Local Feature

Interest Points and Corners

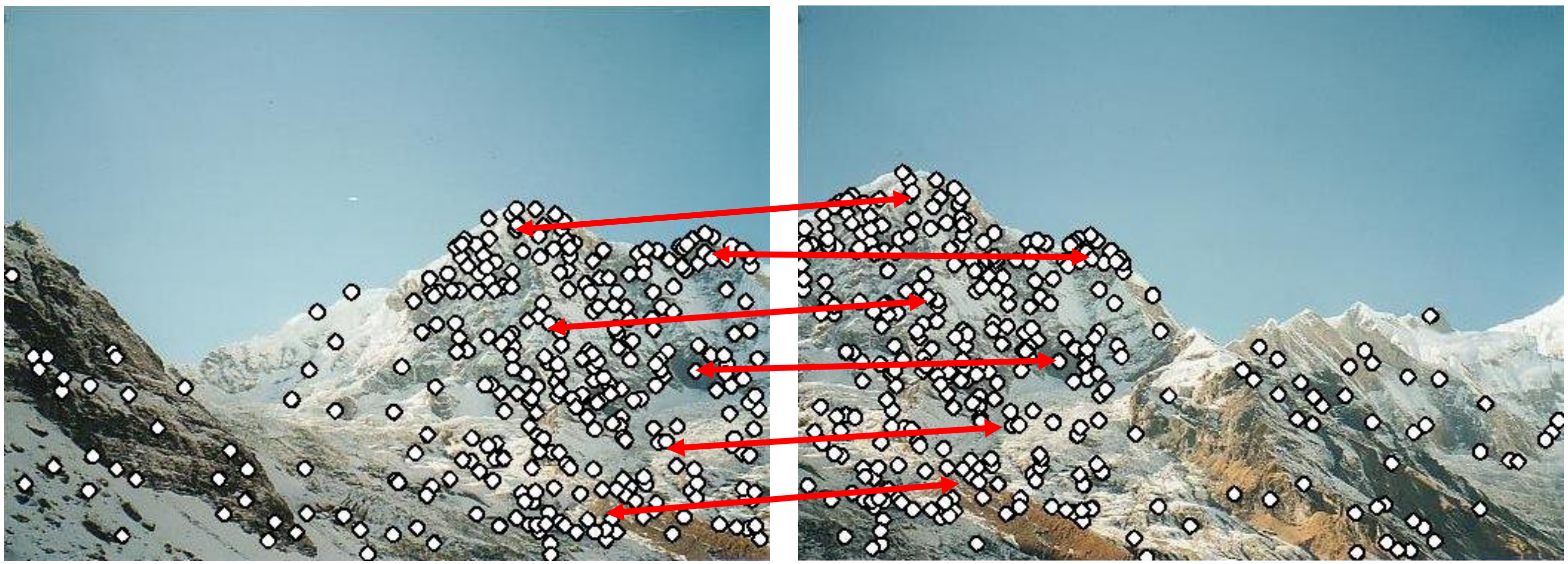
Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Note: “interest points” = “keypoints”, also sometimes called “features”

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?



Step 1: Detect feature points in both images

Step 2: Find corresponding pairs

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

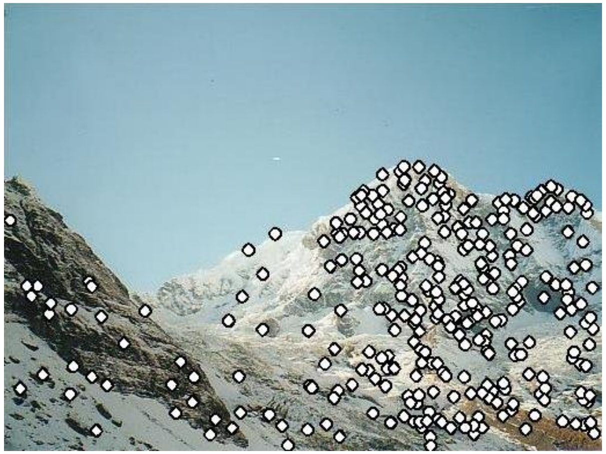
Step 3: Use these pairs to align images

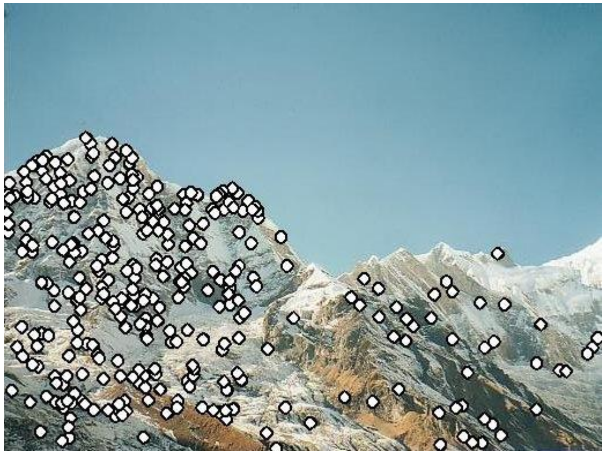
Applications

Feature points are used for:

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

Characteristics of good features





- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature is distinctive
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Characteristics of good features

Problem 1:

- Detect the *same* point *independently* in both images

no chance to match!

- We want to detect (at least some of) the same points in both images.

We need a repeatable detector

[Darya Frolova and Denis Simakov]

Characteristics of good features

Problem 2:

- For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor

[Darya Frolova and Denis Simakov]

Characteristics of good features

Geometric
transformations

Photometric
transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Invariant local features

Subset of local feature types designed to be invariant to common geometric and photometric transformations.

Basic steps:

1) Detect distinctive interest points
2) Extract invariant descriptors

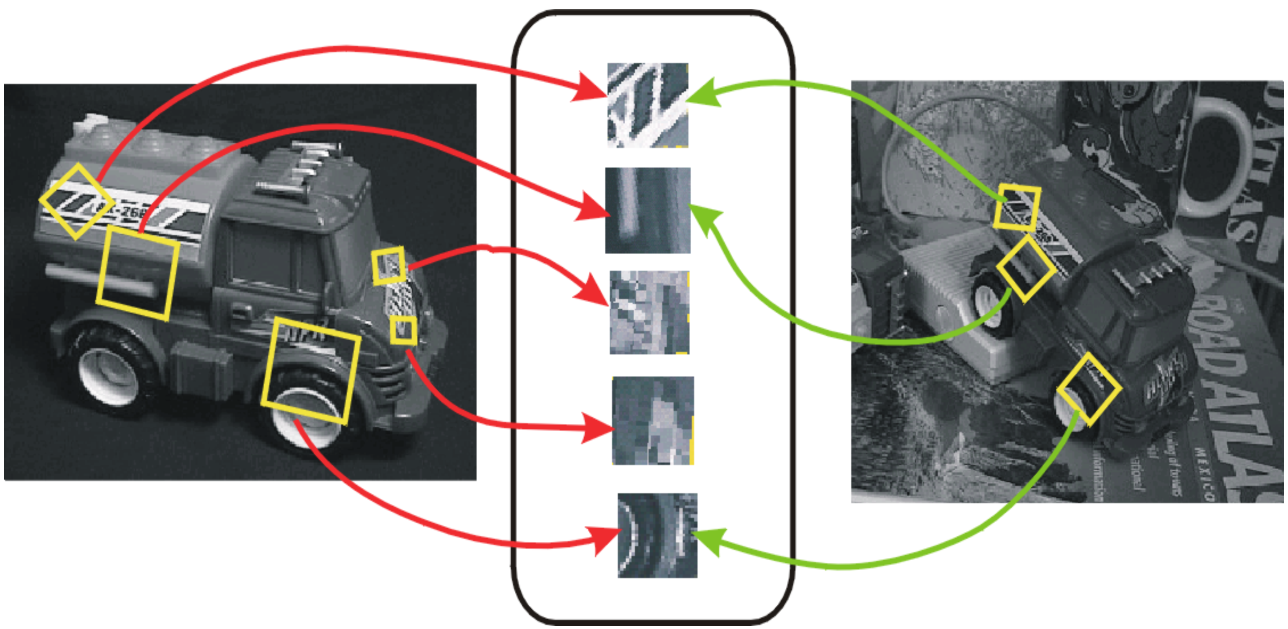


Figure: David Lowe

Main questions

Where will the interest points come from?

- What are salient features that we'll *detect* in multiple views?

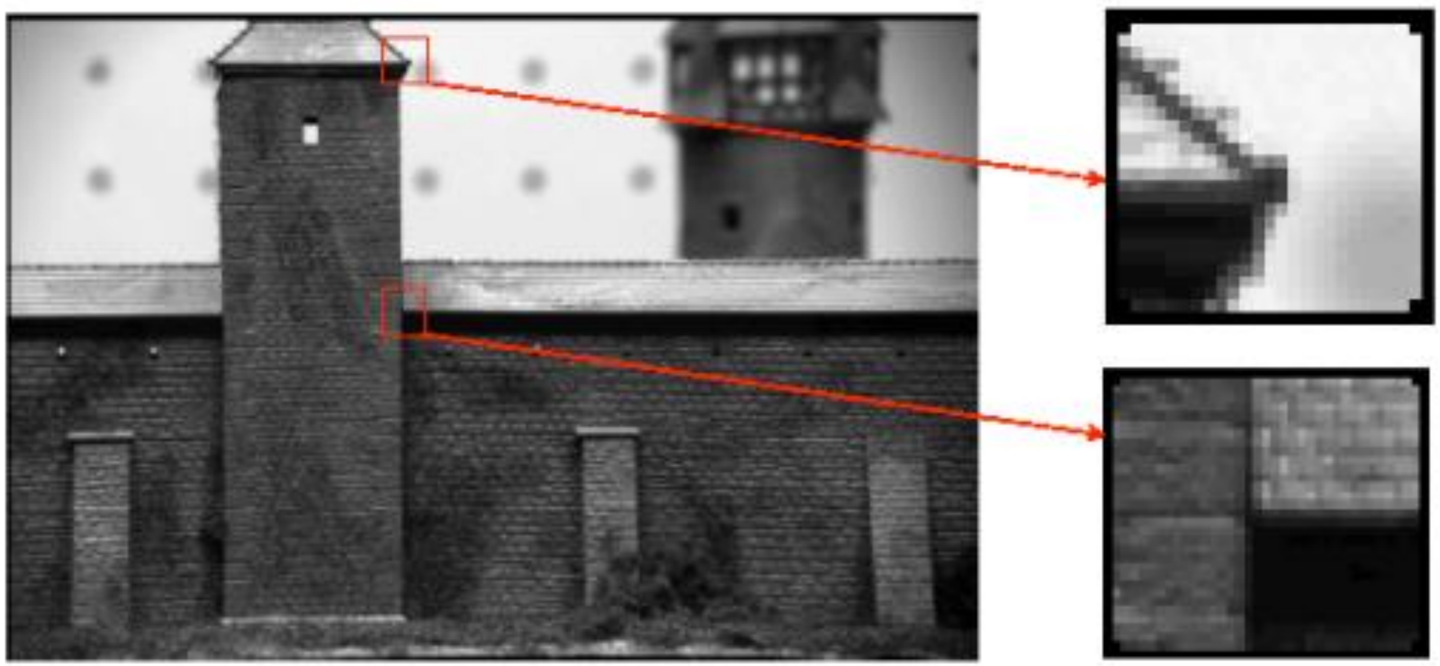
How to *describe* a local region?

How to establish *correspondences*, i.e., compute matches?

What points would you choose?

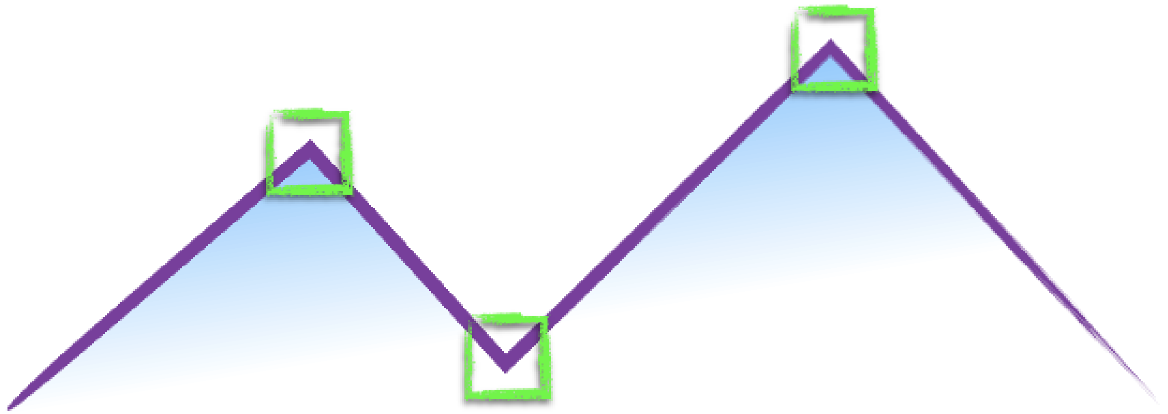
Corners as distinctive interest points

- What are Corners?
  - Intuitively, junctions of contours.
  - Generally more stable features over changes of viewpoint
  - Intuitively, large variations in the neighborhood of the point in all directions



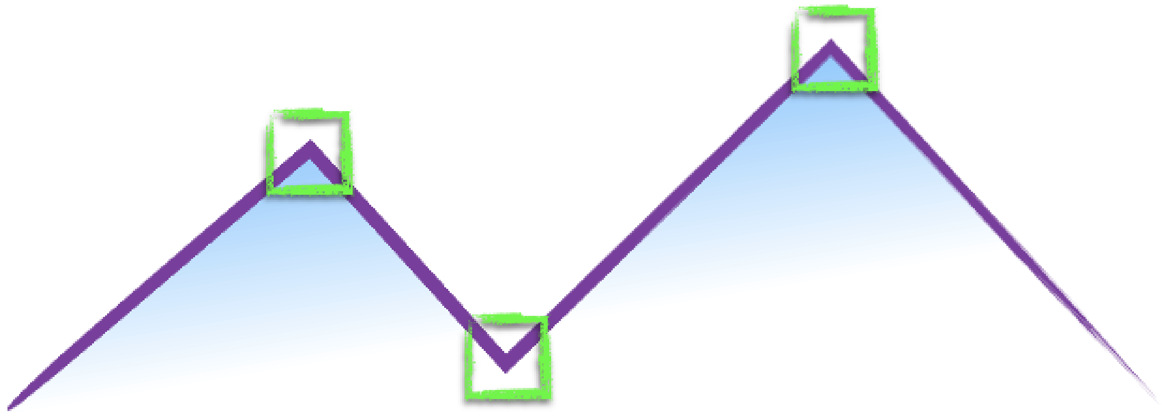
Harris corner detector

How do you find a corner?



How do you find a corner?

[Moravec 1980]

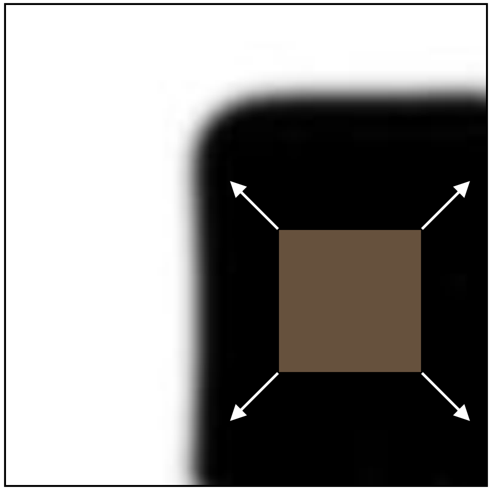


Easily recognized by looking through a small window

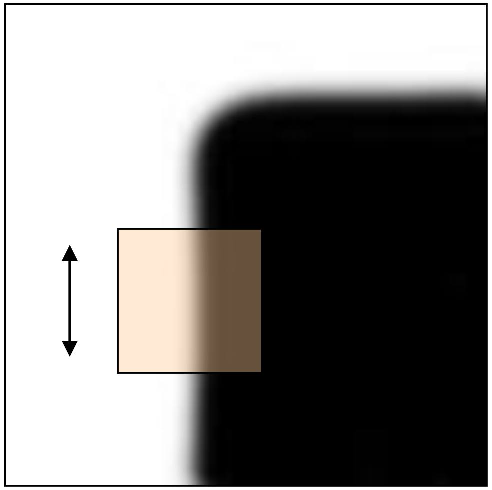
Shifting the window should give large change in intensity

Corners as distinctive interest points

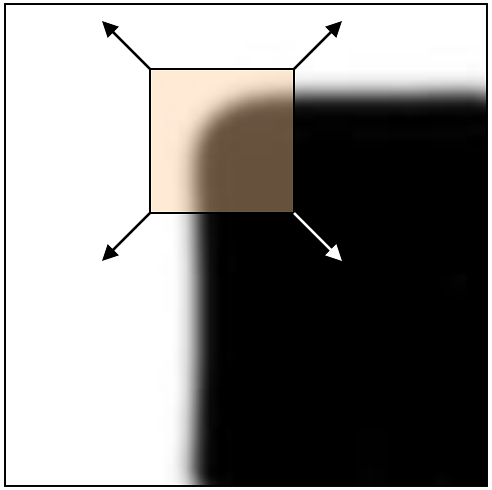
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction



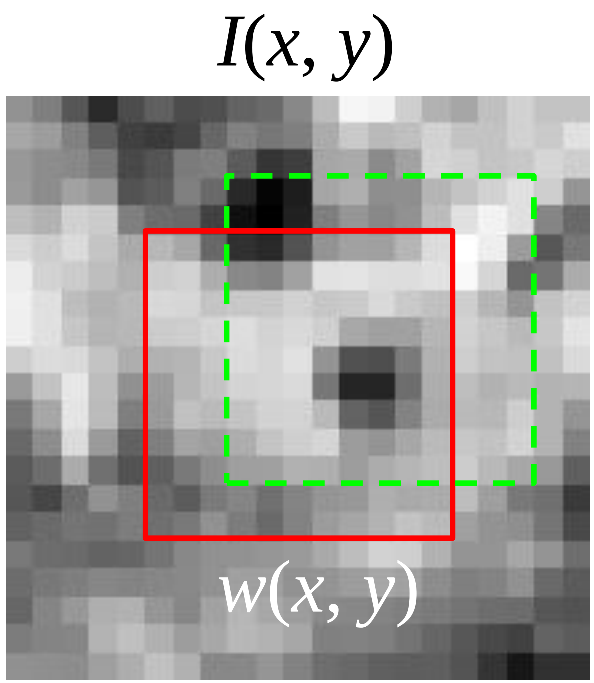
"corner":
significant change in all directions

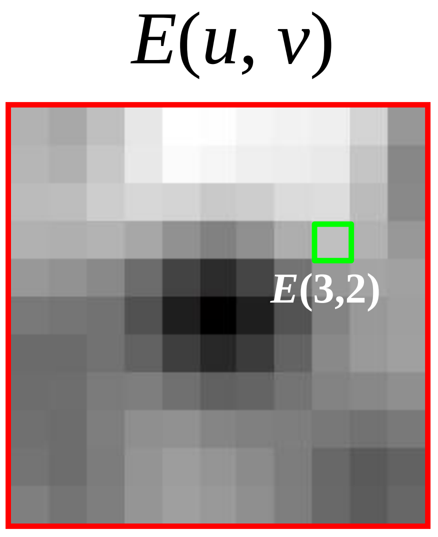
Source: A. Efros

Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

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



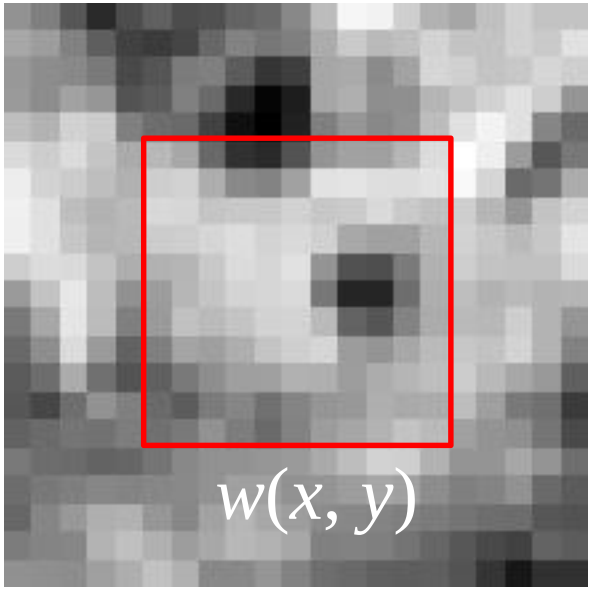


Corner Detection: Mathematics

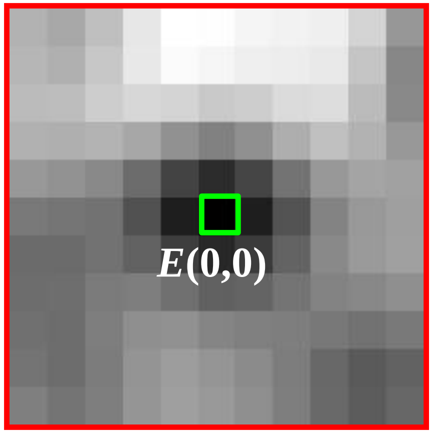
Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

