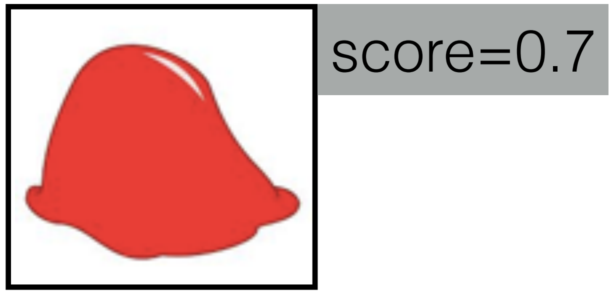


CMPSCI 670: Computer Vision

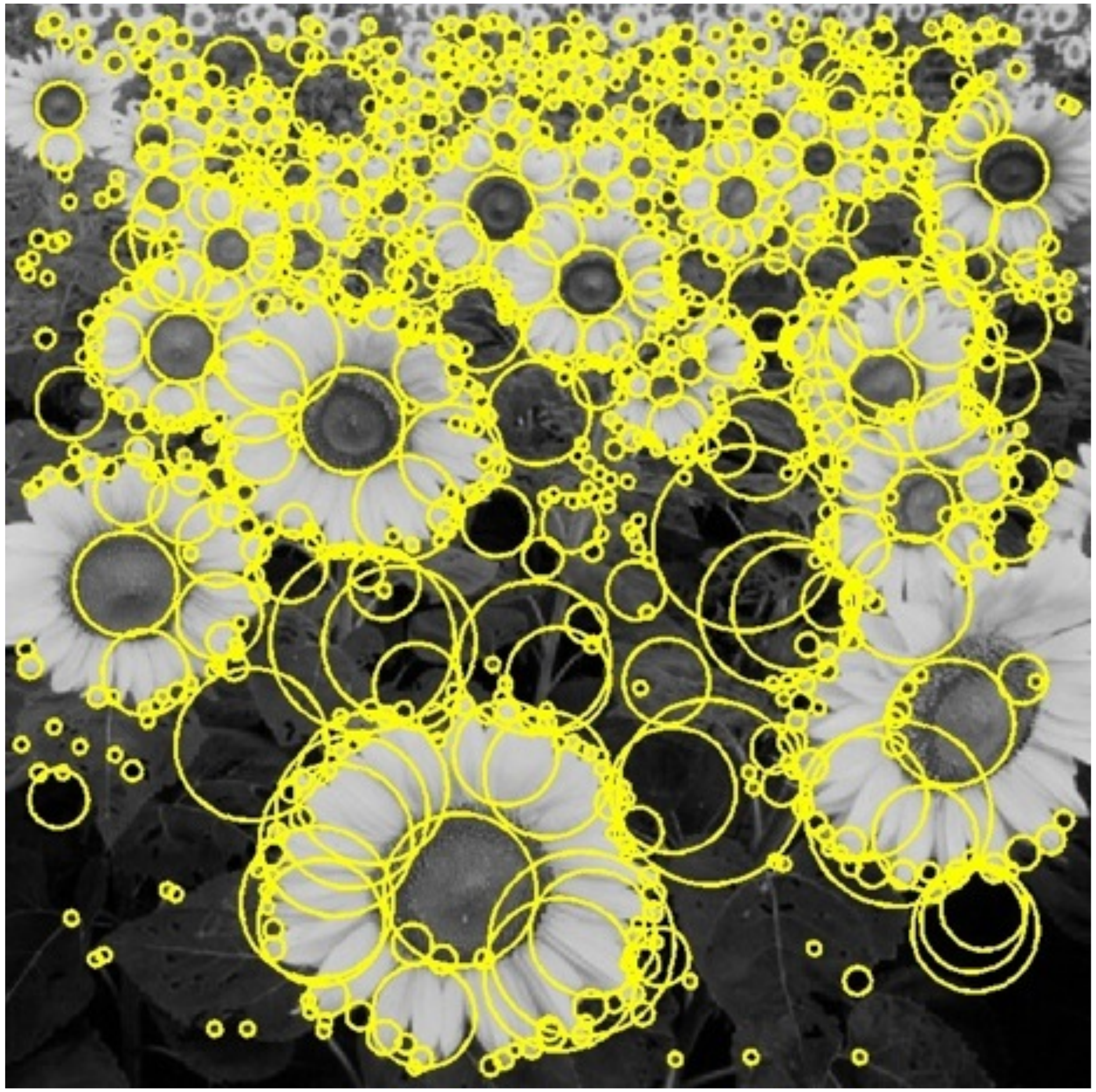
Blob detection

University of Massachusetts, Amherst

October 1, 2014

Instructor: Subhransu Maji

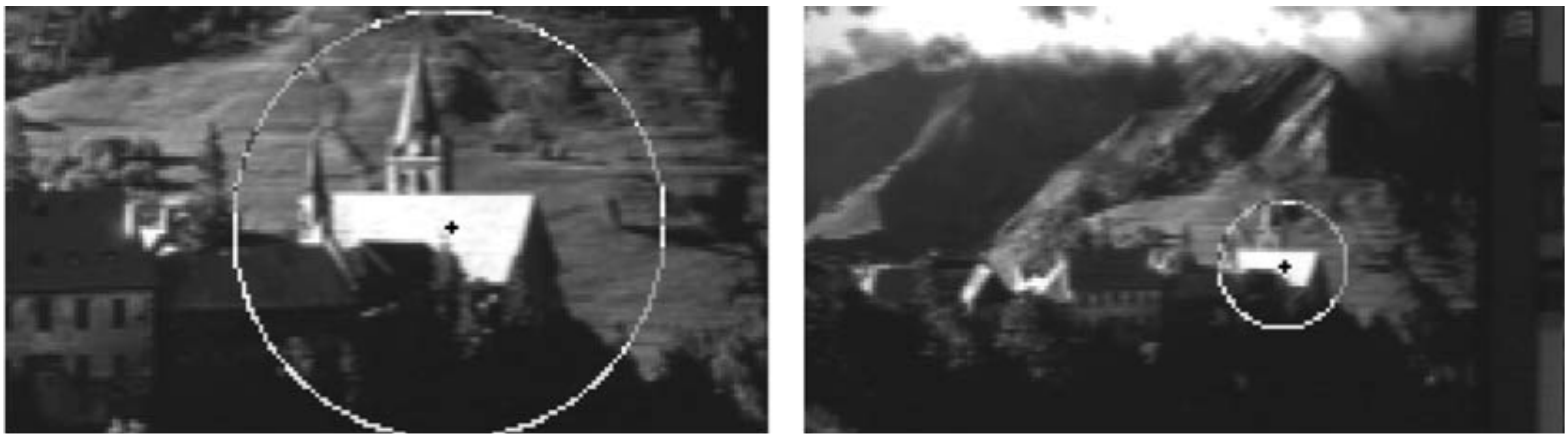
Blob detection



Source: L. Lazebnik

Feature detection with scale selection

- We want to extract features with characteristic scale that is *covariant* with the image transformation such as scaling and translation

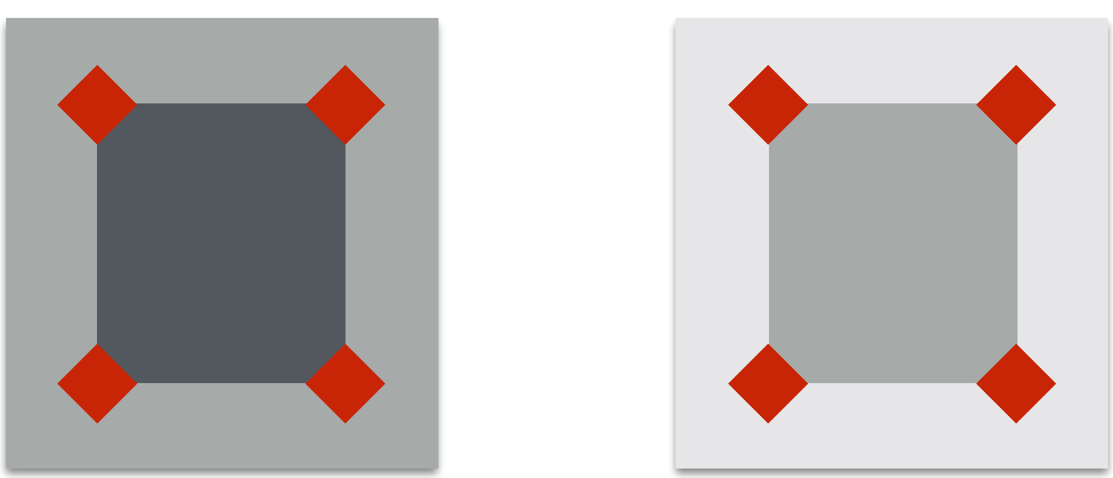


Matching regions across scales

Recall: invariance and covariance

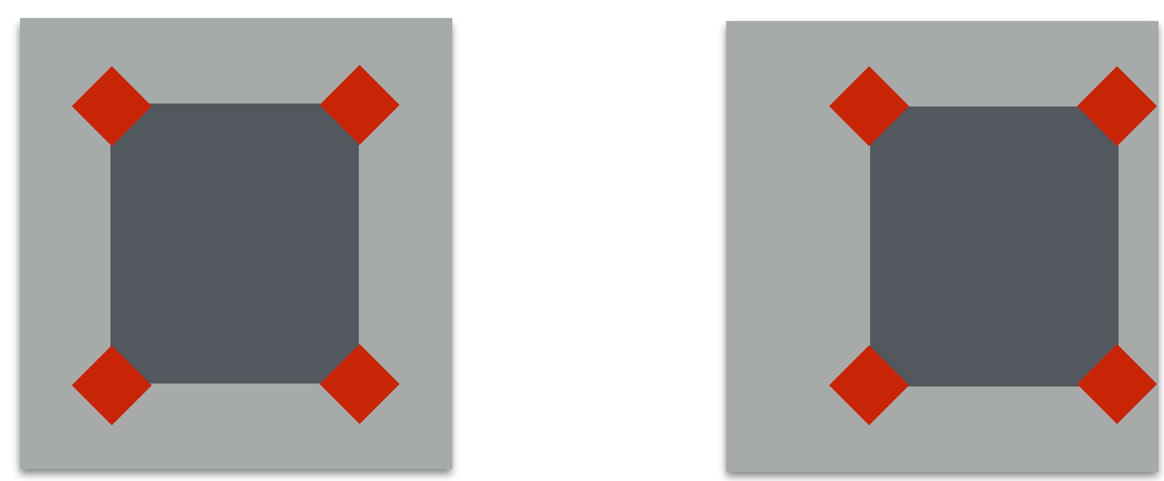
Invariance

- The property should not change when the input is transformed
- For e.g., an *intensity invariant* corner detector finds the same corners even if the intensity of the image is changed

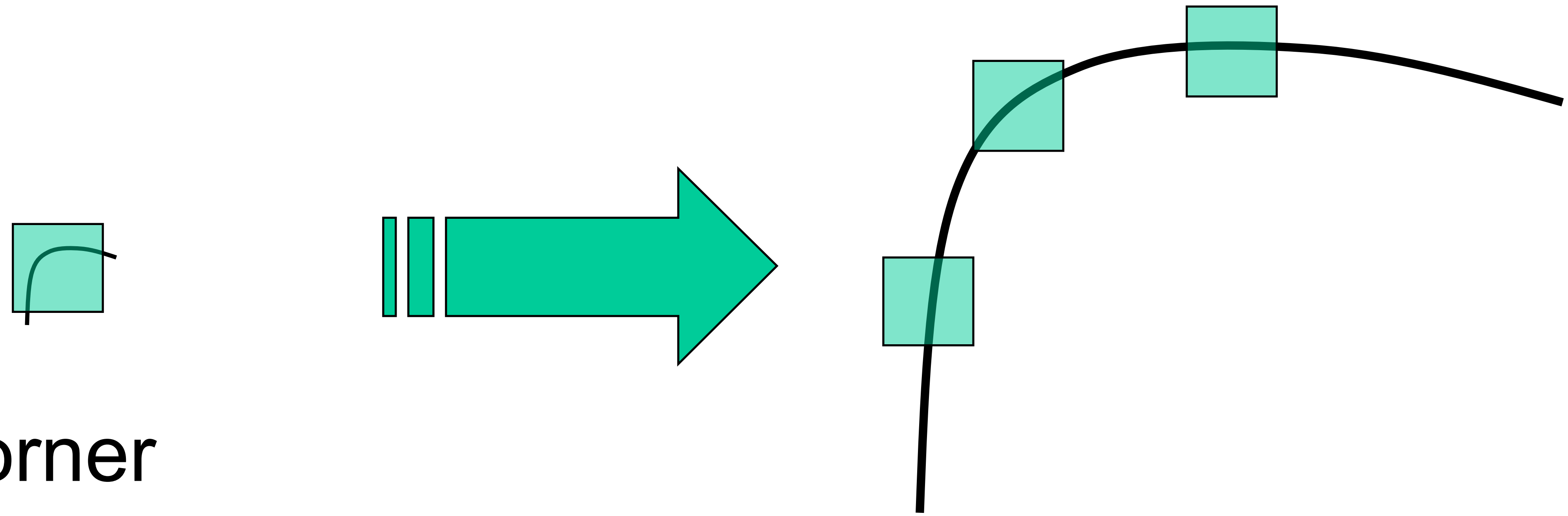


Covariance

- The property should be transformed according to the image transformation
- For e.g., a *translation covariant* corner detector finds the same corners translated by the amount the image is translated



