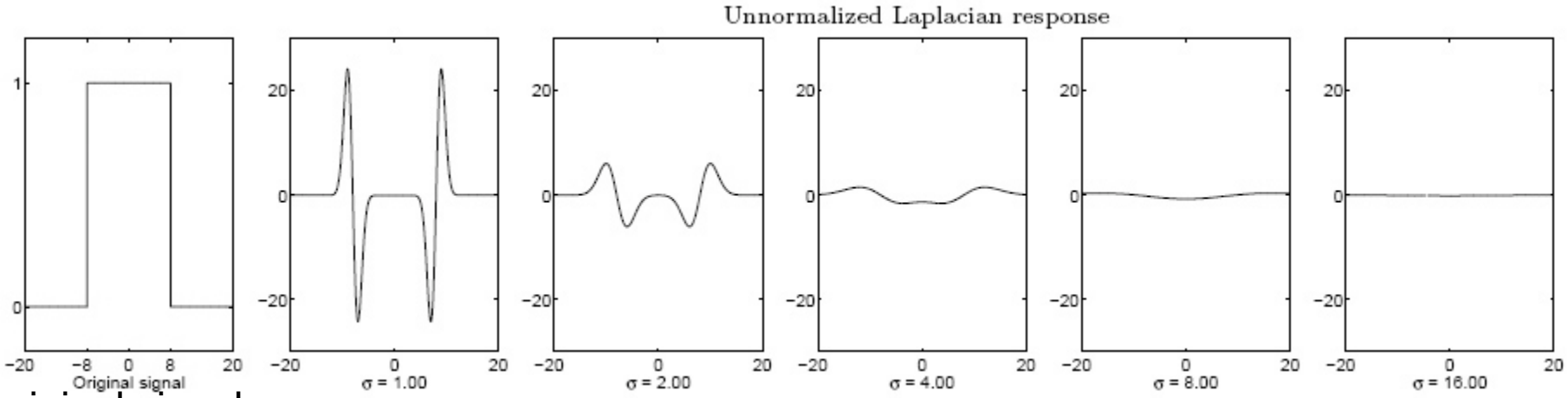
Blob Detection is done by convolving the image with Laplacian filter at different scales

Look for the maximum response.

Slide: S. J Bedros

How to find the spatial extent of blob using LOG?

- Idea: Find the characteristic scale of the blob by convolving it with Laplacian filters at several scales and looking for the maximum response.



original signal (radius=8)

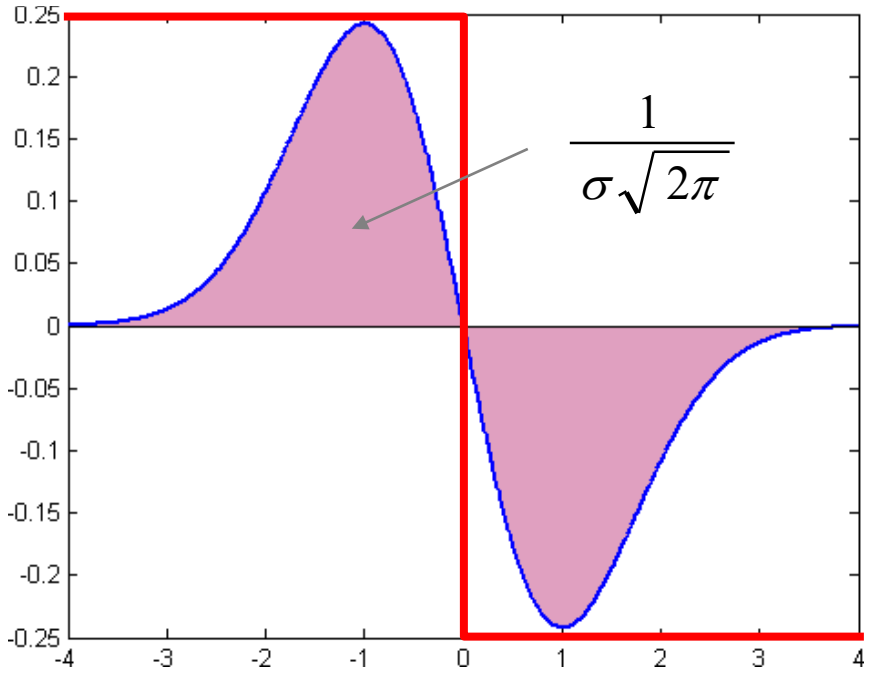
increasing σ →

Why does this happen?

Slide: S. J Bedros

Scale Normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases



Slide: S. J Bedros

Scale Normalization

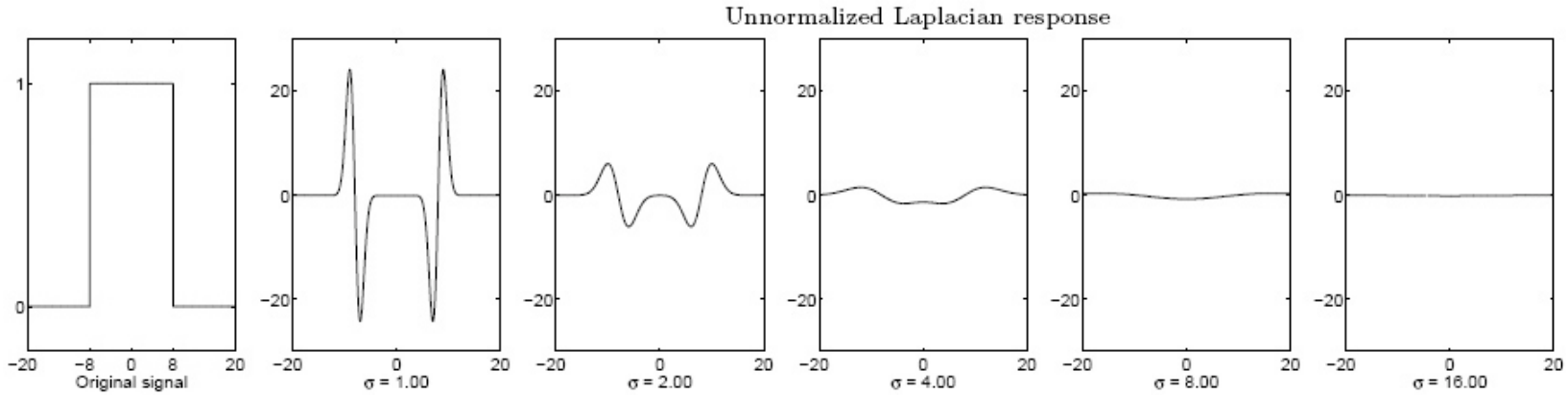
- To keep response the same (scale- invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

Slide: S. J Bedros

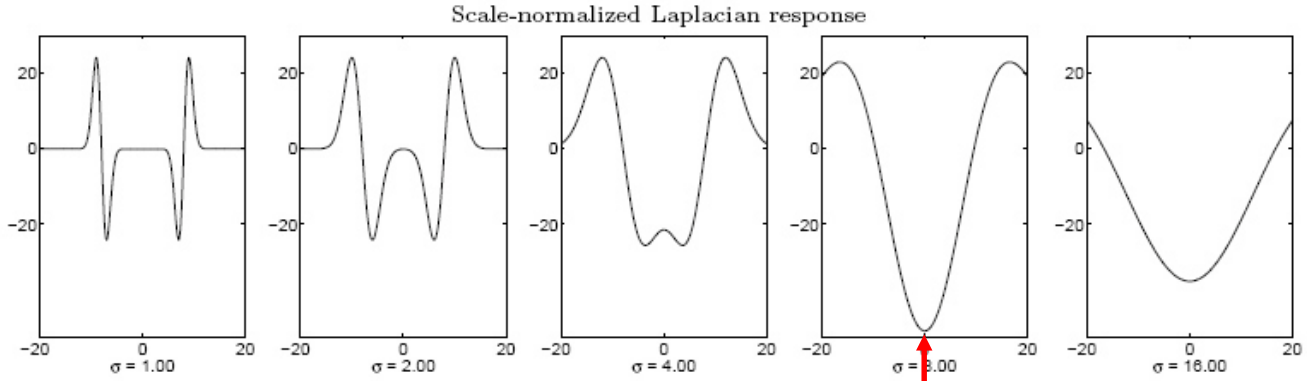
Effect of scale normalization

Original signal

Unnormalized Laplacian response



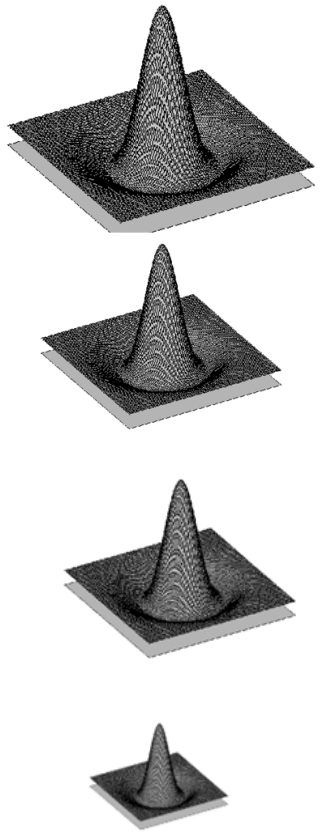
Scale-normalized Laplacian response (x σ^2)

