Computer Vision
- Feature Detection

Junjie Cao @ DLUT

Spring 2019

Image matching: a challenging problem

- keypoint features or interest points (or even **corners**):
  - mountain peaks
  - building corners
  - Doorways
  - interestingly shaped patches of snow

- **Edge features**:
  - profile of mountains against the sky.

  - local appearance
  - Orientation

  - straight line segments => vanishing points => camera parameters

- **Two pairs of images to be matched. What kinds of feature might one use to establish a set of *correspondences* between these images?**

Image matching: a challenging problem

by Diva Sian

by swashford

Slightly different color and viewpoint, but still easy

Slide credit: Steve Seitz

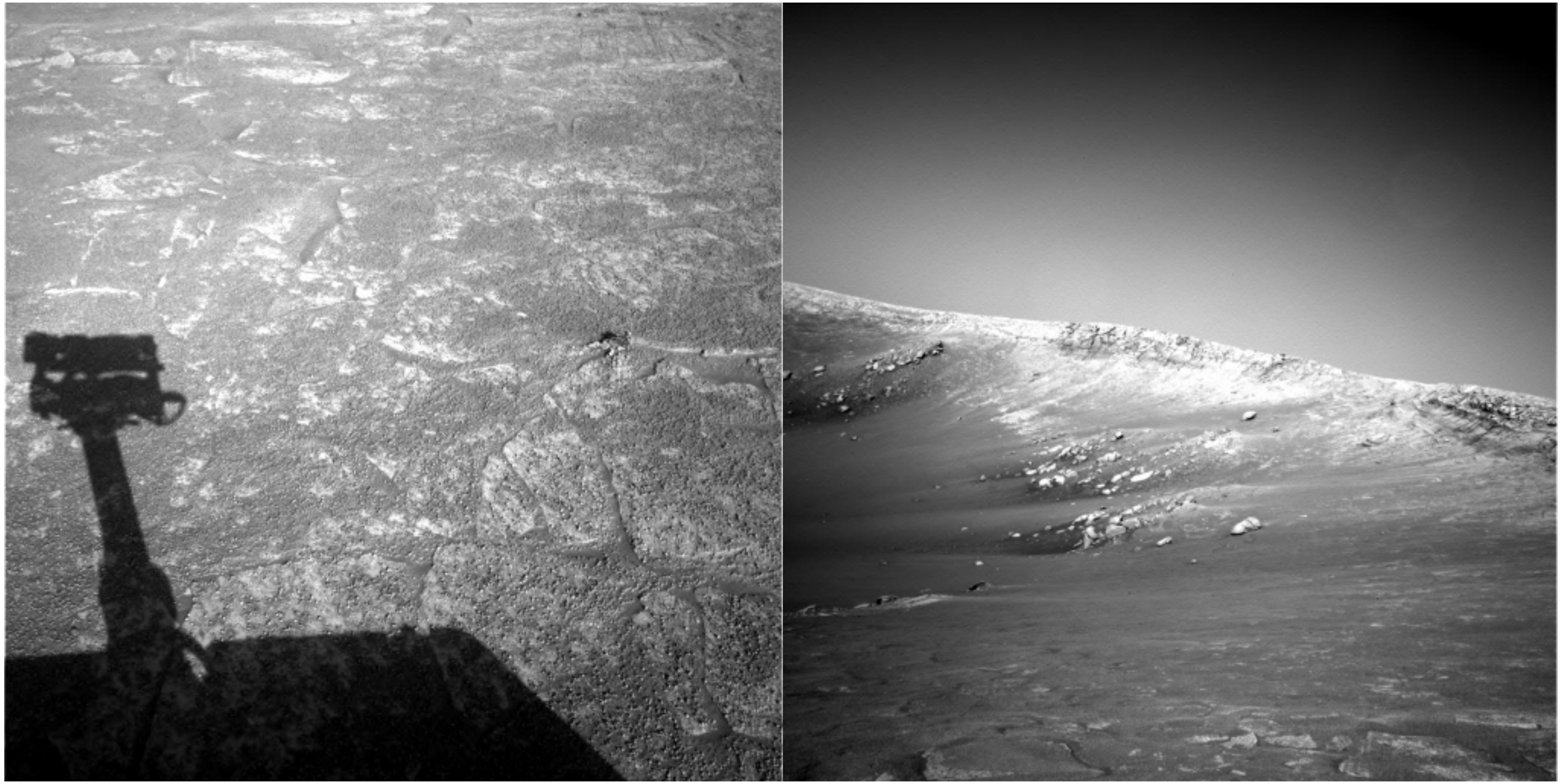
Harder Case

by Diva Sian

by scgbt

Slide credit: Steve Seitz

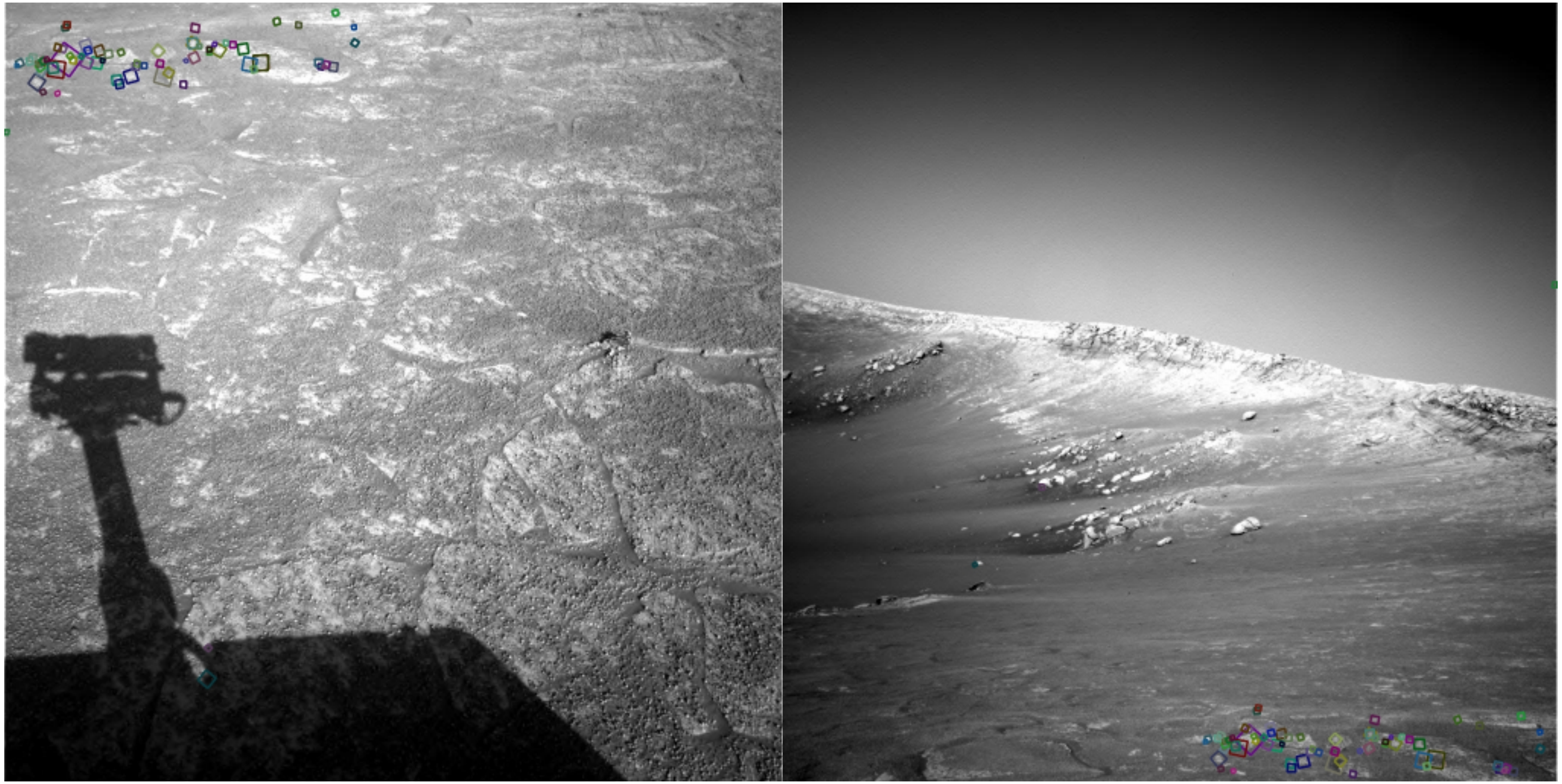
Harder Still?



NASA Mars Rover images

Slide credit: Steve Seitz

Answer Below (Look for tiny colored squares)

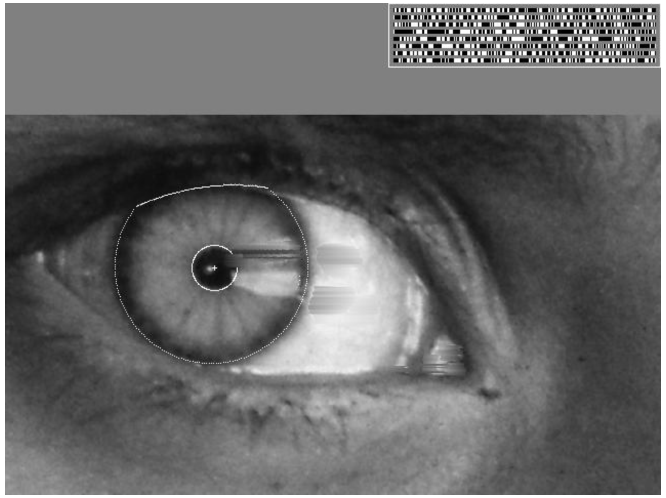


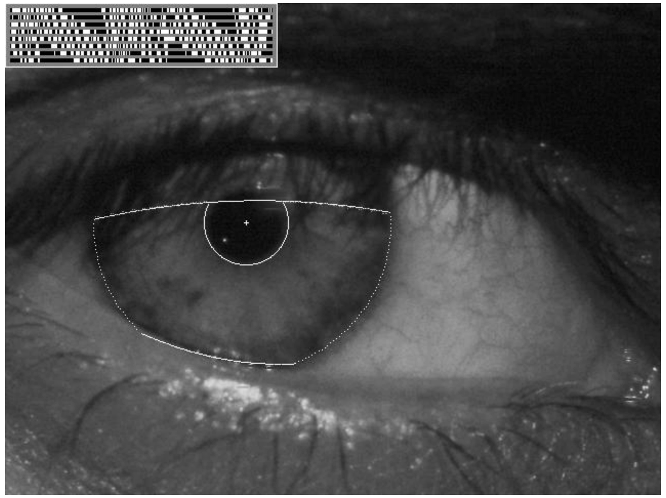
NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)

Slide credit: Steve Seitz

Even harder case

"How the Afghan Girl was Identified by Her Iris Patterns" **Read the [story]()**





What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector

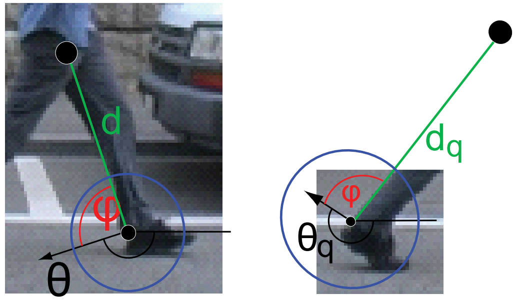
Some background reading:
Rick Szeliski, Chapter 4.1;

- (optional) K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10):1615-1630
  - http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_