Scaling



Corner

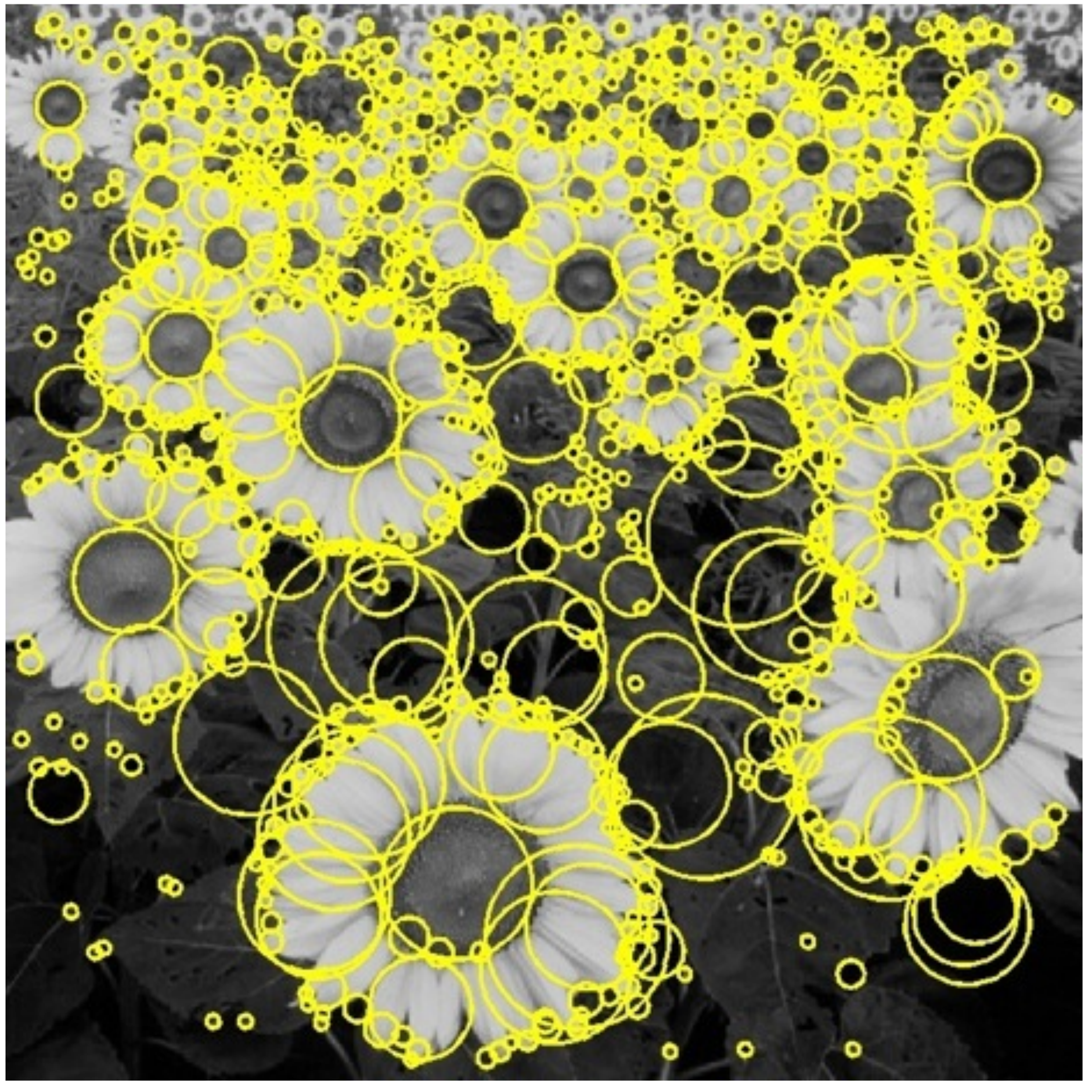
All points will be classified as edges

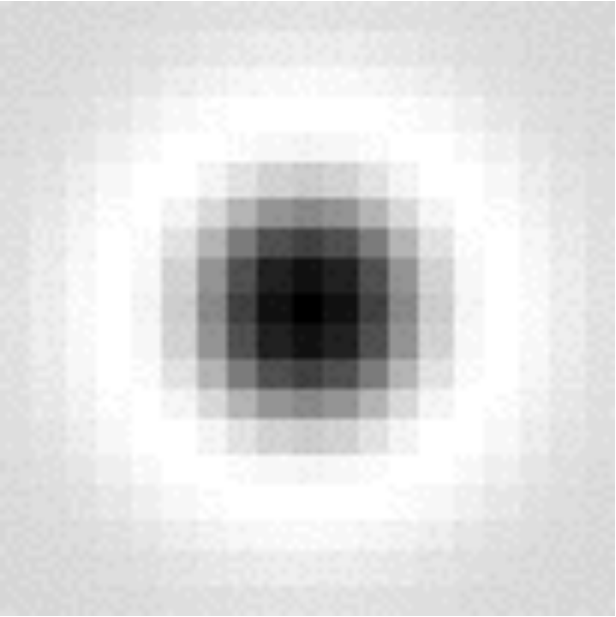
Corner location is not covariant to scaling!

Source: L. Lazebnik

Blob detection: Basic idea

- To detect blobs, convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting *scale space*
- This will give us a scale and space covariant detector

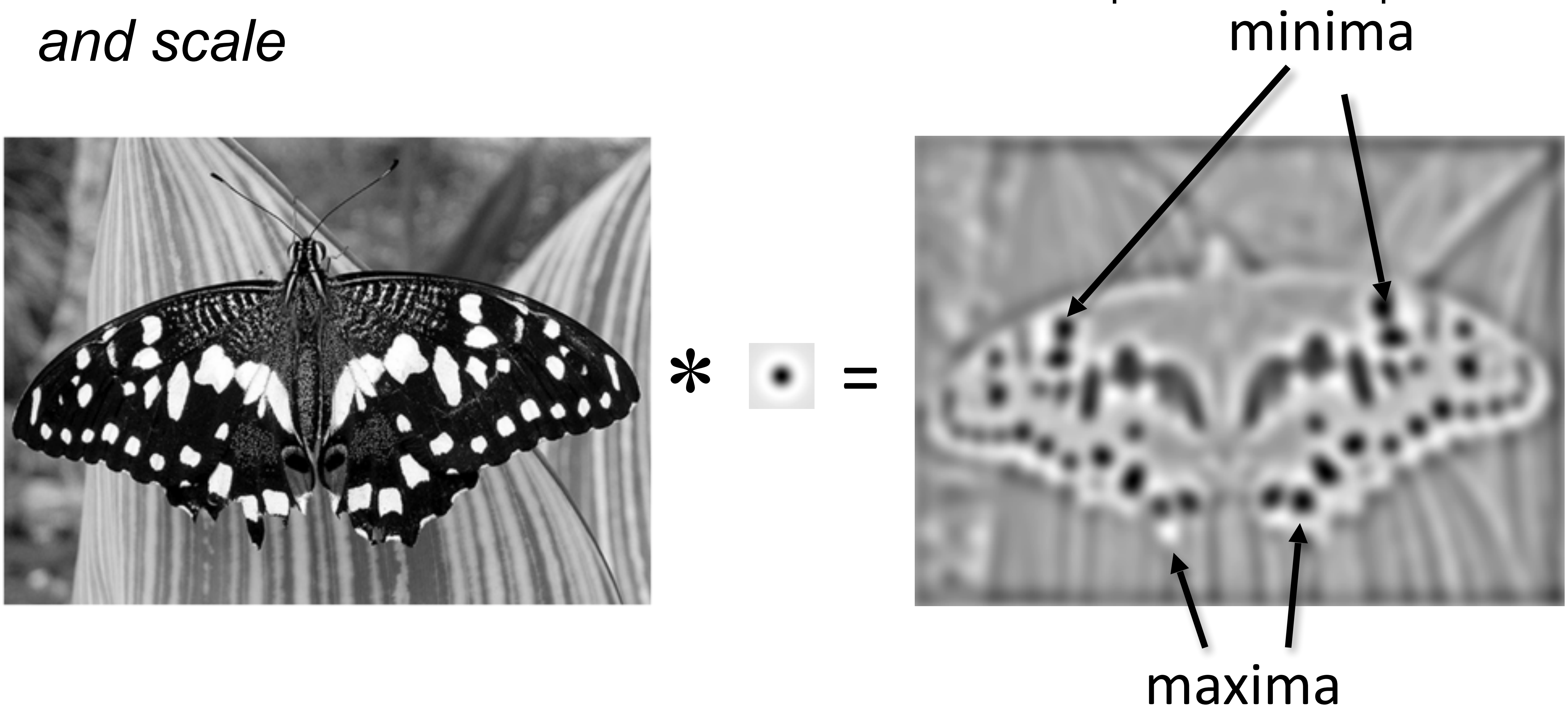




Source: L. Lazebnik

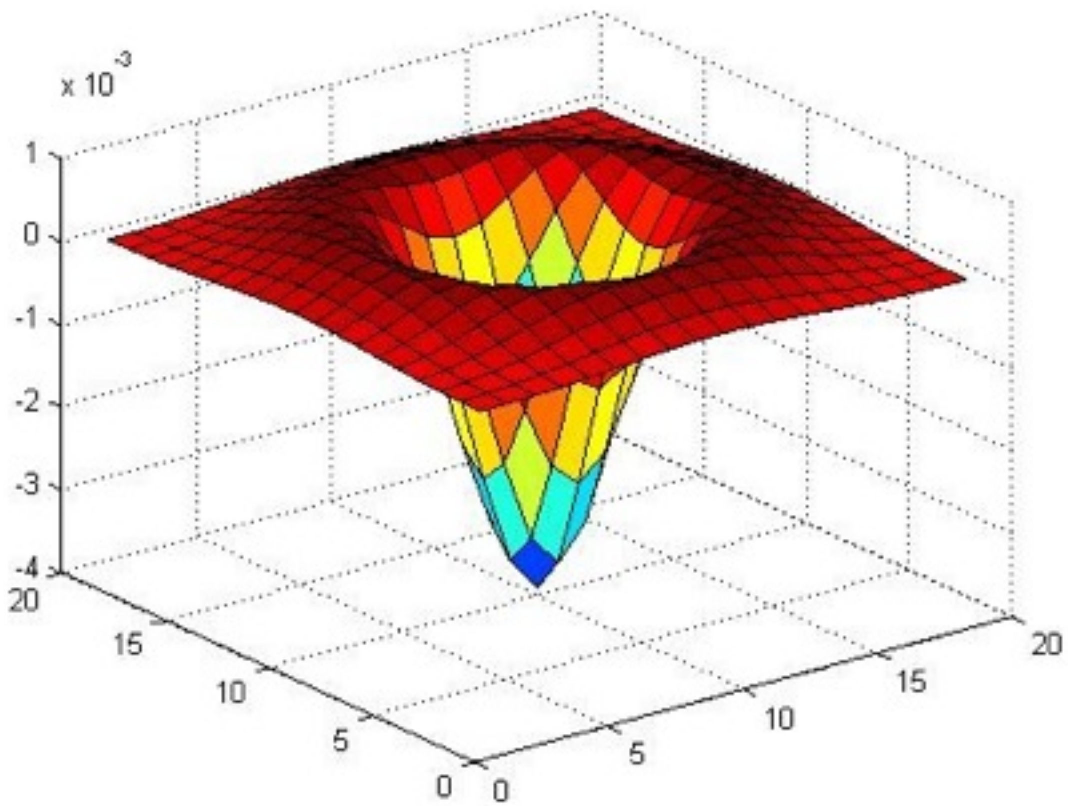
Blob detection: Basic idea

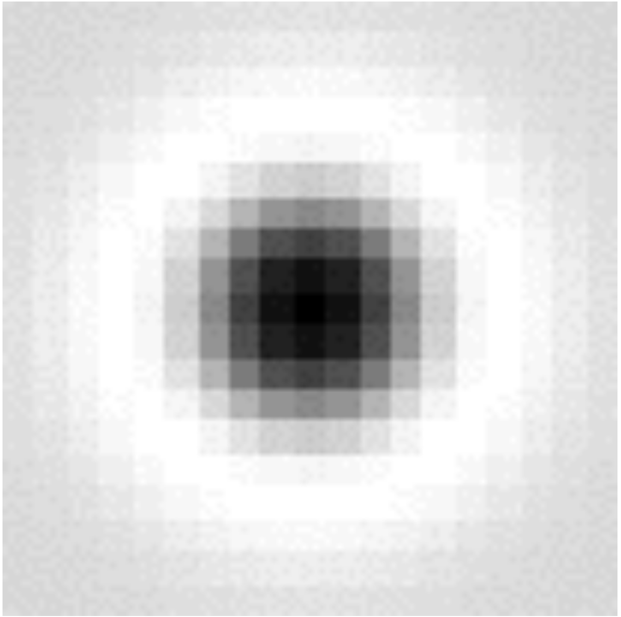
Find maxima *and minima* of blob filter response in space *and scale*



