Visual Object Recognition

Bastian Leibe & Kristen Grauman

Computer Vision Laboratory
ETH Zurich

Department of Computer Sciences
University of Texas in Austin

Chicago, 14.07.2008

ETH BIWI Computer Vision
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Computer Sciences

Outline

1. Detection with Global Appearance & Sliding Windows
2. Local Invariant Features: Detection & Description
3. Specific Object Recognition with Local Features

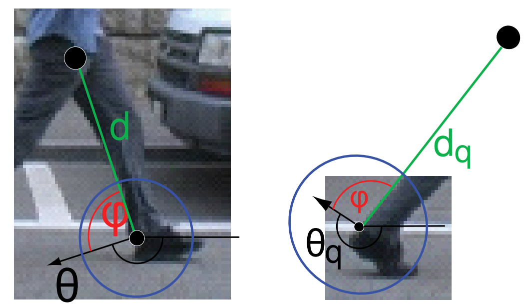
– *Coffee Break* –

4. Visual Words: Indexing, Bags of Words Categorization
5. Matching Local Features
6. Part-Based Models for Categorization
7. Current Challenges and Research Directions

Motivation

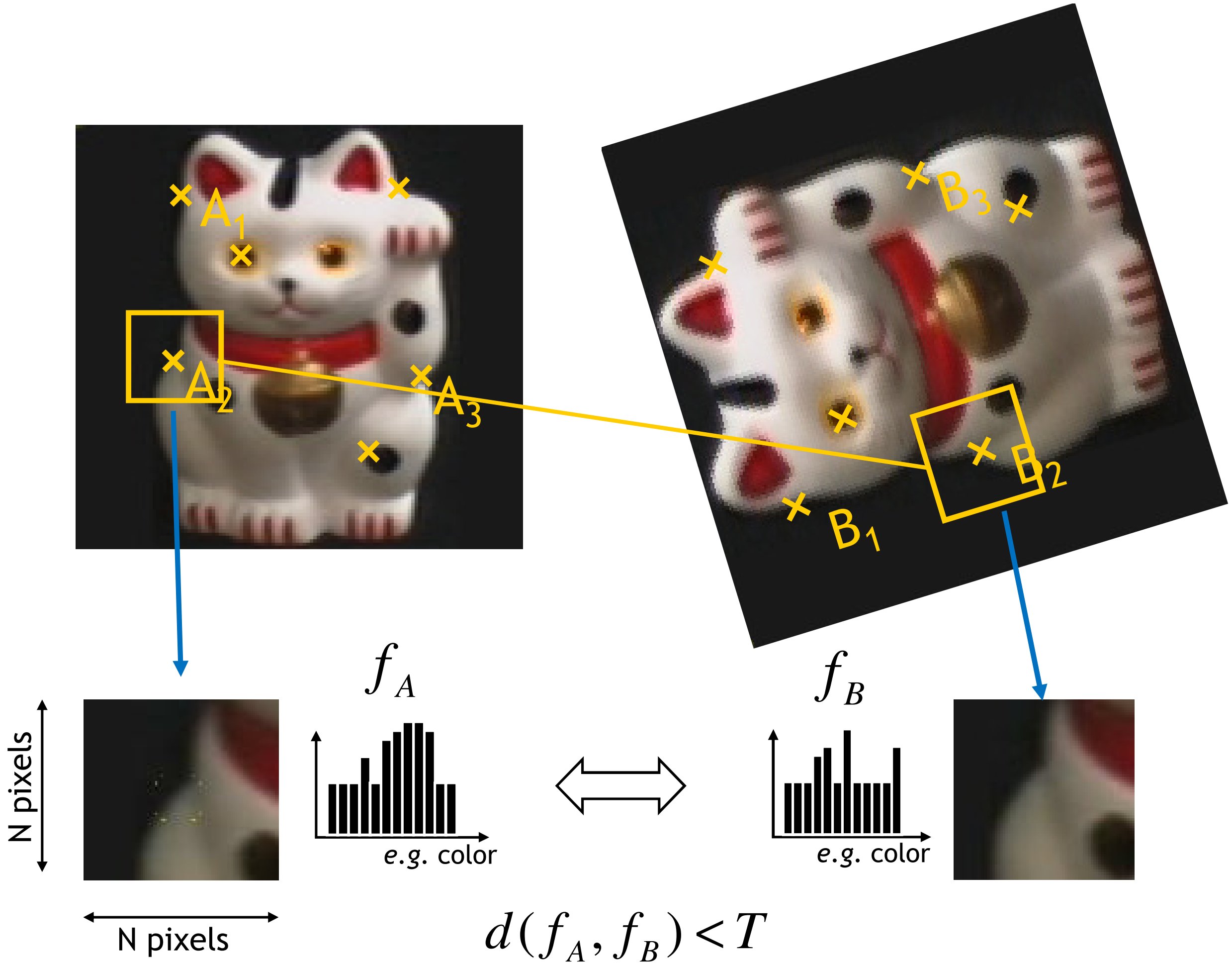
- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions

  - Articulation



  - Intra-category variations

Approach



$$d(f_A, f_B) < T$$

1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

K. Grauman, B. Leibe

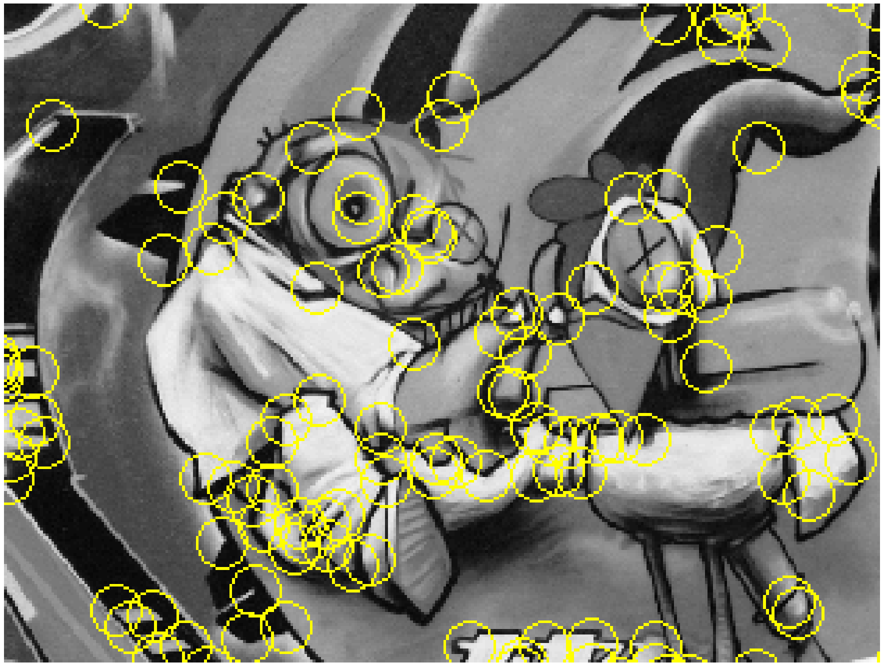
Requirements

- Region extraction needs to be repeatable and precise
  - Translation, rotation, scale changes
  - (Limited out-of-plane (≈affine) transformations)
  - Lighting variations
- We need a sufficient number of regions to cover the object
- The regions should contain "interesting" structure

Many Existing Detectors Available

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Lindeberg ‘98], [Lowe 1999]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR [Tuytelaars & Van Gool ‘04]
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- Others…

