Scale Invariant Feature Transform
SIFT Descriptor

SIFT citation

- Cited 34442 (google scholar, 30.3.2016)
- Cited 34663 (google scholar, 14.4.2016)

David Lowe - Google Schol...

scholar.google.ca/citations?user=8vs5HGYAAAAJ&hl=en

Search

Web Images More... Sign in

Google Scholar

Follow

David Lowe

Professor of Computer Science,
University of British Columbia
Computer Vision, Object Recognition
Verified email at cs.ubc.ca - Homepage

Citation indices	All	Since 2011
Citations	65427	38953
h-index	47	36
i10-index	81	53

Co-authors View all...

James Little, Marius Muja, Matthew Brown, Sancho Mc

Cited 34442 (google scholar, 30.3.2016)

Title 1–20	Cited by	Year
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	34442	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	10914	1999
Perceptual Organization and Visual Recognition DG Lowe Kluwer Academic Publishers, Boston	1628 *	1985
Three-dimensional object recognition from single two-dimensional images		

David Lowe - Google Schol...

scholar.google.ca/citations?user=8vs5HGYAAAAJ&hl=en

Web Images More... Sign in

Google Scholar

Follow

David Lowe

Professor of Computer Science, University of British Columbia

Computer Vision, Object Recognition

Verified email at cs.ubc.ca - Homepage

Citation indices	All	Since 2011
Citations	65798	39308
h-index	47	36
i10-index	82	53

Co-authors View all...

James Little, Marius Muja, Matthew Brown, Sanch

Title 1–20	Cited by	Year
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	34663	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	10995	1999
Perceptual Organization and Visual Recognition DG Lowe Kluwer Academic Publishers, Boston	1629 *	1985
Three-dimensional object recognition from single two-dimensional images DG Lowe	1551	1987

May 2020

Google Scholar

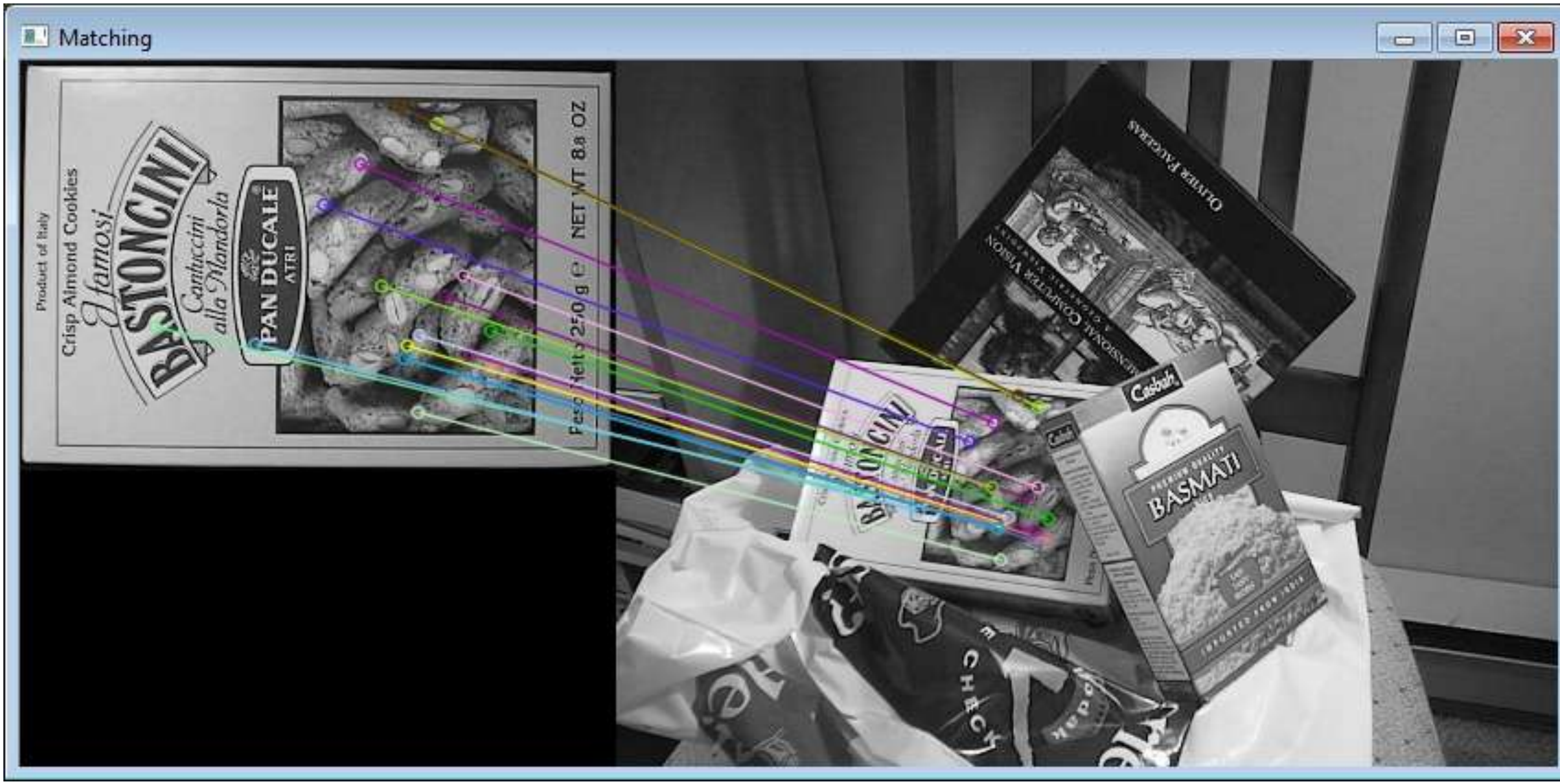
David Lowe

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

FOLLOW

ARTICLES   CITED BY   CO-AUTHORS

TITLE	CITED BY	YEAR
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	56478	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	18910	1999
Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.	3021	2009


Matching
Product of Italy
Crisp Almond Cookies
Famosi
BASTONCINI
Cantuccini
alla Mandorla
PAN DUCALE
ATRI
NET WT 8.8 OZ
Casbah
BASMATI

Outline

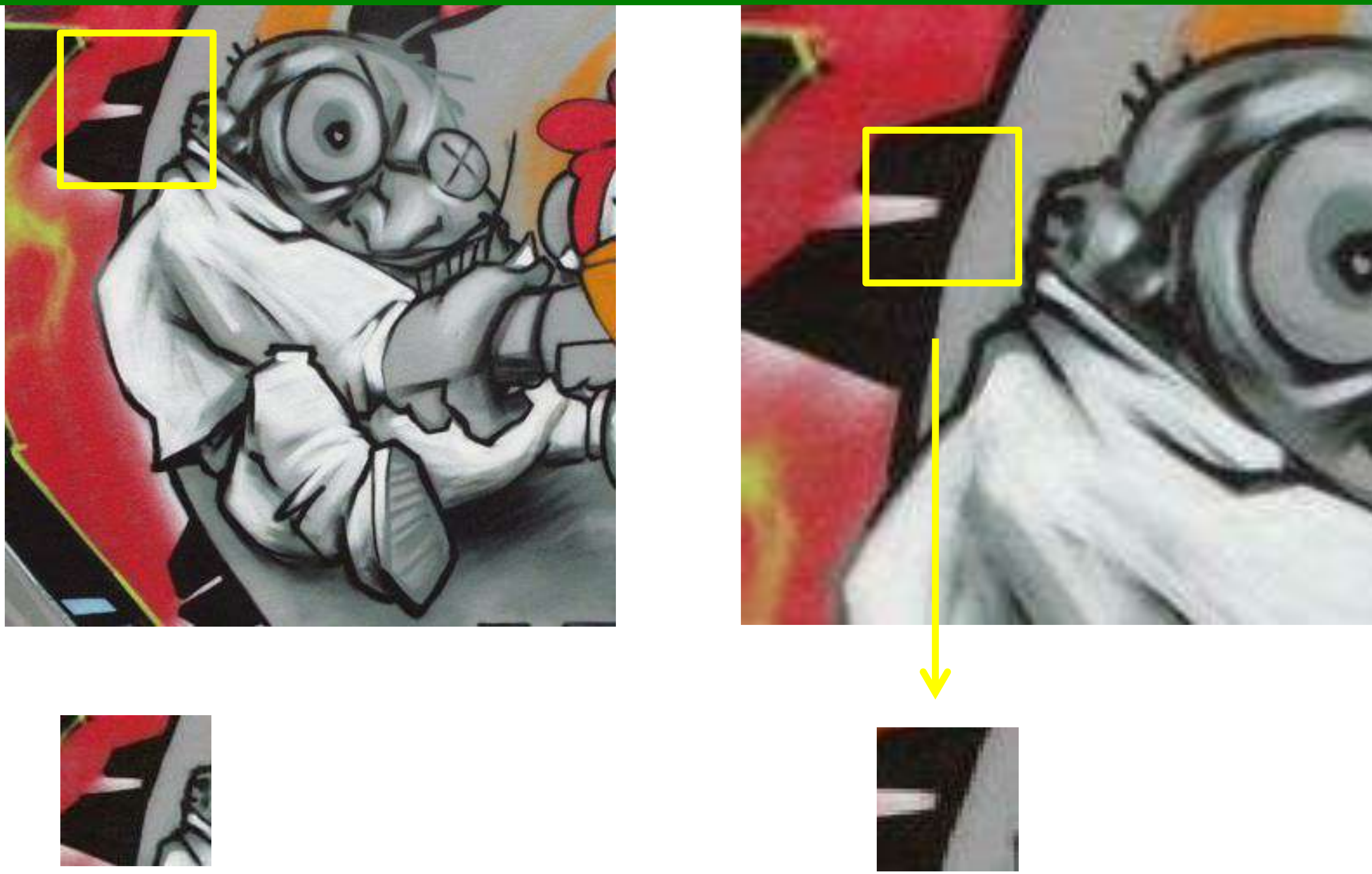
- Automatic scale selection
- SIFT Descriptor

Local Descriptors

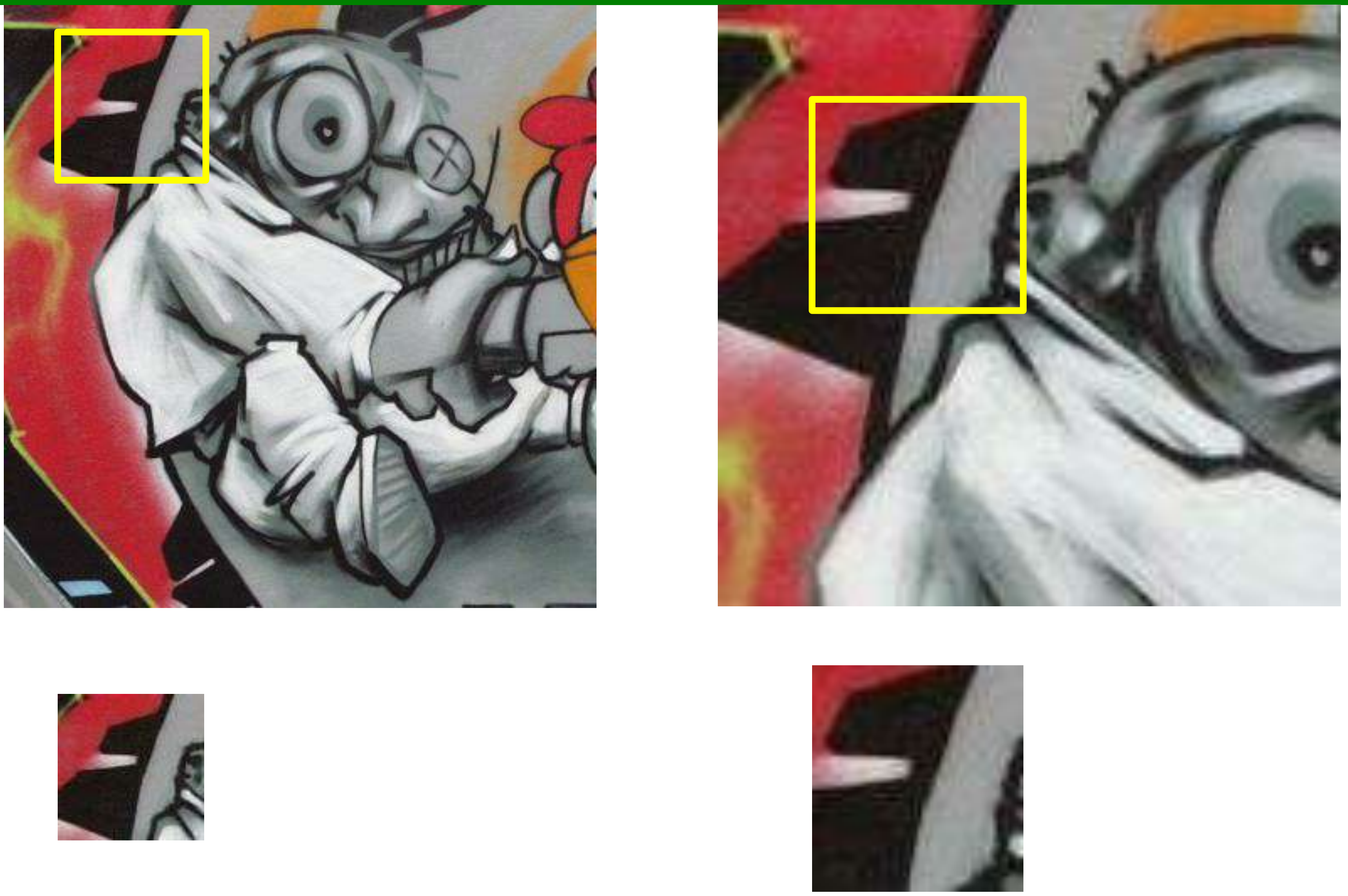
- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient

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

Exhaustive search (Multi-scale approach)

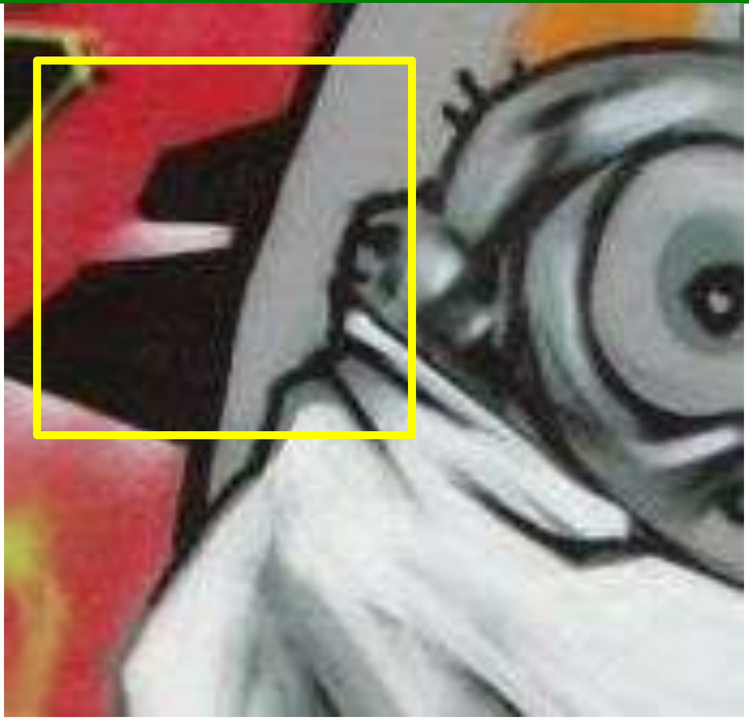


Exhaustive search (Multi-scale approach)

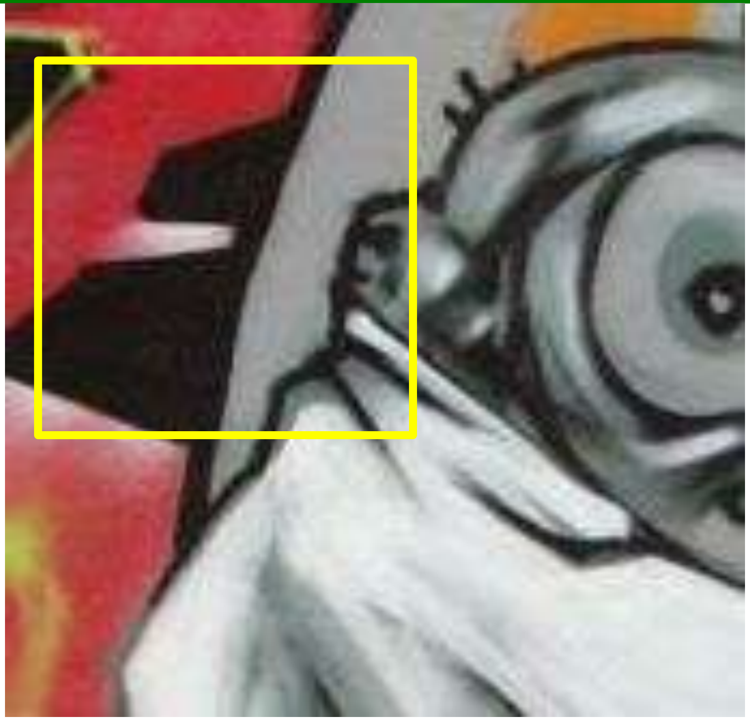


Exhaustive search (Multi-scale approach)

Exhaustive search (Multi-scale approach)



Invariance
Extract patch frrom each image individually

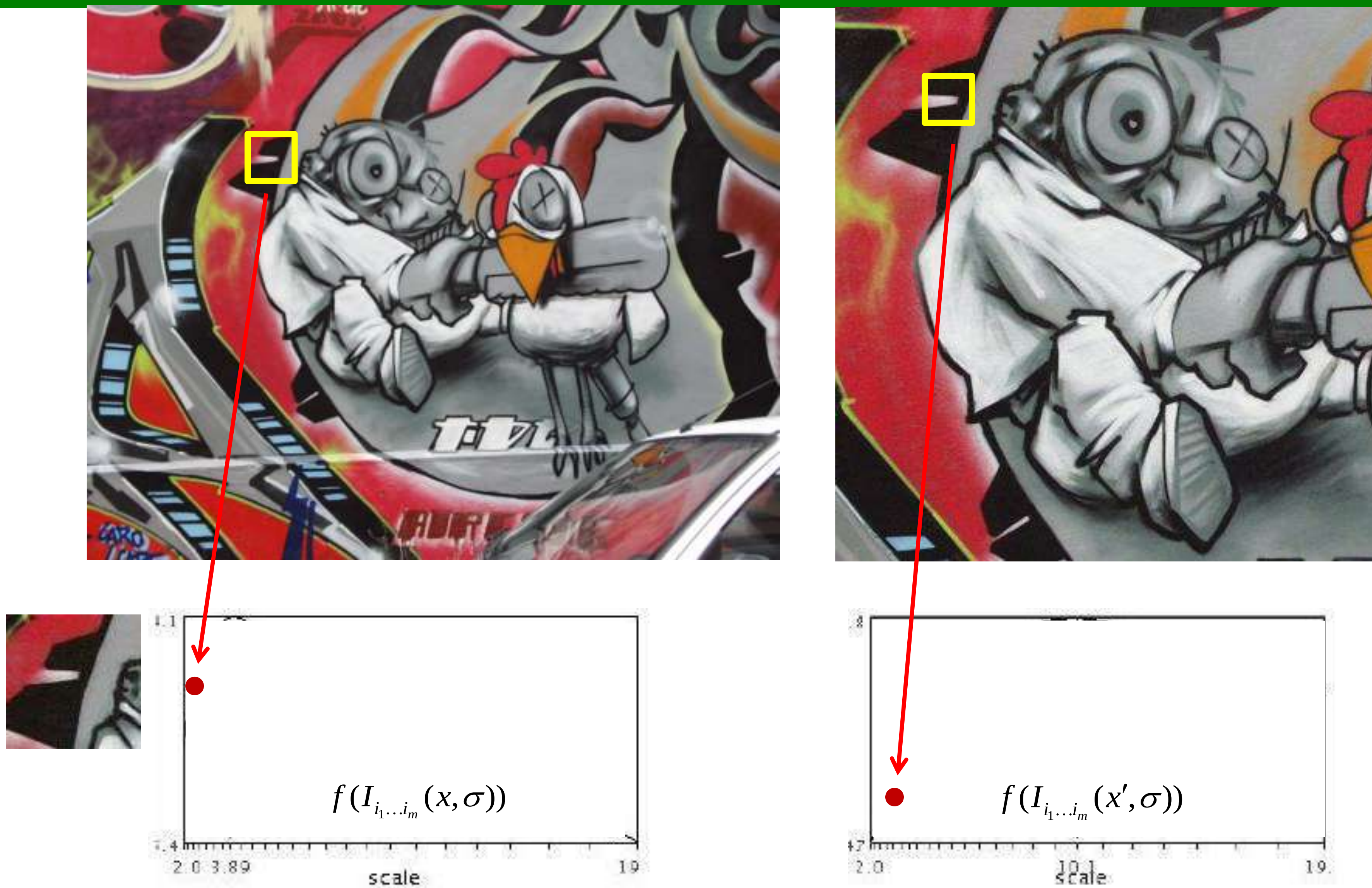


Automatic scale selection

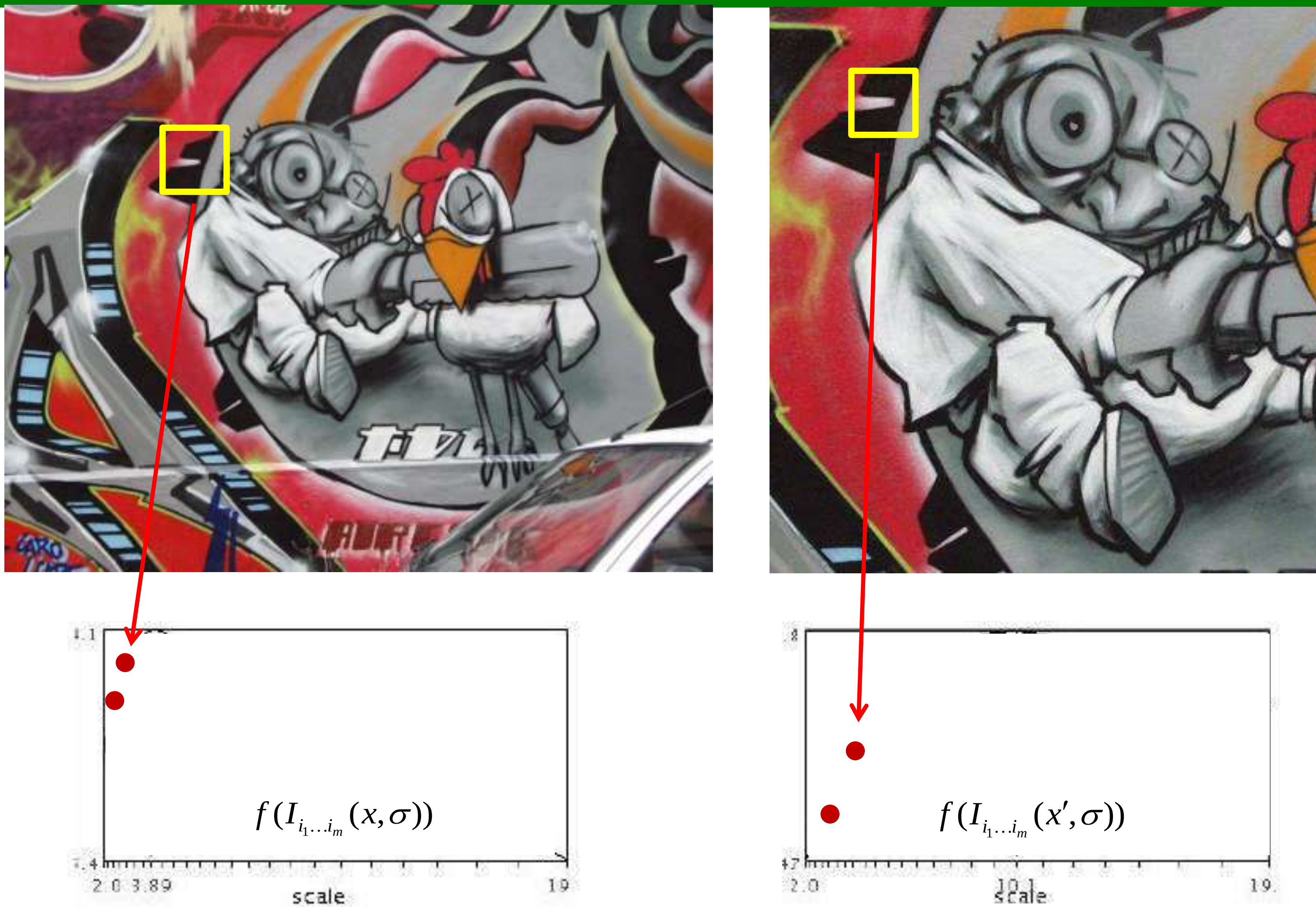
$$f(I_{i_1\ldots i_m}(x,\sigma)) = f(I_{i_1\ldots i_m}(x',\sigma'))$$

How to find corresponding patch sizes?

