Local Feature Extraction for

Wide-Baseline Matching, Object Recognition and Image Retrieval Methods, Stitching and more …

Jiří Matas and Ondra Chum and James Pritts

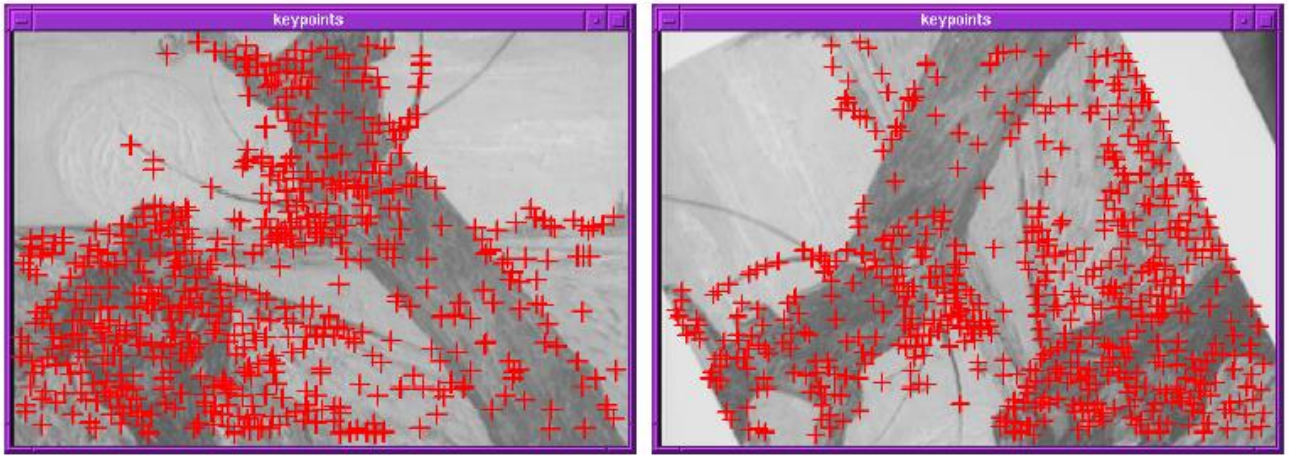
Center for Machine Perception, Czech Technical University Prague

Includes slides by:

- Darya Frolova, Denis Simakov,The Weizmann Institute of Science
- Martin Urban , Stepan Obdrzalek, Ondra Chum Center for Machine Perception Prague
- Matthew Brown,David Lowe, University of British Columbia
- Svetlana Lazenbik, University of Illinois

Feature extraction: Corners

Finding Corners

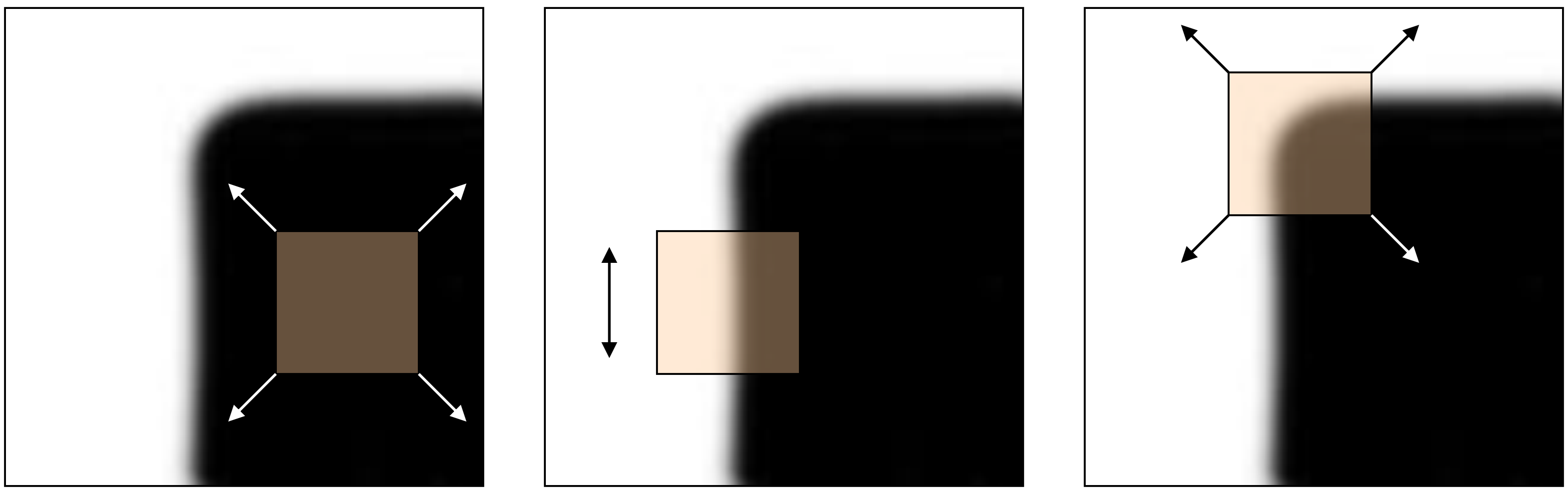


- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

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

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 intensity



"flat" region:
no change in
all directions

"edge":
no change
along the edge
direction

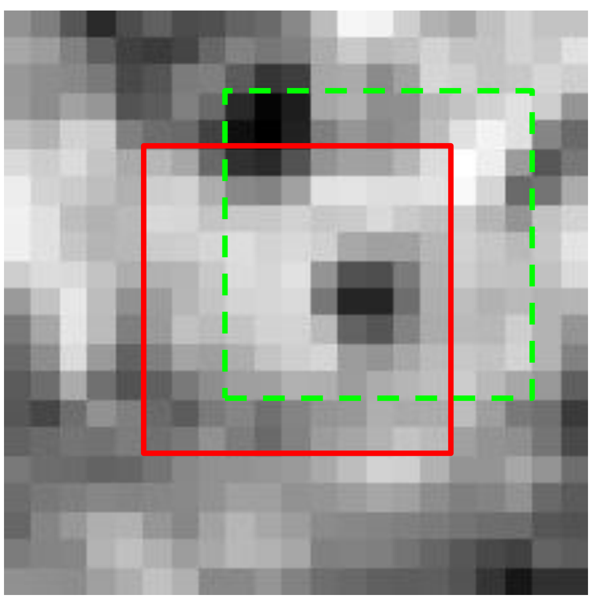
"corner":
significant
change in all
directions

Corner Detection: Mathematics

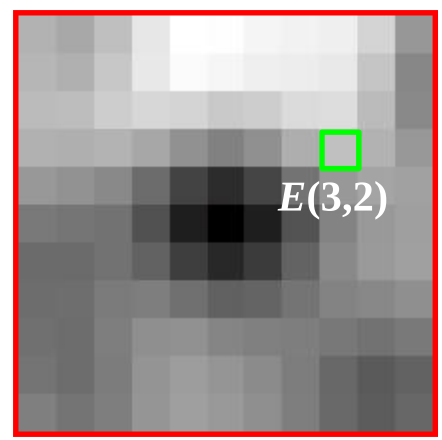
Change in appearance of window *W* for the shift [*u*,*v*]:

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

$I(x, y)$



$E(u, v)$

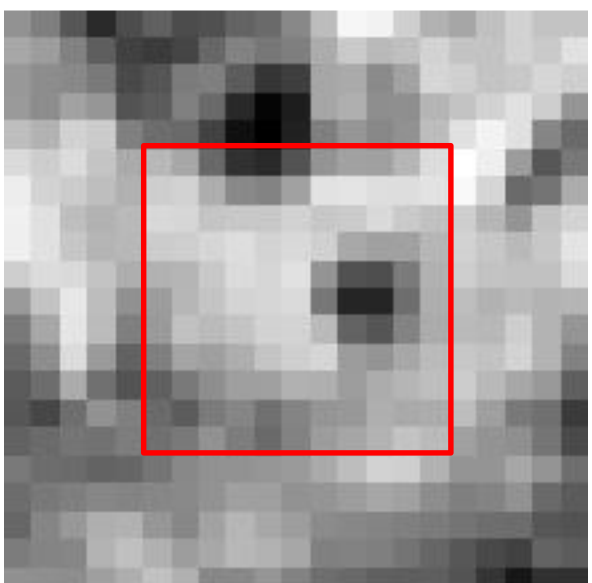


Corner Detection: Mathematics

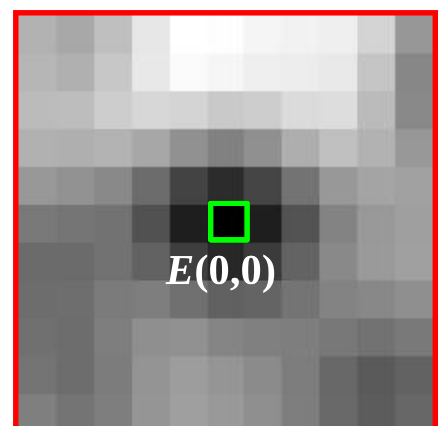
Change in appearance of window *W* for the shift [*u,v*]:

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

$I(x, y)$



$E(u, v)$



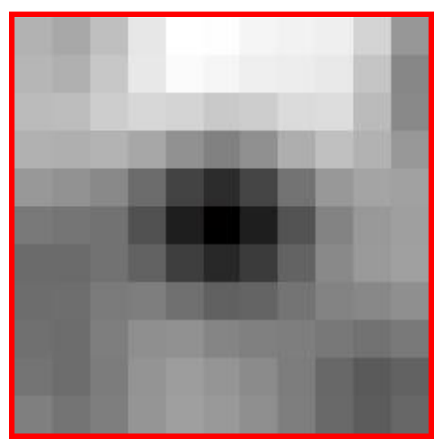
Corner Detection: Mathematics

Change in appearance of window *W* for the shift [*u*,*v*]:

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

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

$E(u, v)$



Corner Detection: Mathematics

- First-order Taylor approximation for small motions [*u*, *v*]:

$$I(x+u, y+v) \approx I(x,y) + I_x u + I_y v$$

- Let's plug this into *E*(*u*,*v*):

$$\begin{aligned} E(u,v) &= \sum_{(x,y)\in W} [I(x+u,y+v) - I(x,y)]^2 \\ &\approx \sum_{(x,y)\in W} [I(x,y) + I_x u + I_y v - I(x,y)]^2 \\ &= \sum_{(x,y)\in W} [I_x u + I_y v]^2 = \sum_{(x,y)\in W} I_x^2 u^2 + 2 I_x I_y uv + I_y^2 v^2 \end{aligned}$$

***WAIT! Why not just maximize E(u,v) directly?

Corner Detection: Mathematics

CENTER FOR MACHINE PERCEPTION

The quadratic approximation can be written as

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

where *M* is a *second moment matrix* computed from image derivatives:

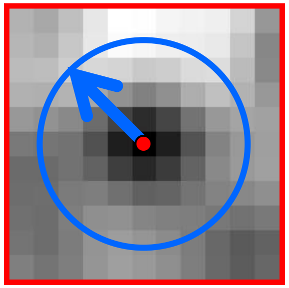
$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix}$$

(the sums are over all the pixels in the window *W*)

Interpreting the second moment matrix

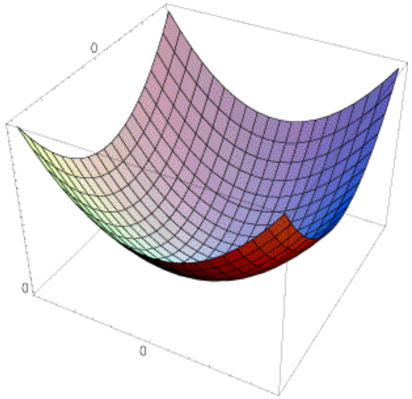
- The surface $E(u,v)$ is locally approximated by a quadratic form. Let's try to understand its shape.
  - Specifically, in which directions does it have the smallest/greatest change?

$E(u,v)$



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

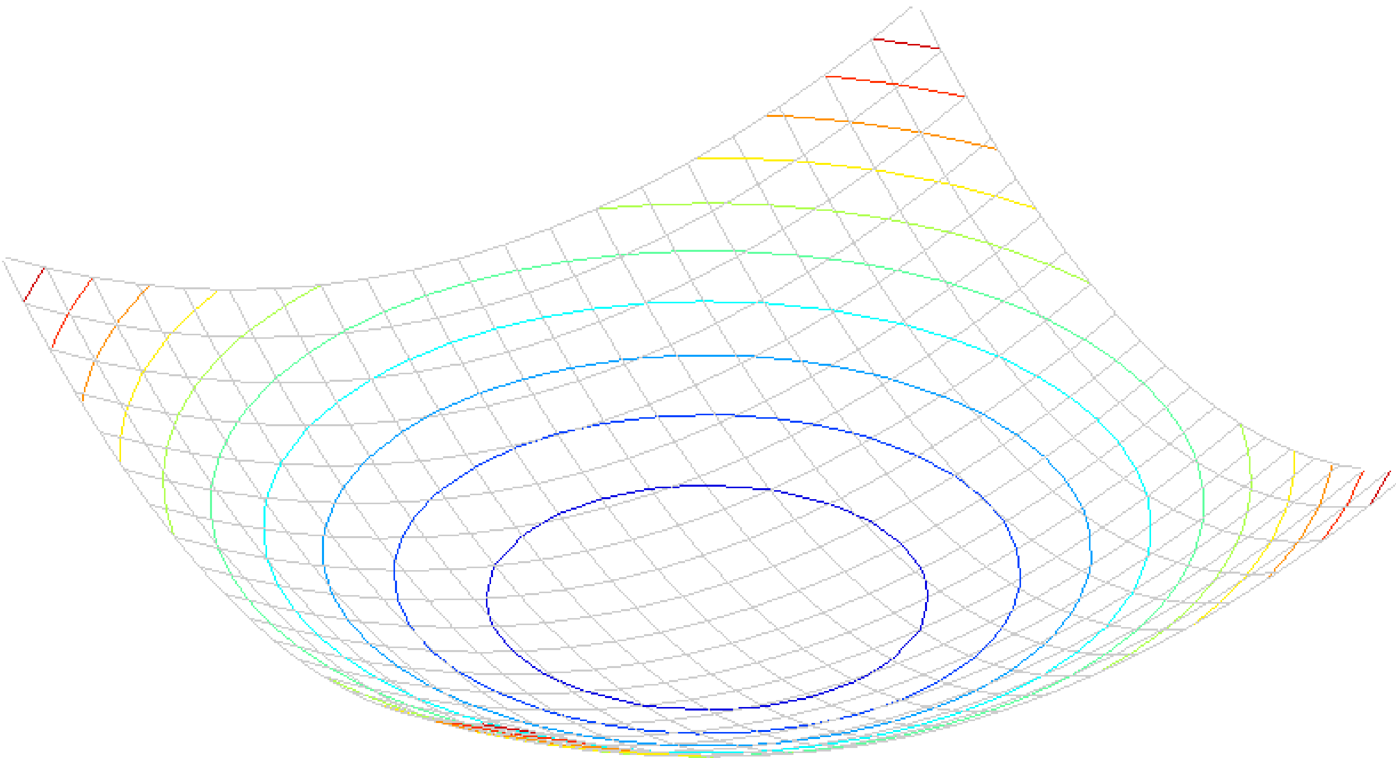
$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix}$$



Interpreting the second moment matrix

Consider a horizontal "slice" of *E*(*u*, *v*): $[u \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.



