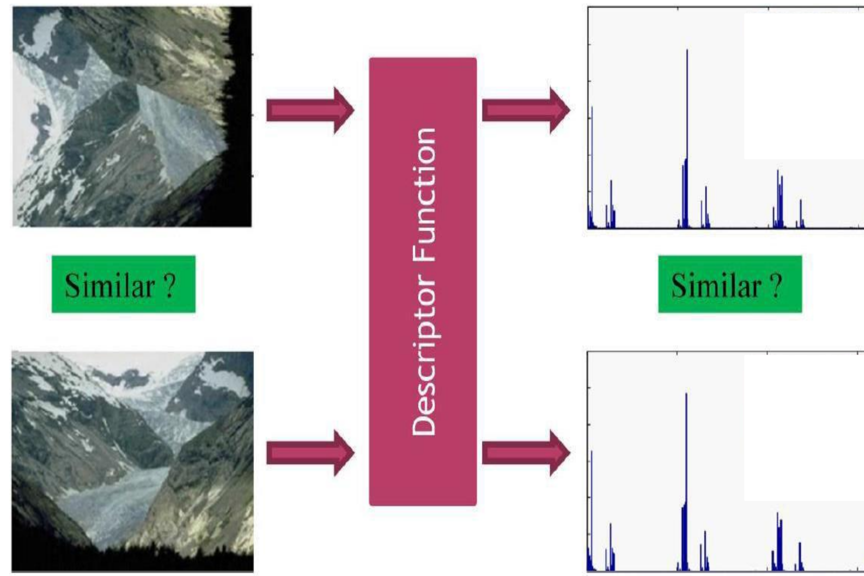
Computer Vision

Region Detection & Local Descriptors

Dr. Mrinmoy Ghorai

Indian Institute of Information Technology
Sri City, Chittoor

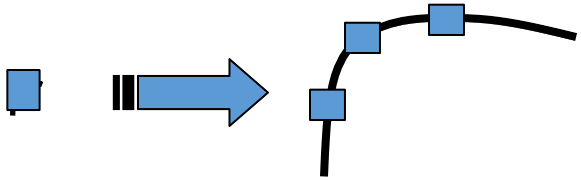
Region Detection & Local Descriptors



Previous Class

- Keypoint detection: repeatable and distinctive
  - Corners, Harris
  - Invariant to scale, rotation, etc.

- Harris Corner Detection
  - Rotation Invariant
  - Partial Intensity Change Invariant
  - *Not Invariant to Scale*

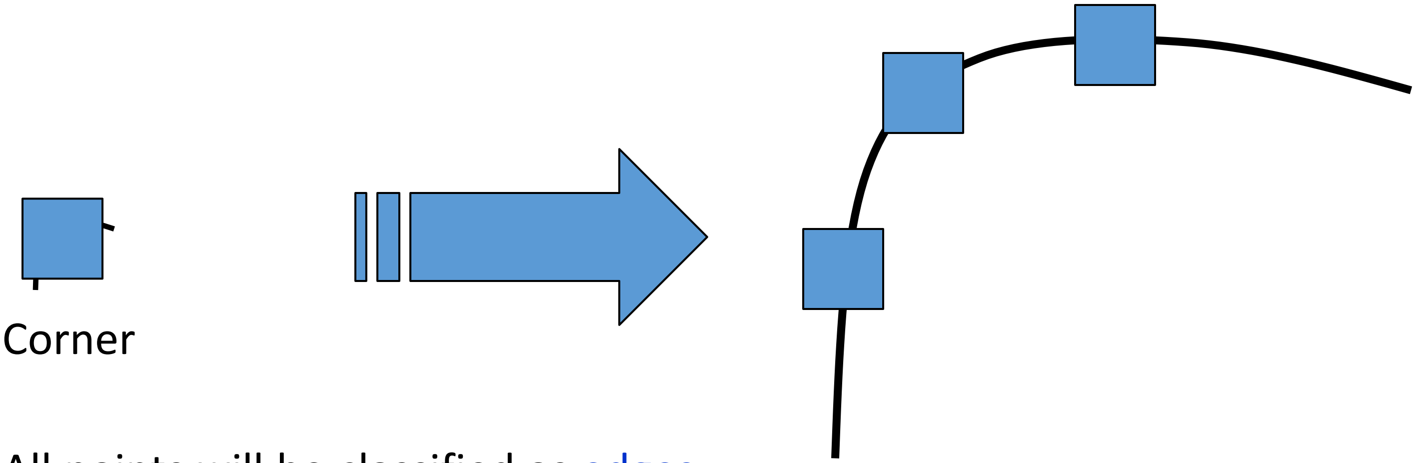


Today’s class

- Scale Invariance
- Region Detection
- Local Descriptors
- Image Matching

Harris Detector: Invariance Properties

- Scaling

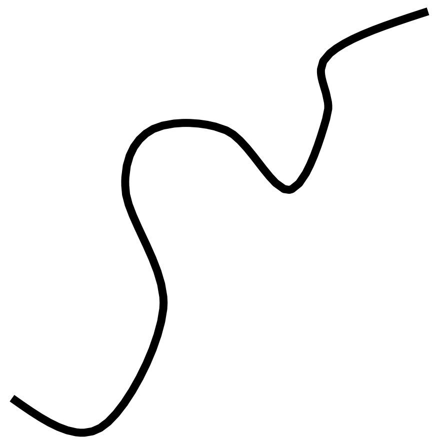


All points will be classified as edges

Not invariant to scaling

Scale invariant detection

Suppose you're looking for corners

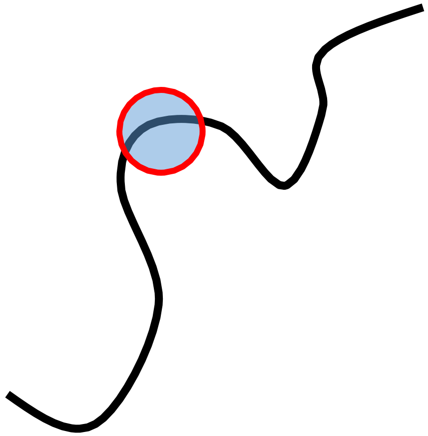


Key idea: find scale that gives local maximum of *f*

- in both position and scale
- One definition of *f*: the Harris operator

Scale invariant detection

Suppose you're looking for corners

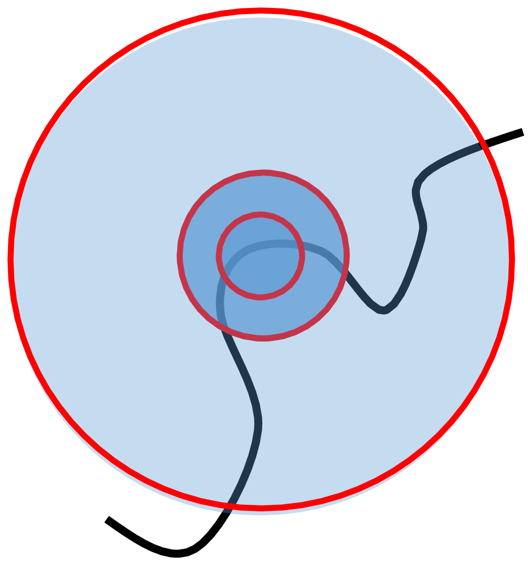


Key idea: find scale that gives local maximum of *f*

- in both position and scale
- One definition of *f*: the Harris operator

Scale invariant detection

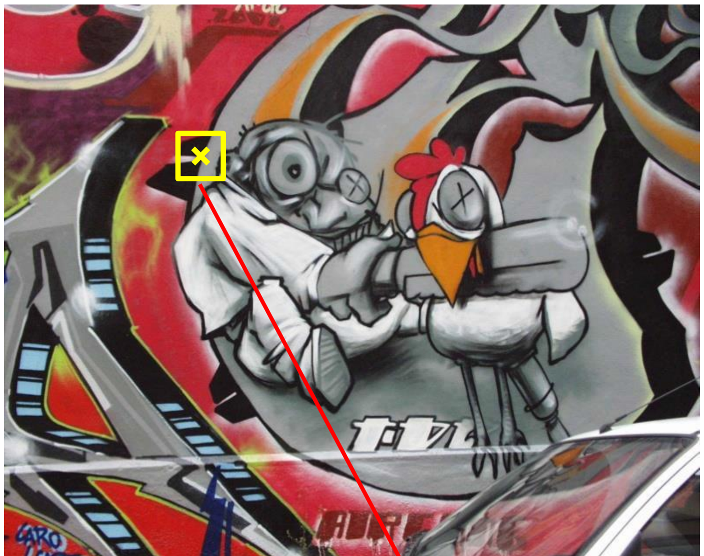
Suppose you're looking for corners



Key idea: find scale that gives local maximum of *f*

- in both position and scale
- One definition of *f*: the Harris operator

Automatic Scale Selection

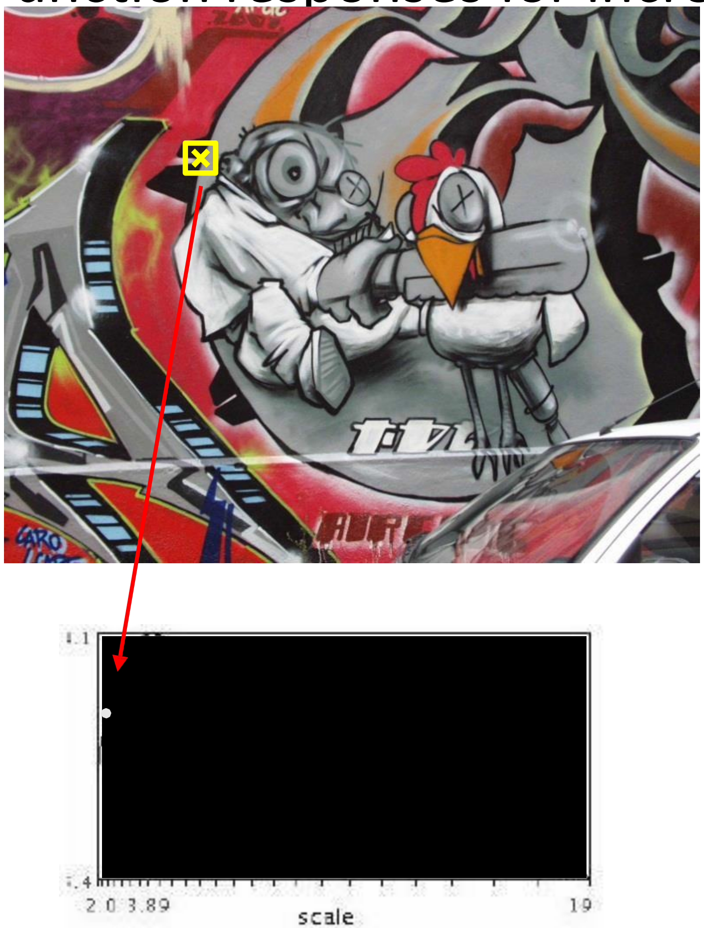


$$f(I_{i_1\ldots i_m}(x,\sigma)) = f(I_{i_1\ldots i_m}(x',\sigma'))$$

How to find corresponding patch sizes?

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

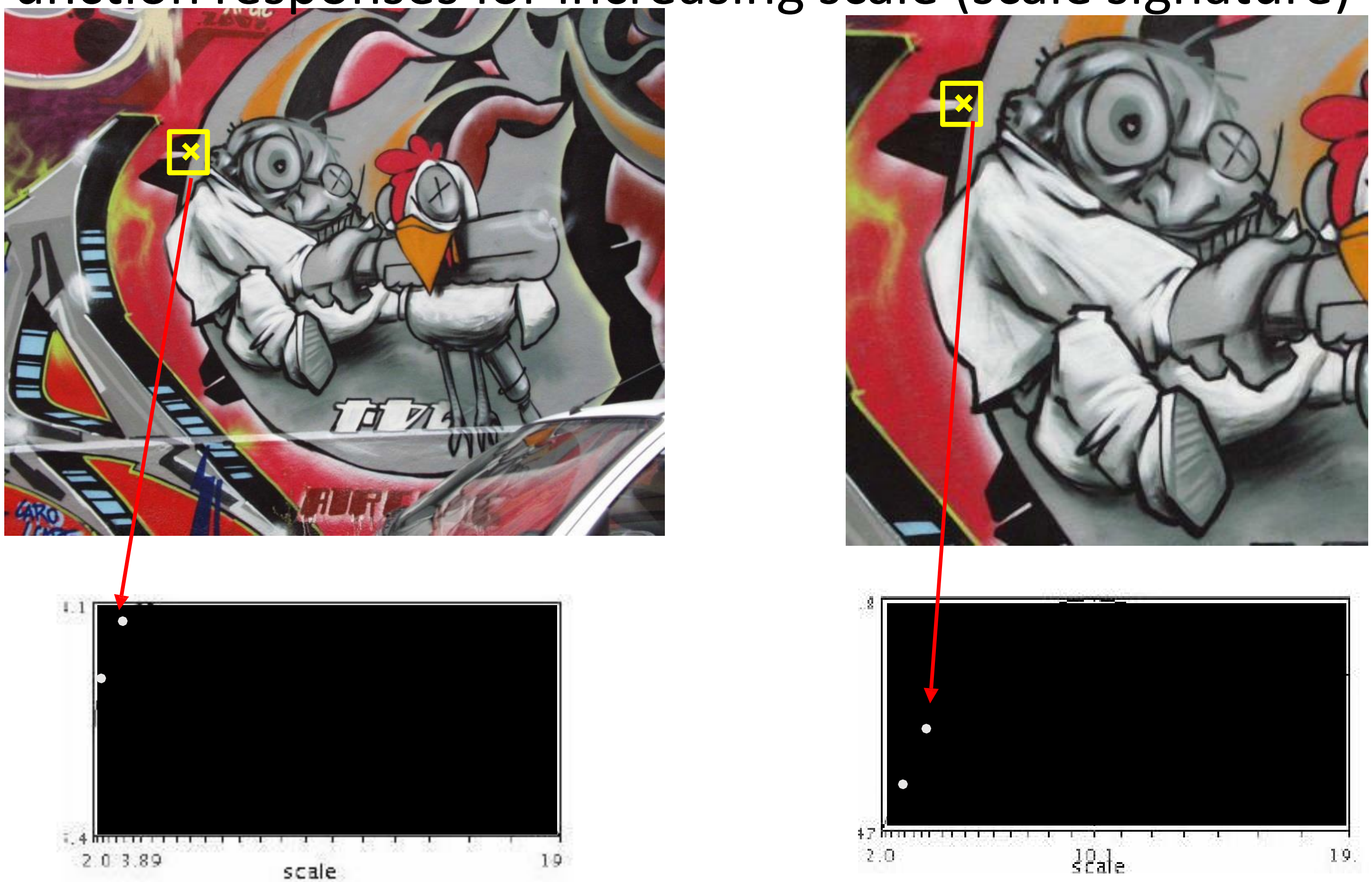


$f(I_{i_1 \dots i_m}(x, \sigma))$

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

Automatic Scale Selection

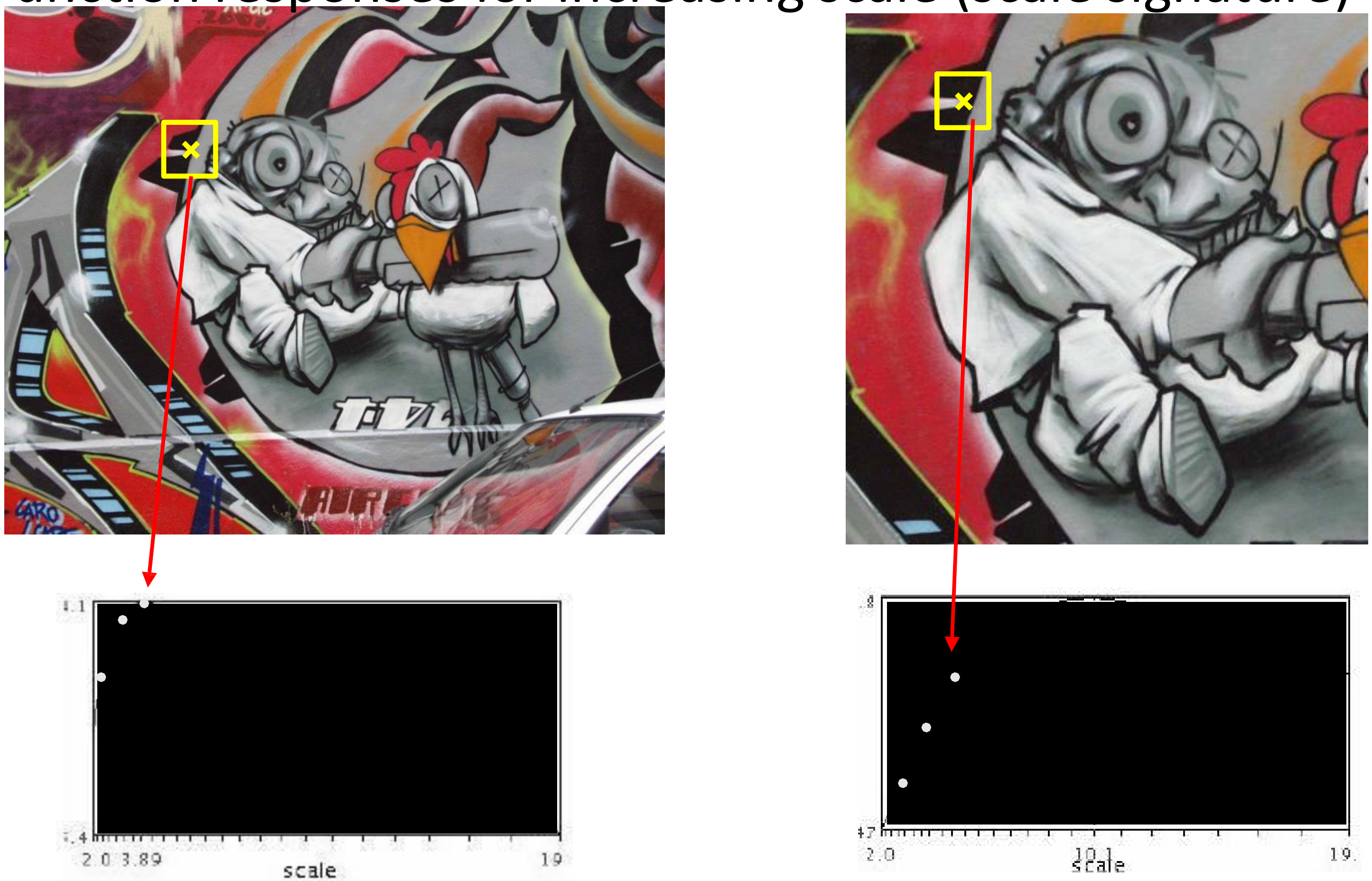
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

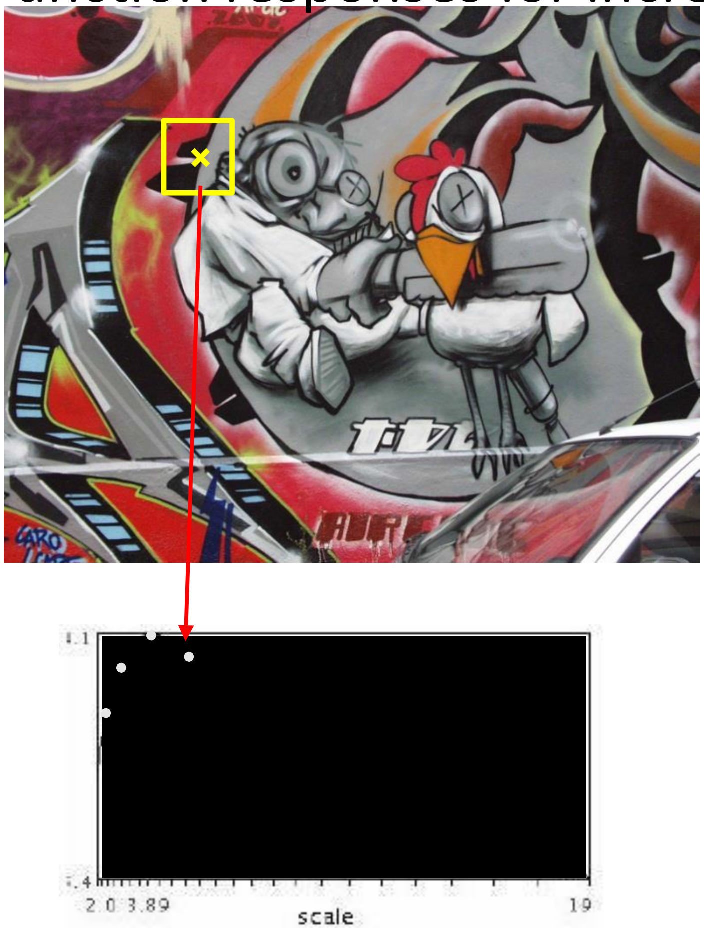
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



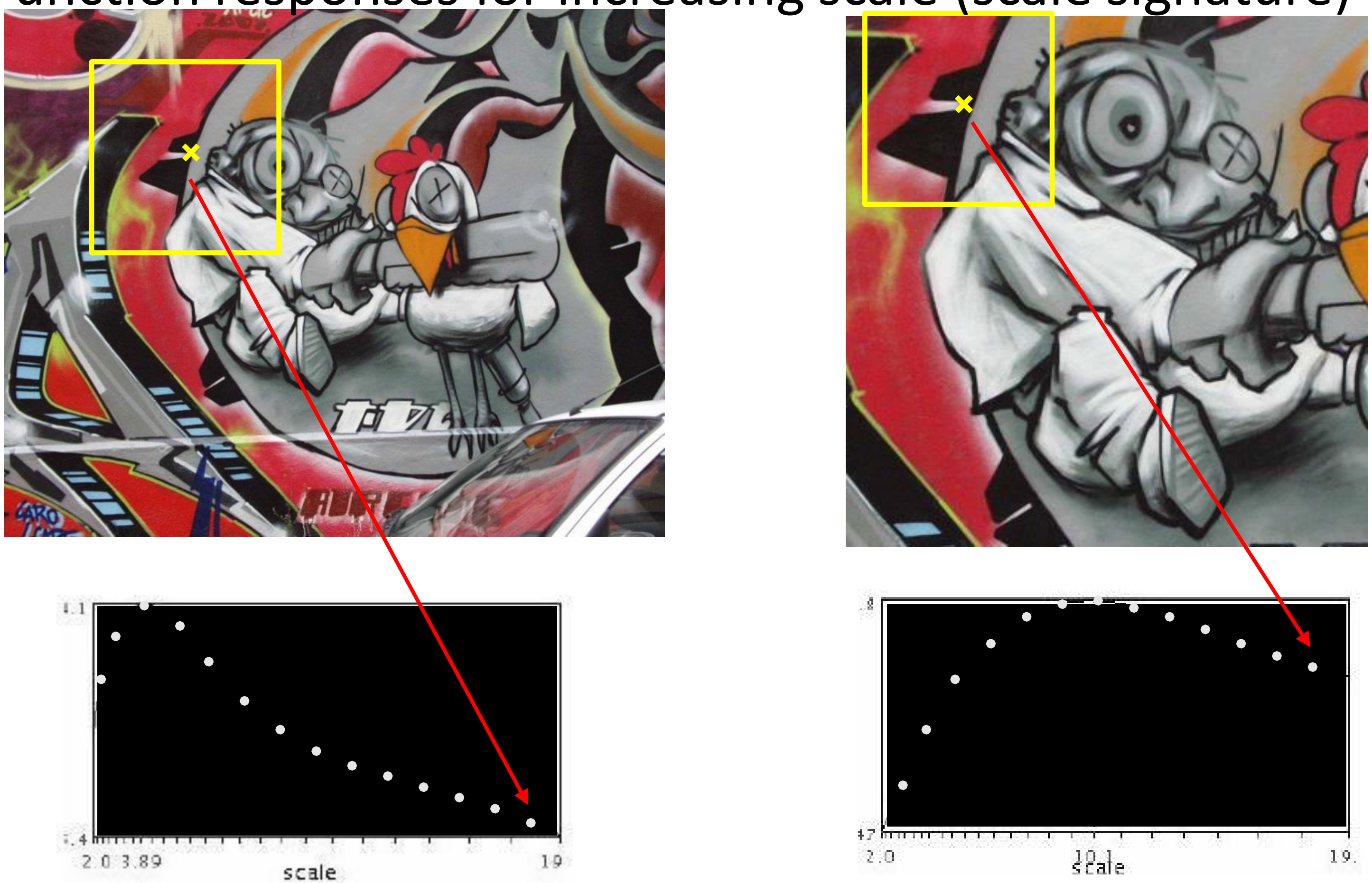
$f(I_{i_1 \dots i_m}(x, \sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