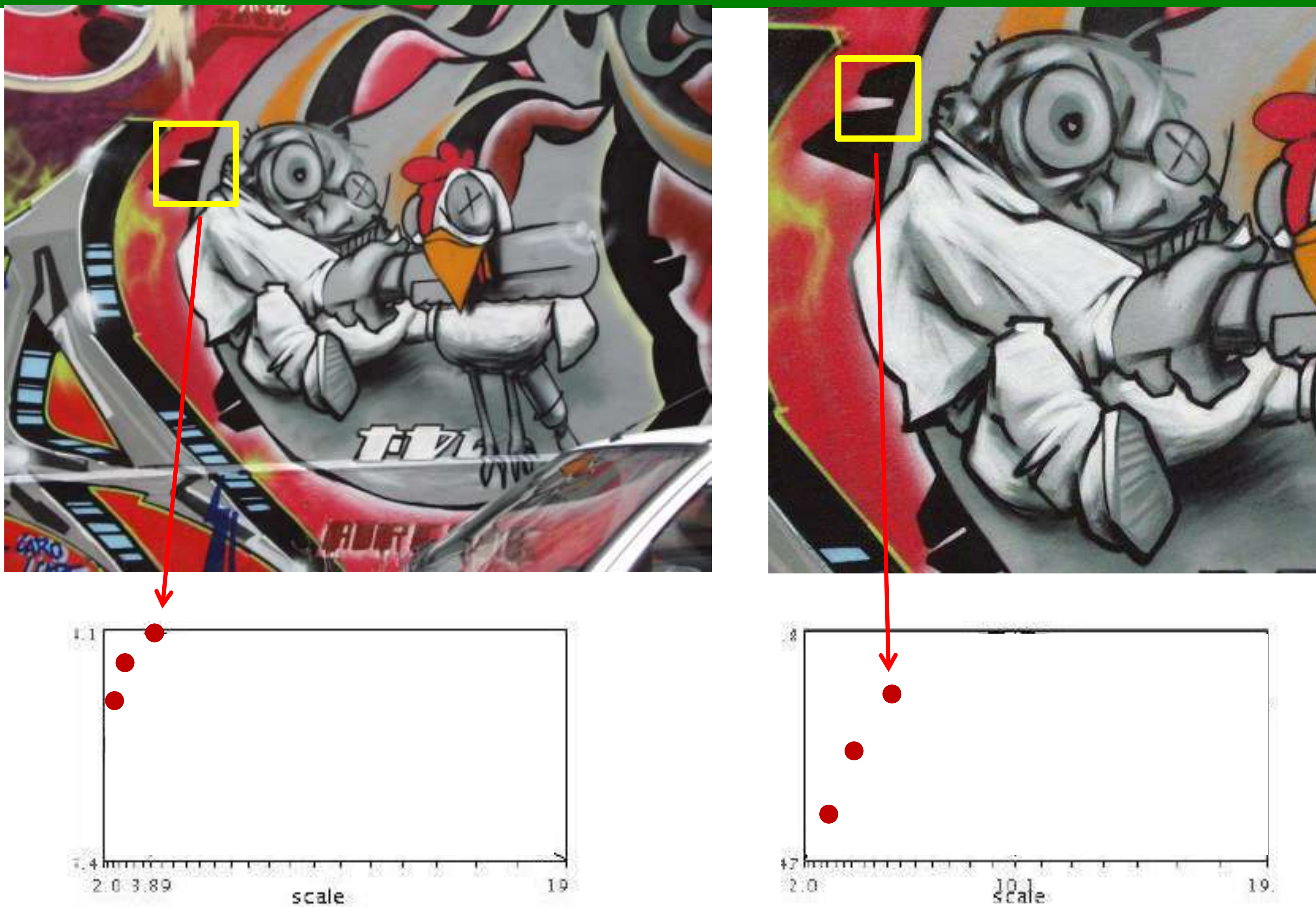
Automatic scale selection
Function responses for increasing scale



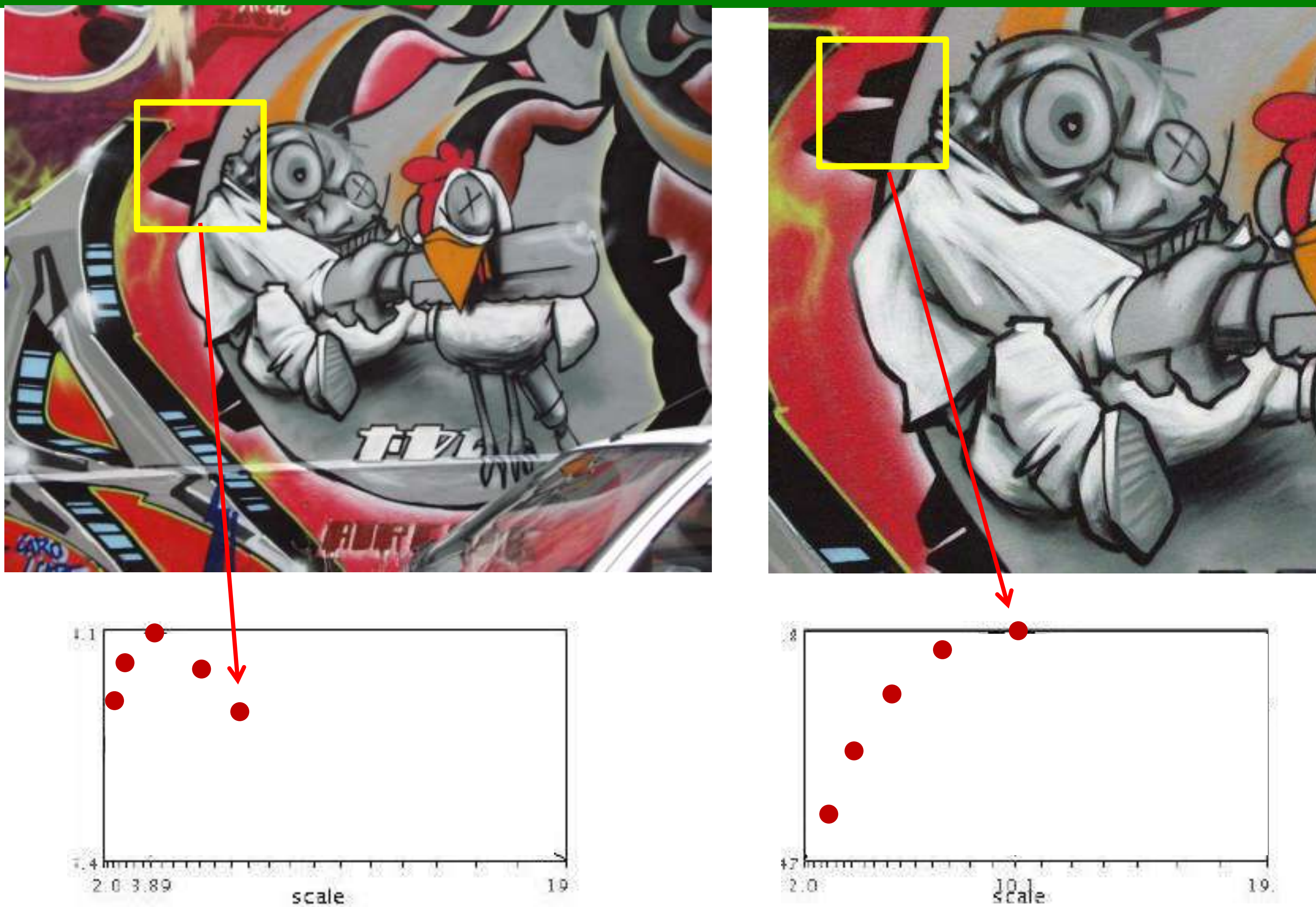
Automatic scale selection



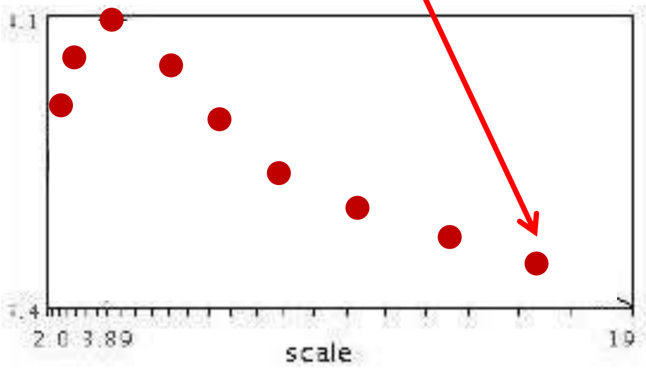
Automatic scale selection

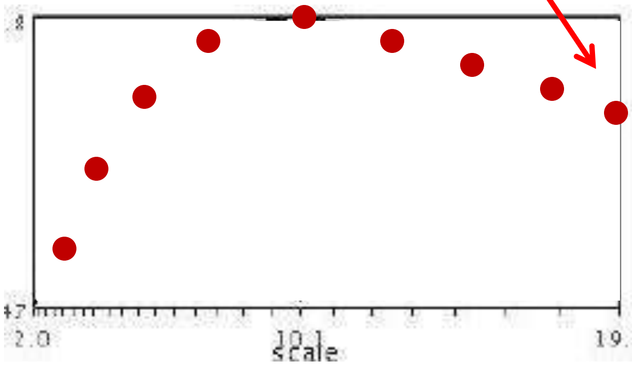


Automatic scale selection

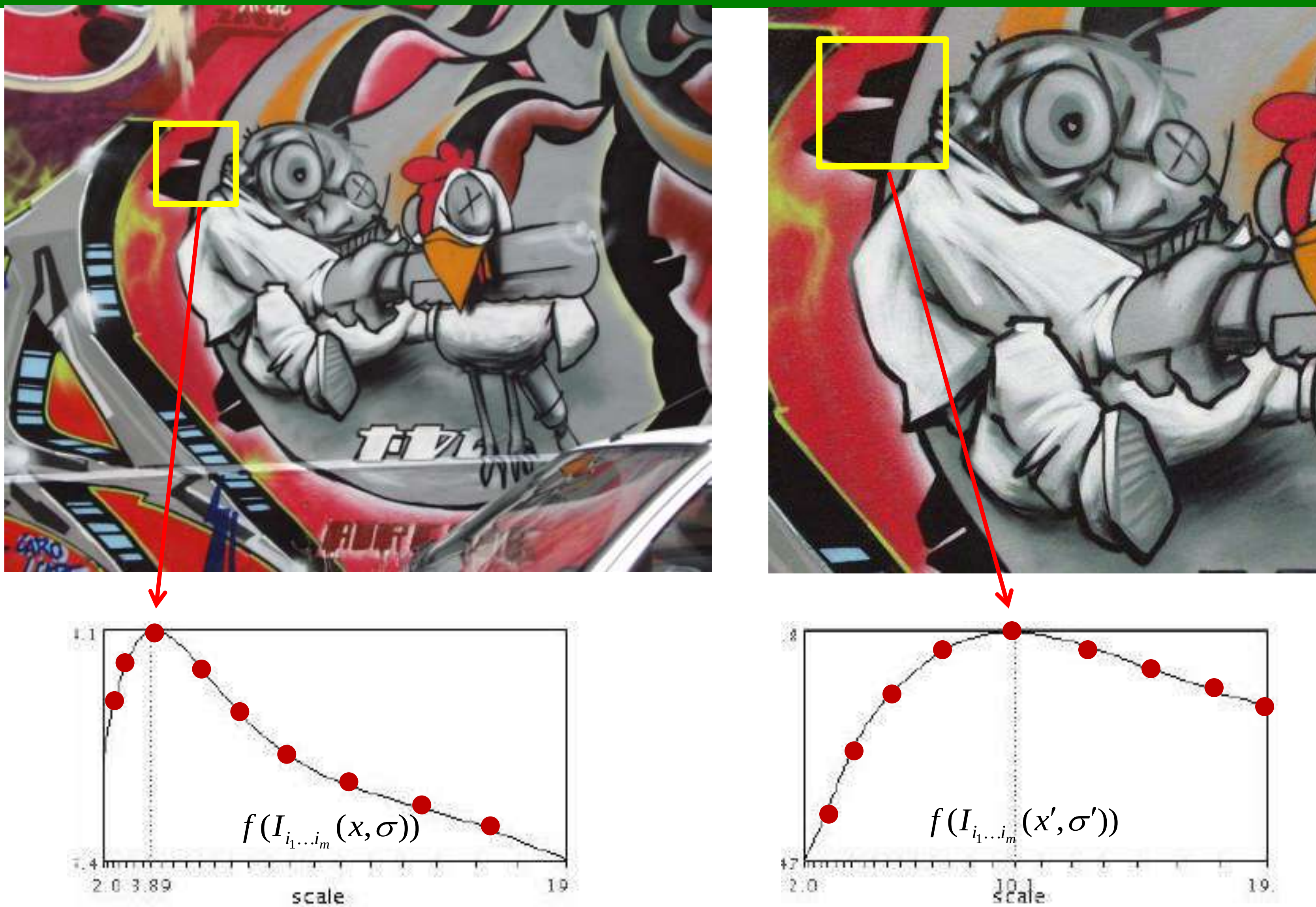


Automatic scale selection

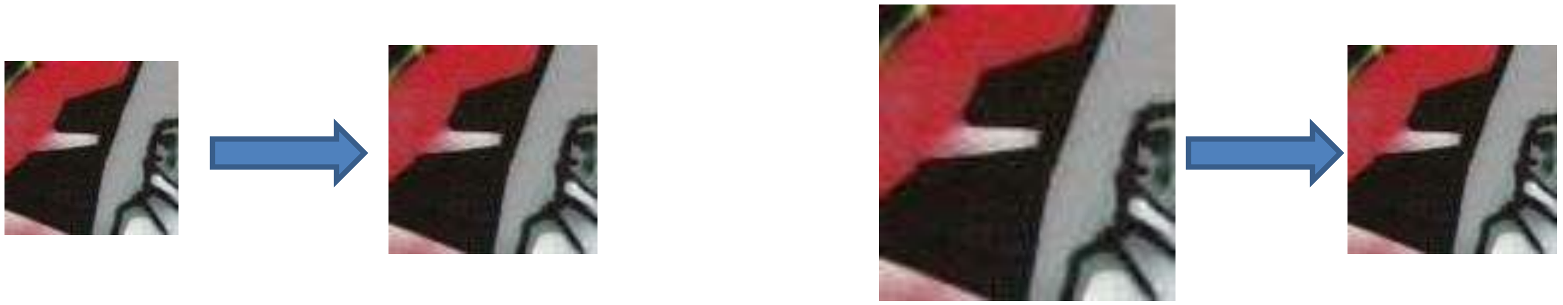




Automatic scale selection



Automatic scale selection



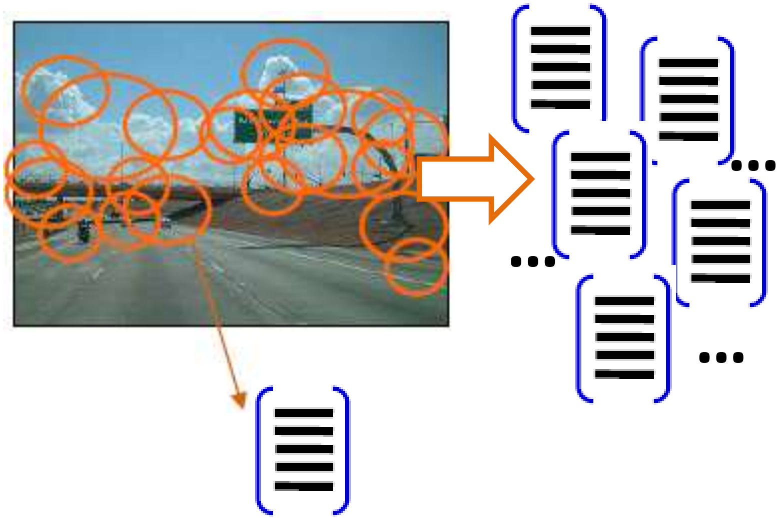
- Normalize: rescale to fixed size

Outline

- Automatic scale selection
- SIFT Descriptor

What is SIFT?

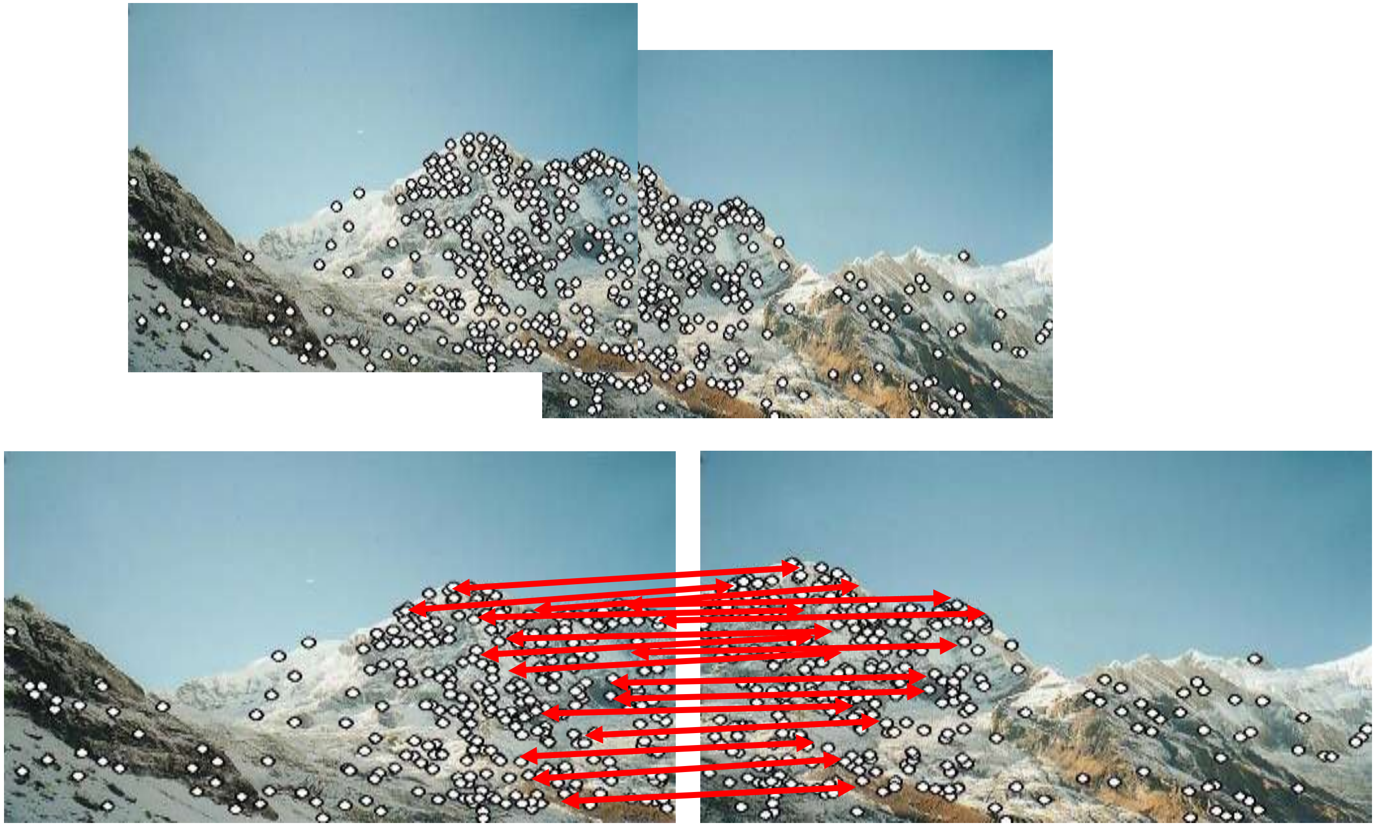
- Scale Invariant Feature Transform (SIFT) is an approach for detecting and extracting ***local features*** from an image.



What is SIFT?

- Originally proposed for panorama stitching

Panorama stitching



How to detect *which features* to match?

Slide credit: Prof. Grauman

Applications

- Object recognition
- Robot localization and mapping
- 3D scene modeling, recognition and tracking
- Human action recognition
- Analyzing the human brain in 3D magnetic resonance images
- ……

What is SIFT?