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

$I(x, y)$



$E(u, v)$



Corner Detection: Mathematics

Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

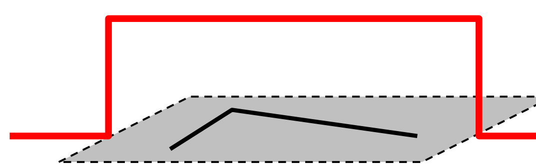
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

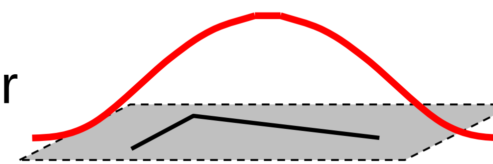
Intensity

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

Source: R. Szeliski

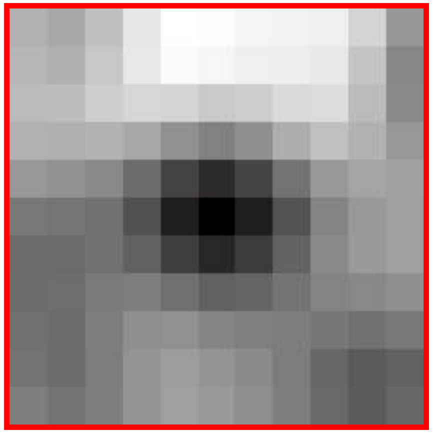
Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

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

We want to find out how this function behaves for small shifts

$E(u, v)$



Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for shift $[u,v]$:

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

We want to discover how E behaves for small shifts

But this is very slow to compute naively.
O(window_width^2 * shift_range^2 * image_width^2)

O(11^2 * 11^2 * 600^2) = 5.2 billion of these
14.6 thousand per pixel in your image

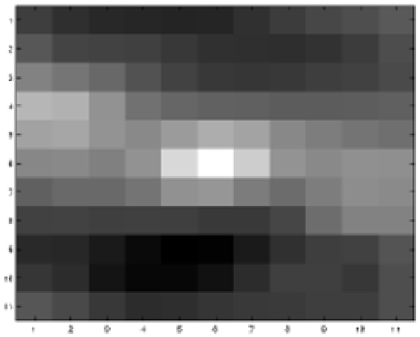
Corner Detection: Mathematics

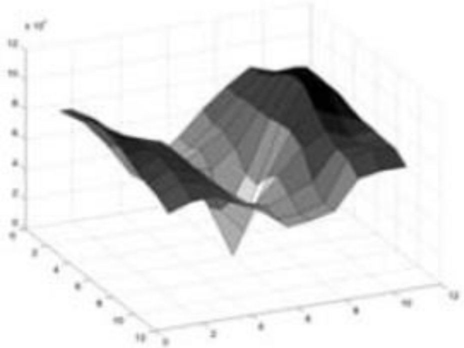
Change in appearance of window *w*(*x*,*y*) for shift [*u*,*v*]:

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

We want to discover how E behaves for small shifts

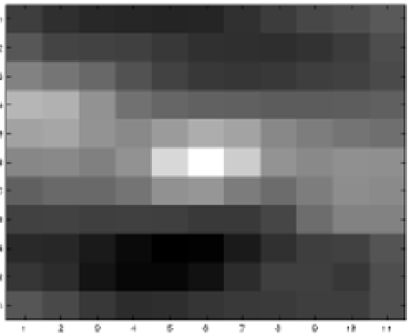
But we know the response in *E* that we are looking for – strong peak.

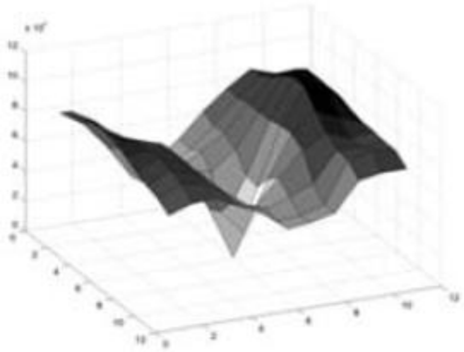




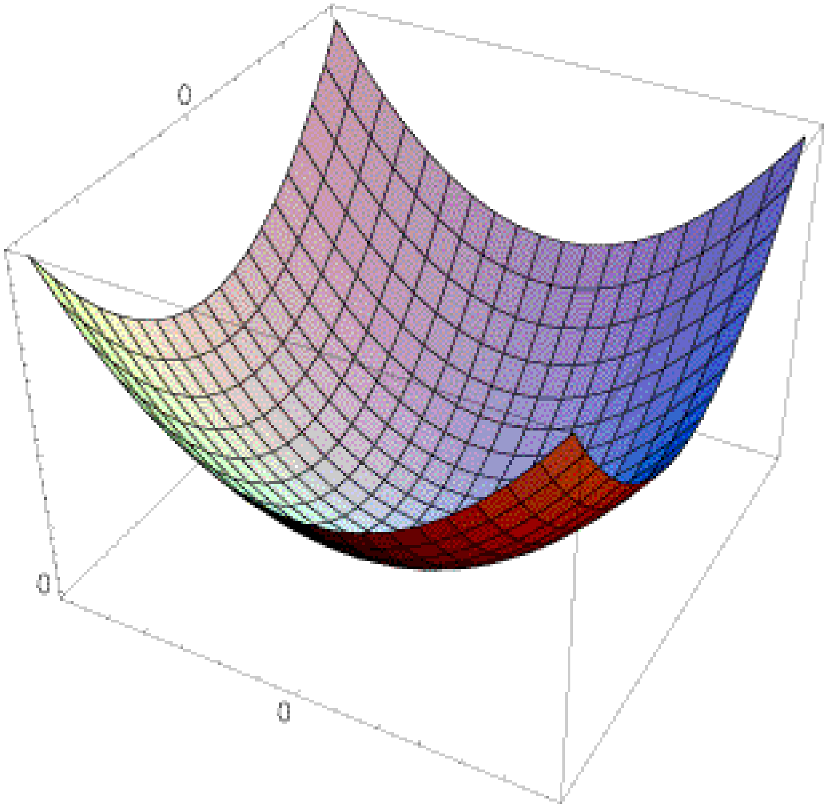
Corner Detection: Mathematics

Can we just approximate $E(u,v)$ locally by a quadratic surface?





$\approx$



Corner Detection: Mathematics

Change in appearance of window *w*(*x*,*y*) for the shift [*u*,*v*]:

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

We want to find out how this function behaves for small shifts

Local quadratic approximation of *E*(*u*,*v*) in the neighborhood of (0,0) is given by the *second-order Taylor expansion*:

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector formulation

This measure of change can be approximated by:

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

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

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

Gradient with respect to x, times gradient with respect to y

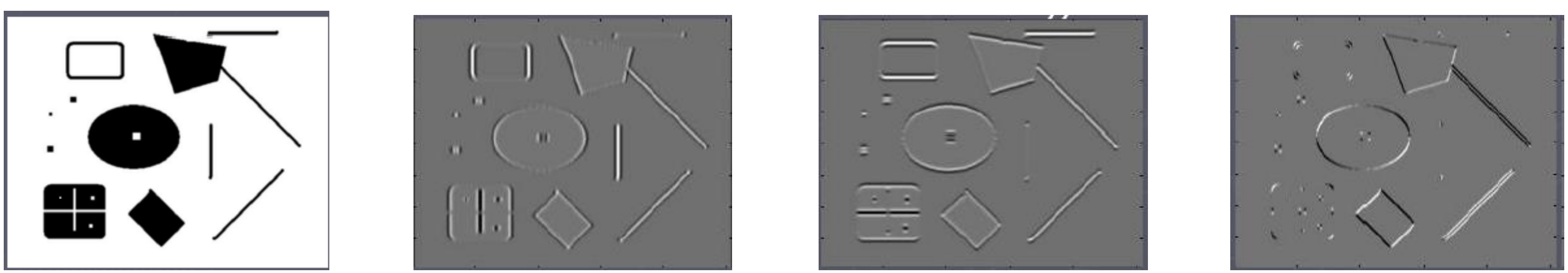
Sum over image region – area we are checking for corner

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

Harris Detector formulation

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

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).



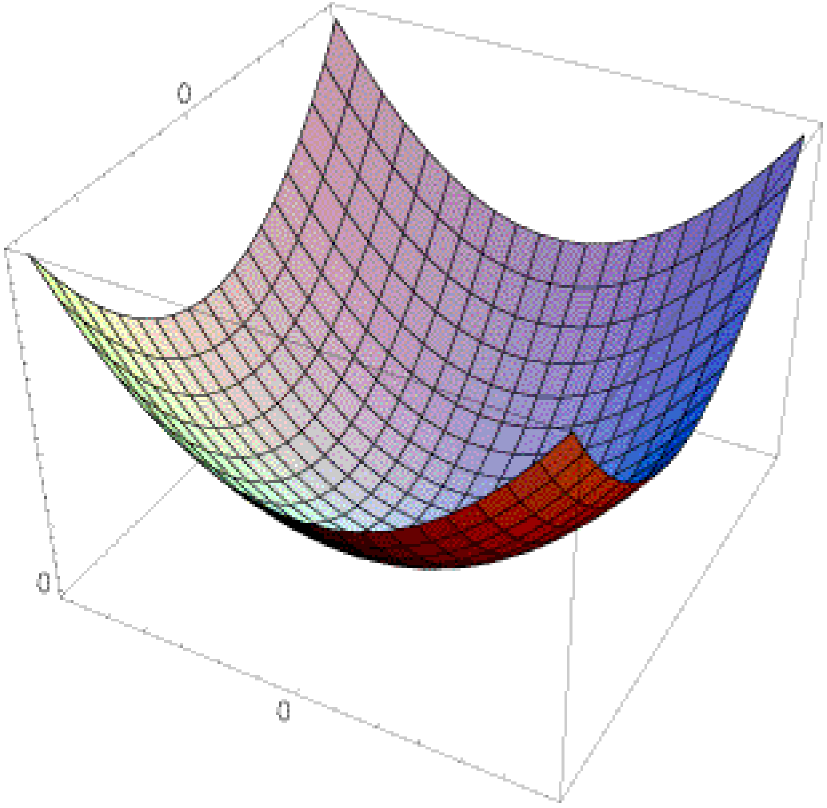
Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $\quad I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $\quad I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

Interpreting the second moment matrix

The surface *E*(*u*,*v*) is locally approximated by a quadratic form. Let's try to understand its shape.

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

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



James Hays

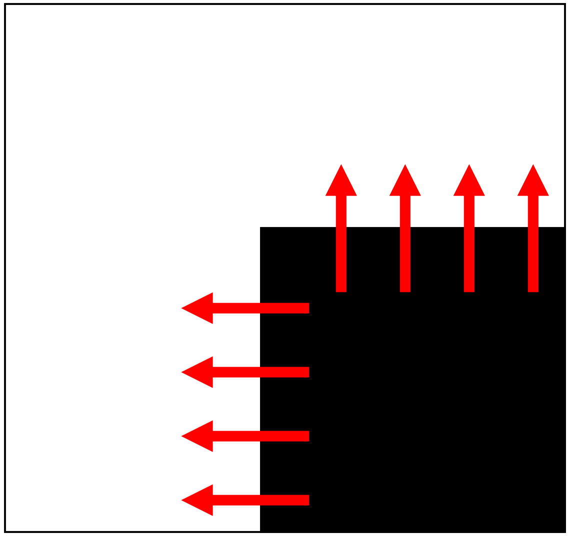
What does this matrix reveal?

Intensity change in shifting window: eigenvalue analysis

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

λ_1, λ_2 – eigenvalues of M

First, consider an axis-aligned corner:



What does this matrix reveal?

First, consider an axis-aligned corner:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

This means dominant gradient directions align with x or y axis

If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

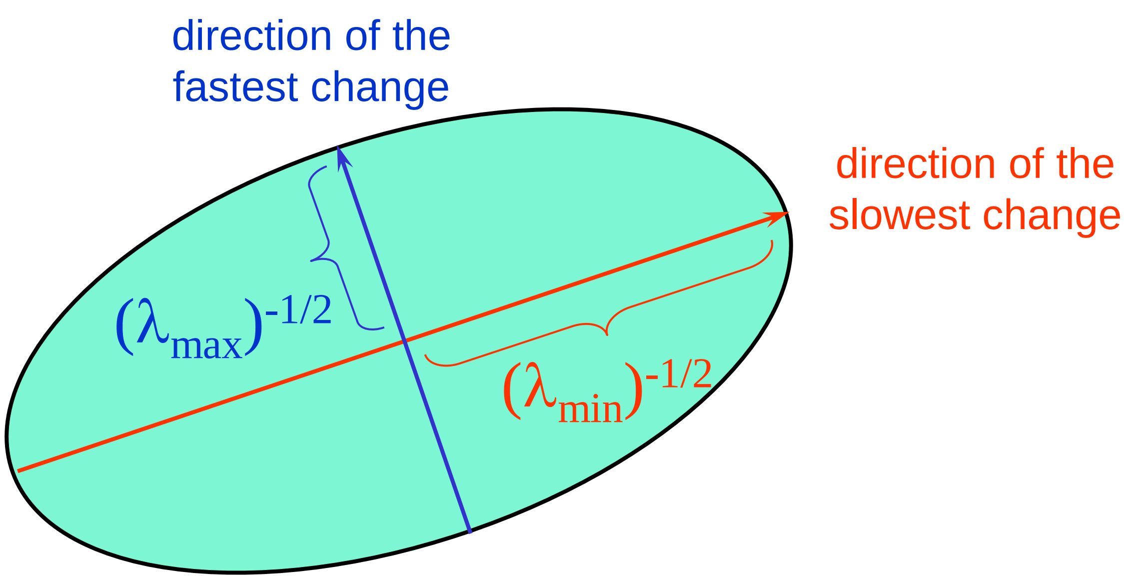
What if we have a corner that is not aligned with the image axes?

Slide credit: David Jacobs

General Case

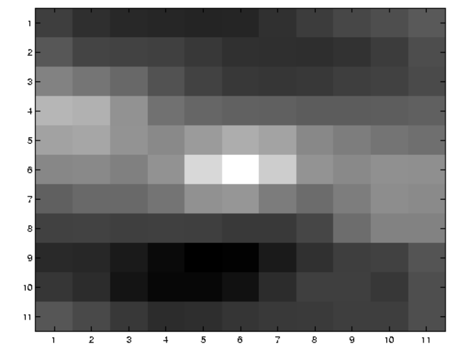
Since M is symmetric, we have $M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

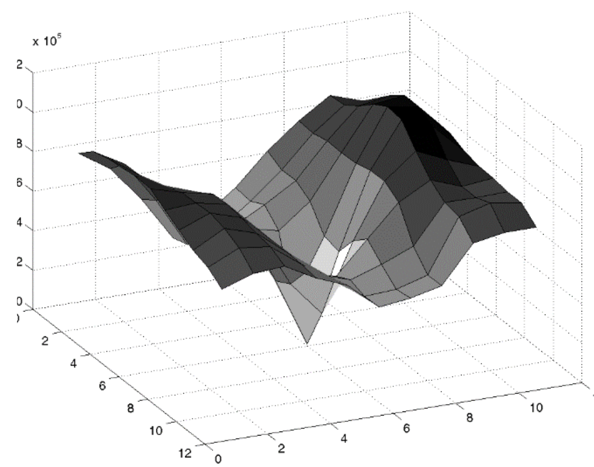
We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*



Slide adapted form Darya Frolova, Denis Simakov.

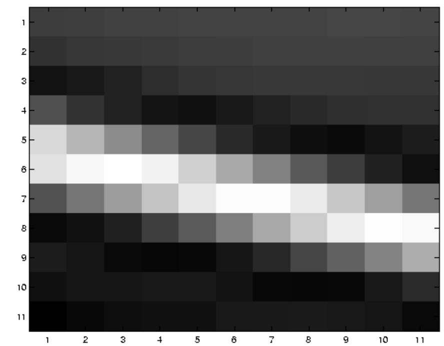
Selecting Good Features

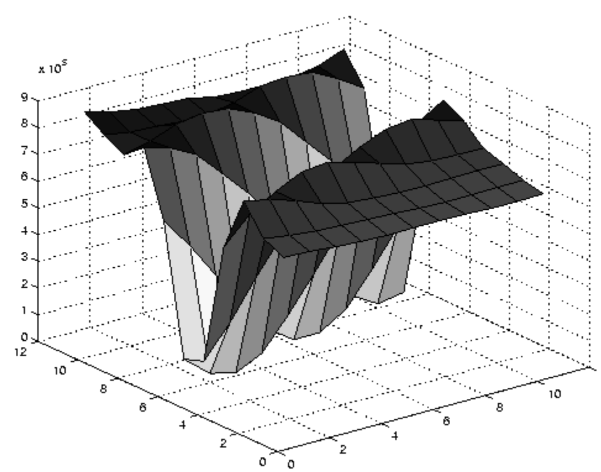




λ_1 and λ_2 are large

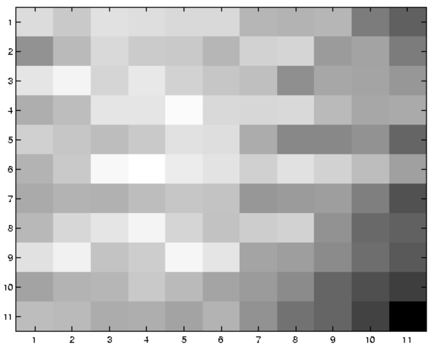
Selecting Good Features

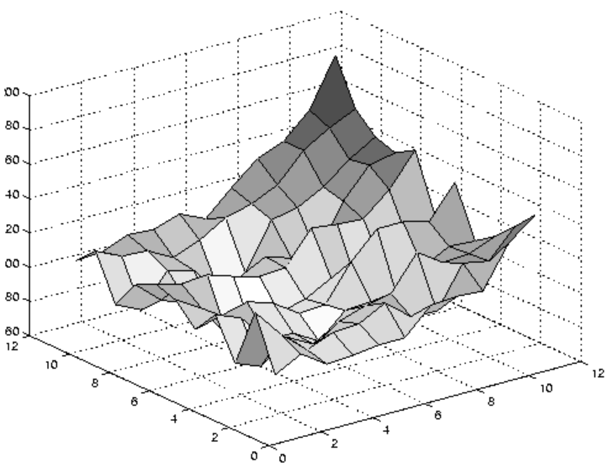




large λ_1, small λ_2

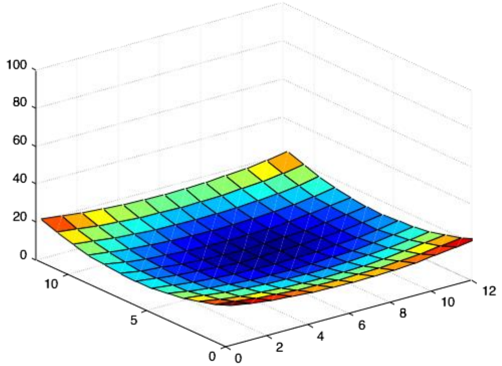
Selecting Good Features

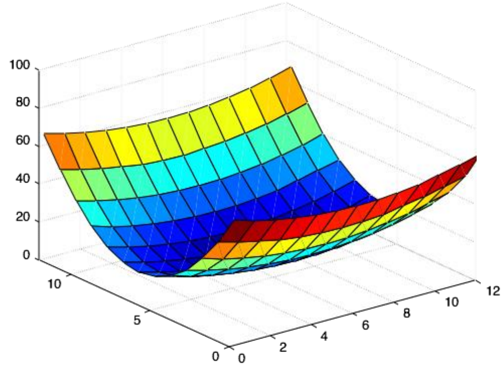


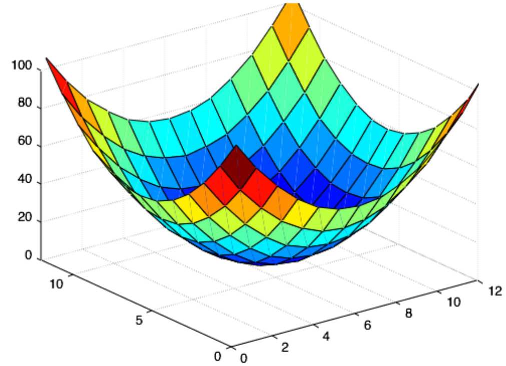


small λ_1, small λ_2

Which error surface indicates a good image feature?







