CS588: Image Search

Scale Invariant Region Selection and SIFT

Sung-Eui Yoon
(윤성의)

Course URL:
http://sgvr.kaist.ac.kr/~sungeui/IR

KAIST

Class Objectives (Ch. 2.2 and Ch. 2.3)

- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gradients (LoG) ≈ Difference of Gradients (DoG)
  - SIFT as a local descriptor

- At last time, we discussed:
  - Different conferences
  - Image descriptors that are invariant to various changes
  - Harris corner detector

From Points to Regions…

- The Harris and Hessian operators define interest points.
  - Precise localization
  - High repeatability

- In order to compare those points, we need to compute a descriptor over a region.
  - How can we define such a region in a scale invariant manner?

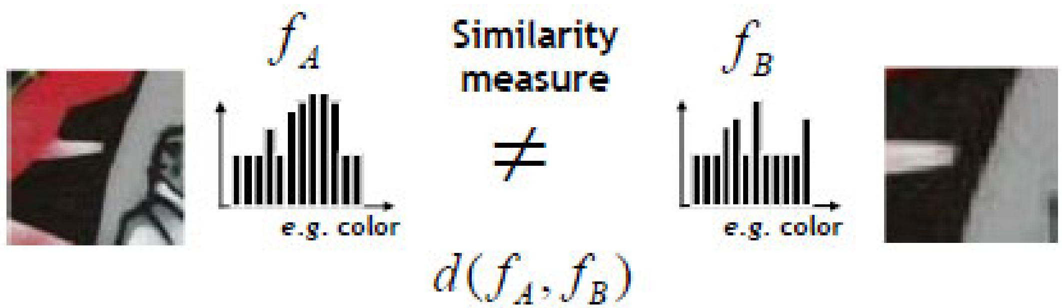
- *I.e. how can we detect scale invariant interest regions?*

Source: Bastian Leibe

KAIST

Naïve Approach: Exhaustive Search

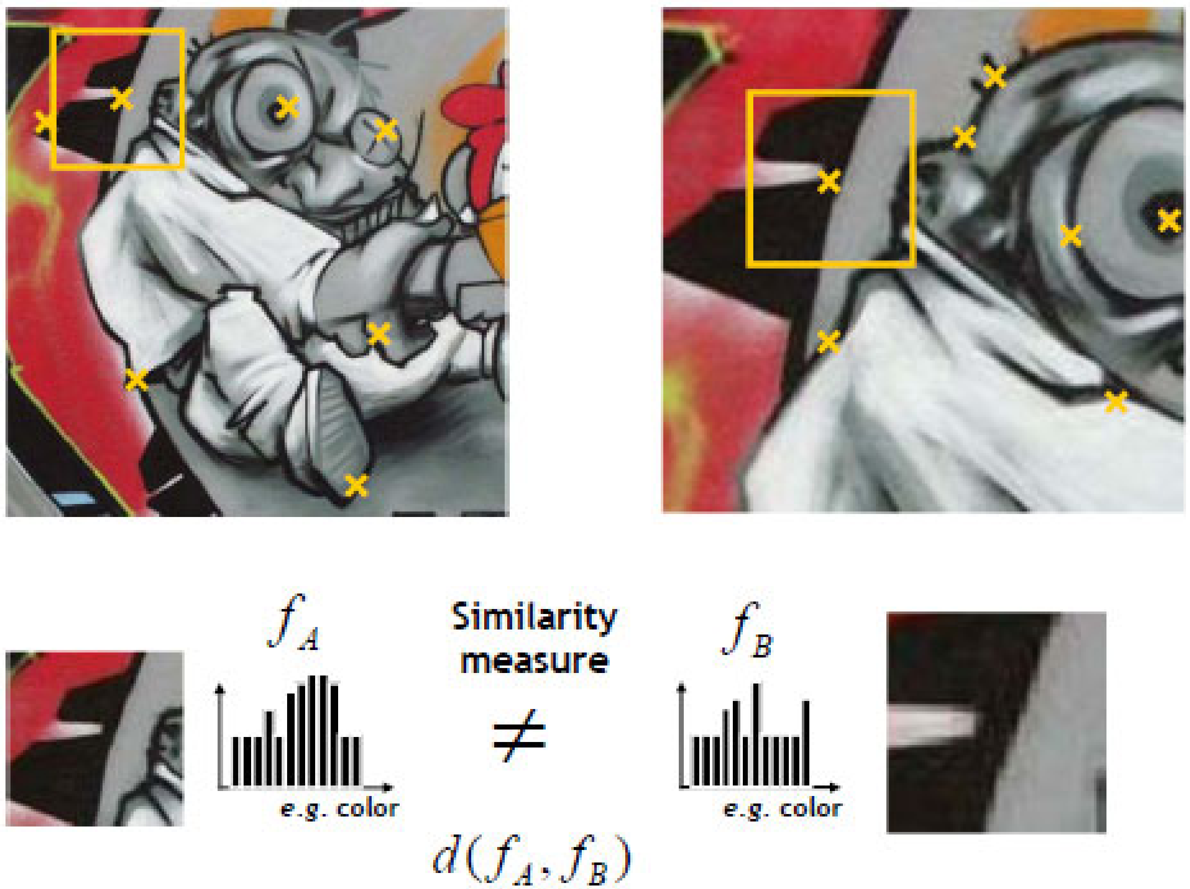
- Multi-scale procedure
  - Compare descriptors while varying the patch size



Slide credit: Krystian Mikolajczyk

Naïve Approach: Exhaustive Search

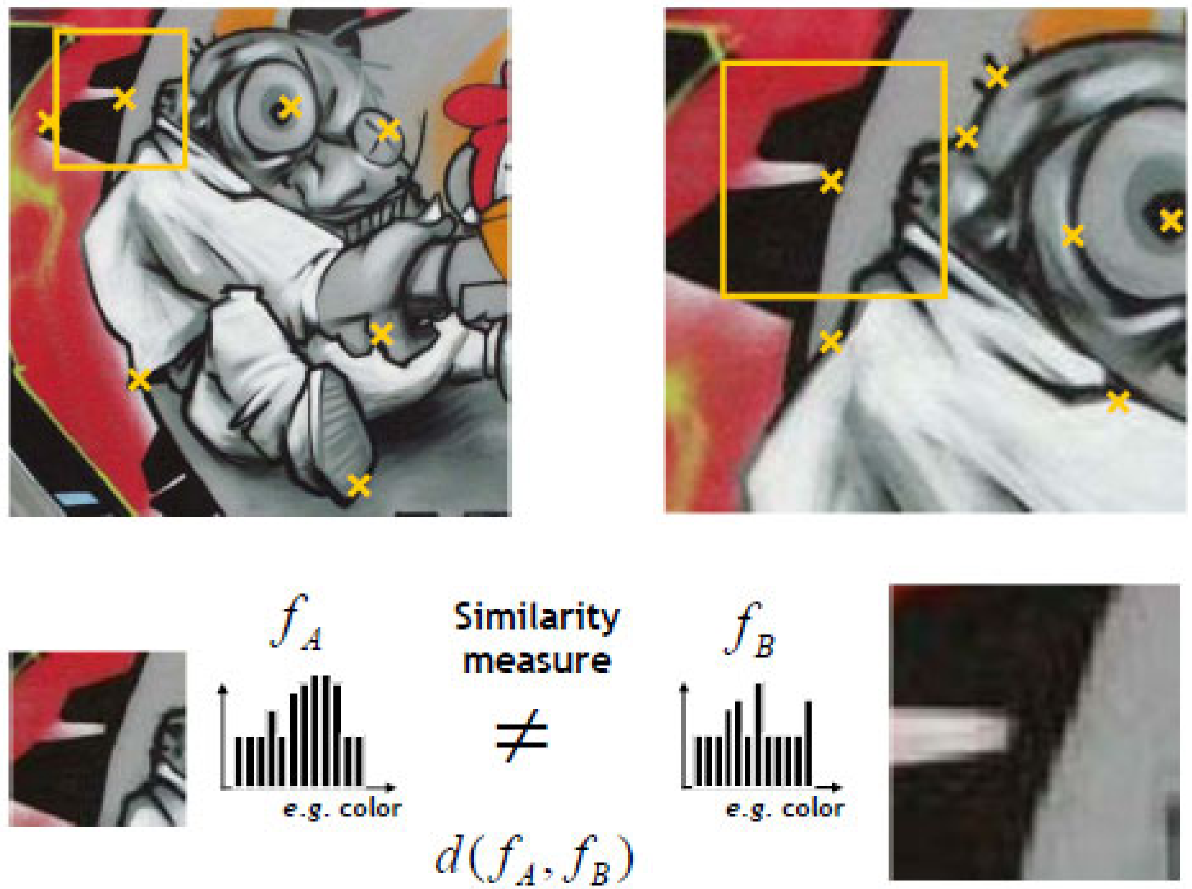
- Multi-scale procedure
  - Compare descriptors while varying the patch size



Slide credit: Krystian Mikolajczyk

Naïve Approach: Exhaustive Search

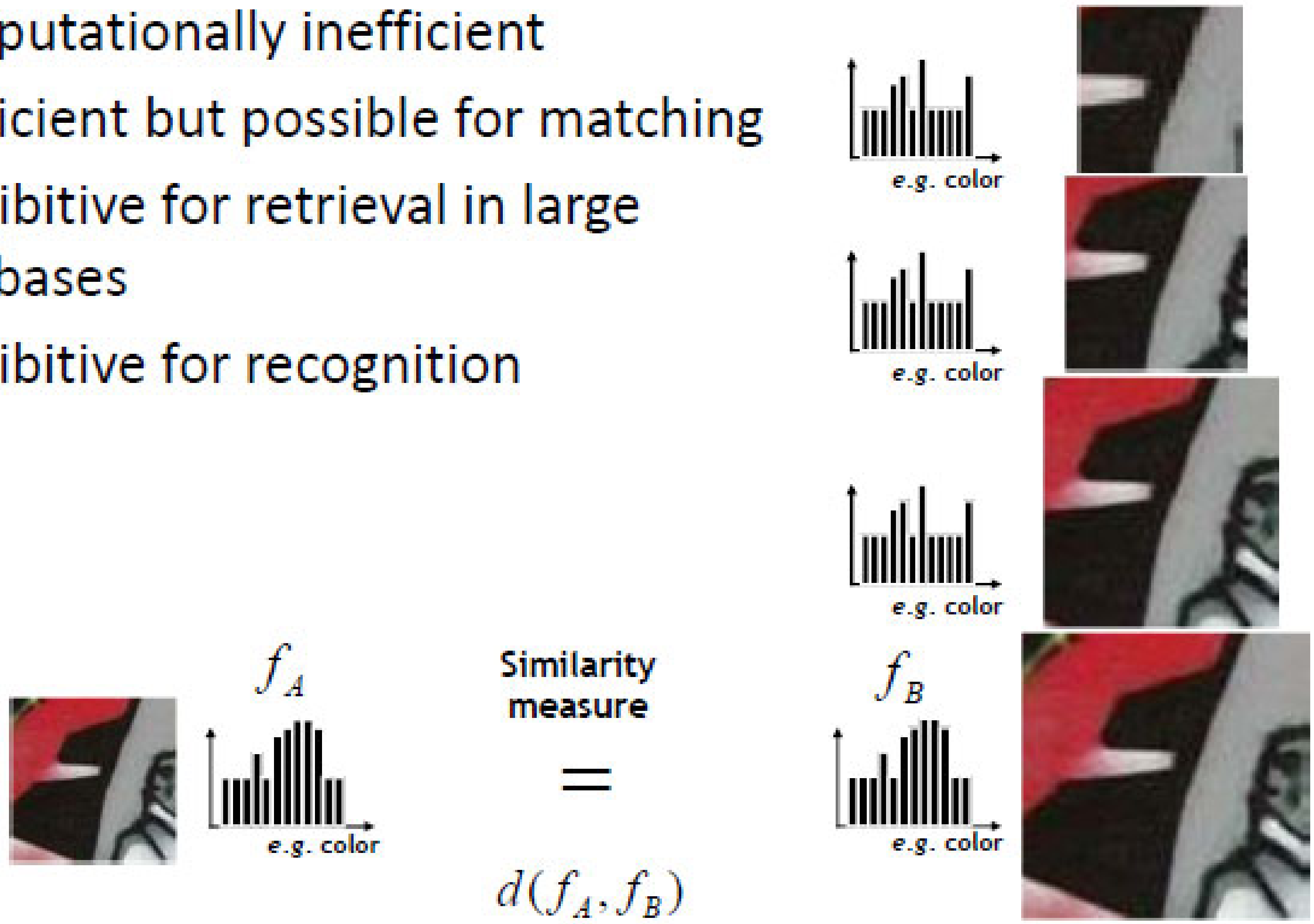
- Multi-scale procedure
  - Compare descriptors while varying the patch size



