# Interest Points and Corners

Read Szeliski 4.1

Computer Vision

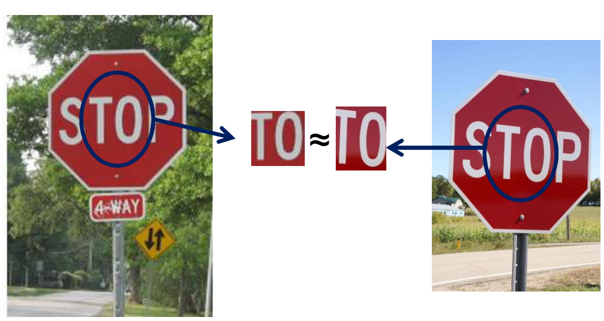
James Hays

Slides from Rick Szeliski, Svetlana Lazebnik, Derek Hoiem and Grauman&Leibe 2008 AAAI Tutorial

1

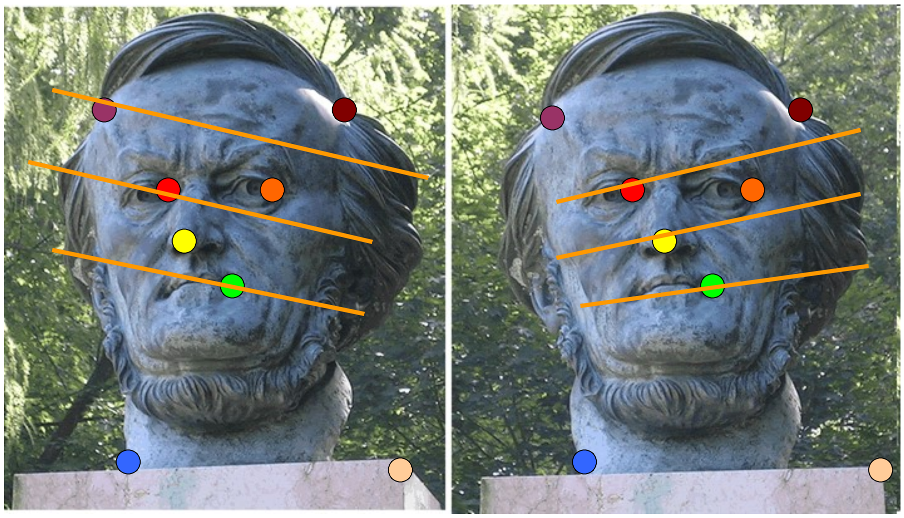
## Correspondence across views

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




2

## Example: estimating "fundamental matrix" that corresponds two views



Slide from Silvio Savarese

3

## Example: structure from motion

4

# Applications

- Feature points are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition

5

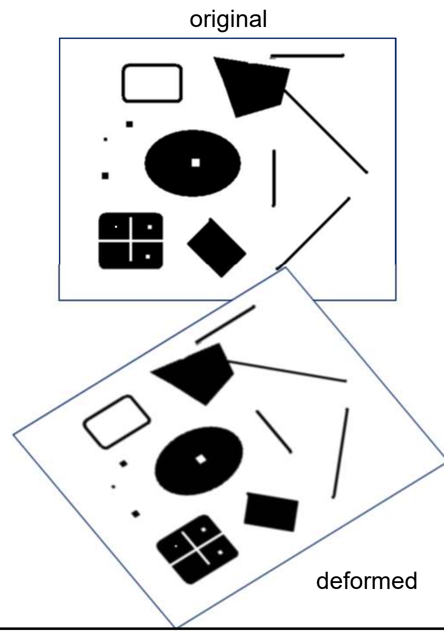
# This class: interest points (continued) and local features

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

6

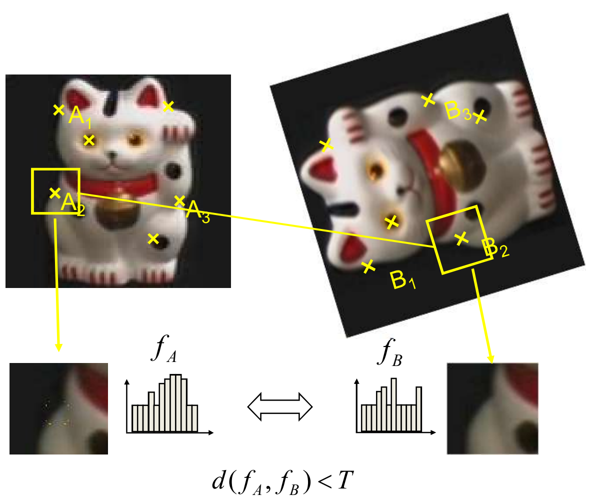
# This class: interest points

- 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?




7

# Overview of Keypoint Matching




**1. Find a set of distinctive key-points**

**2. Define a region around each keypoint**

**3. Compute a local descriptor from the normalized region**

**4. Match local descriptors**

K. Grauman, B. Leibe

8

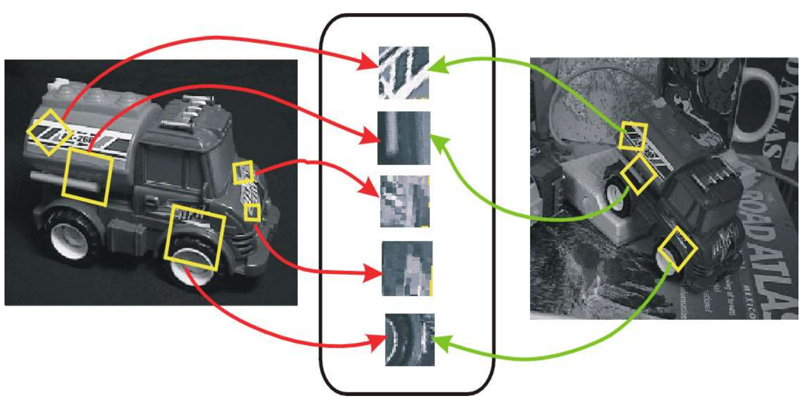
# Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

9

## Invariant Local Features

Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



**Features Descriptors**

10

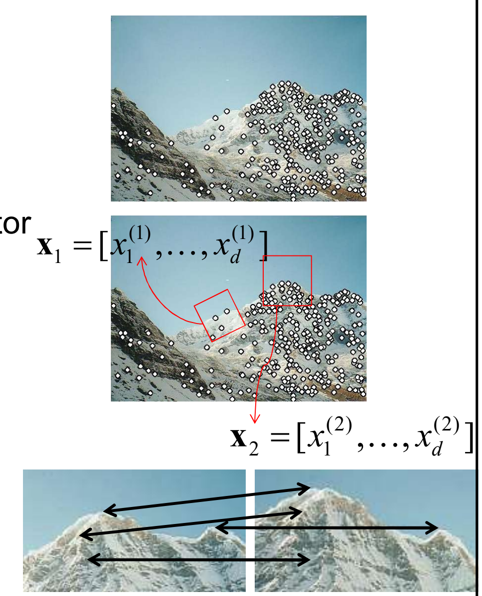
## Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