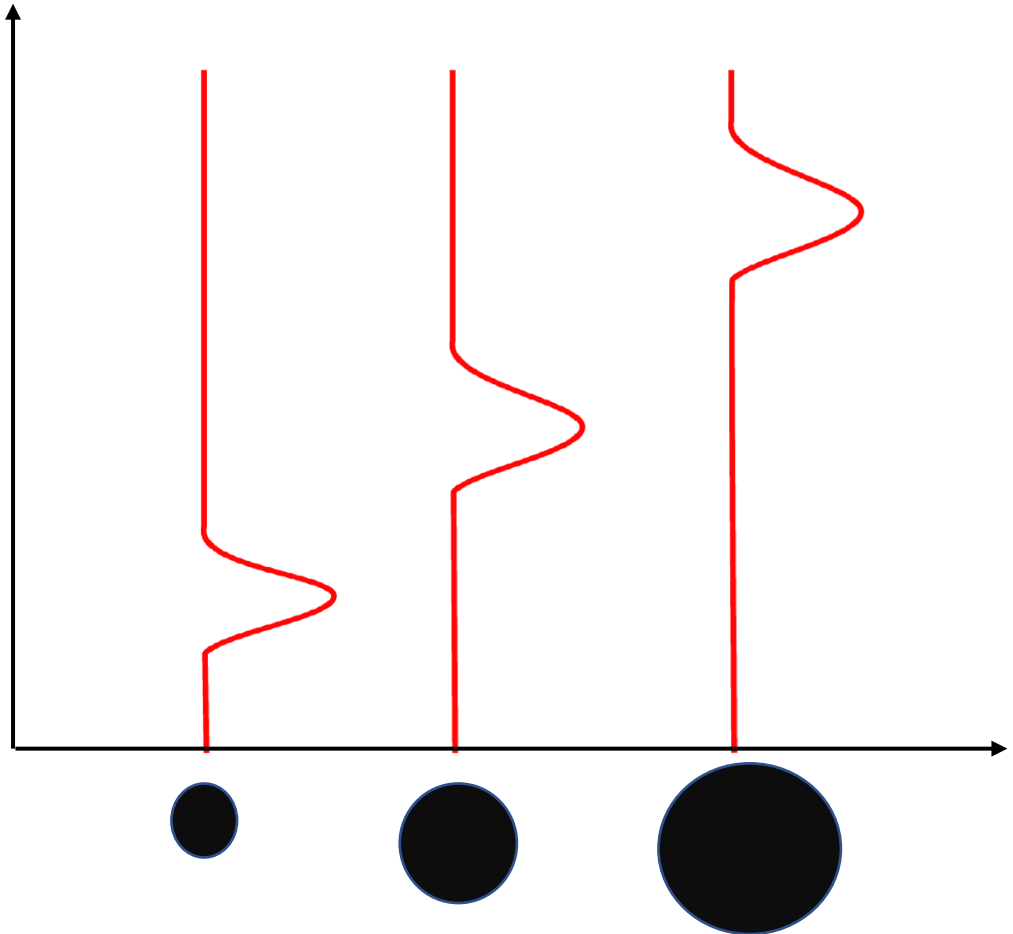


Slide: S. J Bedros

What A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector

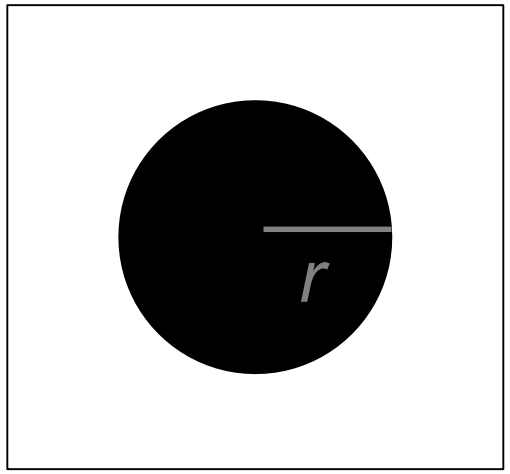




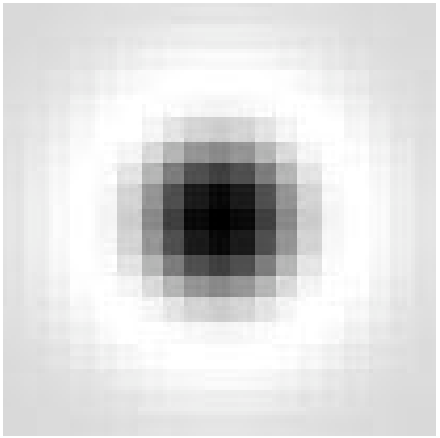
Slide source: Bastian Leibe

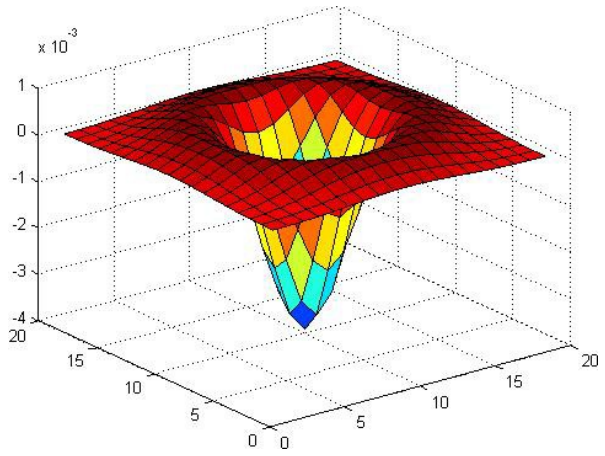
Scale Selection

- At what scale does the Laplacian achieve a maximum response for circle with radius R?



image



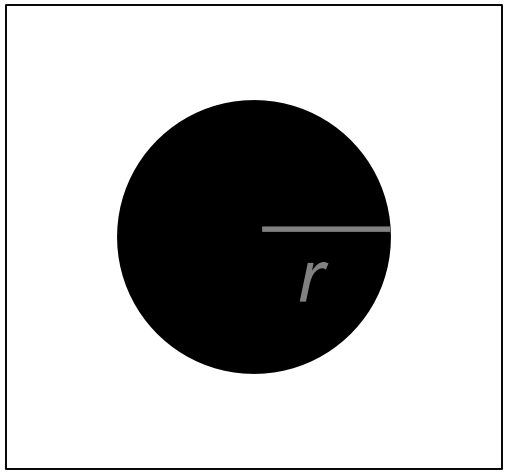


Laplacian

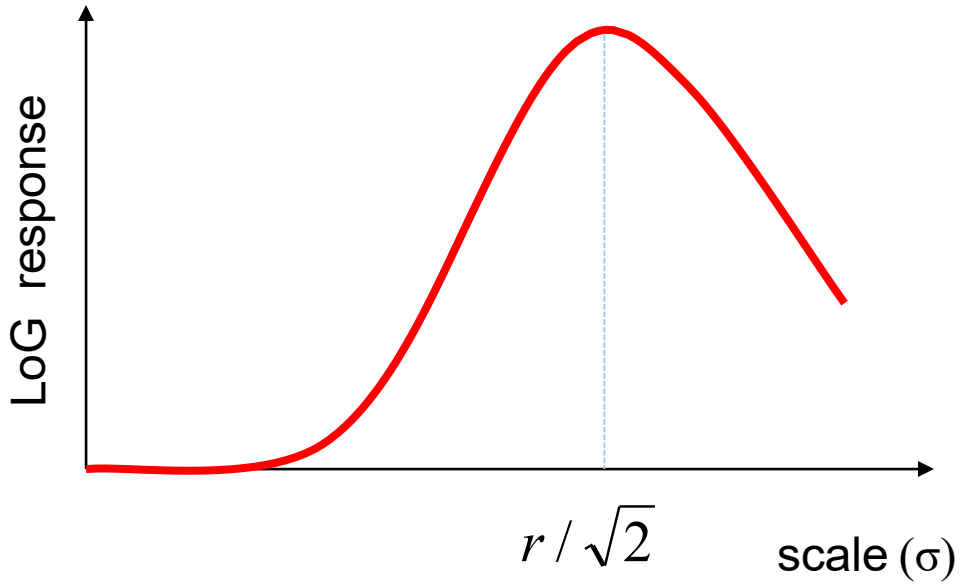
Slide: S. J Bedros

Scale Selection:

- At what scale does the Laplacian achieve a maximum response for circle with radius R?
- LoG is t maximium at $\sigma = r/\sqrt{2}$



