Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2020-2021

Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

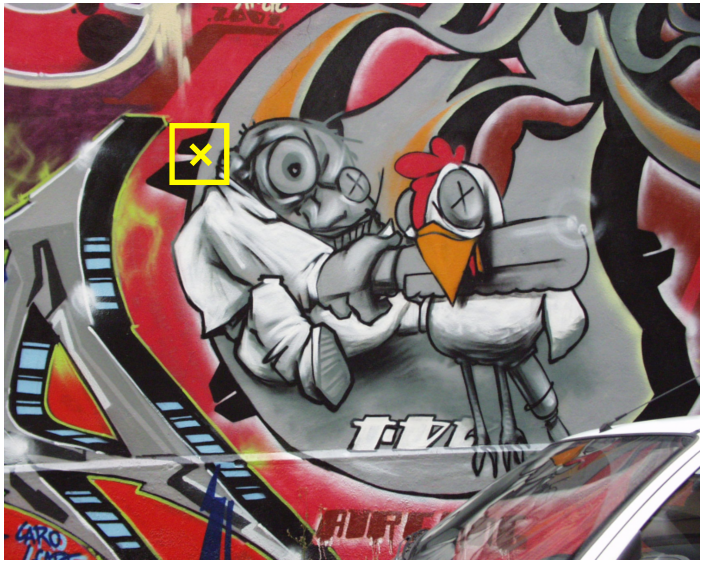
   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

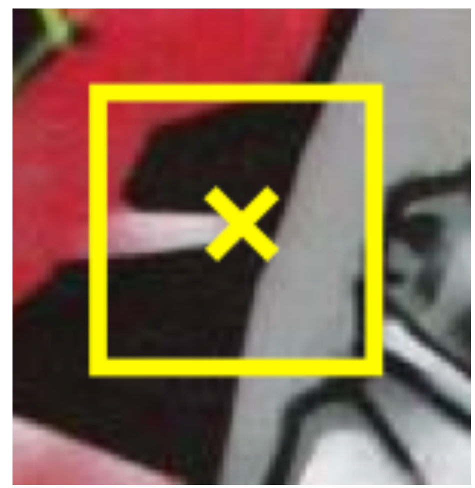
Local features

- local feature = interest points = keypoint + region descriptor
- local image feature = a tiny patch in the image that is invariant to image scaling, rotation and change in illumination. It is geometrically (translation, rotation, ...) and photometrically (brightness, exposure, ...) invariant.

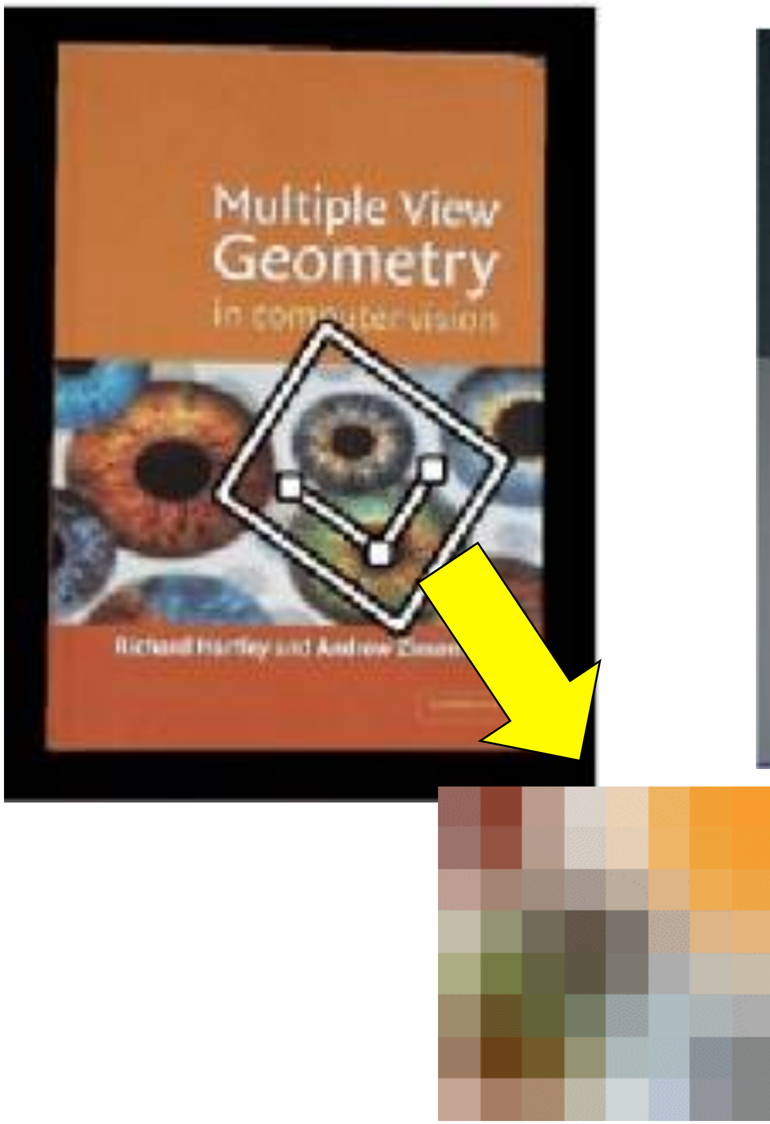


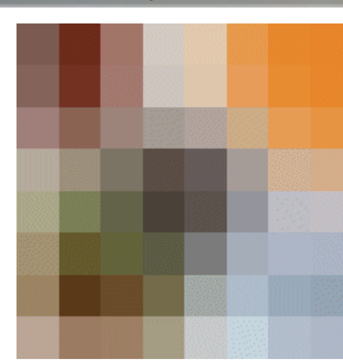
Local features

- local feature = interest points = keypoint + region descriptor
- local image feature = a tiny patch in the image that is invariant to image scaling, rotation and change in illumination. It is geometrically (translation, rotation, ...) and photometrically (brightness, exposure, ...) invariant.



Geometric transformations





Exemples:
translation,
rotation,
scaling

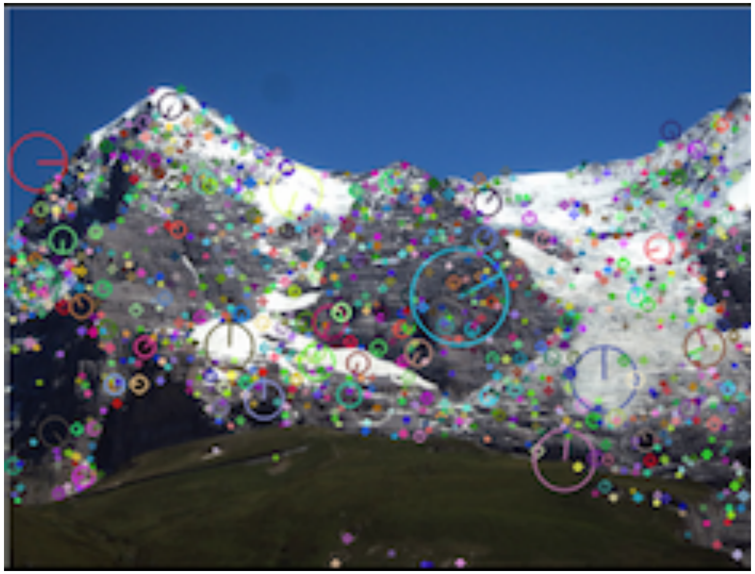
Photometric transformations

Local features – lab class 4

- local image feature = a tiny region in the image that's invariant to image scaling, rotation and change in illumination.

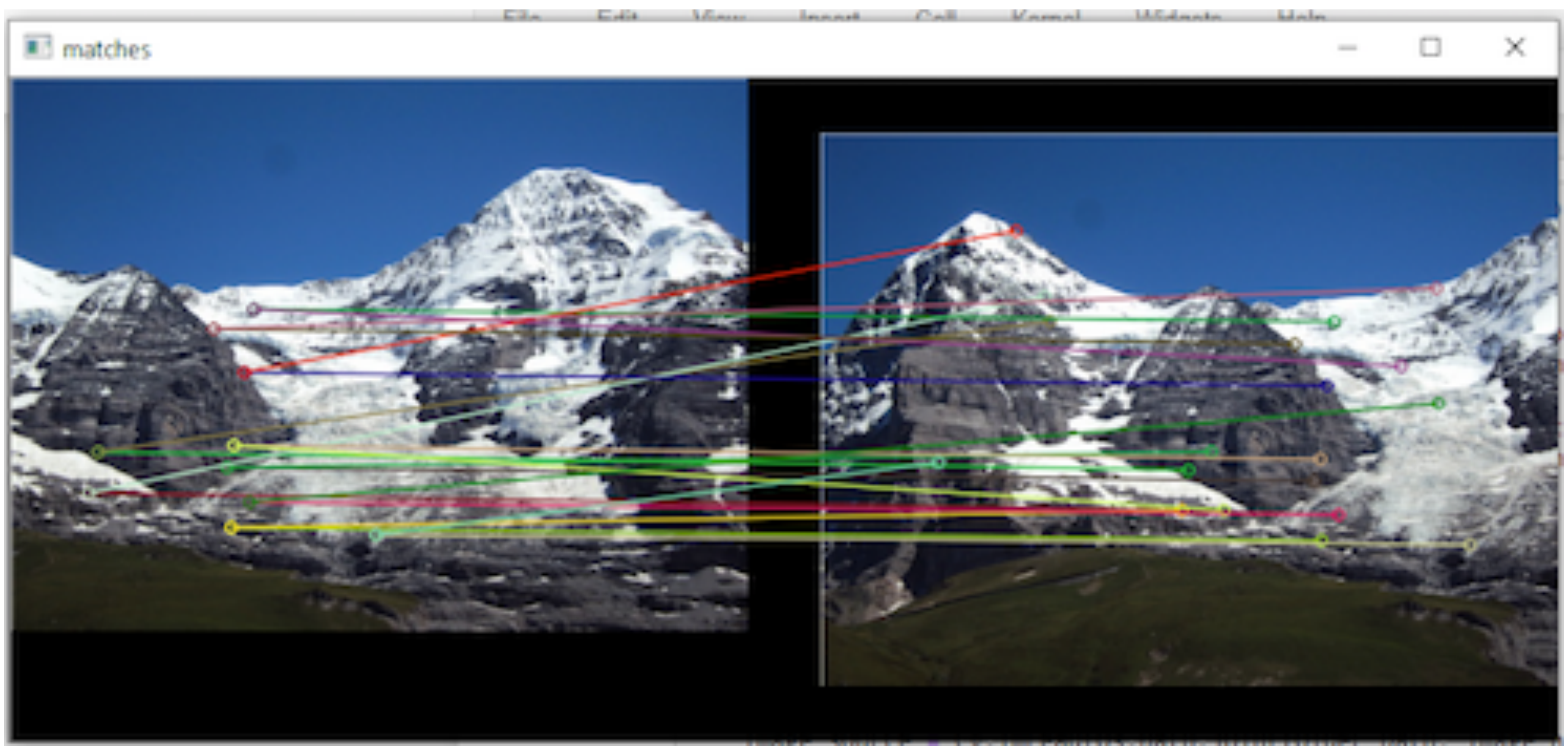
Local features – lab class 4

- local image feature = a tiny region in the image that's invariant to image scaling, rotation and change in illumination.



Matching local features – lab class 4

- matching local image feature = find correspondence points between images based on local features that look the same

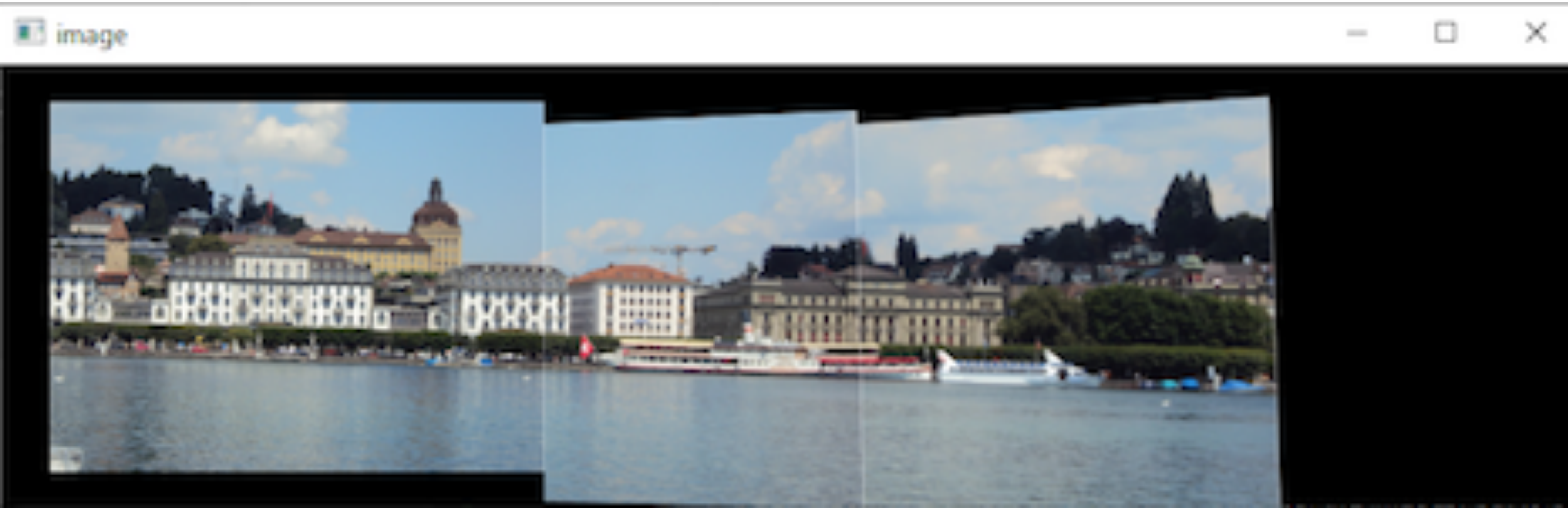


Panorama stitching – lab class 4

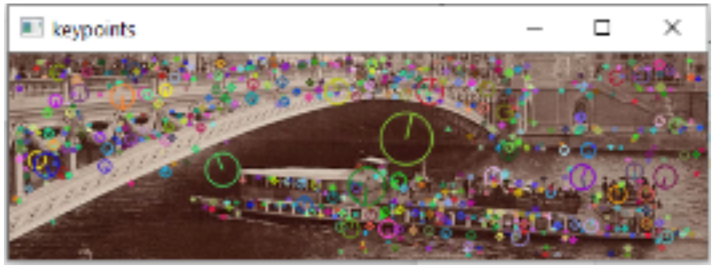
- align images in order to obtain panorama images

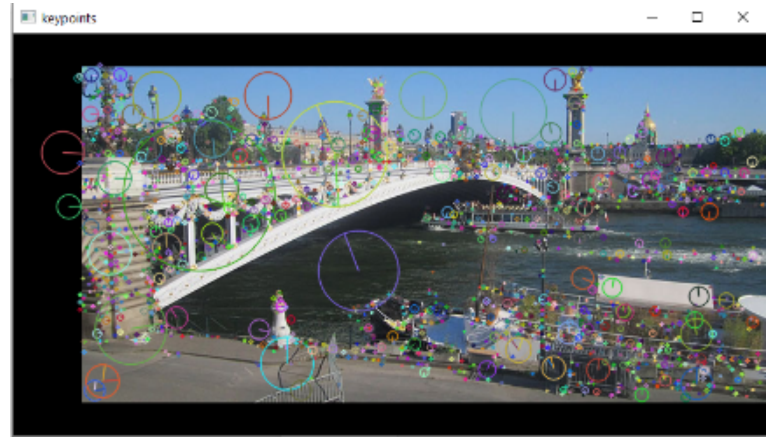
Panorama stitching – lab class 4

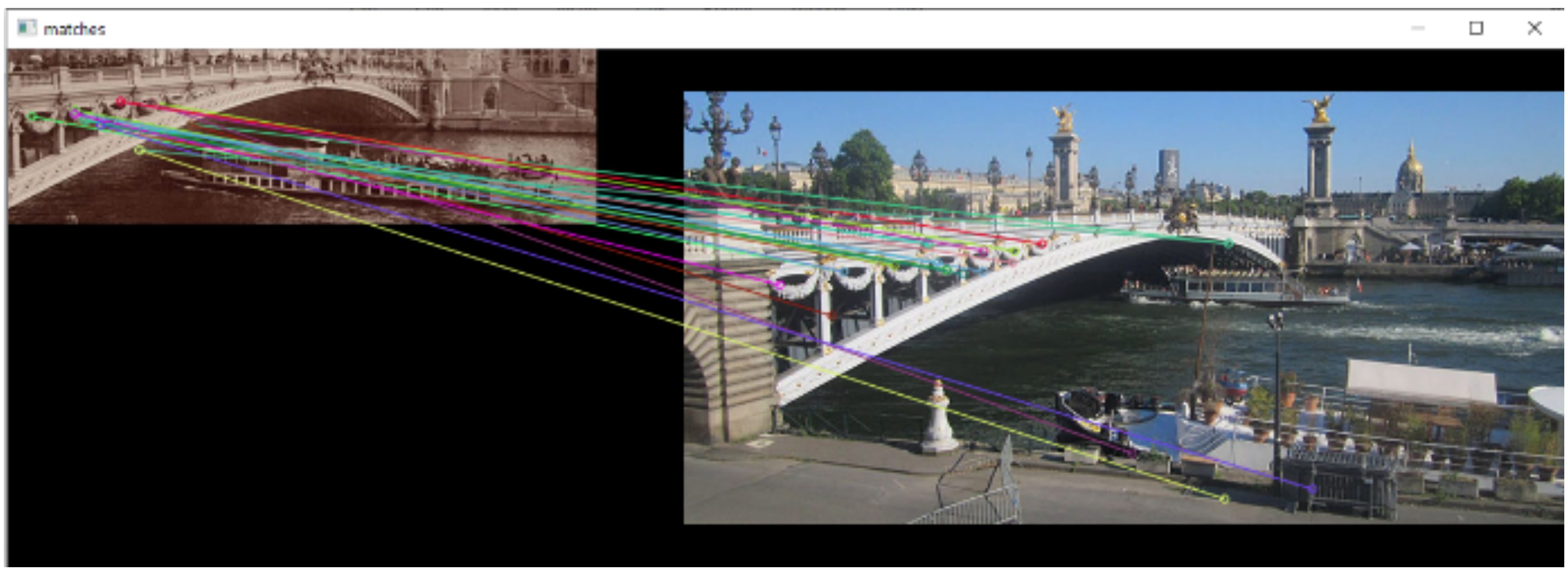
- align images in order to obtain panorama images



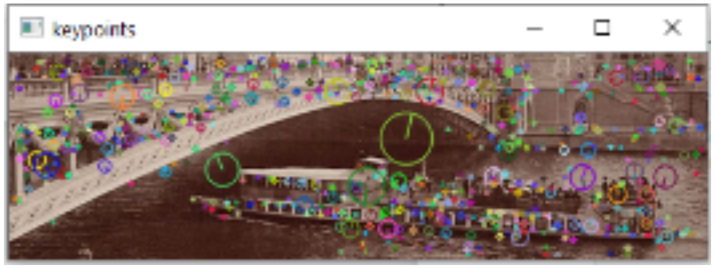
Then and now – lab class 4

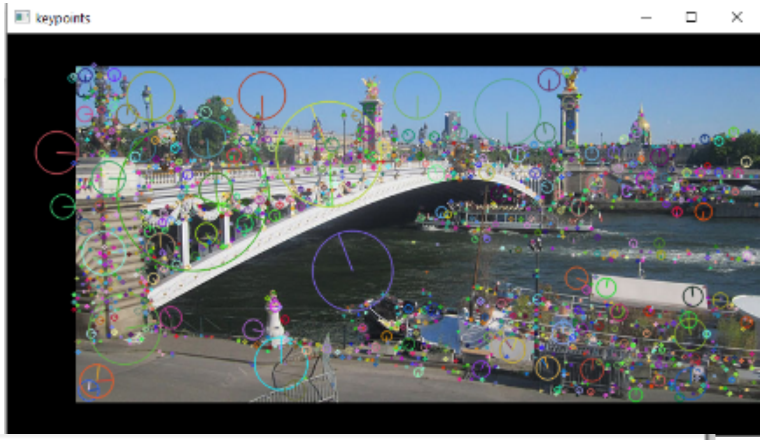


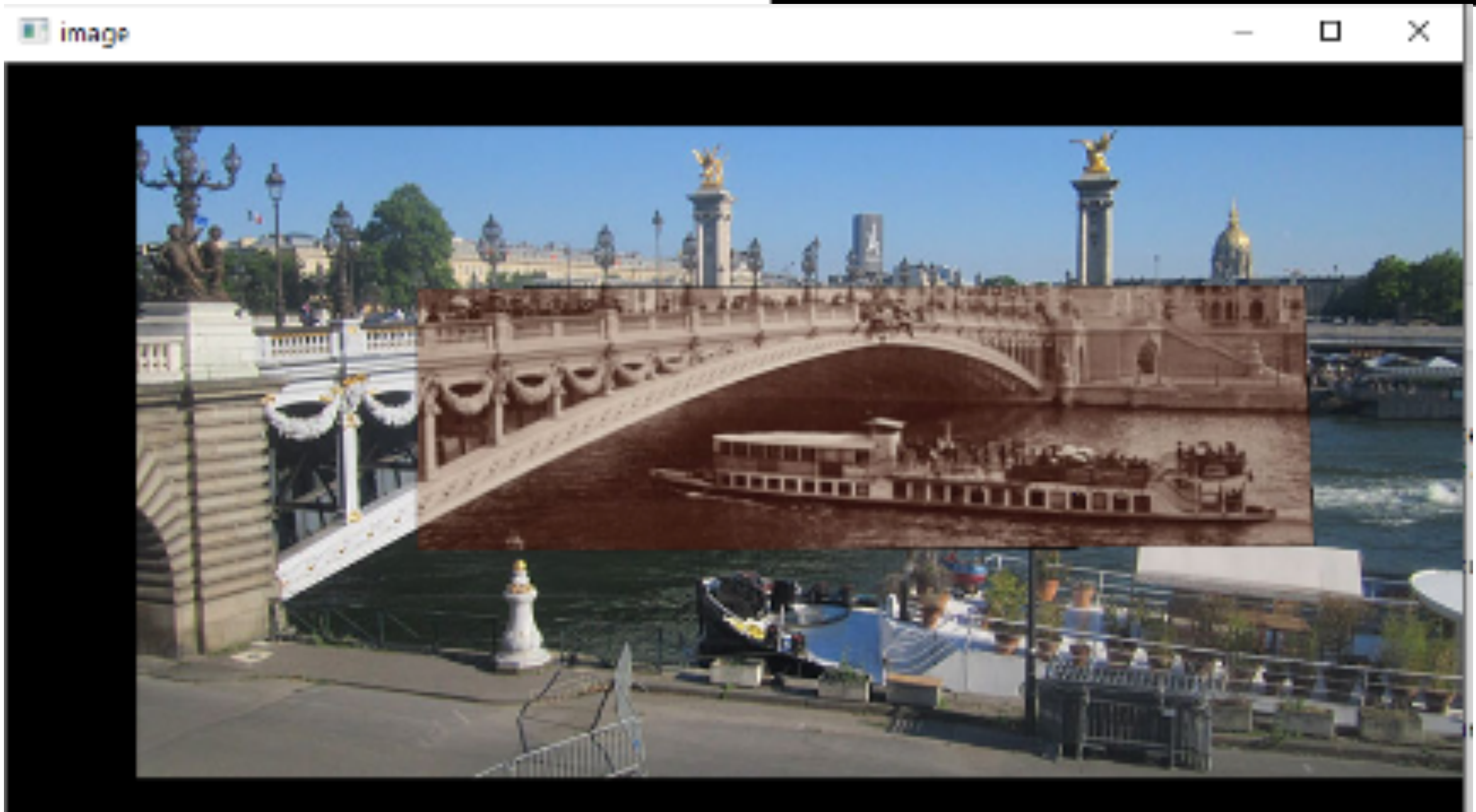




Then and now – lab class 4







Using SIFT in OpenCV

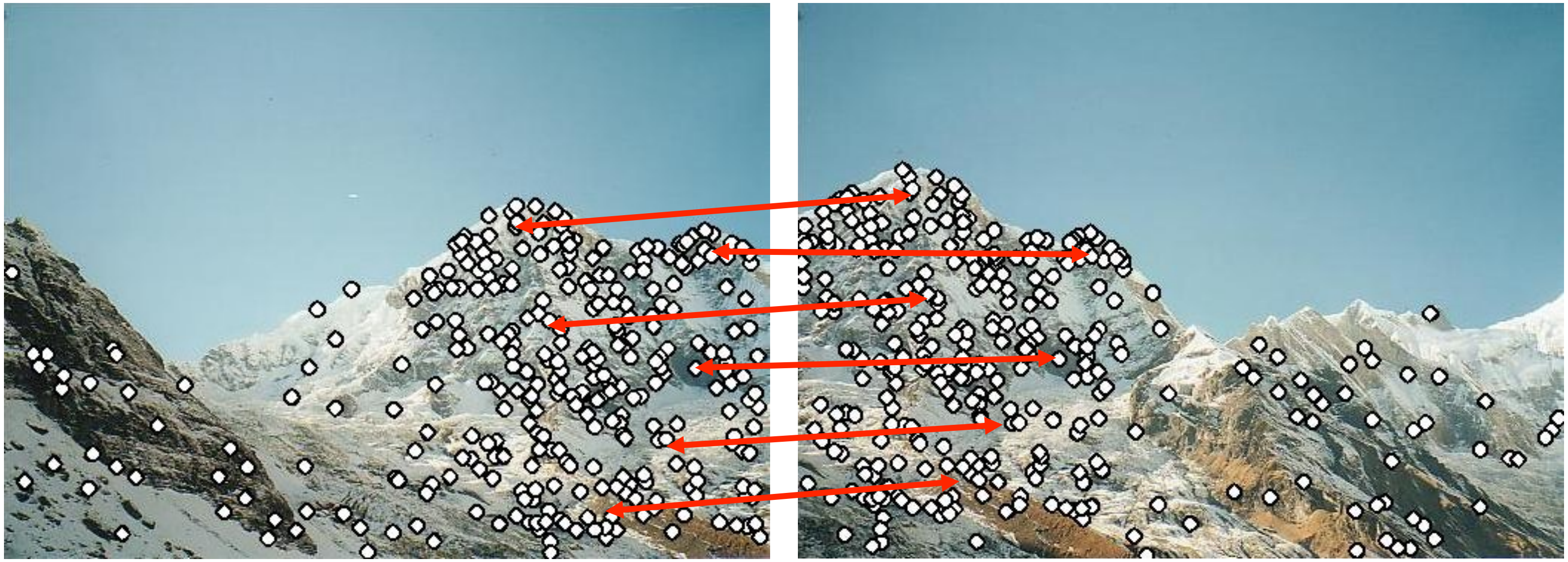
- (make a new environment)
- pip uninstall opencv-python
- pip install opencv-contrib-python

Why extract local features?