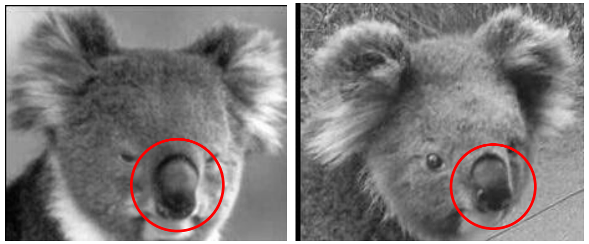
Motivation for using local features

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

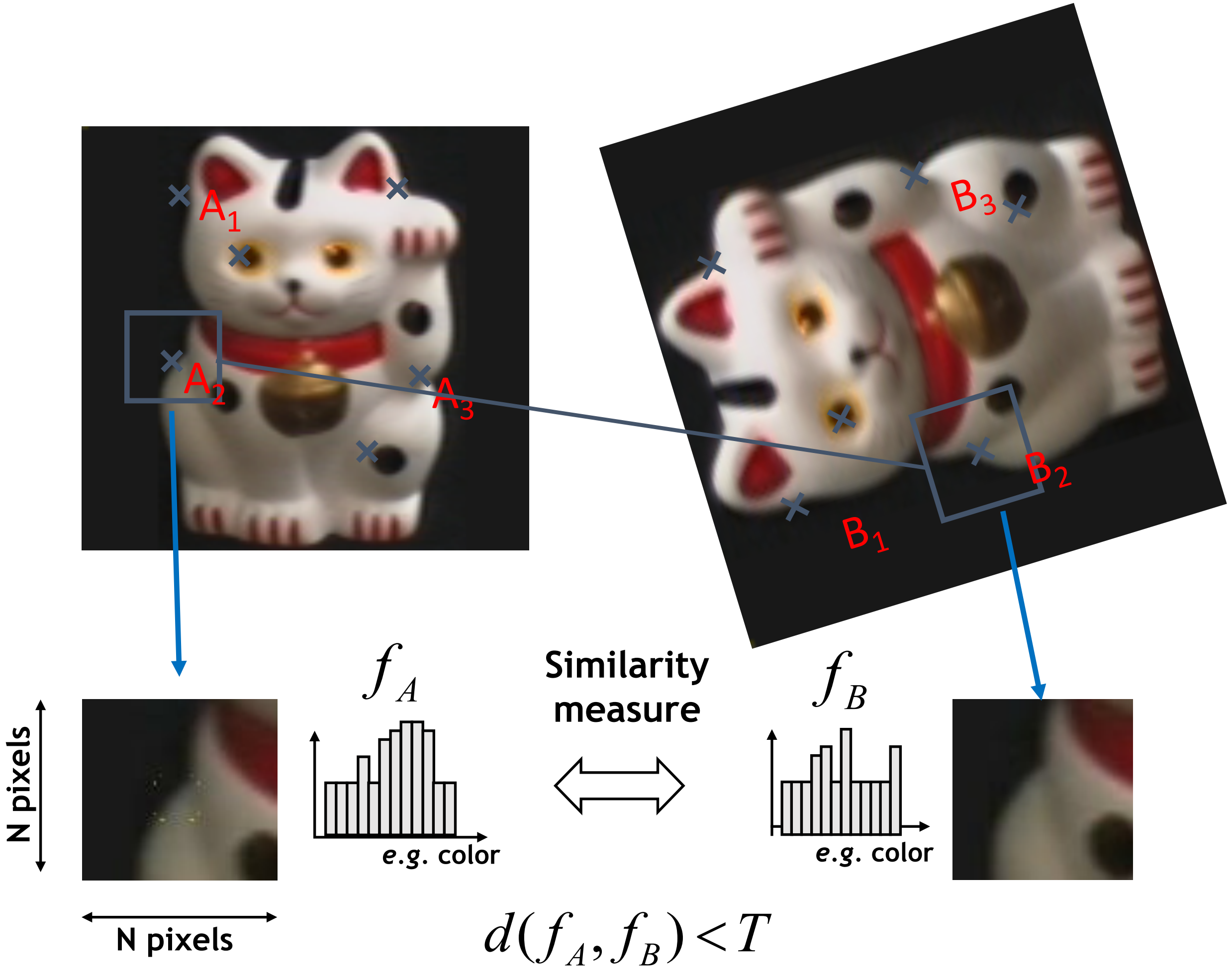
  - Articulation



  - Intra-category variations



General Approach



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

Slide credit: Bastian Leibe

Common Requirements: **independently & Repeatability**

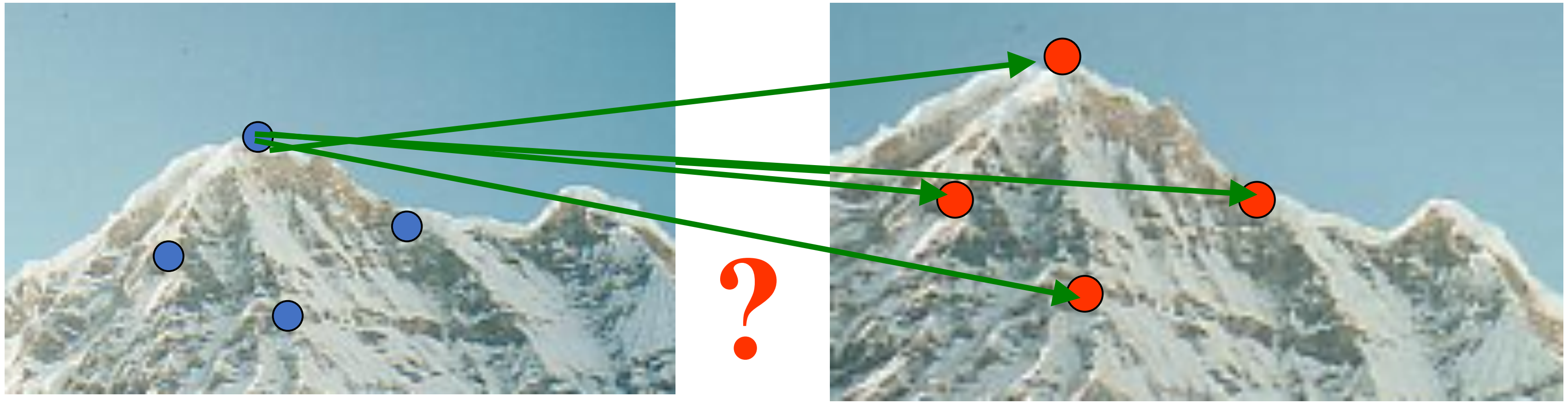
- Yet we have to be able to run the detection procedure ***independently*** per image.

No chance to find true matches!

- We need a **repeatable** detector!
  - We want to detect (at least **some** of) the same points in both images.

Common Requirements: **descriptor distinctiveness**

- We need a reliable and distinctive descriptor!
  - be able to **reliably** determine which point goes with which.

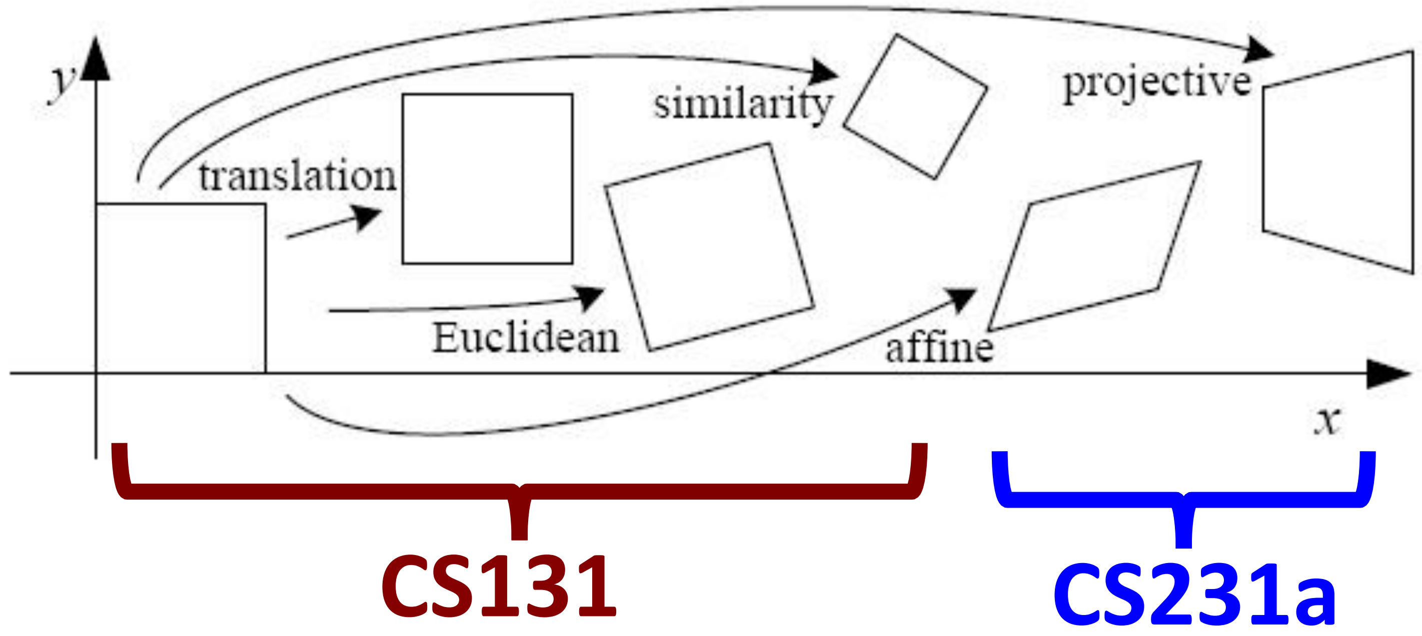


- Must provide some **invariance** to geometric and photometric differences between the two views.

Invariance: Geometric Transformations

Slide credit: Steve Seitz

Levels of Geometric Invariance



Slide credit: Bastian Leibe

Invariance: Photometric Transformations

- Often modeled as a linear transformation:
  - Scaling + Offset

Slide credit: Tinne Tuytelaars

Requirements

- Region extraction needs to be repeatable and accurate
  - Invariant to translation, rotation, scale changes
  - Robust or covariant to out-of-plane (≈affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- Locality: Features are local, therefore robust to occlusion and clutter.
- Quantity: We need a sufficient number of regions to cover the object.
- Distinctivenes : The regions should contain "interesting" structure.
- Efficiency: Close to real-time performance.