Blob filter

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





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

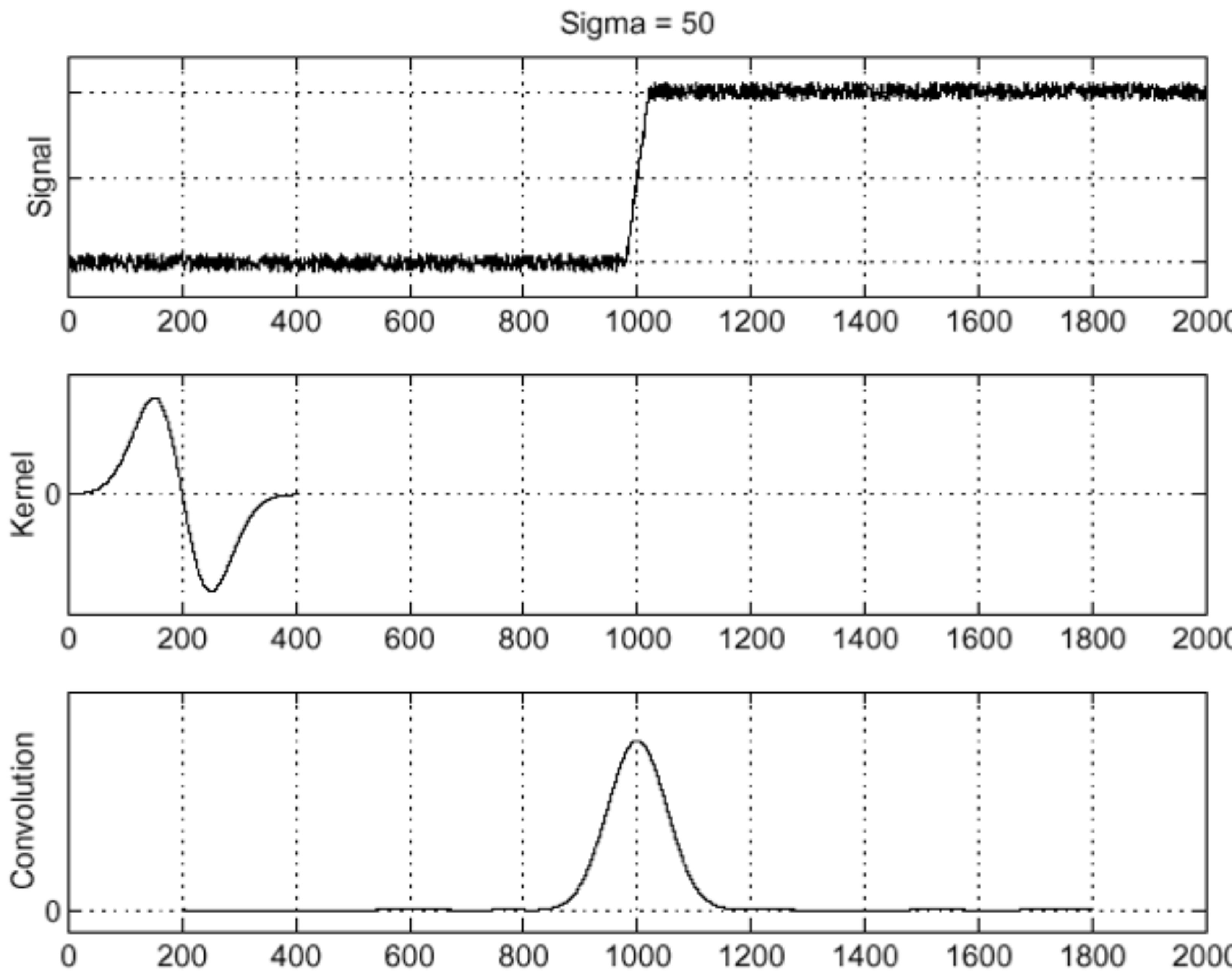
Source: L. Lazebnik

Recall: Edge detection

f

$\frac{d}{dx} g$

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



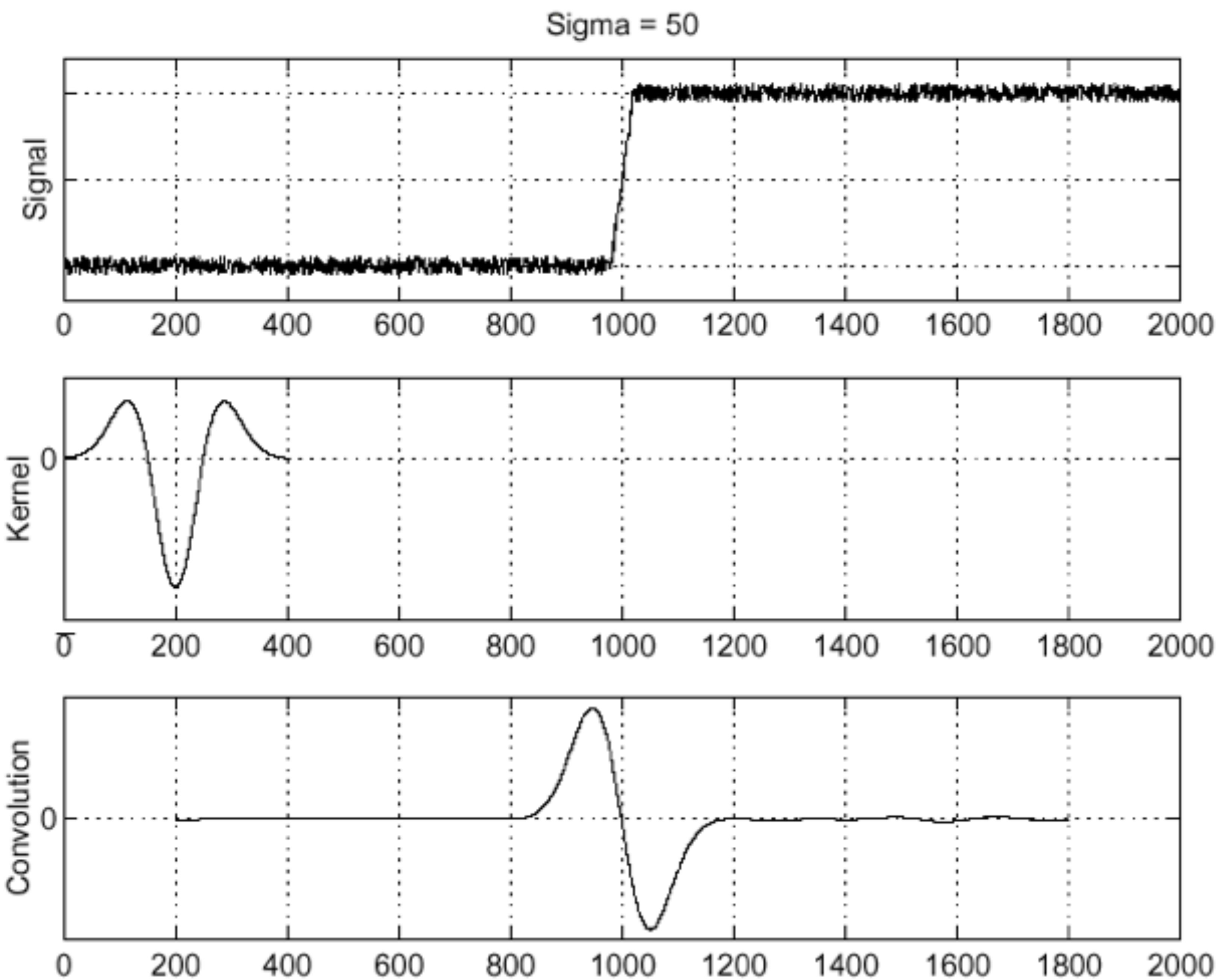
Edge

Derivative of Gaussian

Edge = maximum of derivative

Edge detection, take 2

f



Edge

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

Second derivative of Gaussian (Laplacian)

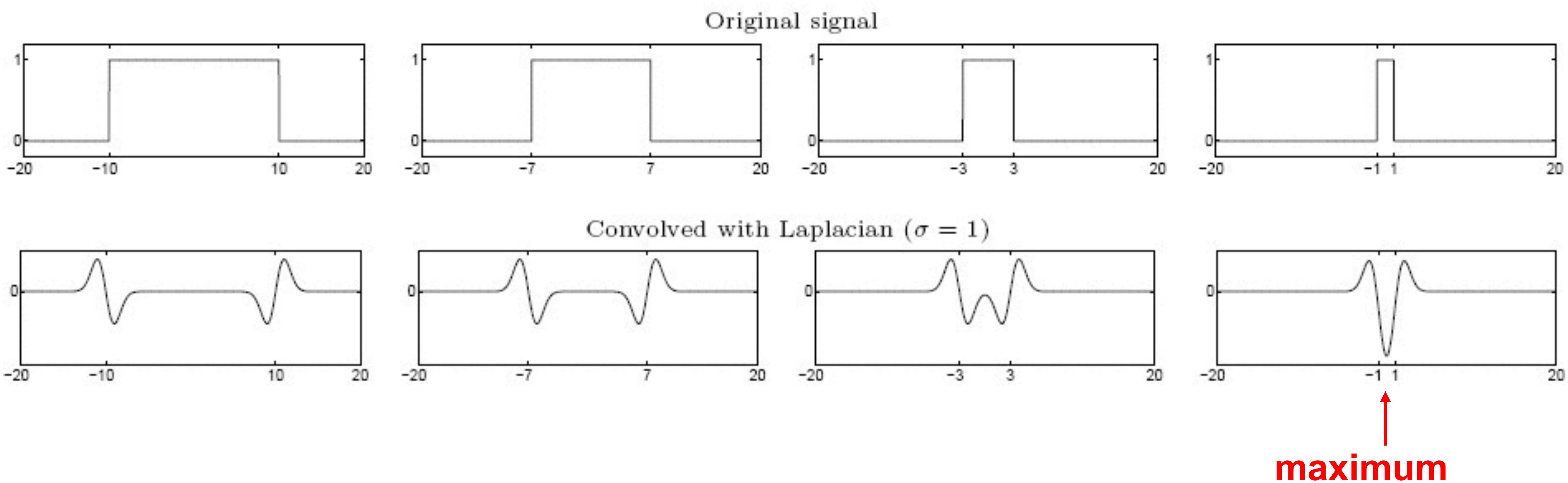
$$f * \frac{d^2}{dx^2} g$$

Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

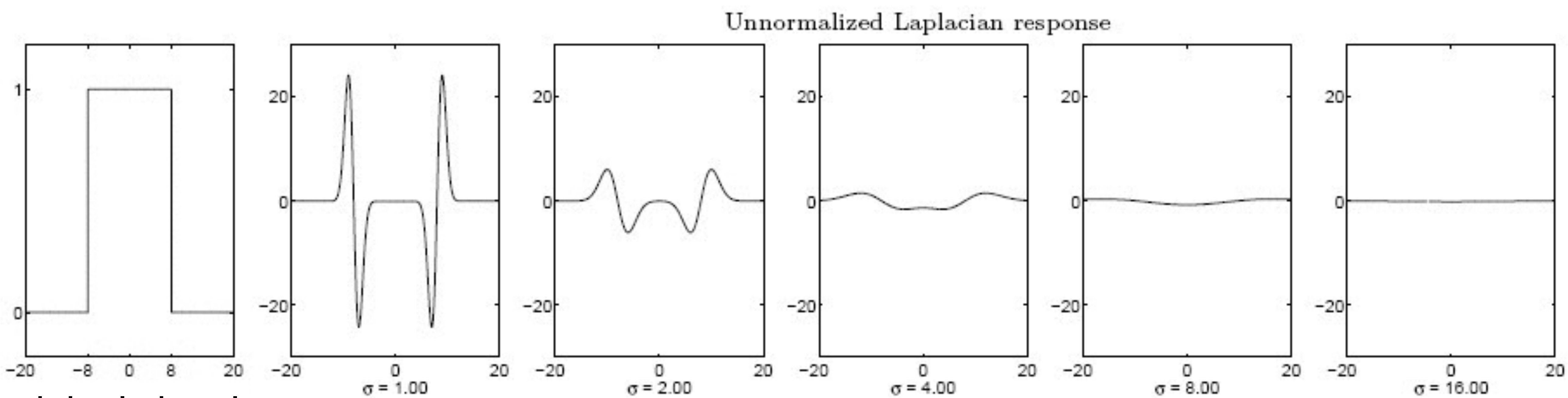


Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

Source: L. Lazebnik

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:



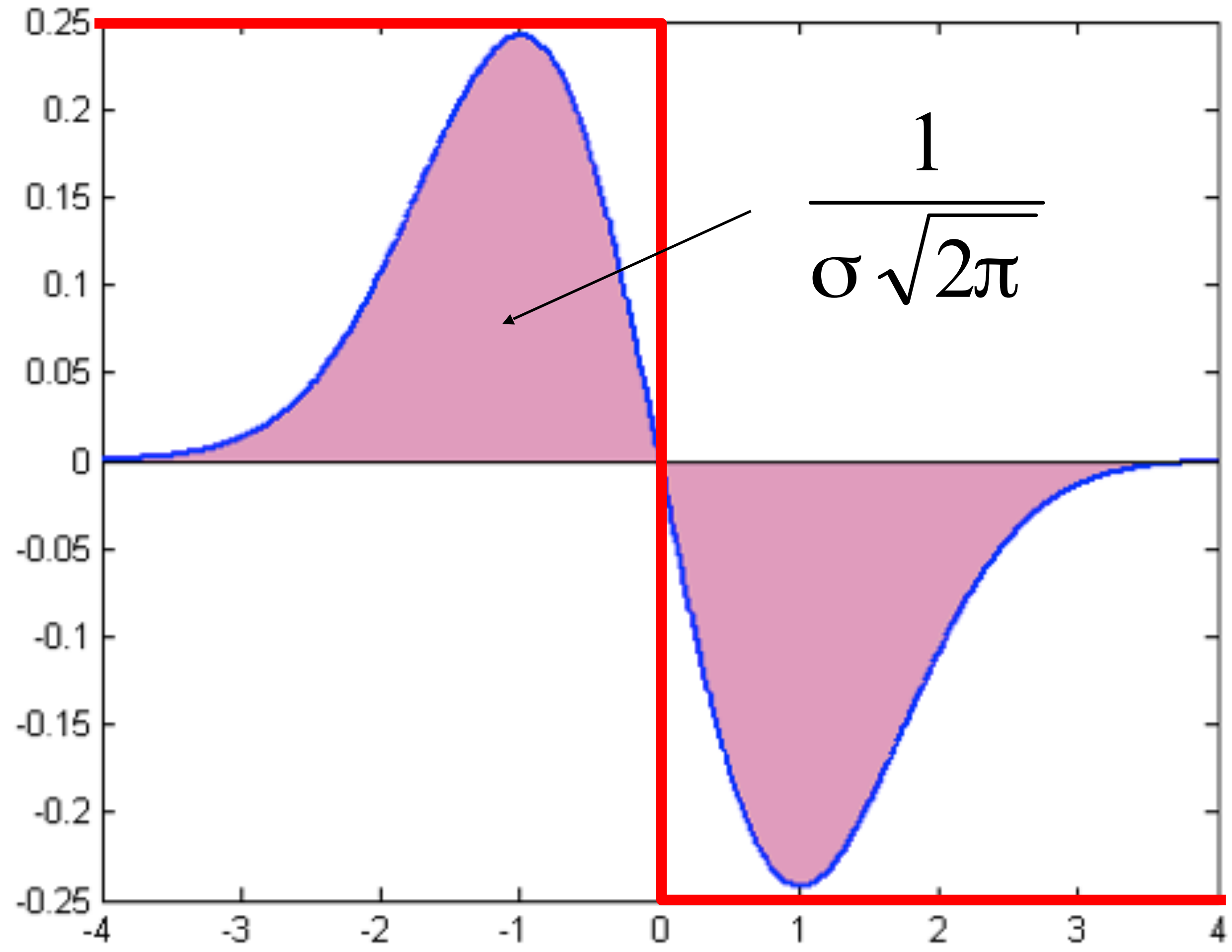
original signal (radius=8)