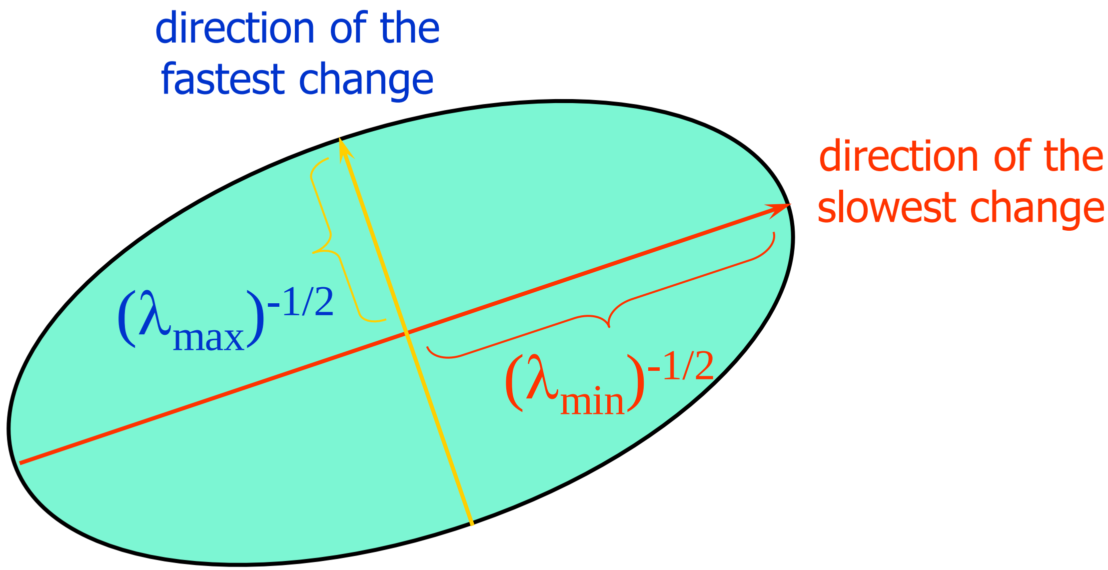
Interpreting the second moment matrix

Consider a horizontal "slice" of *E*(*u*, *v*): $[u\ v]\ M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by *R*



Interpreting the second moment matrix

CENTER FOR MACHINE PERCEPTION

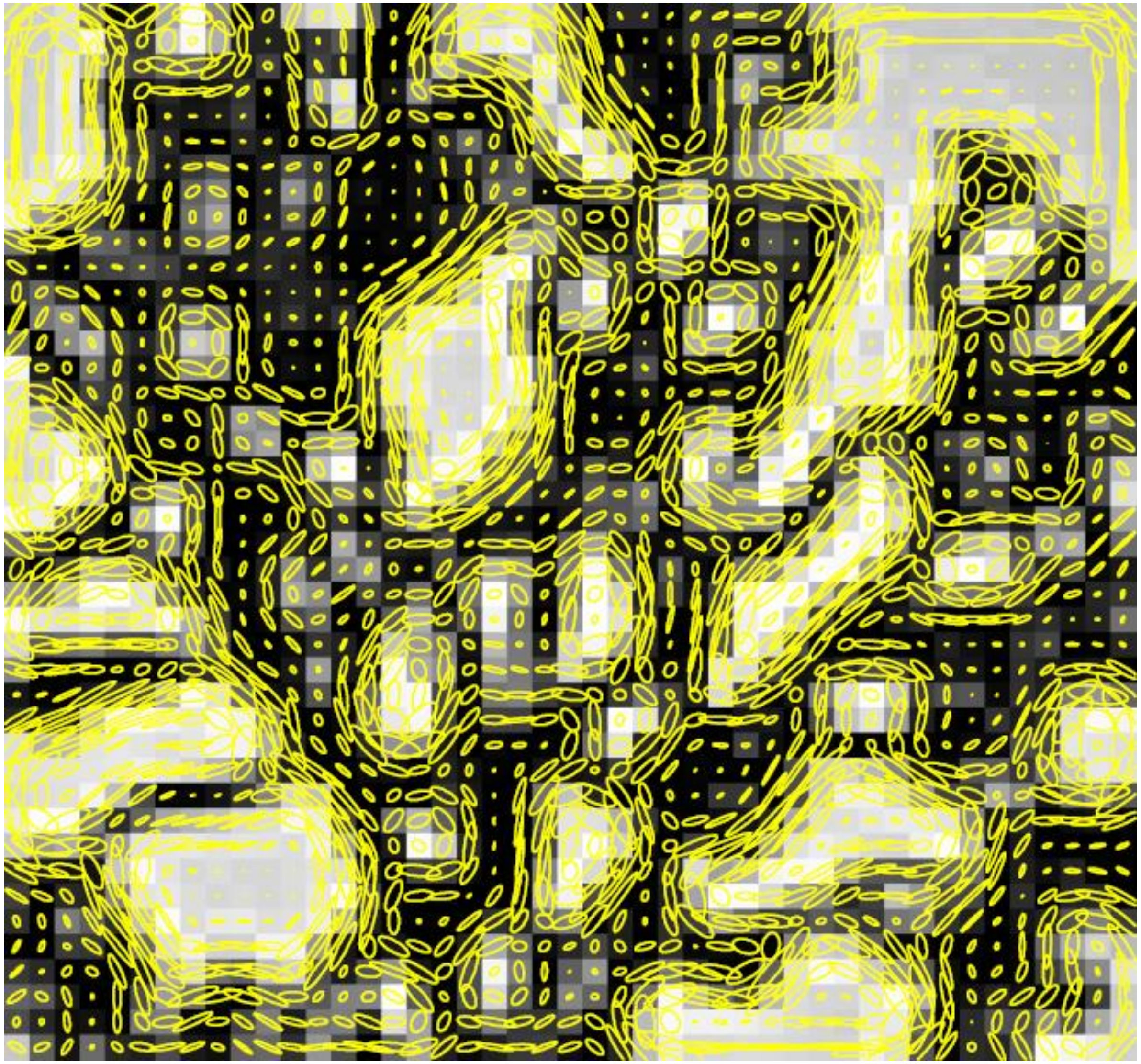
Consider the axis-aligned case (gradients are either horizontal or vertical)

$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

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

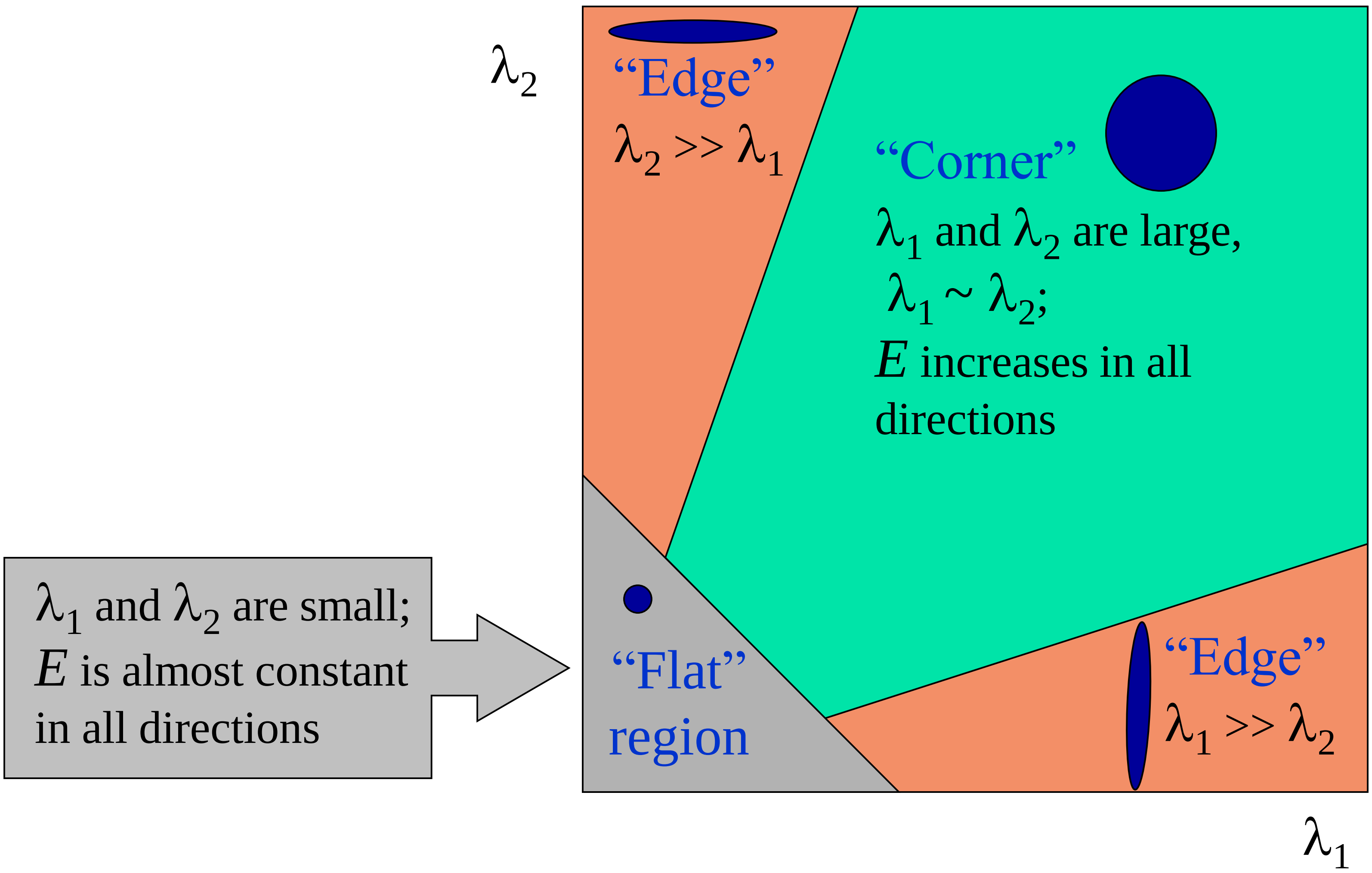
Visualization of second moment matrices

Visualization of second moment matrices



Interpreting the eigenvalues

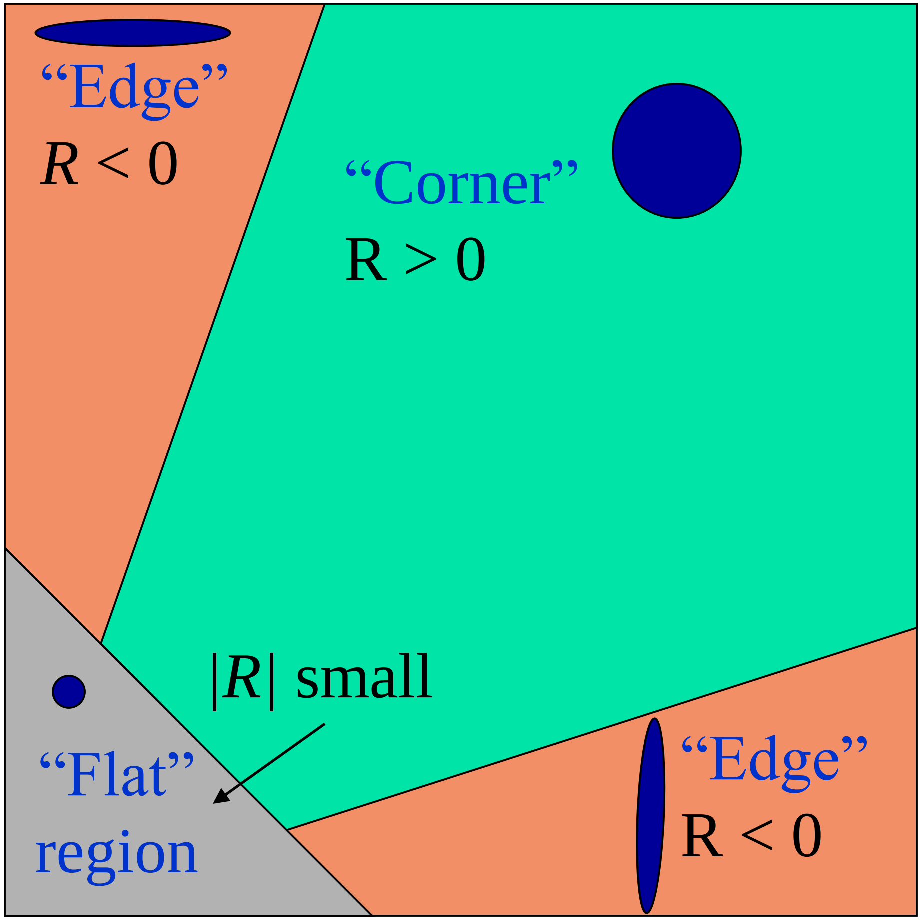
Classification of image points using eigenvalues of *M*:



Corner response function

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

α: constant (0.04 to 0.06)



The Harris corner detector

CENTER FOR MACHINE PERCEPTION

1. Compute partial derivatives at each pixel
2. Compute second moment matrix *M* in a Gaussian window around each pixel:

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

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

The Harris corner detector

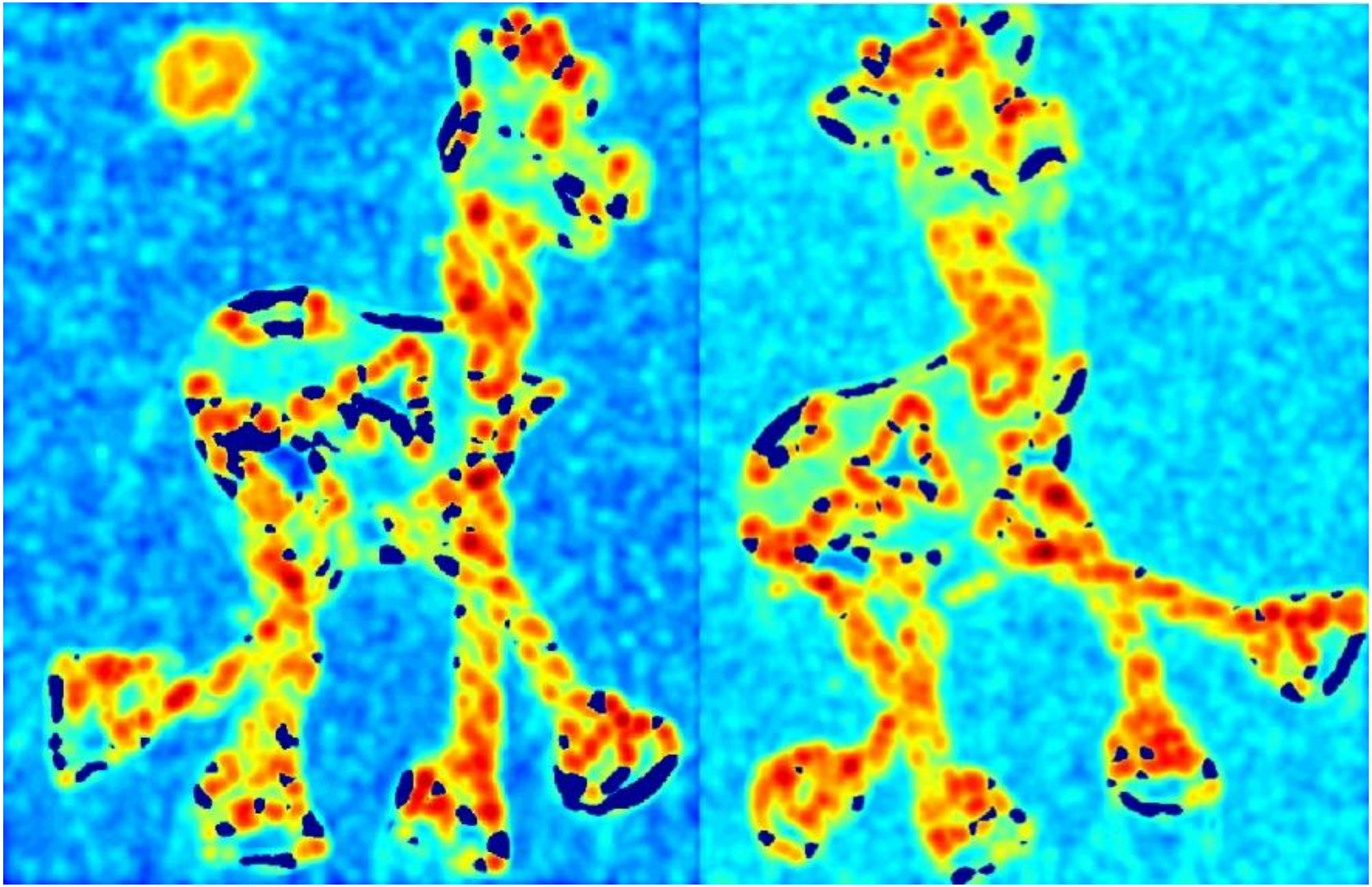
1. Compute partial derivatives at each pixel
2. Compute second moment matrix *M* in a Gaussian window around each pixel
3. Compute corner response function *R*

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

Harris Detector: Steps

Harris Detector: Steps

Compute corner response *R*



The Harris corner detector

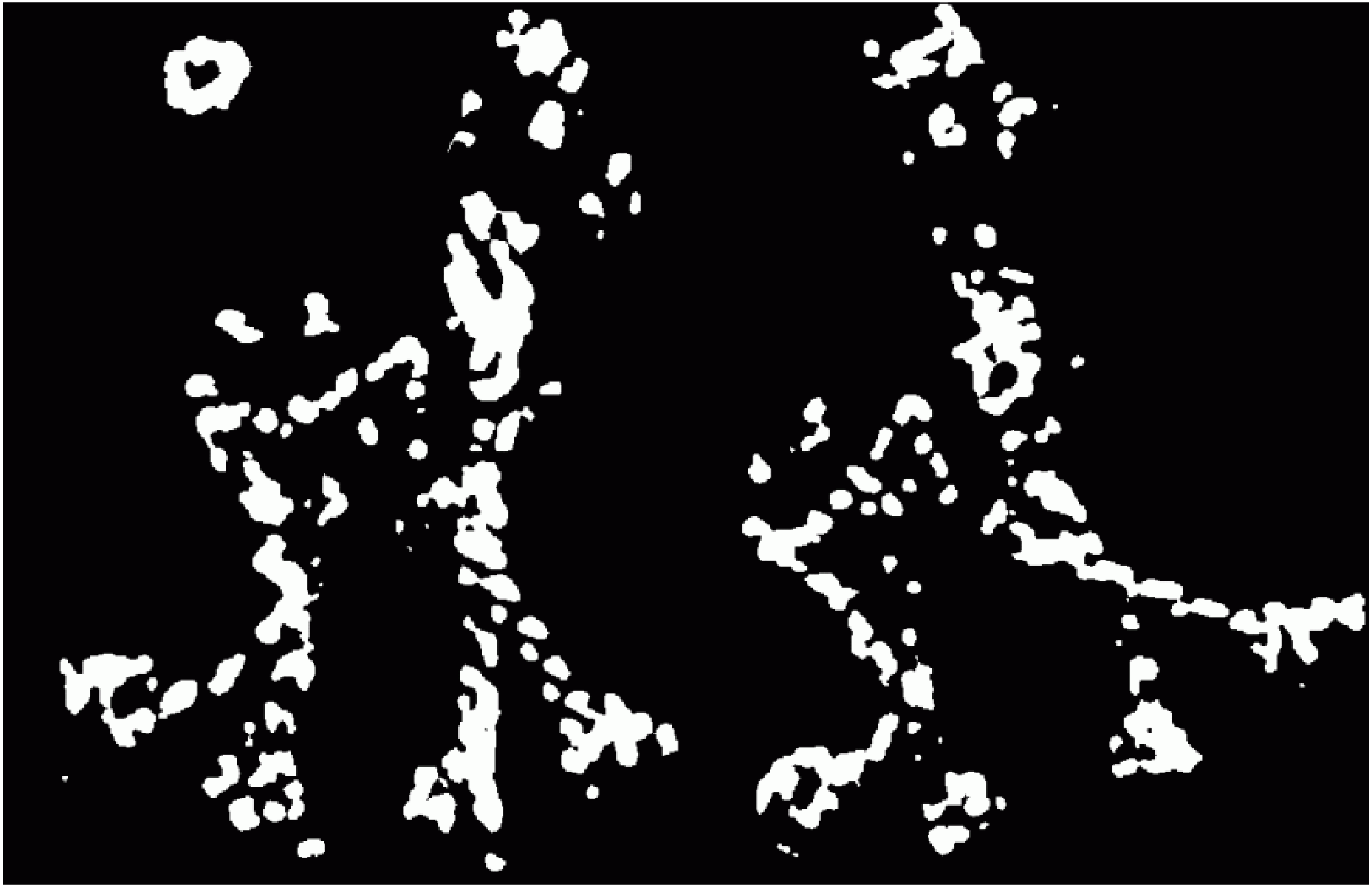
1. Compute partial derivatives at each pixel
2. Compute second moment matrix M in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maxima of response function (nonmaximum suppression)