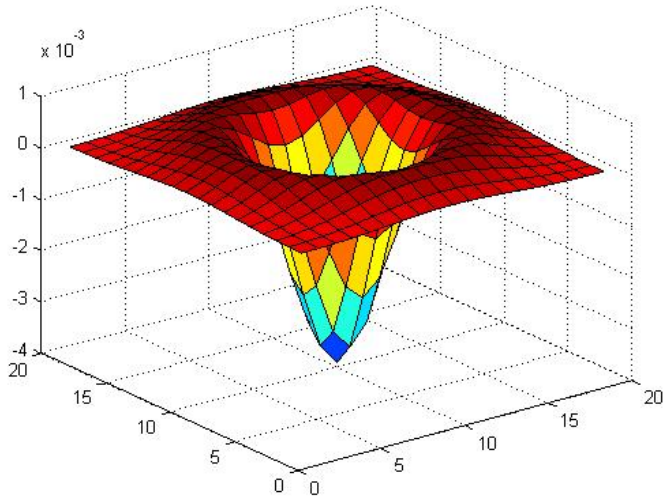
image



T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

Slide: S. J Bedros

Example



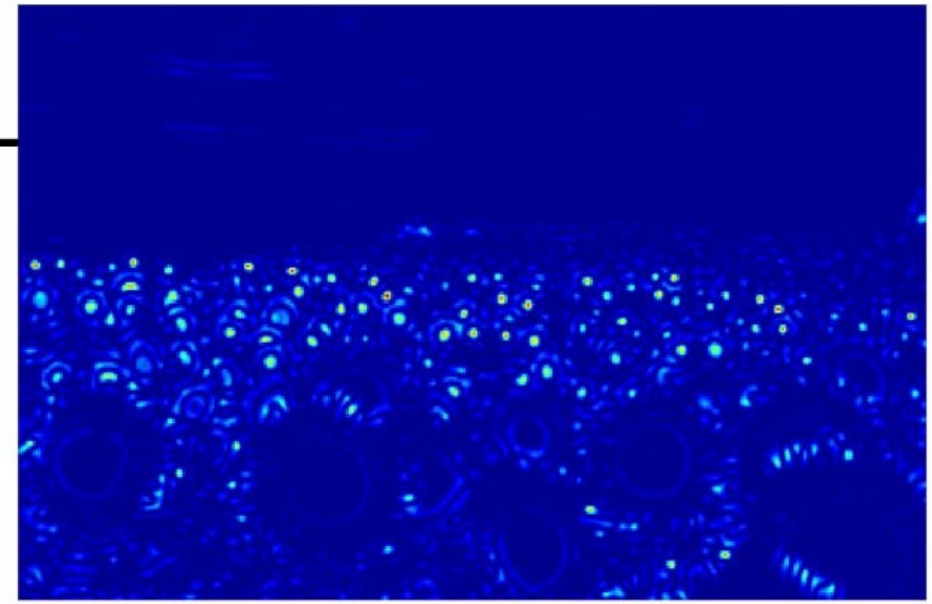
LoG Detector

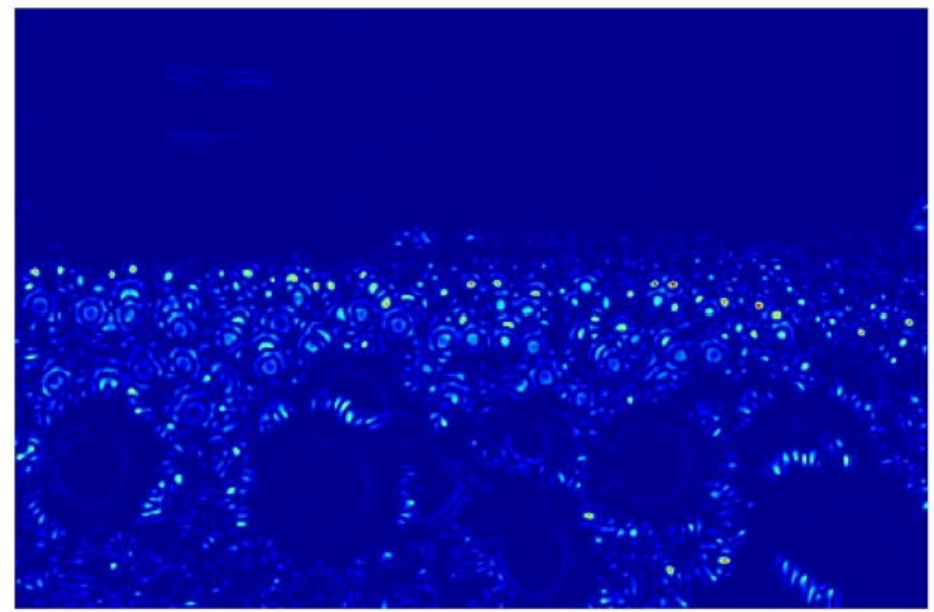
Image at ¾ of original Size

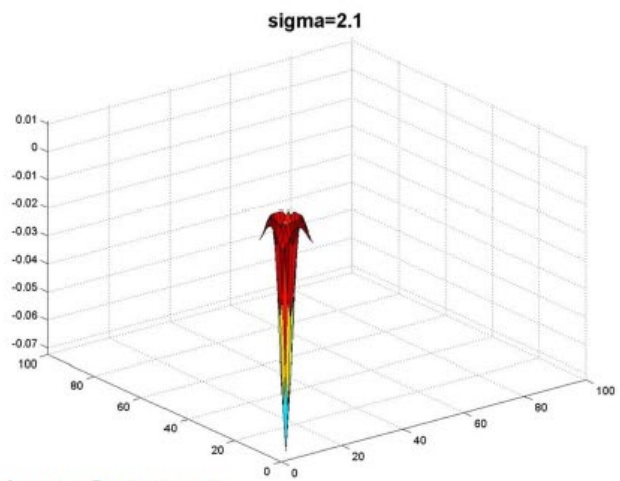
Original Image

Source: Kristen Grauman

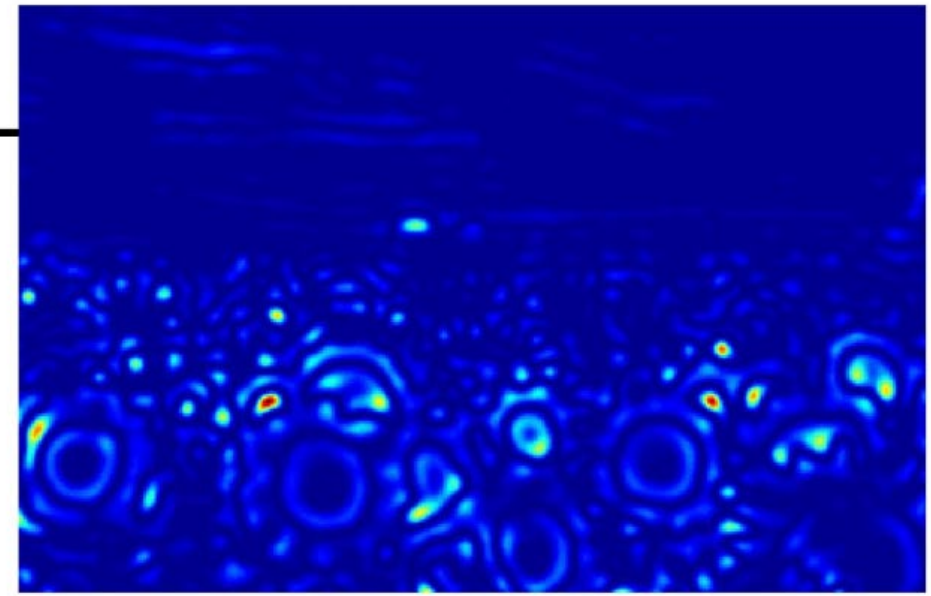
Original image
at ¾ the size