increasing σ

Source: L. Lazebnik

Scale normalization

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

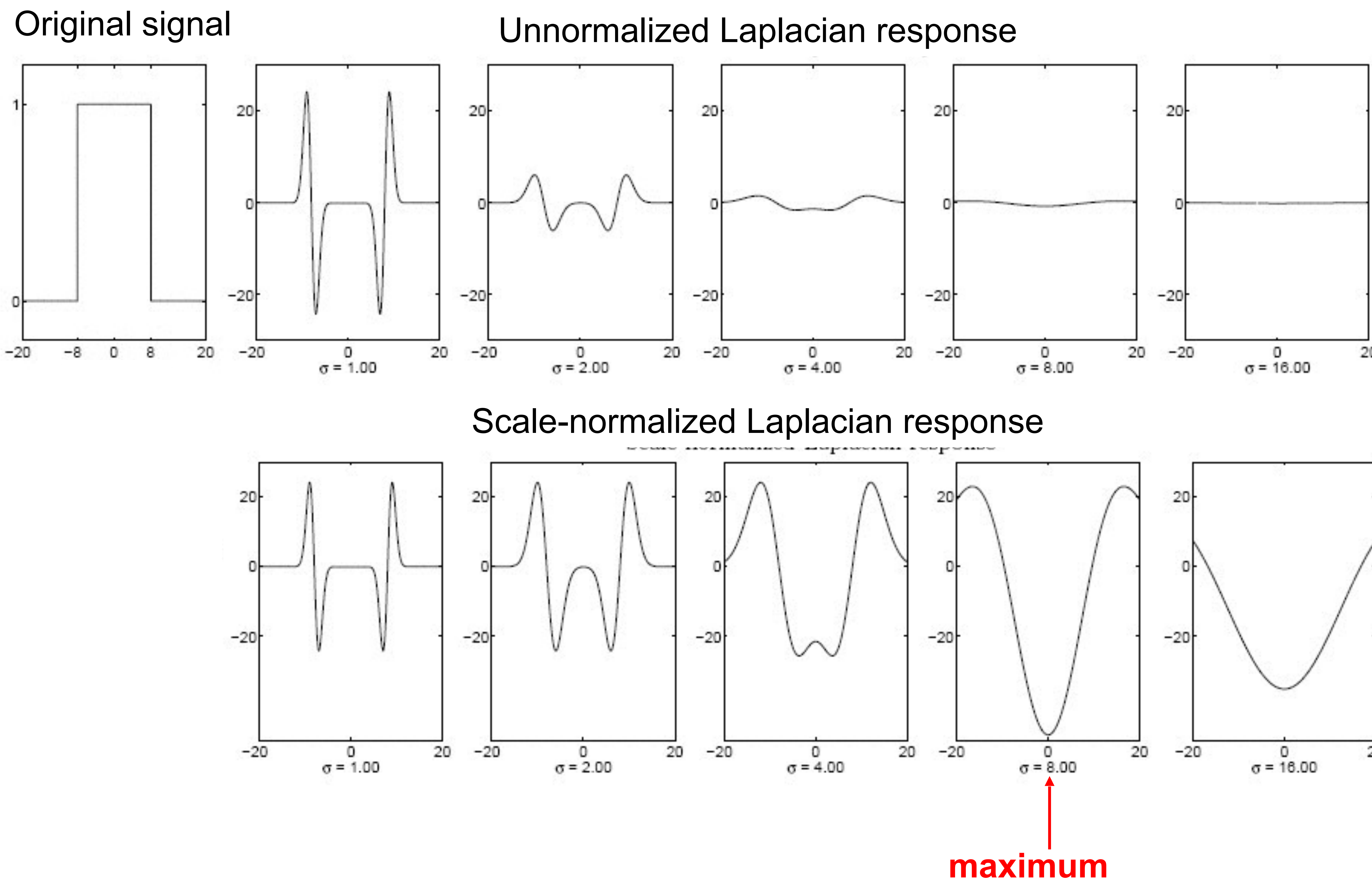


Scale normalization

- 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

Source: L. Lazebnik

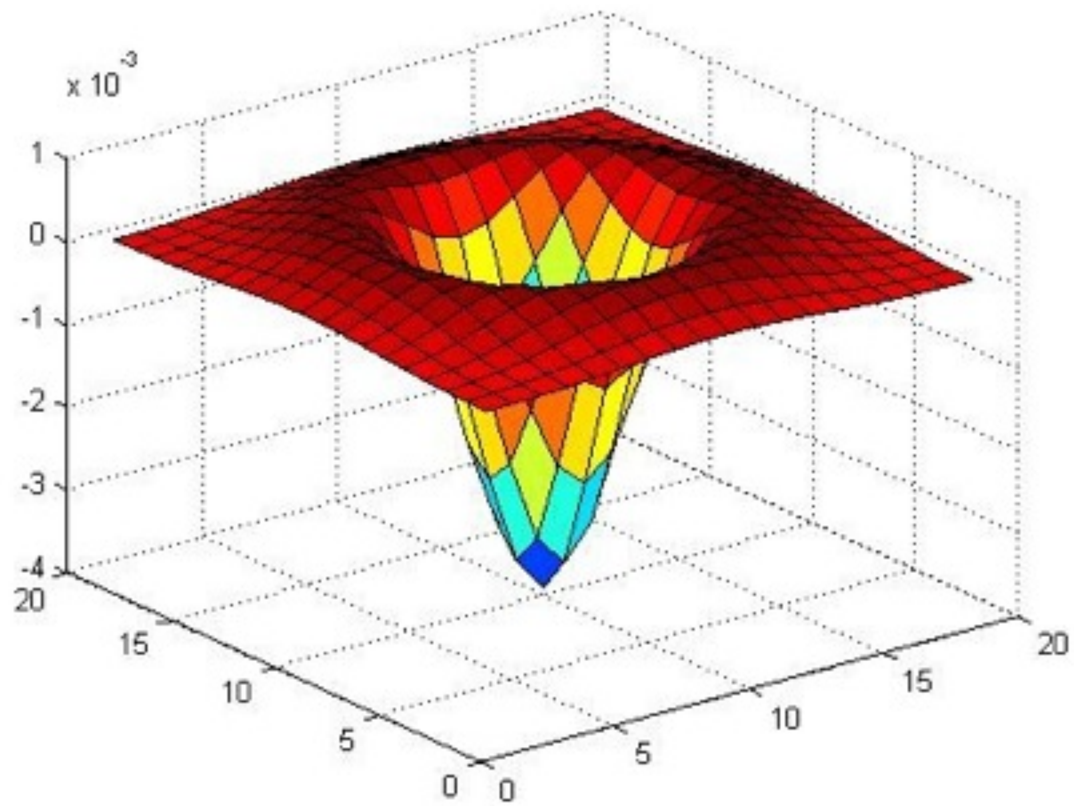
Effect of scale normalization

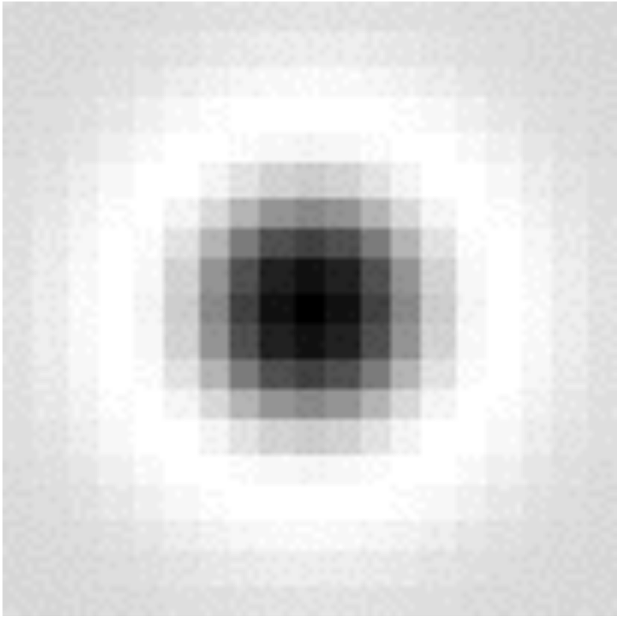


Source: L. Lazebnik

Blob detection in 2D

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



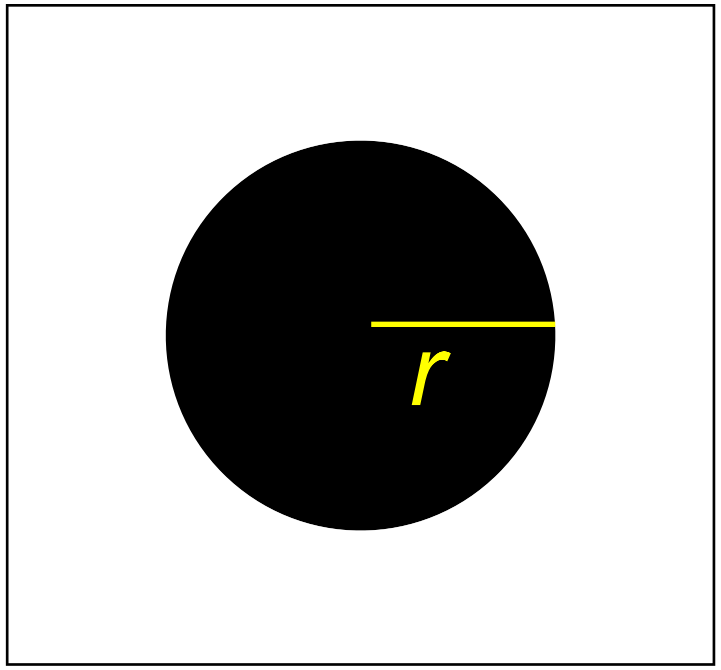


Scale-normalized: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

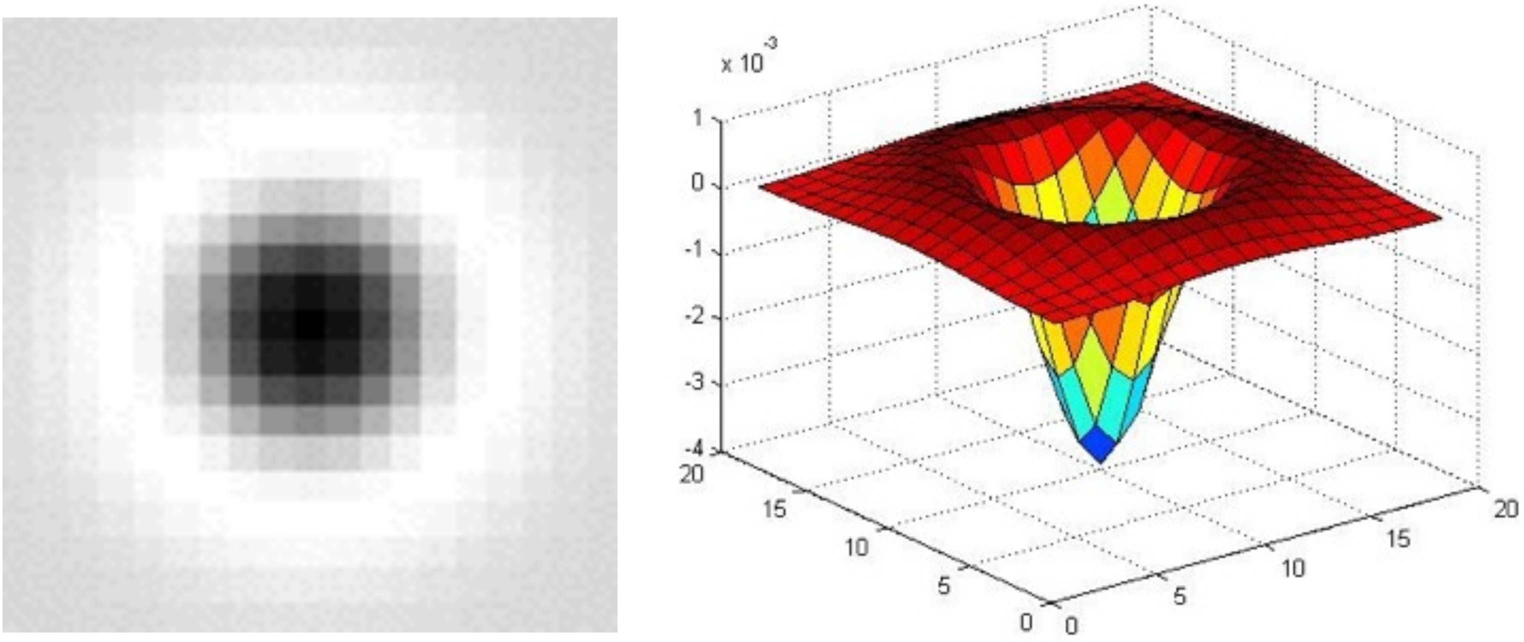
Source: L. Lazebnik

Scale selection

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



image



Laplacian

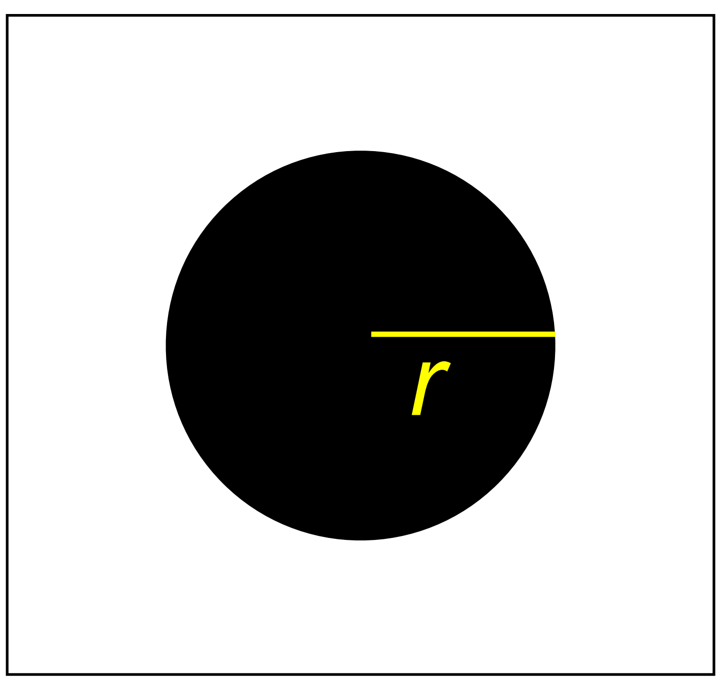
Source: L. Lazebnik

Scale selection

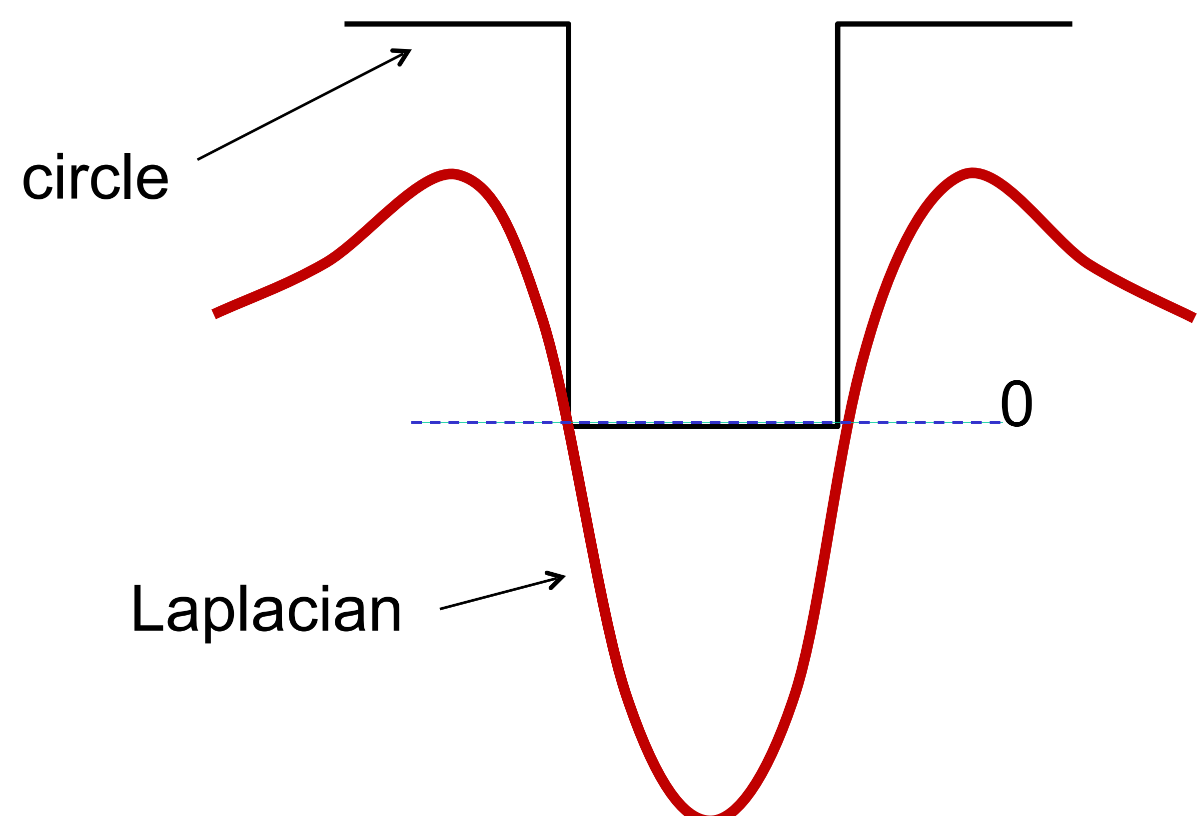
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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



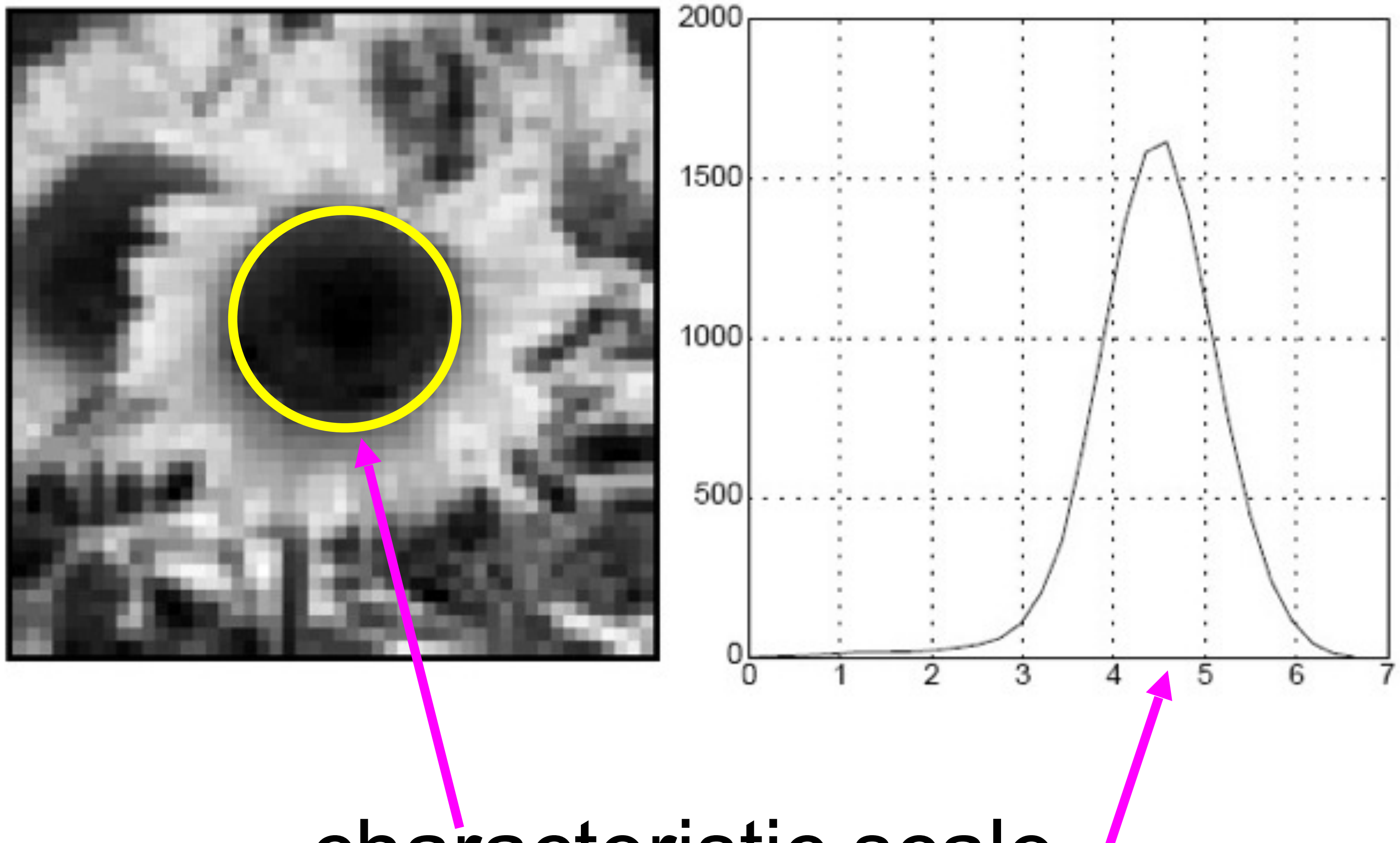
image



Source: L. Lazebnik

Characteristic scale

- We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center



characteristic scale

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

Source: L. Lazebnik

Scale-space blob detector

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

Source: L. Lazebnik

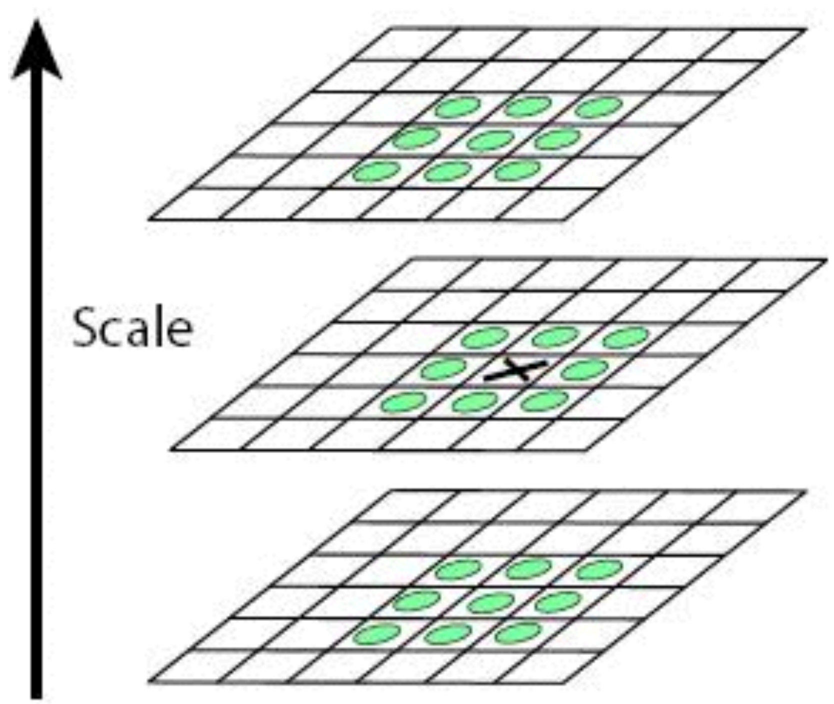
Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

