Lecture 8:
Interest Point Detection

Saad J Bedros

sbedros@umn.edu

Review of Edge Detectors

- 3×3 gradient operators (Prewitt, Sobel) are **simple and fast**. Used when
  - fine edges are only needed
  - noise level is low
- By varying σ parameter, **Canny operator** can be used
  - to detect fine as well as rough edges
  - at different noise levels
- All **gradient operators**
  - provide edge orientation
  - need localisation: non-maxima suppression, hysteresis threshoding
- **Zero-crossing** edge detector
  - is supported by neurophysiological experiments
  - was popular in the 1980's
  - today, **less frequently used** in practice

Today's Lecture

- Interest Points Detection
- What do we mean with Interest Point Detection in an Image
- Goal: Find Same features between multiple images taken from different position or time

Applications

- Image alignment
- Image Stitching
- 3D reconstruction
- Object recognition
- Indexing and database retrieval
- Object tracking
- Robot navigation

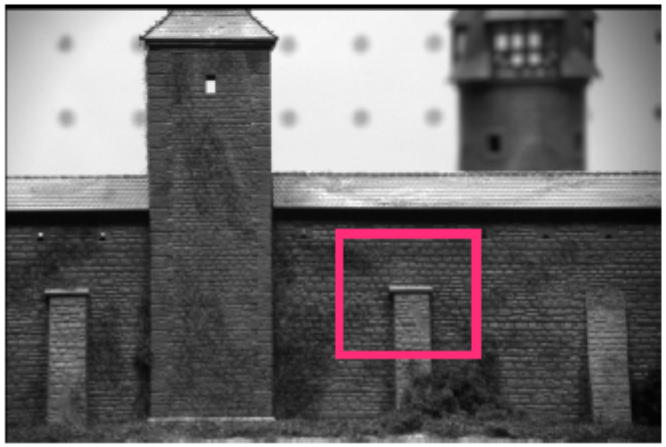
Image Alignment

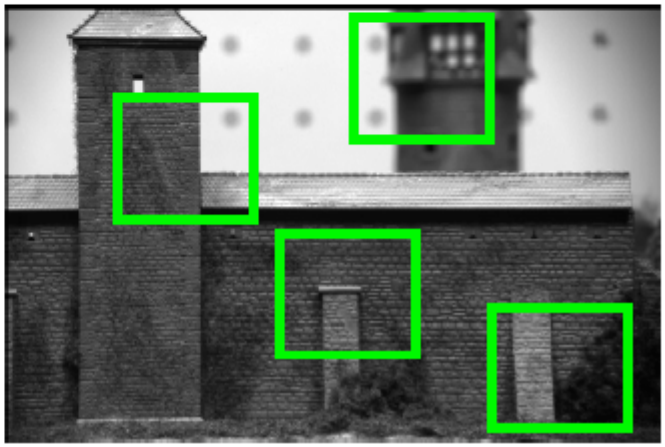
- Homography
- Ransac

Robert Collins
CSE486, Penn State

Motivation: Patch Matching

Elements to be matched are image patches of fixed size





Task: find the best (most similar) patch in a second image

Known as:
Template Matching Method, with option to use correlation as a metric

Correlation

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

f = Image

h = Kernel

f

f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

$\otimes$

h

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

$$f * h = f_1h_1 + f_2h_2 + f_3h_3 + f_4h_4 + f_5h_5 + f_6h_6 + f_7h_7 + f_8h_8 + f_9h_9$$

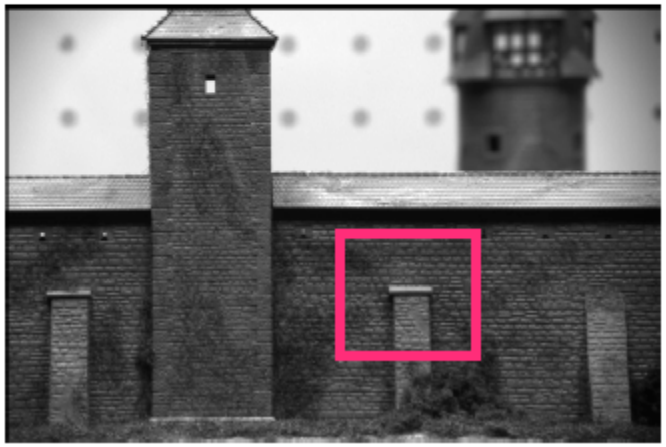
Correlation

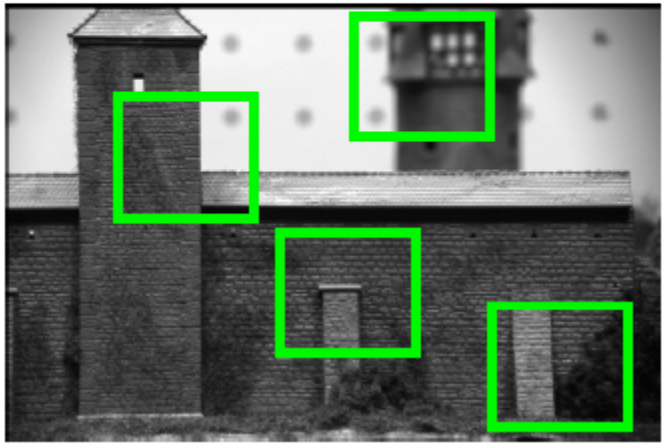
$$f \otimes h = \sum_k \sum_l f(k,l) h(i+k, j+l)$$ Cross correlation

$$f \otimes f = \sum_k \sum_l f(k,l) f(i+k, j+l)$$ Auto correlation

- Use Cross Correlation to find the template in an image,
- Maximum indicate high similarity

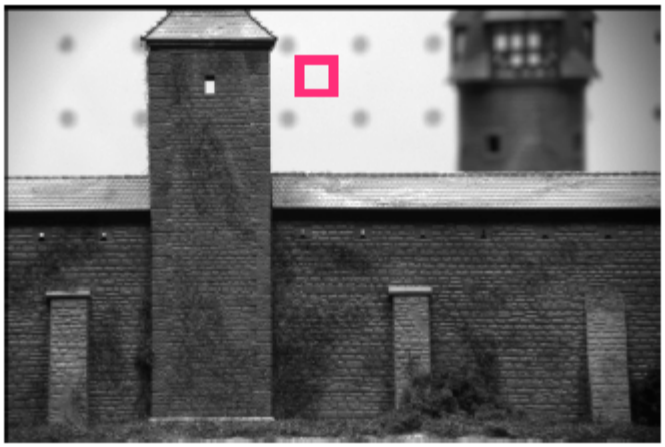
Not all Patches are Created Equal!



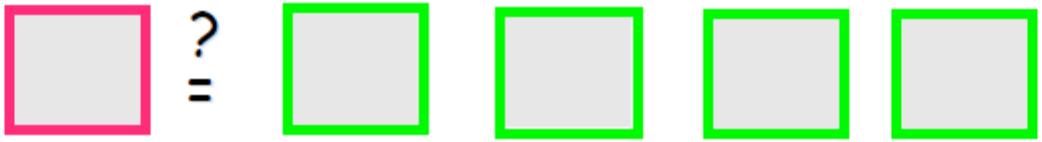


Inituition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar).

Not all Patches are Created Equal!



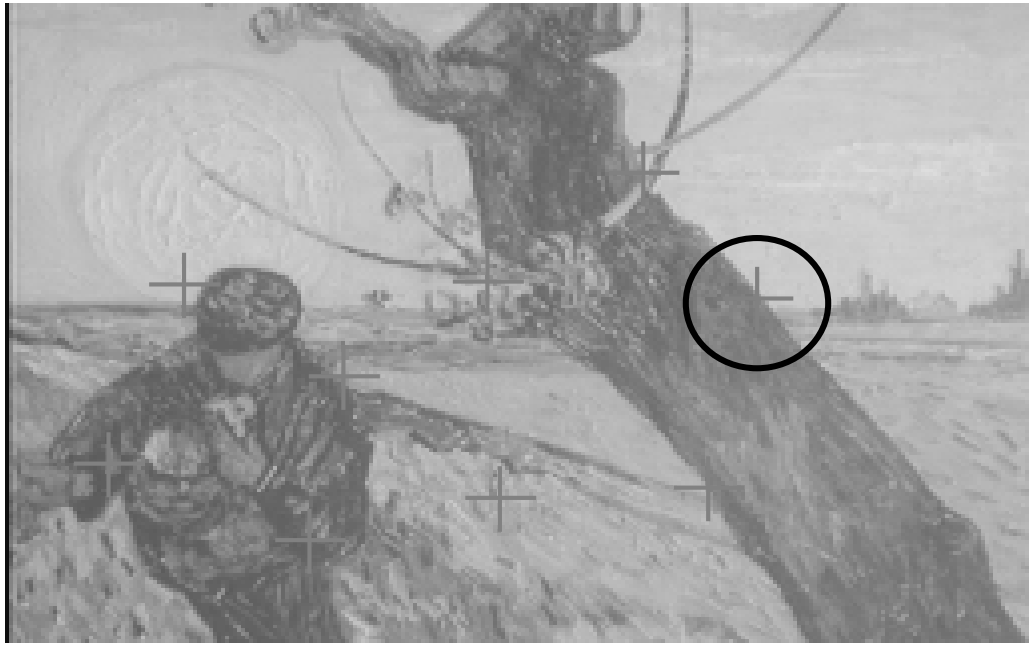
Inituition: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame)



We need more robust feature descriptors for matching !!!!!

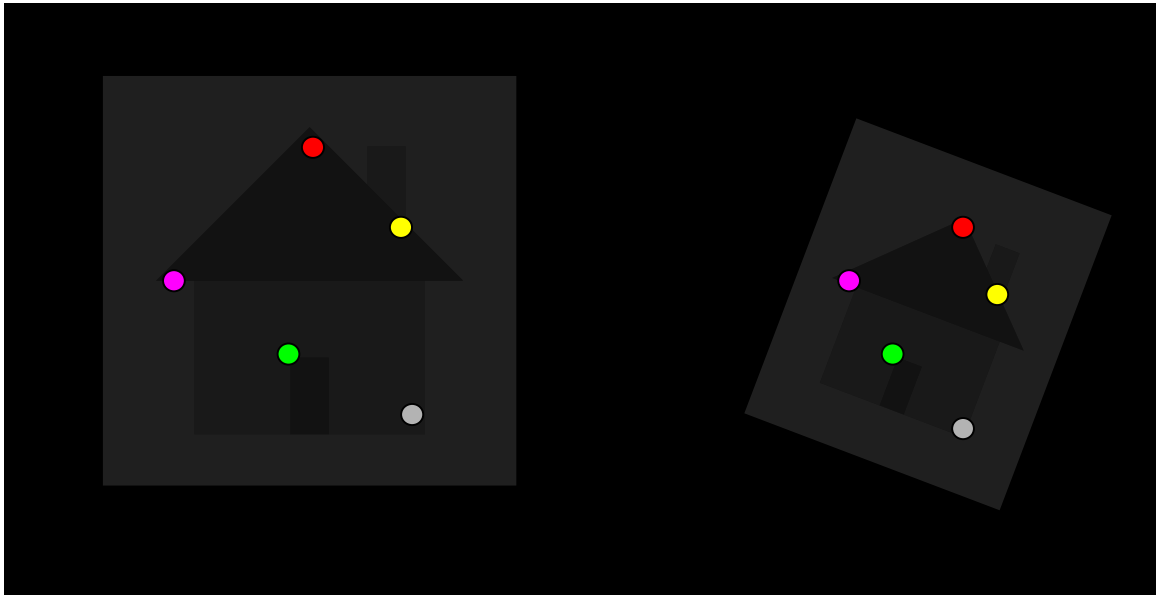
Interest Points

- Local features associated with a significant change of an image property of several properties simultaneously (e.g., intensity, color, texture).

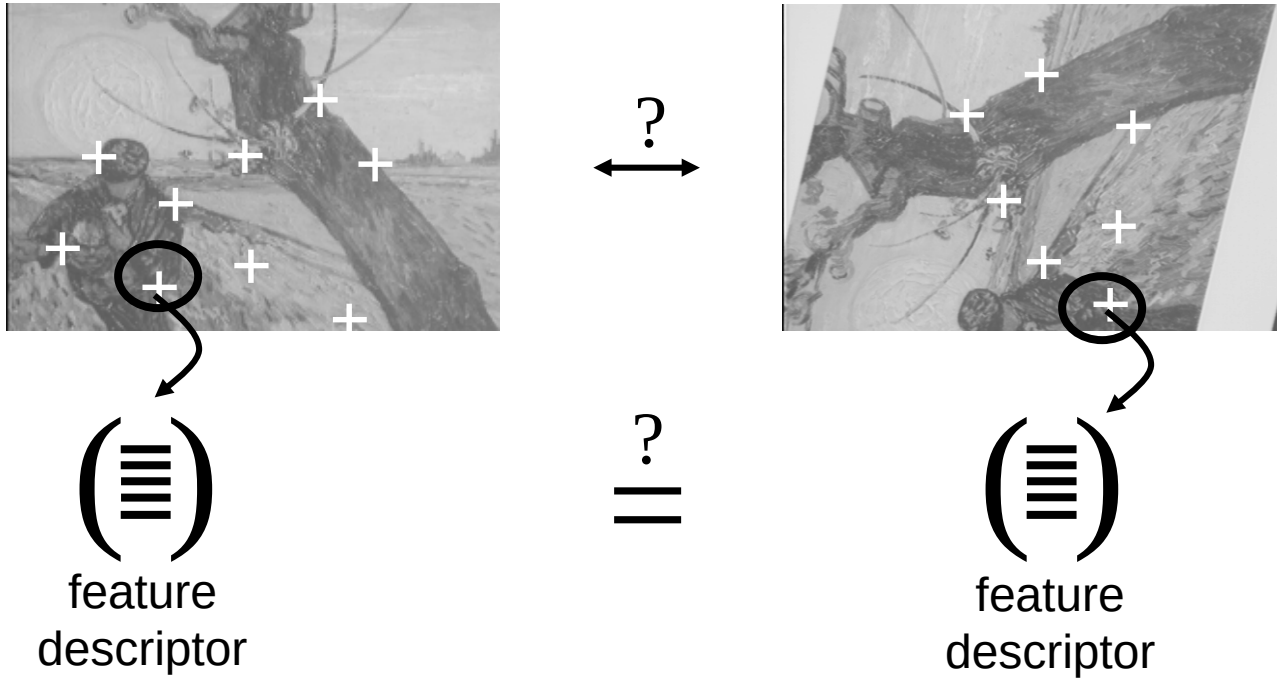


Why Extract Interest Points?

- Corresponding points (or features) between images enable the estimation of parameters describing geometric transforms between the images.



What if we don't know the correspondences?



- Need to compare *feature descriptors* of local patches surrounding interest points

Properties of good features

- **Local:** features are local, robust to occlusion and clutter (no prior segmentation!).
- **Accurate:** precise localization.

- **Invariant** (or **covariant**)
- **Robust:** noise, blur, compression, etc. do not have a big impact on the feature.

} **Repeatable**

- **Distinctive:** individual features can be matched to a large database of objects.
- **Efficient:** close to real-time performance.

Invariant / Covariant

- A function f( ) is **invariant** under some transformation T( ) if its value does change when the transformation is applied to its argument:

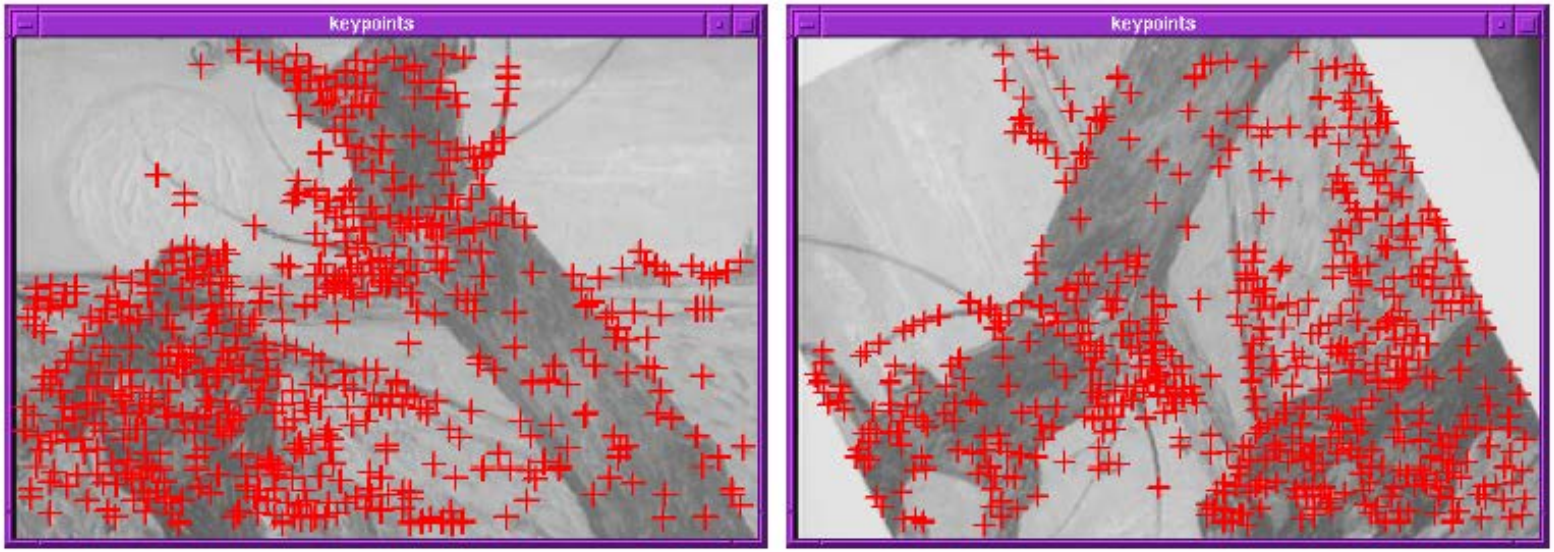
$$\text{if } f(x) = y \text{ then } f(T(x))=y$$

- A function f( ) is **covariant** when it commutes with the transformation T( ):

$$\text{if } f(x) = y \text{ then } f(T(x))=T(f(x))=T(y)$$

Invariance

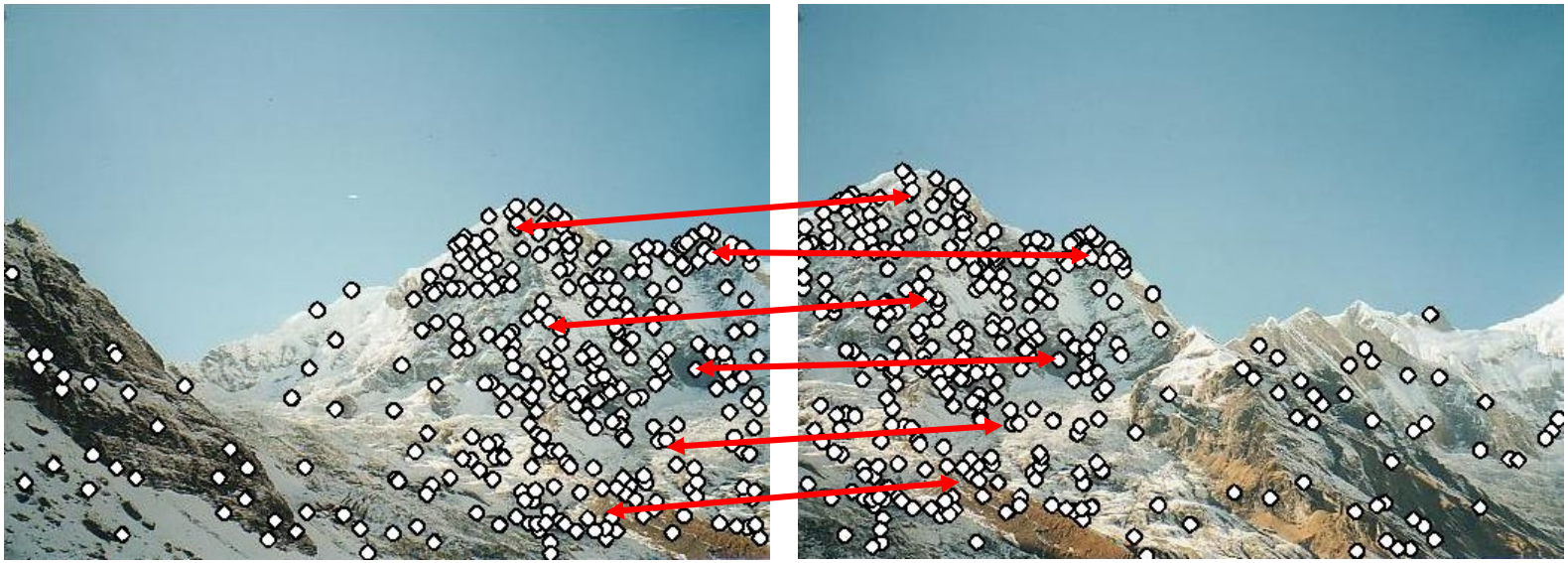
- Features should be detected despite geometric or photometric changes in the image.
- Given two transformed versions of the same image, features should be detected in corresponding locations.



Example: Panorama Stitching

- How do we combine these two images?

Panorama stitching (cont'd)



Step 1: extract features

Step 2: match features

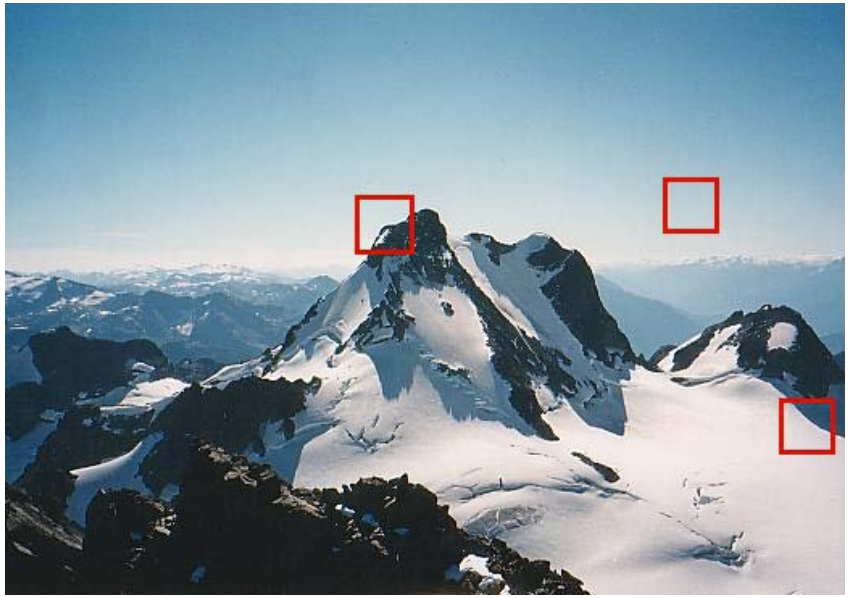
Panorama stitching (cont'd)

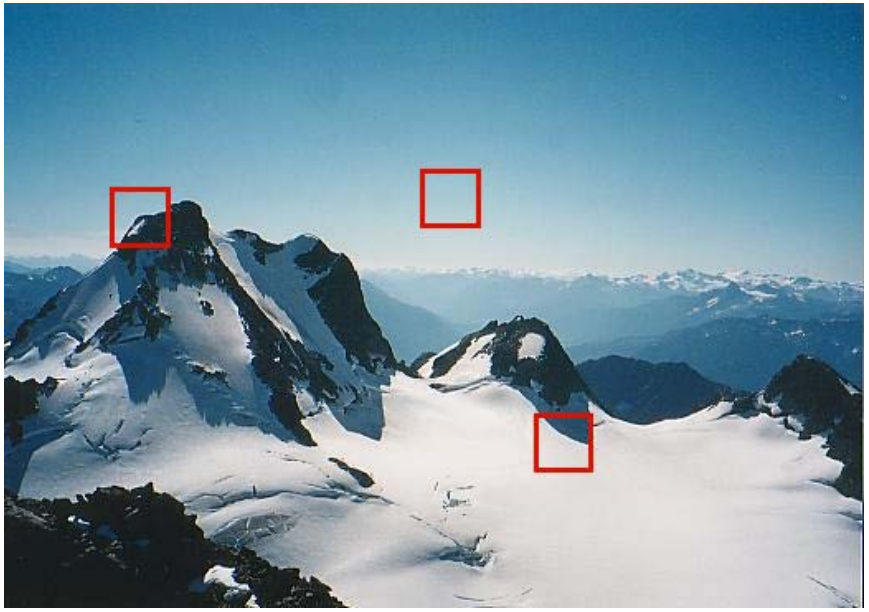
Step 1: extract features

Step 2: match features

Step 3: align images

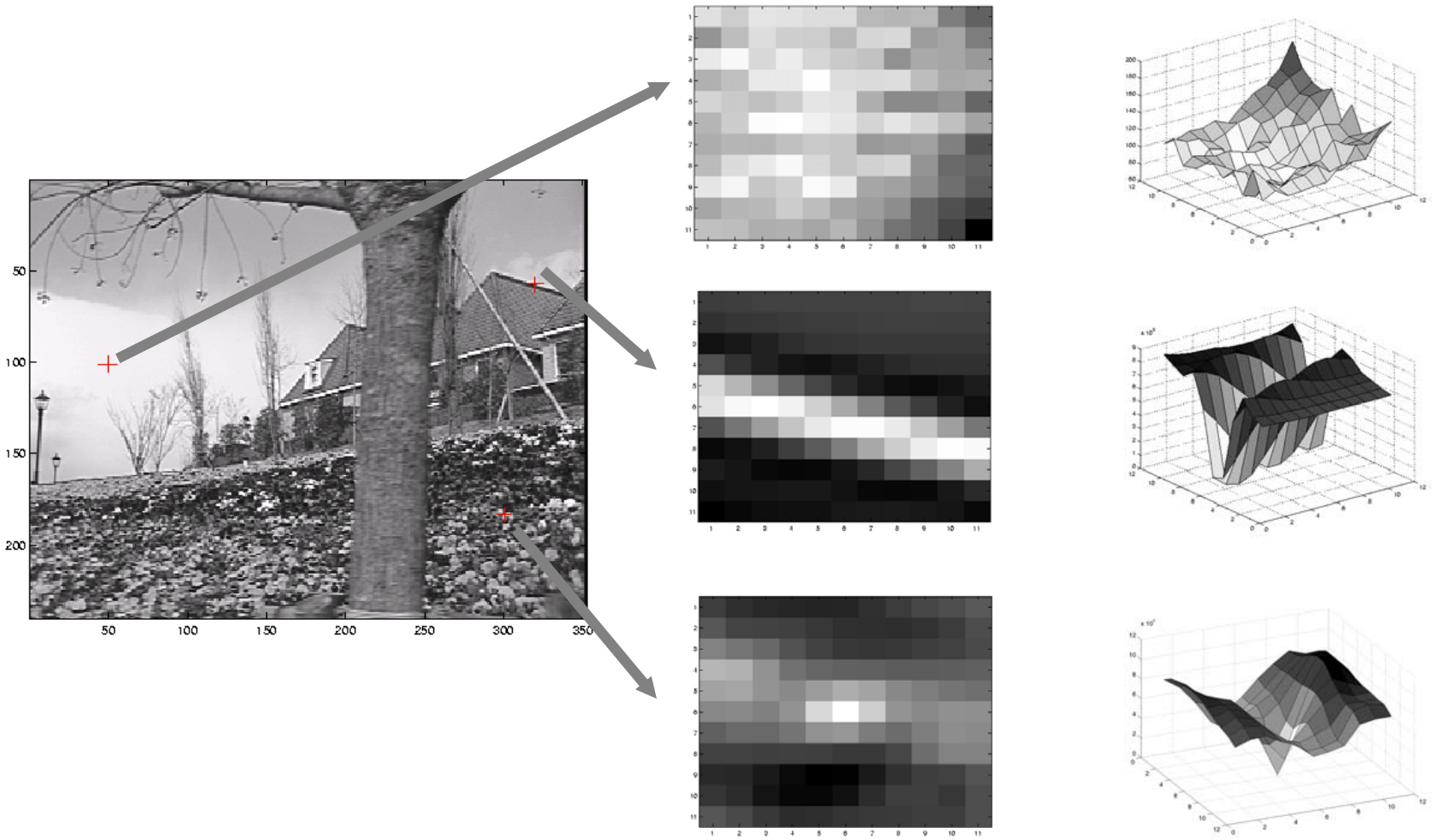
What features should we use?