flat

edge
‘line’

corner
‘dot’

4. Compute eigenvalues and eigenvectors

eigenvalue

$$M\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

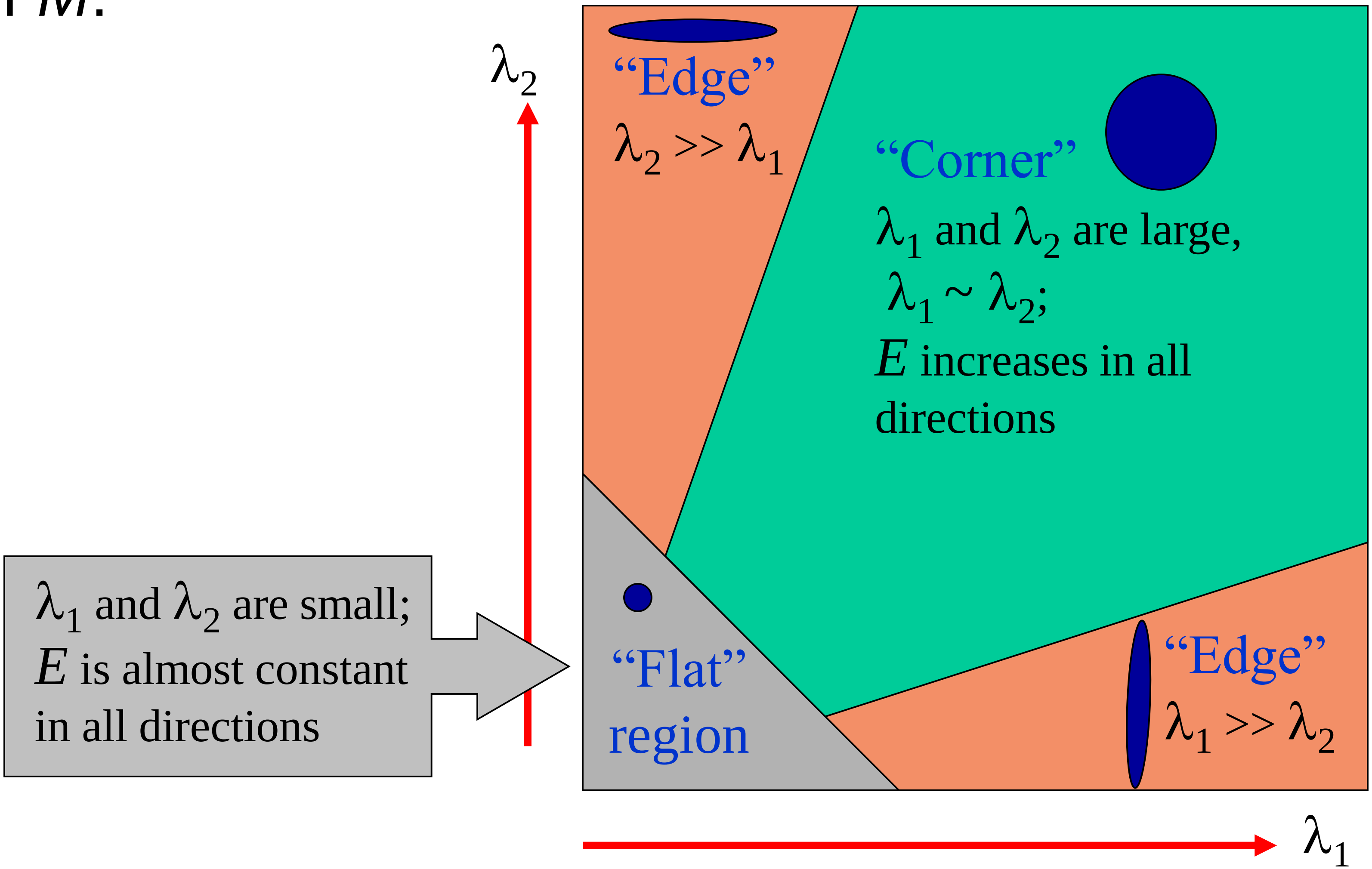
1. Compute the determinant of $M - \lambda I$
(returns a polynomial)

2. Find the roots of polynomial $\det(M - \lambda I) = 0$
(returns eigenvalues)

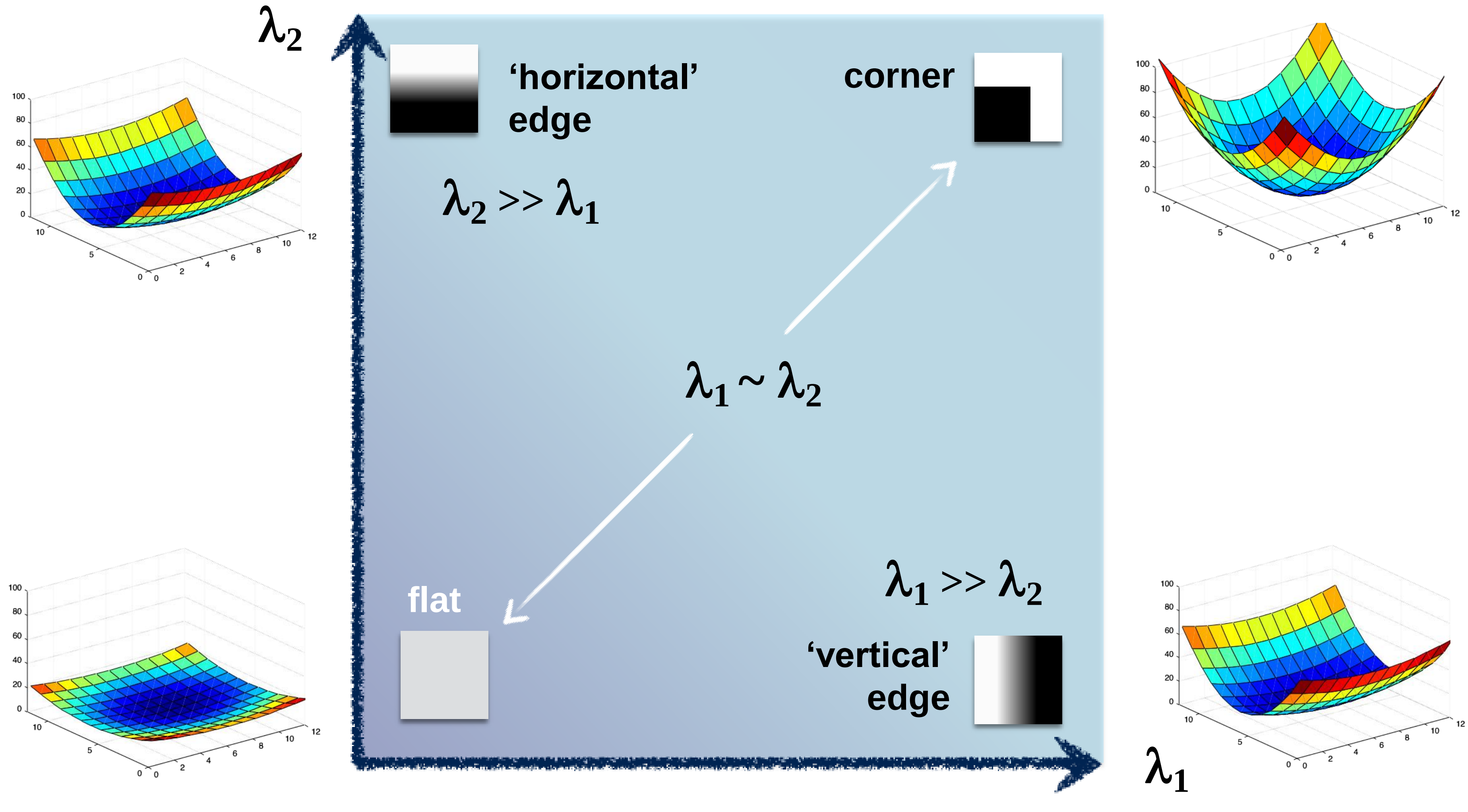
3. For each eigenvalue, solve $(M - \lambda I)\boldsymbol{e} = 0$
(returns eigenvectors)

Interpreting the eigenvalues

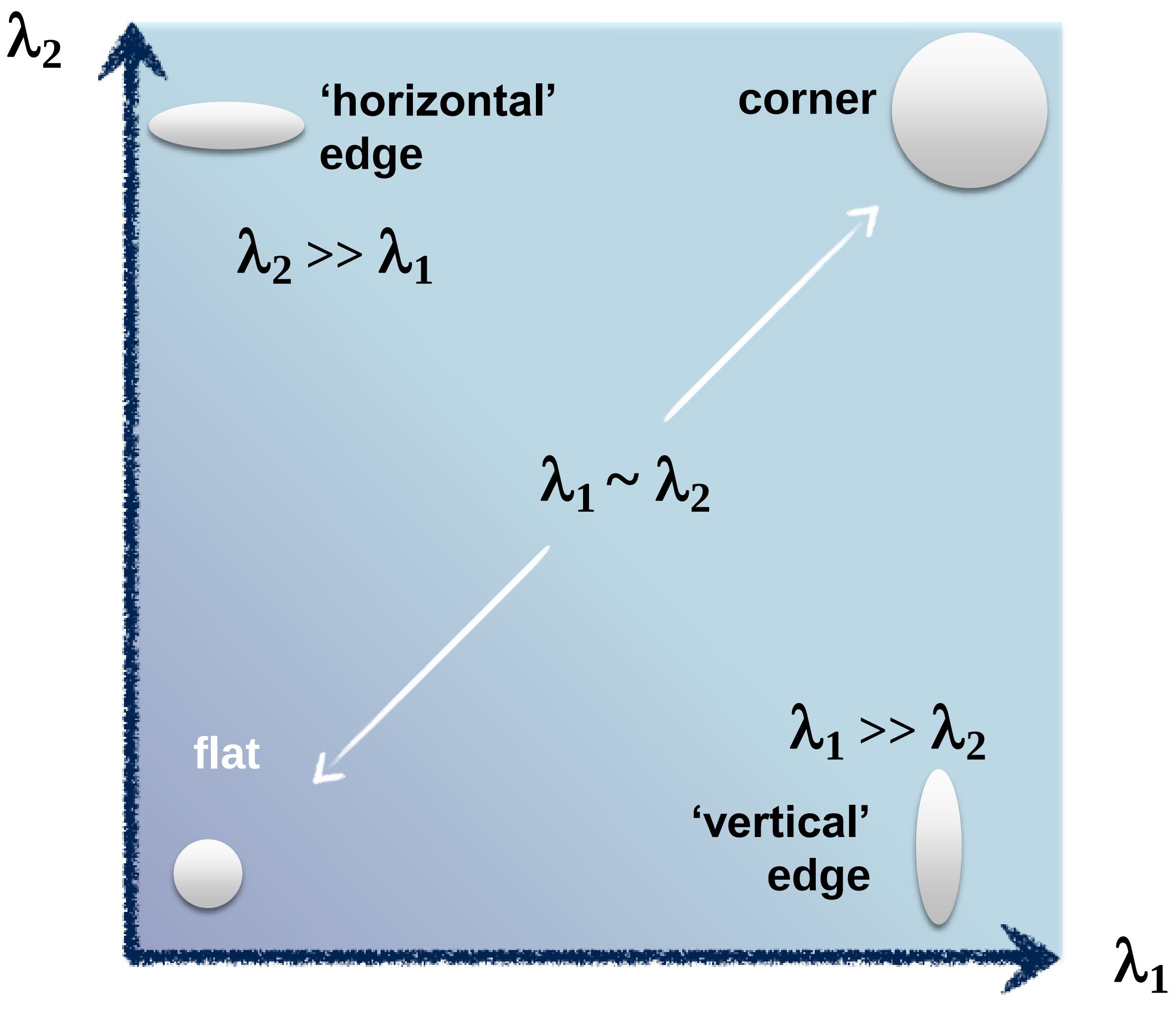
Classification of image points using eigenvalues of *M*:



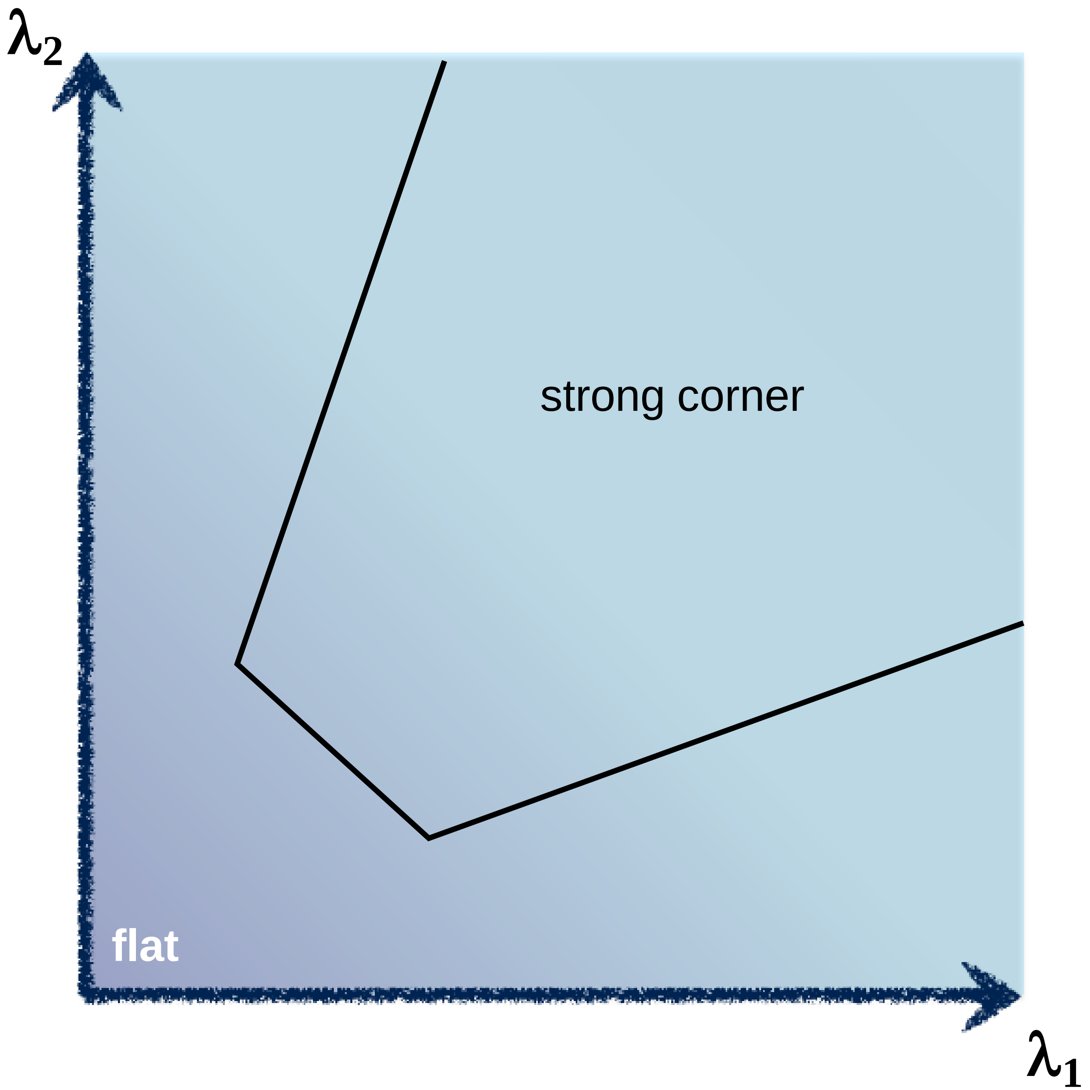
interpreting eigenvalues



interpreting eigenvalues



5. Use threshold on eigenvalues to detect corners

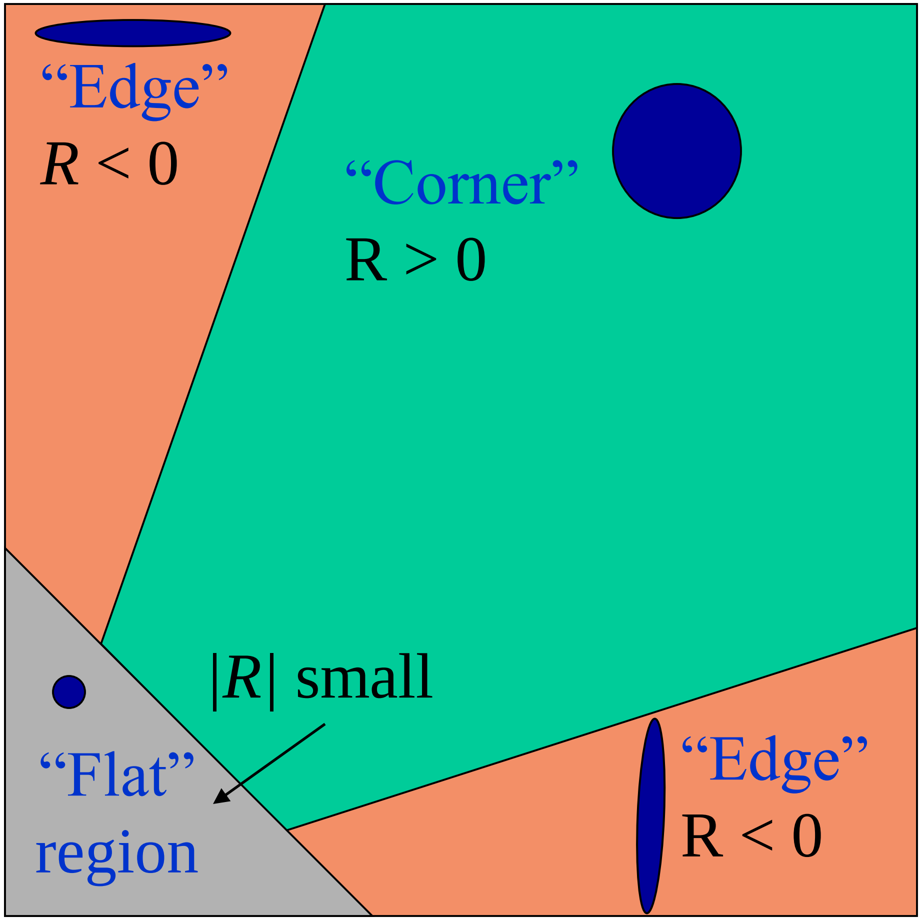


Think of a function to score ‘cornerness’

Corner response function

$$R = \det(M) - \alpha \, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: empirical constant (0.04 to 0.06)



Harris & Stephens (1988)

$$R = \det(M) - \kappa \operatorname{trace}^2(M)$$

Kanade & Tomasi (1994)

$$R = \min(\lambda_1, \lambda_2)$$

Nobel (1998)

$$R = \frac{\det(M)}{\operatorname{trace}(M) + \epsilon}$$

Harris detector: Steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix *M* in a Gaussian window around each pixel
3. Compute corner response function *R*
4. Threshold *R*
5. Find local maxima of response function (nonmaximum suppression)

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris Detector

C.Harris and M.Stephens. "A Combined Corner and Edge Detector."1988.

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x \qquad I_{y^2} = I_y \cdot I_y \qquad I_{xy} = I_x \cdot I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x^2} = G_{\sigma'} * I_{x^2} \qquad S_{y^2} = G_{\sigma'} * I_{y^2} \qquad S_{xy} = G_{\sigma'} * I_{xy}$$

Harris Detector

C.Harris and M.Stephens. “A Combined Corner and Edge Detector.”1988.