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

Why extract local features?

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



Step 1: extract local features

Step 2: match local features

Why extract local features?

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

Step 1: extract local features

Step 2: match local features

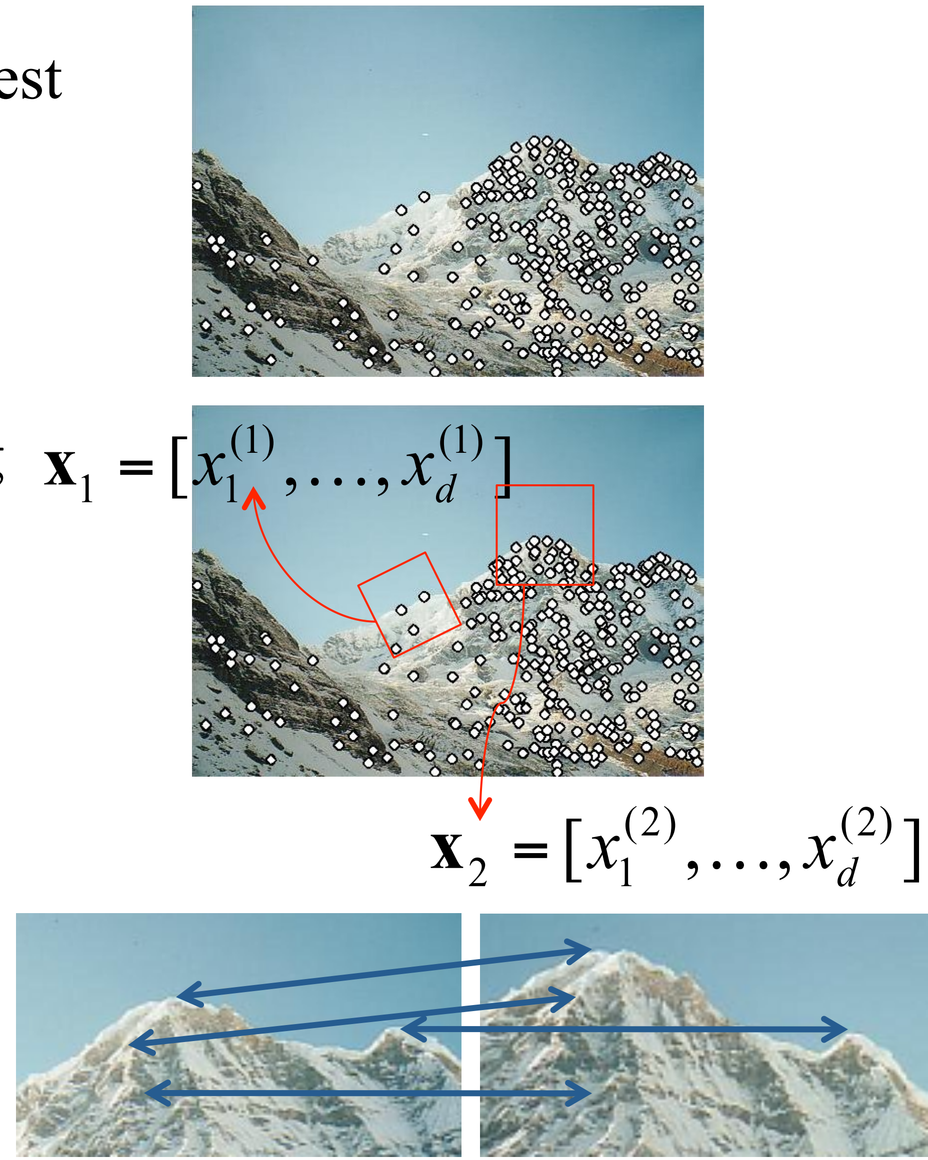
Step 3: align images

Local features: main components

1) **Detection:** Identify the interest points

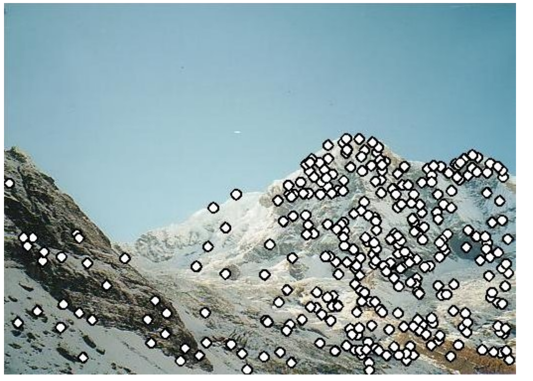
2) **Description:** Extract vector feature descriptor surrounding each interest point.

3) **Matching:** Determine correspondence between descriptors in two views



Local features: main components

1) Detection: Identify the interest points



2) Description: Extract vector feature descriptor surrounding each interest point.

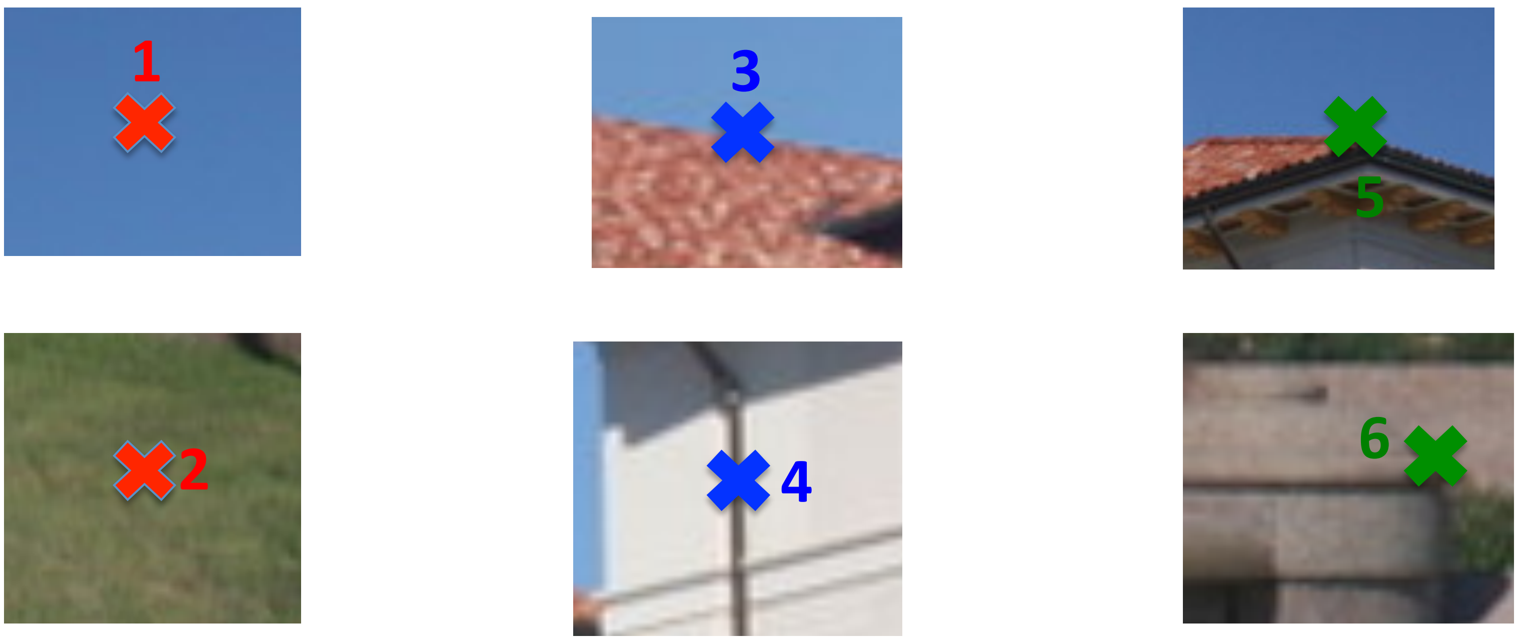
3) Matching: Determine correspondence between descriptors in two views

• What points would you choose?


1
2
3
4
5
6

Detecting local invariant features

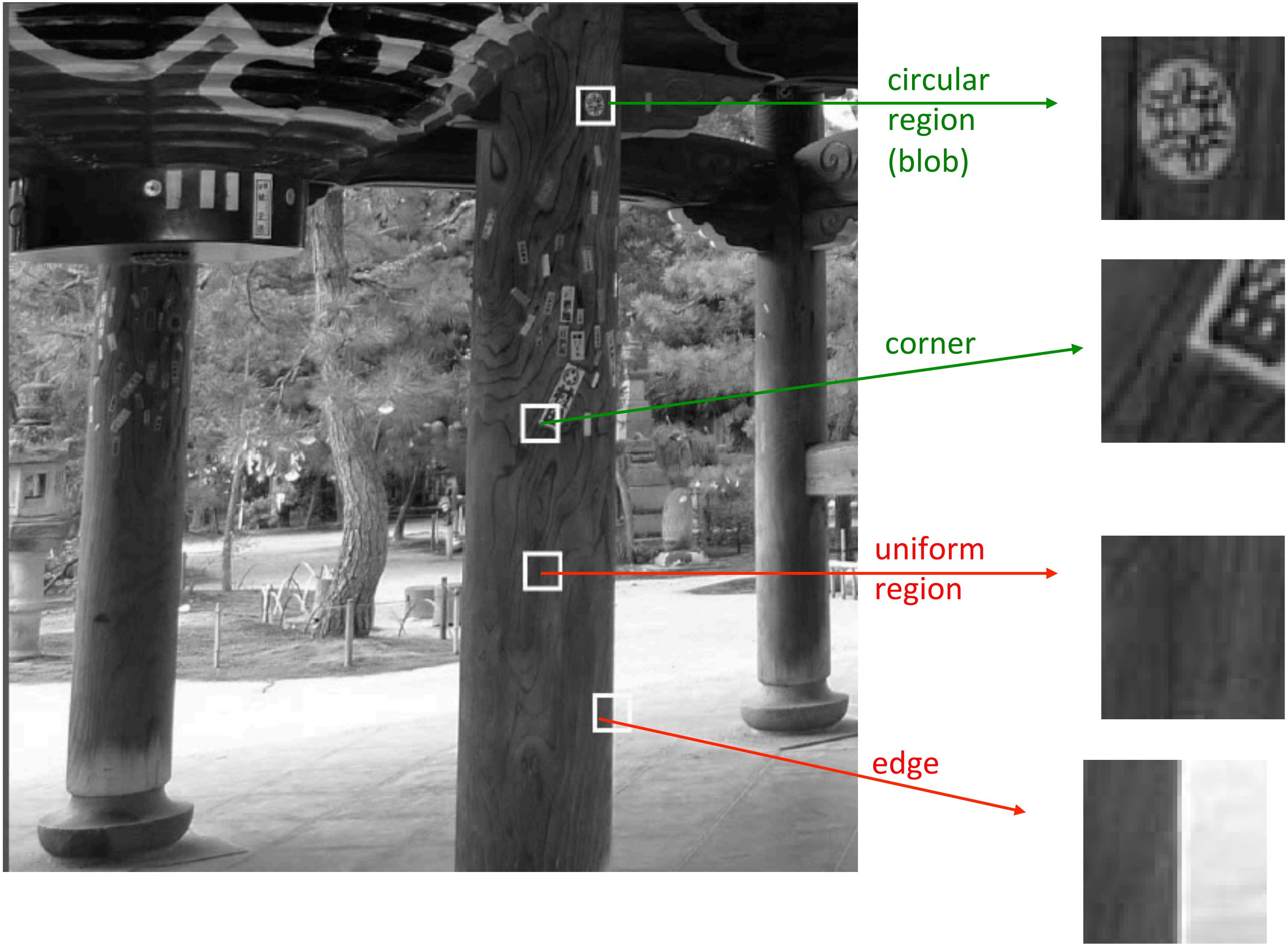
- Detection of interest points



Uniform regions
Ambiguous for
matching

Edges
Ambiguous for
matching

Corners
Good pentru
matching

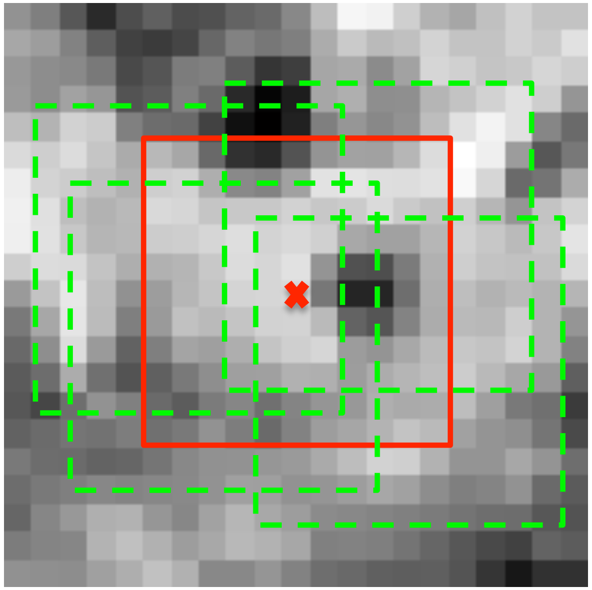

circular
region
(blob)
corner
uniform
region
edge

Detecting local invariant features

- Detection of interest points
  - Harris corner detection
  - Scale invariant blob detection: LoG
- Description of local patches

Detecting local invariant features

- How do we mathematically formulate what is a good local feature?

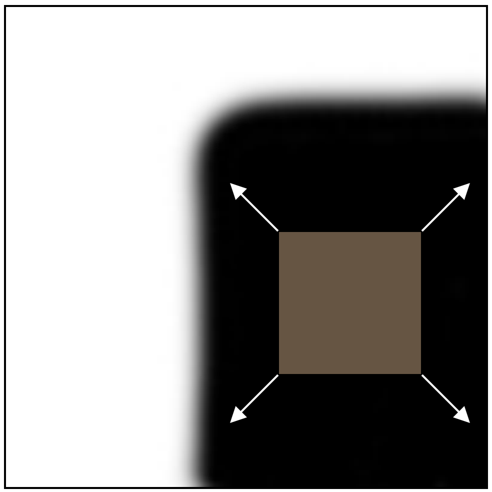


- Characterize the point by a small window centered in it

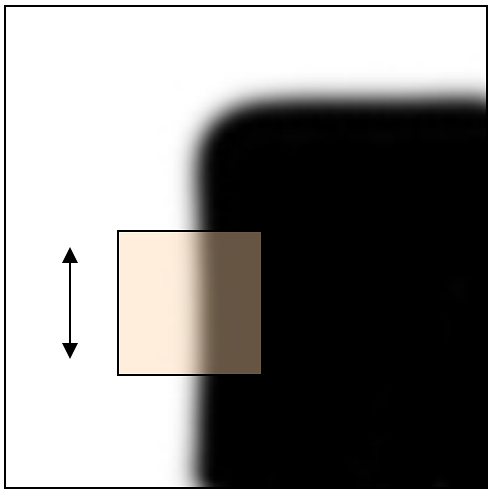
- Good local feature: shifting a window in *any direction* should give *a large change* in intensity (sum of squared distances)

Corner detection: Basic idea

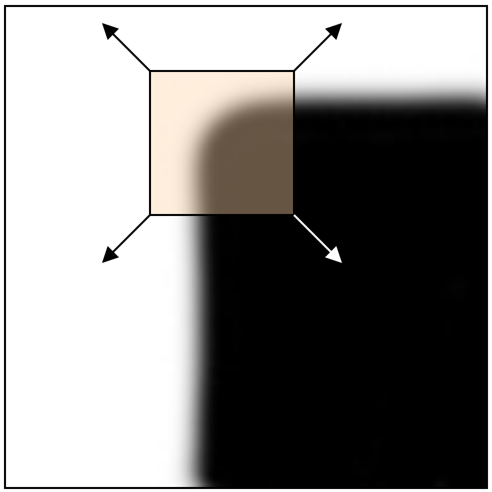
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction



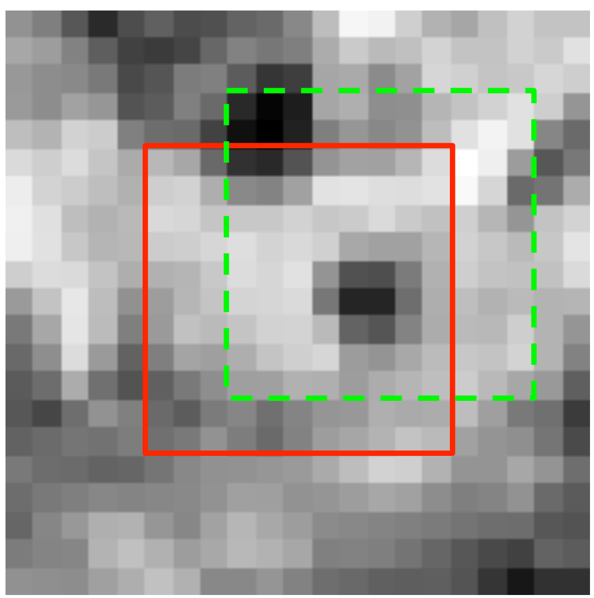
"corner":
significant change in all directions

Corner Detection: Derivation

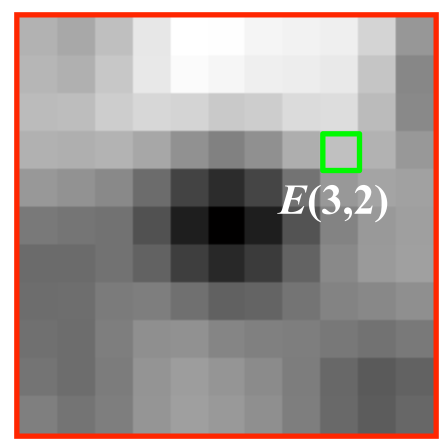
Change in appearance of window W for the shift $[u,v]$:

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

$I(x, y)$



$E(u, v)$

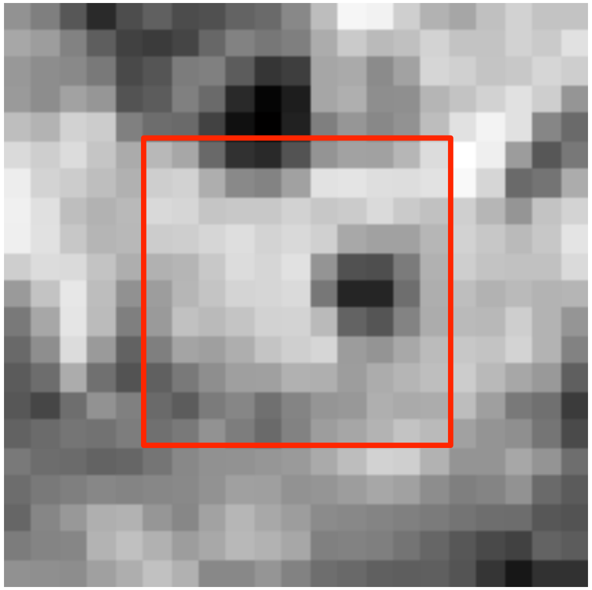


Corner Detection: Derivation

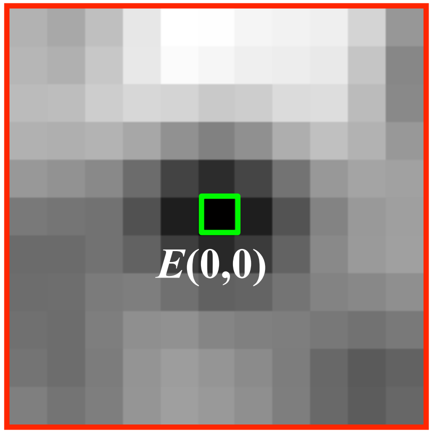
Change in appearance of window W for the shift $[u,v]$:

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

$I(x, y)$



$E(u, v)$



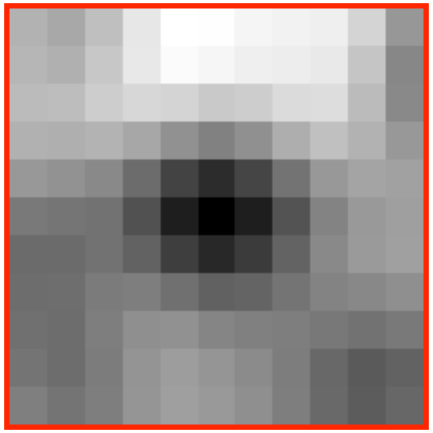
Corner Detection: Derivation

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

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

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

$E(u, v)$



Corner Detection: Derivation

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

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

$$I(x+u, y+v) \approx I(x,y) + u \cdot \frac{\partial I}{\partial x}(x,y) + v \cdot \frac{\partial I}{\partial y}(x,y)$$

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

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

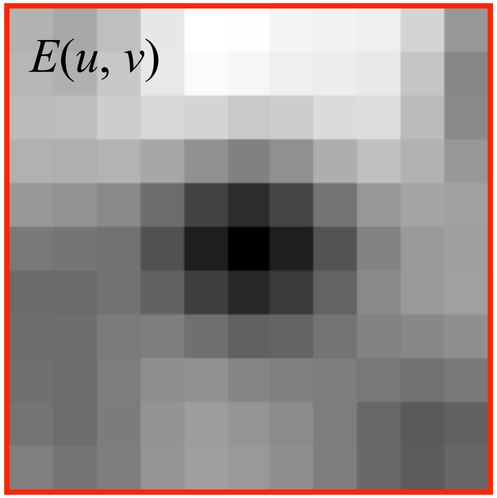
$$E(u,v) = \sum_{(x,y)\in W} \left(I(x+u, y+v) - I(x,y)\right)^2$$

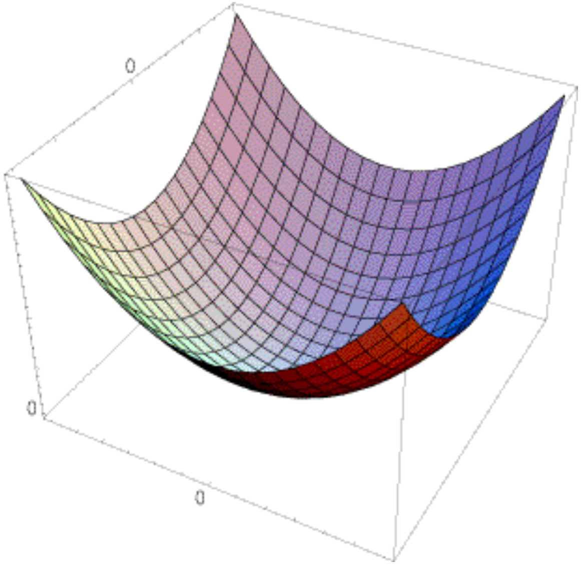
$$E(u,v) \approx \sum_{(x,y)\in W} (u \cdot I_x + v \cdot I_y)^2 = \sum_{(x,y)\in W} u^2 \cdot I_x^2 + 2uv \cdot I_x \cdot I_y + v^2 \cdot I_y^2$$

Corner Detection: Derivation

•$E(u,v)$ can be locally approximated by a quadratic surface:

$$E(u,v) \approx u^2 \sum_{x,y} I_x^2 + 2uv \sum_{x,y} I_x I_y + v^2 \sum_{x,y} I_y^2$$





•In which directions does this surface have the fastest/ slowest change?