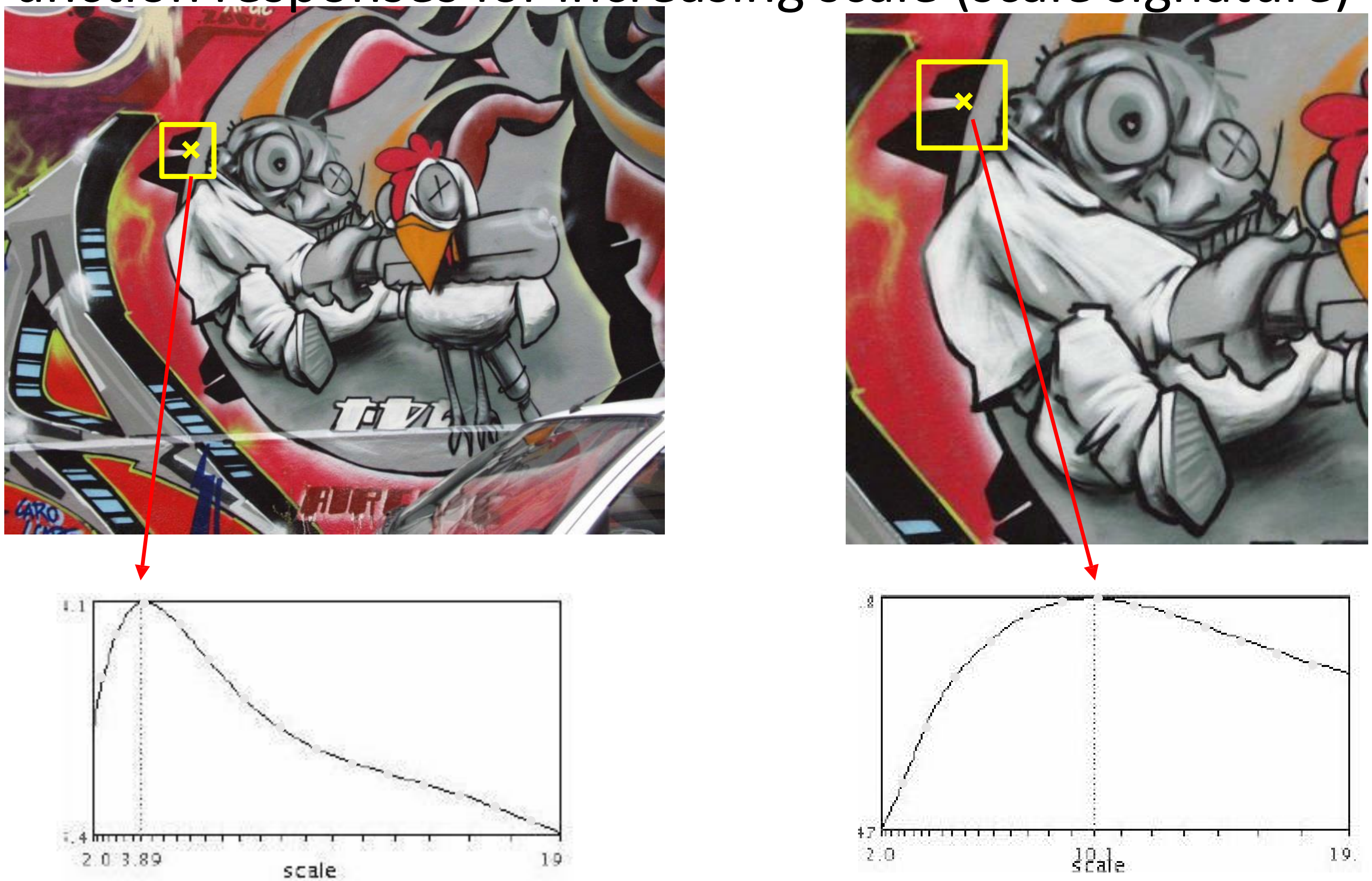


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

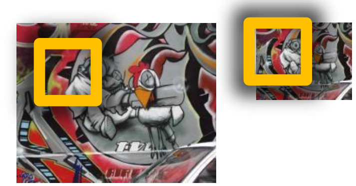


$f(I_{i_1 \ldots i_m}(x, \sigma))$ $f(I_{i_1 \ldots i_m}(x', \sigma'))$

K. Grauman, B. Leibe

Implementation

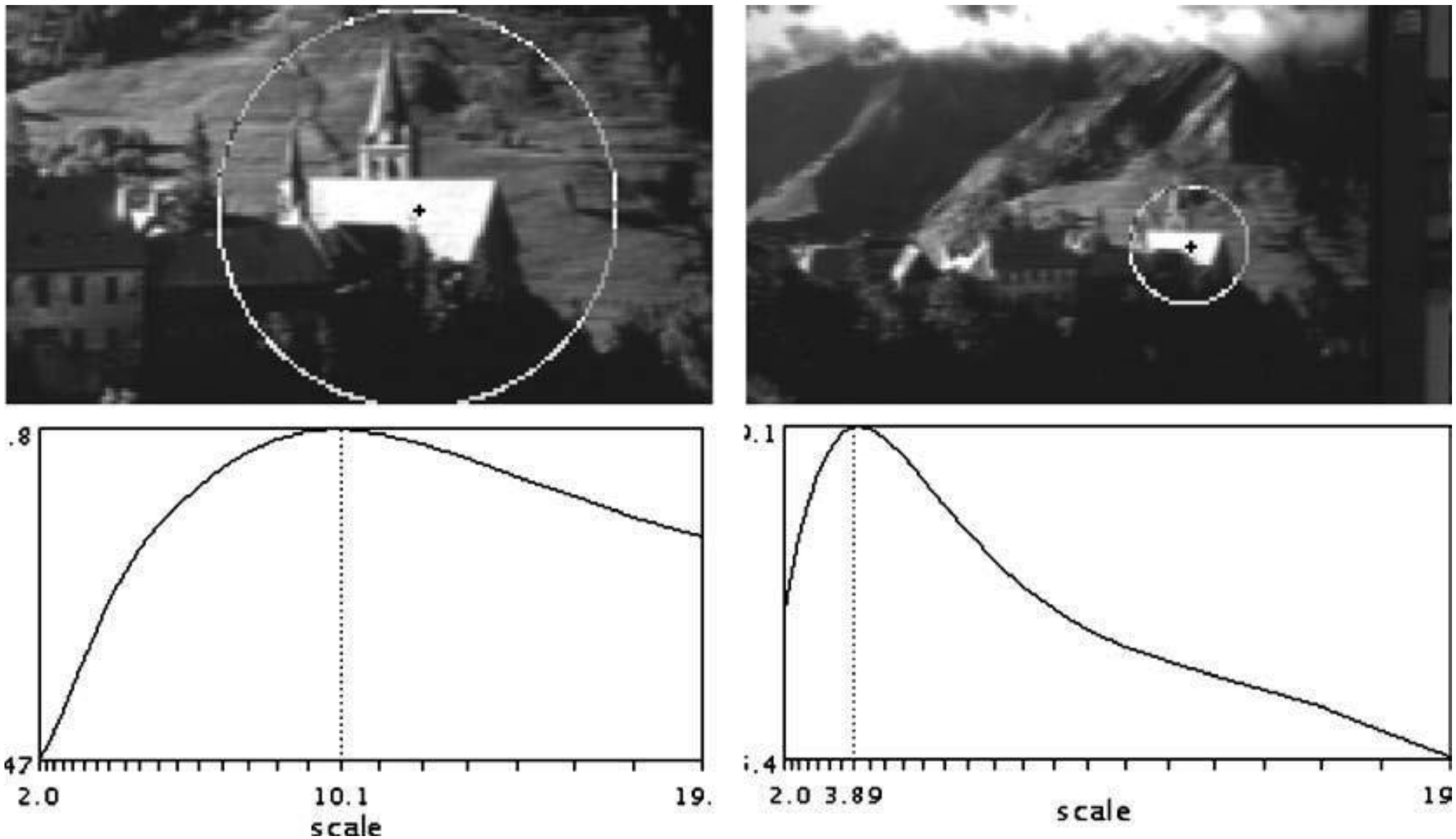
- Instead of computing *f* for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid



(sometimes need to create in-between levels, e.g. a ¾-size image)

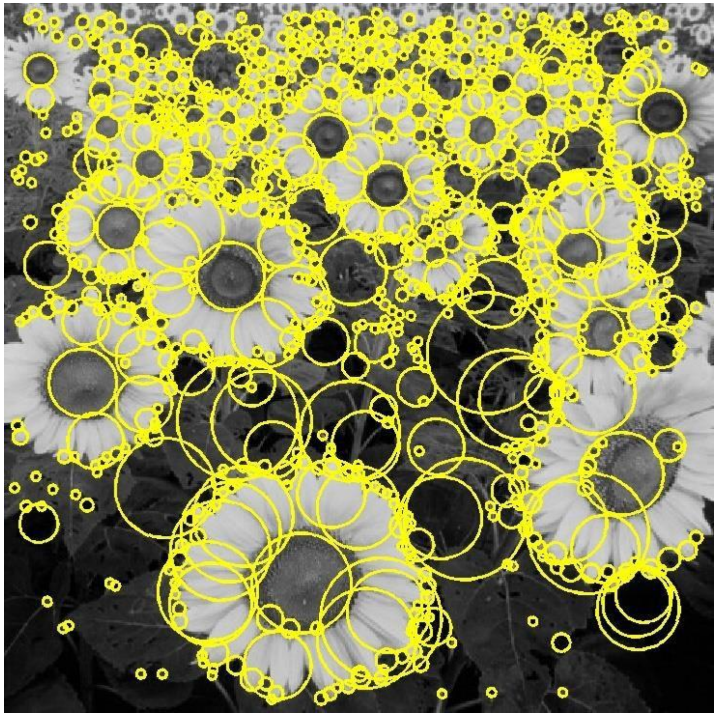
Keypoint detection with scale selection

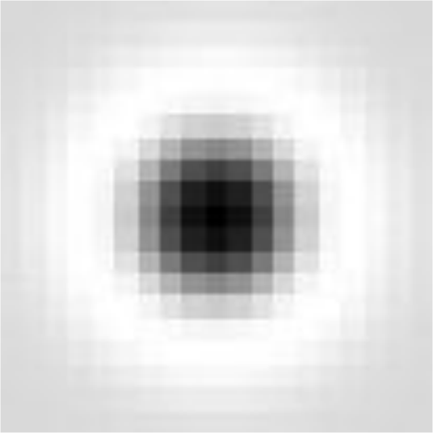
- We want to extract keypoints with characteristic scale that is *covariant* with the image transformation



Basic idea

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

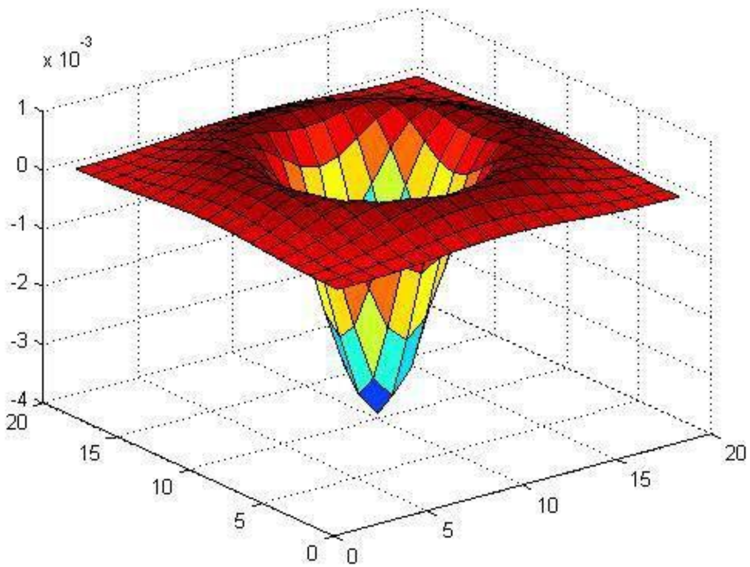


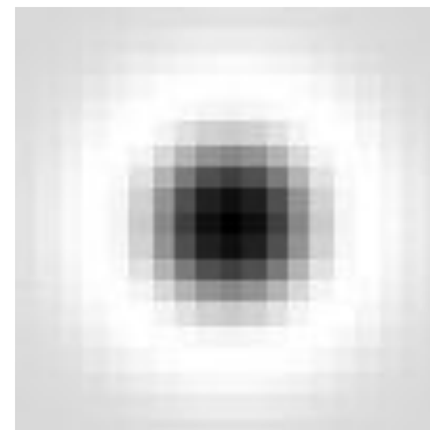


T. Lindeberg. [Feature detection with automatic scale selection.](#) *IJCV* 30(2), pp 77-116, 1998.

Blob filter

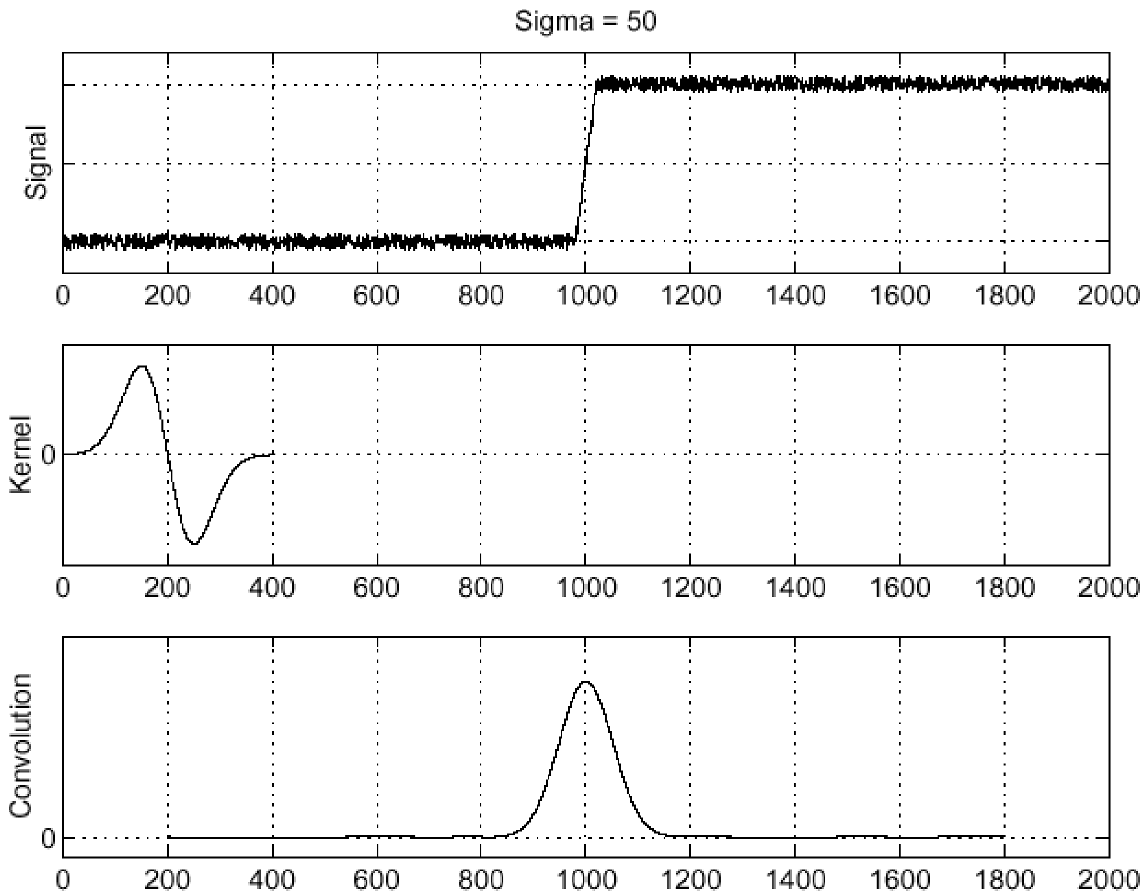
- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D





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

Recall: Edge detection



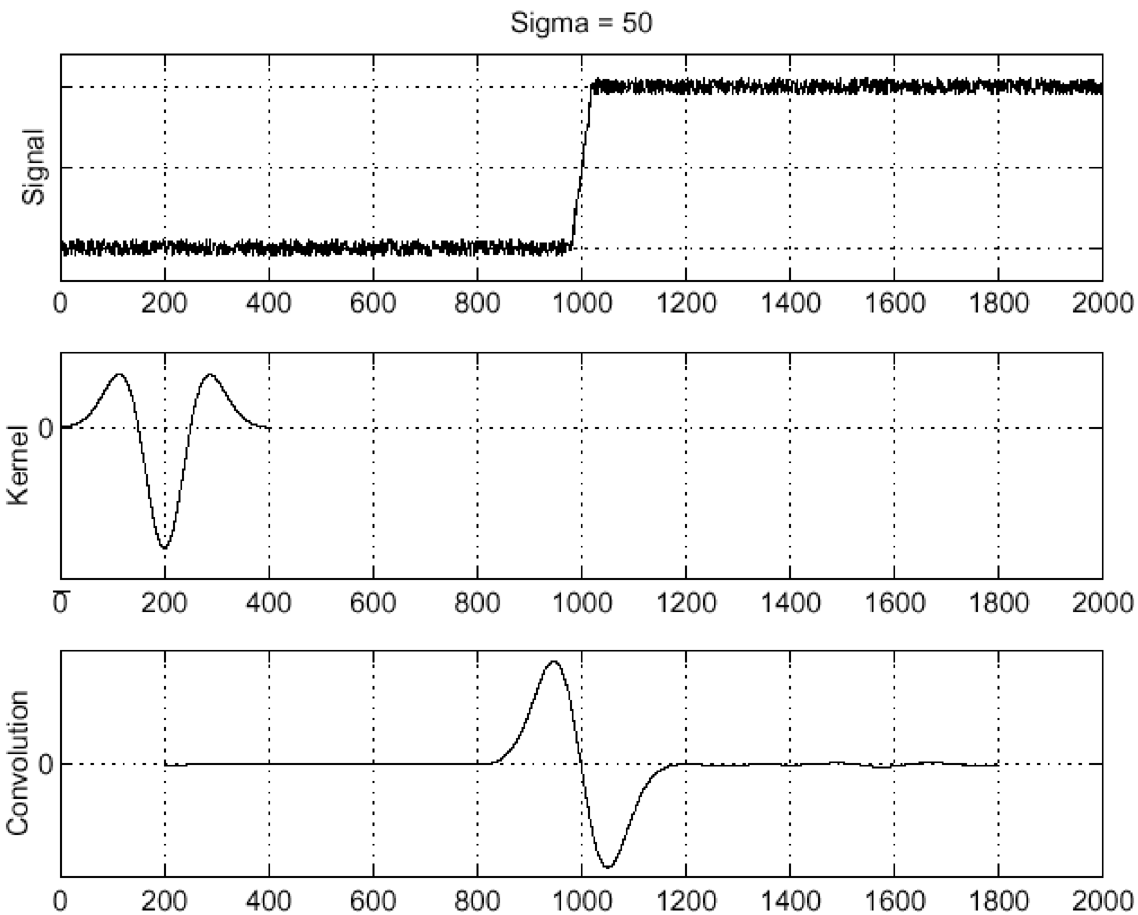
f — Edge

$\frac{d}{dx} g$ — Derivative of Gaussian

$f * \frac{d}{dx} g$ — Edge = maximum of derivative

Source: S. Seitz

Edge detection, Take 2



f

Edge

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

Second derivative
of Gaussian
(Laplacian)

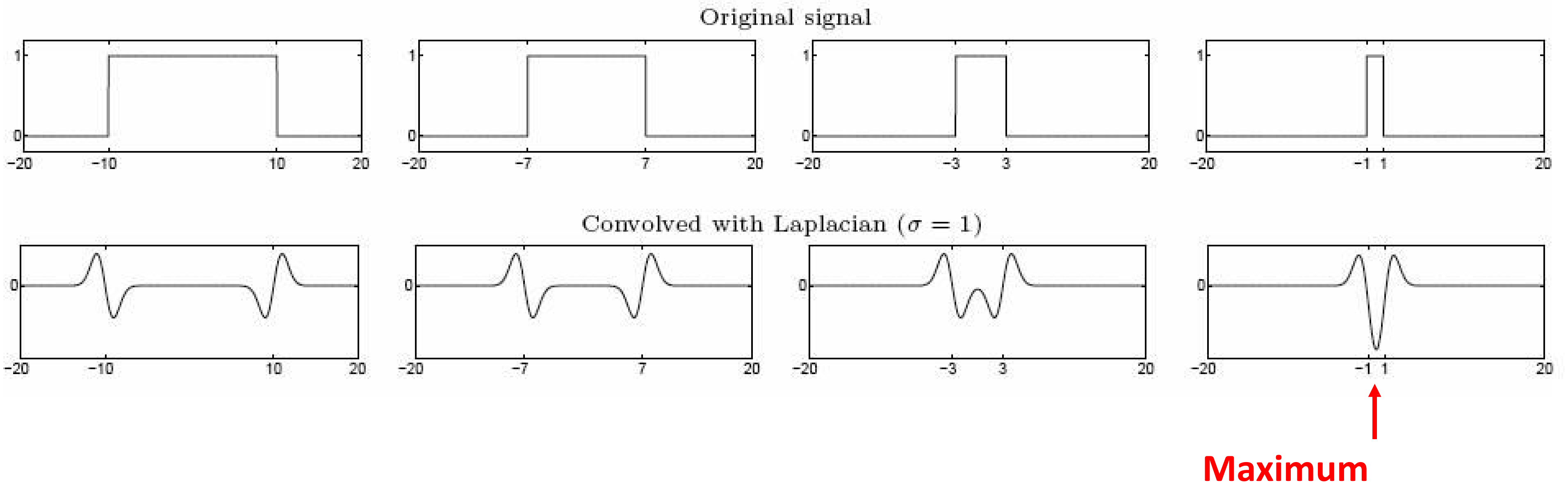
$f * \frac{d_2}{dx^2} g$

Edge = zero crossing
of second derivative

Source: S. Seitz

From edges to blobs

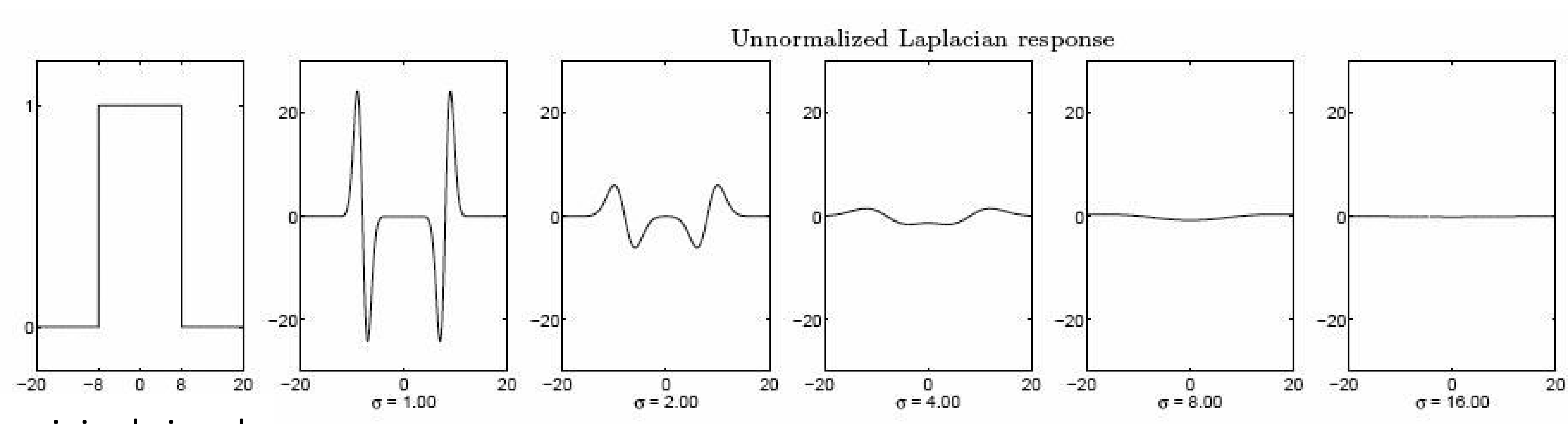
- Edge = ripple
- Blob = superposition of two ripples