11

## Local features: main components

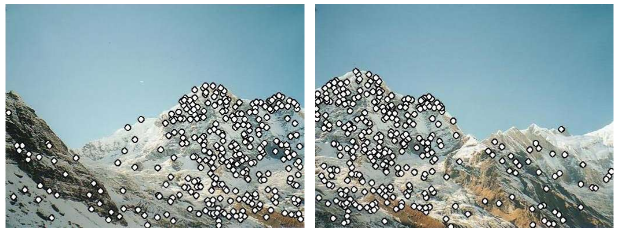
1) **Detection:** Identify the interest points
2) **Description:** Extract vector feature descriptor surrounding each interest point.
3) **Matching:** Determine correspondence between descriptors in two views




Kristen Grauman

12

## Characteristics of good features

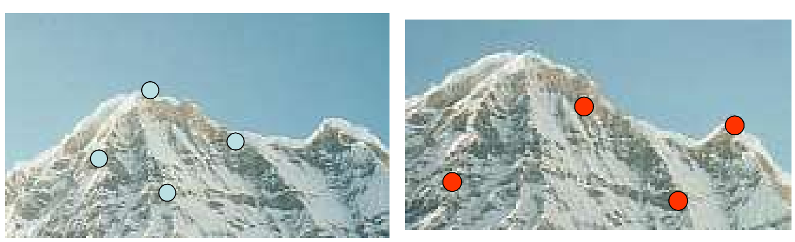


- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature is distinctive
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

13

## Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.



No chance to find true matches!

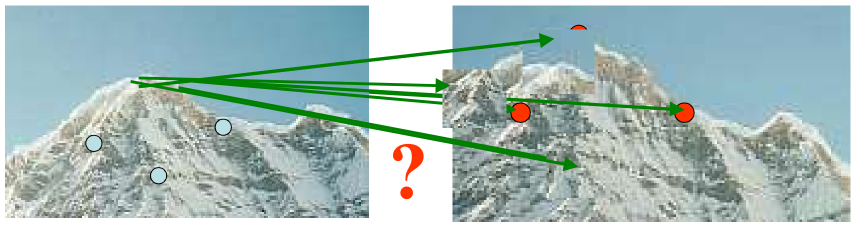
- Yet we have to be able to run the detection procedure *independently* per image.

Kristen Grauman

14

## Goal: descriptor distinctiveness

- We want to be able to reliably determine which point goes with which.



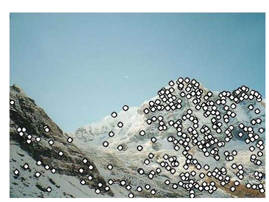

- Must provide some invariance to geometric and photometric differences between the two views.

Kristen Grauman

15

## Local features: main components

1) Detection: Identify the interest points



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

3) Matching: Determine correspondence between descriptors in two views

16

# Many Existing Detectors Available

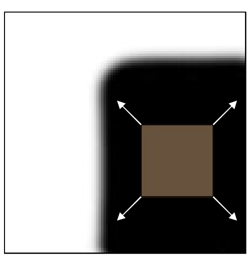
| | |
|---|---|
| Hessian & Harris | [Beaudet '78], [Harris '88] |
| Laplacian, DoG | [Lindeberg '98], [Lowe 1999] |
| Harris-/Hessian-Laplace | [Mikolajczyk & Schmid '01] |
| Harris-/Hessian-Affine | [Mikolajczyk & Schmid '04] |
| EBR and IBR | [Tuytelaars & Van Gool '04] |
| MSER | [Matas '02] |
|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

K. Grauman, B. Leibe

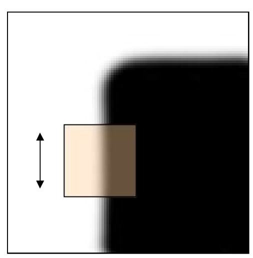
17

## Corner Detection: Basic Idea

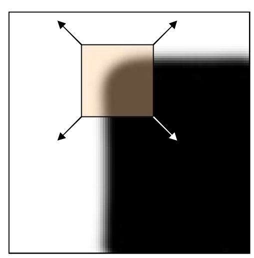
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region: no change in all directions



"edge": no change along the edge direction

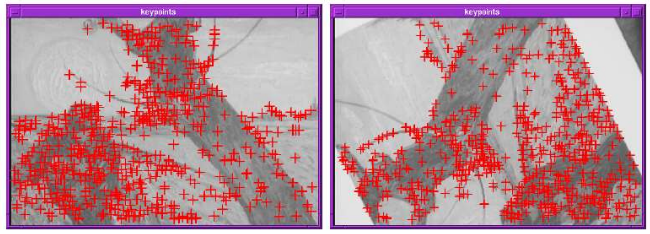


"corner": significant change in all directions

Source: A. Efros

18

## Finding Corners



- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147--151.

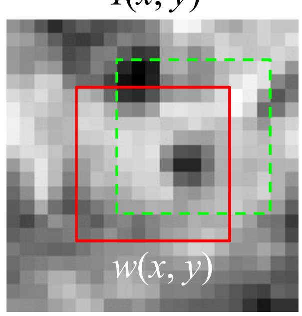
19

## Corner Detection: Mathematics

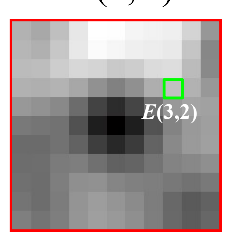
Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

$I(x, y)$




$E(u, v)$



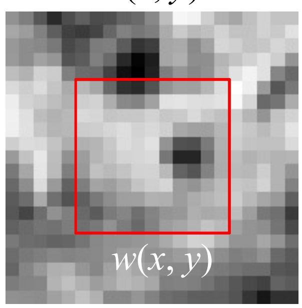

20

## Corner Detection: Mathematics

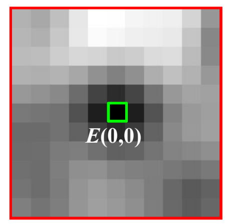
Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

$I(x, y)$




$E(u, v)$



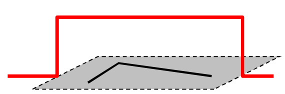

21

## Corner Detection: Mathematics

Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

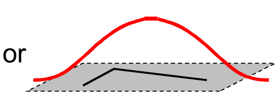
$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{Window function}}\left[\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}} - \underbrace{I(x,y)}_{\text{Intensity}}\right]^2$$

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

Source: R. Szeliski

22

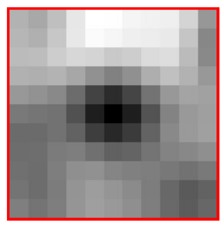
## Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