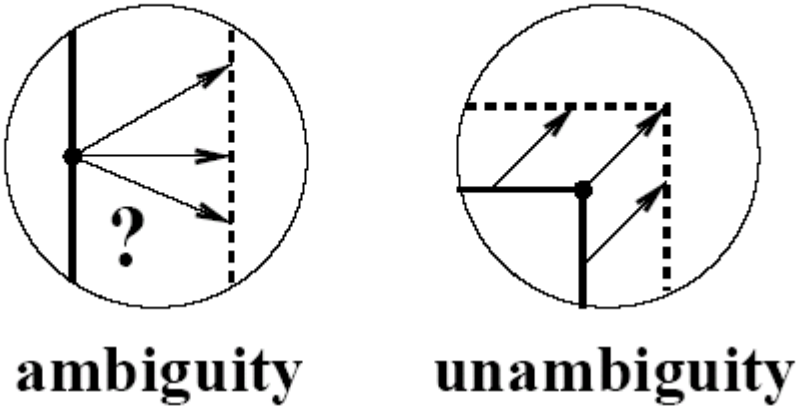


Use features with gradients in at least two (significantly) different orientations → patches ? Corners ?

What features should we use? (cont'd)

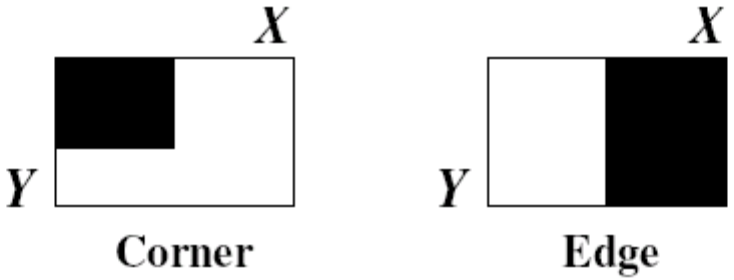


The Aperture Problem



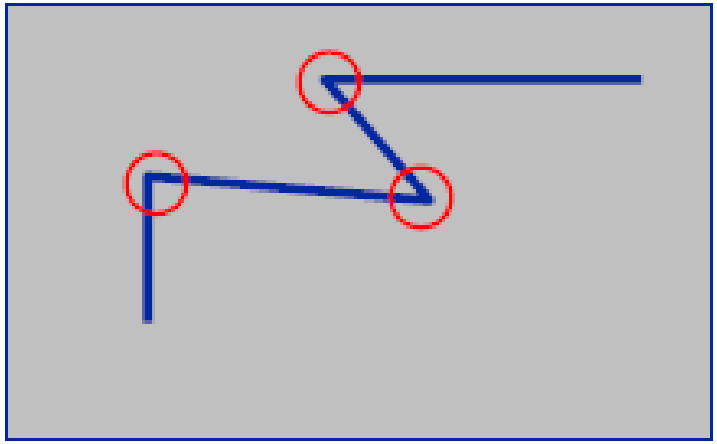
- Motion vectors are **ambiguous** at edge
  - locally, normal component can only be determined
  - tangential component cannot be determined
- Motion vectors are **unambiguous** at corner

Corners vs Edges



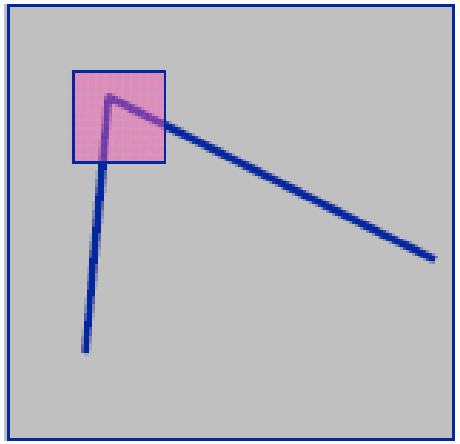
- **Corners** are locations where variations of intensity function $f(x, y)$ in both X **and** Y are high
  - $\Rightarrow$ both partial derivatives f_x and f_y are large
- **Edges** are locations where variation of $f(x, y)$ in certain direction is high, while variation in the orthogonal direction is low
  - $\Rightarrow$ when edge is oriented along Y, f_x is large, f_y small

Corner Detection

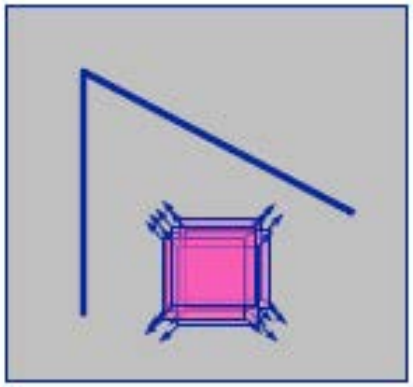


Corner Detection-Basic Idea

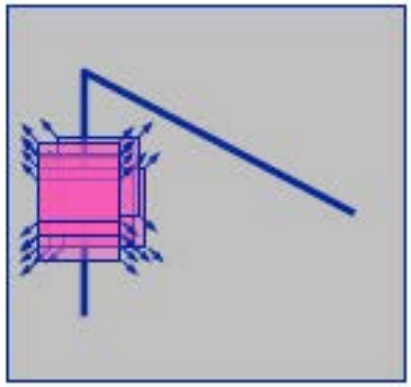
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in response



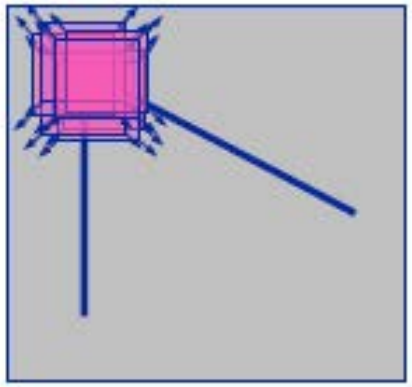
Corner Detector: Basic Idea



"flat" region:
no change in
all directions

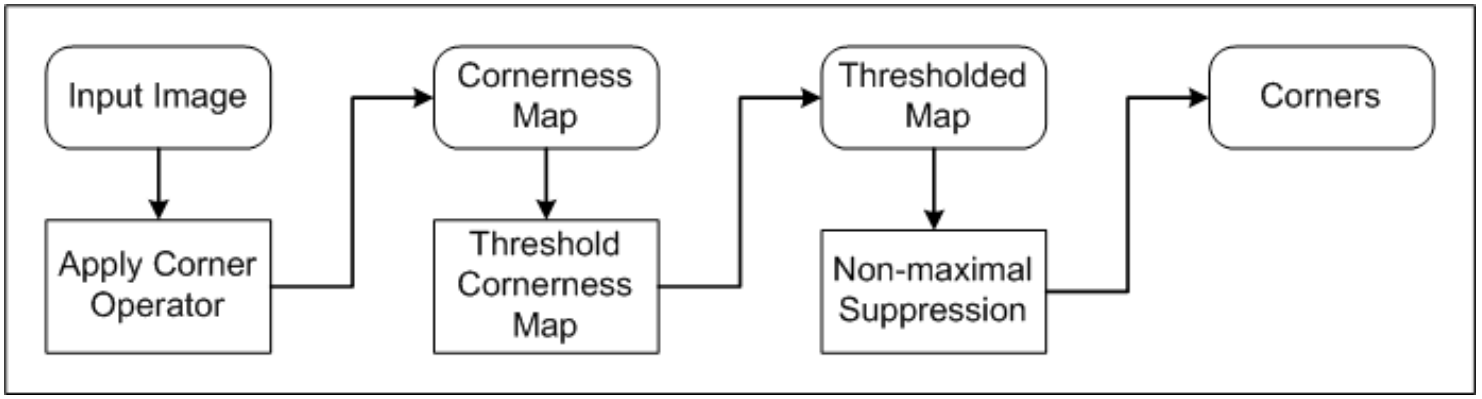


"edge":
no change along
the edge direction



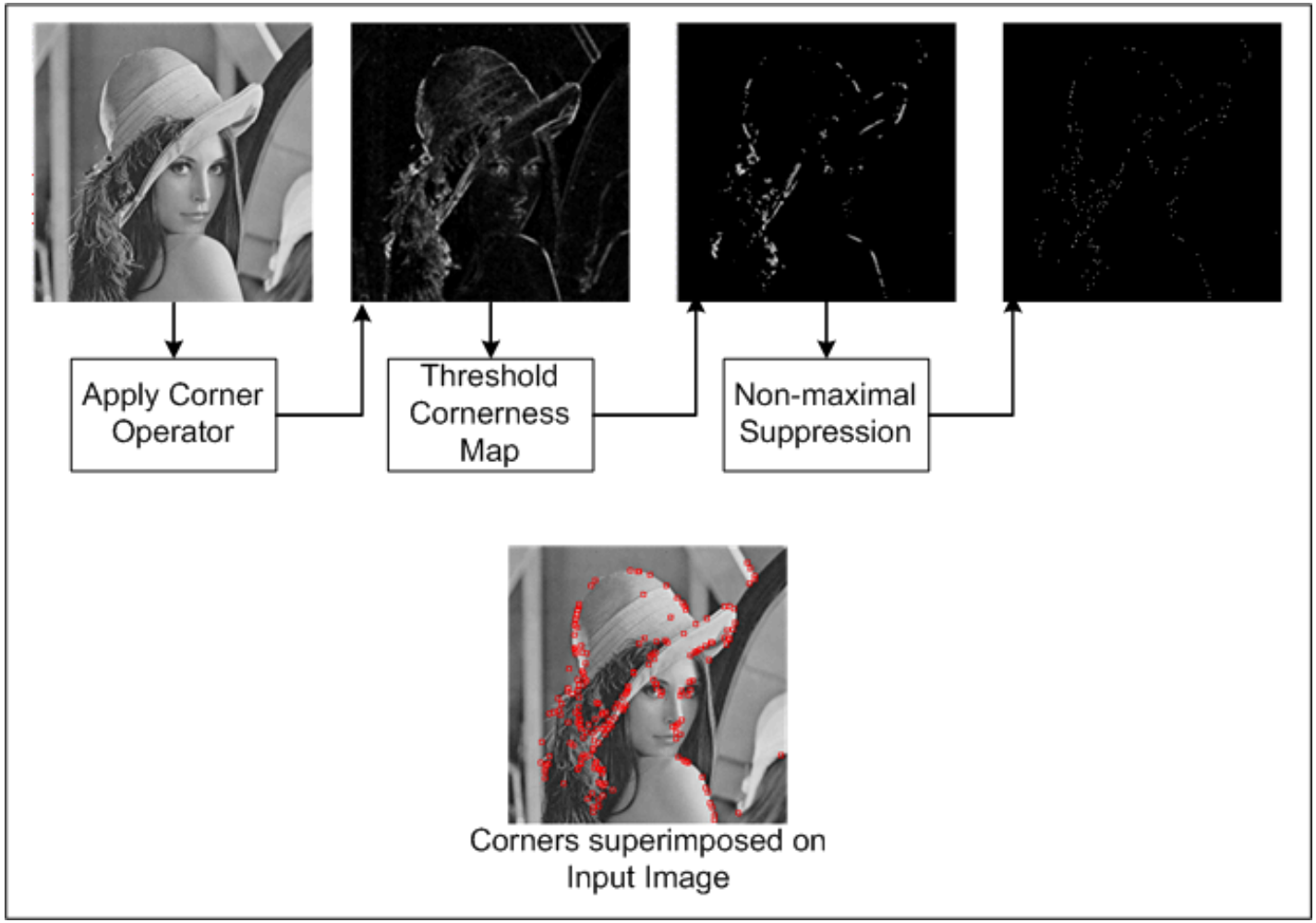
"corner":
significant change
in all directions

Mains Steps in Corner Detection



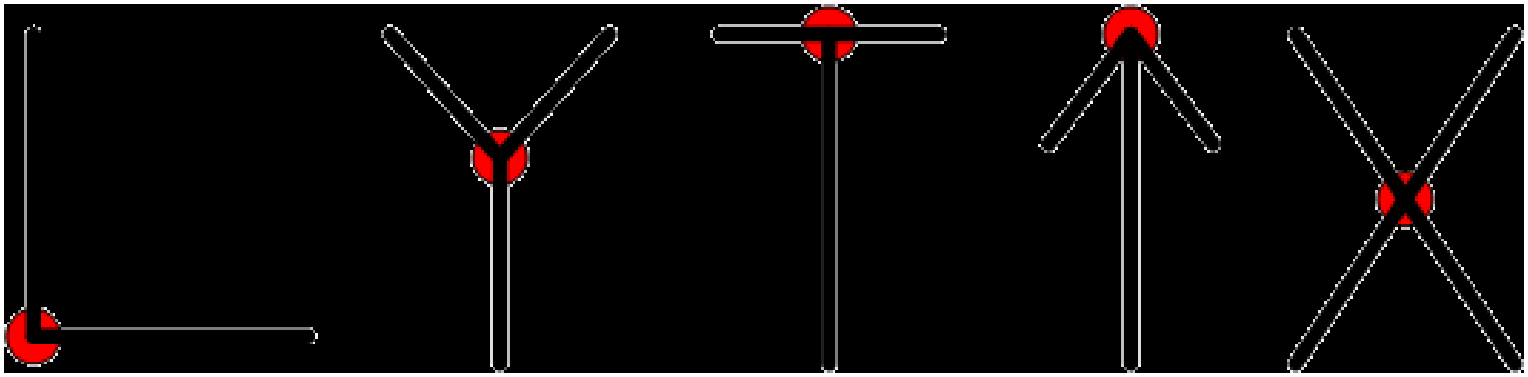
1. For each pixel in the input image, the corner operator is applied to obtain a *cornerness* measure for this pixel.
2. Threshold cornerness map to eliminate weak corners.
3. Apply non-maximal suppression to eliminate points whose cornerness measure is not larger than the cornerness values of all points within a certain distance.

Mains Steps in Corner Detection (cont'd)



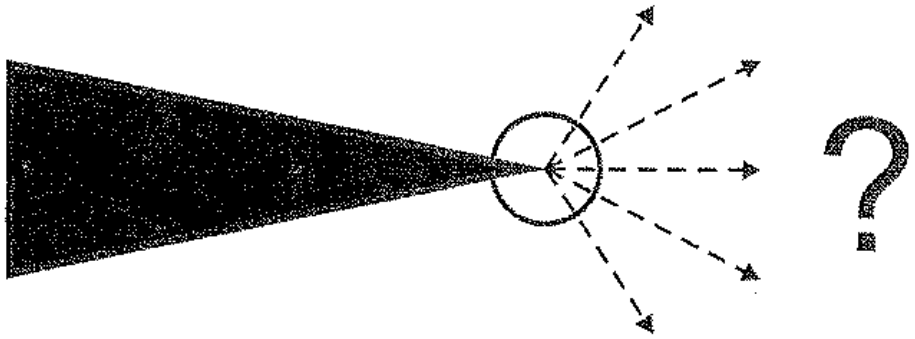
Corner Types

Example of L-junction, Y-junction, T-junction, Arrow-junction, and X-junction corner types



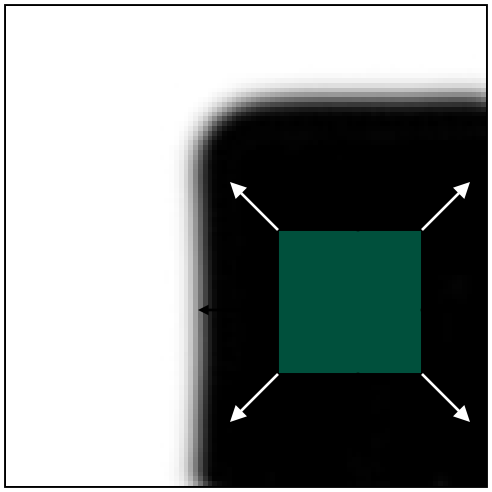
Corner Detection Using Edge Detection?

- Edge detectors are not stable at corners.
- Gradient is ambiguous at corner tip.
- Discontinuity of gradient direction near corner.

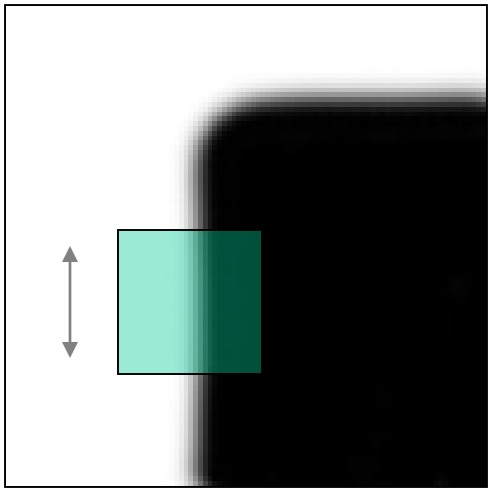


Corner Detection Using Intensity: Basic Idea

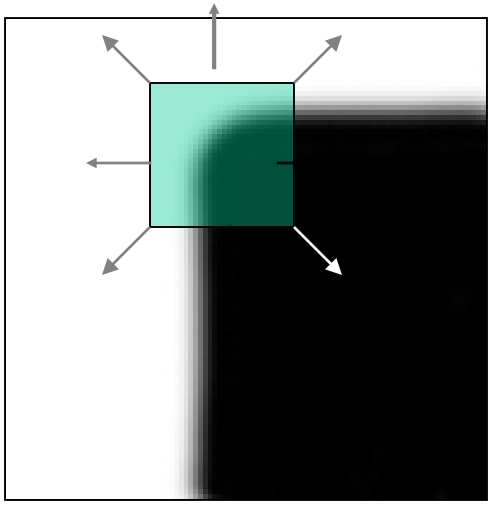
- Image gradient has two or more dominant directions near a corner.
- Shifting a window in *any direction* should give *a large change* in intensity.



"flat" region:
no change in all directions



"edge": no change along the edge direction



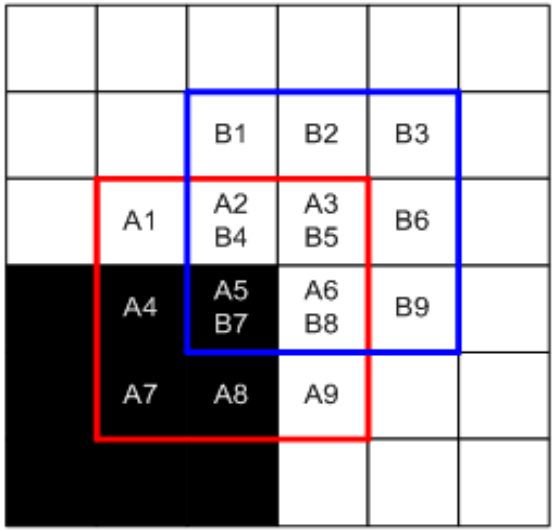
"corner": significant change in all directions

Moravec Detector (1977)

- Measure intensity variation at (x,y) by shifting a small window (3x3 or 5x5) by one pixel in each of the **eight** principle directions (horizontally, vertically, and four diagonals).

Moravec Detector (1977)

- Calculate intensity variation by taking the sum of squares of intensity differences of corresponding pixels in these two windows.



$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y)\right)^2 .$$

8 directions

$\Delta x, \Delta y$ in {-1,0,1}

$S_W(-1,-1), S_W(-1,0), \ldots S_W(1,1)$

Moravec Detector (cont'd)

- The "cornerness" of a pixel is the **minimum** intensity variation found over the eight shift directions:

Cornerness(x,y) = $\min\{S_W(-1,-1), S_W(-1,0), \ldots S_W(1,1)\}$

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Cornerness Map (normalized)

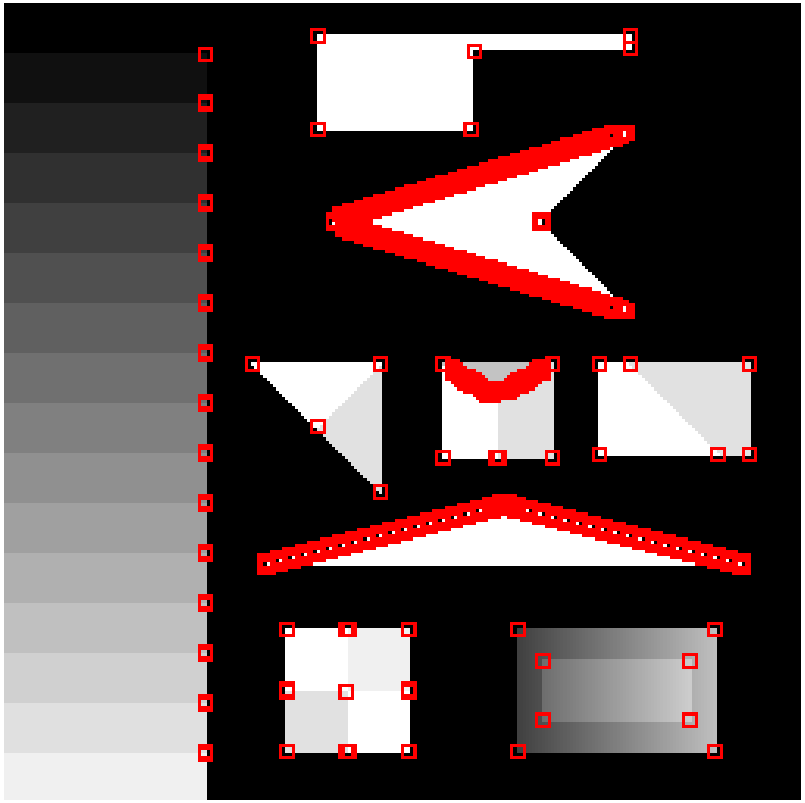
Note response to isolated points!

Moravec Detector (cont'd)

- Non-maximal suppression will yield the final corners.

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

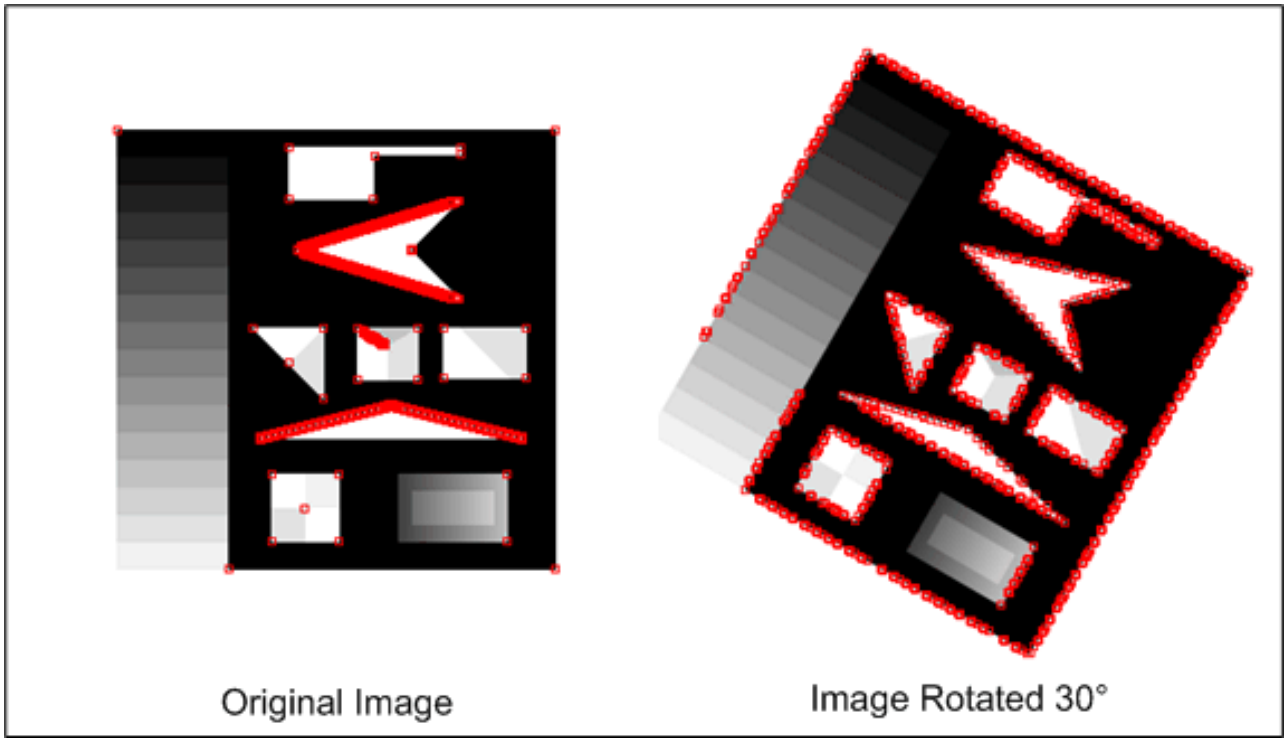
Moravec Detector (cont'd)



- Does a reasonable job in finding the majority of true corners.
- Edge points not in one of the eight principle directions will be assigned a relatively large cornerness value.

Moravec Detector (cont'd)

- The response is anisotropic as the intensity variation is only calculated at a discrete set of shifts (i.e., not rotationally invariant)



Harris Detector

Change of intensity for the shift $[u, v]$:

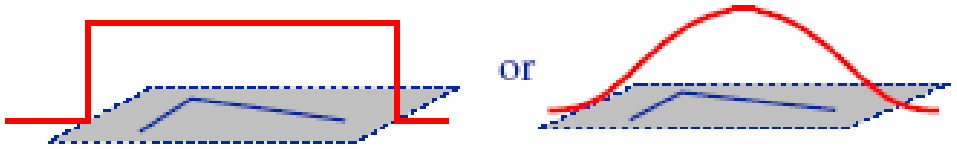
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

Window function $w(x,y) =$



- Improves the Moravec operator by avoiding the use of discrete directions and discrete shifts.
- Uses a Gaussian window instead of a square window.