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

Scale selection

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



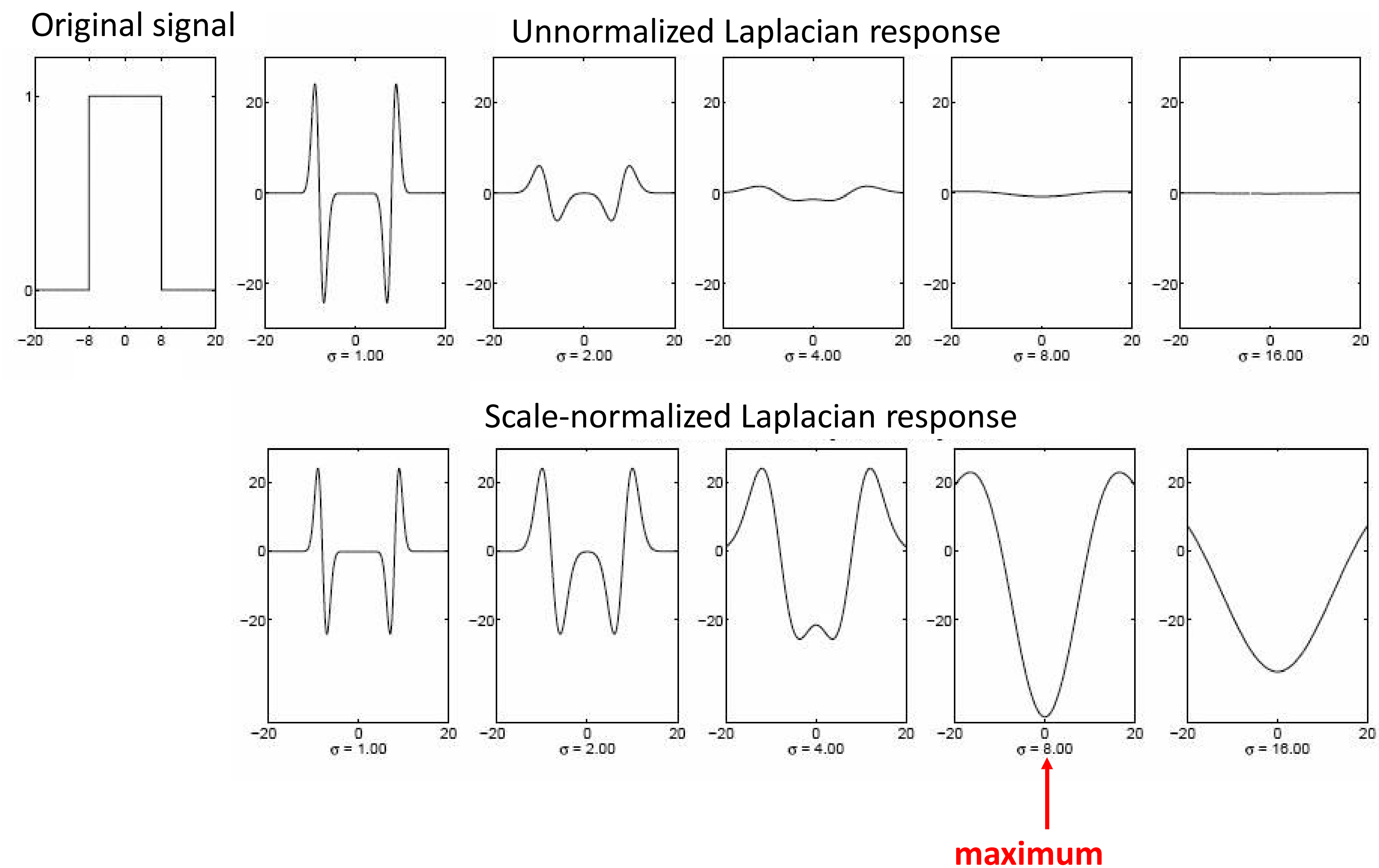
original signal
(radius=8)

increasing σ

Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases
- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

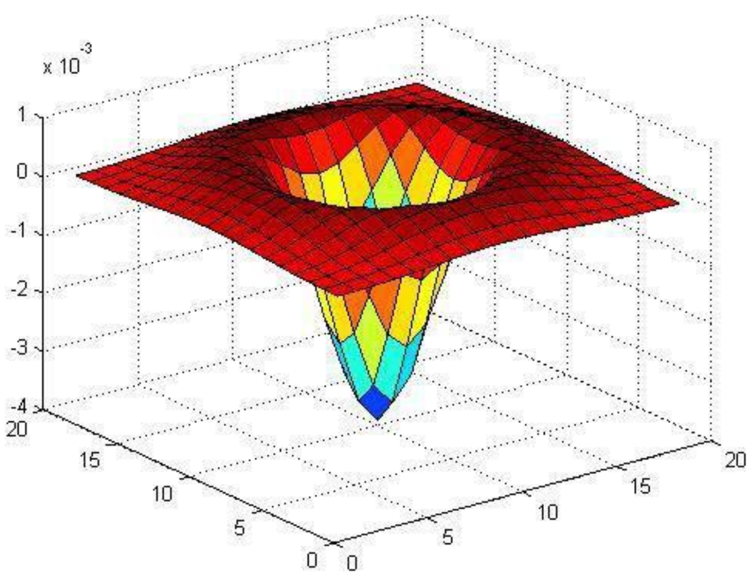
Effect of scale normalization

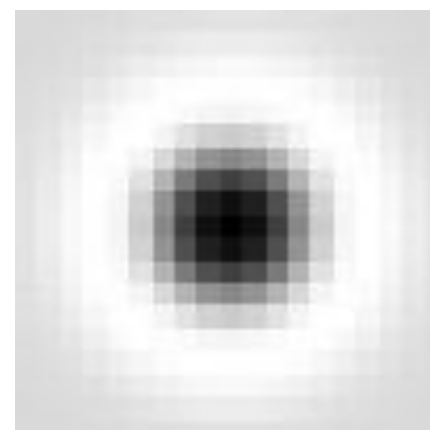


Blob detection in 2D

- *Scale-normalized* Laplacian of Gaussian:

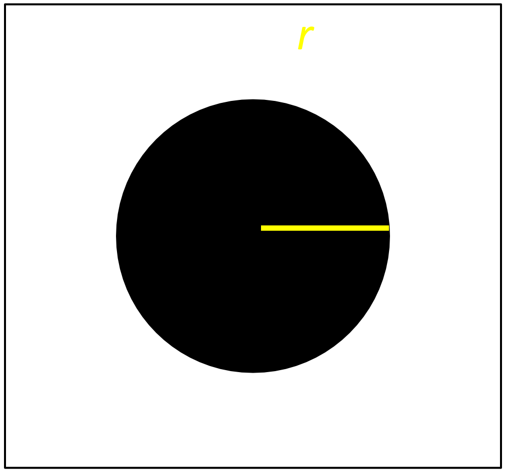
$$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$



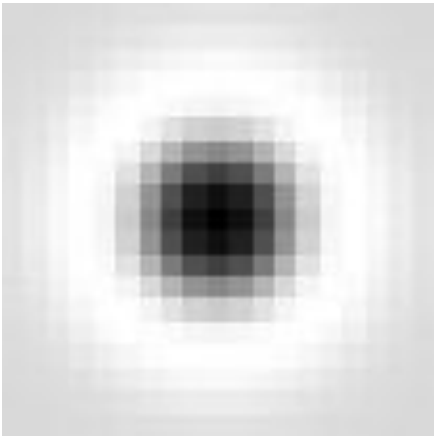


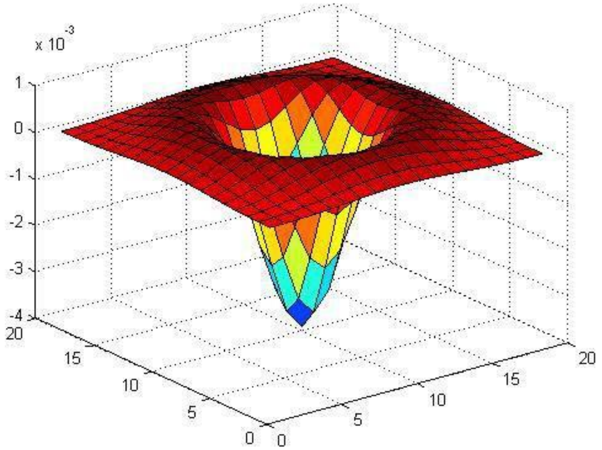
Blob detection in 2D

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



image

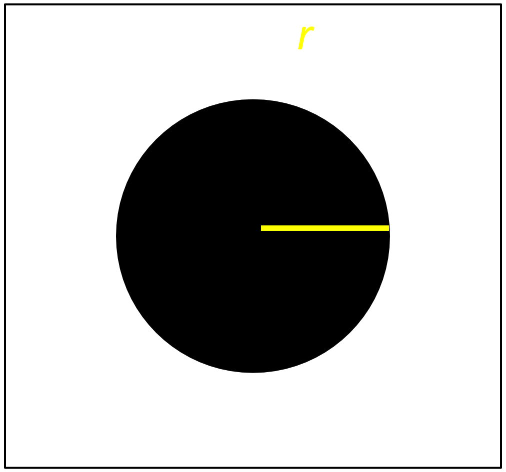




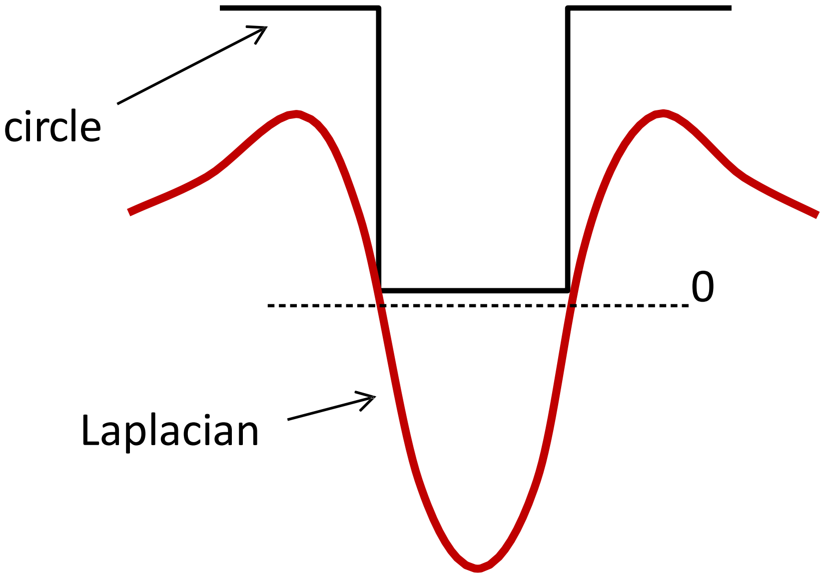
Laplacian

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



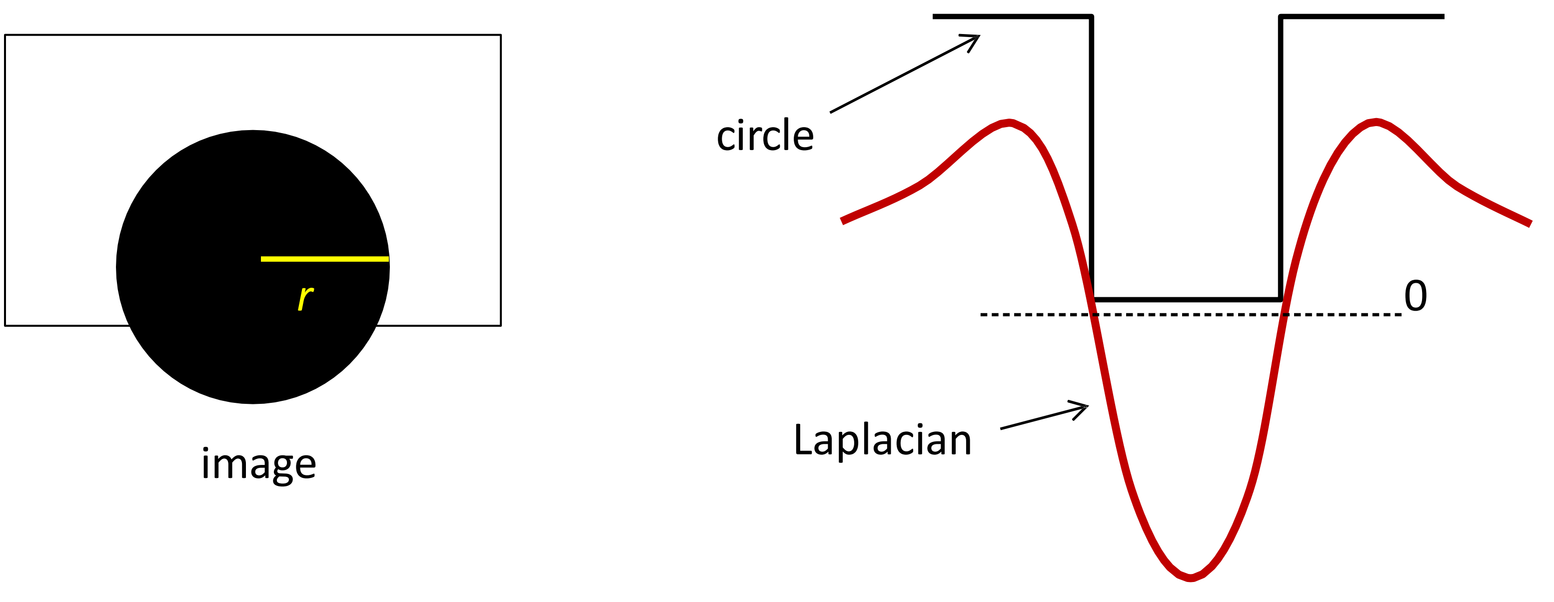
image



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):

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

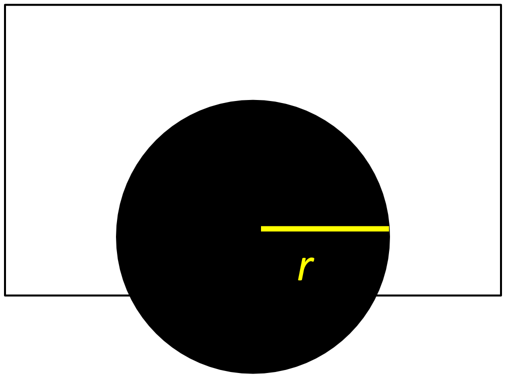


Blob detection in 2D

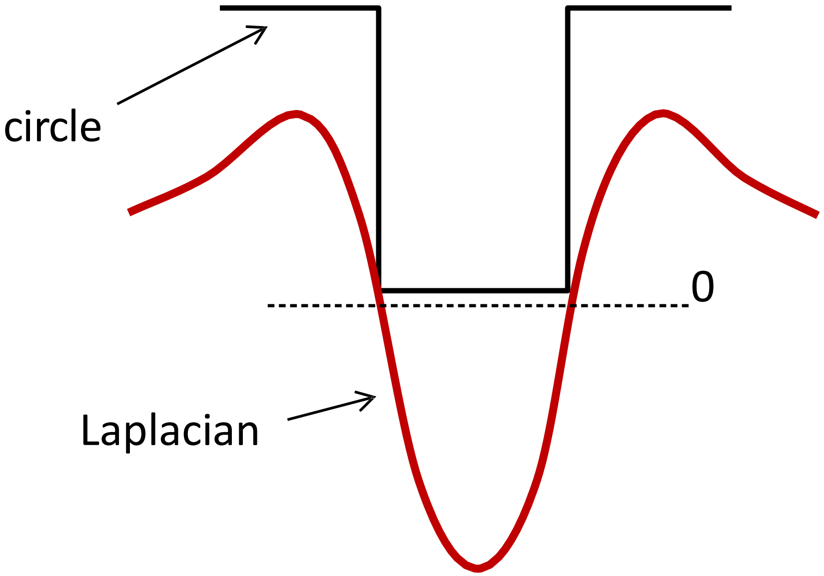
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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