We want to find out how this function behaves for small shifts

$E(u, v)$



23

## Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

We want to find out how this function behaves for small shifts

But this is very slow to compute naively.
O(window_width$^2$ * shift_range$^2$ * image_width$^2$)

O( $11^2$ * $11^2$ * $600^2$ ) = 5.2 billion of these
14.6 thousand per pixel in your image

24

## Corner Detection: Mathematics

Change in appearance of window *w*(*x*,*y*)
for the shift [*u*,*v*]:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

We want to find out how this function behaves for small shifts

Recall Taylor series expansion. A function f can be approximated at point a as

$$f(a)+\frac {f'(a)}{1!} (x-a)+ \frac{f''(a)}{2!} (x-a)^2+\frac{f'''(a)}{3!}(x-a)^3+ \cdots.$$

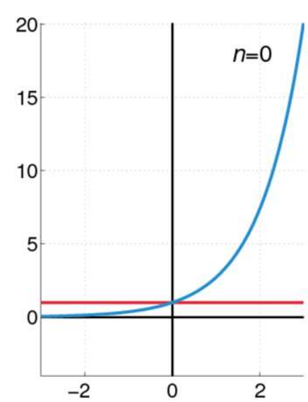
25

## Recall: Taylor series expansion

A function f can be approximated as

$$f(a)+\frac {f'(a)}{1!} (x-a)+ \frac{f''(a)}{2!} (x-a)^2+\frac{f'''(a)}{3!}(x-a)^3+ \cdots.$$

Approximation of
f(x) = $e^x$
centered at f(0)




26

Robert Collins
CSE486, Penn State

# Taylor Series for 2D Functions

$$f(x+u,y+v) = f(x,y) + uf_x(x,y) + vf_y(x,y) +$$

**First partial derivatives**

$$\frac{1}{2!}\left[u^2 f_{xx}(x,y) + uv f_{xy} x,y + v^2 f_{yy}(x,y)\right] +$$

**Second partial derivatives**

$$\frac{1}{3!}\left[u^3 f_{xxx}(x,y) + u^2 v f_{xxy}(x,y) + uv^2 f_{xyy}(x,y) + v^3 f_{yyy}(x,y)\right]$$

**Third partial derivatives**

$+ \ldots$ (Higher order terms)

First order approx

$$f(x+u,y+v) \approx f(x,y) + uf_x(x,y) + vf_y(x,y)$$

27

Robert Collins
CSE486, Penn State

# Harris Corner Derivation

$$\sum [I(x+u,y+v) - I(x,y)]^2$$

$$\approx \sum [I(x,y) + uI_x + vI_y - I(x,y)]^2$$ First order approx

$$= \sum u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2$$

$$= \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$ Rewrite as matrix equation

$$= \begin{bmatrix} u & v \end{bmatrix} \left( \sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

28

Robert Collins
CSE486, Penn State

# Harris Detector: Mathematics

For small shifts $[u,v]$ we have a *bilinear* approximation:

$$E(u,v) \cong [u,v] \quad M \begin{bmatrix} u \\ v \end{bmatrix}$$

where $M$ is a 2×2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

**Windowing function - computing a weighted sum (simplest case, w=1)**

**Note: these are just products of components of the gradient, Ix, Iy**

29

Robert Collins
CSE486, Penn State

# Intuitive Way to Understand Harris

Treat gradient vectors as a set of (dx,dy) points with a center of mass defined as being at (0,0).

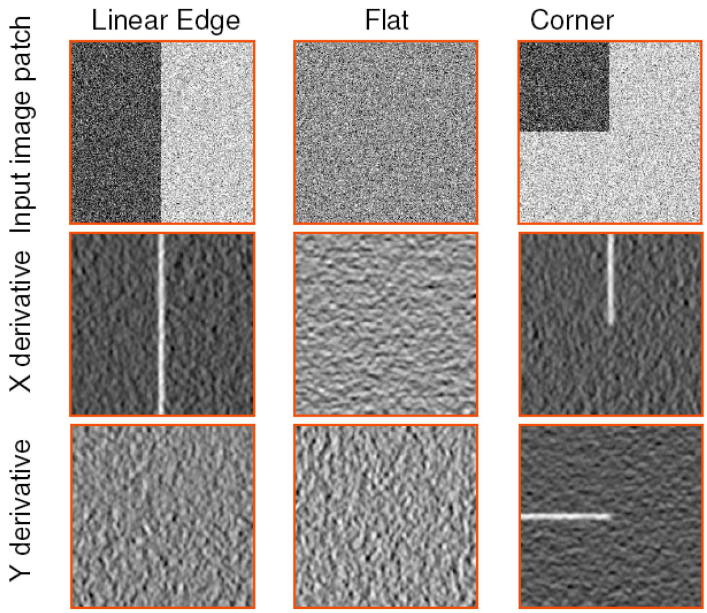
Fit an ellipse to that set of points via scatter matrix

Analyze ellipse parameters for varying cases…

30

Robert Collins
CSE486, Penn State

# Example: Cases and 2D Derivatives

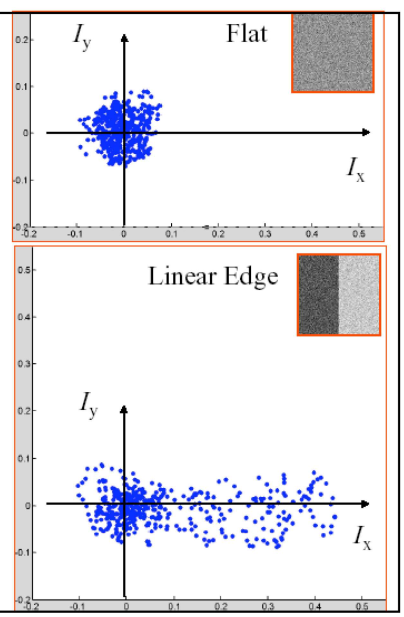



31

Robert Collins
CSE486, Penn State

# Plotting Derivatives as 2D Points

The distribution of the $x$ and $y$ derivatives is very different for all three types of patches

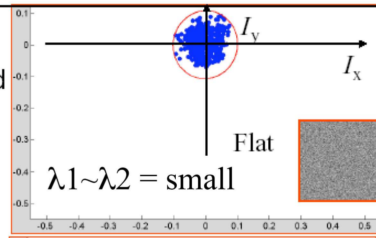


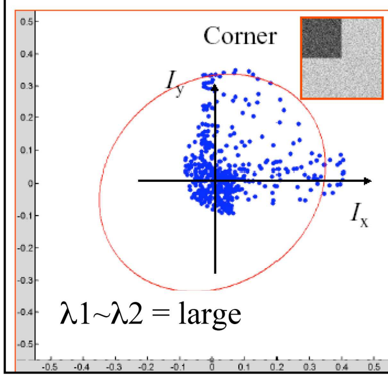

32

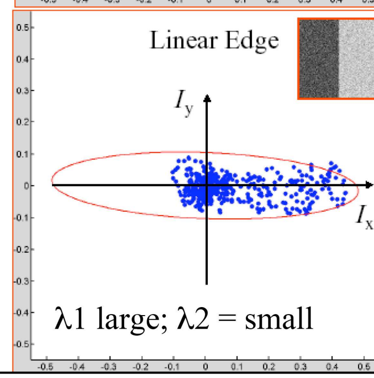
Robert Collins
CSE486, Penn State

## Fitting Ellipse to each Set of Points

The distribution of $x$ and $y$ derivatives can be characterized by the shape and size of the principal component ellipse