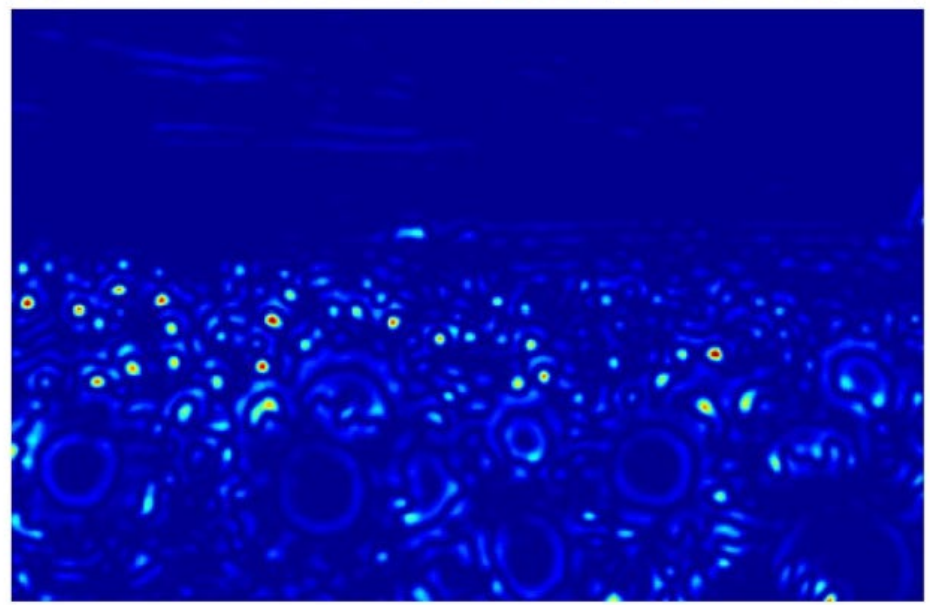


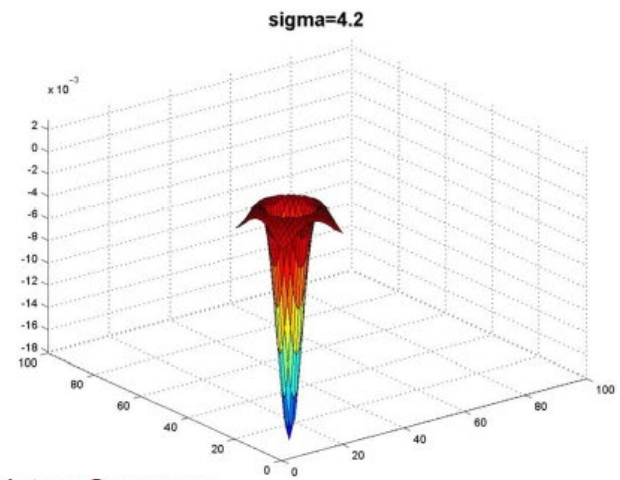




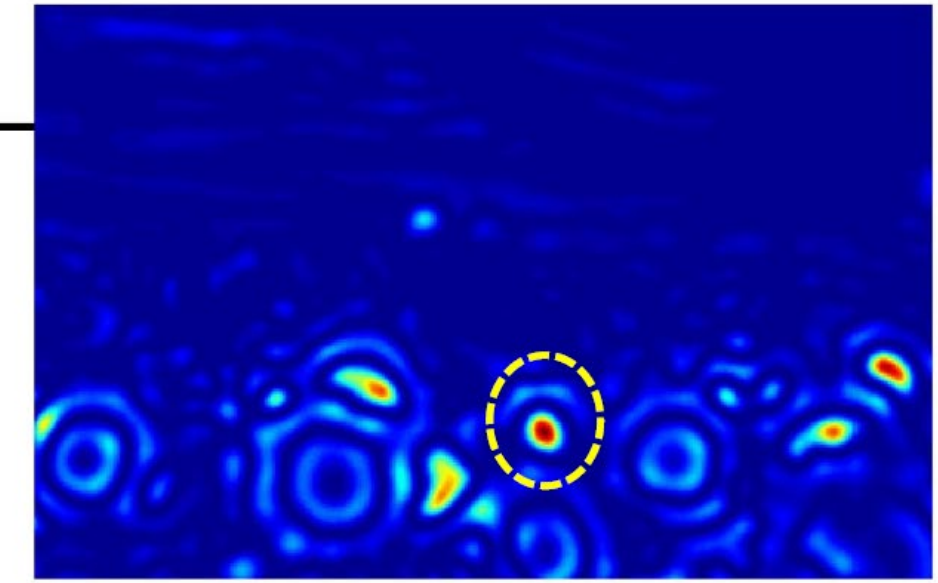
Kristen Grauman

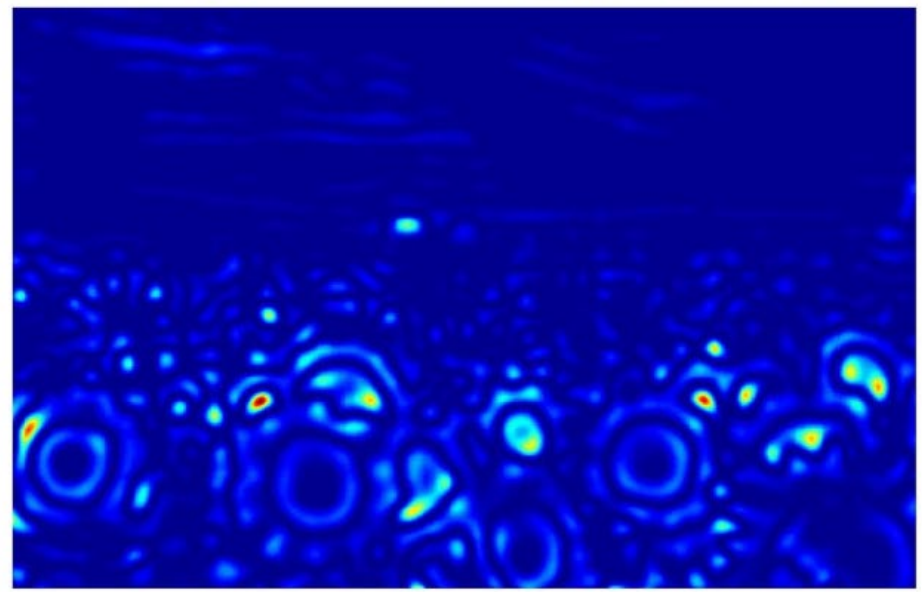


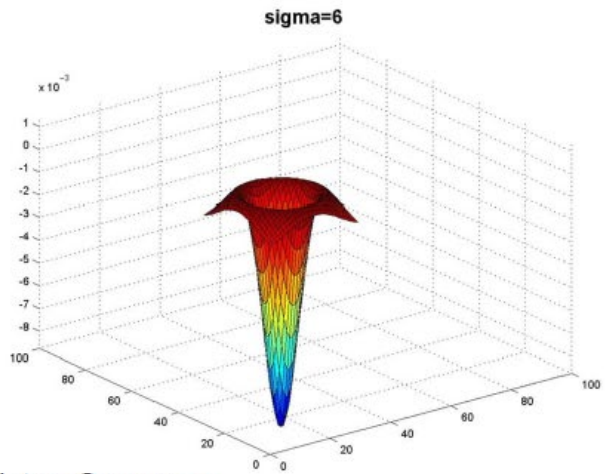




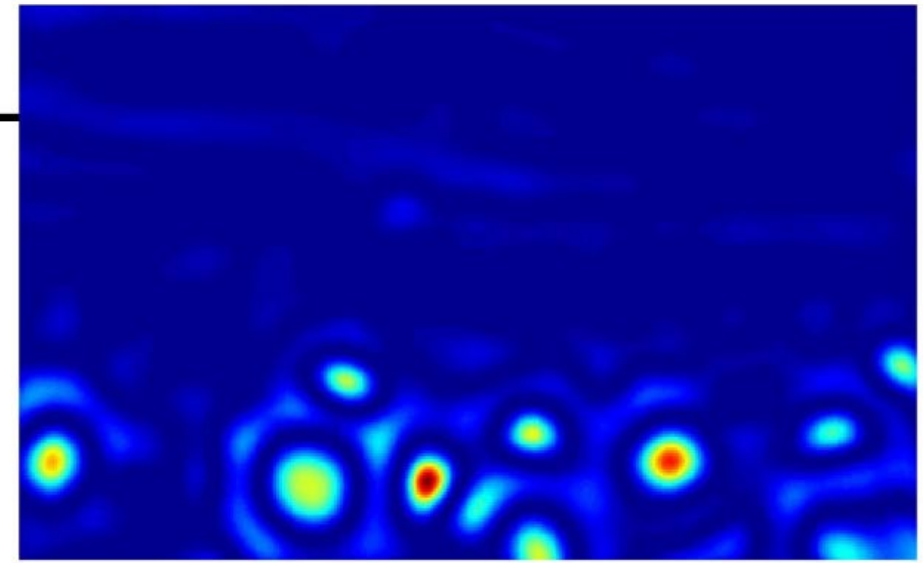
Kristen Grauman

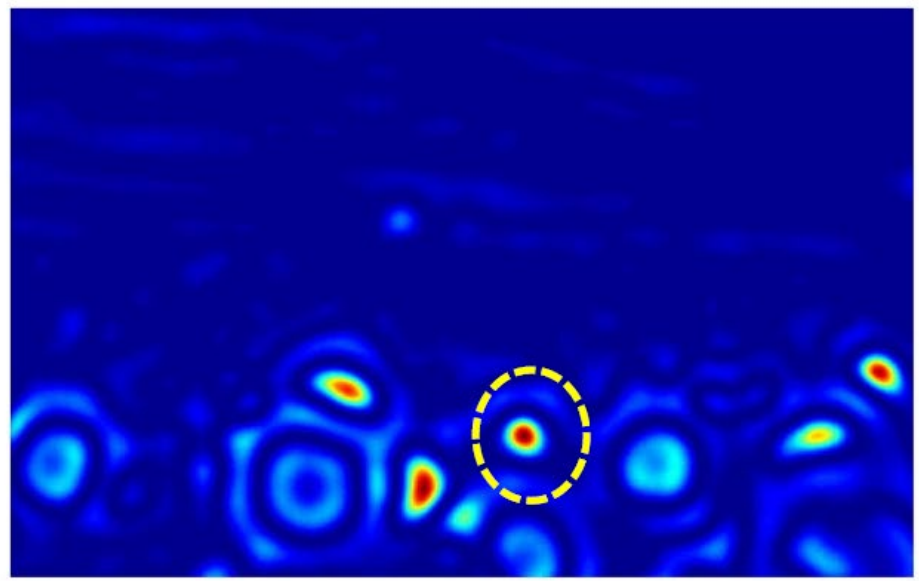


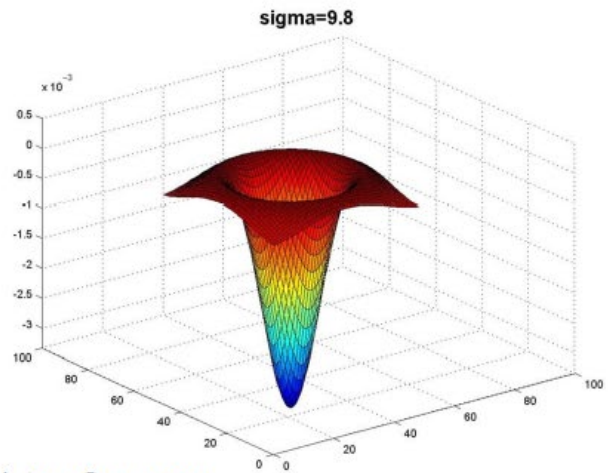




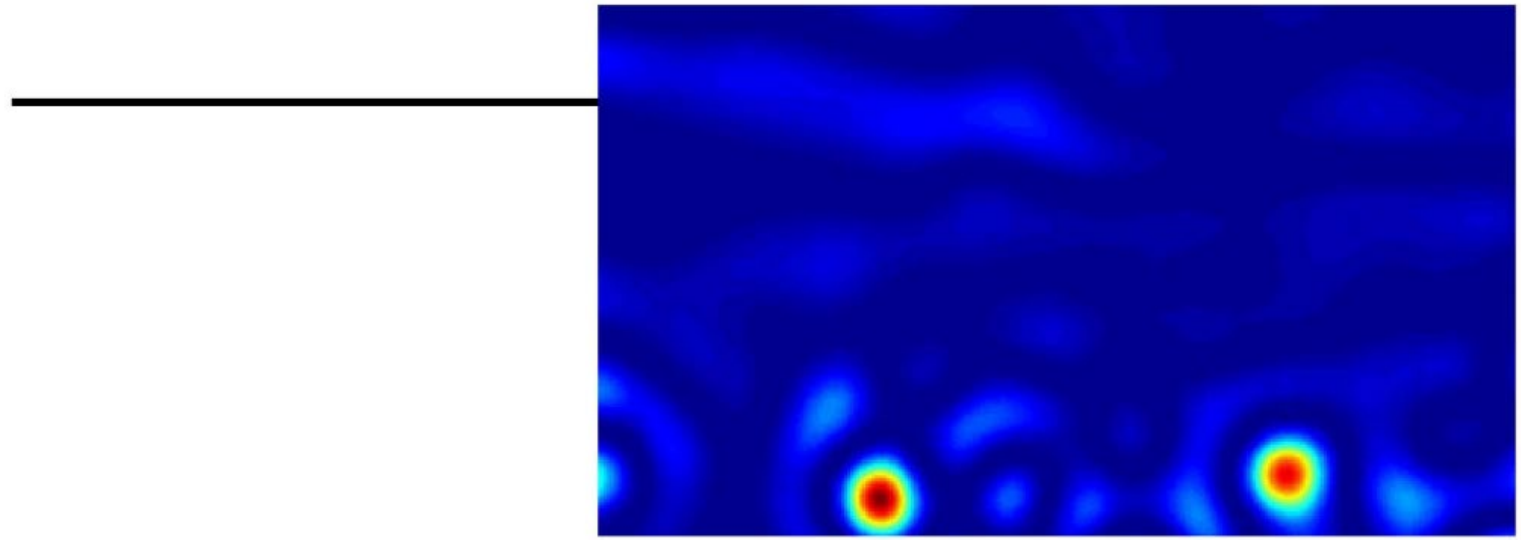
Kristen Grauman