4. Define the matrix at each pixel

$$M(x,y) = \begin{bmatrix} S_{x^2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y^2}(x,y) \end{bmatrix}$$

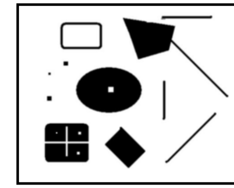
5. Compute the response of the detector at each pixel

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

6. Threshold on value of R; compute non-max suppression.

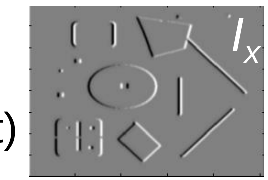
Harris Detector [Harris88]

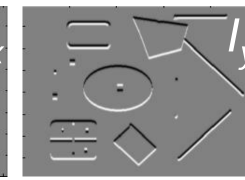
Compute matrix M



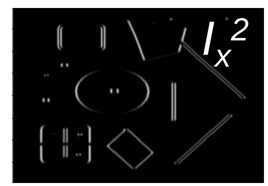
$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

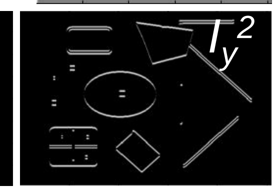
1. Image derivatives (optionally, blur first)

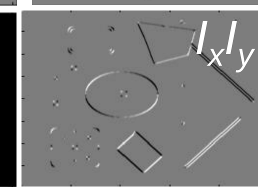




2. Square of derivatives







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

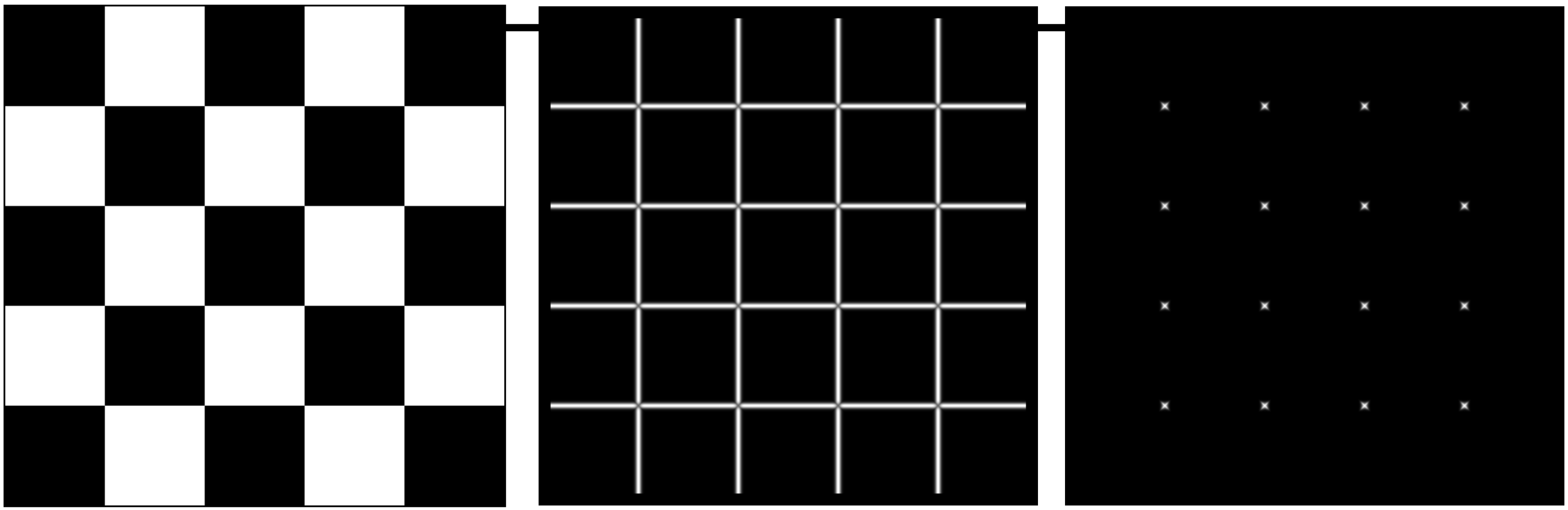
3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

$$har = \det[M(\sigma_I,\sigma_D)] - \alpha[\operatorname{trace}(M(\sigma_I,\sigma_D))^2] =$$

$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

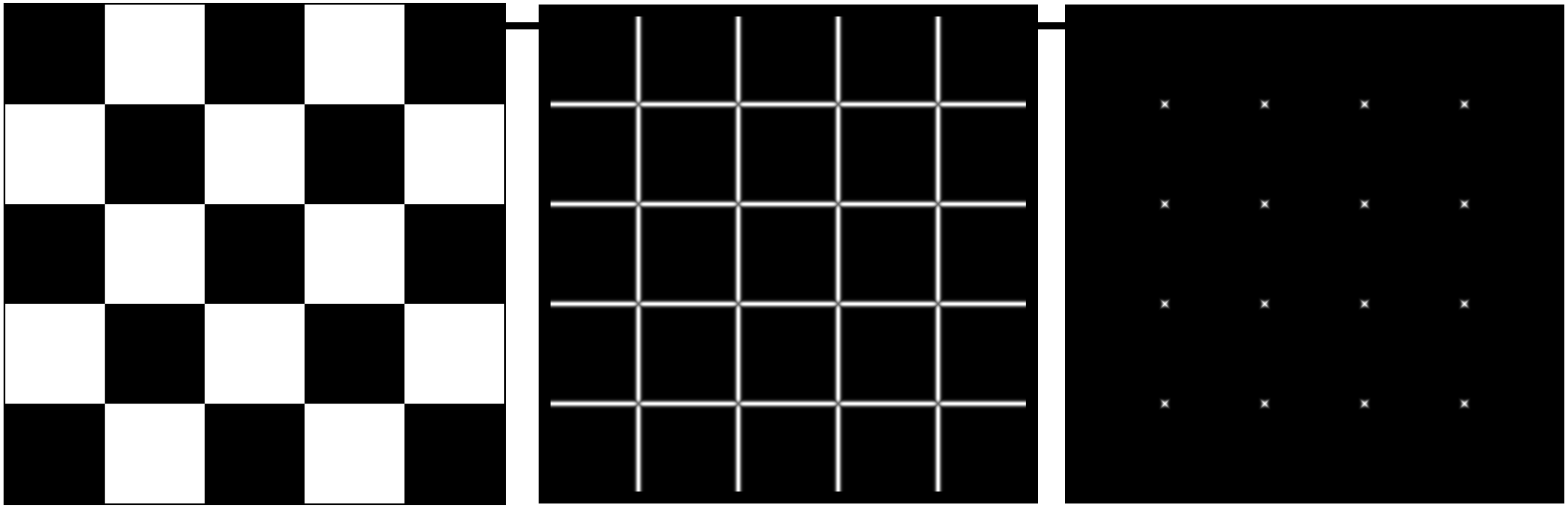
5. Non-maxima suppression



I $\lambda_{\max}$ $\lambda_{\min}$

Yet another option: $$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

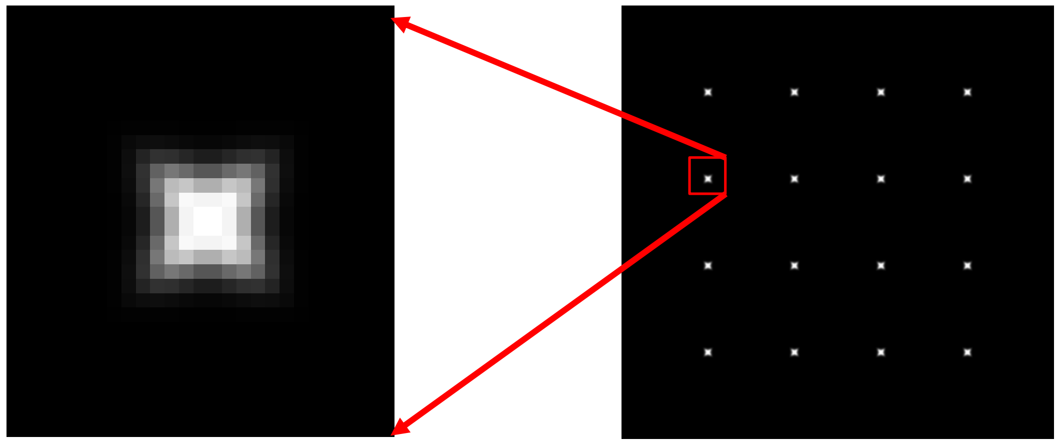
How do you write this equivalently using determinant and trace?



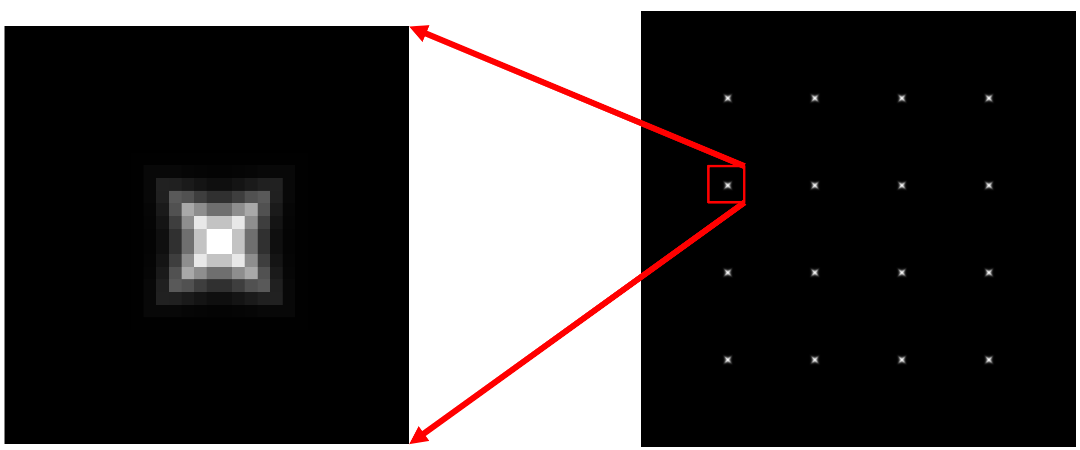
I $\lambda_{\max}$ $\lambda_{\min}$

Yet another option: $$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{determinant(H)}{trace(H)}$$