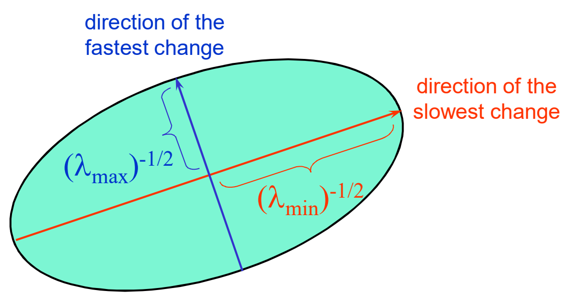

## Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

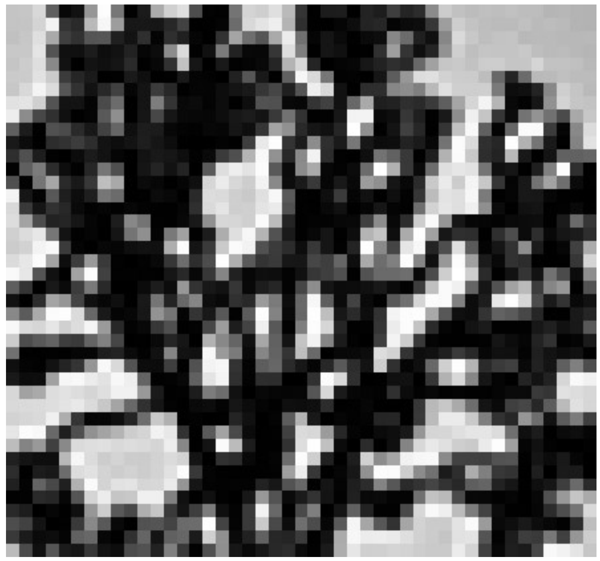
This is the equation of an ellipse.

Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by $R$

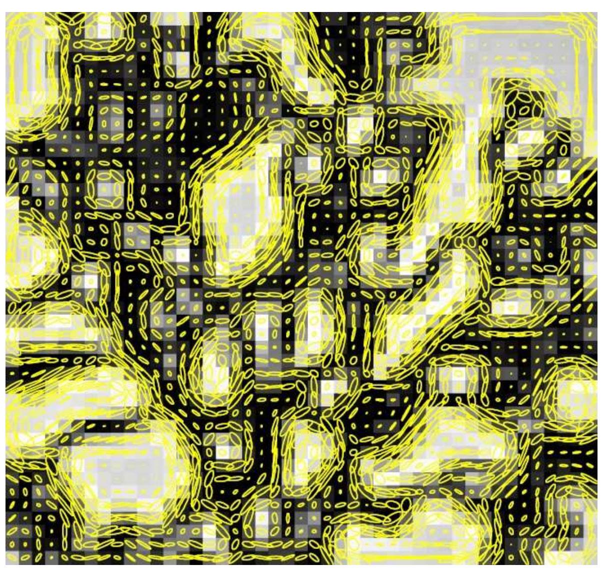


## Visualization of second moment matrices



35

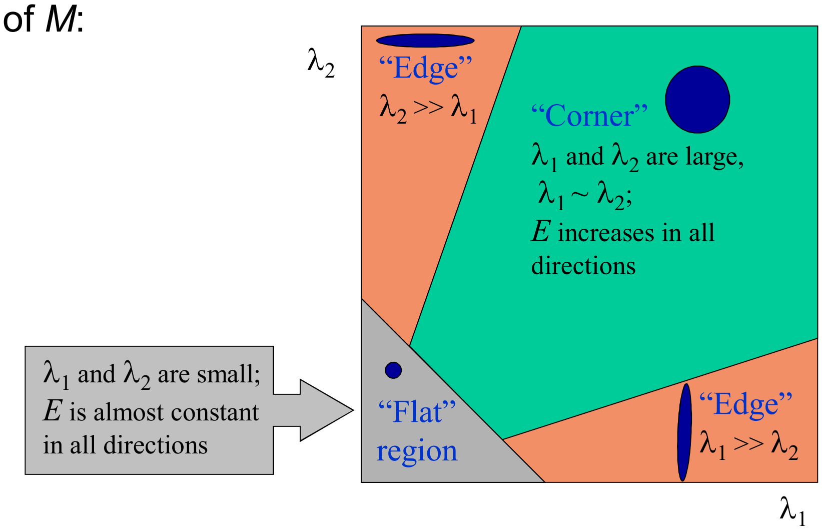
## Visualization of second moment matrices



36

## Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:

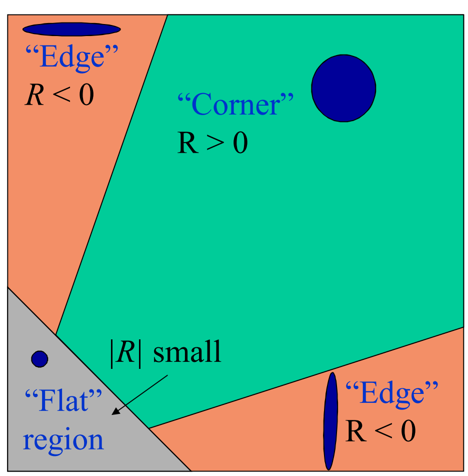



37

## Corner response function

$$R = \det(M) - \alpha \operatorname{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

$\alpha$: constant (0.04 to 0.06)




38

## Harris corner detector

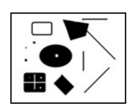
1) Compute *M* matrix for each image window to get their *cornerness* scores.
2) Find points whose surrounding window gave large corner response (*f*> threshold)
3) Take the points of local maxima, i.e., perform non-maximum suppression

C.Harris and M.Stephens. [“A Combined Corner and Edge Detector.”]() *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

39

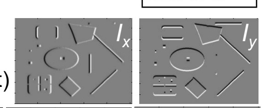
## Harris Detector [Harris88]

- Second moment matrix



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

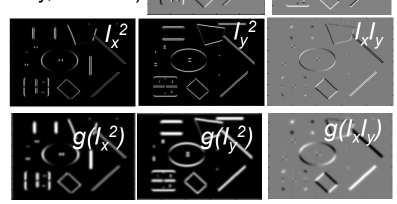
1. Image derivatives (optionally, blur first)