Slide credit: Bastian Leibe

Many Existing Detectors Available

• Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
• Laplacian, DoG	[Lindeberg ‘98], [Lowe ‘99]
• 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…	

- *Those detectors have become a basic building block for many recent applications in Computer Vision.*

Slide credit: Bastian Leibe

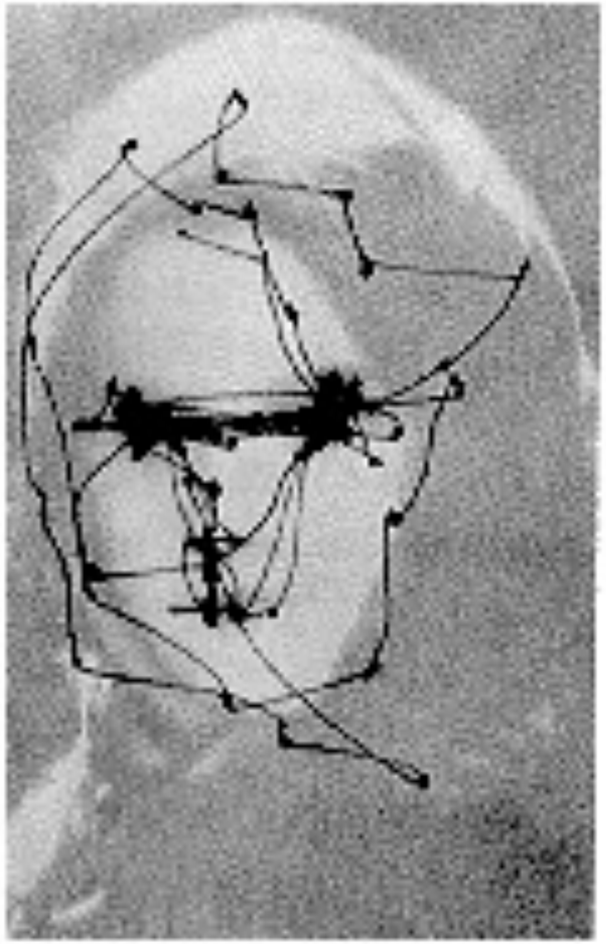
What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector

Some background reading:
Rick Szeliski, Chapter 4.1;

- (optional) K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10):1615-1630
  - http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_

Human eye movements



Yarbus eye tracking

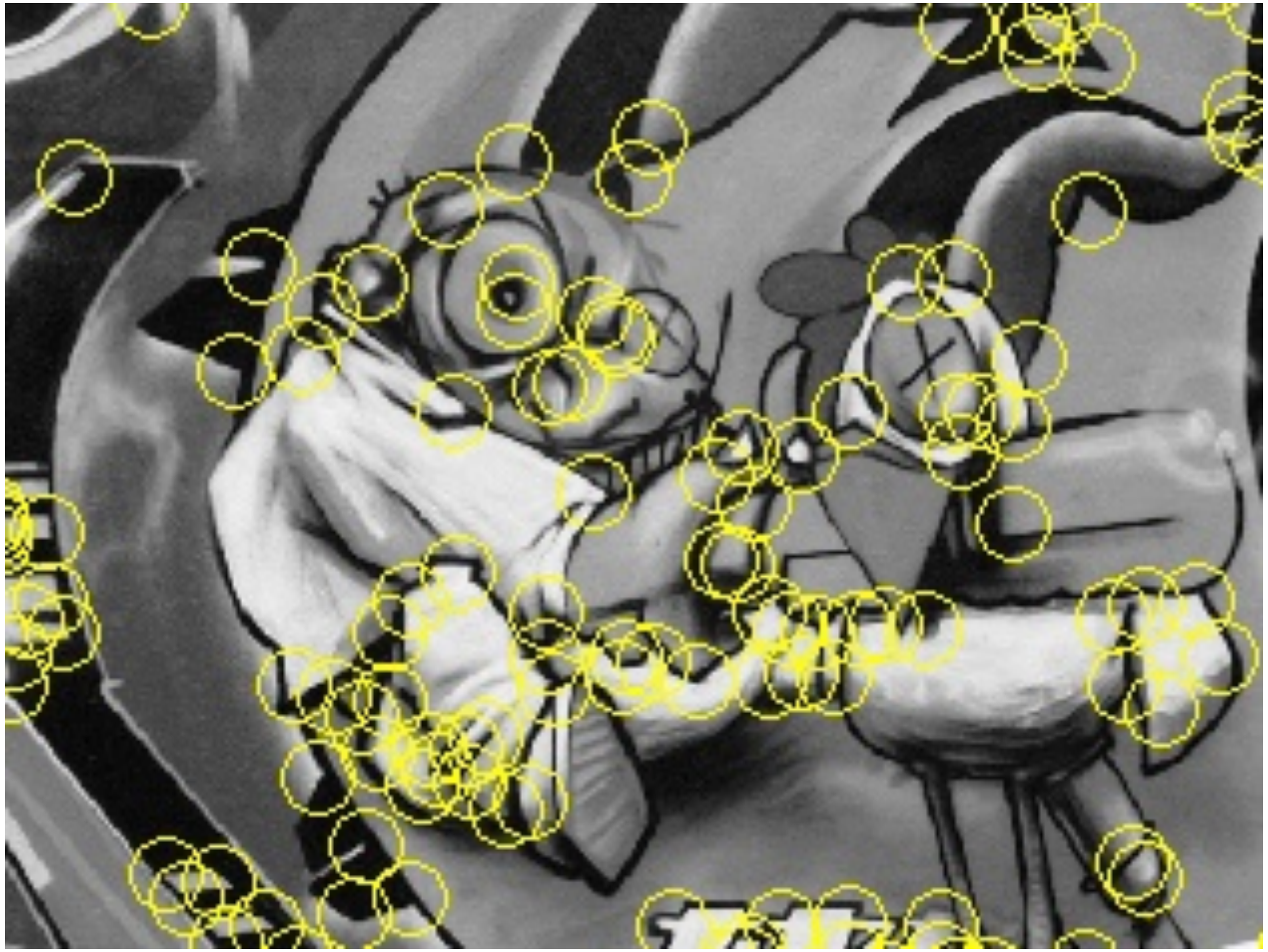
What catches your interest?

Fundamental to Applications

- Feature points are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking (robots, drones, AR)
  - Indexing and database retrieval
  - Object recognition
  - …

Hays

Keypoint Localization

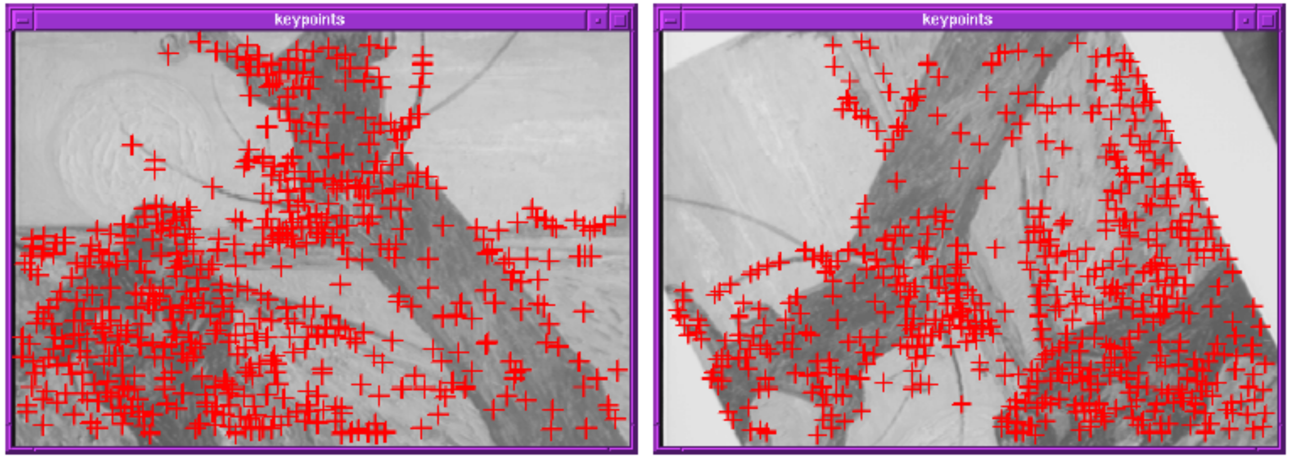


- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

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

Slide credit: Bastian Leibe

Finding Corners

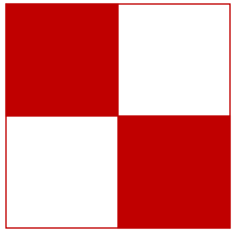


- Key property:
  - In the region around a corner, image gradient has two or more dominant directions
- Corners are *repeatable* and *distinctive*

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*, 1988.

Slide credit: Svetlana Lazebnik

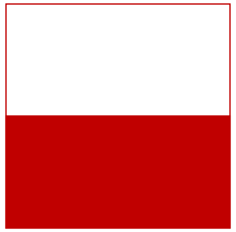
Corners versus edges



$\sum I_x^2 \longrightarrow$ Large

$\sum I_y^2 \longrightarrow$ Large

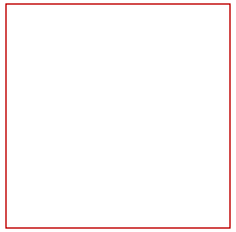
Corner



$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Large

Edge

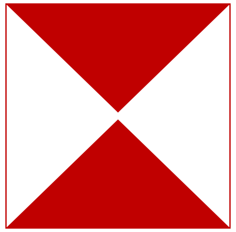


$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Small

Nothing

Corners versus edges



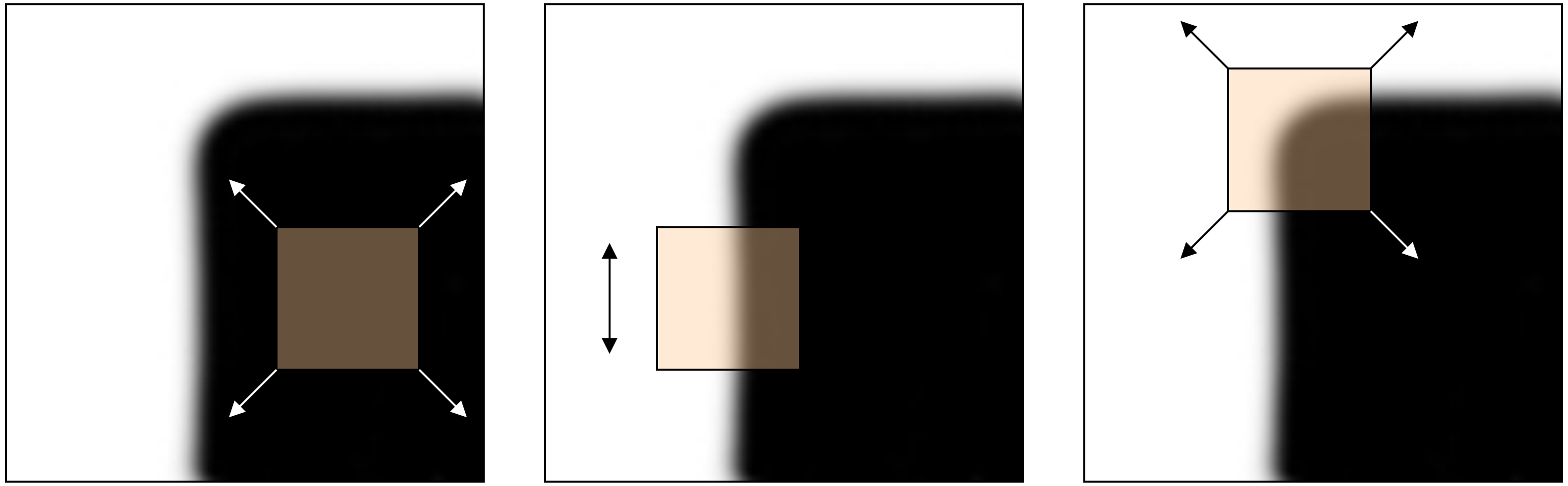
$$\sum I_x^2 \longrightarrow ??$$

$$\sum I_y^2 \longrightarrow ??$$

Corner

Corners as Distinctive Interest Points

- Design criteria
    - We should easily recognize the point by looking through a small window (*locality*)
    - Shifting the window in *any direction* should give *a large change* in intensity (*good localization*)



"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Slide credit: Alyosha Efros

Harris Detector Formulation

- Change of intensity for the shift [u,v]:

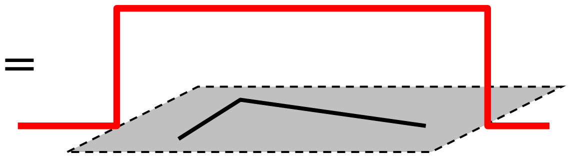
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

