Locating and Describing Interest Points

Computer Vision

CS 543 / ECE 549

University of Illinois

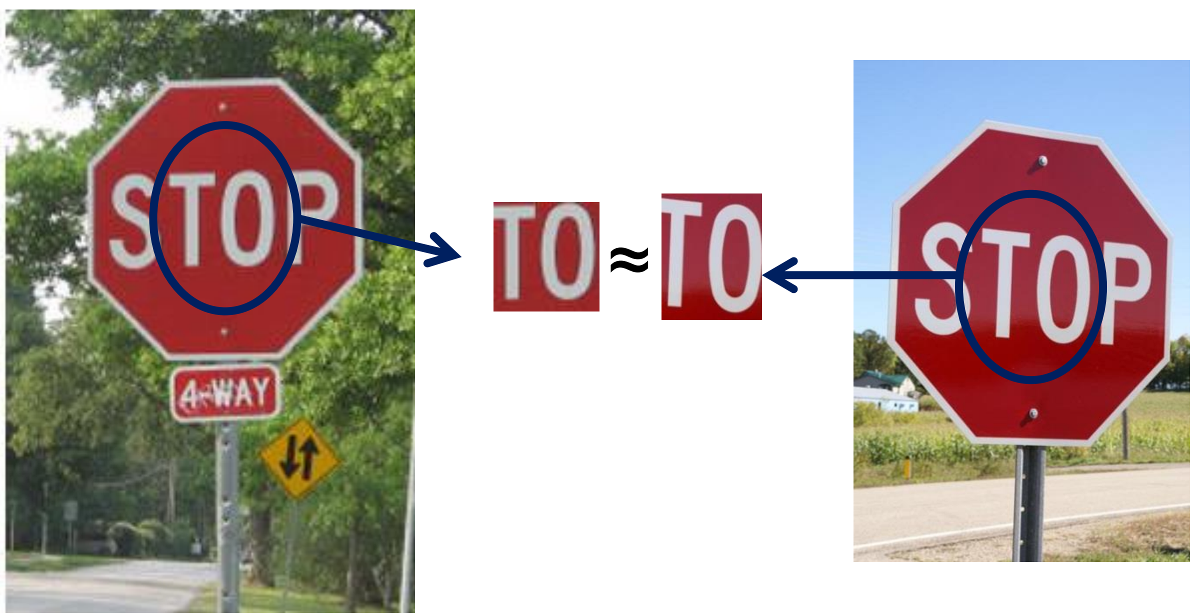
Derek Hoiem

Acknowledgment: Many keypoint slides from Grauman&Leibe 2008 AAAI Tutorial

Laplacian Pyramid clarification

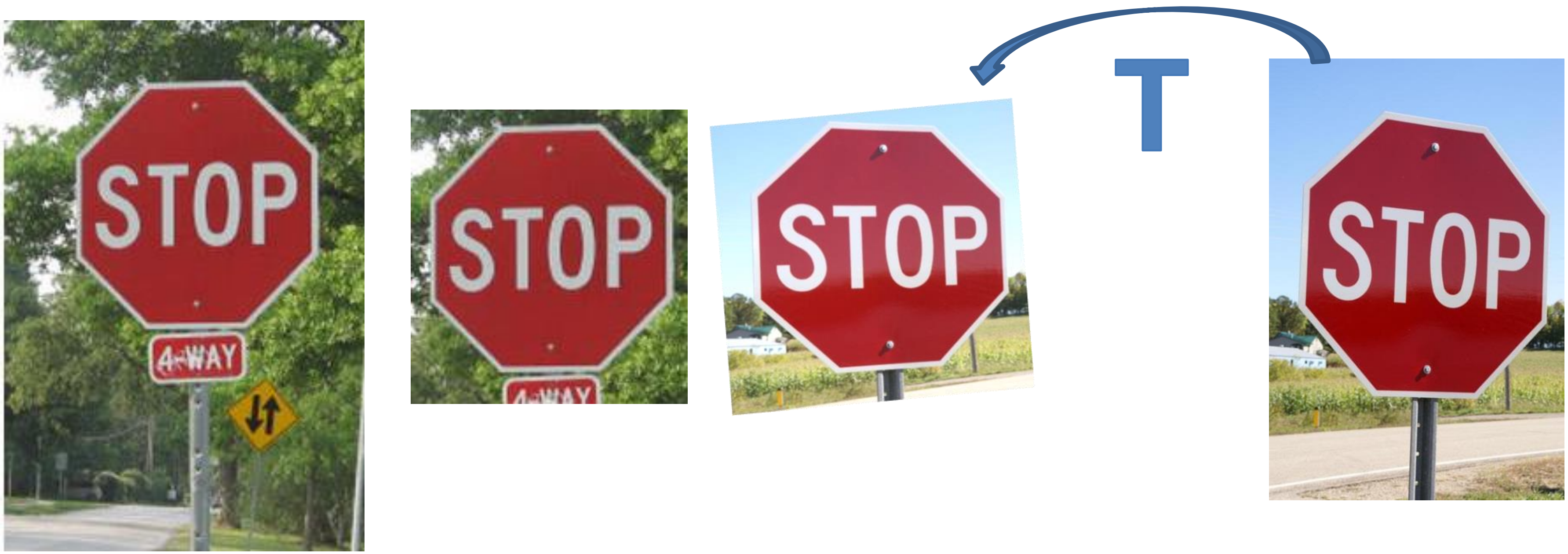
This section: correspondence and alignment