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

Harris Detector: Steps

CENTER FOR MACHINE PERCEPTION

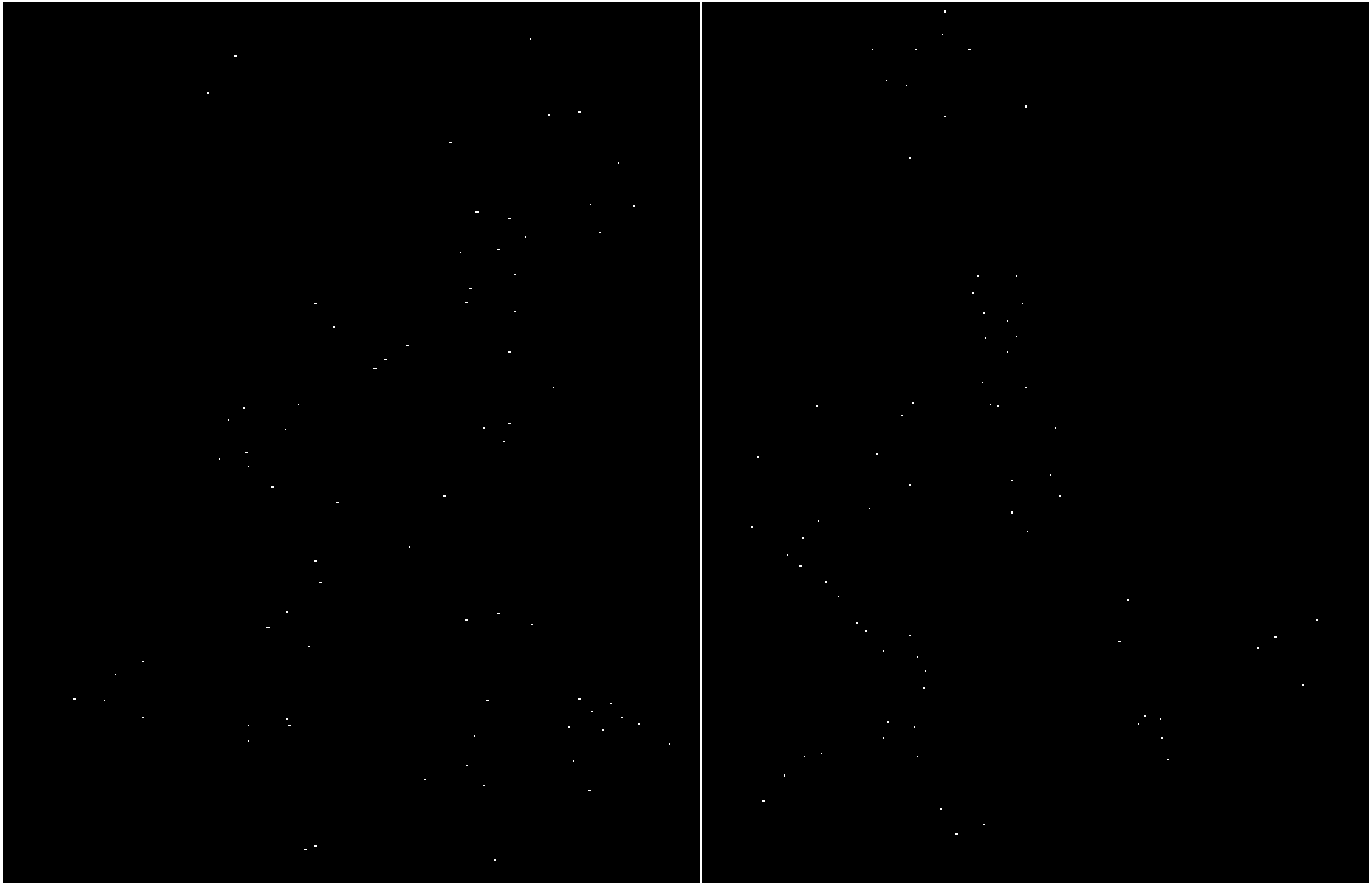
Find points with large corner response: $R >$ threshold



Harris Detector: Steps

CENTER FOR MACHINE PERCEPTION

Take only the points of local maxima of *R*

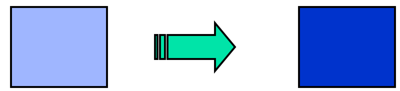


Harris Detector: Steps

Robustness of corner features

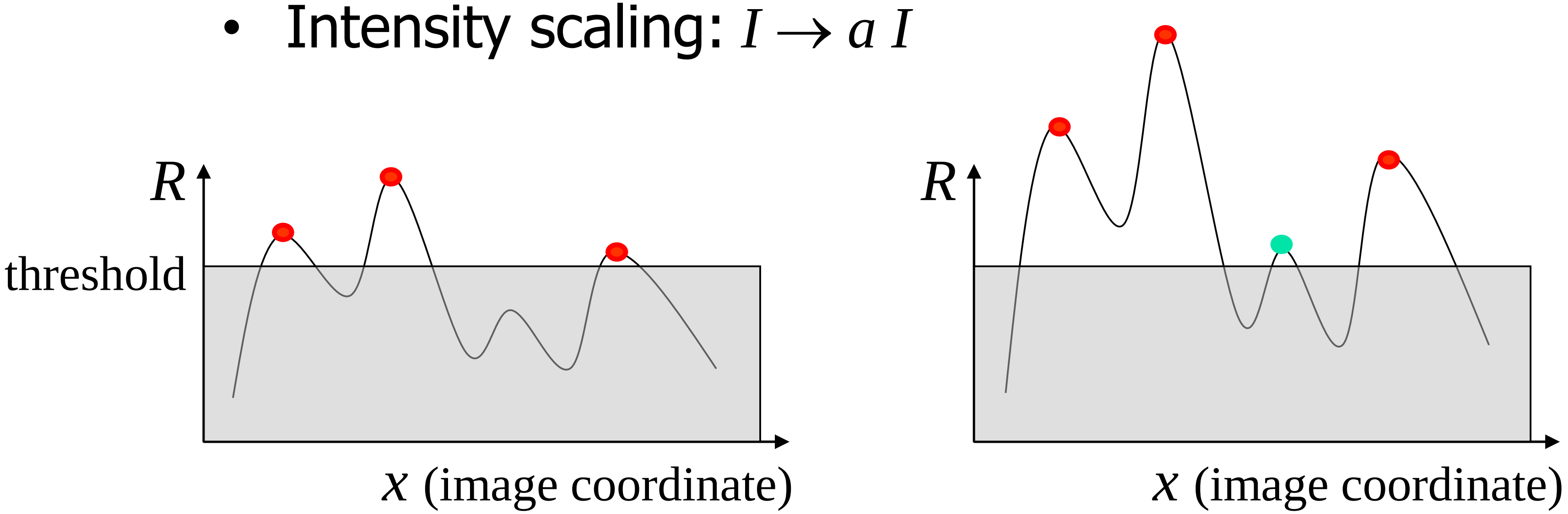
- What happens to corner features when the image undergoes geometric or photometric transformations?

Affine intensity change



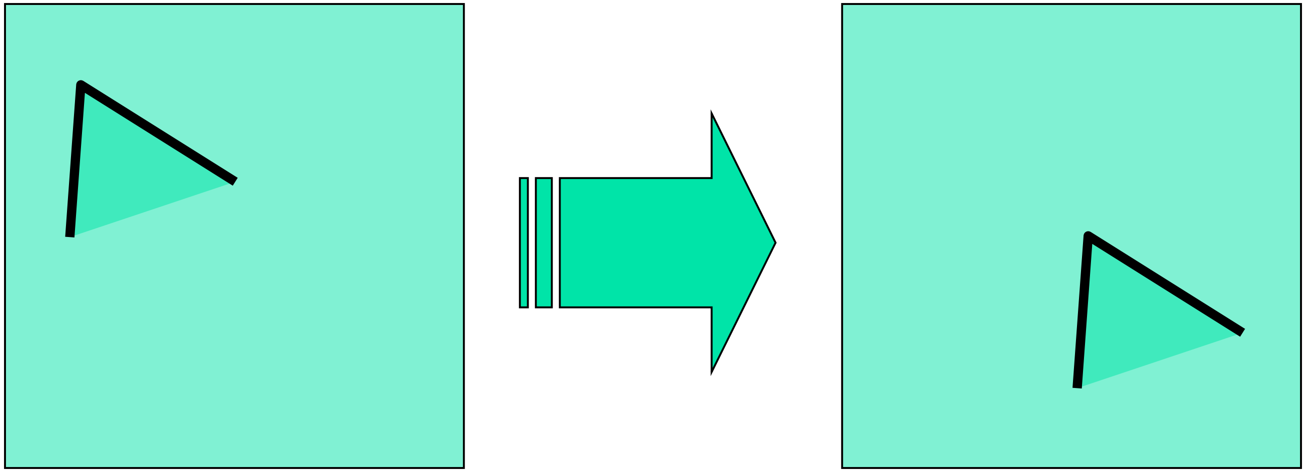
$$I \to a\, I + b$$

- Only derivatives are used => invariance to intensity shift $I \to I + b$
- Intensity scaling: $I \to a\, I$



Partially invariant to affine intensity change

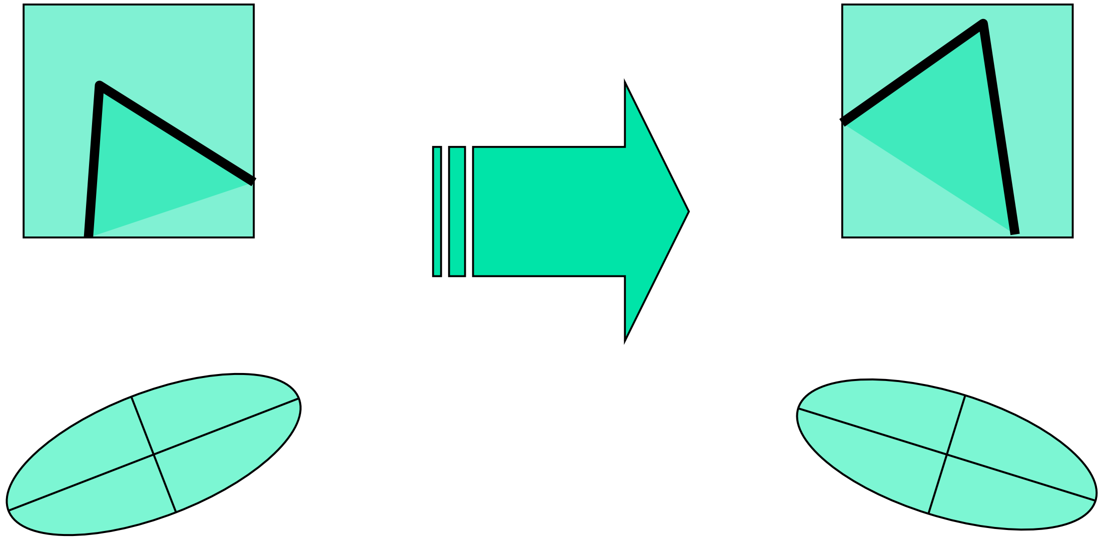
Image translation



- Derivatives and window function are shift-invariant

Corner location is *covariant* w.r.t. translation

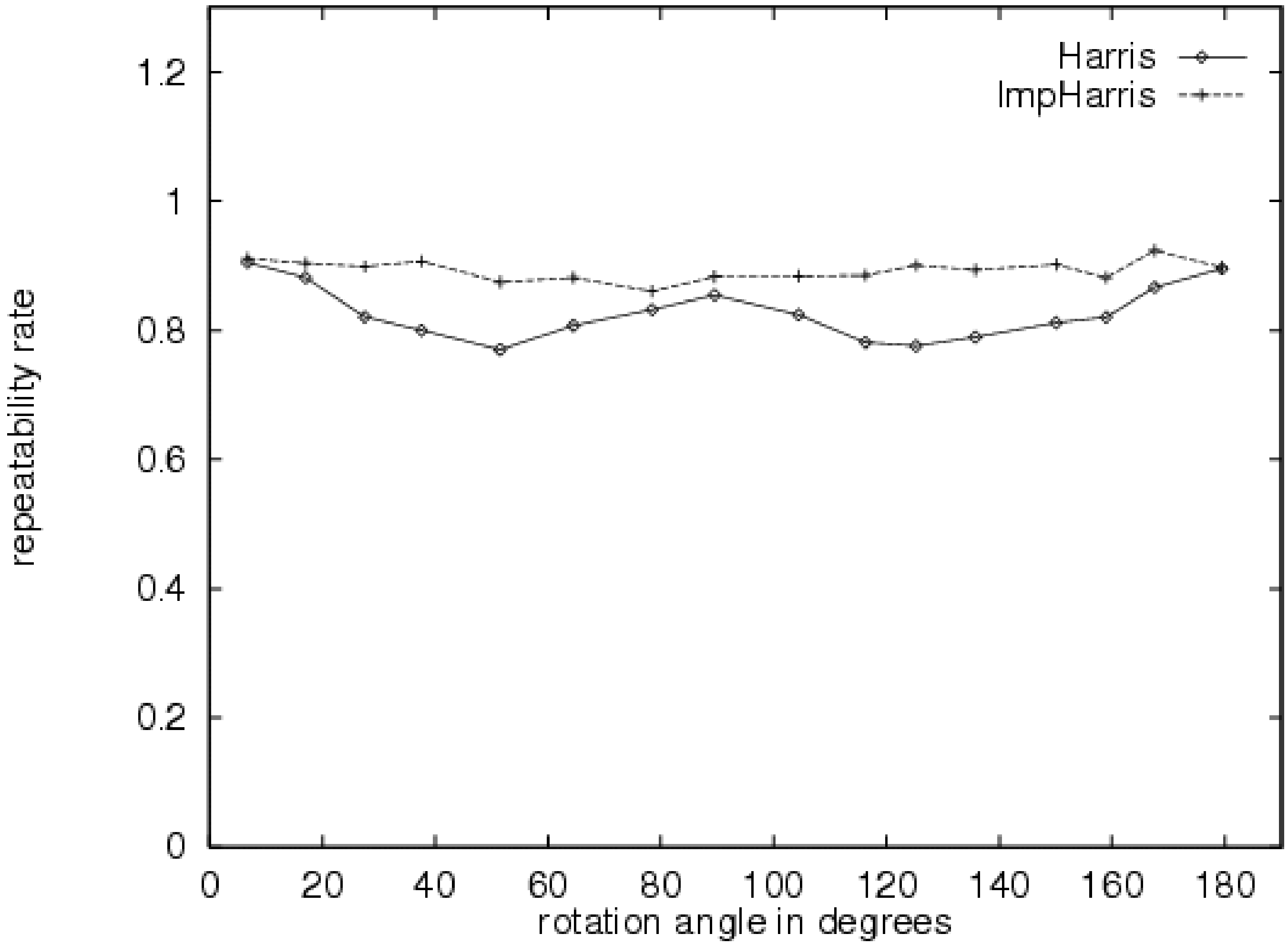
Image rotation



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

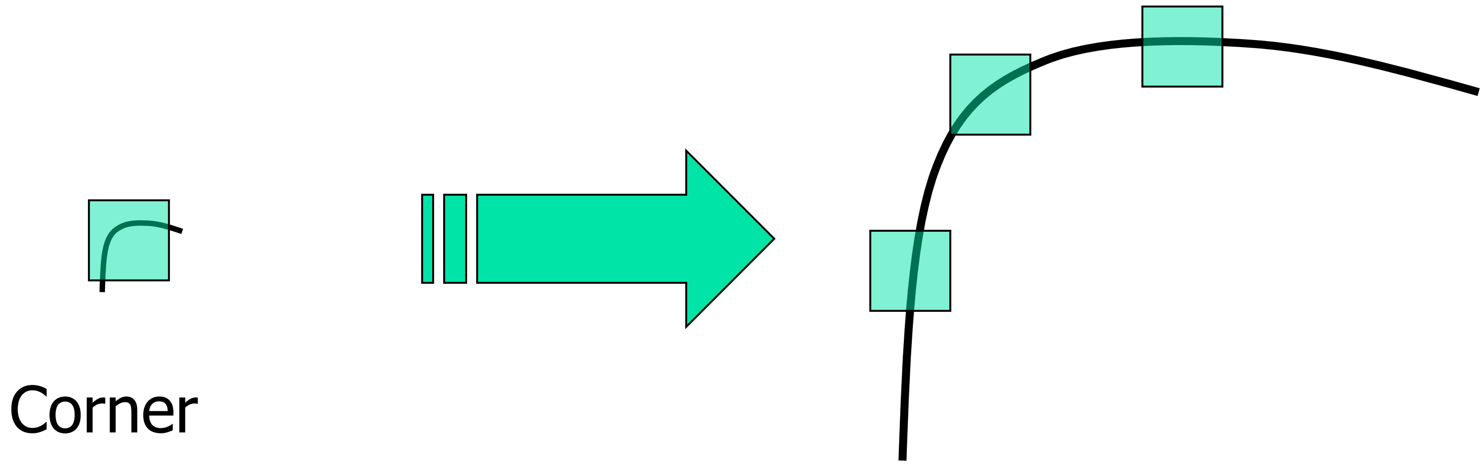
Corner location is covariant w.r.t. rotation

Rotation Invariance of Harris Detector



C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

Scaling



All points will be classified as edges

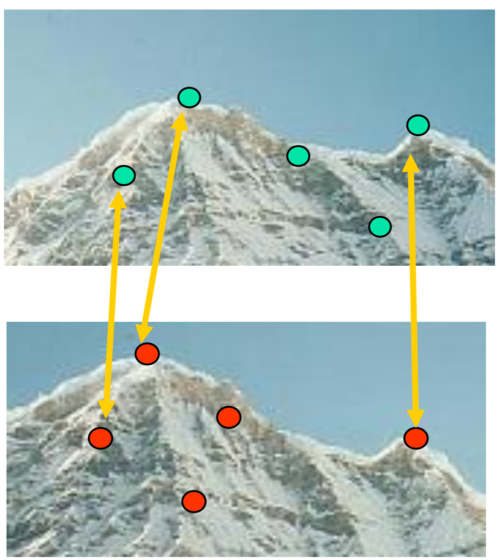
Corner location is not covariant to scaling!