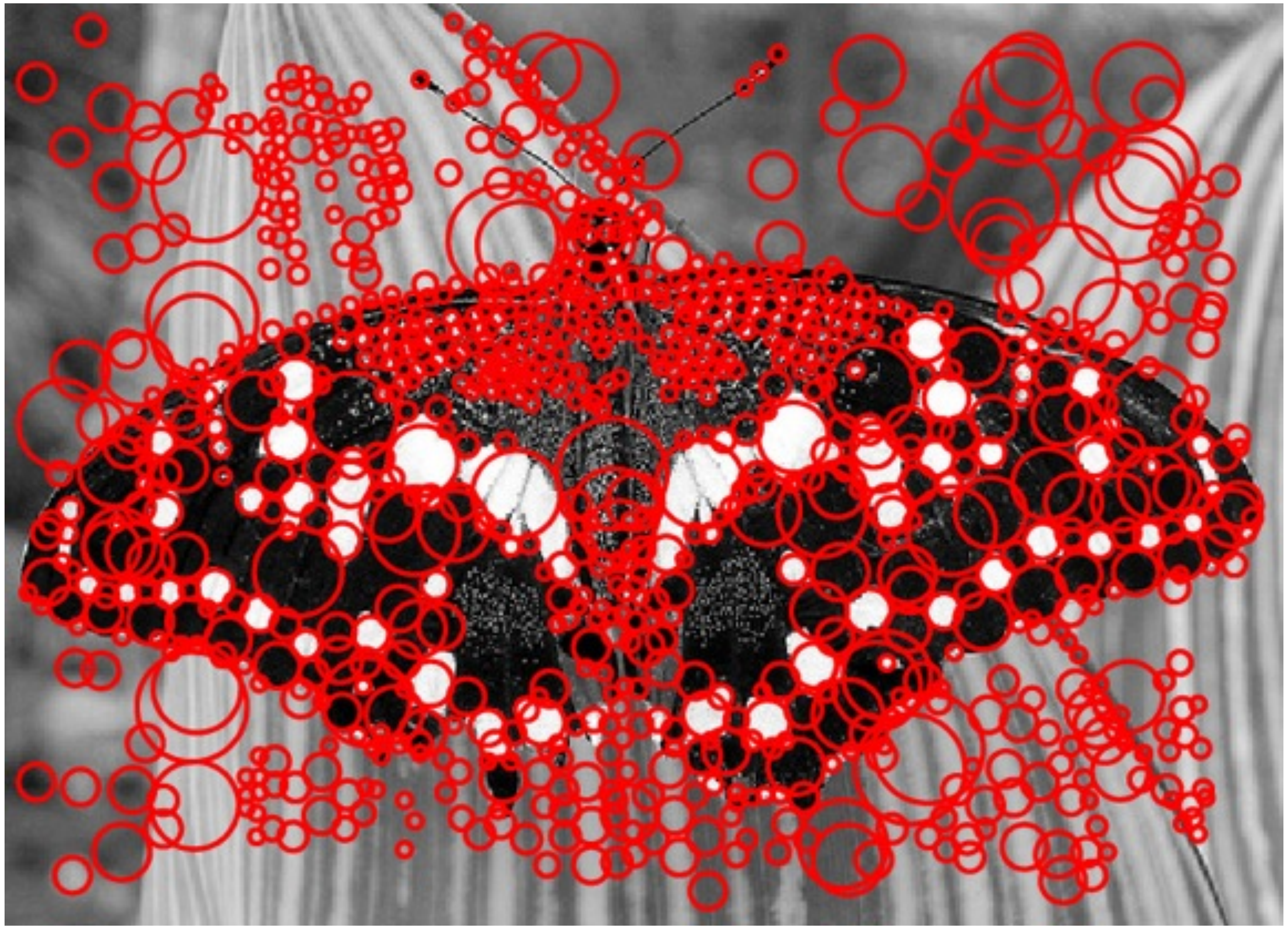
Scale-space blob detector

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



Source: L. Lazebnik

Scale-space blob detector: Example



Source: L. Lazebnik

Efficient implementation

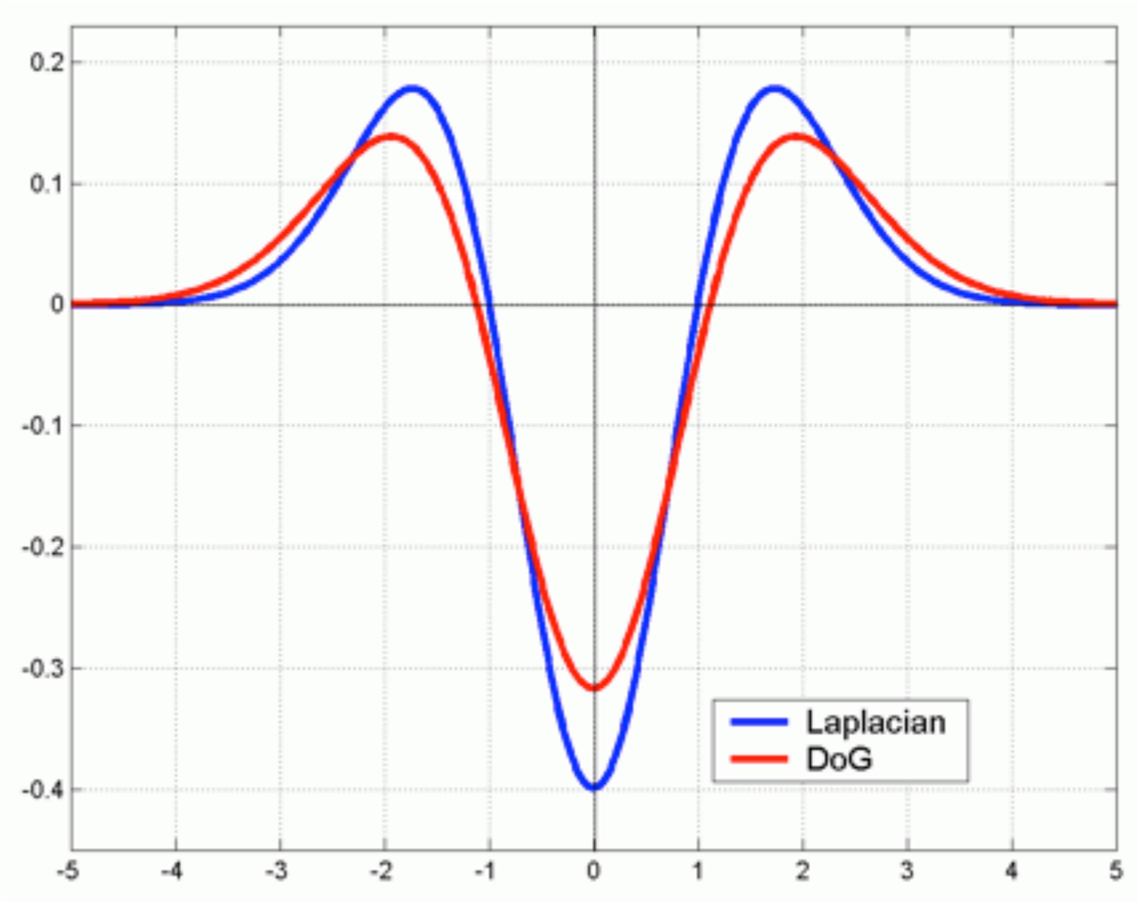
- Approximating the Laplacian with a difference of Gaussians:

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

(Laplacian)

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

(Difference of Gaussians)

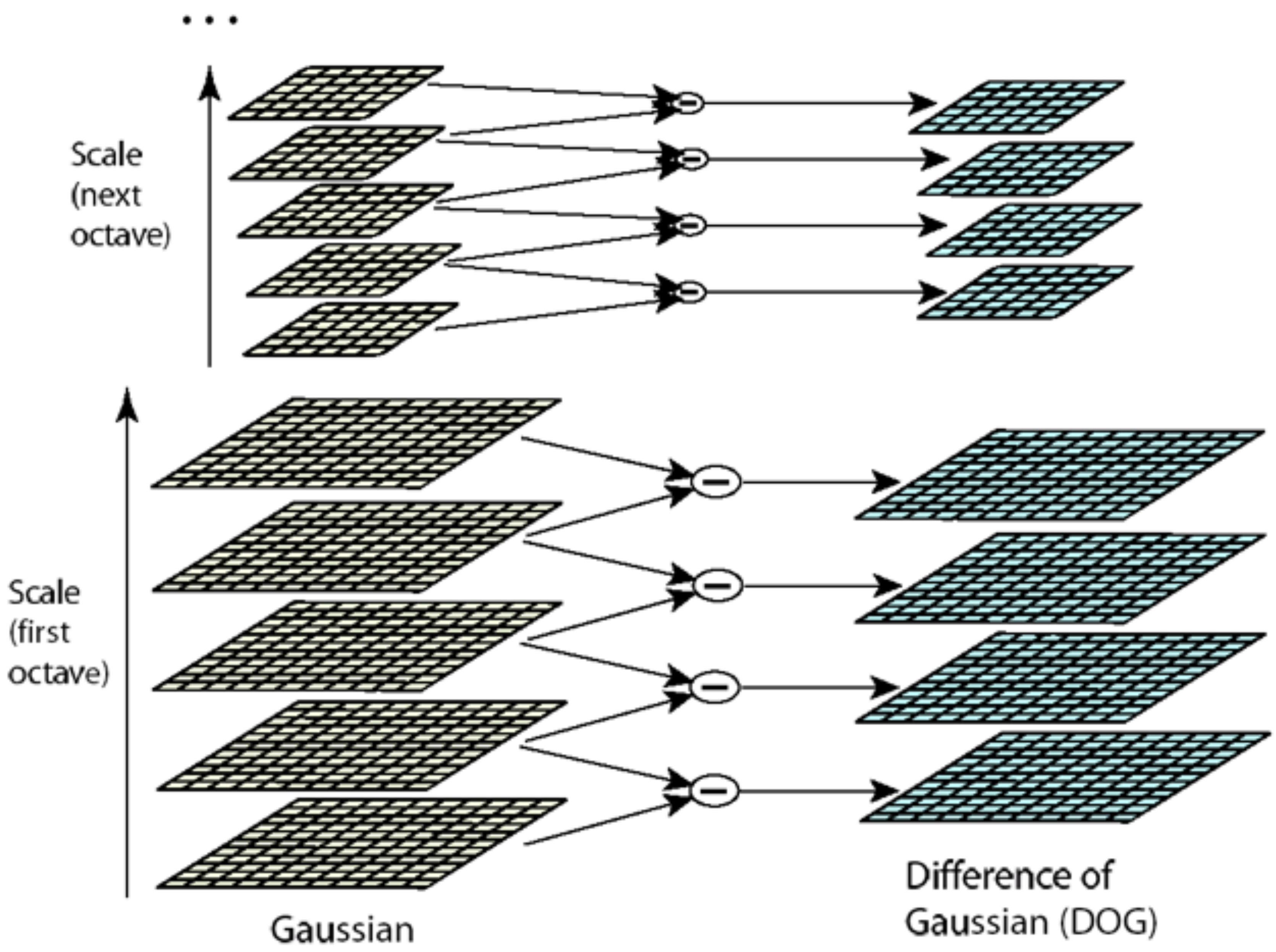


$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2 + y^2)/2\sigma^2}$$

Is the Laplacian separable?

Source: L. Lazebnik

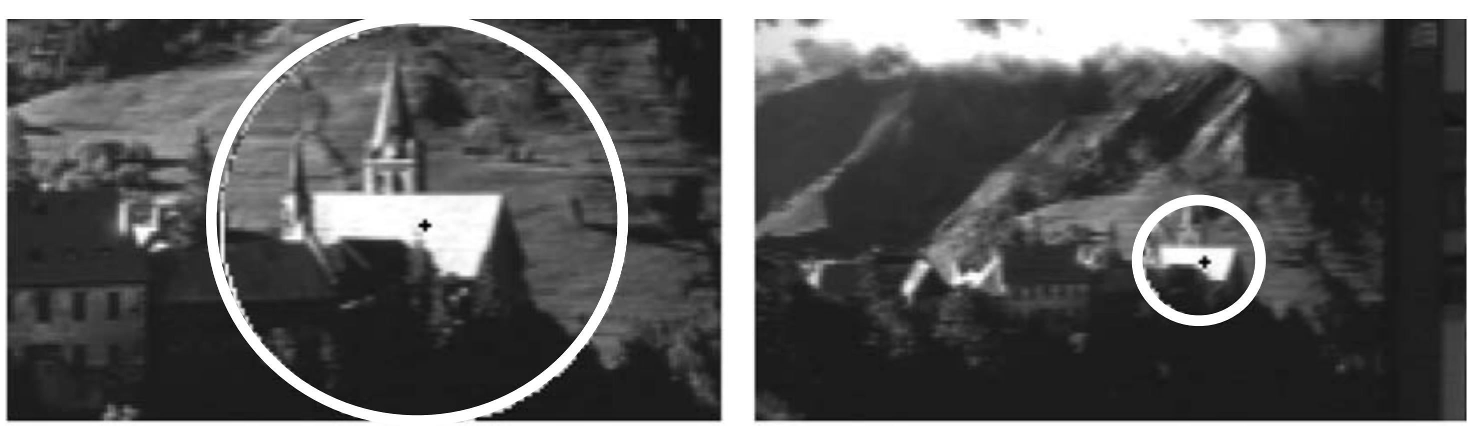
Efficient implementation

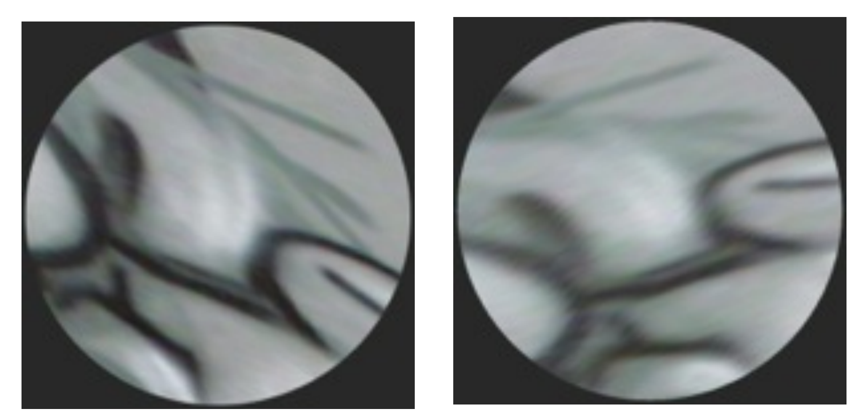


David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

From feature detection to description

- Scaled and rotated versions of the same neighborhood will give rise to blobs that are related by the same transformation
- What to do if we want to compare the appearance of these image regions?
  - **Normalization**: transform these regions into same-size circles
  - Problem: rotational ambiguity