Corner Detection: Derivation

- $E(u,v)$ can be locally approximated by a quadratic surface:

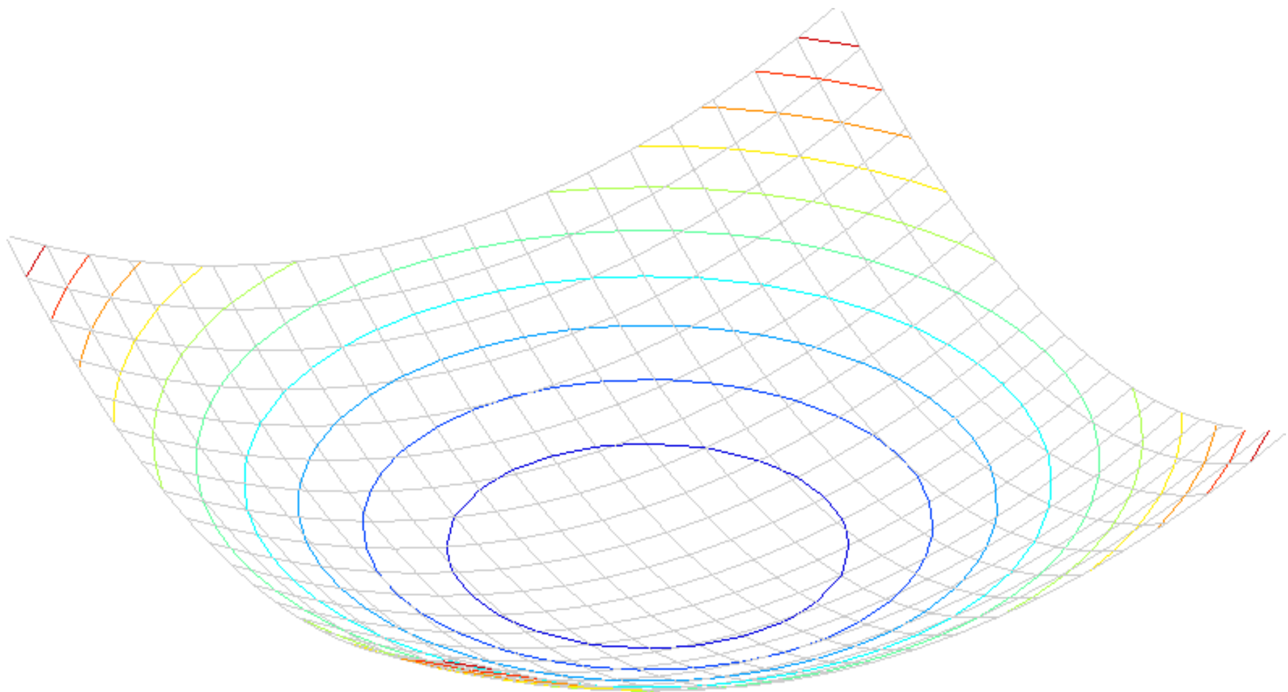
$$E(u,v) \approx u^2 \sum_{x,y} I_x^2 + 2uv \sum_{x,y} I_x I_y + v^2 \sum_{x,y} I_y^2$$

$$= \begin{bmatrix} u & v \end{bmatrix} \underbrace{\begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix}}_{\text{Second moment matrix } M} \begin{bmatrix} u \\ v \end{bmatrix}$$

Interpreting the second moment matrix

A horizontal "slice" of $E(u, v)$ is given by the equation of an ellipse:

$$[u \;\; v] \;\; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

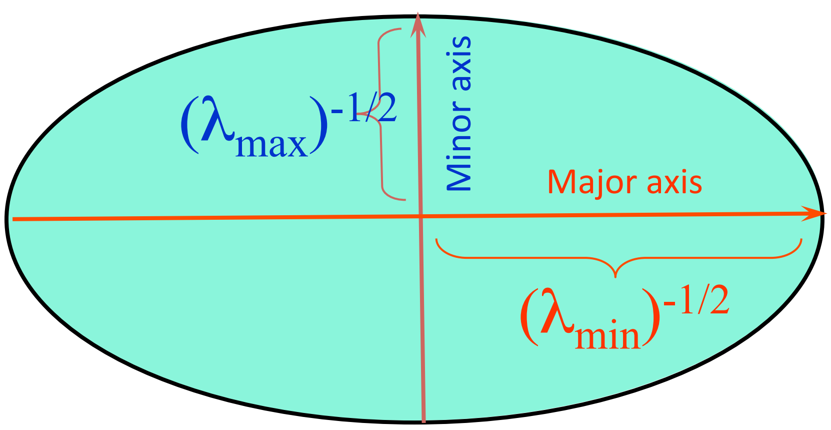


Interpreting the second moment matrix

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

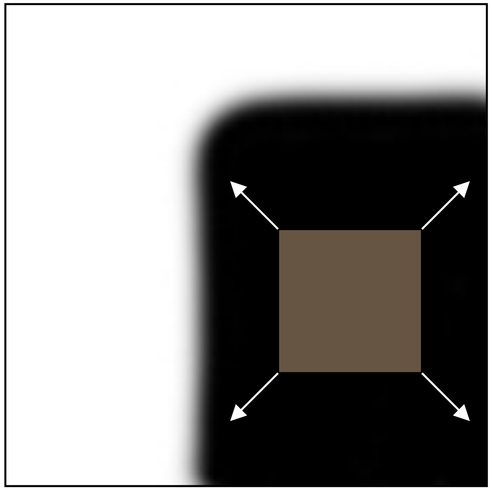
$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix} =$$

$$[u \;\; v] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 1$$



Interpreting the second moment matrix

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



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

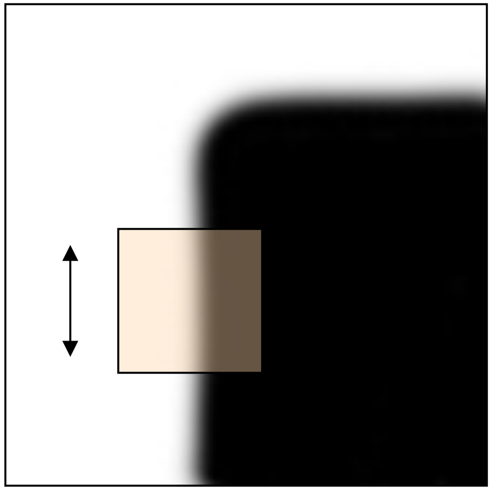
"flat" region:
no change in all directions

$$M \approx \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The eigenvalues of M = solutions of the equation $\det(M - \lambda I_2) = 0$ are $\lambda_1 \approx \lambda_2 \approx 0$

Interpreting the second moment matrix

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



"edge":
no change along the edge direction

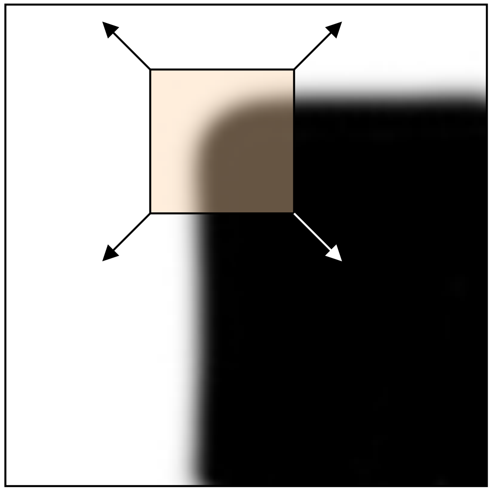
$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

$$M \approx \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix}$$

The eigenvalues of M = solutions of the equation $\det(M - \lambda I_2) = 0$ are $\lambda_1 >> 0$, $\lambda_2 \approx 0$

Interpreting the second moment matrix

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



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

"corner":
significant change in all directions

$$M \approx \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

The eigenvalues of M = solutions of the equation $\det(M - \lambda I_2) = 0$ are $\lambda_1, \lambda_2 >> 0$

Interpreting the second moment matrix

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

$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

This means dominant gradient directions align with x or y axis

Look for locations where **both** λ's are large.

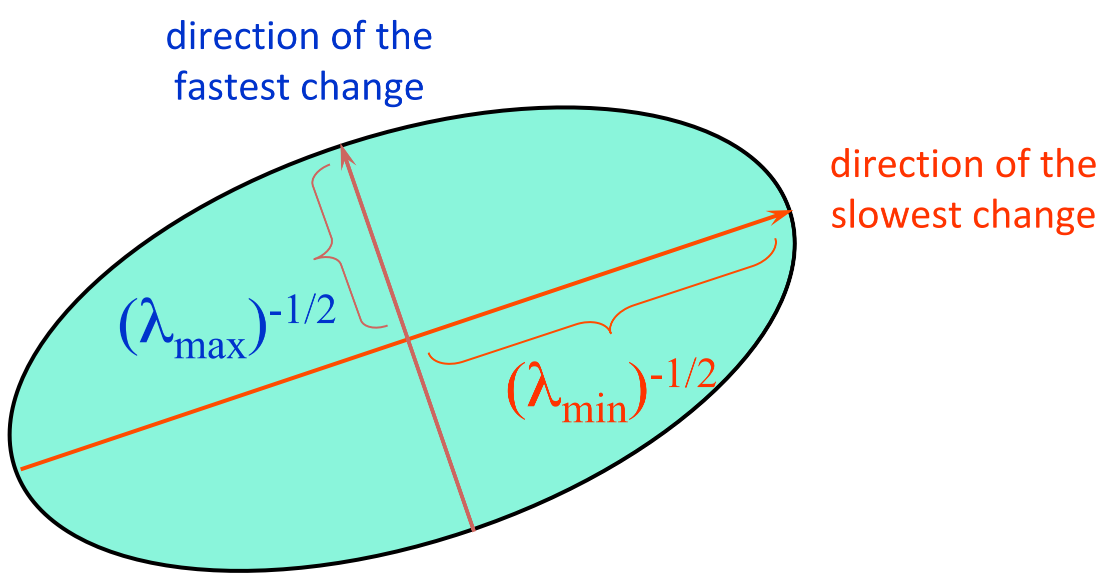
If either λ_1 or λ_2 is close to 0, then this is **not** corner-like.

Interpreting the second moment matrix

In the general case, need to *diagonalize* 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

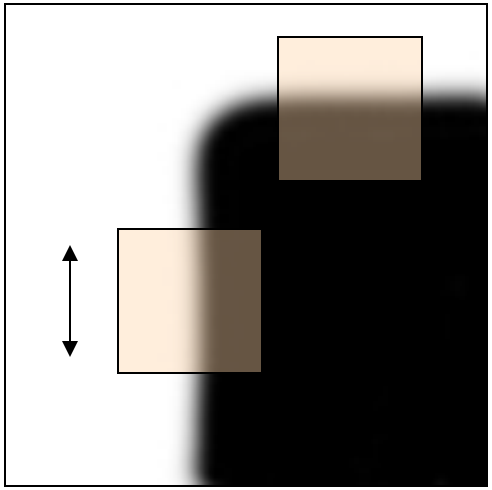
In the general case, need to *diagonalize* M:

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

$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of *M* reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.

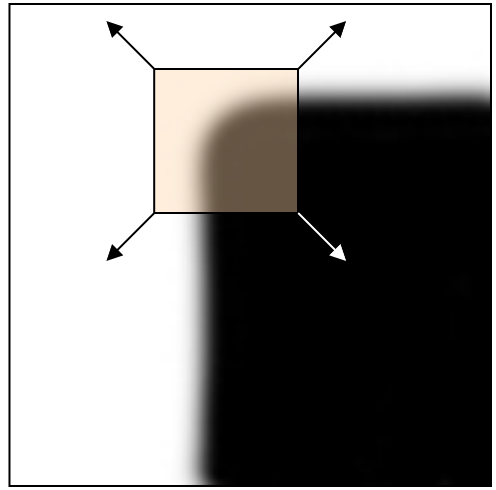
Corner response function



"edge":

$\lambda_1 >> \lambda_2$

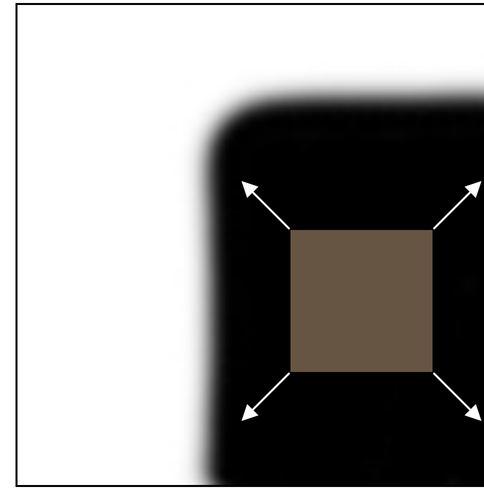
$\lambda_2 >> \lambda_1$



"corner":

λ_1 and λ_2 are large,

$\lambda_1 \sim \lambda_2$;



"flat" region

λ_1 and λ_2 are small

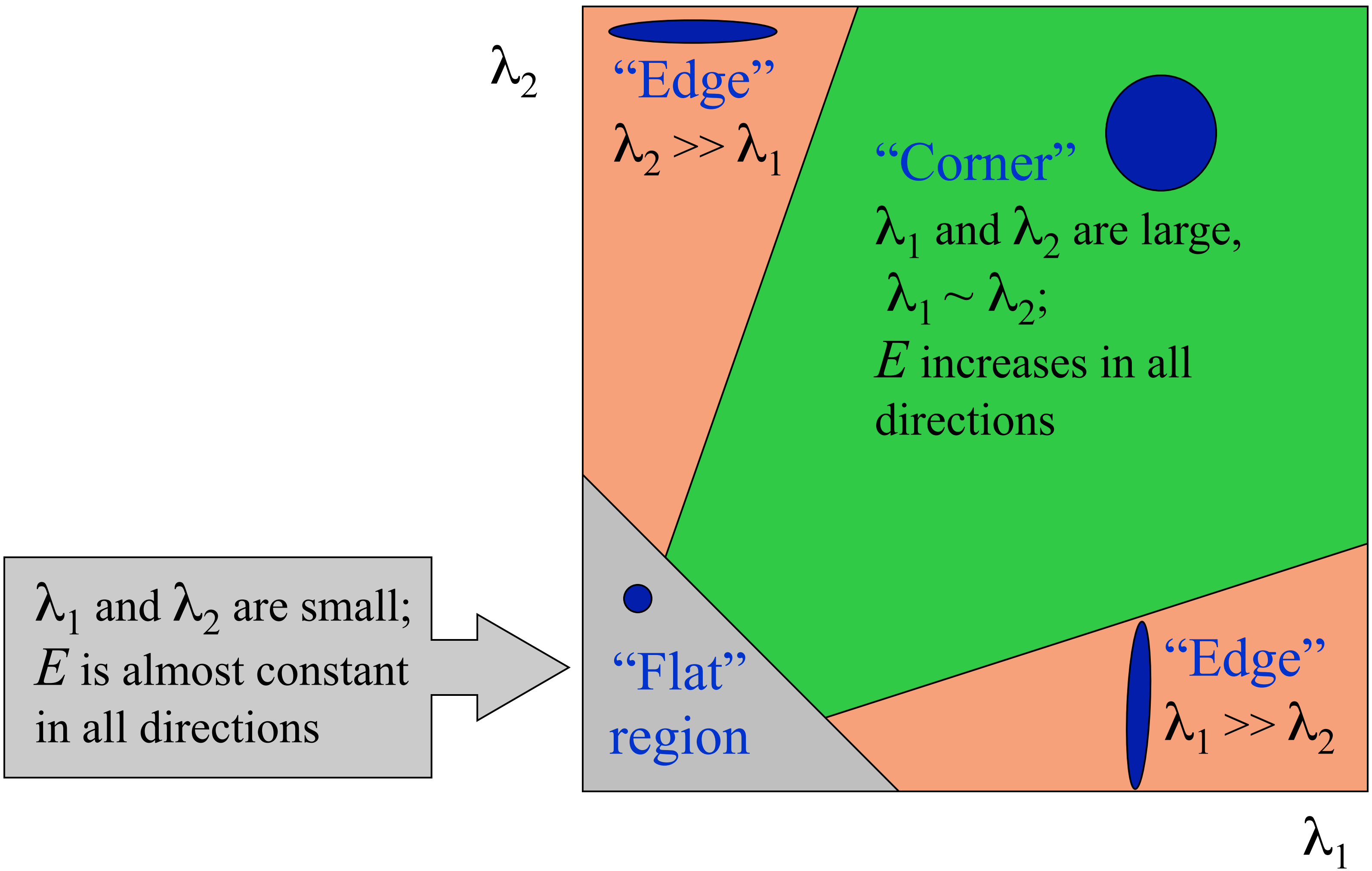
Cornerness score
(other variants possible)

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$f = det(M) - \alpha \cdot trace(M)^2 = \lambda_1 \cdot \lambda_2 - \alpha * (\lambda_1 + \lambda_2)^2$$

Interpreting the eigenvalues

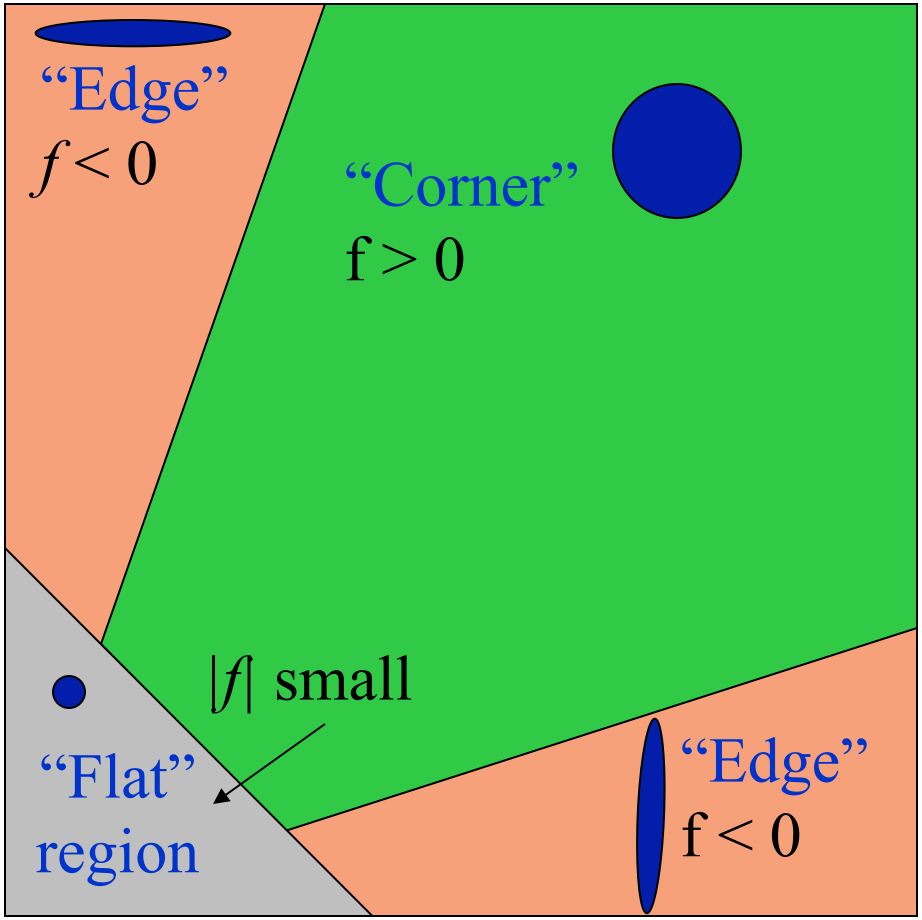
Classification of image points using eigenvalues of *M*:



Corner response function

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

α: constant (0.04 to 0.06)



Harris corner detector

1) Compute M matrix for each image window to get their *cornerness* scores.

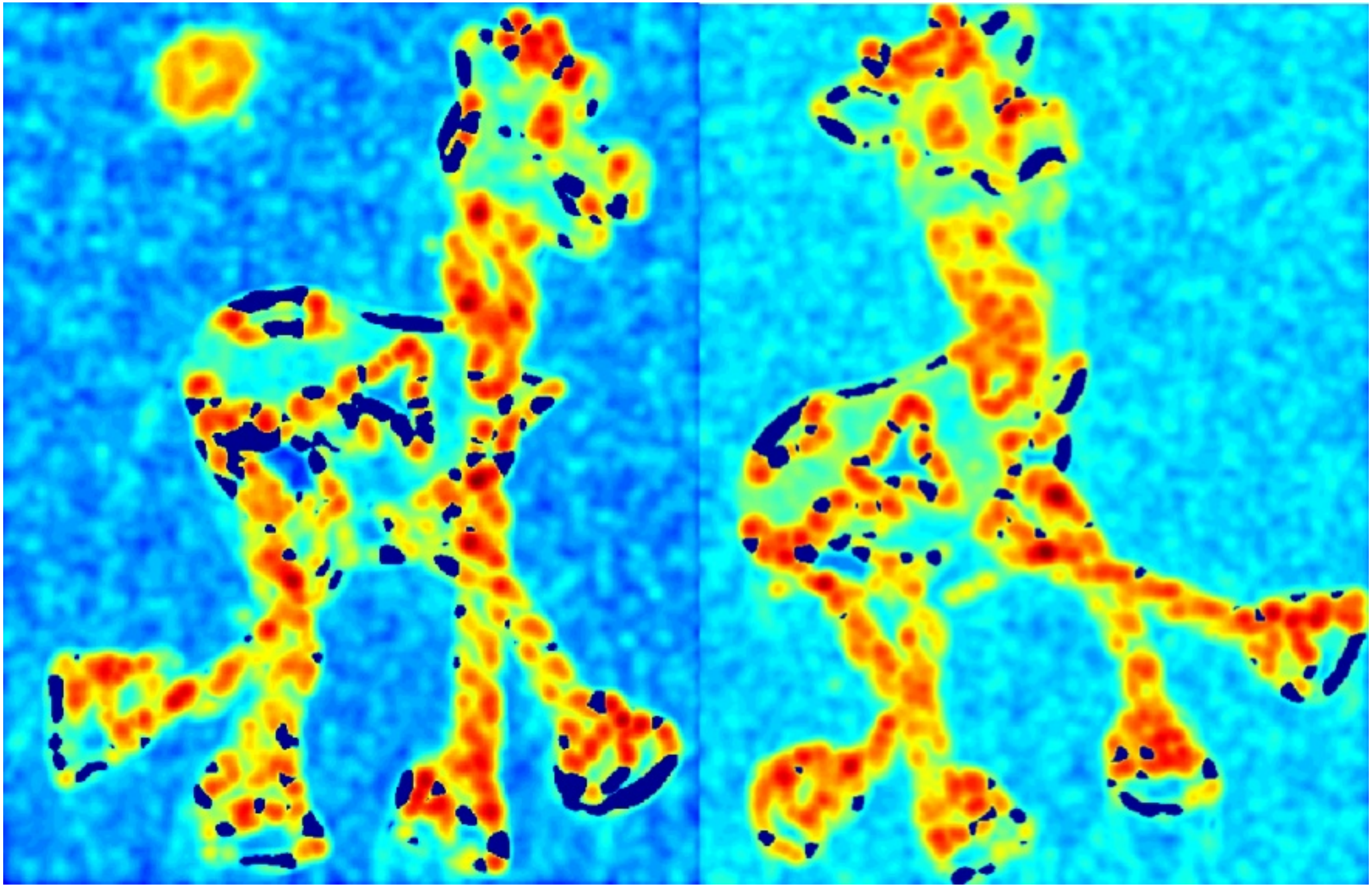
2) Find points whose surrounding window give large corner response (f > threshold)