Keypoint Localization



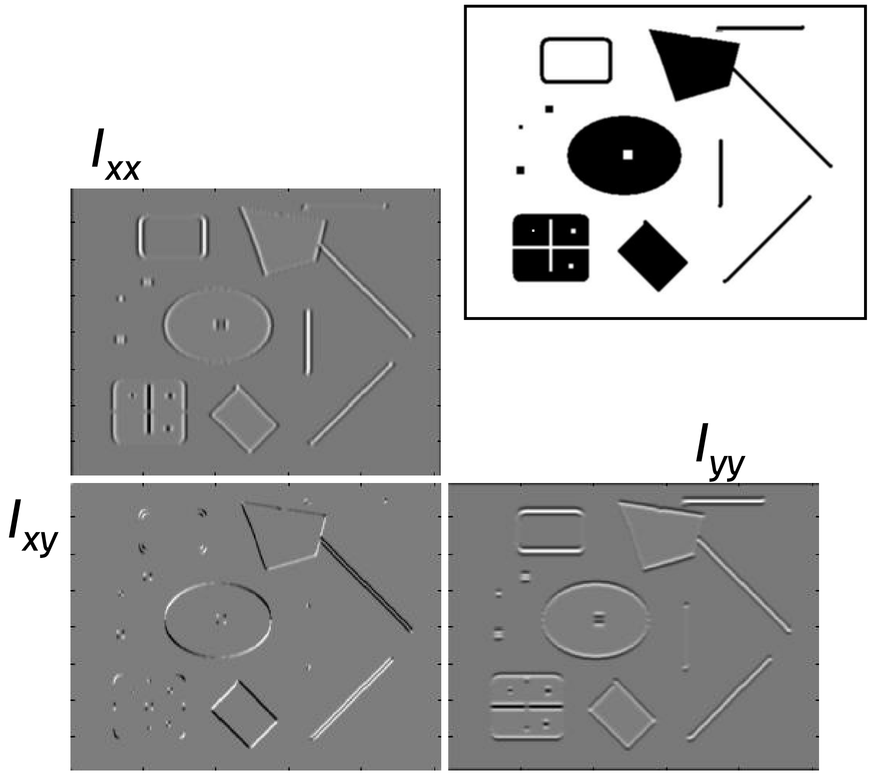
- **Goals:**
  - **Repeatable detection**
  - **Precise localization**
  - **Interesting content**

$\Rightarrow$ ***Look for two-dimensional signal changes***

Hessian Detector [Beaudet78]

- **Hessian determinant**

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

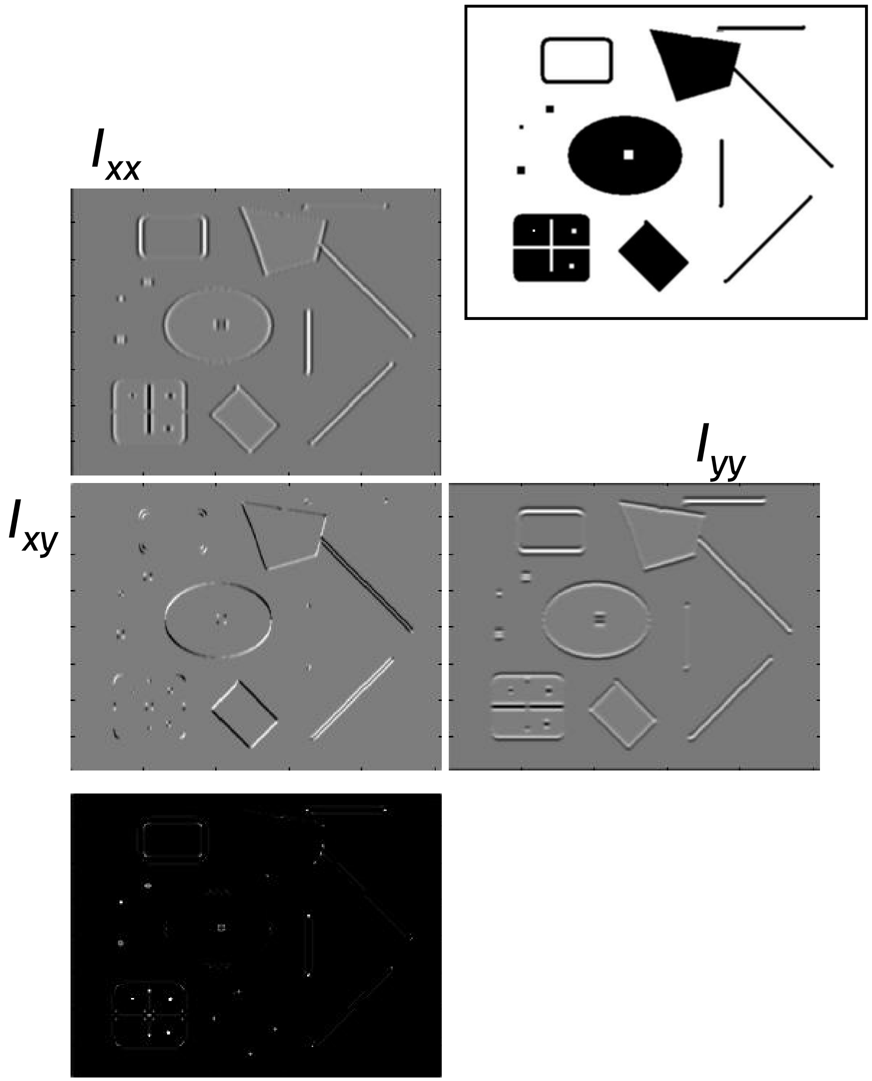


Intuition: **Search for strong derivatives in two orthogonal directions**

Hessian Detector [Beaudet78]

- **Hessian determinant**

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

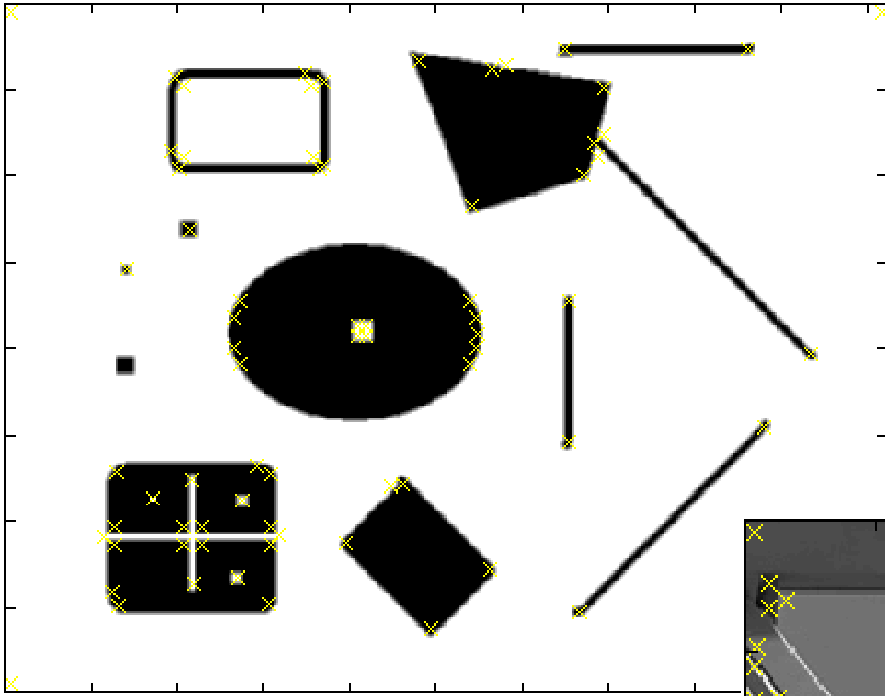


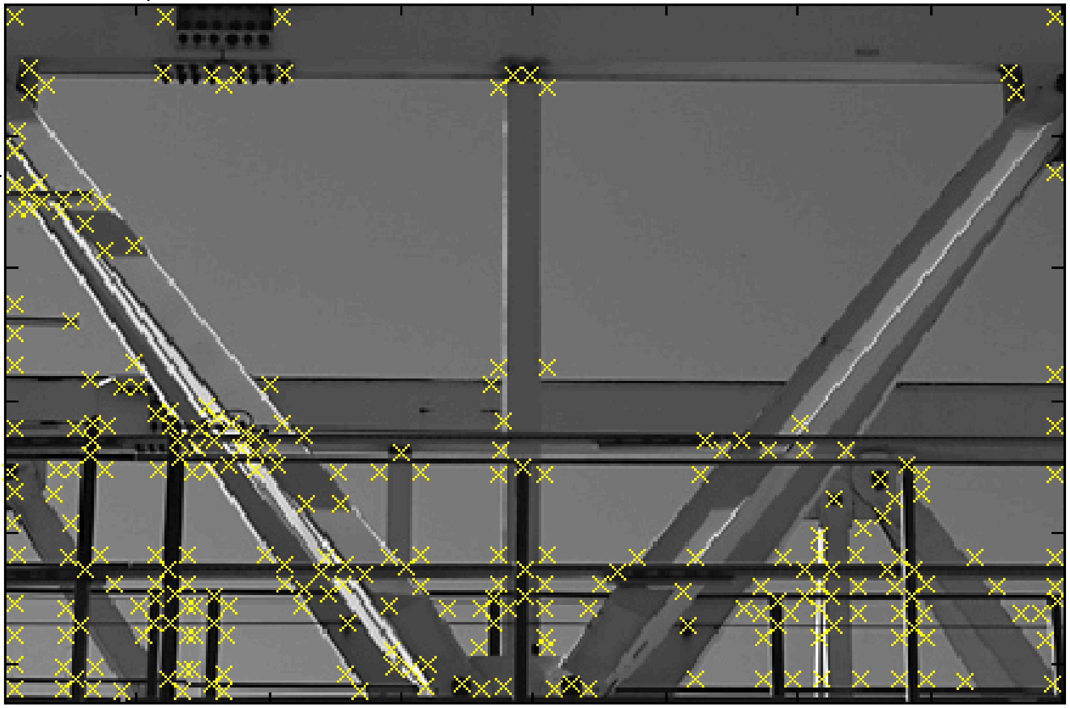
$$\det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$$