Slide credit: Krystian Mikolajczyk

Naïve Approach: Exhaustive Search

- Comparing descriptors while varying the patch size
  - Computationally inefficient
  - Inefficient but possible for matching
  - Prohibitive for retrieval in large databases
  - Prohibitive for recognition



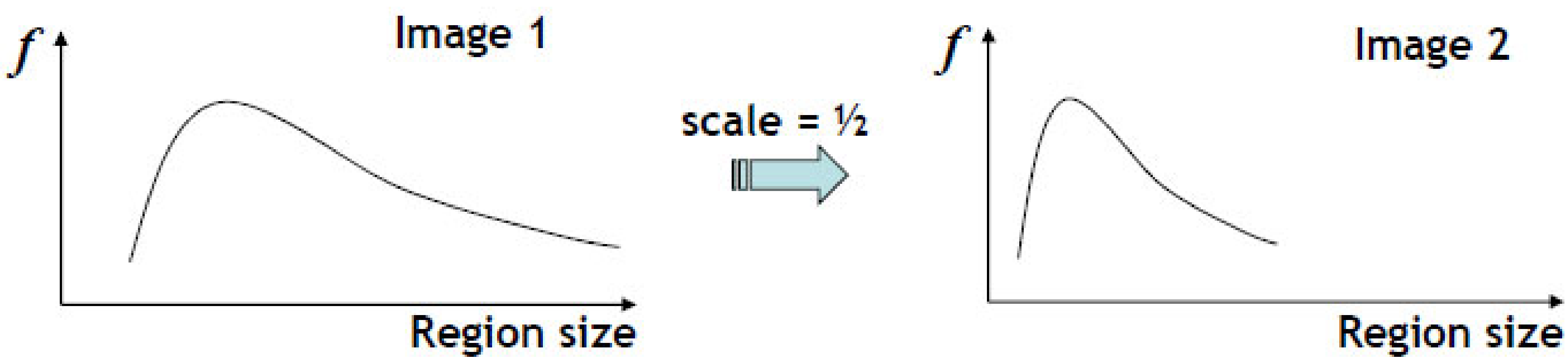
Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Solution:
  - Design a function on the region, which is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (patch width)

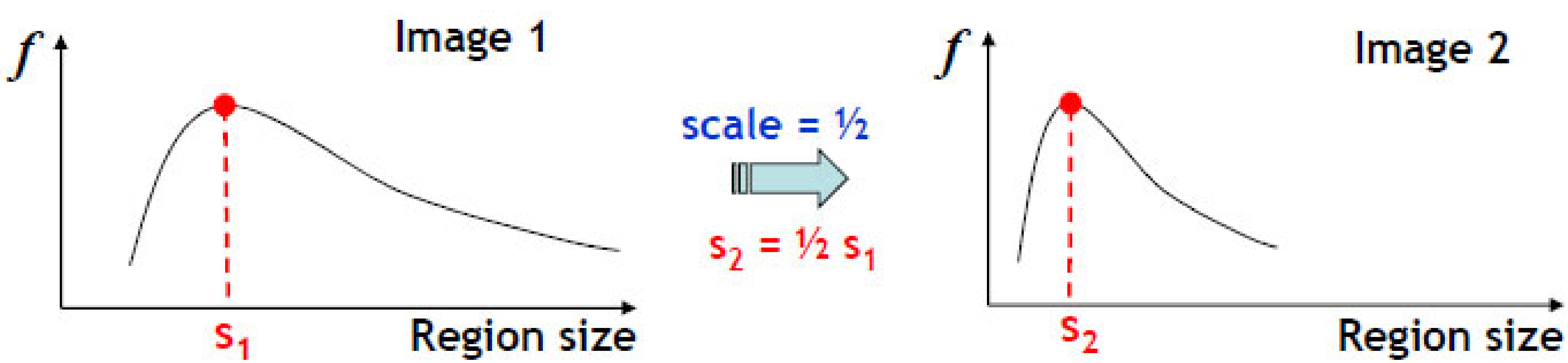


Slide credit: Kristen Grauman

KAIST

Automatic Scale Selection

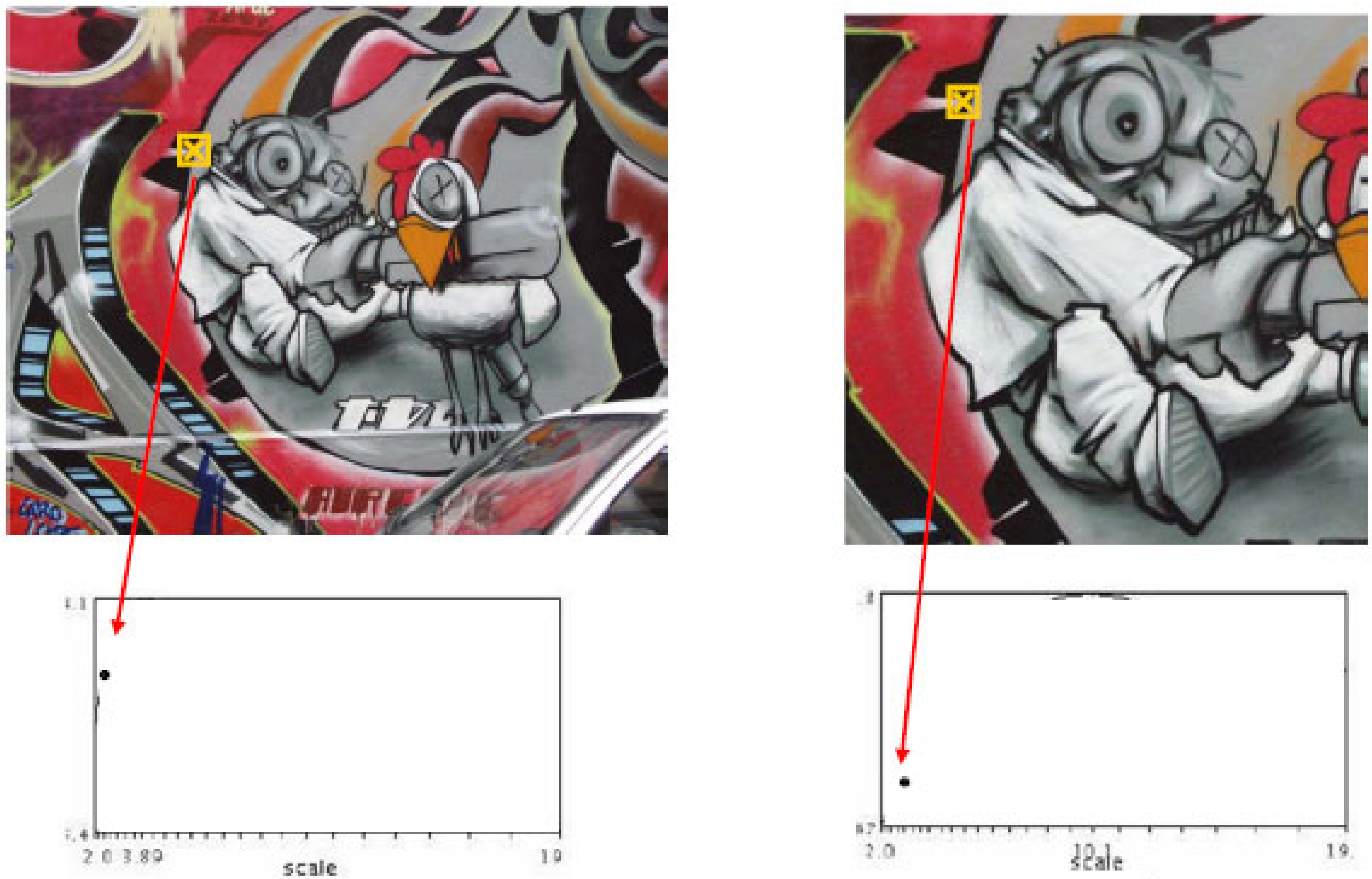
- Common approach:
  - Take a local maximum of this function.
  - Observation: region size for which the maximum is achieved should be *invariant* to image scale.



Slide credit: Kristen Grauman

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

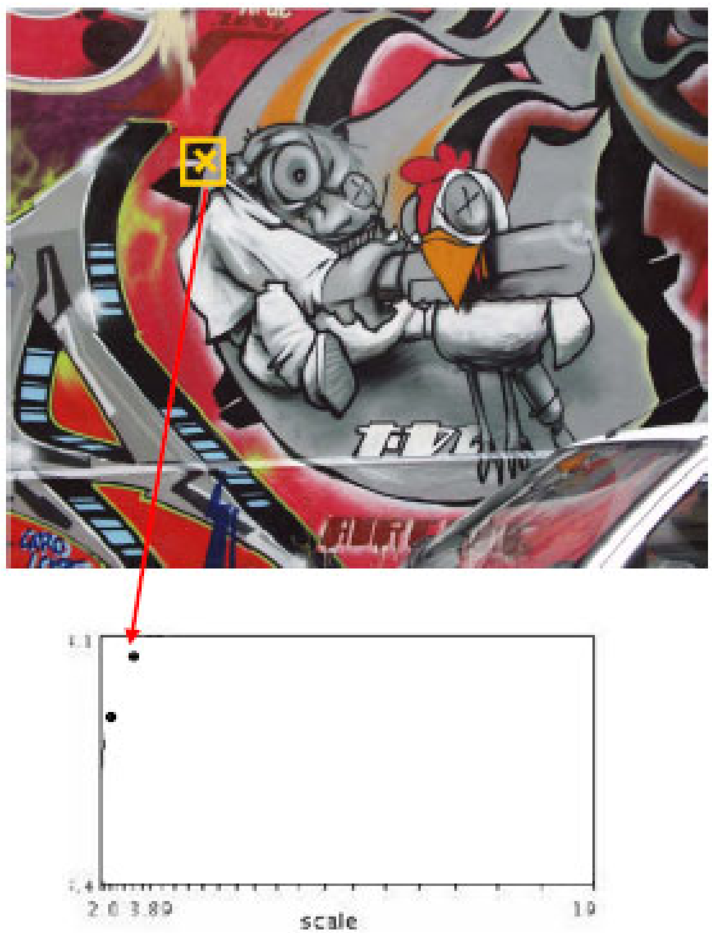


$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



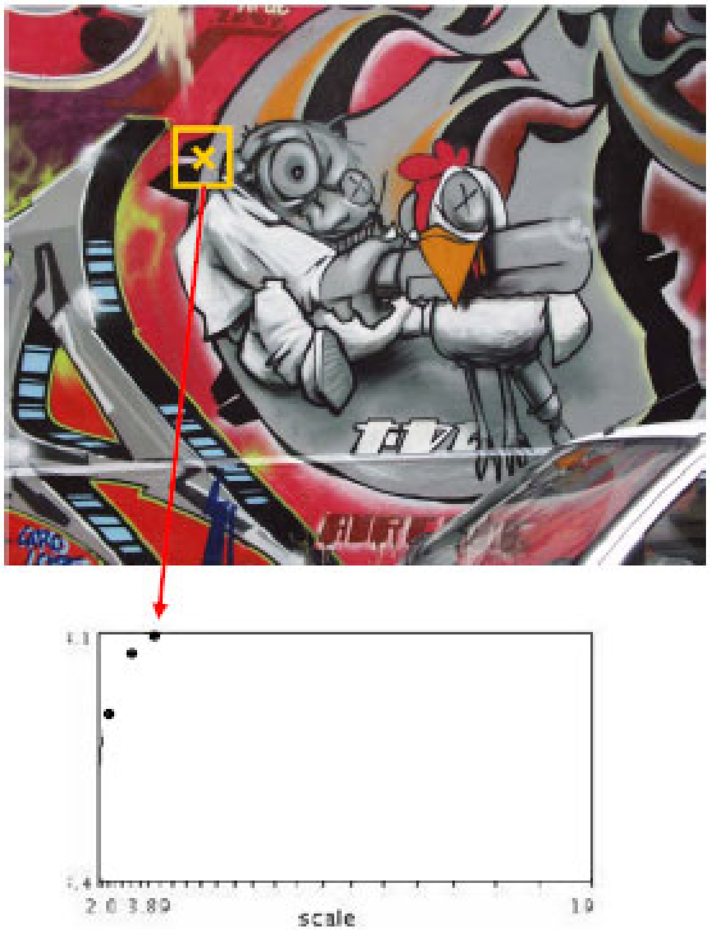
$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