- Correspondence: matching points, patches, edges, or regions across images

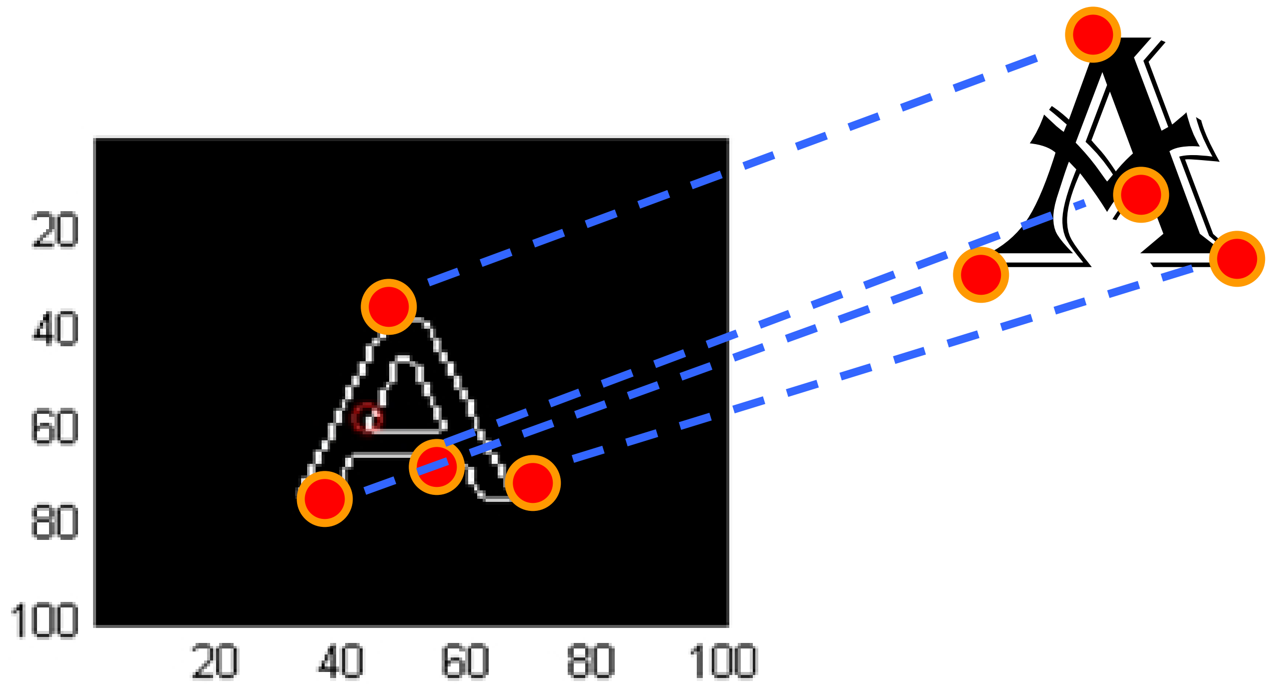


This section: correspondence and alignment

- Alignment: solving the transformation that makes two things match better

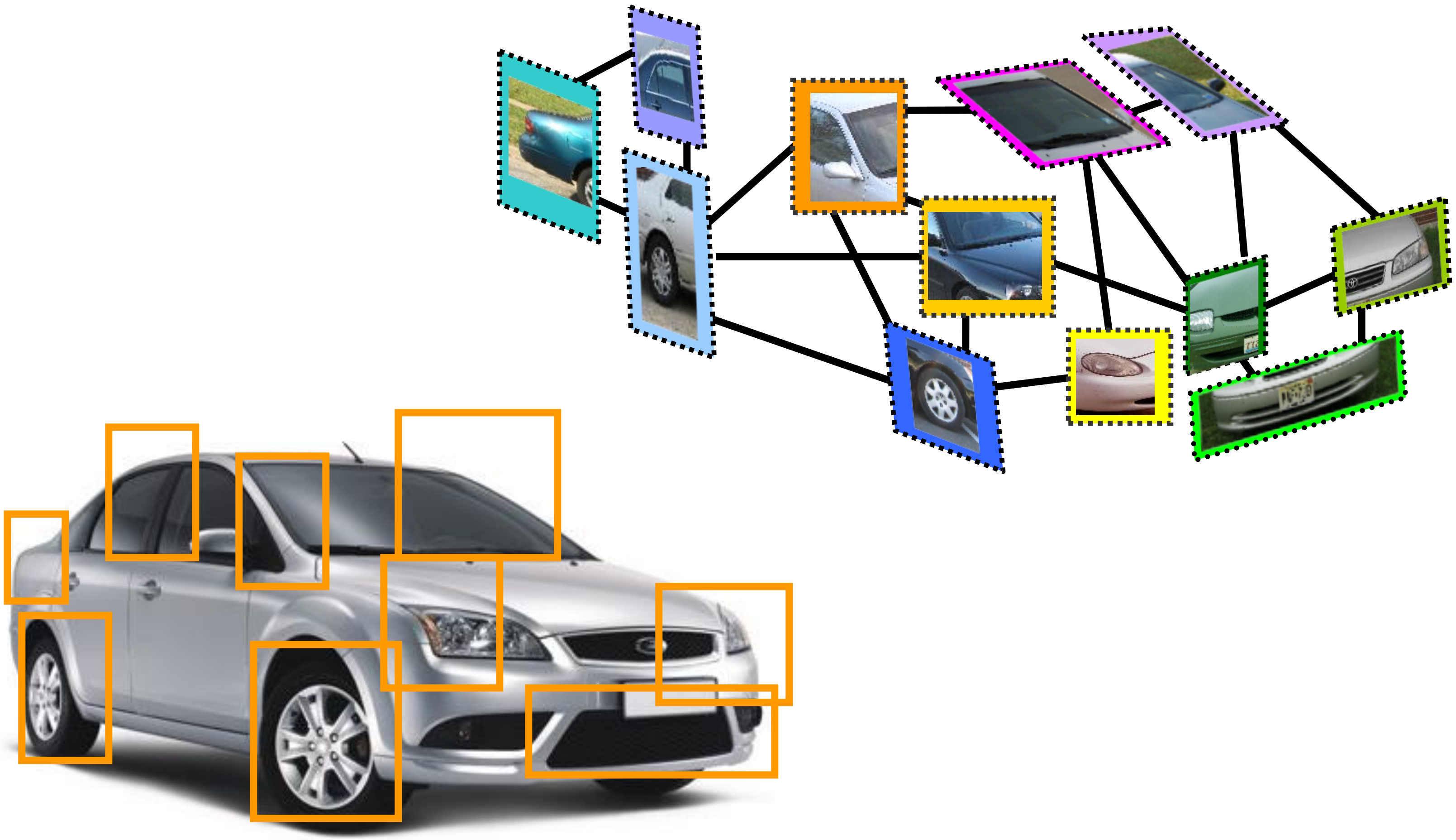


Example: fitting an 2D shape template



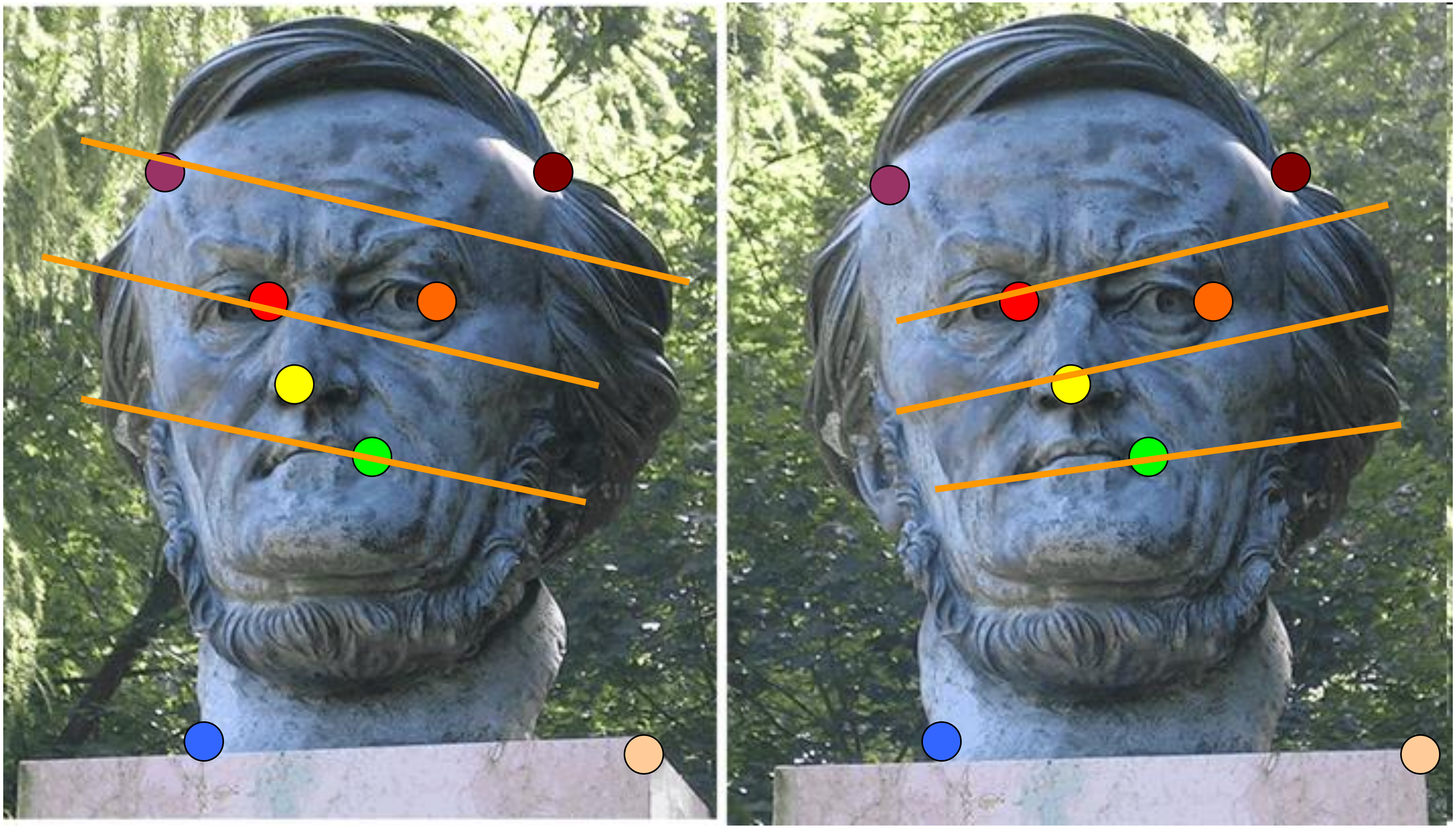
Slide from Silvio Savarese

Example: fitting a 3D object model



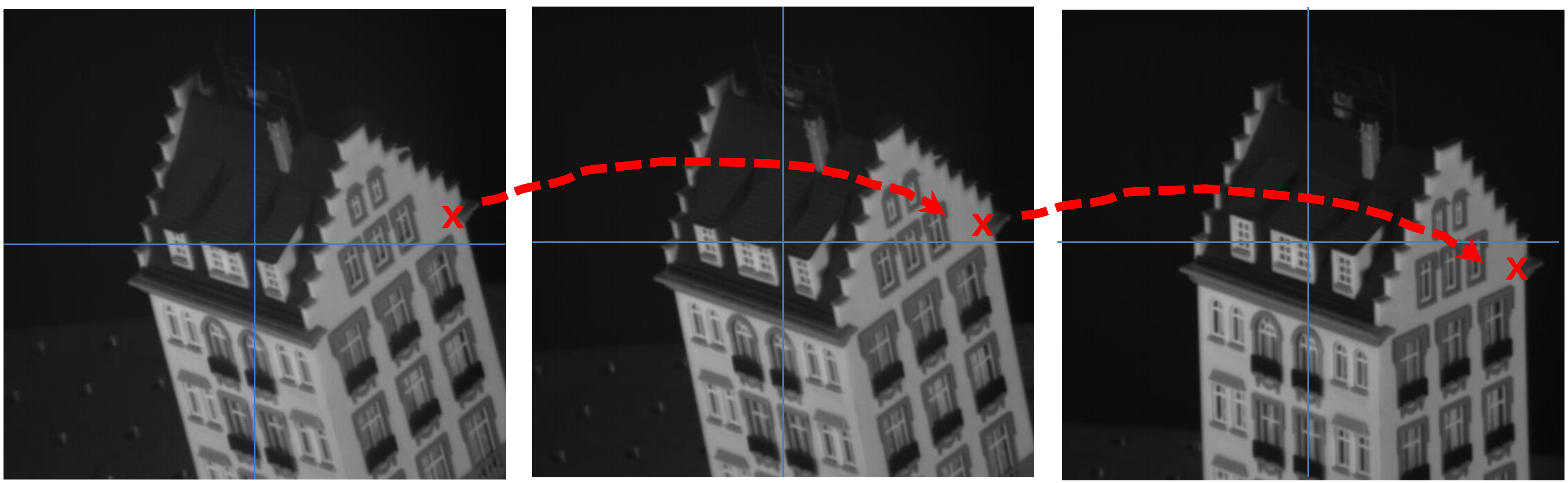
Slide from Silvio Savarese

Example: estimating "fundamental matrix" that corresponds two views



Slide from Silvio Savarese

Example: tracking points



frame 0 | frame 22 | frame 49

Your problem 1 for HW 2!

HW 2

- Interest point detection and tracking
  - Detect trackable points
  - Track them across 50 frames
  - In HW 3, you will use these tracked points for structure from motion

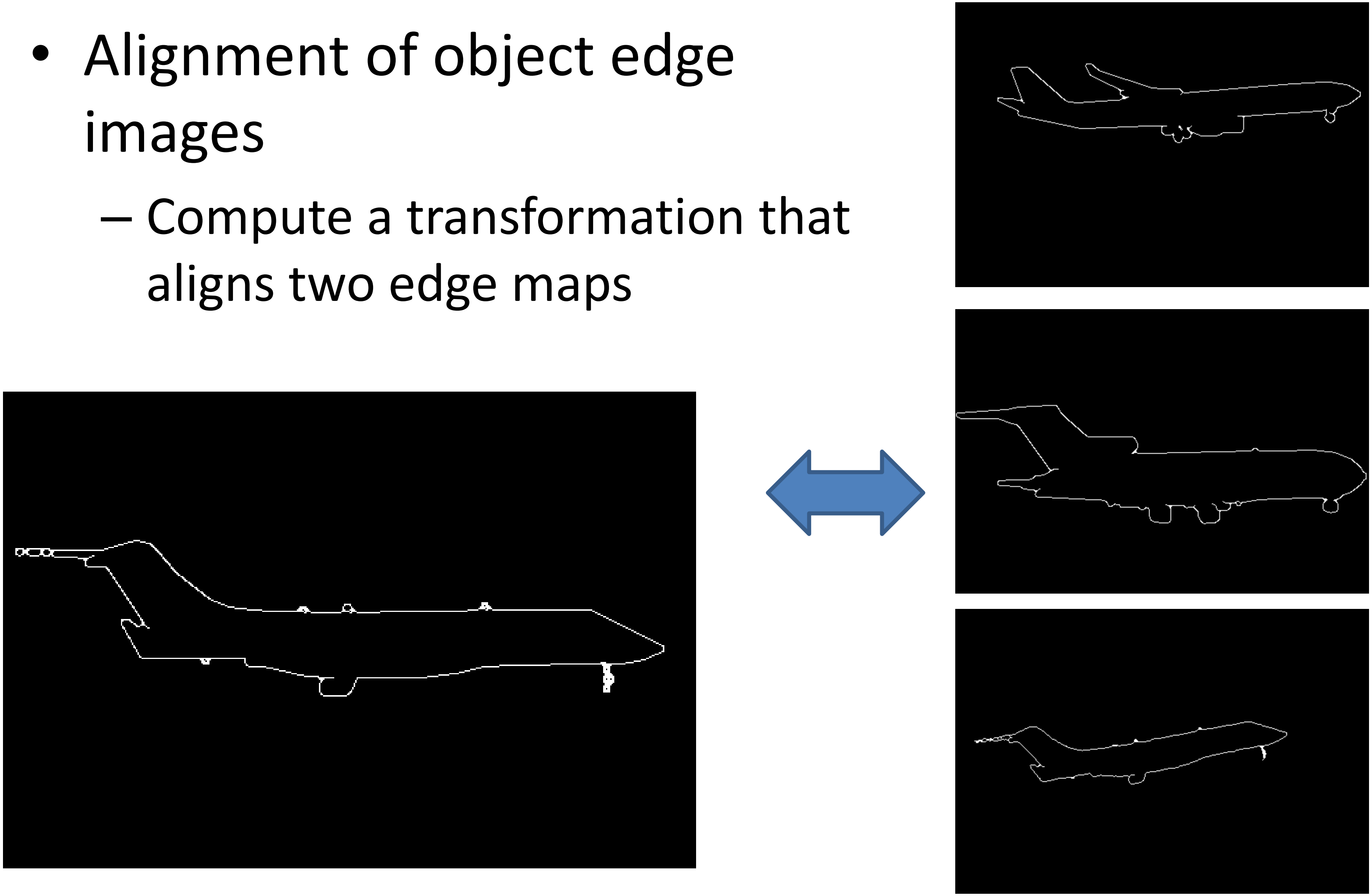
frame 0

frame 22

frame 49

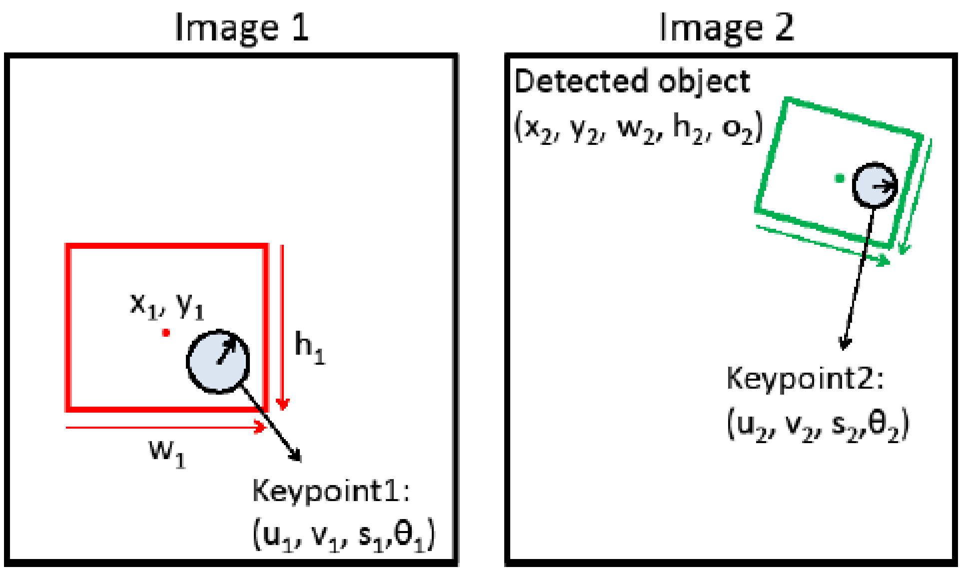
HW 2

- Alignment of object edge images
  - Compute a transformation that aligns two edge maps



HW 2

- Initial steps of object alignment
  - Derive basic equations for interest-point based alignment

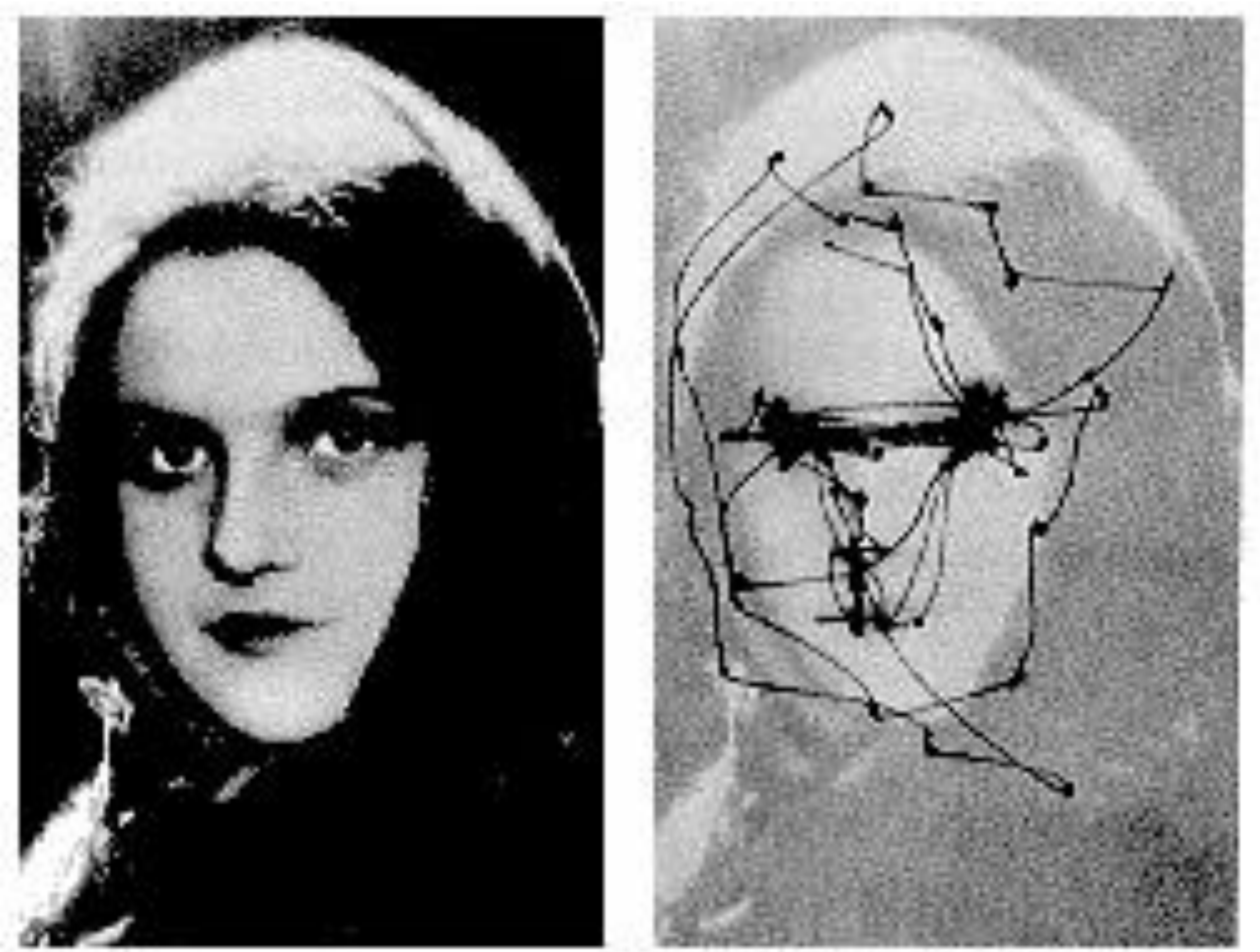


This class: interest points

- Note: “interest points” = “keypoints”, also sometimes called “features”

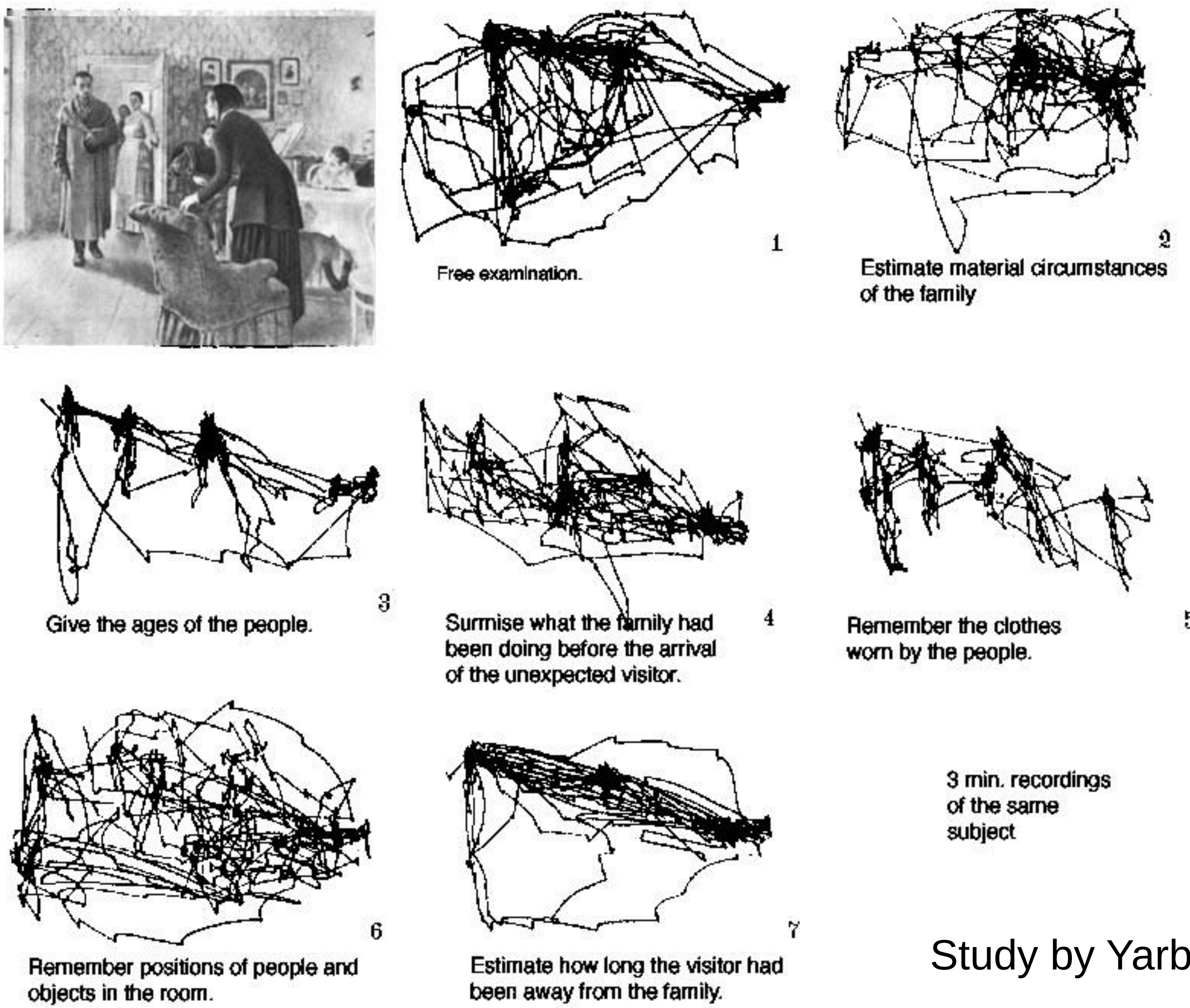
- Many applications
  - tracking: which points are good to track?
  - recognition: find patches likely to tell us something about object category
  - 3D reconstruction: find correspondences across different views