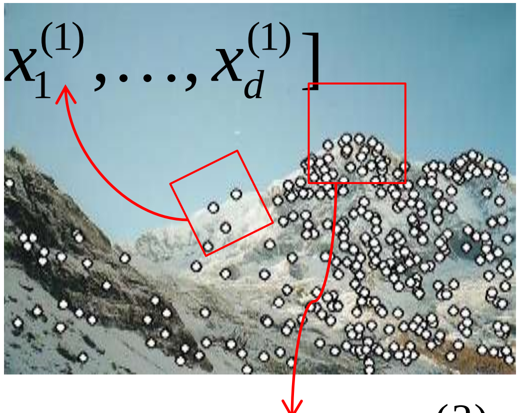
- SIFT feature descriptors are reasonably ***invariant*** to
  - scaling
  - rotation
  - image noise
  - changes in illumination
  - small changes in viewpoint

Local features: main components

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

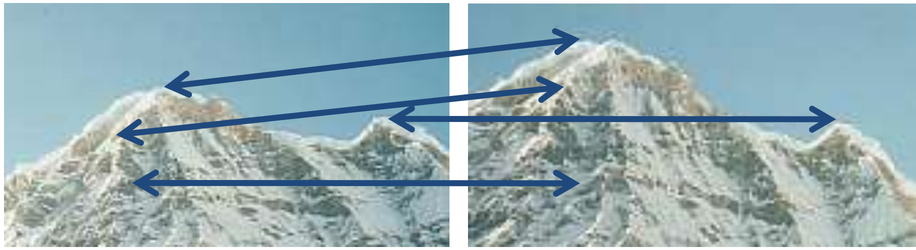
2) **Description**: Extract a feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

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

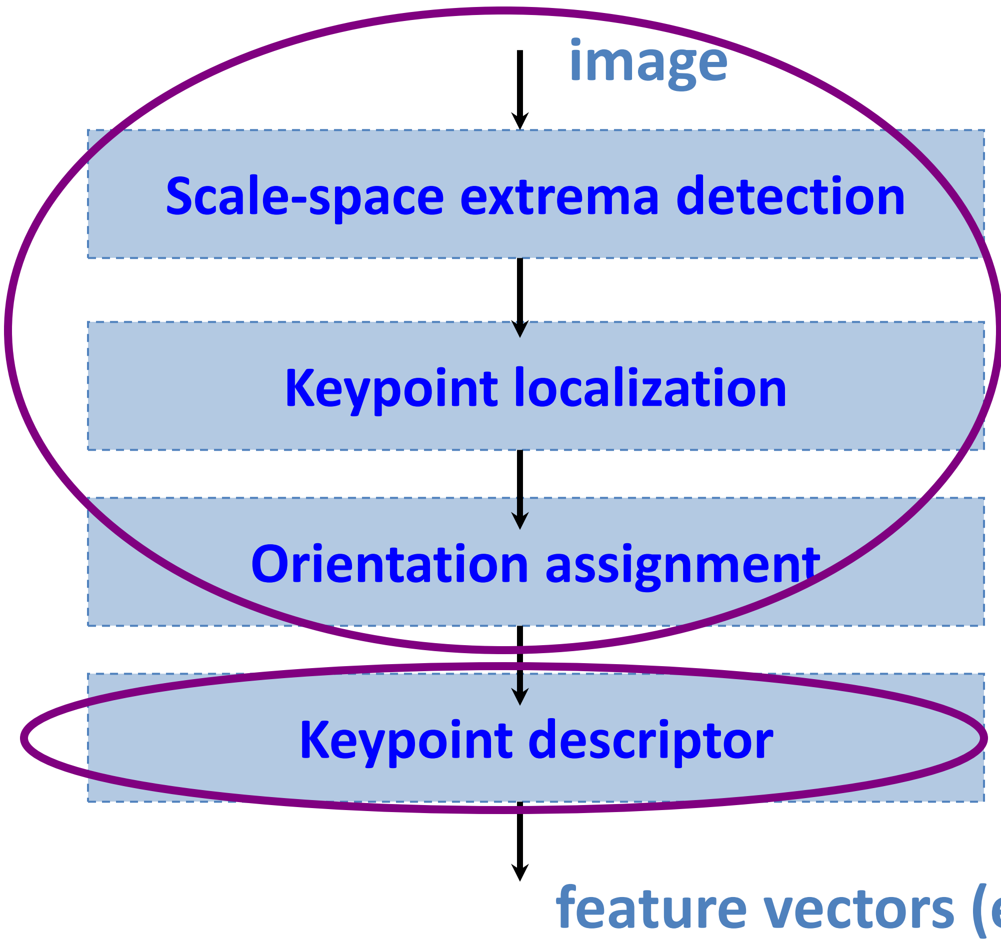


Slide credit: Prof. Grauman

SIFT: Overview

- Major stages of SIFT computation

Interest point detector
+
descriptor



Identify potential interest points
(location, scale)

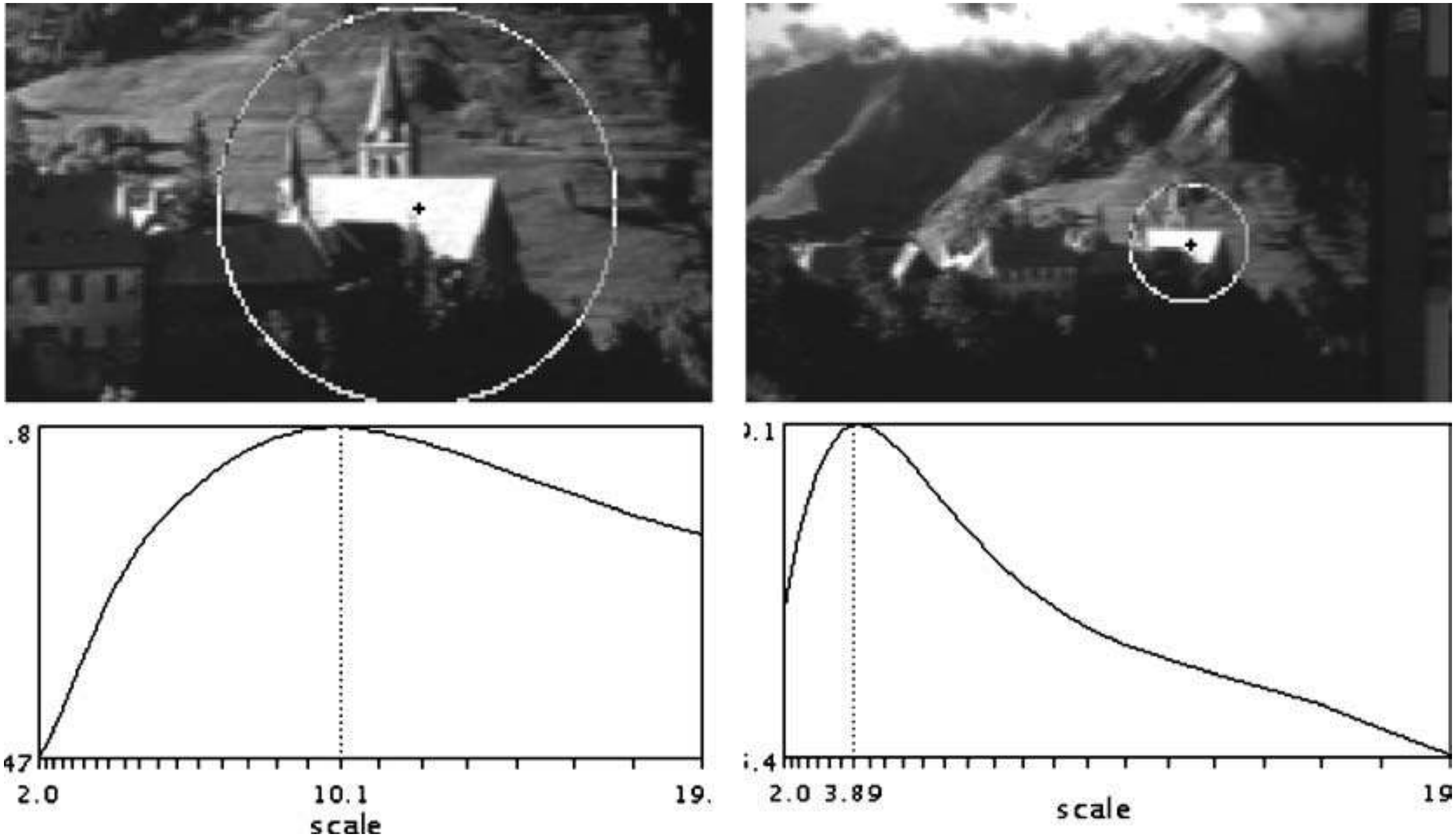
Localize candidate keypoints
Reduced sets of (location, scale)

Identify the dominant orientations
(location, scale, orientation)

Build a descriptor based on
histogram of gradients in local
neighborhood

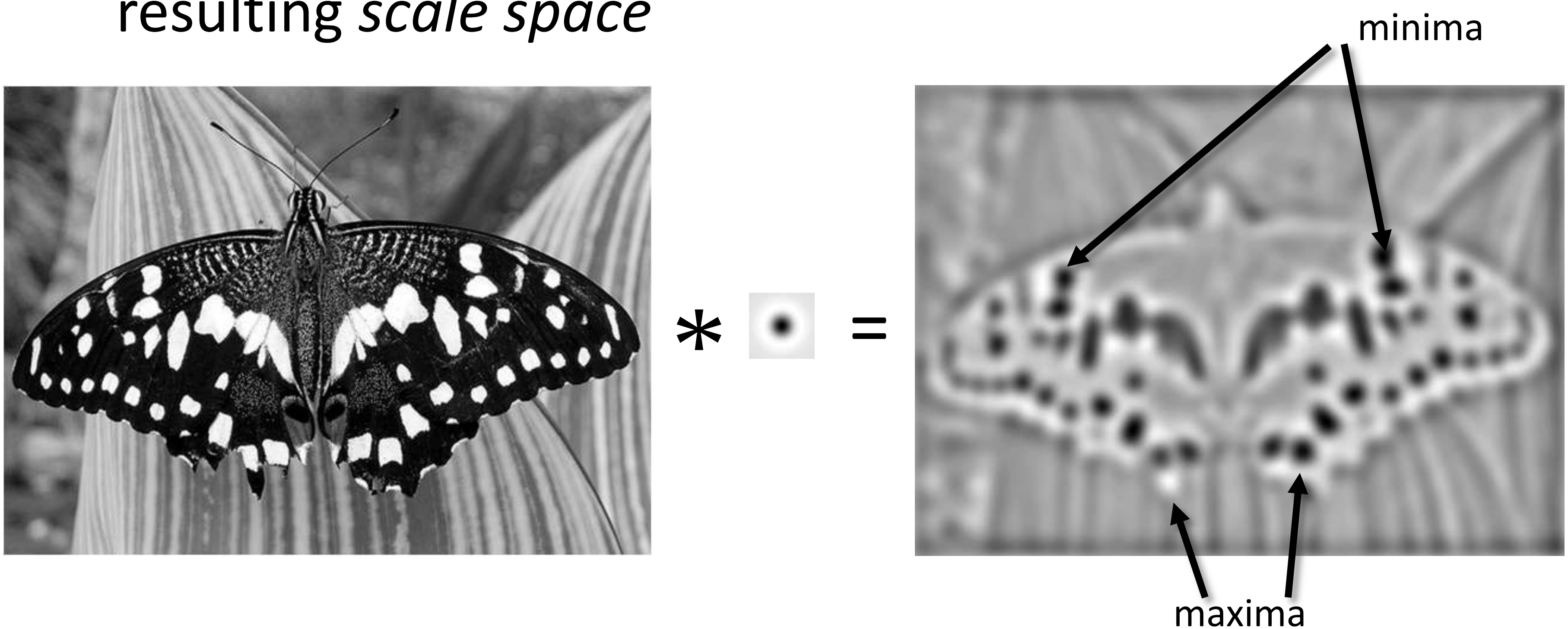
Keypoint detection with scale selection

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



Basic idea → Blob detection

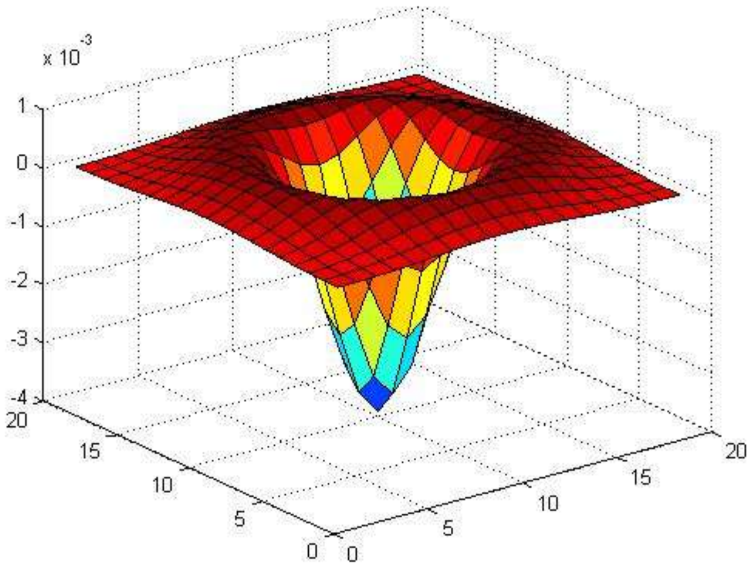
- Convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting *scale space*

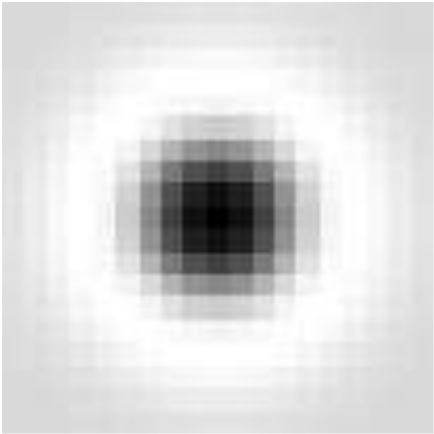


- Find maxima *and minima* of blob filter response in space *and scale*

Blob filter

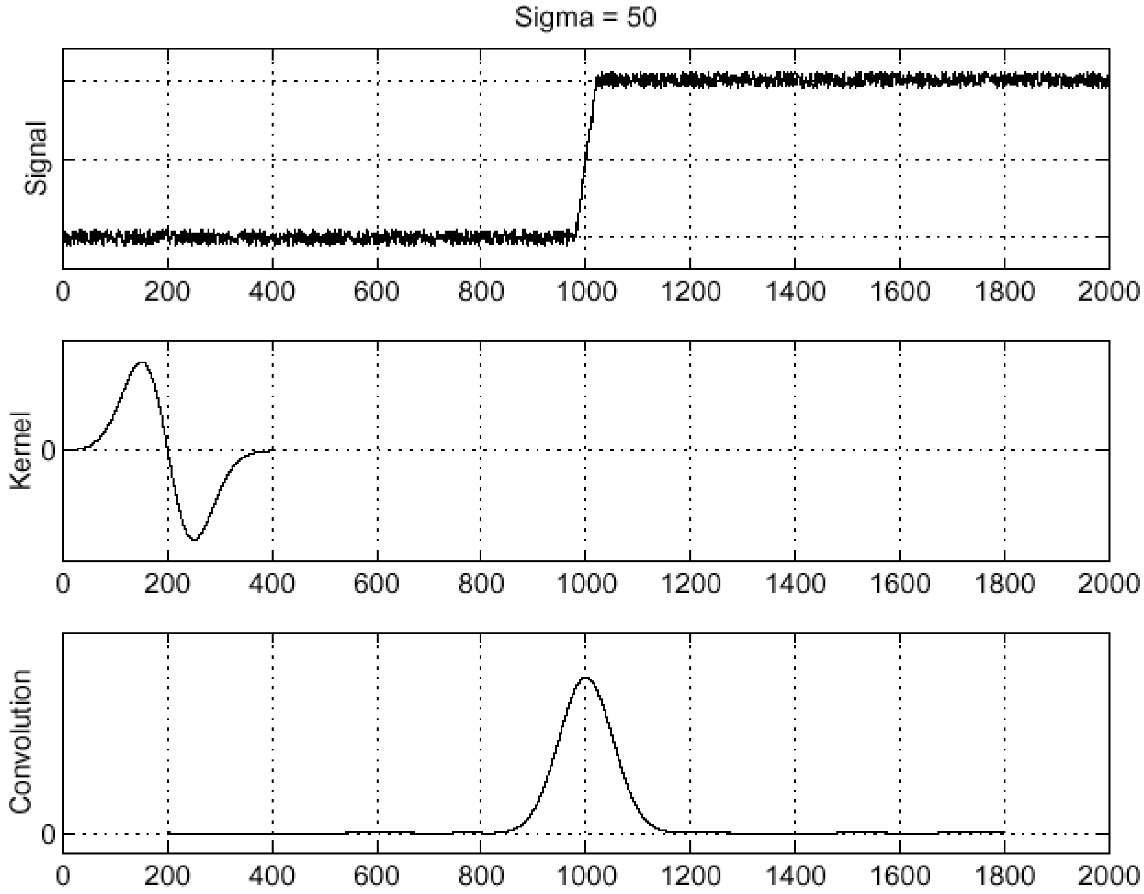
- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D