3) Take the points of local maxima, i.e., perform non-maximum suppression

Harris Detector: Steps

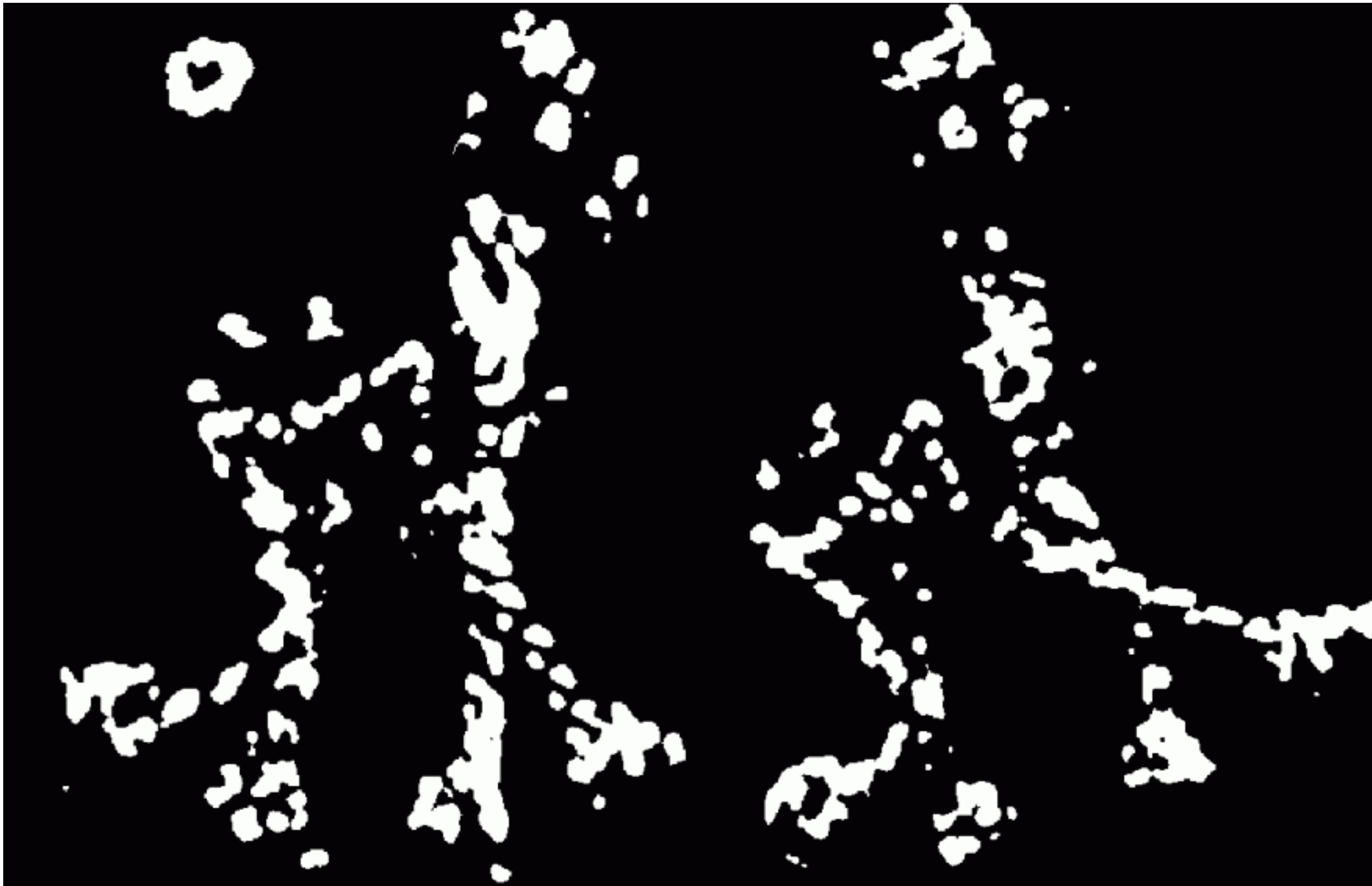
Harris Detector: Steps

Compute corner response f



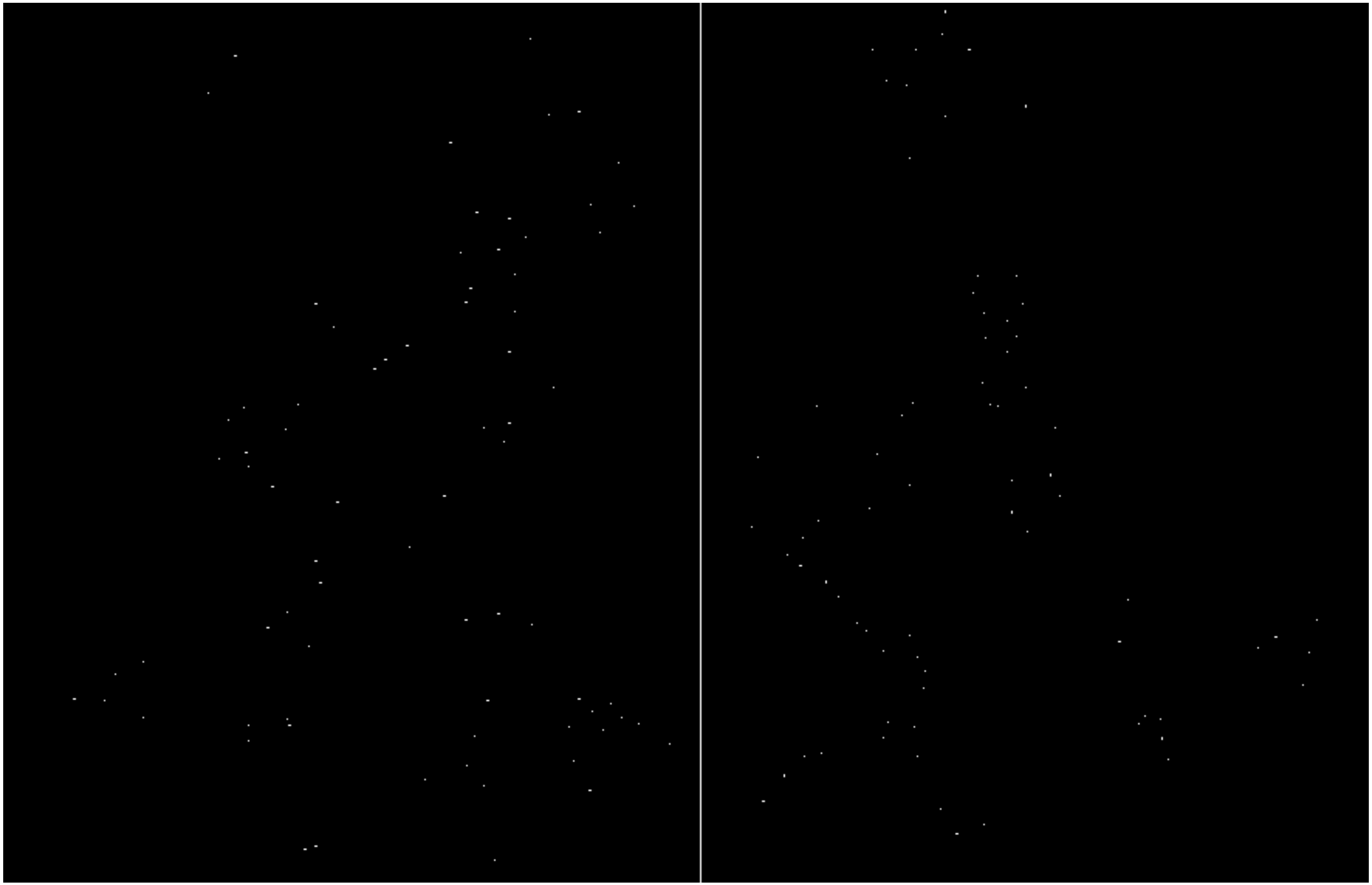
Harris Detector: Steps

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

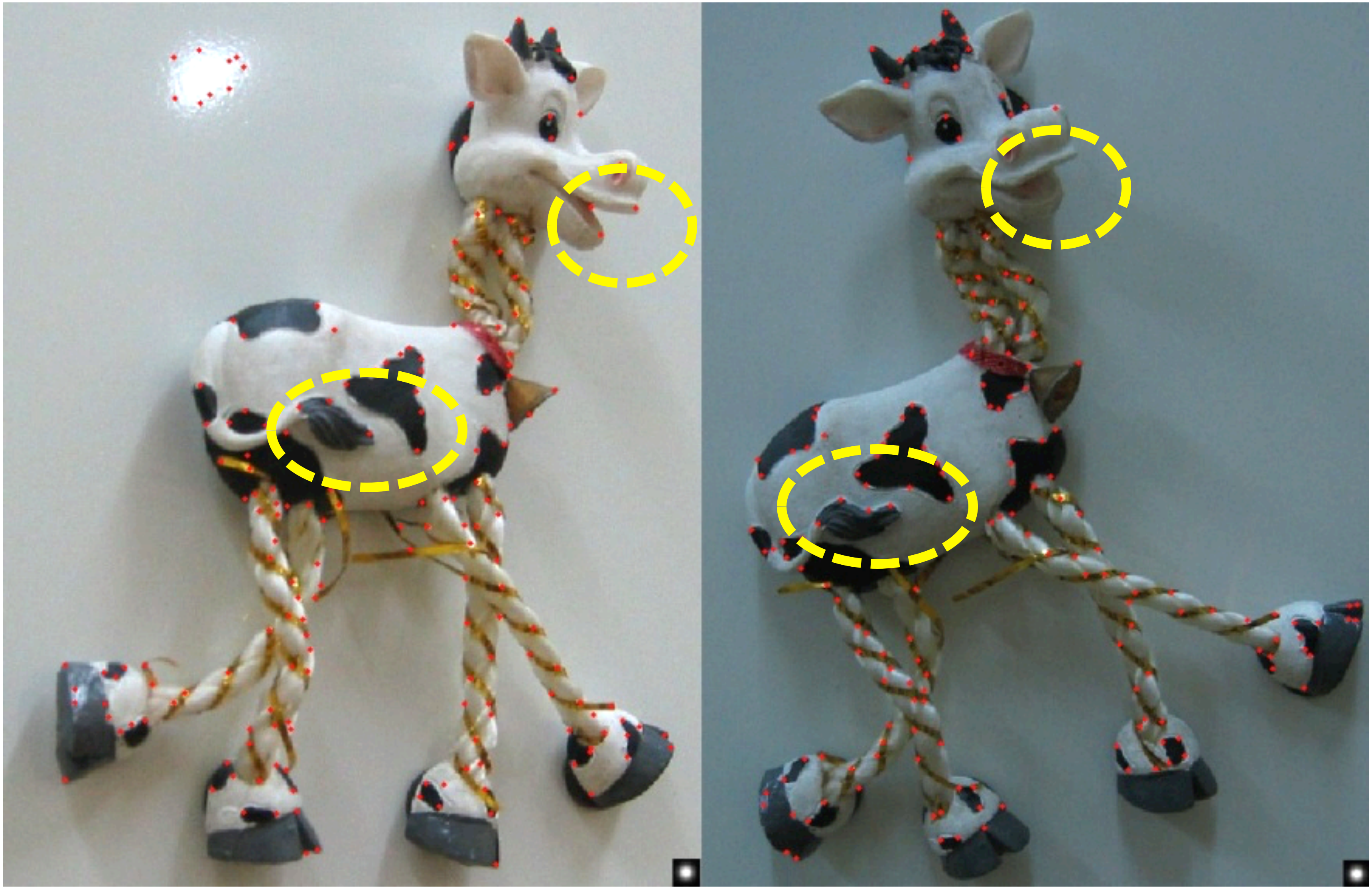


Harris Detector: Steps

Take only the points of local maxima of f



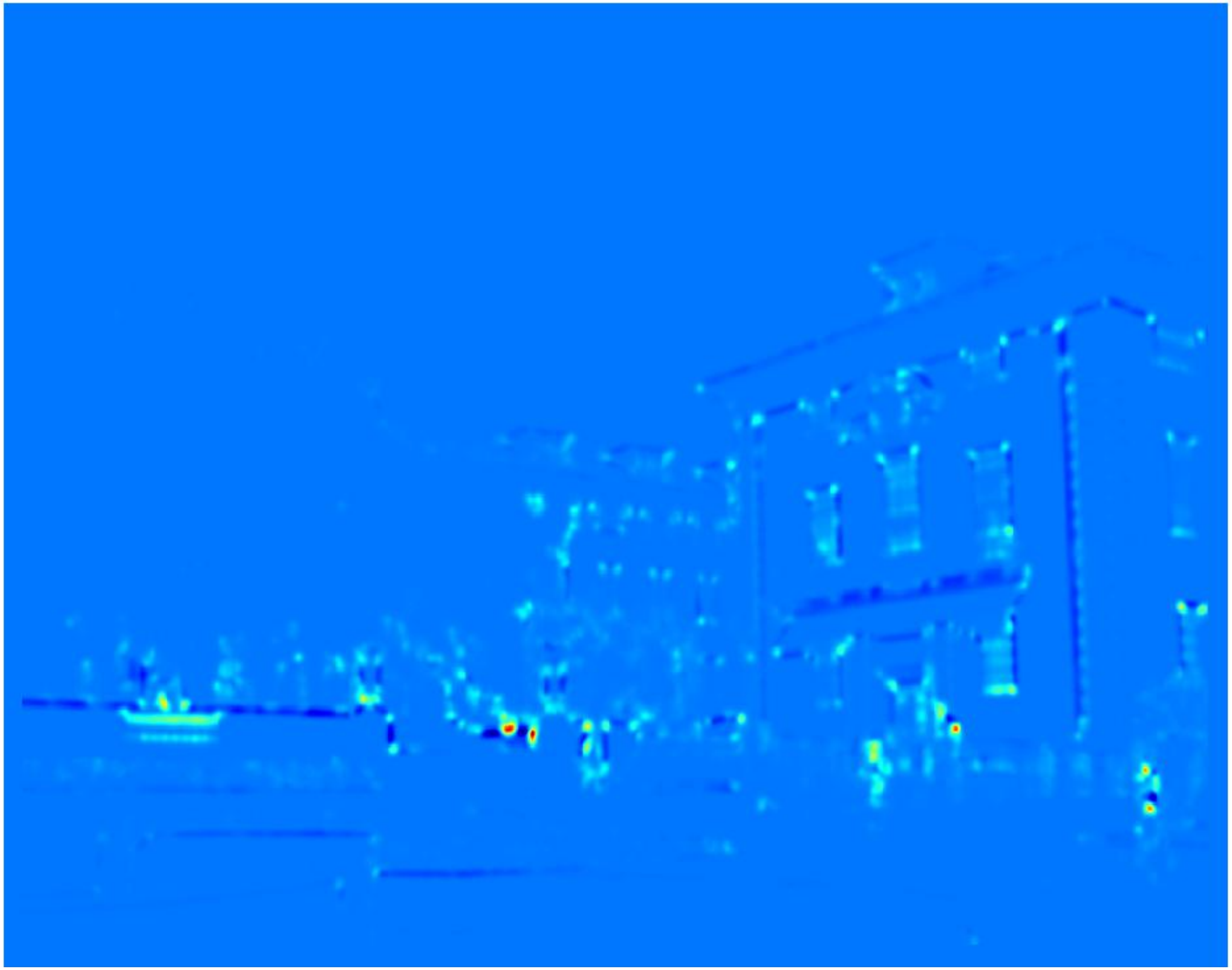
Harris Detector: Steps



Harris Detector: Steps

Harris Detector: Steps

Compute corner response f



Harris Detector: Steps

Take only the points of local maxima of f

Robustness of corner features

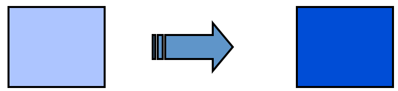
- What happens to corner features when the image undergoes geometric or photometric transformations?

Invariance and covariance

We want to find corners that are:

- ***invariant*** to ***photometric transformations***
- ***covariant*** to ***geometric transformation***

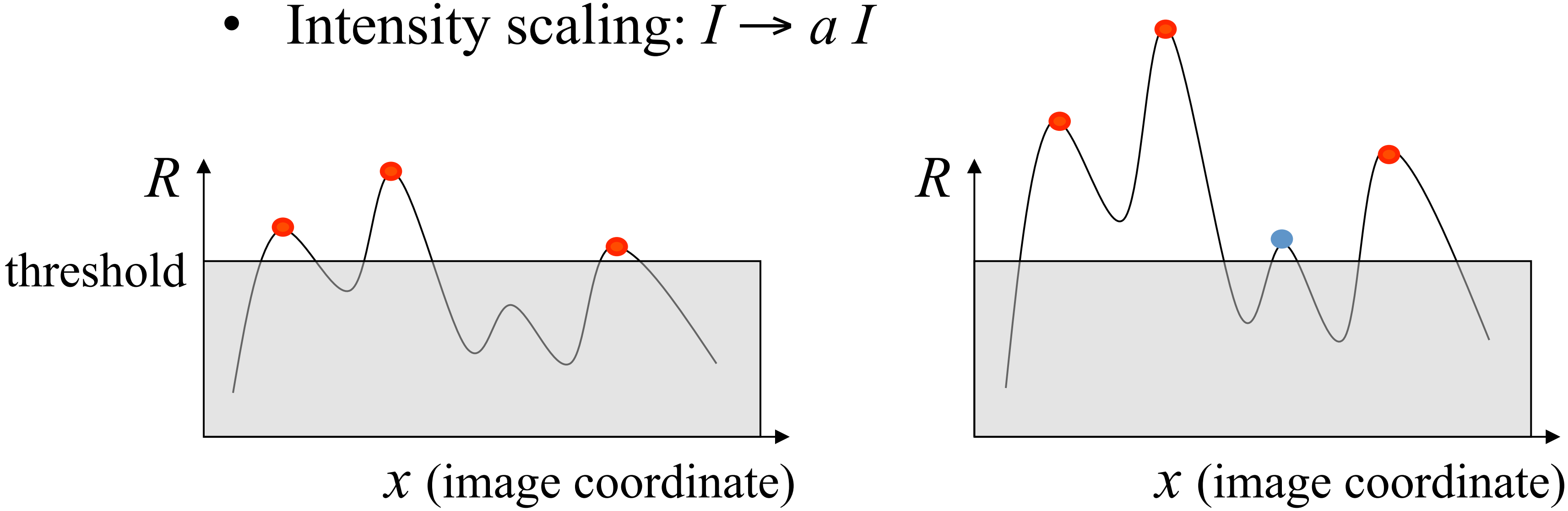
Photometric transformations: affine intensity changes



$I \rightarrow a\,I + b$

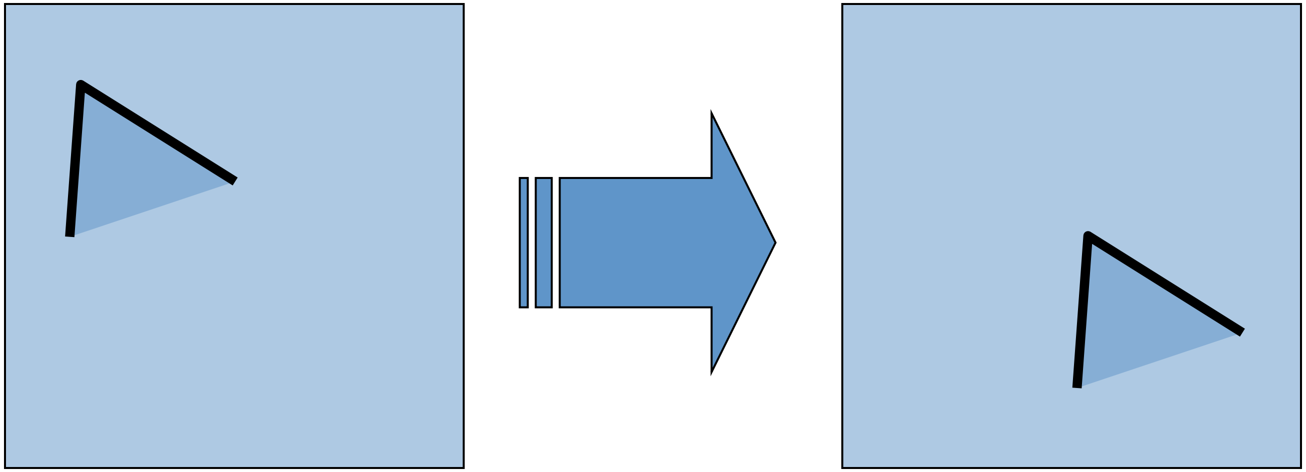
- Only derivatives are used, so invariant to intensity shift $I \rightarrow I + b$

- Intensity scaling: $I \rightarrow a\,I$



Partially invariant to affine intensity change

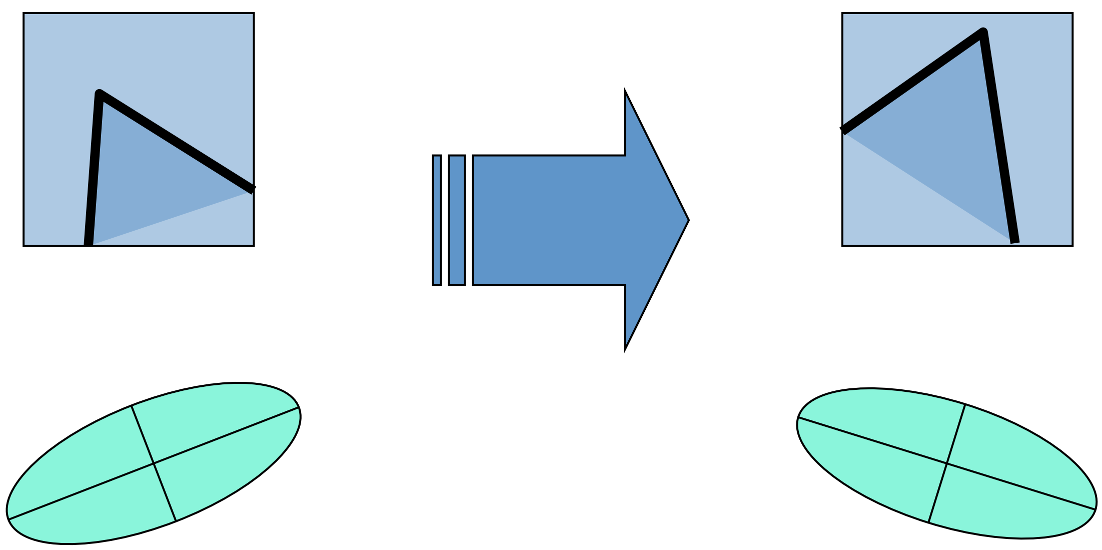
Geometric transformations: translations



- Derivatives and window function are shift-invariant

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

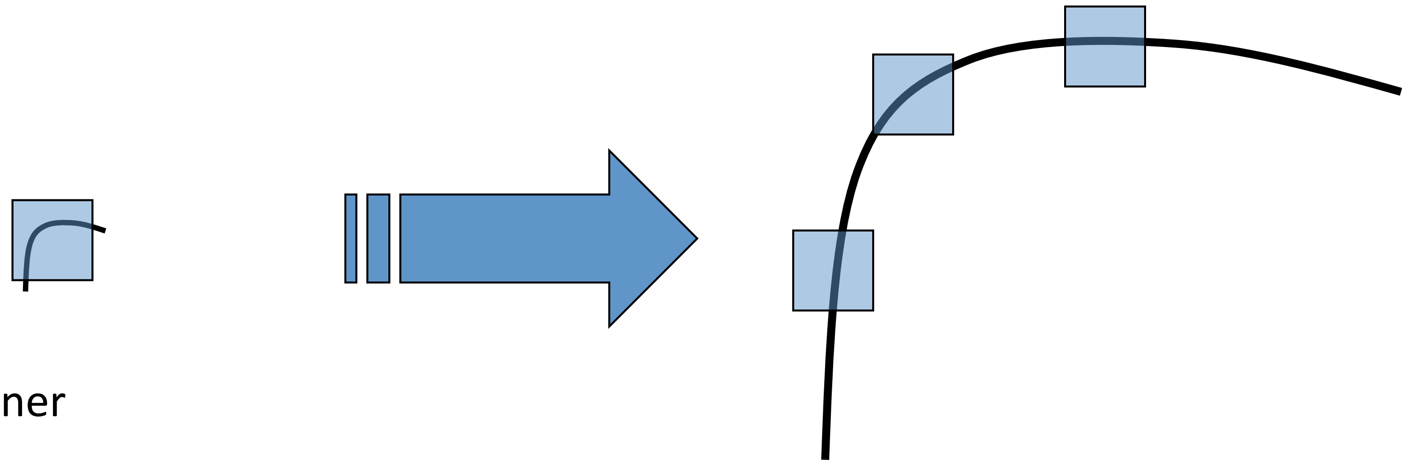
Geometric transformations: rotations



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

Corner location is covariant w.r.t. rotation

Geometric transformations: scaling



Corner

All points will be classified as edges

Corner location is not covariant w.r.t. scaling!

Robustness of corner features

- What happens to corner features when the image undergoes geometric or photometric transformations?
- We want to find corners that are:
  - ***invariant*** to ***photometric transformations***
  - ***covariant*** to ***geometric transformation***

Properties of Harris corner detector

Partially invariant to affine intensity change

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

Corner location is covariant w.r.t. rotation

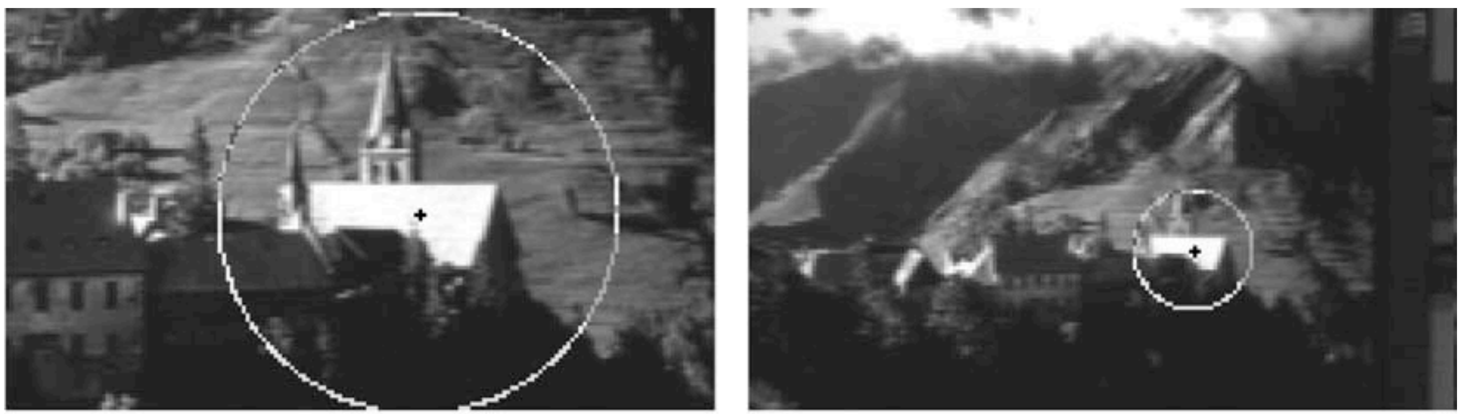
Corner location is not covariant w.r.t. scaling!

Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

Keypoint detection with scale selection

- We want to extract keypoints with *characteristic scales* that are *covariant* w.r.t. the image transformation



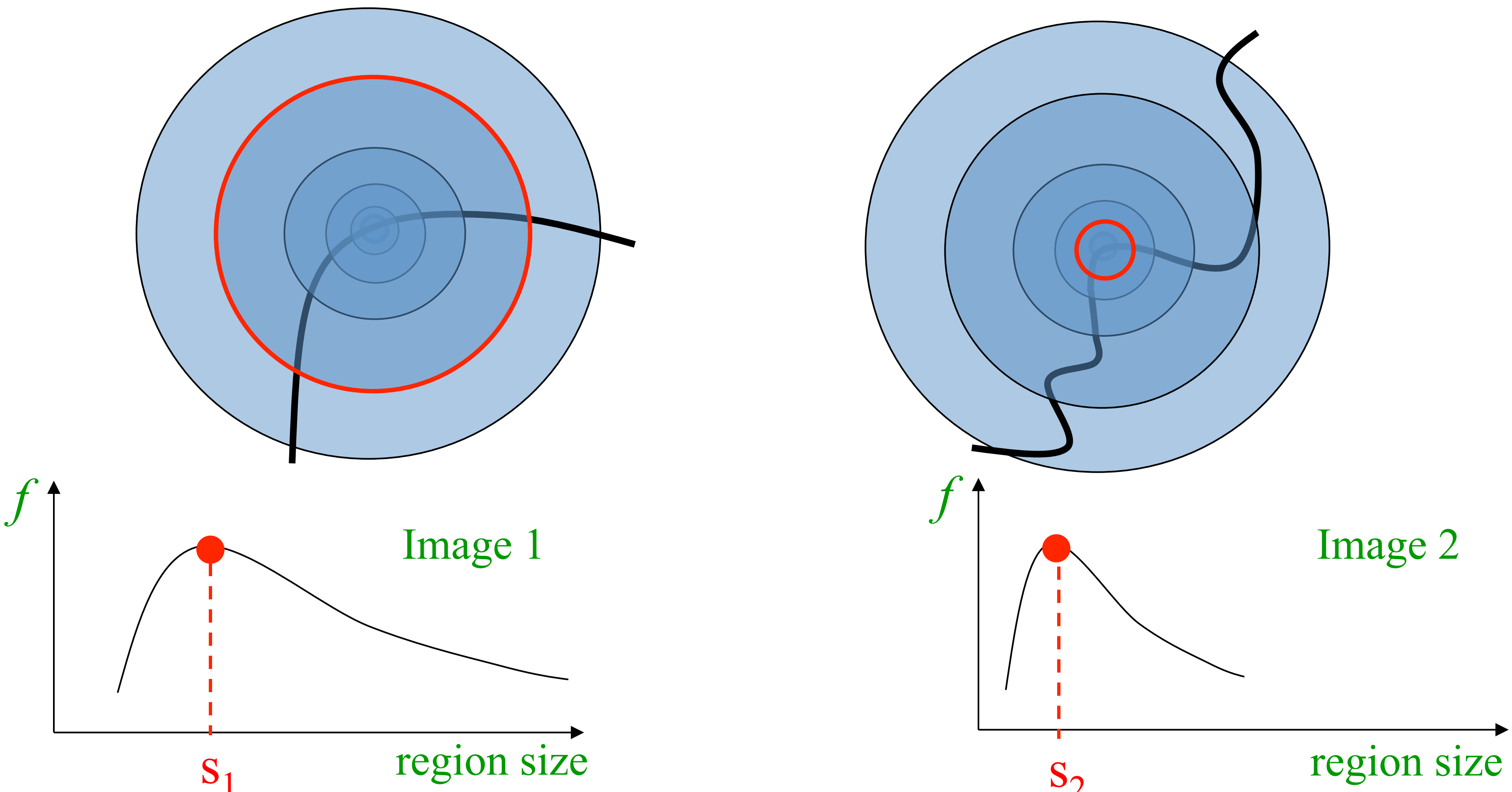
Automatic Scale Selection

How to find corresponding patch sizes,
with only one image in hand?

Automatic Scale Selection

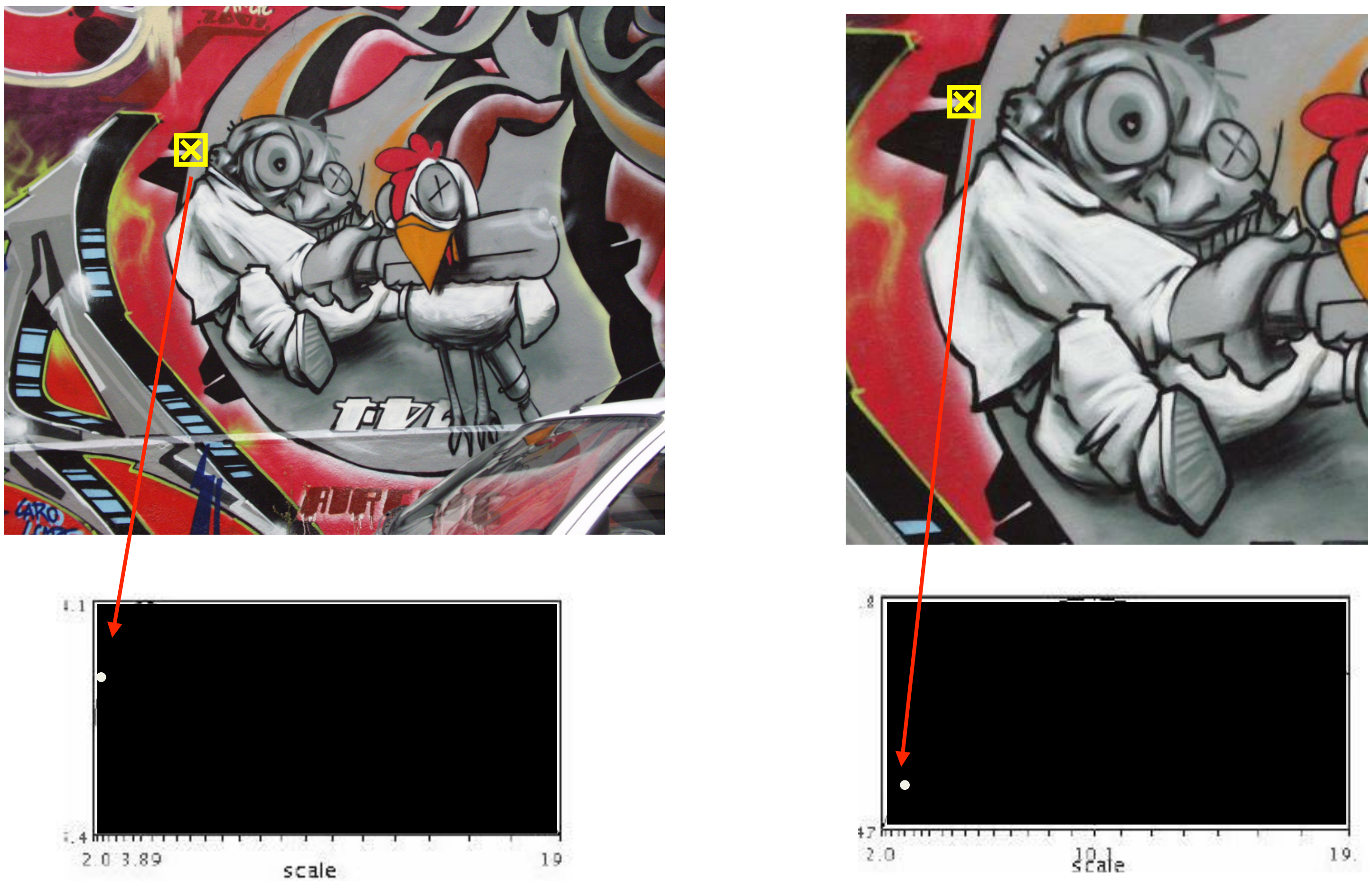
Intuition:

- Find scale that gives local maxima of some function *f* in both position and scale.



Automatic Scale Selection

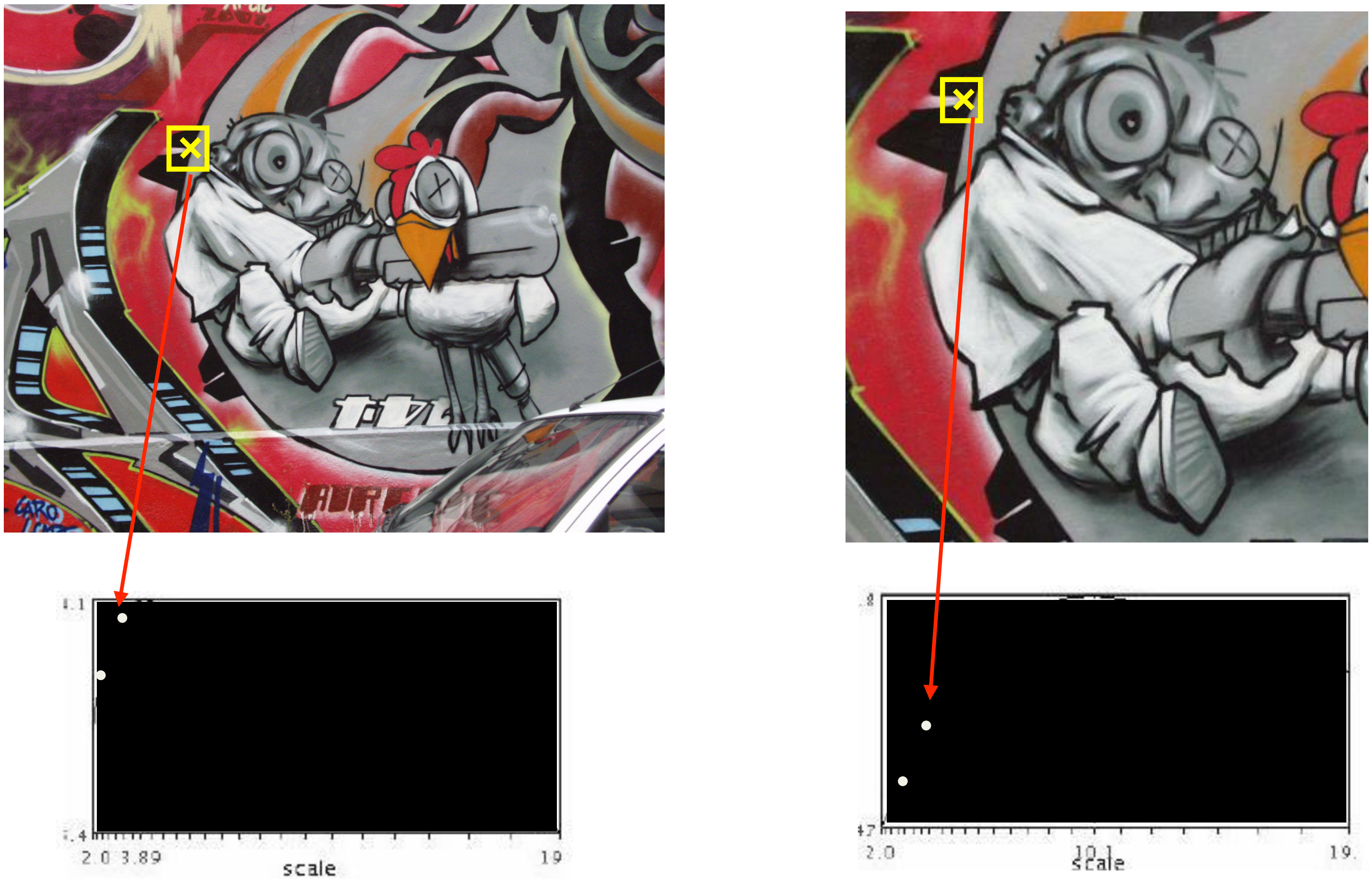
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

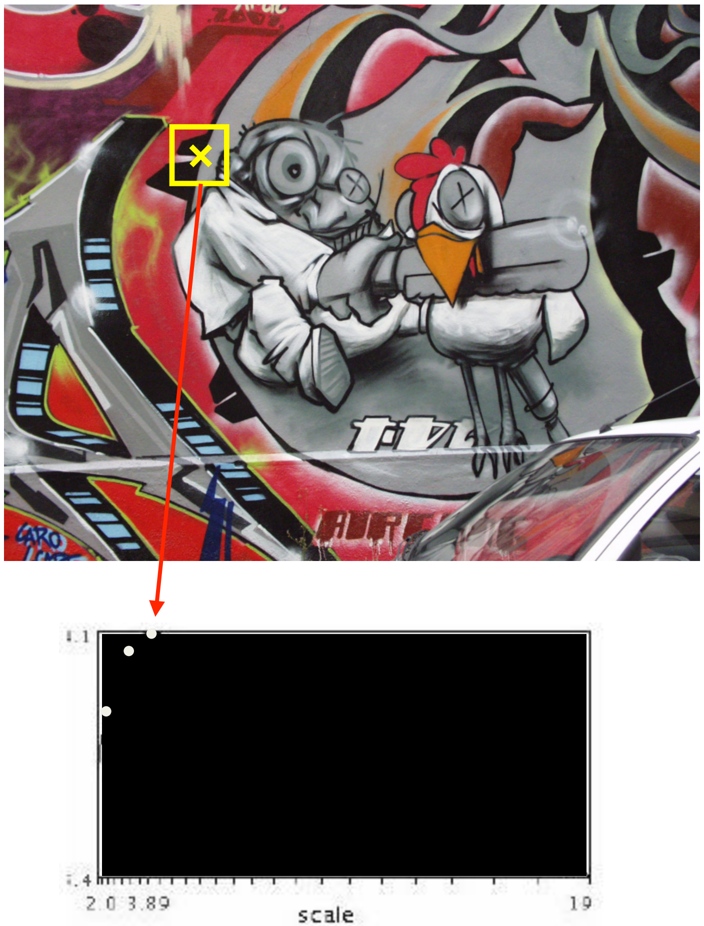
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

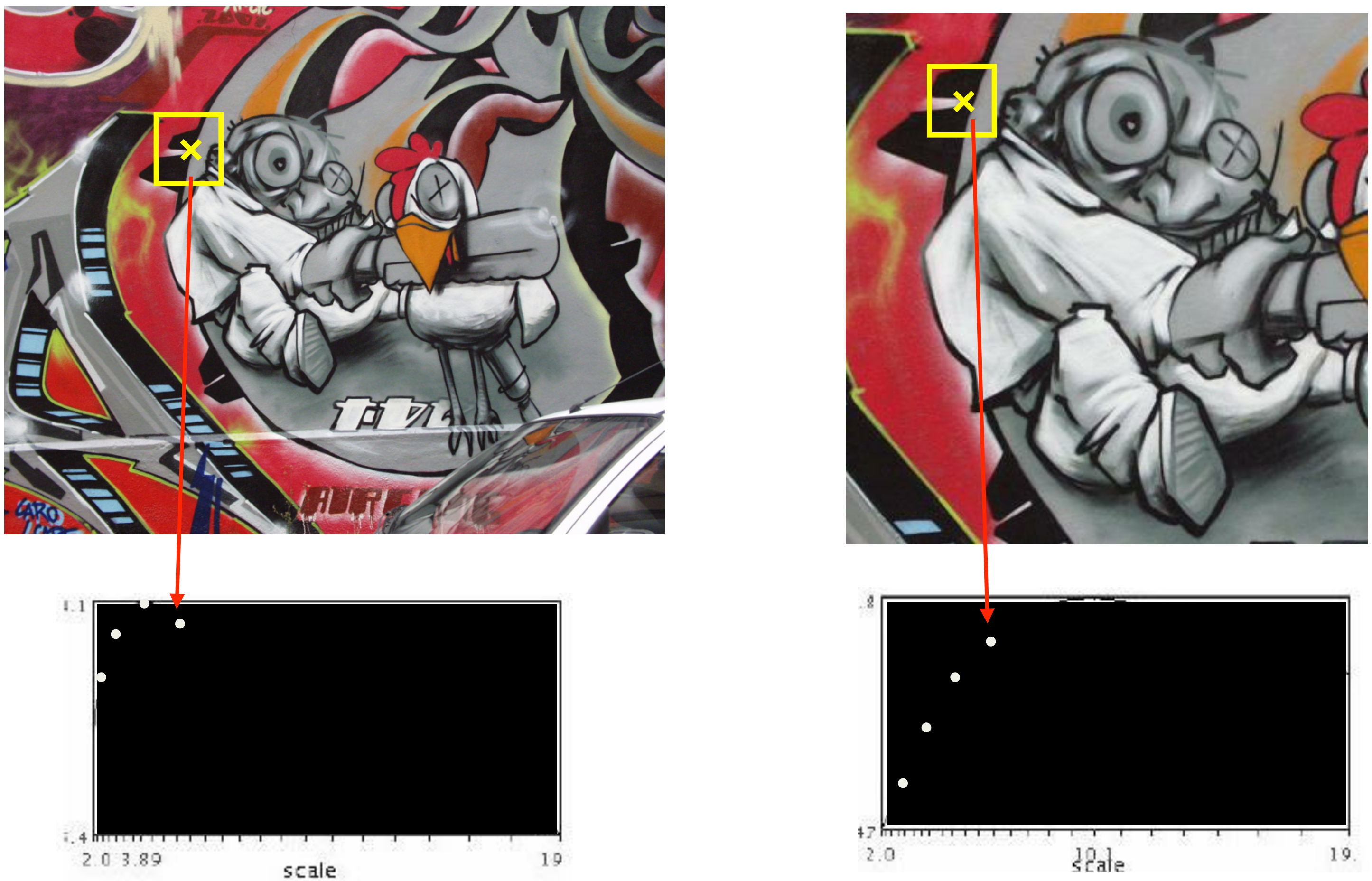


$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

Automatic Scale Selection

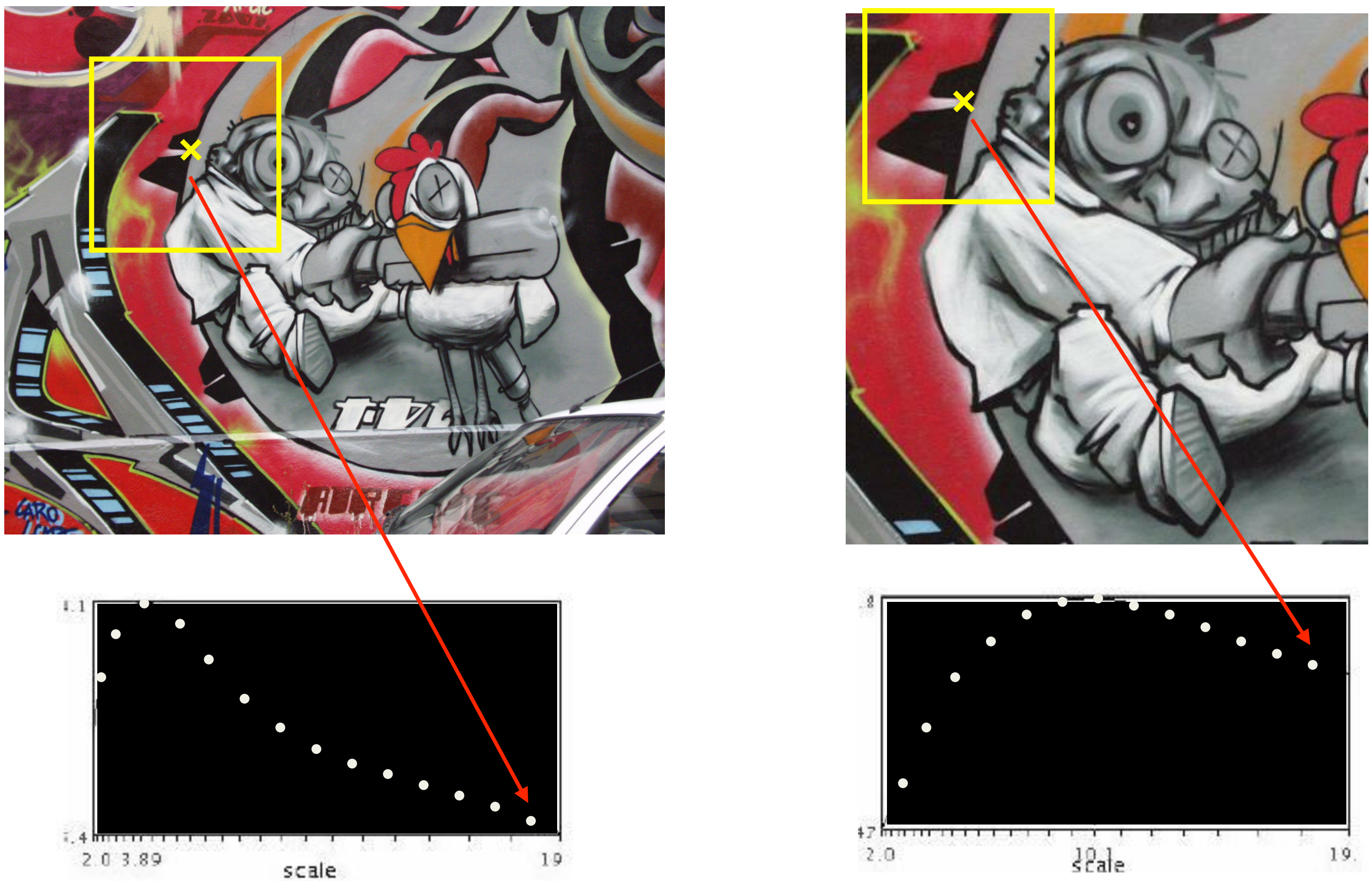
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

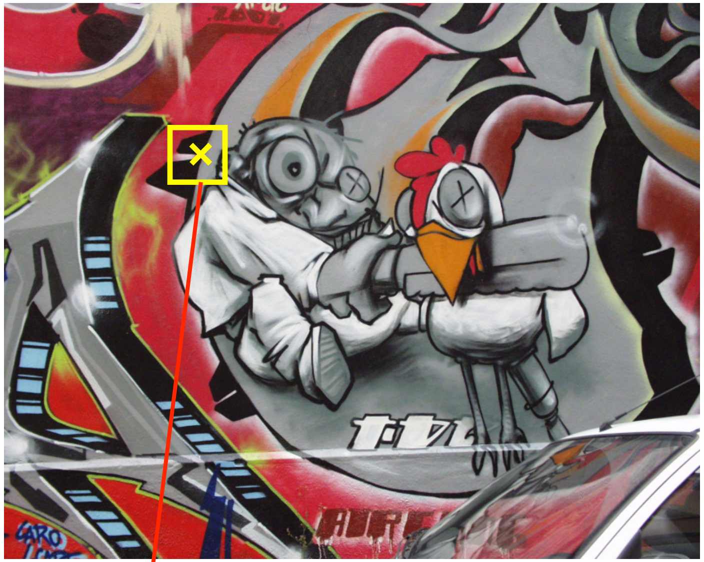
- Function responses for increasing scale (scale signature)

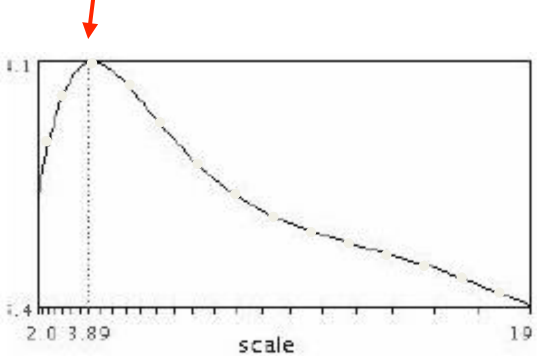


$f(I_{i_1 \ldots i_m}(x, \sigma))$ $f(I_{i_1 \ldots i_m}(x', \sigma))$

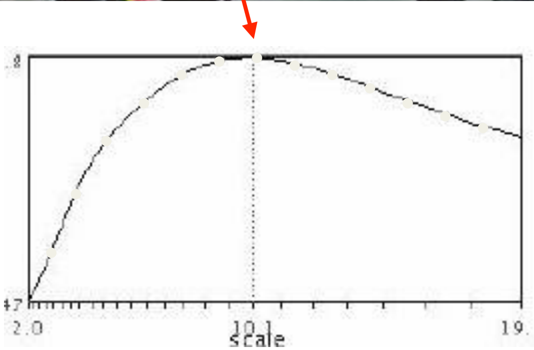
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