Human eye movements



Yarbus eye tracking

Human eye movements

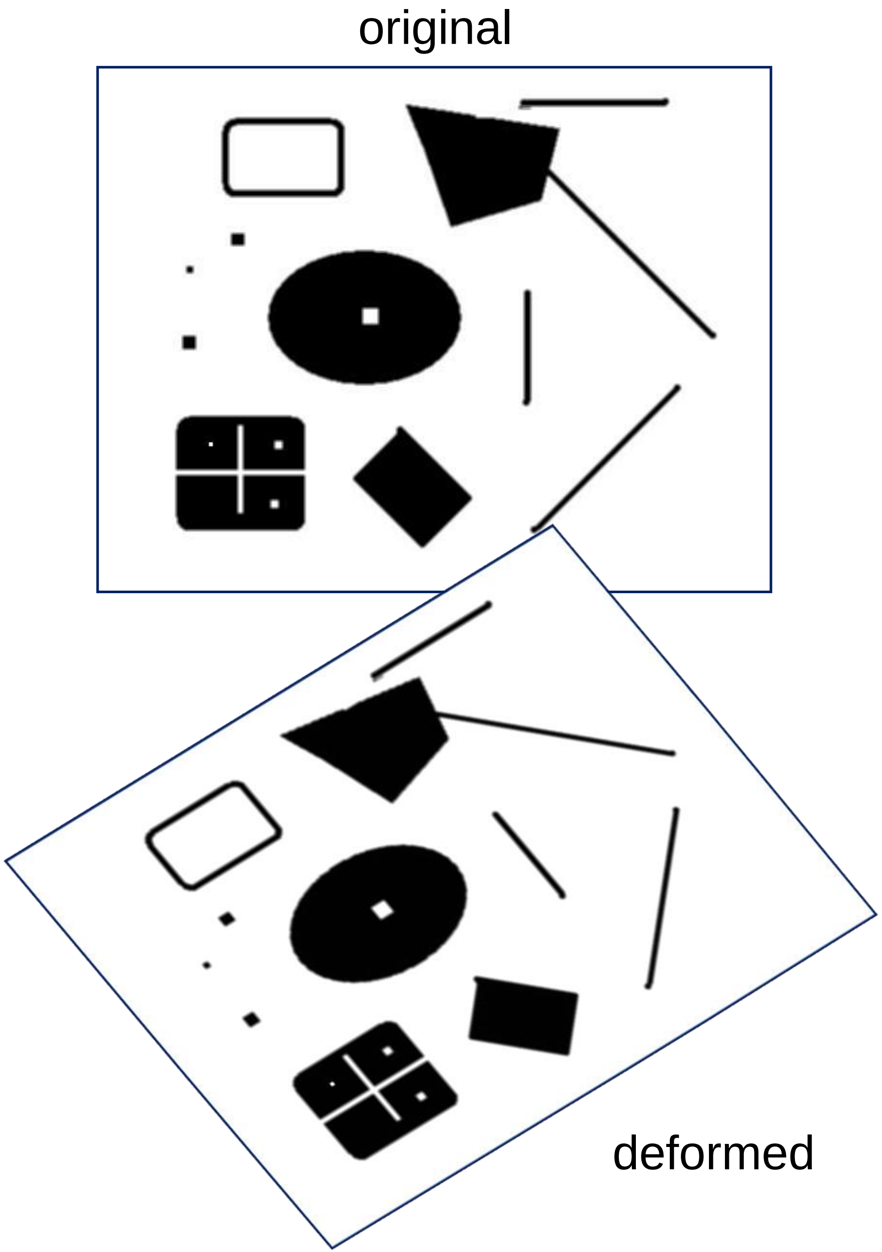


Study by Yarbus

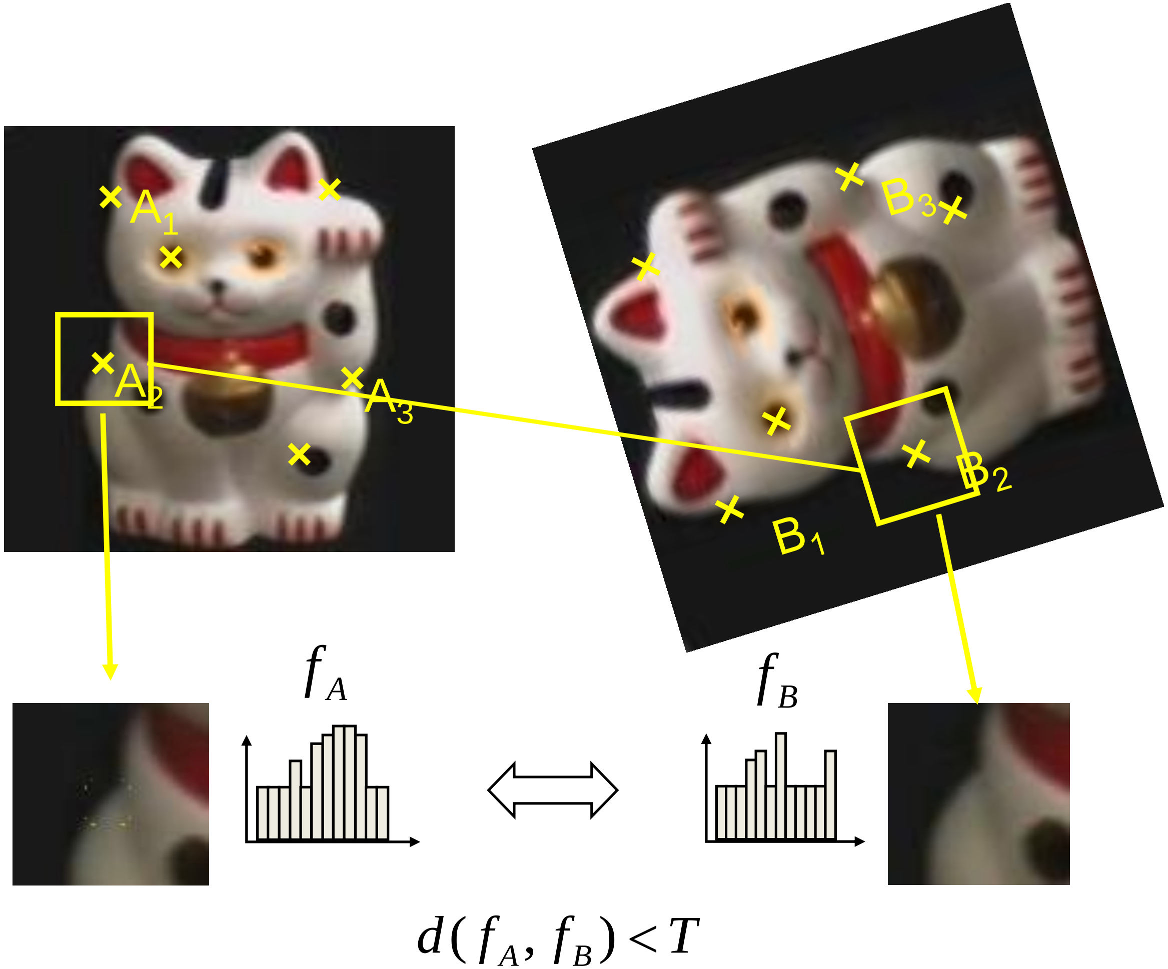
Change blindness: http://www.simonslab.com/videos.html

This class: interest points

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
  - Which points would you choose?



Overview of Keypoint Matching

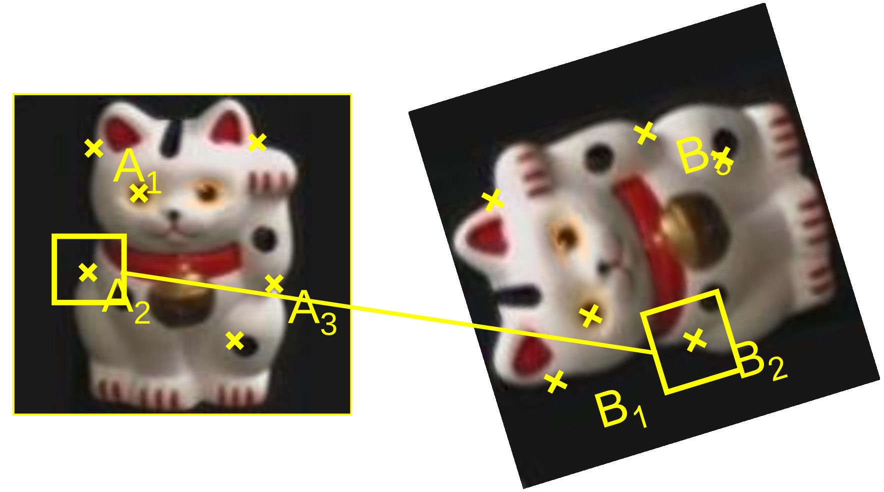


1. **Find a set of distinctive key-points**
2. **Define a region around each keypoint**
3. **Extract and normalize the region content**
4. **Compute a local descriptor from the normalized region**
5. **Match local descriptors**

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Key trade-offs



Detection

More Repeatable

Robust detection

Precise localization

More Points

Robust to occlusion

Works with less texture

Description

More Distinctive

Minimize wrong matches

More Flexible

Robust to expected variations

Maximize correct matches

Choosing interest points

Where would you tell your friend to meet you?

Choosing interest points

Where would you tell your friend to meet you?

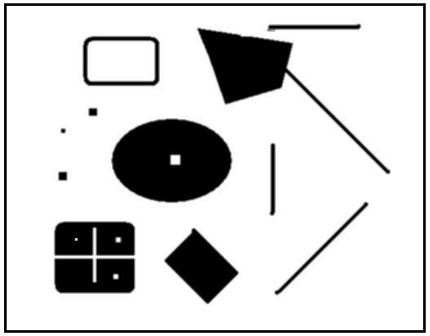
Many Existing Detectors Available

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…	

Harris Detector [Harris88]

- Second moment matrix

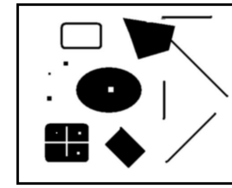
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



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

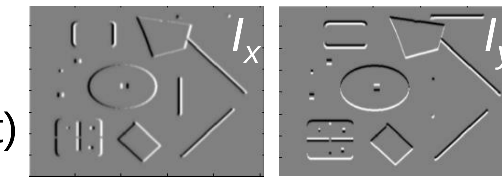
Harris Detector [Harris88]

- Second moment matrix



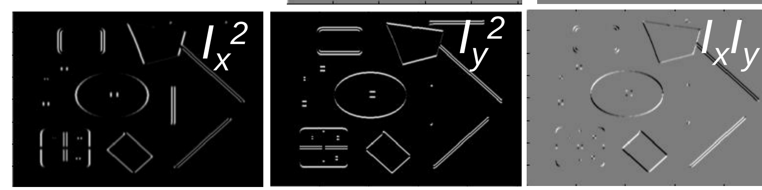
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives (optionally, blur first)



$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

2. Square of derivatives



3. Gaussian filter *g(σI)*

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\operatorname{trace}(\mu(\sigma_I, \sigma_D))^2] =$$
$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression

Harris Detector: Mathematics

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

1. Want large eigenvalues, and small ratio $\frac{\lambda_1}{\lambda_2} < t$

2. We know

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

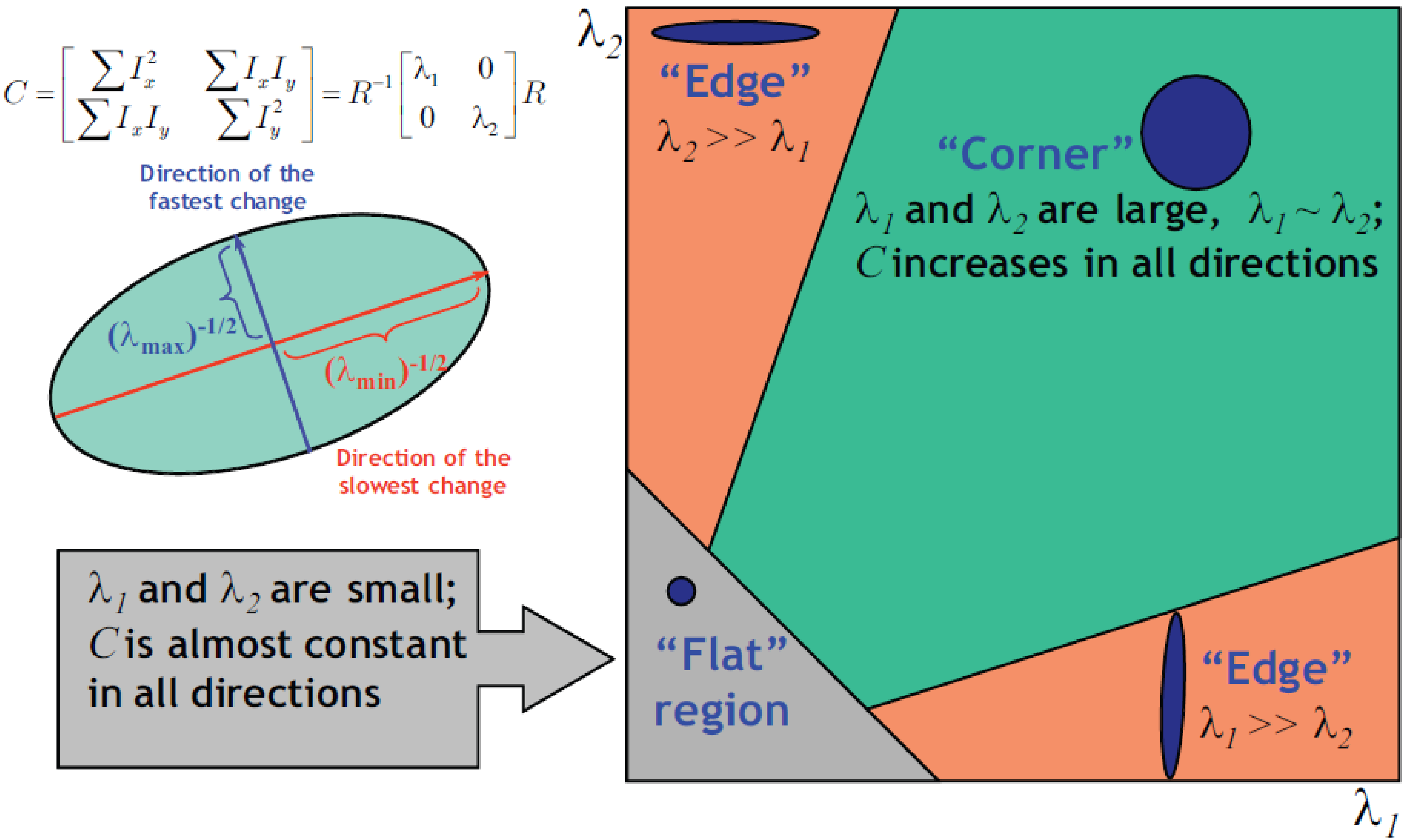
3. Leads to

$$\det M - k \cdot \operatorname{trace}^2(M) > t$$

(k :empirical constant, k = 0.04-0.06)