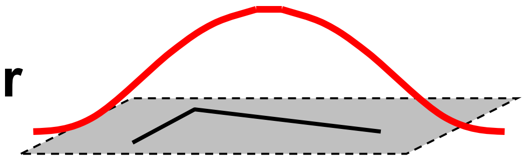
Shifted intensity

Intensity

Window function $w(x,y) =$



or



1 in window, 0 outside

Gaussian

Slide credit: Rick Szeliski

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

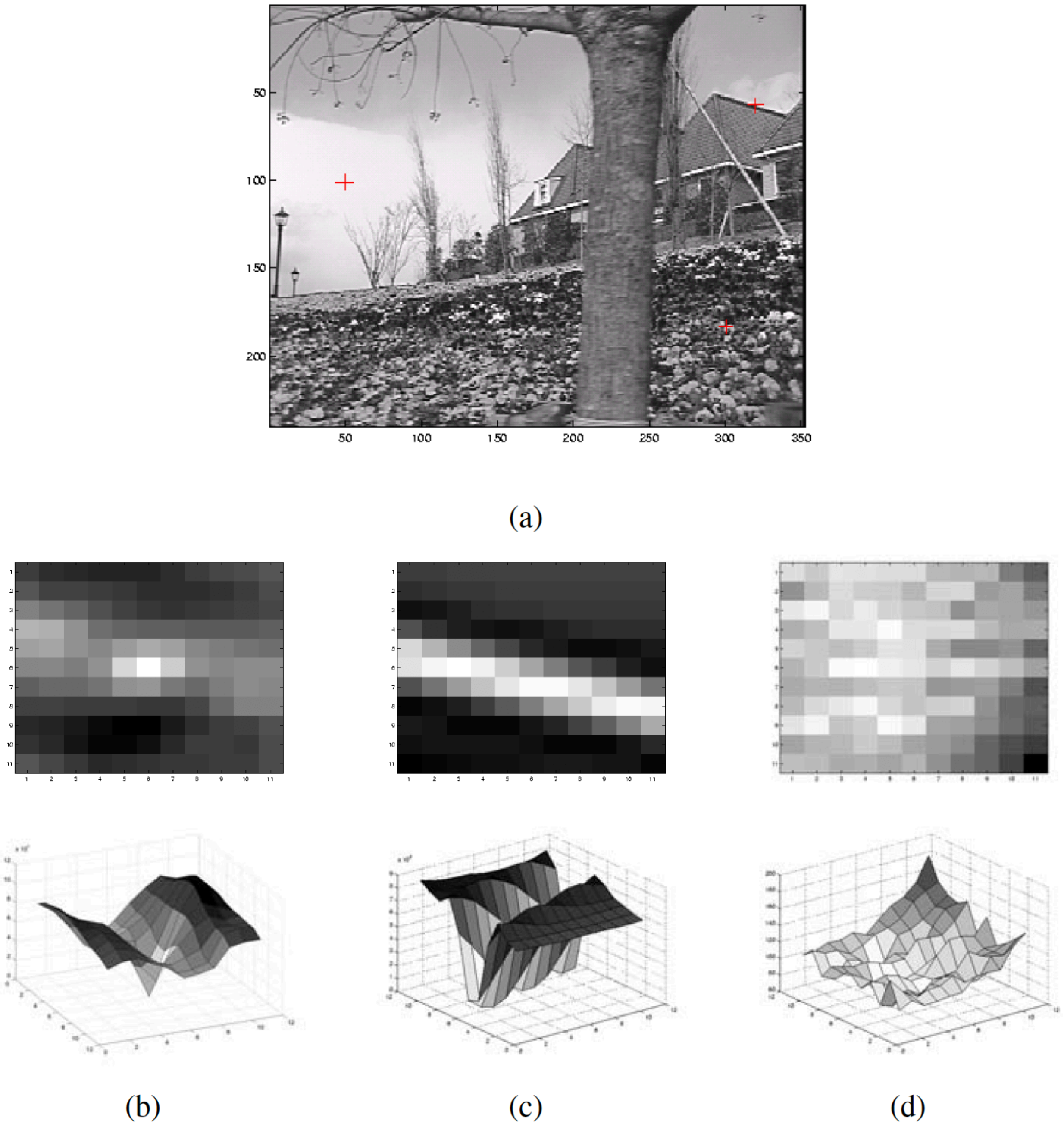


Figure 4.5 Three auto-correlation surfaces E_AC(u) shown as both grayscale images and surface plots:

Corner Detection by Auto-correlation

Change in appearance of window $w(x,y)$ for 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 discover how E 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

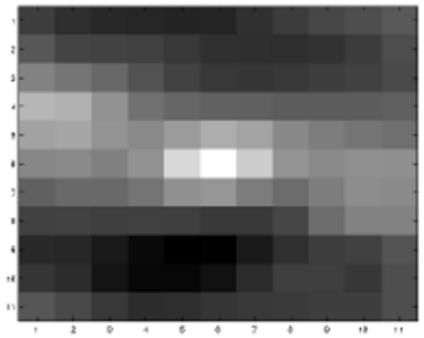
Corner Detection by Auto-correlation

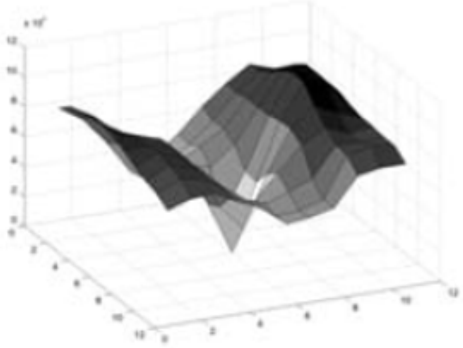
Change in appearance of window $w(x,y)$ for shift $[u,v]$:

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

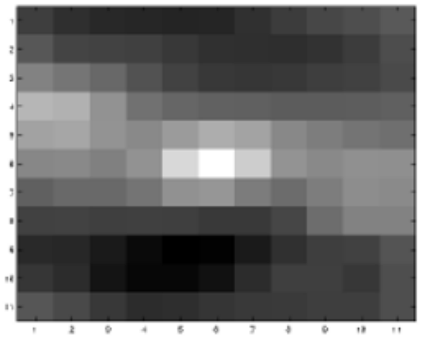
We want to discover how E behaves for small shifts

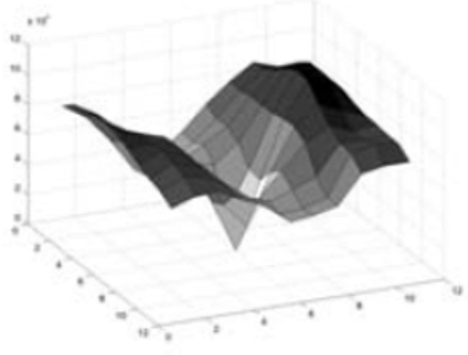
But we know the response in *E* that we are looking for – strong peak.



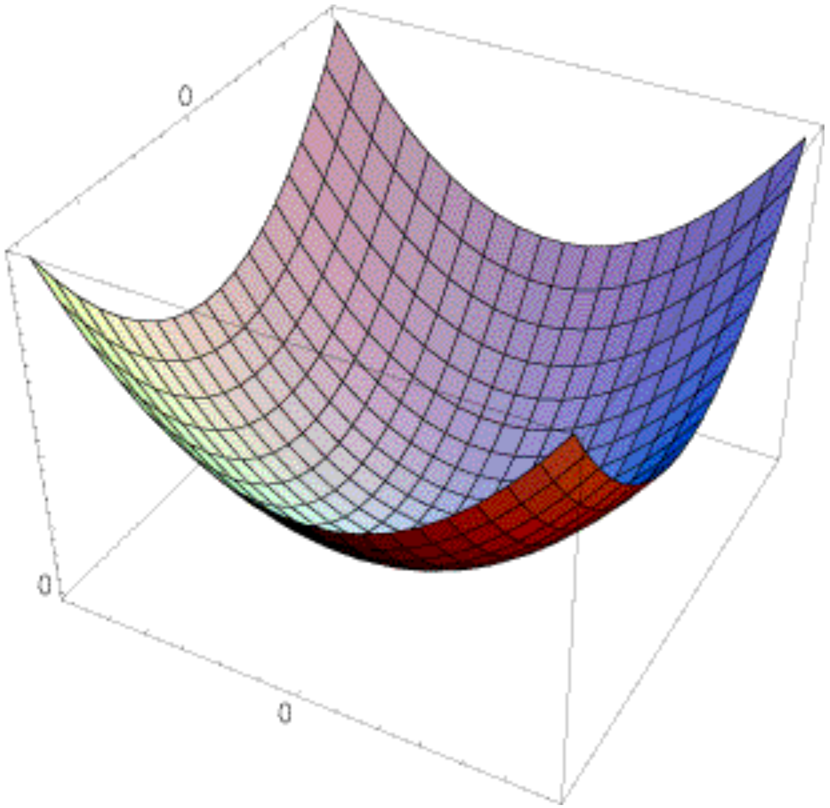


Can we just approximate $E(u,v)$ locally by a quadratic surface?





$\approx$



Recall: Taylor series expansion

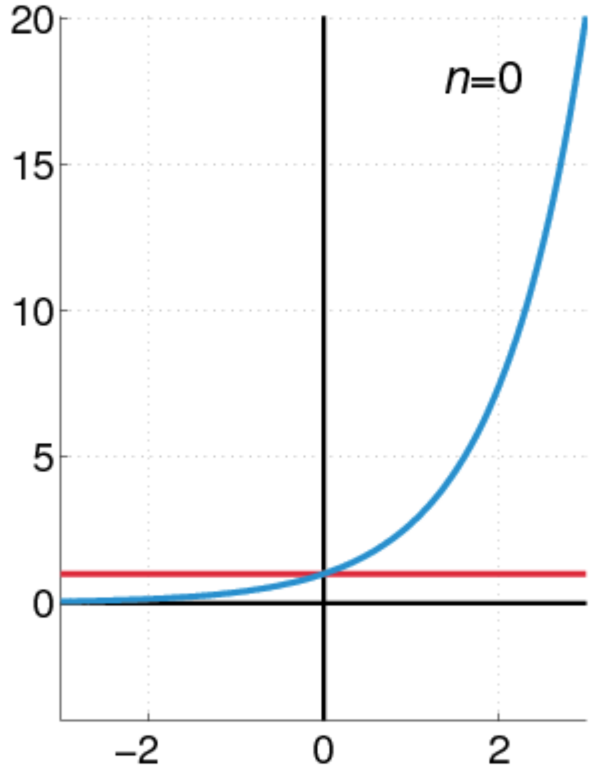
A function f can be represented by an infinite series of its derivatives at a single point *a*:

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

Wikipedia

As we care about window centered, we set *a = 0* (MacLaurin series)

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



Local quadratic approximation of *E*(*u*,*v*) in the neighborhood of (0,0) is given by the *second-order Taylor expansion*:

$$E(u,v) \approx E(0,0) + [u \quad v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \quad v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Notation: partial derivative

Local quadratic approximation of *E*(*u*,*v*) in the neighborhood of (0,0) is given by the *second-order Taylor expansion*:

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Ignore function value;

Ignore first derivative

Just look at shape of second derivative

We can prove that it is 0.