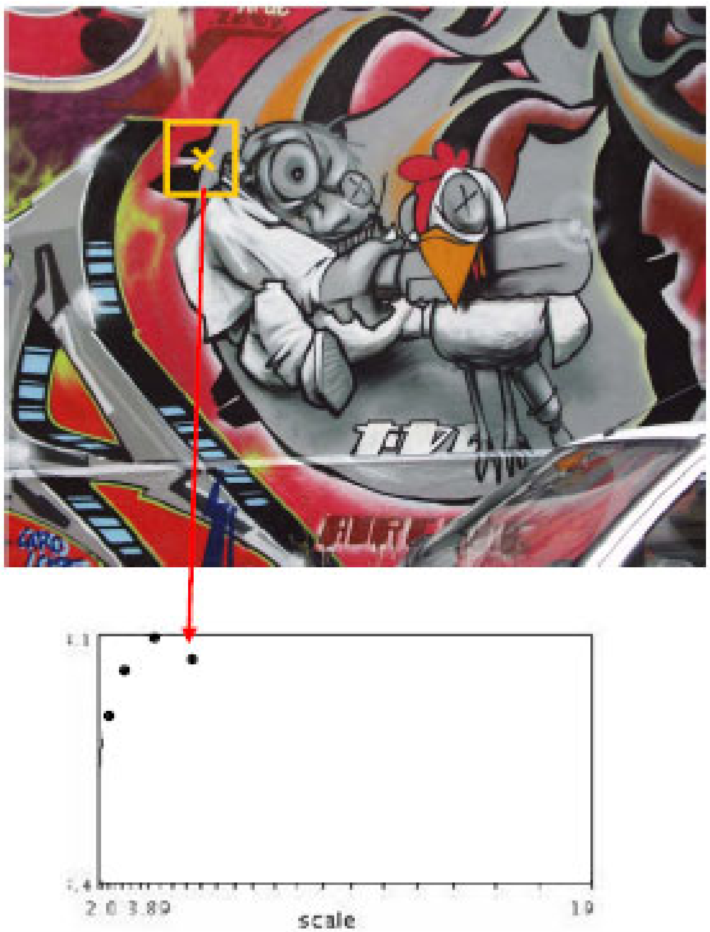
$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



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

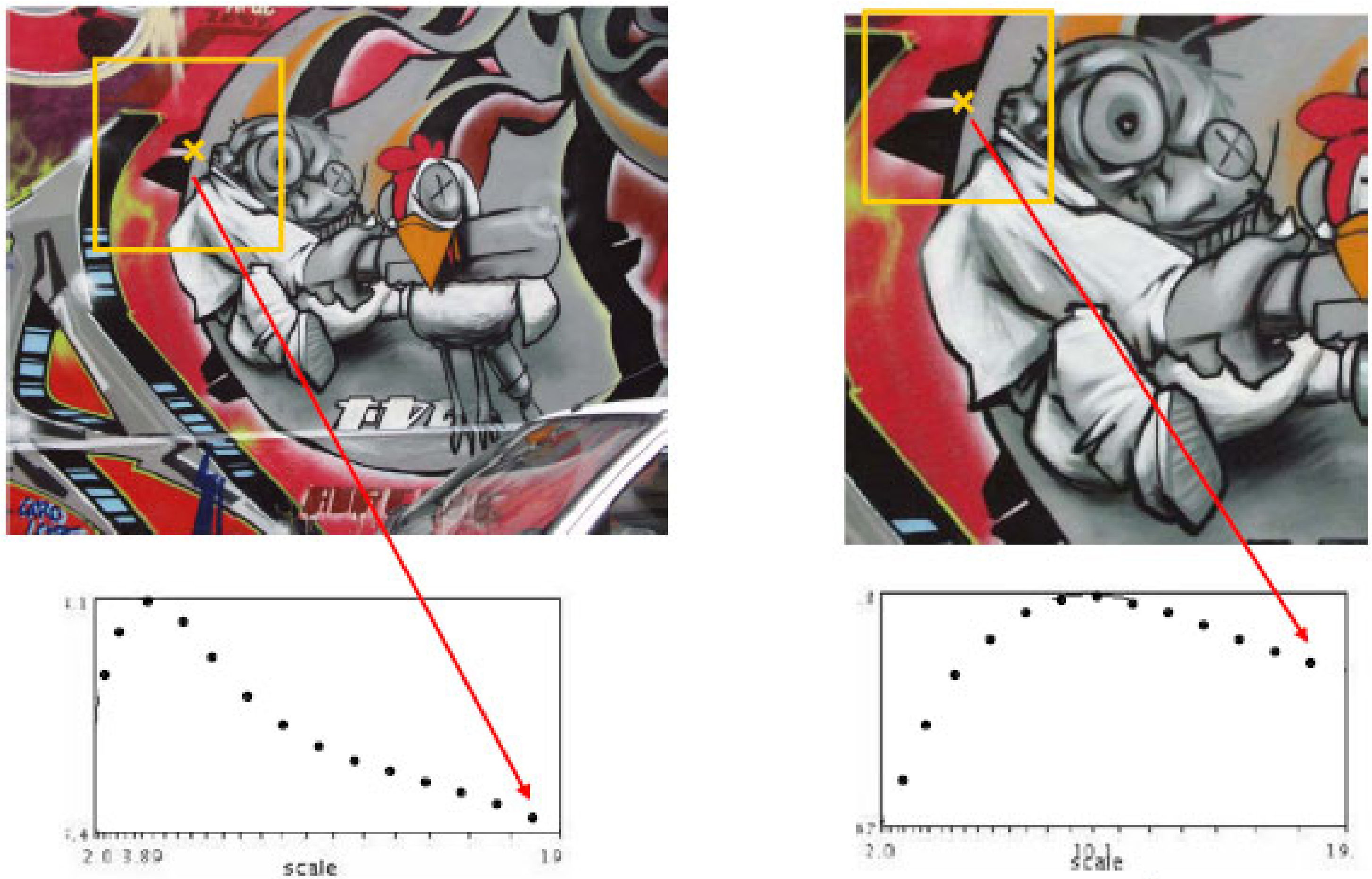
$f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

KAIST

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

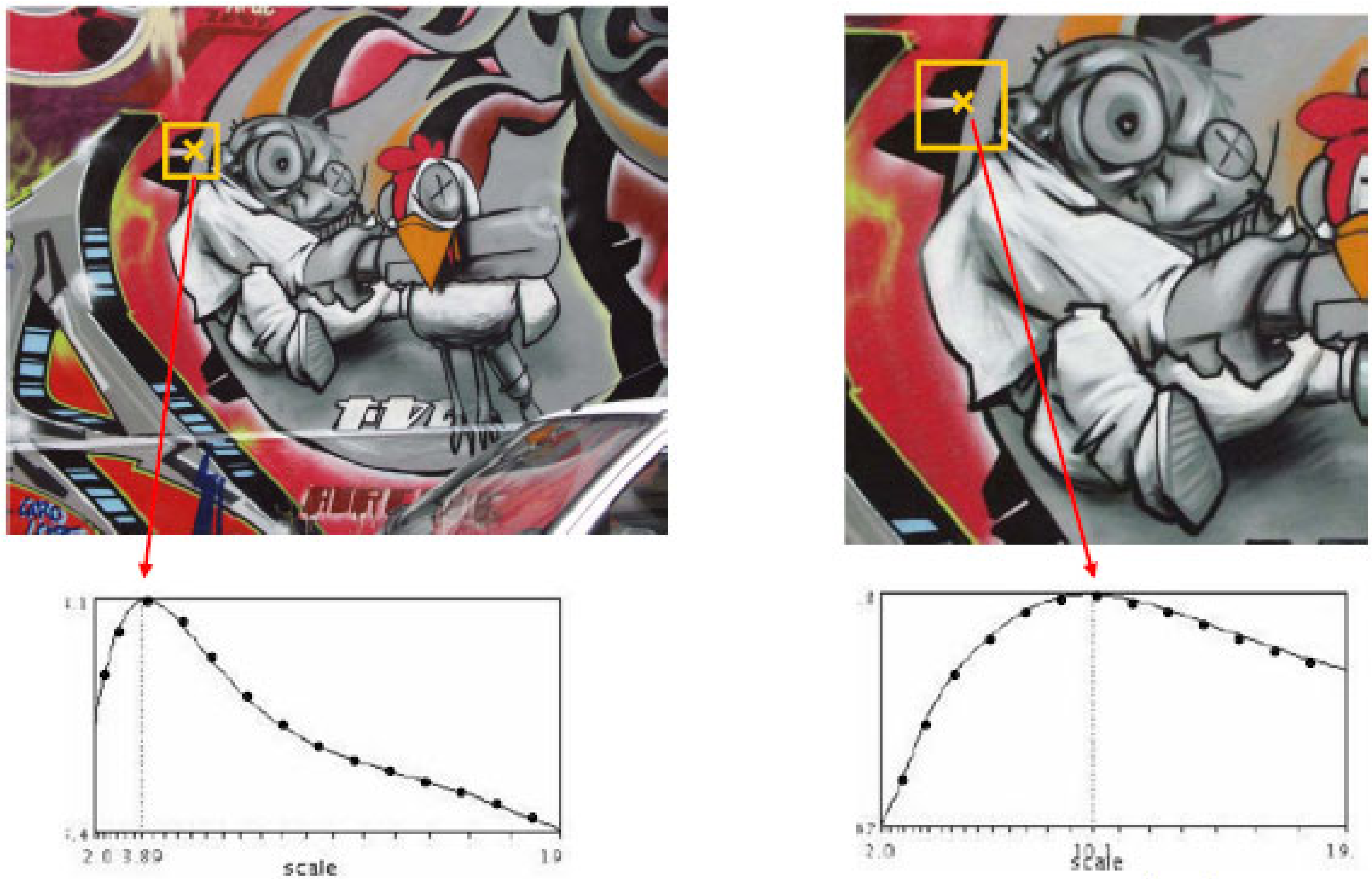


$f(I_{i_1 \dots i_m}(x, \sigma))$ $f(I_{i_1 \dots i_m}(x', \sigma))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