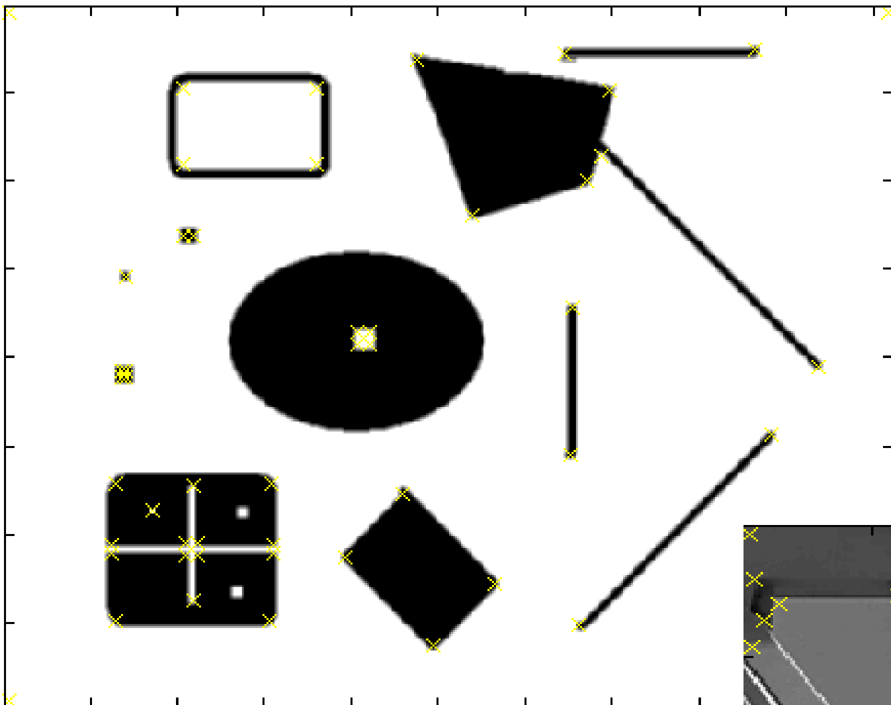
[Nice brief derivation on wikipedia]

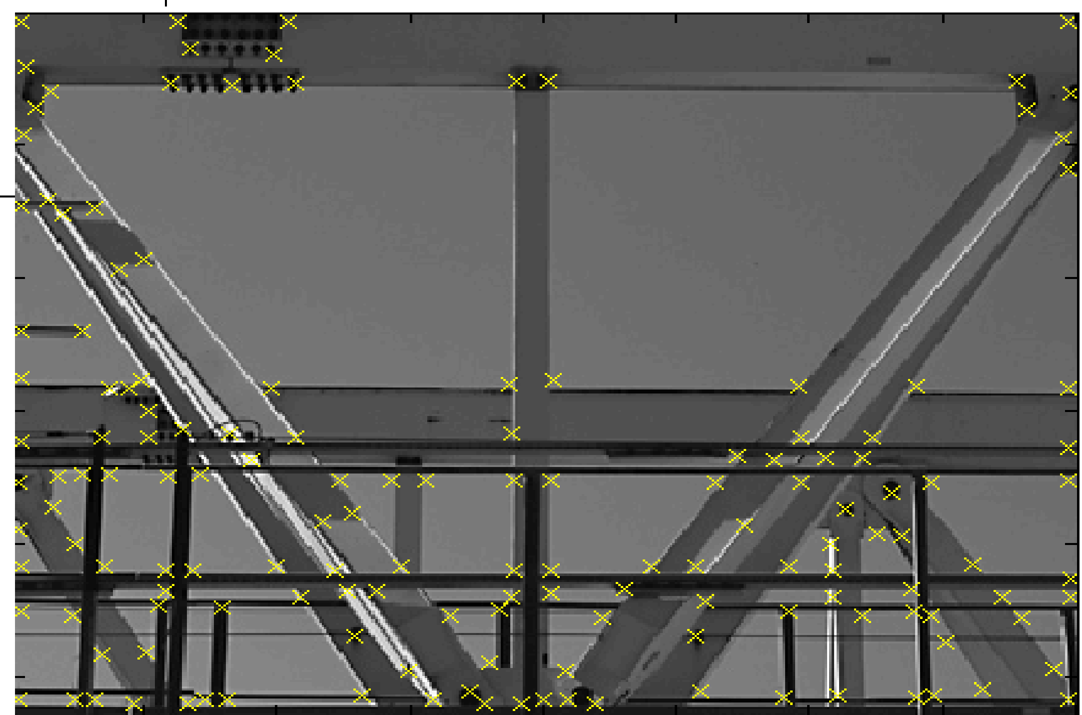
Explanation of Harris Criterion



From [Grauman and Leibe]

Harris Detector – Responses [Harris88]





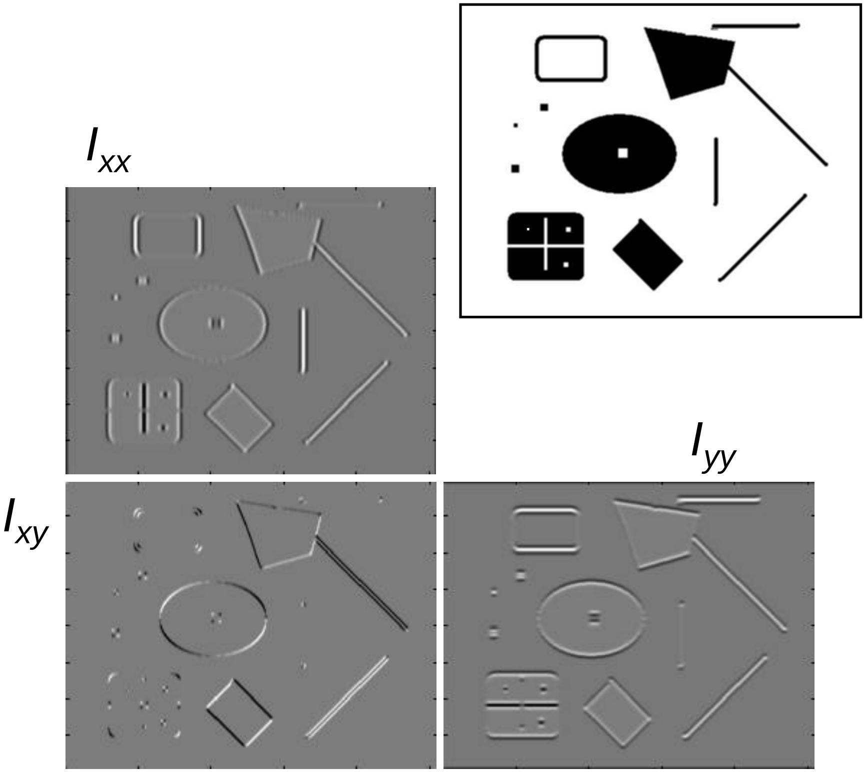
Effect: **A very precise corner detector.**

Harris Detector - Responses [Harris88]

Hessian Detector [Beaudet78]

- Hessian determinant

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



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

Hessian Detector [Beaudet78]

- Hessian determinant

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

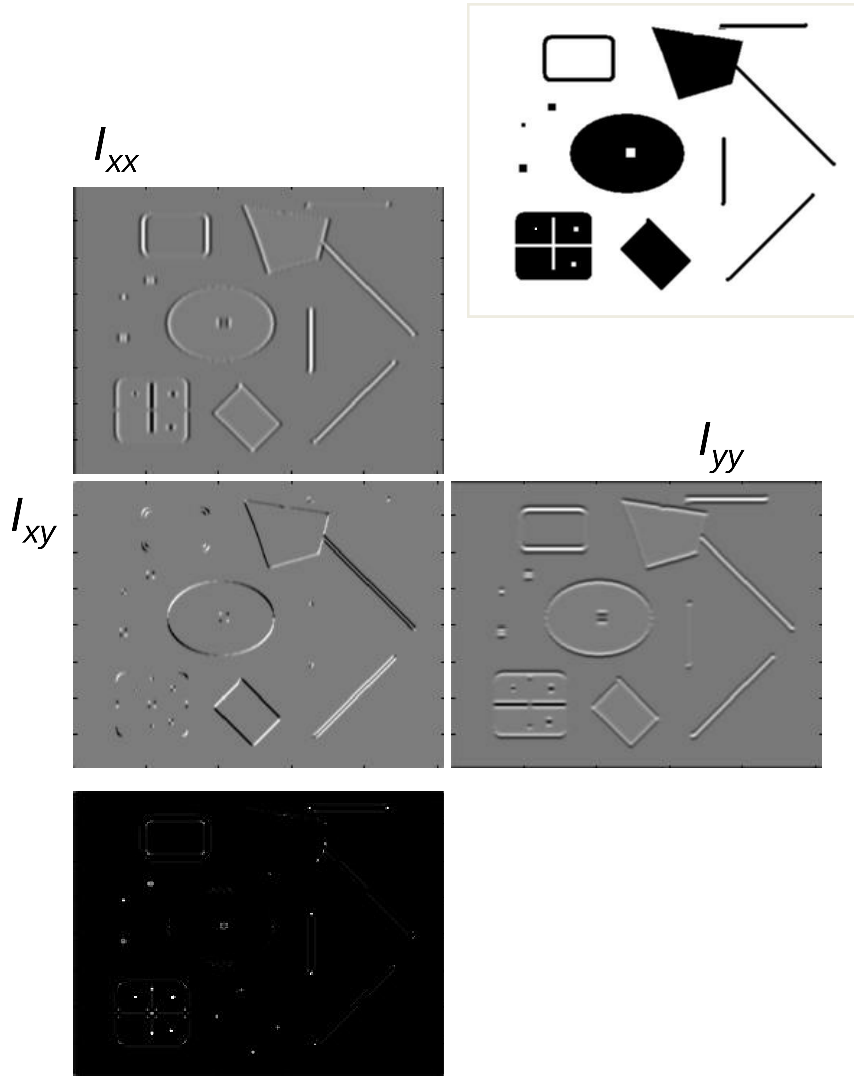
$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

Find maxima of determinant

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

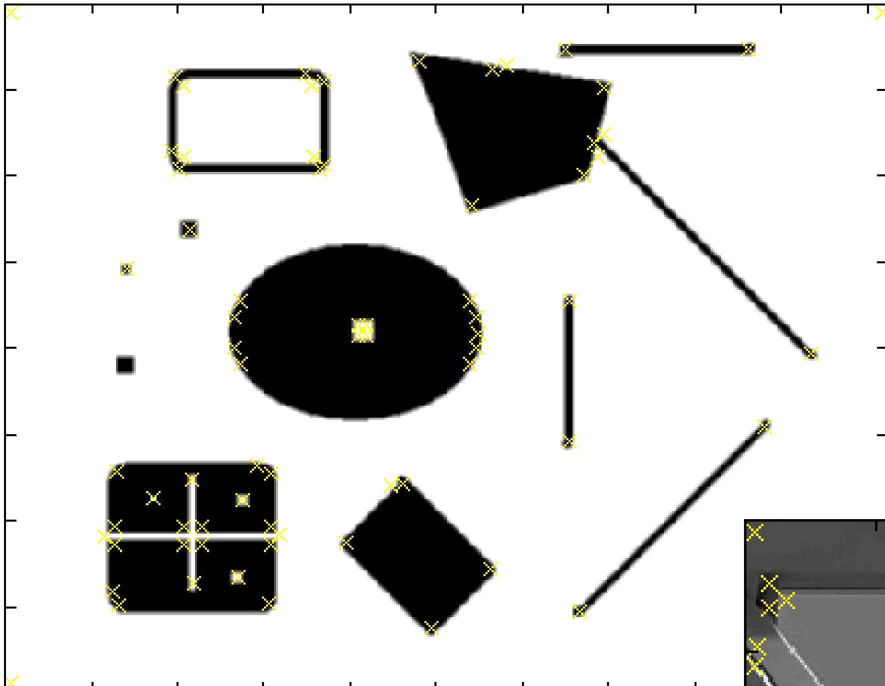
In Matlab:

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

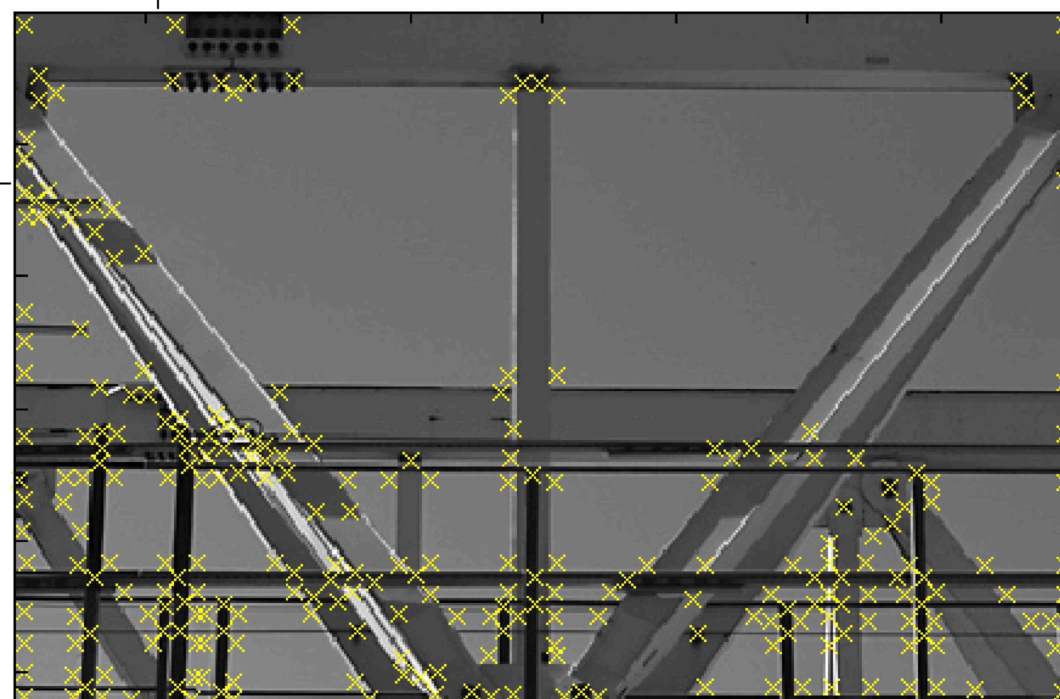


K. Grauman, B. Leibe

Hessian Detector – Responses [Beaudet78]



Effect: **Responses mainly on corners and strongly textured areas.**



Hessian Detector – Responses [Beaudet78]

So far: can localize in x-y, but not scale

Automatic Scale Selection

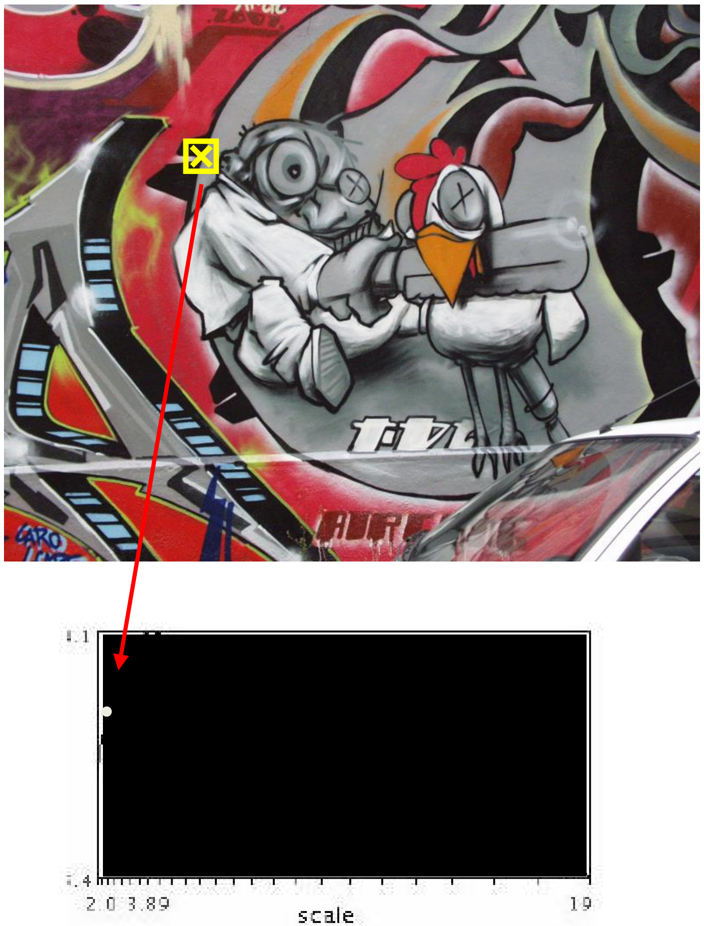
$$f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma'))$$

How to find corresponding patch sizes?