image

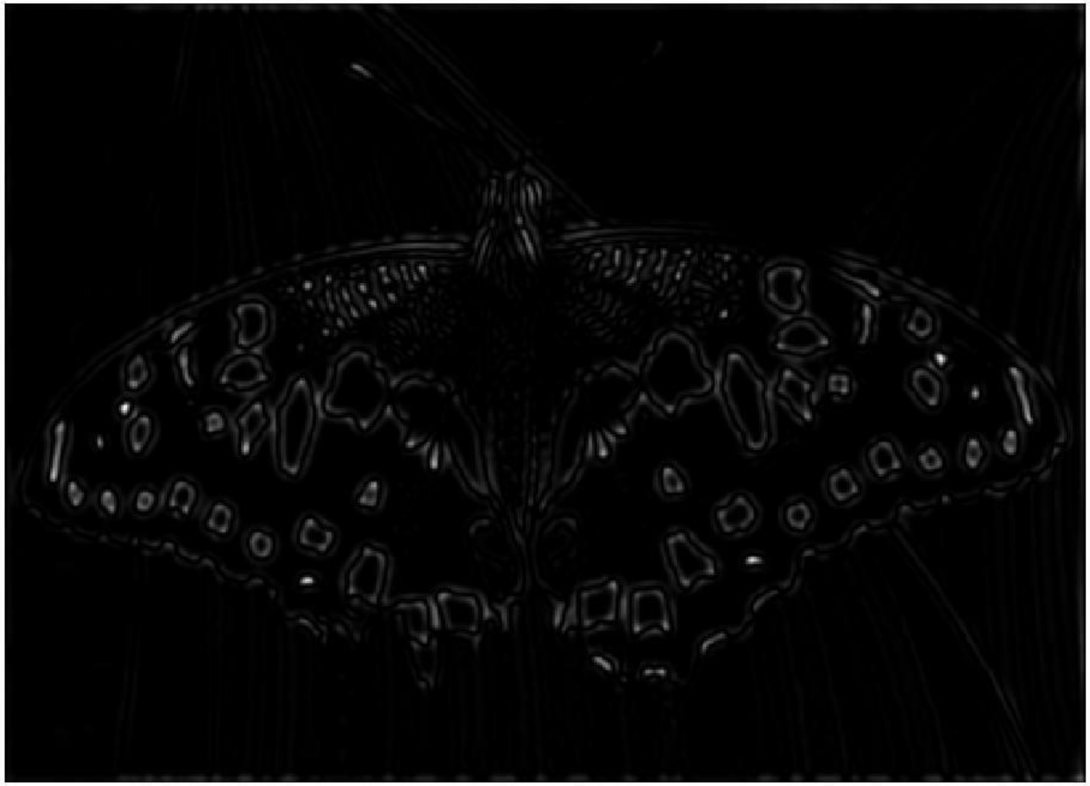


Scale-space blob detector

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

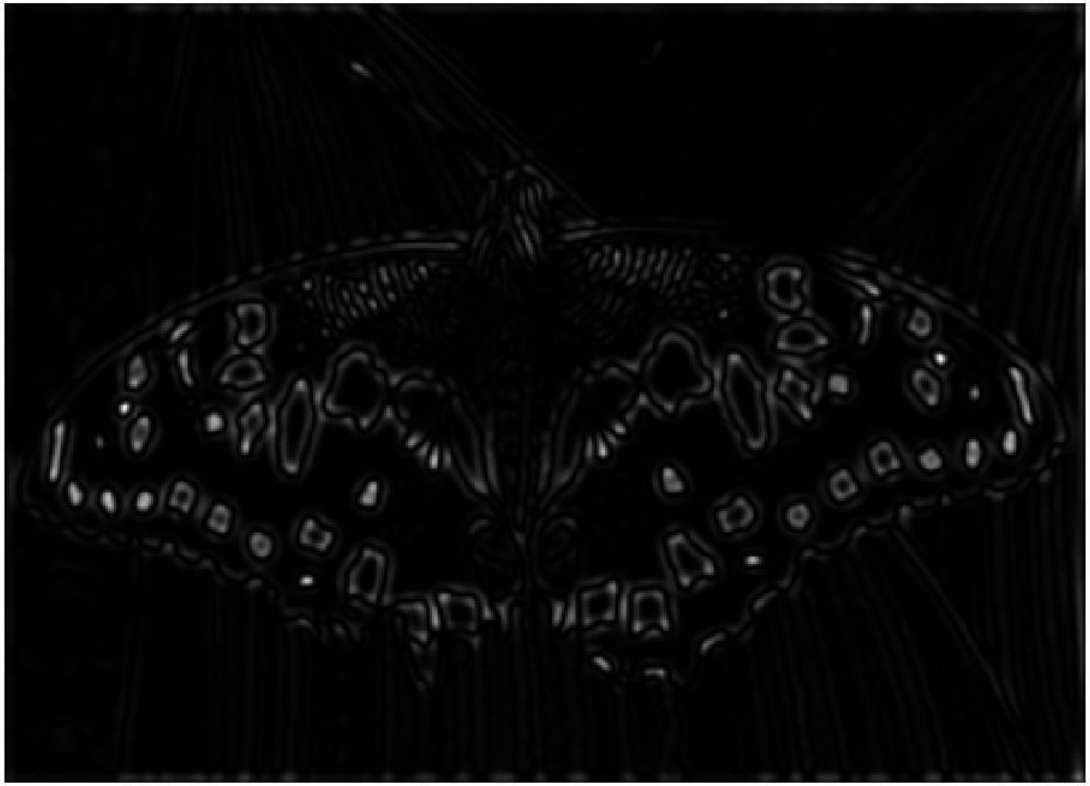
Scale-space blob detector: Example

Scale-space blob detector: Example



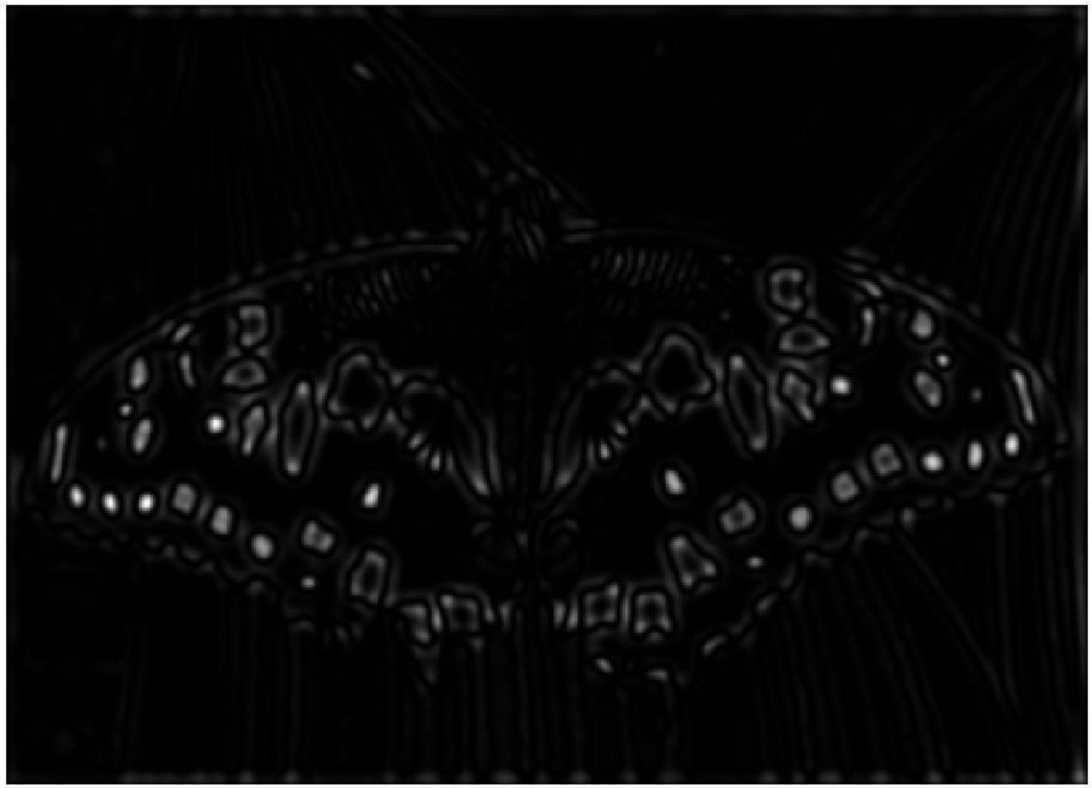
sigma = 2

Scale-space blob detector: Example



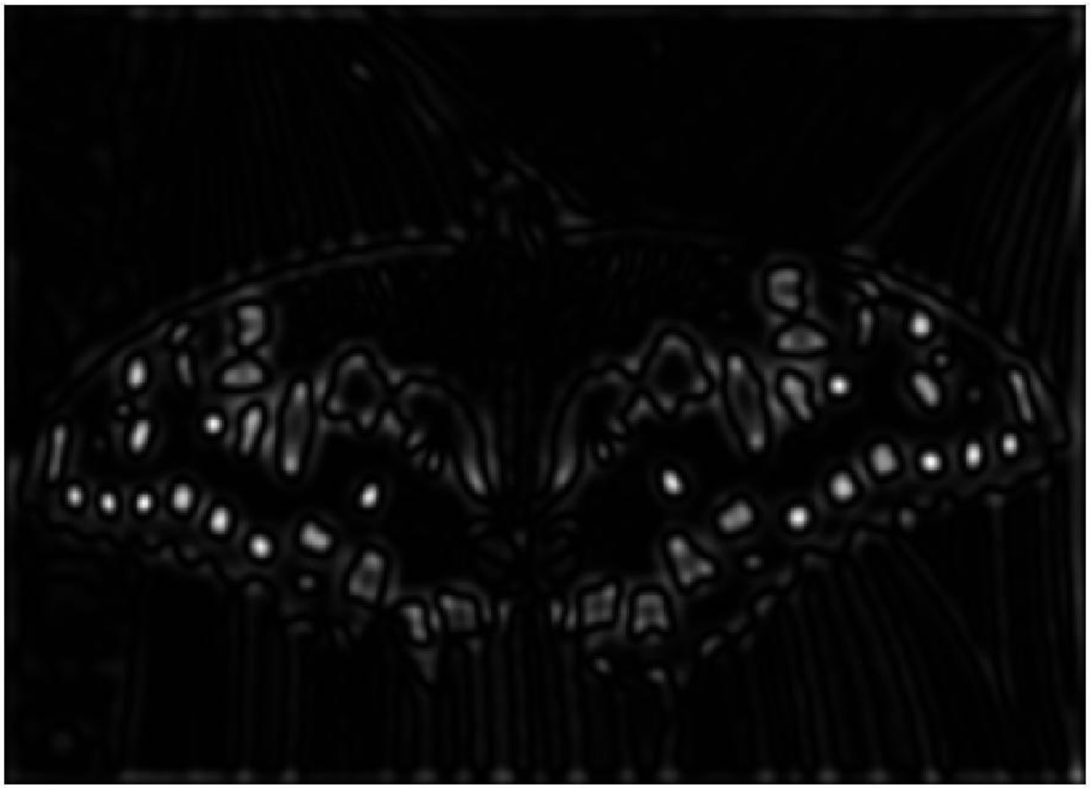
sigma = 2.5018

Scale-space blob detector: Example



sigma = 3.1296

Scale-space blob detector: Example

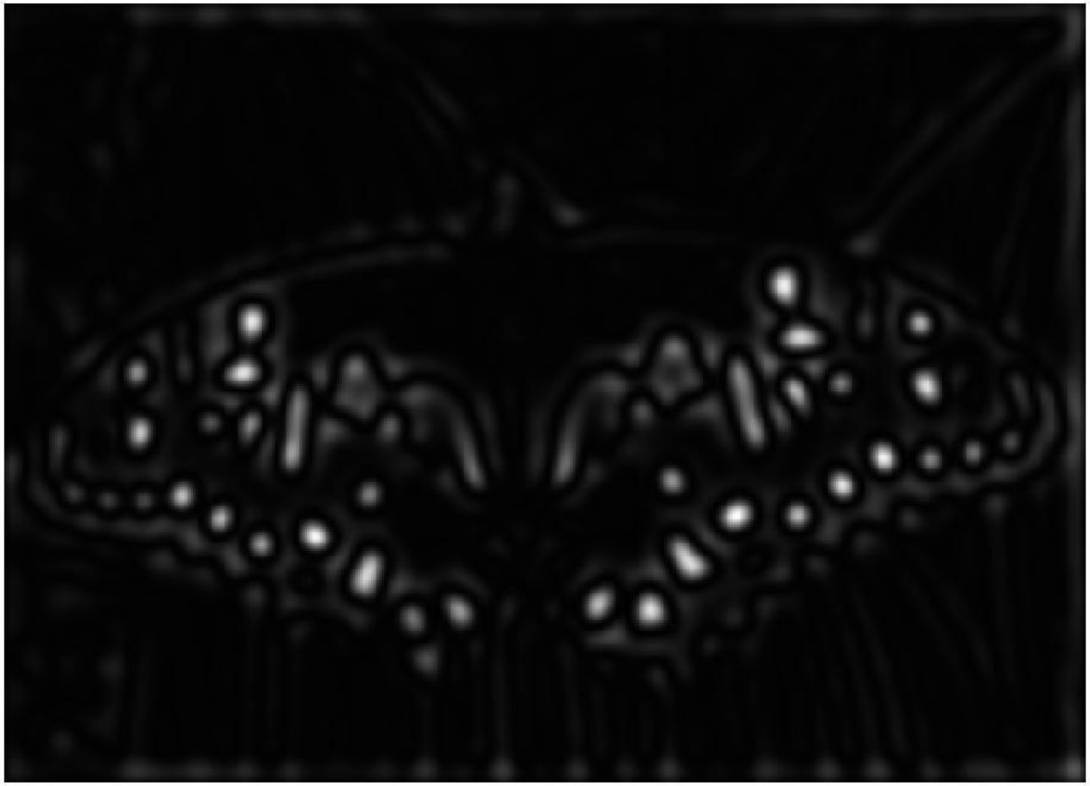


sigma = 3.9149

Scale-space blob detector: Example

sigma = 4.8972

Scale-space blob detector: Example



sigma = 6.126

Scale-space blob detector: Example

sigma = 7.6631

Scale-space blob detector: Example

sigma = 9.5859

Scale-space blob detector: Example

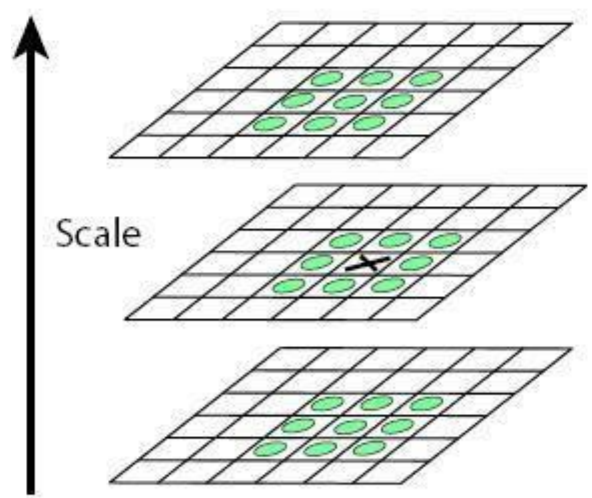
sigma = 11.9912

Scale-space blob detector Example

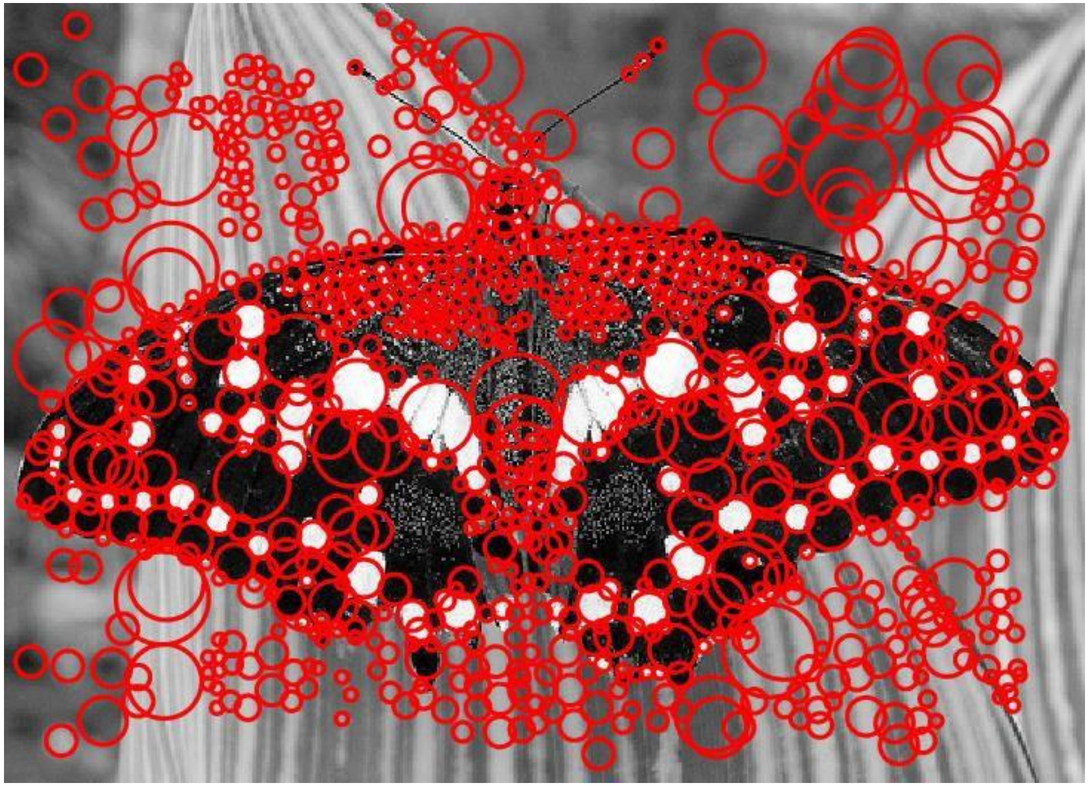
sigma = 15

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



Efficient implementation (SIFT)

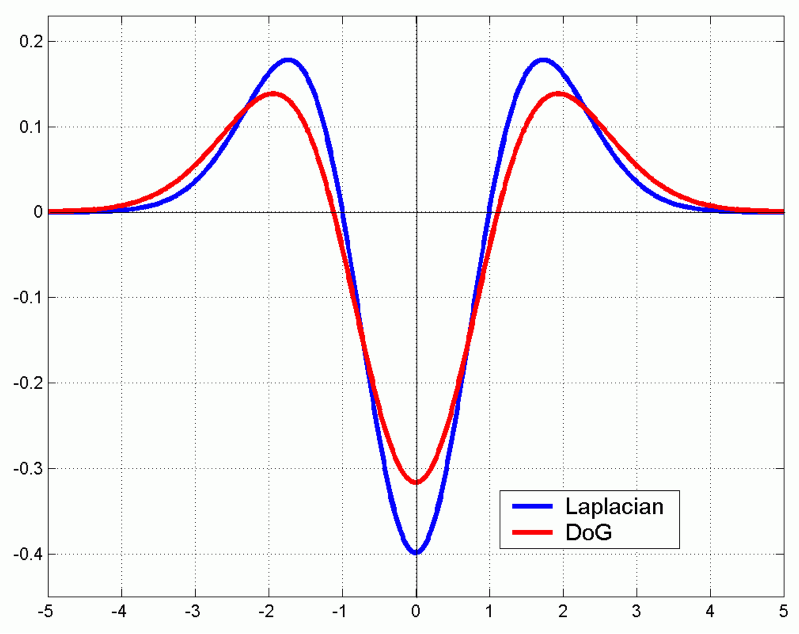
- Approximating the Laplacian with a difference of Gaussians by SIFT detector:

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

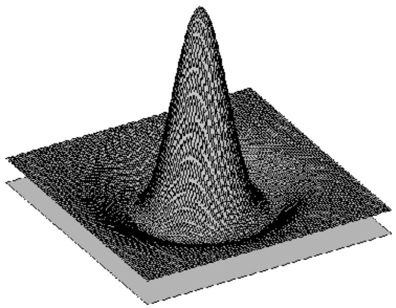
(Laplacian)

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

(Difference of Gaussians)



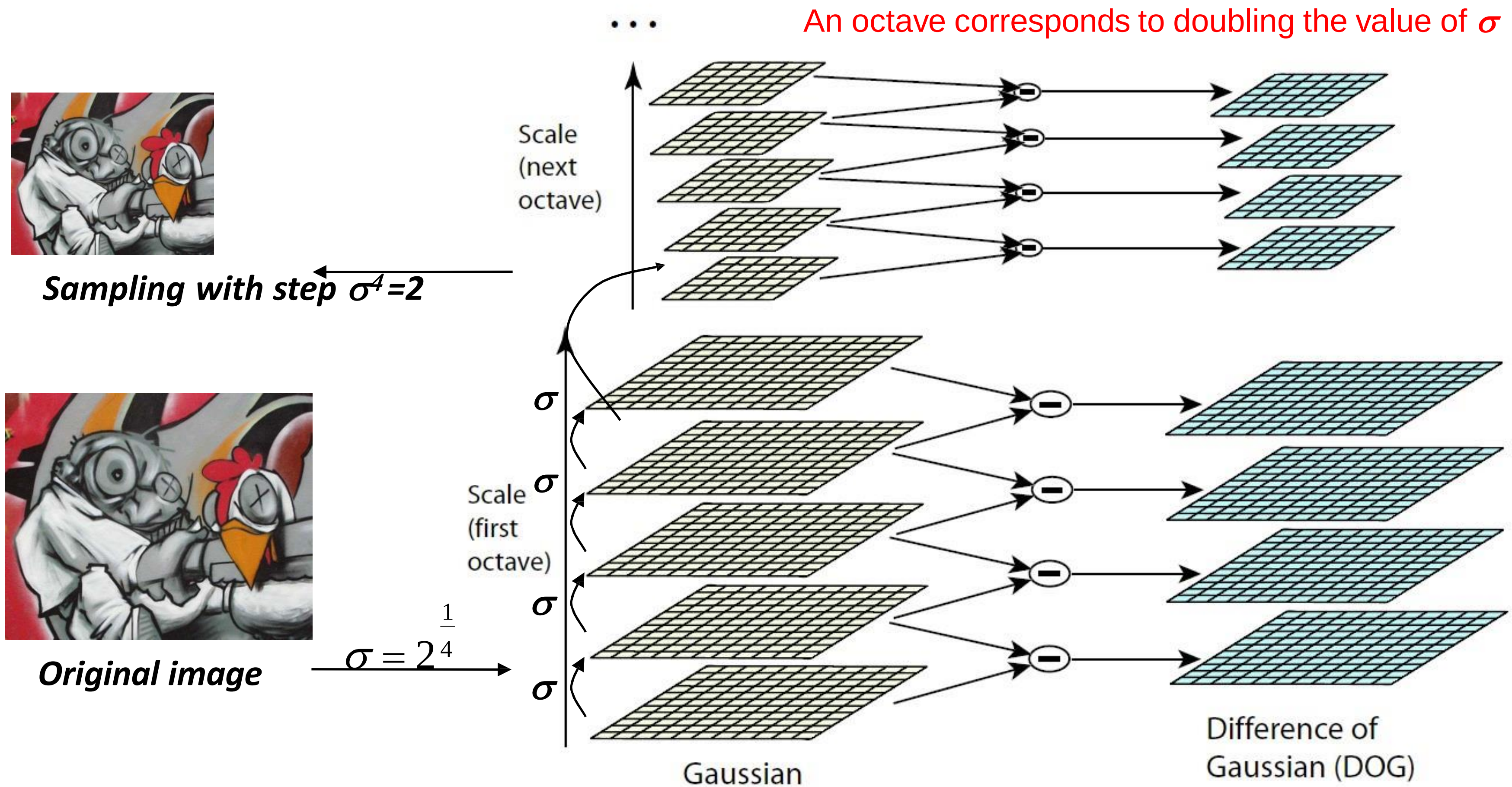
Difference-of-Gaussian (DoG)



K. Grauman, B. Leibe

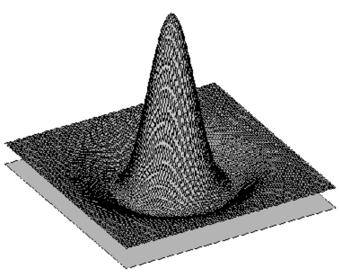
DoG – Efficient Computation

- Computation in Gaussian scale pyramid

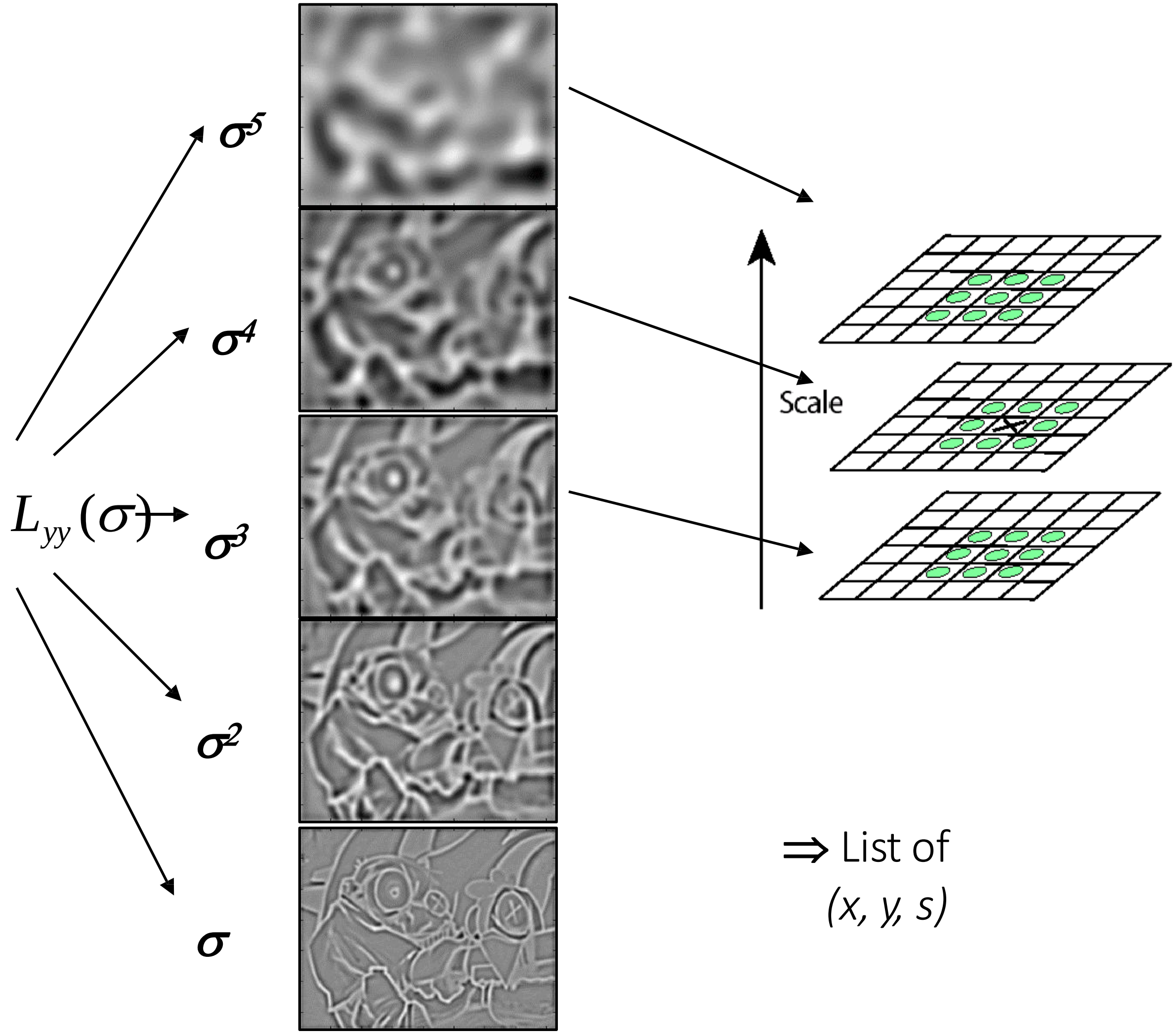


K. Grauman, B. Leibe

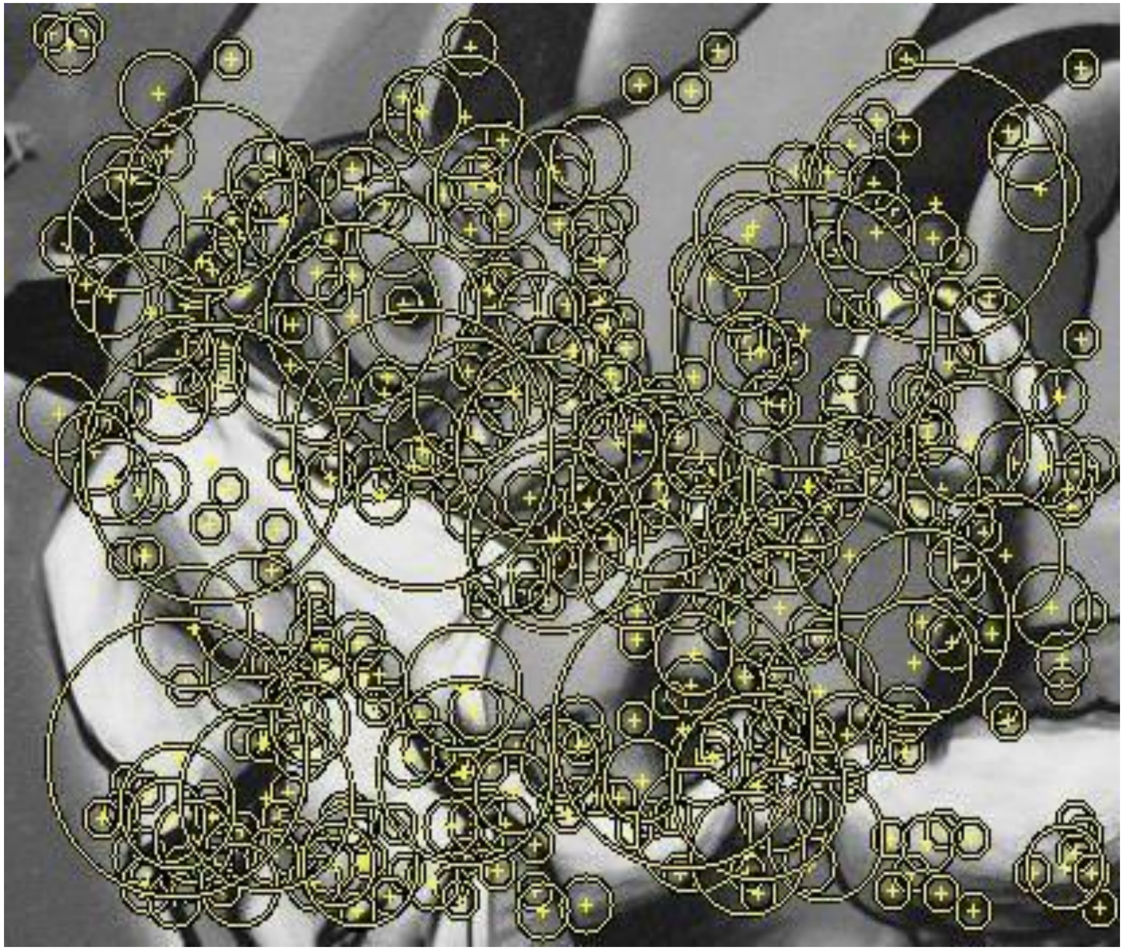
Find local maxima in position-scale space of Difference-of-Gaussian