Harris Detector

For small shifts $[u, v]$ we have a *bilinear* approximation:

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

where M is a 2×2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Harris Detector (cont'd)

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y) \right)^2 .$$

- Using first-order Taylor approximation:

$$f(x_i - \Delta x, y_i - \Delta y) \approx f(x_i, y_i) + \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} ,$$

$$f(x) = \mathrm{f}(a) + \frac{\mathrm{f}'(a)}{1!}(x-a) + \frac{\mathrm{f}''(a)}{2!}(x-a)^2 + \ldots + \frac{\mathrm{f}^{(n)}(a)}{n!}(x-a)^n + O(x^{n+1})$$

Harris Detector (cont'd)

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i, y_i) - \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$

$$= \sum_{x_i \in W} \sum_{y_i \in W} \left(- \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$

$$= \sum_{x_i \in W} \sum_{y_i \in W} \left(\left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$

Since $\mathbf{u}^2 = \mathbf{u}^\top \mathbf{u}$

Harris Detector (cont'd)

$$= \sum_{x_i \in W} \sum_{y_i \in W} [\Delta x, \Delta y] \left(\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= [\Delta x, \Delta y] \left(\sum_{x_i \in W} \sum_{y_i \in W} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= [\Delta x, \Delta y] \, A_W(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} ,$$

$A_W(x,y)=$

$$\begin{bmatrix} \sum_{x_i \in W} \sum_{y_i \in W} \left(\frac{\partial f(x_i, y_i)}{\partial x}\right)^2 & \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} \\ \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} & \sum_{x_i \in W} \sum_{y_i \in W} \left(\frac{\partial f(x_i, y_i)}{\partial y}\right)^2 \end{bmatrix}$$

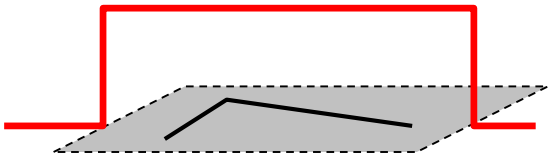
2 x 2 matrix (auto-correlation or 2nd order moment matrix)

Harris Detector

- General case – use window function:

$$S_W(\Delta x, \Delta y) = \sum_{x_i, y_i} w(x_i, y_i)\left[f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y)\right]^2$$

default window function $w(x,y)$:



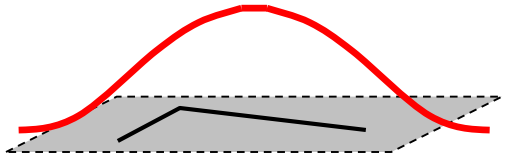
1 in window, 0 outside

$$A_W = \begin{bmatrix} \sum_{x,y} w(x,y) f_x^2 & \sum_{x,y} w(x,y) f_x f_y \\ \sum_{x,y} w(x,y) f_x f_y & \sum_{x,y} w(x,y) f_y^2 \end{bmatrix} = \sum_{x,y} w(x,y) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

Harris Detector (cont'd)

- Harris uses a Gaussian window: $\mathbf{w(x,y)=G(x,y,\sigma_I)}$ where σ_I is called the "integration" scale

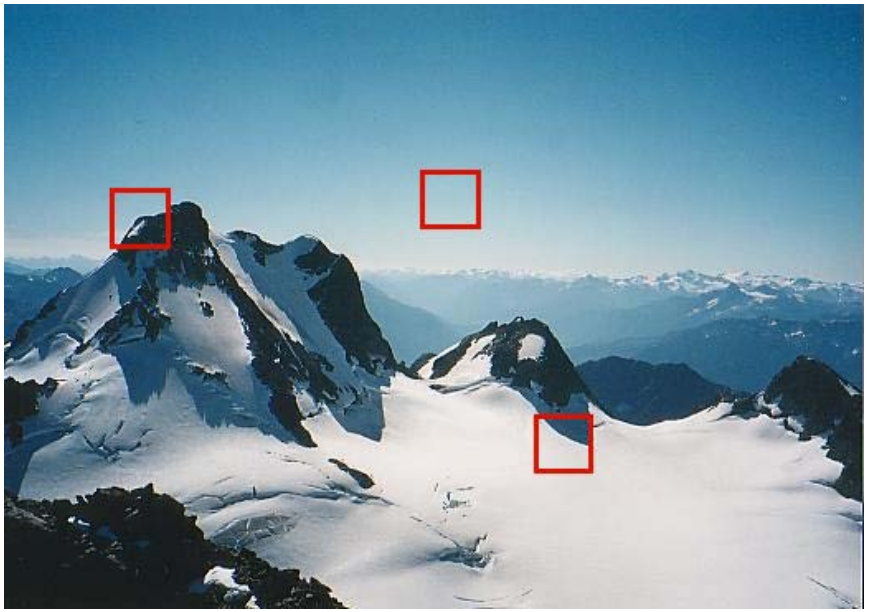
window function $w(x,y)$:



Gaussian

$$A_W = \begin{bmatrix} \sum_{x,y} w(x,y) f_x^2 & \sum_{x,y} w(x,y) f_x f_y \\ \sum_{x,y} w(x,y) f_x f_y & \sum_{x,y} w(x,y) f_y^2 \end{bmatrix} = \sum_{x,y} w(x,y) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

Harris Detector



$$A_W = \sum_{x,y} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

Describes the gradient distribution (i.e., local structure) inside window!

Does not depend on $\Delta x, \Delta y$

Harris Detector (cont'd)

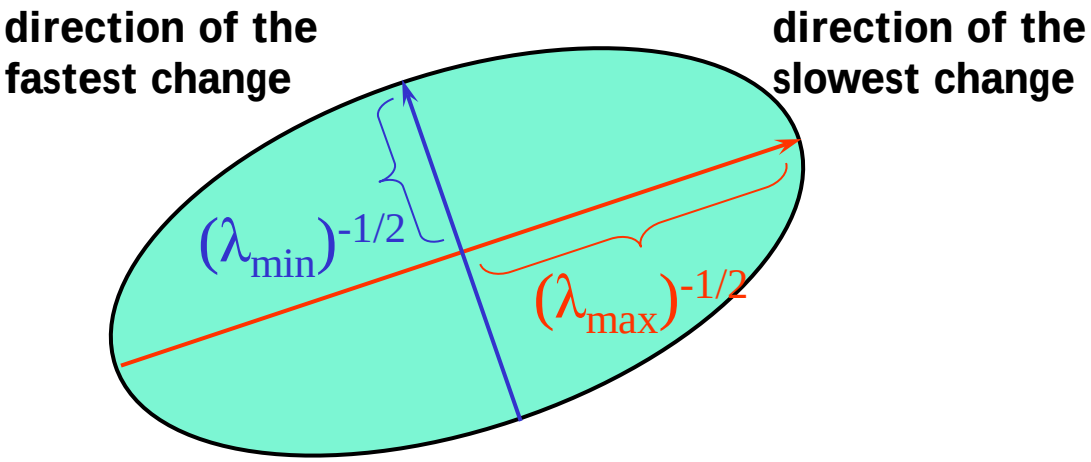
Since M is symmetric, we have:

$$A_W = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

We can visualize A_W as an ellipse with axis lengths determined by the eigenvalues and orientation determined by R

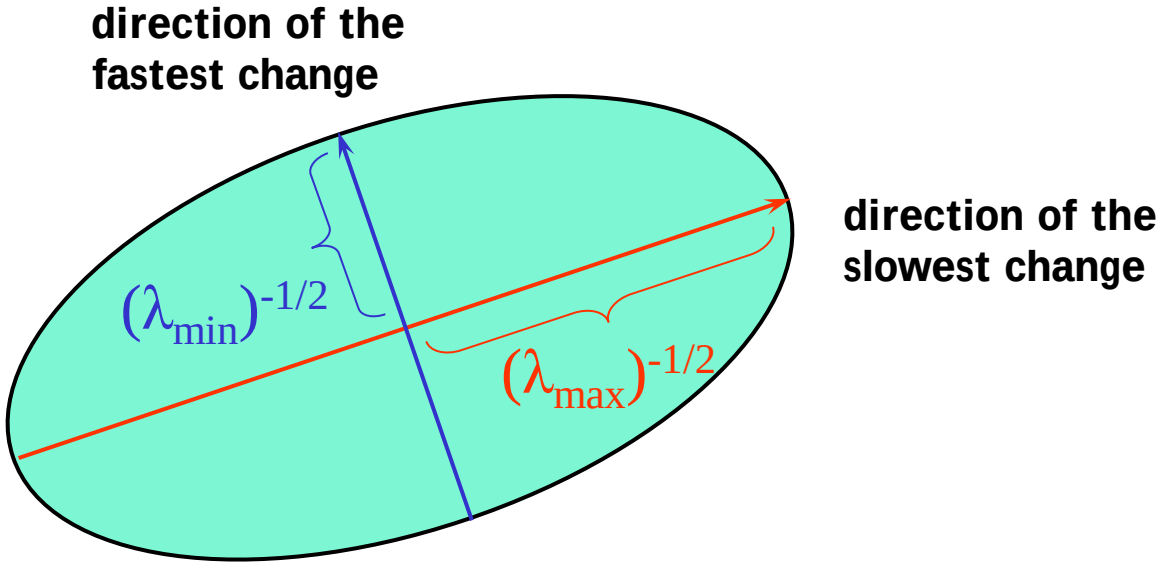
Ellipse equation:

$$[\Delta x \;\; \Delta y] \; A_W \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \text{const}$$



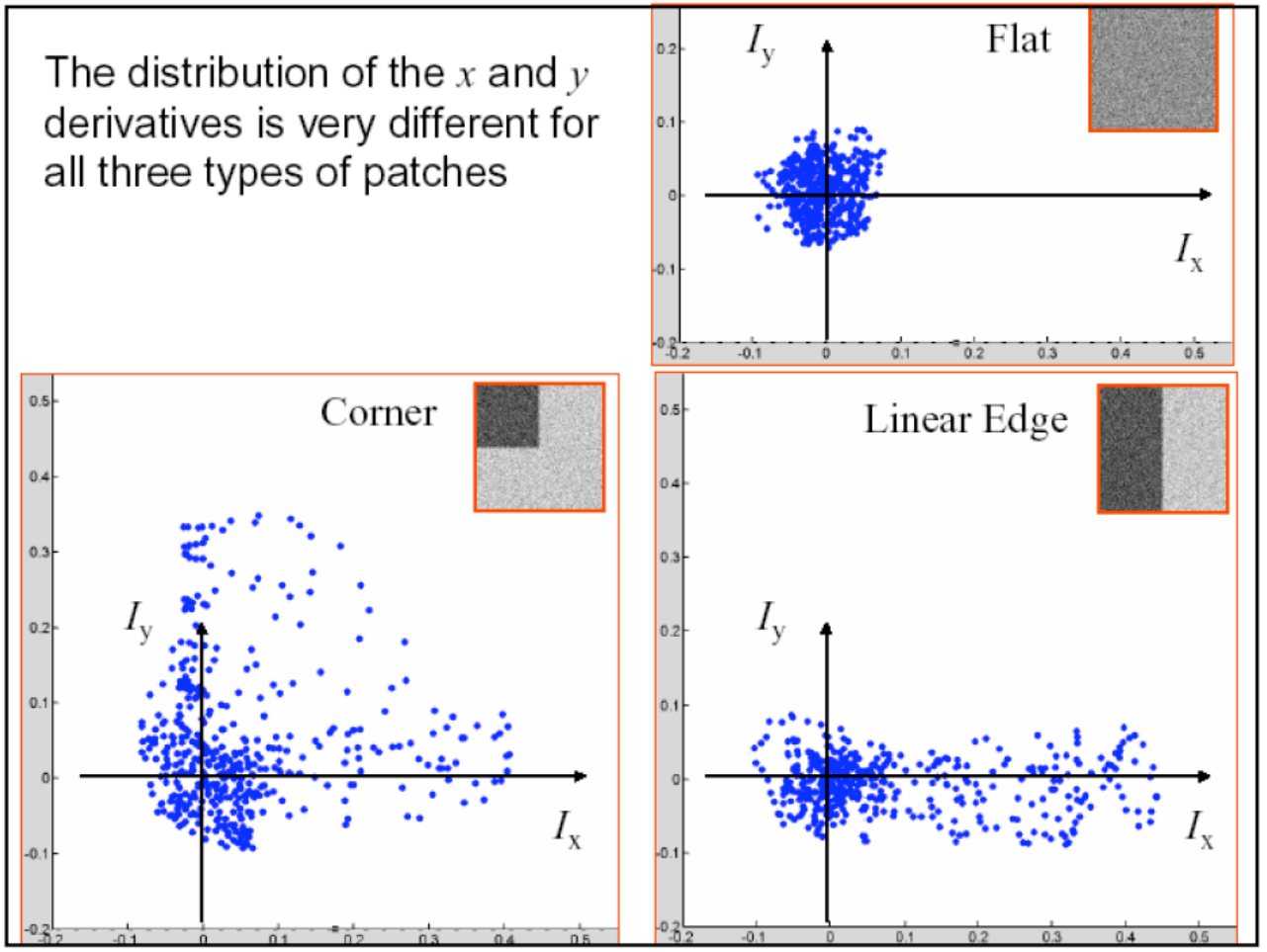
Harris Detector (cont'd)

- Eigenvectors encode edge direction
- Eigenvalues encode edge strength

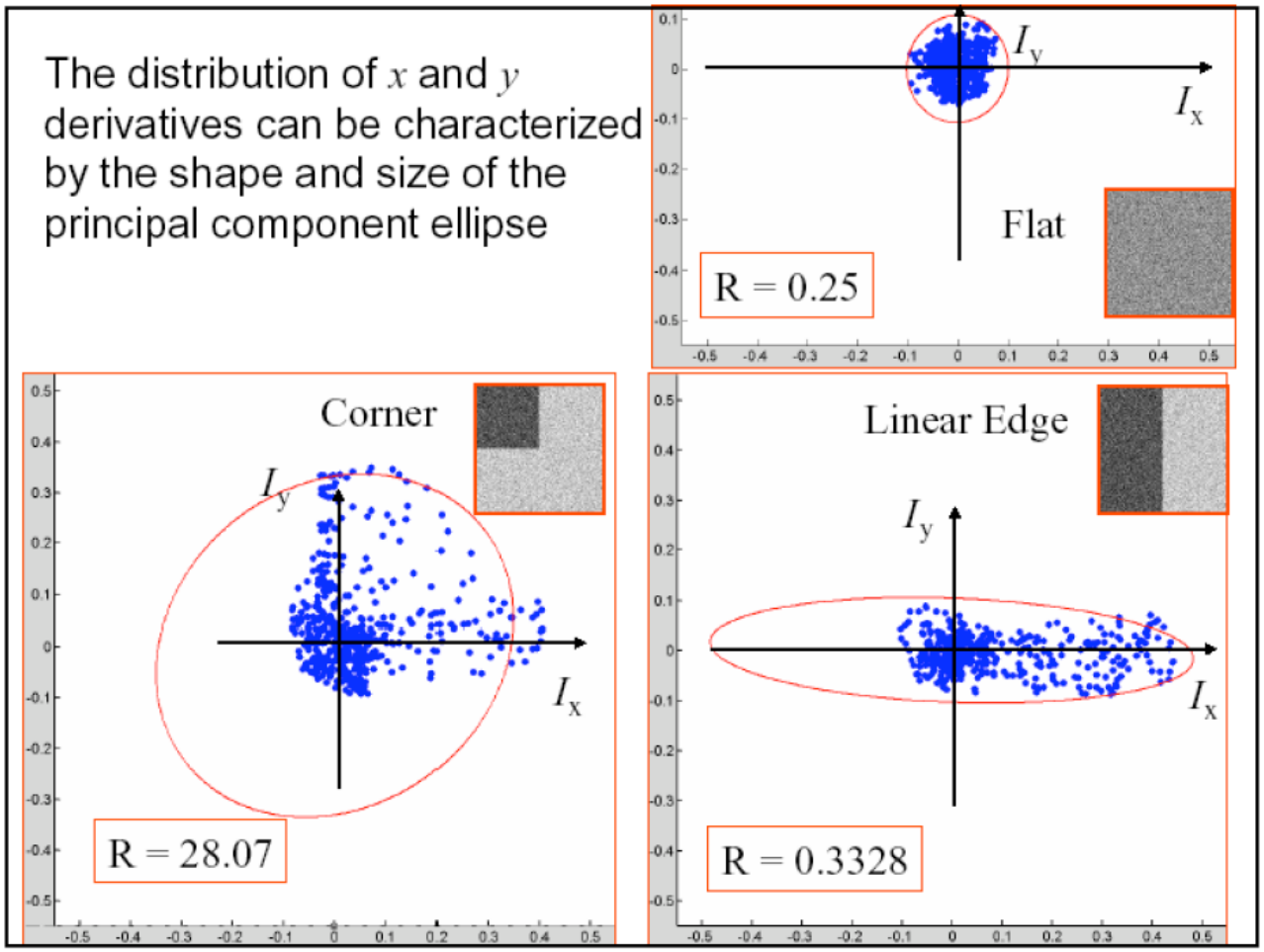


Distribution of f_x and f_y

The distribution of the x and y derivatives is very different for all three types of patches

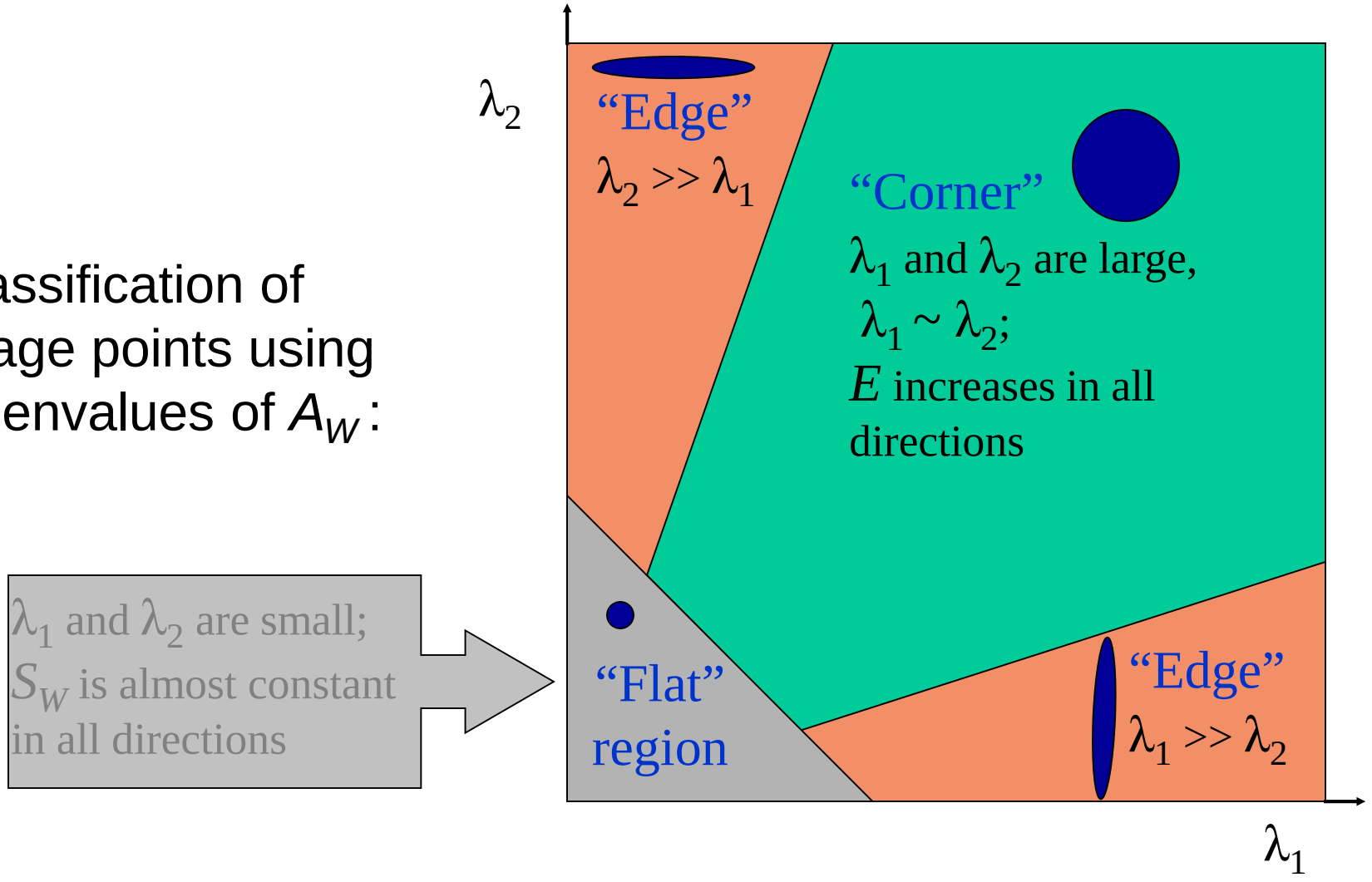


Distribution of f_x and f_y (cont'd)



Harris Detector (cont'd)

Classification of image points using eigenvalues of A_W:



Harris Detector (cont'd)

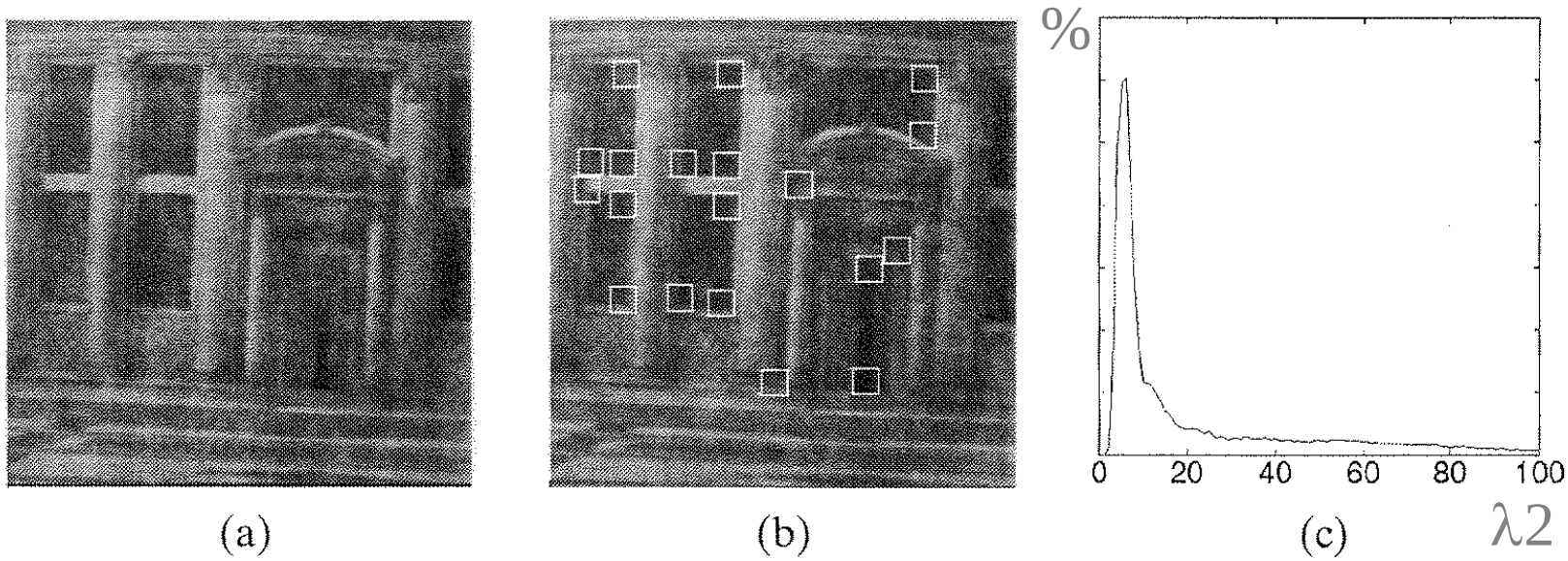


Figure 4.9 (a): original image of a building. (b): the 15 × 15 pixel neighbourhoods of some of the image points for which $\lambda_2 > 20$. (c): histogram of λ_2 values across the image.