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

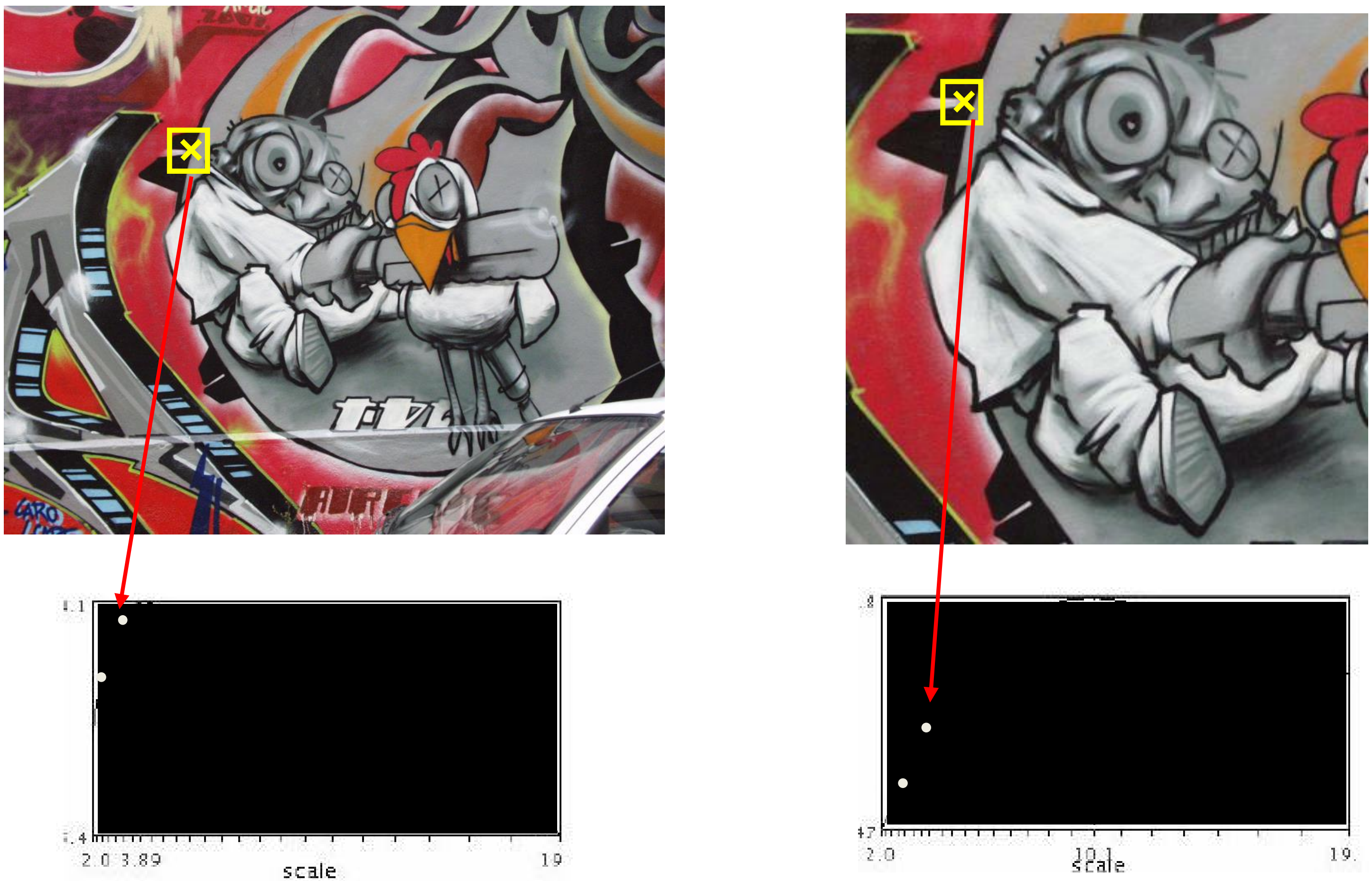


$f(I_{i_1 \dots i_m}(x, \sigma))$

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

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

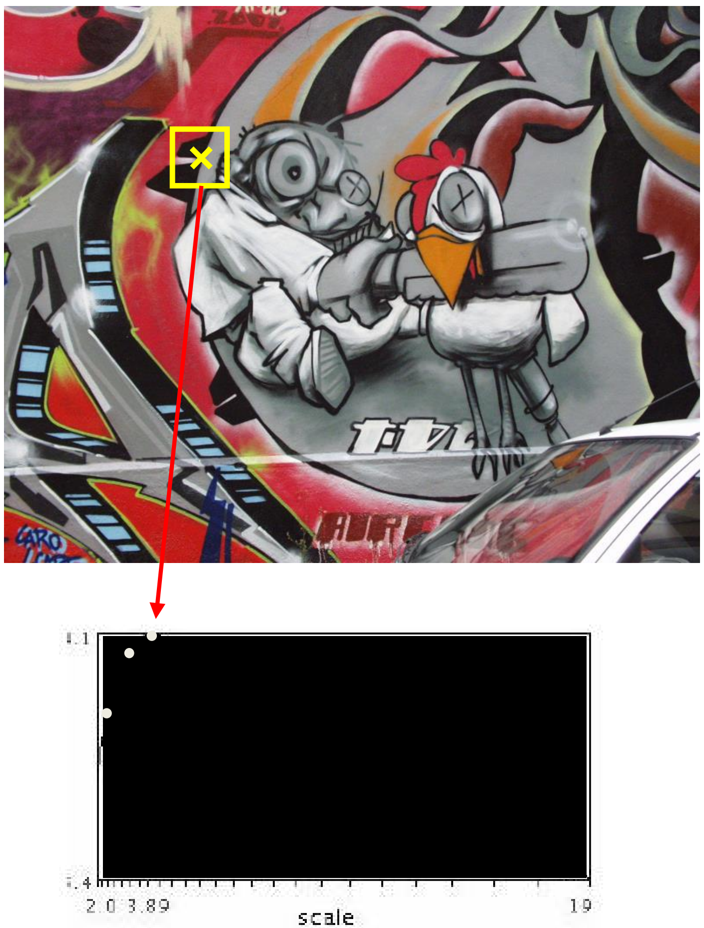


$$f(I_{i_1 \ldots i_m}(x,\sigma)) \qquad f(I_{i_1 \ldots i_m}(x',\sigma))$$

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



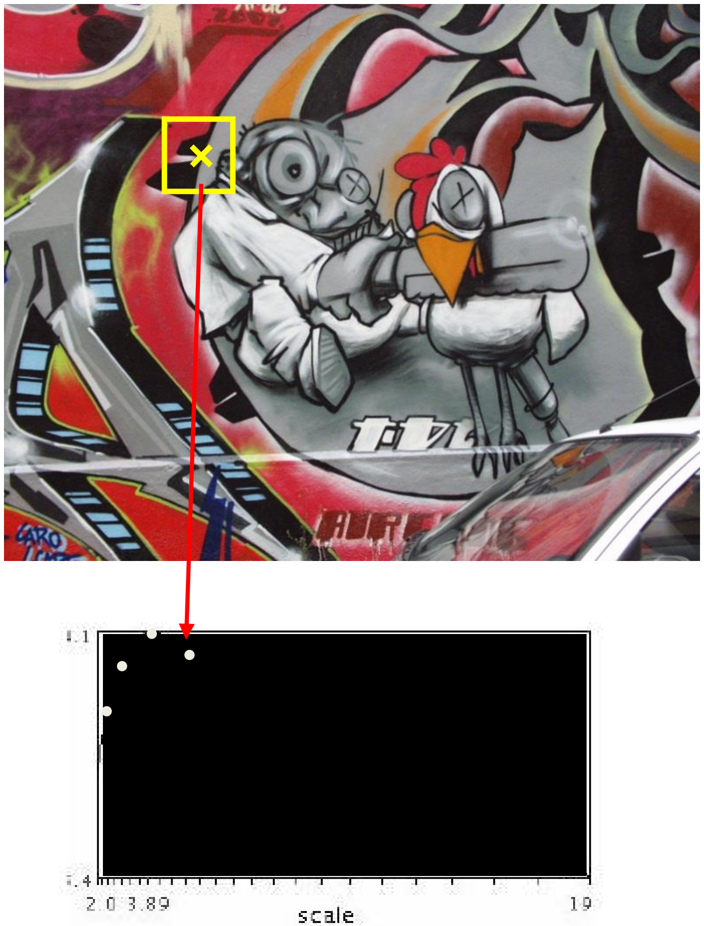
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



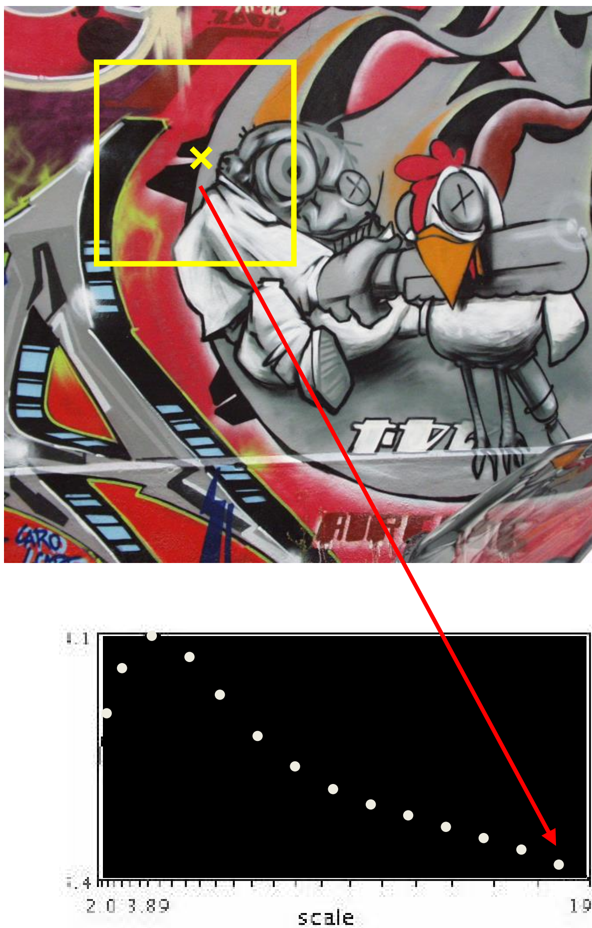
$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

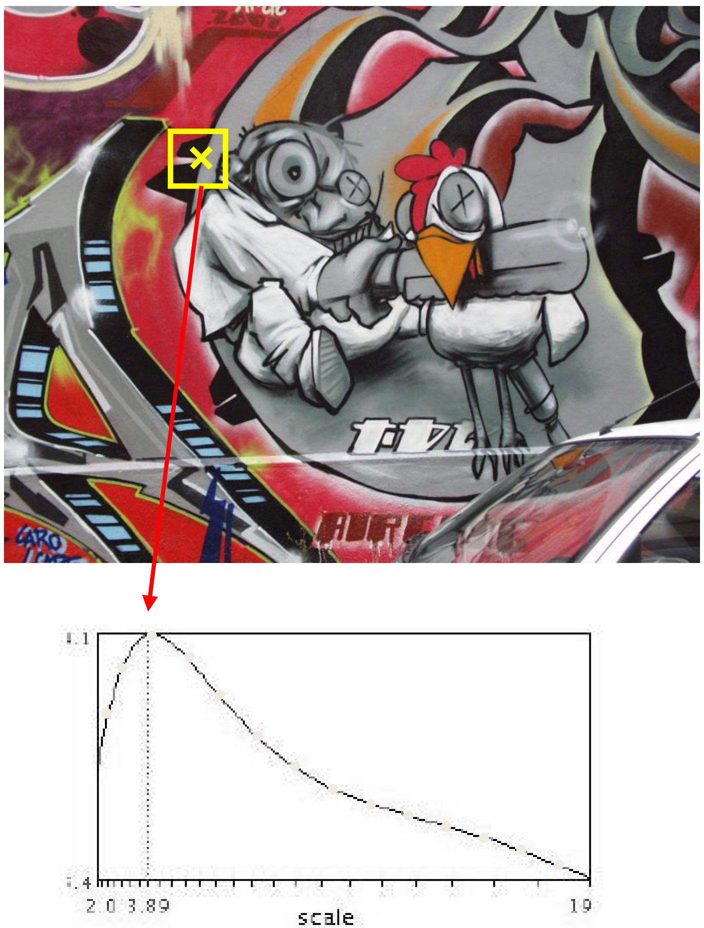


$f(I_{i_1 \ldots i_m}(x,\sigma))$

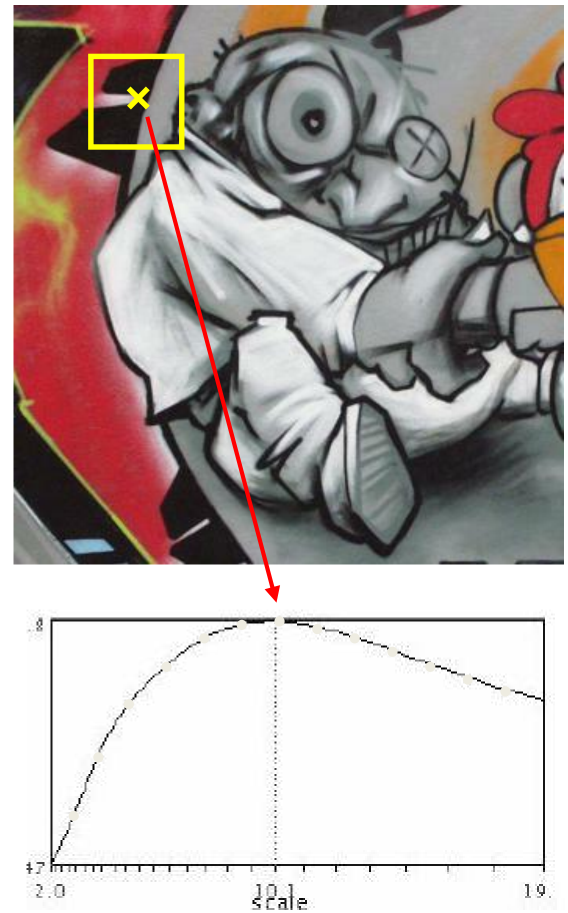
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