$$L_{xx}(\sigma) + L_{yy}(\sigma)$$



Results: Difference-of-Gaussian

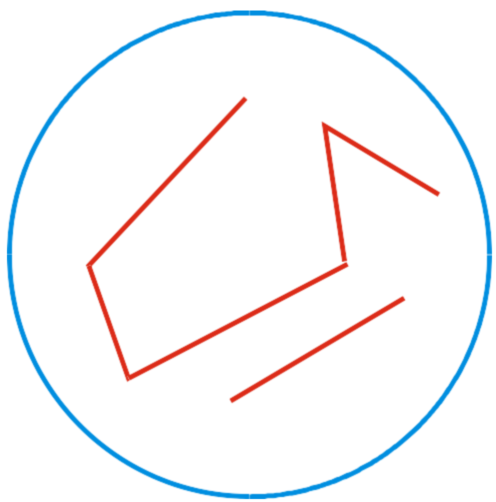


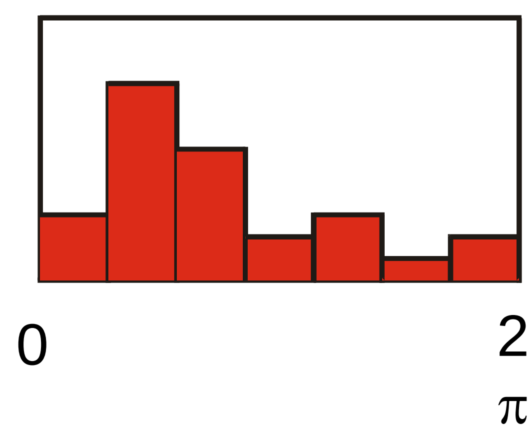
K. Grauman, B. Leibe

Orientation Normalization

[Lowe, SIFT, 1999]

- Compute orientation histogram



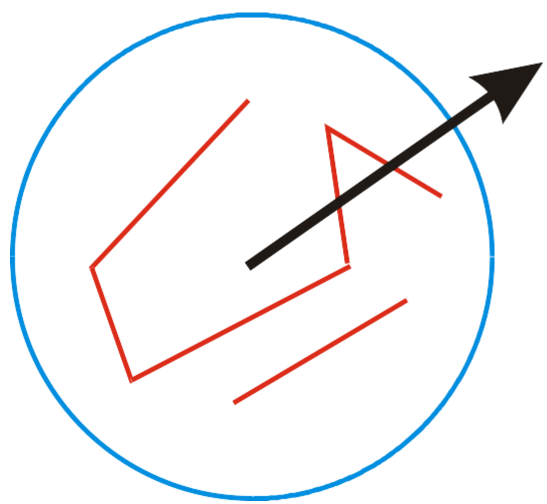


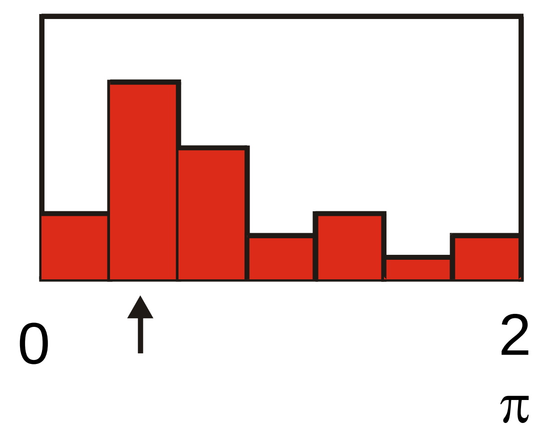
T. Tuytelaars, B. Leibe

Orientation Normalization

[Lowe, SIFT, 1999]

- Compute orientation histogram
- Select dominant orientation



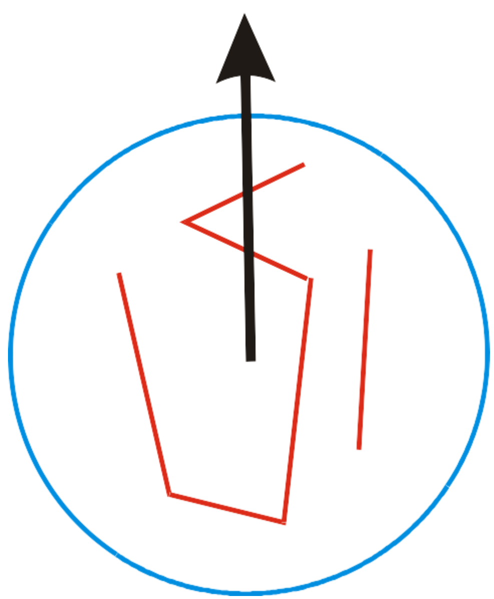


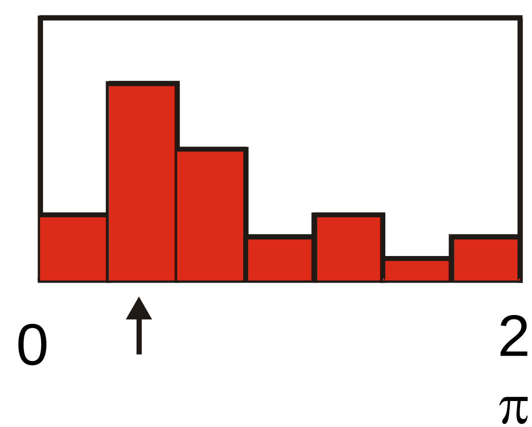
T. Tuytelaars, B. Leibe

Orientation Normalization

[Lowe, SIFT, 1999]

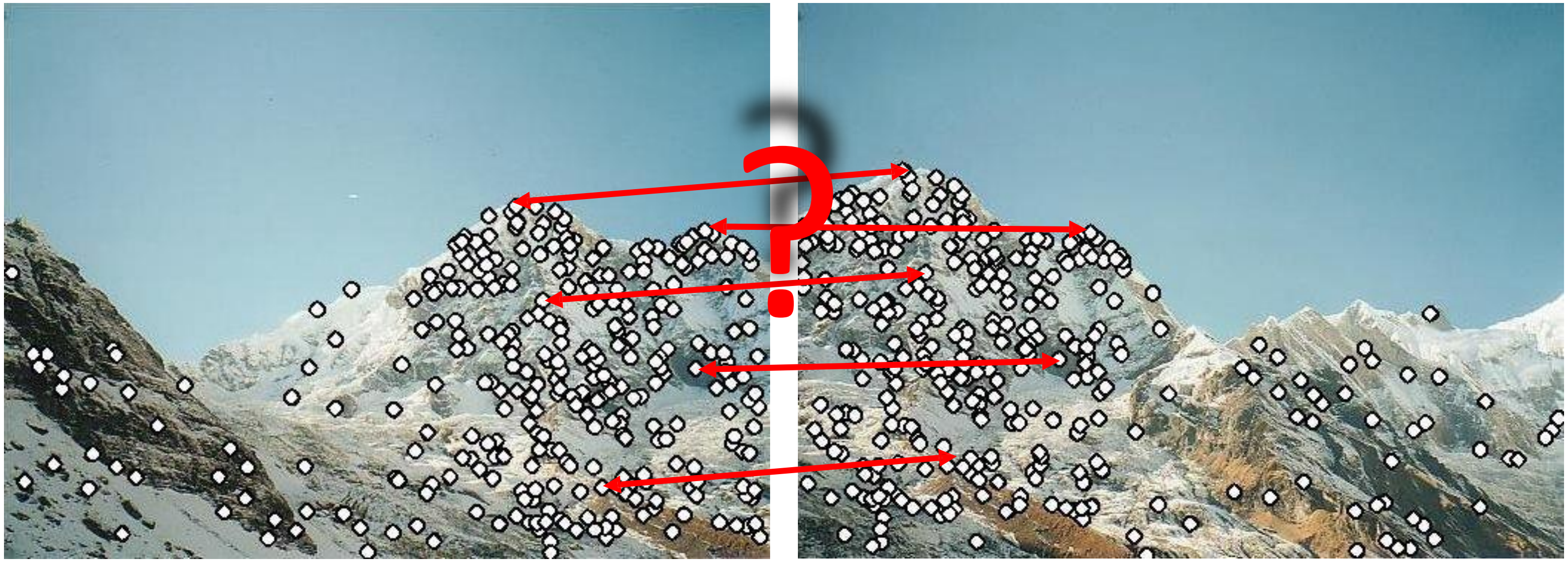
- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation





Feature descriptors

We know how to detect good points
Next question: **How to match them?**

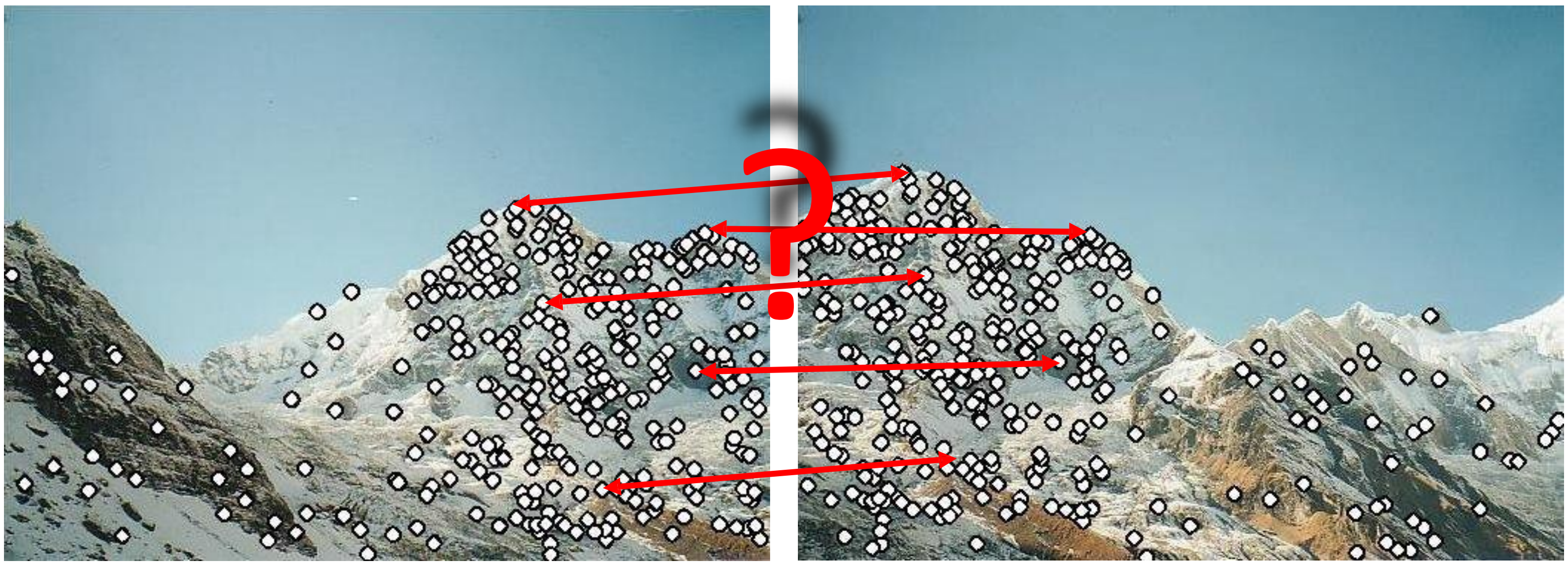


Answer: Come up with a *descriptor* for each point, find similar descriptors between the two images

Feature descriptors

We know how to detect good points Next question:

How to match them?



Lots of possibilities (this is a popular research area)

- Simple option: match square windows around the point
- State of the art approach: SIFT
    - David Lowe, UBC http://www.cs.ubc.ca/~lowe/keypoints/

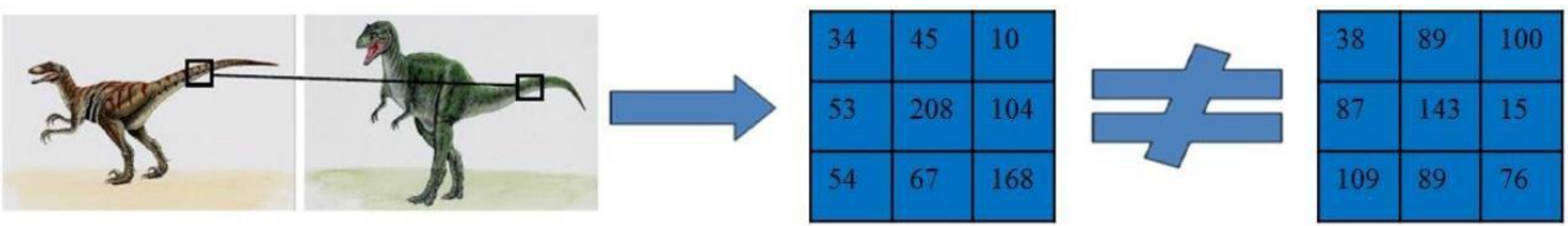
Image/Region Matching

- Automatically recognize whether two images/regions contain the similar content.

Image/Region Matching

- Automatically recognize whether two images/regions contain the similar content.

- Comparing the image pixels as they are, will not work.



Image/Region Matching

- Pixel-based distances on high-dimensional data (and images especially) can be very unintuitive.

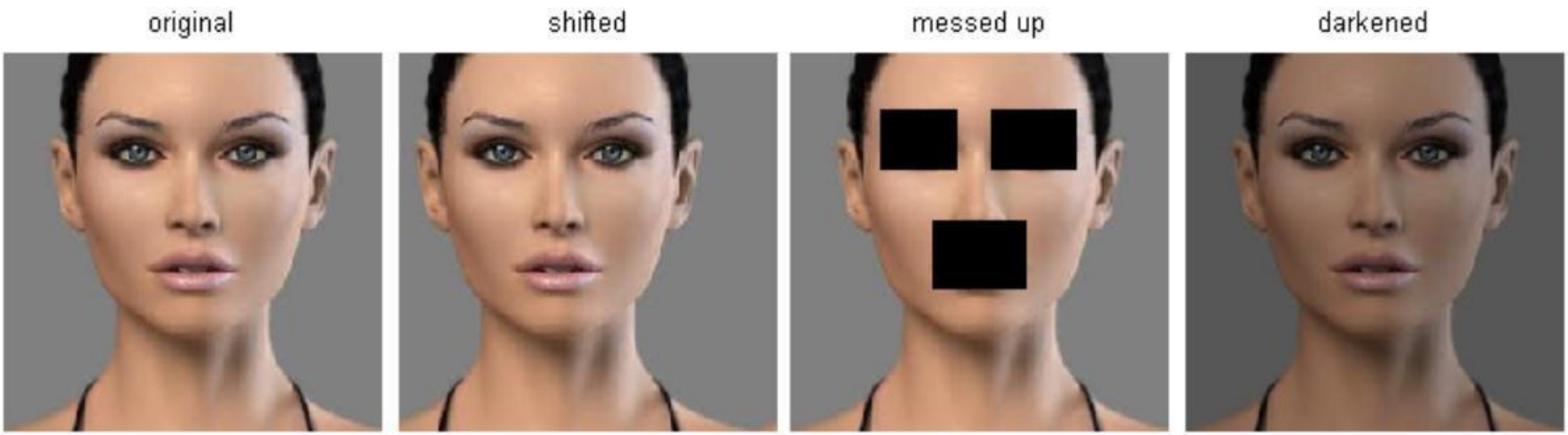


Image Source: cs231n, Oxford University

Challenges

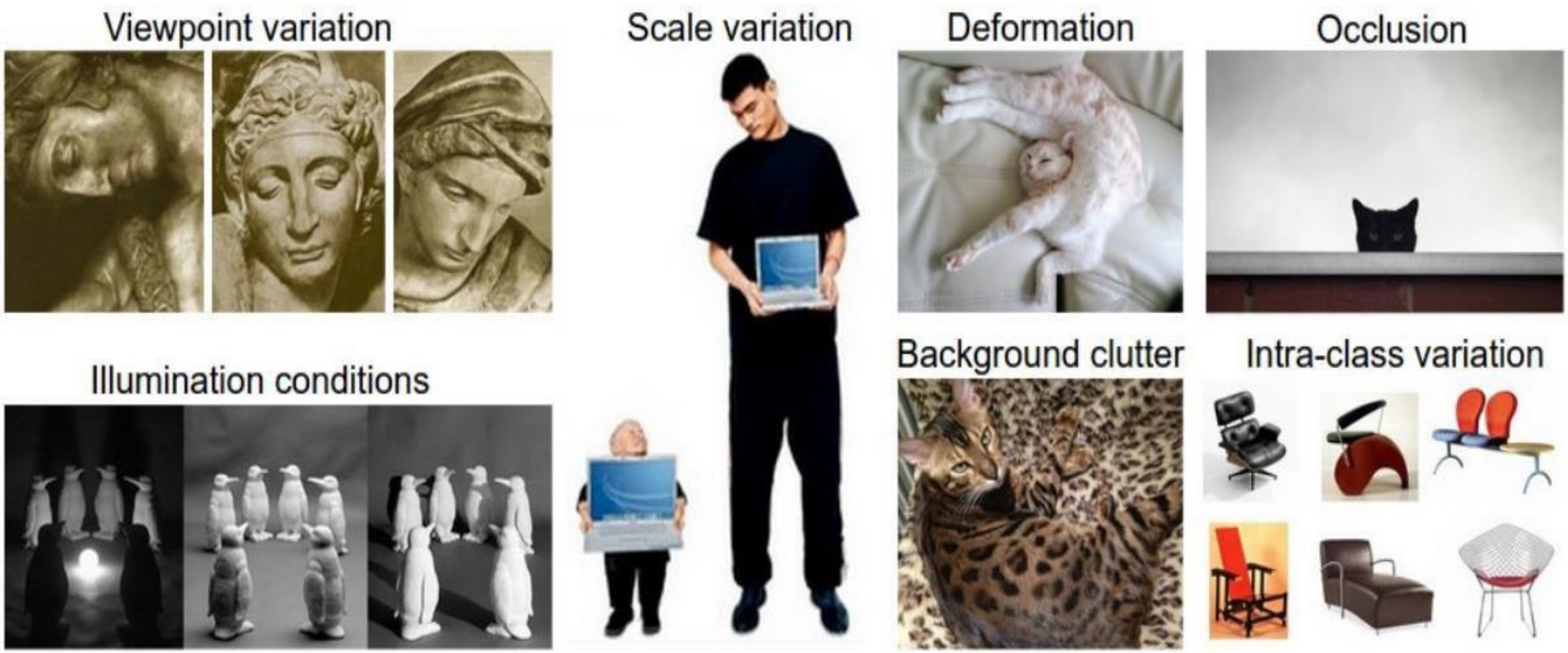
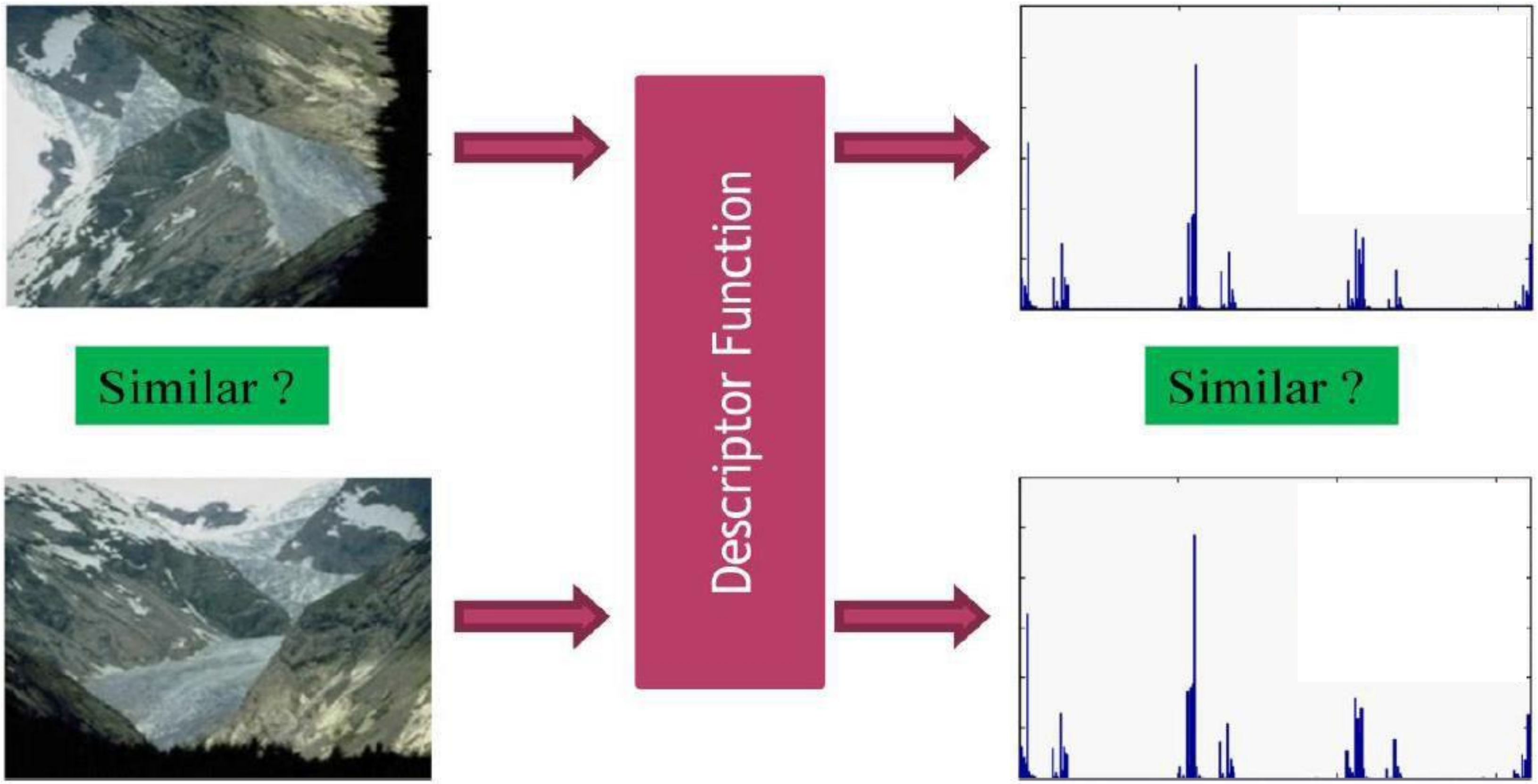


Image Source: cs231n, Oxford University

Solution

- Descriptors allow certain differences between the images.