Corner Detection: Mathematics

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

Second-order Taylor expansion of $E(u,v)$ about (0,0):

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

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

$$E_{uu}(u,v) = \sum_{x,y} 2w(x,y) I_x(x+u,y+v) I_x(x+u,y+v)$$

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

$$E_{uv}(u,v) = \sum_{x,y} 2w(x,y) I_y(x+u,y+v) I_x(x+u,y+v)$$

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

Corner Detection: Mathematics

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

Second-order Taylor expansion of $E(u,v)$ about (0,0):

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(0,0) = 0$$

$$E_u(0,0) = 0$$

$$E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_x(x,y)$$

$$E_{vv}(0,0) = \sum_{x,y} 2w(x,y) I_y(x,y) I_y(x,y)$$

$$E_{uv}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_y(x,y)$$

Corner Detection: Mathematics

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

Second-order Taylor expansion of $E(u,v)$ about (0,0):

$$E(u,v) \approx [u \quad v] \begin{bmatrix} \sum_{x,y} w(x,y) I_x^2(x,y) & \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y) \\ \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y) & \sum_{x,y} w(x,y) I_y^2(x,y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(0,0) = 0$$

$$E_u(0,0) = 0$$

$$E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_x(x,y)$$

$$E_{vv}(0,0) = \sum_{x,y} 2w(x,y) I_y(x,y) I_y(x,y)$$

$$E_{uv}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_y(x,y)$$

Corner Detection: Mathematics

The quadratic approximation simplifies to

$$E(u,v) \approx [u \;\; v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

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

where M is a *second moment matrix* computed from image derivatives:

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

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Harris Detector Formulation

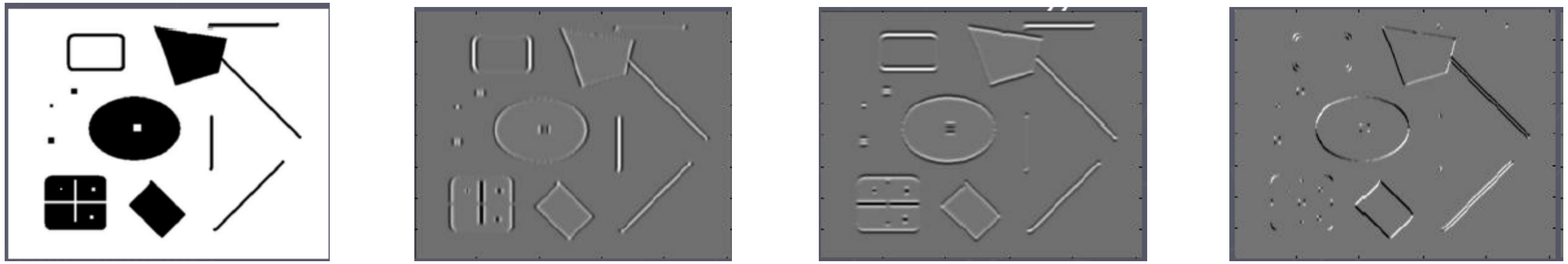


Image I $\qquad I_x \qquad I_y \qquad I_xI_y$

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

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

Gradient with respect to x**, times gradient with respect to** y

Sum over image region – the area we are checking for corner

$$M = \begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

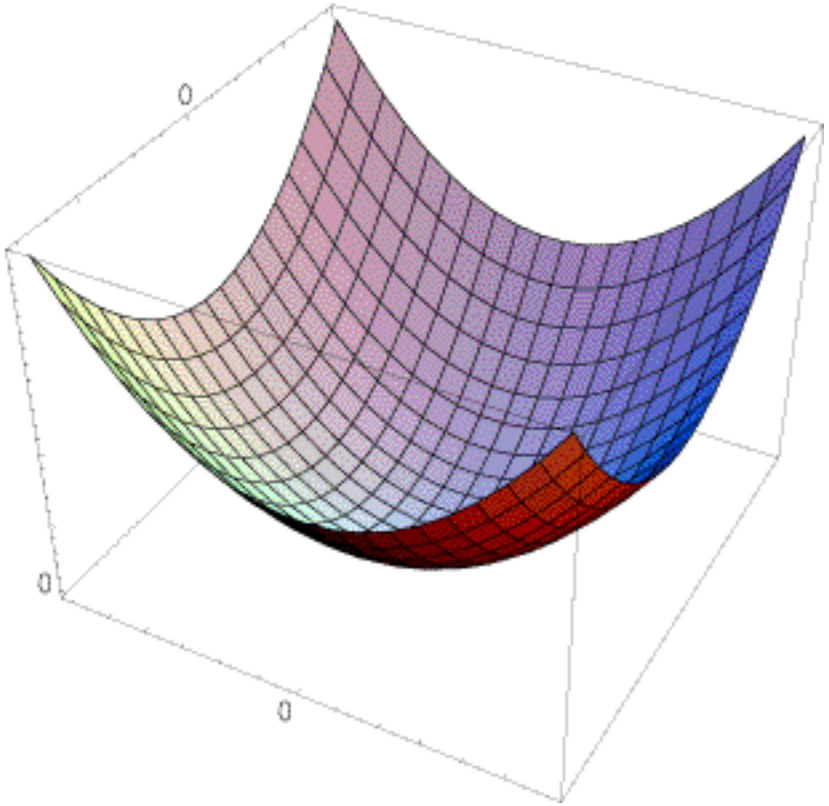
Slide credit: Rick Szeliski

Interpreting the second moment matrix

The surface *E*(*u*,*v*) is locally approximated by a quadratic form. Let's try to understand its shape.

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

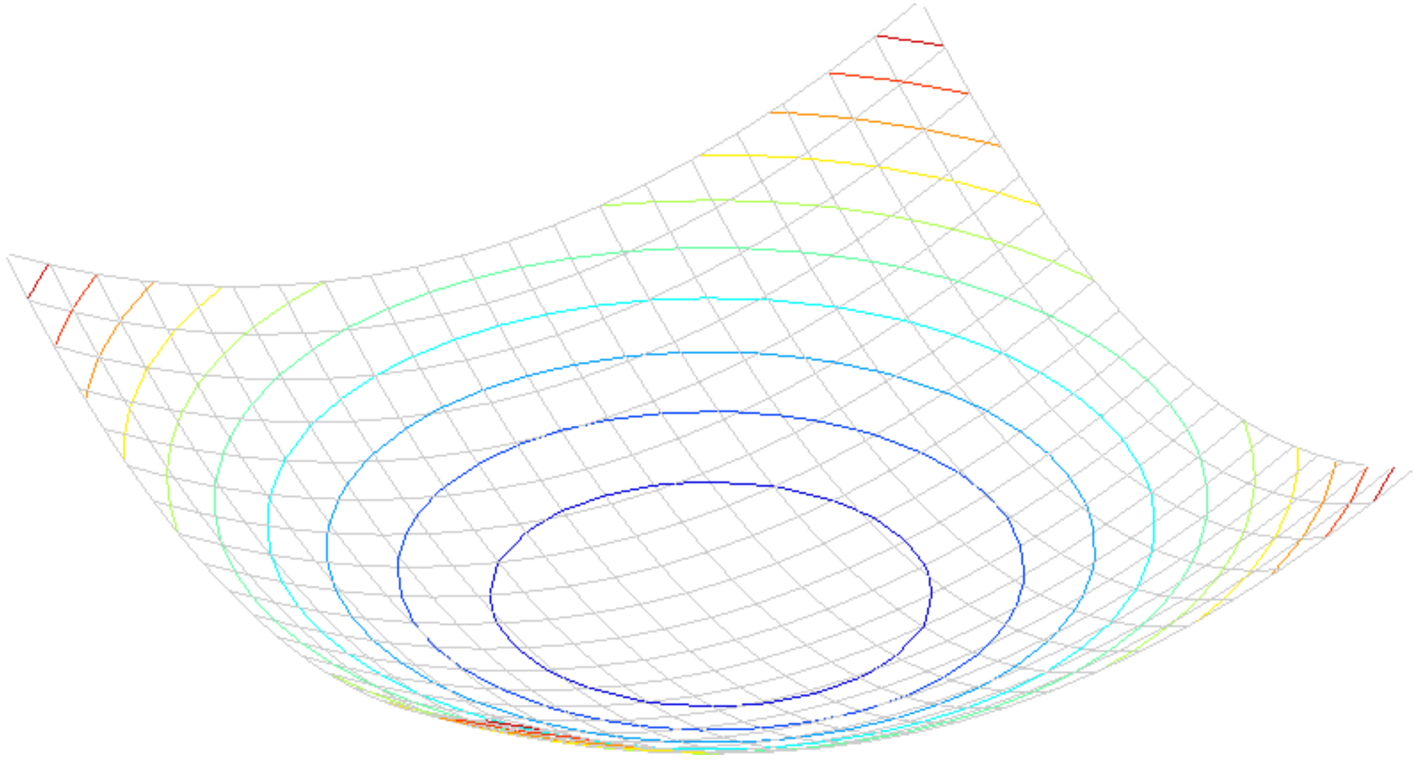
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