(assuming that $\lambda_1 > \lambda_2$)

Harris Detector

Measure of corner response:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

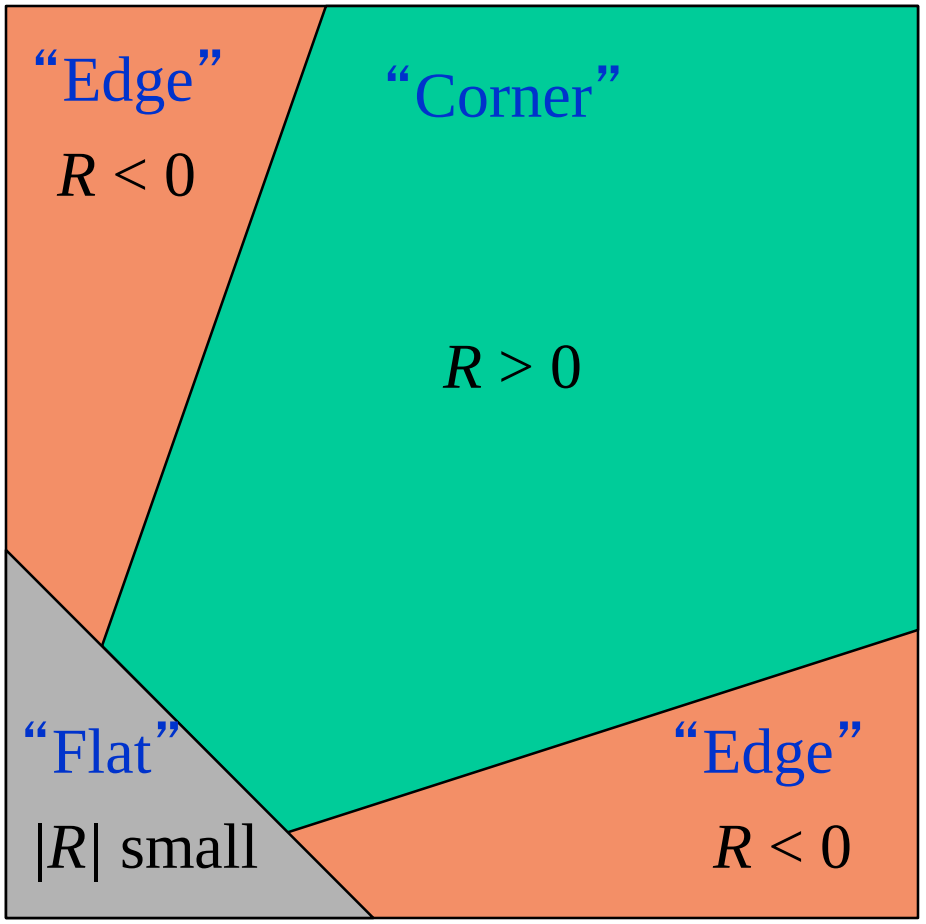
$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

(k – empirical constant, k = 0.04-0.06)

(Shi-Tomasi variation: use min(λ_1,λ_2) instead of R)

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Harris Detector

- The Algorithm:
  - Find points with large corner response function R ($R >$ threshold)
  - Take the points of *local maxima* of R

Harris corner detector algorithm

- Compute image gradients I_x I_y for all pixels
- For each pixel
  - Compute

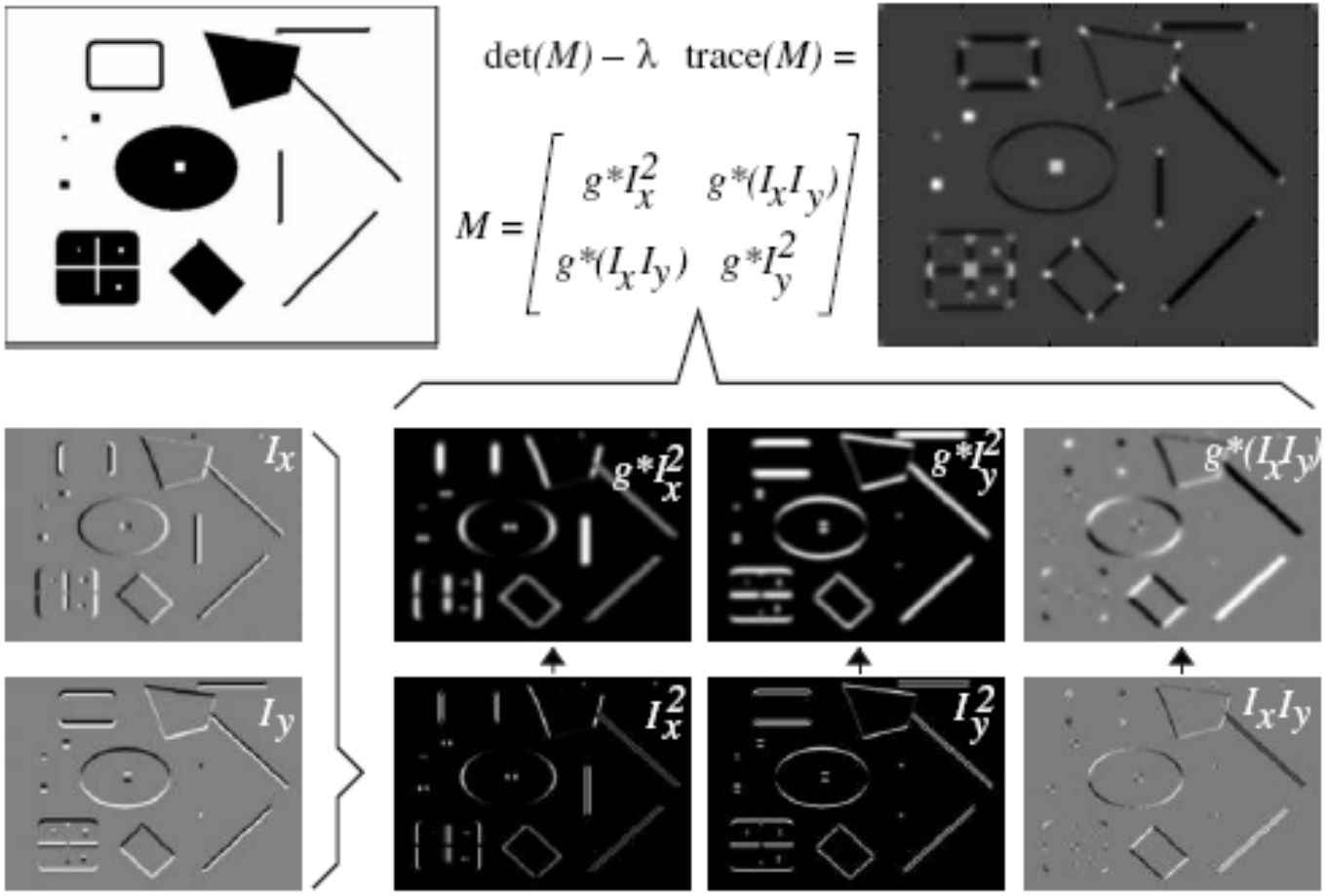
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

    by looping over neighbors x,y
  - compute

$$R = \det M - k\left(\text{trace } M\right)^2$$

- Find points with large corner response function R (R > threshold)
- Take the points of locally maximum R as the detected feature points (ie, pixels where R is bigger than for all the 4 or 8 neighbors).

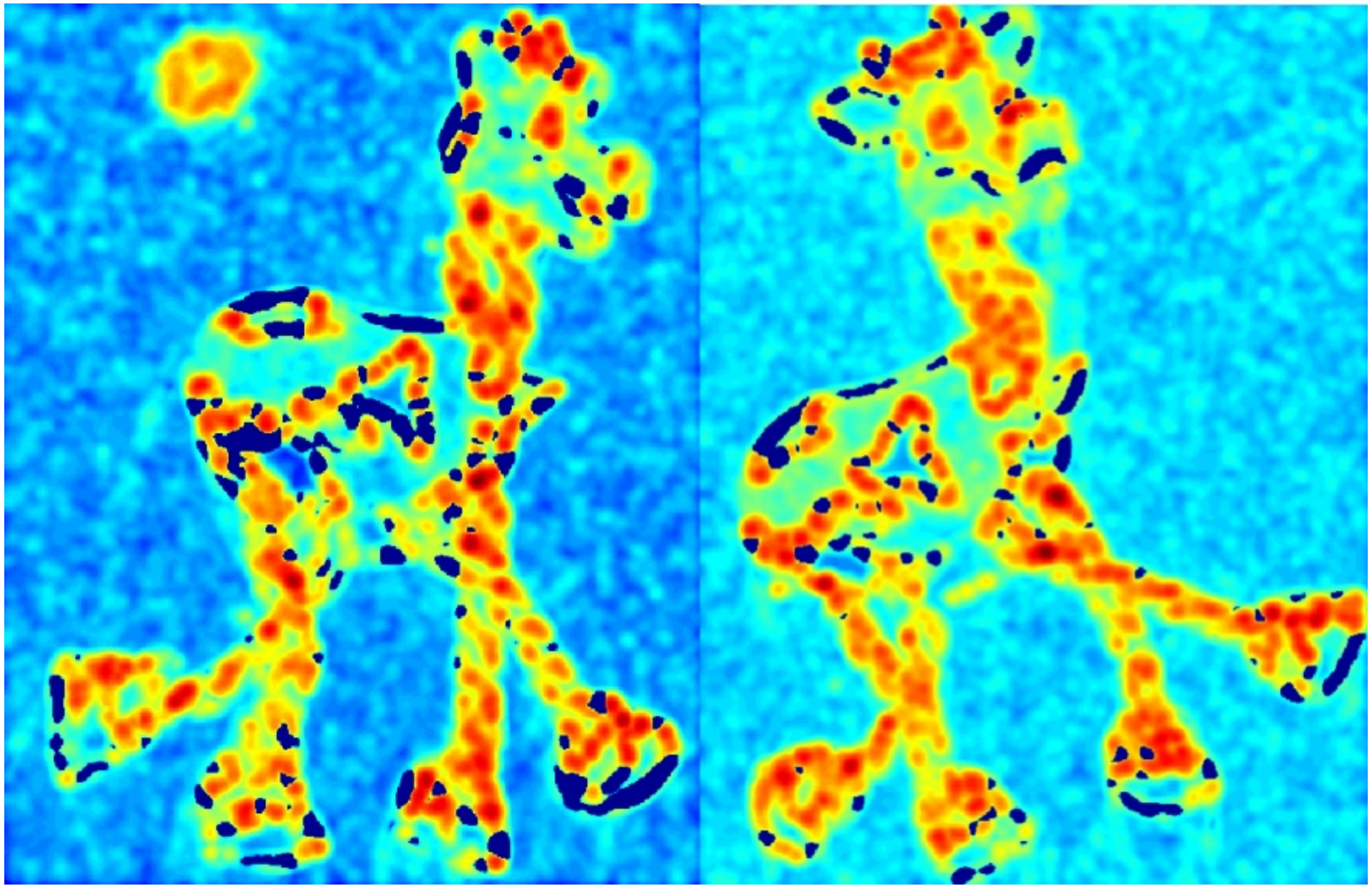
Harris Detector - Example



Harris Detector - Example

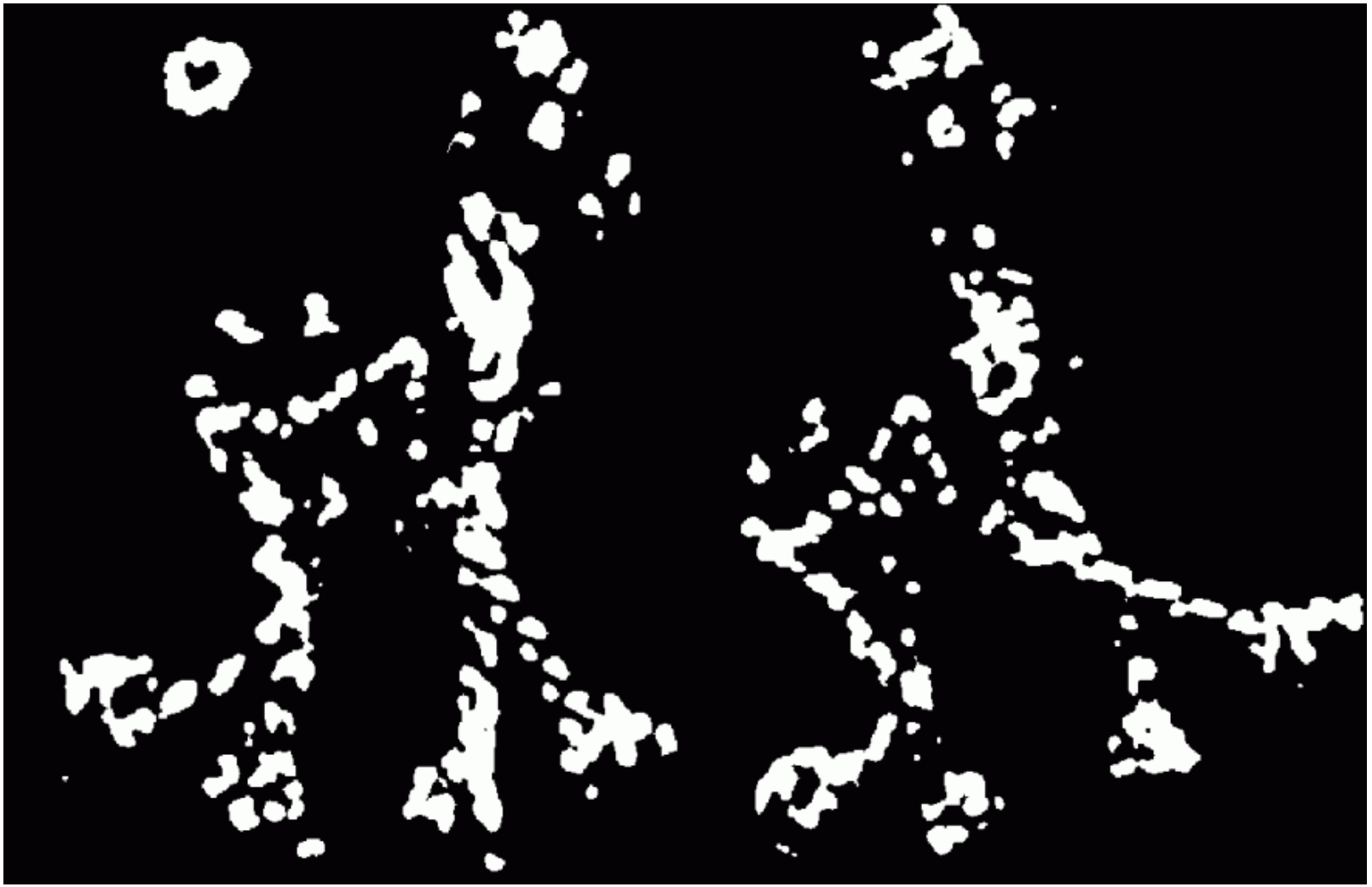
Harris Detector - Example

Compute corner response R



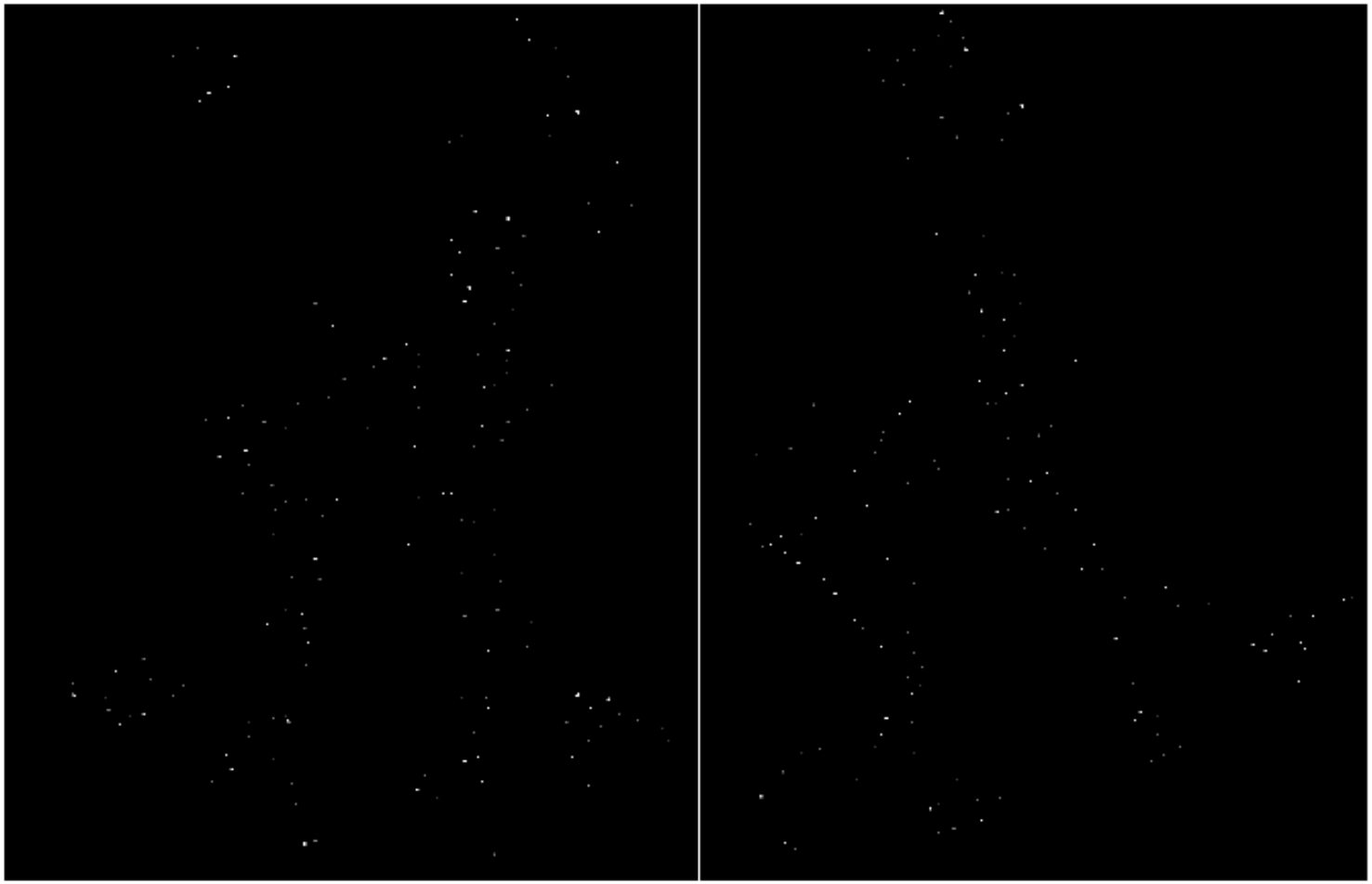
Harris Detector - Example

Find points with large corner response: R>threshold



Harris Detector - Example

Take only the points of local maxima of R



Harris Detector - Example

Map corners on the original image

Harris Detector – Scale Parameters

- The Harris detector requires two scale parameters:

  (i) a *differentiation scale* σ_D for smoothing prior to the computation of image derivatives,

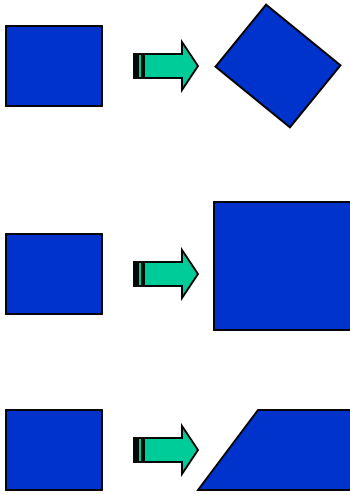
  &

  (ii) an *integration scale* σ_I for defining the size of the Gaussian window (i.e., integrating derivative responses). $A_W(x,y) \rightarrow A_W(x,y,\sigma_I,\sigma_D)$

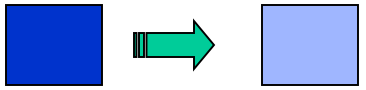
- Typically, $\sigma_I = \gamma\sigma_D$

Invariance to Geometric/Photometric Changes

- Is the Harris detector invariant to geometric and photometric changes?
- Geometric
  - Rotation
  - Scale
  - Affine

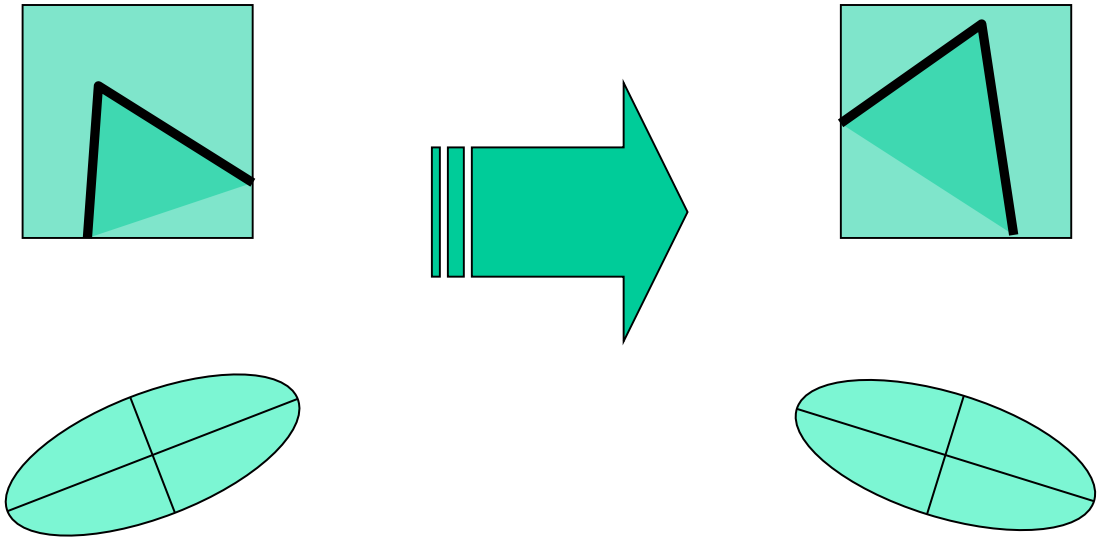


- Photometric
  - Affine intensity change: $I(x,y) \rightarrow a\, I(x,y) + b$



Harris Detector: Rotation Invariance

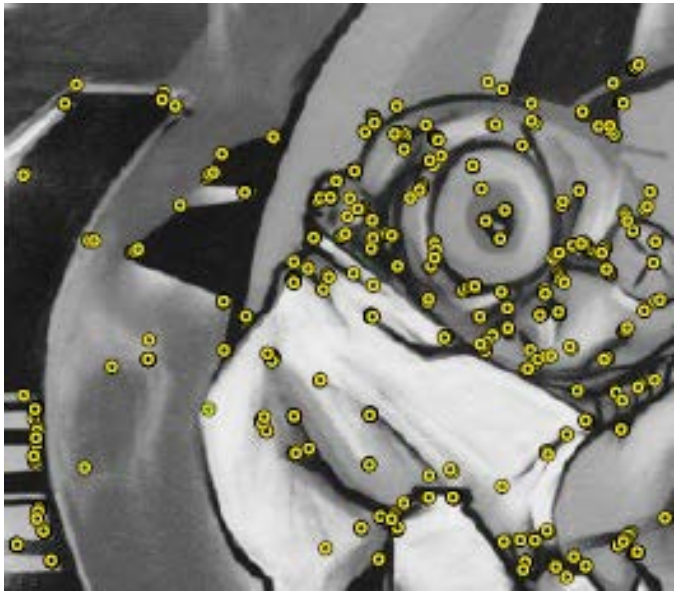
- Rotation

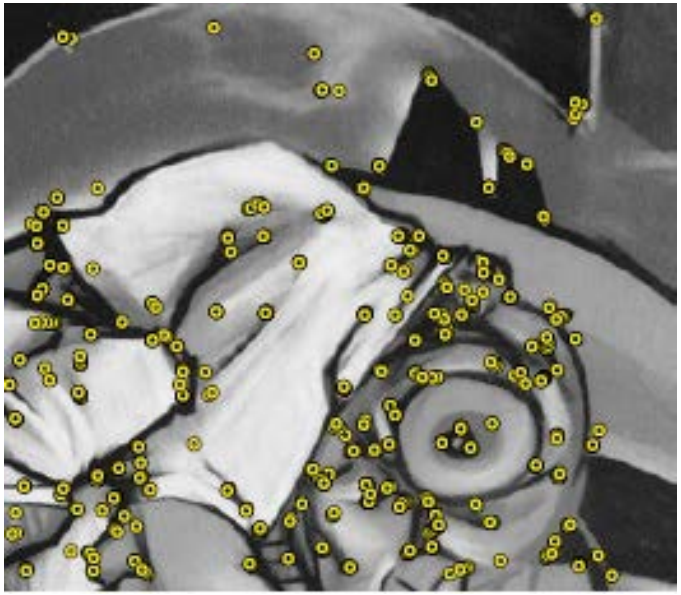


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

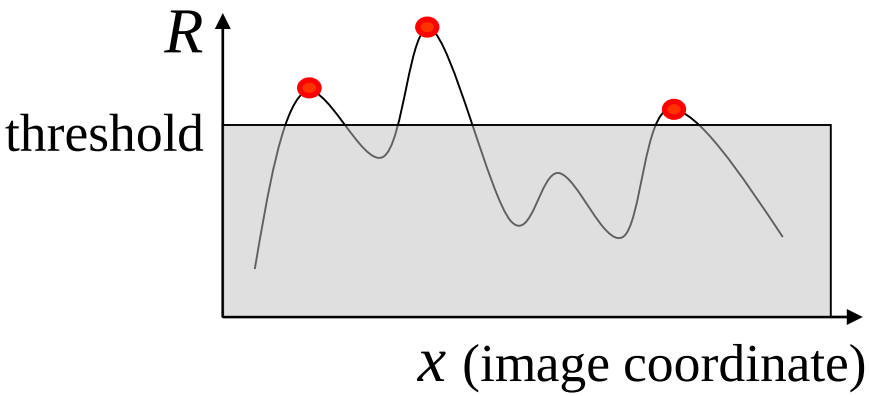
Harris Detector: Rotation Invariance (cont'd)

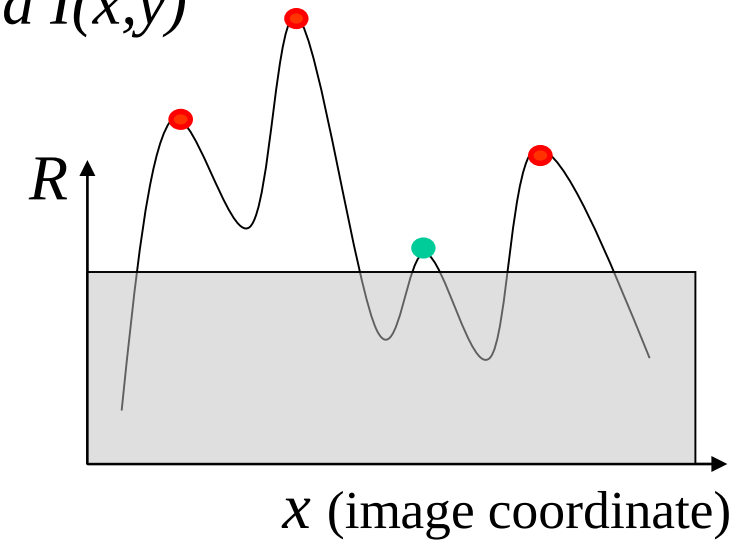




Harris Detector: Photometric Changes

- Affine intensity change
  - ✓ Only derivatives are used => invariance to intensity shift $I(x,y) \rightarrow I(x,y) + b$
  - ✓ Intensity scale: $I(x,y) \rightarrow a\, I(x,y)$

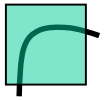


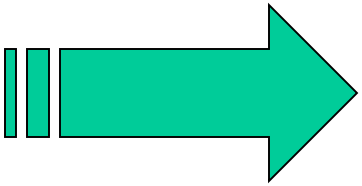


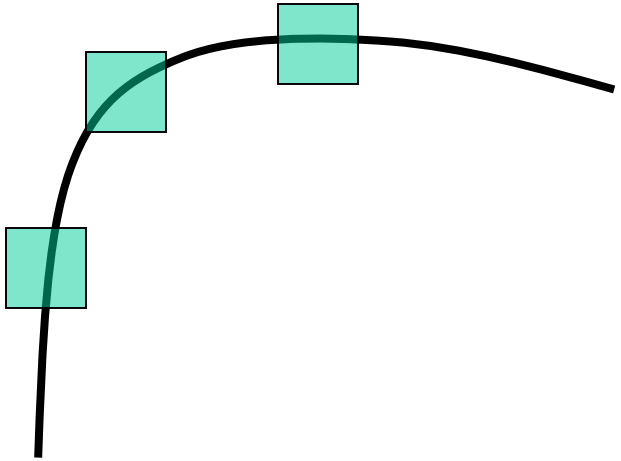
Partially invariant to affine intensity change

Harris Detector: Scale Invariance

- Scaling







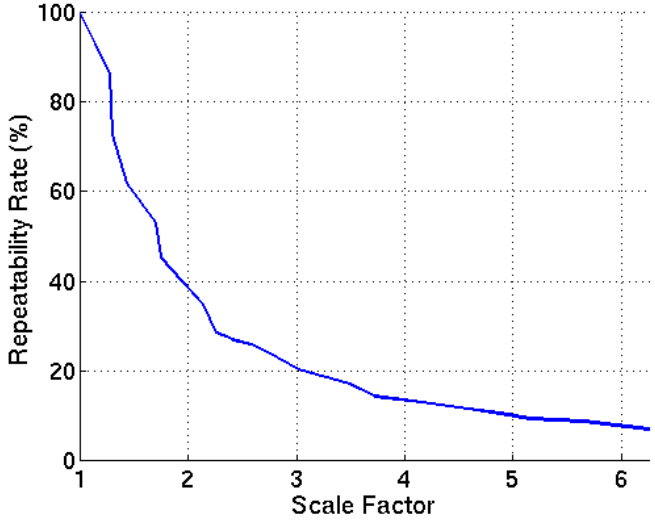
Corner

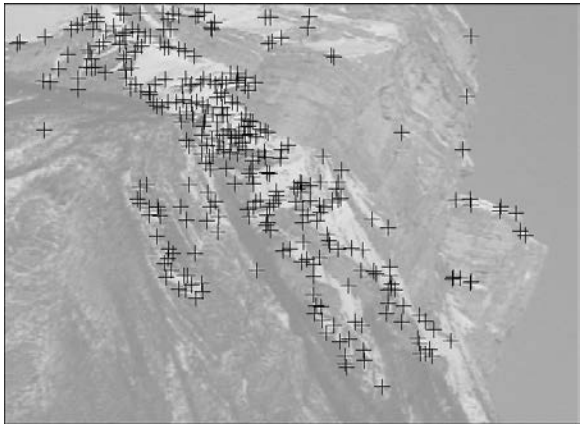
All points will be classified as edges

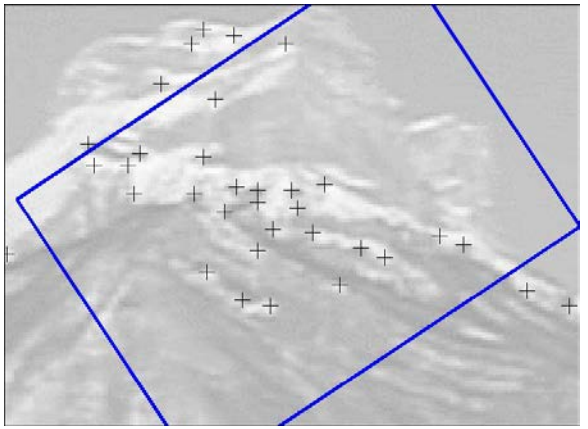
Not invariant to scaling (and affine transforms)

Harris Detector: Disadvantages

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







How to handle scale changes?