In Matlab:

$$I_{xx} .* I_{yy} - (I_{xy})\text{^}2$$

Hessian Detector – Responses [Beaudet78]





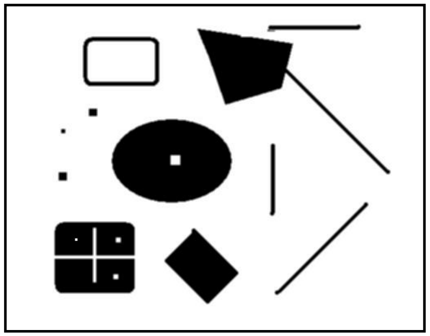
Effect: Responses mainly on corners and strongly textured areas.

Hessian Detector – Responses [Beaudet78]

Harris Detector [Harris88]

- **Second moment matrix (autocorrelation matrix)**

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

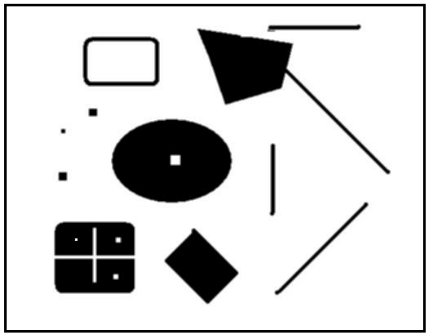


***Intuition:* Search for local neighborhoods where the image content has two main directions (eigenvectors).**

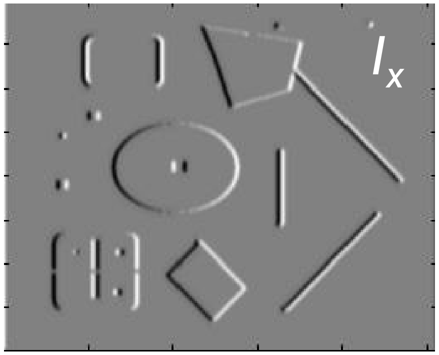
Harris Detector [Harris88]

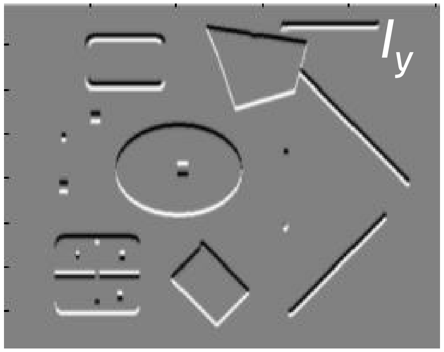
- **Second moment matrix (autocorrelation matrix)**

$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



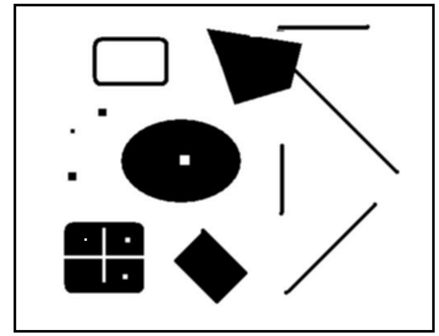
1. Image derivatives
   $g_x(\sigma_D)$, $g_y(\sigma_D)$,



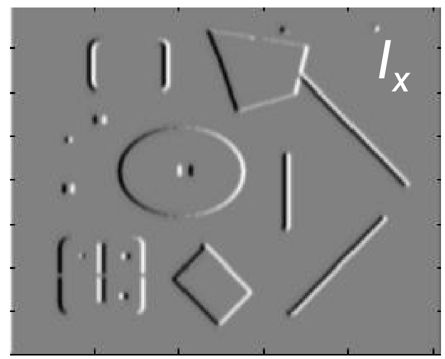


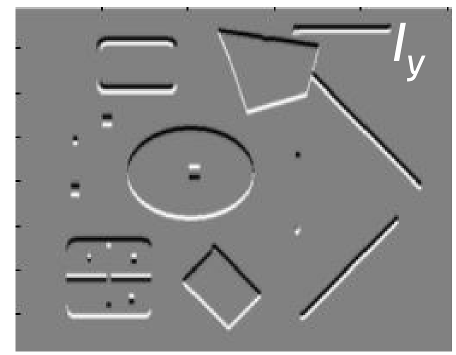
Harris Detector [Harris88]

- **Second moment matrix (autocorrelation matrix)**

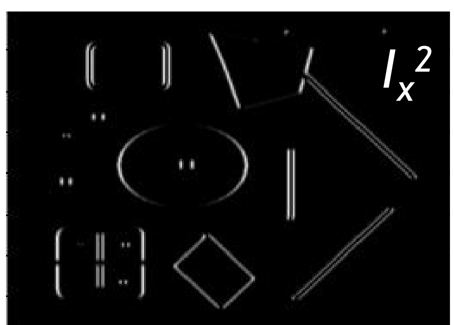


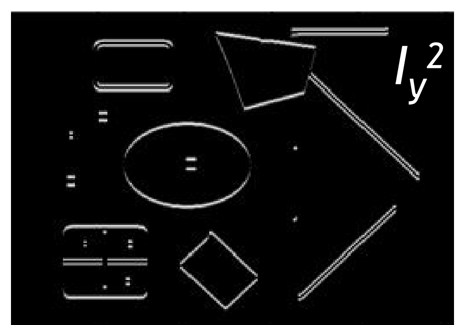
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

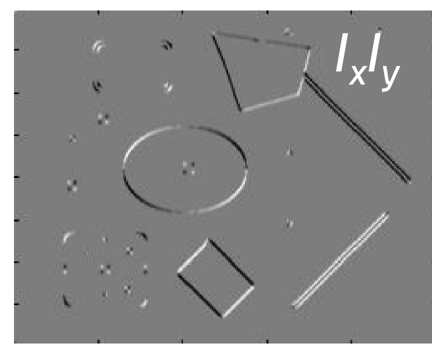




1. Image derivatives
   $g_x(\sigma_D)$, $g_y(\sigma_D)$,





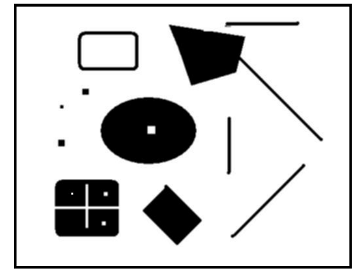


2. Square of derivatives

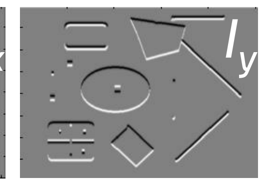
Harris Detector [Harris88]

- **Second moment matrix (autocorrelation matrix)**

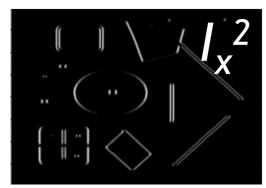
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