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

Recall: Edge detection



f

Edge

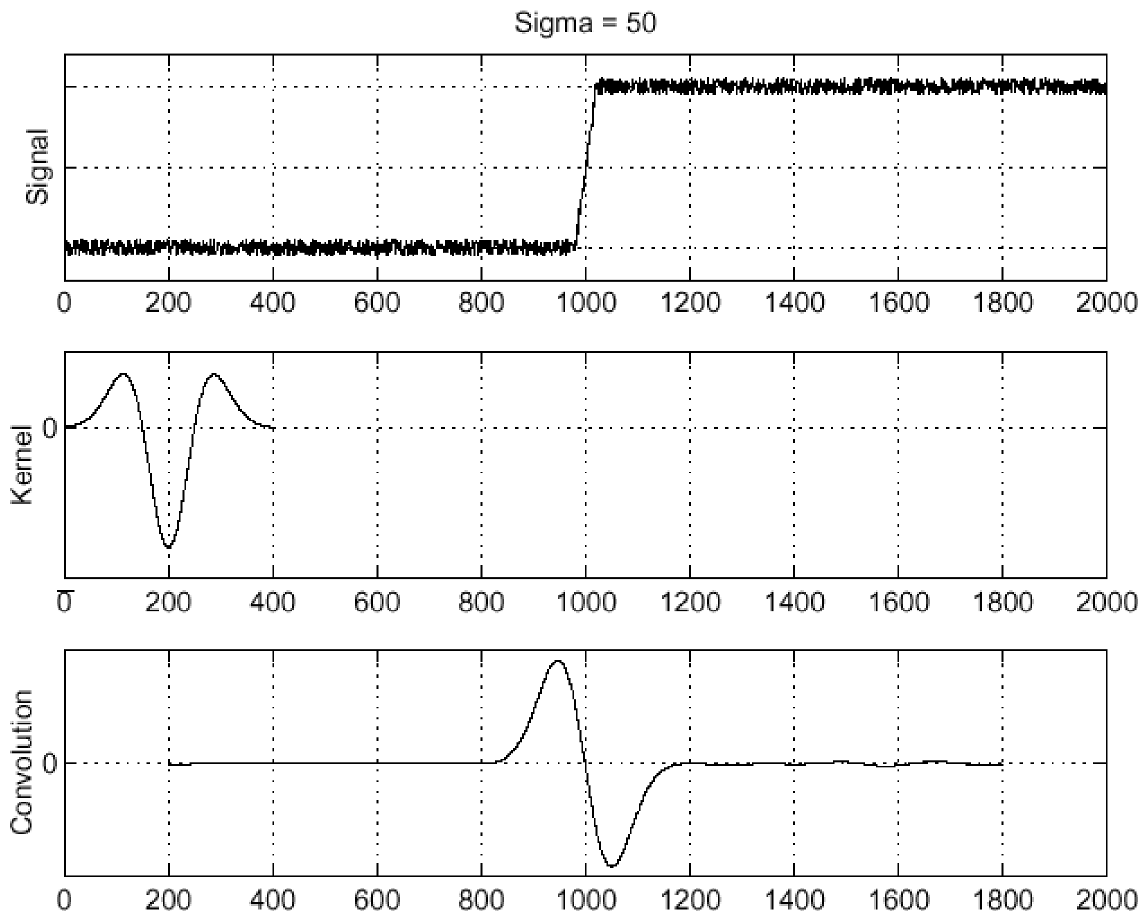
$\frac{d}{dx} g$

Derivative of Gaussian

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

Edge = maximum of derivative

Edge detection, Take 2



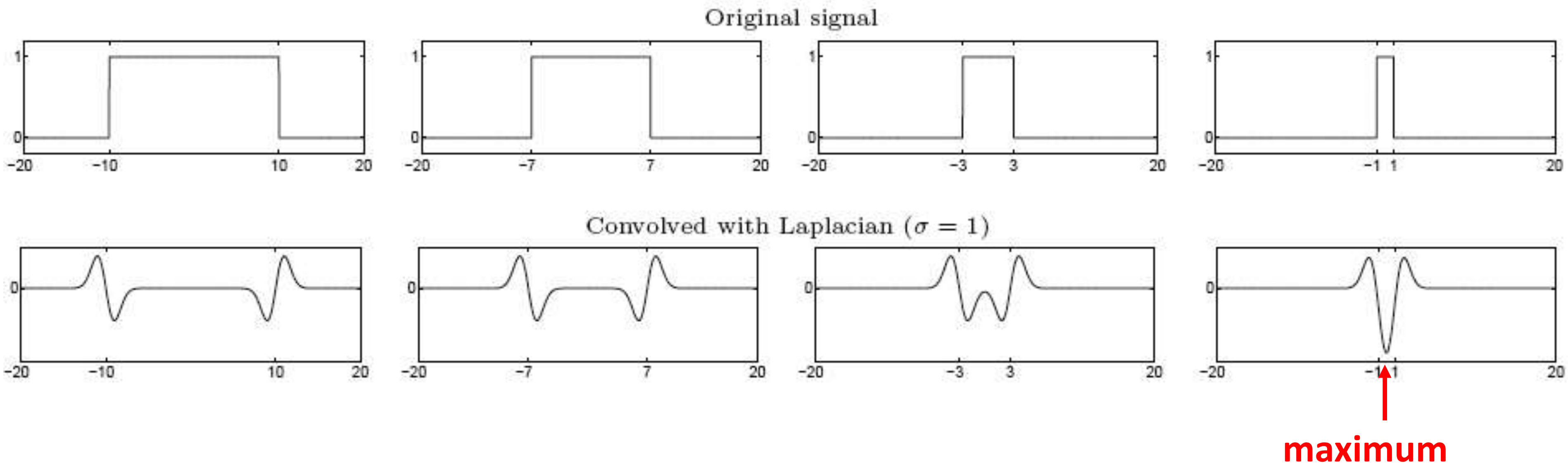
f — Edge

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

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

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

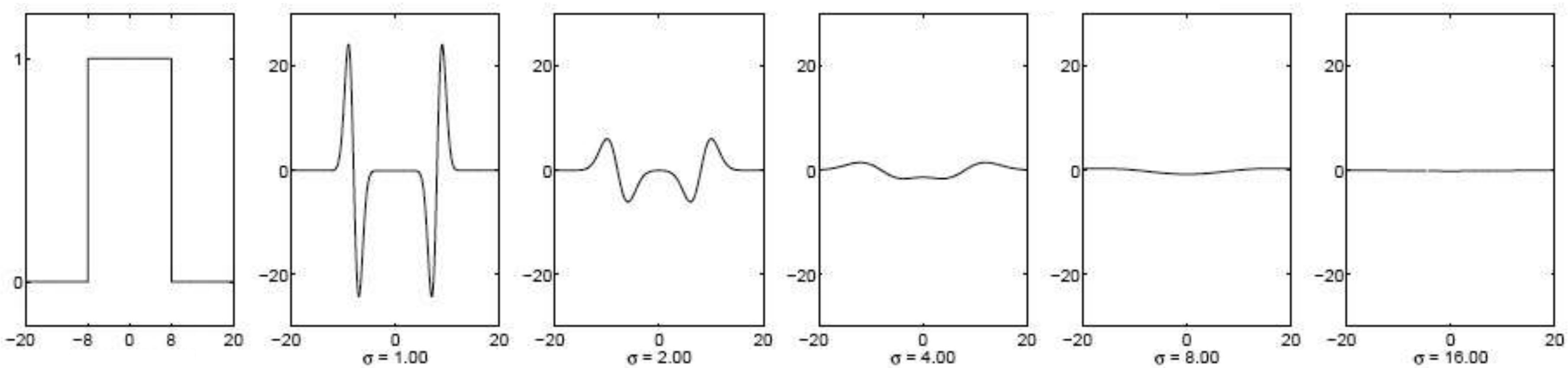
Scale normalization

- 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

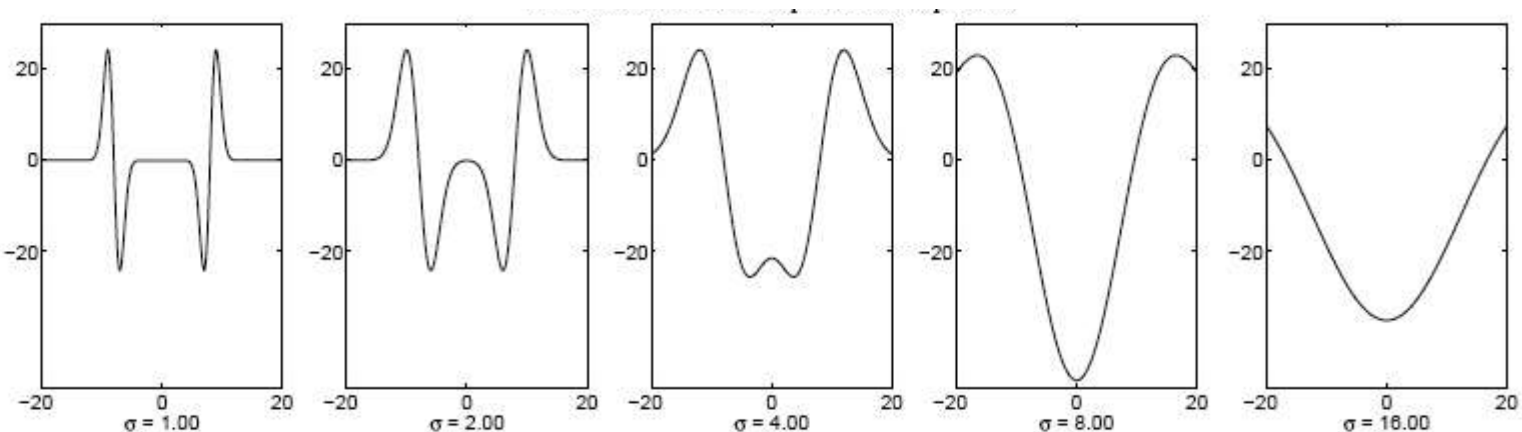
Effect of scale normalization

Original signal

Unnormalized Laplacian response



Scale-normalized Laplacian response

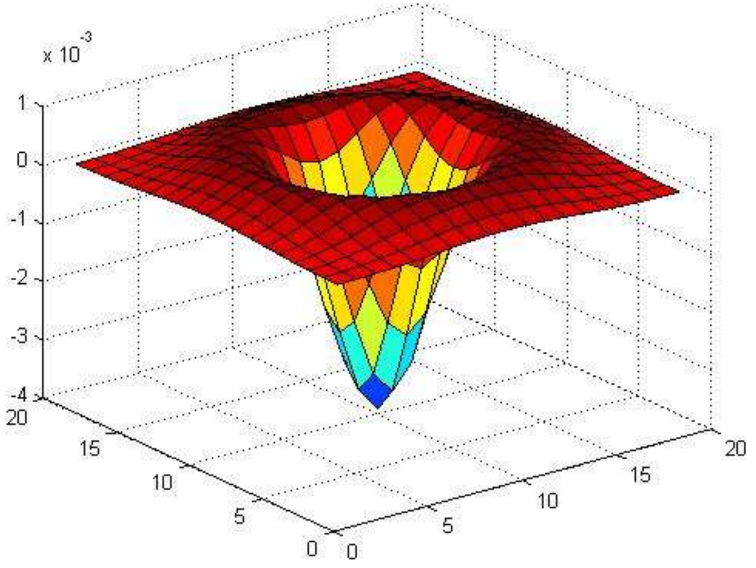


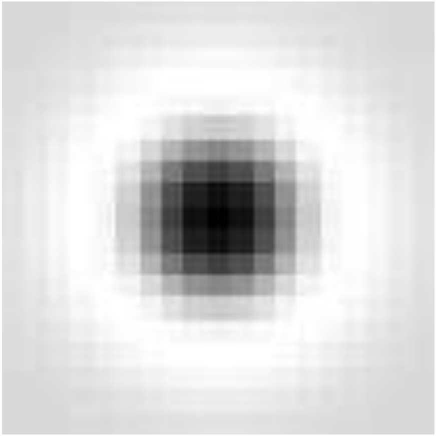
maximum

Blob detection in 2D

- *Scale-normalized* Laplacian of Gaussian:

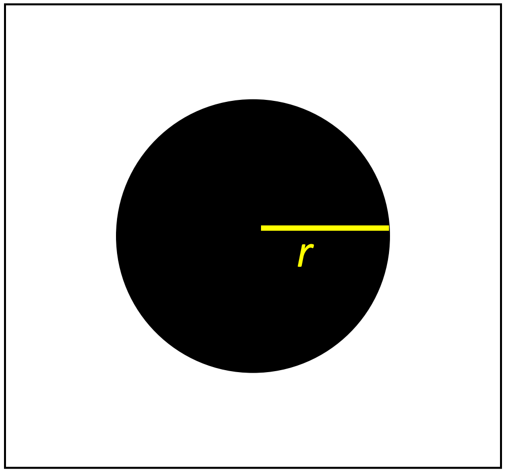
$$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$



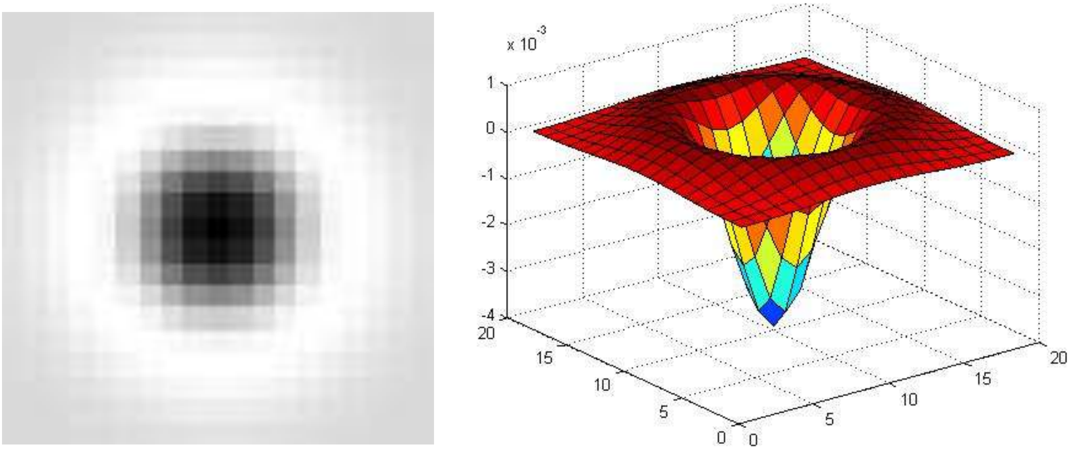


Blob detection in 2D

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



image



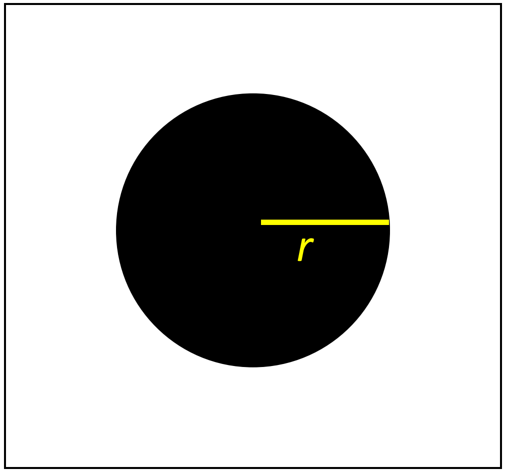
Laplacian

Blob detection in 2D

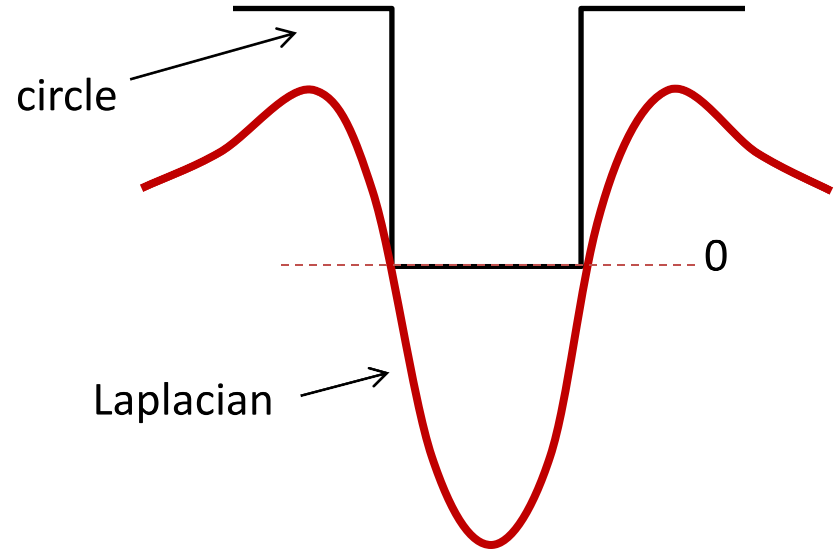
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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



image



Scale-space blob detector

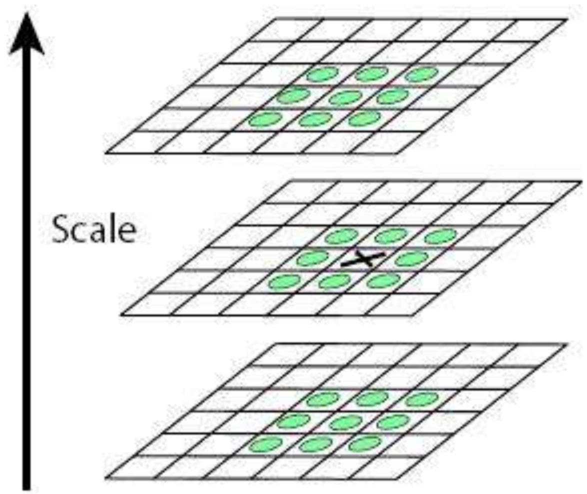
1. Convolve image with scale-normalized Laplacian at several scales

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



Efficient implementation

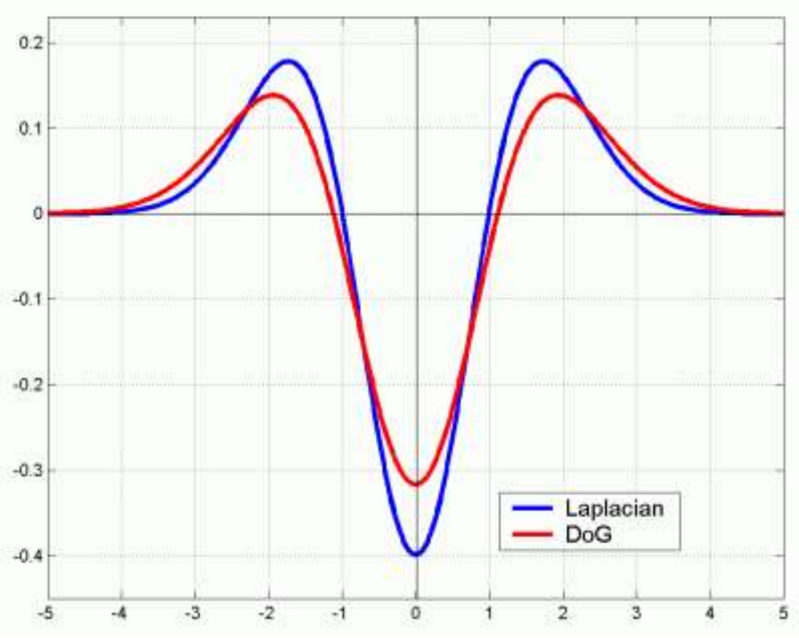
- Approximating the Laplacian with a difference of Gaussians:

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

(Laplacian)

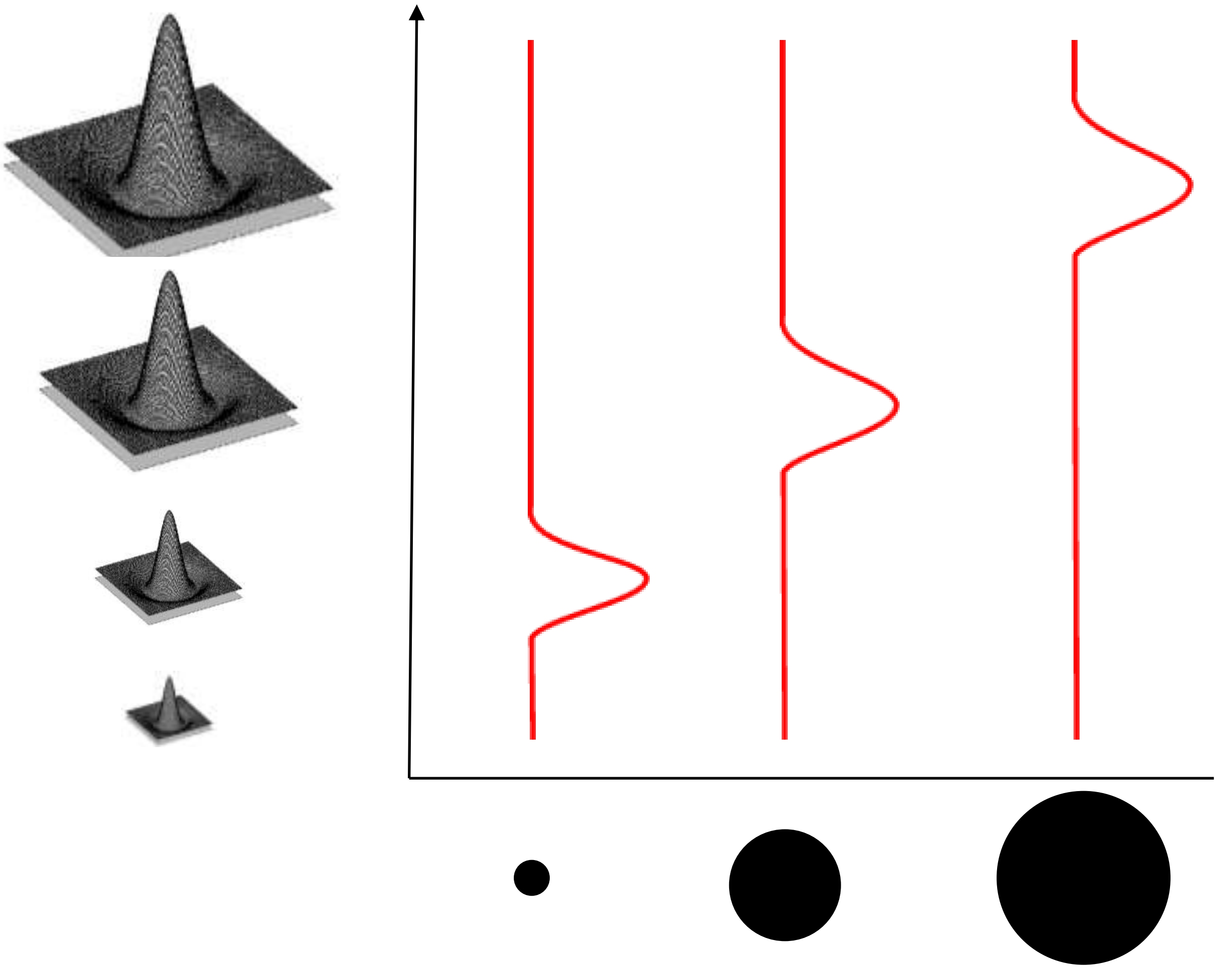
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