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

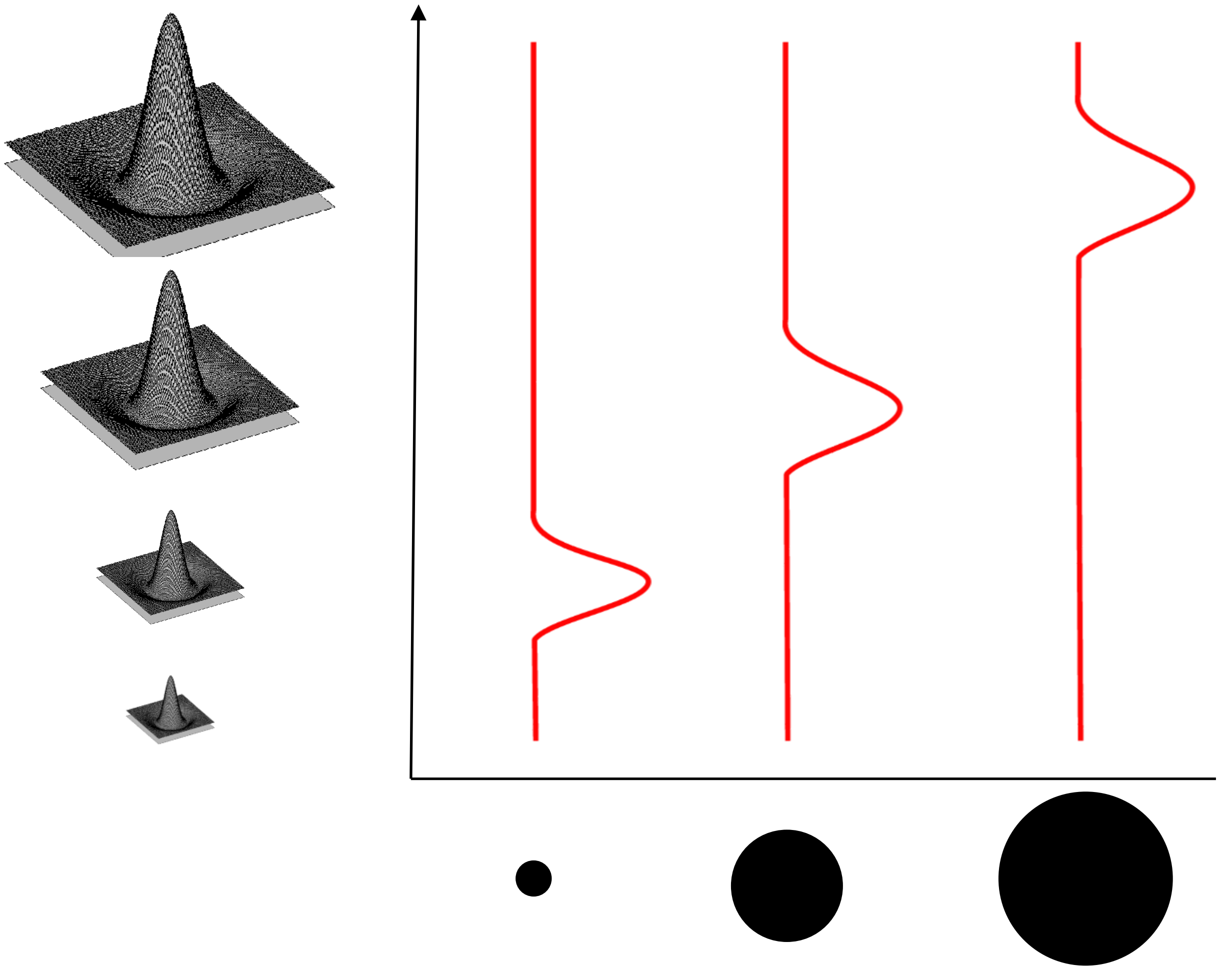


$f(I_{i_1 \ldots i_m}(x', \sigma'))$

K. Grauman, B. Leibe

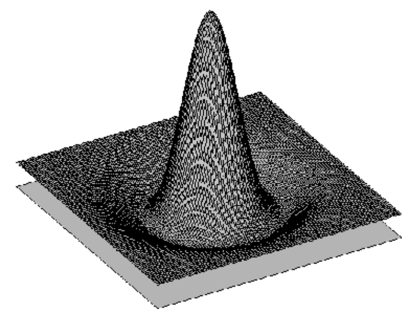
What Is A Useful Signature Function?

- Difference-of-Gaussian = “blob” detector



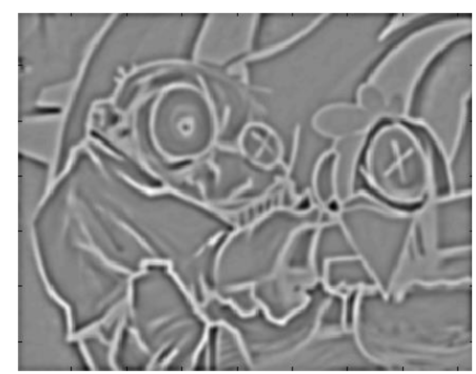
K. Grauman, B. Leibe

Difference-of-Gaussian (DoG)



-

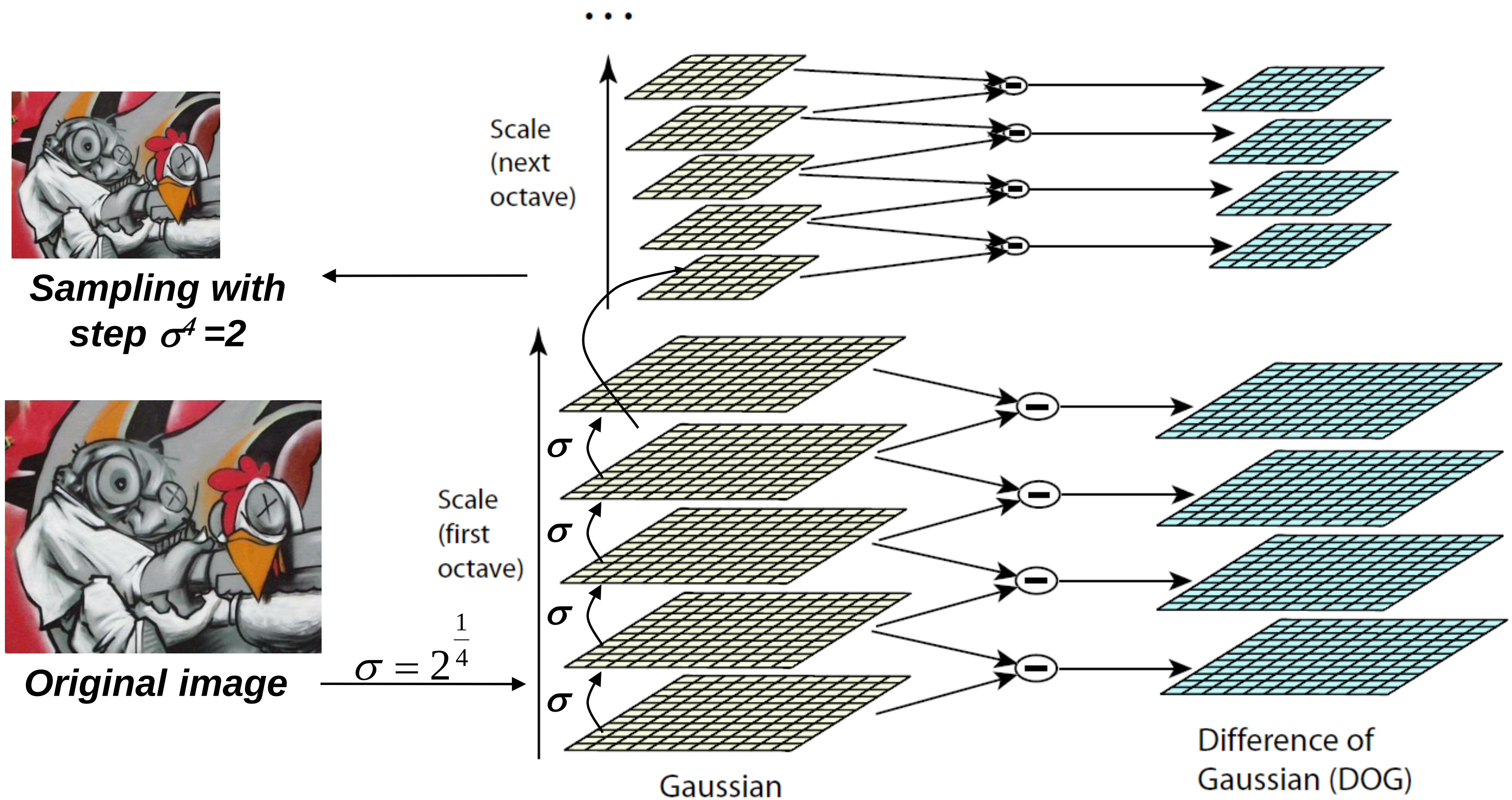
=



K. Grauman, B. Leibe

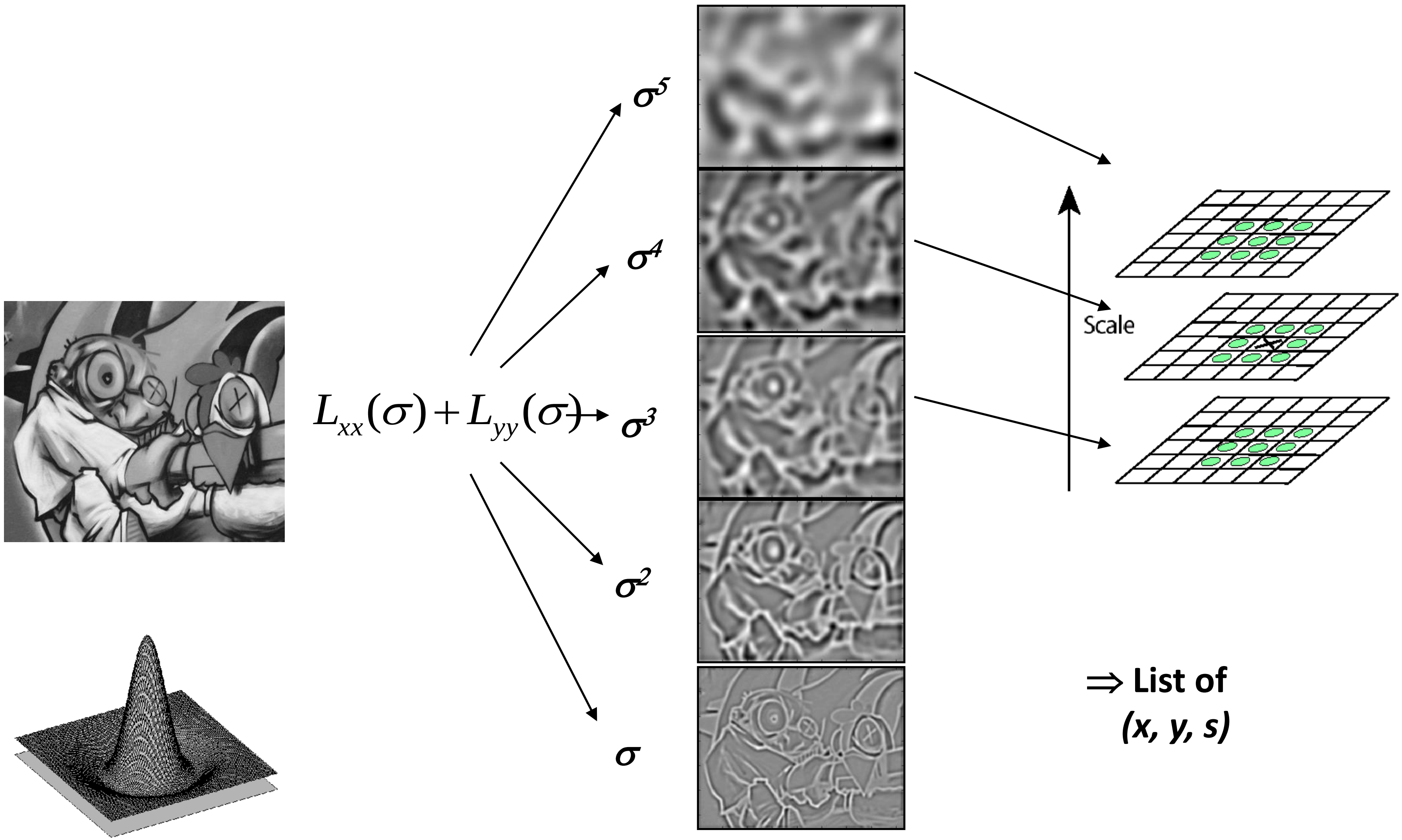
DoG – Efficient Computation

- Computation in Gaussian scale pyramid



K. Grauman, B. Leibe

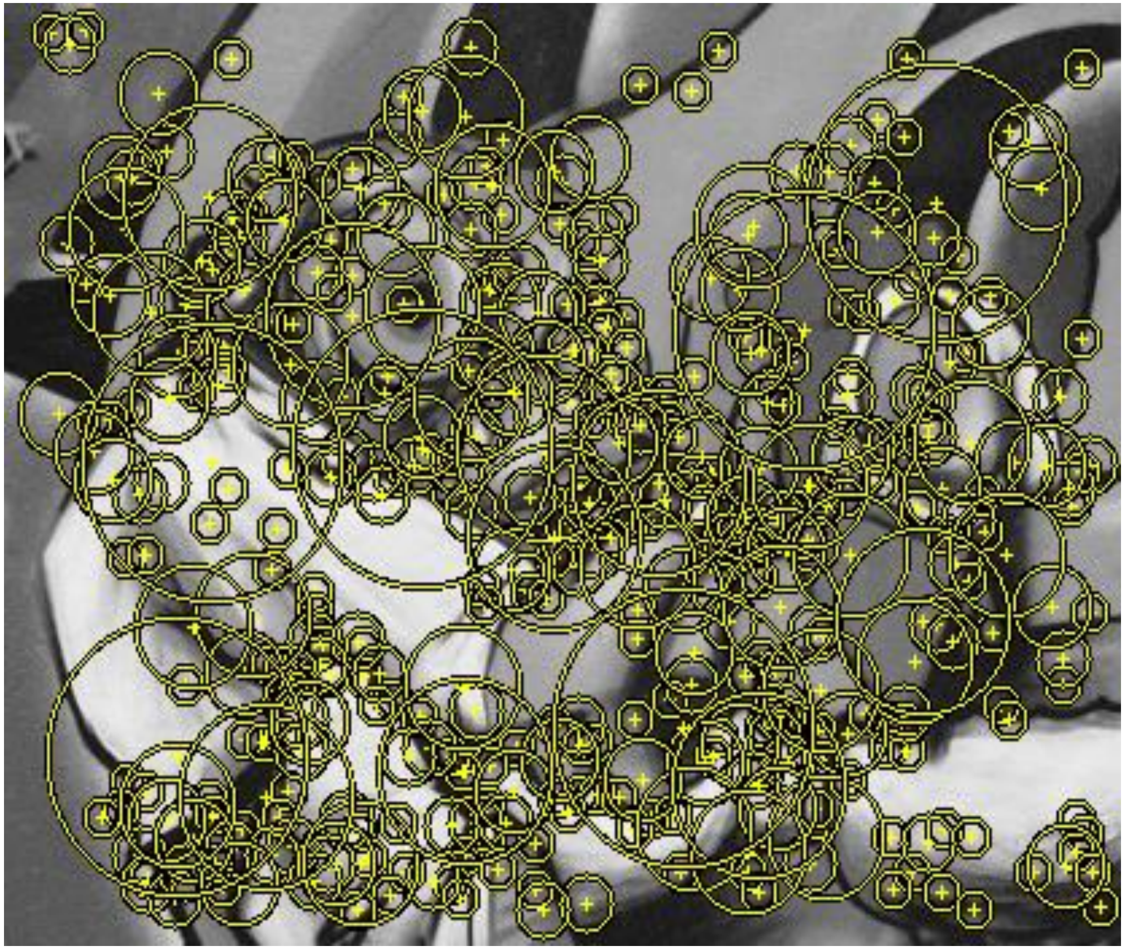
Find local maxima in position-scale space of Difference-of-Gaussian



$\Rightarrow$ **List of** ***(x, y, s)***

K. Grauman, B. Leibe

Results: Difference-of-Gaussian



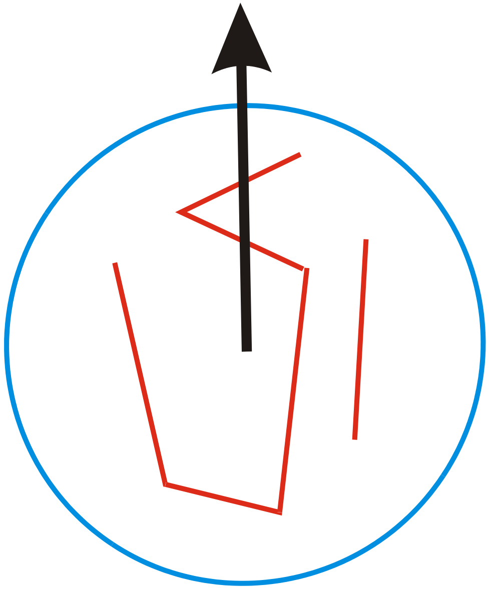
K. Grauman, B. Leibe

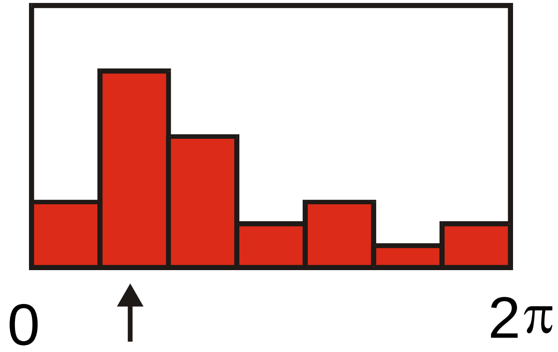
Orientation Normalization

- Compute orientation histogram

[Lowe, SIFT, 1999]