- A_W must be adapted to scale changes.
- If the scale change is known, we can adapt the Harris detector to the scale change (i.e., set properly σ_I , σ_D).
- What if the scale change is unknown?

Multi-scale Harris Detector

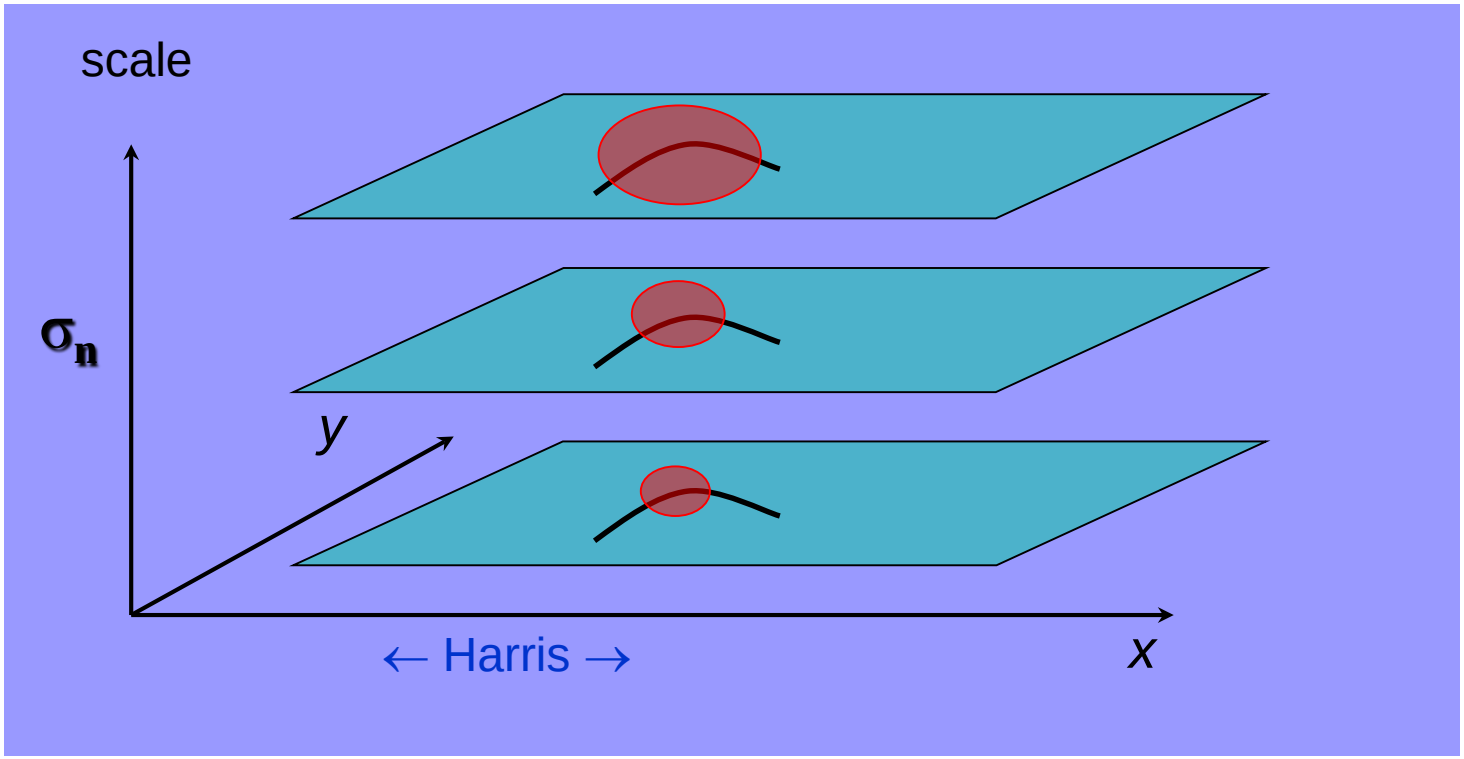
- Detects interest points at varying scales.

$$\mathbf{R(A_W) = det(A_W(x,y,\sigma_I,\sigma_D)) - \alpha\ trace^2(A_W(x,y,\sigma_I,\sigma_D))}$$

$\sigma_n = k^n \sigma$

$\sigma_D = \sigma_n$

$\sigma_I = \gamma \sigma_D$

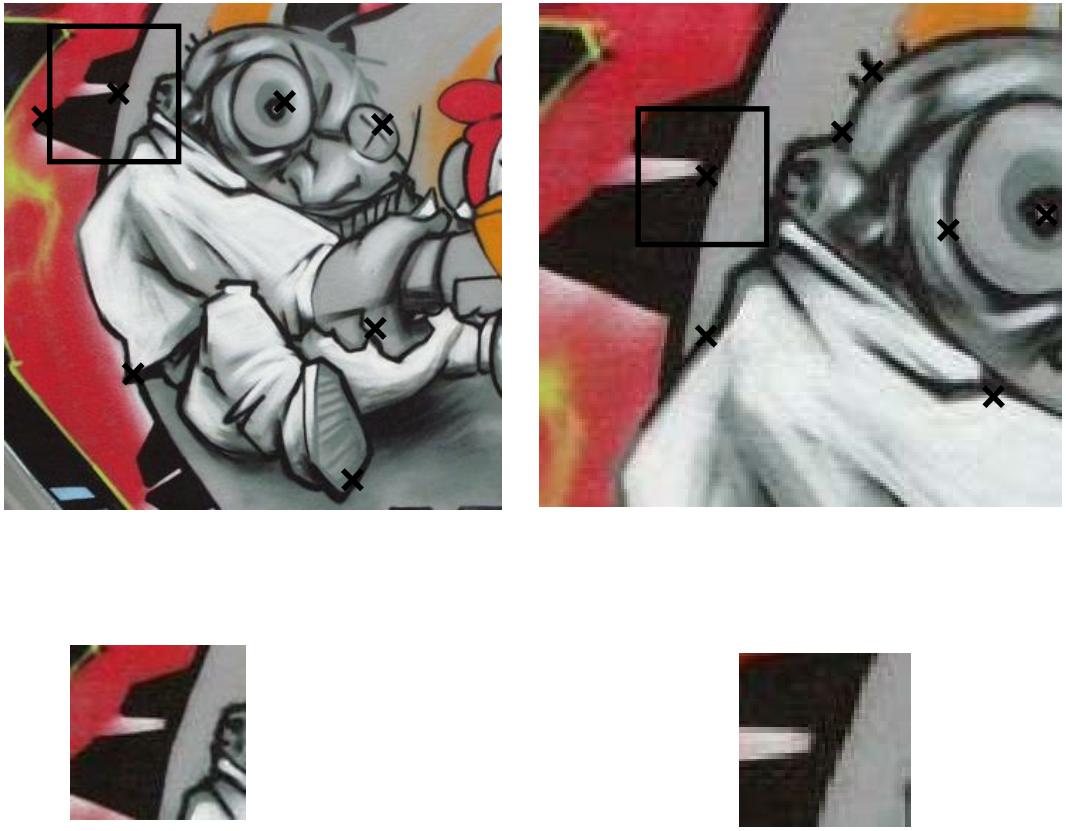


How to cope with transformations?

- Exhaustive search
- Invariance
- Robustness

Exhaustive search

- Multi-scale approach



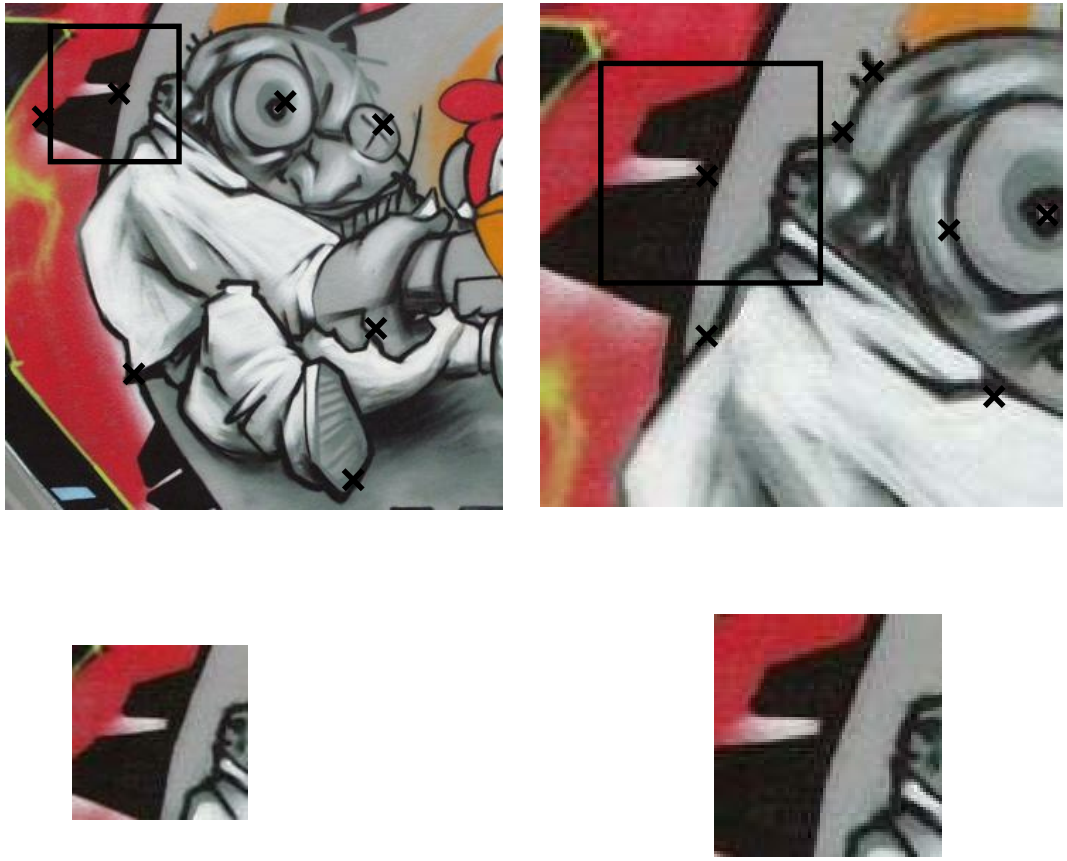
Slide from T. Tuytelaars ECCV 2006 tutorial

Exhaustive search

- Multi-scale approach

Exhaustive search

- Multi-scale approach

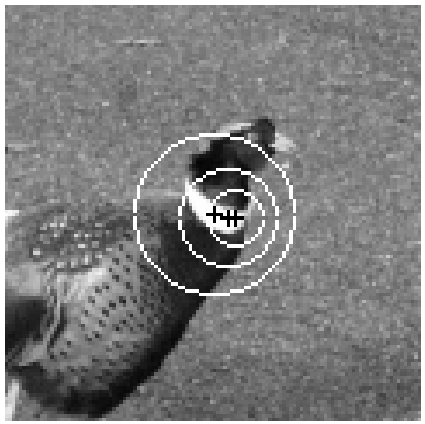


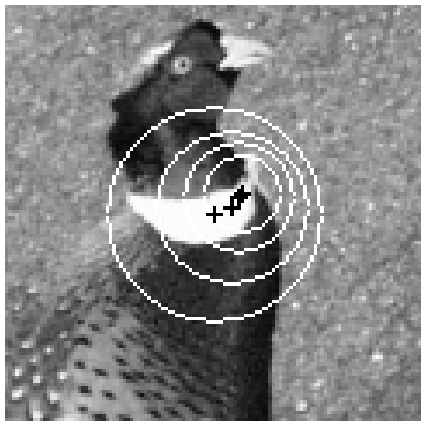
Exhaustive search

- Multi-scale approach

How to handle scale changes?

- Not a good idea!
  - There will be many points representing the same structure, complicating matching!
  - Note that point locations shift as scale increases.

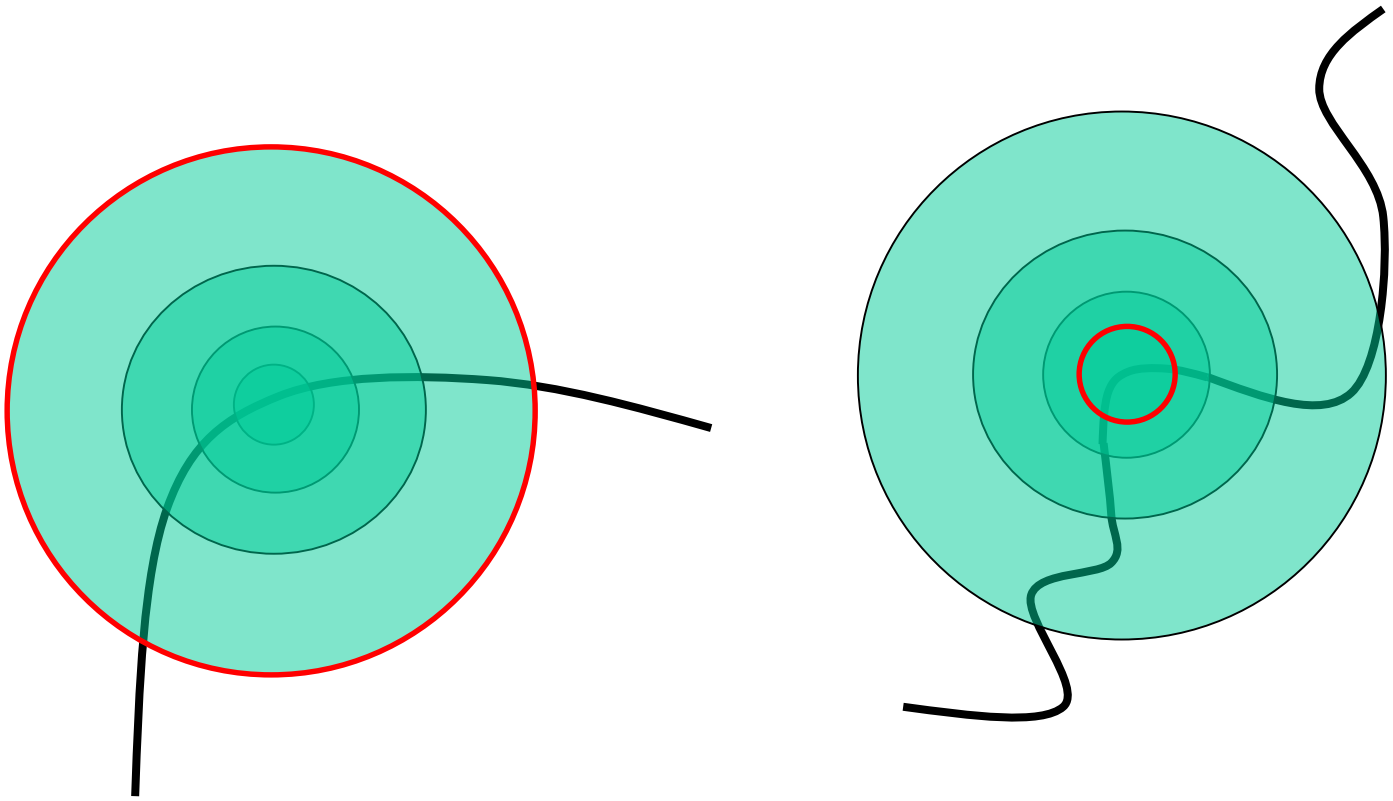




The size of the circle corresponds to the scale at which the point was detected

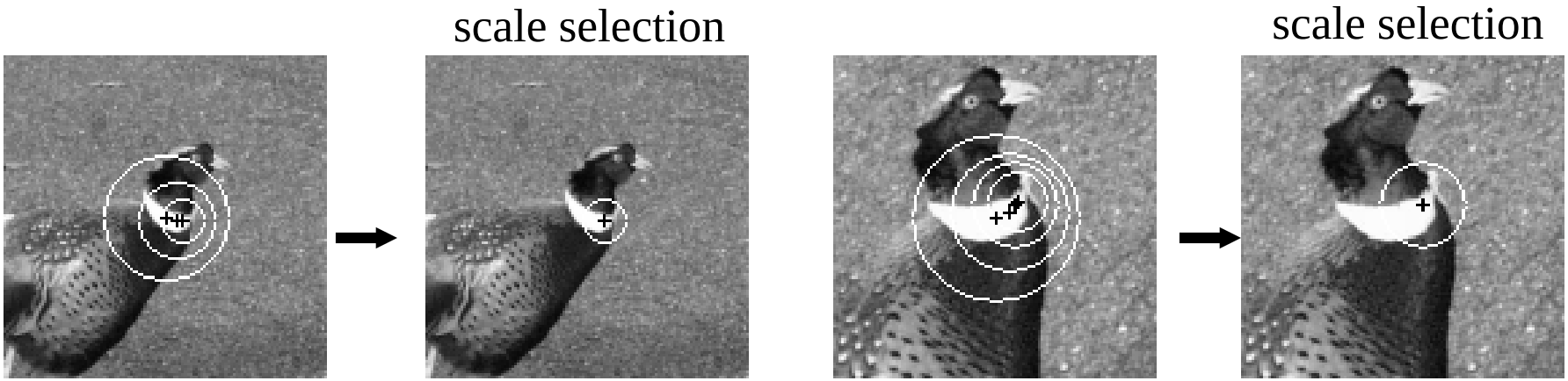
How to handle scale changes? (cont'd)

- How do we choose corresponding circles ***independently*** in each image?



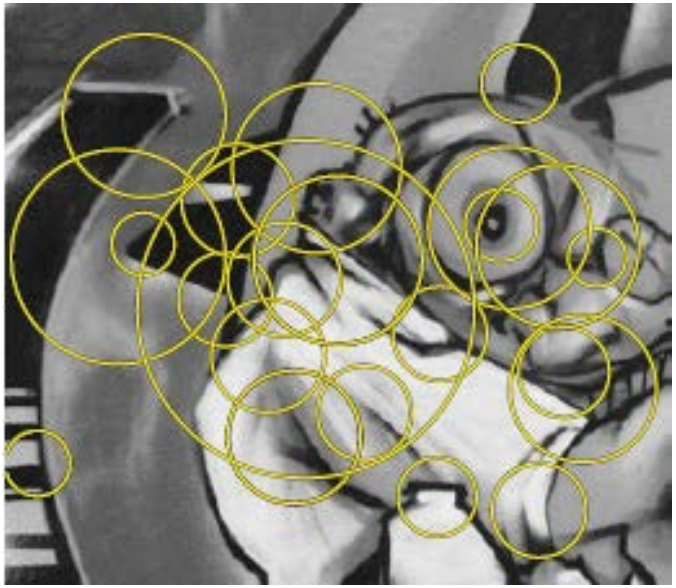
How to handle scale changes? (cont'd)

- Alternatively, use **scale selection** to find the characteristic scale of each feature.
- Characteristic scale depends on the feature's spatial extent (i.e., local neighborhood of pixels).



How to handle scale changes?

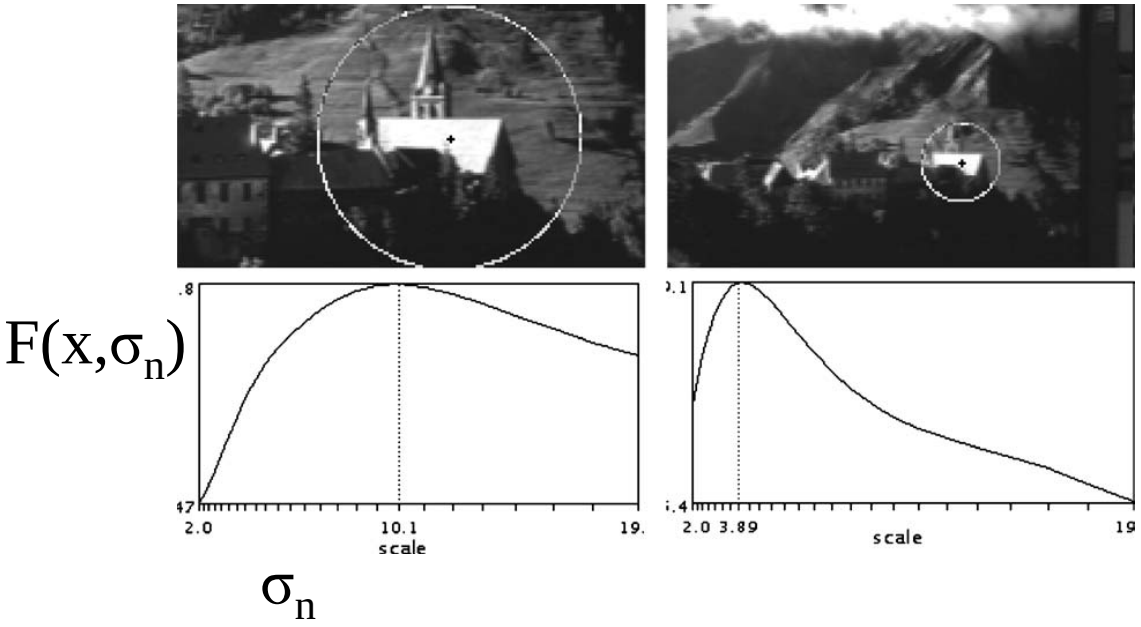
- Only a subset of the points computed in scale space are selected!



The size of the circles corresponds to the scale at which the point was selected.

Automatic Scale Selection

- Design a function $F(x,\sigma_n)$ which provides some local measure.
- Select points at which $F(x,\sigma_n)$ is maximal over σ_n.

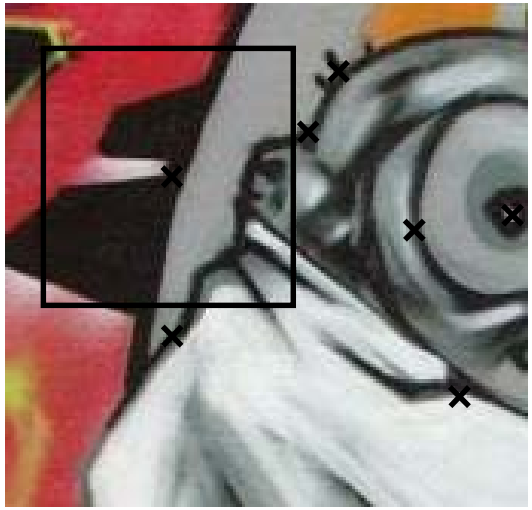


max of $F(x,\sigma_n)$ corresponds to characteristic scale!

T. Lindeberg, "Feature detection with automatic scale selection" *International Journal of Computer Vision,* vol. 30, no. 2, pp 77-116, 1998.