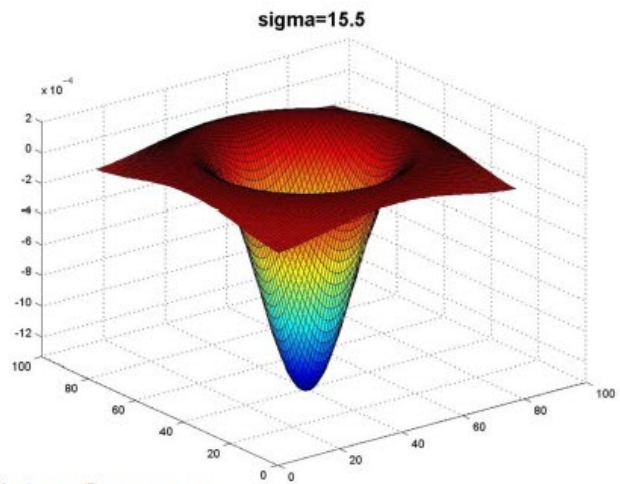


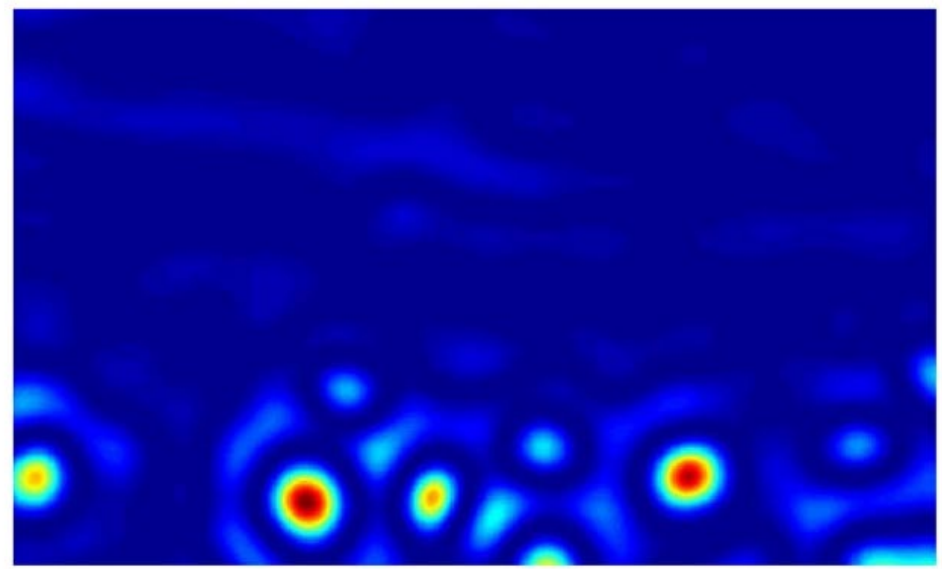




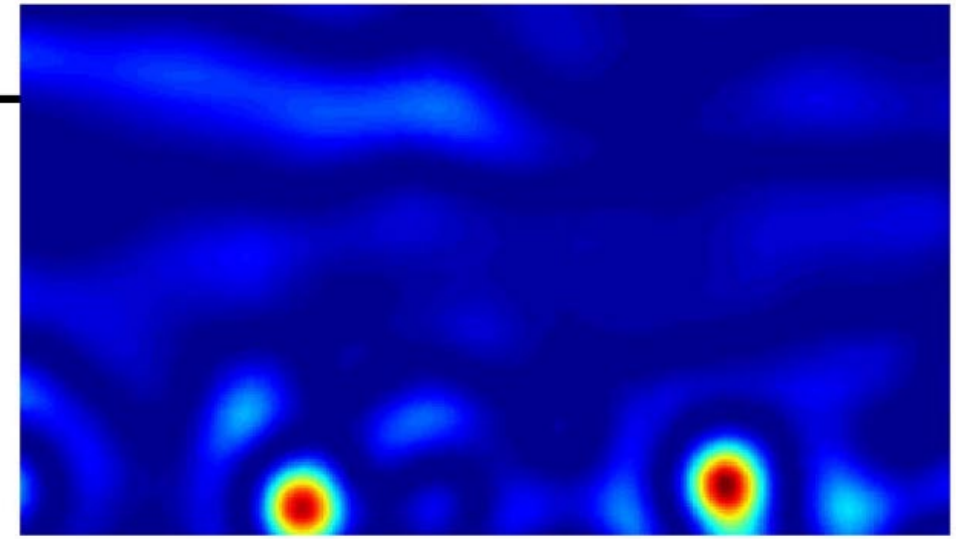
Kristen Grauman

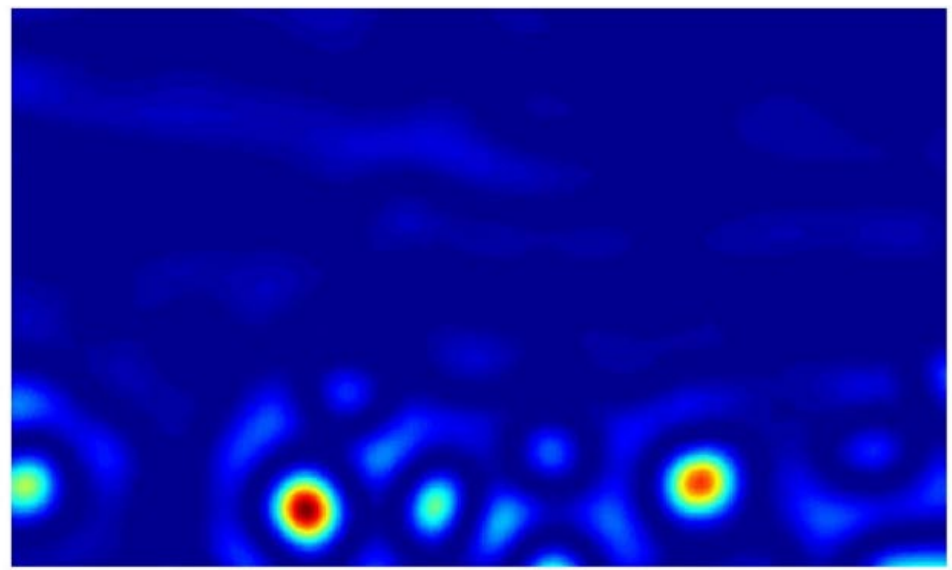


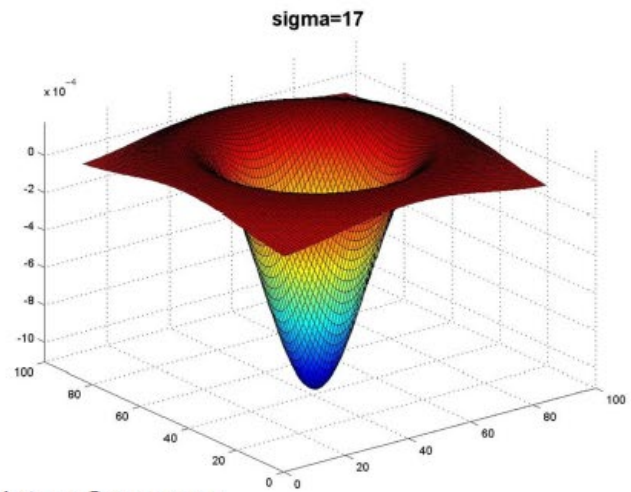




Kristen Grauman



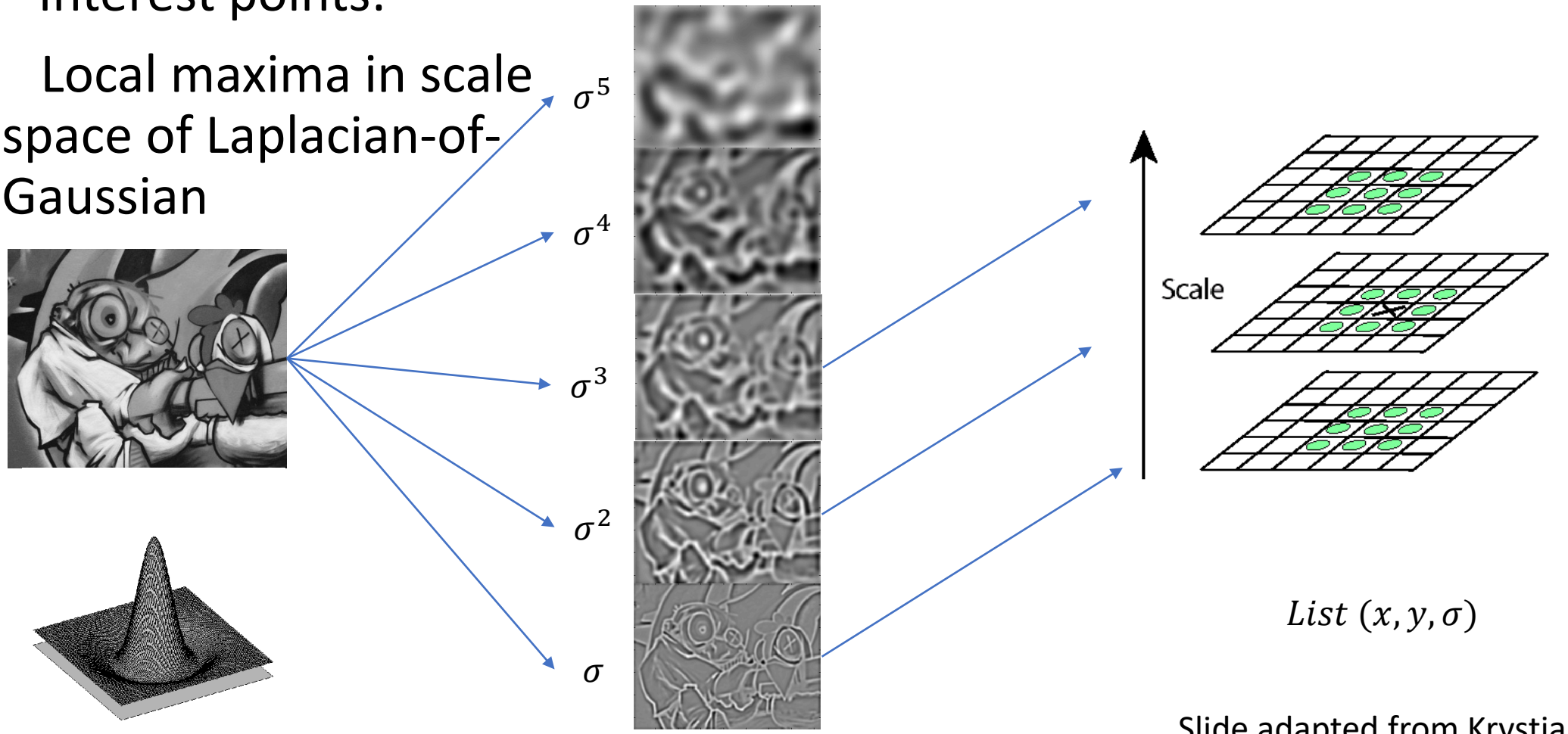



sigma=17
x 10^-4
0
-2
-4
-6
-8
-10
100
80
60
40
20
0
0
20
40
60
80
100

Laplacian-of-Gaussian (LoG)

- Interest points:

  Local maxima in scale space of Laplacian-of-Gaussian

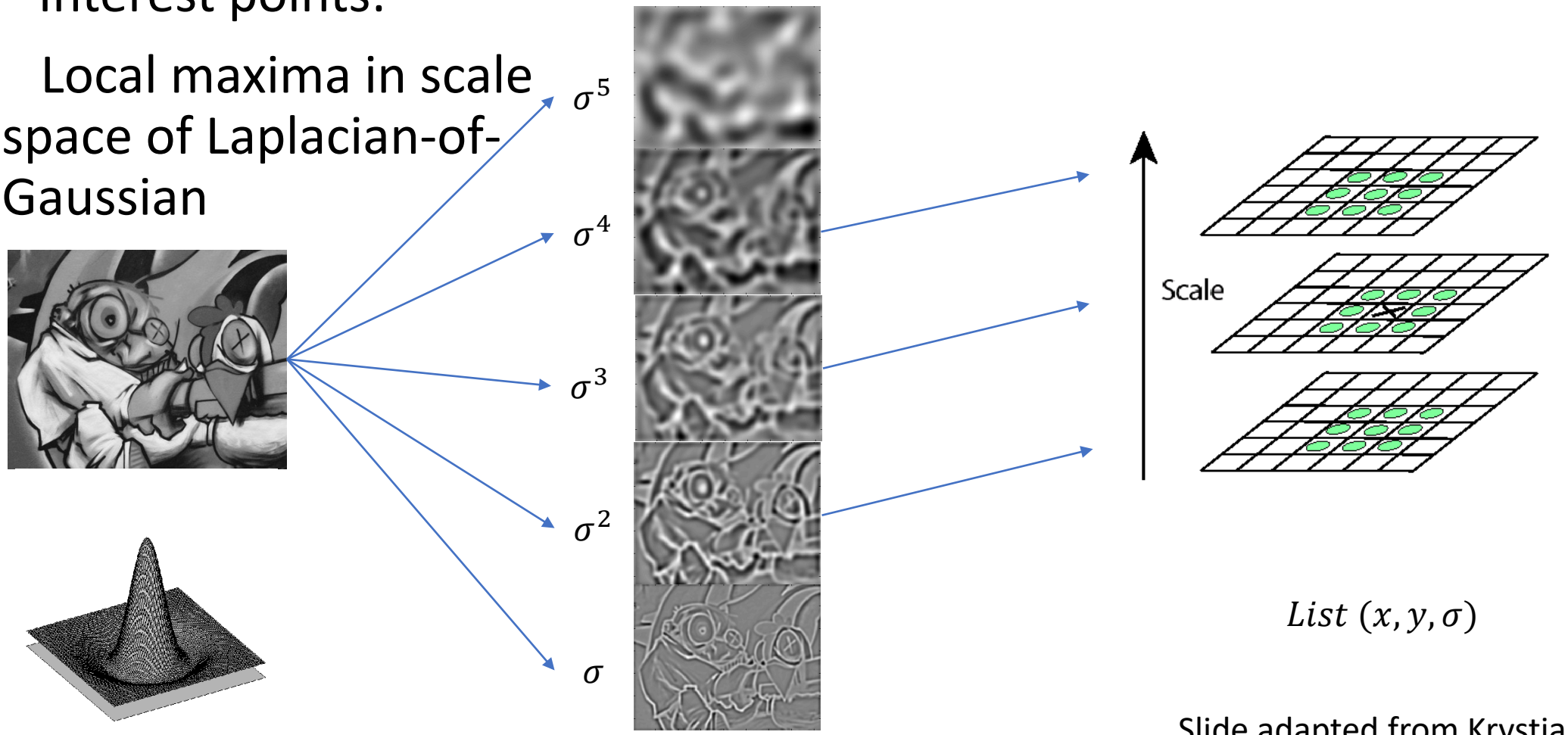


Slide adapted from Krystian Mikolajczyk

Laplacian-of-Gaussian (LoG)

- Interest points:

  Local maxima in scale space of Laplacian-of-Gaussian

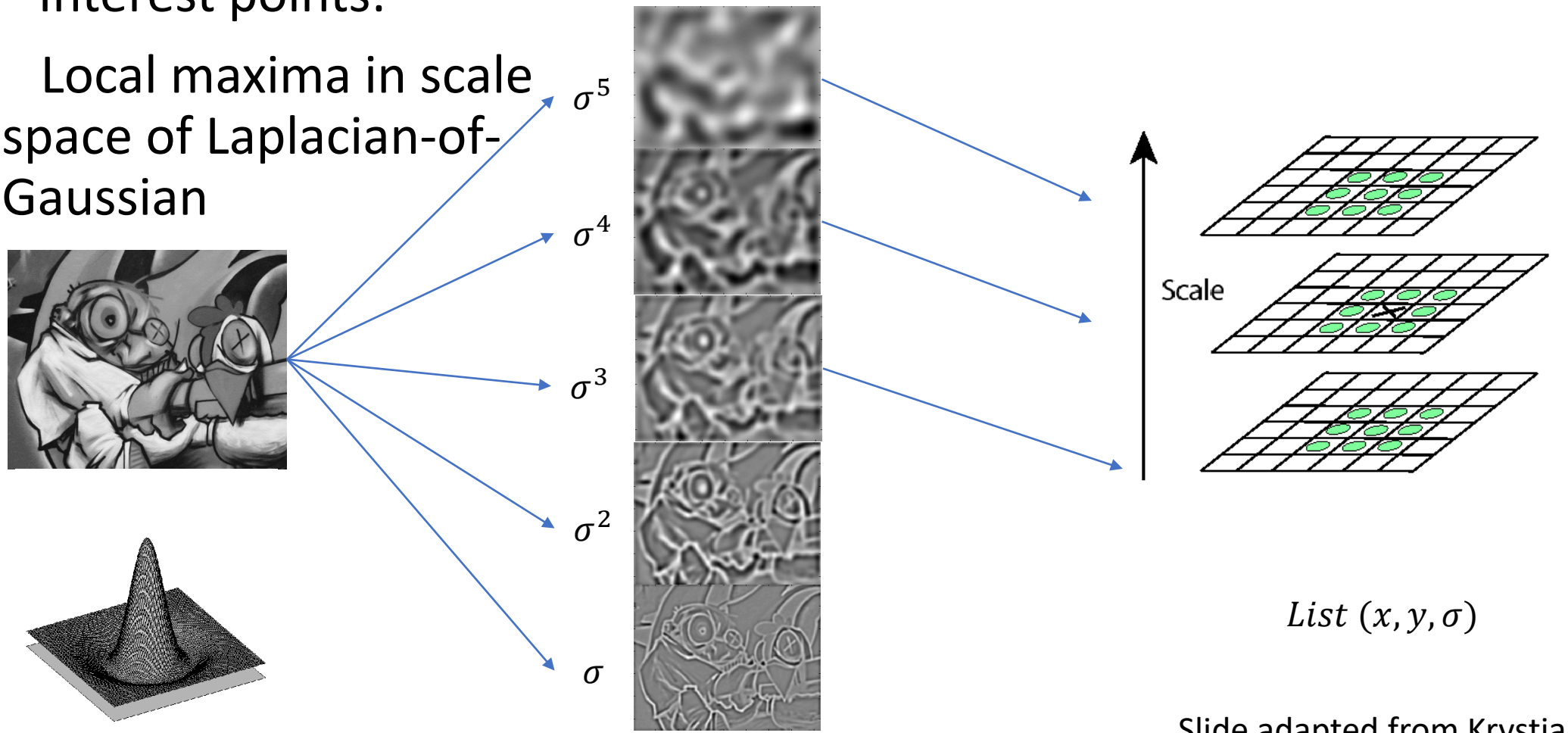


Slide adapted from Krystian Mikolajczyk

Laplacian-of-Gaussian (LoG)

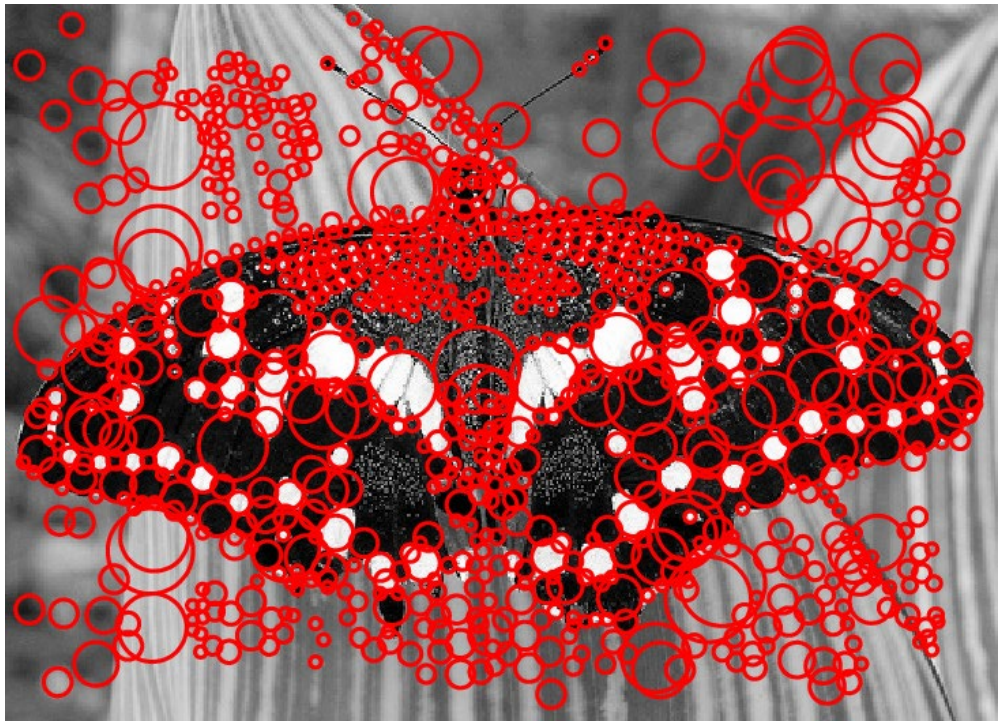
- Interest points:

  Local maxima in scale space of Laplacian-of-Gaussian



Slide adapted from Krystian Mikolajczyk

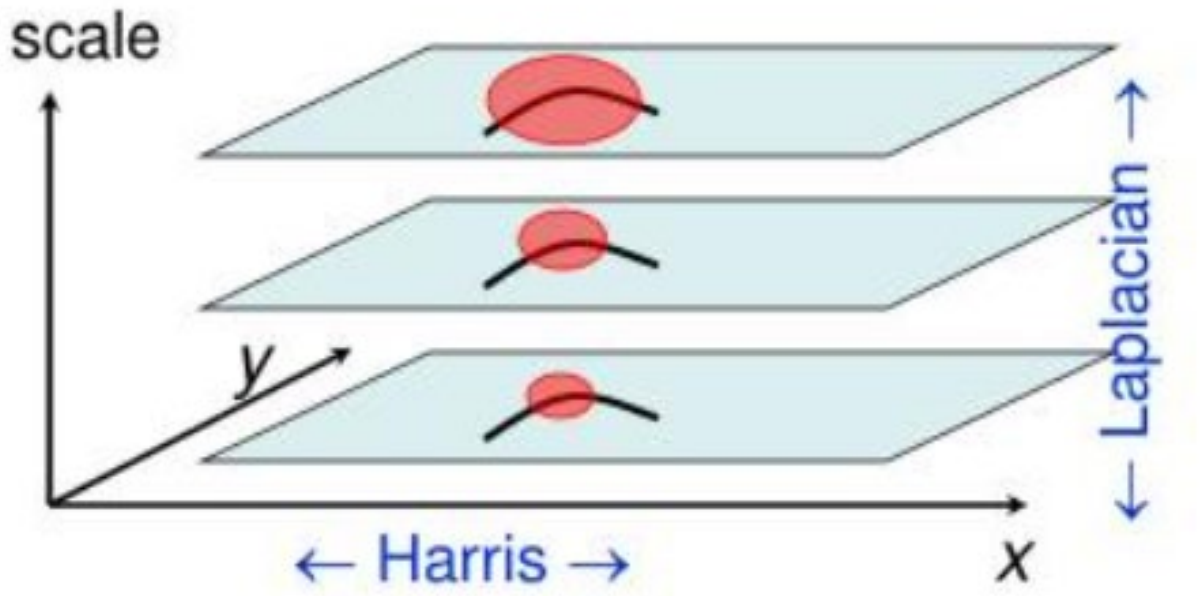
LoG Detector with different scales



Source: Lana Lazebni

Summary: Scale-invariant Region Selection

- Methodology:
- Find local maximum of
  - Harris corner detector in space
  - Laplacian in scale
- Known as ***Harris Laplace Detect***

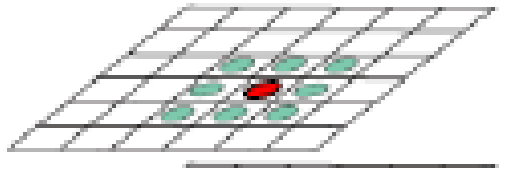


Harris-Laplace Detector

(1) Find interest points at multiple scales using Harris detector.