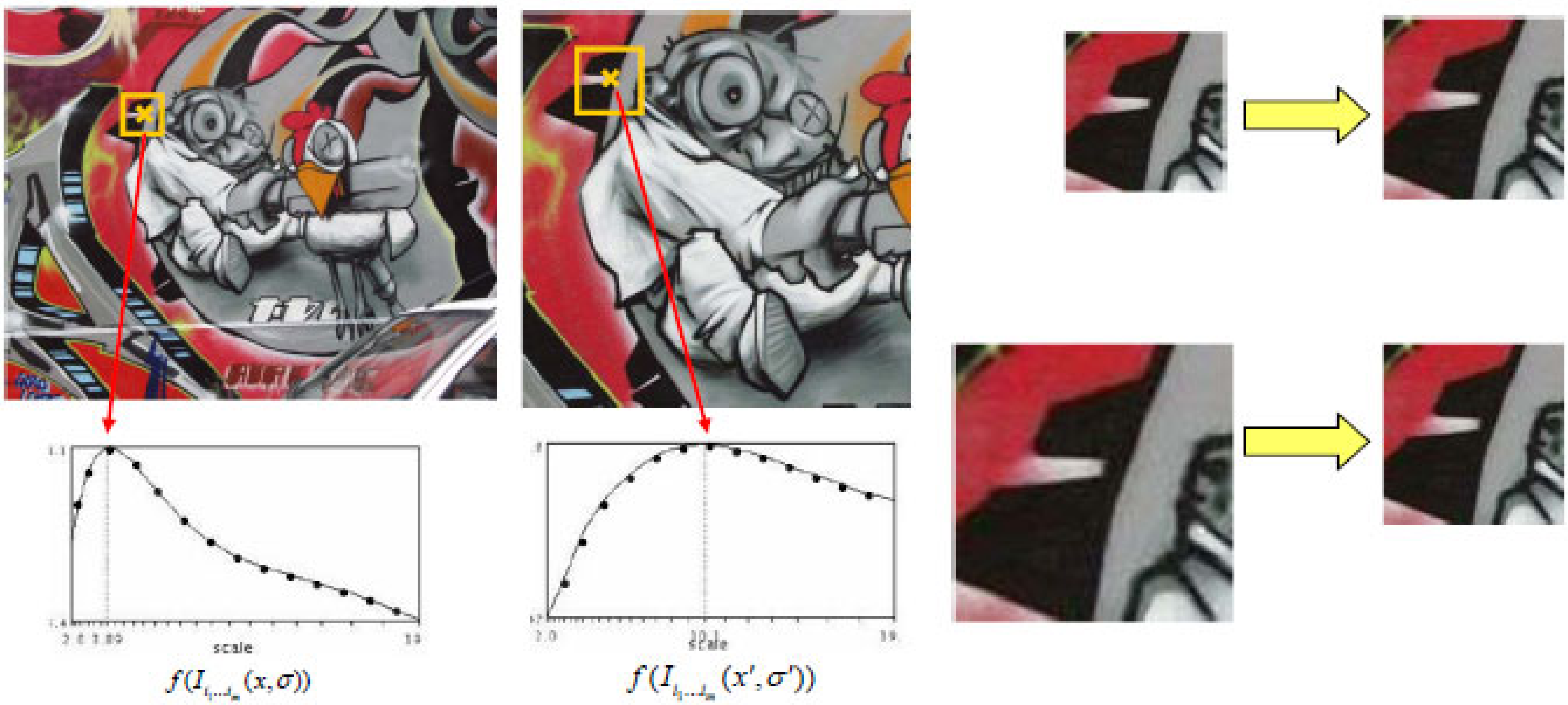


$f(I_{i_1 \ldots i_m}(x, \sigma))$ $f(I_{i_1 \ldots i_m}(x', \sigma'))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

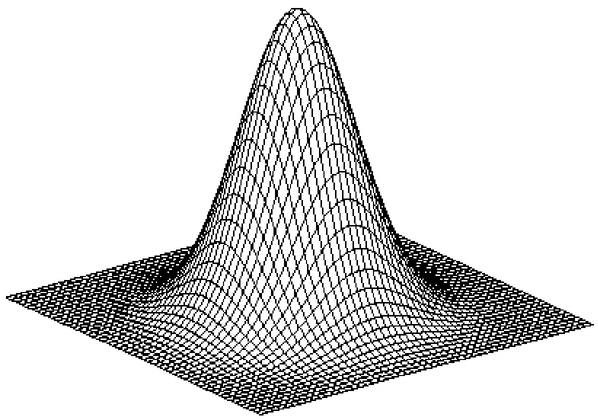
- Normalize: Rescale to fixed size



Slide credit: Tinne Tuytelaars

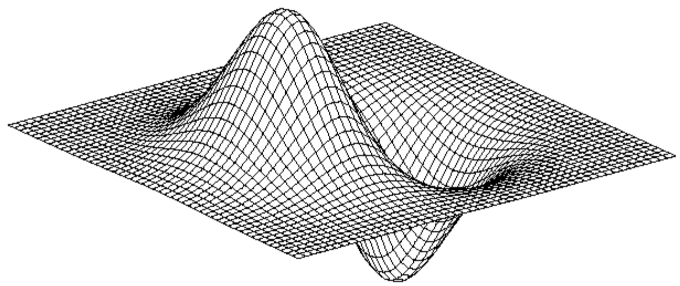
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector



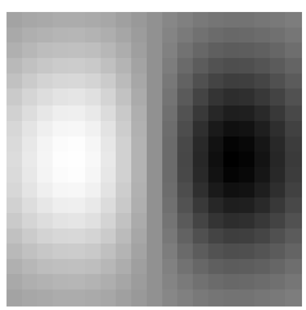
Gaussian

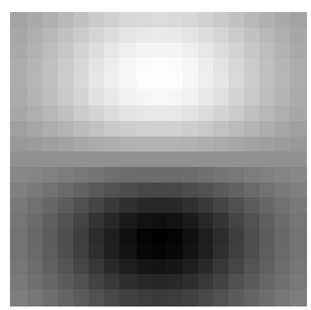
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



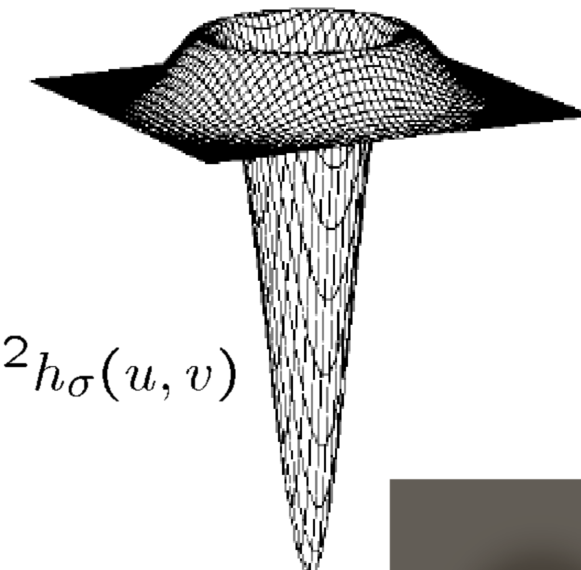
Derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$





Laplacian of Gaussian

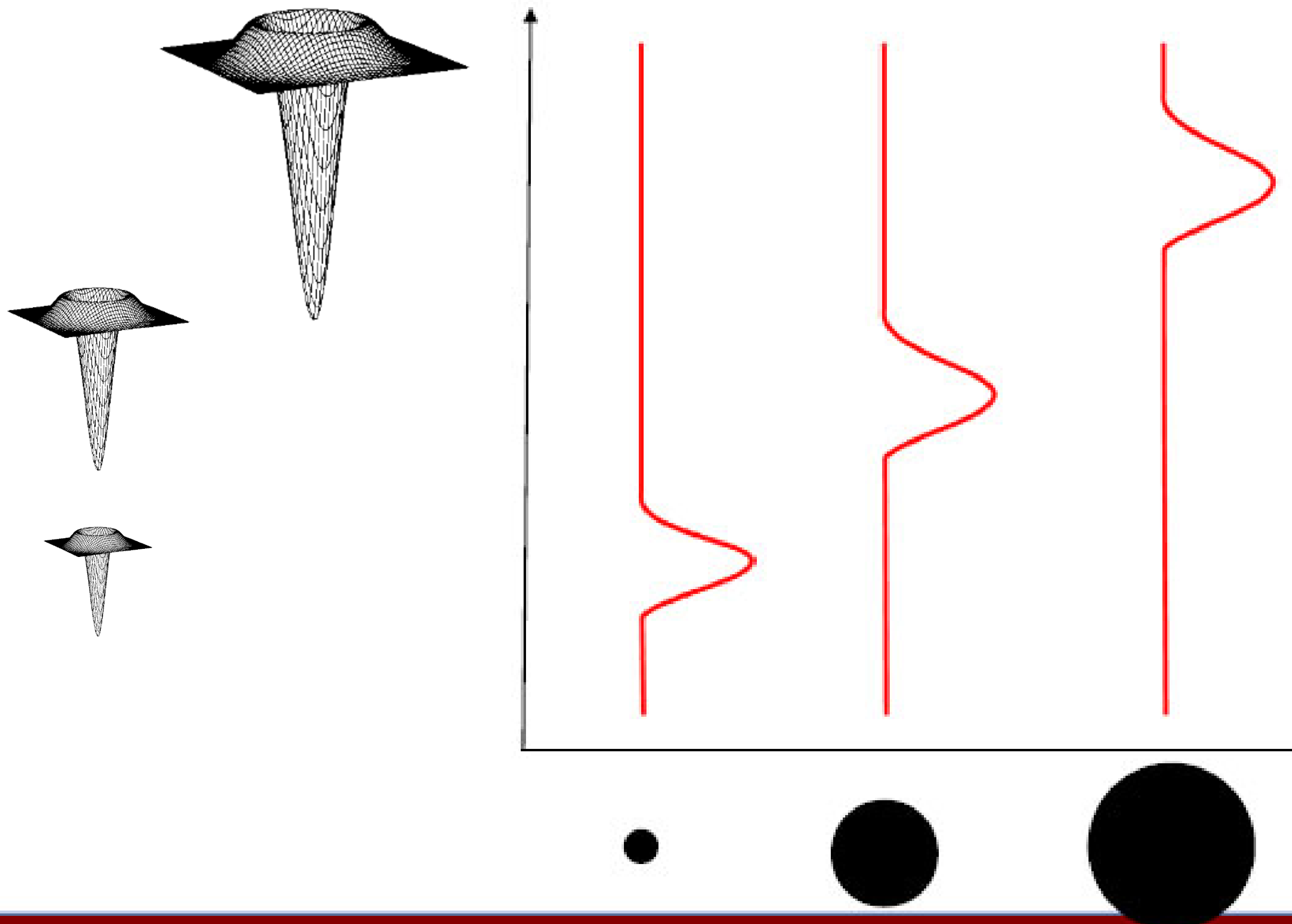


$$\nabla^2 h_\sigma(u,v)$$

Adapted from Steve Seitz, U of Washington

What Is A Useful Signature Function?