James Hays

Interpreting the second moment matrix

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

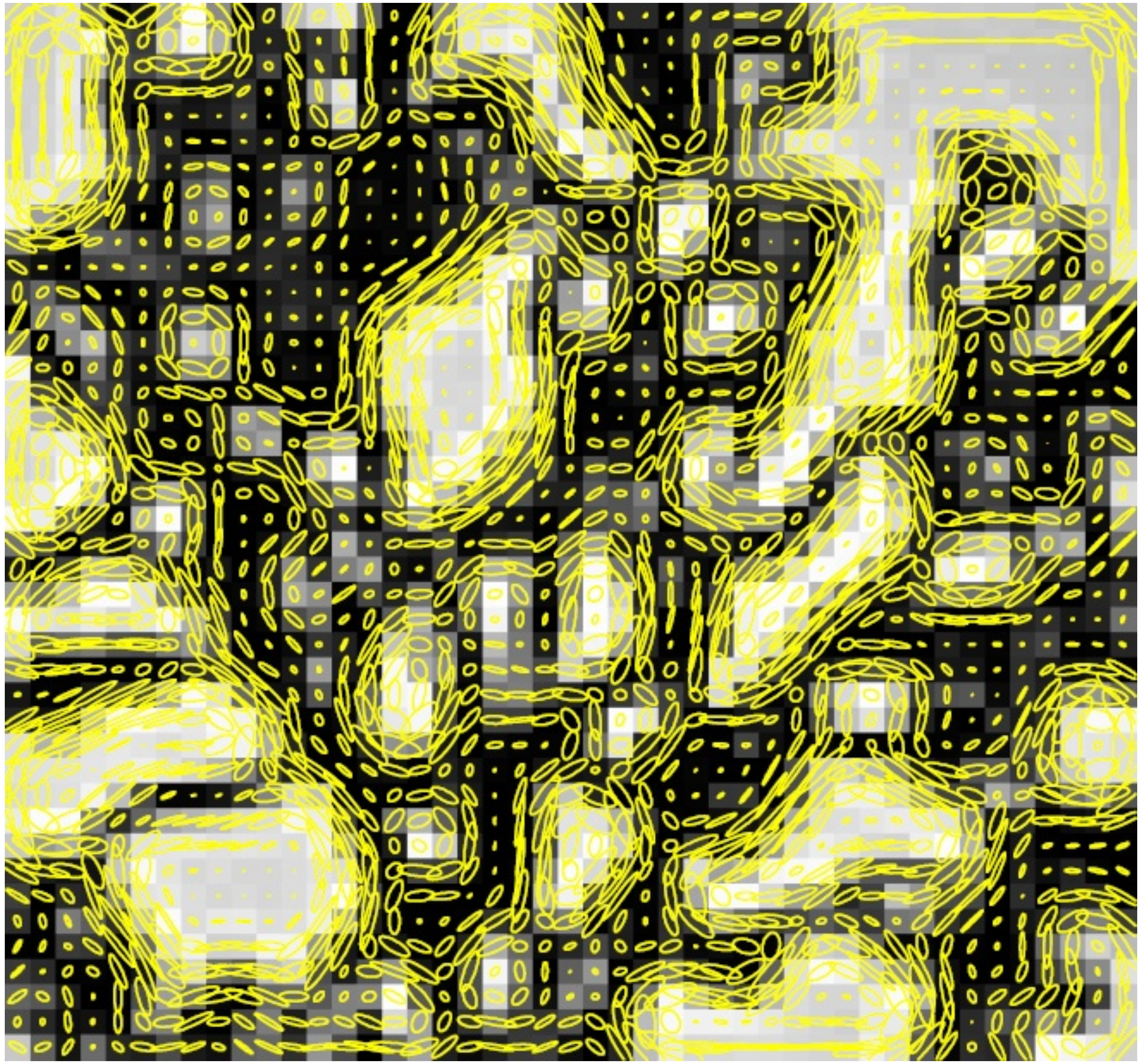


James Hays

Visualization of second moment matrices

James Hays

Visualization of second moment matrices



James Hays

Interpreting the second moment matrix

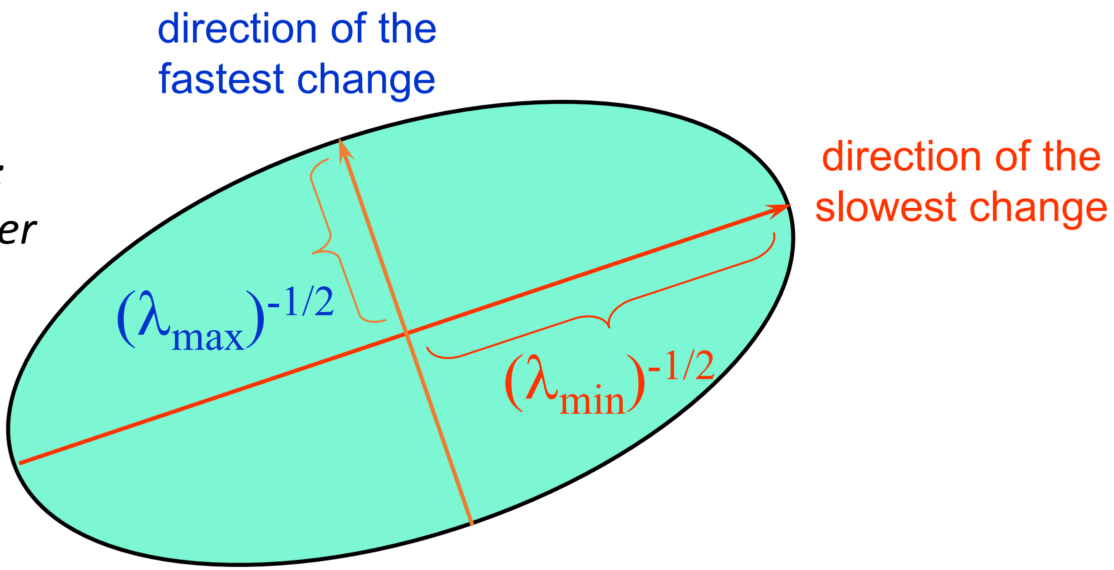
Consider a horizontal "slice" of $E(u, v)$: $[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

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 a rotation matrix R.

Note inverse relationship: larger eigenvalue = steeper slope; smaller ellipse in visualization



James Hays

The Principal Axes Theorem:

Let Abe an n x n symmetric matrix. Then there is an orthogonal change

of variable, **x**=P**y**, that transforms the quadratic form $\mathbf{x}^T A\mathbf{x}$ into a

vectors are orthogonal, if the dot product equals

ogonal.

quadratic from $\mathbf{y}^T D\mathbf{y}$ with no cross-product term (x_1x_2) (Lay, 453).

$$\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=0$$

5. $A = PDP^{-1}$:

$$P=\begin{bmatrix}-1 & 1\\ 1 & 1\end{bmatrix} \quad D=\begin{bmatrix}7 & 0\\ 0 & 3\end{bmatrix} \quad P^{-1}=\begin{bmatrix}-1/2 & 1/2\\ 1/2 & 1/2\end{bmatrix}$$

6. $\mathbf{x} = P\mathbf{y}$,

$$\mathbf{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix} \quad \mathbf{y}=\begin{bmatrix}y_1\\y_2\end{bmatrix}$$

7. Compute

$$5x_1^2 - 4x_1x_2 + 5x_2^2 = \mathbf{x}^T A\mathbf{x} = (P\mathbf{y})^T A(P\mathbf{y}) = \mathbf{y}^T P^T AP\mathbf{y} = \mathbf{y}^T D\mathbf{y}$$

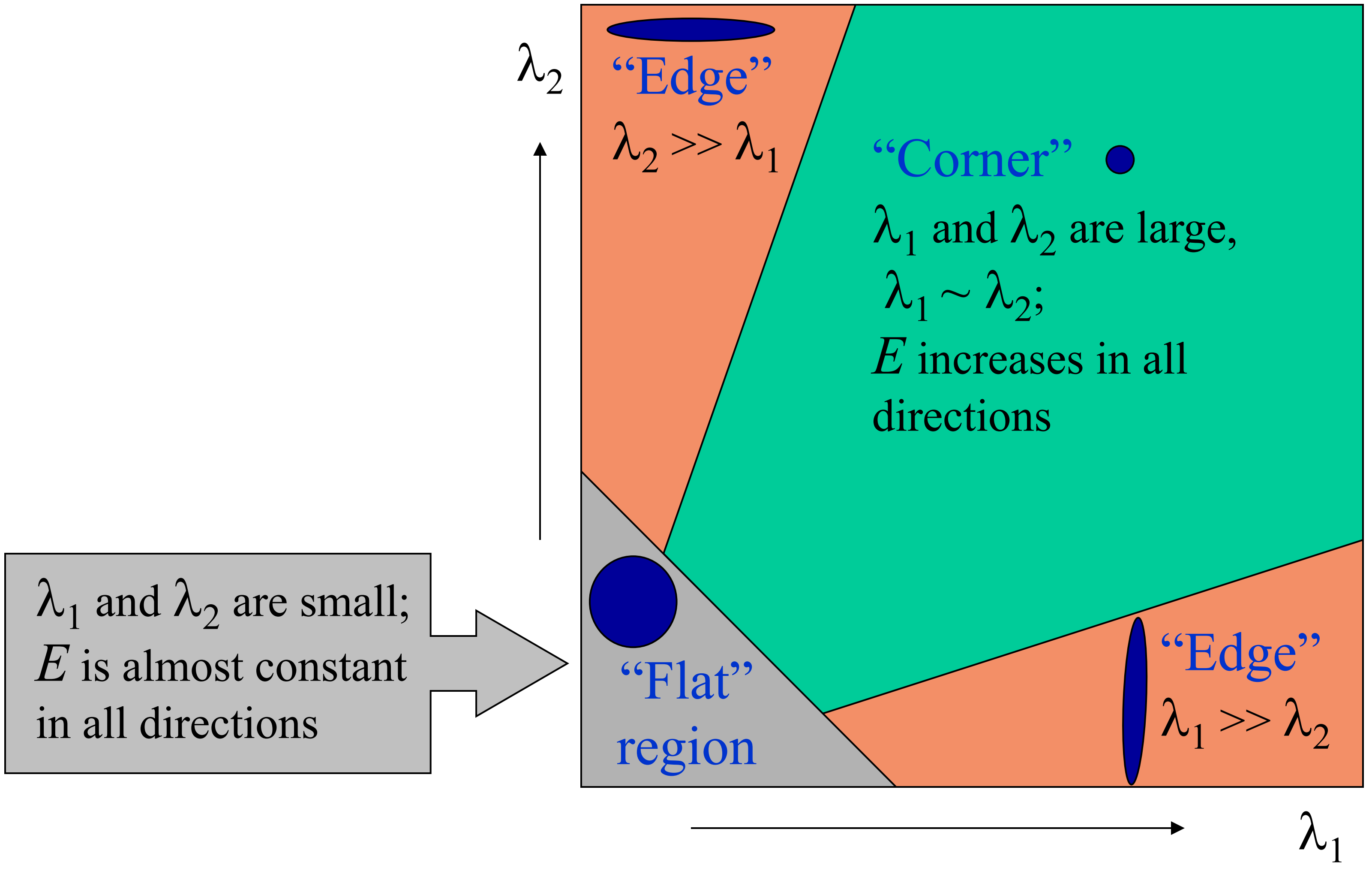
8.

$$=\begin{bmatrix}y_1 & y_2\end{bmatrix}\begin{bmatrix}7 & 0\\ 0 & 3\end{bmatrix}\begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}y_1 & y_2\end{bmatrix}\begin{bmatrix}7y_1\\3y_2\end{bmatrix}=7y_1^2+3y_2^2$$

9. Therefore

$$\therefore 5x_1^2 - 4x_1x_2 + 5x_2^2 = 7y_1^2 + 3y_2^2$$

Classification of image points using eigenvalues of M

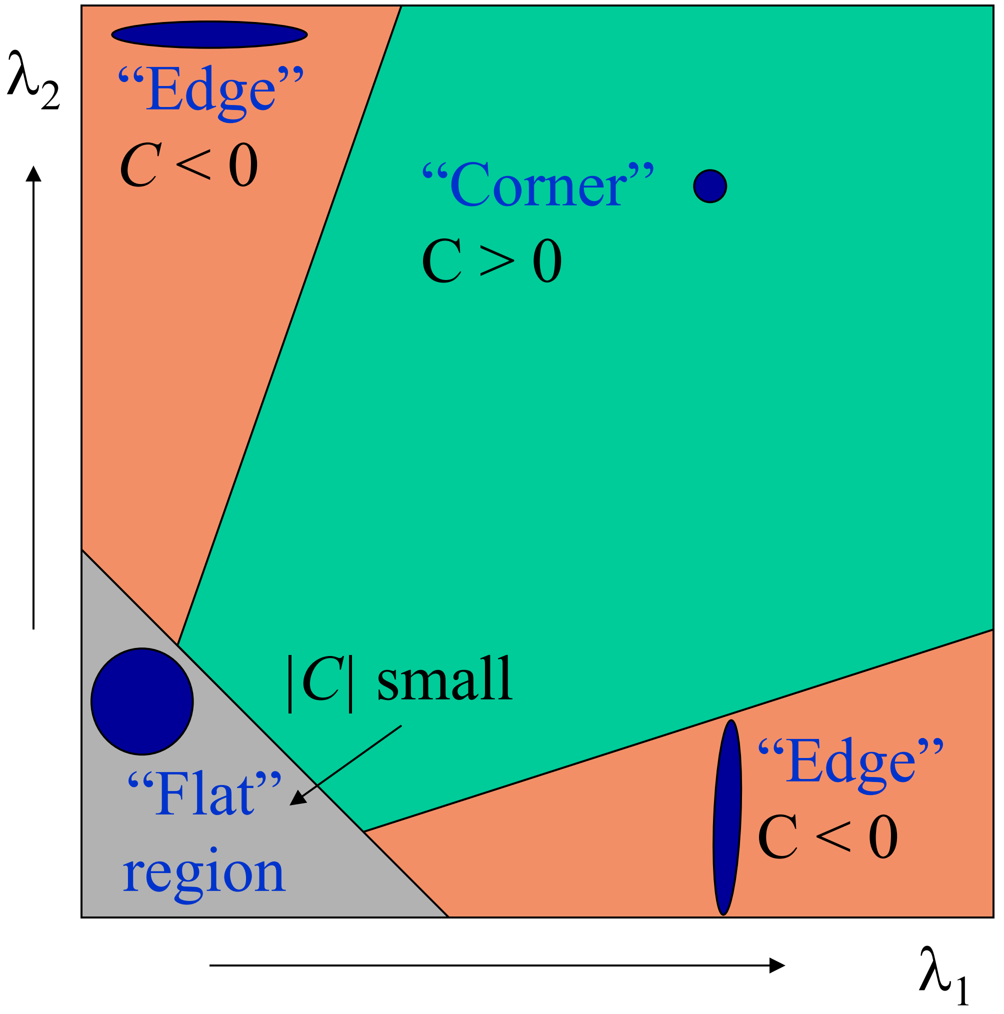


Classification of image points using eigenvalues of *M*

Cornerness

$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)