Comparing using descriptor function
(images are taken from Corel-database and RSHD descriptor is used)

Dubey et al. Rotation and scale invariant hybrid image descriptor and retrieval. *Computers & Electrical Engineering*, 2015.

Applications

- Image Matching
- Image Retrieval
- Biomedical Image Analysis
- Texture Classification
- Image Correspondence
- Face Analysis
- Biometrics
- Building Panorama
- And many more…

Image descriptor

- Descriptions of the visual features
- Described by appearance based characteristics such as color, shape, etc.

Image descriptor

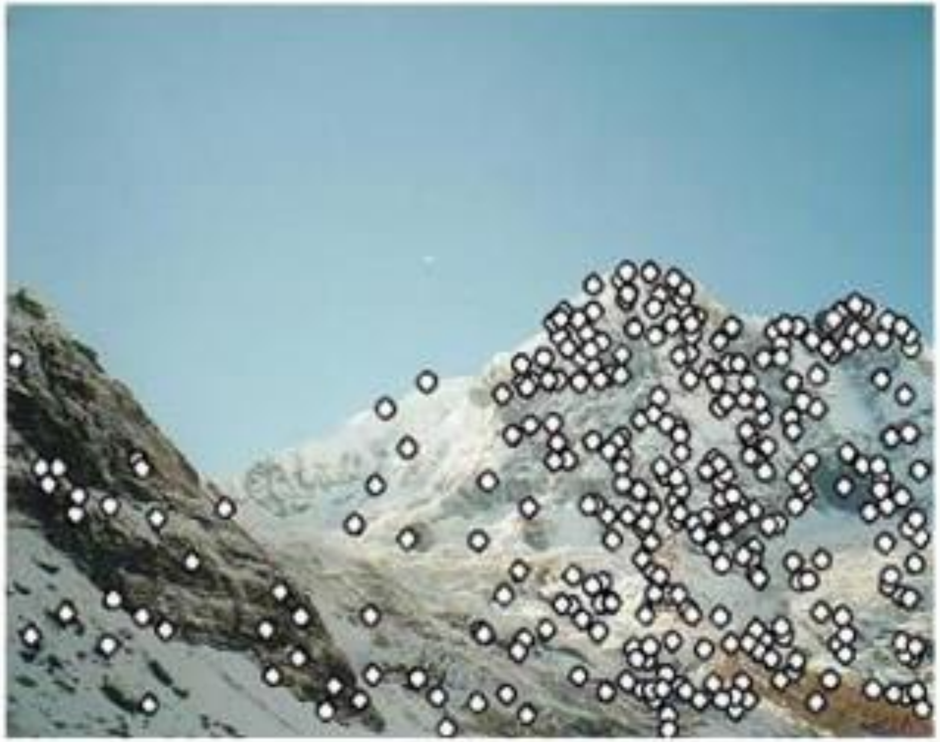
- Descriptions of the visual features
- Described by appearance based characteristics such as color, shape, etc.

A descriptor must be

- Distinctive
- Robust
- Compact
- Low Dimensional

Where to compute the descriptors?

- Over interest regions.
- Interest region may be
  - Key-Points or Global based.



Local Descriptors

- Most available descriptors focus on -
  - Edge/gradient information
  - Capture texture information
  - Exploit local relationship
  - Color also play a vital role
  - Shape features
  - Feature fusion

Widely Used Local Descriptors

- SIFT – Scale Invariant Feature Transform

Distinctive image features from scale-invariant keypoints

DG Lowe - International journal of computer vision, 2004 - Springer

… the assigned orientation, **scale**, and loca- tion for each **feature**, thereby providing **invariance** to these … **Invariant** Fea- ture **Transform** (SIFT), as it **transforms** image data into **scale-invariant** coordinates relative … that densely cover the image over the full range of **scales** and locations …

Cited by 50252 Related articles All 179 versions

- LBP – Local Binary Pattern

Multiresolution gray-scale and rotation invariant texture classification with local binary patterns

T Ojala, M Pietikainen… - … Transactions on **pattern** …, 2002 - ieeexplore.ieee.org

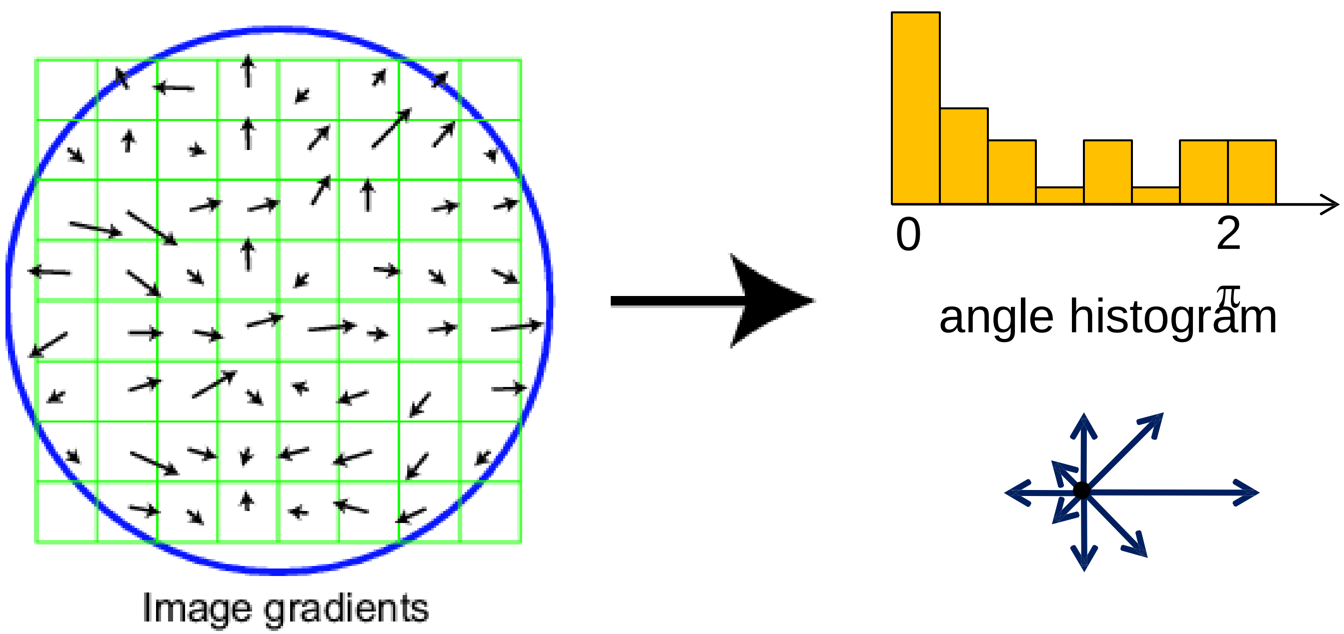
Presents a theoretically very simple, yet efficient, multiresolution approach to gray-scale and rotation invariant texture classification based on **local binary patterns** and nonparametric discrimination of sample and prototype distributions. The method is based on recognizing …

Cited by 12251 Related articles All 16 versions

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

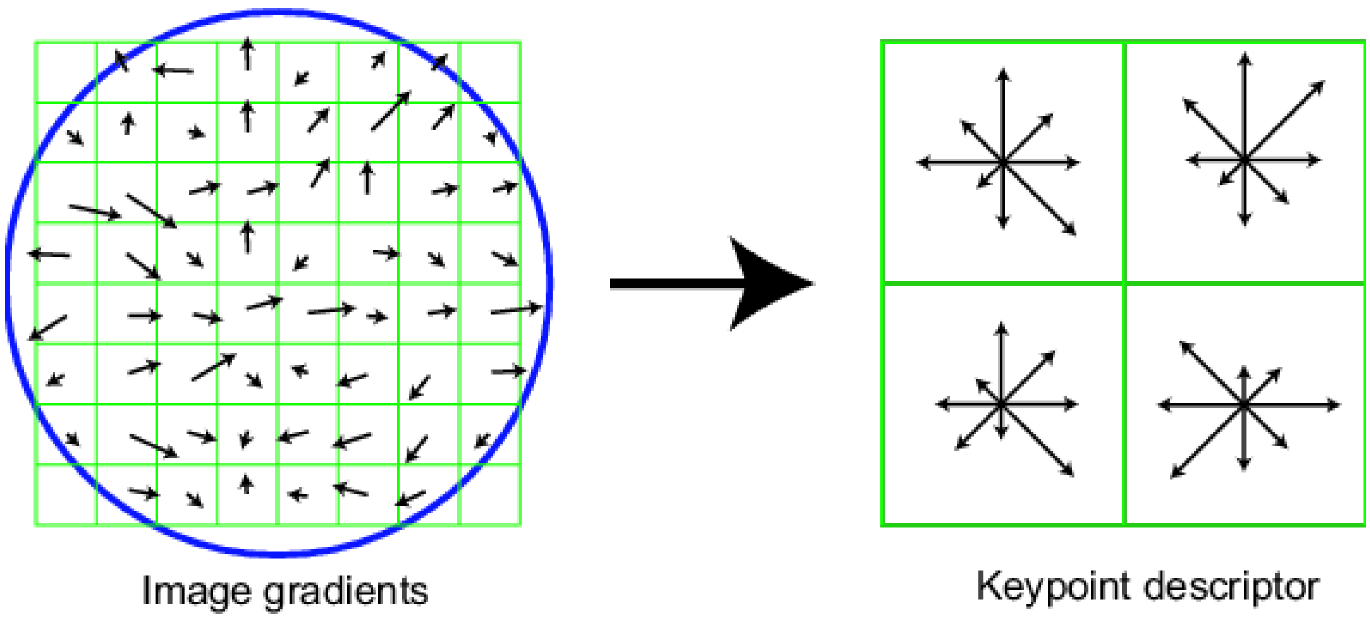


Adapted from slide by David Lowe

SIFT descriptor

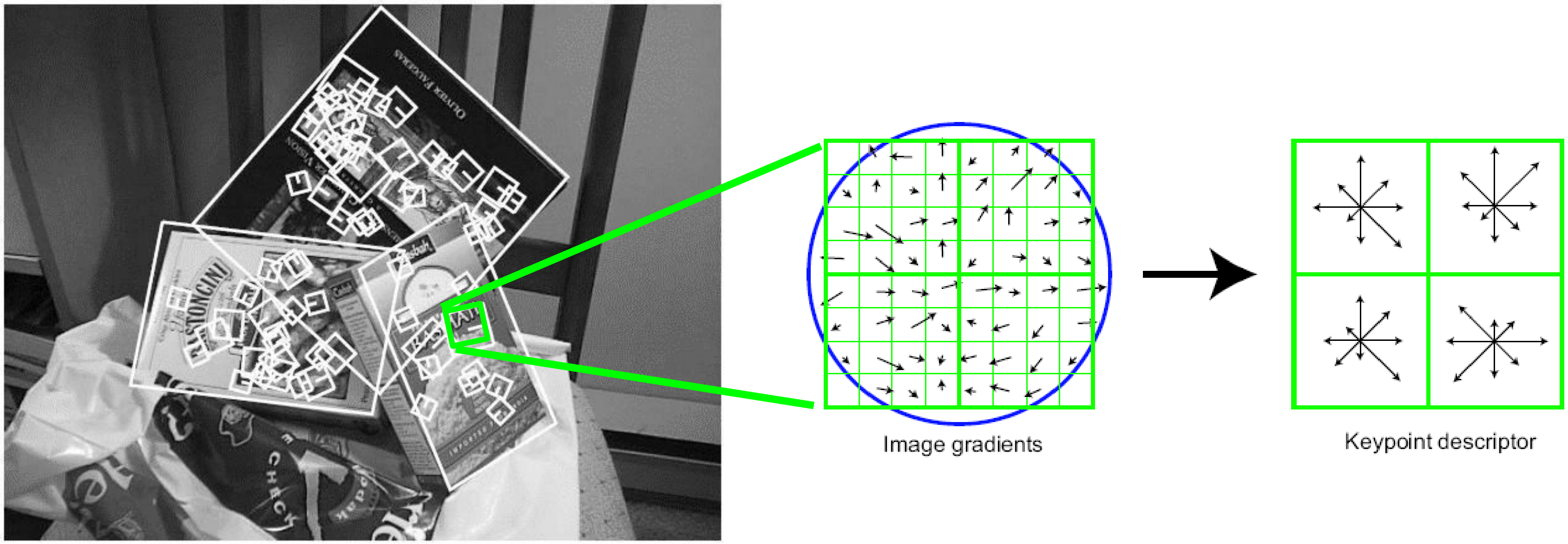
Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

Local Descriptors: SIFT Descriptor



Histogram of oriented gradients

•Captures important texture information

•Robust to small translations / affine deformations

[Lowe, ICCV 1999]

K. Grauman, B. Leibe

Feature matching

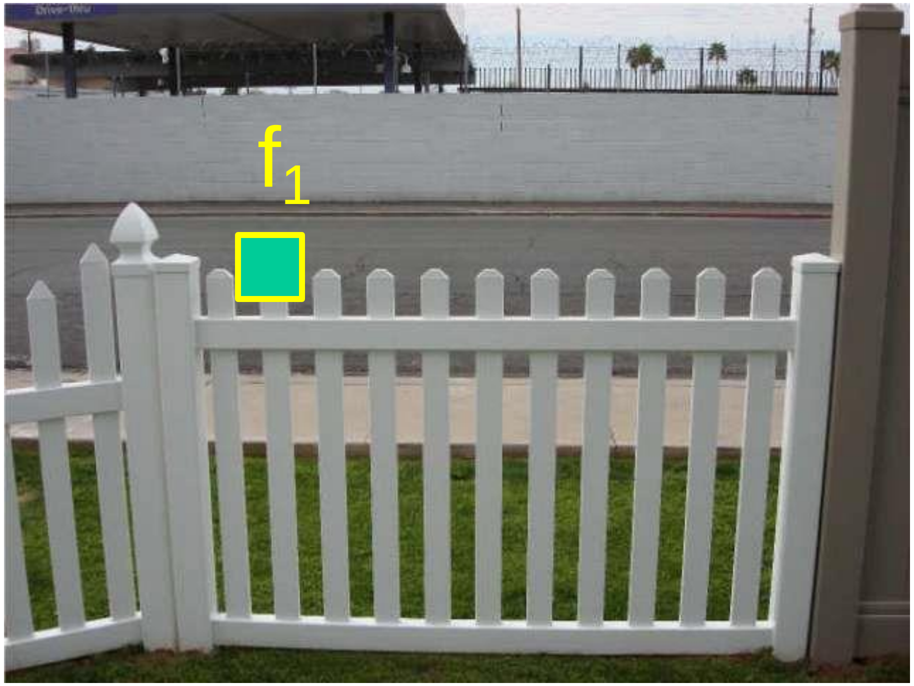
Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

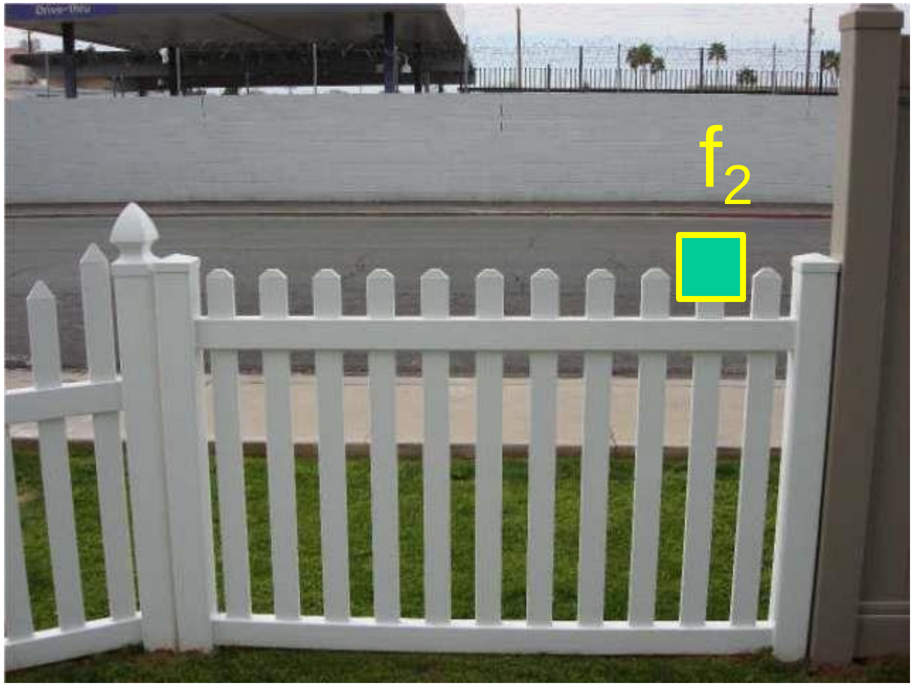
Feature distance

How to define the difference between two features f_1, f_2?

- Simple approach: L_2 distance, $||f_1 - f_2||$
- can give good scores to ambiguous (incorrect) matches



I_1

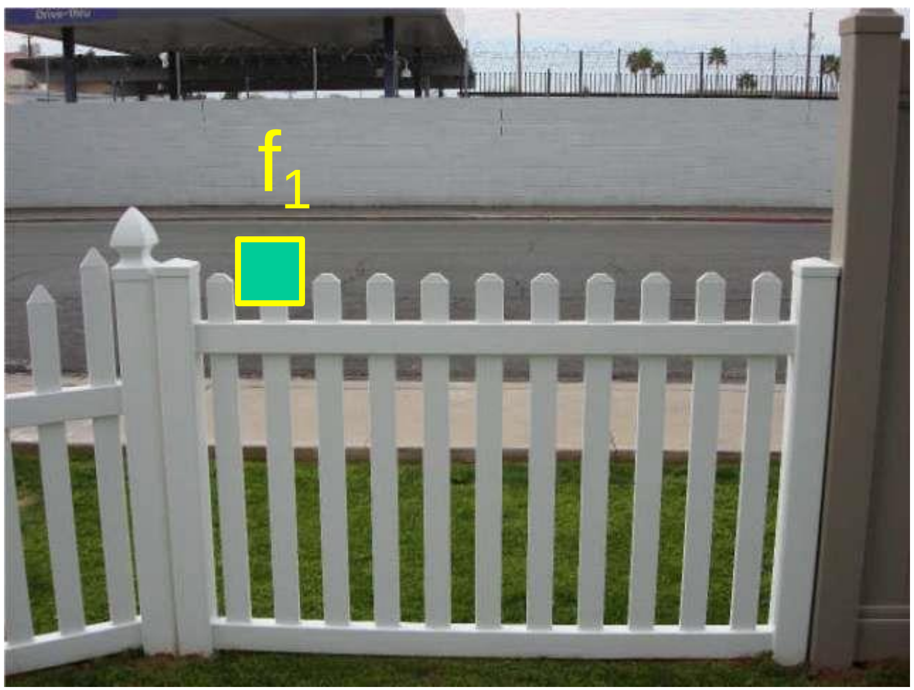


I_2

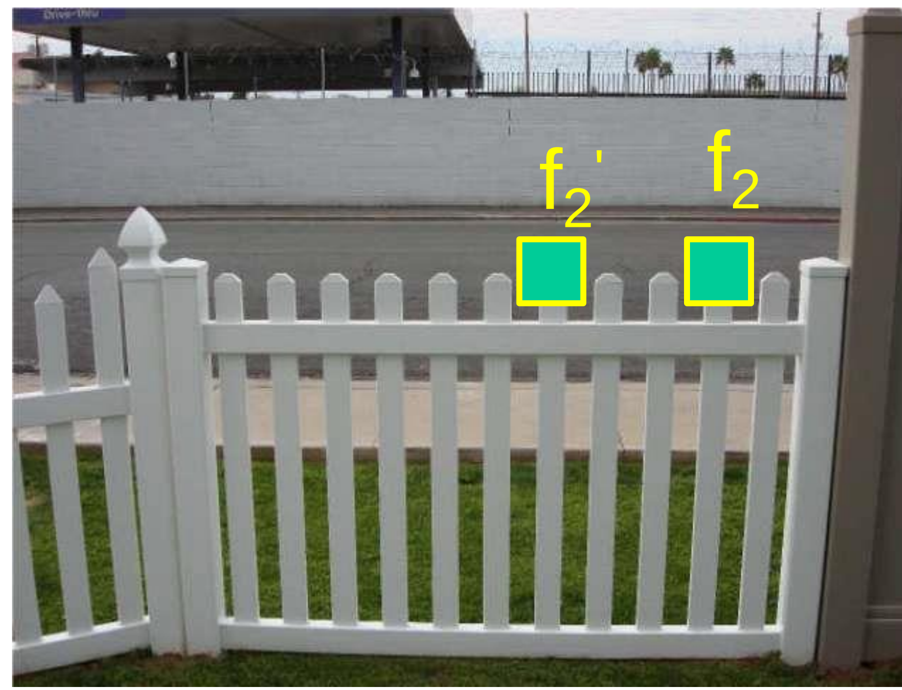
Feature distance

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $||f_1 - f_2|| / || f_1 - f_2' ||$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2nd best SSD match to f_1 in I_2
  - gives large values for ambiguous matches