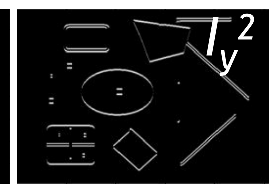


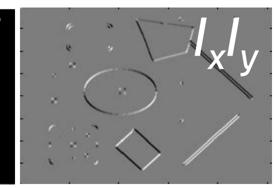
1. Image derivatives



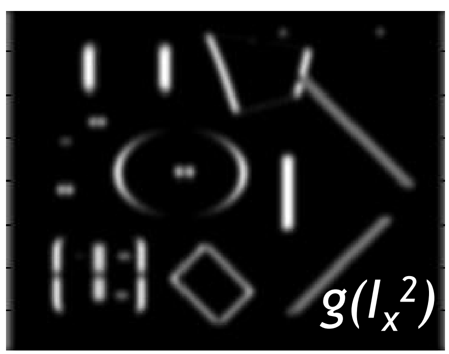
2. Square of derivatives

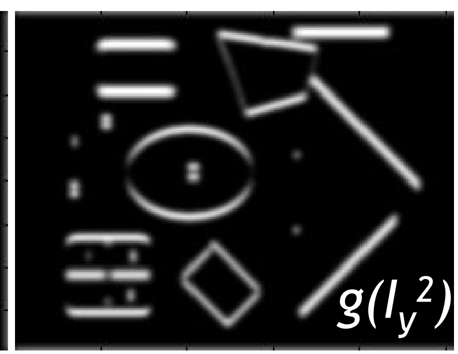


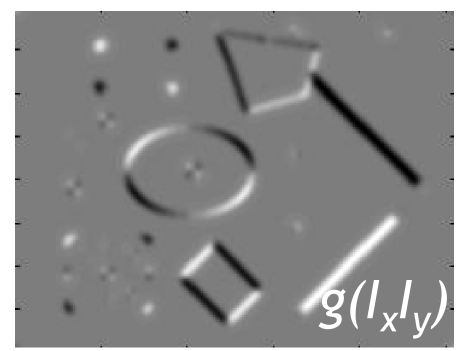




3. Gaussian filter $g(\sigma_I)$

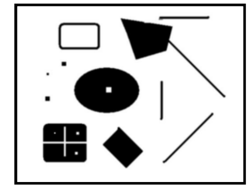






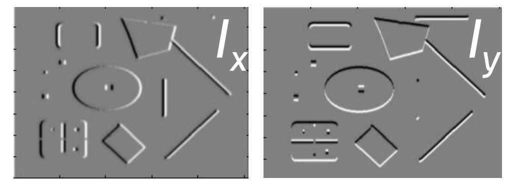
Harris Detector [Harris88]

- **Second moment matrix (autocorrelation matrix)**

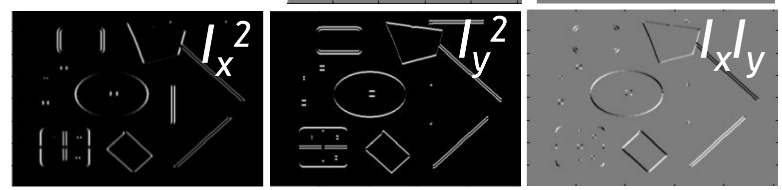


$$\mu(\sigma_I,\sigma_D)=g(\sigma_I)*\begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

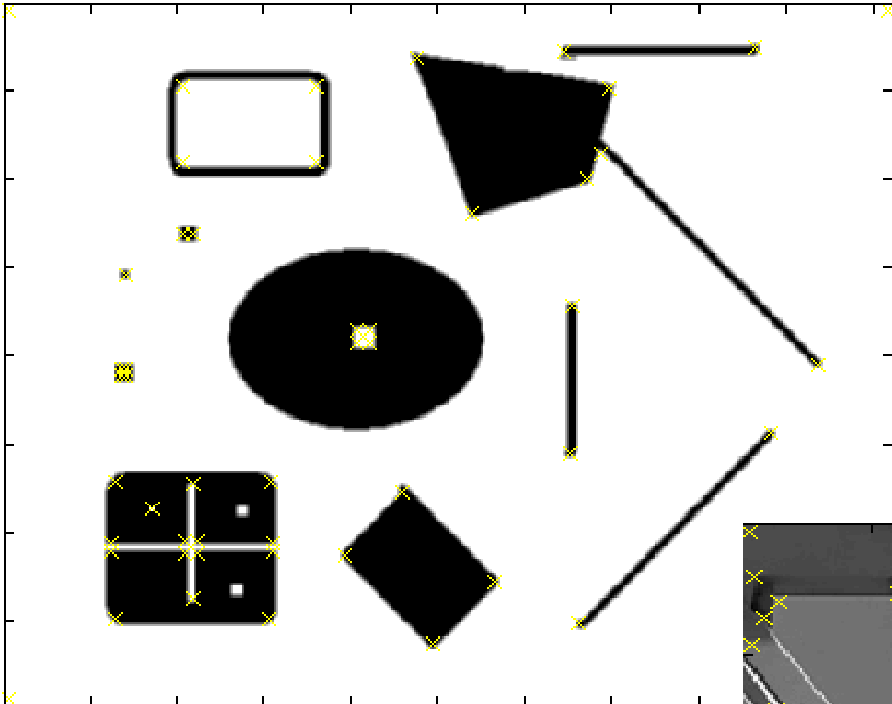
4. Corneress function – both eigenvalues are strong

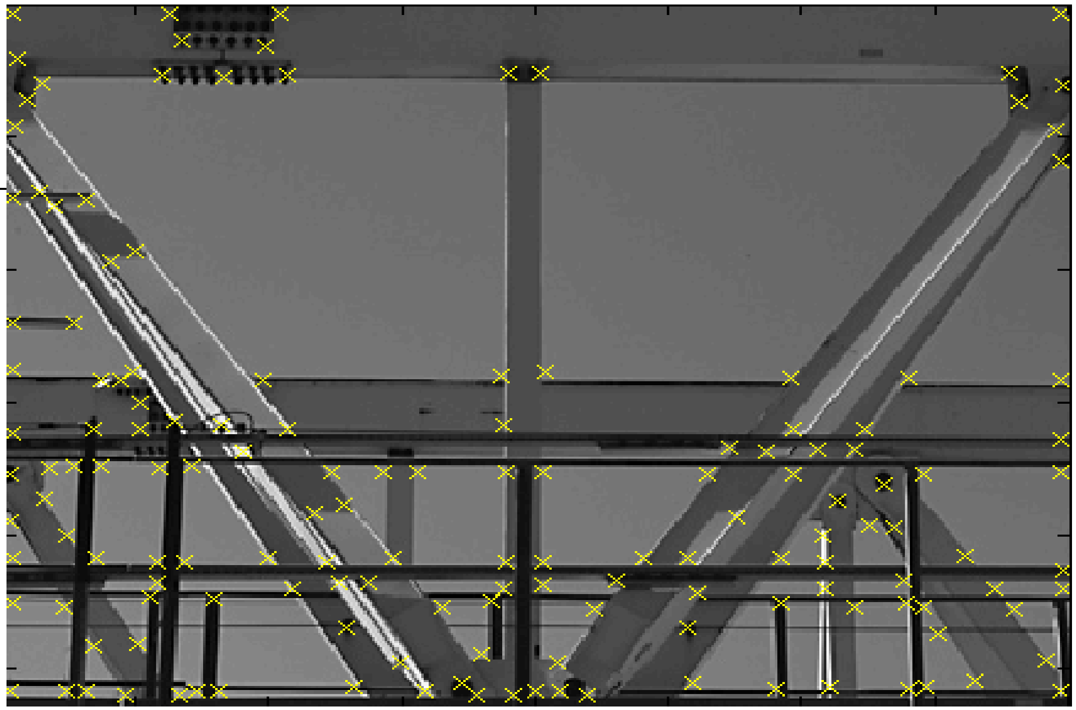
$$har=\det[\mu(\sigma_I,\sigma_D)]-\alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))]=$$

$$g(I_x^2)g(I_y^2)-[g(I_xI_y)]^2-\alpha[g(I_x^2)+g(I_y^2)]^2$$

5. Non-maxima suppression

Harris Detector – Responses [Harris88]





Effect: A very precise corner detector.

Harris Detector – Responses [Harris88]

Automatic Scale Selection

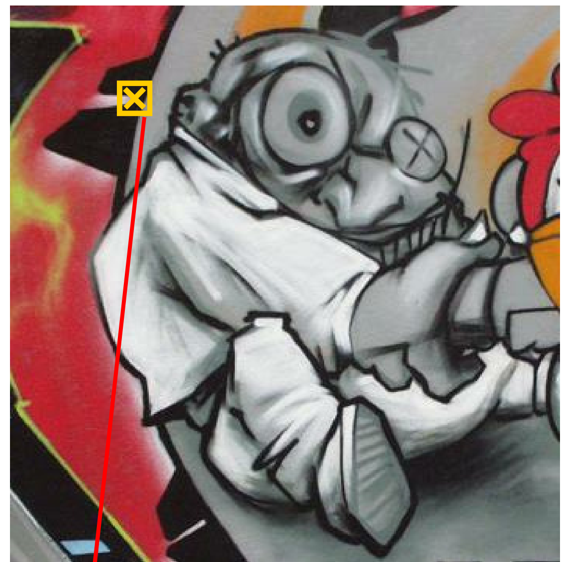
$$f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma'))$$

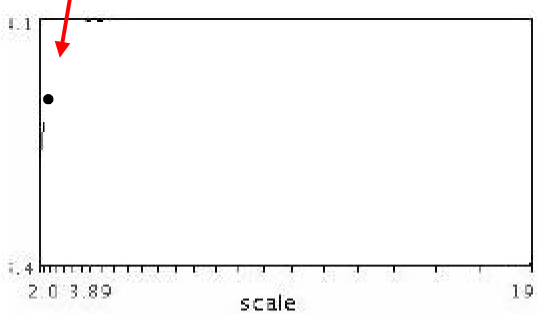
Same operator responses if the patch contains the same image up to scale factor

How to find corresponding patch sizes?