Classification of image points using eigenvalues of *M*

Cornerness

$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

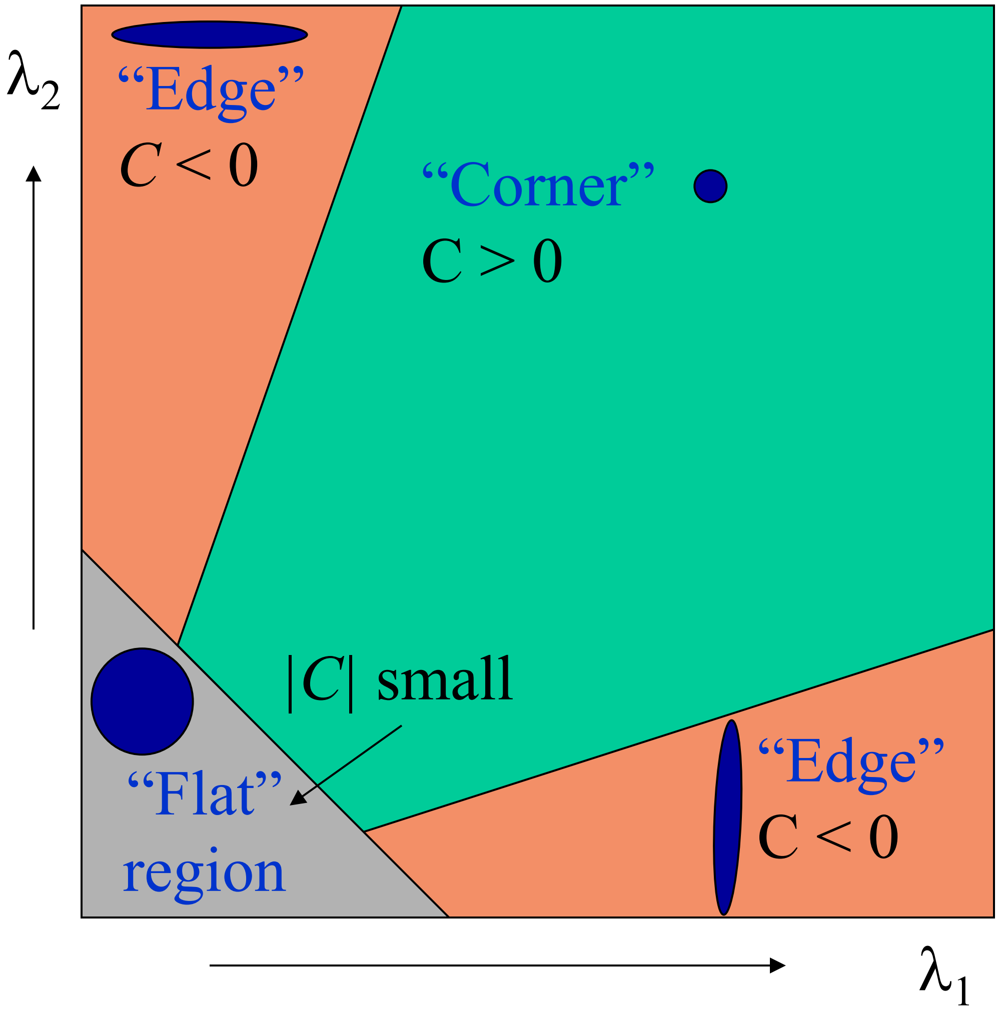
α: constant (0.04 to 0.06)

Remember your linear algebra:

Determinant: $\det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \cdots \lambda_n \, .$

Trace: $\operatorname{tr}(A) = \sum_i \lambda_i \, .$

$$C = \det(M) - \alpha \, \operatorname{trace}(M)^2$$

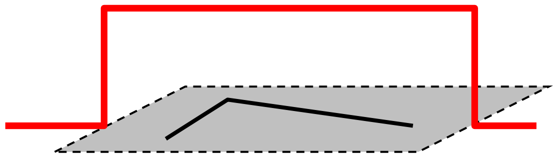


Window Function $w(x,y)$

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

- Option 1: uniform window
  - Sum over square window

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

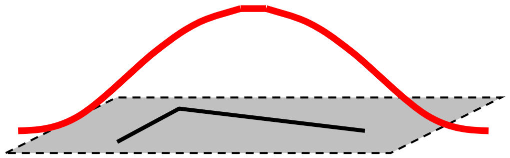


1 in window, 0 outside

  - Problem: not rotation invariant

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum

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



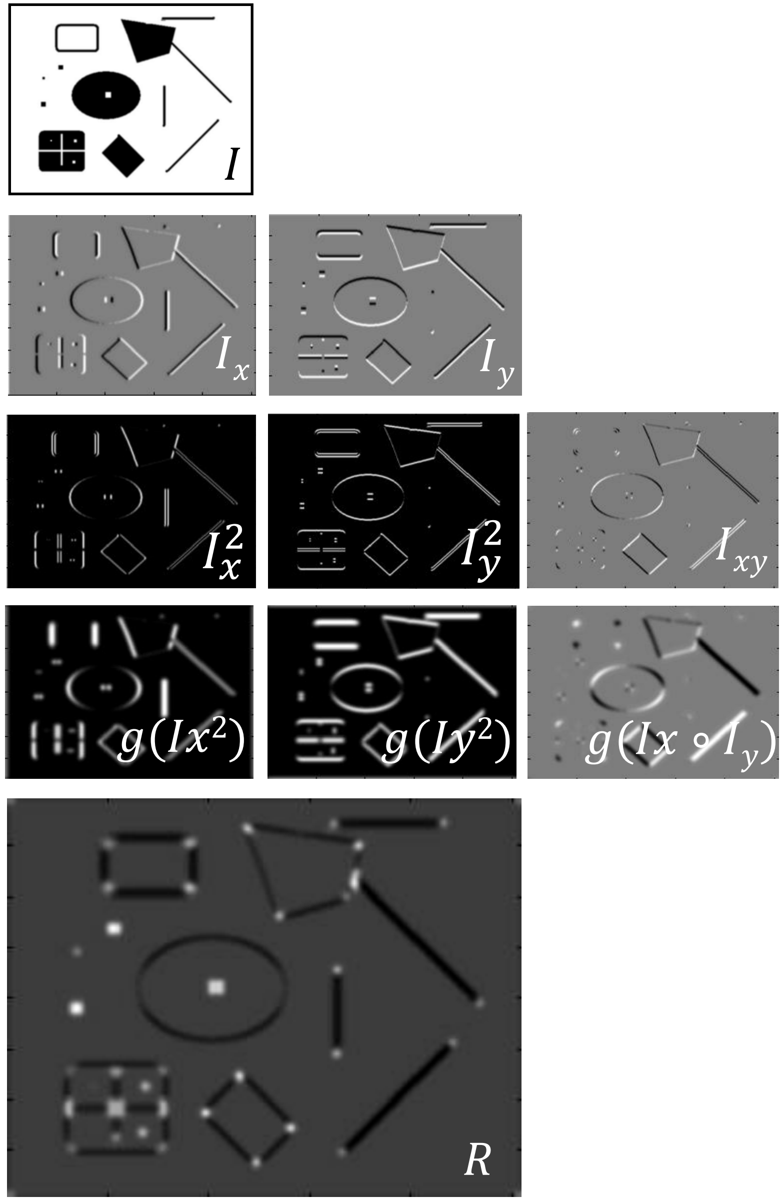
Gaussian

  - Result is rotation invariant

Slide credit: Bastian Leibe

Harris Corner Detector [Harris88]

James Hays



0. Input image
   We want to compute M at each pixel.

1. Compute image derivatives (optionally, blur first).

2. Compute M components as squares of derivatives.

3. Gaussian filter *g()* with width σ

4. Compute cornerness

$$C = \det(M) - \alpha\, \mathrm{trace}(M)^2$$
$$= g(I_x^2) \circ g(I_y^2) - g(I_x \circ I_y)^2$$
$$-\alpha\left[g(I_x^2) + g(I_y^2)\right]^2$$

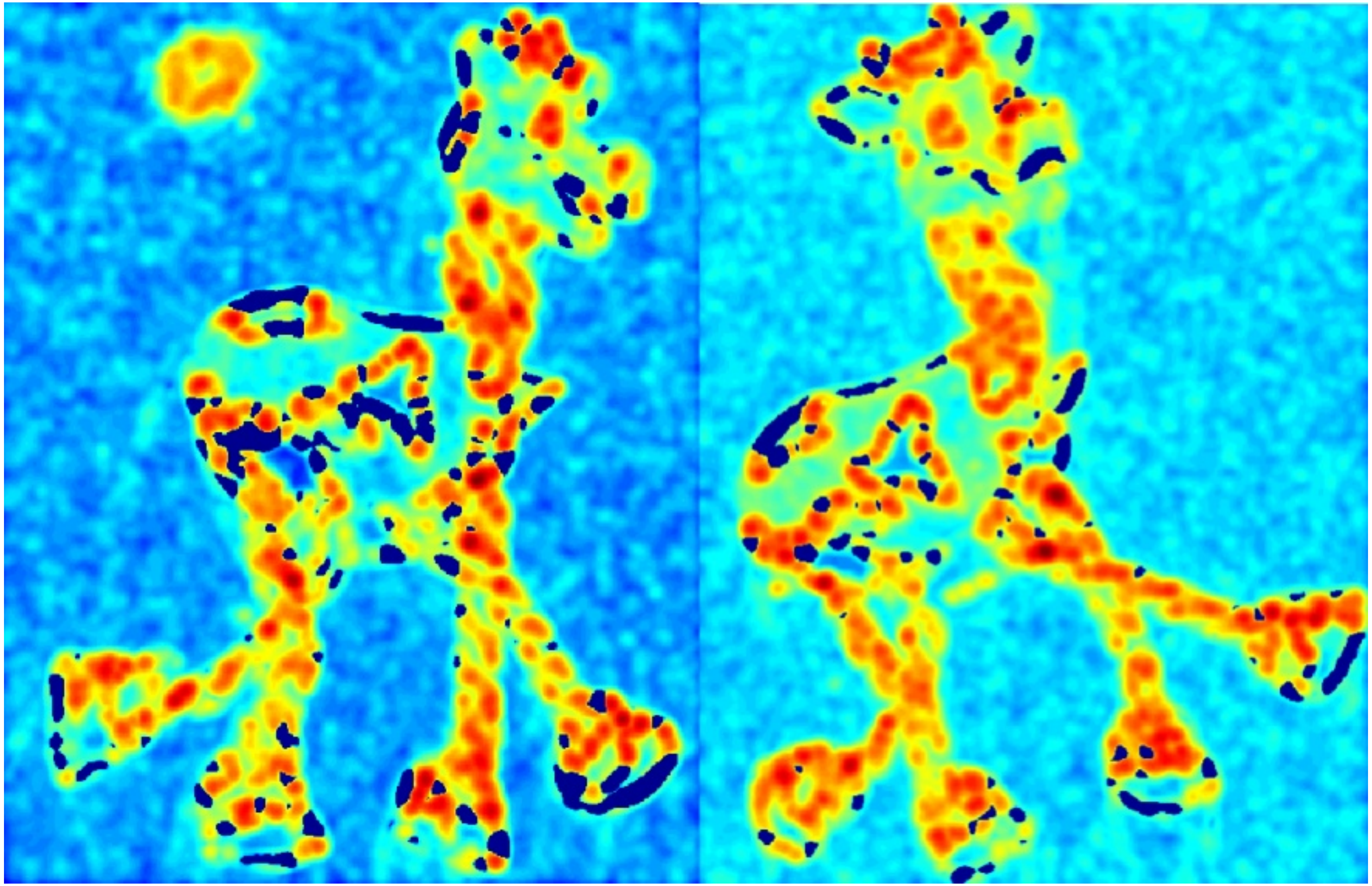
5. Threshold on C to pick high cornerness