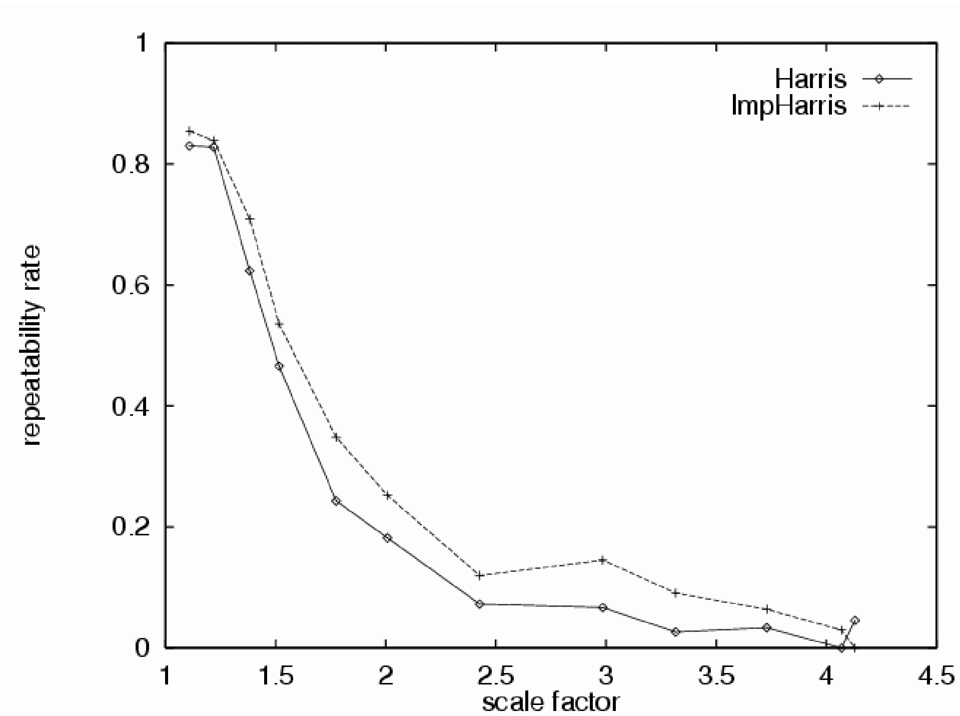
Harris Detector: Scale Change

- Quality of Harris detector for different scale changes

Repeatability rate:

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



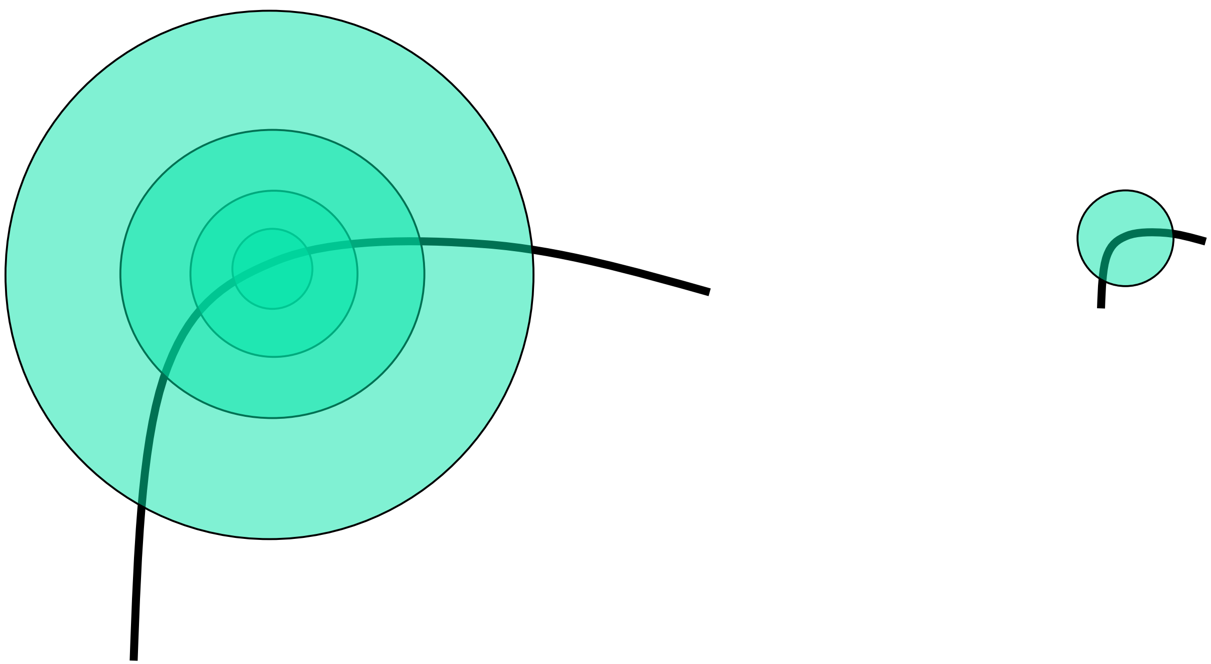


C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

Scale Invariant Detection

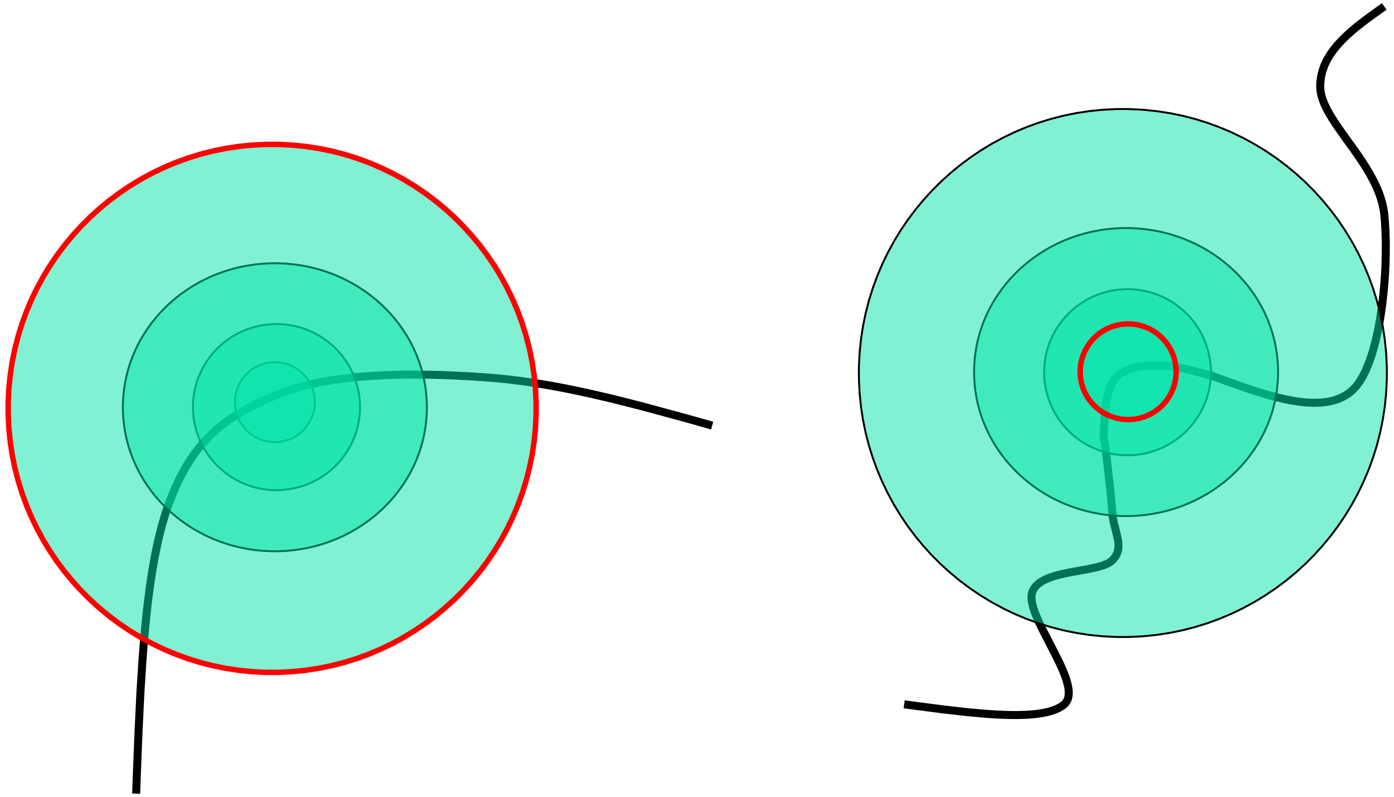
CENTER FOR MACHINE PERCEPTION

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



Scale Invariant Detection

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



What is scale space



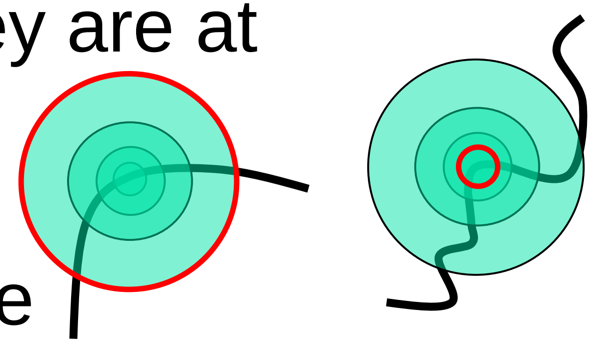


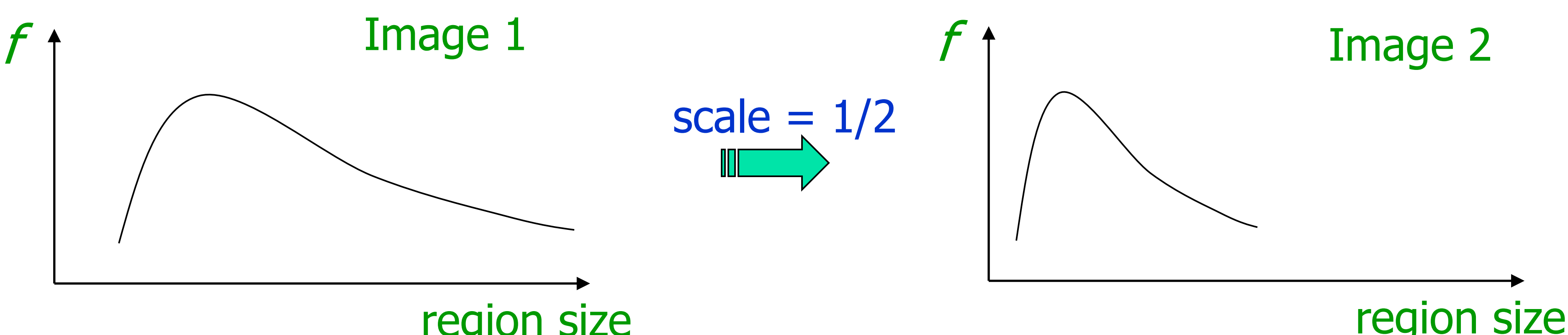
- Progression of Gaussian blurs
- Intuition: Simulate a point spread function applied to larger parts of the scene
- Theory: Scale space axioms

Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is “scale covariant” (the same for corresponding regions, even if they are at different scales)

– For a point in one image, we can consider it as a function of region size (circle radius)

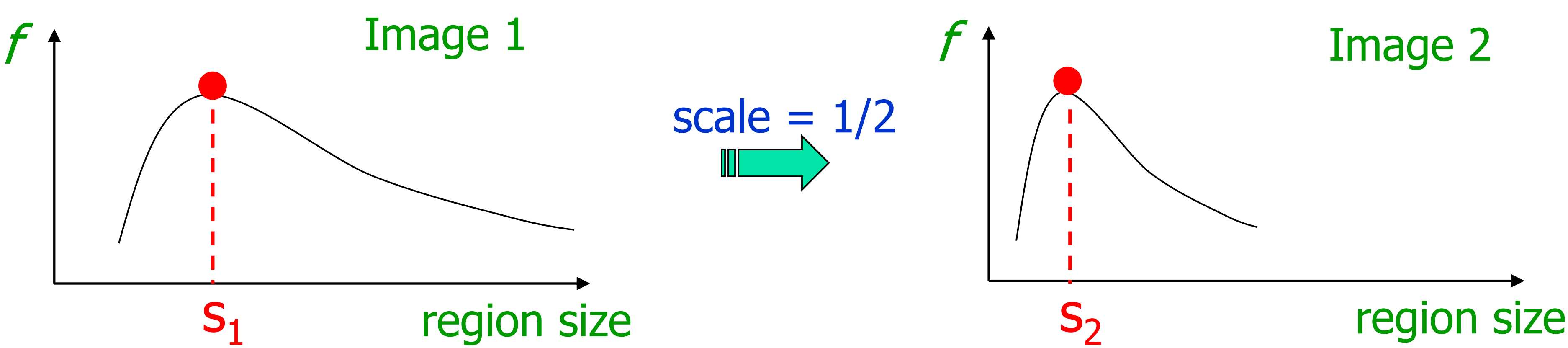




Scale Invariant Detection

- Common approach:
  - Take a local maximum of some 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!



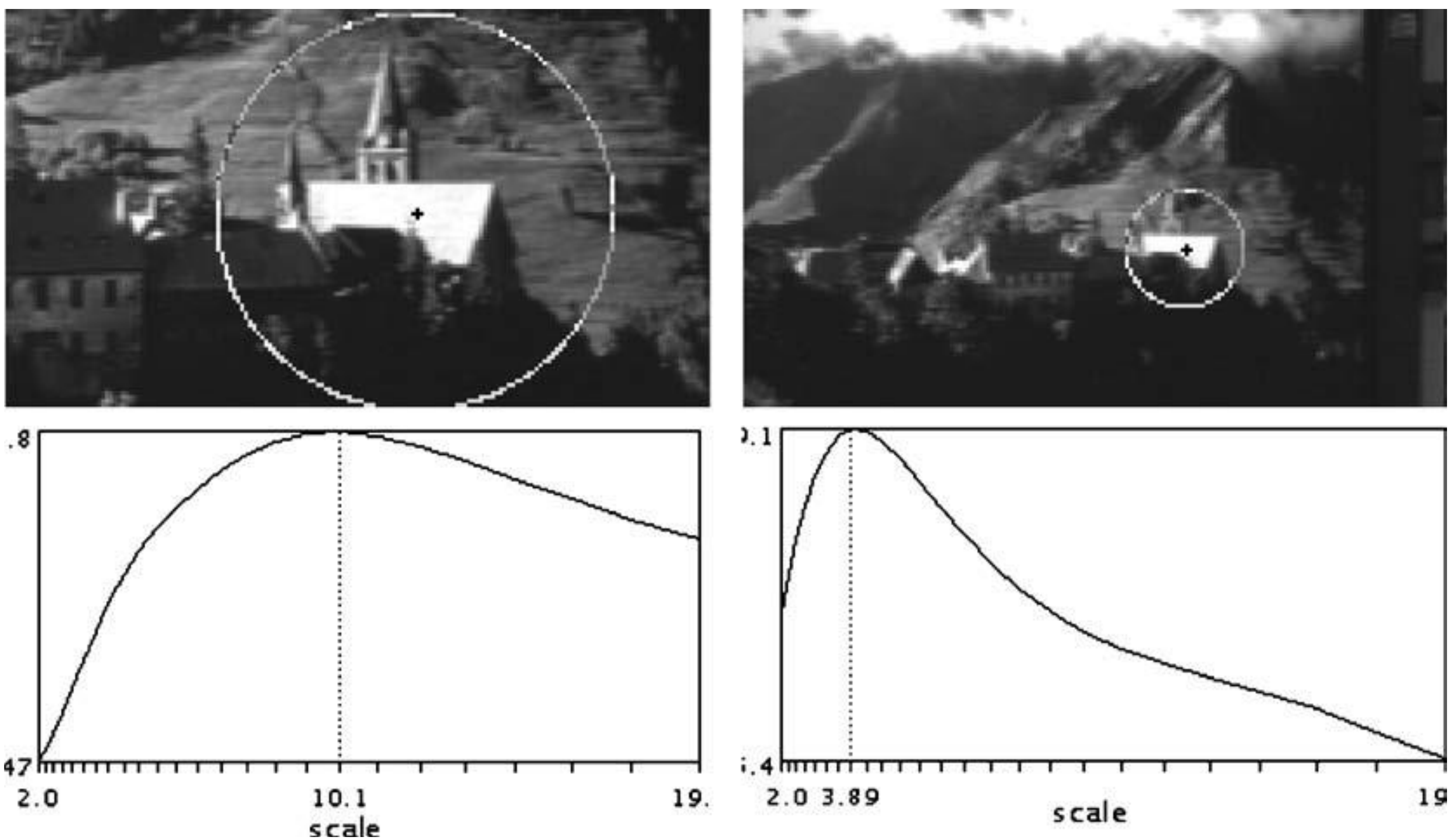
Scale Invariant Detection

- 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)

