Corner detection

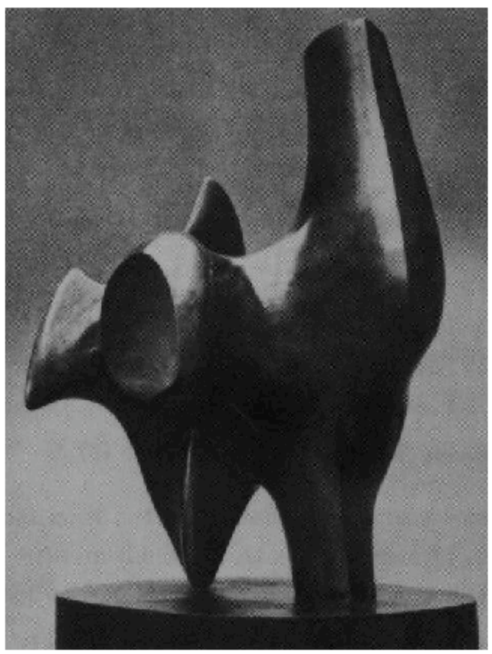
Lecture 4

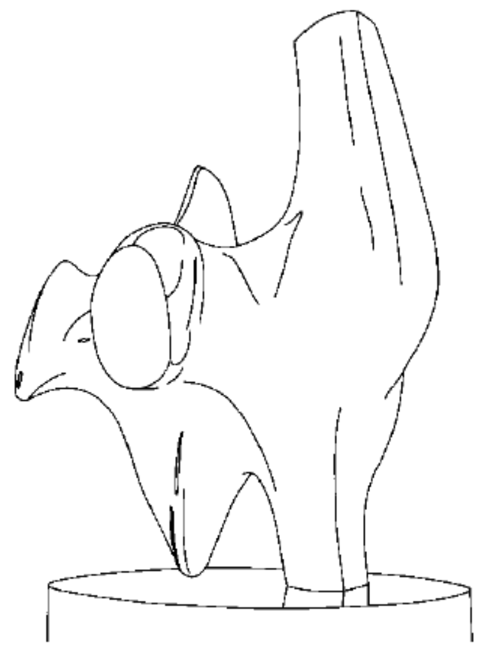
CS 664 – Spring 2008

Last time: Edge detection

- Convert a gray or color image into set of curves
  - Represented as binary image

- Capture properties of shapes



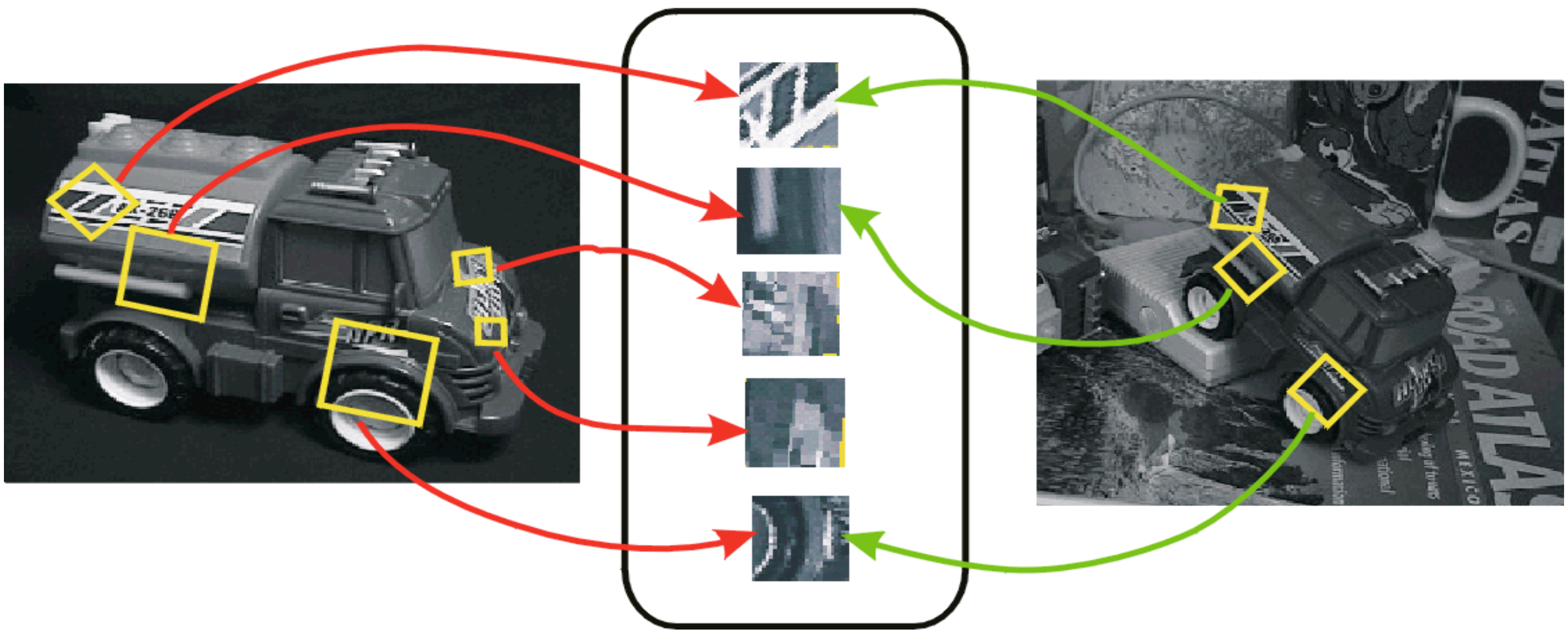


A problem with edges

- Edges are insensitive to intensity changes, but not to other image transformations

Enter interest point detection

- Goal: Find points that are stable across scaling, rotation, etc.
  - e.g. corners



SIFT Features

Corners

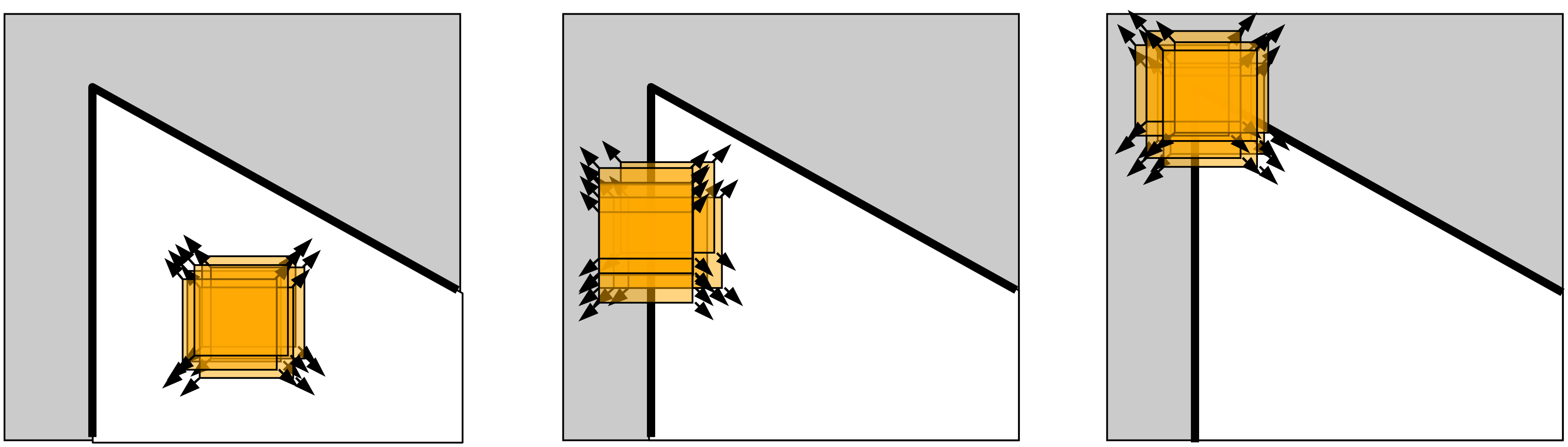
- A corner is characterized by a region with intensity change in two different directions

- Use local derivative estimates
  - Gradient oriented in different directions

- Not as simple as looking at gradient (partial derivatives) wrt coordinate frame

Corner detection: the basic idea

- At 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

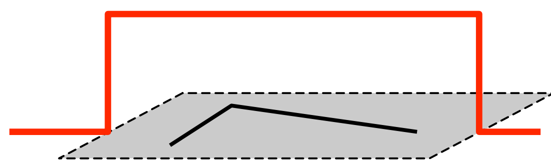
"corner":
significant change
in all directions

A simple corner detector

- Define the sum squared difference (SSD) between an image patch and a patch shifted by offset (x,y):