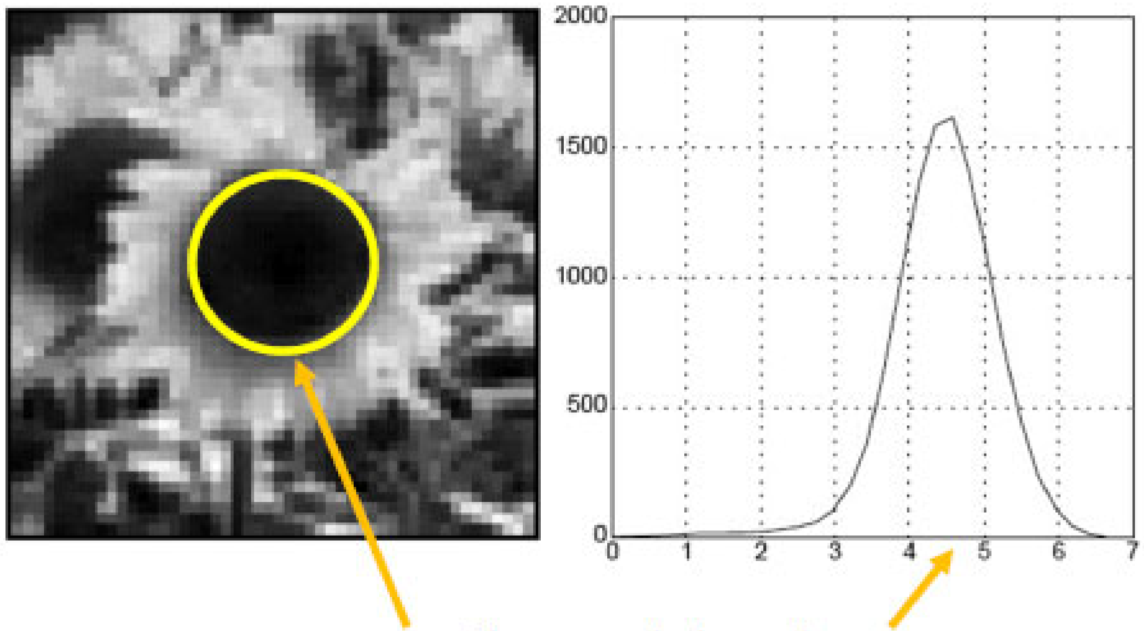
- Laplacian-of-Gaussian = “blob” detector



Slide credit: Bastian Leibe

Characteristic Scale

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



Characteristic scale

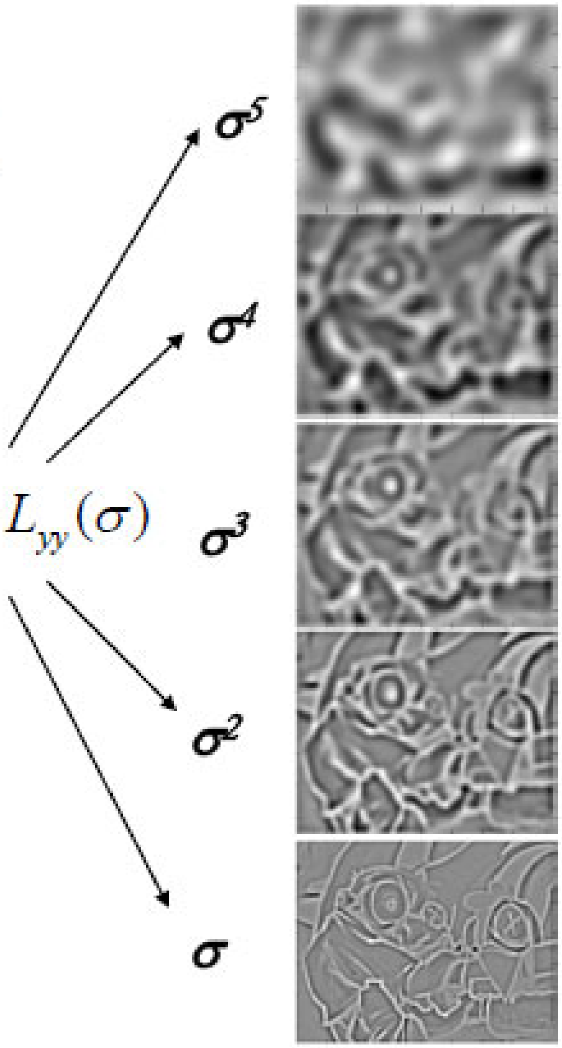
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