Invariance

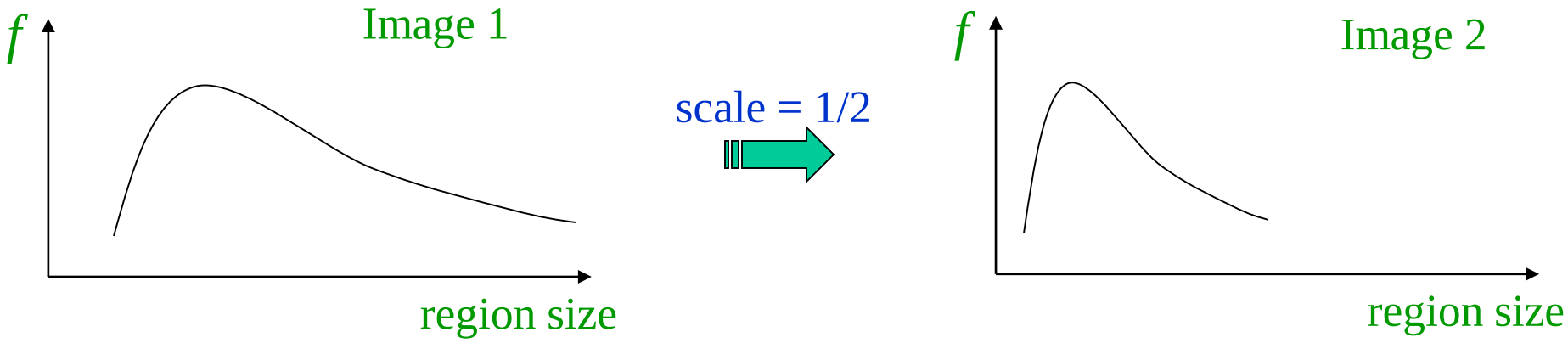
- Extract patch from each image individually



Automatic scale selection

- Solution:
  - Design a function on the region, which is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)

    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 width)



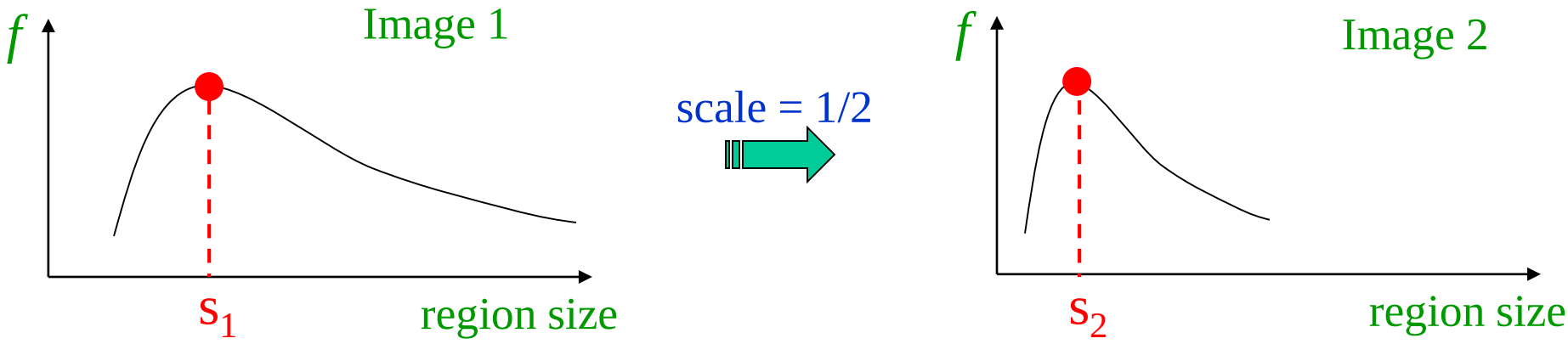
Automatic scale selection

- Common approach:

  Take a local maximum of this function

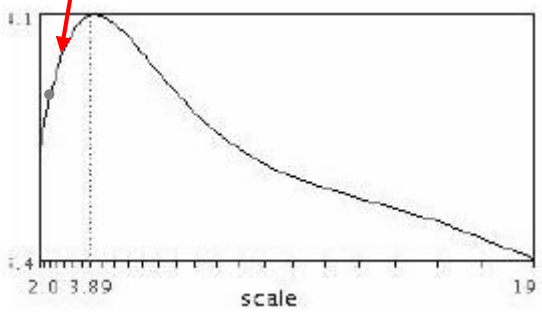
  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

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

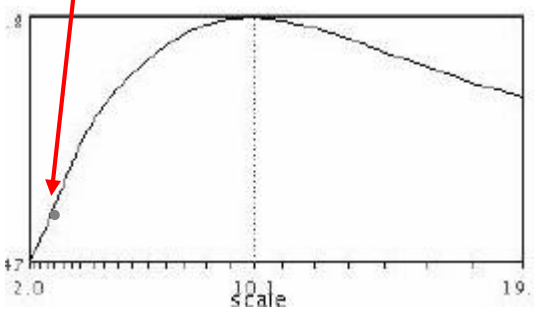


Automatic Scale Selection

- Function responses for increasing scale (scale signature)



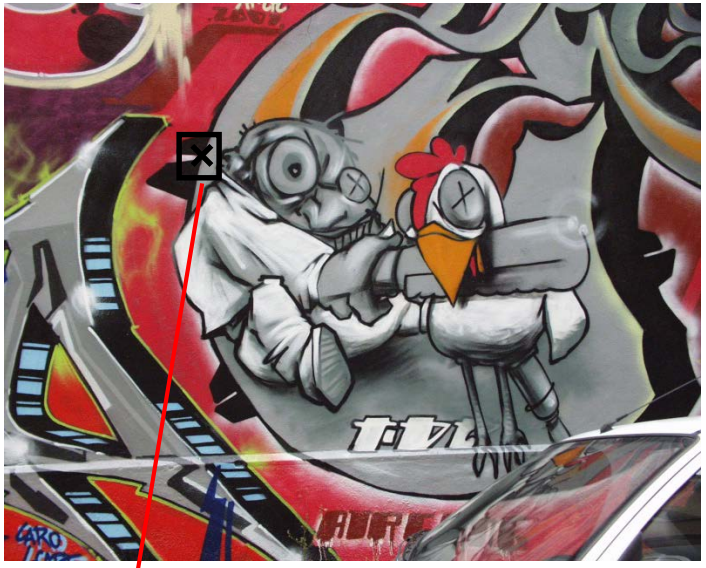
$f(I_{i_1 \ldots i_m}(x,\sigma))$

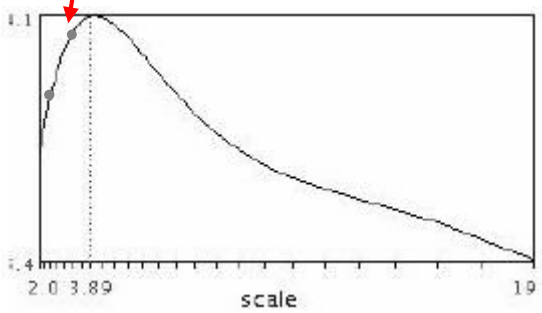


$f(I_{i_1 \ldots i_m}(x',\sigma))$

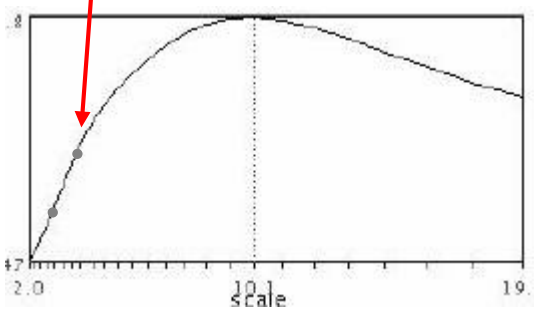
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





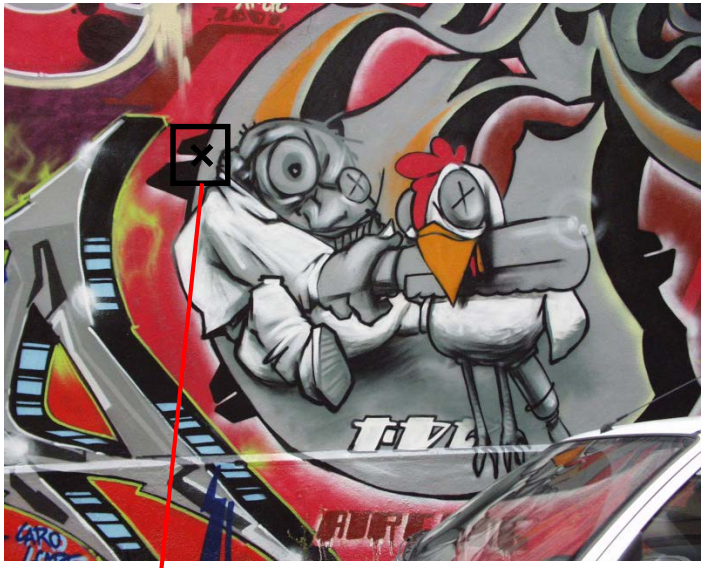
$f(I_{i_1 \ldots i_m}(x,\sigma))$

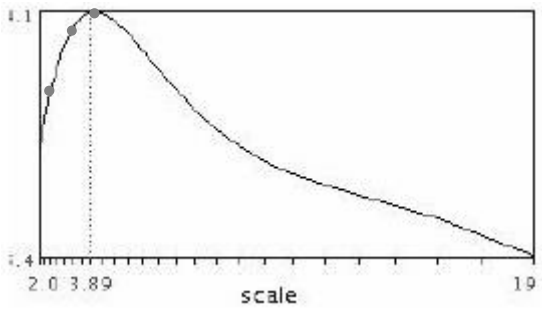


$f(I_{i_1 \ldots i_m}(x',\sigma))$

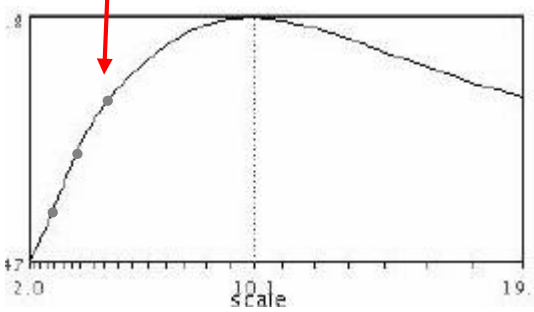
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





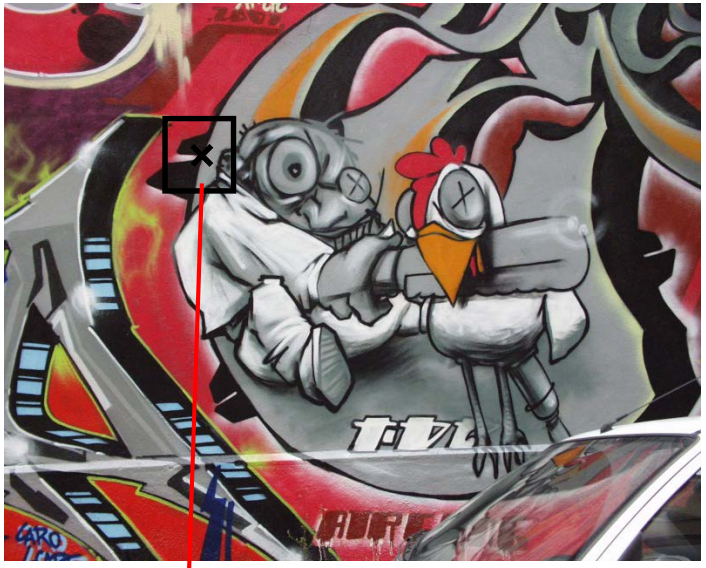
$f(I_{i_1 \ldots i_m}(x,\sigma))$

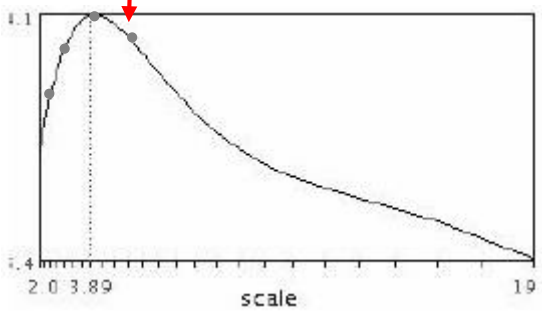


$f(I_{i_1 \ldots i_m}(x',\sigma))$

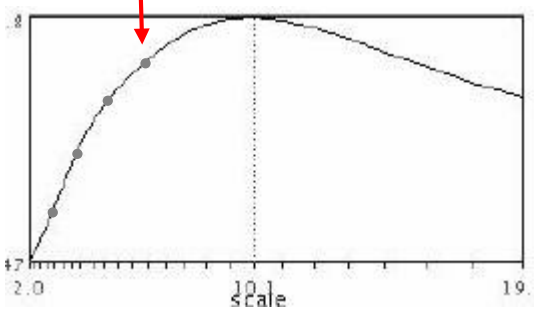
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





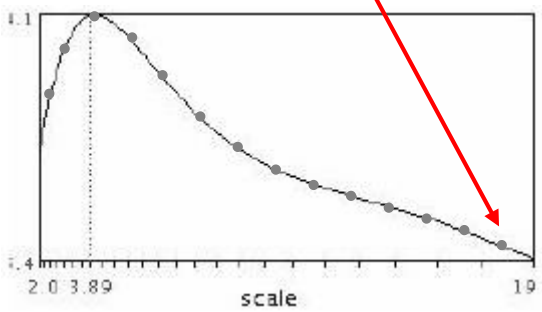
$f(I_{i_1 \ldots i_m}(x,\sigma))$



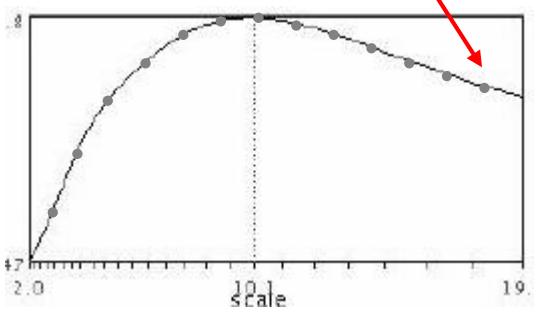
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



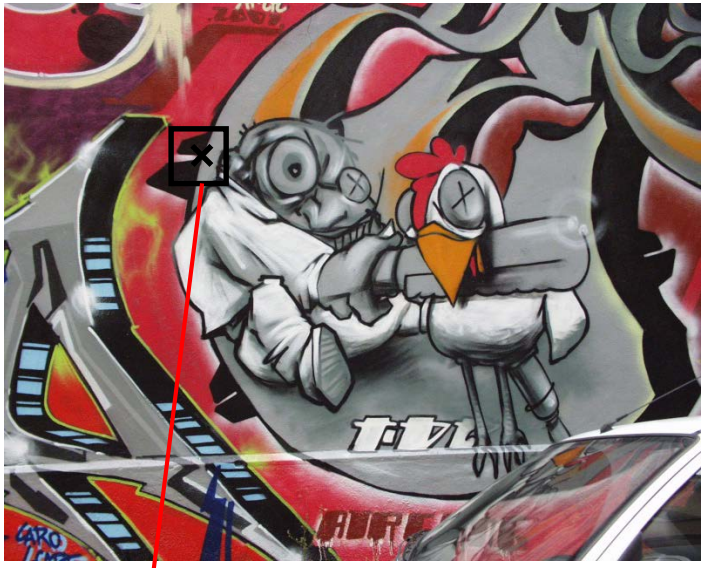
$f(I_{i_1 \ldots i_m}(x,\sigma))$

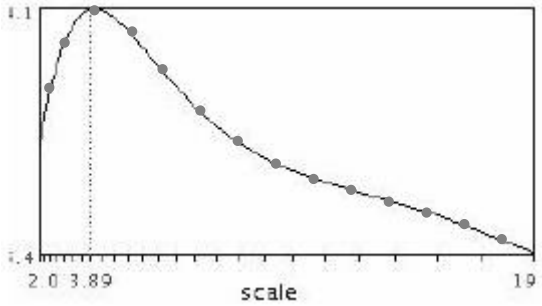


$f(I_{i_1 \ldots i_m}(x',\sigma))$

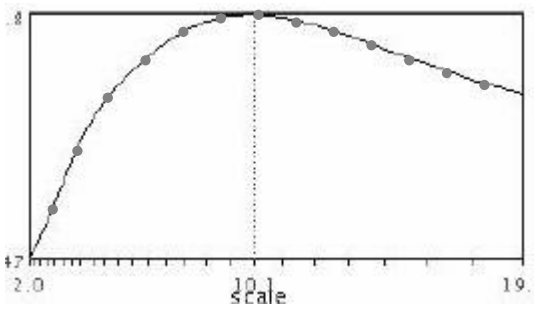
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





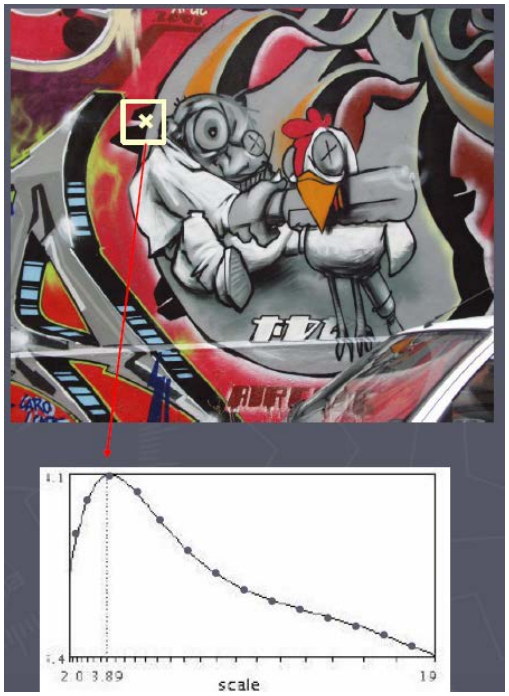
$f(I_{i_1 \ldots i_m}(x,\sigma))$

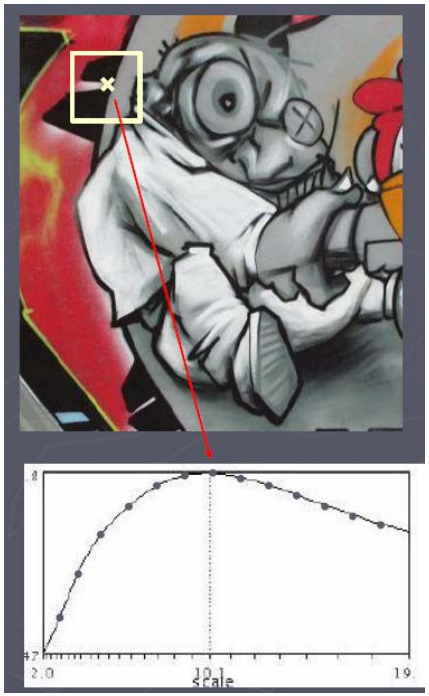


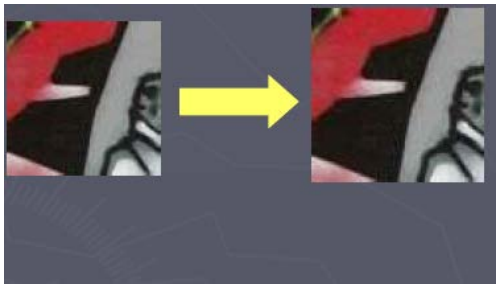
$f(I_{i_1 \ldots i_m}(x',\sigma'))$

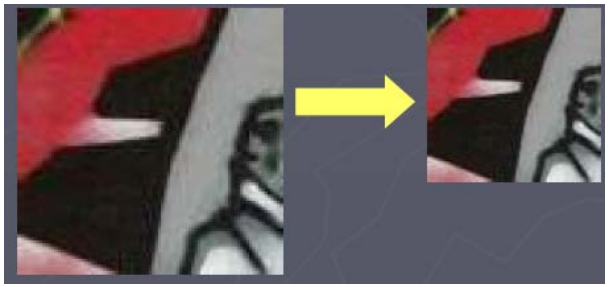
Scale selection

- Use the scale determined by detector to compute descriptor in a normalized frame







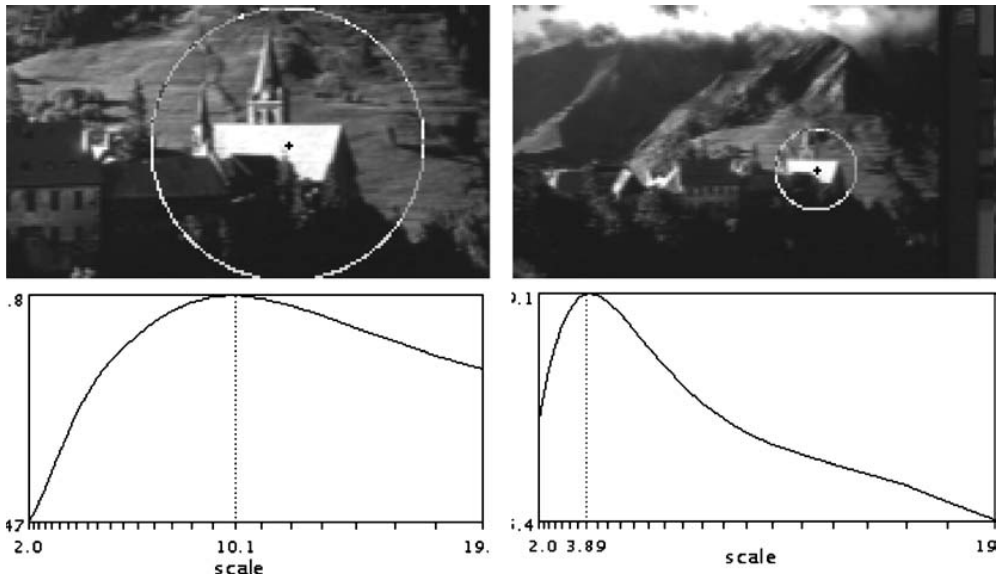


Automatic Scale Selection (cont'd)

- Characteristic scale is relatively independent of the image scale.

- The ratio of the scale values corresponding to the max values, is equal to the scale factor between the images.

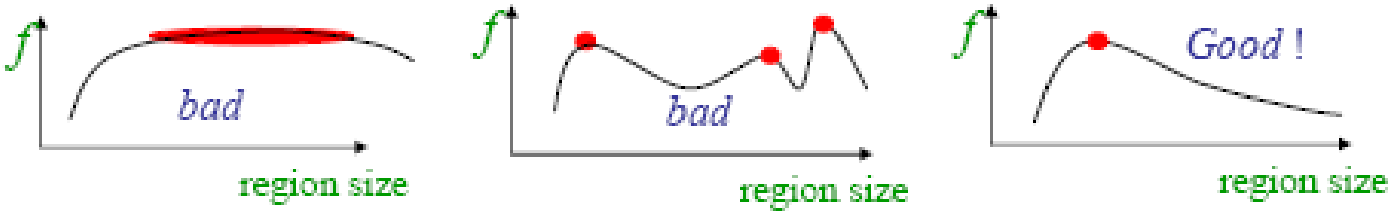
Scale factor: 2.5



Scale selection allows for finding spatial extend that is ***covariant*** with the image transformation.

Automatic Scale Selection

- What local measures should we use?
  - Should be rotation invariant
  - Should have one stable sharp peak



How should we choose $F(x,\sigma_n)$?

- Typically, $F(x,\sigma_n)$ is defined using derivatives, e.g.:

$$Square\ gradient:\ \sigma^2(L_x^2(x,\sigma)+L_y^2(x,\sigma))$$

$$LoG:|\sigma^2(L_{xx}(x,\sigma)+L_{yy}(x,\sigma))|$$

$$DoG:|I(x)*G(\sigma_{n-1})-I(x)*G(\sigma_n)|$$

$$Harris\ function:\det(A_W)-\alpha\,trace^2(A_W)$$

- LoG yielded best results in a evaluation study; DoG was second best.

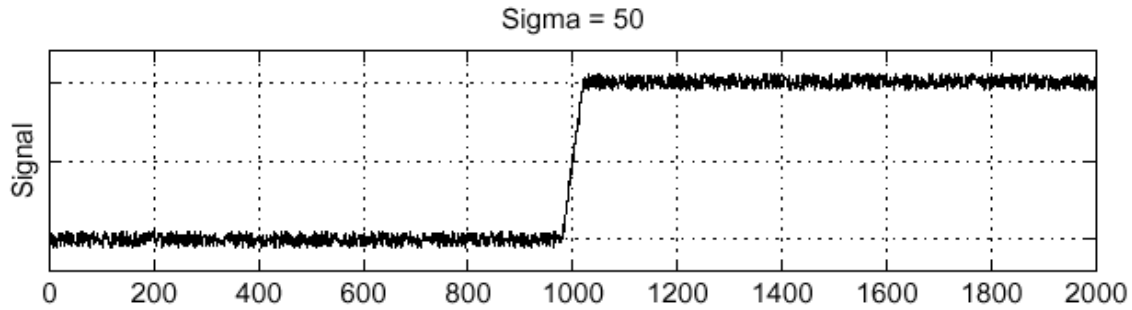
C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors", *International Journal of Computer Vision*, 37(2), pp. 151-172, 2000.

How should we choose $F(x,\sigma_n)$?

- Let's see how LoG responds at blobs …

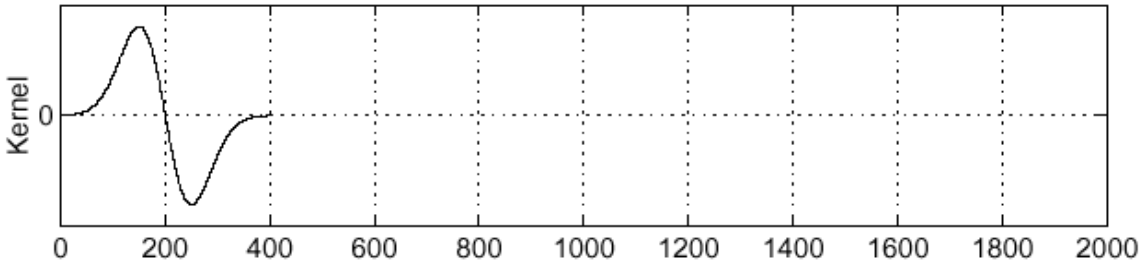
Recall: Edge detection Using 1st derivative

f



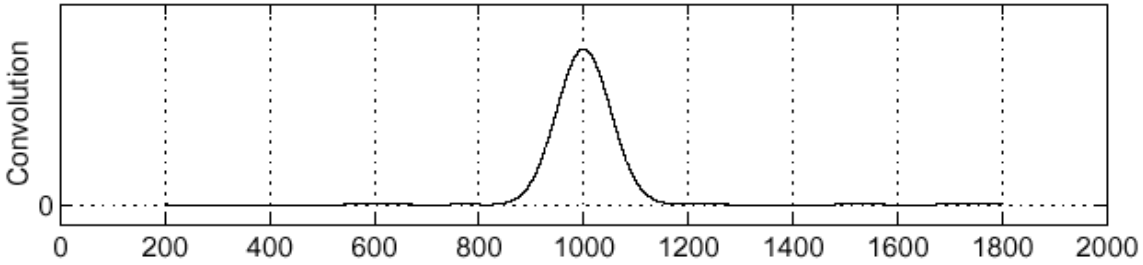
Edge

$\frac{d}{dx} g$



Derivative of Gaussian

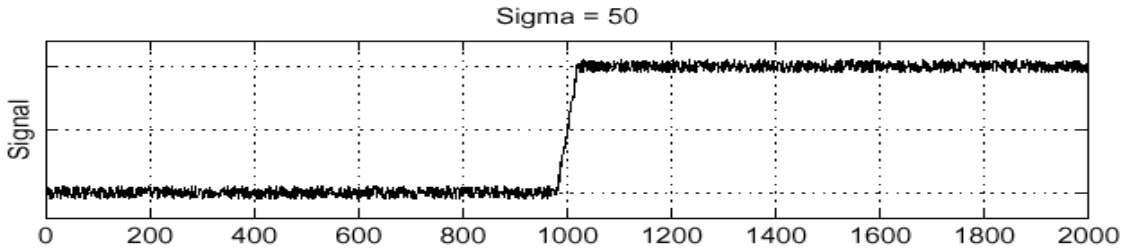
$f * \frac{d}{dx} g$



Edge = maximum of derivative

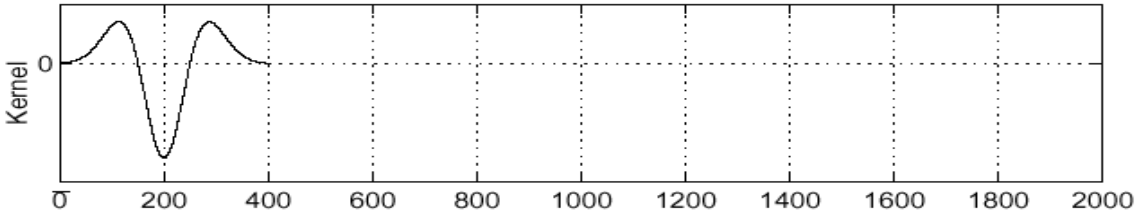
Recall: Edge detection Using 2nd derivative

f



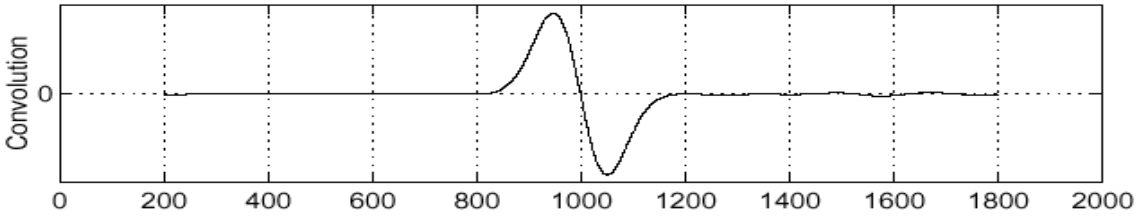
Edge

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



Second derivative of Gaussian (Laplacian)