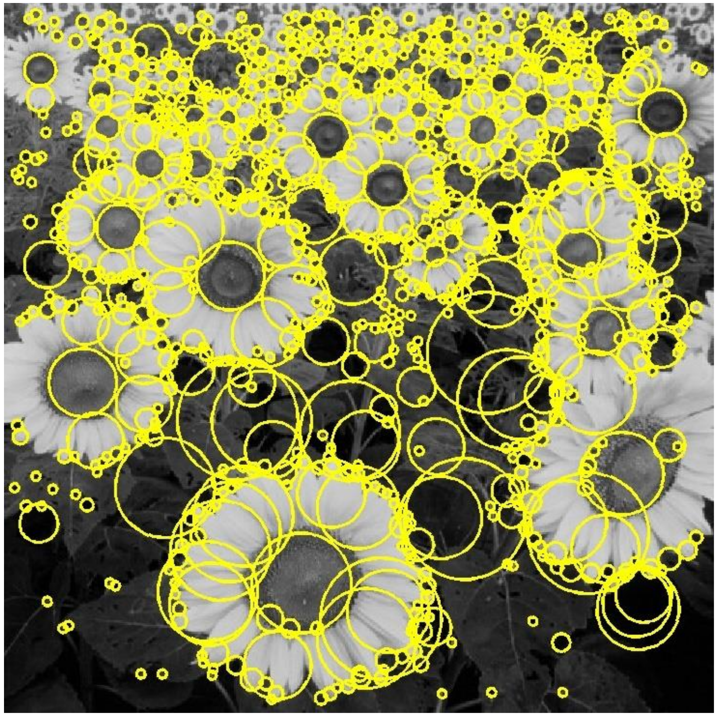
Keypoint detection with scale selection

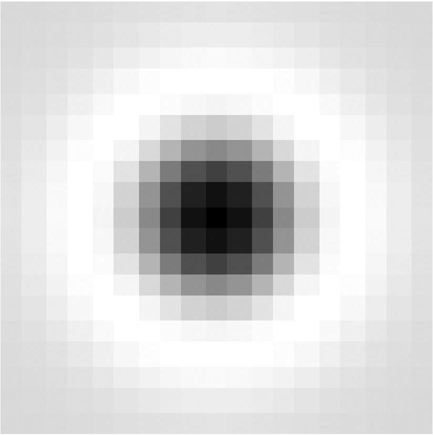
- We want to extract keypoints with characteristic scale that is *covariant* with the image transformation



Basic idea

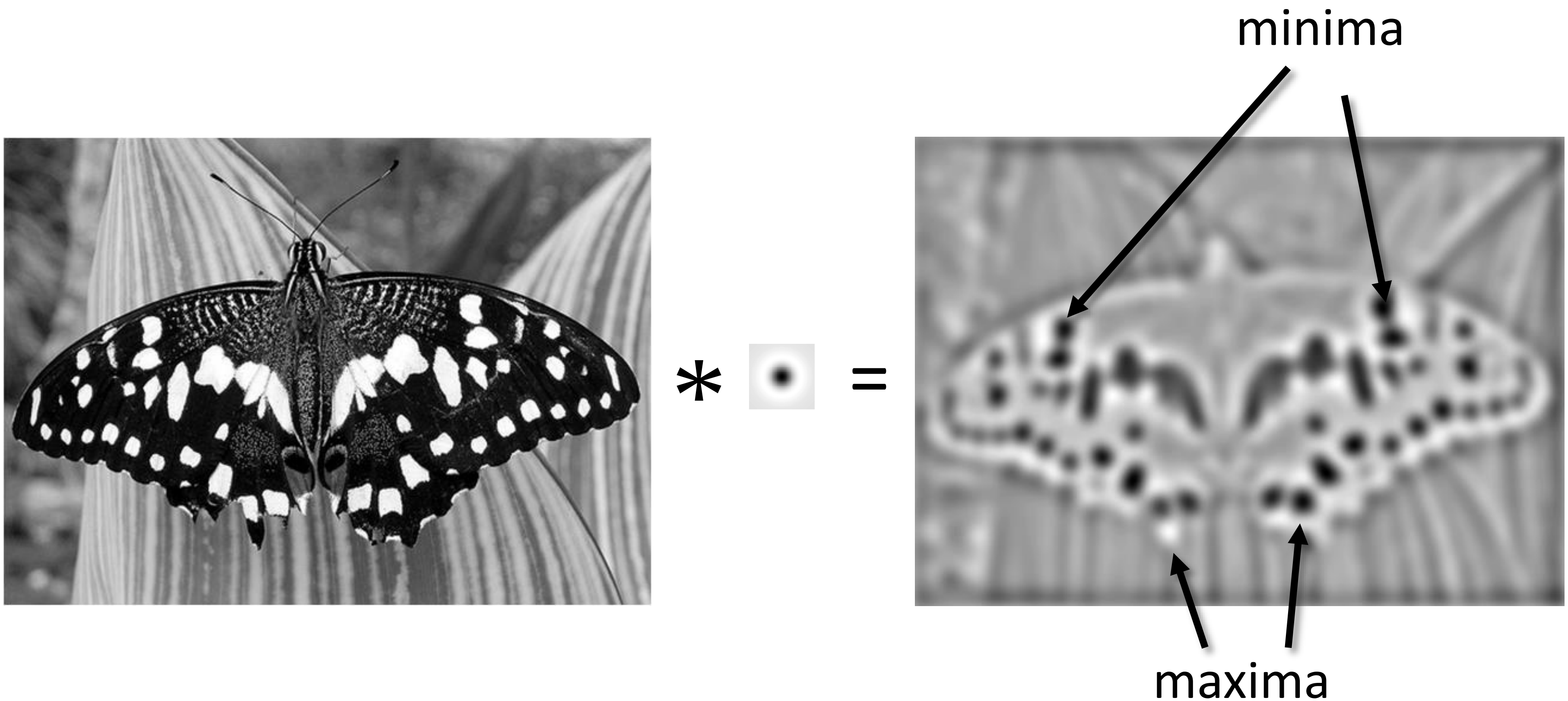
- Convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting *scale space*





T. Lindeberg. Feature detection with automatic scale selection. *IJCV* 30(2), pp 77-116, 1998.

Blob detection

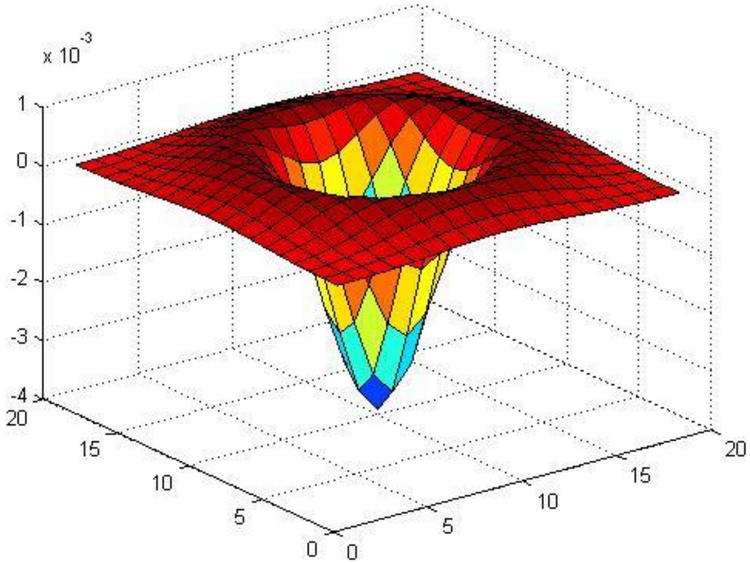


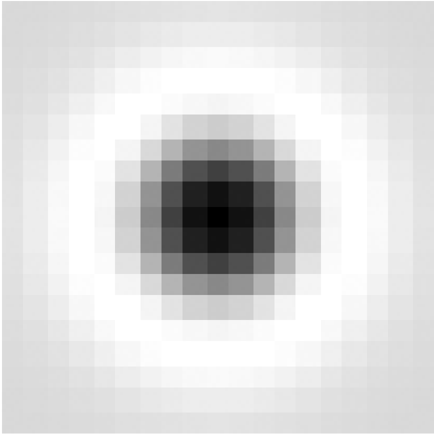
- Find maxima *and minima* of blob filter response in space *and scale*

Source: N. Snavely

Blob filter

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D





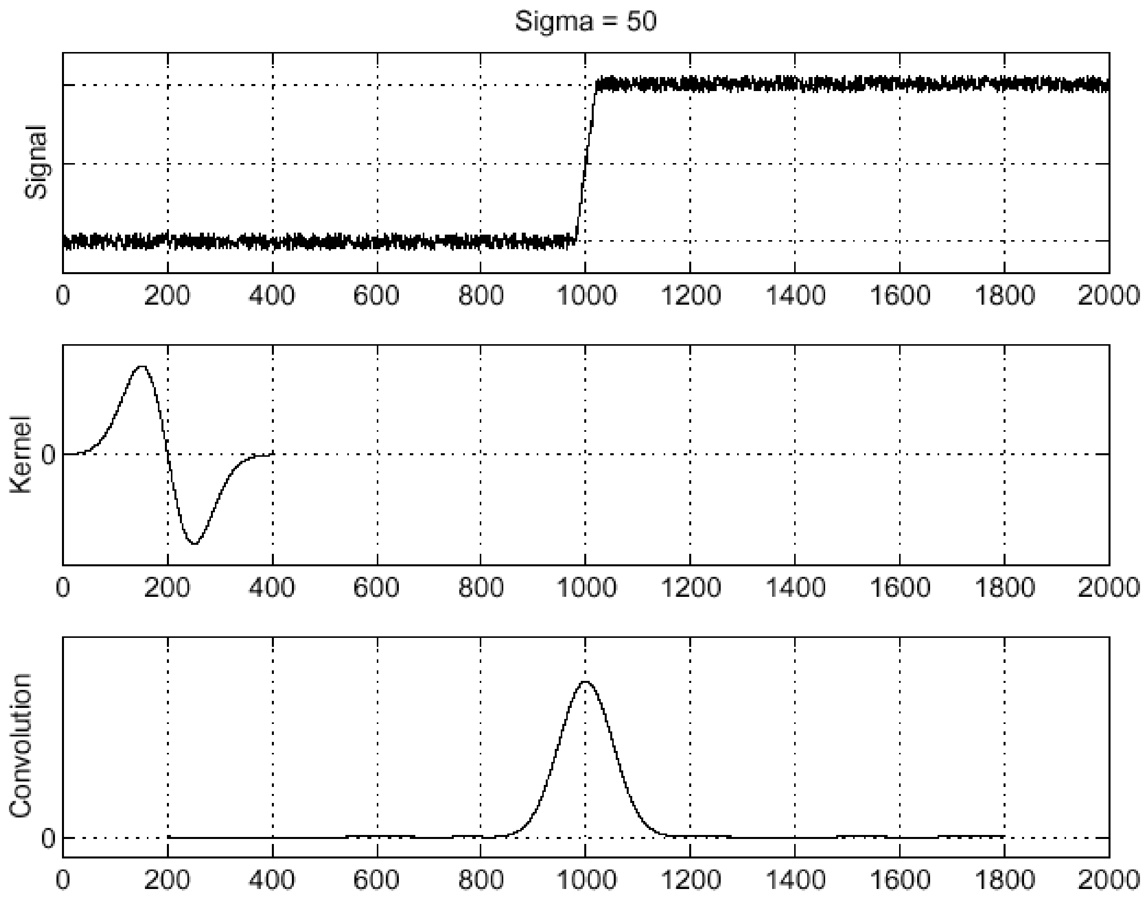
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Recall: Edge detection

f

$\frac{d}{dx} g$

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



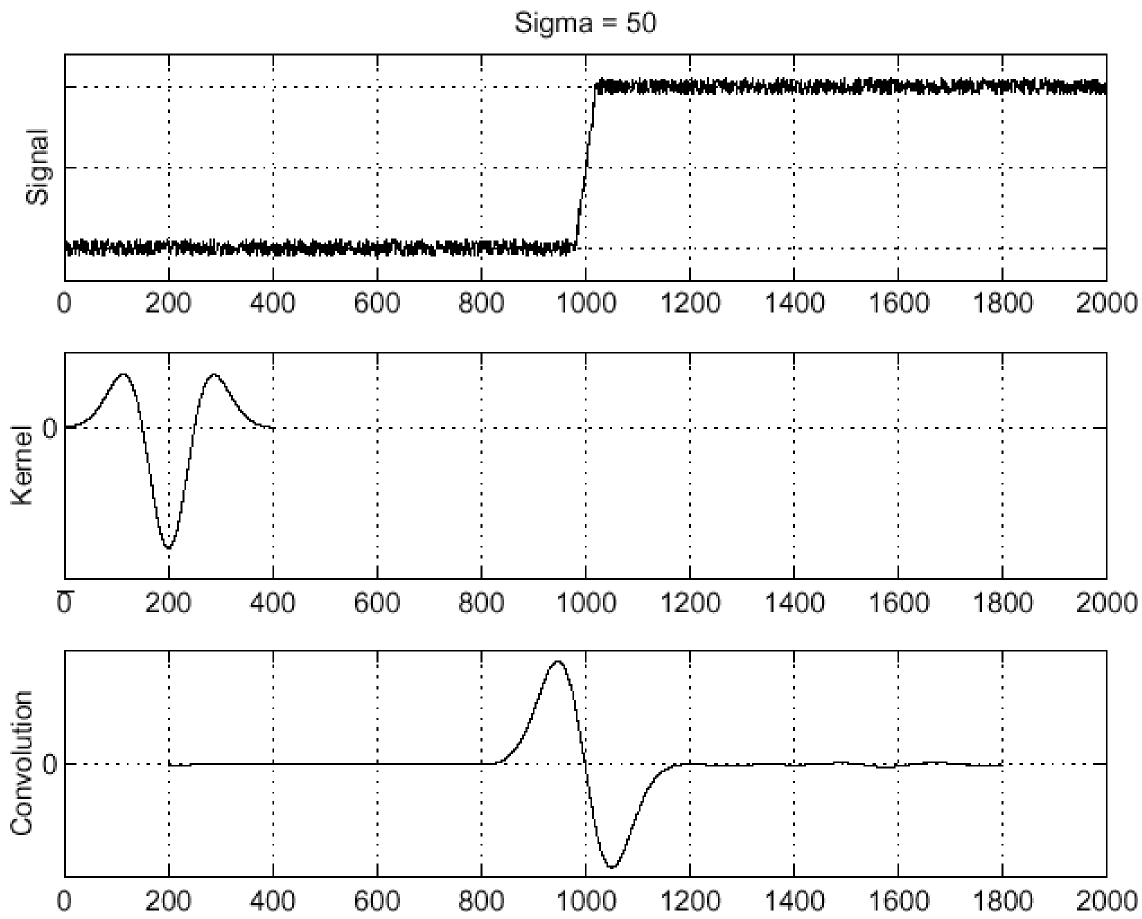
Edge

Derivative of Gaussian

Edge = maximum of derivative

Source: S. Seitz

Edge detection, Take 2



f — Edge

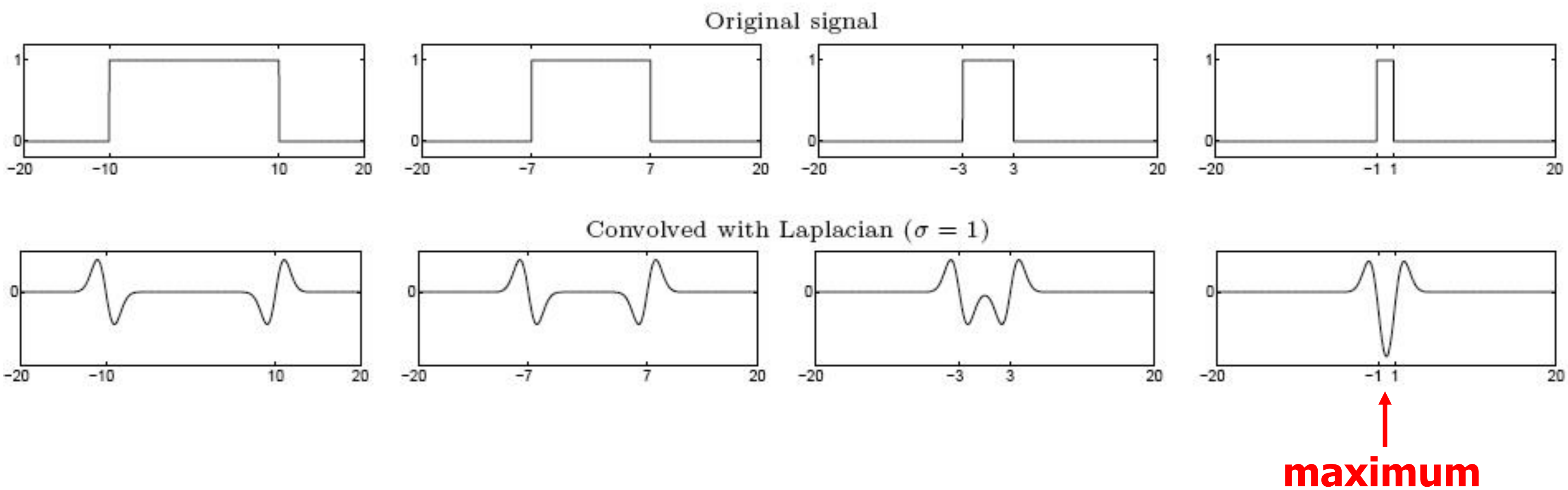
$\frac{d^2}{dx^2} g$ — Second derivative of Gaussian (Laplacian)