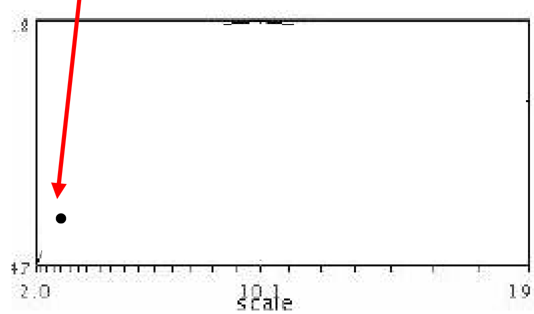
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





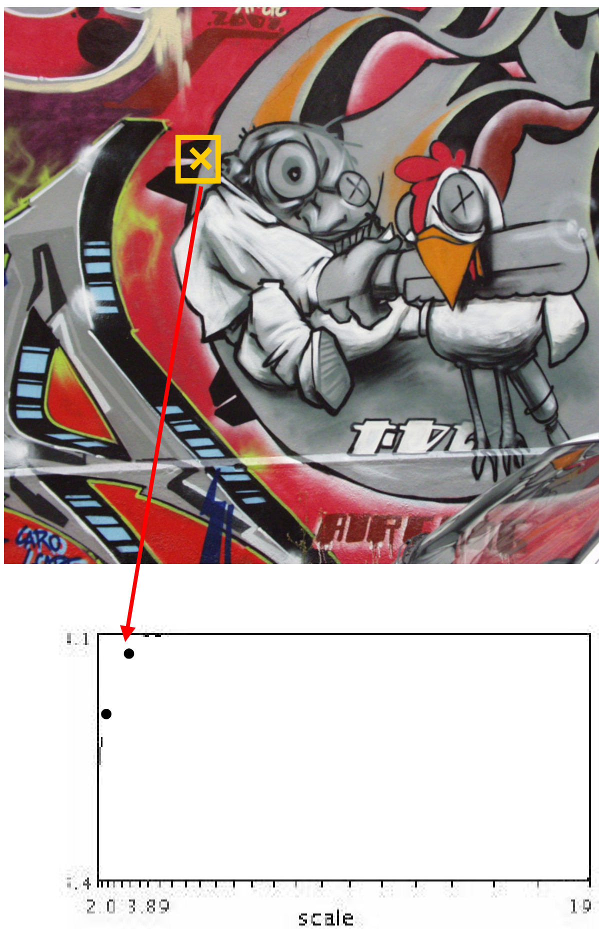
$f(I_{i_1 \ldots i_m}(x, \sigma))$



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

Automatic Scale Selection

- **Function responses for increasing scale (scale signature)**

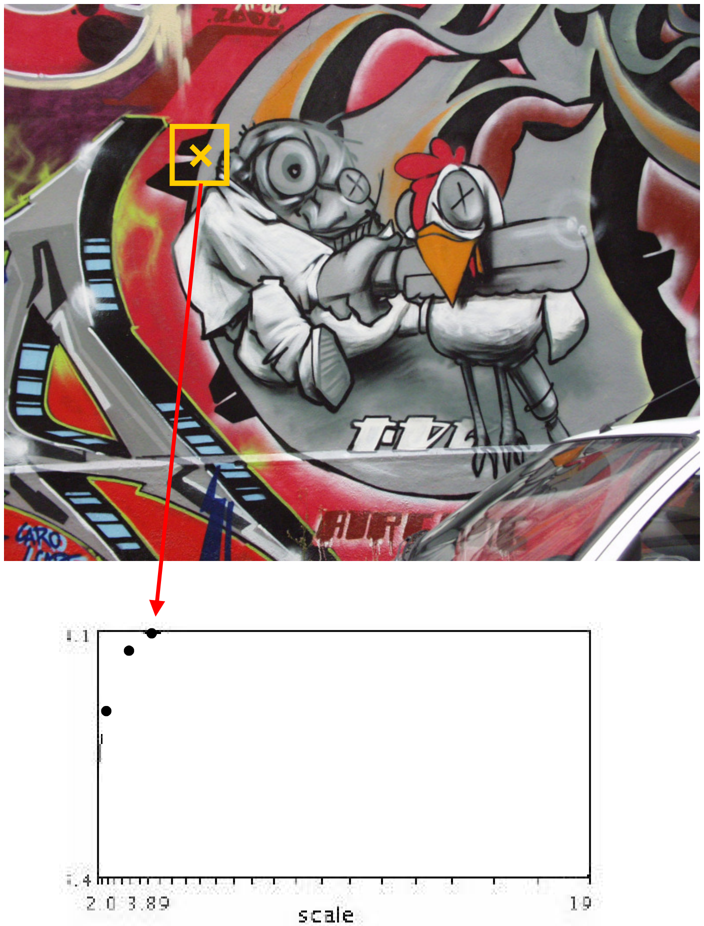


$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

Automatic Scale Selection

- **Function responses for increasing scale (scale signature)**