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

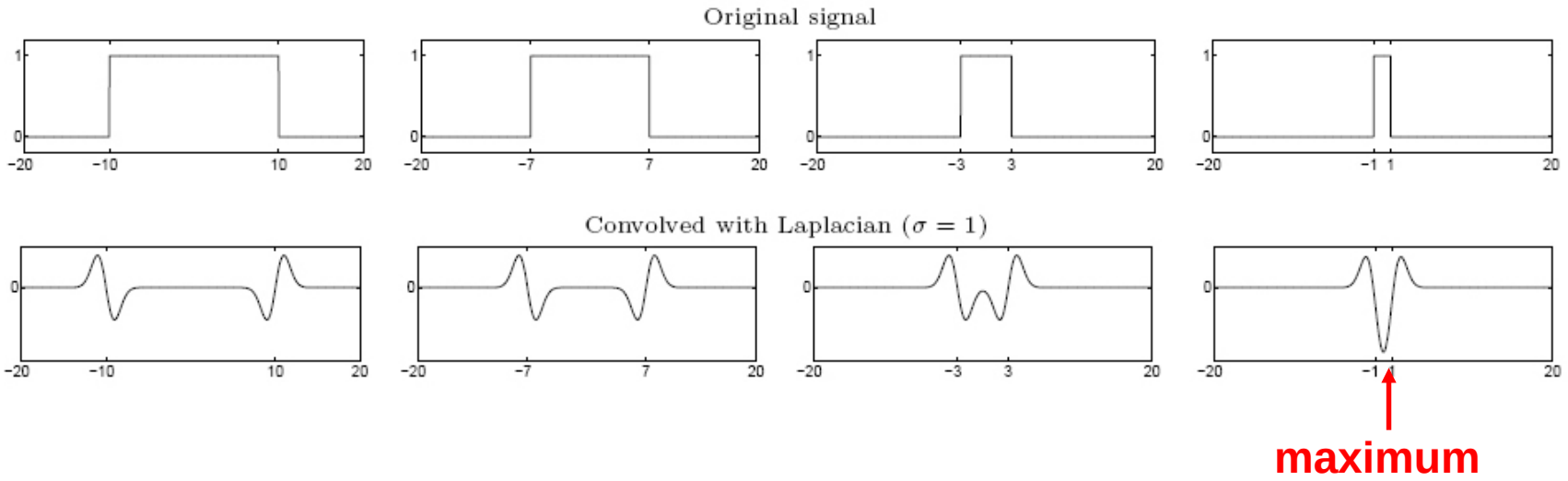


Edge = zero crossing of second derivative

From edges to blobs (i.e., small regions)

- Blob = superposition of two edges

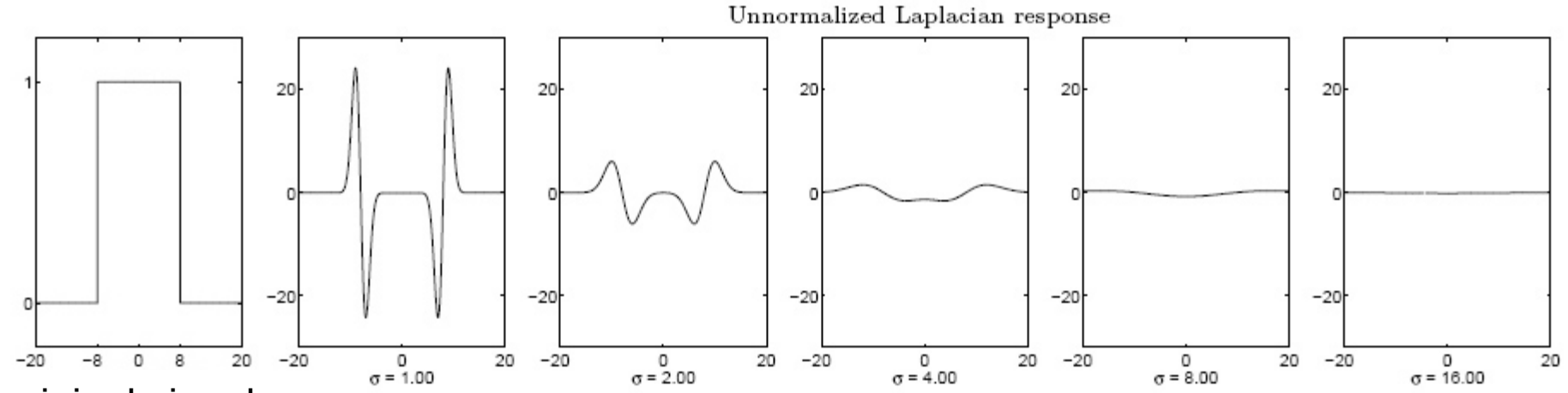
(blobs of different spatial extent)



Spatial selection: the magnitude of the Laplacian response will achieve a maximum (absolute value) at the center of he blob, provided the scale of the Laplacian is "matched" to the scale of the blob (e.g, spatial extent)

How could we find the spatial extent of a 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.



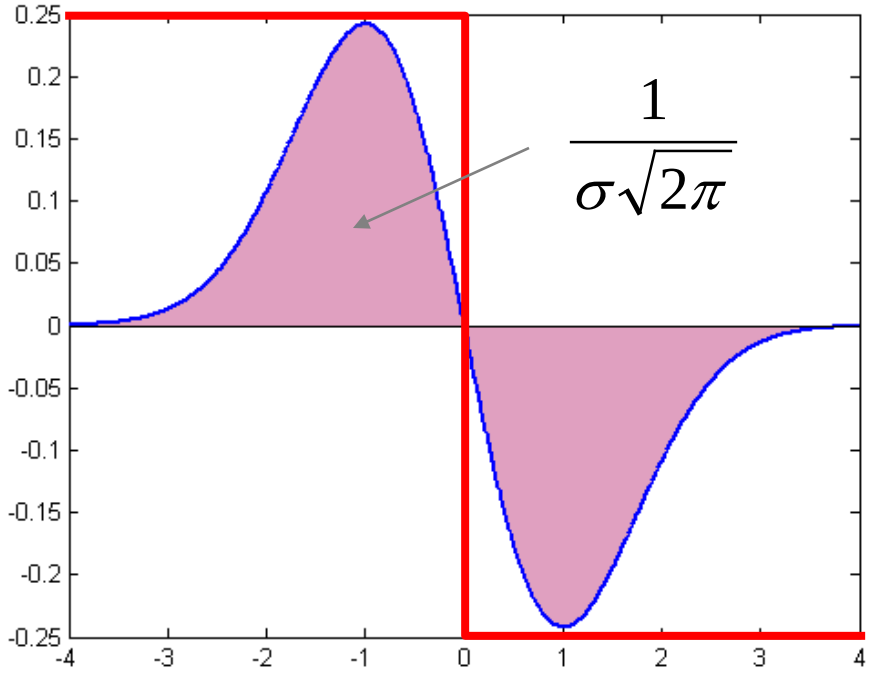
original signal (radius=8)

increasing σ →

Why does this happen?

Scale normalization

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



Scale normalization (cont'd)

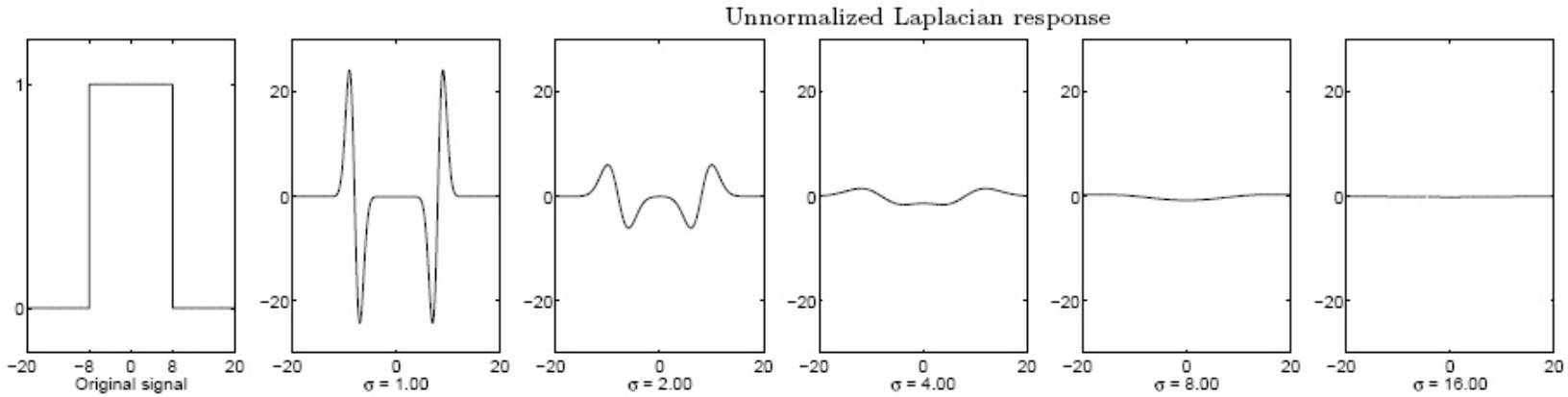
- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ

- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

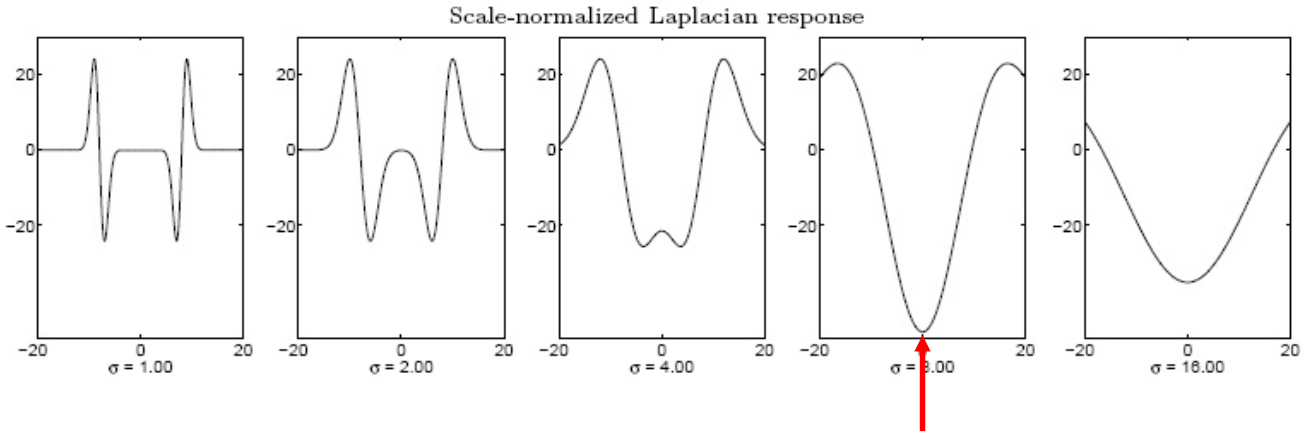
Effect of scale normalization

Original signal

Unnormalized Laplacian response

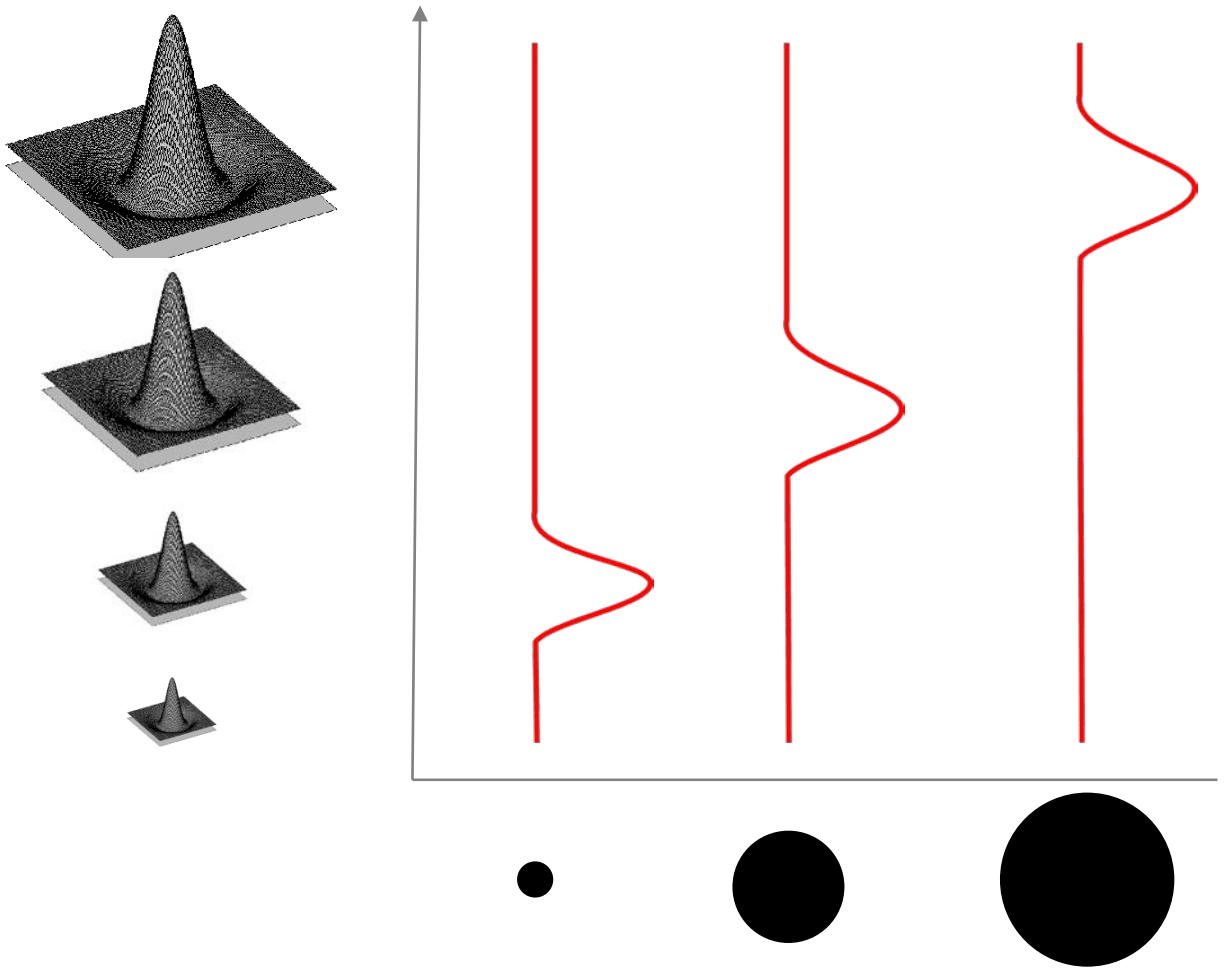


Scale-normalized Laplacian response (x σ^2)



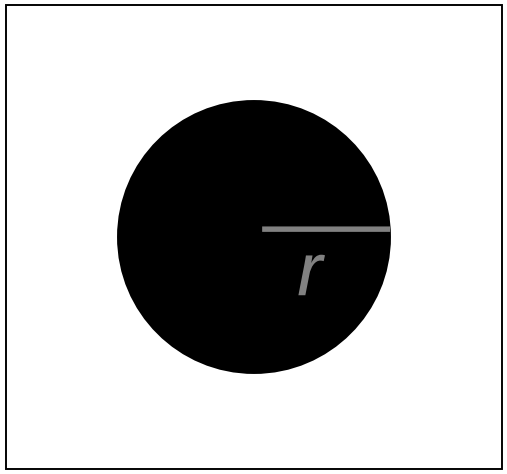
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector

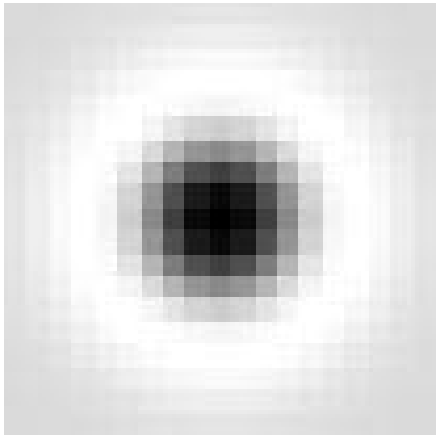


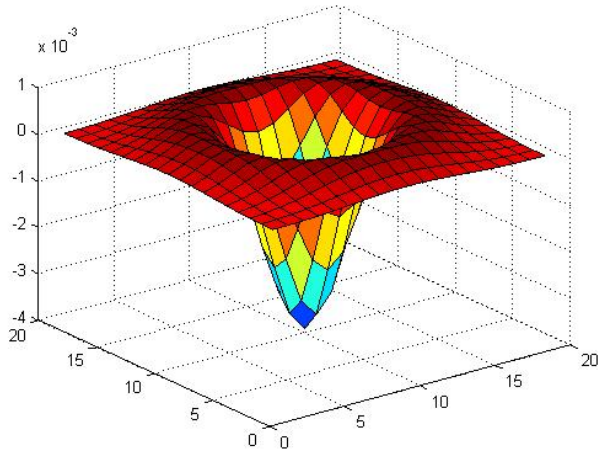
Scale selection: case of circle

- At what scale does the Laplacian achieve a maximum response for a binary circle of radius r?



image



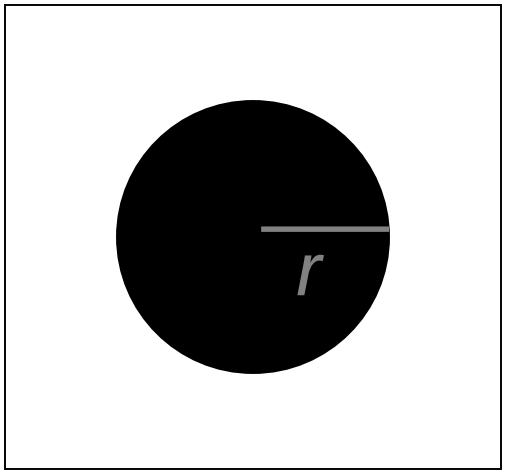


Laplacian

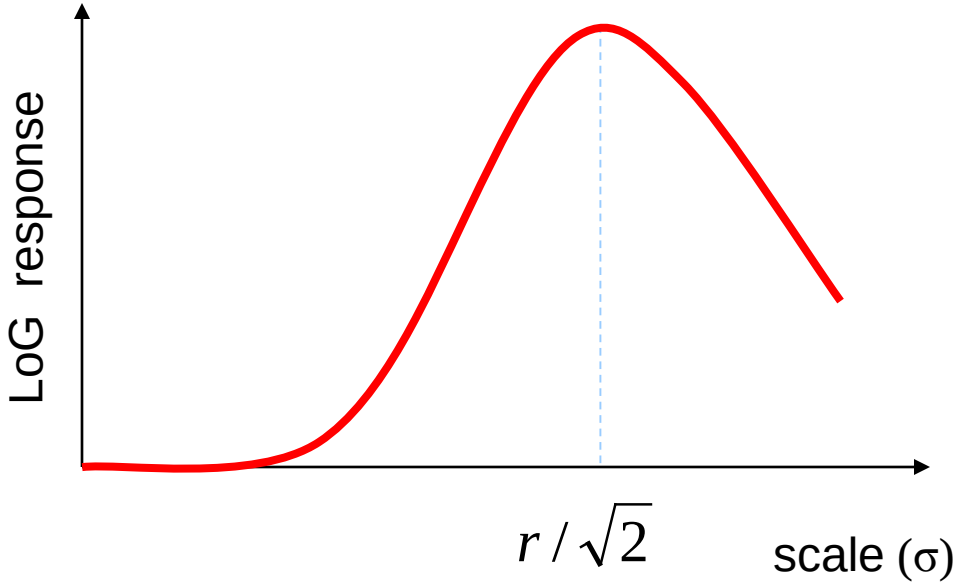
Scale selection: case of circle

- At what scale does the LoG achieve a maximum response for a binary circle of radius r?

- LoG is maximized at $\sigma = r / \sqrt{2}$ (characteristic scale)

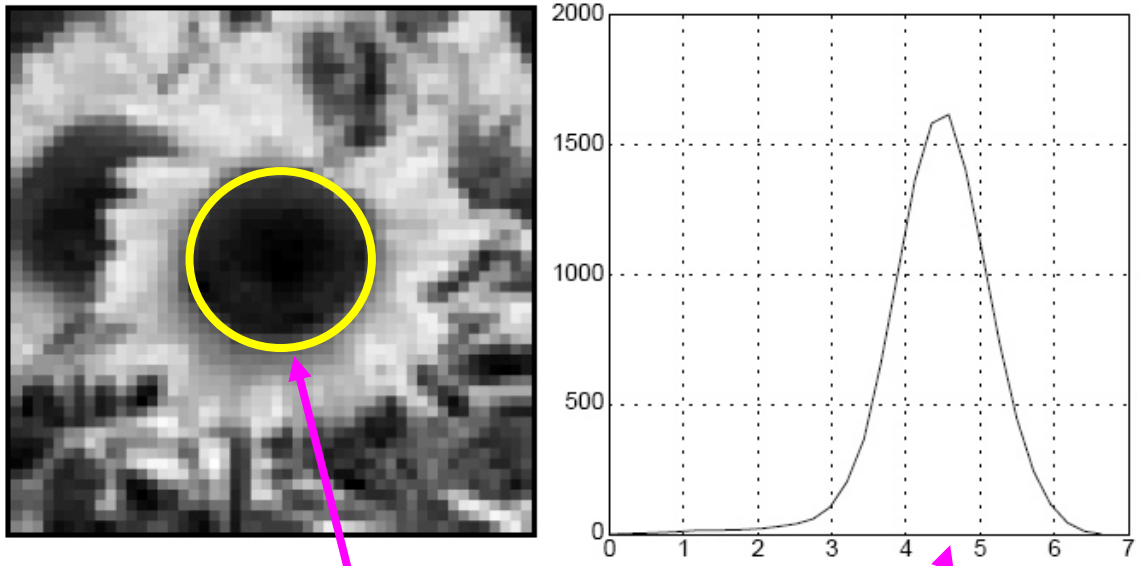


image



Characteristic scale

- We define the *characteristic scale* as the scale that produces peak of Laplacian response



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

Source: Lana Lazebnik

Scale-space blob detector: Example

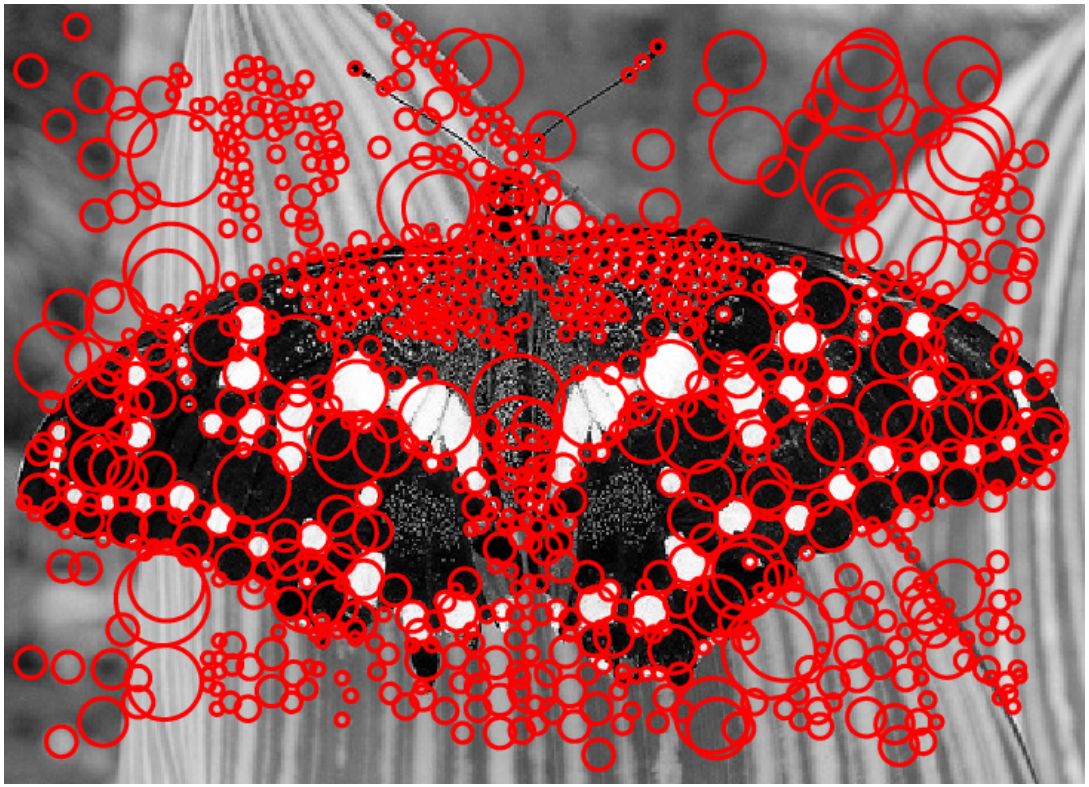
Source: Lana Lazebnik

Scale-space blob detector: Example

sigma = 11.9912

Source: Lana Lazebnik

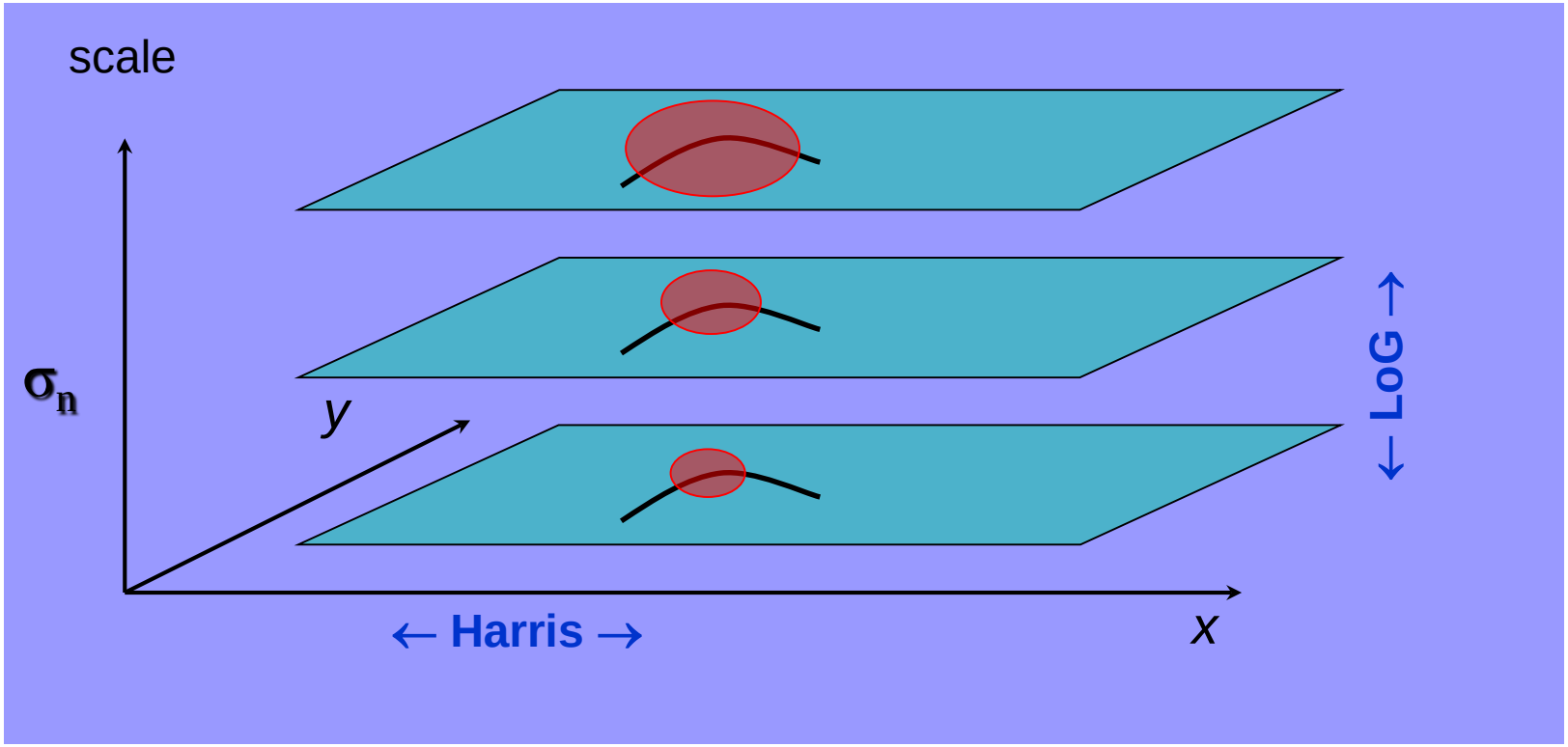
Scale-space blob detector: Example



Source: Lana Lazebnik

Harris-Laplace Detector

- Multi-scale Harris with scale selection.
- Uses LoG maxima to find characteristic scale.