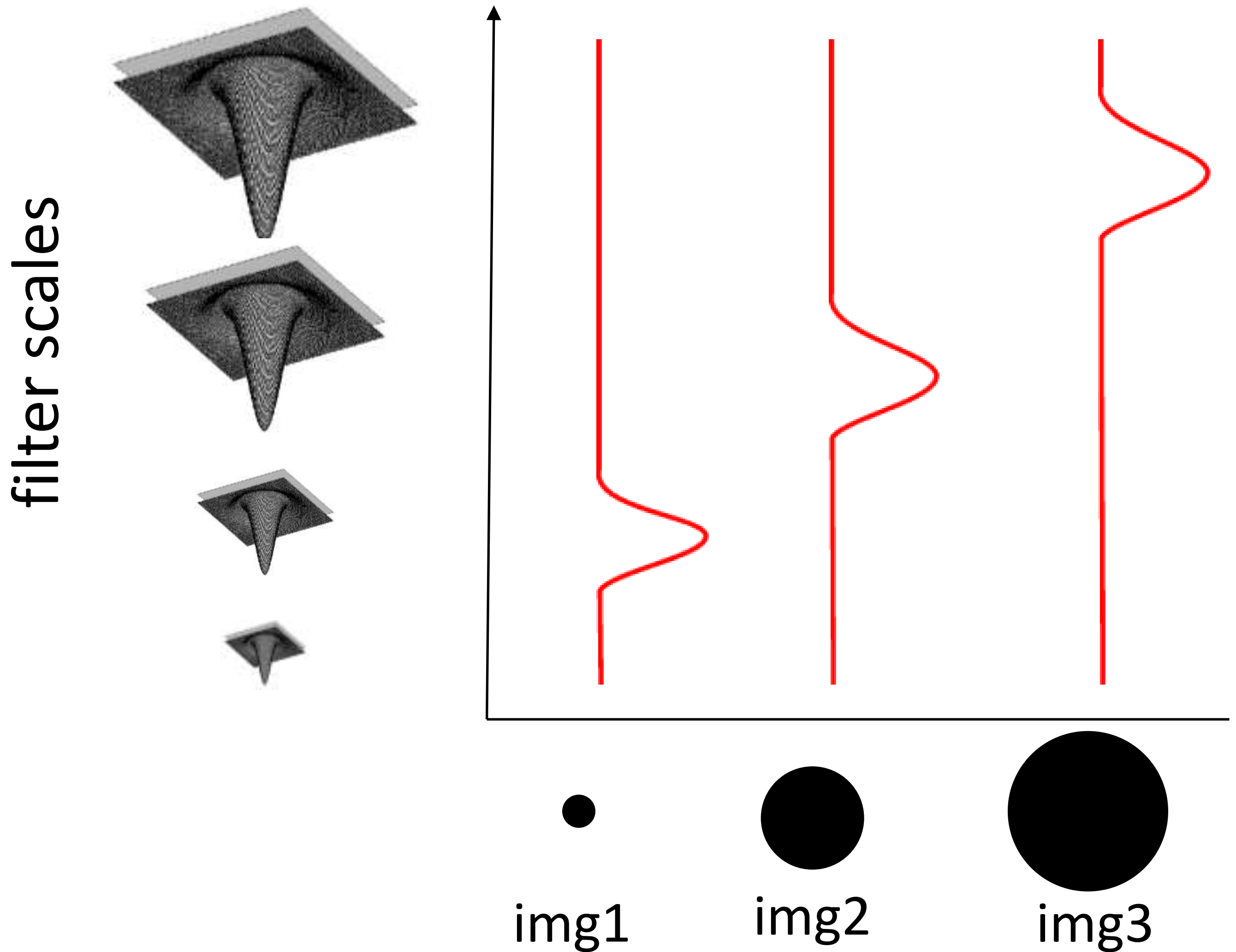
Useful Signature Function

- Difference-of-Gaussian = "blob" detector



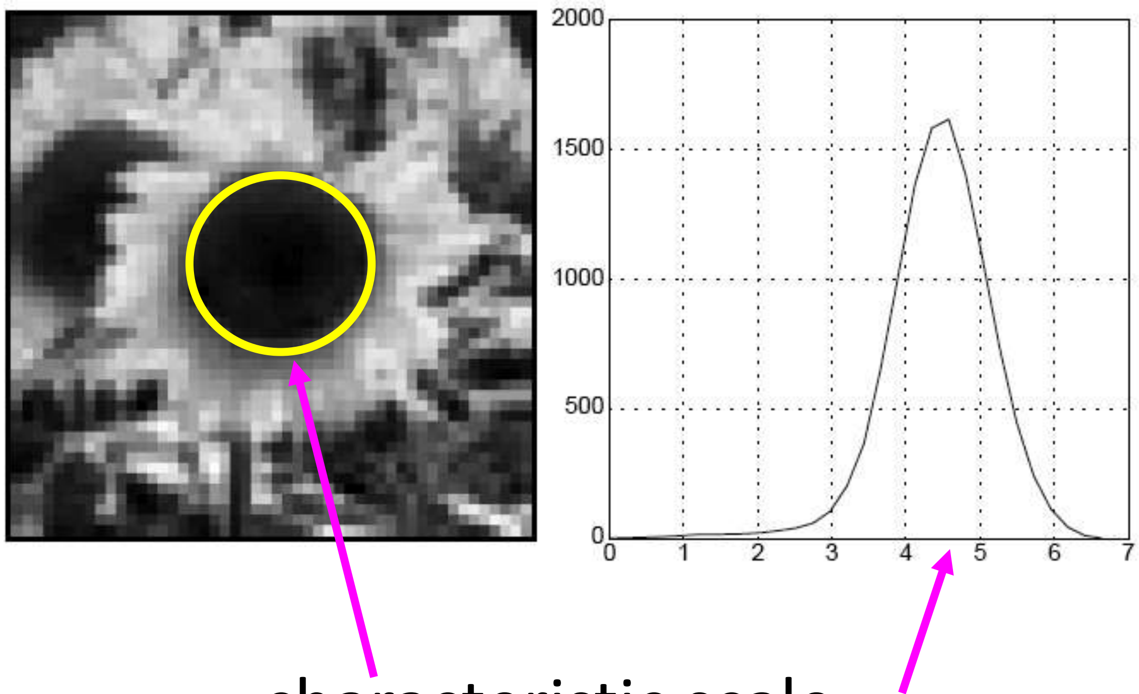
Laplacian of Gaussian

- Difference-of-Gaussian = "blob" detector



Blob detection in 2D

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



characteristic scale

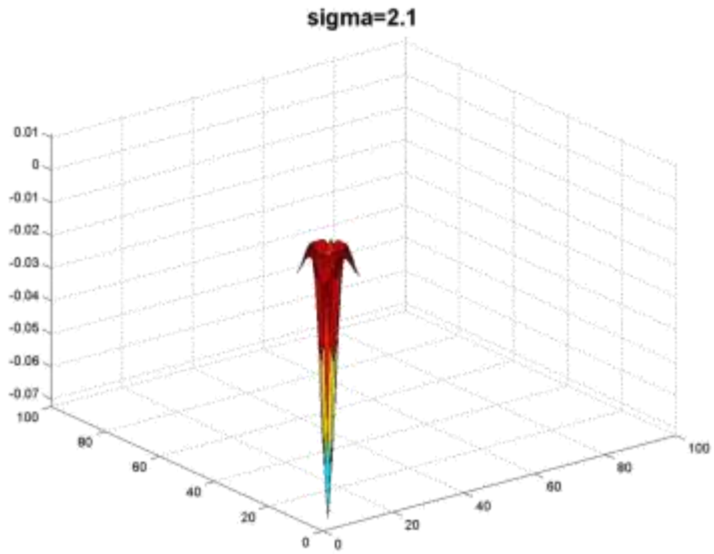
Slide credit: Lana Lazebnik

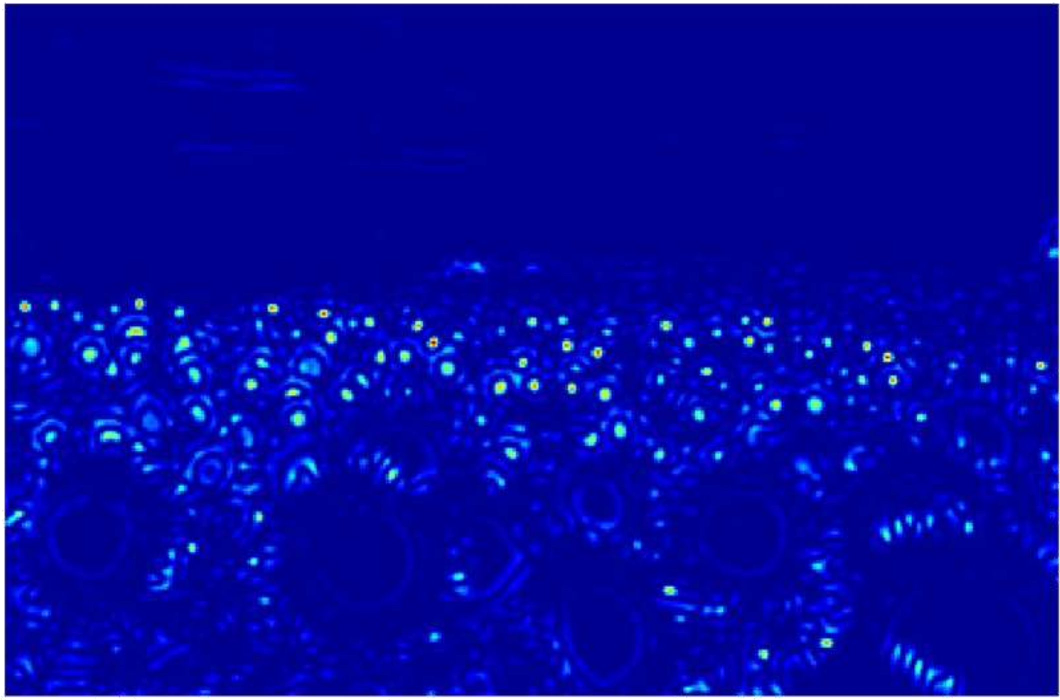
Exa

Original image at
¾ the size

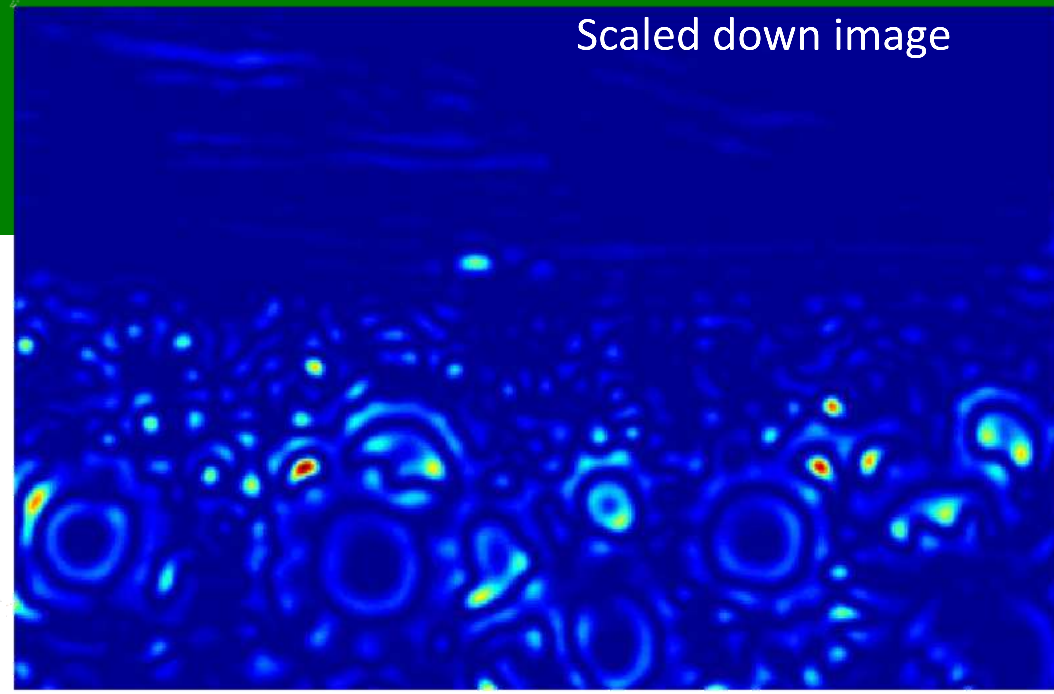
Slide credit: Kristen Grauman

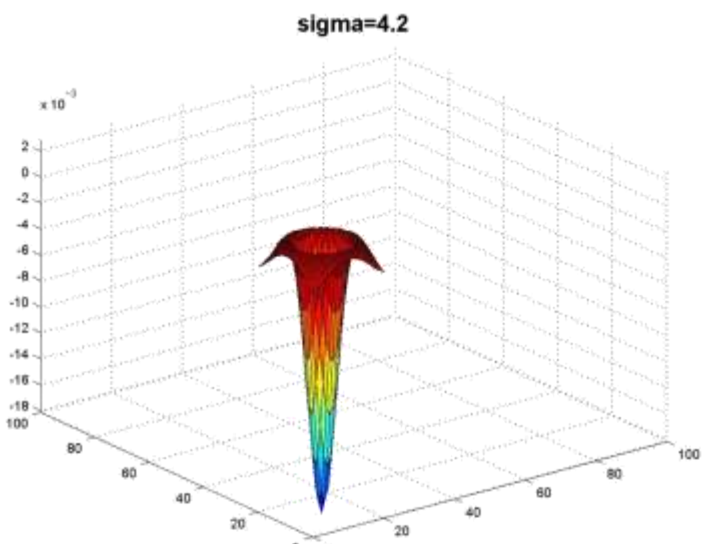
Example

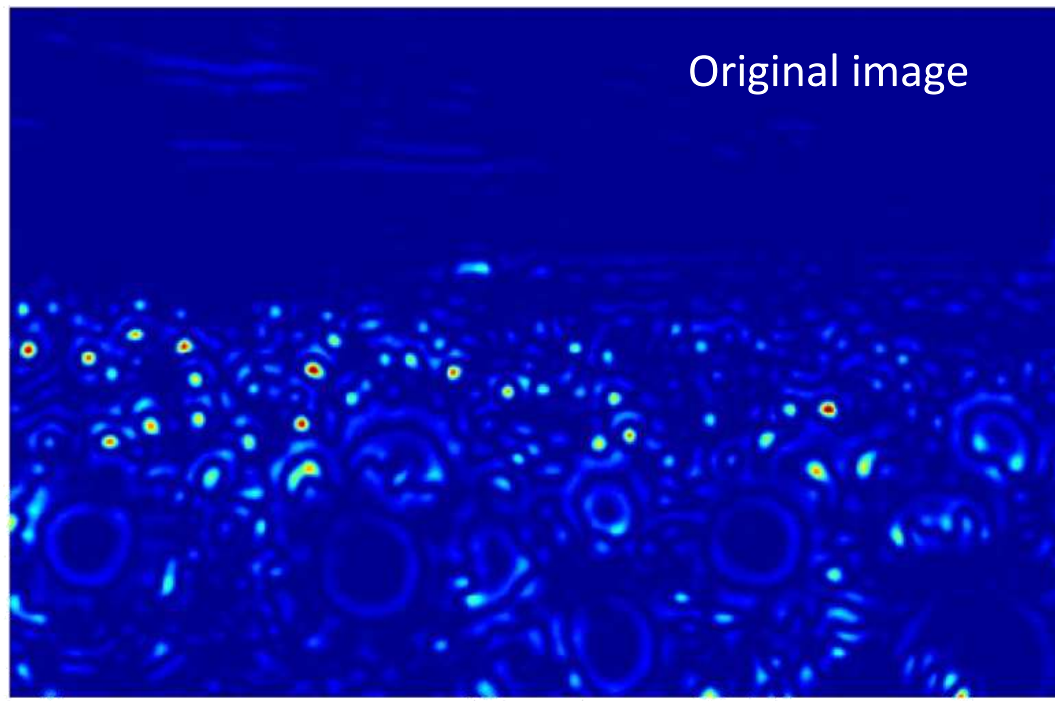




Slide credit: Prof. Grauman

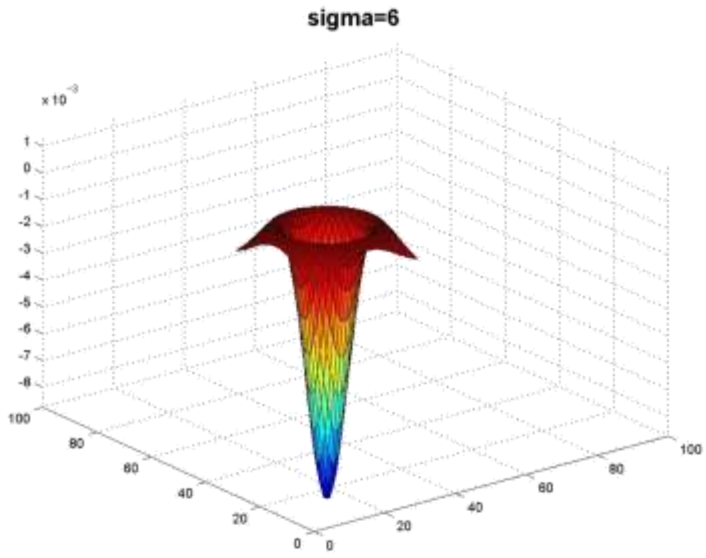


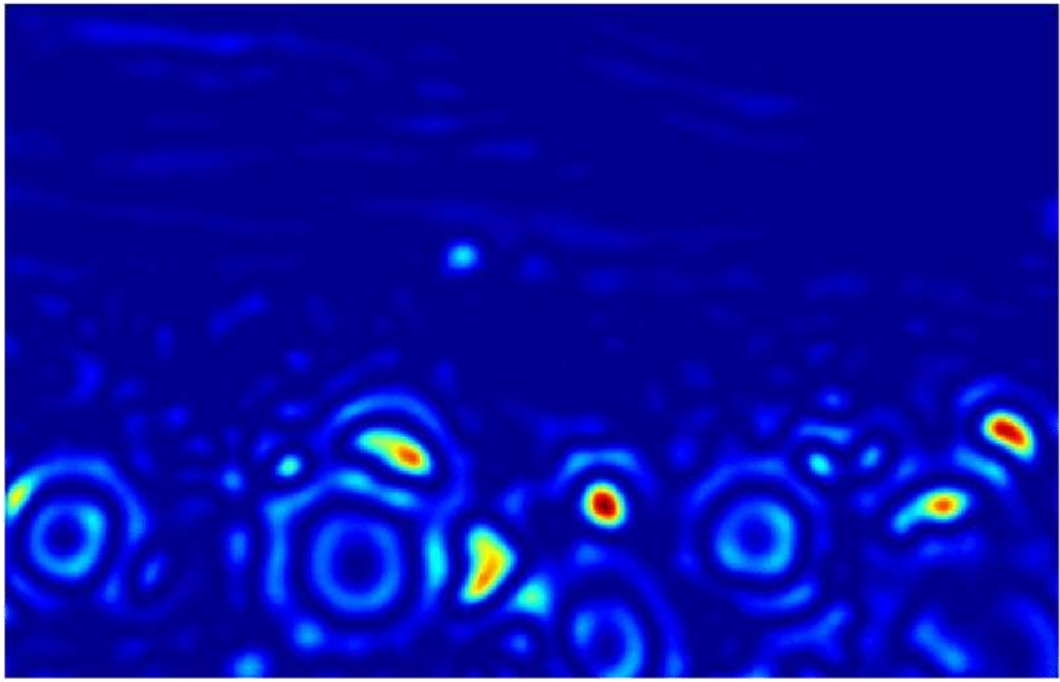




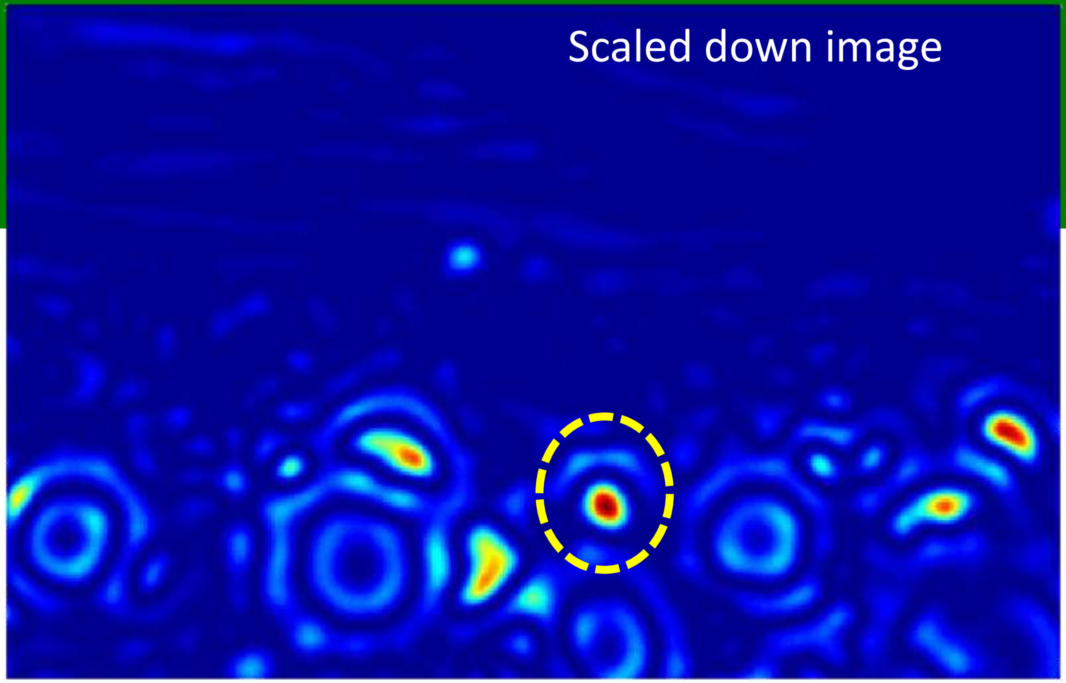
Slide credit: Kristen Grauman

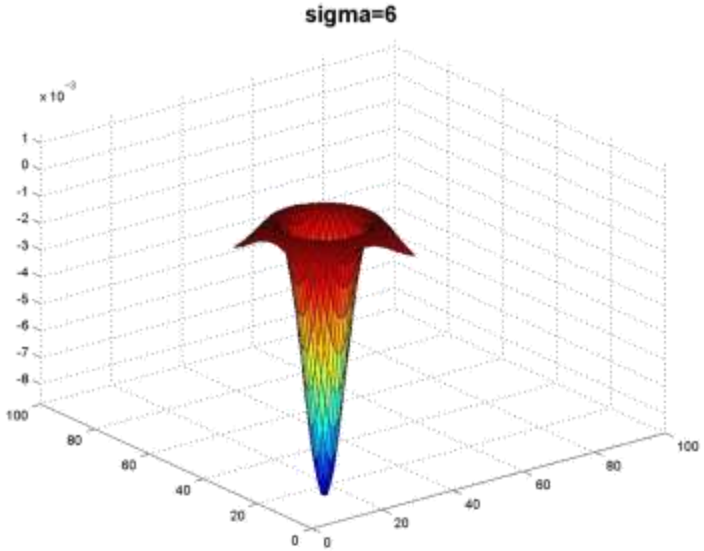