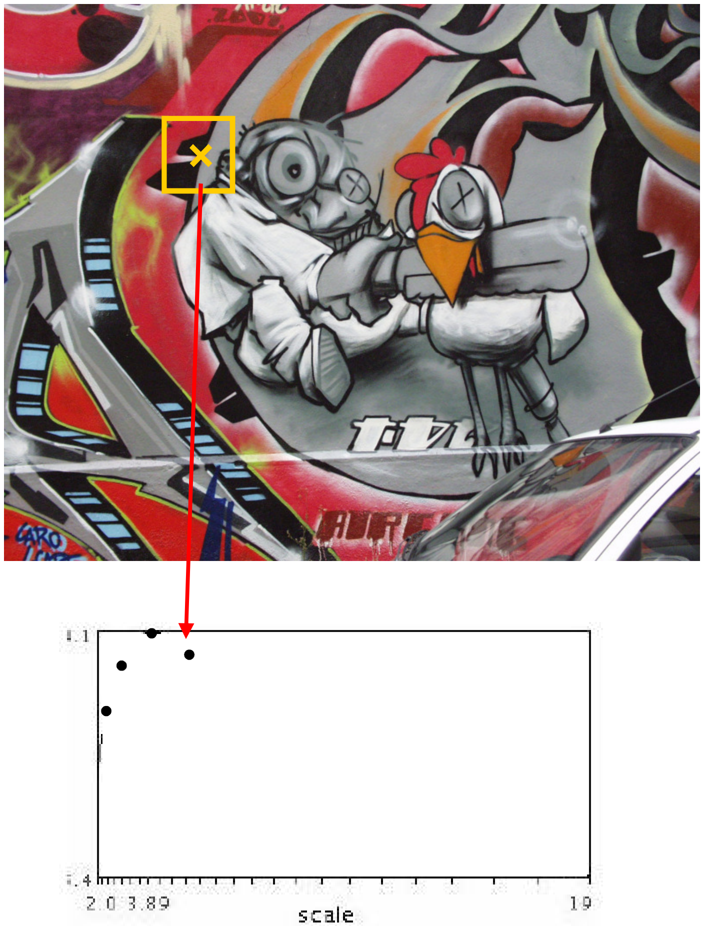


$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

Automatic Scale Selection

- **Function responses for increasing scale (scale signature)**

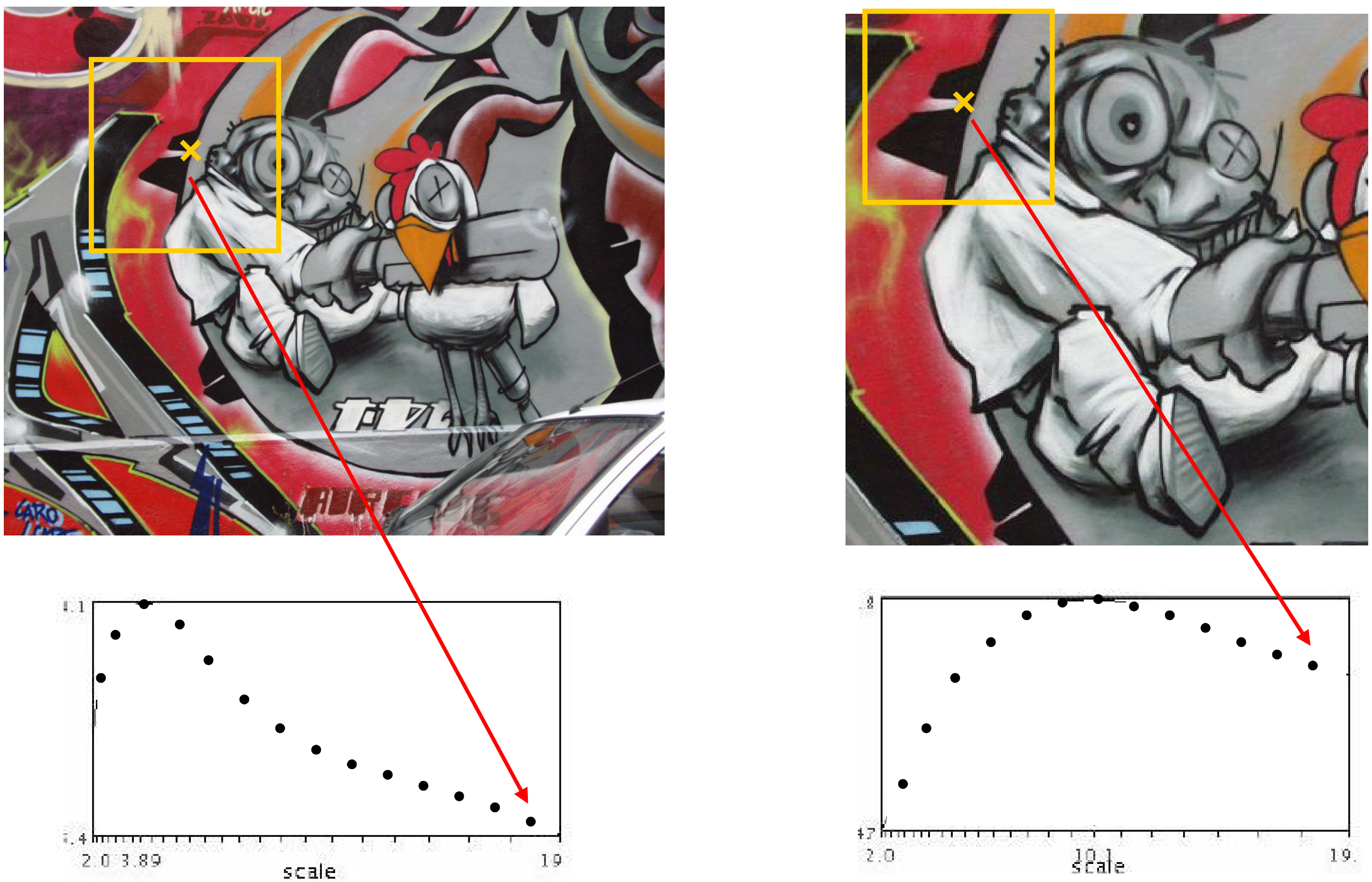


$f(I_{i_1 \dots i_m}(x, \sigma))$

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

Automatic Scale Selection

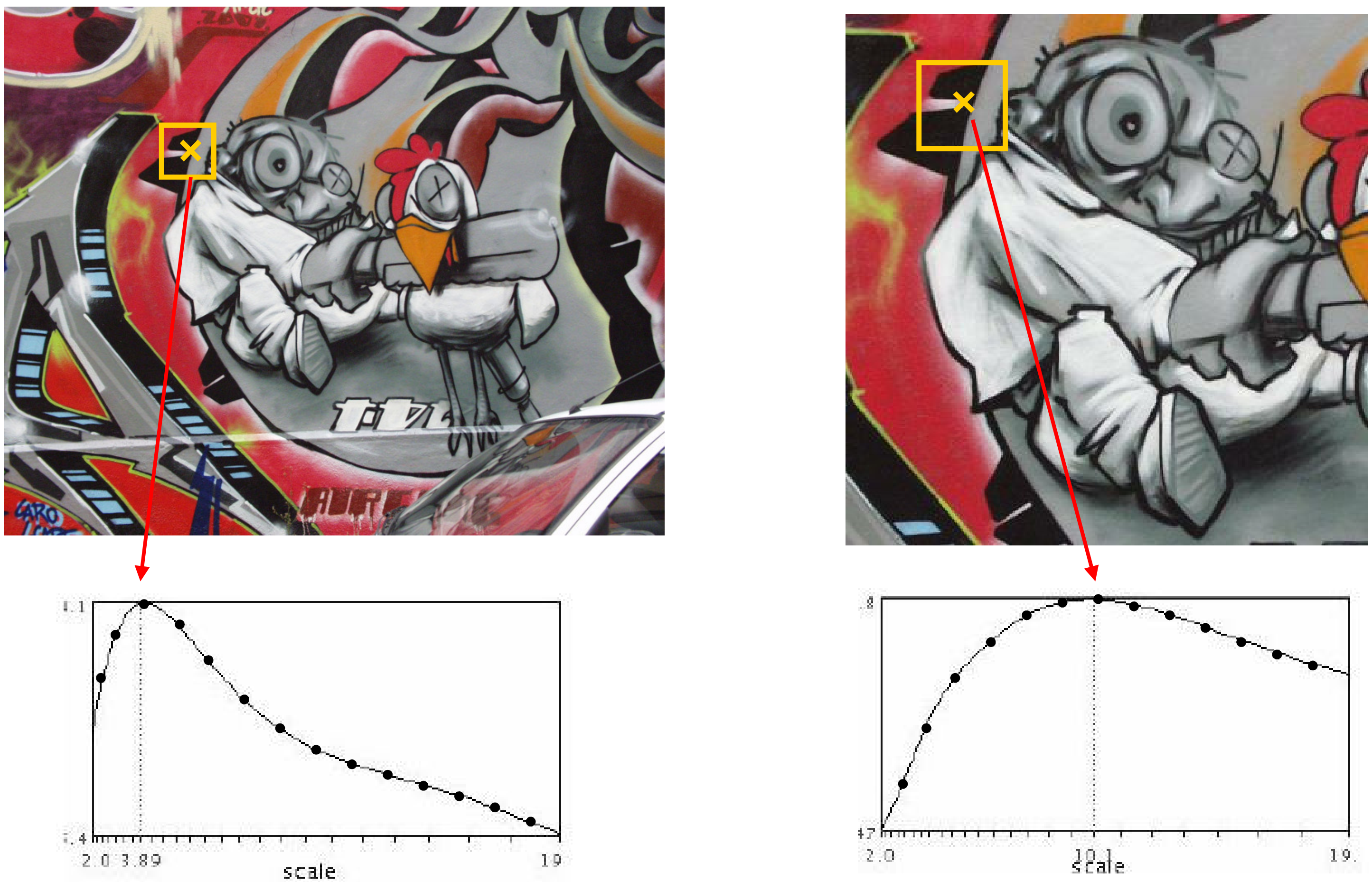
- **Function responses for increasing scale (scale signature)**