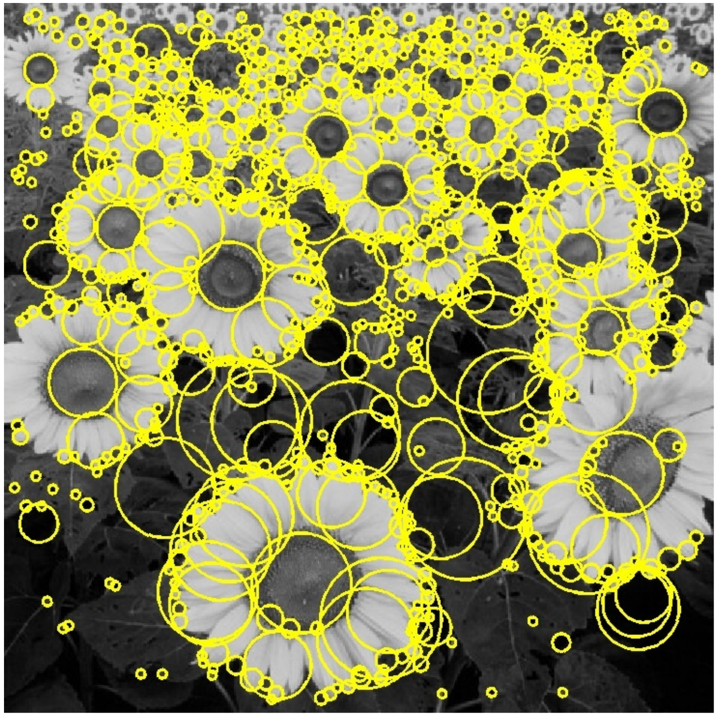
$f(I_{i_1 \ldots i_m}(x, \sigma))$

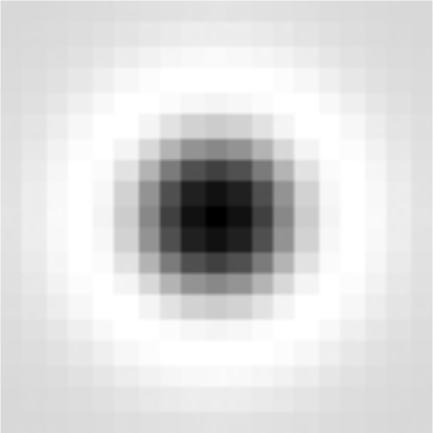


$f(I_{i_1 \ldots i_m}(x', \sigma'))$

Basic idea

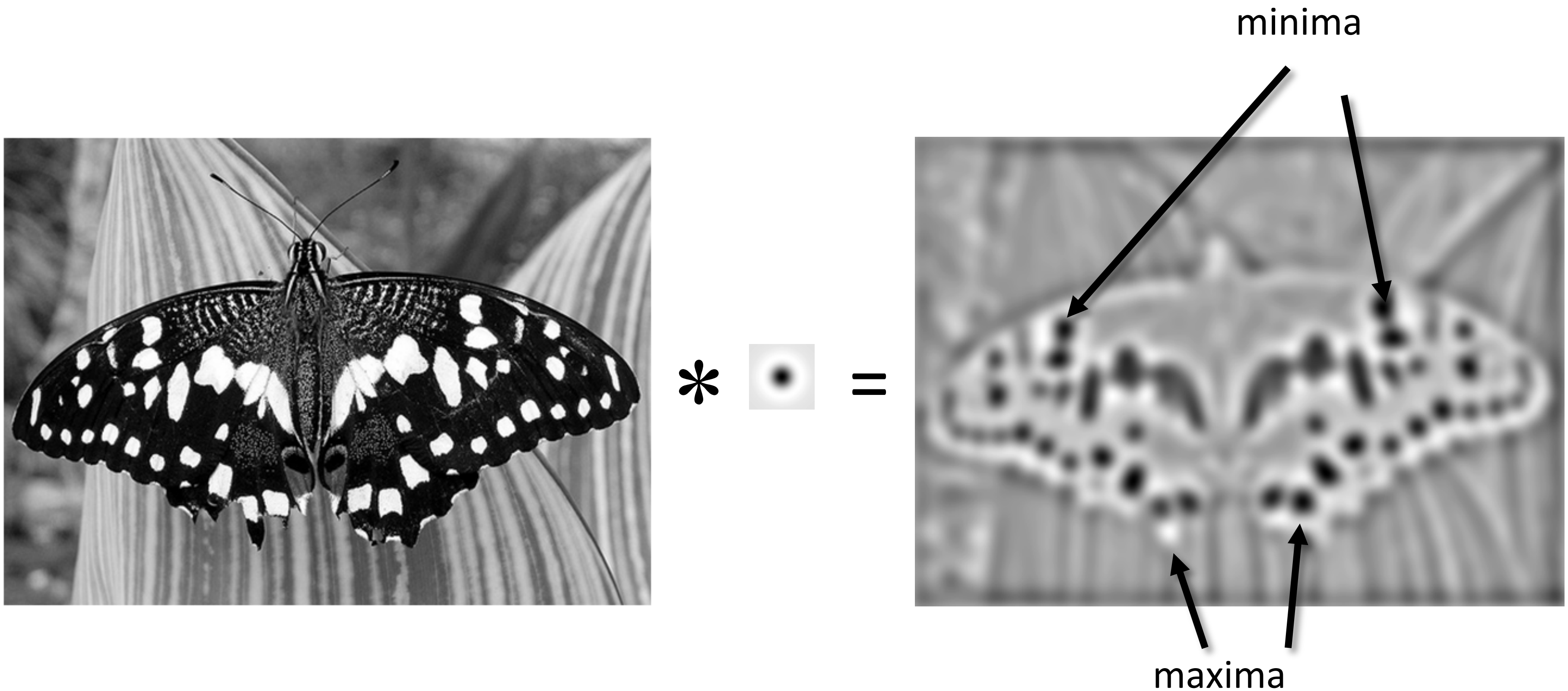
- Convolve the image with a "blob filter" at multiple scales and look for extrema of filter response in the resulting *scale space*





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

Blob detection

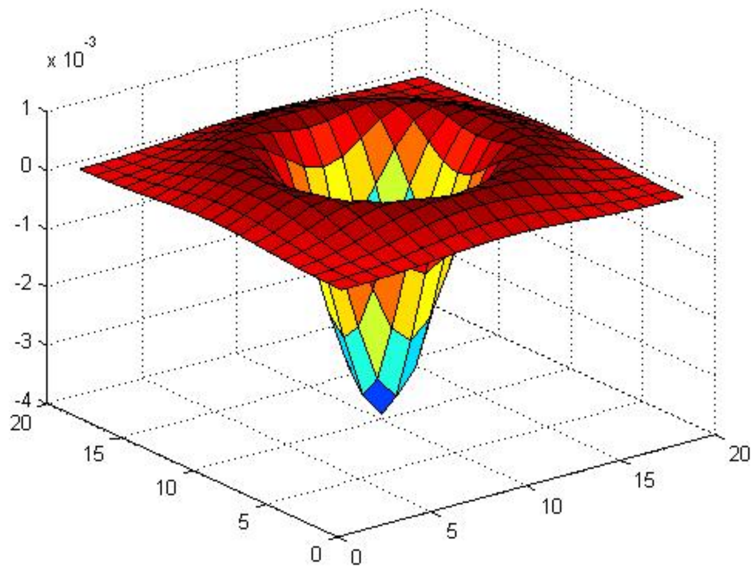


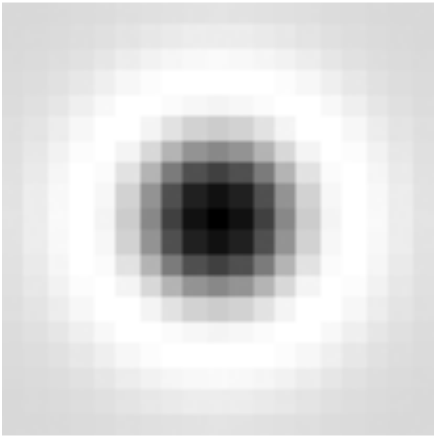
- Find maxima *and minima* of blob filter response in space *and scale*

Slide credit: N. Snavely

Blob detection in 2D

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



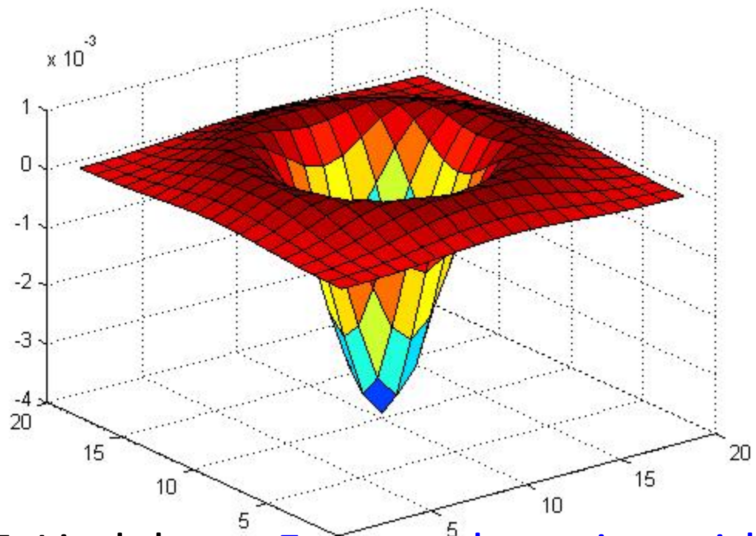


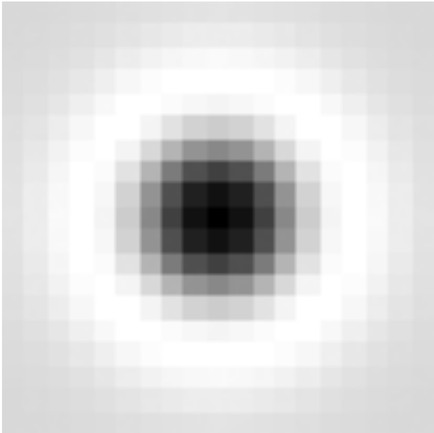
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection in 2D

- Better performance: *Scale-normalized* Laplacian of Gaussian:

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

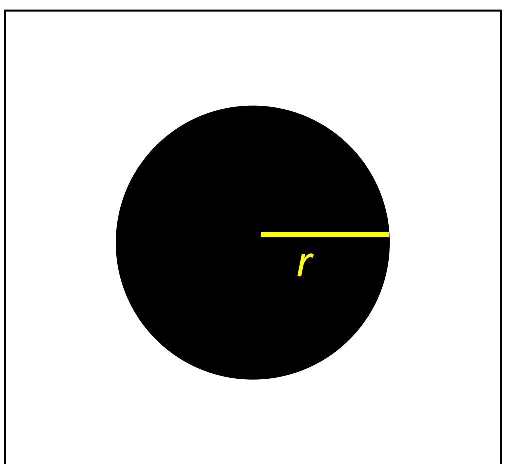




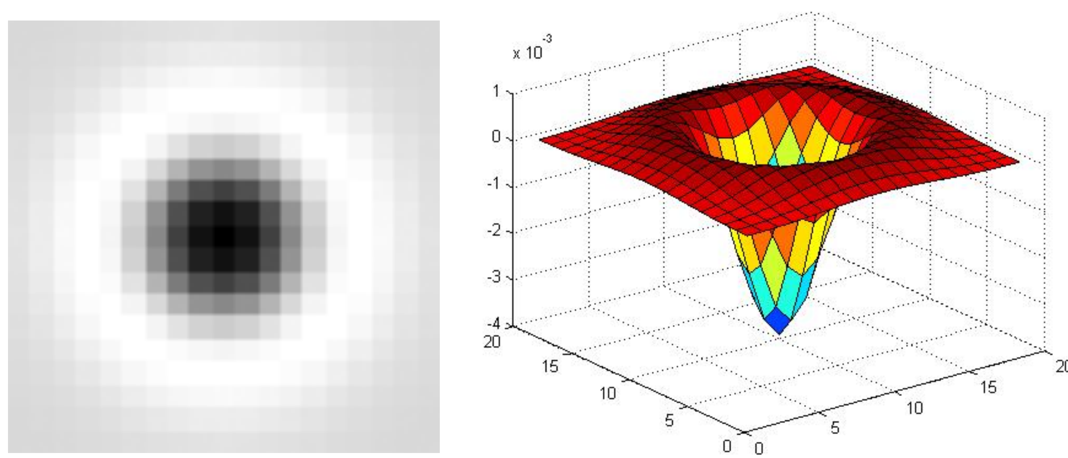
T. Lindeberg, Feature detection with automatic scale selection, *IJCV* 30(2), pp 77-116, 1998

Blob detection in 2D

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



image



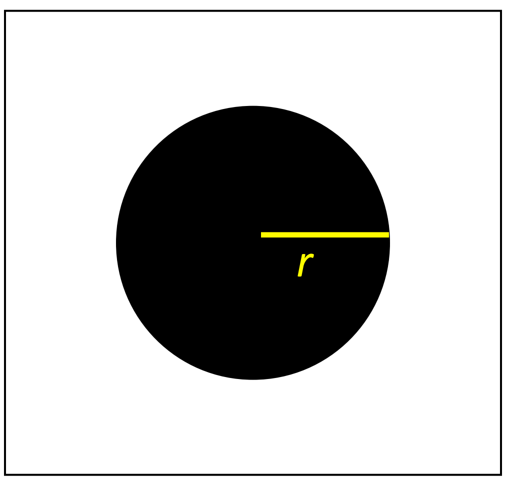
Laplacian

Blob detection in 2D

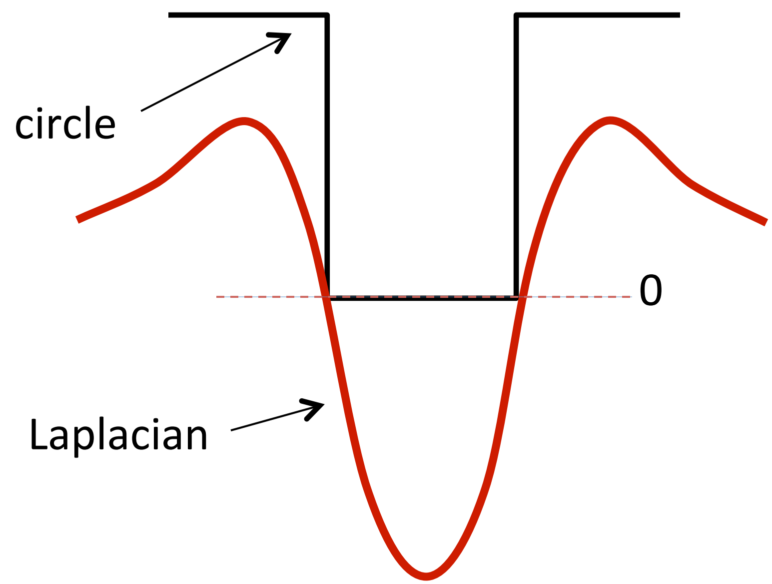
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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

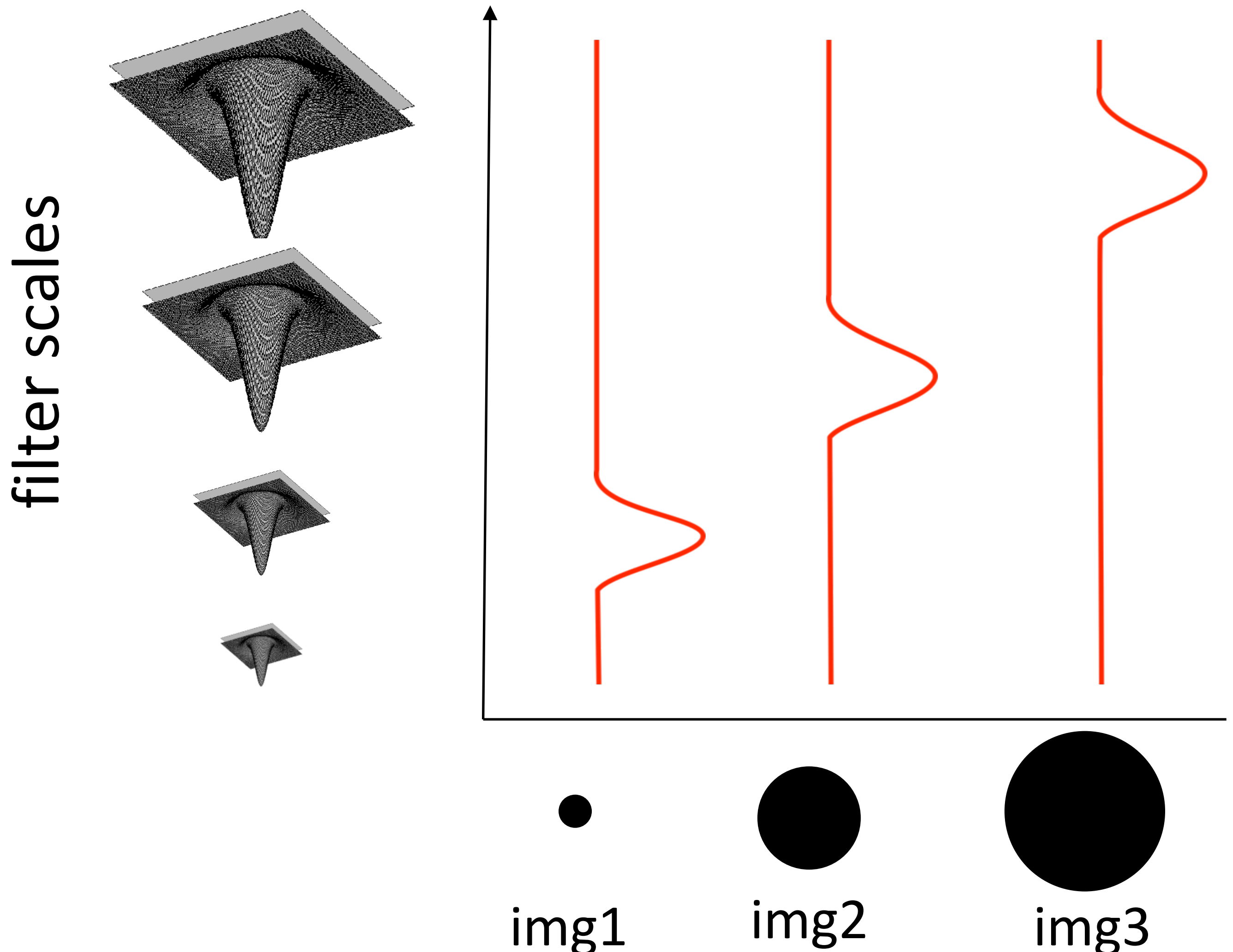


image



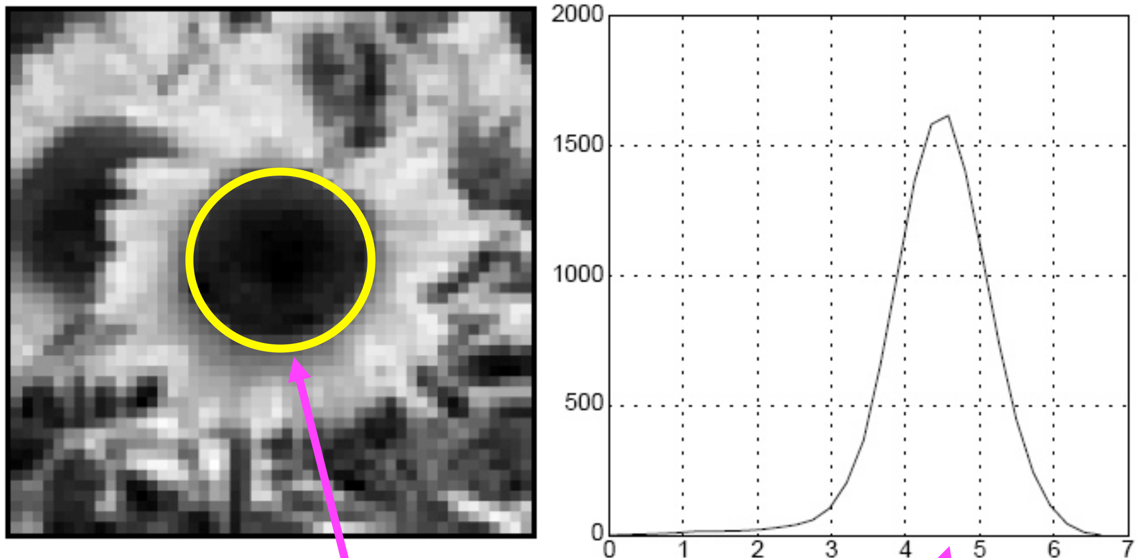
Blob detection in 2D: scale selection

- Laplacian-of-Gaussian = "blob" detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$



Blob detection in 2D

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



characteristic scale

Slide credit: Lana Lazebnik

Scale-space blob detector

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

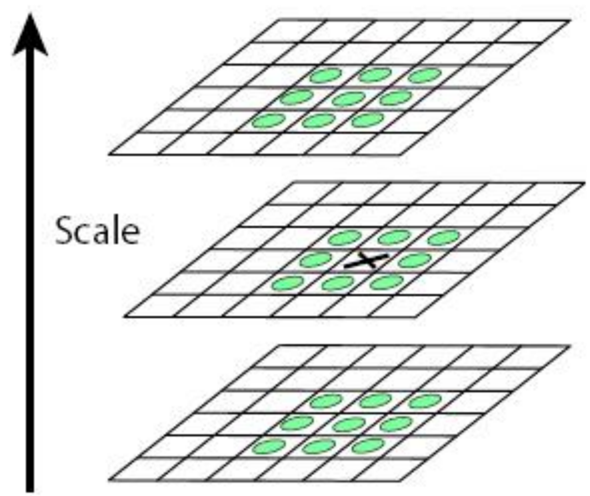
Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