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

2. Square of derivatives

3. Gaussian filter $g(\sigma_I)$




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

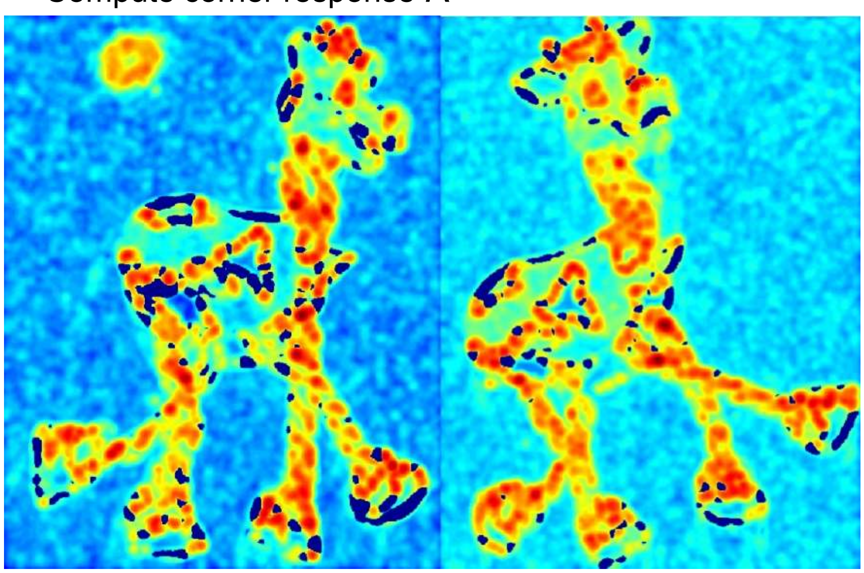

40

## Harris Detector: Steps

41

## Harris Detector: Steps

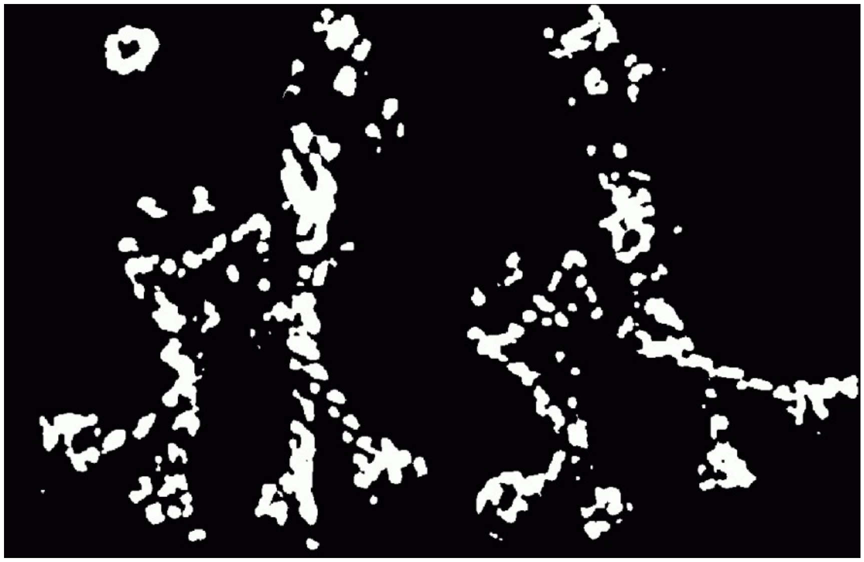
Compute corner response $R$



42

## Harris Detector: Steps

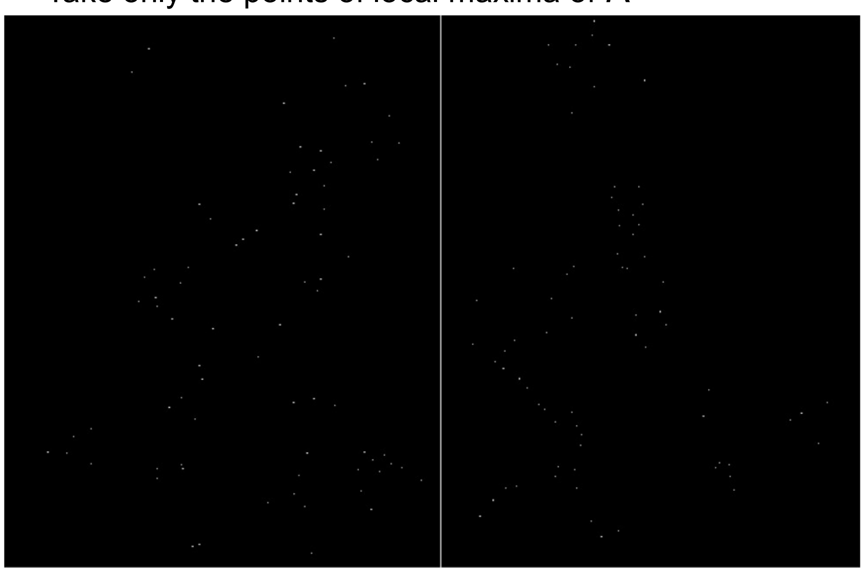
Find points with large corner response: $R$>threshold



43

## Harris Detector: Steps

Take only the points of local maxima of $R$



44

## Harris Detector: Steps

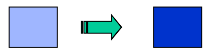
45

## Invariance and covariance

- We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations
  - **Invariance:** image is transformed and corner locations do not change
  - **Covariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations

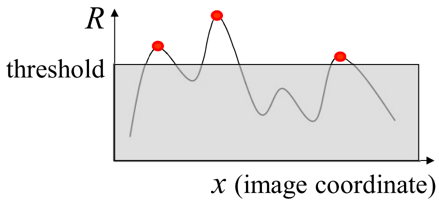
46

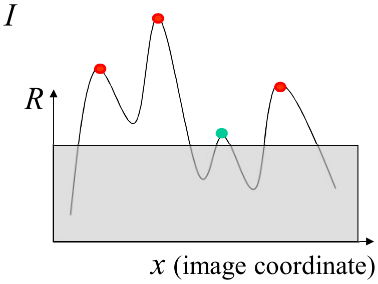
## Affine intensity change



$I \rightarrow a\,I + b$

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scaling: $I \rightarrow a\,I$