- Scales are chosen as follows: $\sigma_n = k^n \sigma$
- At each scale, choose local maxima assuming 3 x 3 window (i.e., non-maximal suppression)

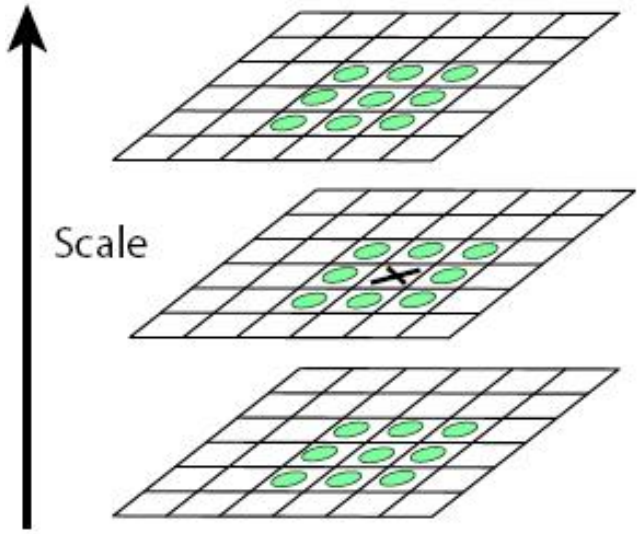
$$F(x, \sigma_n) > F(x_W, \sigma_n) \forall x_W \in W$$
$$F(x, \sigma_n) > t_h$$



where $F(x, \sigma_n) = \det(A_W) - \alpha trace^2(A_w)$ $(\sigma_D = \sigma_n, \sigma_I = \gamma\sigma_D)$

Harris-Laplace Detector

- (2) Select points at which the normalized LoG is maximal across scales and the maximum is above a threshold



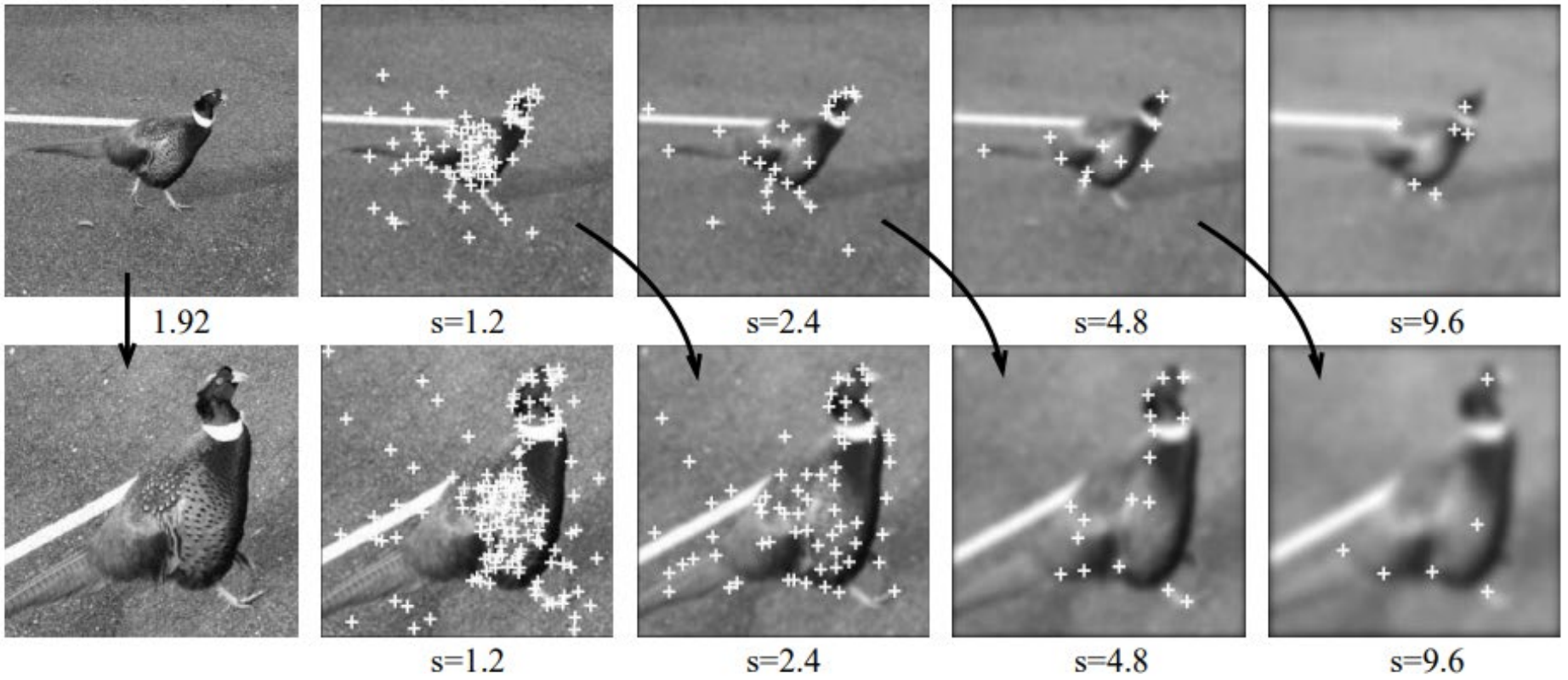
$$F(x,\sigma_n) > F(x,\sigma_{n-1}) \wedge F(x,\sigma_n) > F(x,\sigma_{n+1})$$
$$F(x,\sigma_n) > t$$

where:

$$F(x,\sigma_n) = |\sigma^2(L_{xx}(x,\sigma_n) + L_{yy}(x,\sigma_n))|$$

Example

- Image differ by a scale factor of 1.92
- Points detected on different resolution level with Harris Laplacian



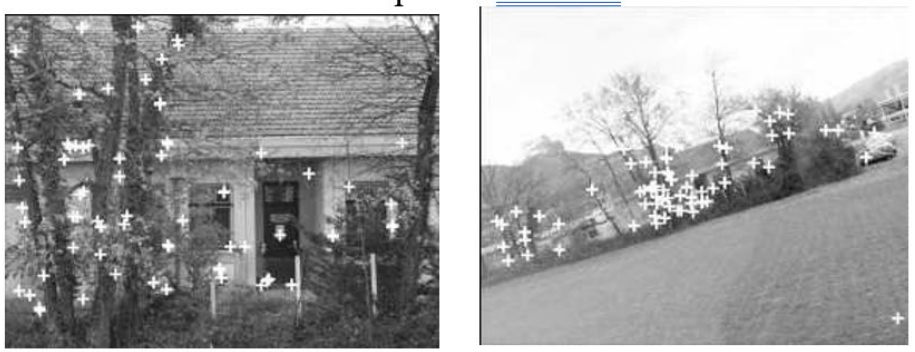
Example

- Two images are taken from
  - the same viewpoint
  - But different focal length and image orientation
- 190 and 213 points are initially detected (Top)
- 58 points are initially matched (Middle)
- 32 inliers found by RANSAC (Bottom)

- The rotation between image can then be estimated

Extracted interest points

Initial points matches



Inliers to the estimated homography

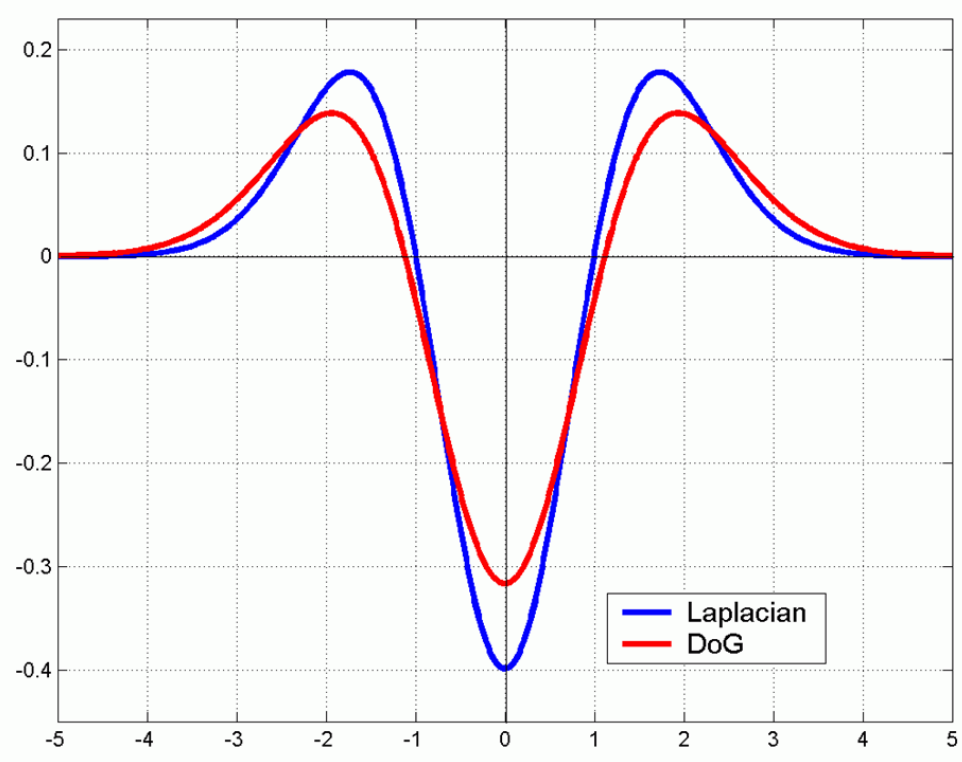
Example

- Images are taken from different view points
- 14 inliers is found for estimating the fundamental matrix and they are all correct.