Example

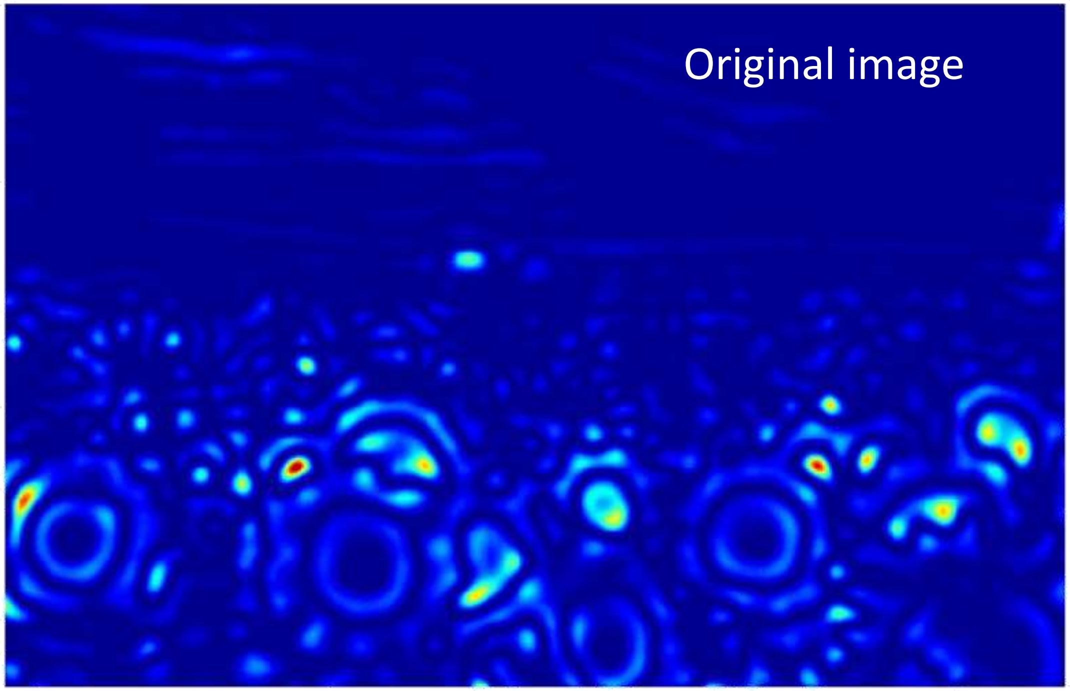




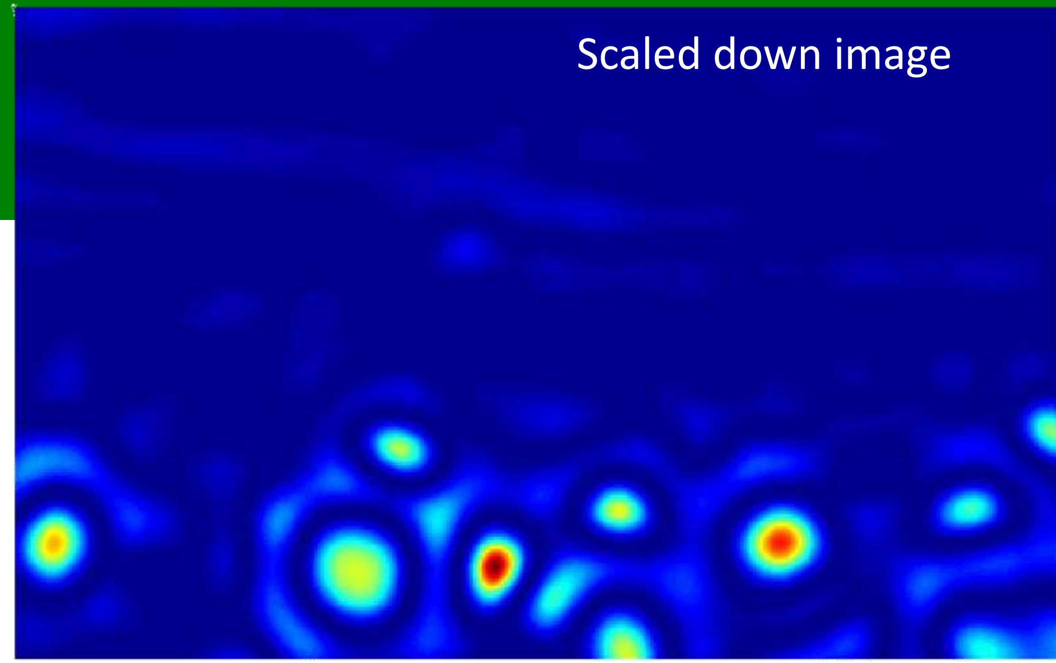
Slide credit: Prof. Grauman

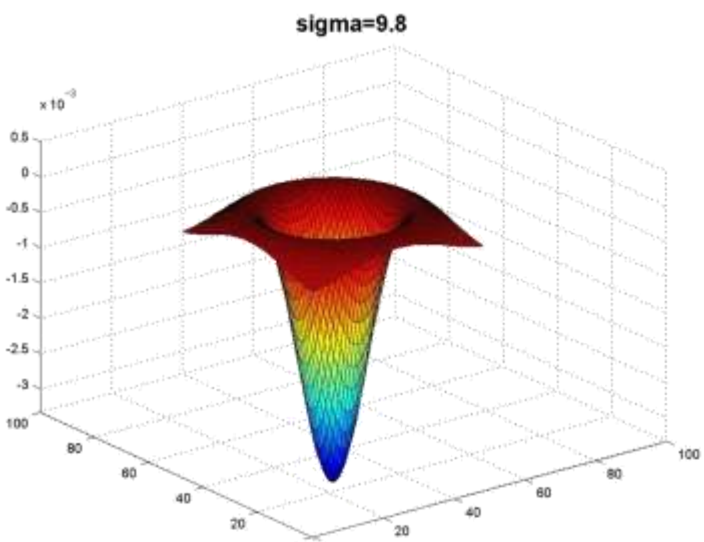


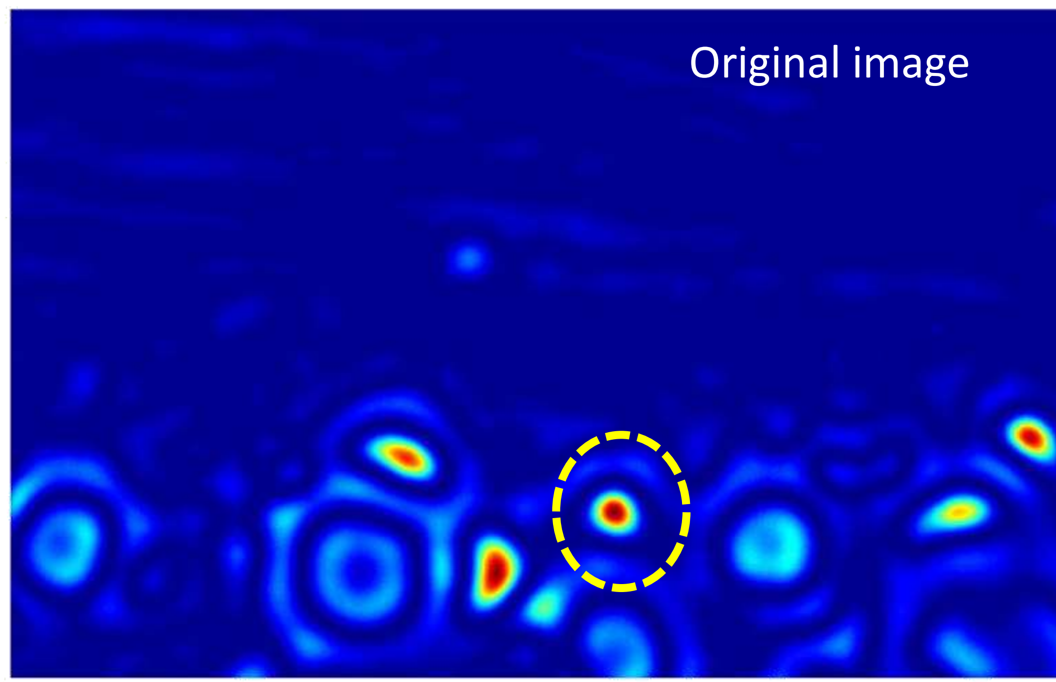




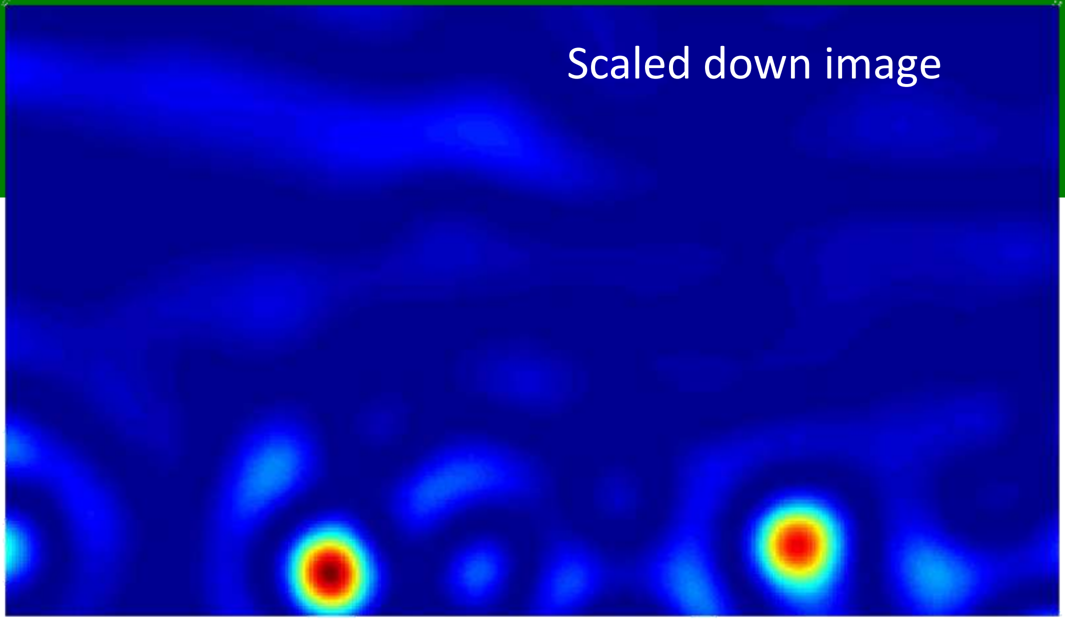
Slide credit: Kristen Grauman

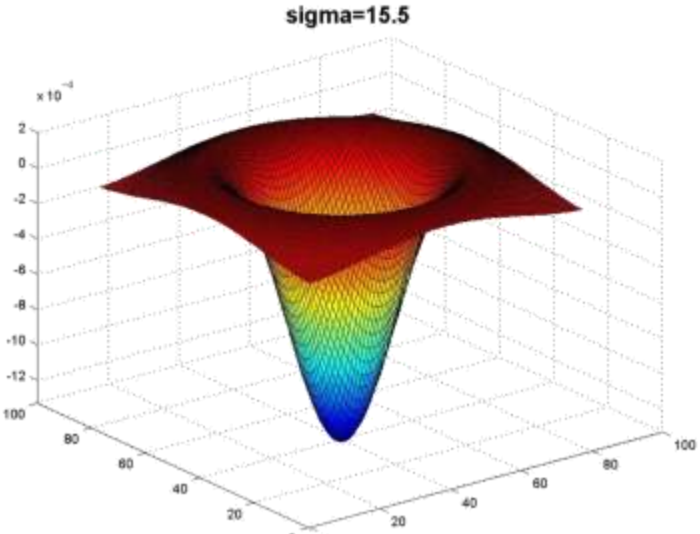


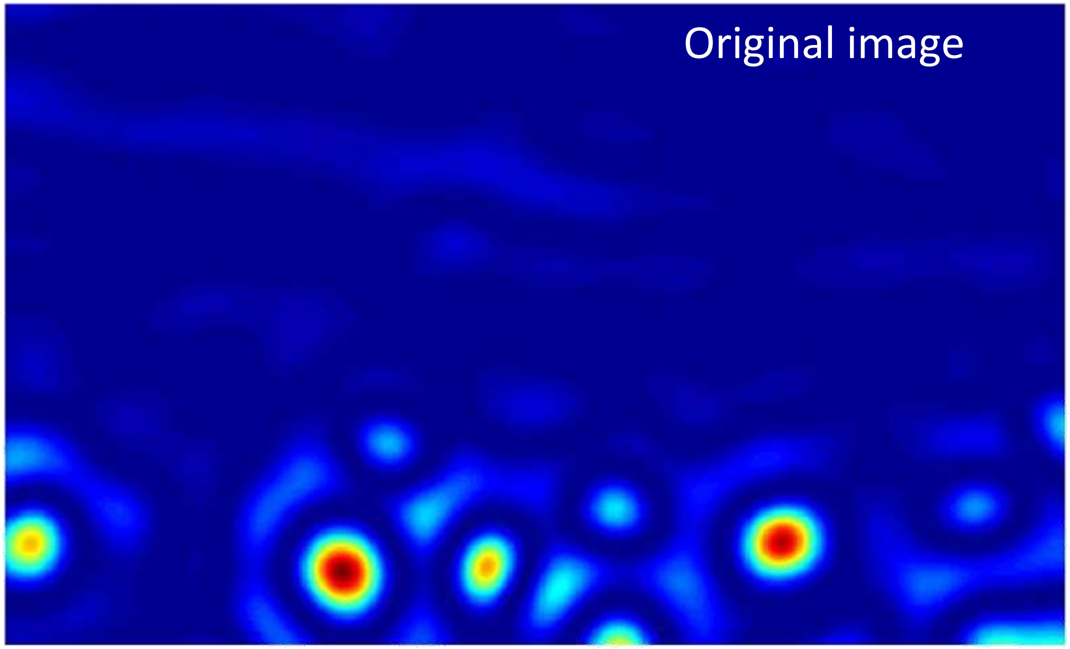




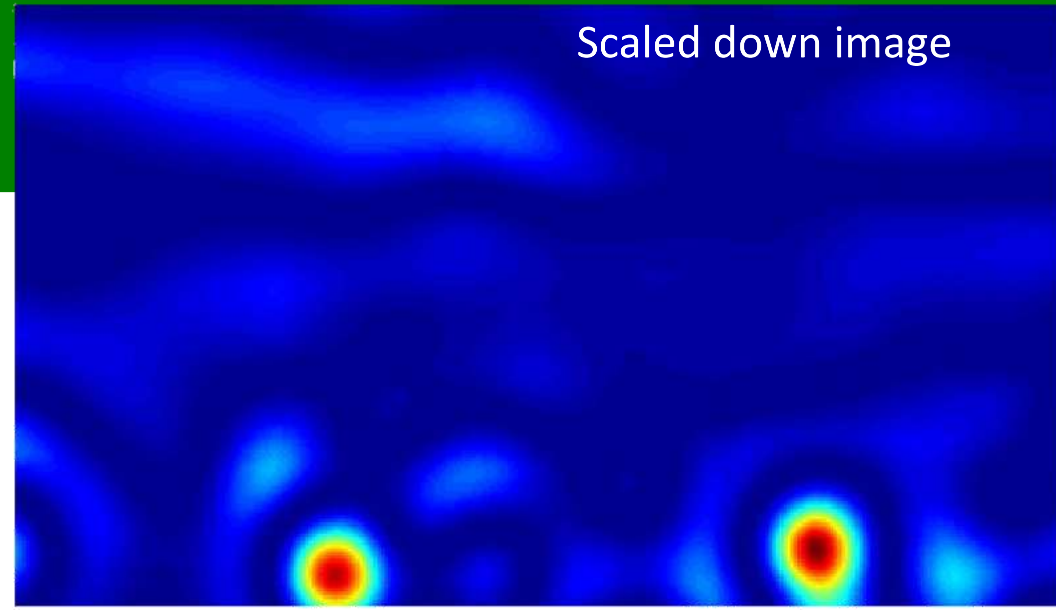
Slide credit: Kristen Grauman

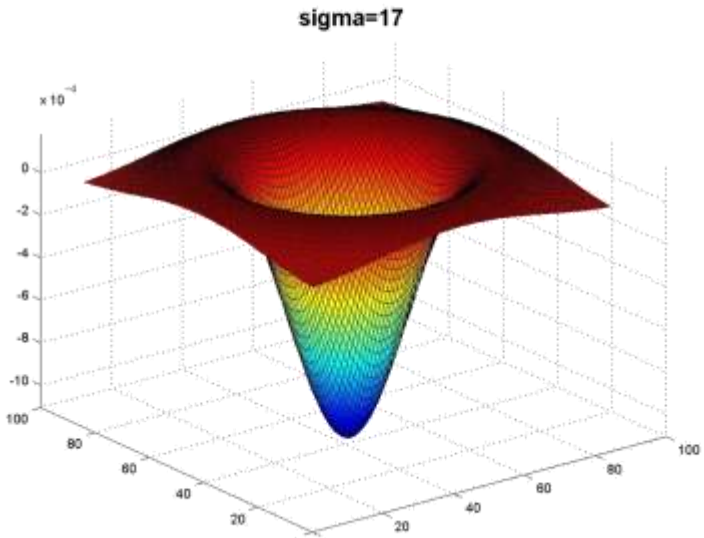


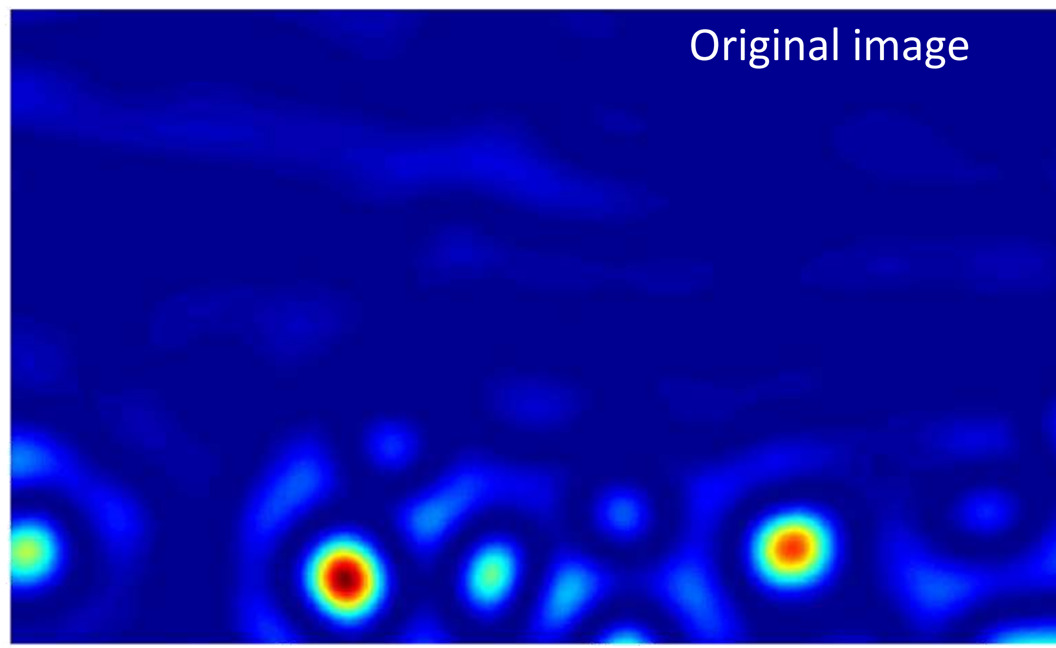




Slide credit: Kristen Grauman



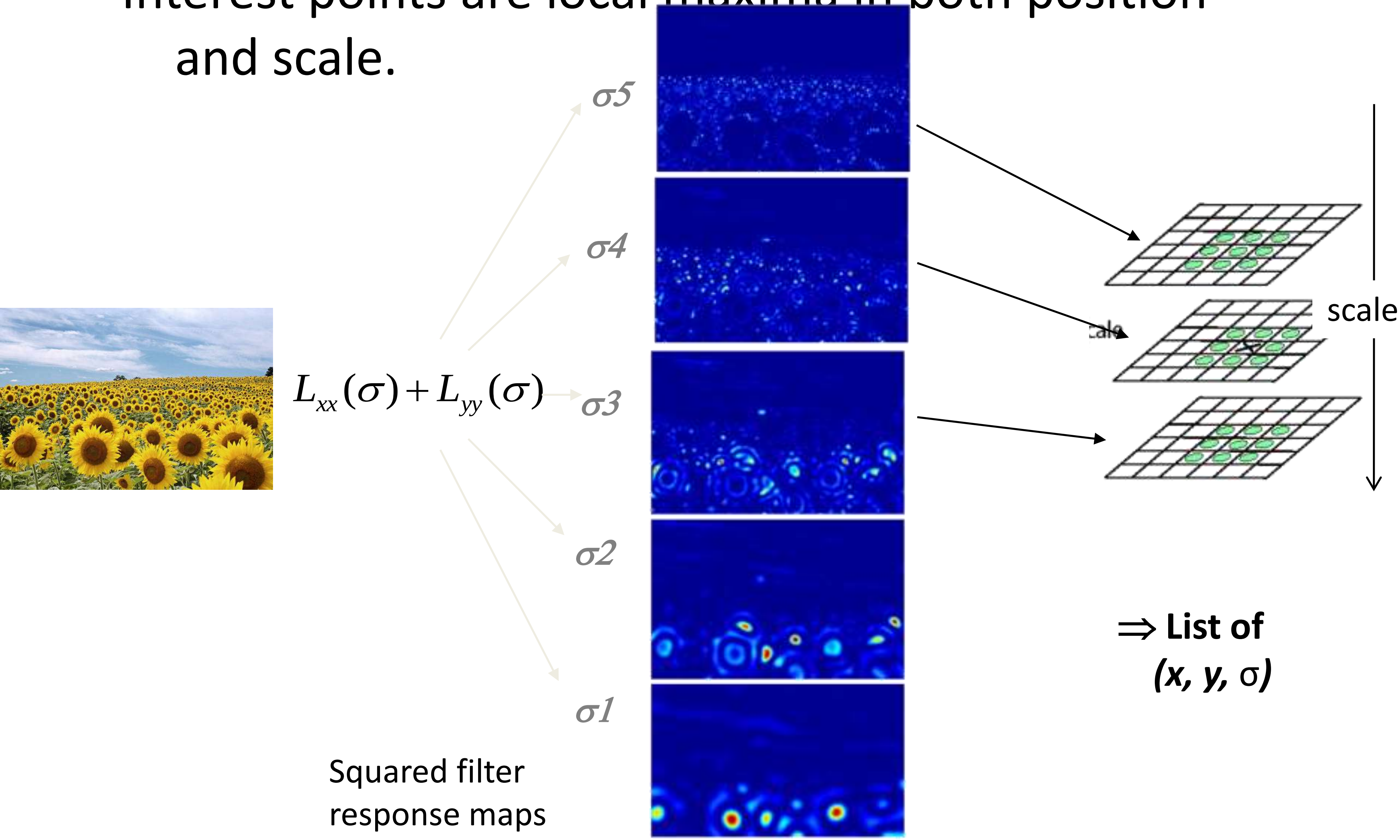




Slide credit: Kristen Grauman

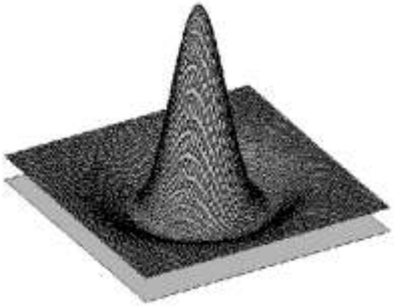
Scale invariant interest points

Interest points are local maxima in both position and scale.



⇒ **List of *(x, y, σ)***

Difference-of-Gaussian (DoG)