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

Automatic Scale Selection

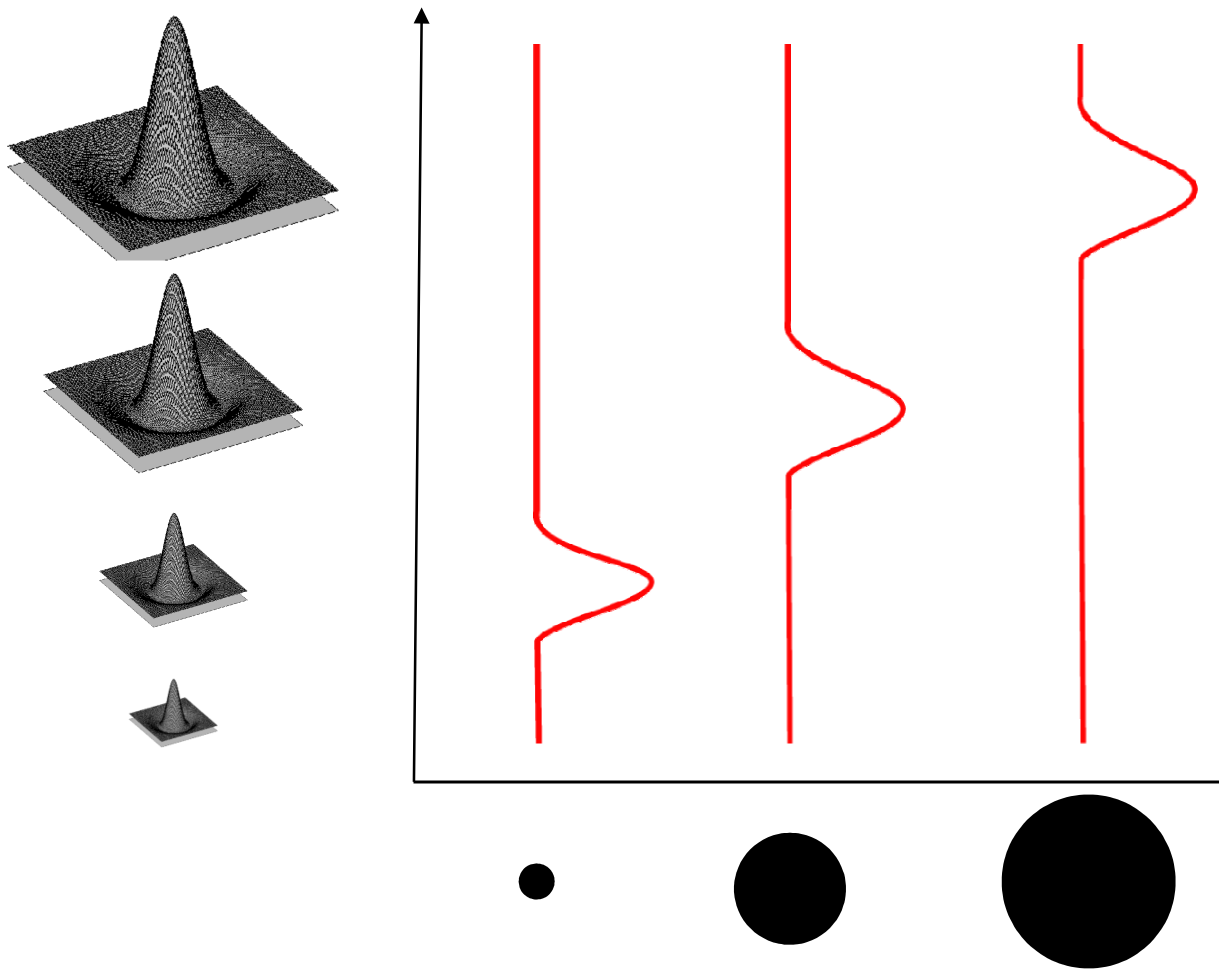
- **Function responses for increasing scale (scale signature)**



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

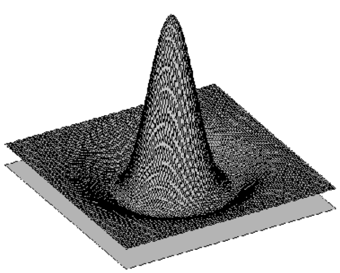
What Is A Useful Signature Function?

- **Laplacian-of-Gaussian = "blob" detector**



Laplacian-of-Gaussian (LoG)