$f * \frac{d^2}{dx^2} g$ — Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

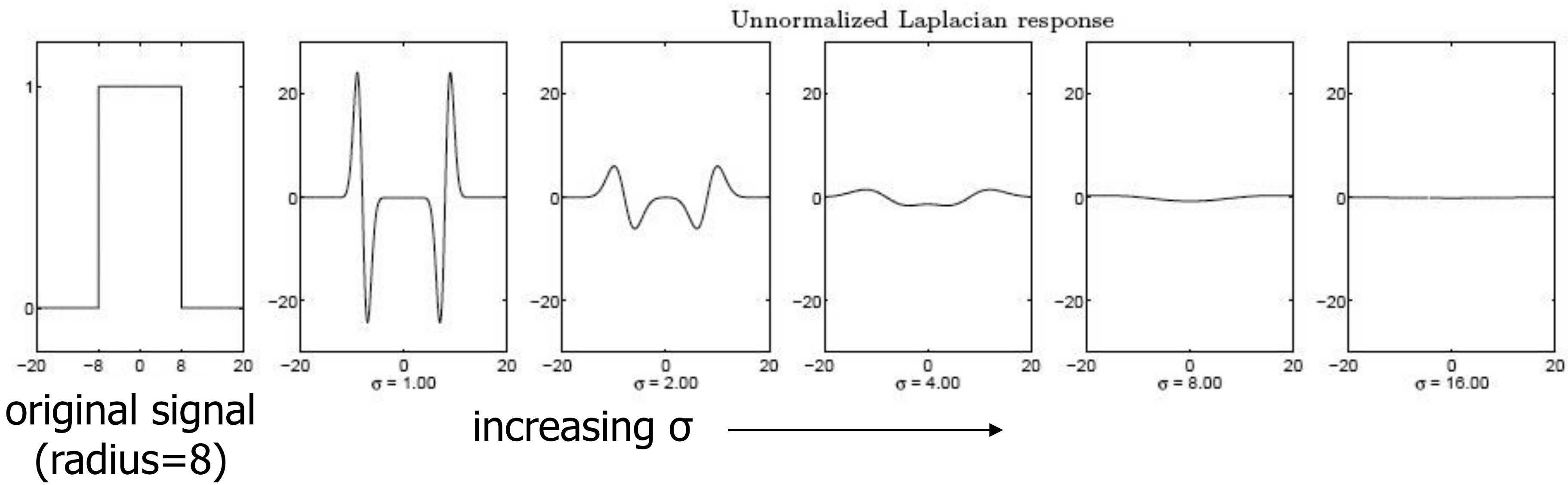
- Edge = ripple
- Blob = superposition of two ripples



Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

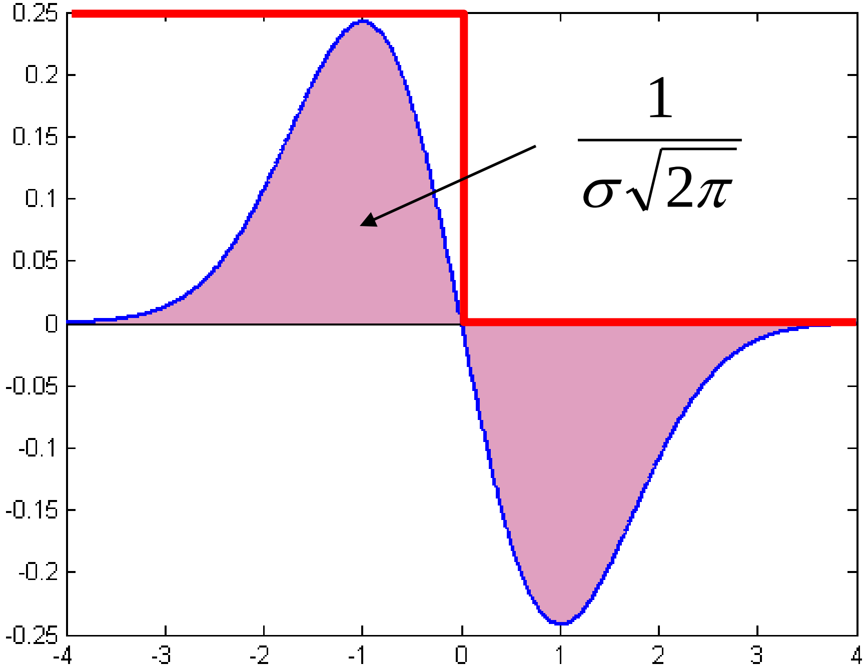
Scale selection

- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response
- However, Laplacian response decays as scale increases:



Scale normalization

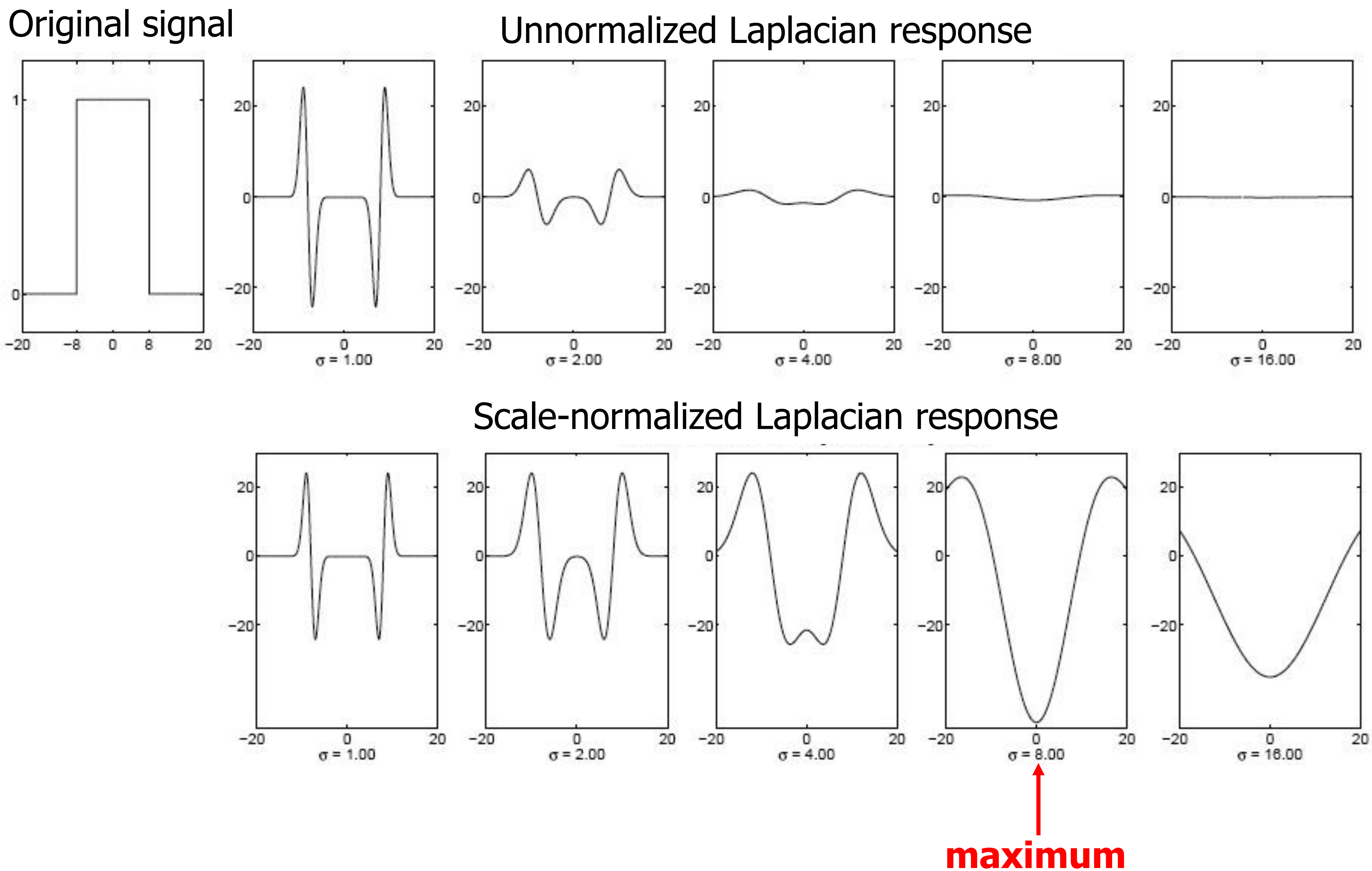
- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases



Scale normalization

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

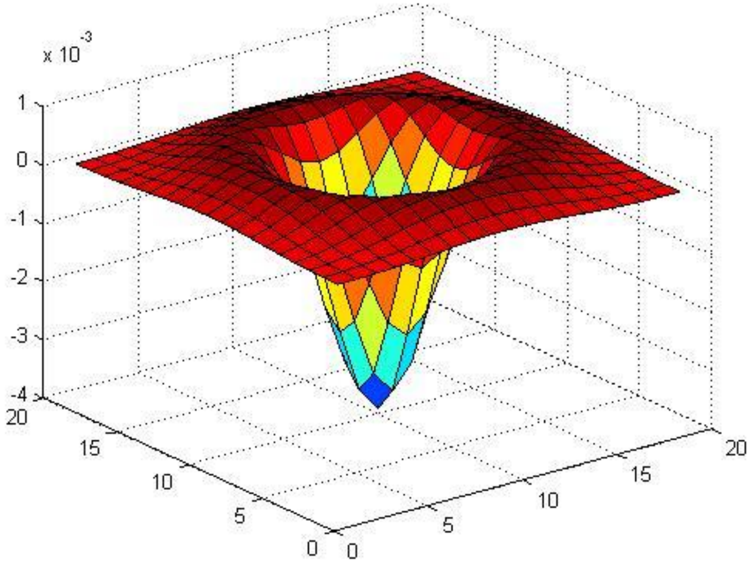
Effect of scale normalization

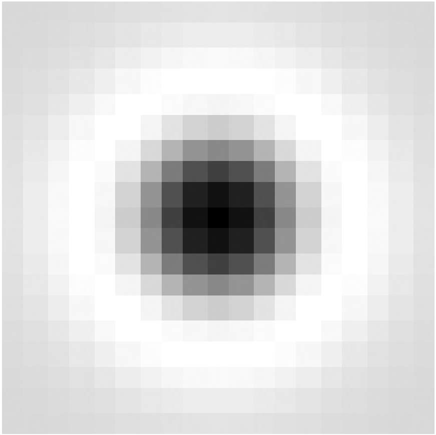


Blob detection in 2D

- *Scale-normalized* Laplacian of Gaussian:

$$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

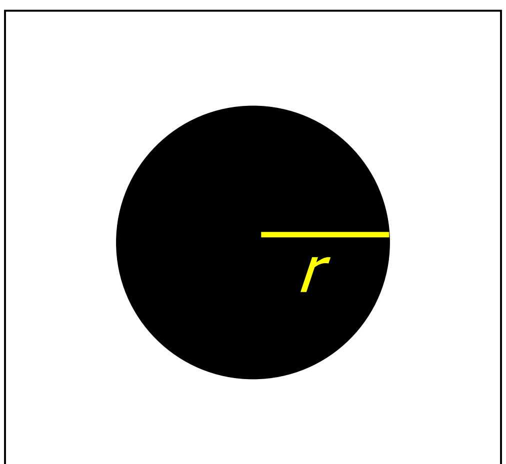




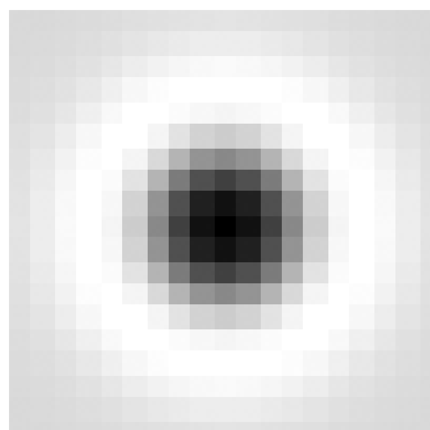
Blob detection in 2D

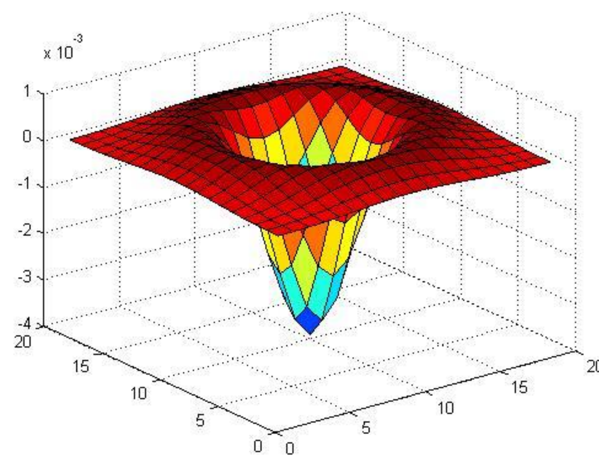
CENTER FOR MACHINE PERCEPTION

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- Laplacian measures curvature, think of one dimension
- Gives how much the pixels differ from it’s average value



image





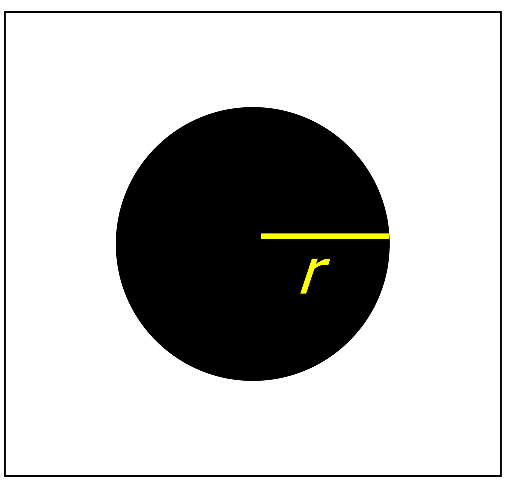
Laplacian

Blob detection in 2D

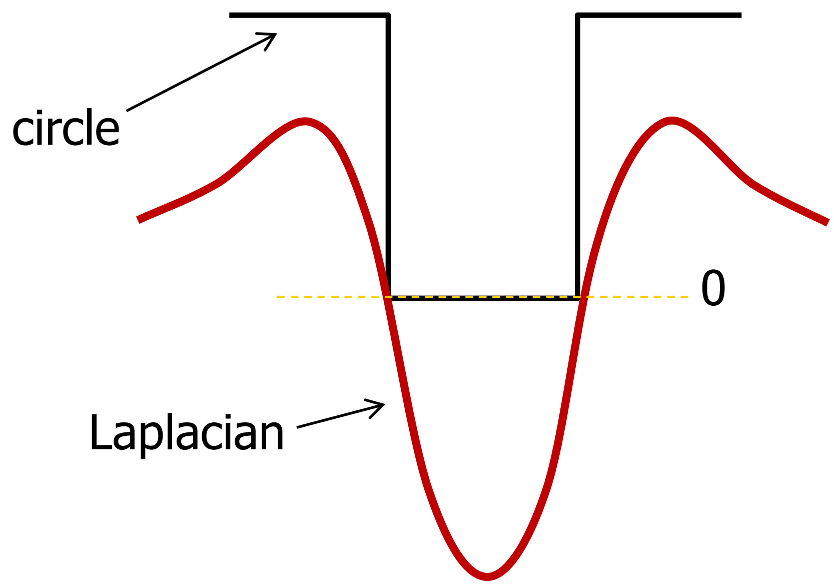
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$

- Therefore, the maximum response occurs at $\sigma = r/\sqrt{2}$.