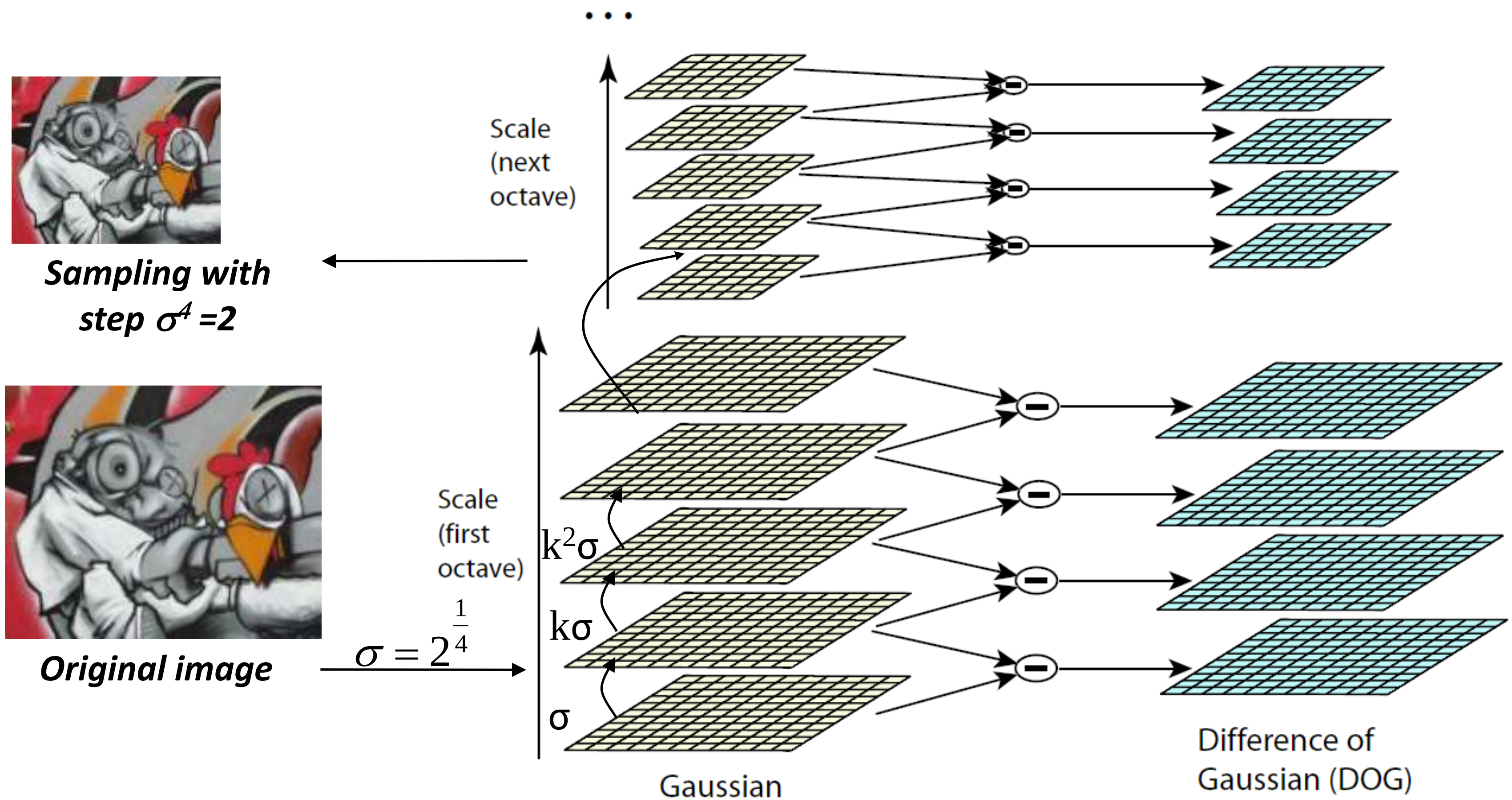


-

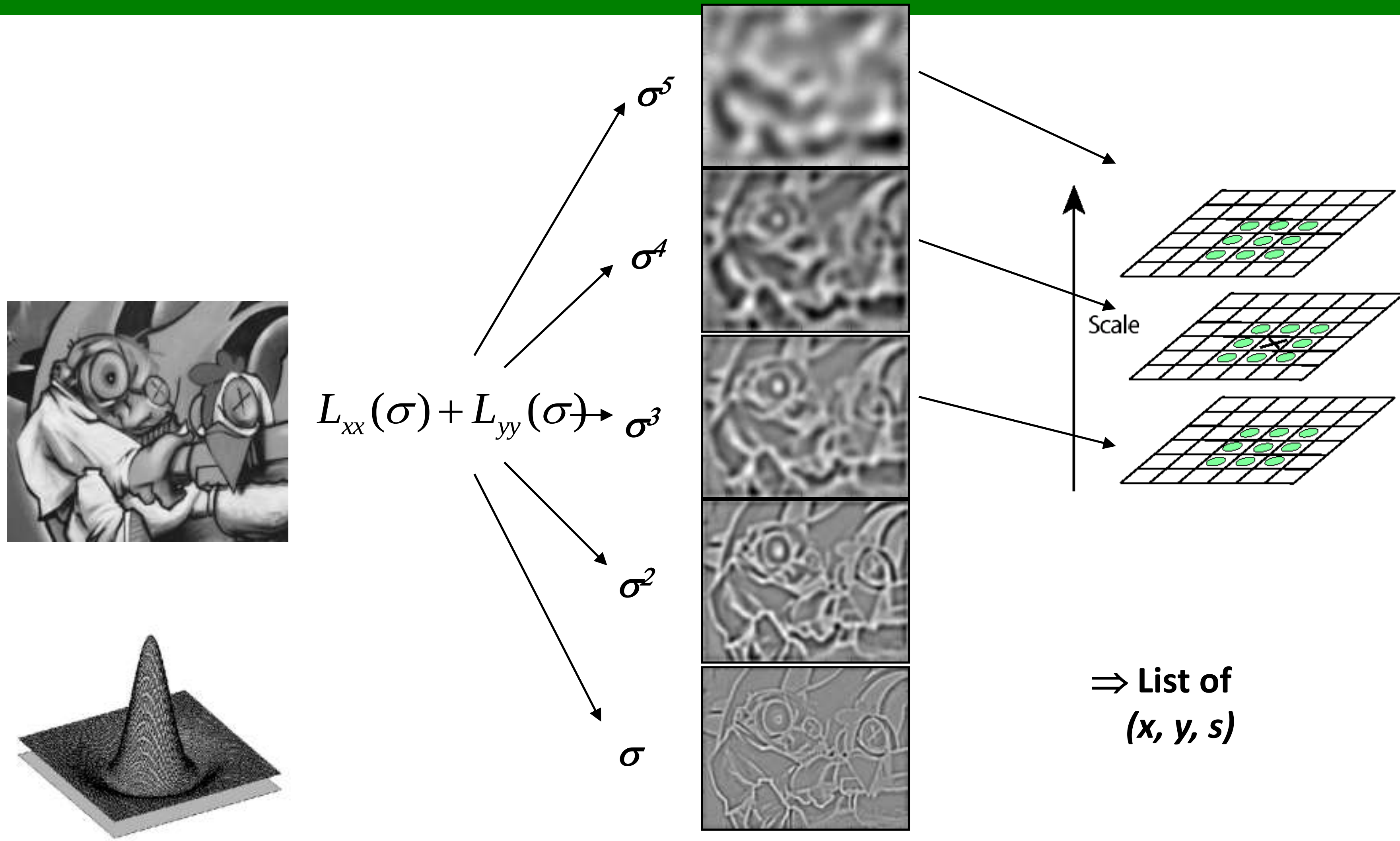
=

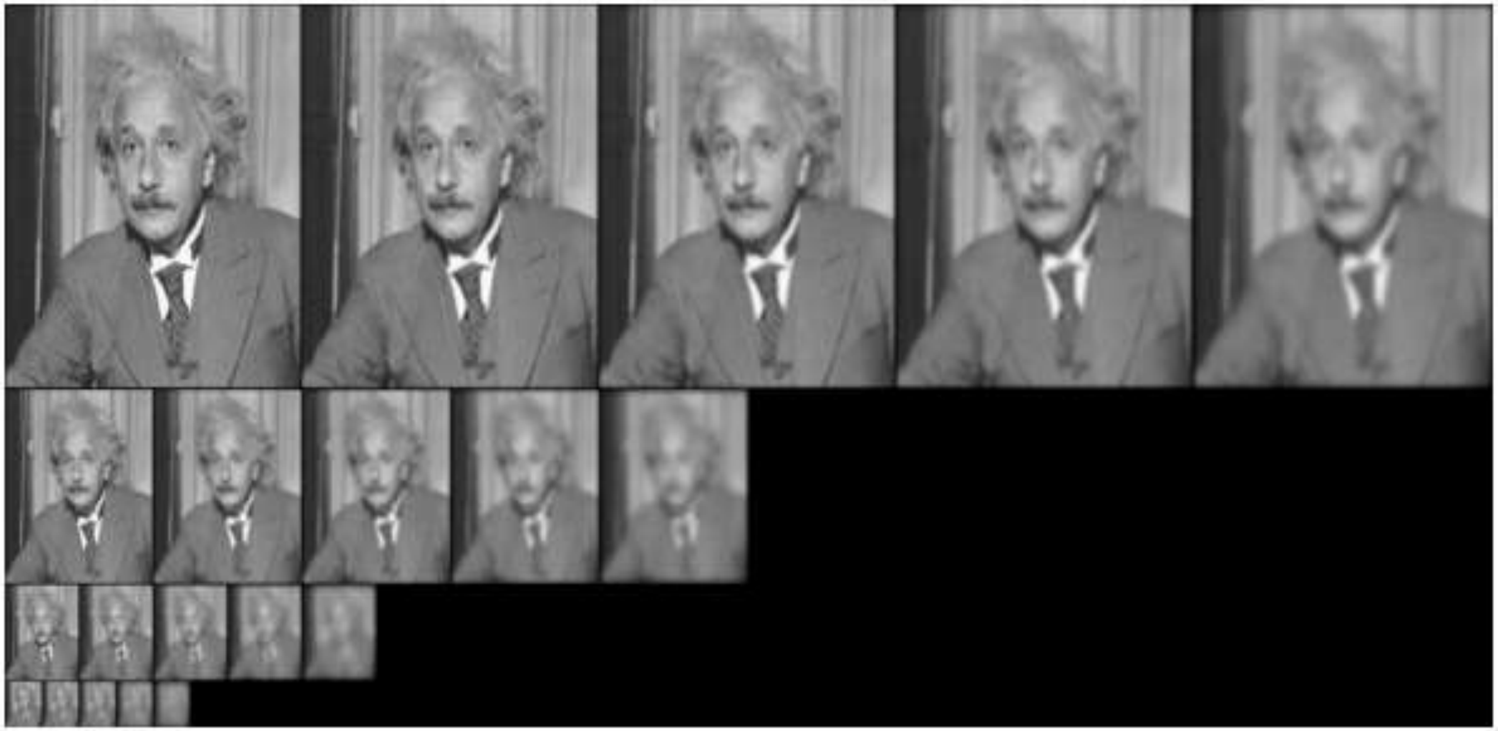
DoG – Efficient Computation

- Computation in Gaussian scale pyramid



Find local maxima in position-scale space of Difference-of-Gaussian

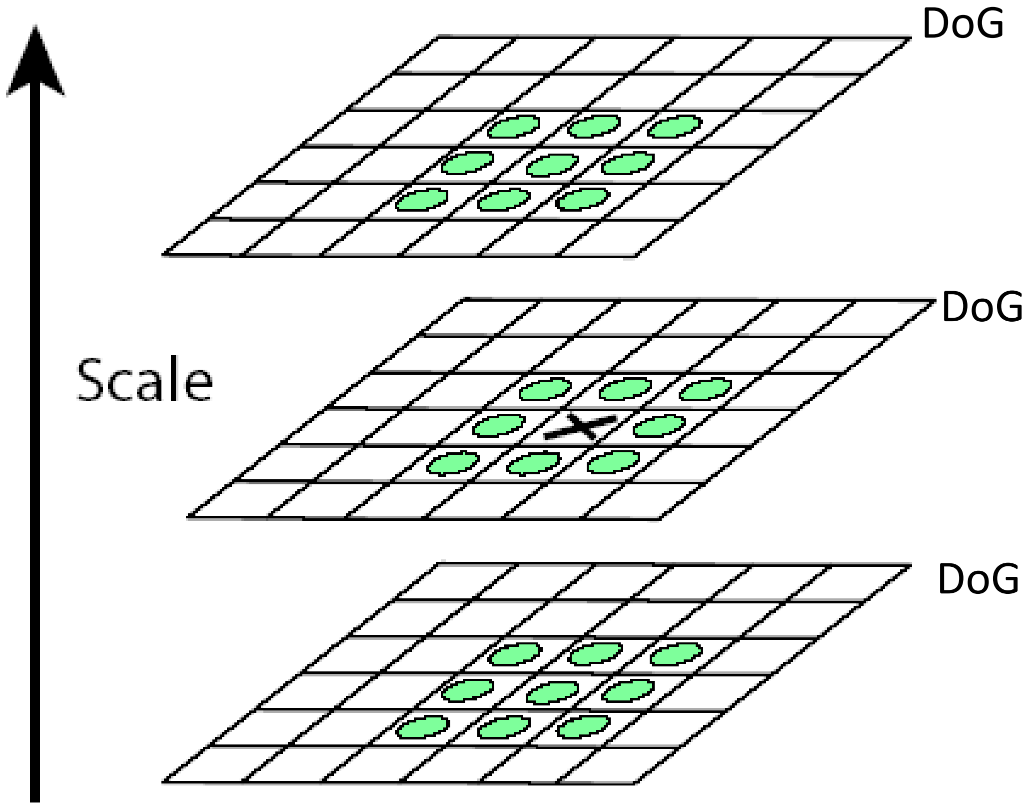




Gaussian images

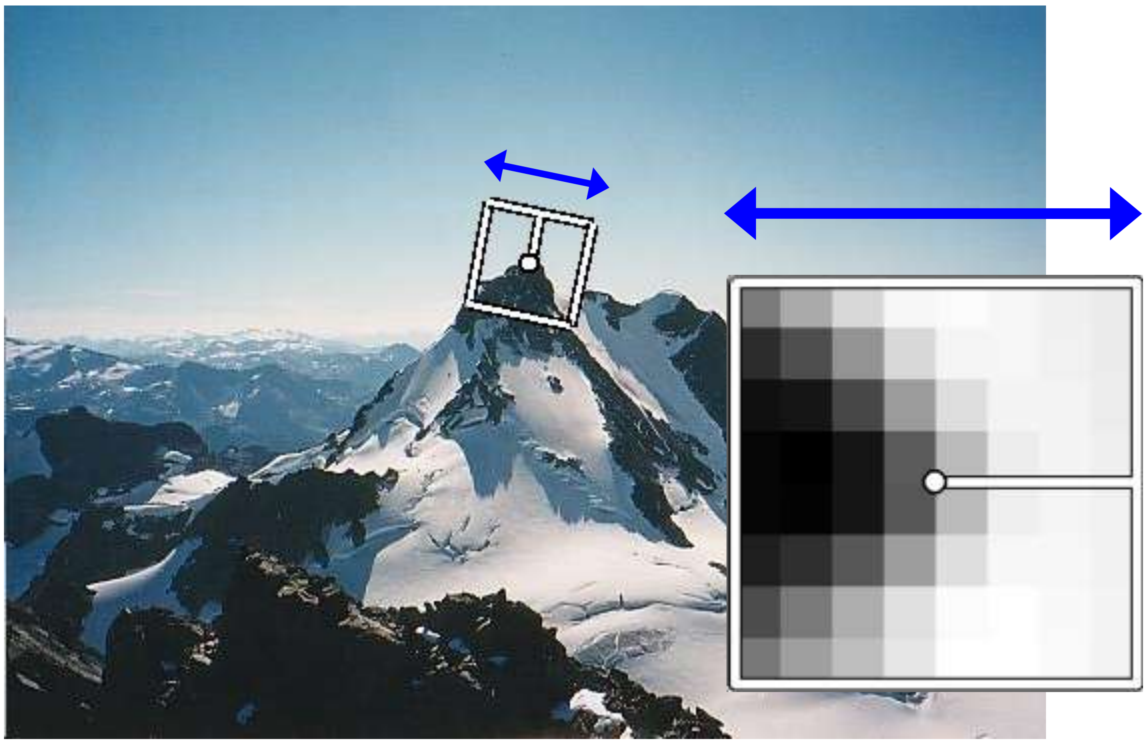
DoG images

Scale-space extrema detection



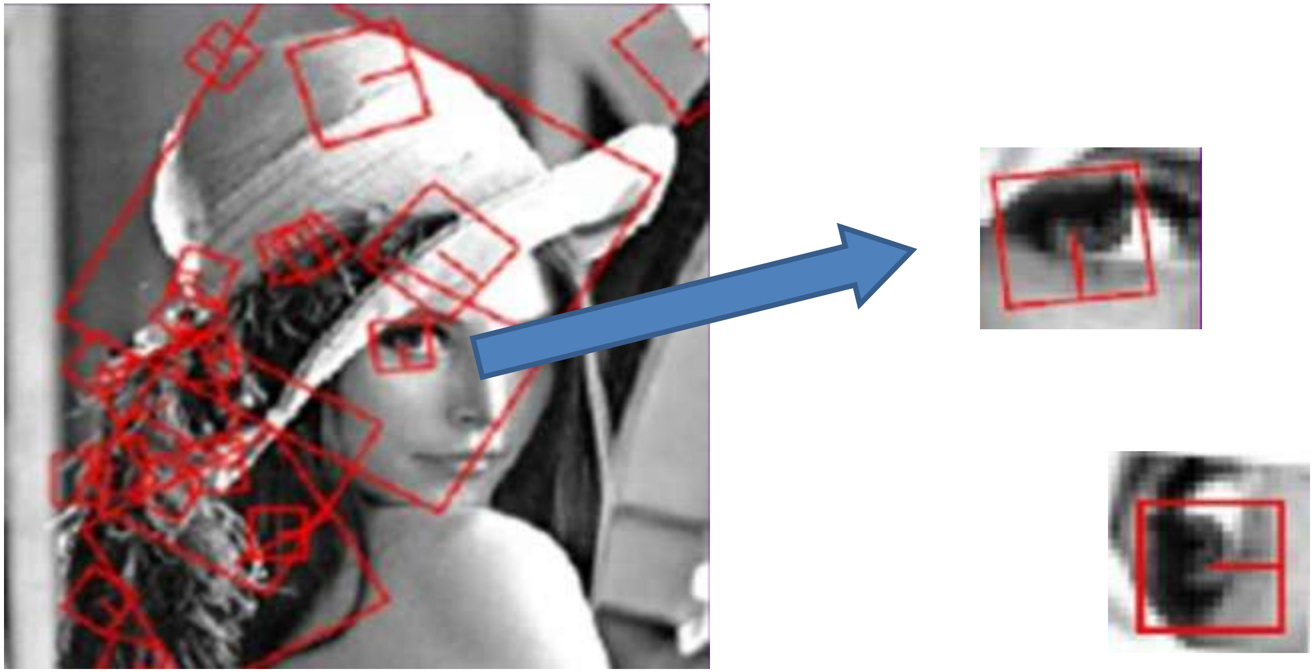
If ***X*** is the *largest* or the *smallest* of all of its neighbors, ***X*** is called a ***keypoint***.

Making descriptor rotation invariant



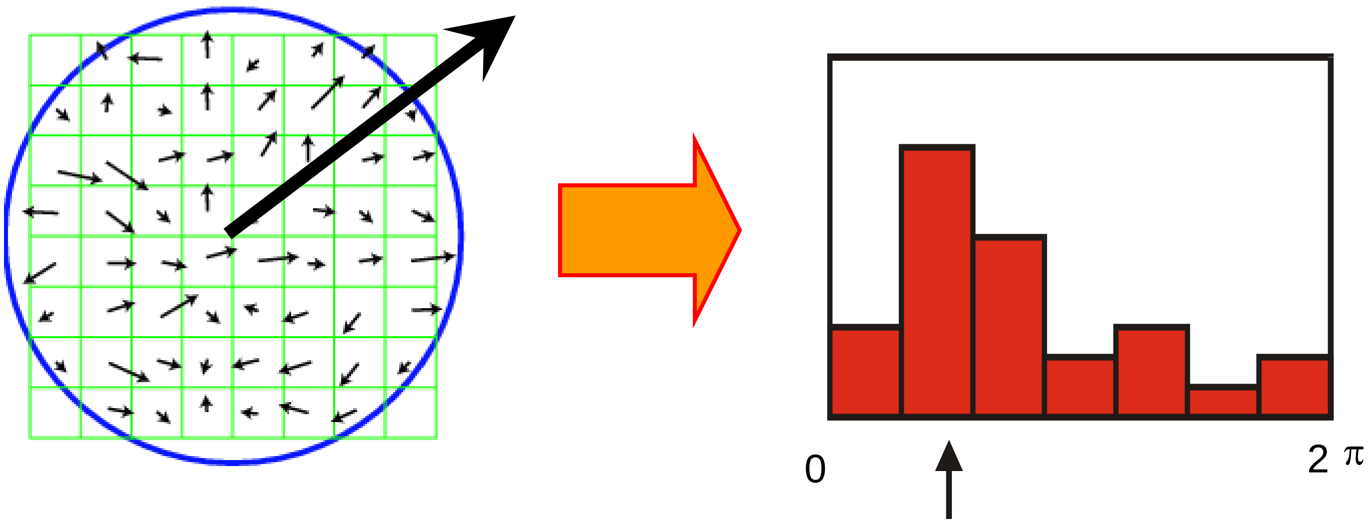
- Rotate patch according to its *dominant* gradient orientation
- This puts the patches into a canonical orientation.

Orientation assignment



Eliminating rotation ambiguity

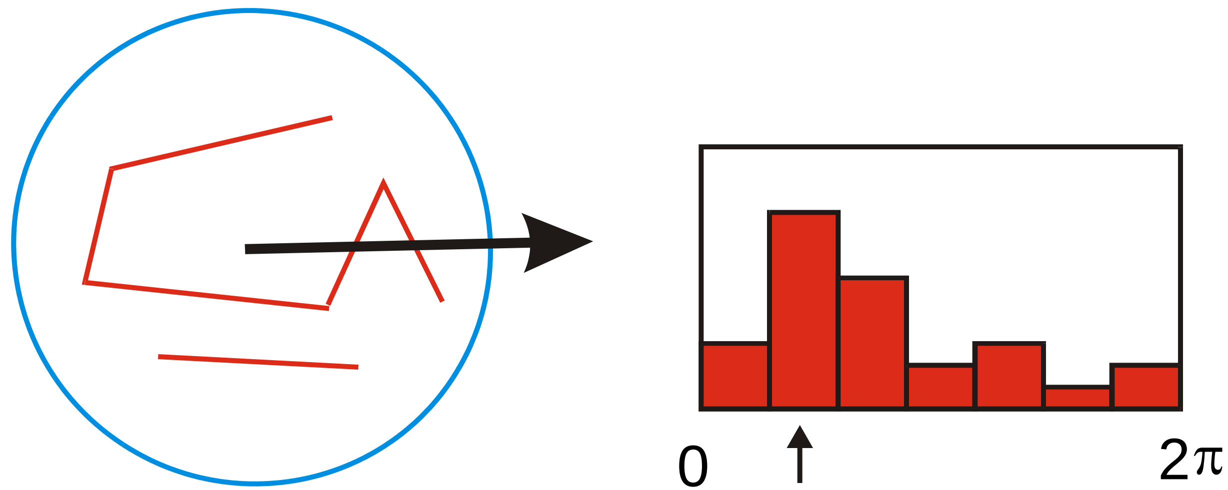
- To assign a unique orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram



Orientation Normalization

[Lowe, SIFT, 1999]

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation



Orientation assignment

- To achieve invariance to ***rotation***
- Compute gradient magnitude and orientation for each image sample *L*(x, y, σ)

$$m = \sqrt{(L_{x+1,y} - L_{x-1,y})^2 + (L_{x,y+1} - L_{x,y-1})^2}$$

$$\theta = \tan^{-1}((L_{x,y+1} - L_{x,y-1})/(L_{x+1,y} - L_{x-1,y}))$$

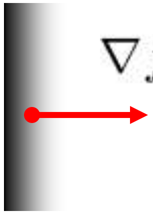
- Form an ***orientation histogram*** from the gradient orientations of sample points within a region around the keypoint, weighted by its gradient magnitude and a Gaussian-weighted window
- Detect the highest peak

Review: Image gradient

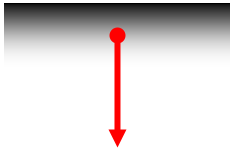
The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

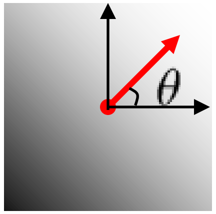
The gradient points in the direction of most rapid change in intensity



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The **gradient direction** is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The **gradient magnitude** is given by:

$$||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

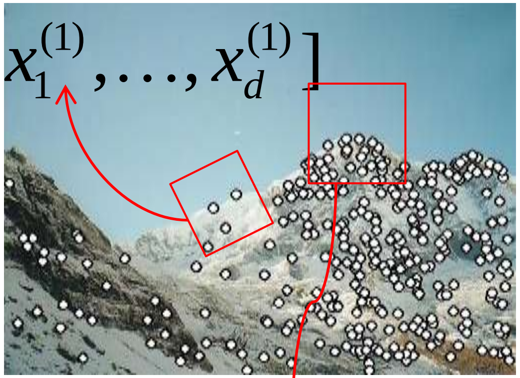
Local features: main components

1) Detection: Identify the interest points

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

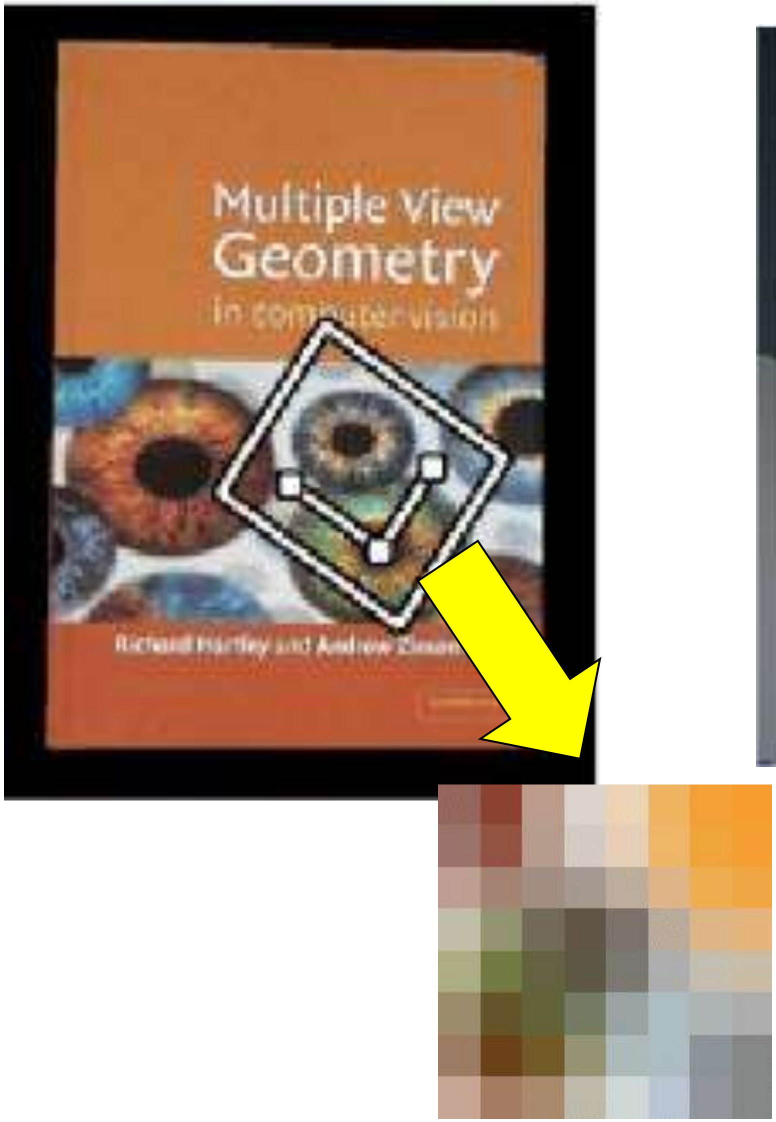
3) Matching: Determine correspondence between descriptors in two views