Different criteria



Harris criterion

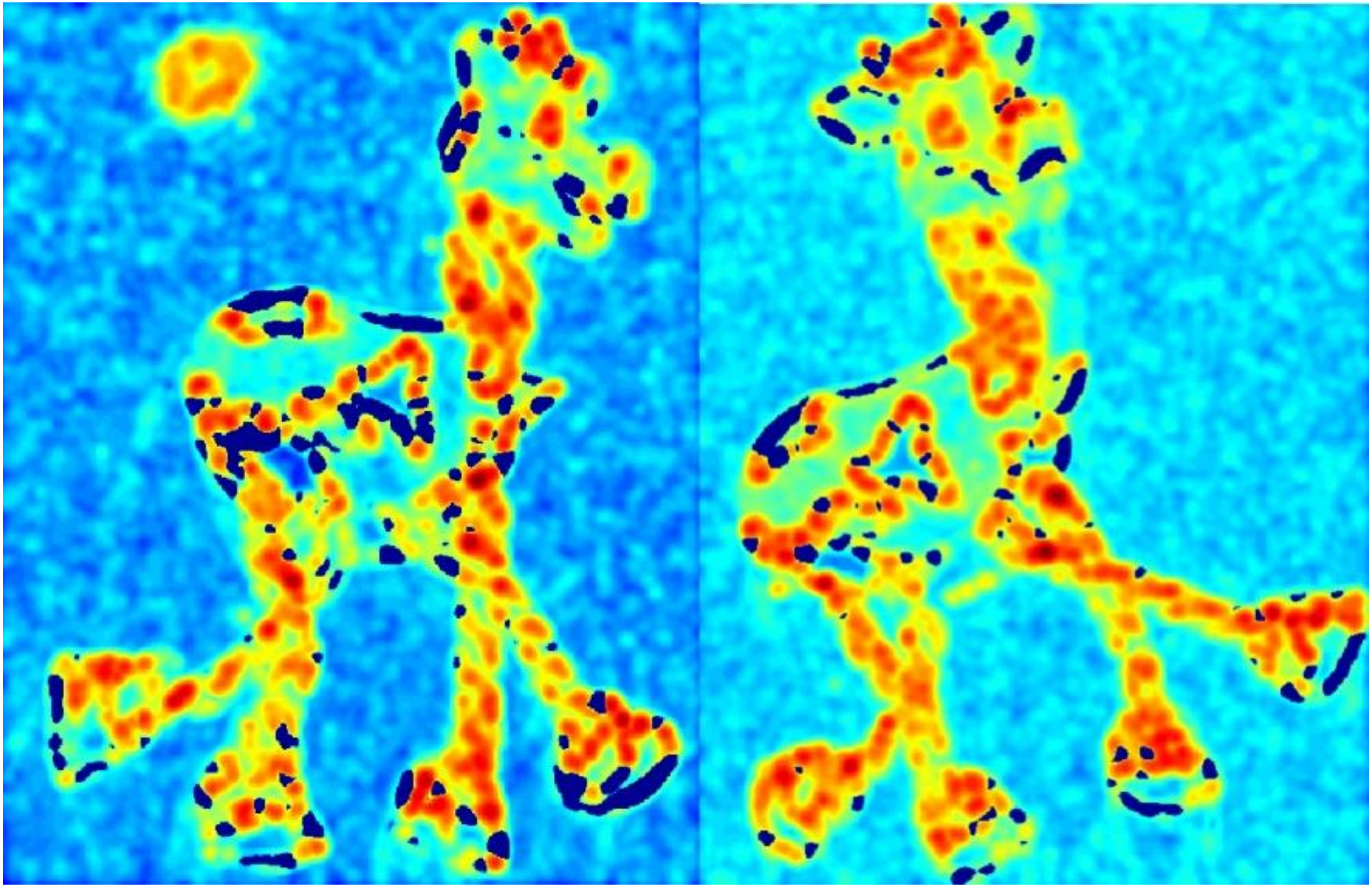


$\lambda_{\min}$

Harris Detector: Steps

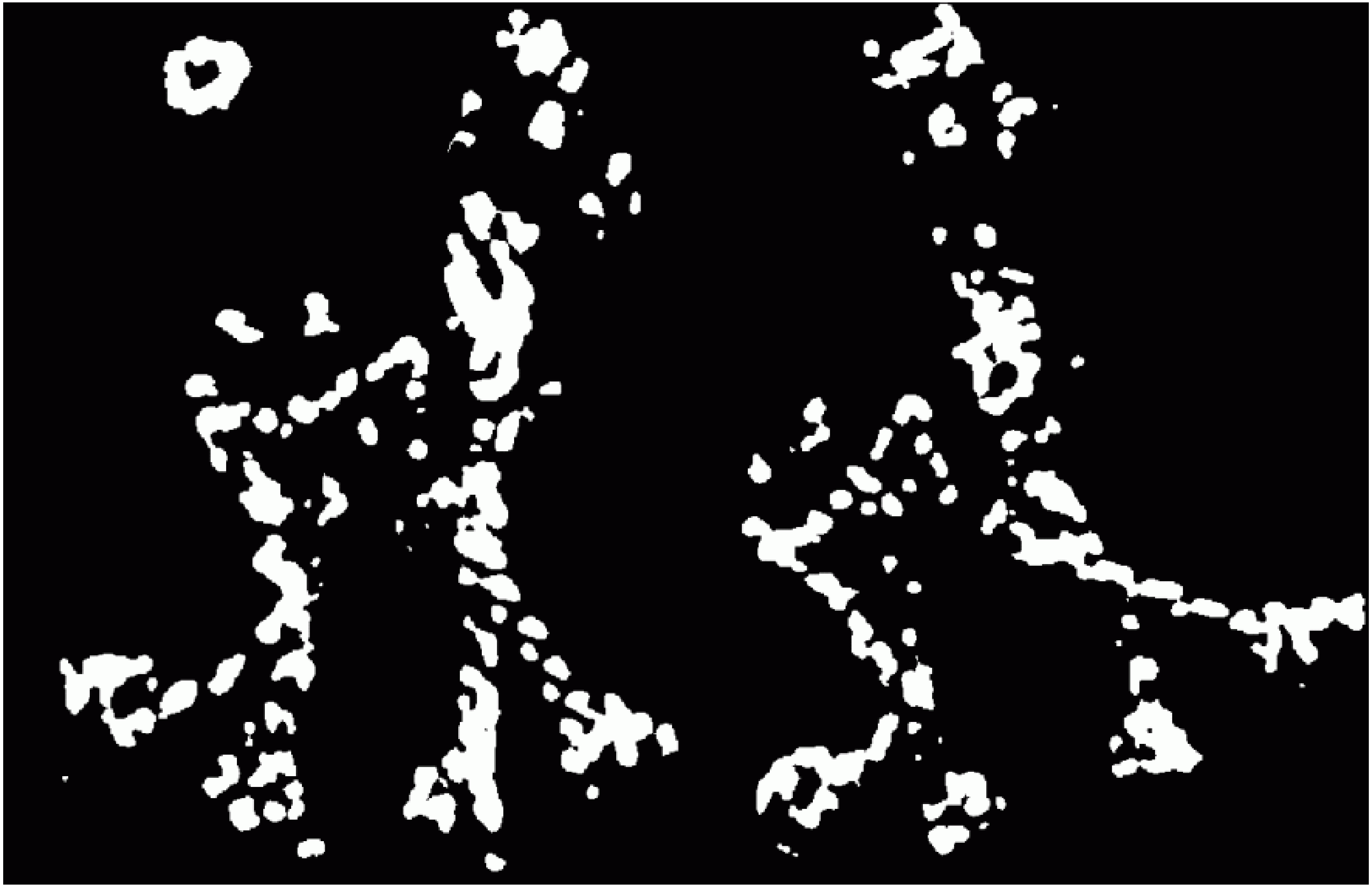
Harris Detector: Steps

Compute corner response R



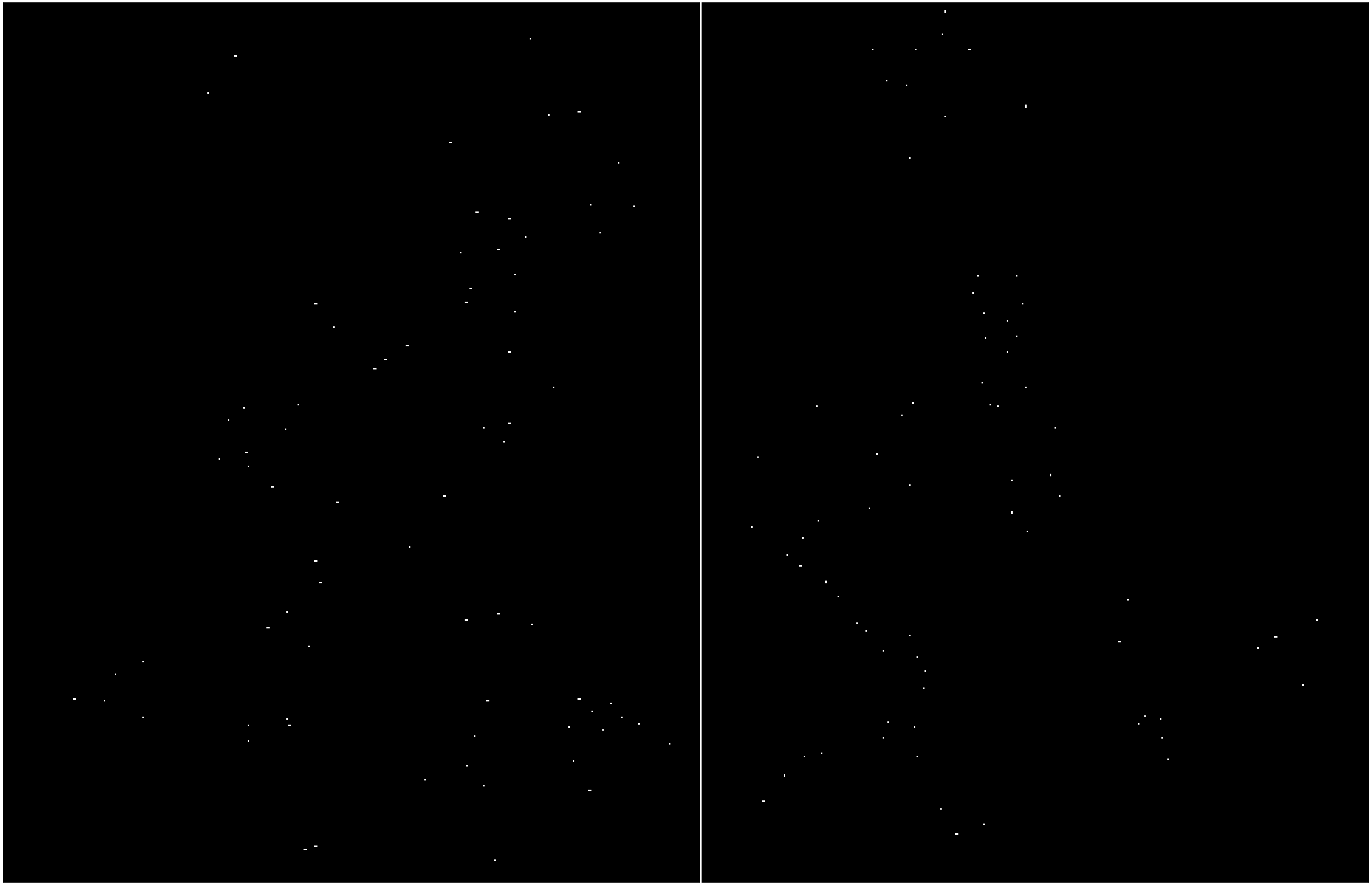
Harris Detector: Steps

Find points with large corner response: R>threshold



Harris Detector: Steps

Take only the points of local maxima of R

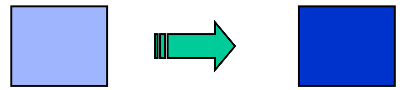


Harris Detector: Steps

Invariance and covariance

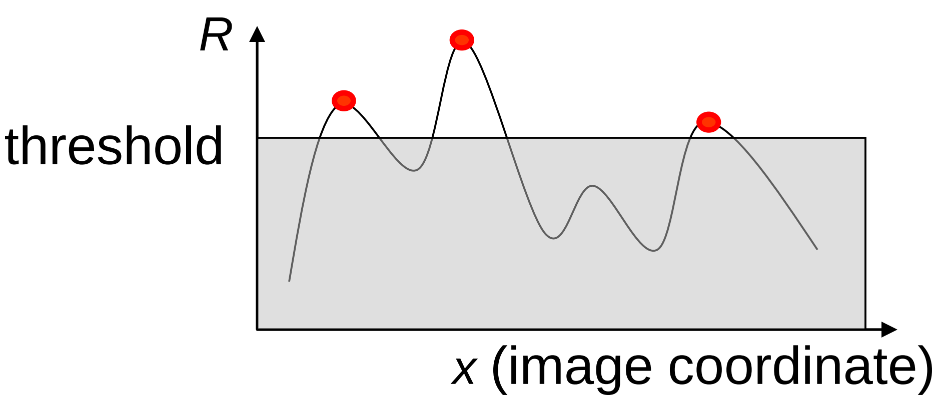
- We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations
  - **Invariance:** image is transformed and corner locations do not change
  - **Covariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations

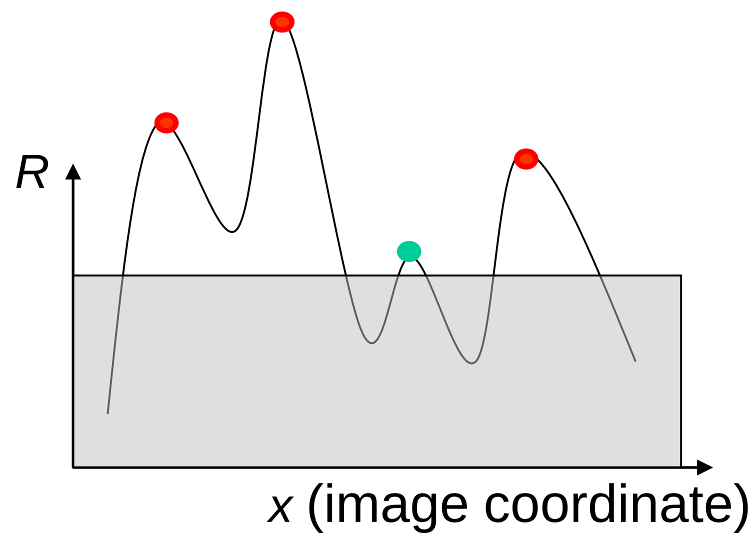
Harris Detector: Some Properties

- Partial invariance to additive and multiplicative intensity changes  $I \to a\,I + b$

✓ Only derivatives used => invariance to intensity shift $I \to I + b$

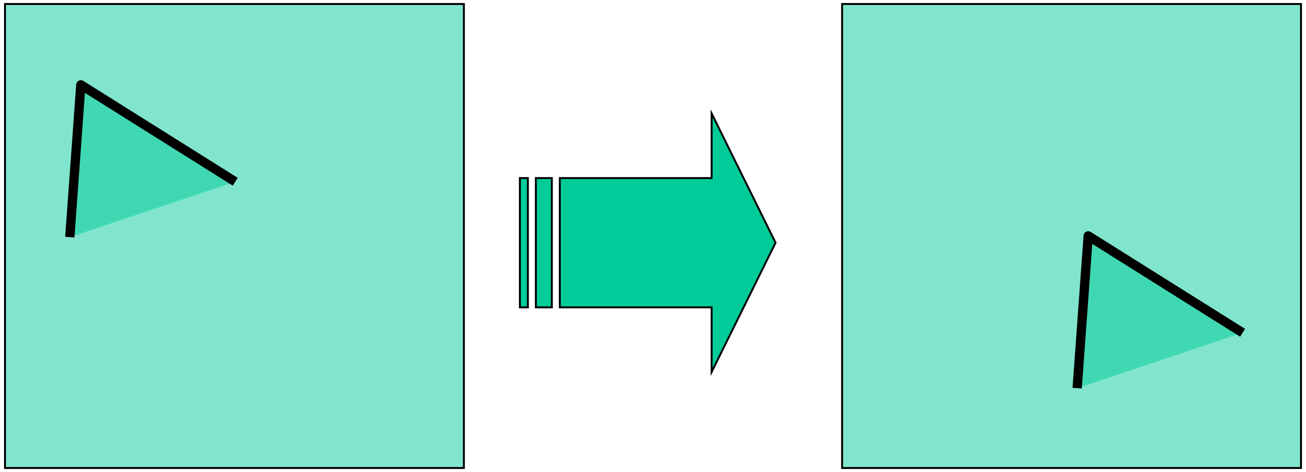
✓ Intensity scale: $I \to a\,I$ Because of fixed intensity threshold on local maxima, only partial invariance to multiplicative intensity changes.





Harris Detector: Some Properties

Image translation

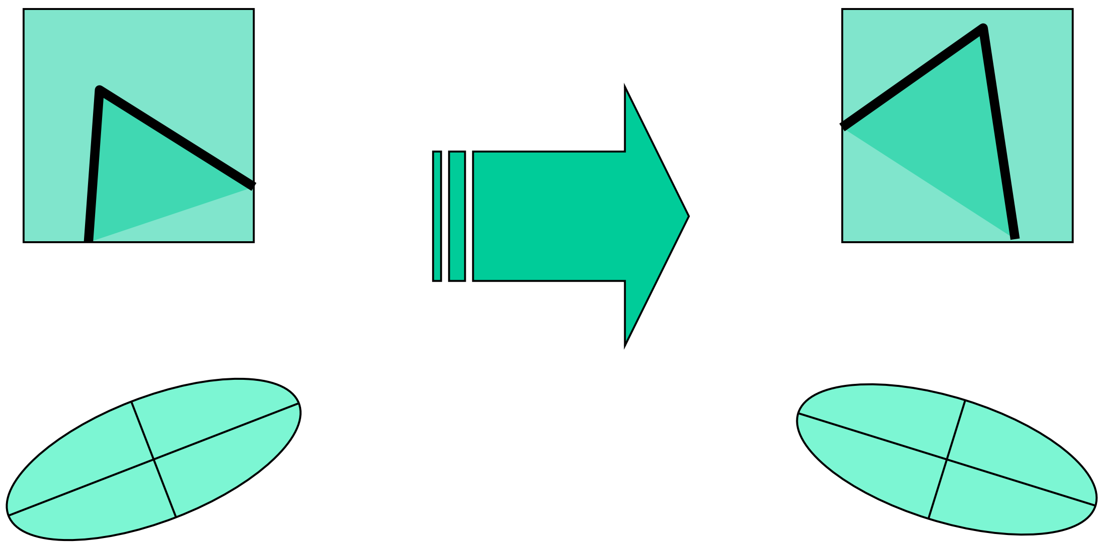


- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation

Harris Detector: Some Properties

Image Rotation

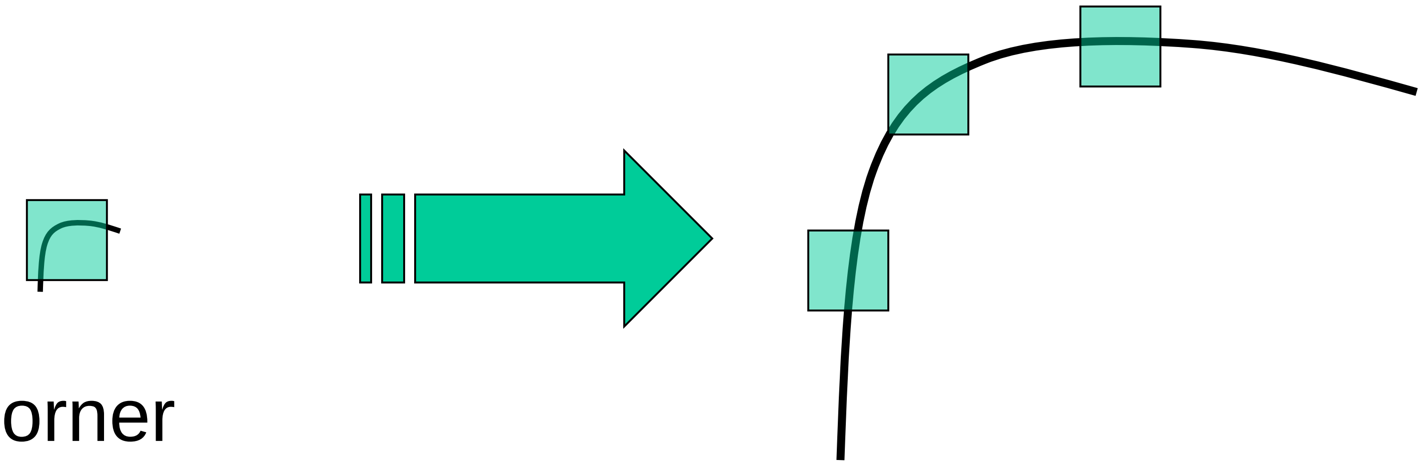


Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation

Harris Detector: Some Properties

Image Scaling

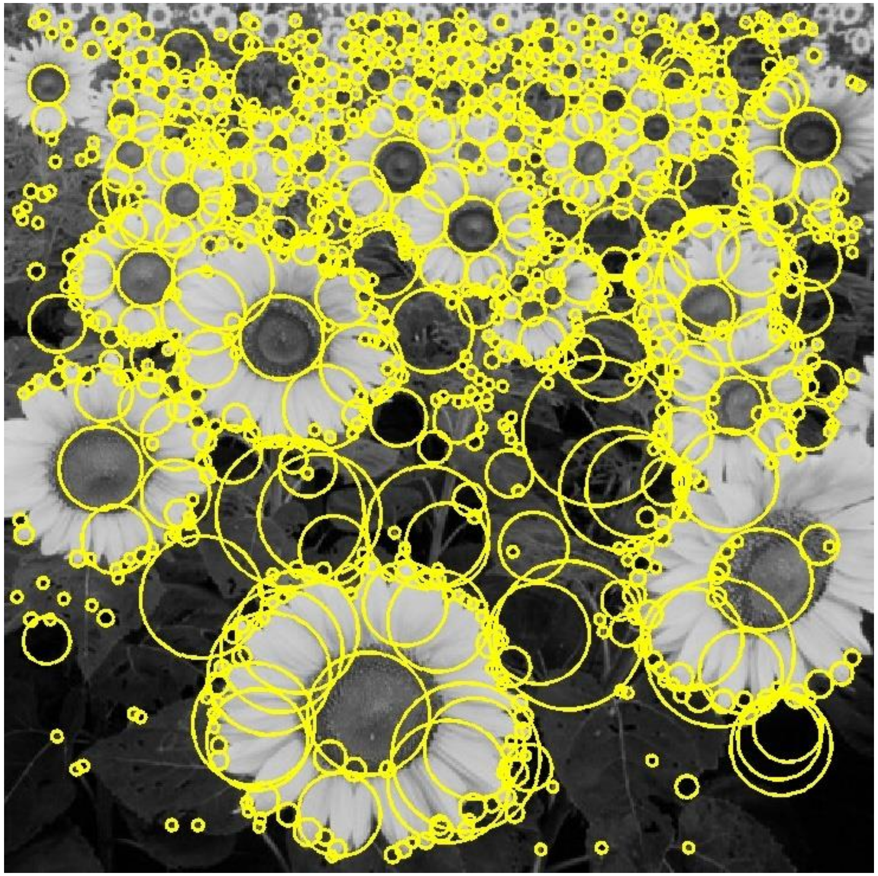


Corner

All points will be classified as edges

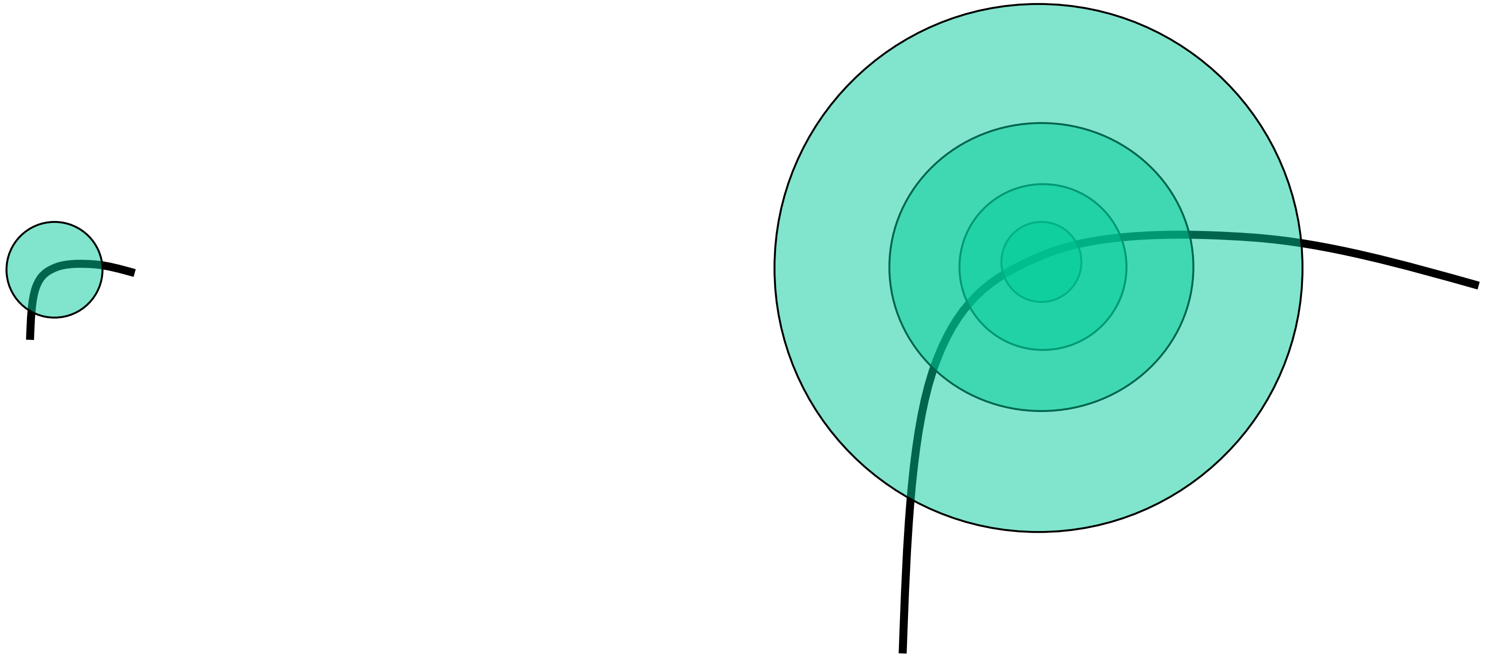
Corner location is not covariant to scaling!

Blob Detection



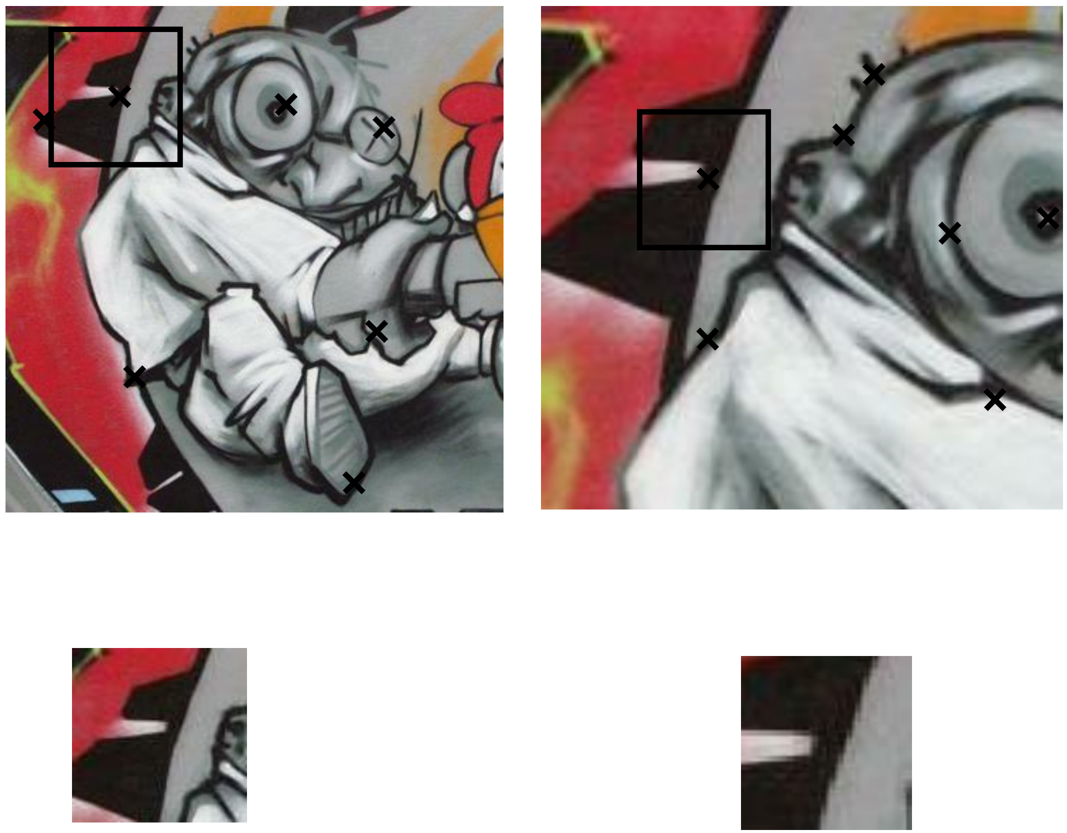
Scale Invariant Detection

- Consider regions (e.g. circles, square) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Exhaustive search