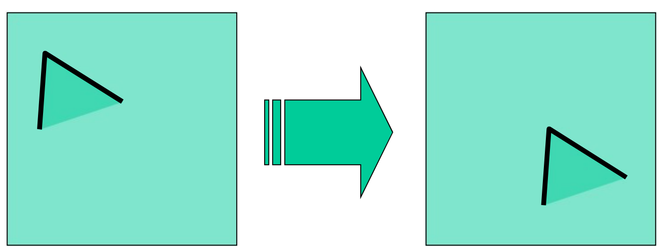



*Partially invariant* to affine intensity change

47

## Image translation

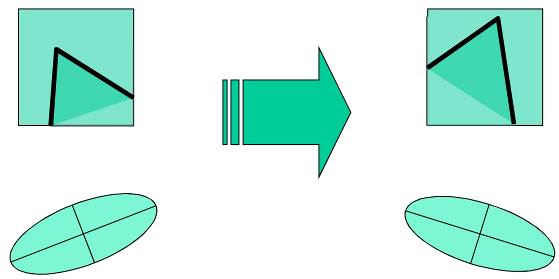


- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation

48

## Image rotation

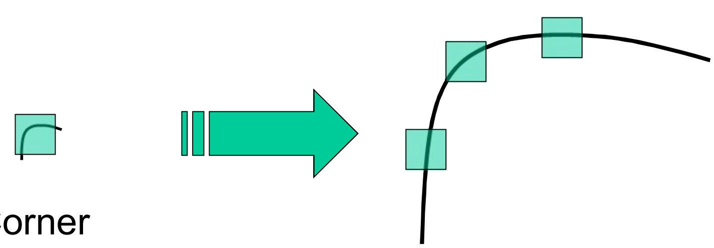


Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation

49

## Scaling



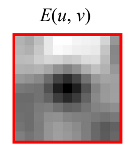
Corner

All points will be classified as edges

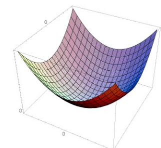
Corner location is not covariant to scaling!

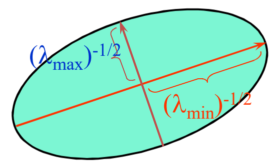
50

# Review: Harris corner detector




- Approximate distinctiveness by local auto-correlation.
- Approximate local auto-correlation by second moment matrix
- Quantify distinctiveness (or cornerness) as function of the eigenvalues of the second moment matrix.
- But we don't actually need to compute the eigenvalues by using the determinant and trace of the second moment matrix.





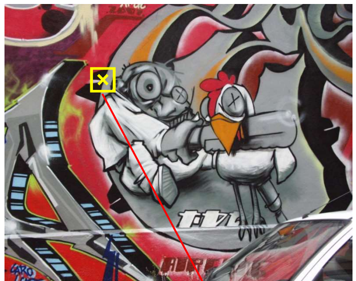

- Video chess https://youtu.be/vkWdzWeRfC4

51

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

52

# Automatic Scale Selection



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

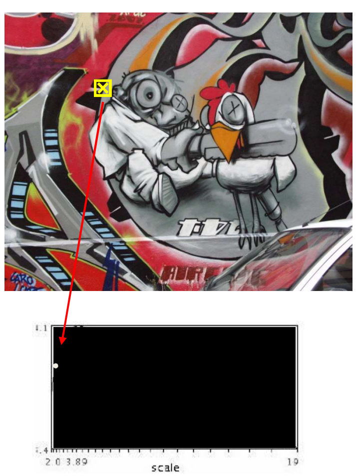
How to find corresponding patch sizes?

K. Grauman, B. Leibe

53

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)




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

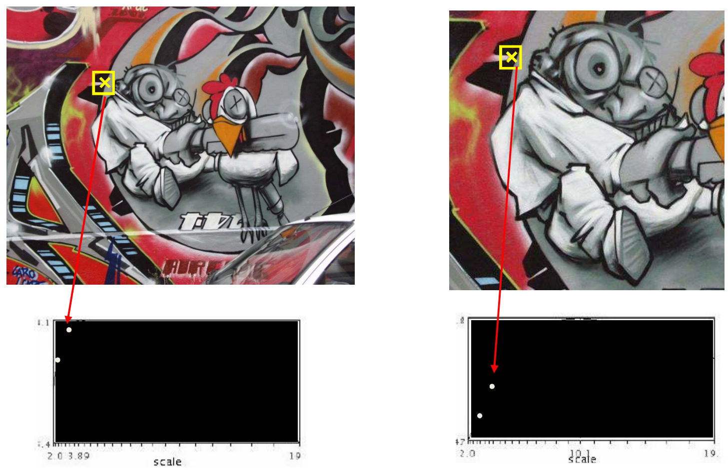

$f(I_{i_1 \ldots i_m}(x', \sigma))$

K. Grauman, B. Leibe

54

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)



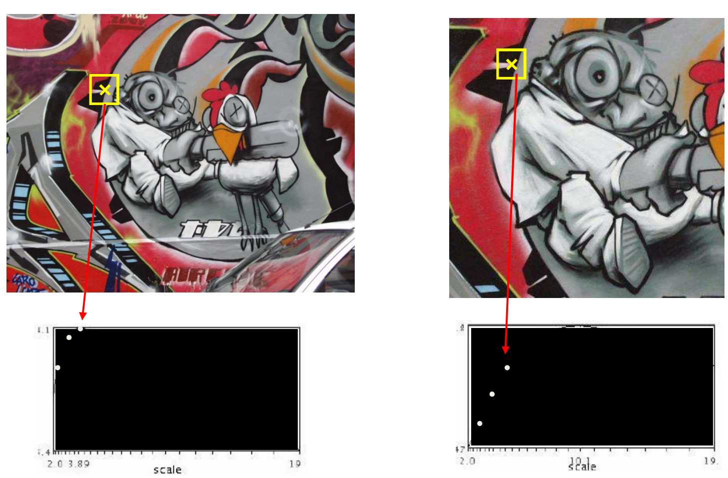

$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

55

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)



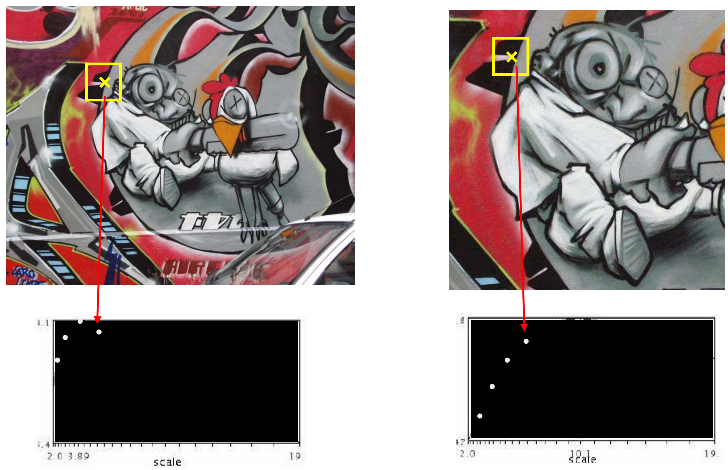

$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

56

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)