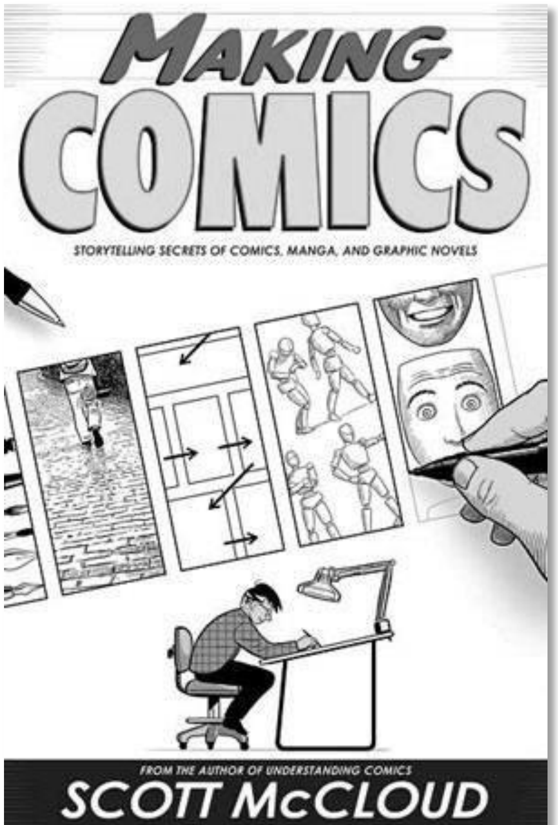


I_1

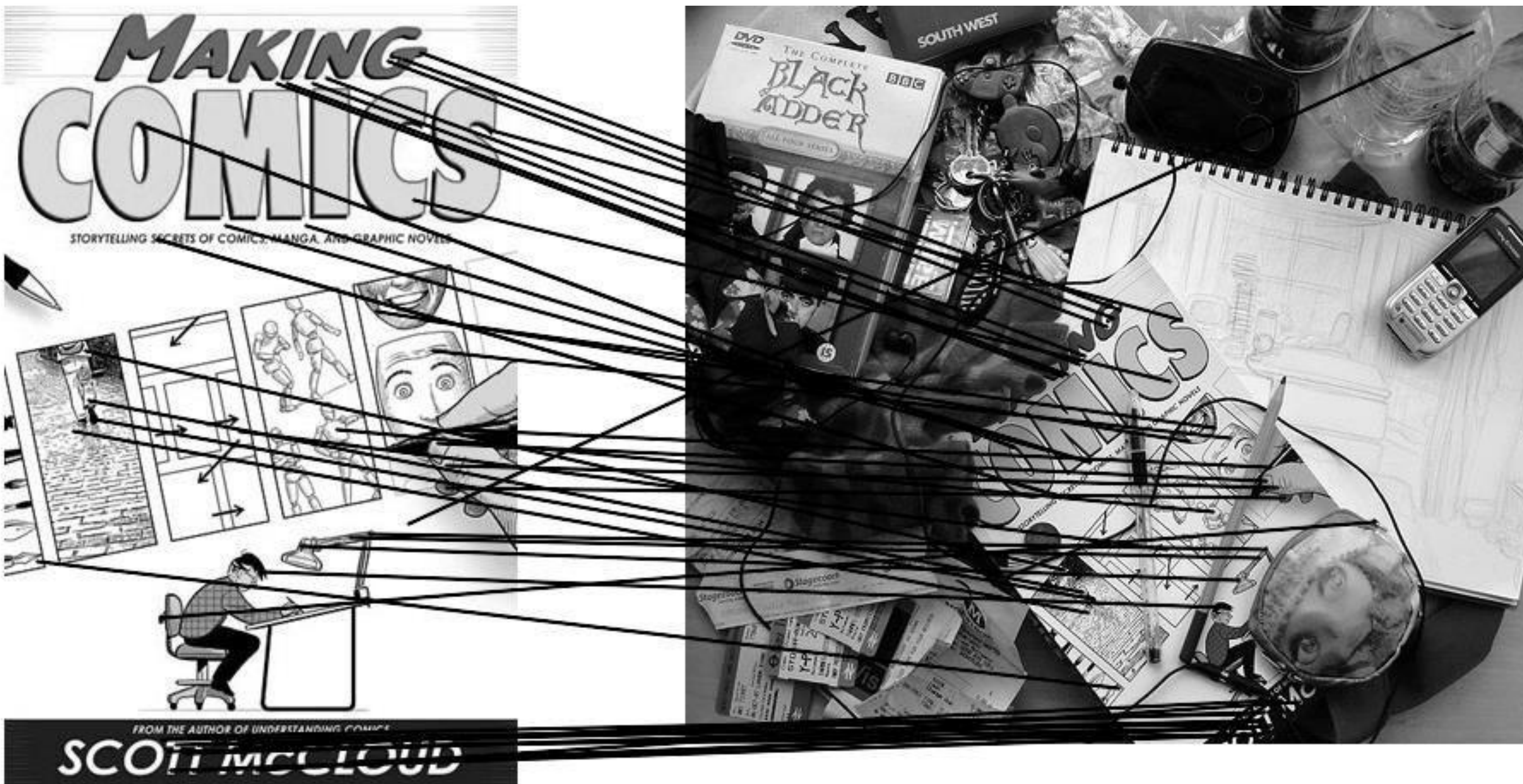


I_2

Feature matching example

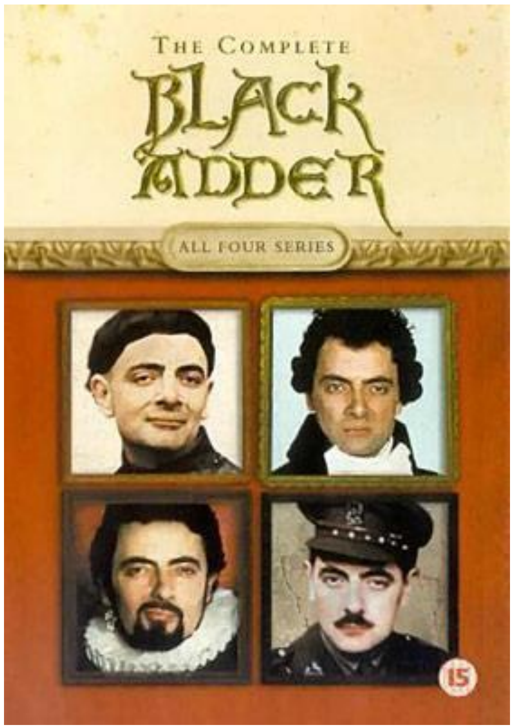


Feature matching example

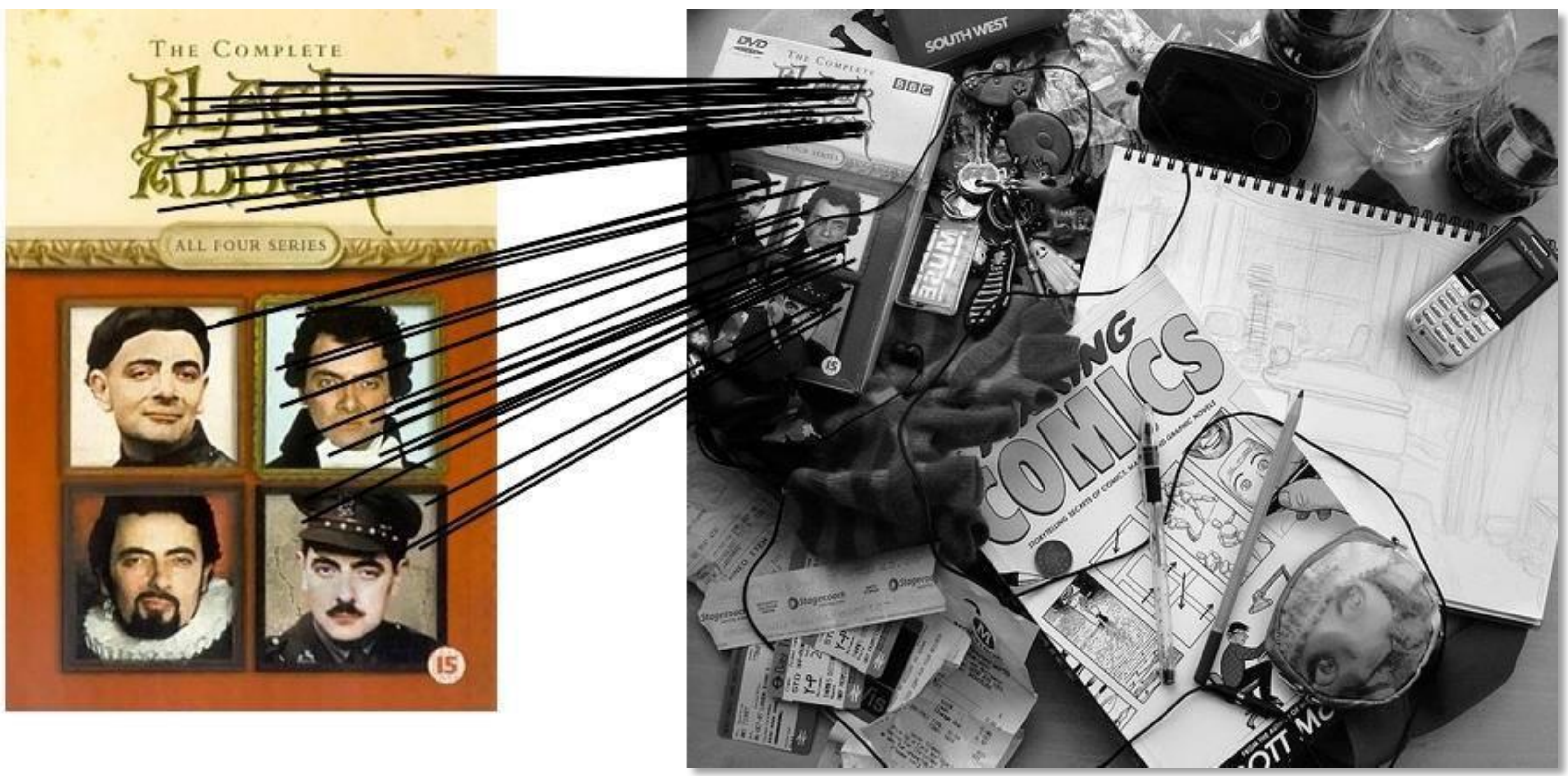


51 matches

Feature matching example



Feature matching example



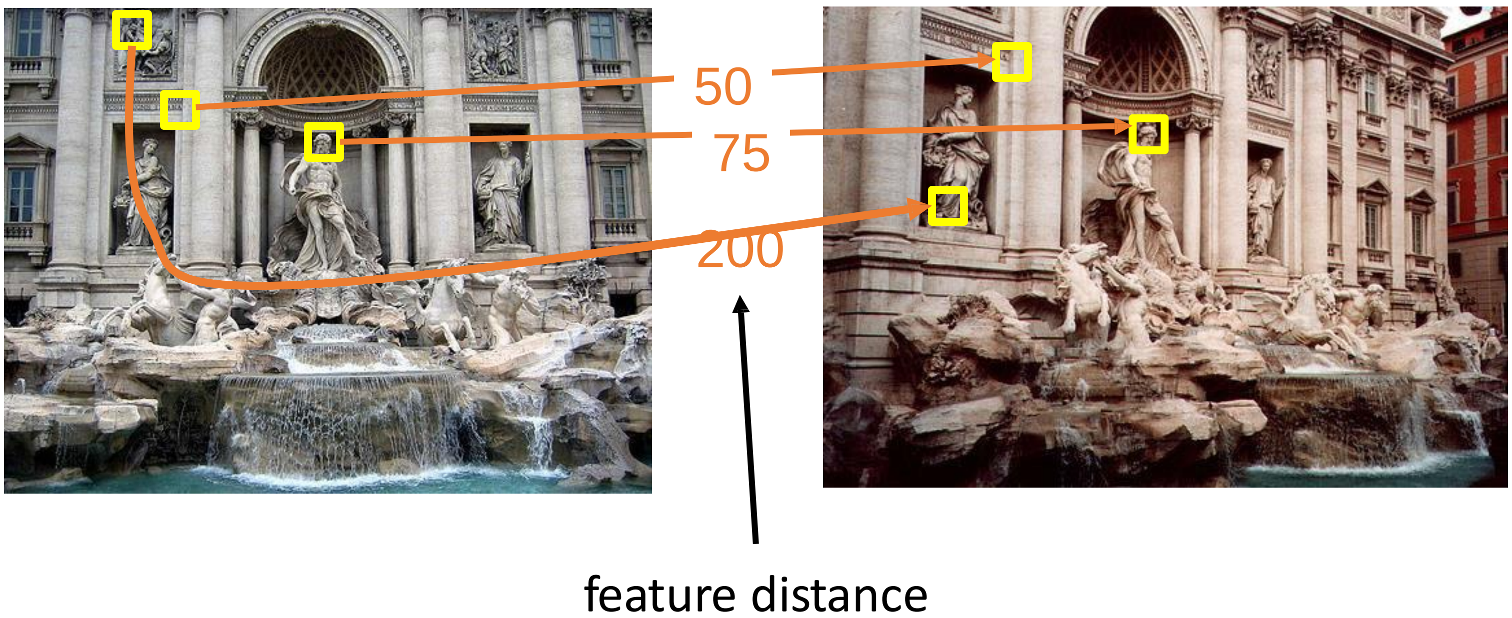
58 matches

Evaluating the results

How can we measure the performance of a feature matcher?

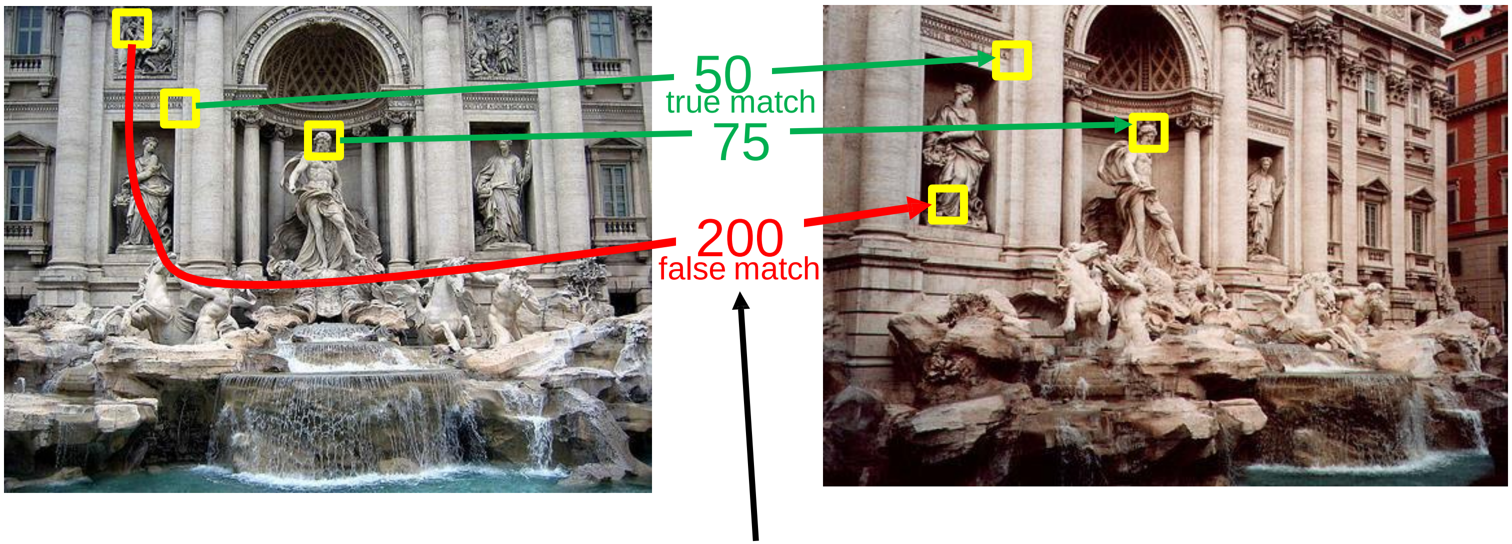
Evaluating the results

How can we measure the performance of a feature matcher?



True/false positives

How can we measure the performance of a feature matcher?

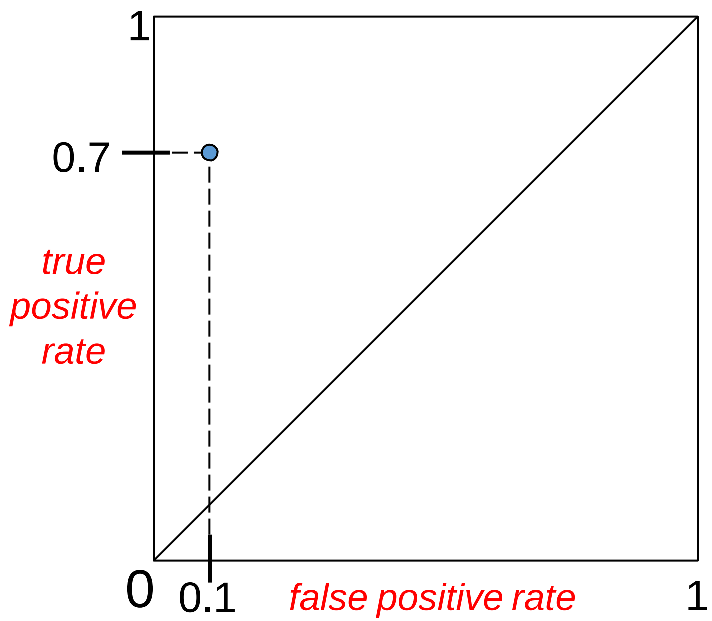


The distance threshold affects performance

- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?

Evaluating the results

How can we measure the performance of a feature matcher?



true positives / # correctly matched features (positives)

"recall"

false positives / # incorrectly matched features (negatives)

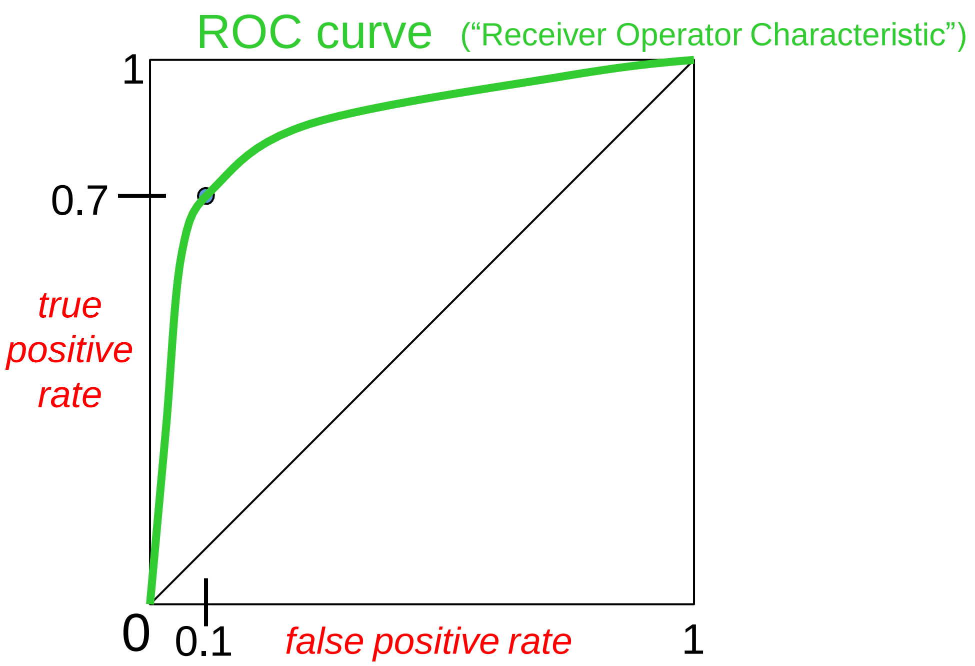
1 - "precision"

Evaluating the results

How can we measure the performance of a feature matcher?

$$\frac{\text{\# true positives}}{\text{\# correctly matched features (positives)}}$$

"recall"



$$\frac{\text{\# false positives}}{\text{\# incorrectly matched features (negatives)}}$$

1 - "precision"

Evaluating the results

- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.
- Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
- ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

Local Binary Pattern (LBP)

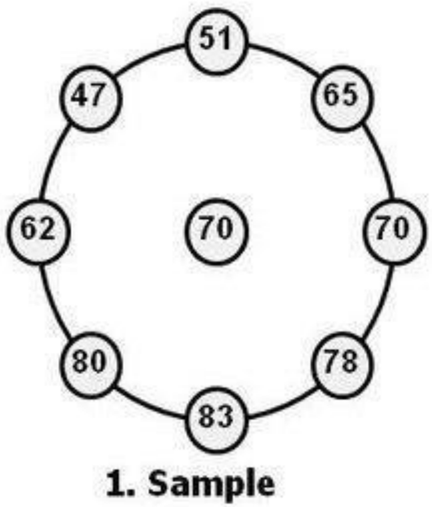


Image Source: scholarpedia

[IEEE TPAMI 2002]

Local Binary Pattern (LBP)

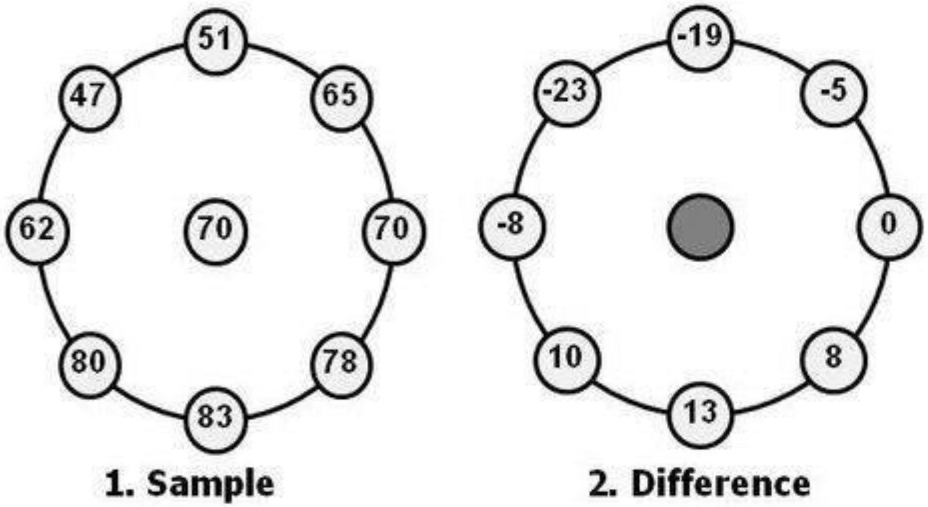


Image Source: scholarpedia

[IEEE TPAMI 2002]

Local Binary Pattern (LBP)