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



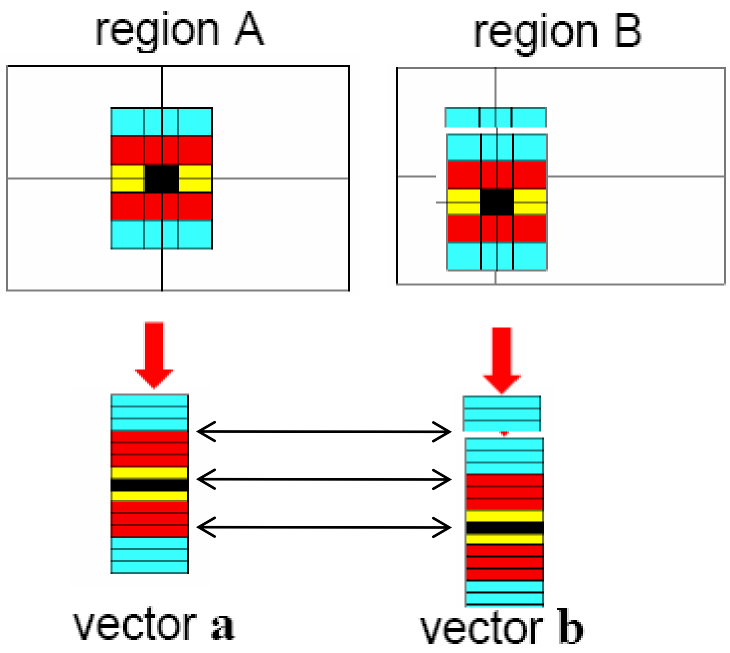
$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

Geometric transformations



e.g. scale,
translation,
rotation

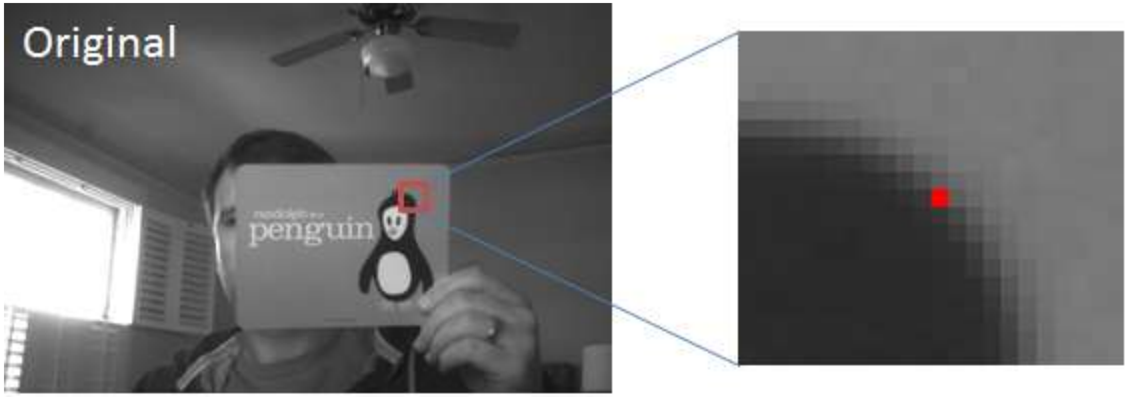
Raw patches as local descriptors



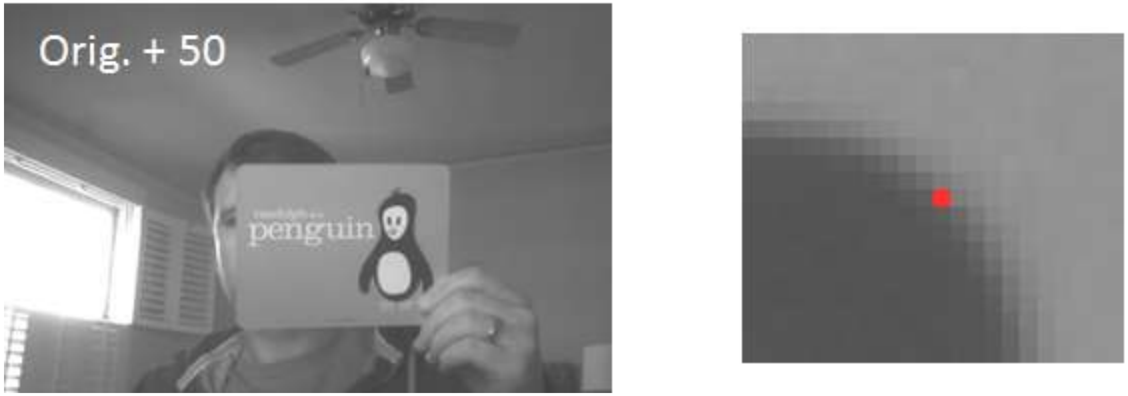
- The simplest way to describe the neighborhood around an interest point is to use *intensities* to form a feature vector.
- But this is very sensitive to even small shifts, rotations.

Gradient vectors

- The first image shows a pixel, highlighted in red, in the original image.

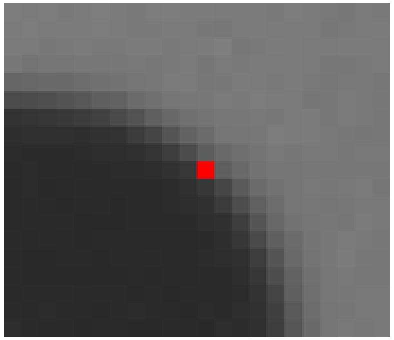


- In the second image, all pixel values have been increased by 50.



Gradient vectors

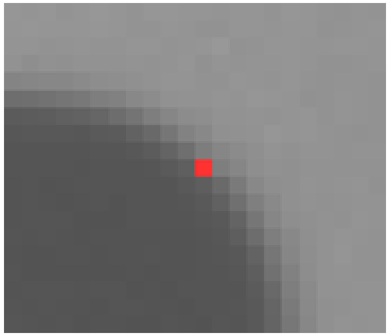
-1	0	1



	93	
56		94
	55	

$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$



	143	
106		144
	105	

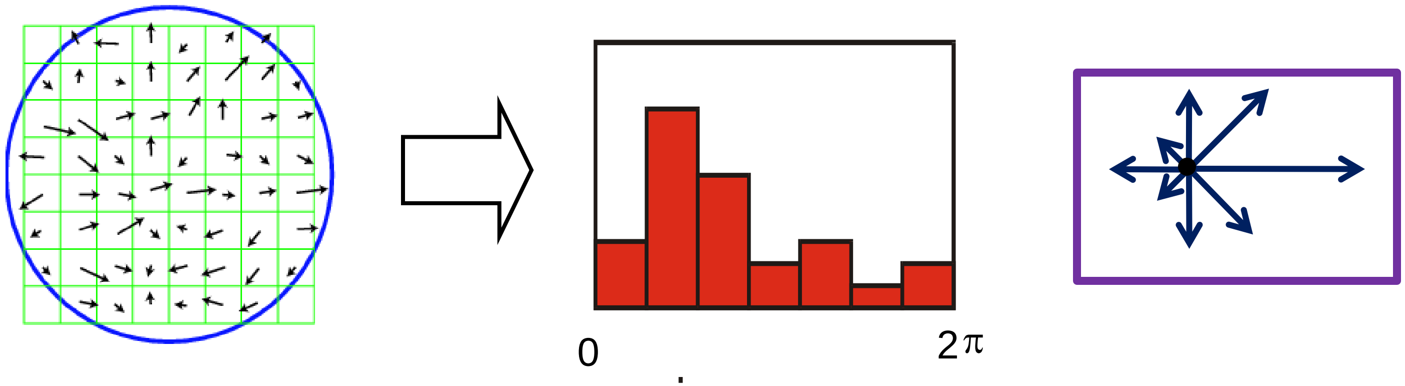
$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$

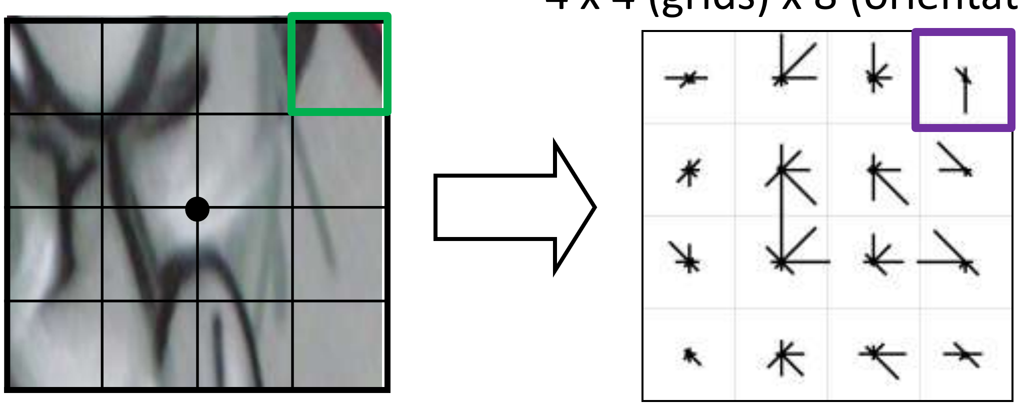
- The gradient vectors are equivalent in the first and second images

Local image descriptor

- Why does SIFT have some illumination invariance?

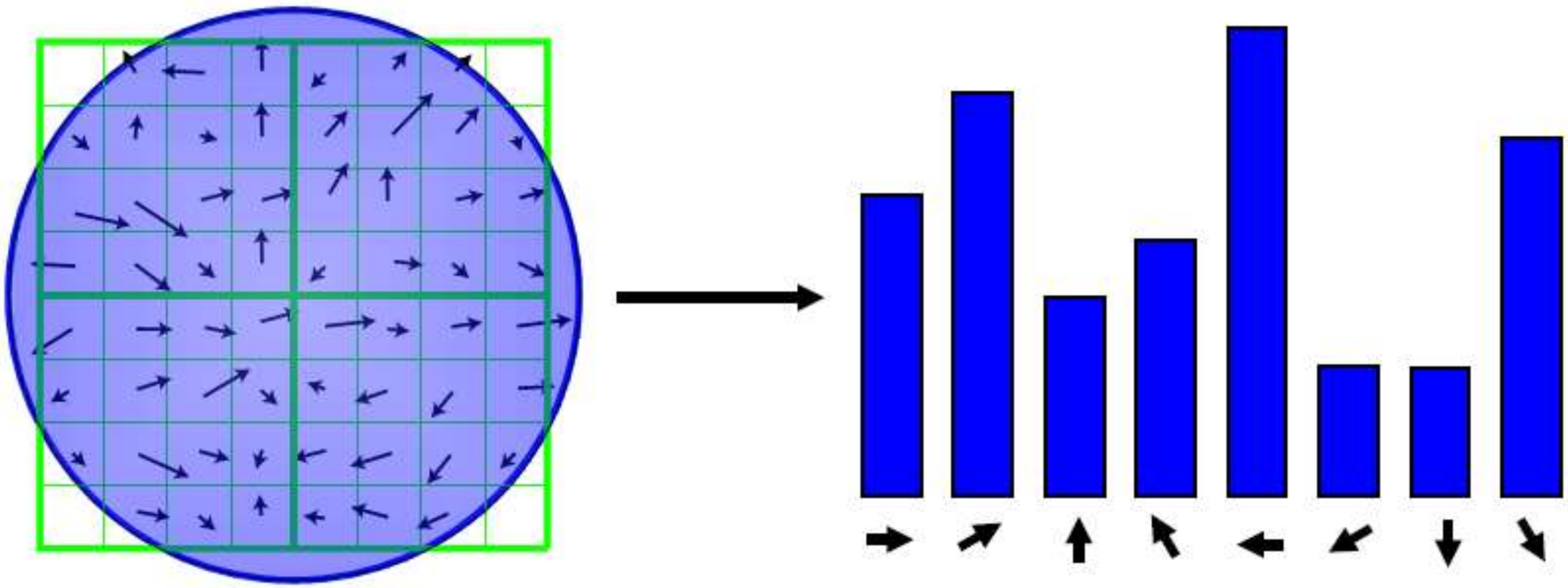


4 x 4 (grids) x 8 (orientations)



Gradient histogram

- 8 (orientations)



SIFT descriptors

- Inspiration: complex neurons in the primary visual cortex

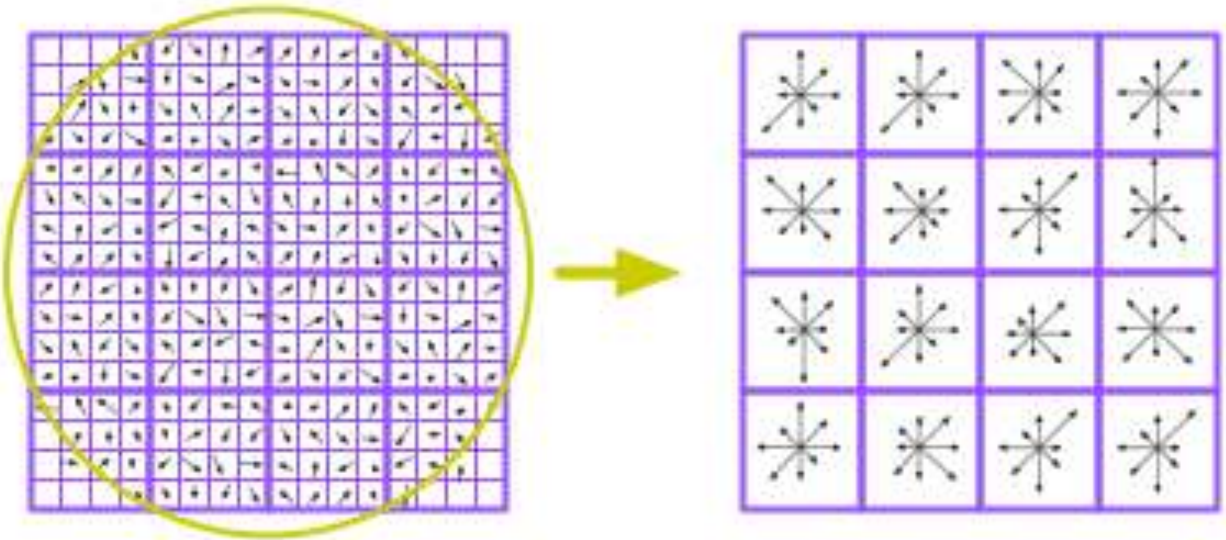


Image gradients Keypoint descriptor

Use a 4x4 grid computed from a 16x16 sample array
128-d = 4 x 4 x 8 (orientations)

D. Lowe. **[Distinctive image features from scale-invariant keypoints.]()** *IJCV* 2004.

621 128

Number of keypoints
Feature dimension

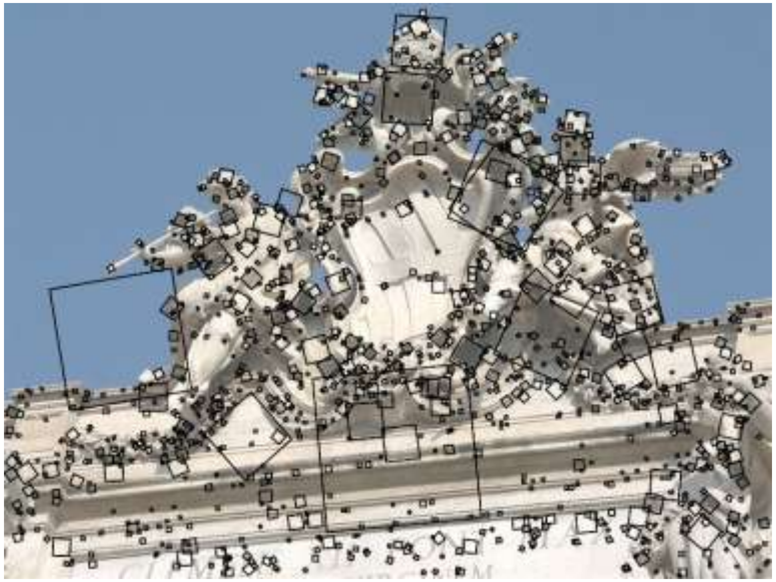
162.38 155.79 44.30 2.615
7 6 0 0 0 0 0 1 58 63 1 0 7 6 1 8 8 9 0 0
24 42 39 14 0 0 0 0 0 0 7 2 44 7 0 0 23 22 6
69 137 64 0 0 0 0 11 137 55 12 0 0 2 25 137
112 0 0 0 0 3 17 30 6 34 1 0 0 20 51 137 89
137 89 0 0 0 15 115 102
137 47 0 0 4 37 26 43 0 0 0 0 19 45 4 0 0 0 0
0 0 16 137 53 33 2 0 0 0 56 137 51 57 2 0 0
0 3 14 35 0 0 0 0 0 0 2 0 0

282.47 185.76 27.80 2.009
0 0 0 0 0 0 0 0 1 41 13 1 0 12 4 0 5 17 15 16
17 83 35 16 19 0 0 1 2 13 24 104 0 1 9 0 0 0
0 0 22 127 127 5 0 0 0 1 127 127 75 16 6 0 0
70 55 2 0 1 0 0 25 127 1 1 9 0 0 1 1 2 115 22
49 4 0 0 0 68
127 127 30 4 0 0 0 58 67 127 69 0 0 0 5 20 2
0 0 0 4 65 5 2 85 50 6 0 1 15 2 30 56 93 53
19 0 0 4 41 22 127 86 1 0 2 17 20
……….

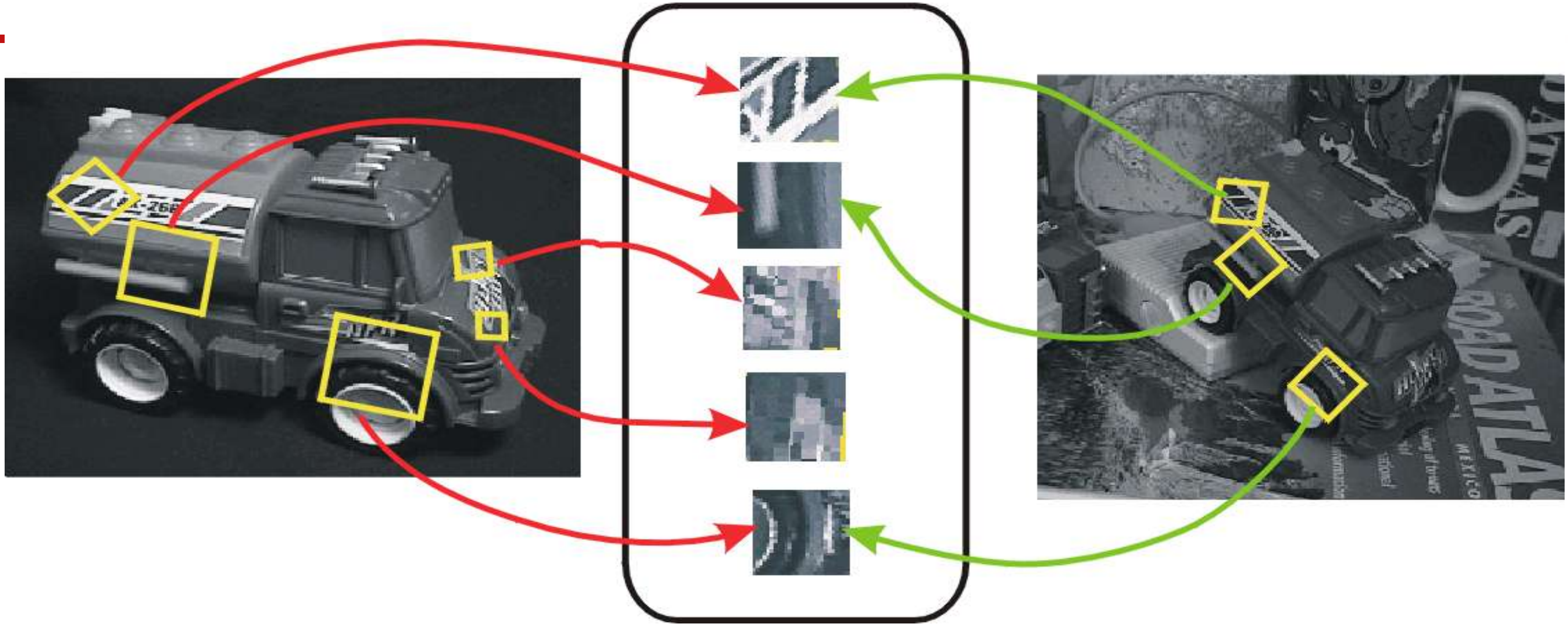
row col scale orientation (-pi~pi)
128 integers (0~255)

SIFT features

- Detected features with characteristic scales and orientations:



From feature detection to feature description



- Description is *invariant*:

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

Details of Lowe's SIFT algorithm

- Run DoG detector
  - Find maxima in location/scale space
  - Remove edge points
- Find all major orientations
  - Bin orientations into 36 bin histogram
    - Weight by gradient magnitude
    - Weight by distance to center (Gaussian-weighted mean)
  - Return orientations within 0.8 of peak
    - Use parabola for better orientation fit
- For each (x,y,scale,orientation), create descriptor:
  - Sample 16x16 gradient mag. and rel. orientation
  - Bin 4x4 samples into 4x4 histograms
  - Threshold values to max of 0.2, divide by L2 norm
  - Final descriptor: 4x4x8 normalized histograms

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

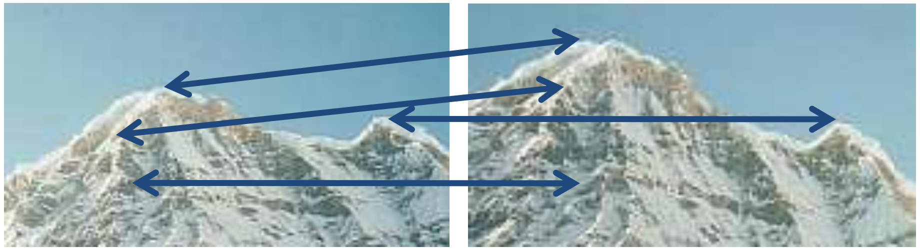
Lowe IJCV 2004

Local features: main components

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

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

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



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