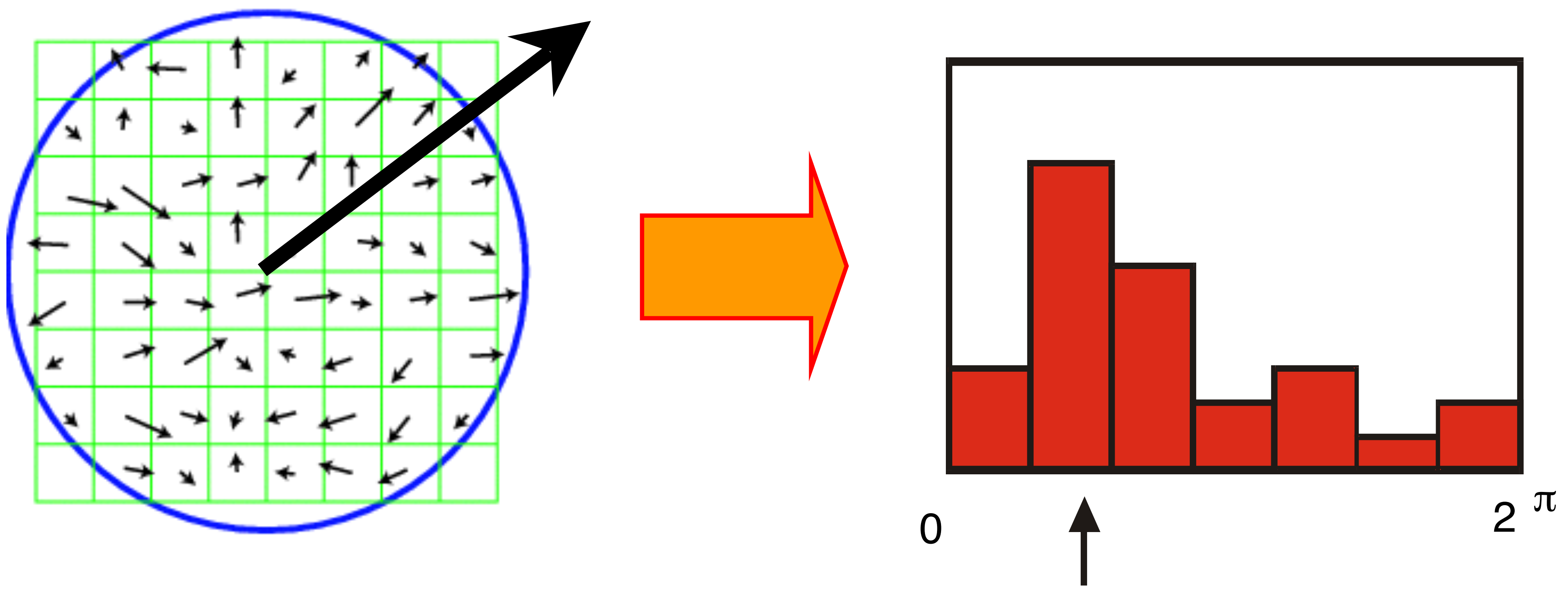




Source: L. Lazebnik

Eliminating rotation ambiguity

- To assign a unique orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram



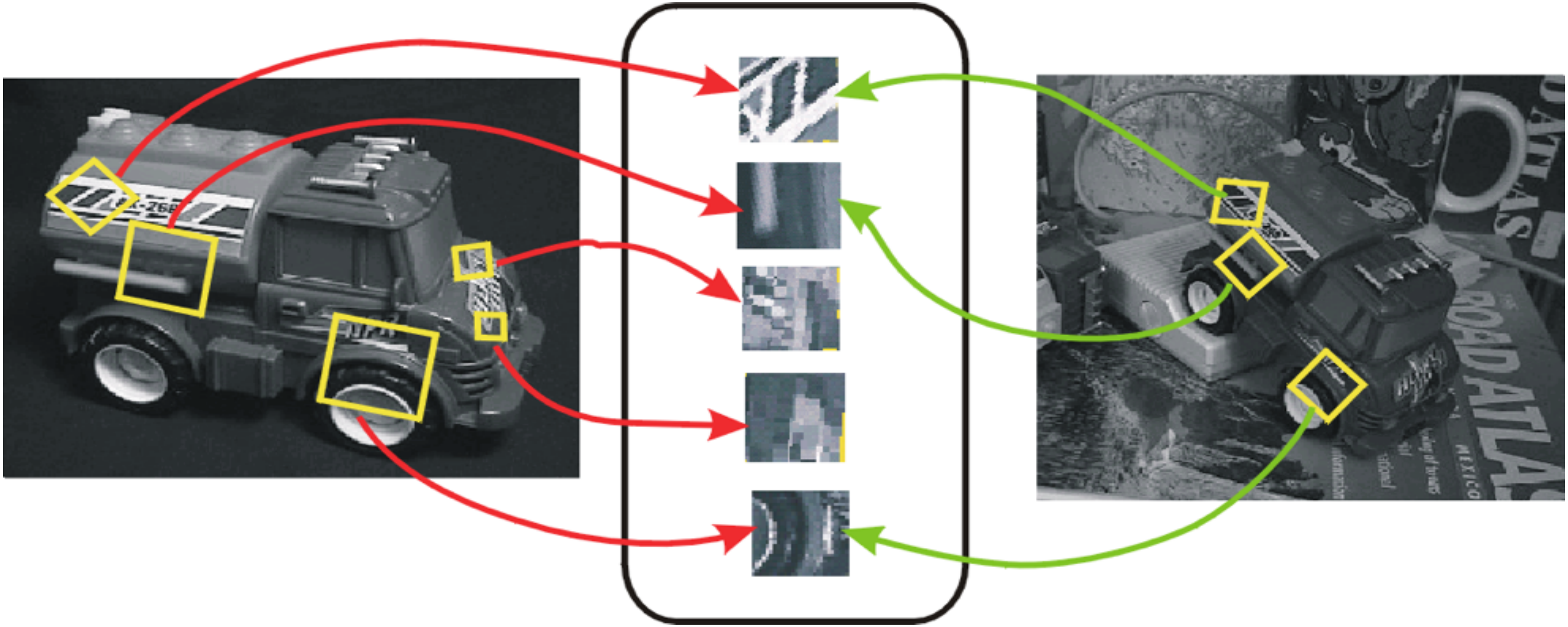
Source: L. Lazebnik

SIFT features

- Detected features with characteristic scales and orientations:

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

From feature detection to description



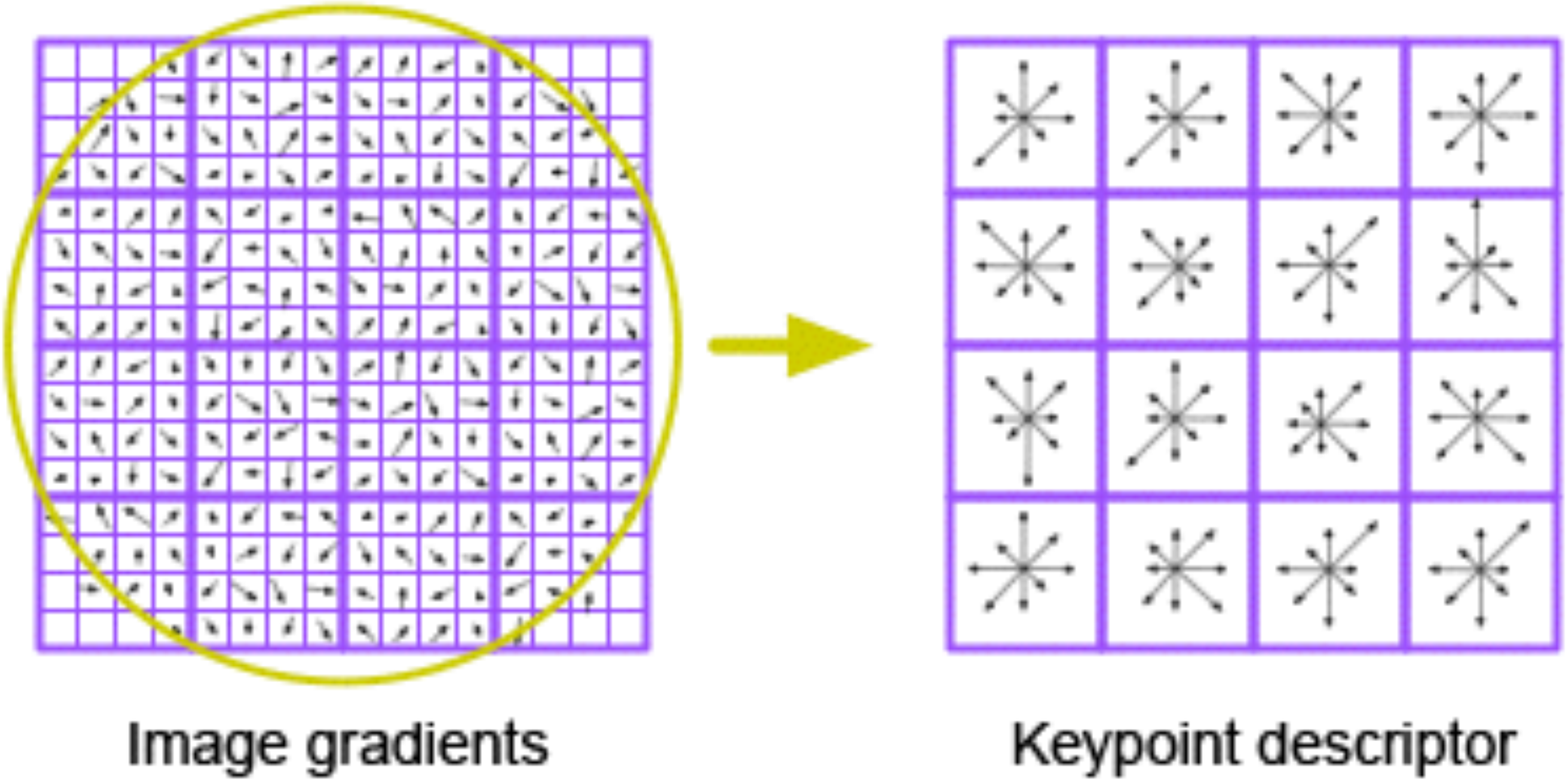
Detection is *covariant*:

features(transform(image)) = transform(features(image))

Description is *invariant*:

features(transform(image)) = features(image)