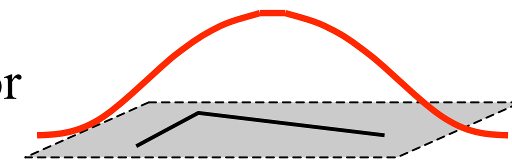
$$S(x,y) = \sum_u \sum_v w(u,v) \left(I(u,v) - I(u-x, v-y) \right)^2$$

where $w(u,v)$ =



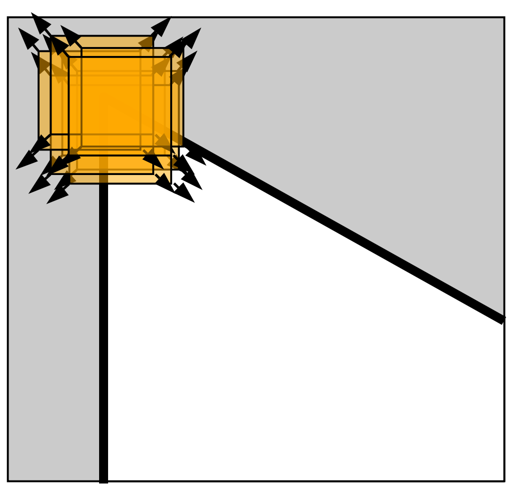
1 in window, 0 outside

or



Gaussian

- If s(x,y) is high for shifts in all 8 directions, declare a corner.
  - Problem: not isotropic



Harris corner detector derivation

- Second-order Taylor series approximation:

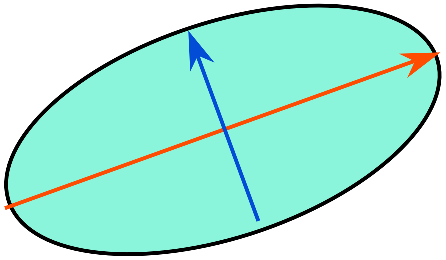
$$S(x,y) = \sum_u \sum_v w(u,v) \; (I(u,v) - I(u-x, v-y))^2$$

$$S(x,y) \approx \frac{1}{2}(x \quad y) A \begin{pmatrix} x \\ y \end{pmatrix}$$

  - where A is defined in terms of partial derivatives $I_x = \partial I / \partial x$ and $I_y = \partial I / \partial y$ summed over (u,v):

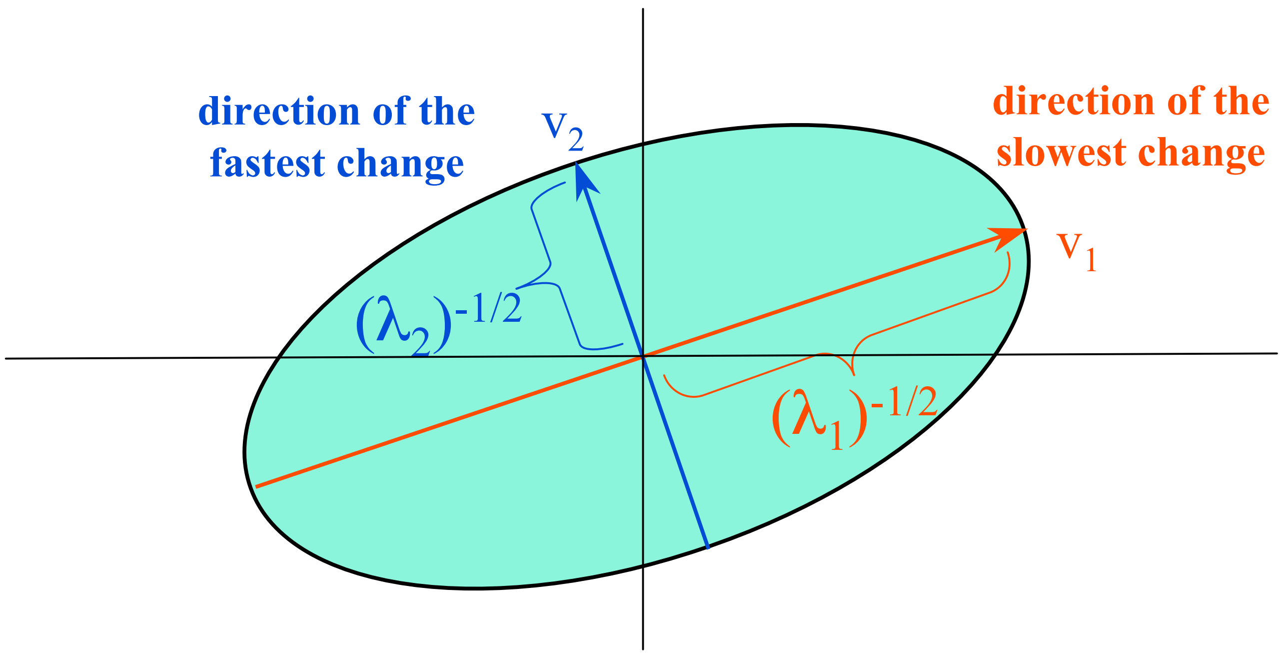
$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

  - For constant t, S(x,y) < t is an ellipse



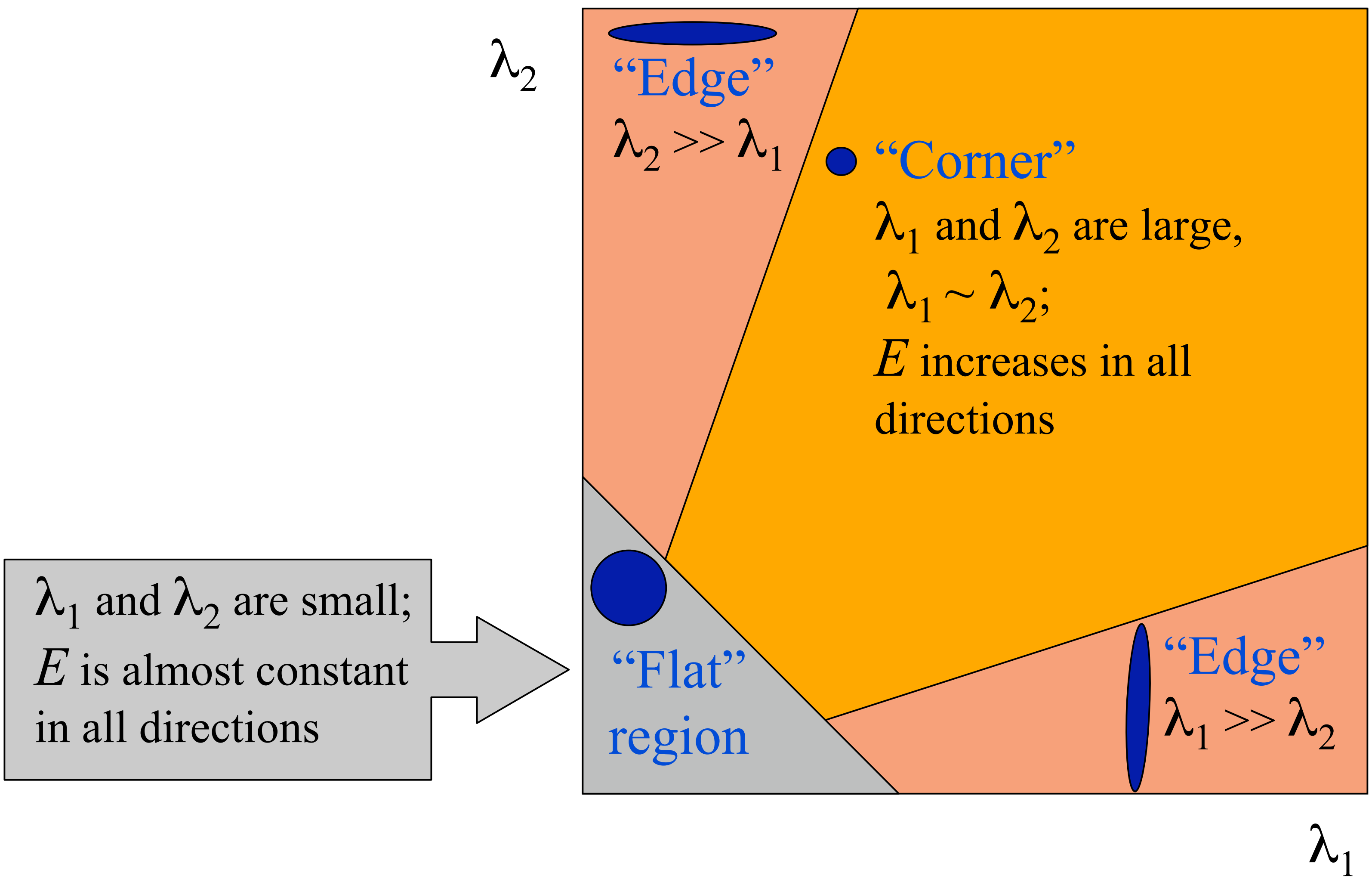
Eigenvector analysis

- The eigenvectors v_1, v_2 of *A* give an orthogonal basis for the ellipse
  - I.e. directions of fastest and slowest change
  - for $\lambda_2 > \lambda_1$, v_1 is the direction of fastest change (minor axis of ellipse) and v_2 is the direction of slowest change (major axis)