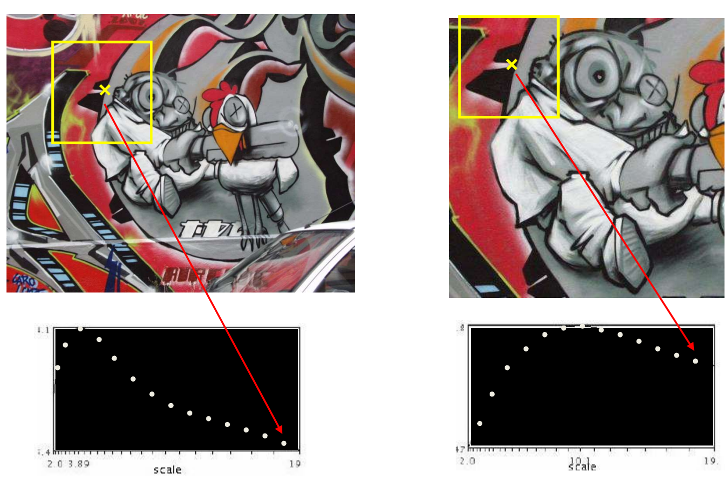

$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

K. Grauman, B. Leibe

57

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)



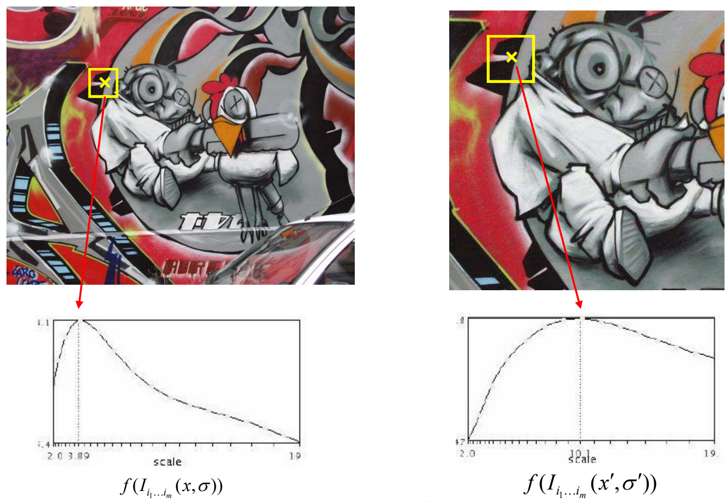

$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

K. Grauman, B. Leibe

58

# Automatic Scale Selection

- Function responses for increasing scale (scale signature)

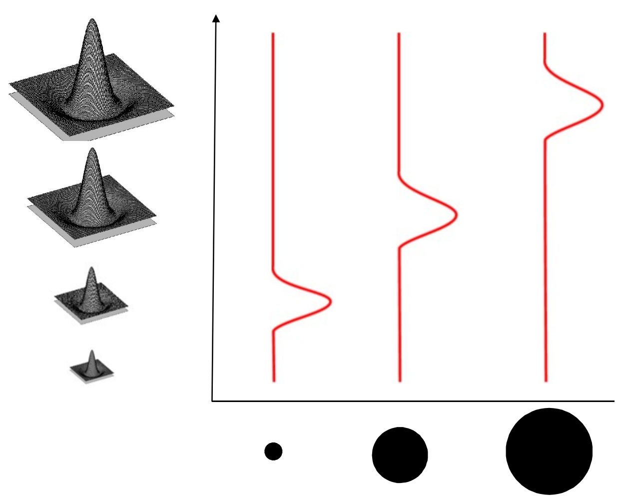



K. Grauman, B. Leibe

59

# What Is A Useful Signature Function?

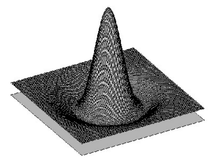
- Difference-of-Gaussian = "blob" detector