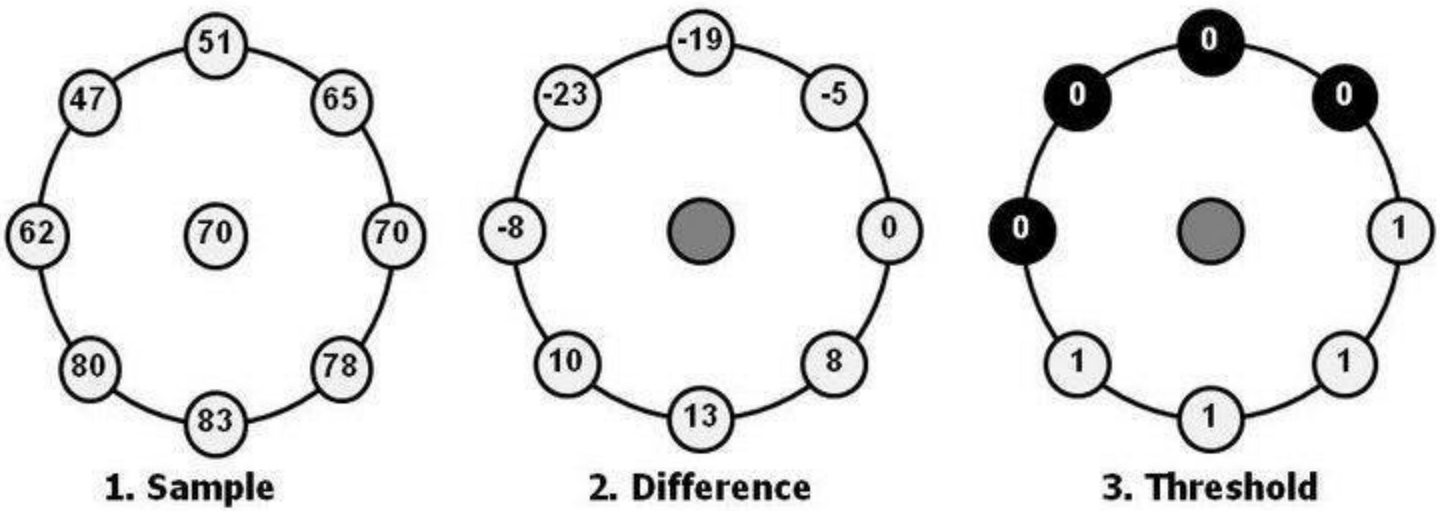


Image Source: scholarpedia

[IEEE TPAMI 2002]

Local Binary Pattern (LBP)

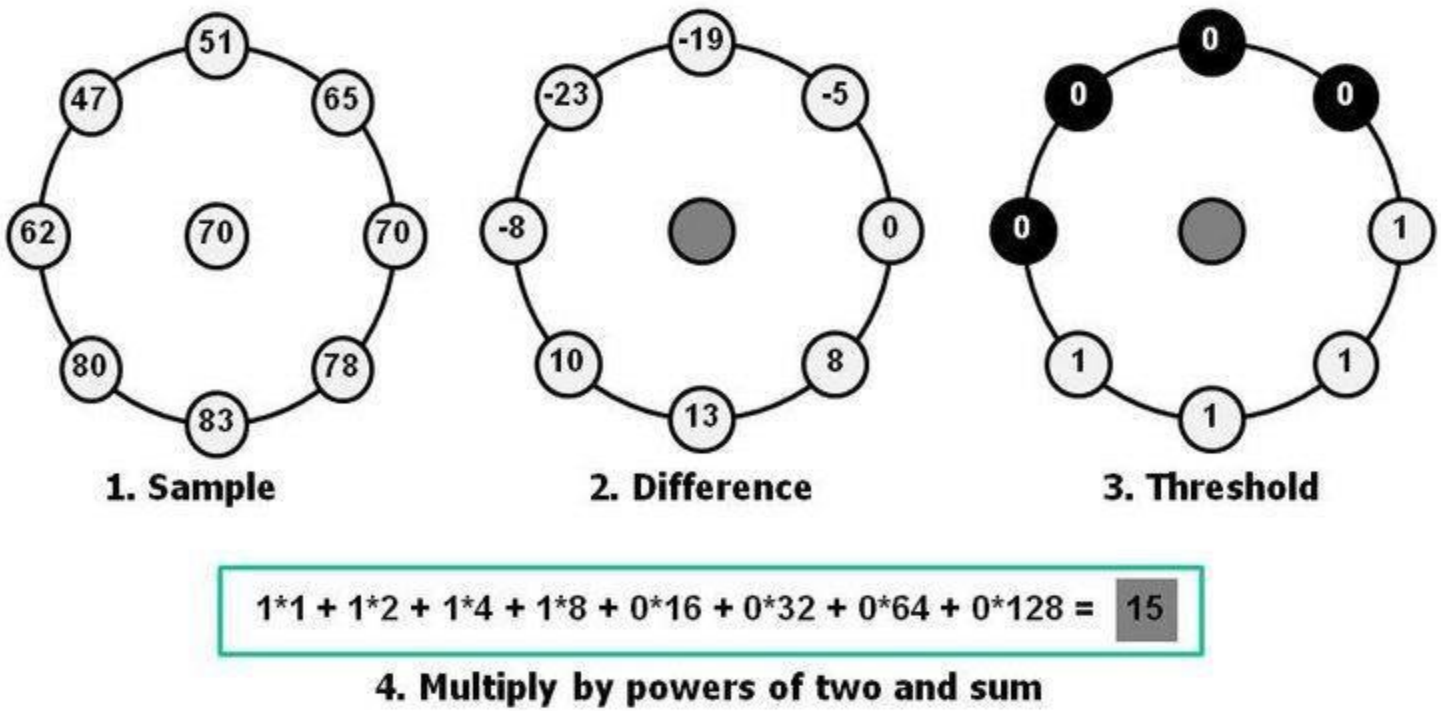


Image Source: scholarpedia

[IEEE TPAMI 2002]

Local Binary Pattern (LBP)

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$

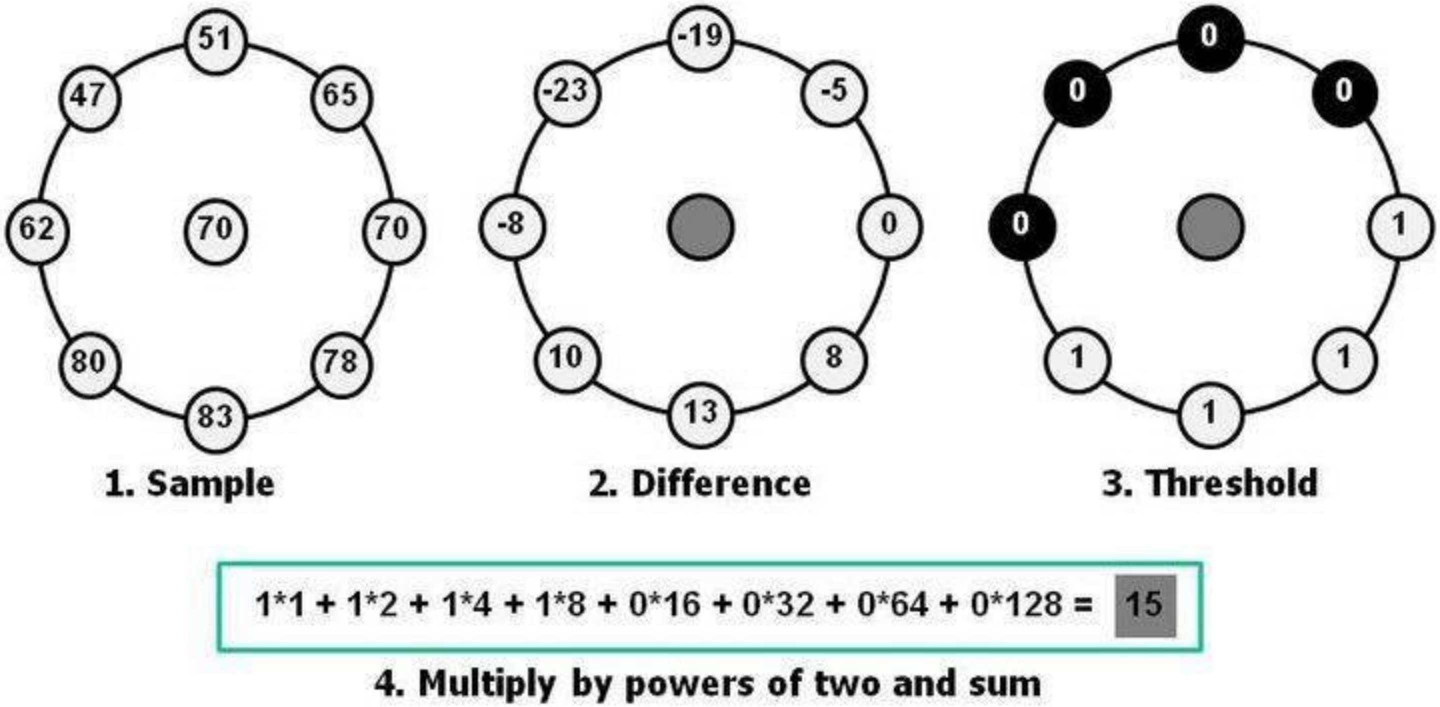


Image Source: scholarpedia

[IEEE TPAMI 2002]

Local Binary Pattern (LBP)

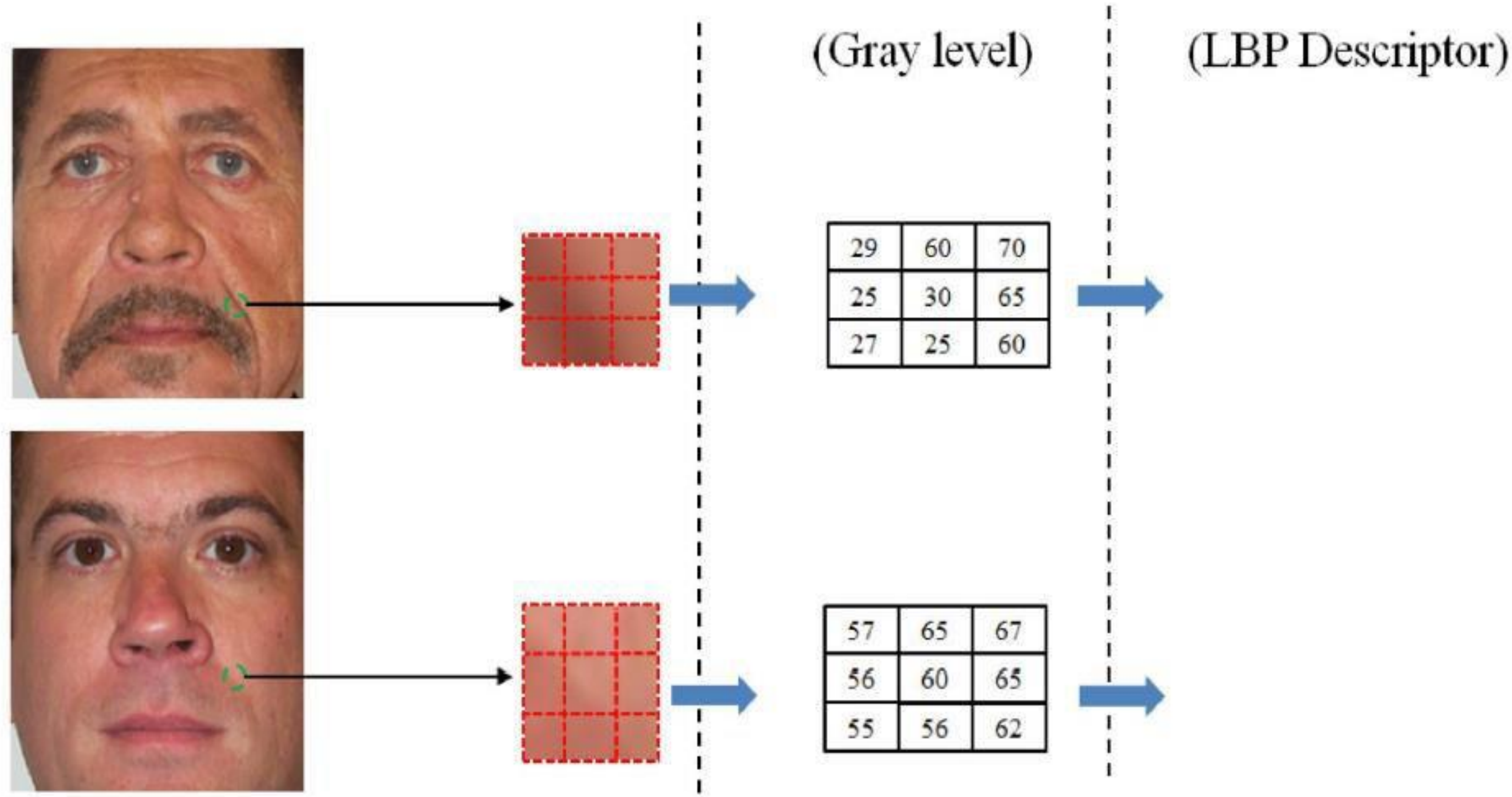


Image Source: Nguyen, D. T., Cho, S. R., & Park, K. R. (2015). Age Estimation-Based Soft Biometrics Considering Optical Blurring Based on Symmetrical Sub-Blocks for MLBP. *Symmetry*, *7*(4), 1882-1913.

Local Binary Pattern (LBP)

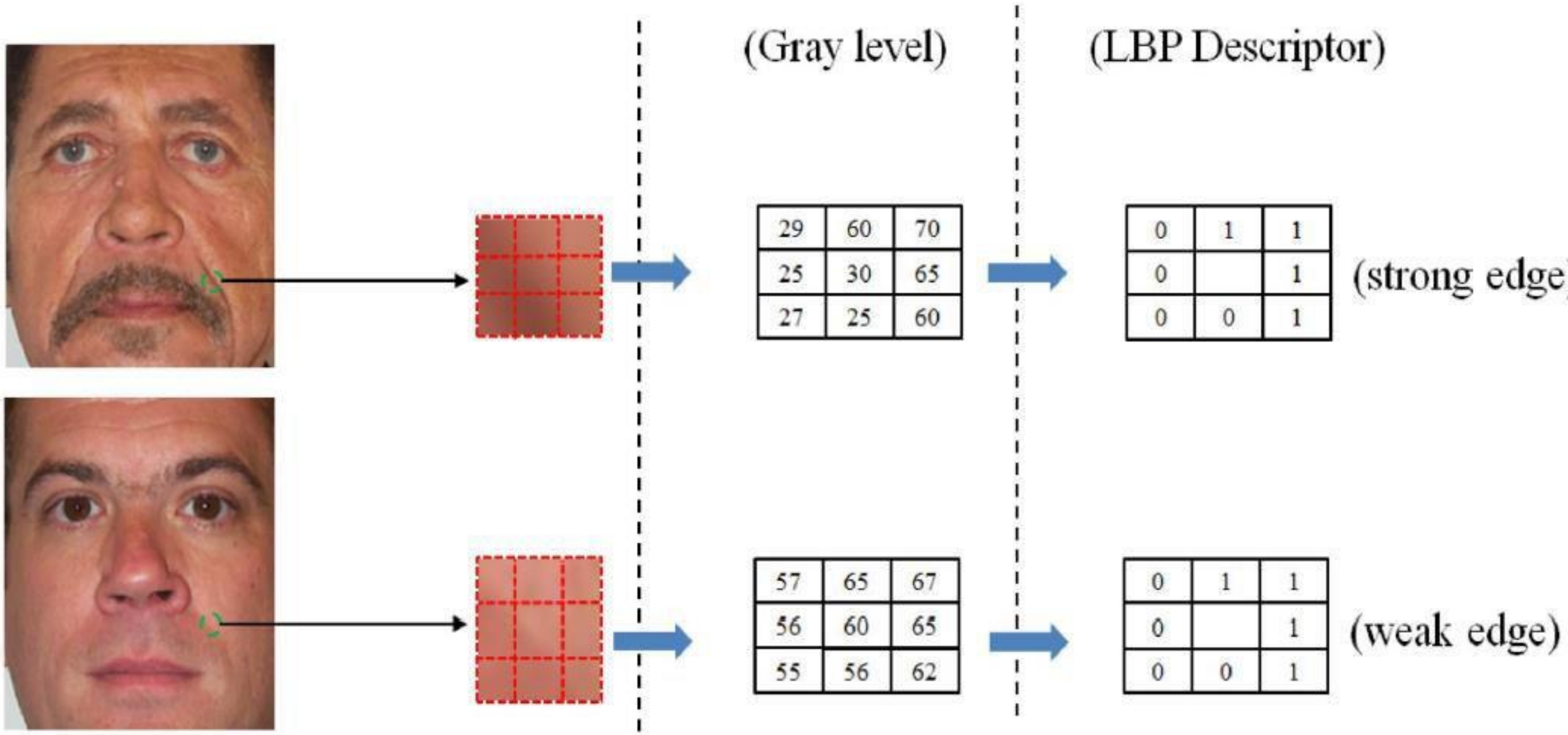


Image Source: Nguyen, D. T., Cho, S. R., & Park, K. R. (2015). Age Estimation-Based Soft Biometrics Considering Optical Blurring Based on Symmetrical Sub-Blocks for MLBP. *Symmetry*, *7*(4), 1882-1913.

Local Binary Pattern (LBP)

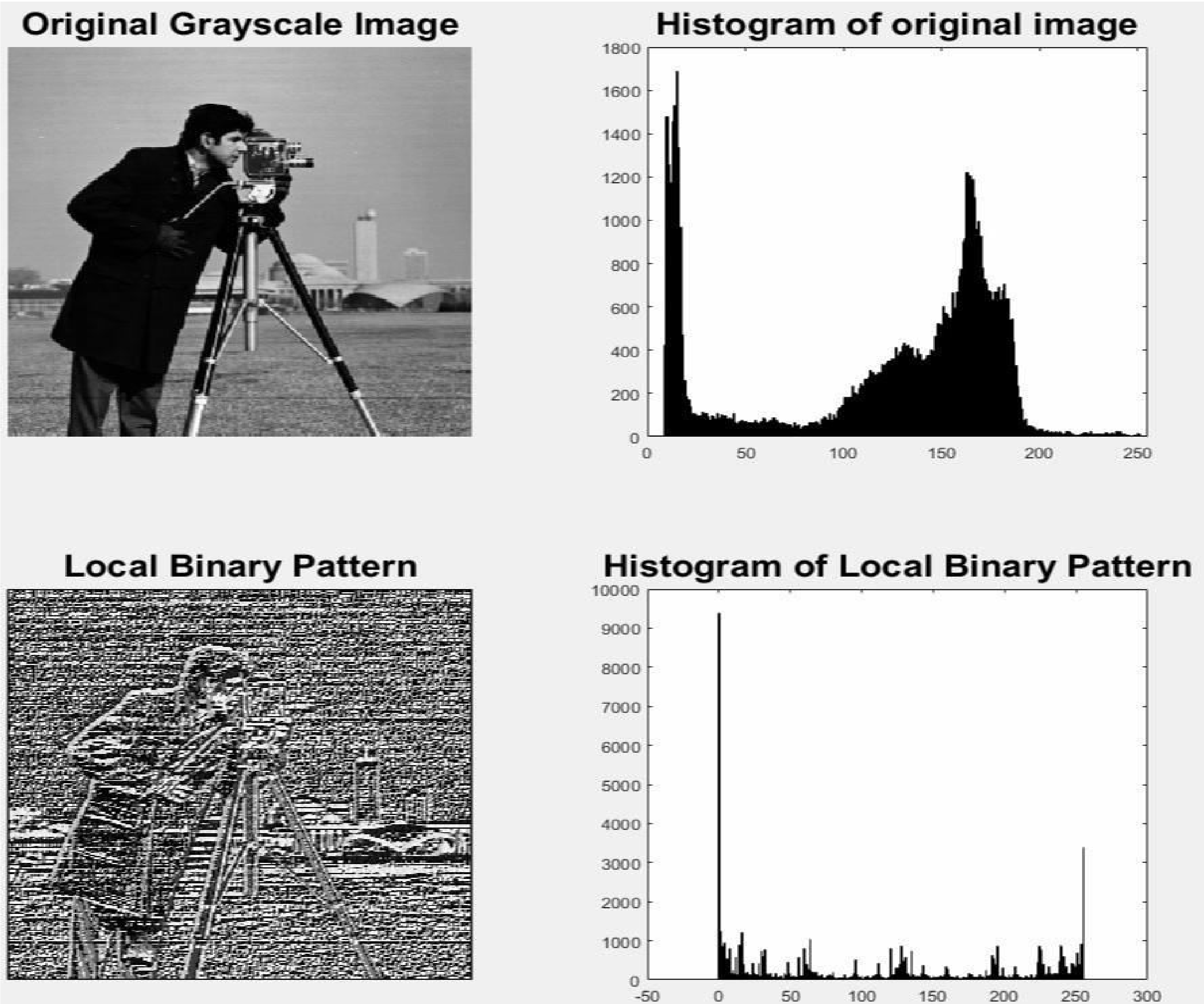
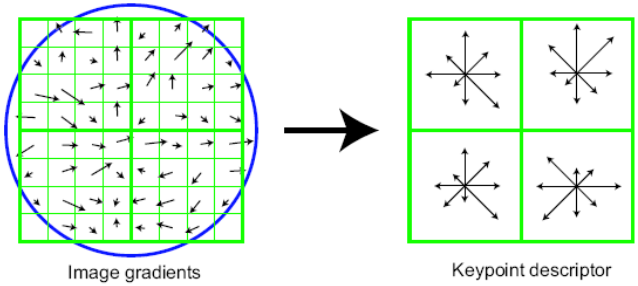


Image Source: Mathworks

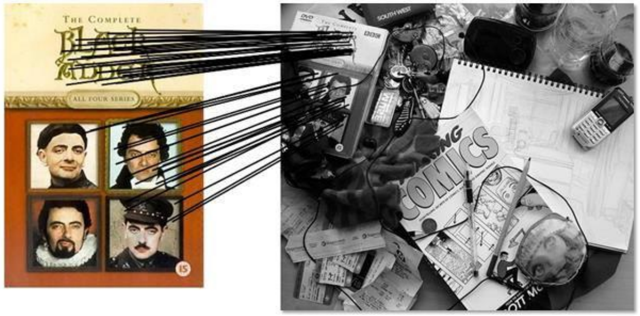
Things to remem

- Region detection: repeatable and distinctive
  - Blobs, DoG, stable regions
  - Invariant to affine transformation

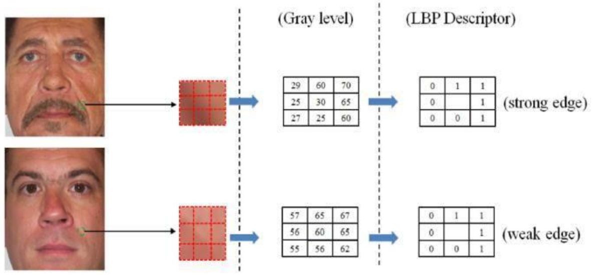
- Local Descriptors:
  - Discriminative
  - Robust
  - Compact
  - Efficient



- Scale Invariant Feature Transform
  - Utilizes gradient information
  - Gradient direction binning



- Local Binary Pattern
  - Exploits local relationship
  - Captures edge, spot, corner, etc.



Acknowledgements

- Thanks to the following researchers for making their teaching/research material online
  - Forsyth
  - Steve Seitz
  - Noah Snavely
  - J.B. Huang
  - Derek Hoiem
  - D. Lowe
  - A. Bobick
  - S. Lazebnik
  - K. Grauman
  - R. Zaleski
  - Leibe

Thank you

- Next class: feature tracking and optical flow