Classify points based on eigenvalues

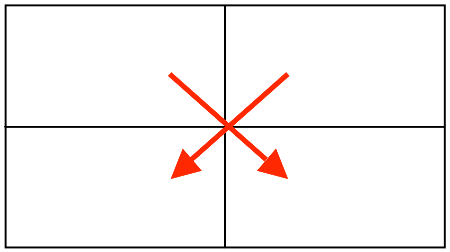
- Classification of image points using eigenvalues of *M*:



Harris corner detection (1988)

- Smooth the image slightly

- Compute derivatives on 45° rotated axis
  - Eigenvectors thus oriented wrt that grid
  - Eigenvalues not affected



- Find eigenvalues λ_1,λ_2 of A ($\lambda_1<\lambda_2$)
  - If both large then high gradient in multiple directions
    - When λ_1 larger than threshold detect a corner

  - Eigenvalues can be computed in closed form

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} \qquad \lambda_1 = \tfrac{1}{2}\left(a+c-\sqrt{(a-c)^2+4b^2}\right)$$

$$\lambda_2 = \tfrac{1}{2}\left(a+c+\sqrt{(a-c)^2+4b^2}\right)$$

Harris corner detection

- But square roots are expensive
  - Approximate corner response function that avoids square roots:

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

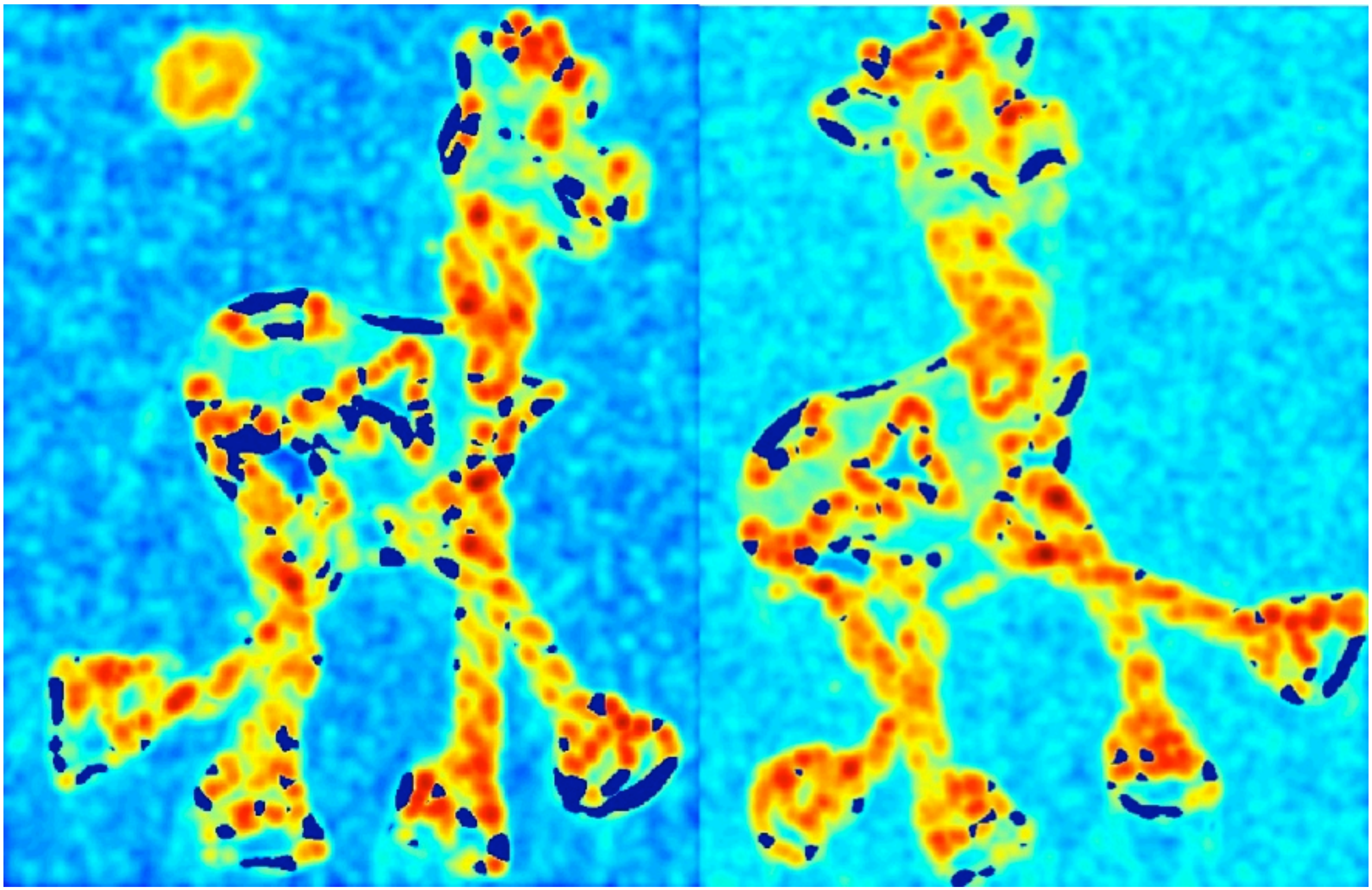
with *k* is set empirically

- After thresholding, keep only local maxima of *R* as corners
  - prevents multiple detections of the same corner