Multi-scale approach



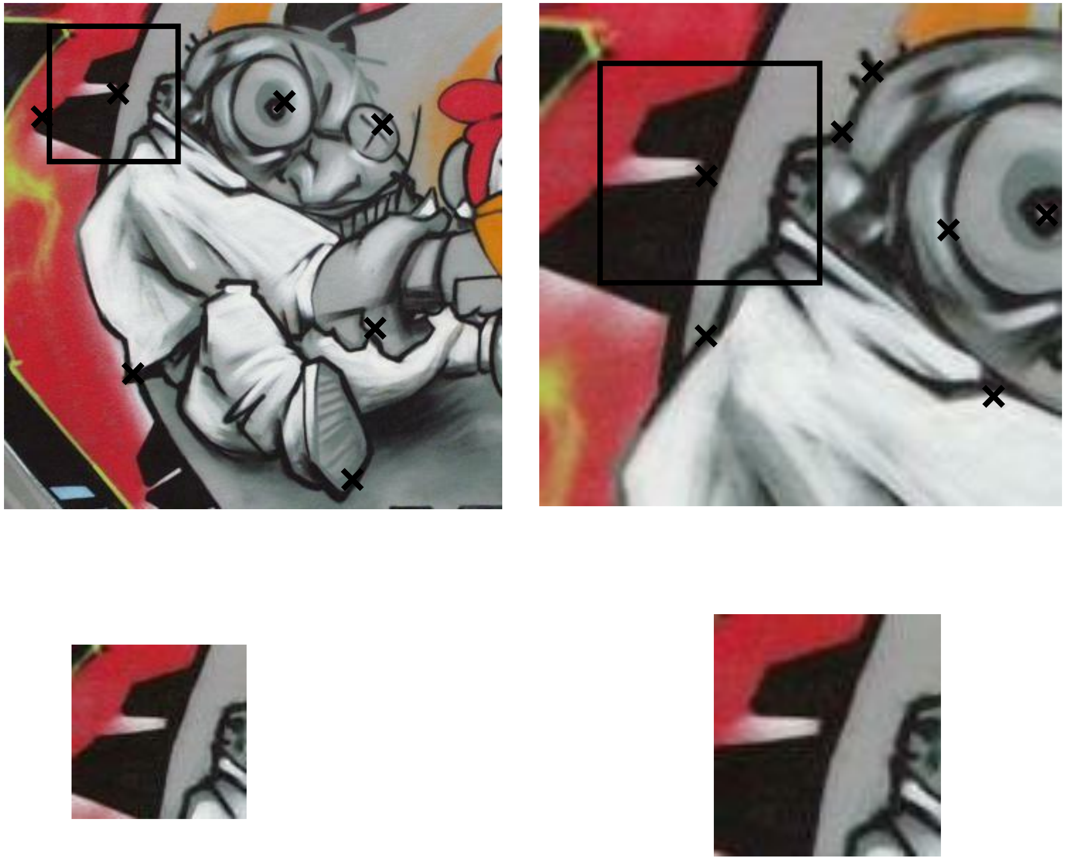
Slide from T. Tuytelaars ECCV 2006 tutorial

Exhaustive search

Multi-scale approach

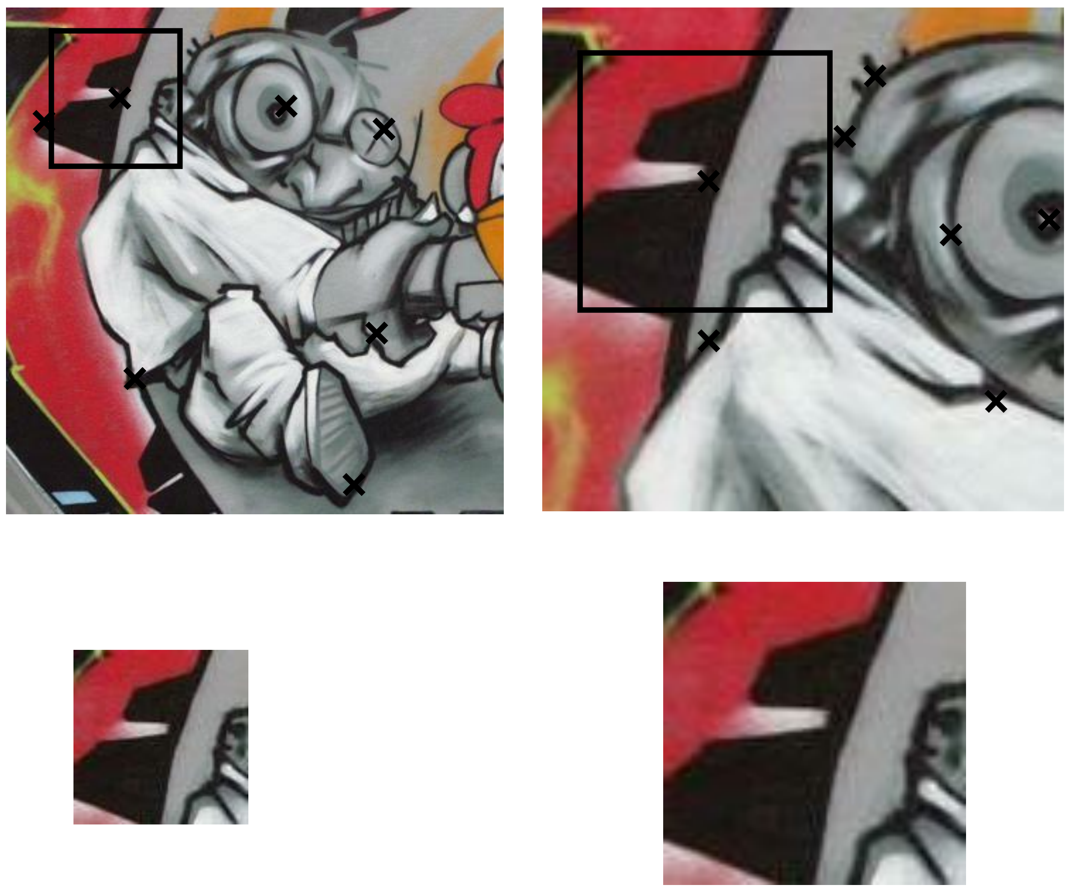
Exhaustive search

Multi-scale approach



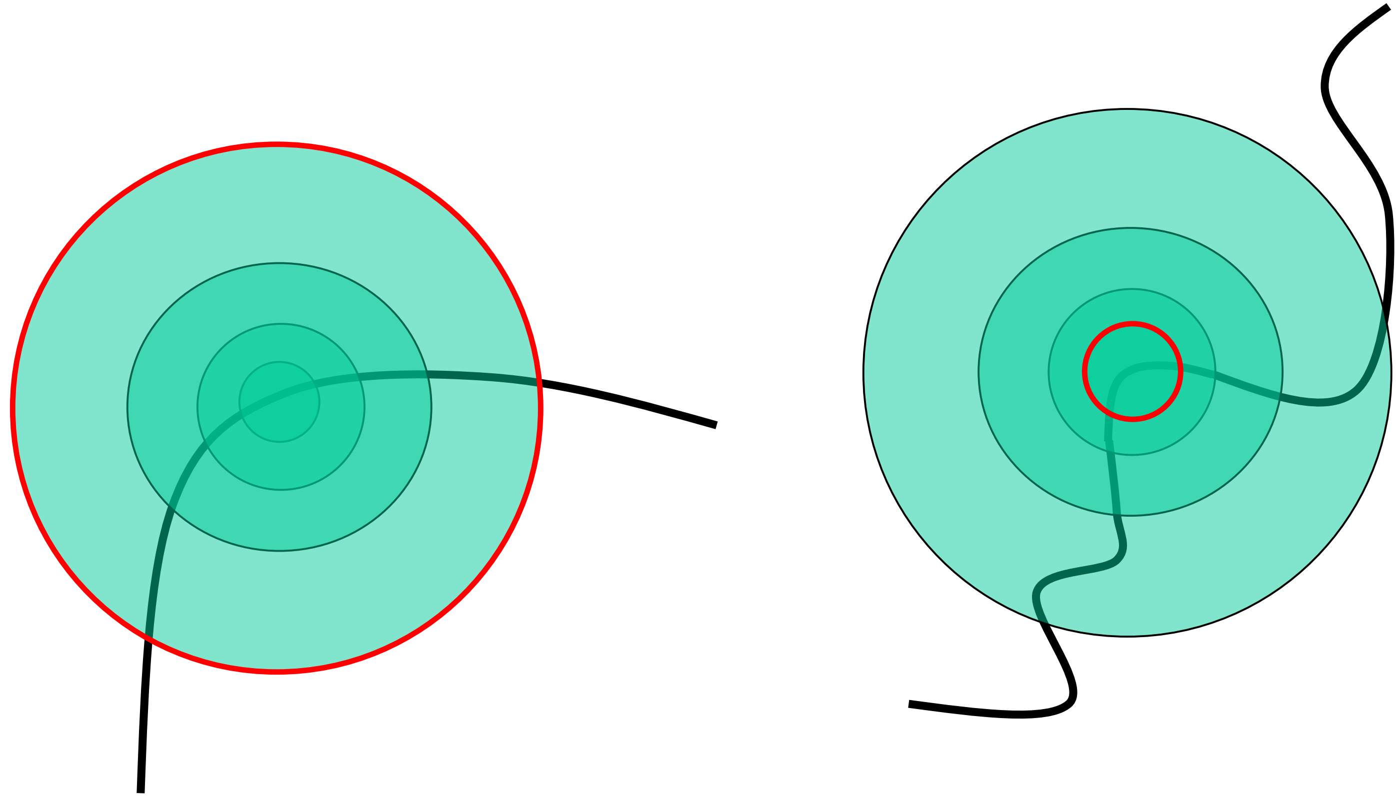
Exhaustive search

Multi-scale approach



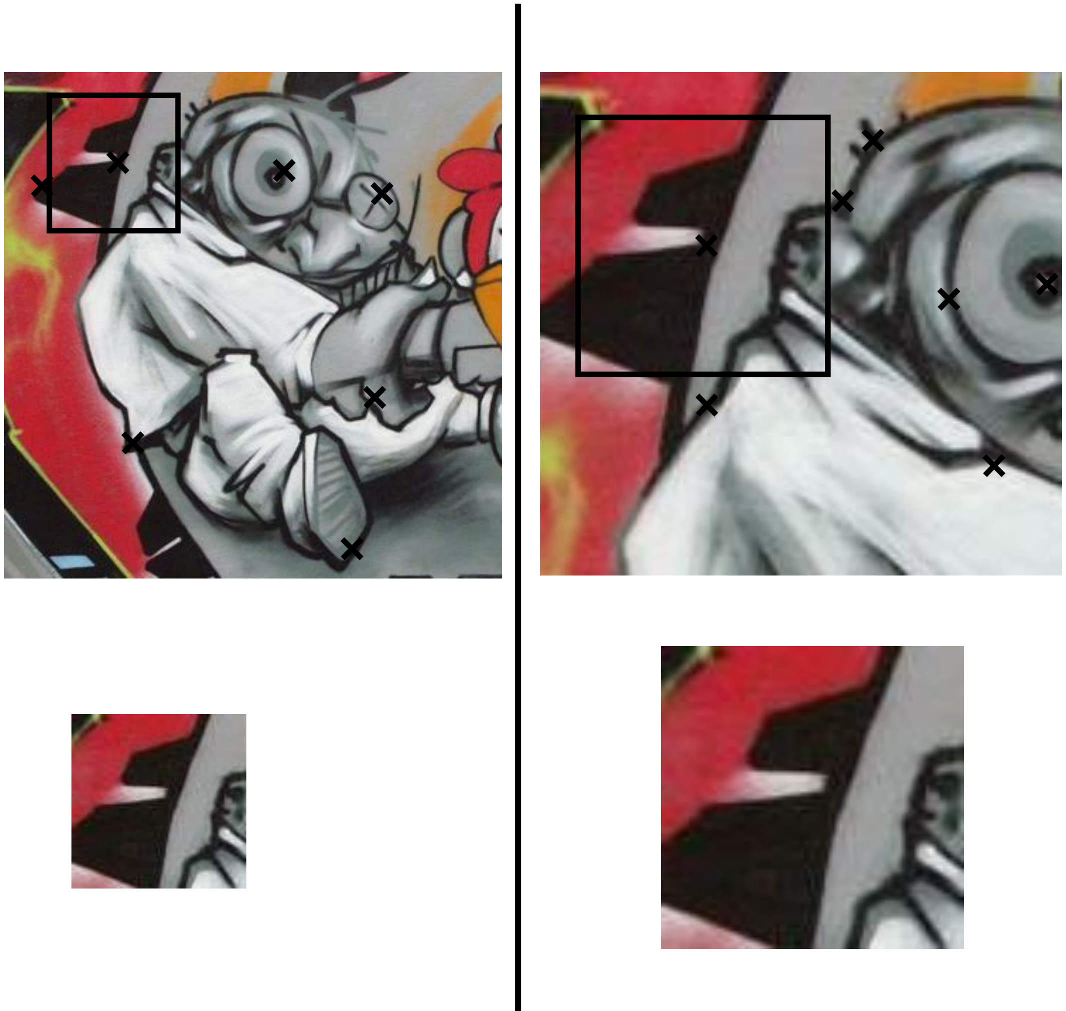
Scale Invariant Detection

The problem: how do we choose corresponding region ***independently*** in each image?



Invariance

Extract patch from each image individually



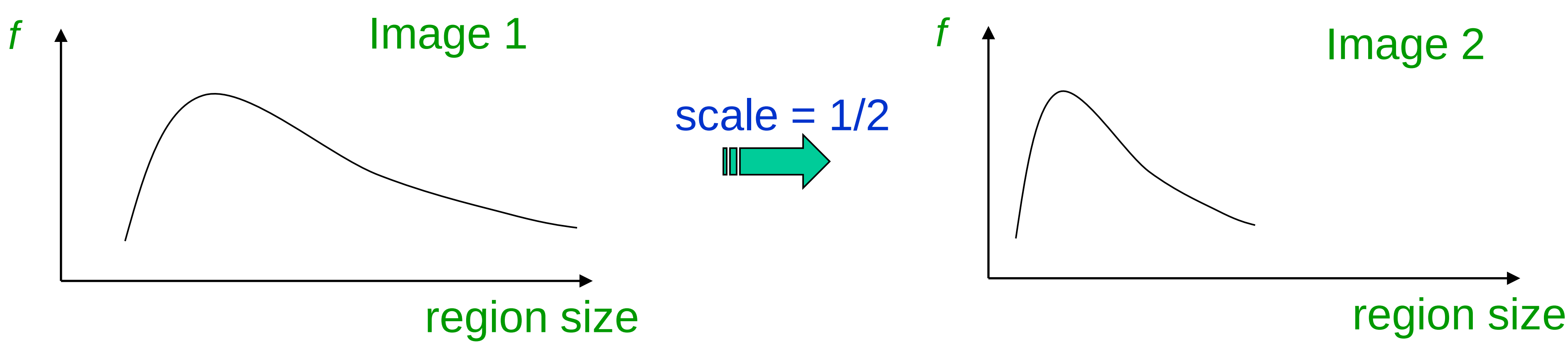
Scale Invariant Detection

Solution:

- Design a function on the region (circle), 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 (circle radius)



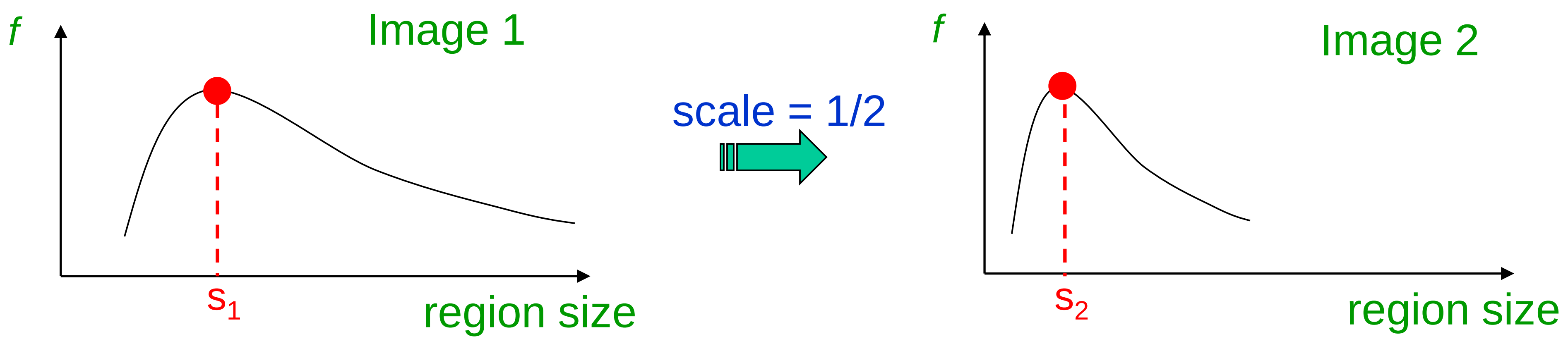
Scale Invariant Detection

Common approach:

Take a local maximum of this function

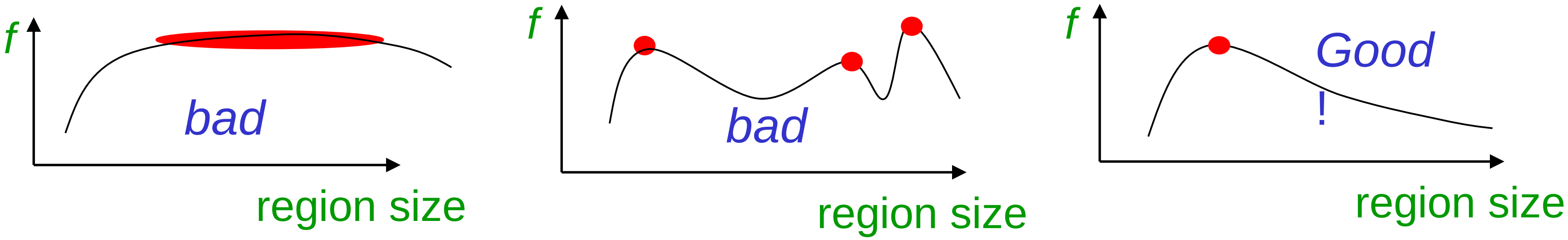
Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

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



Scale Invariant Detection

A “good” function for scale detection:
has one stable sharp peak



- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)

Automatic Scale Selection

Function responses for increasing scale (scale signature)

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

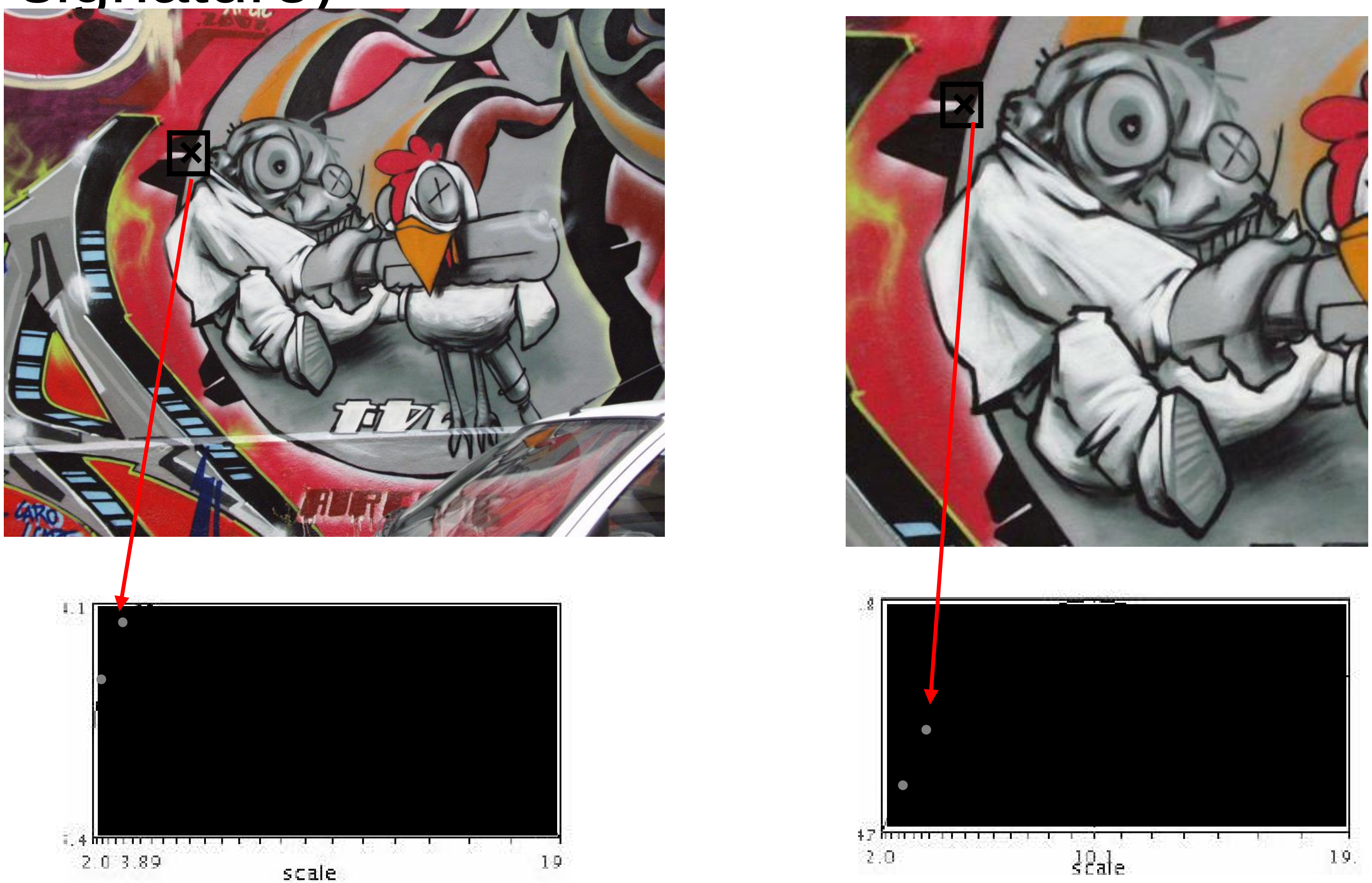
$f(I_{i_1 \dots i_m}(x', \sigma))$

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Automatic Scale Selection

Function responses for increasing scale (scale signature)