Harris-Laplace using DoG

- We can efficiently approximate the Laplacian with a difference of Gaussians:

- $L = \sigma^2(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$
- (Laplacian)
- $DoG = G(x, y, k\sigma) - G(x, y, \sigma)$
- Difference of Gaussians
- Typical values: $\sigma = 1.6, k = \sqrt{2}$



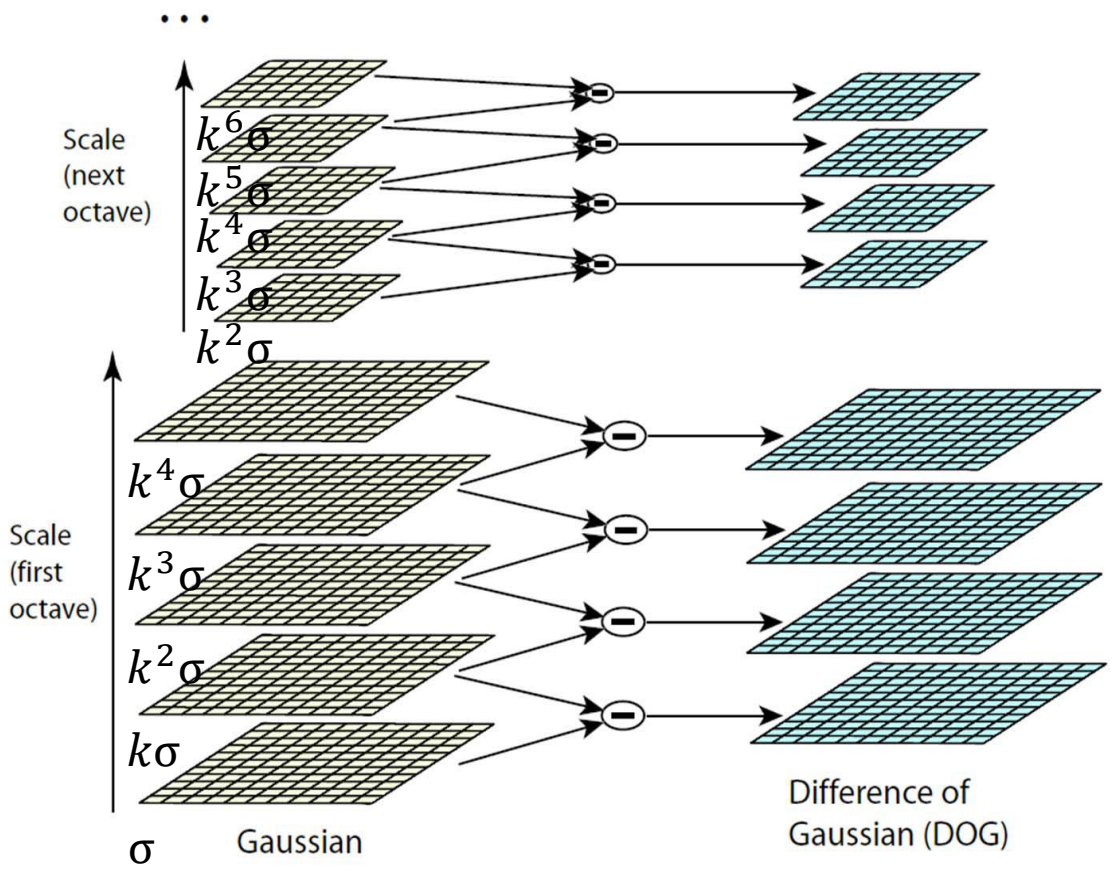
Slide credit: Bastian Leibe

Difference-of-Gaussian (DoG)

- Difference of Gaussians as approximation of the LoG
- Used in Harris-Laplace Lowe's SIFT pipeline for feature detection.
- Advantages
  - No need to compute second derivatives
  - Gaussian are computed in Gaussian pyramid

Slide credit: Bastian Leibe

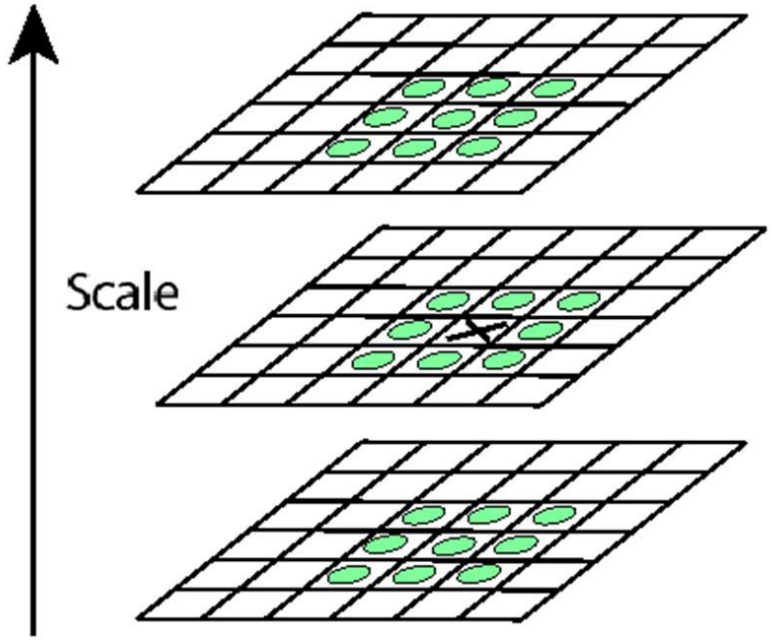
Building a Scale Space



$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-\frac{x^2+y^2}{2k^2\sigma^2}} \qquad k = \sqrt{2}$$

Key point localization in scale space

- Detect Maxima of difference-of-Gaussian (DoG) in scale space
- Compare a pixel (X) with 26 pixel in current and adjacent scales
- Select pixel (X) if it is Maxima
- Reject points with low contrast (threshold)
- Eliminate edge responses



Candiate keypoints list of (x,y,σ)

Slide credit: David Lowe

Summary of Scale Invariant Region Selection

- Given:
  - Two images of the same scene with large scale difference between them
- Goal:
  - Find the same interest points independently in each image
- Solution:
  - Search for maxima of suitable function over the scale
- Two methods
  - Laplacian of Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation

Summary of Today's lesson

- Edge Detector
  - Gradient Detector
    - Roberts
    - Prewitt
    - Sobel
  - Gradient of Gaussian (Canny)
  - Laplacian of Gaussian (Marr-Hildreth)
- Interest Point Detection
  - Harris Corner Detector
- Scale-invariant Region Selection
  - How to make the detector scale invariant.
- Harris-Laplace Detector

Reference

A COMBINED CORNER AND EDGE DETECTOR

Chris Harris & Mike Stephens

Plessey Research Roke Manor, United Kingdom

Consistency of image edge filtering is of prime importance for 3D interpretation of image sequences using feature tracking algorithms. To cater for image regions containing texture and isolated features, a combined corner and edge detector based on the local auto-correlation function is utilised, and it is shown to perform with good consistency on natural imagery.

INTRODUCTION

The problem we are addressing in Alvey Project MMI149 is that of using computer vision to understand the unconstrained 3D world, in which the viewed scenes will in general contain too wide a diversity of objects for top-down recognition techniques to work. For example, we desire to obtain an understanding of natural scenes, containing roads, buildings, trees, bushes, etc., as typified by the two frames from a sequence illustrated in Figure 1. The solution to this problem that we are pursuing is to use a computer vision system based upon motion analysis of a monocular image sequence from a mobile camera. By

they are discrete, reliable and meaningful[2]. However, the lack of connectivity of feature-points is a major limitation in our obtaining higher level descriptions, such as surfaces and objects. We need the richer information that is available from edges[3].

THE EDGE TRACKING PROBLEM

Matching between edge images on a pixel-by-pixel basis works for stereo, because of the known epi-polar camera geometry. However for the motion problem, where the camera motion is unknown, the aperture problem prevents us from undertaking explicit edgel matching. This could be overcome by solving for the motion beforehand, but we are still faced with the task of tracking each individual edge pixel and estimating its 3D location from, for example, Kalman Filtering. This approach is unattractive in comparison with assembling the edgels into edge segments, and tracking these segments as the features.

Now, the unconstrained imagery we shall be considering will contain both curved edges and texture of various

Indexing based on scale invariant interest points

Krystian Mikolajczyk* Cordelia Schmid

INRIA Rhône-Alpes GRAVIR-CNRS
655 av. de l'Europe, 38330 Montbonnot, France
{Krystian.Mikolajczyk,Cordelia.Schmid}@inrialpes.fr

Abstract

This paper presents a new method for detecting scale invariant interest points. The method is based on two recent results on scale space: 1) Interest points can be adapted to scale and give repeatable results (geometrically stable). 2) Local extrema over scale of normalized derivatives indicate the presence of characteristic local structures. Our method first computes a multi-scale representation for the Harris interest point detector. We then select points at which a local measure (the Laplacian) is maximal over scales. This allows a selection of distinctive points for which the characteristic scale is known. These points are invariant to scale, rotation and translation as well as robust to illumination changes and limited changes of viewpoint.

For indexing, the image is characterized by a set of scale invariant points; the scale associated with each point allows the computation of a scale invariant descriptor. Our descriptors are, in addition, invariant to image rotation, to affine illumination changes and robust to small perspective deformations. Experimental results for indexing show an excellent performance up to a scale factor of 4 for a database with more than 5000 images.

1 Introduction

The difficulty in object indexing is to determine the identity of an object under arbitrary viewing conditions in the presence of cluttered real-world scenes or occlusions. Local characterization has shown to be well adapted to this

ant descriptors which are combinations of Gaussian derivatives. Robustness to scale changes is obtained by computing Gaussian derivatives at several scales. Lowe [8] extends these ideas to scale invariance by maximizing the output of difference-of-Gaussian filters in scale-space. Tuytelaars et al. [13] have developed affine invariant descriptors by searching for affine invariant regions and describing them by color invariants. To find these regions they simultaneously use interest points and contours. Instead of using an initial set of features, Chomat et al. [2] select the appropriate scale for every point in the image and compute descriptors at these scales. An object is represented by the set of these descriptors. All of the above methods are limited to a scale factor of 2.

Similar approaches exist for wide-baseline matching [1, 3, 5, 9, 12]. The problem is however more restricted. Additional constraints can be imposed and the search complexity is less prohibitive. For example, Prichett and Zisserman [9] first match regions bound by four line segments. They then use corresponding regions to compute the homography and grow the regions. Such an approach is clearly difficult to extend to the problem of indexing. Two of the papers on wide-baseline matching have specifically addressed the problem of scale. Hansen et al. [5] present a method that uses correlation of scale traces through multi-resolution images to find correspondence between images. A scale trace is a set of values for a pixel at different scales of computation. Dufournaud et al. [3] use a robust multi-scale framework to match images. Interest points and descriptors are computed