Harris detector, step-by-step

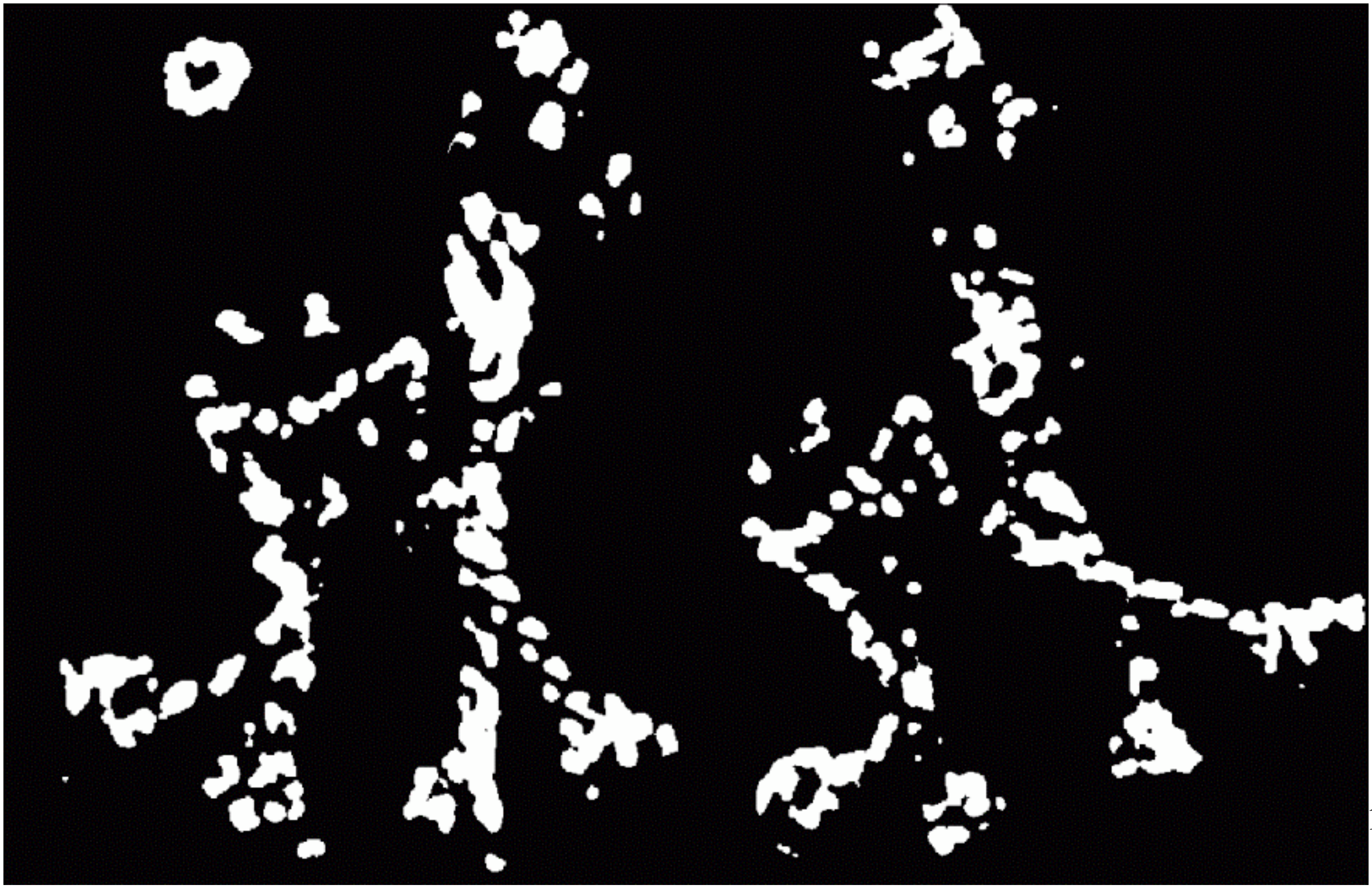
Harris detector, step-by-step

- Compute corner response *R*



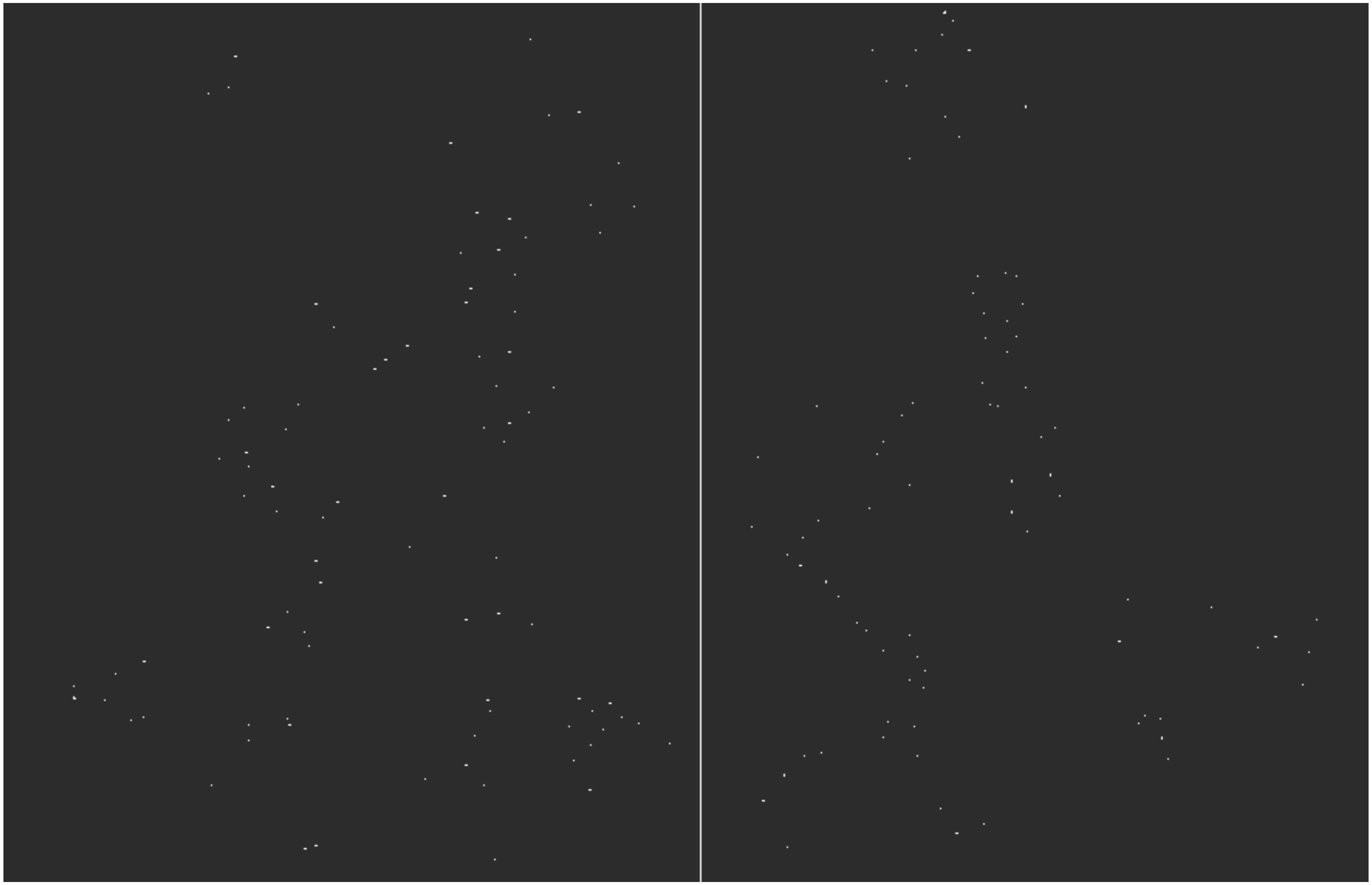
Harris detector, step-by-step

- Threshold on corner response *R*

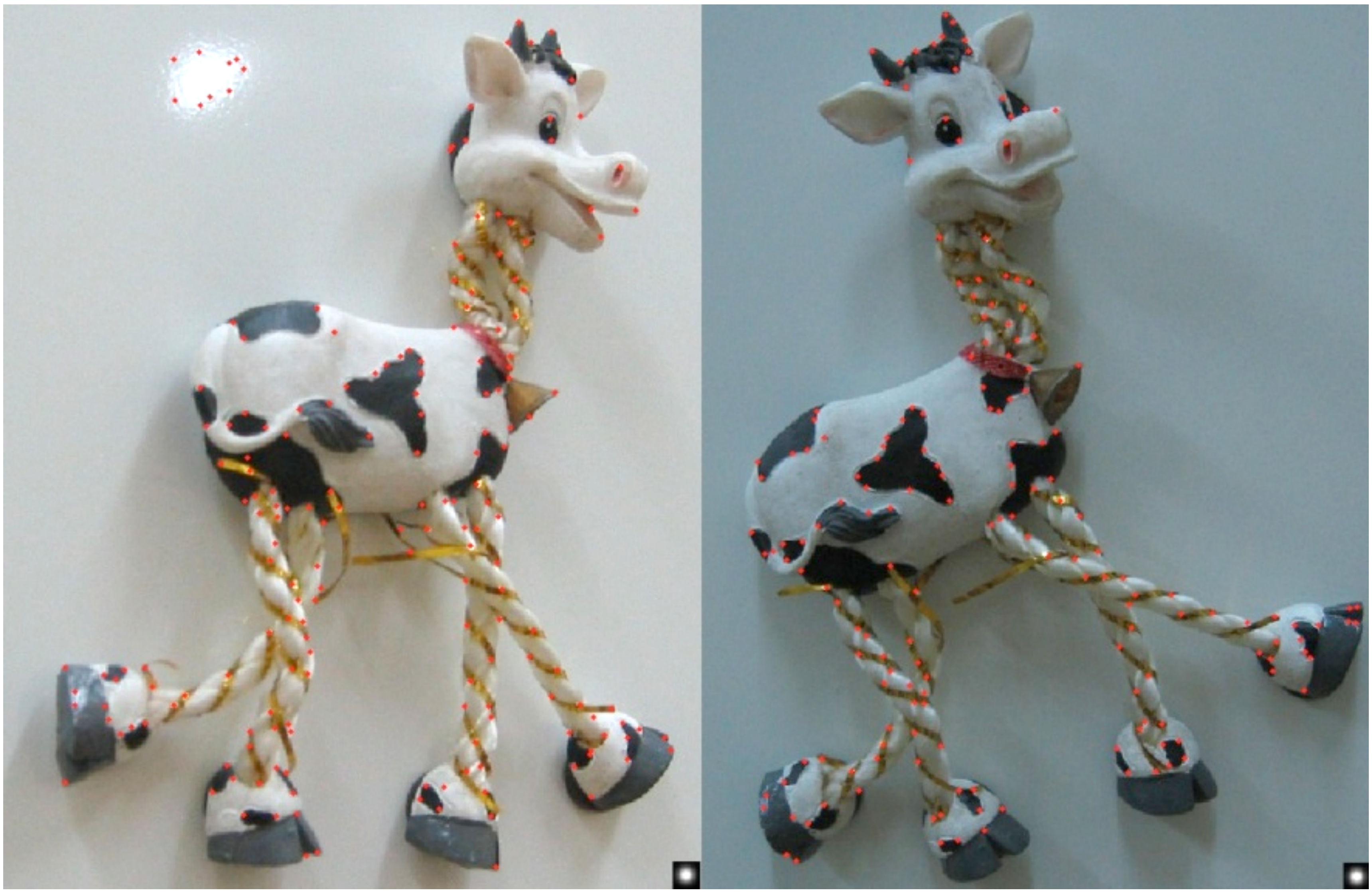


Harris detector, step-by-step

- Take only local maxima of *R*



Harris detector result



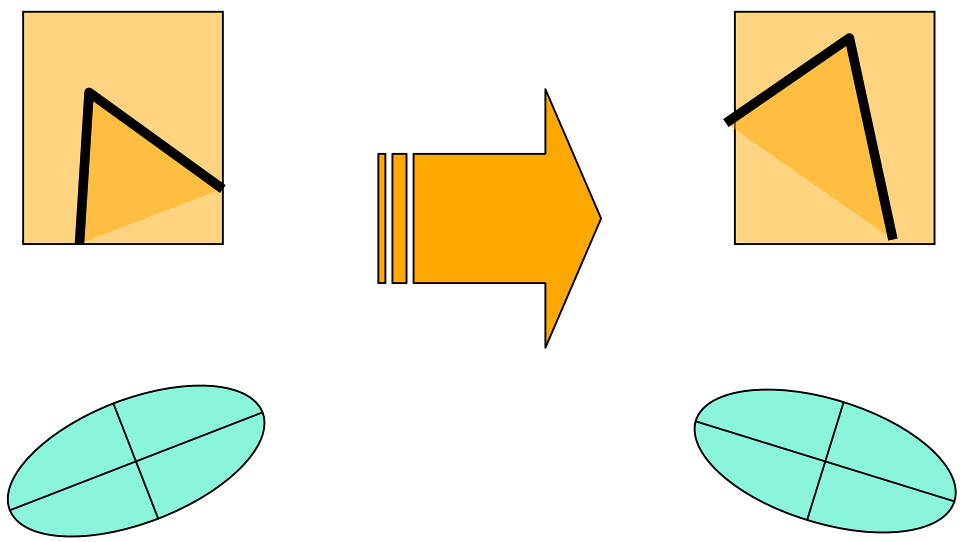
KLT corner detector

- Kanade-Lucas-Tomasi (1994)