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

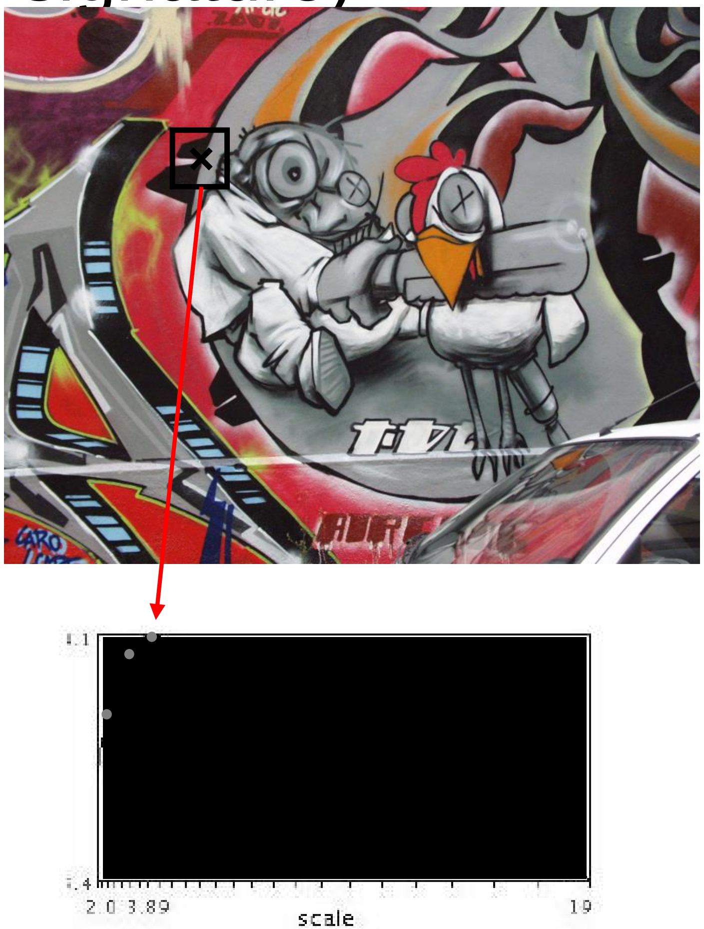
$f(I_{i_1 \ldots i_m}(x', \sigma))$

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Automatic Scale Selection

Function responses for increasing scale (scale signature)



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

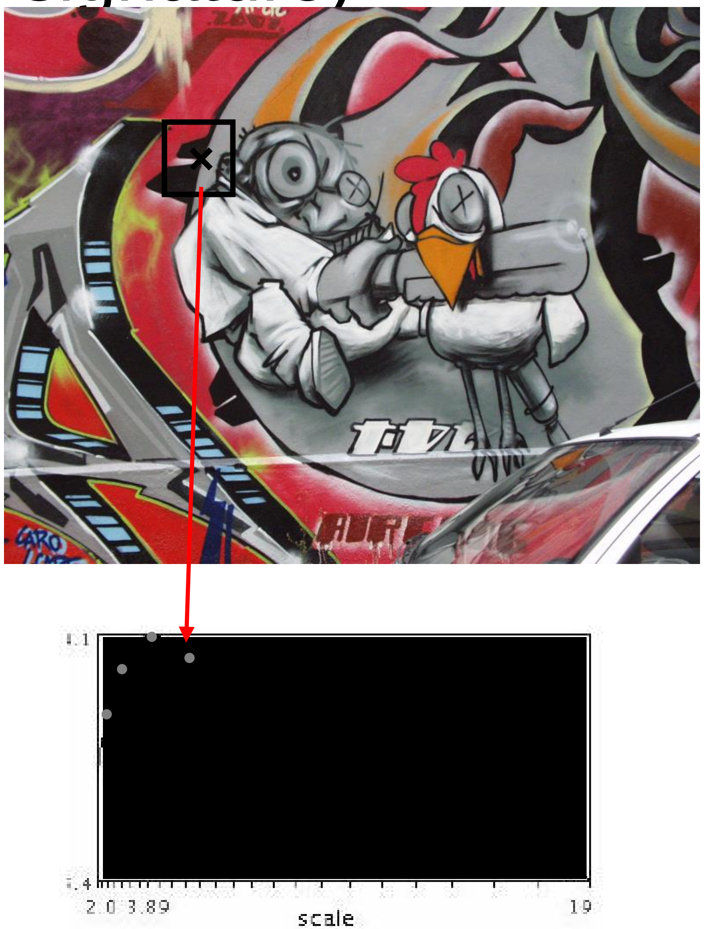
$f(I_{i_1 \ldots i_m}(x', \sigma))$

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Automatic Scale Selection

Function responses for increasing scale (scale signature)



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

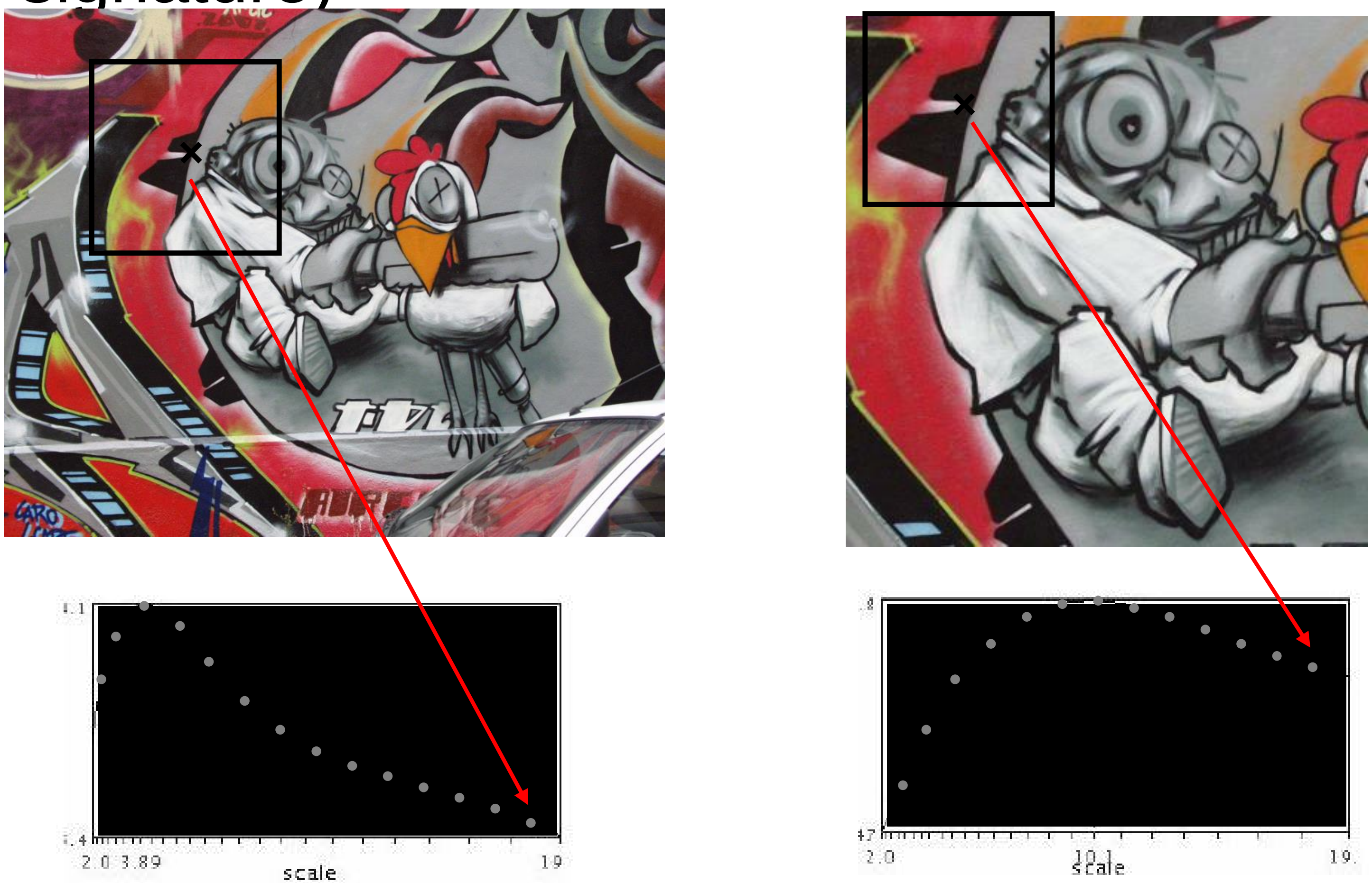
$f(I_{i_1 \dots i_m}(x', \sigma))$

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Automatic Scale Selection

Function responses for increasing scale (scale signature)



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

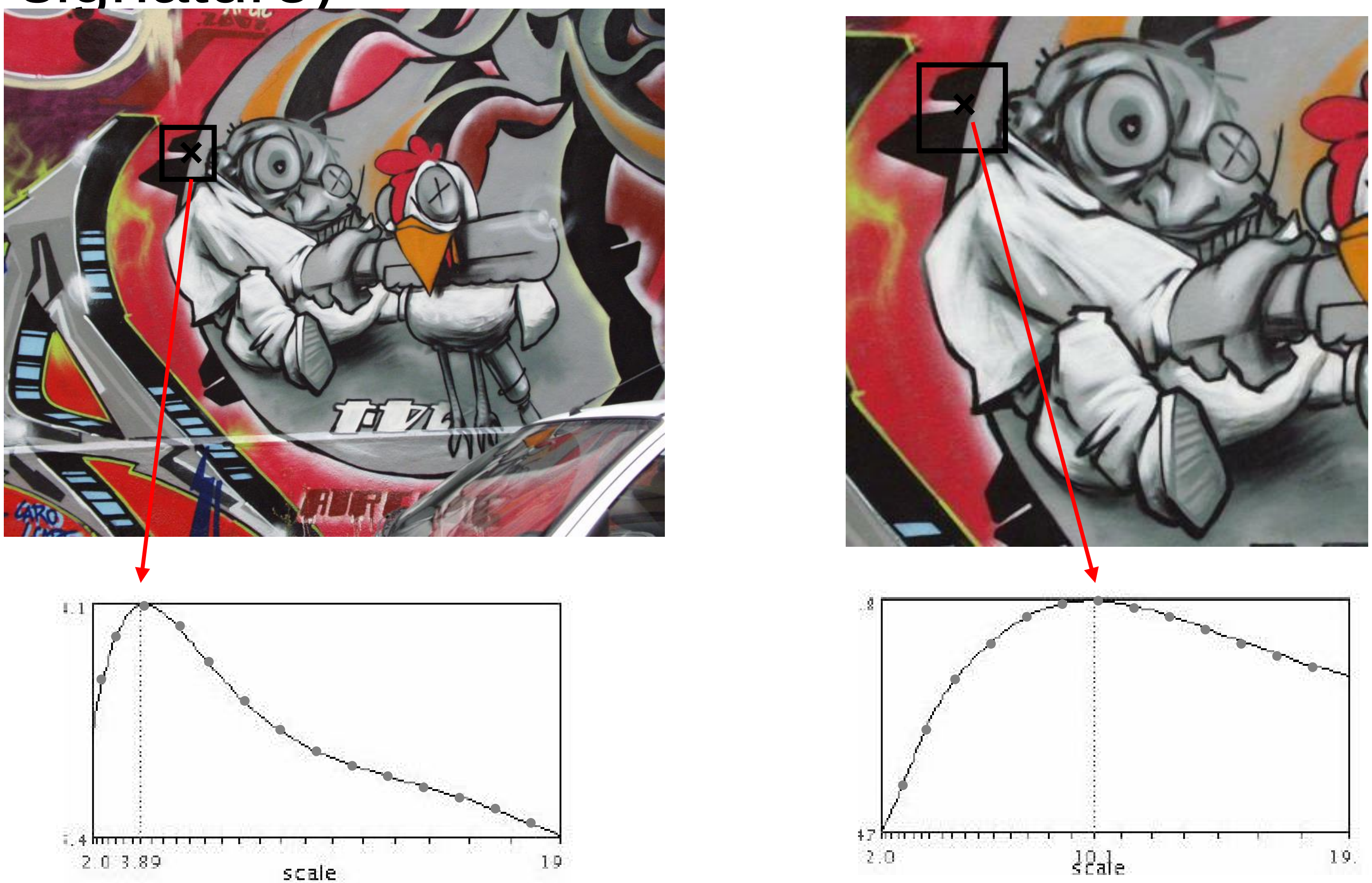
$f(I_{i_1 \ldots i_m}(x', \sigma))$

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Automatic Scale Selection

Function responses for increasing scale (scale signature)



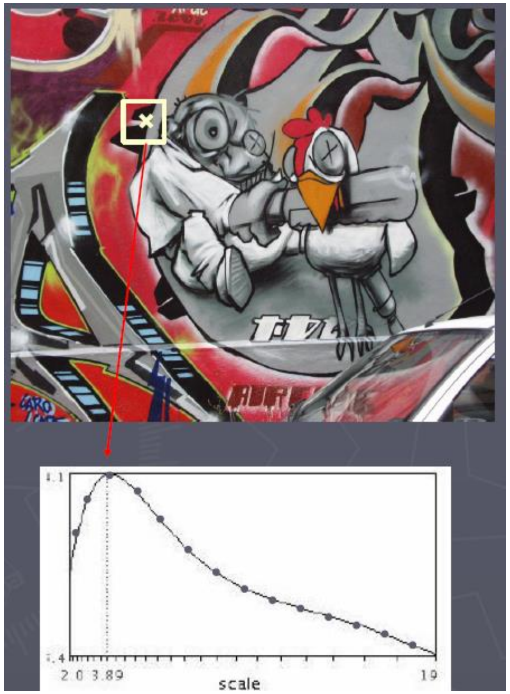
$f(I_{i_1 \ldots i_m}(x,\sigma))$

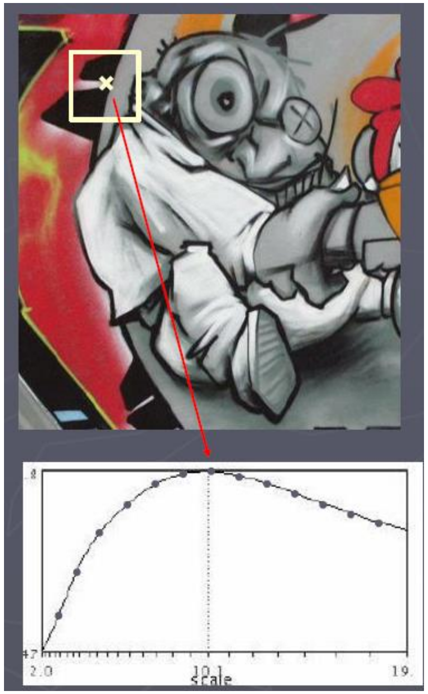
$f(I_{i_1 \ldots i_m}(x',\sigma'))$

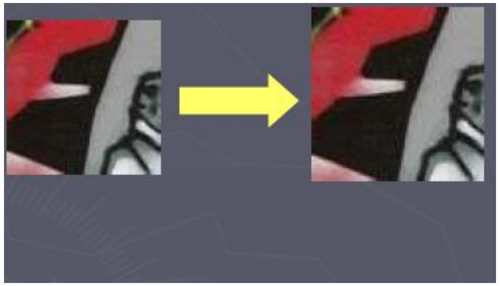
K. Grauman, B. Leibe

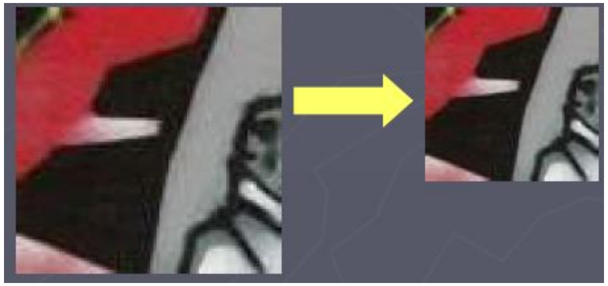
Scale selection

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





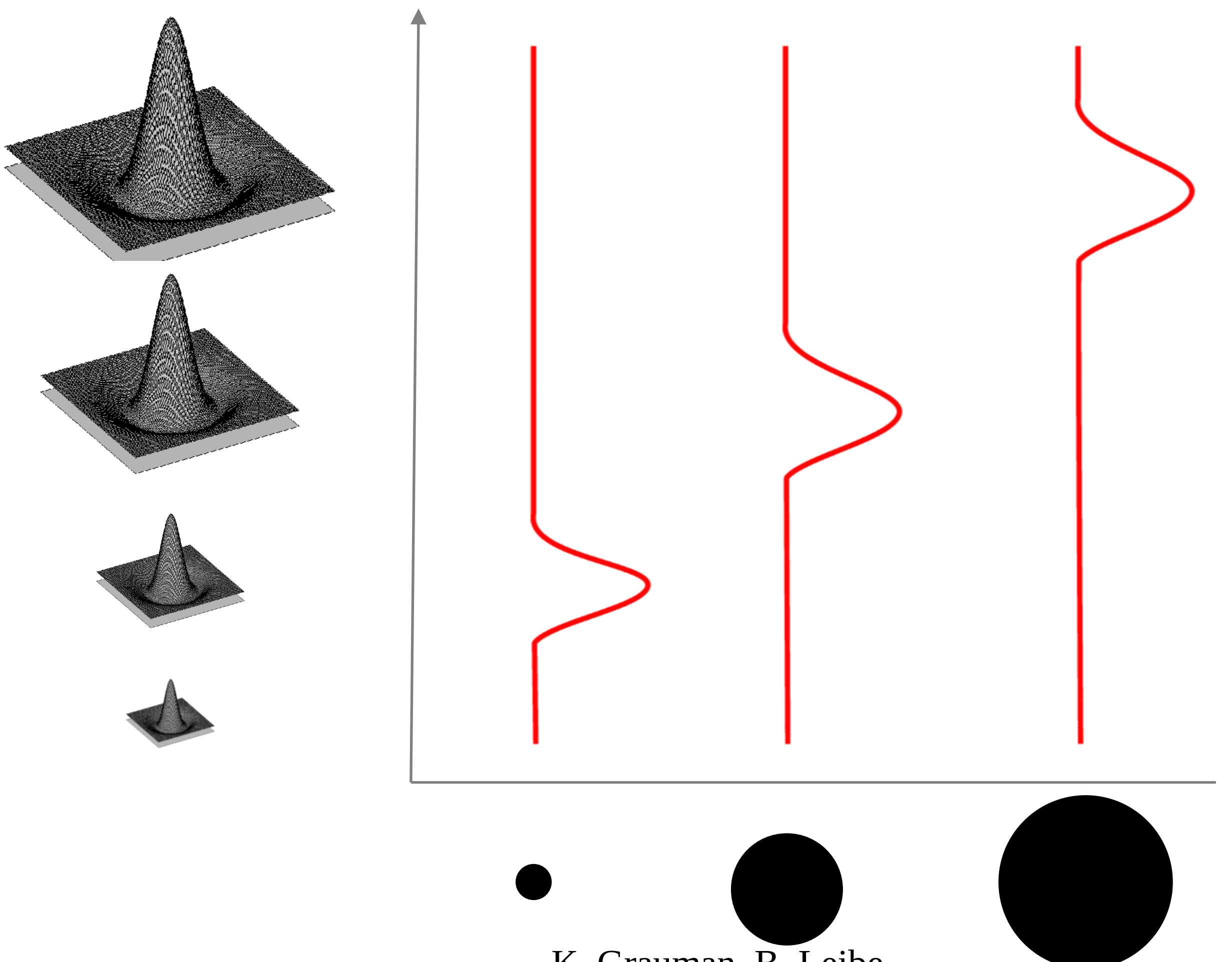




[Images from T. Tuytelaars]

What Is A Useful Signature Function?