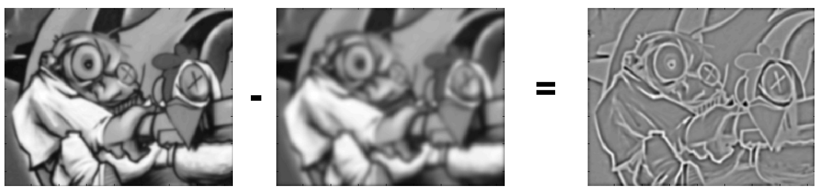


K. Grauman, B. Leibe

60

# Difference-of-Gaussian (DoG)



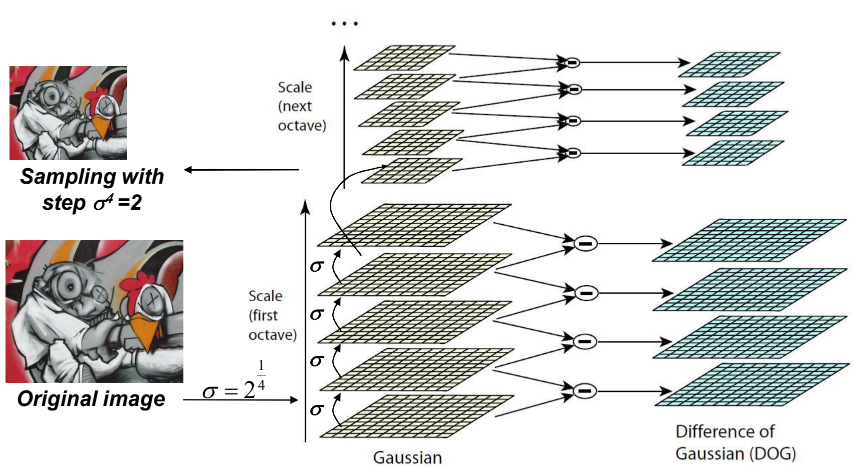



K. Grauman, B. Leibe

61

# DoG – Efficient Computation

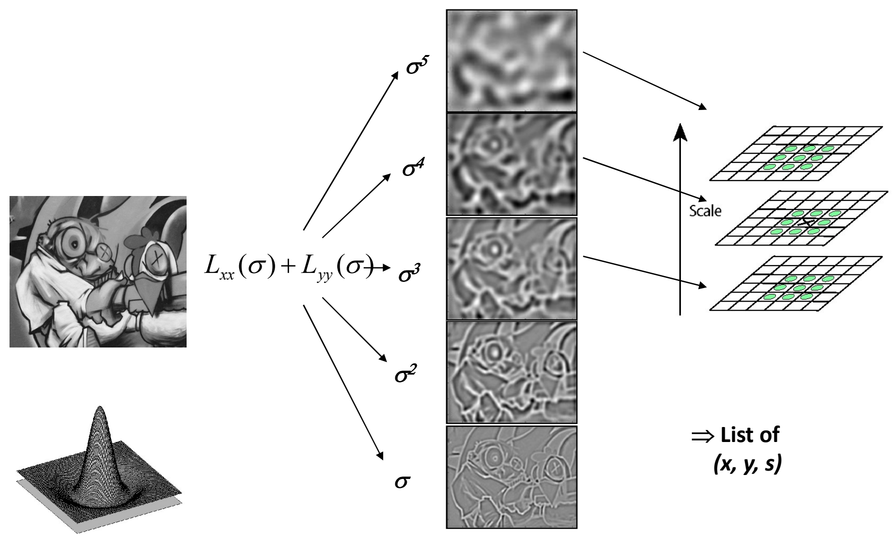
- Computation in Gaussian scale pyramid




K. Grauman, B. Leibe

62

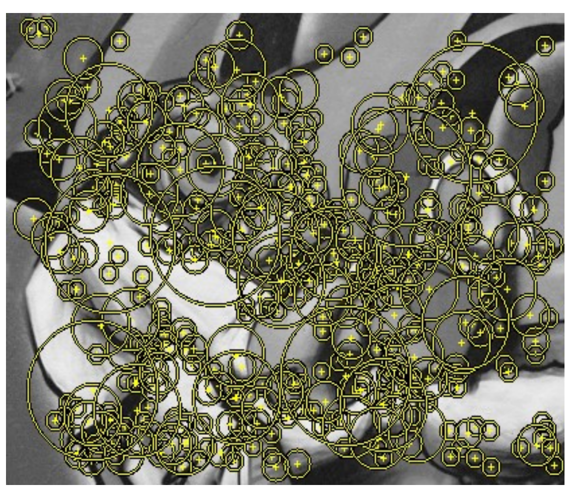
# Find local maxima in position-scale space of Difference-of-Gaussian




K. Grauman, B. Leibe

63

# Results: Difference-of-Gaussian



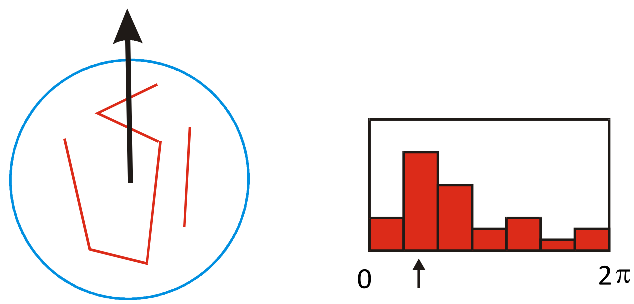
K. Grauman, B. Leibe

64

# Orientation Normalization

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

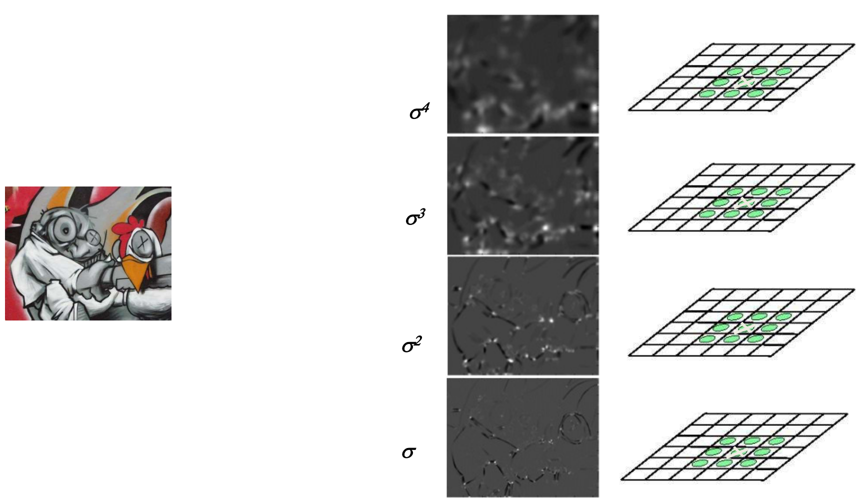
[Lowe, SIFT, 1999]




65

# Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection

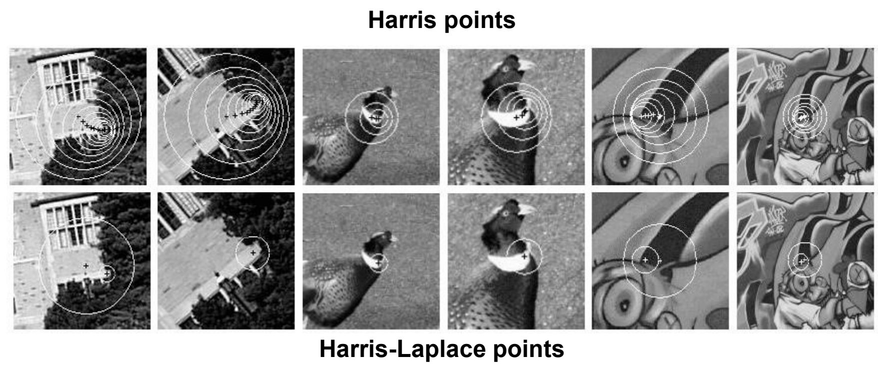



**Computing Harris function** **Detecting local maxima**

66

## Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian
   (same procedure with Hessian $\Rightarrow$ Hessian-Laplace)

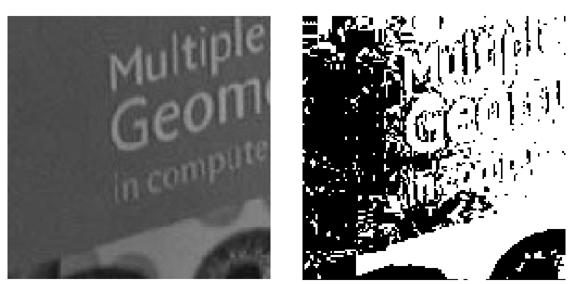



K. Grauman, B. Leibe

67

## 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

68

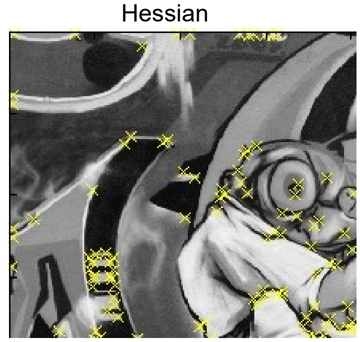
# Example Results: MSER

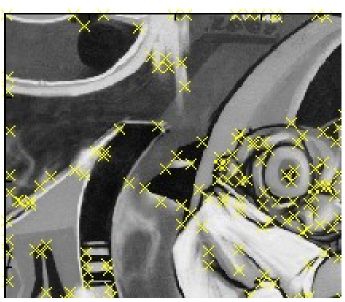



69

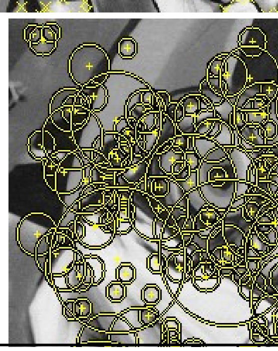
# Comparison

Hessian



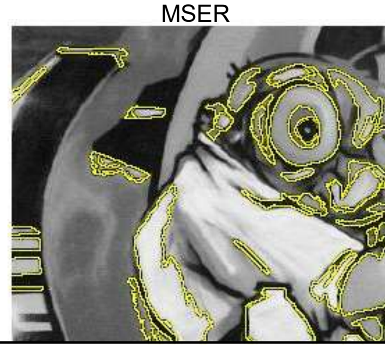


Harris



LoG

MSER



70

# 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

K. Grauman, B. Leibe

71