- **Local maxima in scale space of Laplacian-of-Gaussian**



$$L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow \sigma^3$$

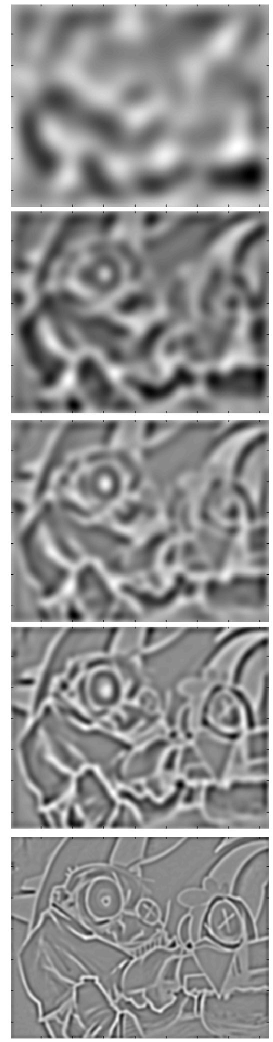
σ^5

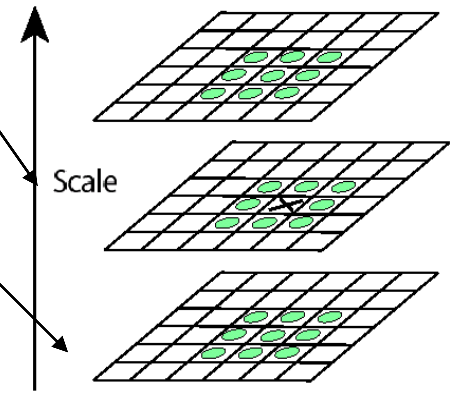
σ^4

σ^3

σ^2

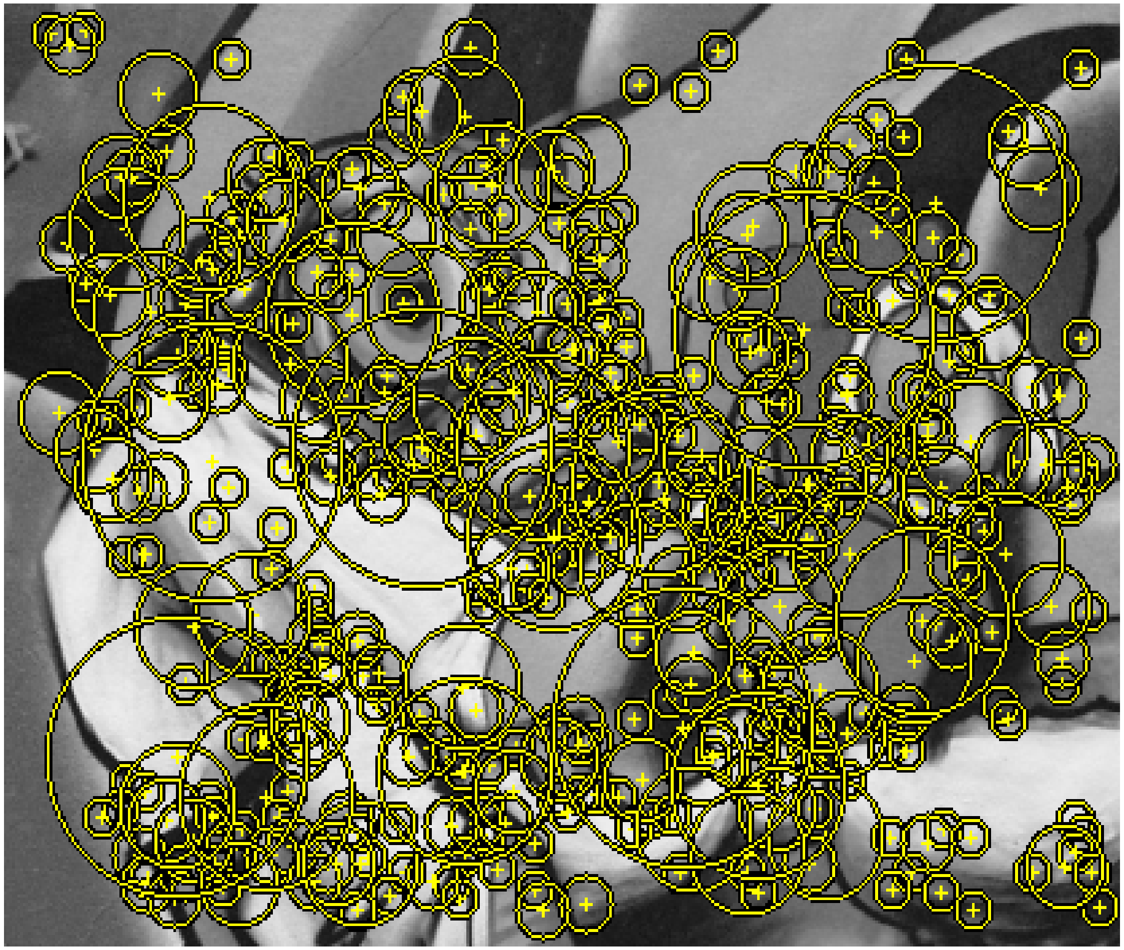
σ





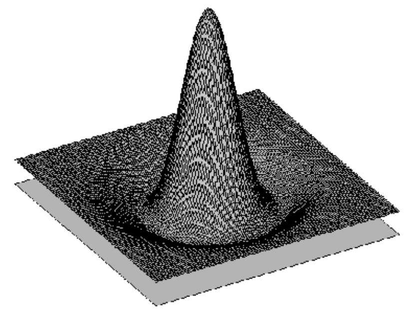
⇒ List of *(x, y, s)*

Results: Laplacian-of-Gaussian



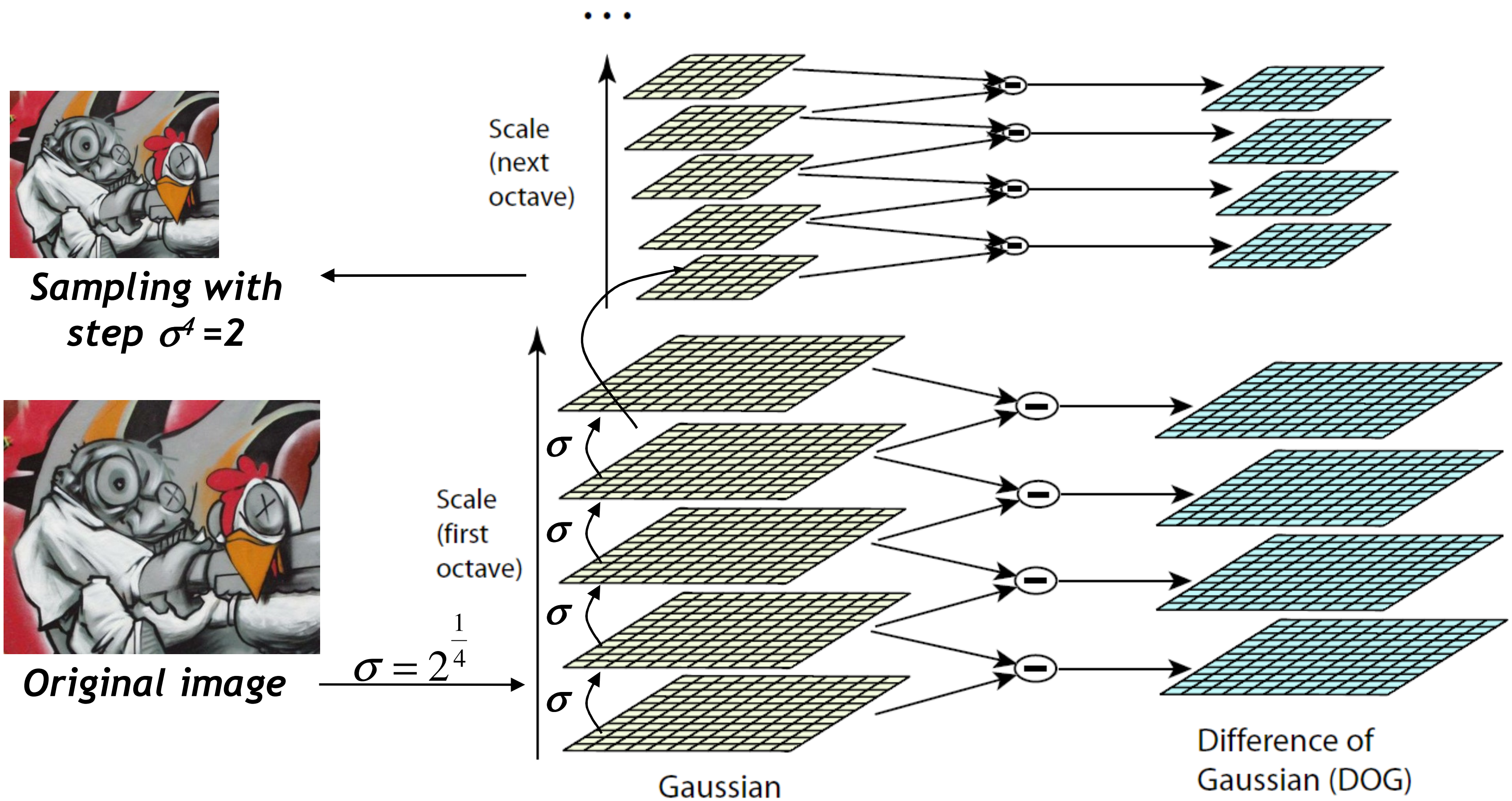
Difference-of-Gaussian (DoG)

- **Difference of Gaussians as approximation of the Laplacian-of-Gaussian**



DoG – Efficient Computation

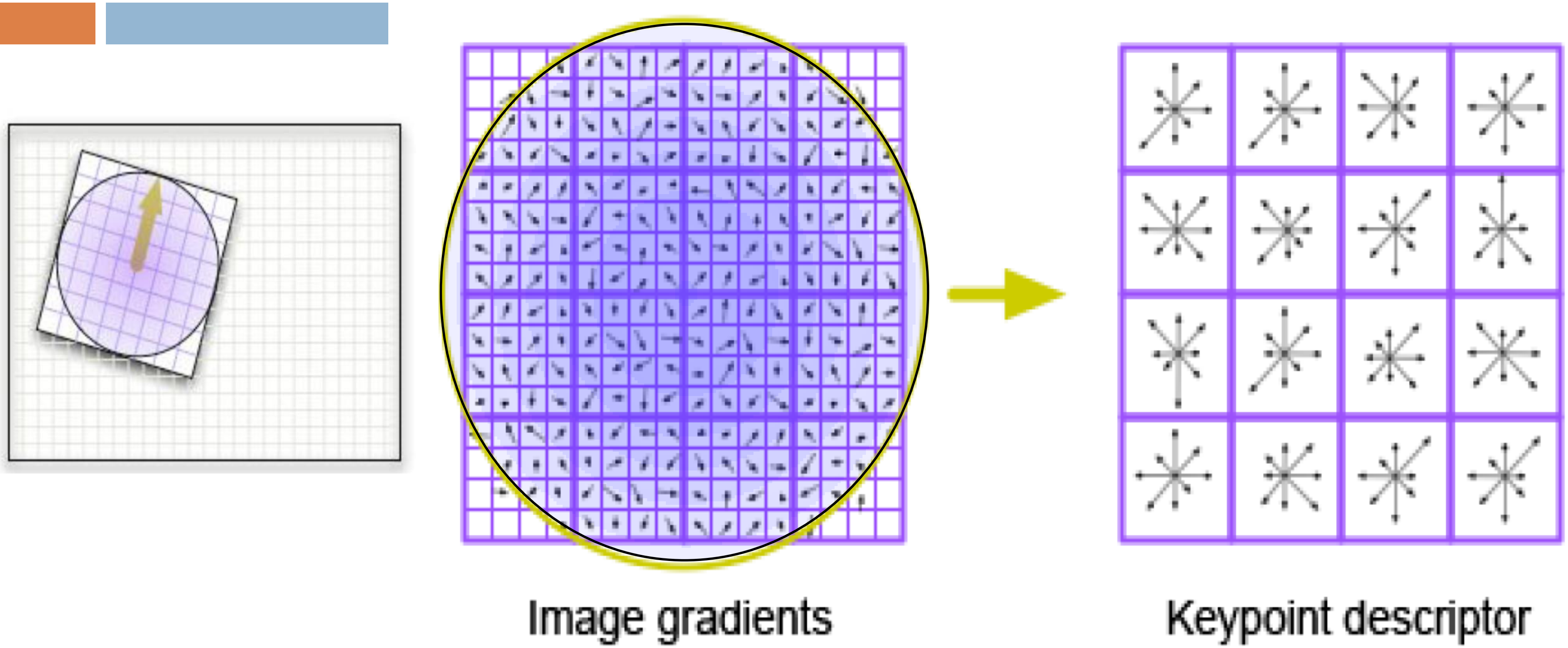
- Computation in Gaussian scale pyramid



K. Grauman, B. Leibe

Results: Lowe's DoG

Building a Descriptor



- Actual implementation uses 4x4 descriptors from 16x16 which leads to a 4x4x8=128 element vector