Laplacian-of-Gaussian = “blob” detector



K. Grauman, B. Leibe

K. Grauman, B. Leibe

Scale Invariant Detection

Functions for determining scale $f = \mathrm{Kernel} * \mathrm{Image}$

Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

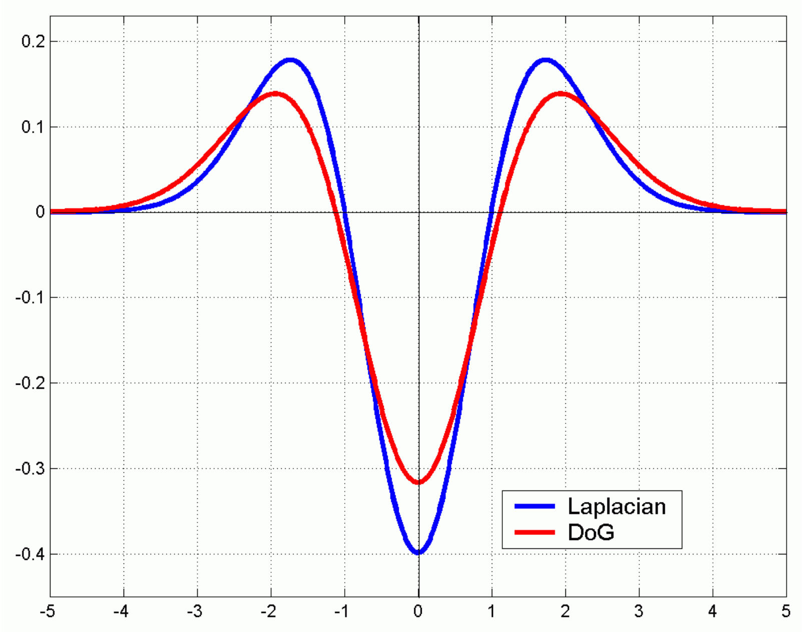
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian

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



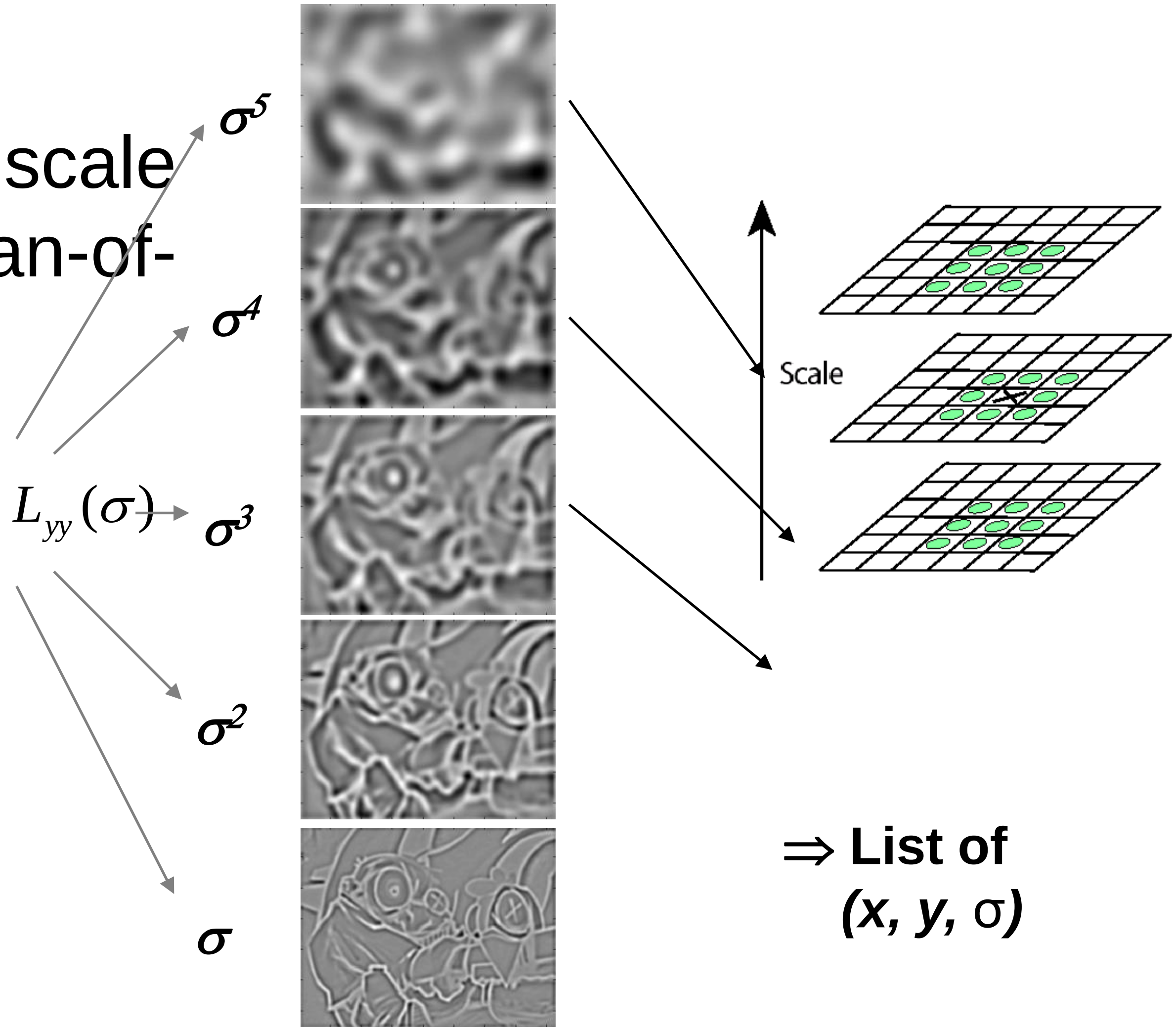
Note: both kernels are invariant to *scale* and *rotation*

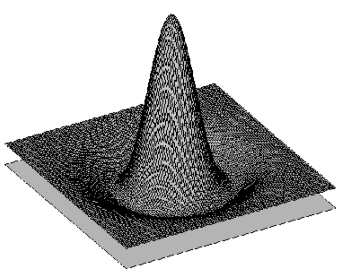
Laplacian-of-Gaussian (LoG)

Interest points:

Local maxima in scale space of Laplacian-of-Gaussian

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$





⇒ **List of *(x, y, σ)***

K. Grauman, B. Leibe

K. Grauman, B. Leibe

Scale-space blob detector: Example

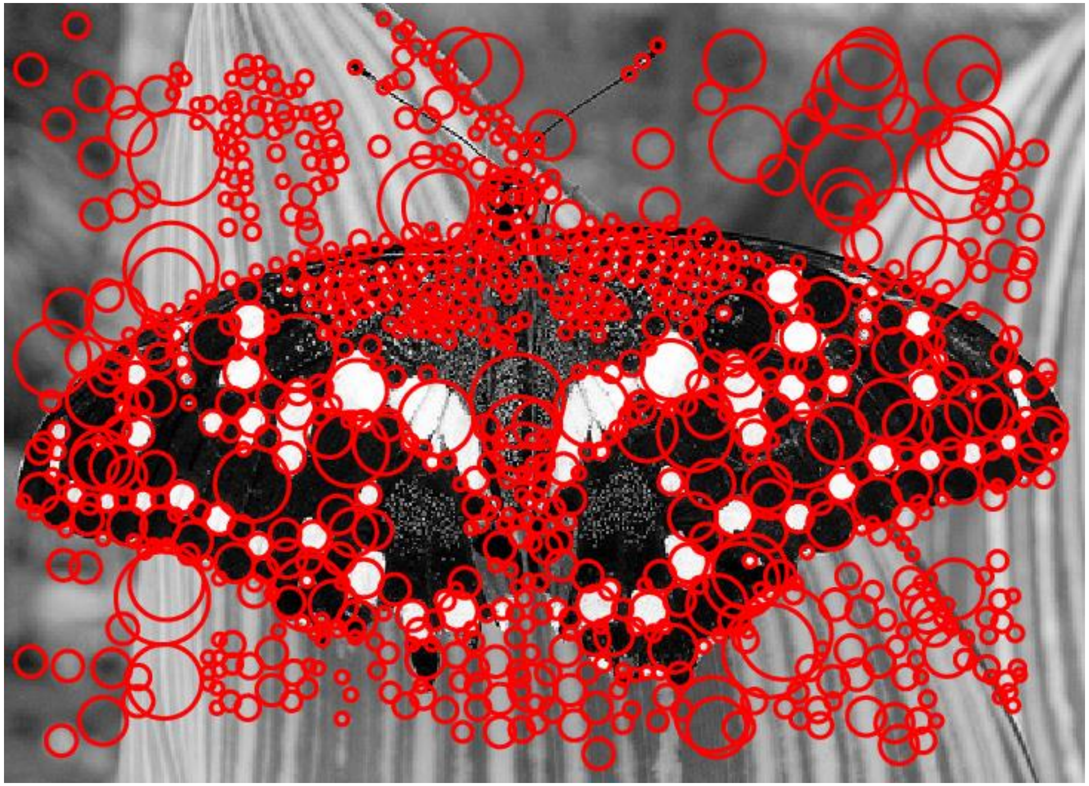
Source: Lana Lazebnik

Scale-space blob detector: Example

sigma = 11.9912

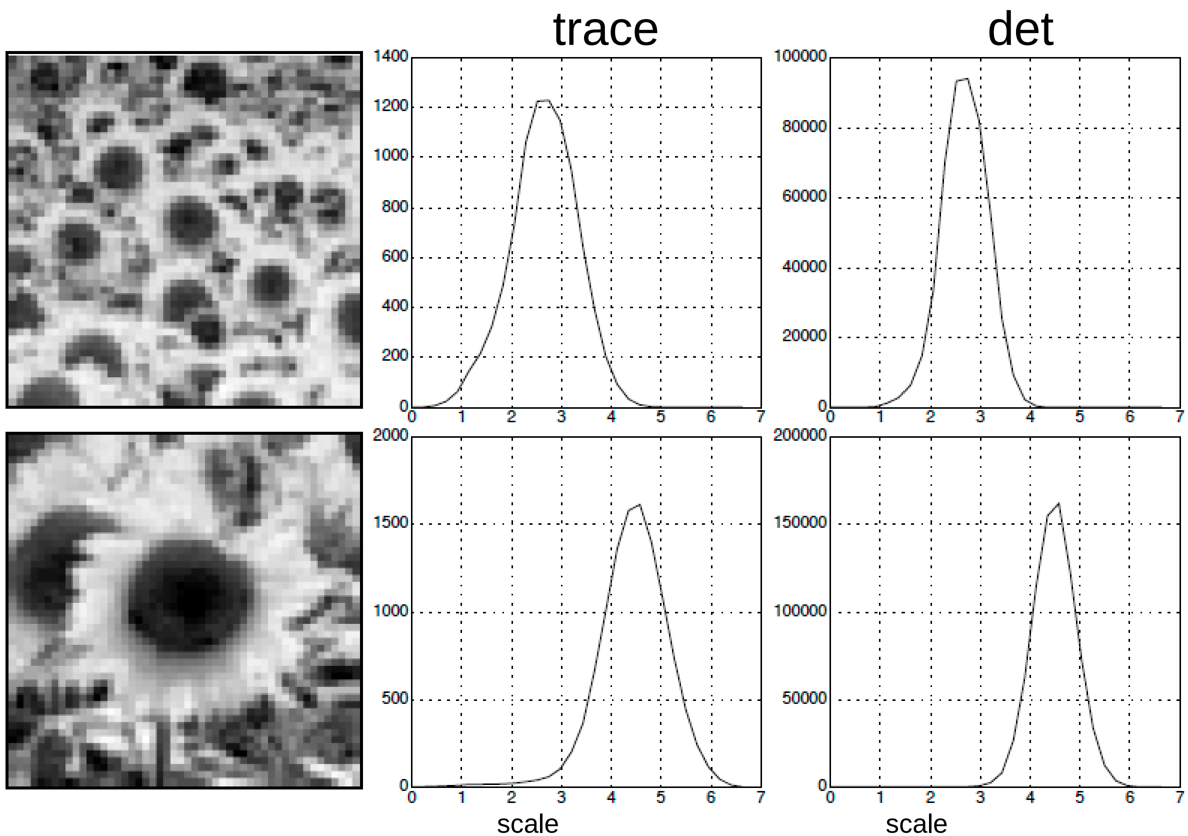
Source: Lana Lazebnik

Scale-space blob detector: Example



Source: Lana Lazebnik

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



From Lindeberg 1998

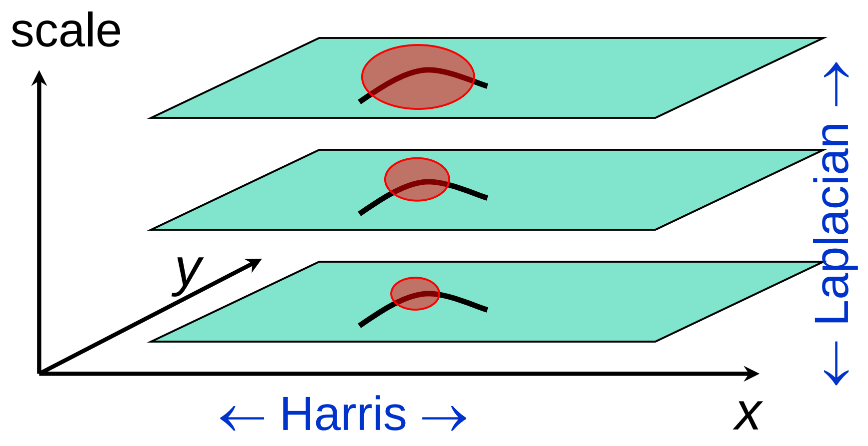
blob detection; *Marr* 1982; Voorhees and Poggio 1987; Blostein and Ahuja 1989; …

Scale Invariant Detectors

Harris-Laplacian[1]

Find local maximum of:

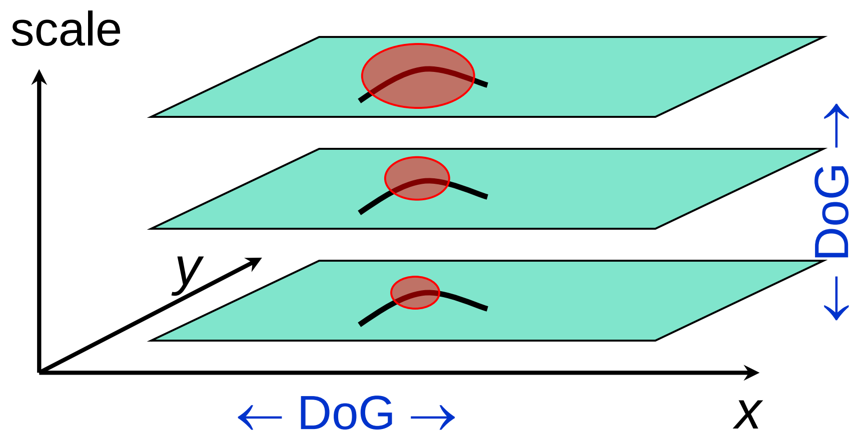
- Harris corner detector in space (image coordinates)
- Laplacian in scale



- SIFT (Lowe)[2]

  Find local maximum of:

  – Difference of Gaussians in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 200

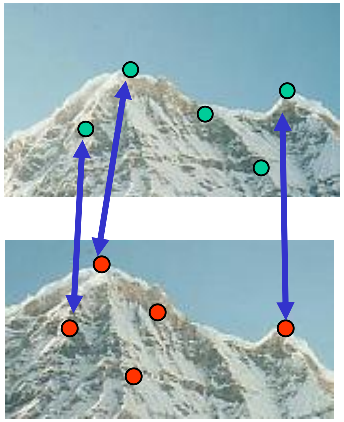
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

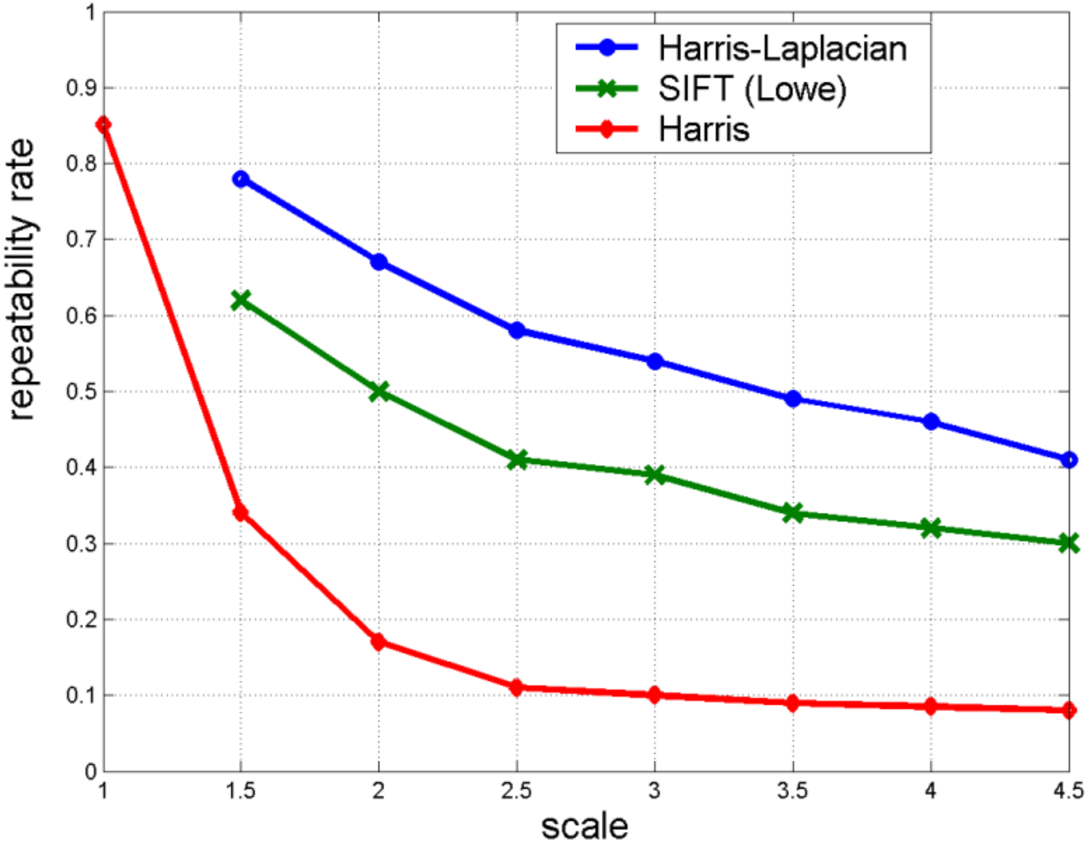
Scale Invariant Detectors

Experimental evaluation of detectors
w.r.t. scale change

Repeatability rate:

$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$





K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

Scale Invariant Detection: Summary

Given: two images of the same scene with a large *scale difference* between them

Goal: find *the same* interest points *independently* in each image

Solution: search for *maxima* of suitable functions in *scale* and in *space* (over the image)

Methods:

1. Harris-Laplacian [Mikolajczyk, Schmid]: maximize Laplacian over scale, Harris' measure of corner response over the image
2. SIFT [Lowe]: maximize Difference of Gaussians over scale and space