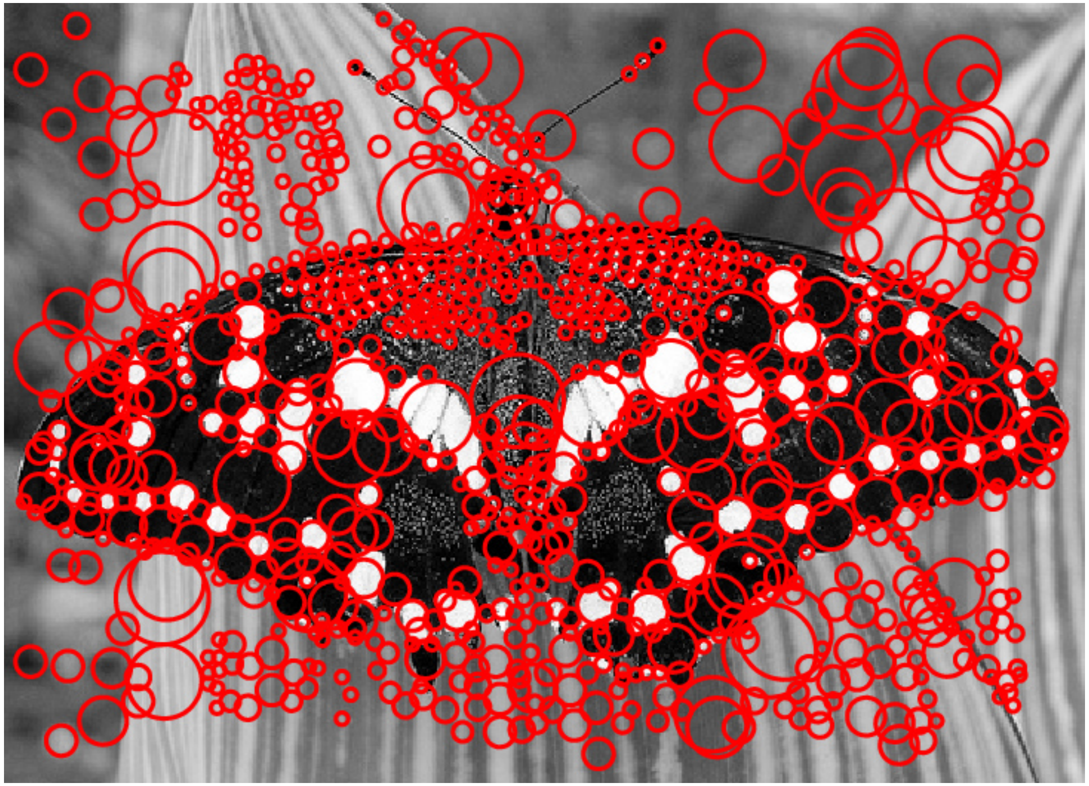
Scale-space blob detector

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



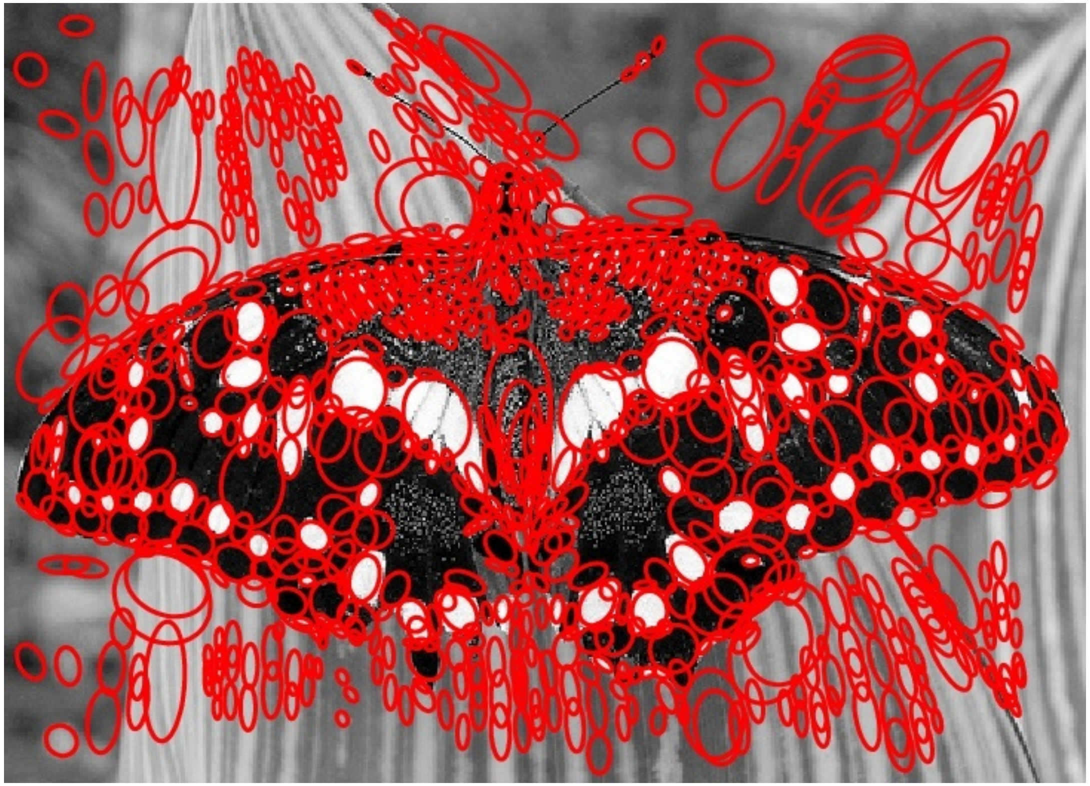
Scale-space blob detector: Example

- Laplacian has strong response along edges



Eliminating edge responses

- Laplacian has strong response along edges



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

Efficient implementation

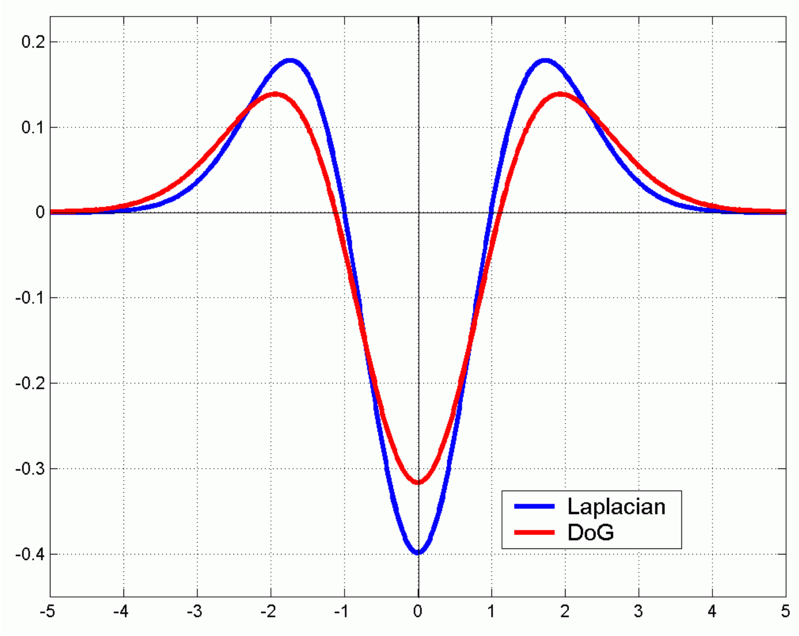
- Approximating the scale-normalized Laplacian with a difference of Gaussians:

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

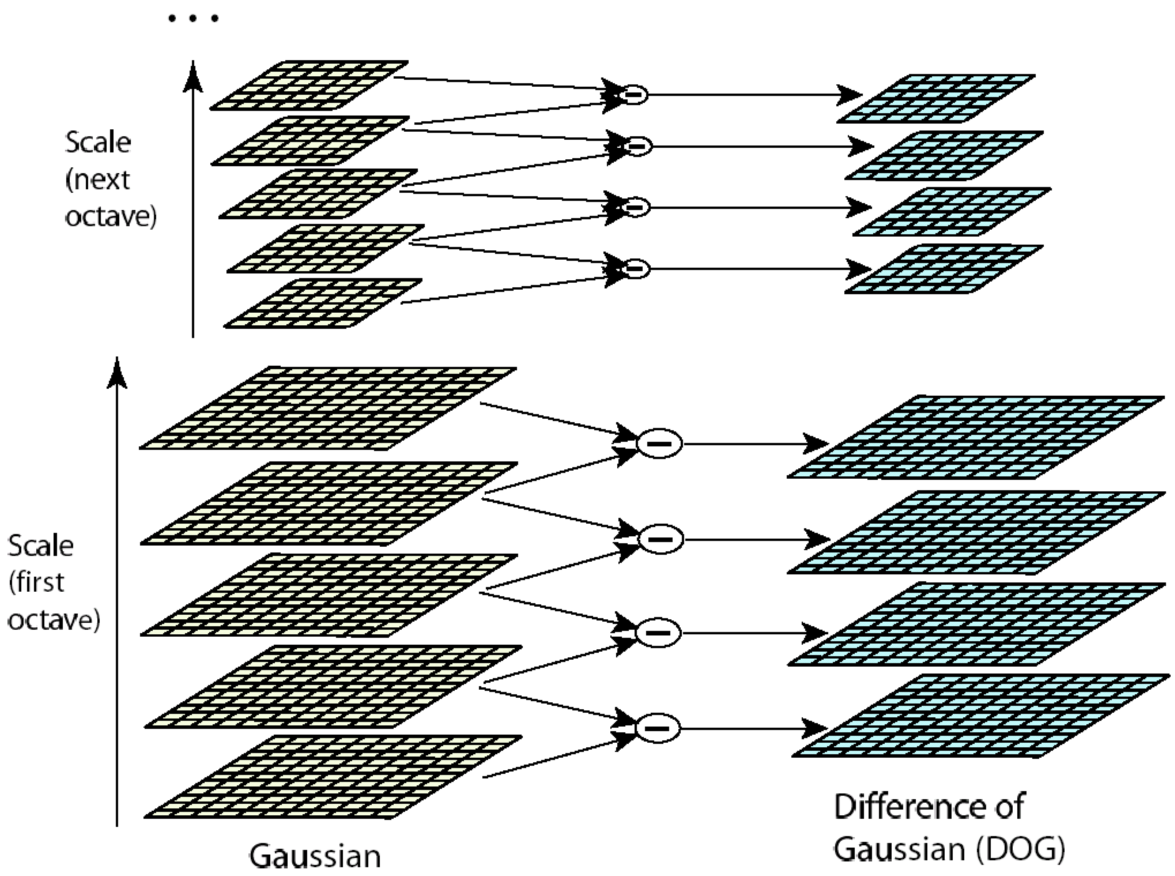
(Scale normalized Laplacian)

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

(Difference of Gaussians)