- Very similar to Harris, but with a greedy corner selection criterion
  - Put all points for which λ_1 > thresh in a list L
  - Sort the list in decreasing order by λ_1
  - Declare highest pixel p in L to be a corner. Then remove all points from L that are within a DxD neighborhood of p
  - Continue until L is empty

Harris detector properties

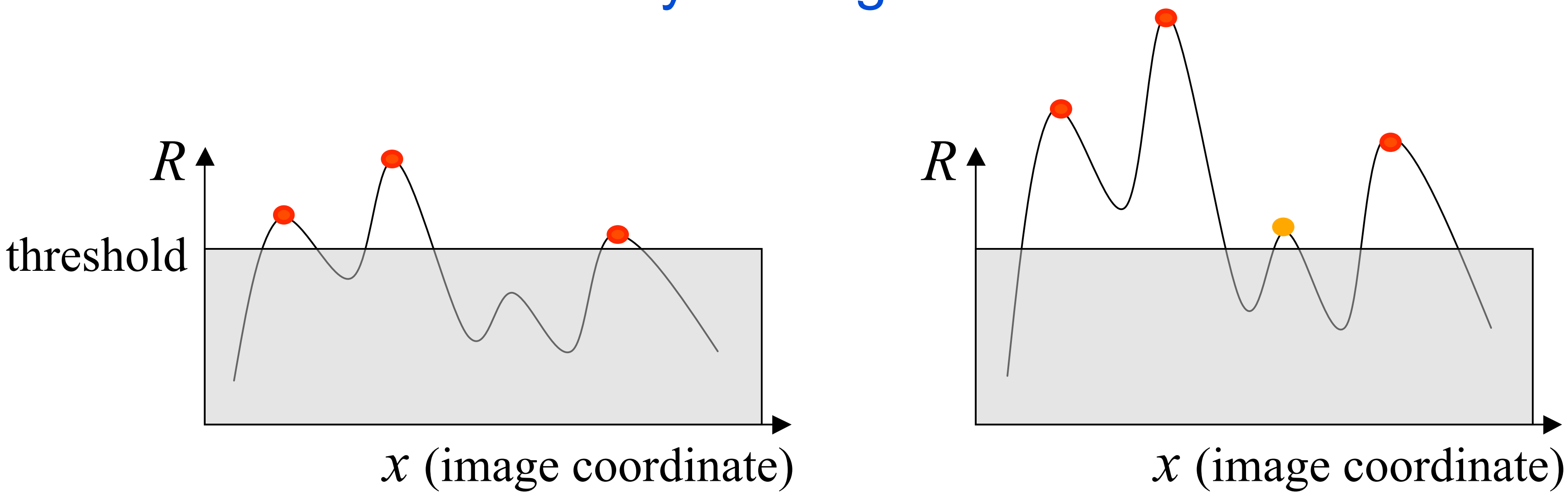
- Rotation invariance
  - Ellipse (eigenvectors) rotate but shape (eigenvalues) remain the same
  - Corner response *R* is invariant to image rotation



Harris detector properties

- Invariant to intensity shift: $I' = I + b$
  - only derivatives are used, not original intensity values

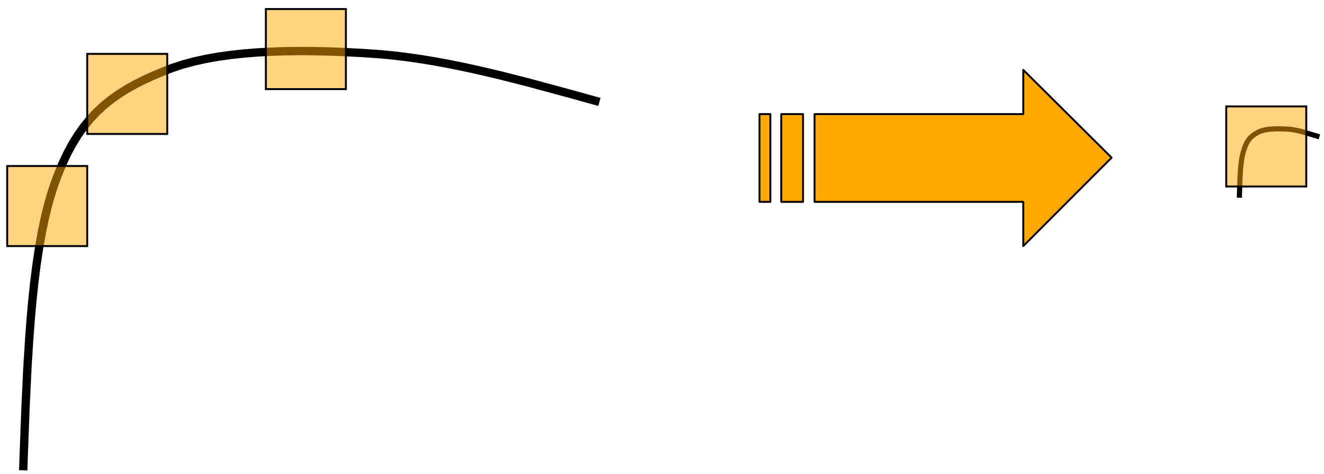
- Insensitive to intensity scaling: $I' = a\,I$



- So Harris is insensitive to affine intensity changes
  - I.e. linear scaling plus a constant offset, $I' = a\,I + b$

Harris detector properties

- But Harris is *not* invariant to image scale



All points will be classified as edges

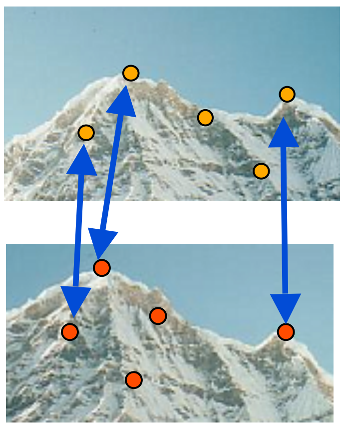
Corner !

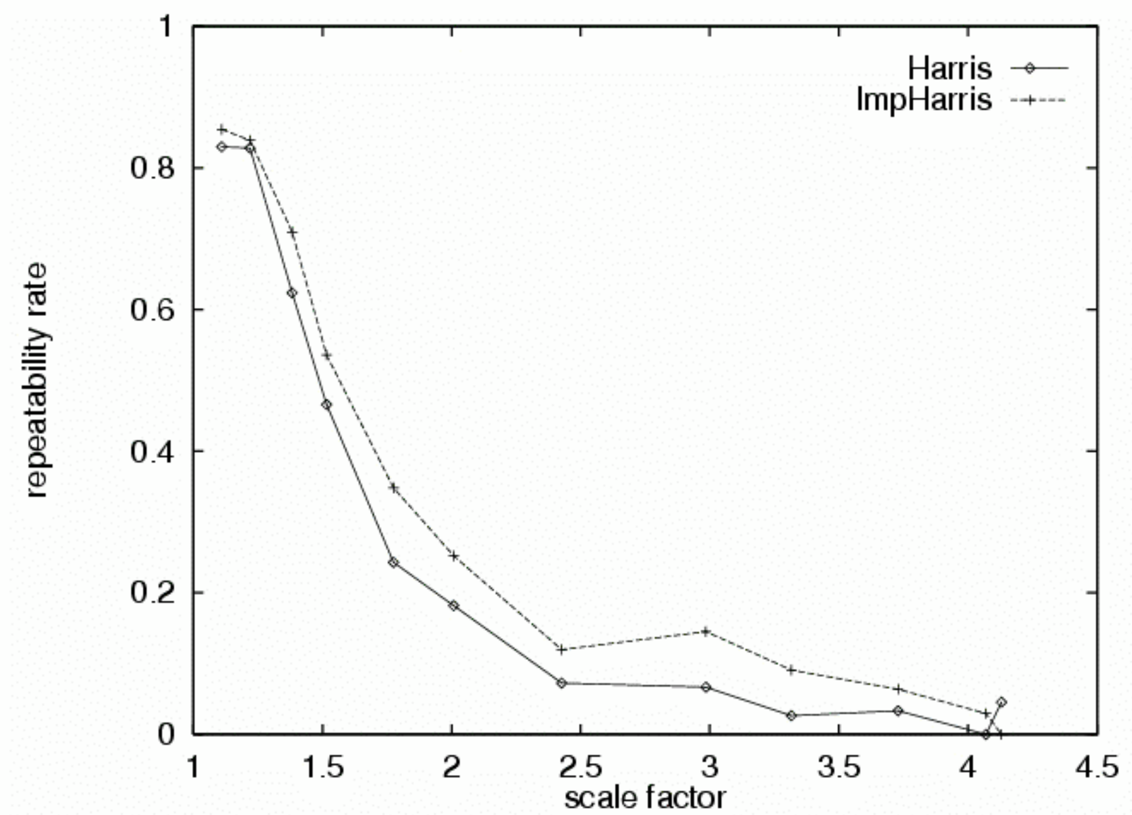
Experimental evaluation

- Quality of Harris detector for different scale changes

Repeatability rate:

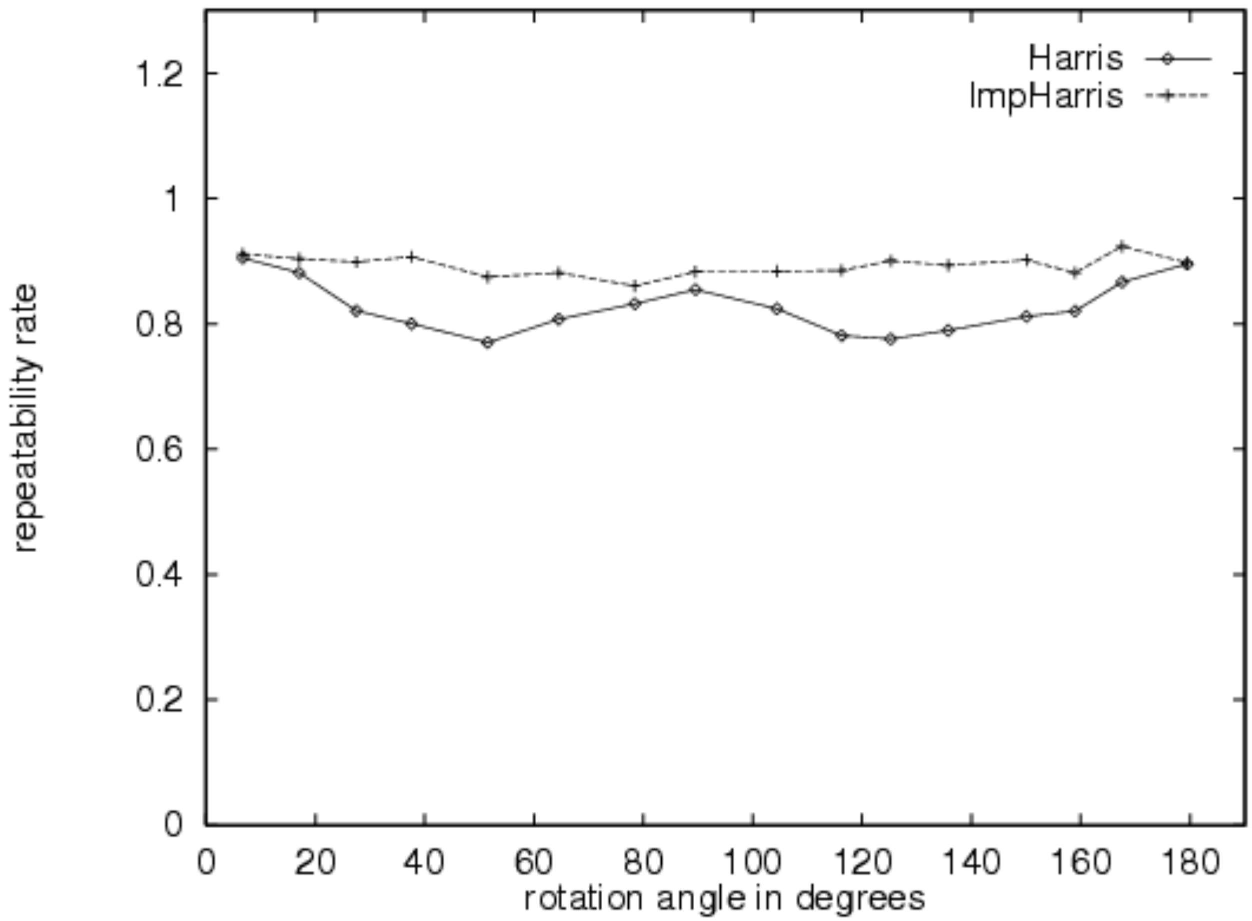
$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$