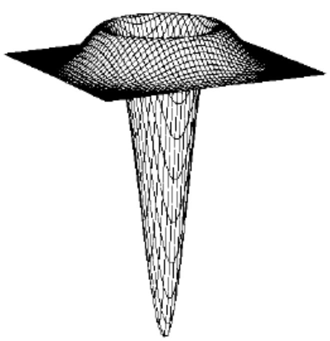
Slide credit: Svetlana Lazebnik

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

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



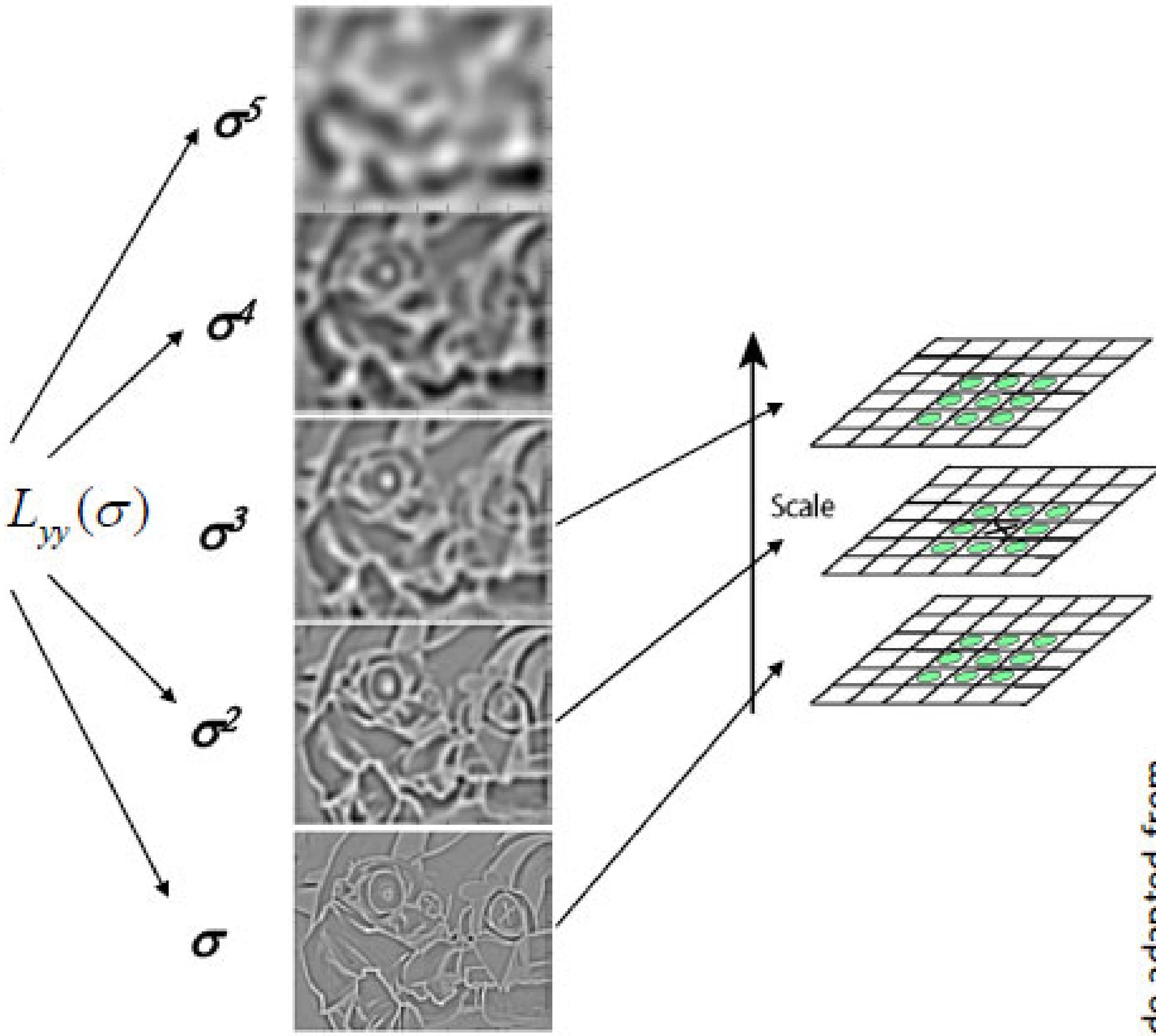


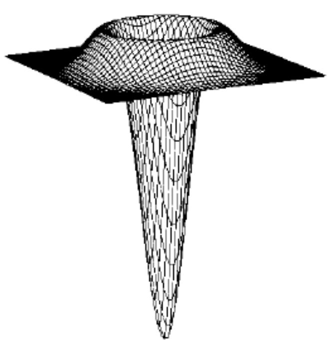
Slide adapted from Krystian Mikolajczyk

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

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



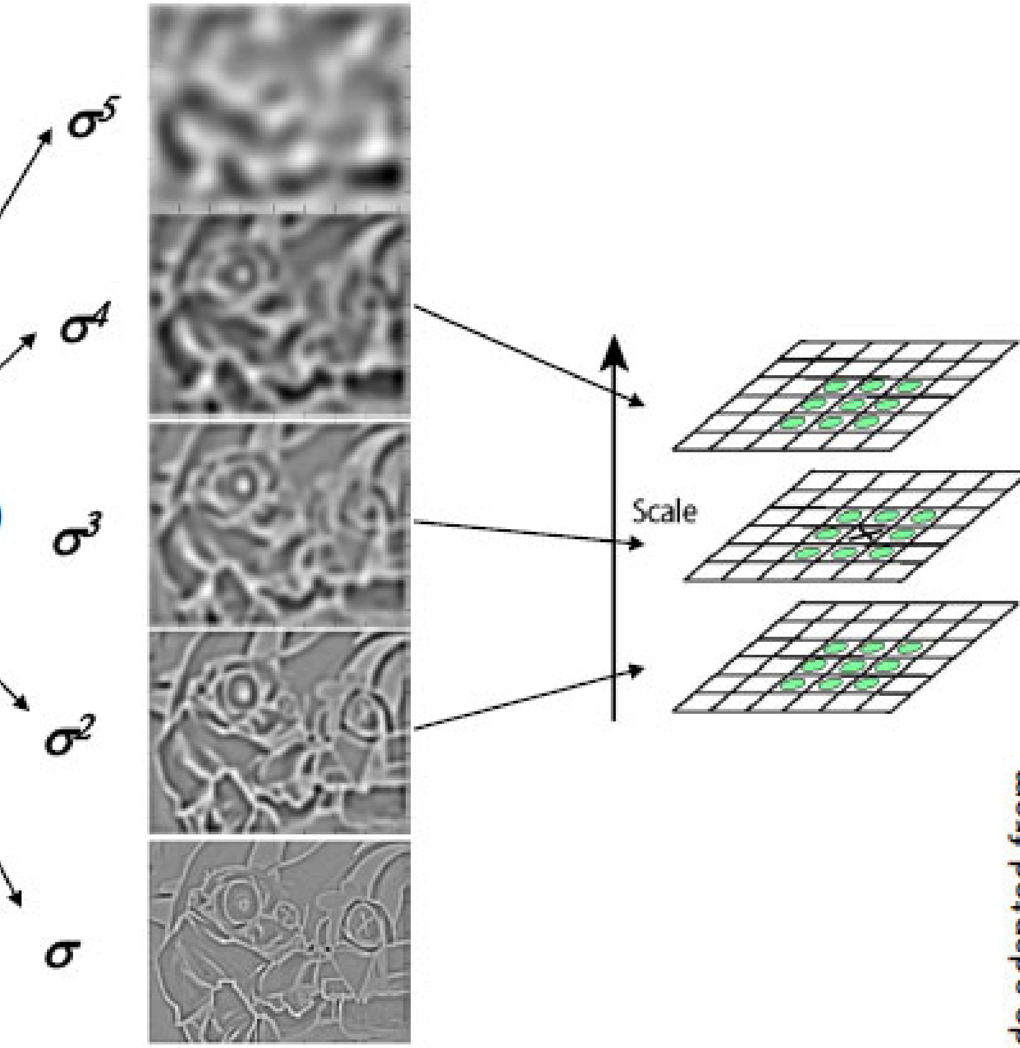


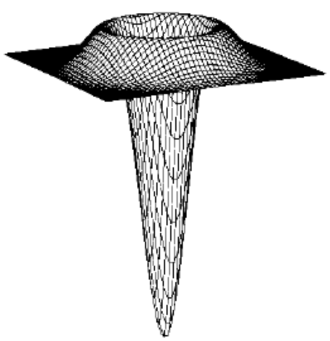
Slide adapted from

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

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



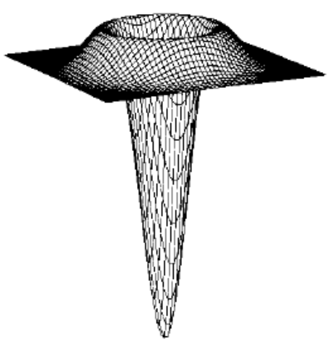


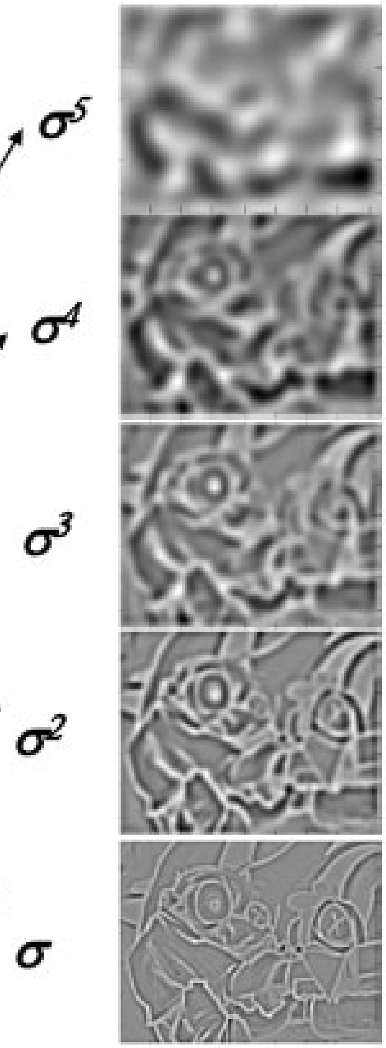
Slide adapted from

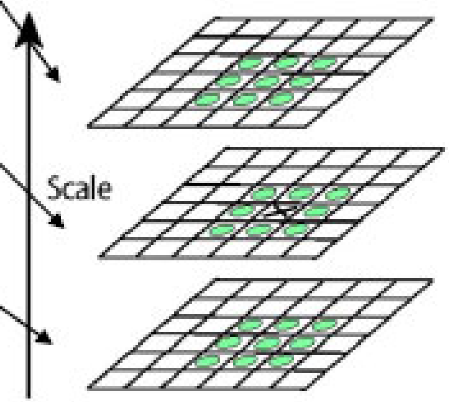
Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

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







⇒ List of *(x, y, σ)*

Slide adapted from

KAIST

LoG Detector: Workflow

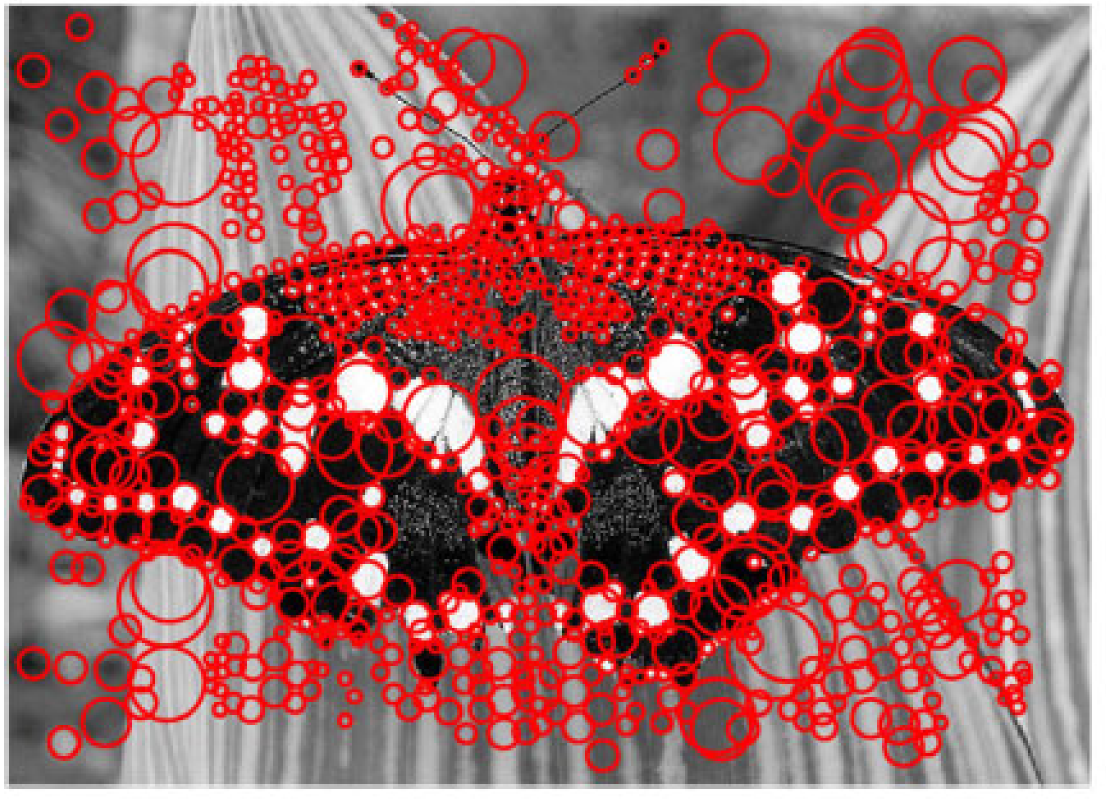
Slide credit: Svetlana Lazebnik

LoG Detector: Workflow

sigma = 11.9912

Slide credit: Svetlana Lazebnik

LoG Detector: Workflow



Slide credit: Svetlana Lazebnik

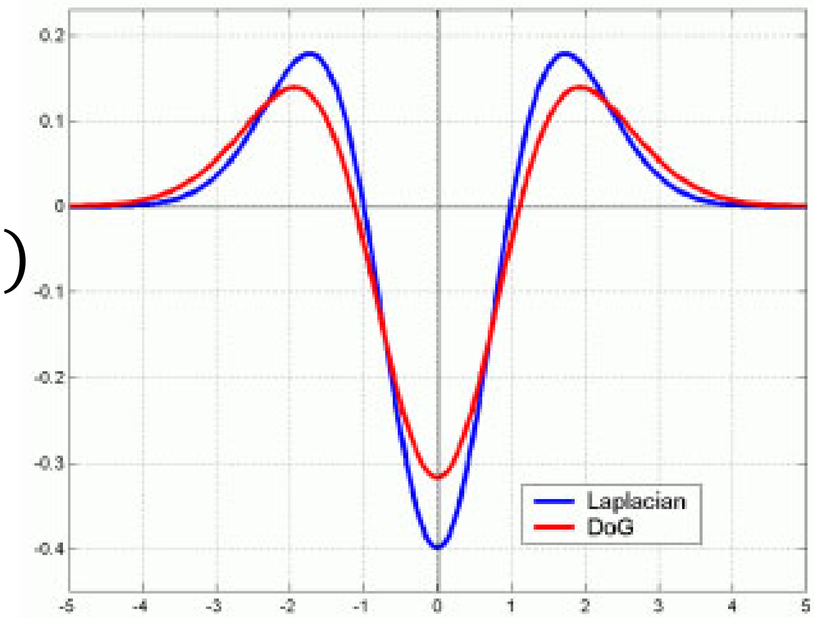
KAIST

Approximating LoG

- **Efficiently approximate LoG with a Difference of Gaussian (DoG)**

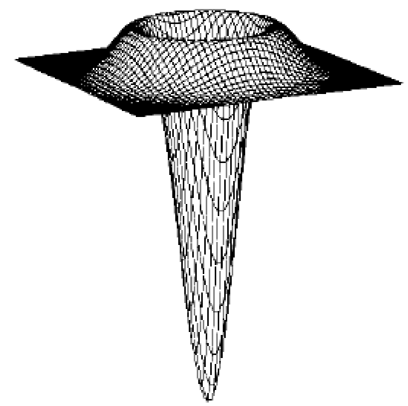
$$LoG = \sigma^2 (L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma))$$

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



Difference-of-Gaussian (DoG)

- Difference of Gaussians as approximation of the LoG
  - This is used e.g. in Lowe's SIFT pipeline for feature detection.

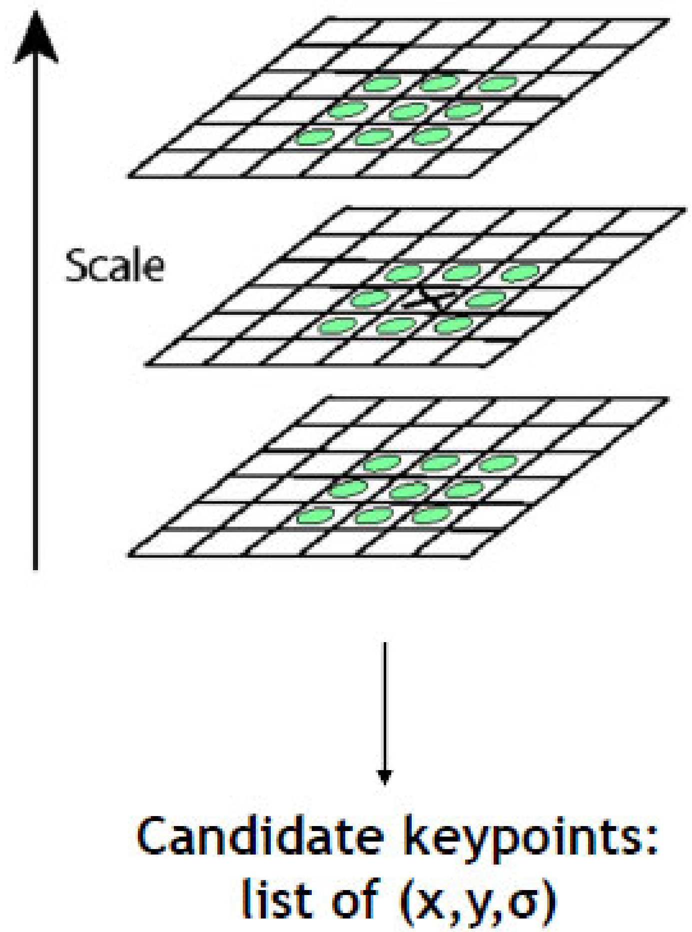


- Advantages
  - No need to compute 2nd derivatives
  - Gaussians are computed anyway, e.g. in a Gaussian pyramid.