SIFT descriptors



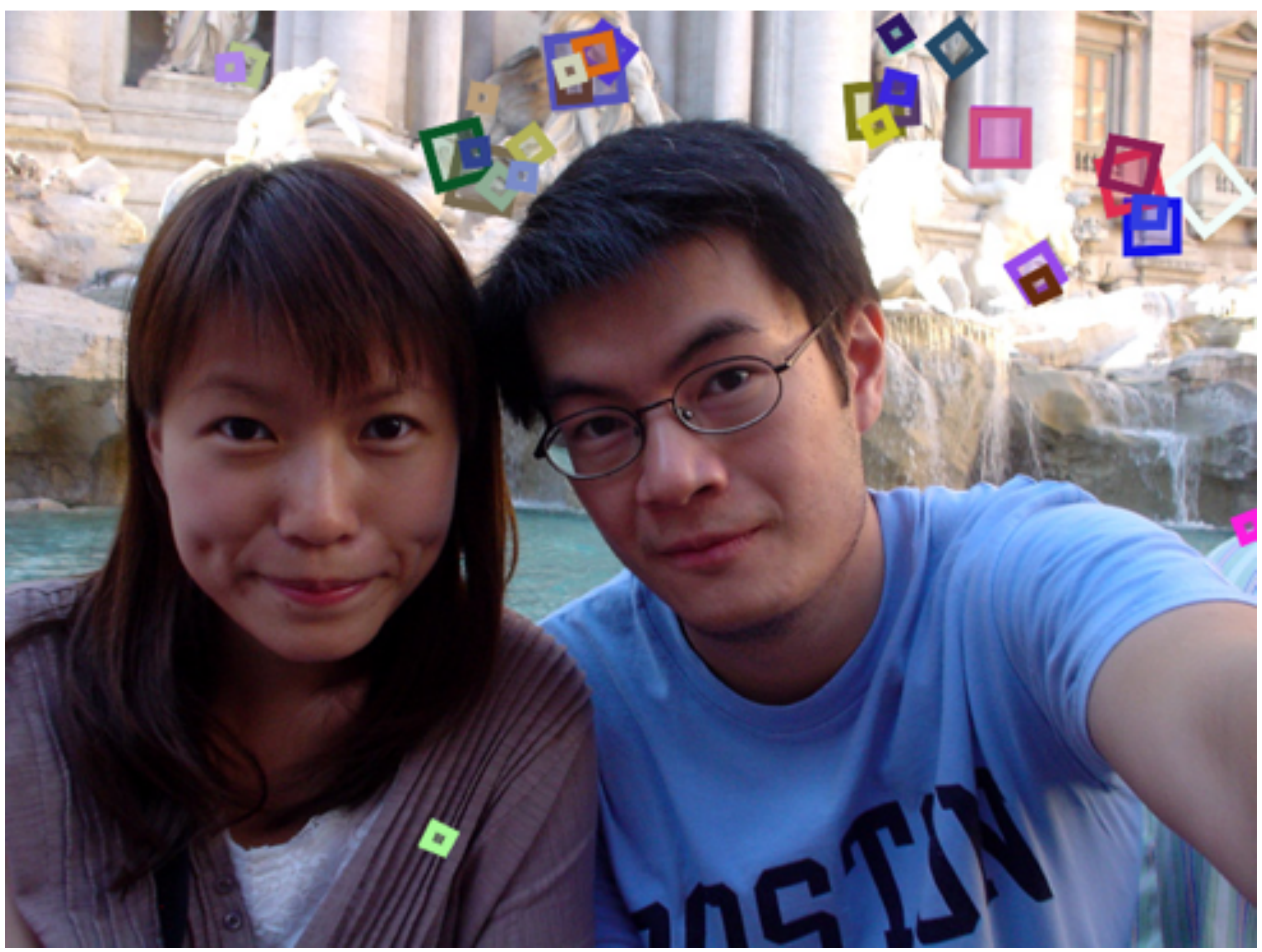
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Source: L. Lazebnik

Properties of SIFT

Extraordinarily robust detection and description technique

- Can handle changes in viewpoint
  - Up to about 60 degree out-of-plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night
- Fast and efficient—can run in real time
- Lots of code available



Source: N. Snavely

Affine adaptation

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

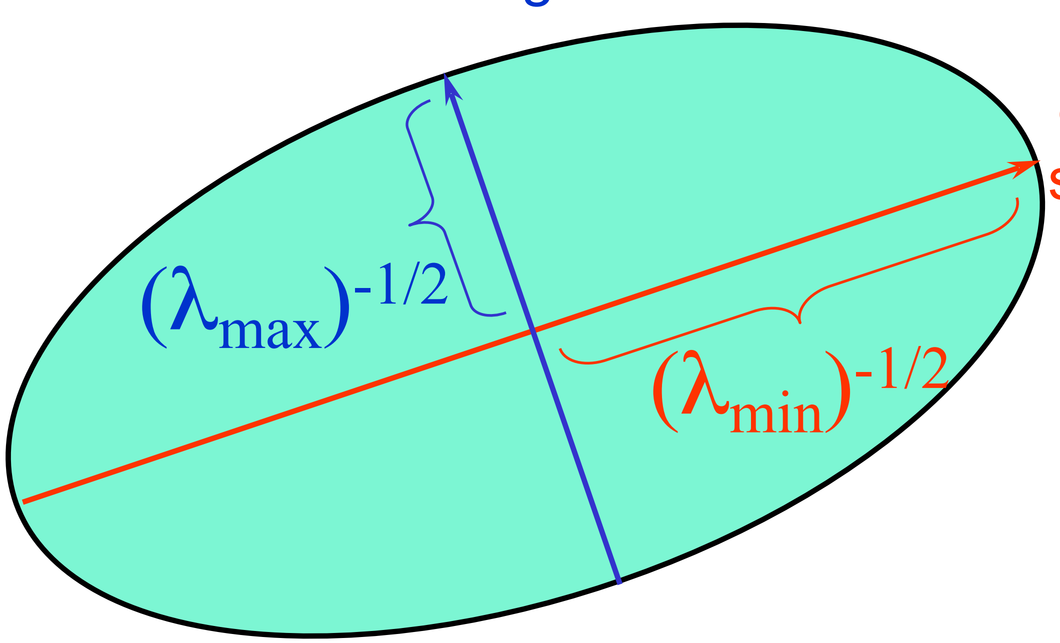
Affine adaptation

Consider the second moment matrix of the window containing the blob:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

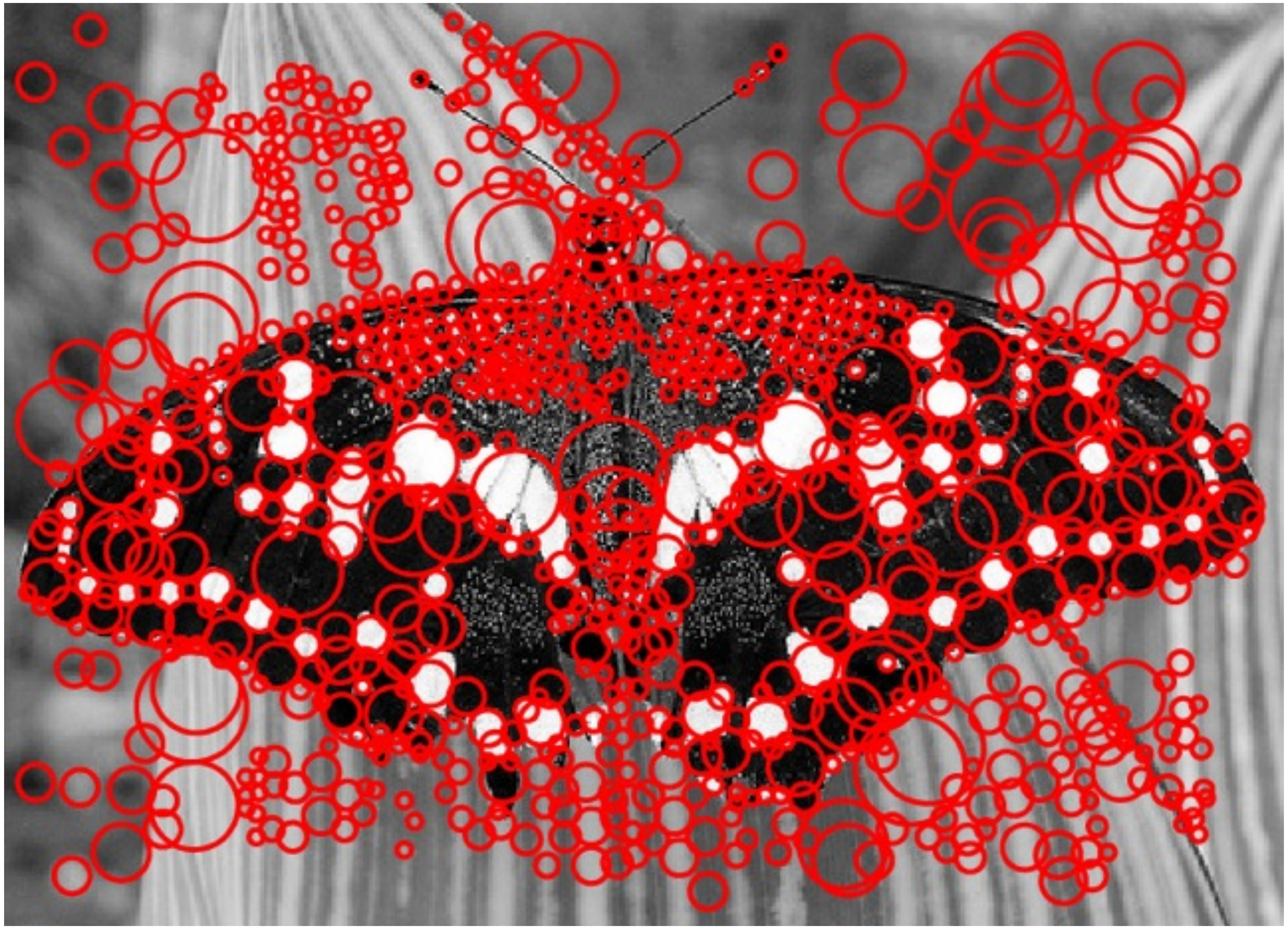
Recall:

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



This ellipse visualizes the “characteristic shape” of the window

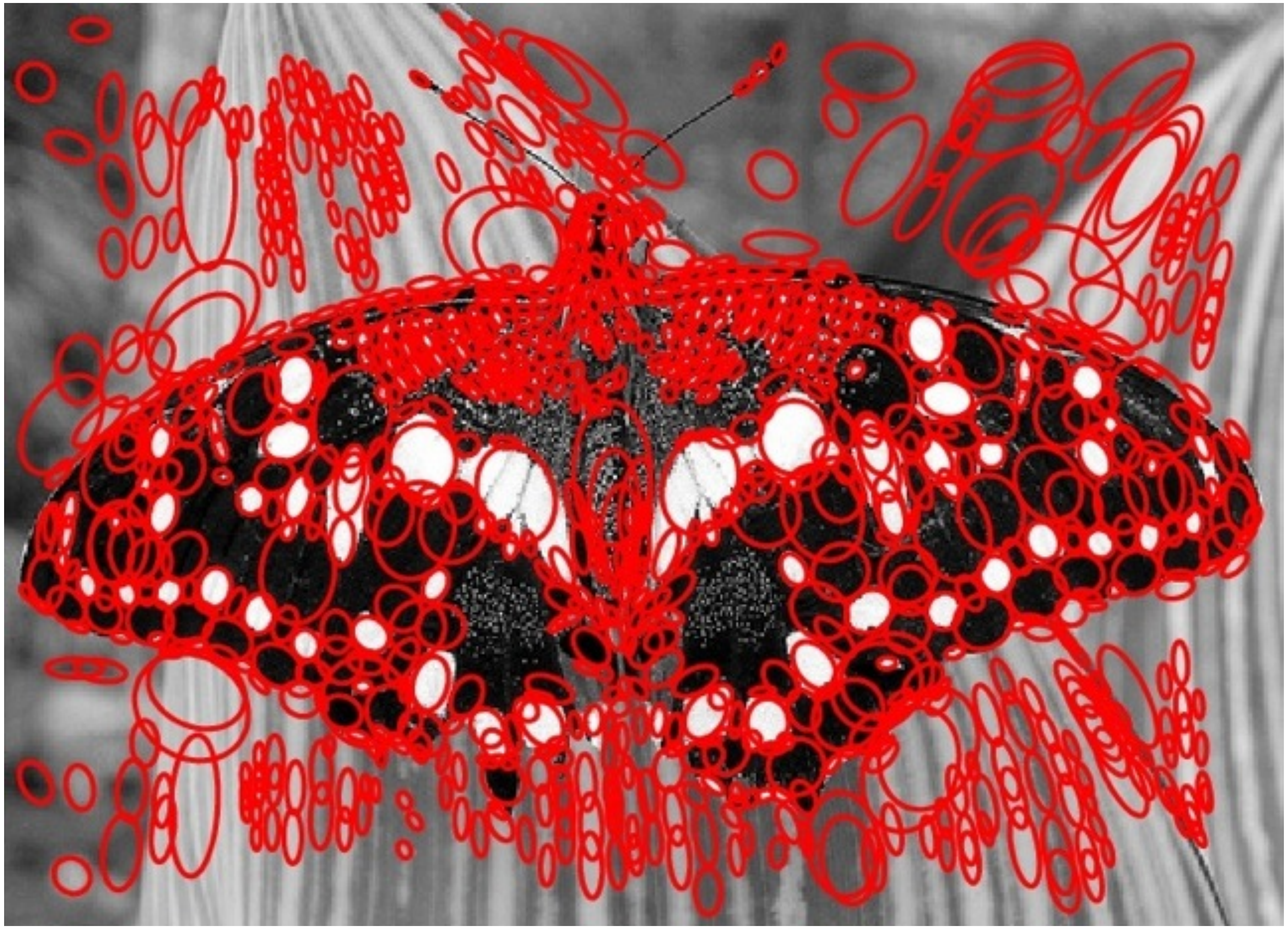
Affine adaptation example



Scale-invariant regions (blobs)

Source: L. Lazebnik

Affine adaptation example



Affine-adapted blobs

Further readings and thoughts ...

- More about scale-space
  - T. Lindeberg, **Scale-space theory: A basic tool for analyzing structures at different scales**, Journal of Applied Statistics, 1994
- SIFT descriptor in detail
  - David G. Lowe, **Distinctive Image Features from Scale-Invariant Keypoints**, IJCV 2004
- How good are local point detectors and descriptors?
  - K. Mikolajczyk, C. Schmid, **A performance evaluation of local descriptors**, IEEE PAMI 2005
- Chapter 4, R. Szeliski's book