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

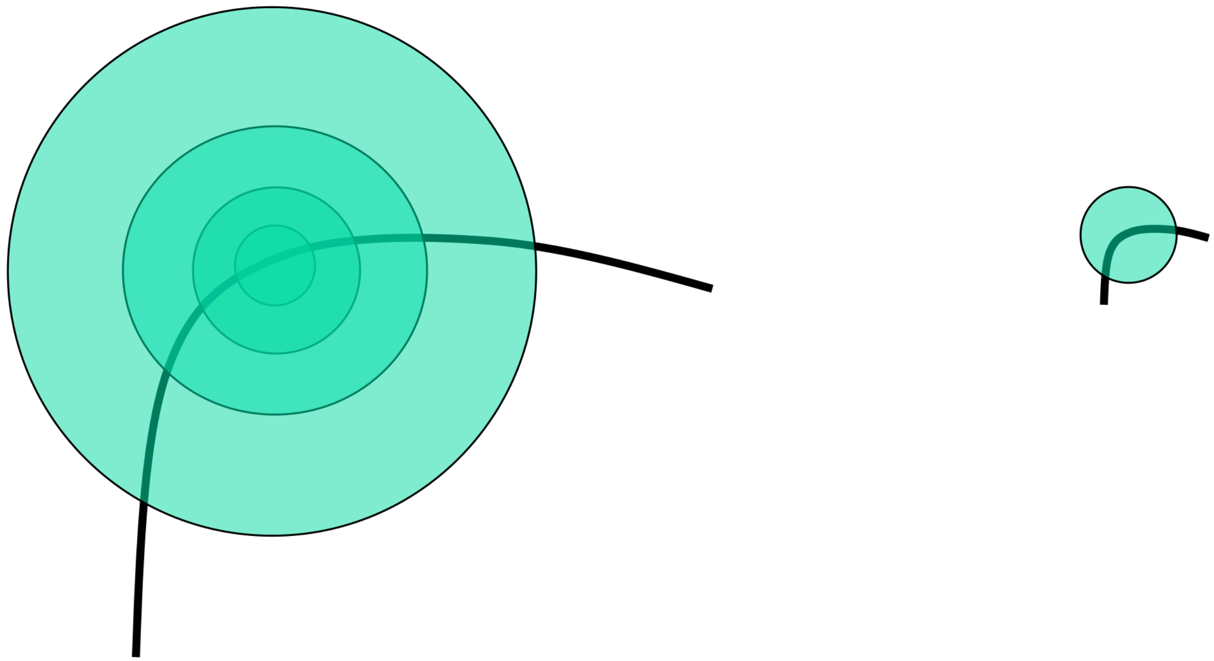
Experimental evaluation



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

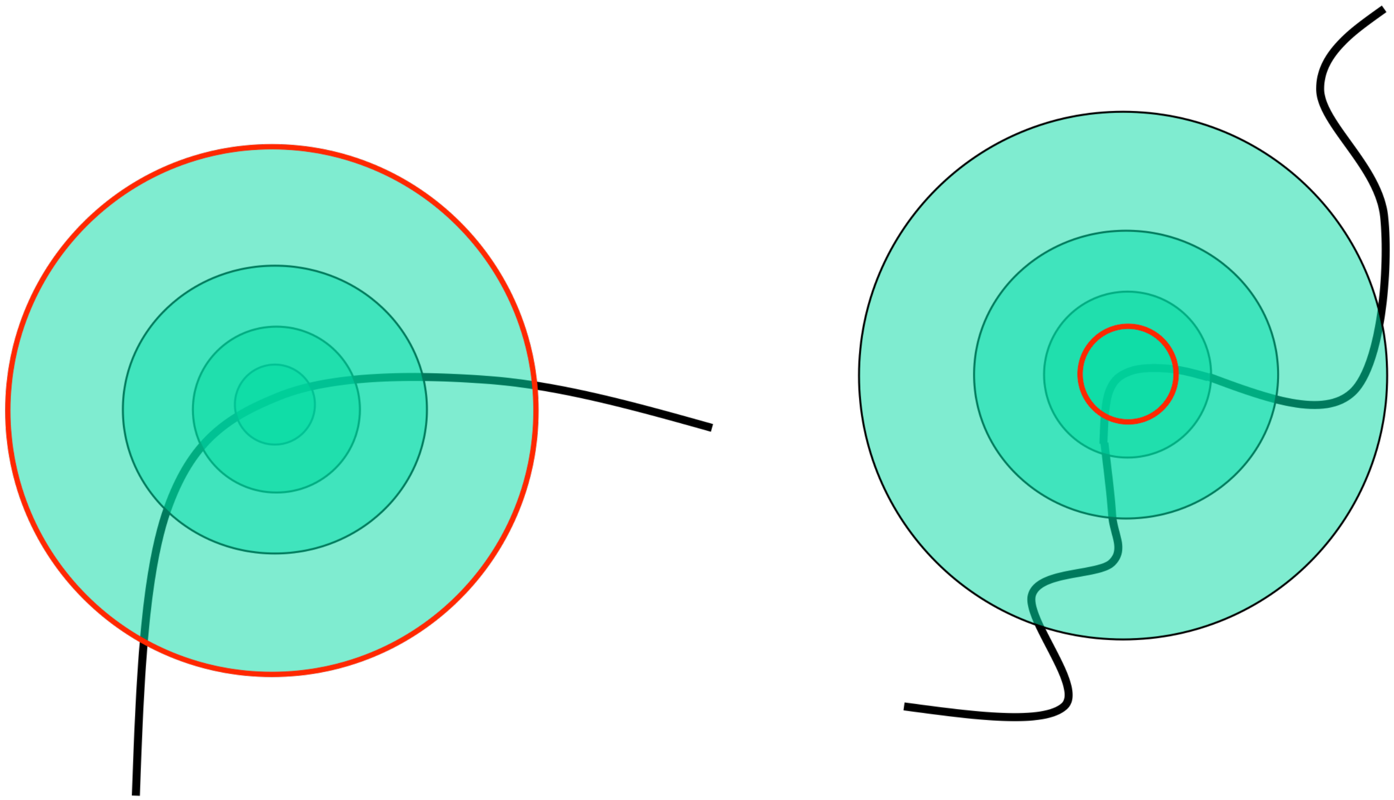
Scale invariant interest point detection

- 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

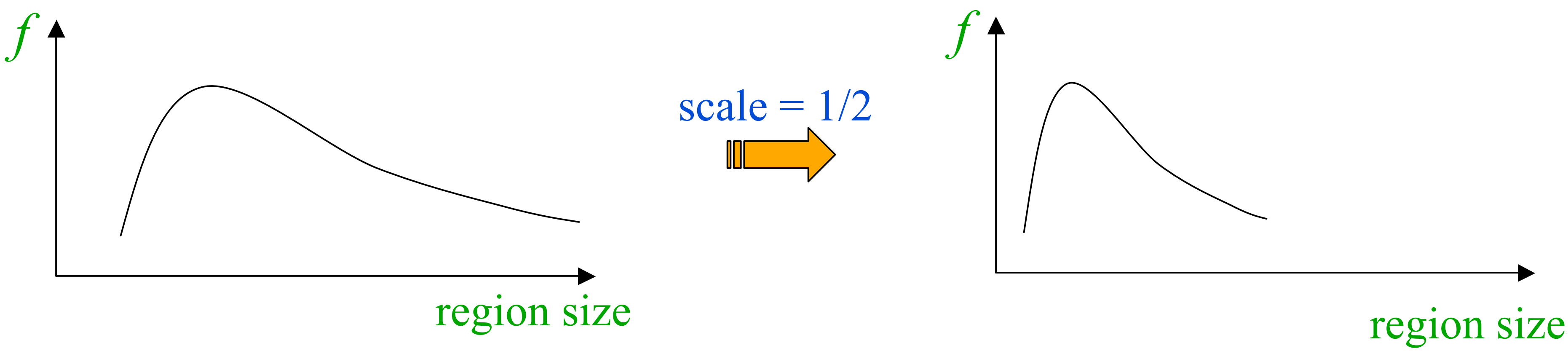
- The problem: how do we choose corresponding circles ***independently*** in each image?



A solution

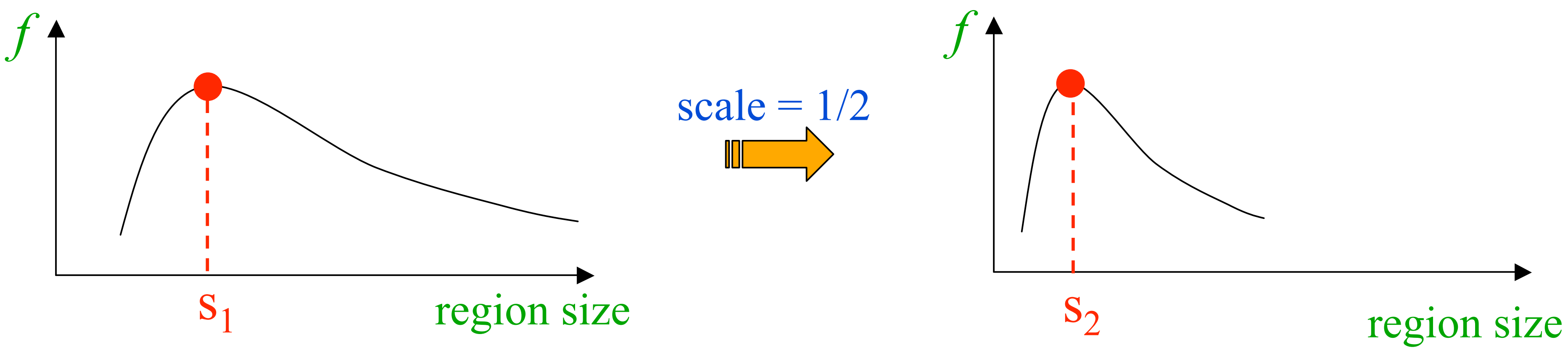
- Design a function which is "scale invariant"
  - I.e. value is the same for two corresponding regions, even if they are at different scales
  - Example: average intensity is the same for corresponding regions, even of different sizes

- For a given point in an image, consider the value of f as a function of region size (circle radius)



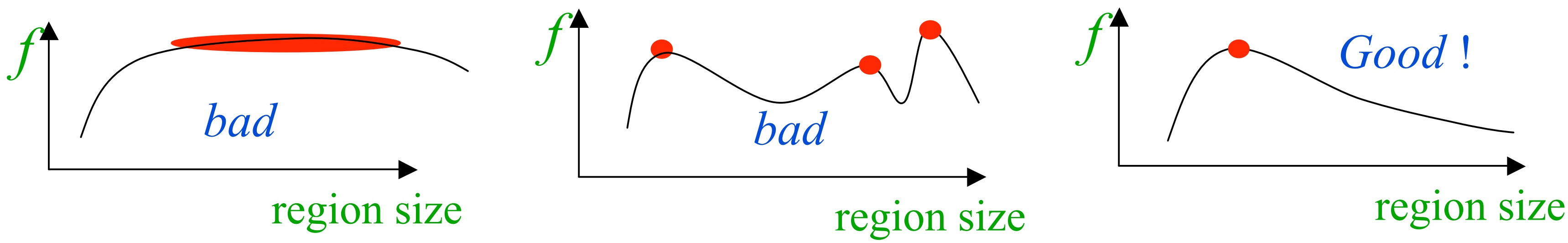
A solution

- Take a local maximum of this function
  - The region size at which maximum is achieved should be invariant to image scale
- This scale invariant region size is determined independently in each image



Choosing a function

- A good function for scale detection has one sharp peak



- A function that responds to image contrast is a good choice
  - e.g. convolve with a kernel like the Laplacian or the Difference of Gaussians

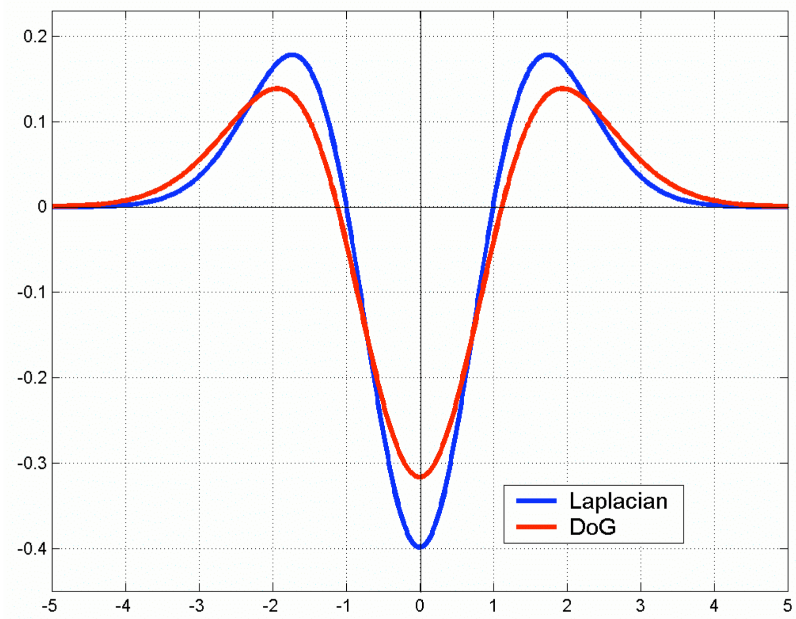
Laplacian vs. Difference of Gaussians

- Common choices:
  - Laplacian:

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

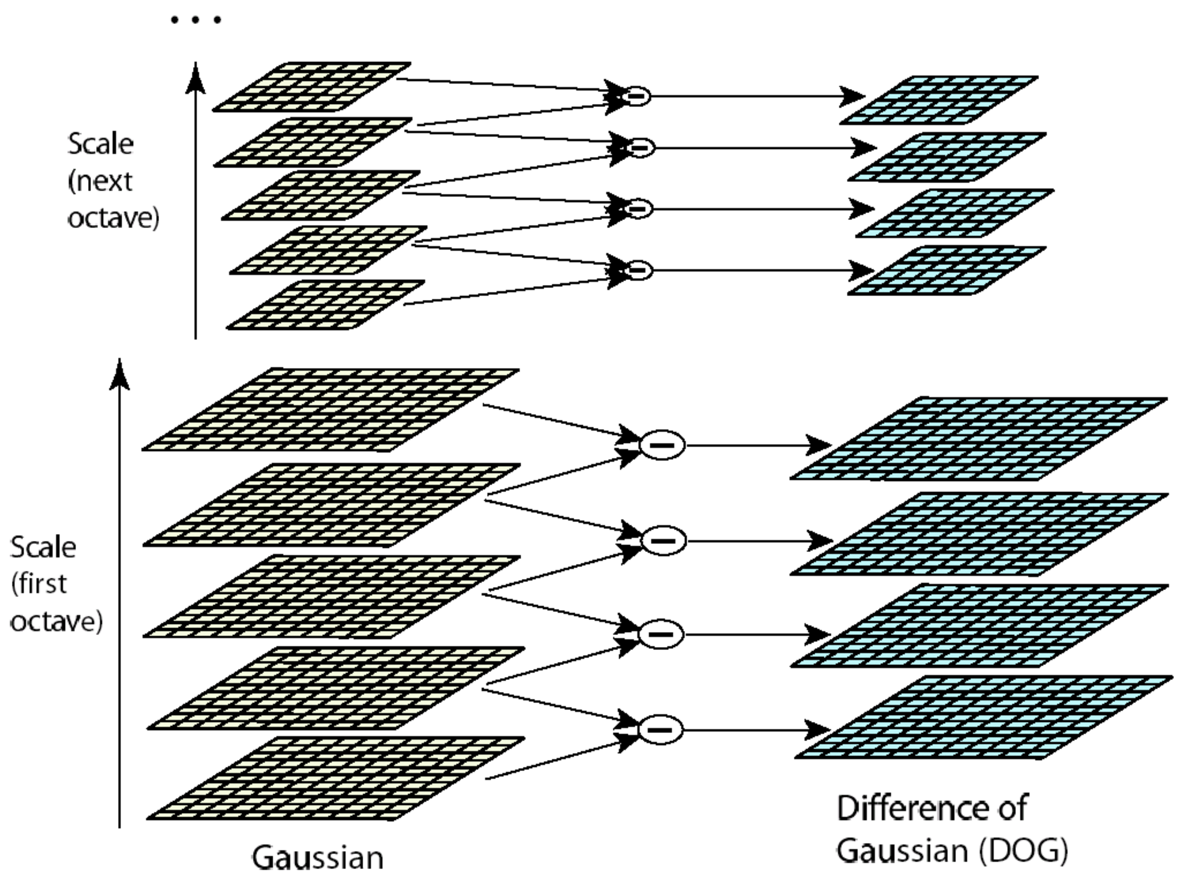
  - Difference of Gaussians:

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



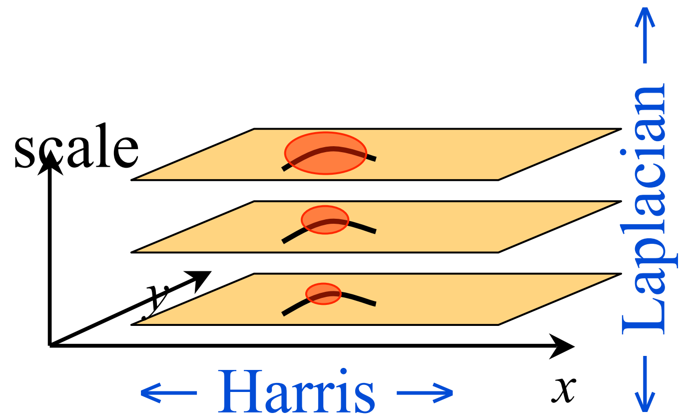
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Differences of Gaussians

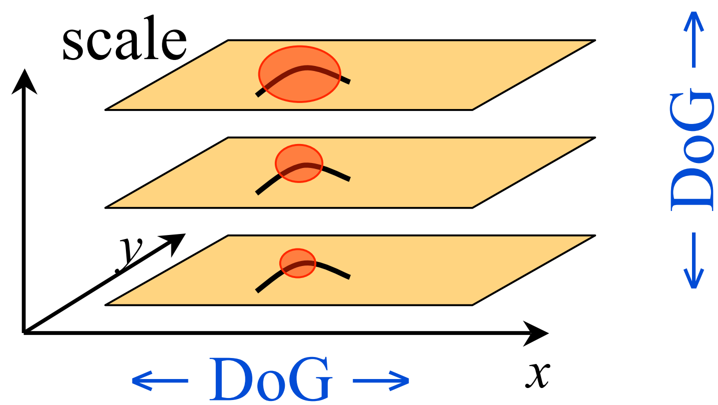


Two approaches: Harris-Laplacian vs. SIFT

- **Harris-Laplacian**[1] finds local maximum of
  - Harris corner detector in image space
  - Laplacian in scale space



- **SIFT (Lowe)**[2] finds local maximum of
  - DoG in image space
  - DoG in scale space



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

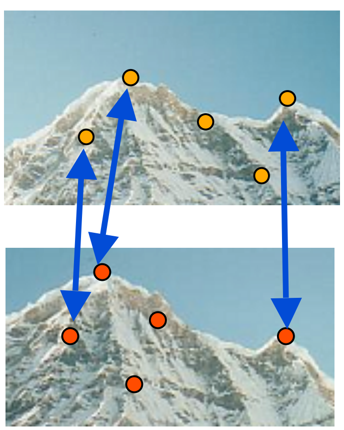
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

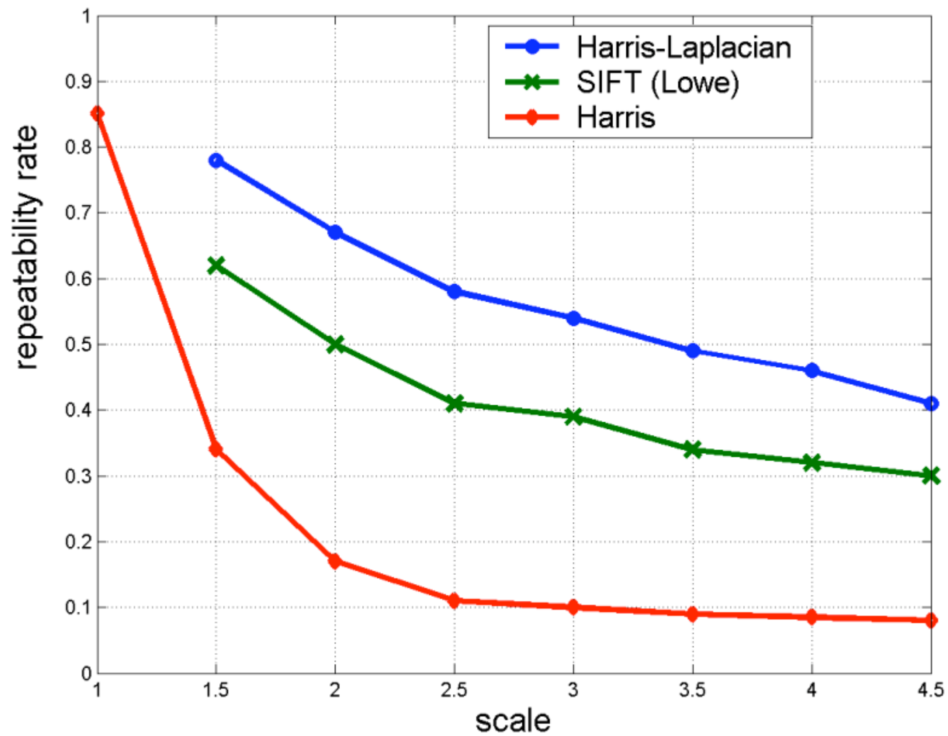
Scale invariance experiments

- 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