Slide credit: Bastian Leibe

Key point localization with DoG

- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses

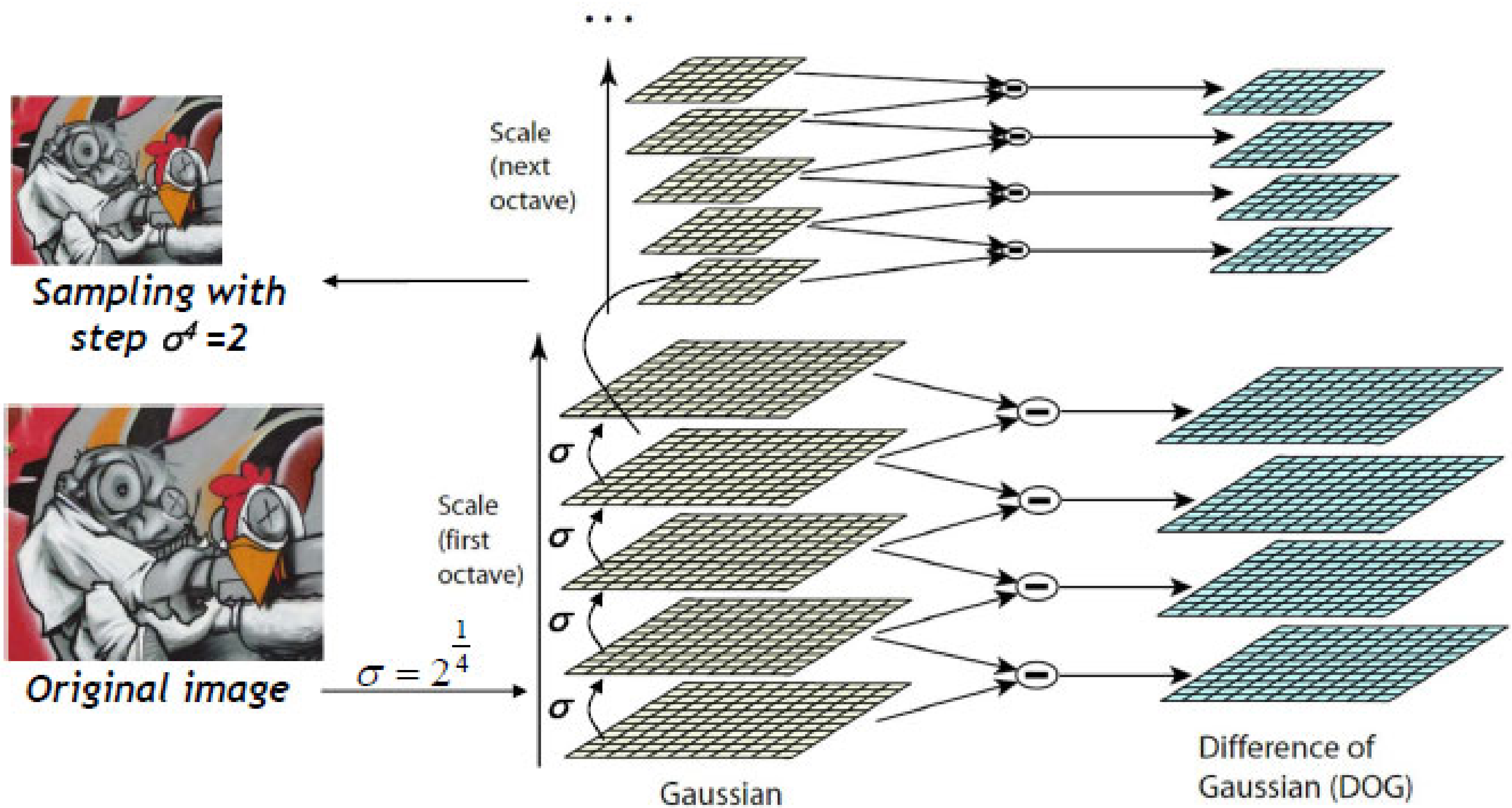


Slide credit: David Lowe

KAIST

DoG – Efficient Computation

- Computation in Gaussian scale pyramid

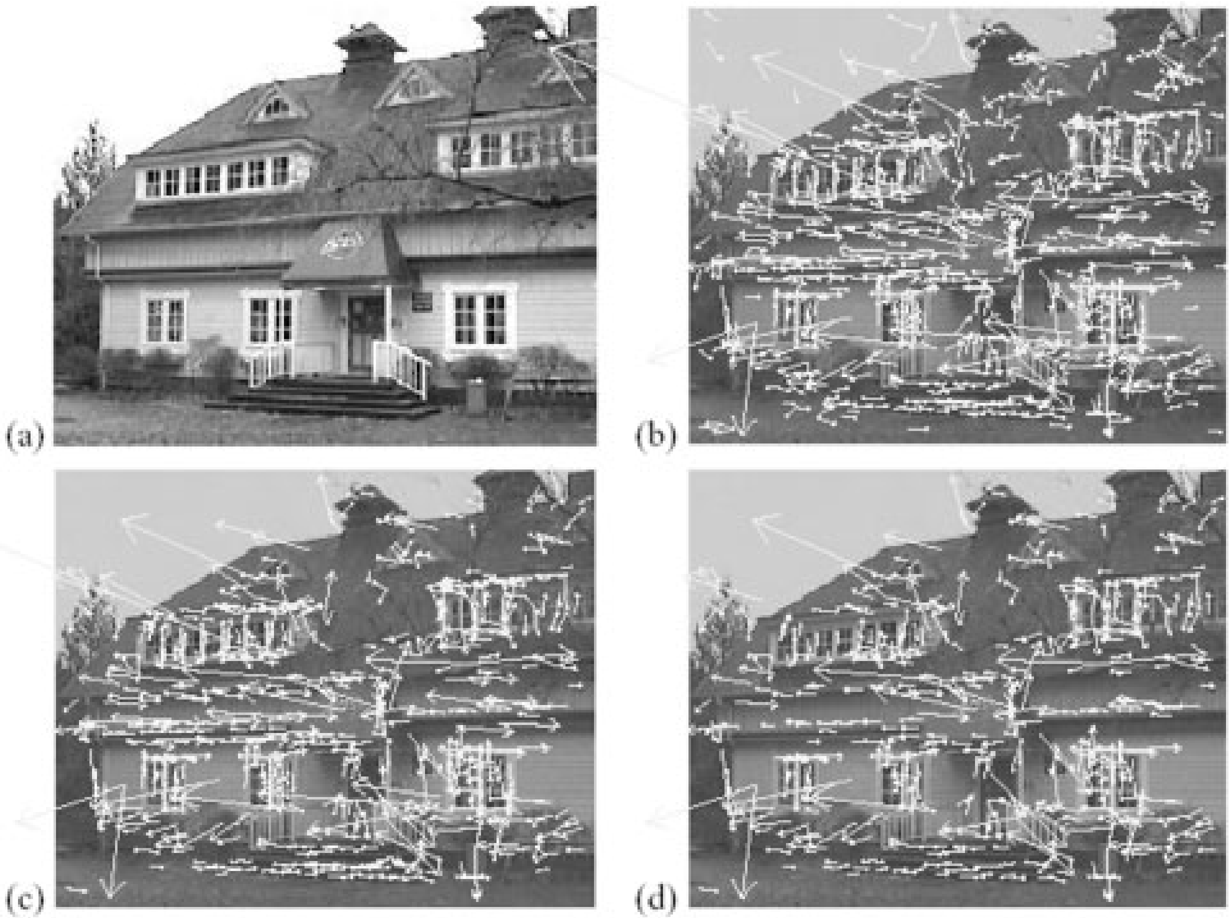


Slide adapted from Krystian Mikolajczyk

Results: Lowe's DoG

Slide credit: Bastian Leibe

Example of Keypoint Detection



(a) 233x189 image

(b) 832 DoG extrema

(c) 729 left after peak value threshold

(d) 536 left after testing ratio of principle curvatures (removing edge responses)

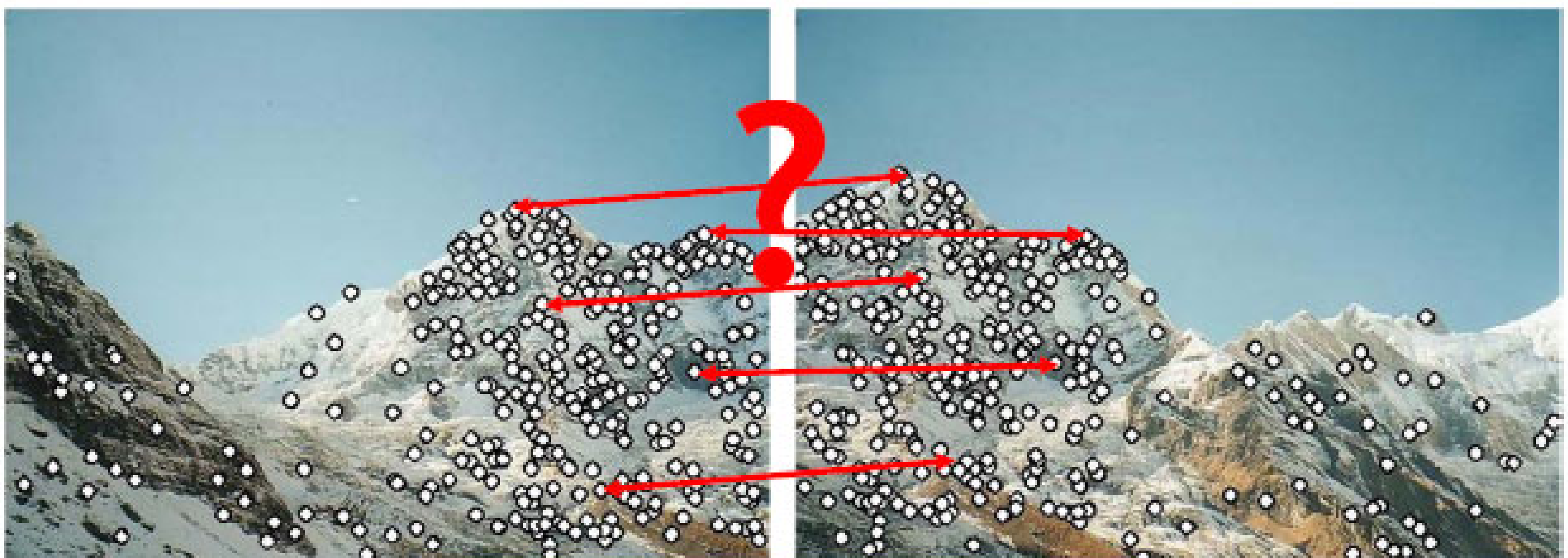
Slide credit: David Lowe

KAIST

Local Descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?



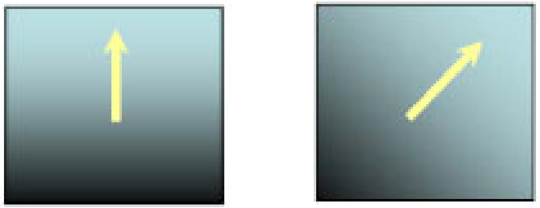
Point descriptor should be:
1. Invariant
2. Distinctive

Slide credit: Kristen Grauman

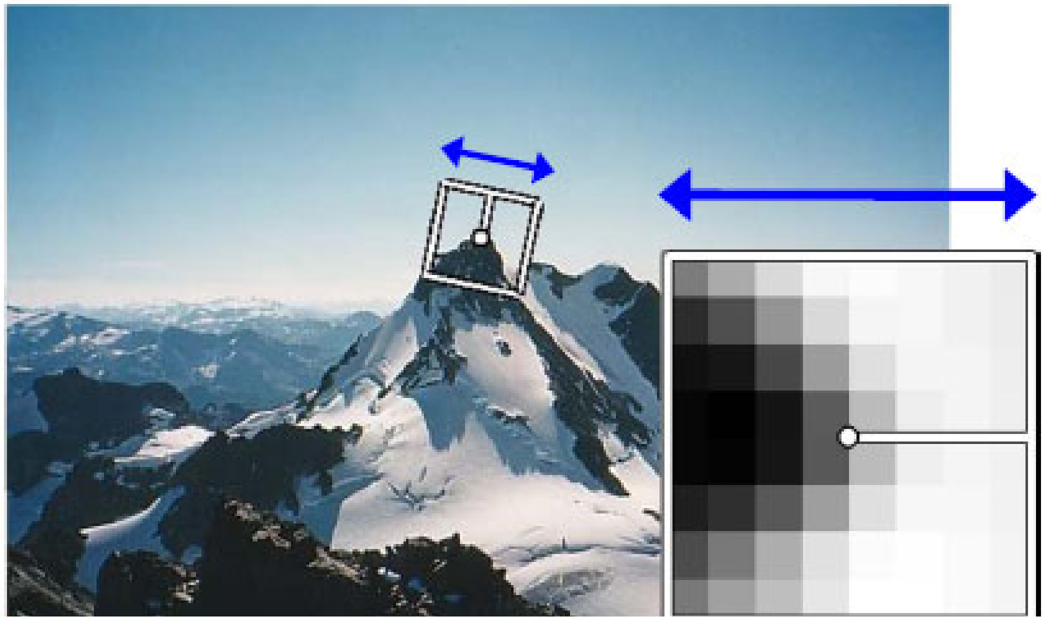
KAIST

Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch



- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.