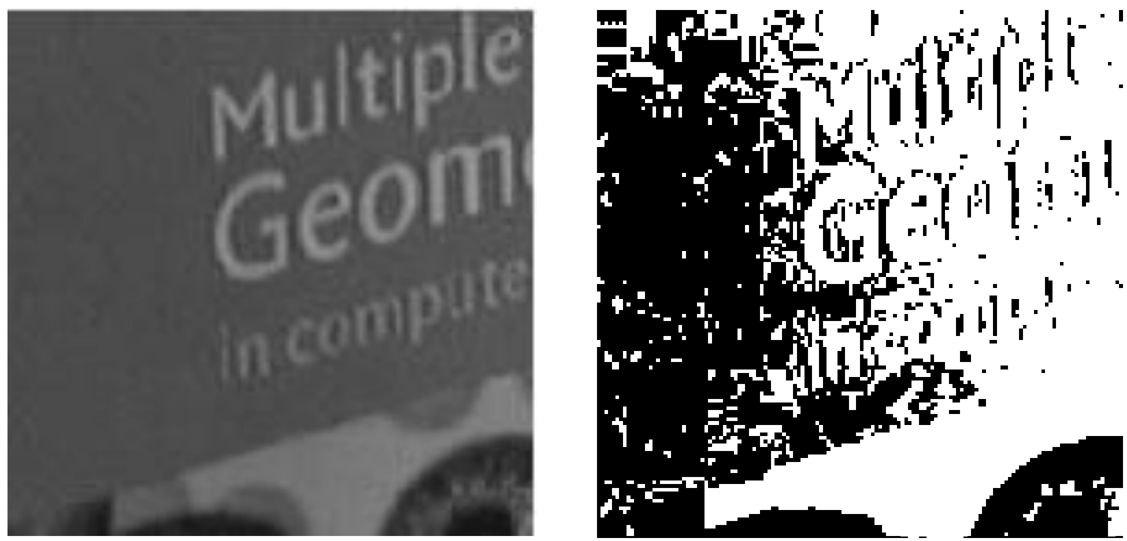
- Select dominant orientation
- Normalize: rotate to fixed orientation





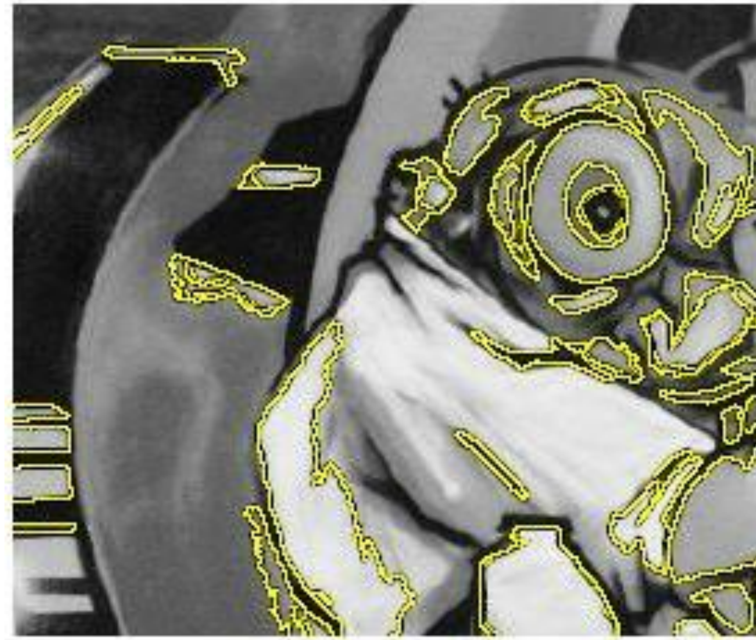
Maximally Stable Extremal Regions [Matas ‘02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range



K. Grauman, B. Leibe

Example Results: MSER



Available at a web site near you…

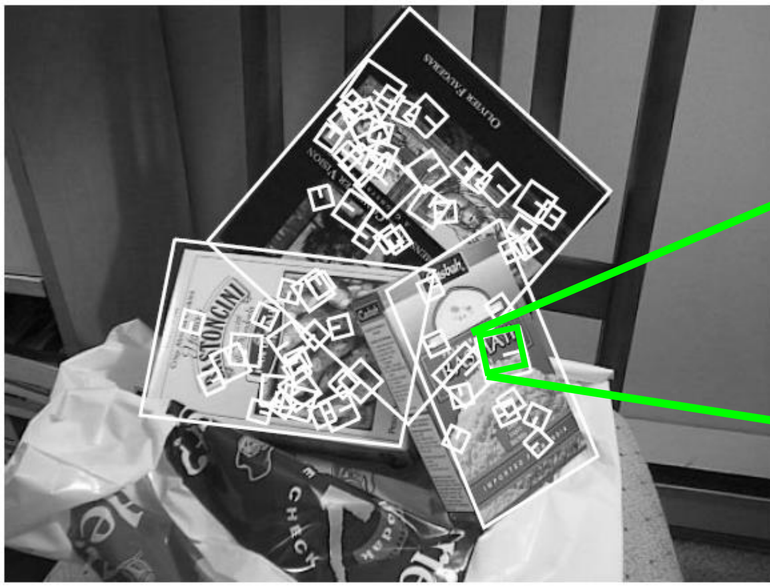
- For most local feature detectors, executables are available online:
  - http://www.robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.vision.ee.ethz.ch/~surf

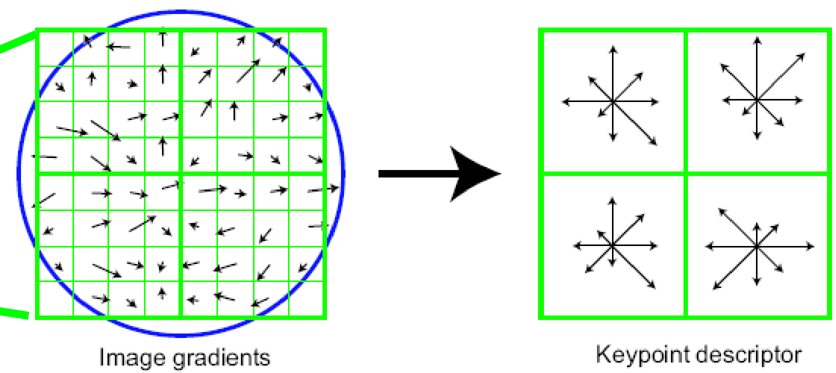
Local Descriptors

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

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

Local Descriptors: SIFT Descriptor





Histogram of oriented gradients

- **Captures important texture information**
- **Robust to small translations / affine deformations**

[Lowe, ICCV 1999]

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

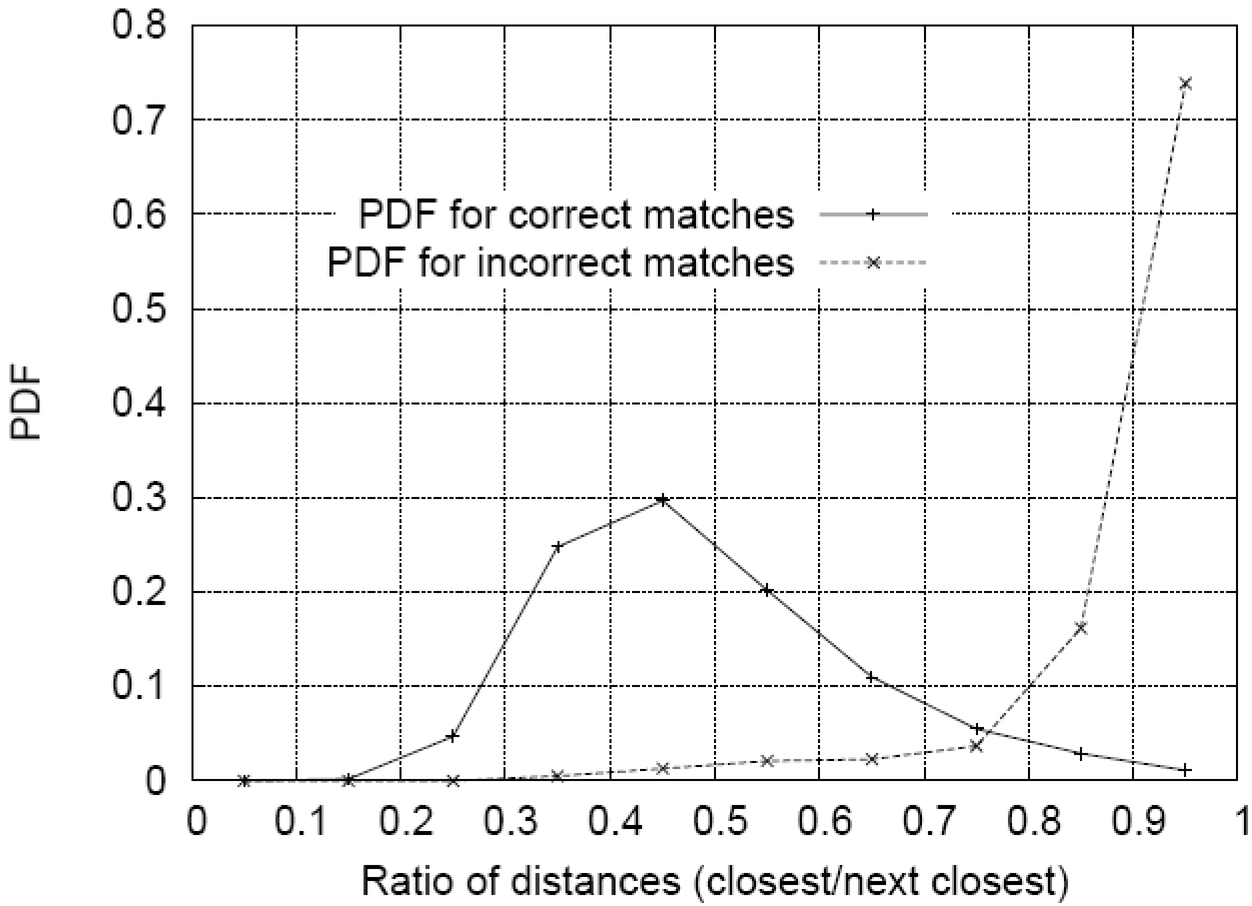
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

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

Lowe IJCV 2004

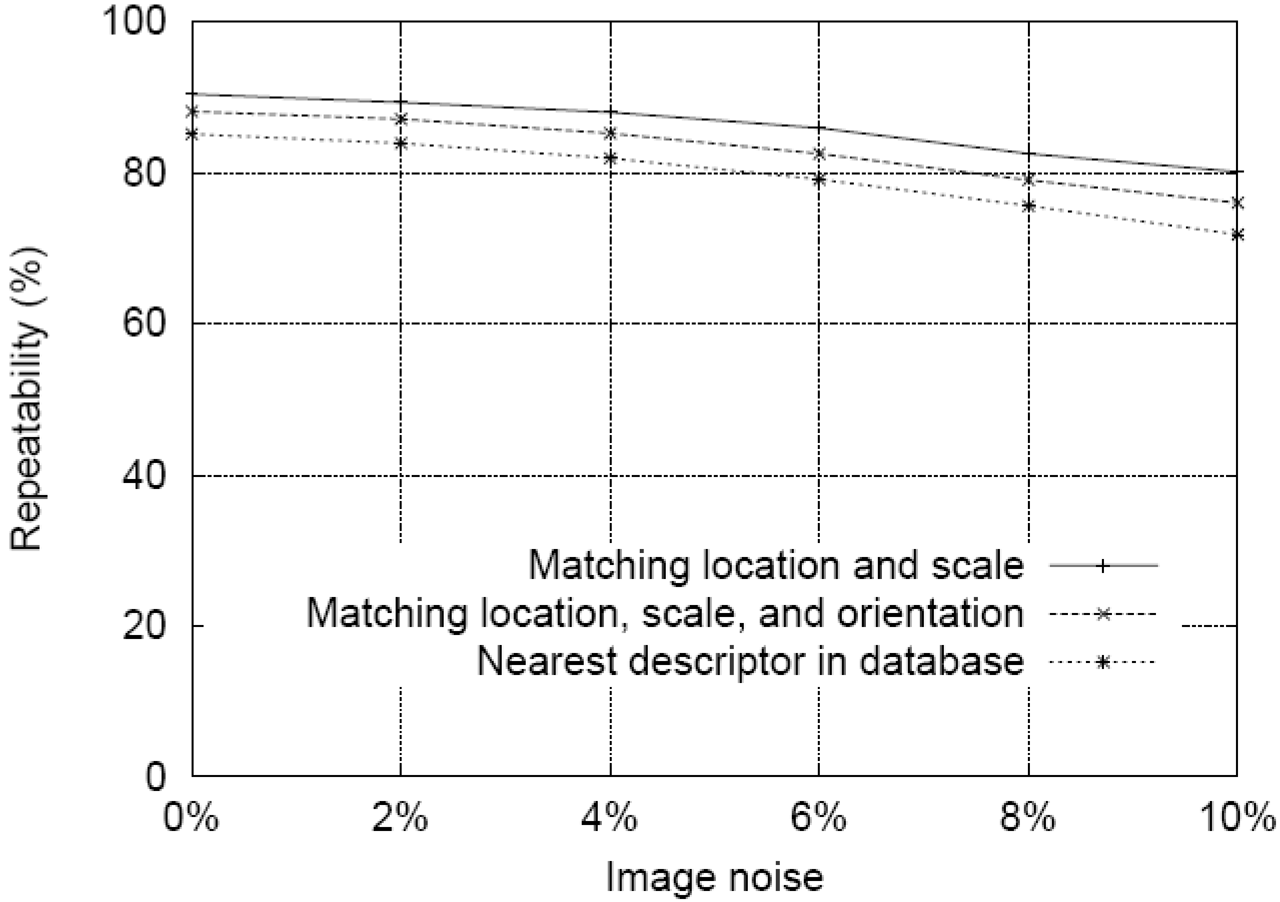
Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2[nd] nearest descriptor