image



Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales

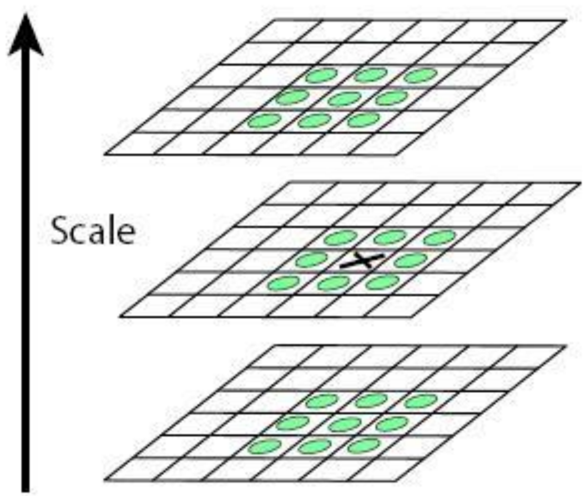
Scale-space blob detector: Example

Scale-space blob detector: Example

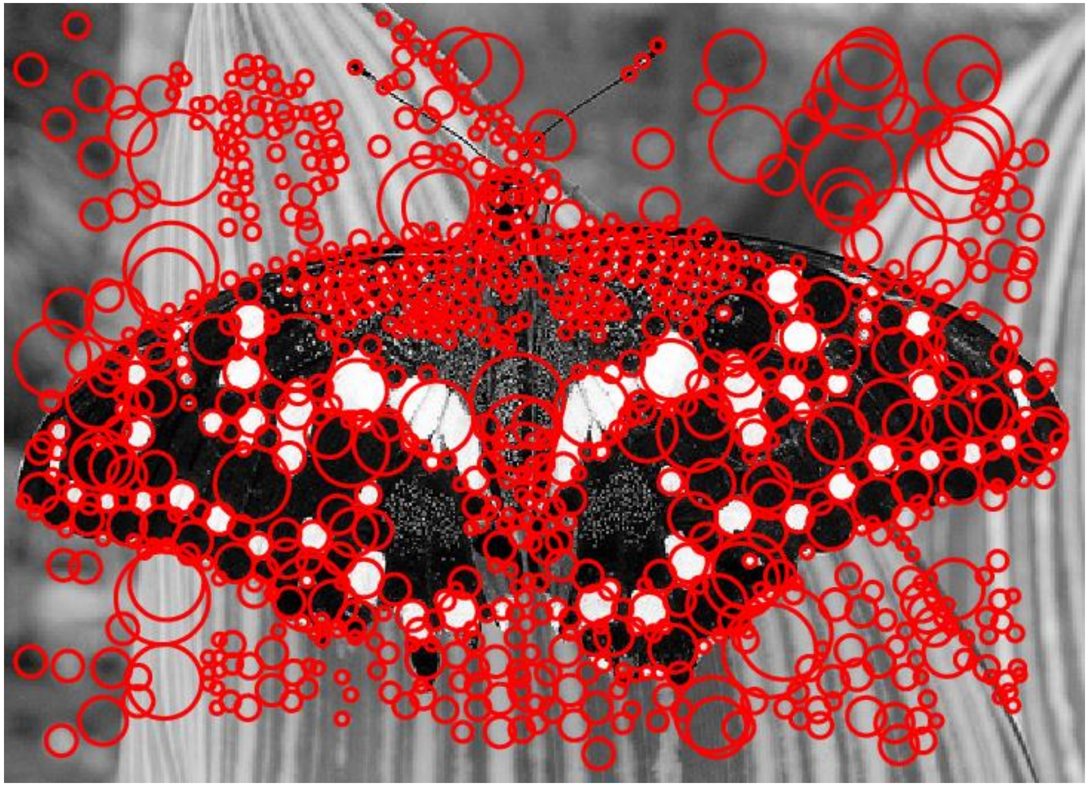
sigma = 11.9912

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space

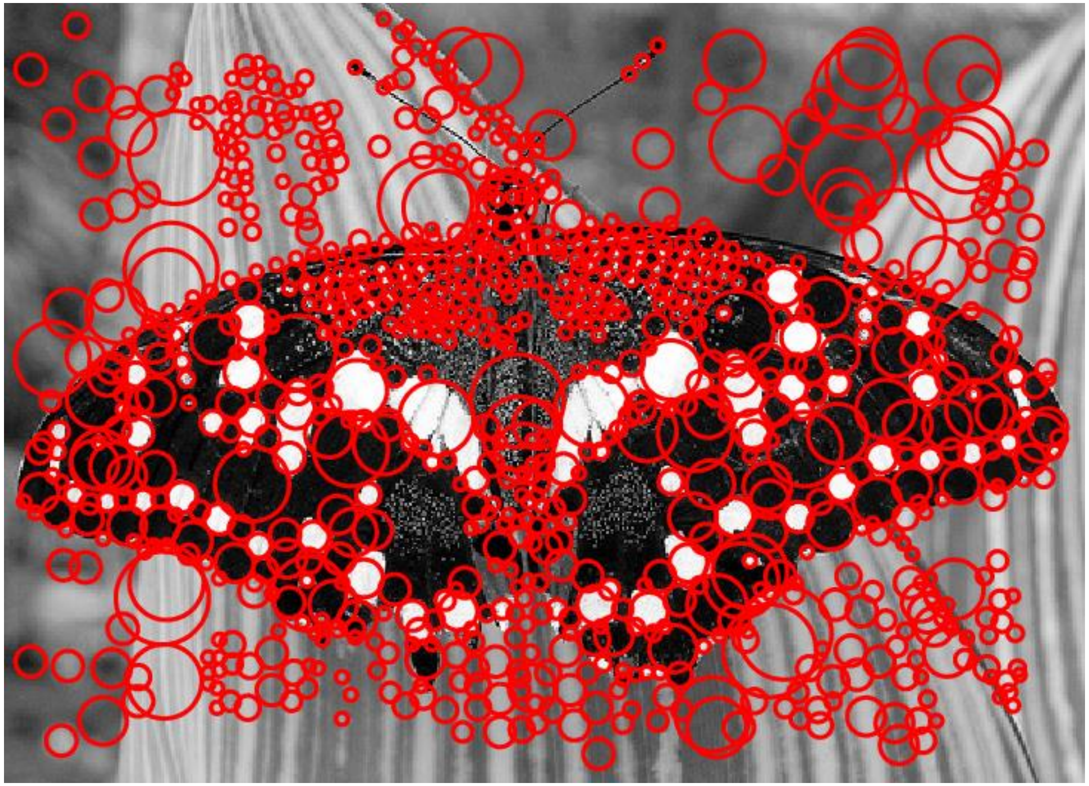


Scale-space blob detector: Example



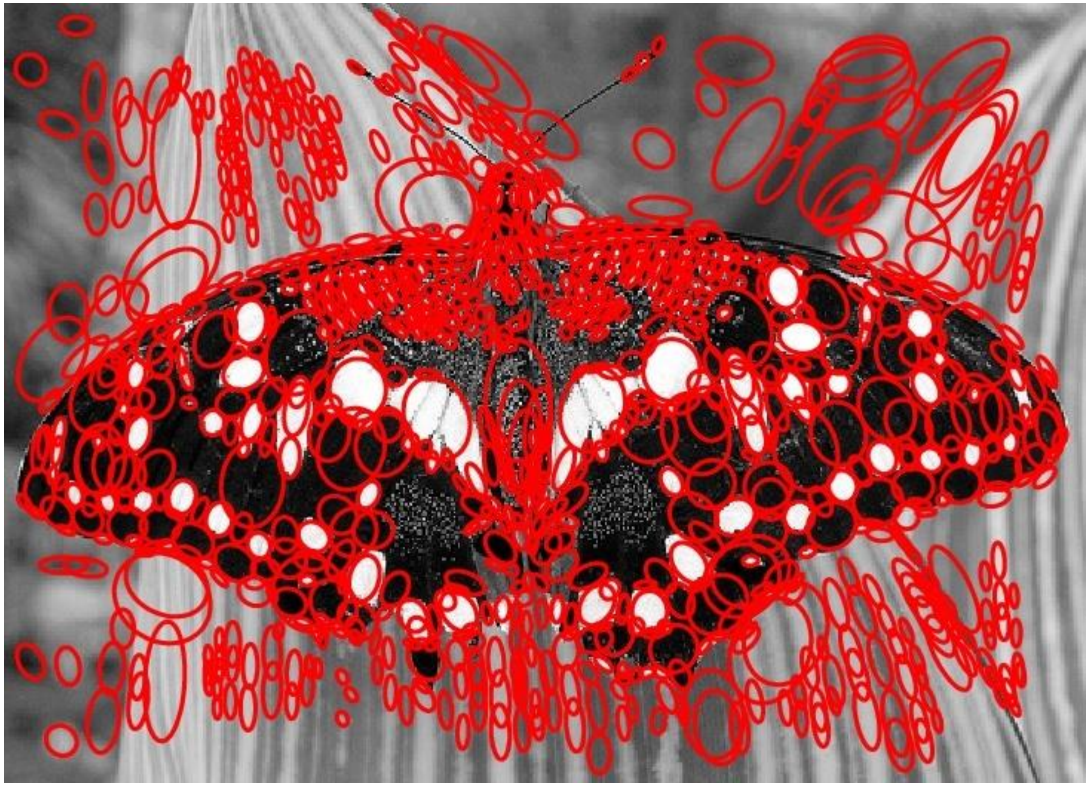
Eliminating edge responses

- Laplacian has strong response along edge



Eliminating edge responses

- Laplacian has strong response along edge



- Solution: filter based on Harris response function over neighboroods containing the “blobs”

Efficient implementation

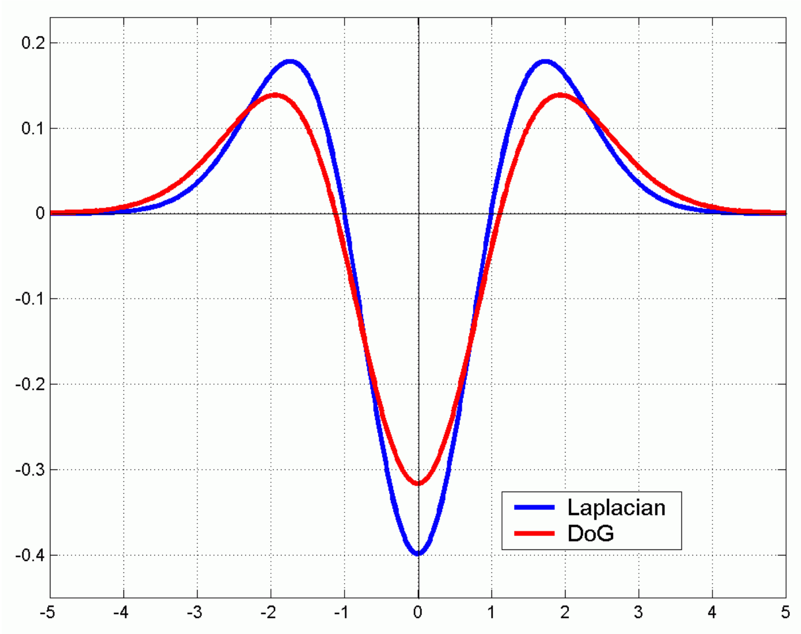
- Approximating the Laplacian with a difference of Gaussians:

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

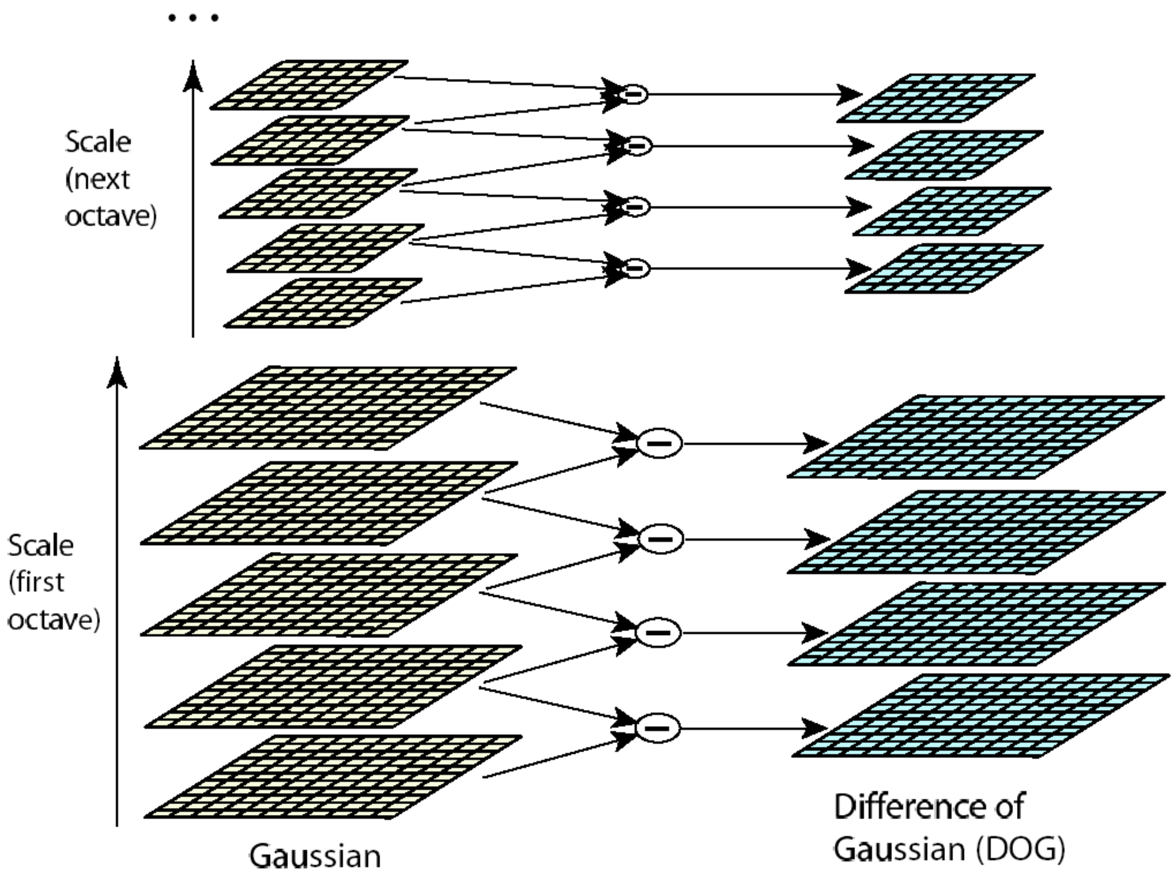
(Laplacian)

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

(Difference of Gaussians)



Efficient implementation



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

Scale Invariant Detection

- Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

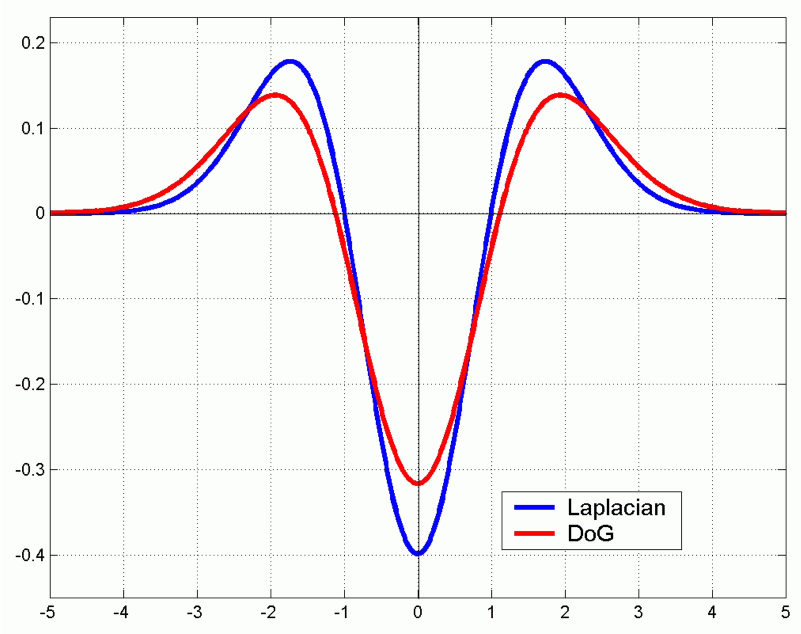
Kernels:

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

(Laplacian)

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

(Difference of Gaussians)



where Gaussian

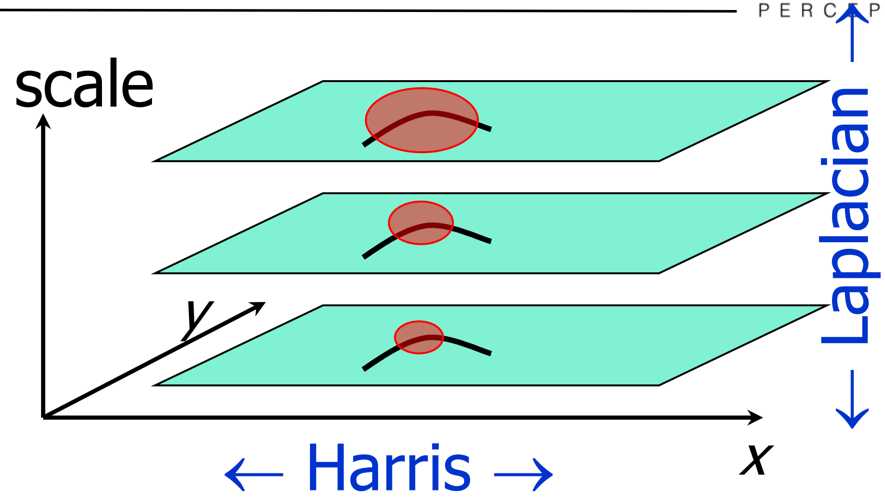
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Scale Invariant Detectors

Harris-Laplacian[1]

Find local maximum of:

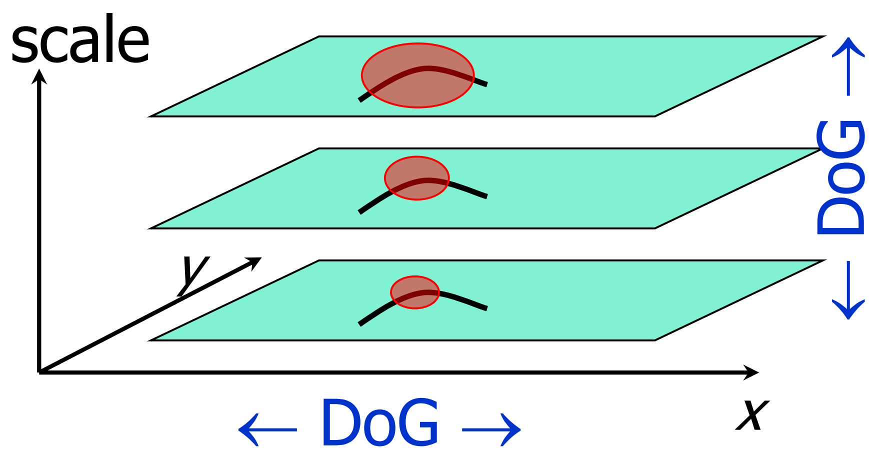
- Harris corner detector in space (image coordinates)
- Laplacian in scale



Laplacian-Laplacian = "SIFT" (Lowe)[2]

Find local maximum of:

- Difference of Gaussians in space and scale



Other options: Hessian, …

Harris does not work well for scale selection

[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

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

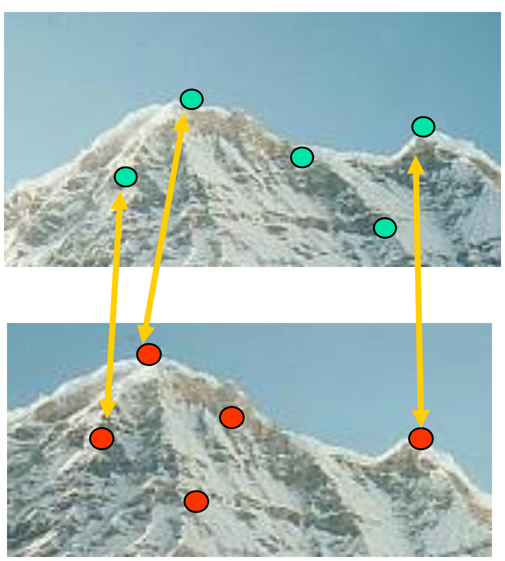
Scale Invariant Detectors

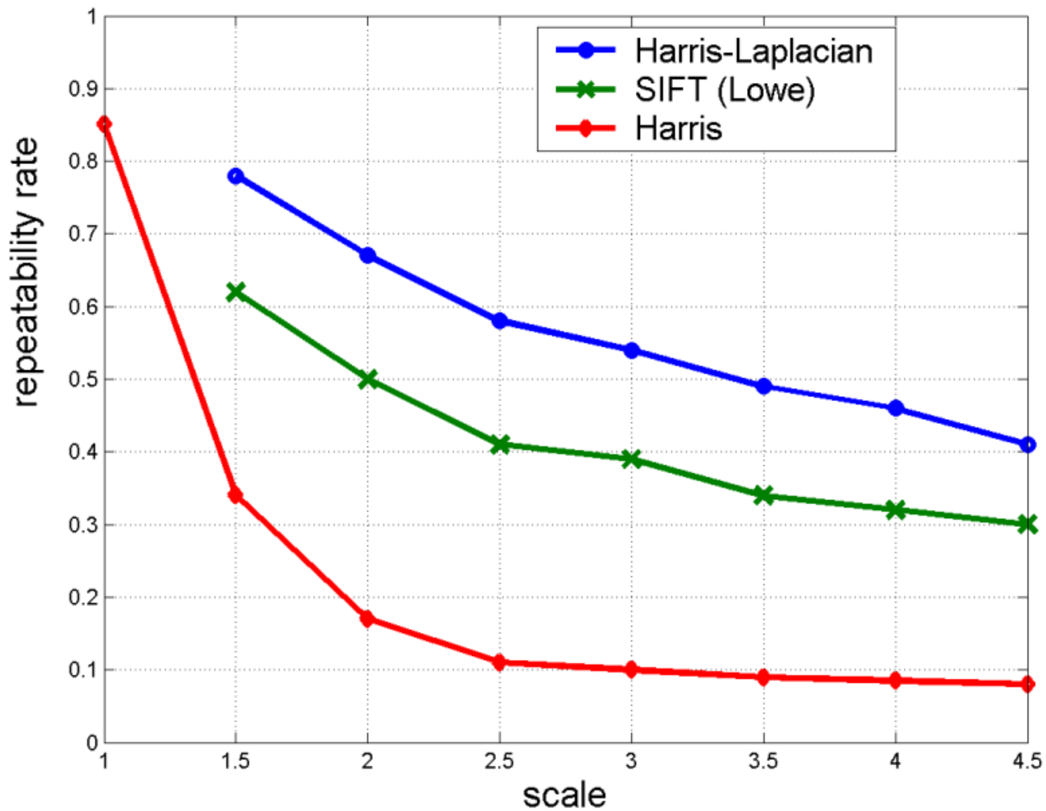
CENTER FOR MACHINE PERCEPTION

- 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

What about 3D rotations?