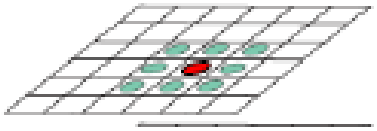


Harris-Laplace Detector (cont'd)

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

- Scales are chosen as follows: $\sigma_n = k^n \sigma_0$ ($\sigma_I = \sigma_n$, $\sigma_D = c\sigma_n$)
- At each scale, choose local maxima assuming 3 x 3 window

$$F(\mathbf{x}, \sigma_n) > F(\mathbf{x_w}, \sigma_n) \quad \forall \mathbf{x_w} \in W$$
$$F(\mathbf{x}, \sigma_n) > t_h$$

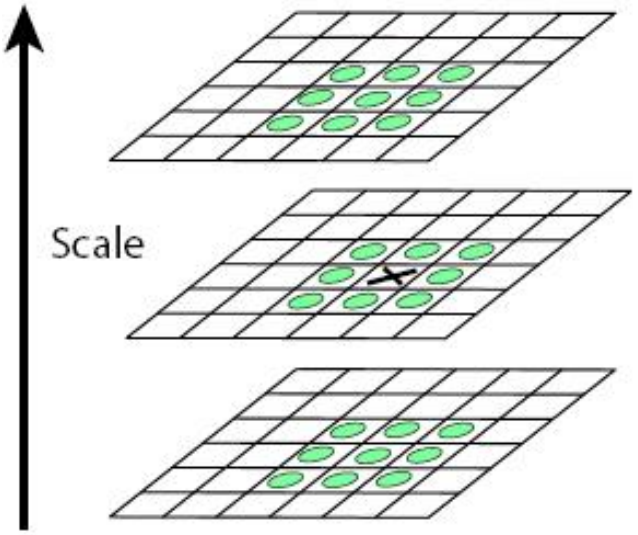


where $F(\mathbf{x}, \sigma_n) = \det(\mathbf{C}) - \alpha \, \mathrm{trace}^2(\mathbf{C})$

K. Mikolajczyk and C. Schmid (2001). "Indexing based on scale invariant interest points" *Int Conference on Computer Vision,* pp 525-531.

Harris-Laplace Detector (cont'd)

(2) Select points at which a local measure (i.e., normalized Laplacian) is maximal over



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

where:

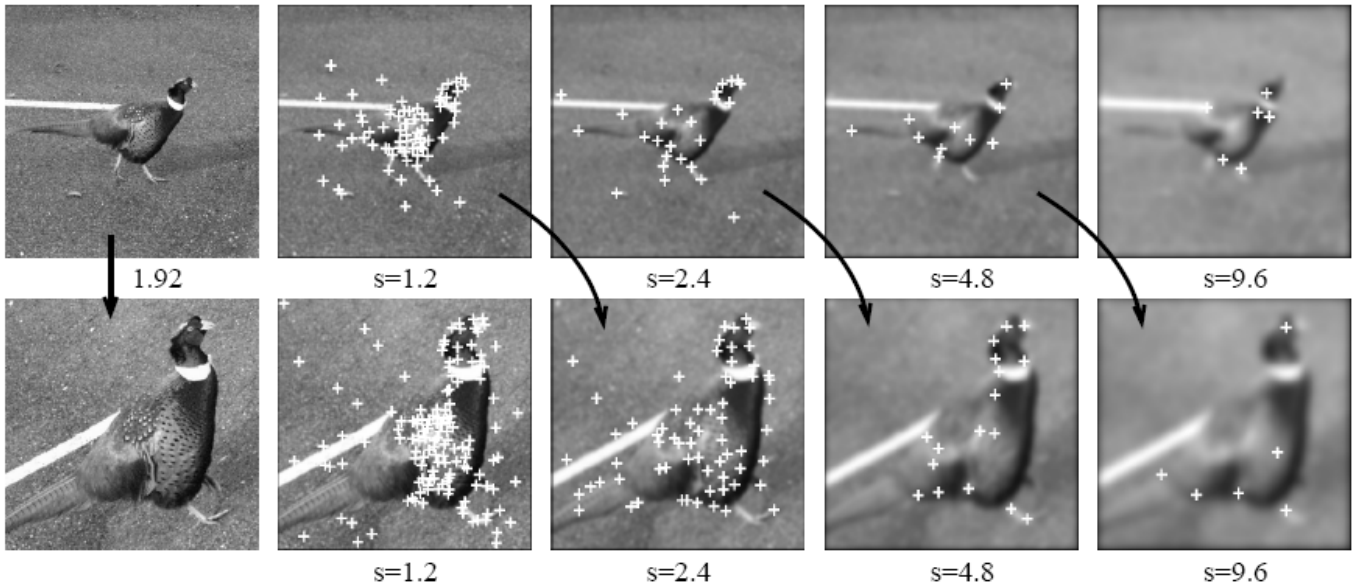
$$F(\mathbf{x}, \sigma_n) = |s^2(L_{xx}(\mathbf{x}, s) + L_{yy}(\mathbf{x}, s))|$$

$(s= \sigma_n)$

K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points" *Int Conference on Computer Vision,* pp 525-531, 2001.

Example

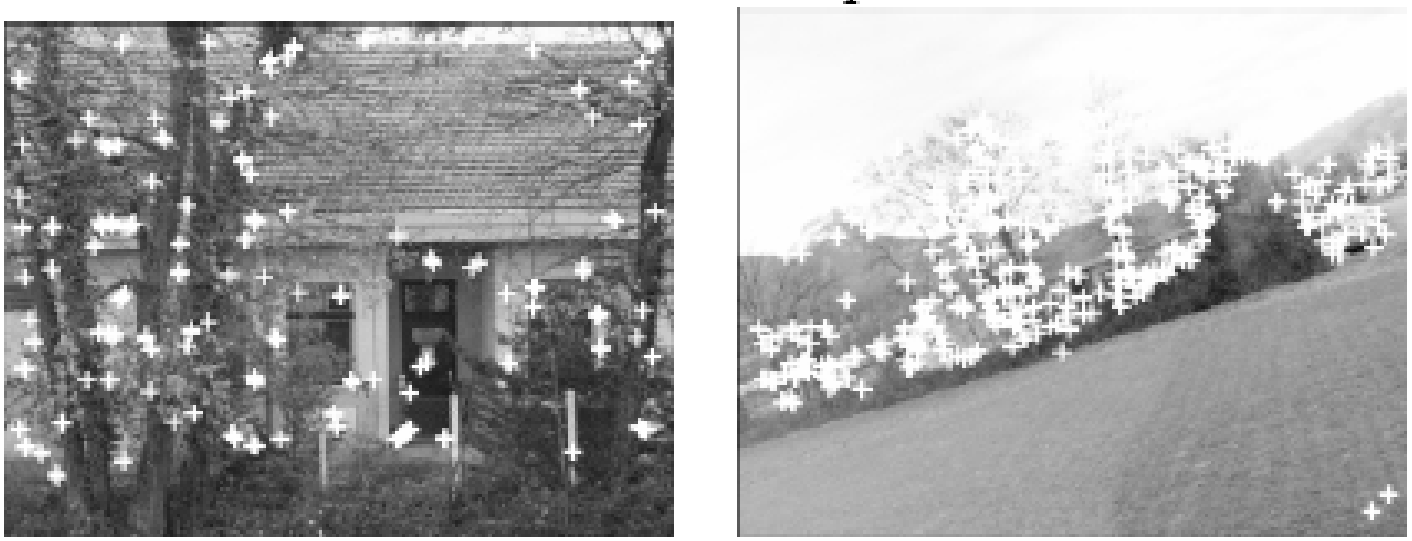
- Points detected at each scale using Harris-Laplace (images differ by a scale factor 1.92)
  - Few points detected in the same location but on different scales.
  - Many correspondences between levels for which scale ratio corresponds to the real scale change between images.



Example

(same viewpoint – change in focal length and orientation)

Extracted interest points



190 and 213 points detected in the left and right images, respectively (more than 2000 points would have been detected without scale selection)

Example (cont'd)

Initial points matches

58 points are initially matched (some not correct)

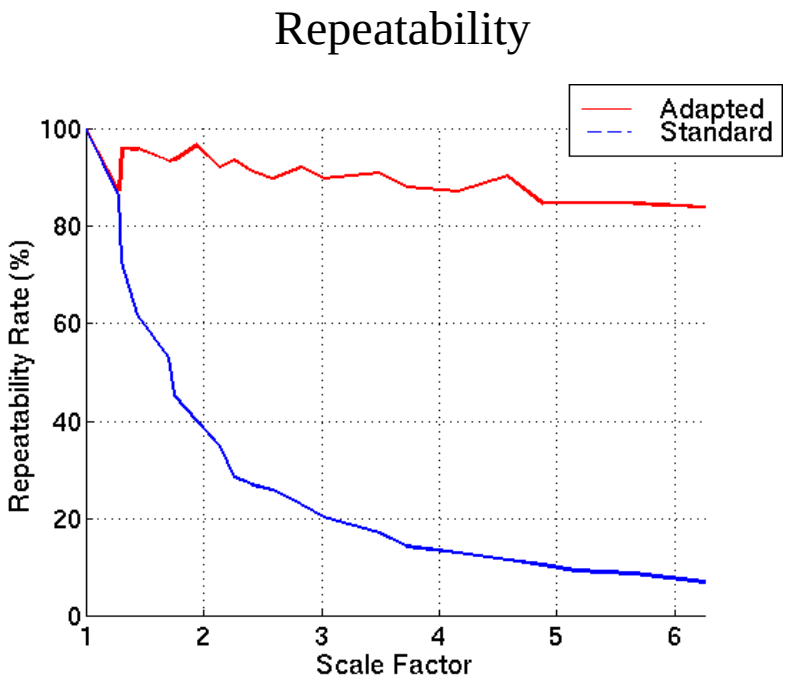
Example (cont'd)

Inliers to the estimated homography

- Reject inconsistent matches using RANSAC to compute the homography between images – left with 32 matches, all of which are correct.
- The estimated scale factor is 4:9 and the estimated rotation angle is 19 degrees.

Harris-Laplace Detector (cont'd)

- Invariant to:
  - Scale
  - Rotation
  - Translation
- Robust to:
  - Illumination changes
  - Limited viewpoint changes

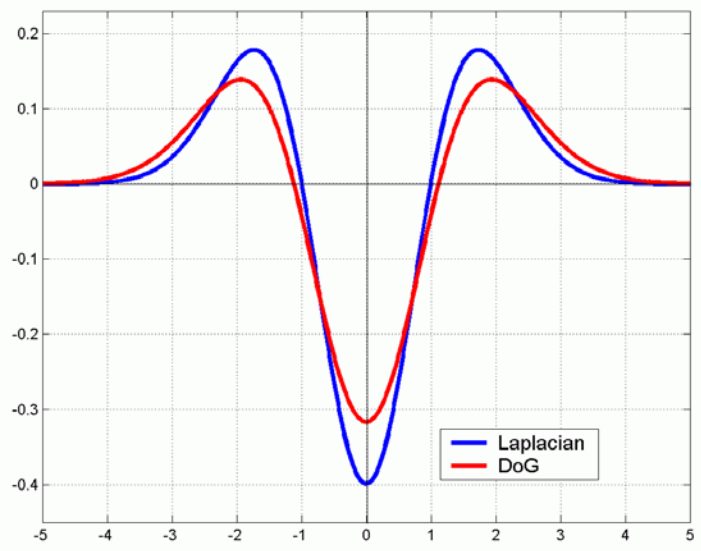


Efficient implementation using DoG

- LoG can be approximated by DoG:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G$$

- Note that DoG already incorporates the σ^2 scale normalization.



Efficient implementation using DoG (cont'd)

- Gaussian-blurred image

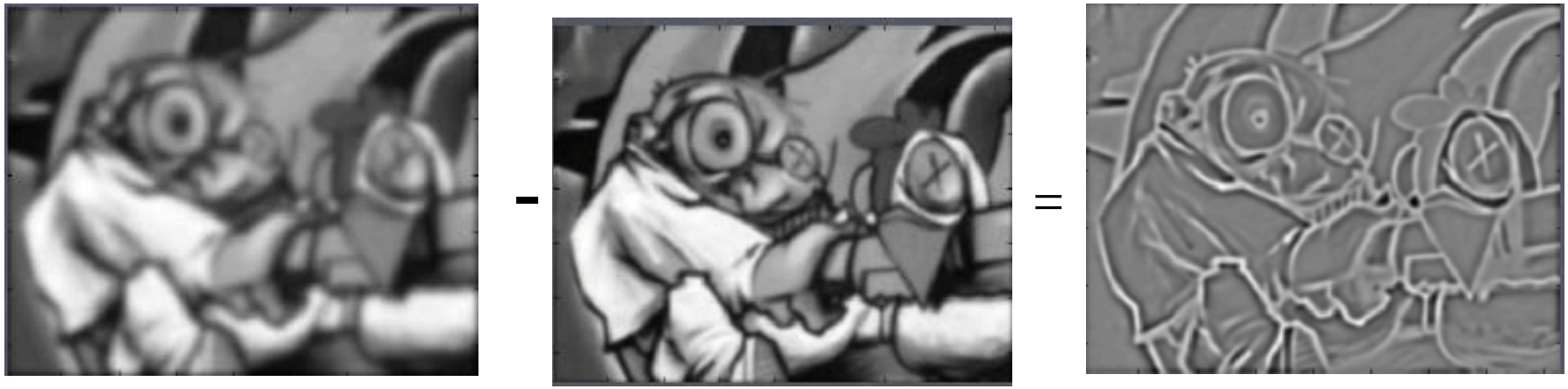
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

- The result of convolving an image with a difference-of Gaussian is given by:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

Example

: $L(x, y, k\sigma) - L(x, y, \sigma)$

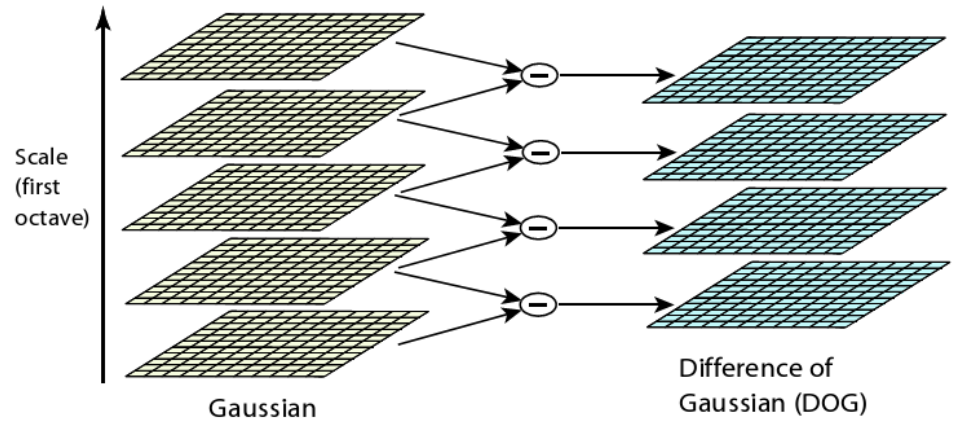


Efficient implementation using DoG (cont'd)

$G(x, y, k^2\sigma) * I$

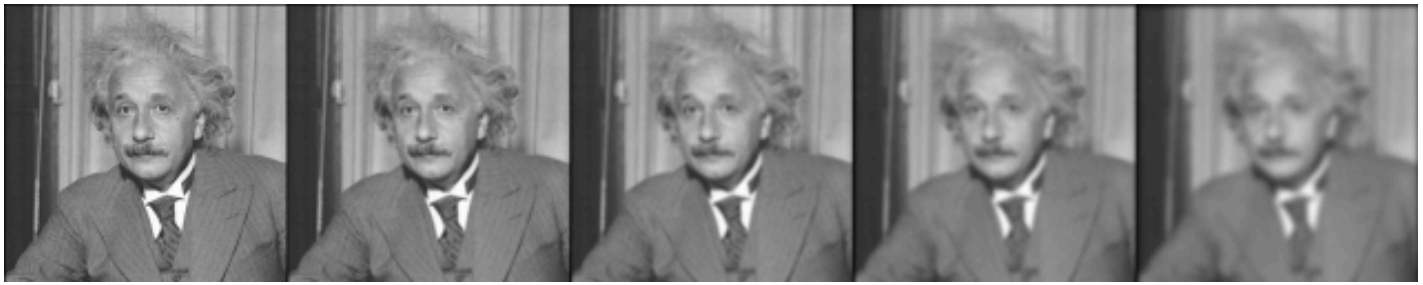
$G(x, y, k\sigma) * I$

$G(x, y, \sigma) * I$



$$D(x, y, \sigma) = \left(G(x, y, k\sigma) - G(x, y, \sigma)\right) * I$$

$\sigma_n = k^n \sigma$

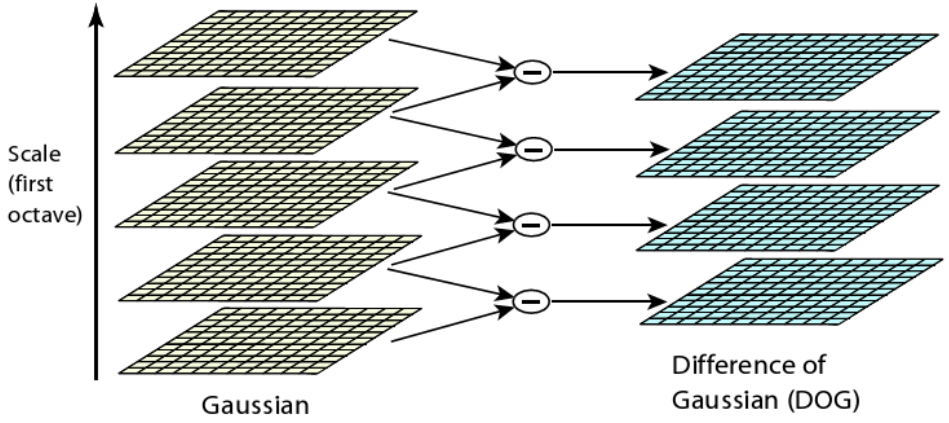


DoG

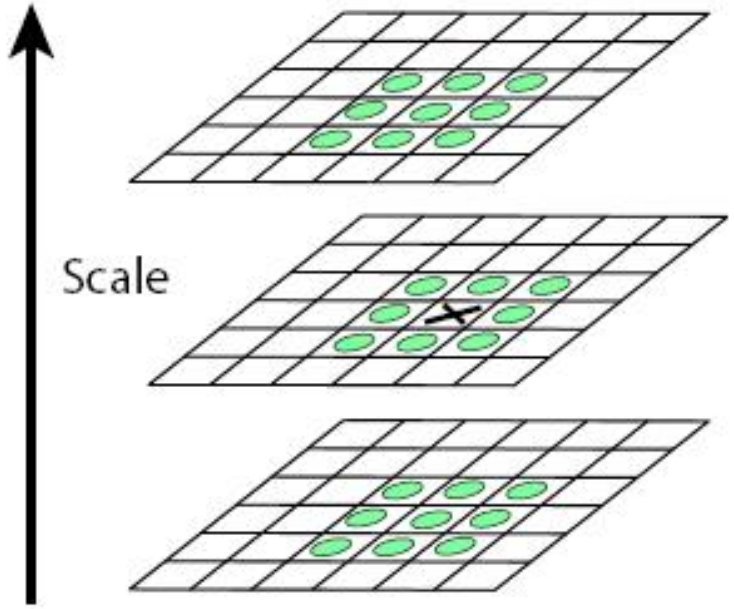
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Efficient implementation using DoG (cont'd)

DoG pyramid

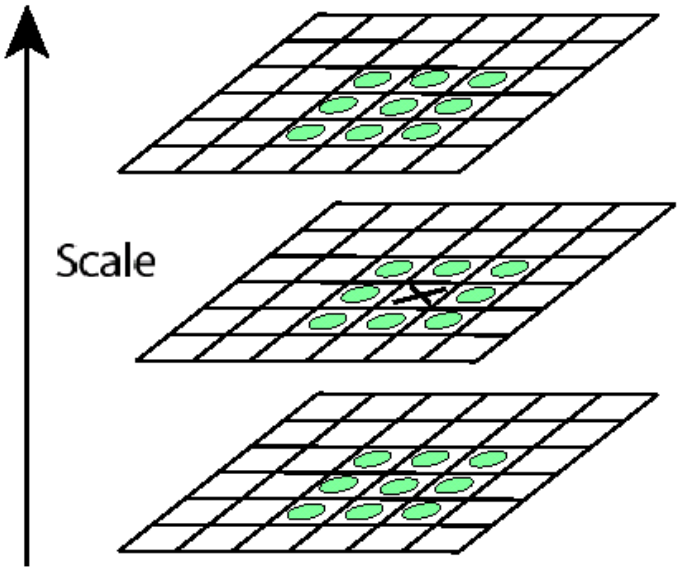


Look for local maxima
in DoG pyramid



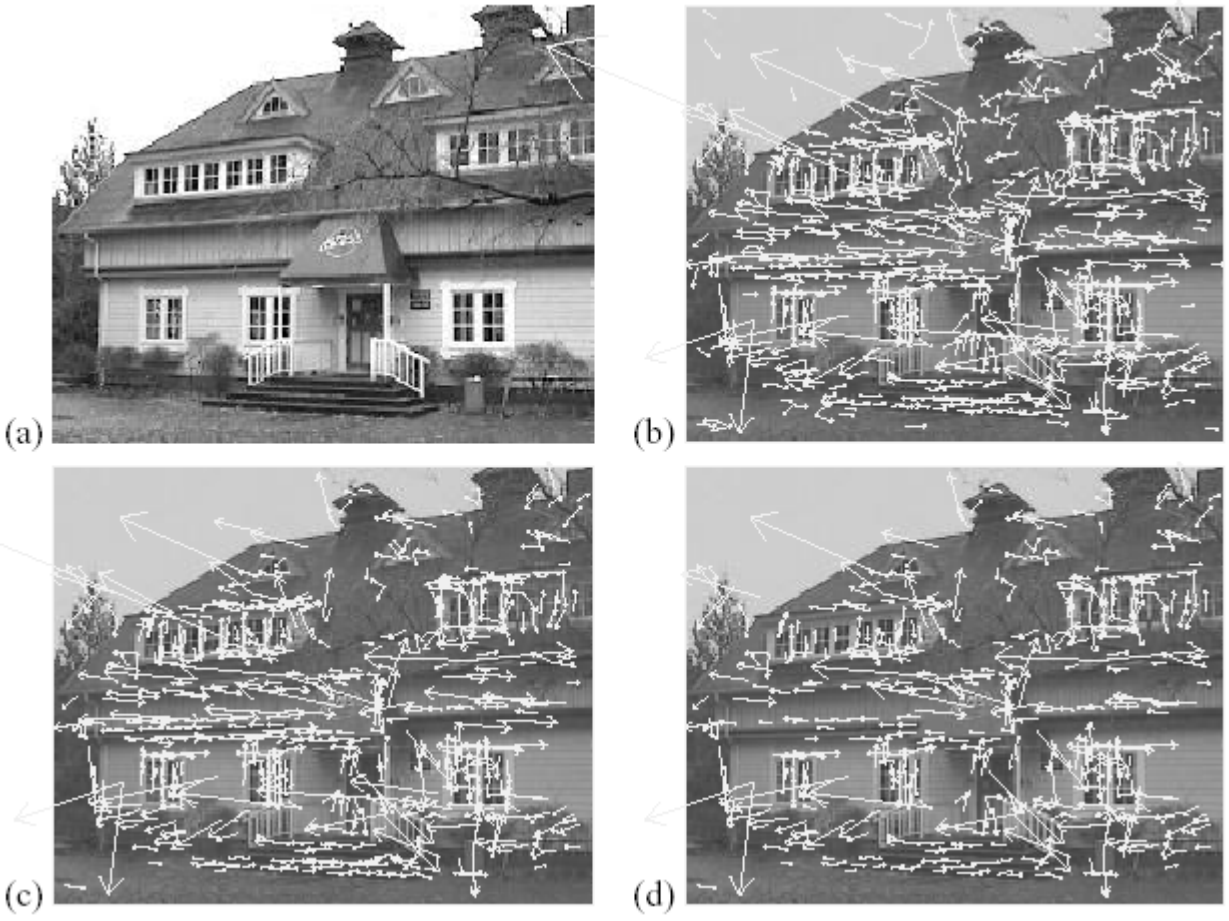
Key point localization with DoG

- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses



Candidate keypoints:
list of (x,y,σ)

Example of keypoint detection



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures (removing edge responses)

Review

- Homography and Ransac
  - ReProjection method from an image to another
  - Robust fitting of the Mapping
- Interest Points
  - Corner Detection: Harris Detector
  - Scaling Issues
    - Harris-Laplace Detector