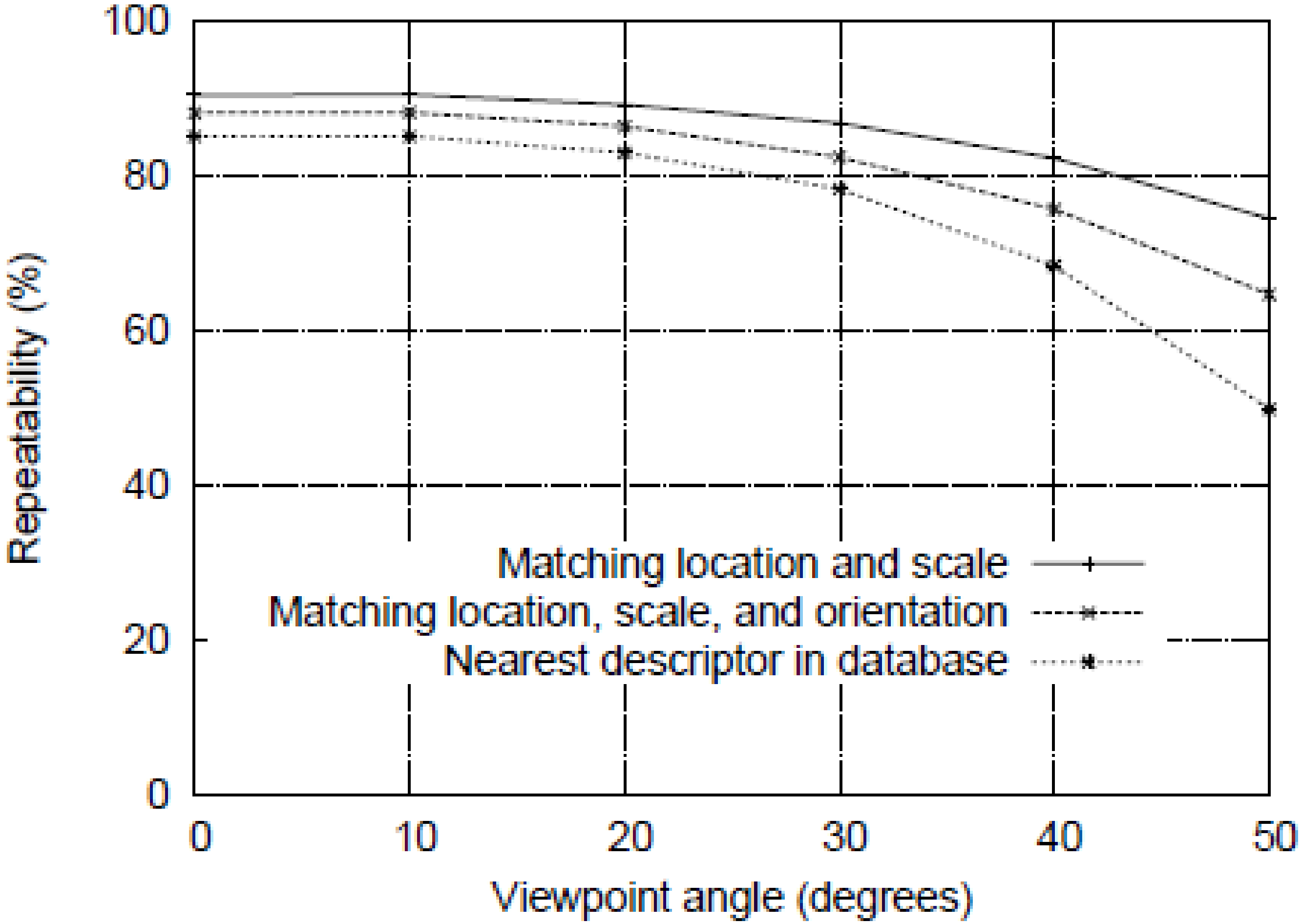
Lowe IJCV 2004

SIFT Repeatability

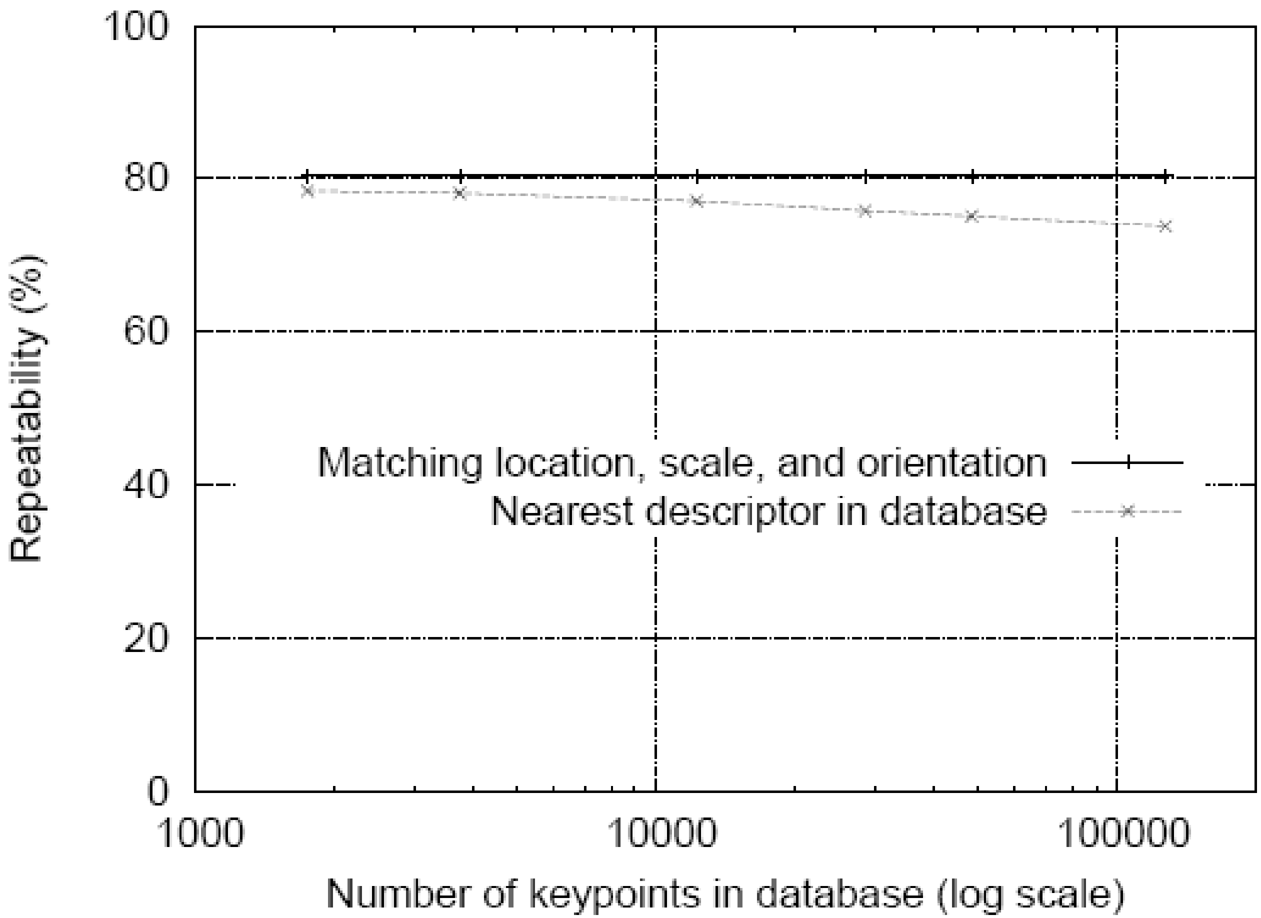


Lowe IJCV 2004

SIFT Repeatability

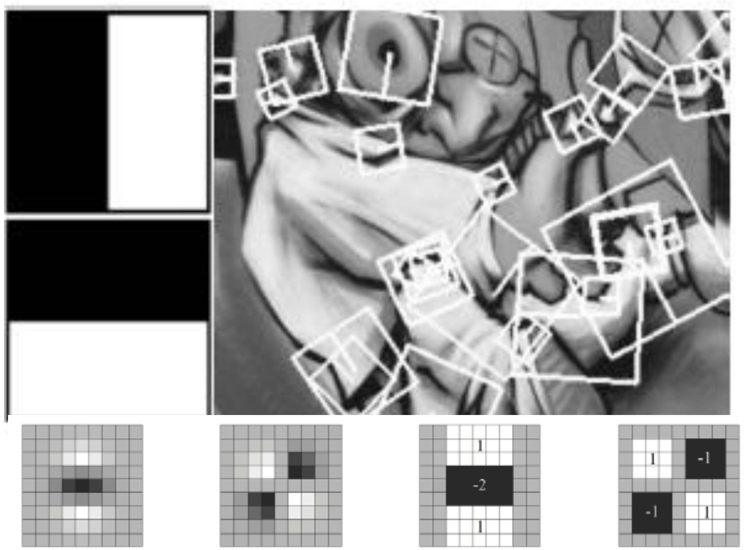


SIFT Repeatability



Lowe IJCV 2004

Local Descriptors: SURF



Fast approximation of SIFT idea

- **Efficient computation by 2D box filters & integral images**
  ⇒ 6 times faster than SIFT
- **Equivalent quality for object identification**

GPU implementation available

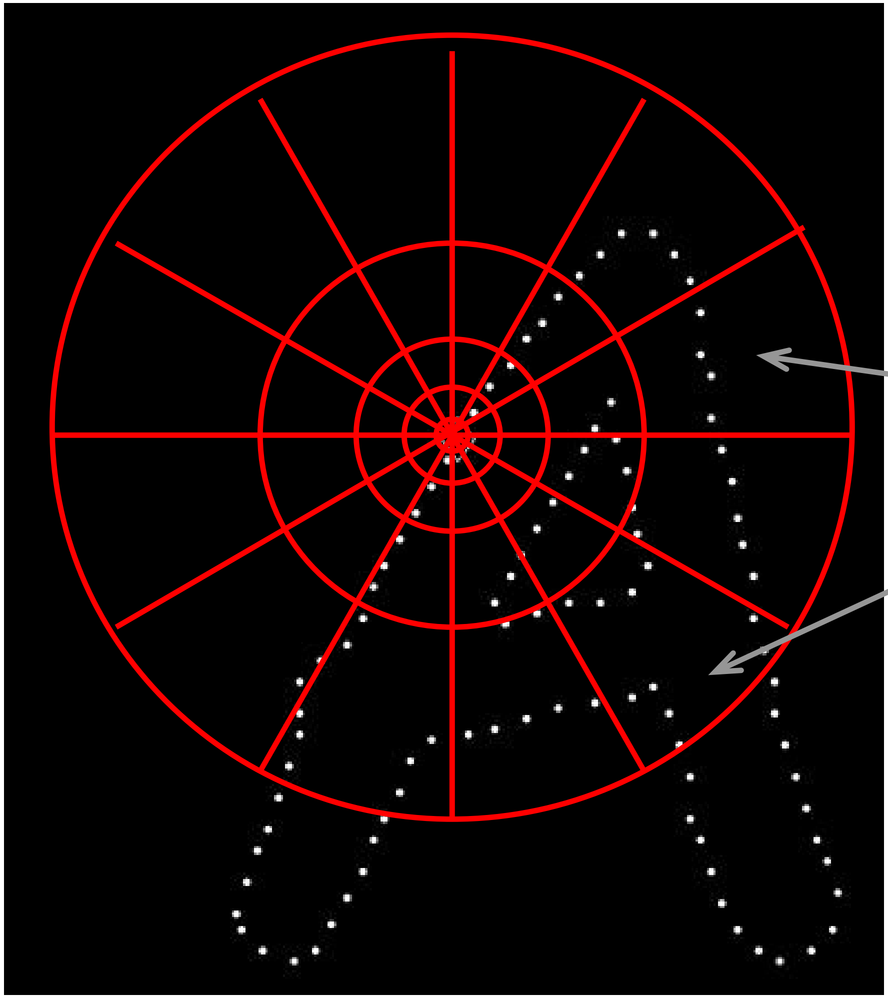
- **Feature extraction @ 200Hz (detector + descriptor, 640×480 img)**
- http://www.vision.ee.ethz.ch/~surf

Many other efficient descriptors are also available

[Bay, ECCV'06], [Cornelis, CVGPU'08]

K. Grauman, B. Leibe

Local Descriptors: Shape Context



Count the number of points inside each bin, e.g.:

Count = 4

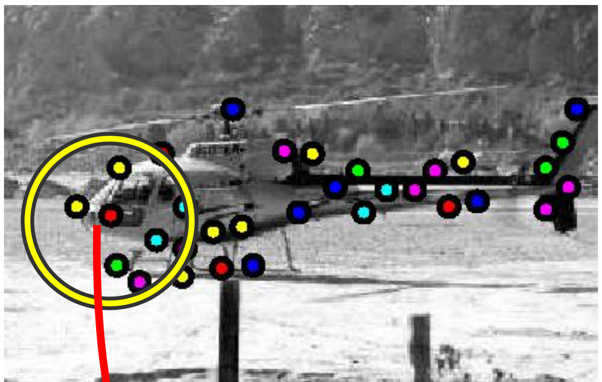
⋮

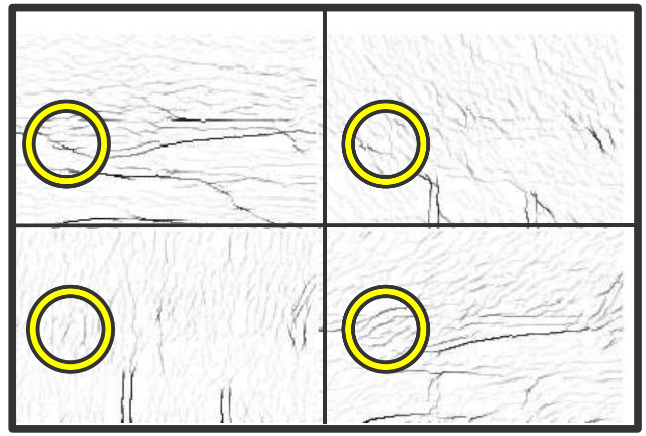
Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Belongie & Malik, ICCV 2001

Local Descriptors: Geometric Blur

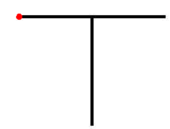




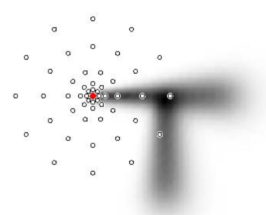
Compute edges at four orientations

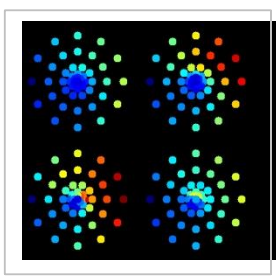
Extract a patch in each channel

Apply spatially varying blur and sub-sample



(Idealized signal)





Example descriptor

Berg & Malik, CVPR 2001

Choosing a detector

- What do you want it for?
  - Precise localization in x-y: Harris
  - Good localization in scale: Difference of Gaussian
  - Flexible region shape: MSER
- Best choice often application dependent
  - Harris-/Hessian-Laplace/DoG work well for many natural categories
  - MSER works well for buildings and printed things
- Why choose?
  - Get more points with more detectors
- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV’05, PAMI’05]
  - All detectors/descriptors shown here work well

Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

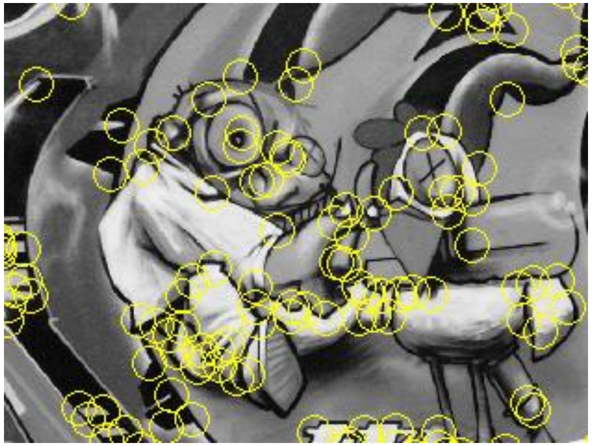
Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Tuytelaars Mikolajczyk 2008

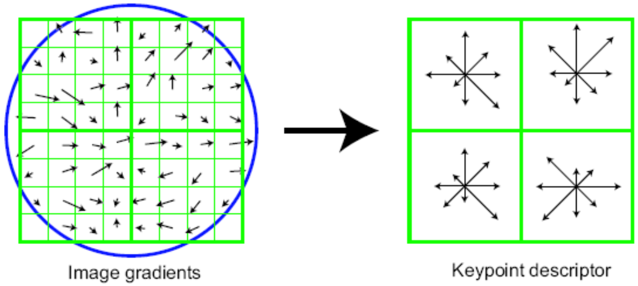
Choosing a descriptor

- Again, need not stick to one
- For object instance recognition or stitching, SIFT or variant is a good choice

Things to remember



- Keypoint detection: repeatable and distinctive
  - Corners, blobs, stable regions
  - Harris, DoG



- Descriptors: robust and selective
  - spatial histograms of orientation
  - SIFT

Next time

- Feature tracking