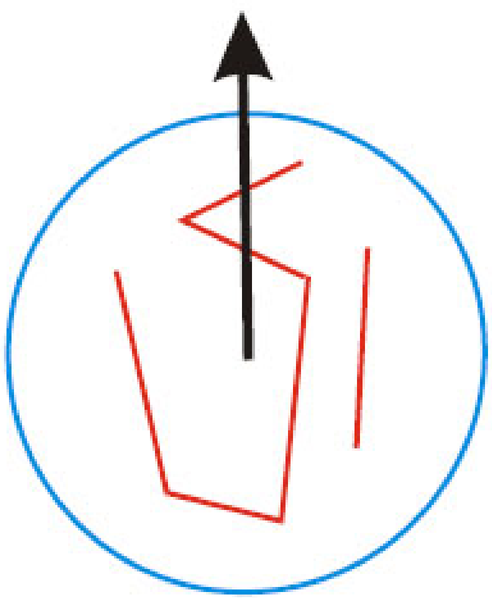


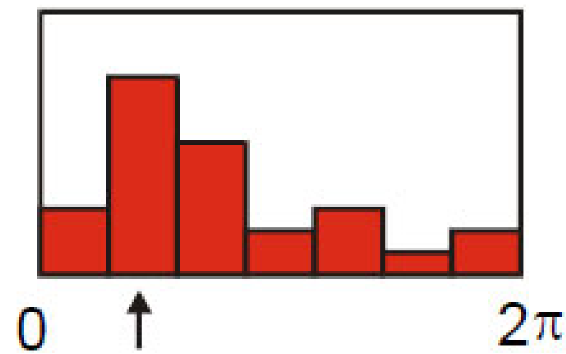
Slide credit: Svetlana Lazebnik, Matthew Brown

Orientation Normalization: Computation

[Lowe, SIFT, 1999]

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

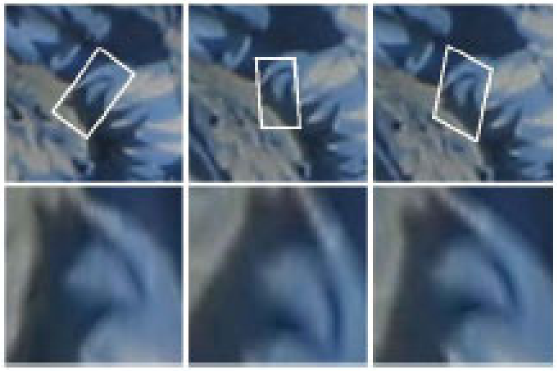




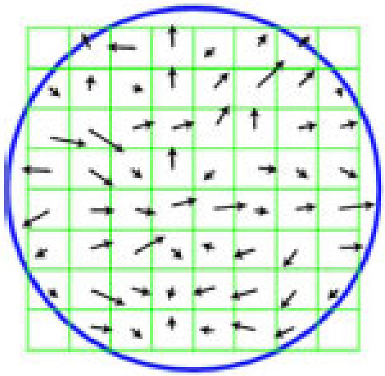
Slide adapted from David Lowe

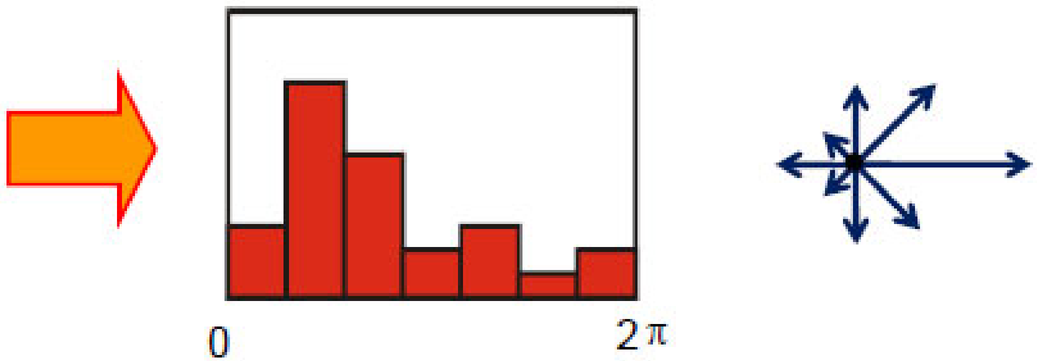
Feature Descriptors

- Disadvantage of patches as descriptors:
  - Small shifts can affect matching score a lot



- Solution: histograms

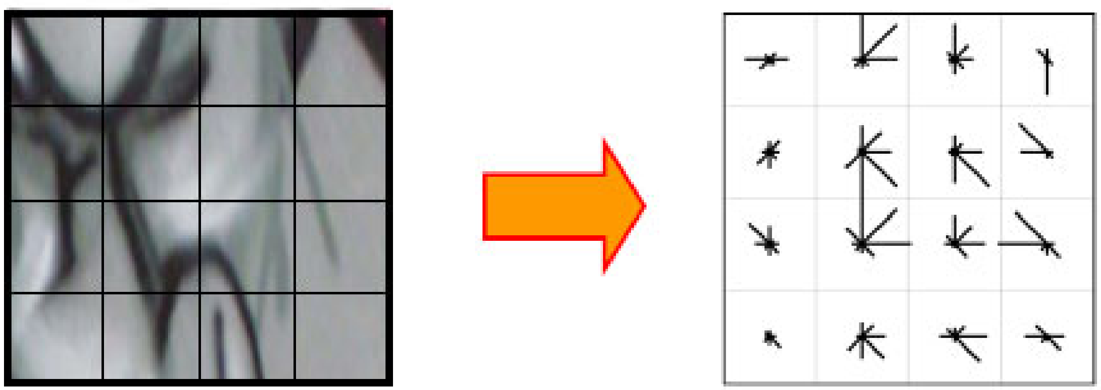




Slide credit: Svetlana Lazebnik

Feature Descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

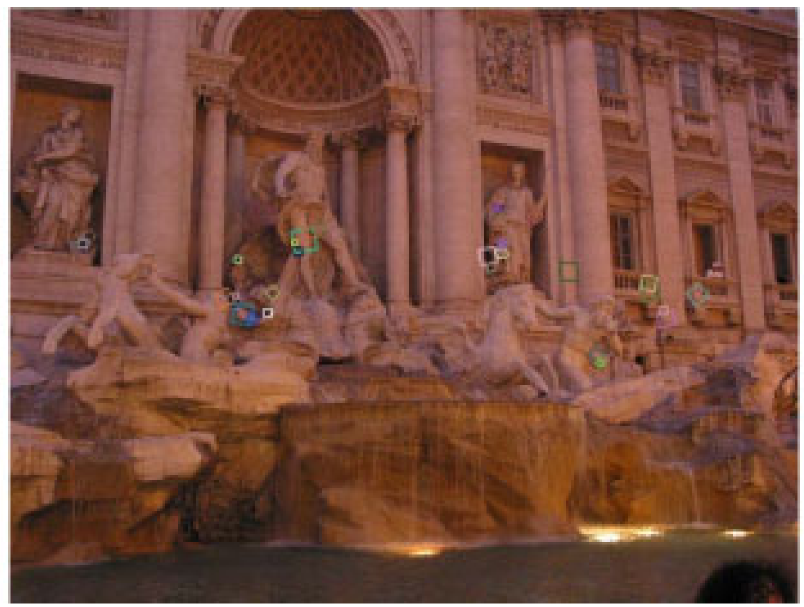


David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Svetlana Lazebnik

Overview: SIFT

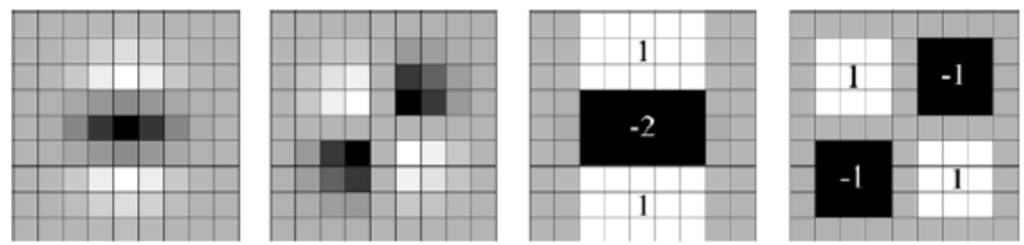
- Extraordinarily robust matching technique
  - Can handle changes in viewpoint up to ~60 deg. out-of-plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



Slide credit: Steve Seitz

Other Descriptors

- **GIST: a kind of SIFT in a global scale**
- **SURF: an acceleration using the integral image, i.e., summed area table**

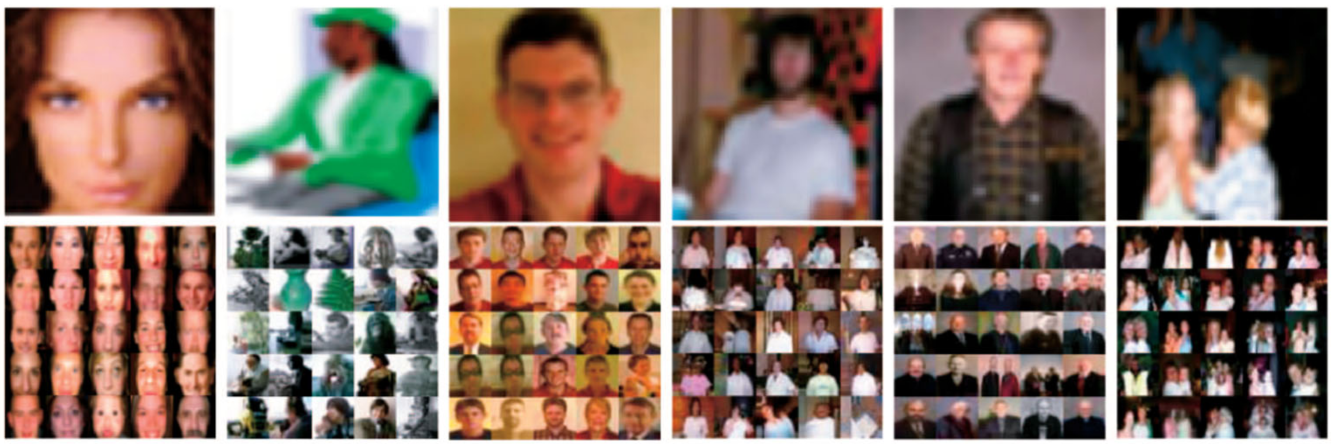


- **CNN features**

KAIST

80M Tiny Images

- Just use 32 by 32 images
- It works well even for recognition with a simple recognition method (nearest neighbor search) with using 80M data



- Indicates the importance of data

PA1 (Optional)

- **Objective**
  - **Understand how to extract SIFT features and to use related libraries (OpenCV, vlfeat, …)**

Class Objectives (Ch. 2.2 & 2.3) were:

- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gradients (LoG) $\approx$ Difference of Gradients (DoG)
  - SIFT as a local descriptor

Next Time…

- **Deep learning based image search**

Homework for Every Class

- Go over the next lecture slides
- Come up with one question on what we have discussed today
  - 1 for typical questions (that were answered in the class)
  - 2 for questions with thoughts or that surprised me
- Write questions 3 times before the mid-term exam
  - Write a question about one out of every four classes
  - Multiple questions in one time will be counted as one time
- Common questions are compiled at the Q&A file
  - Some of questions will be discussed in the class
- If you want to know the answer of your question, ask me or TA on person