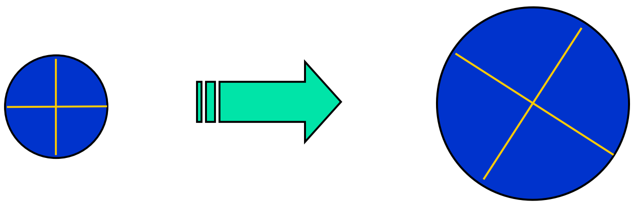
 

What about 3D rotations?

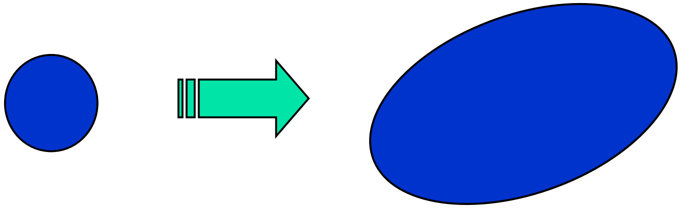
- Affine transformation approximates viewpoint changes for roughly planar objects and roughly orthographic cameras

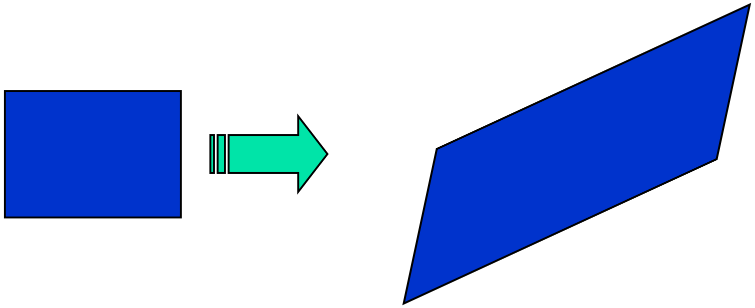
Affine Invariant Detection

- Above we considered:
  Similarity transform (rotation + uniform scale)



- Now we go on to:
  Affine transform (rotation + non-uniform scale)

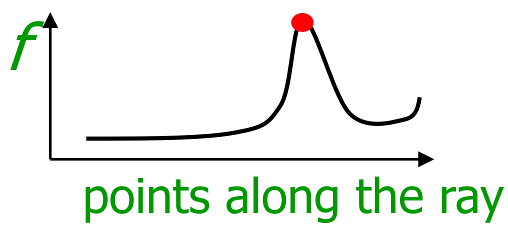




Affine Invariant Detection

CENTER FOR MACHINE PERCEPTION

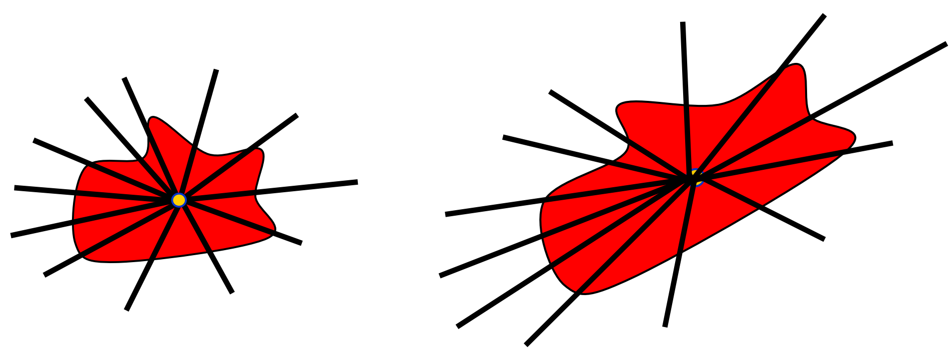
- Take a local intensity extremum as initial point
- Go along every ray starting from this point and stop when extremum of function f is reached



$$f(t) = \frac{\left| I(t) - I_0 \right|}{\frac{1}{t} \int_o^t \left| I(t) - I_0 \right| dt}$$

- We will obtain approximately corresponding regions

Remark: we search for scale in every direction



T.Tuytelaars, L.V.Gool. “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions”. BMVC 2000.

Affine Invariant Detection

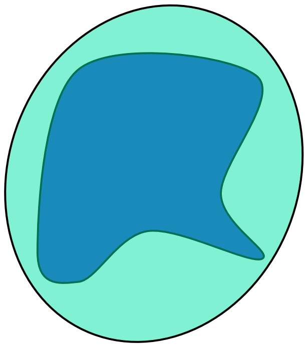
- The regions found may not exactly correspond, so we approximate them with ellipses

- Geometric Moments:

$$m_{pq} = \int_{\mathbb{R}^2} x^p y^q f(x,y)\,dxdy$$

Fact: moments m_{pq} uniquely determine the function f

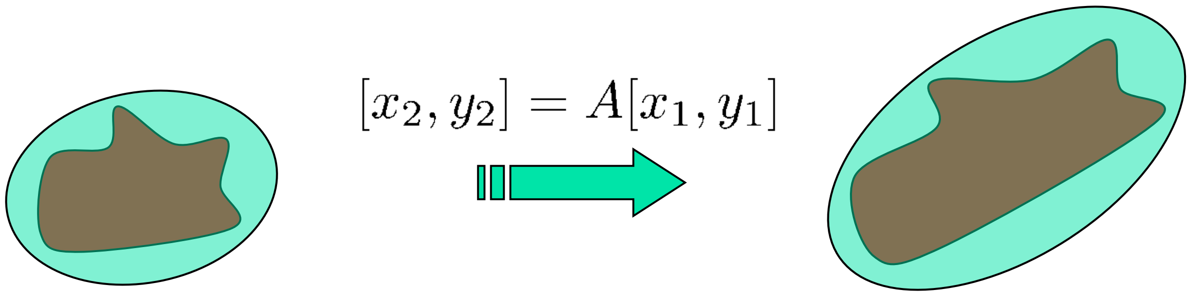
Taking f to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse



This ellipse will have the same moments of orders up to 2 as the original region

Affine Invariant Detection

- Covariance matrix of region points defines an ellipse:



$$[x_2, y_2] = A[x_1, y_1]$$

$$[x_1, y_1]^T \Sigma_1^{-1} [x_1, y_1] = 1 \qquad [x_2, y_2]^T \Sigma_2^{-1} [x_2, y_2] = 1$$

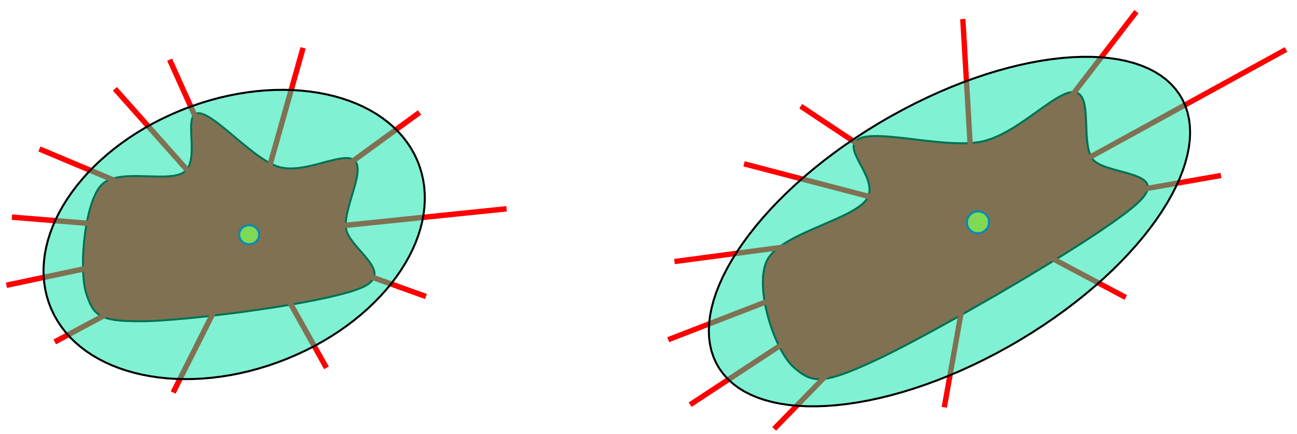
$$\Sigma_1 = \langle [x_1, y_1][x_1, y_1]^T \rangle_{\text{region}_1} \qquad \Sigma_2 = \langle [x_2, y_2][x_2, y_2]^T \rangle_{\text{region}_2}$$

$$\Sigma_2 = A \Sigma_1 A^T$$

Ellipses, computed for corresponding regions, also correspond!

Affine Invariant Detection

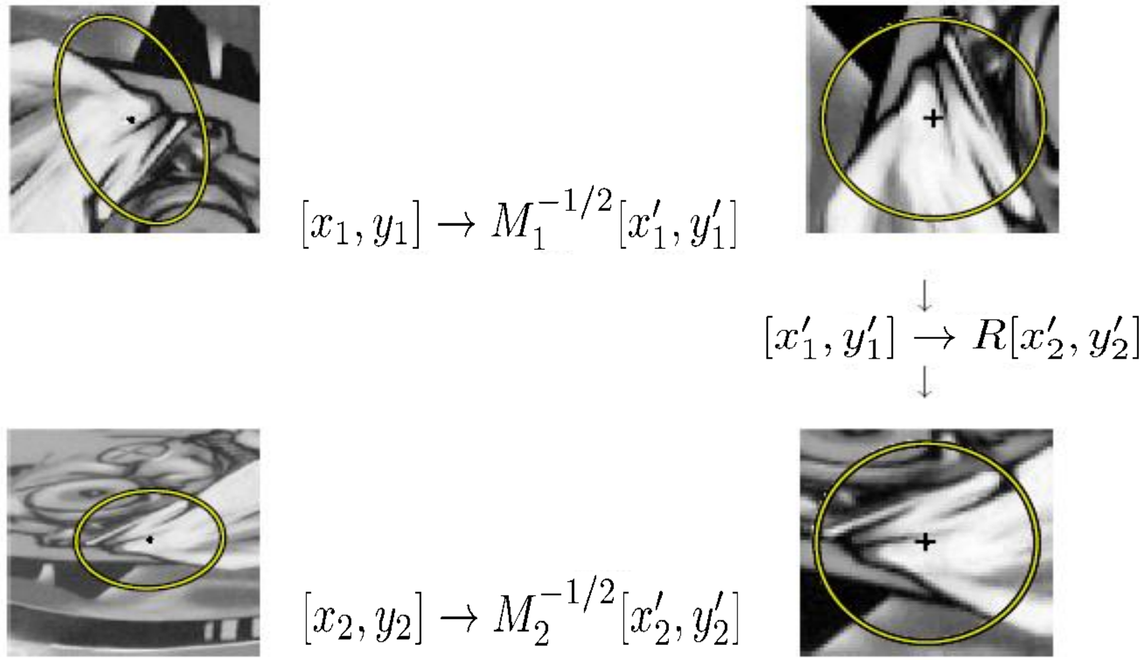
- Algorithm summary (detection of affine invariant region):
  - Start from a *local intensity extremum* point
  - Go in *every direction* until the point of extremum of some function *f*
  - Curve connecting the points is the region boundary
  - Compute *geometric moments* of orders up to 2 for this region
  - Replace the region with *ellipse*



T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions". BMVC 2000.

Harris/Hessian Affine Detector

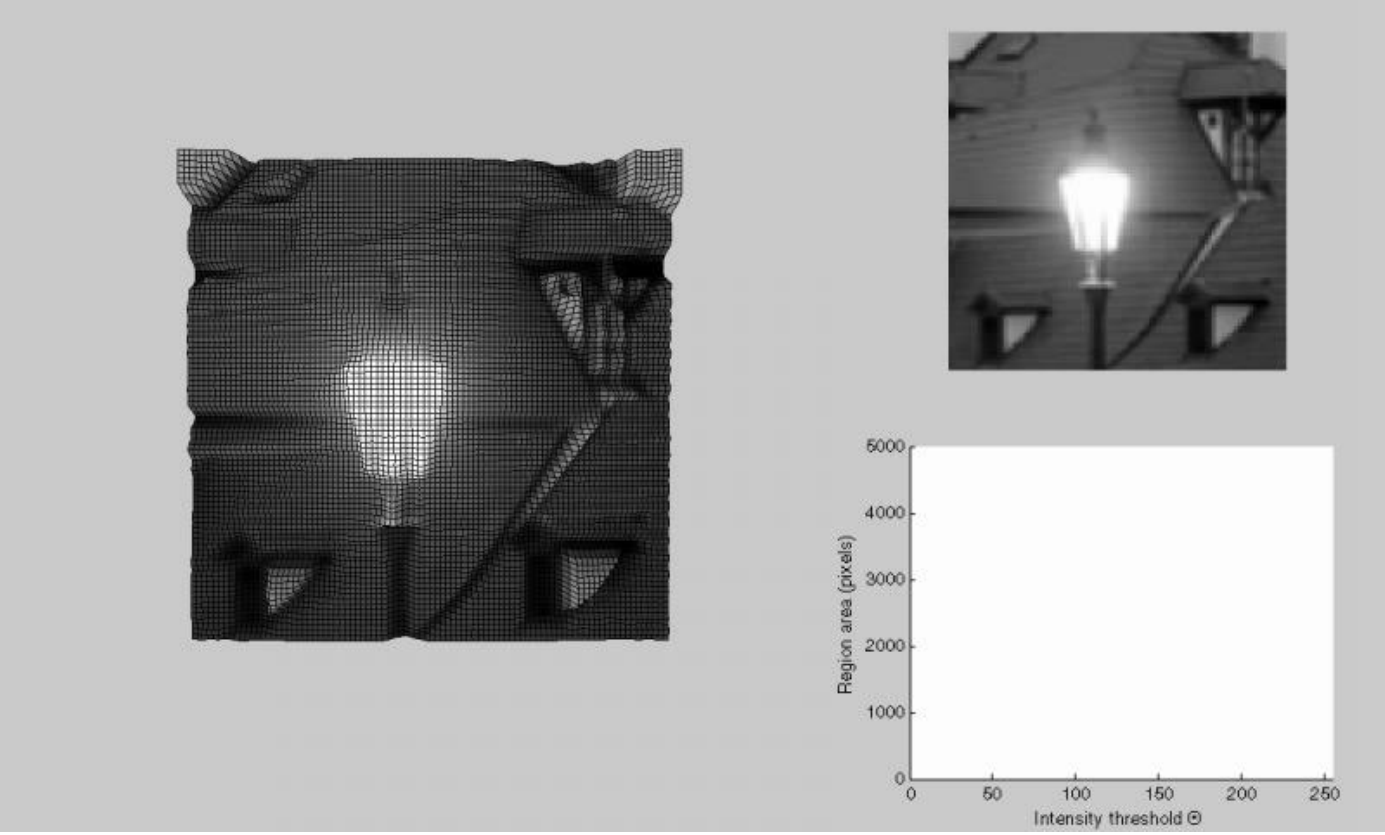
1. Detect initial region with Harris or Hessian detector and select the scale
2. Estimate the shape with the second moment matrix
3. Normalize the affine region to the circular one
4. Go to step 2 if the eigenvalues of the second moment matrix for the new point are not equal



The Maximally Stable Extremal Regions

- Consecutive image thresholding by all thresholds
- Maintain list of Connected Components
- Regions = Connected Components with stable area (or some other property) over multiple thresholds selected

video



J.Matas et.al. "Distinguished Regions for Wide-baseline Stereo". Research Report of CMP, 2001.

The Maximally Stable Extremal Regions

video

MSER Stability

Properties:

- Covariant with continuous deformations of images
- Invariant to affine transformation of pixel intensities
- Enumerated in O(n log log n), real-time computation

MSER regions (in green). The regions 'follow' the object (video1, video2).

Matas, Chum, Urban, Pajdla: "Robust wide baseline stereo from maximally stable extremal regions". BMVC2002

macros.tex
sfmath.sty
cmpitemize.tex

Thank you for your attention.