Efficient implementation



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

Example

Original image at
¾ the size

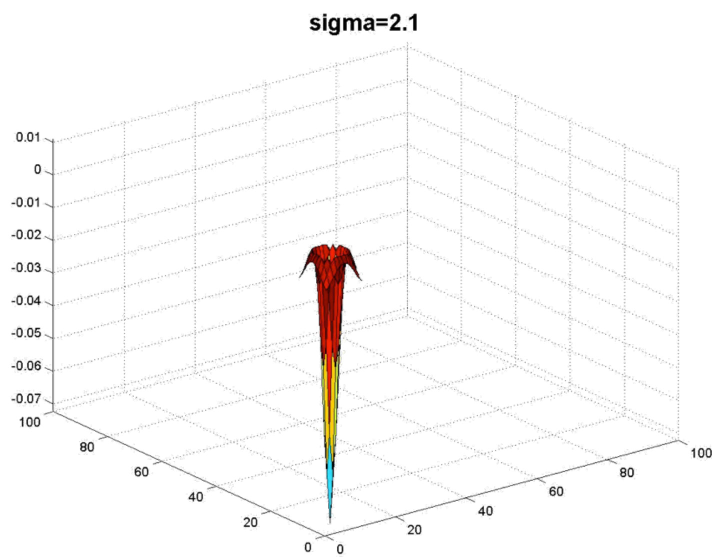
Slide credit: Kristen Grauman

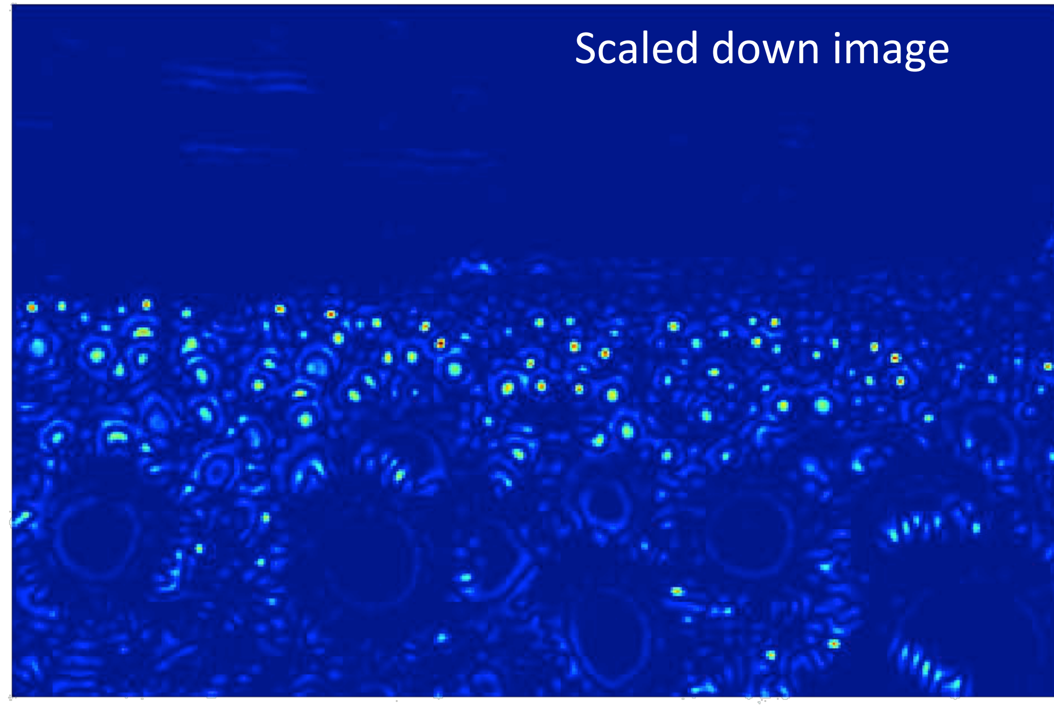
Example

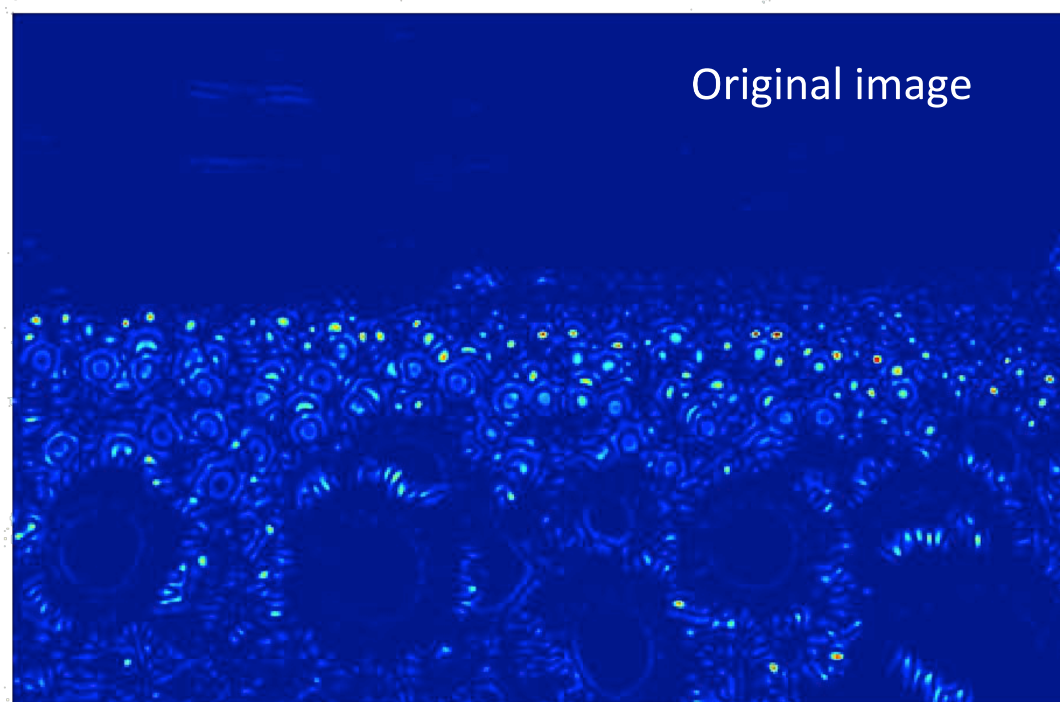
Original image at
¾ the size

Slide credit: Kristen Grauman

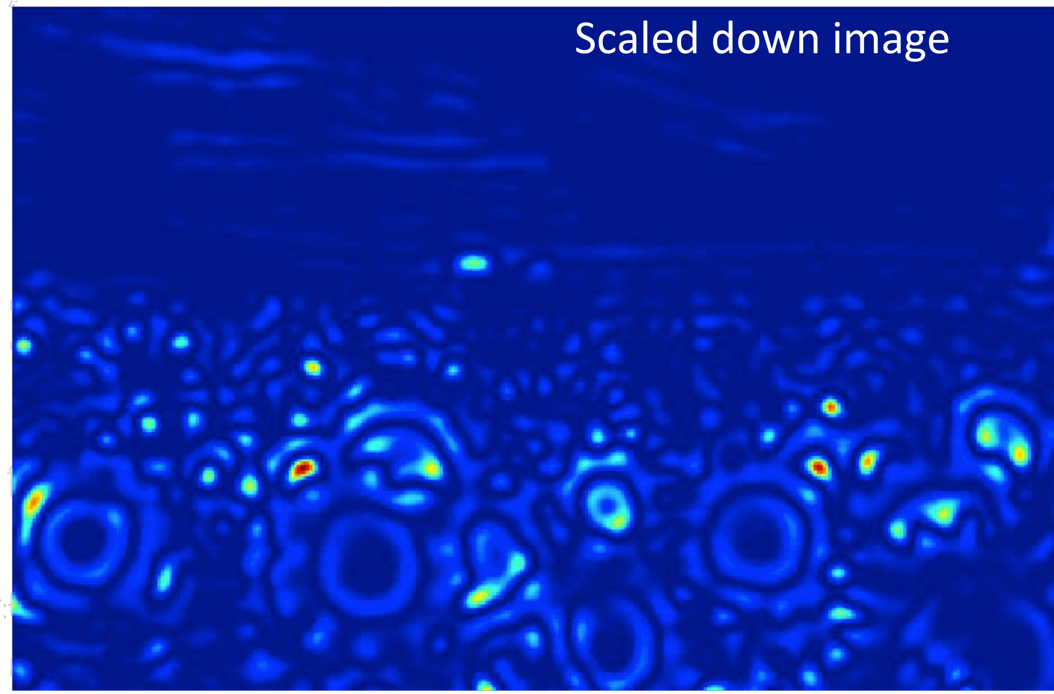
Original image at
¾ the size

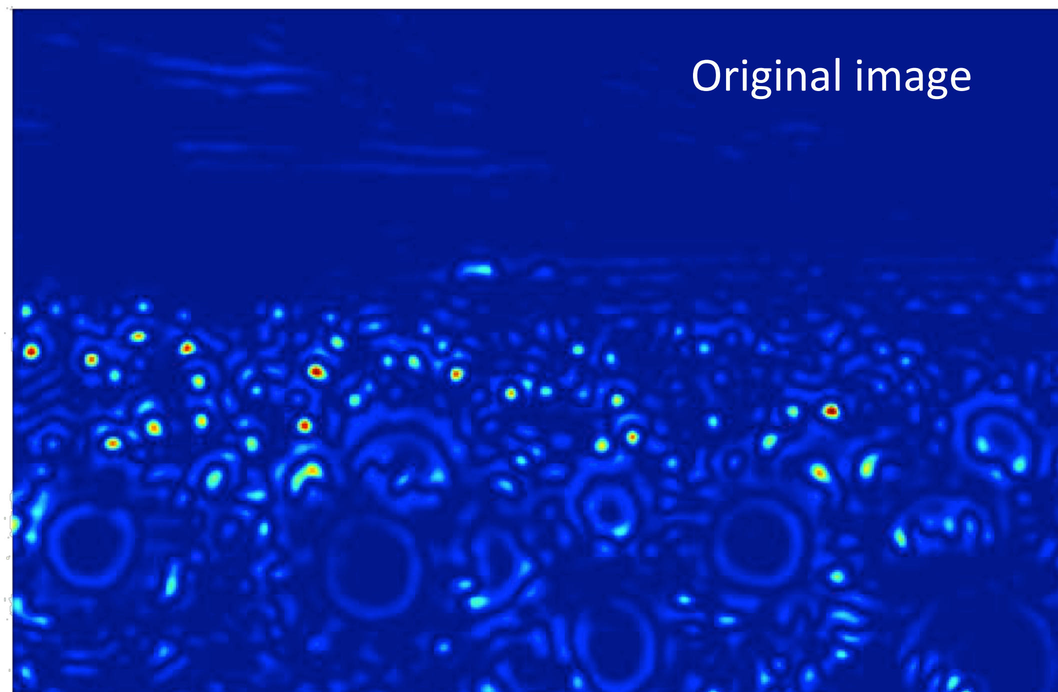


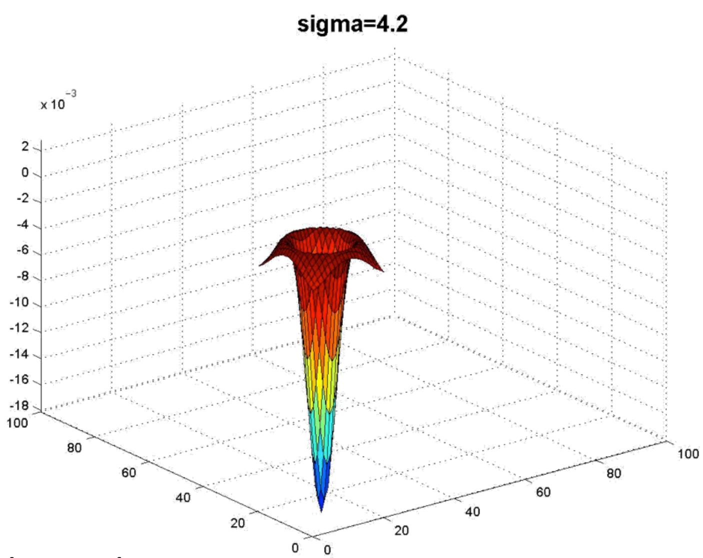




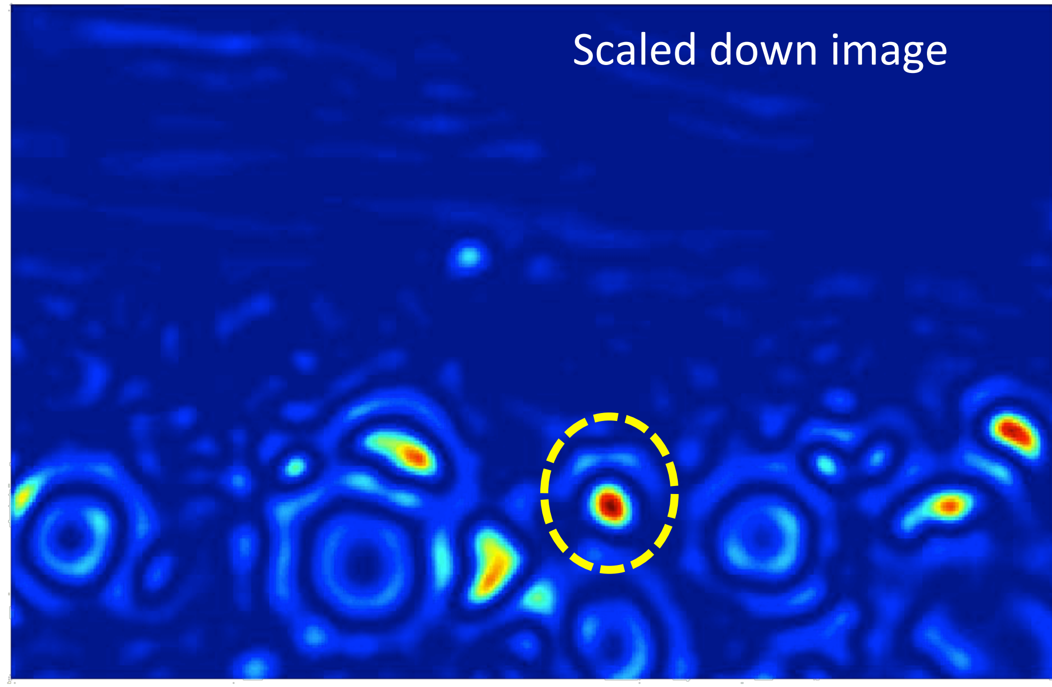
Slide credit: Kristen Grauman

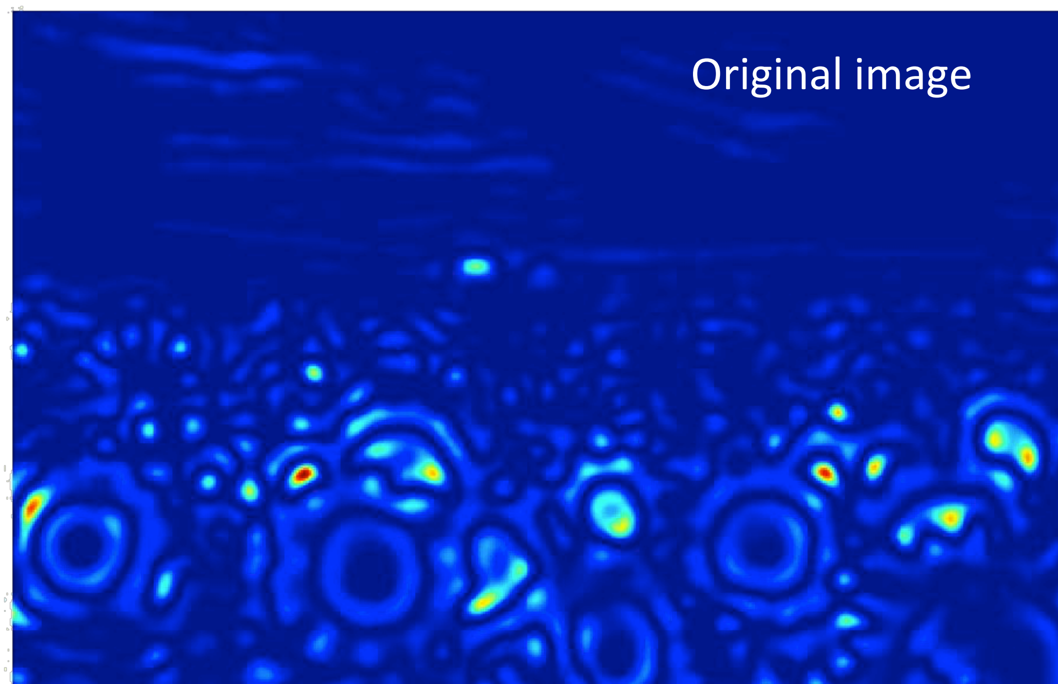


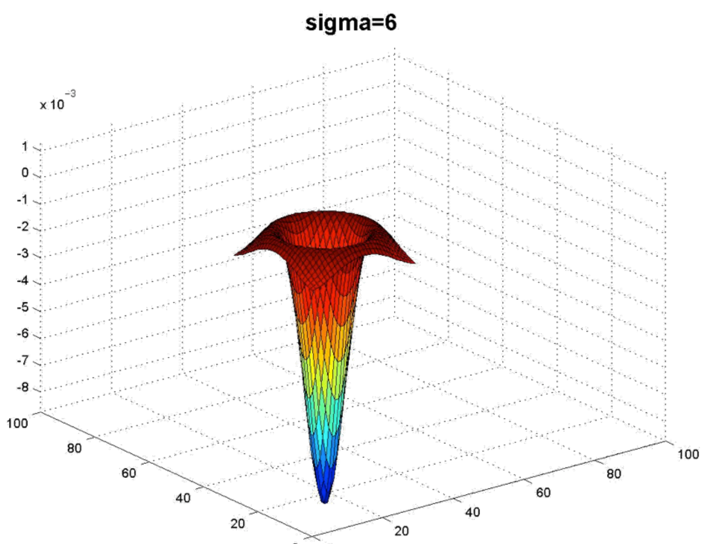




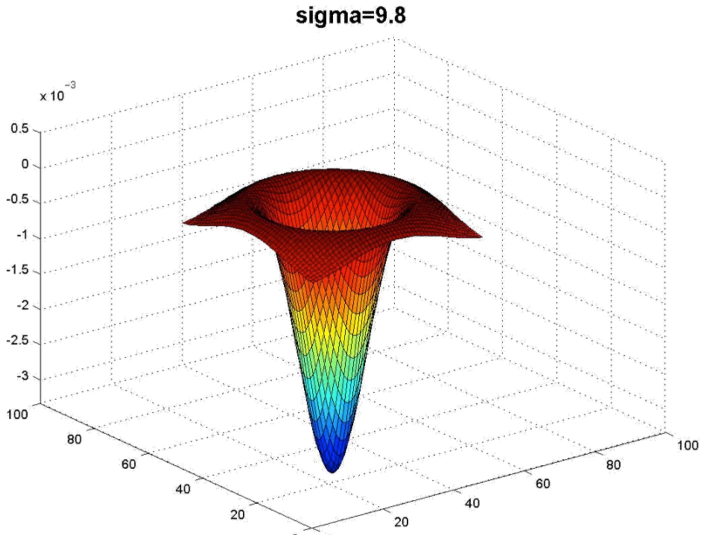
Slide credit: Kristen Grauman

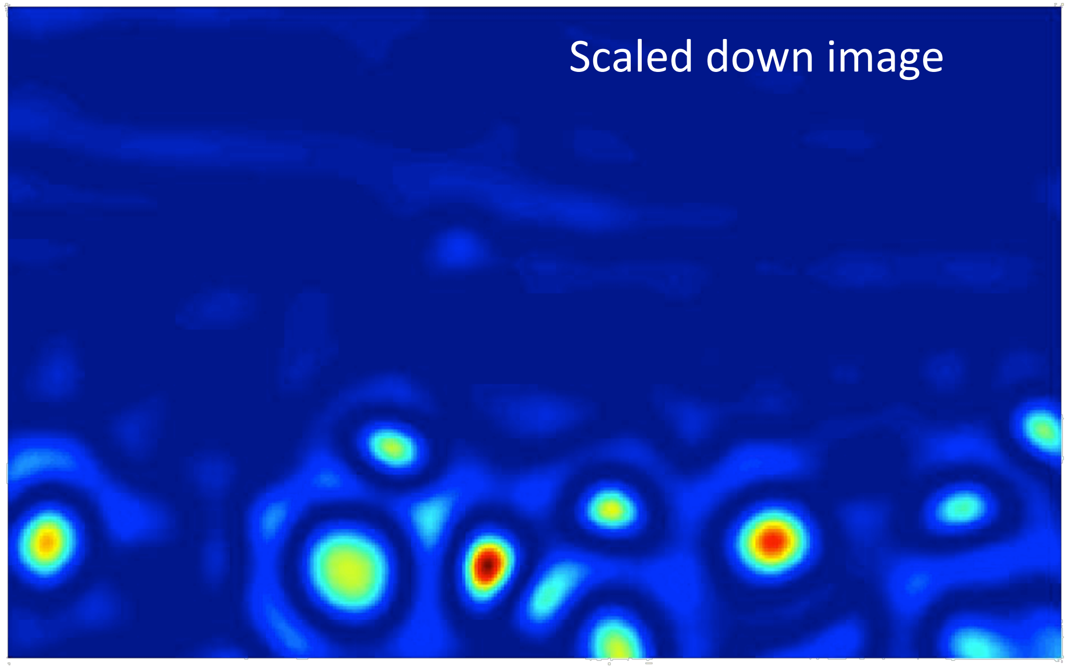


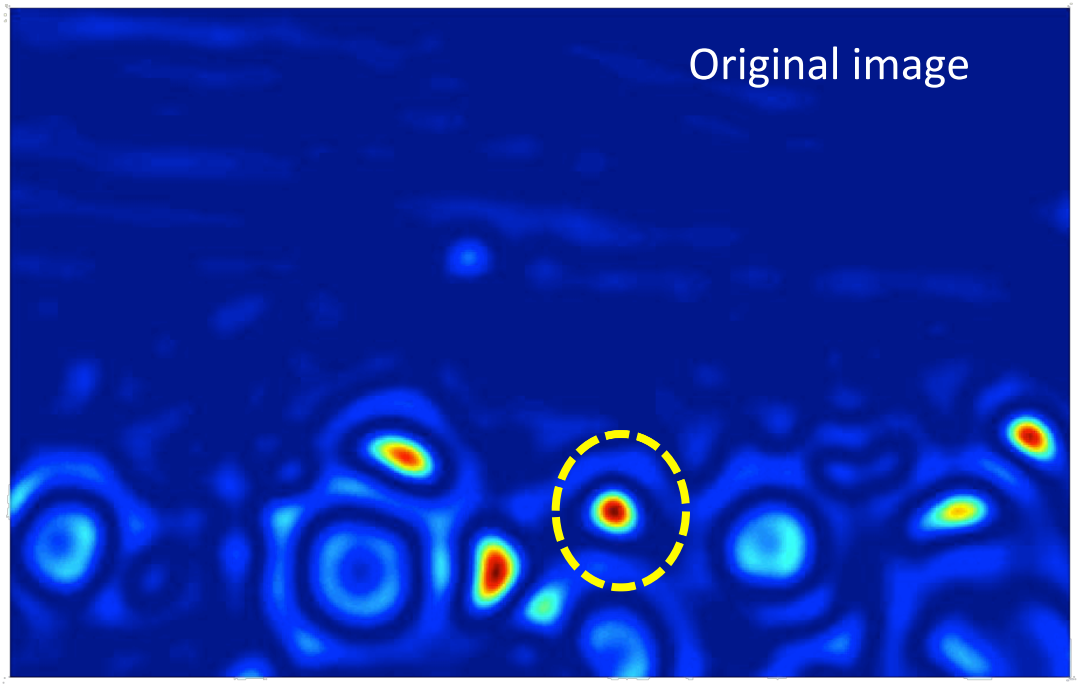




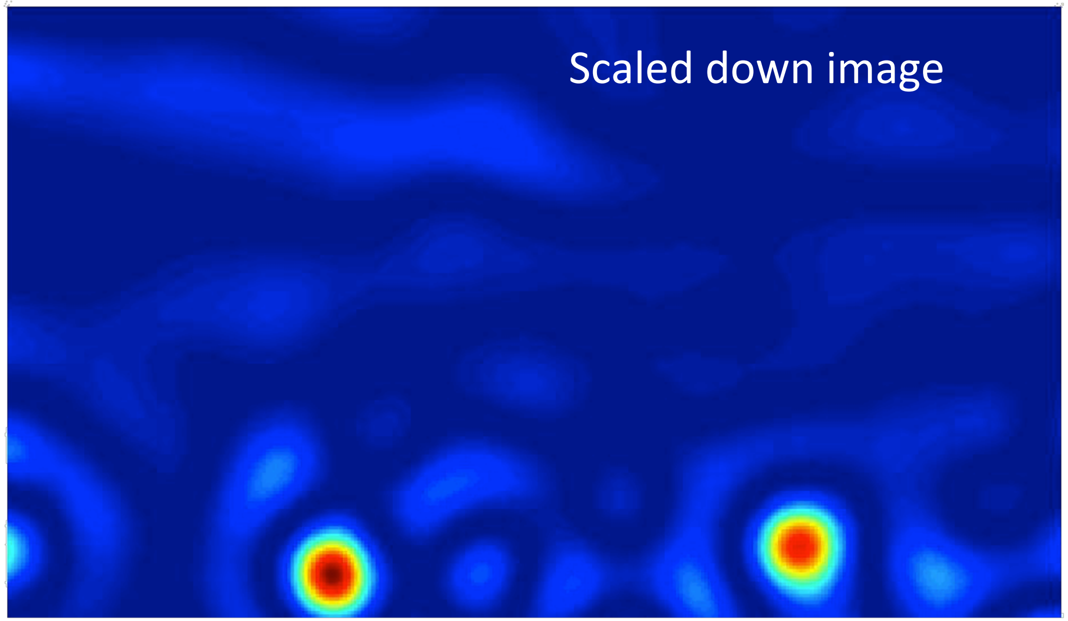
Slide credit: Kristen Grauman

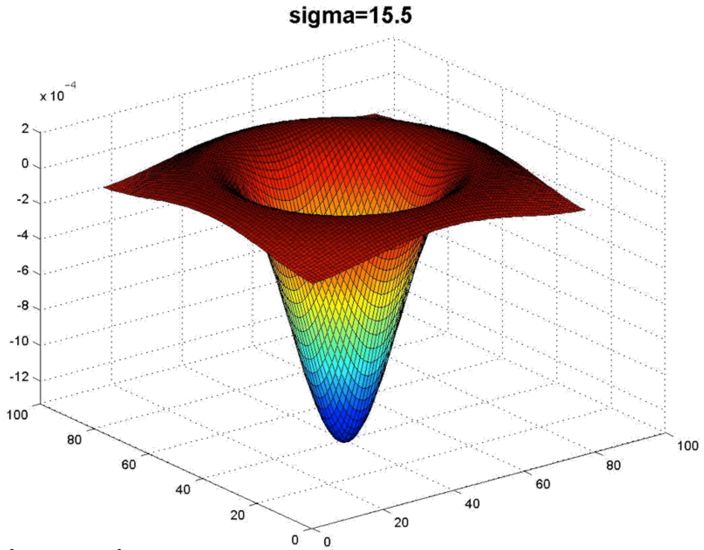


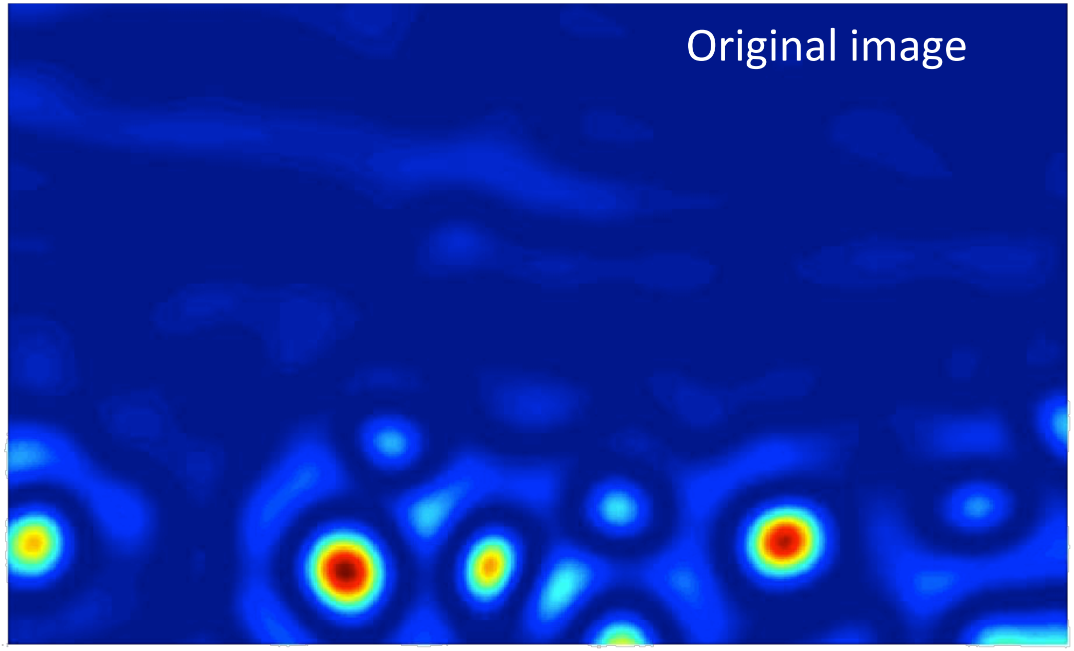




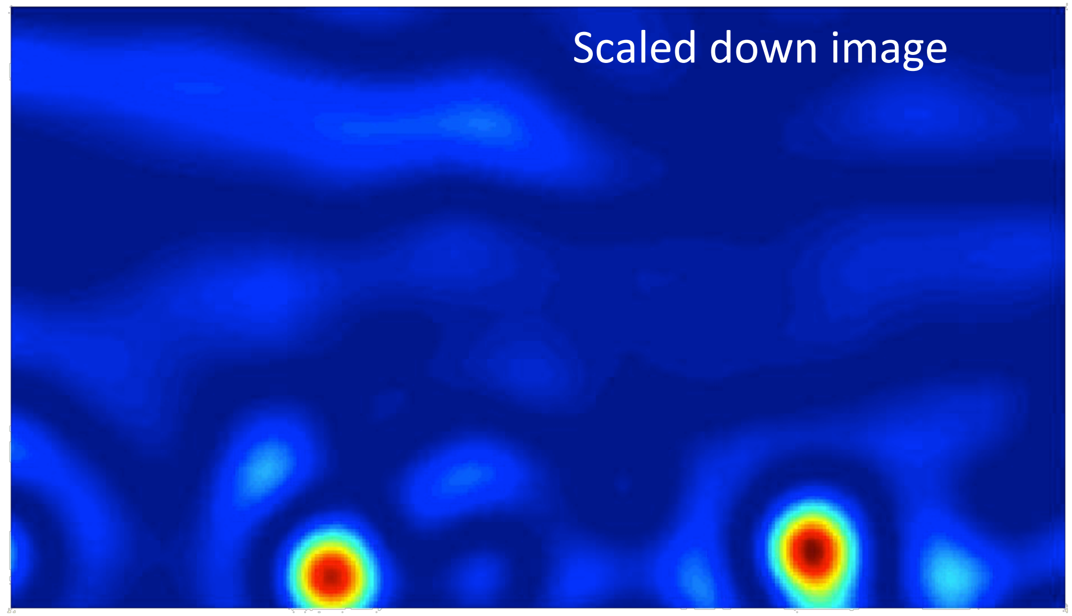
Slide credit: Kristen Grauman

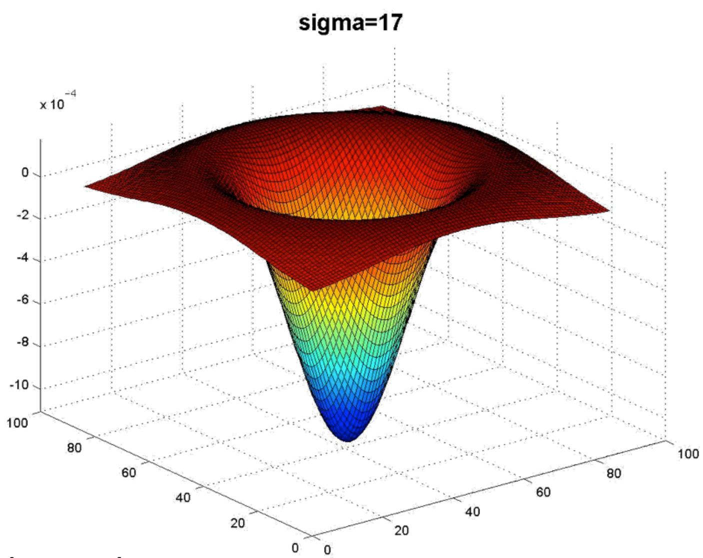


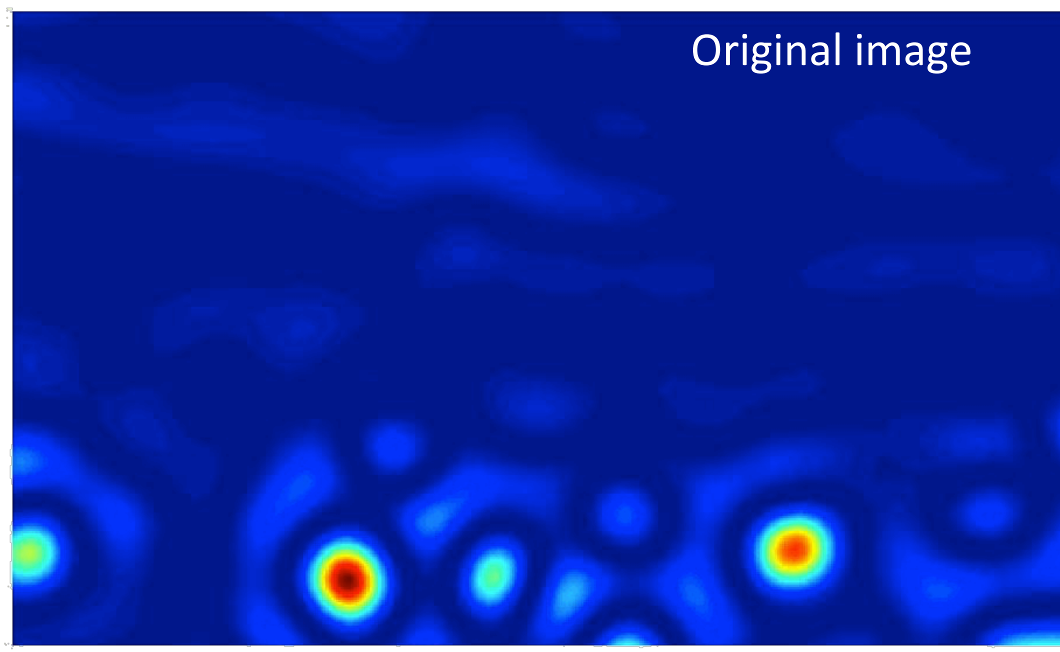




Slide credit: Kristen Grauman



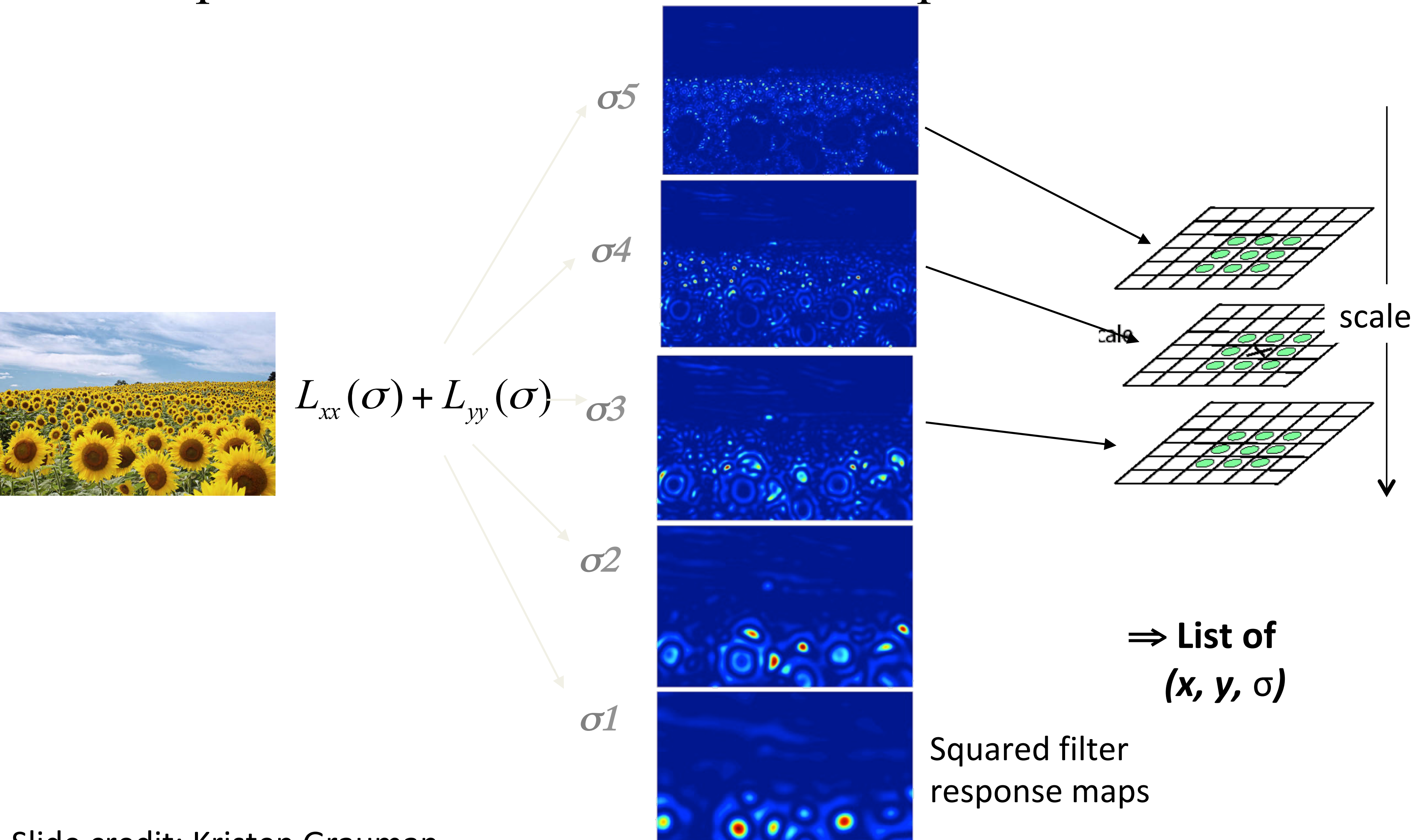




Slide credit: Kristen Grauman

Scale invariant interest points

Interest points are local maxima in both position and scale.



$\Rightarrow$ **List of *(x, y, σ)***

Slide credit: Kristen Grauman

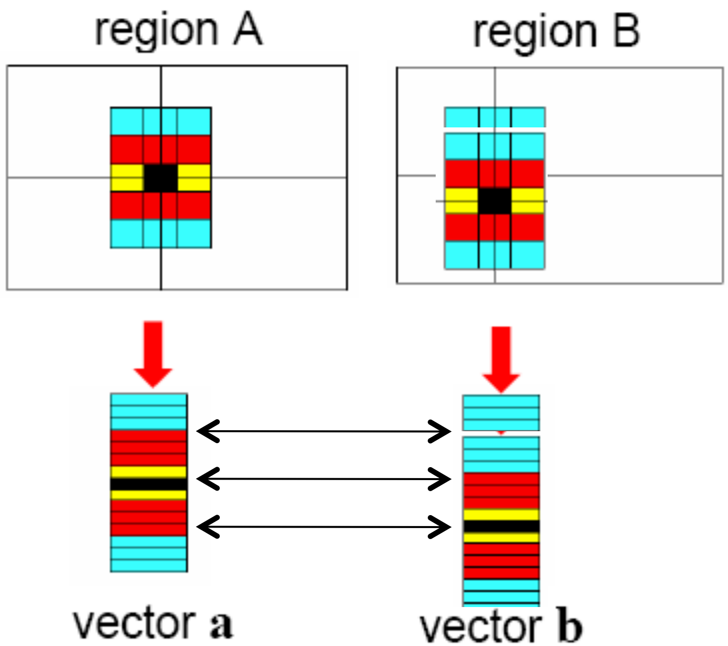
Comparison of Feature Detectors

Table 7.1 Overview of feature detectors.

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Tuytelaars Mikolajczyk 2008

Raw patches as local descriptors



The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

Figure: Andrew Zisserman

Local Descriptors

- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient

- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

David Lowe

FOLLOW

Computer Science Dept., University of British Columbia
Verified email at cs.ubc.ca - Homepage

Computer Vision   Object Recognition

TITLE	CITED BY	YEAR
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	54091	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	18041	1999
Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. M Muja, DG Lowe VISAPP (1) 2, 331-340	2876	2009
Automatic panoramic image stitching using invariant features M Brown, DG Lowe International Journal of Computer Vision 74 (1), 59-73	2427	2007

Distinctive Image Features from Scale-Invariant Keypoints

Abstract

This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

SIFT properties

- Invariant to
  - Scale
  - Rotation

- Partially invariant to
  - Illumination changes
  - Camera viewpoint
  - Occlusion, clutter

Distinctive Image Features from Scale-Invariant Keypoints

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

This approach has been named the Scale Invariant Feature Transform (SIFT), as it transforms image data into scale-invariant coordinates relative to local features.