6. Non-maxima suppression to pick peaks.

Harris Detector: Steps

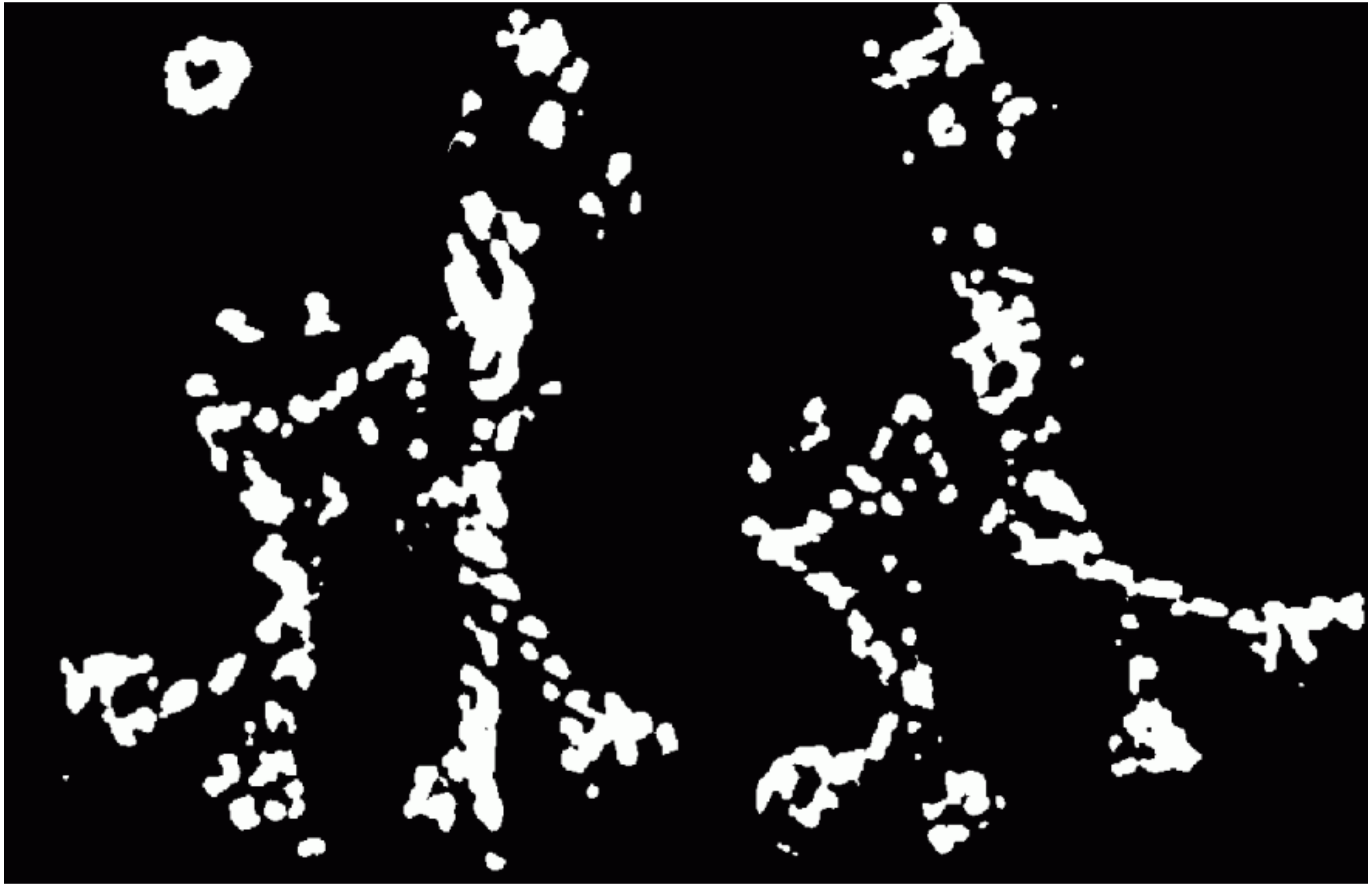
Harris Detector: Steps

Compute corner response C



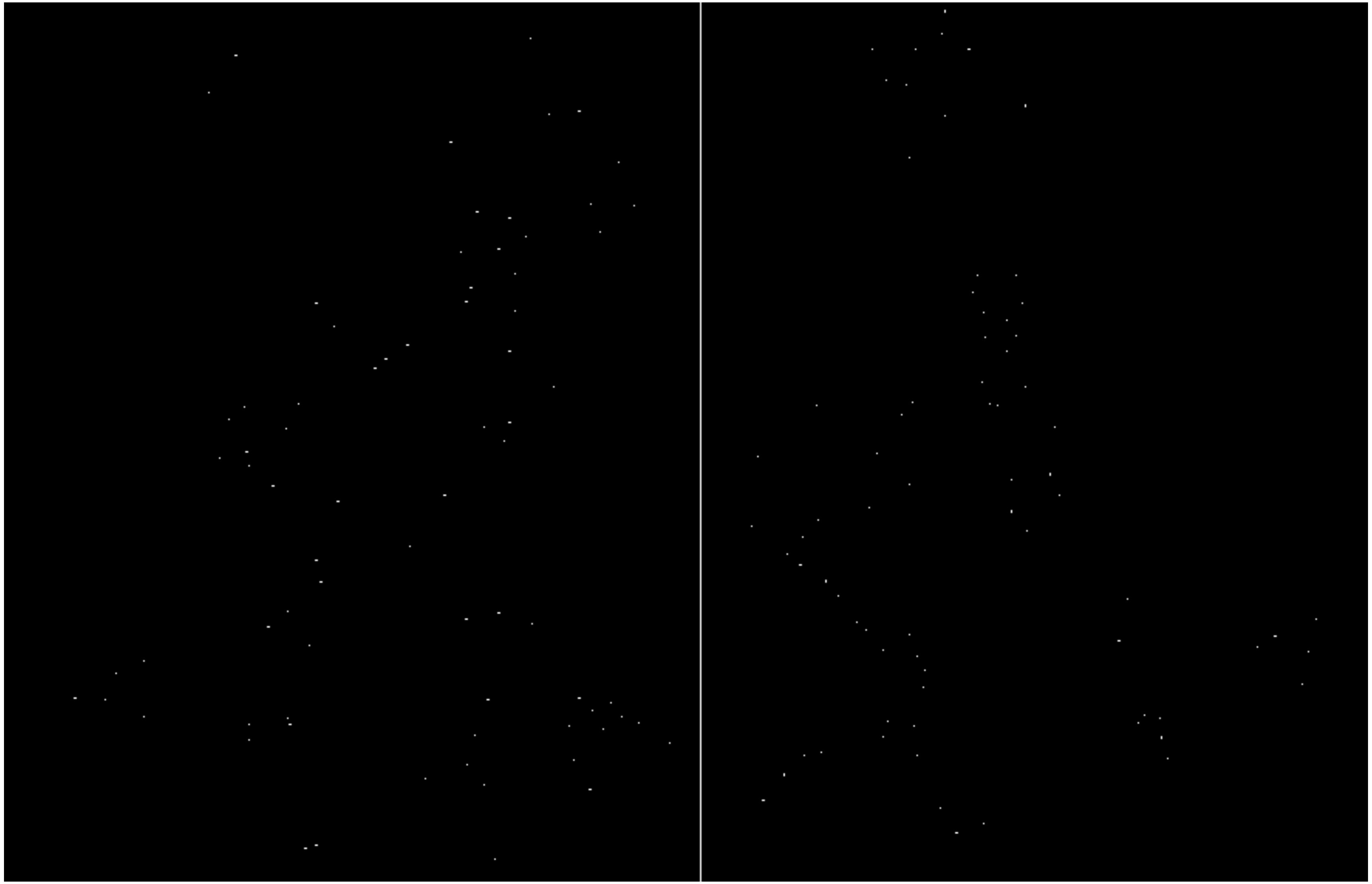
Harris Detector: Steps

Find points with large corner response: C > threshold

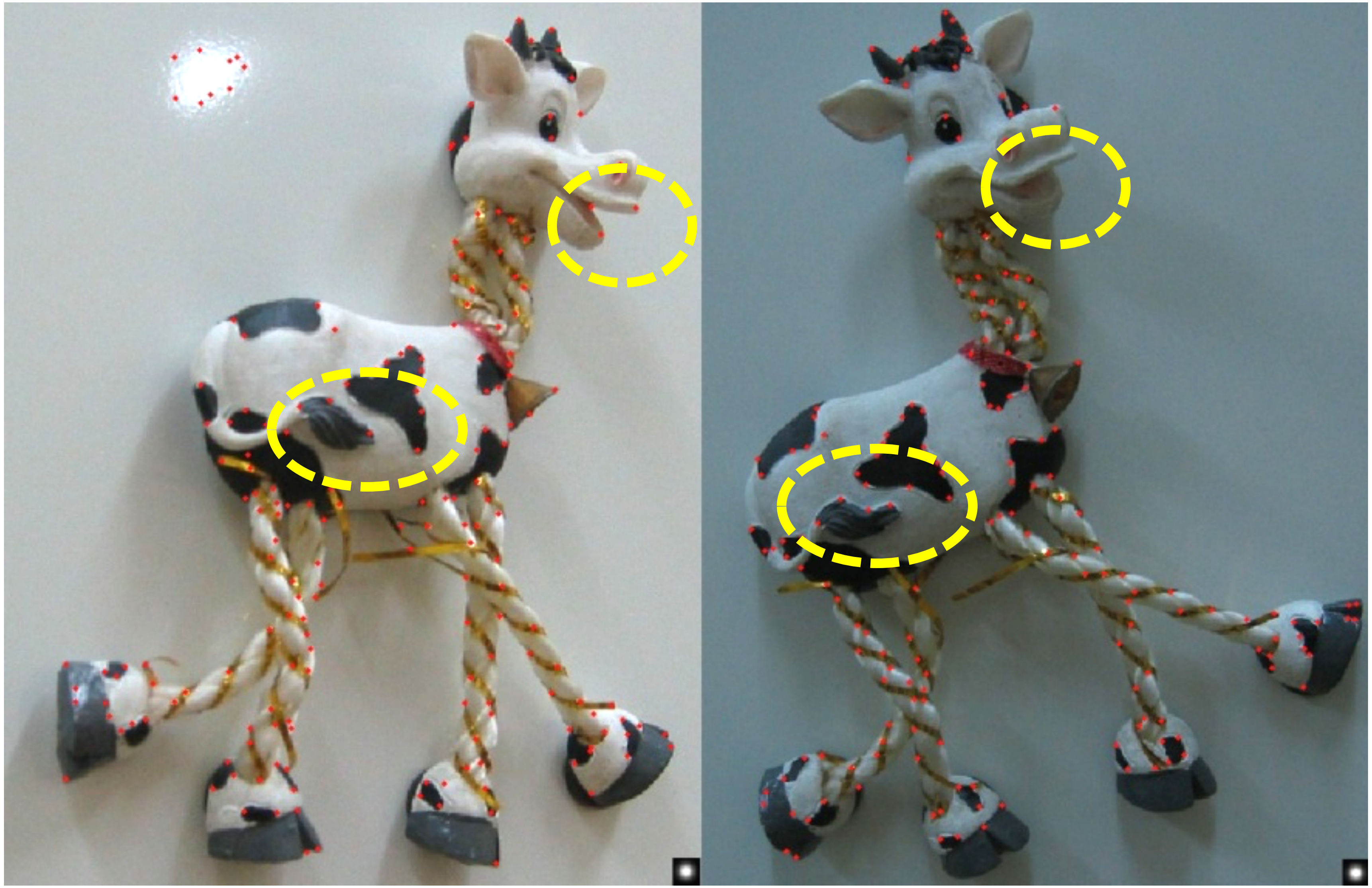


Harris Detector: Steps

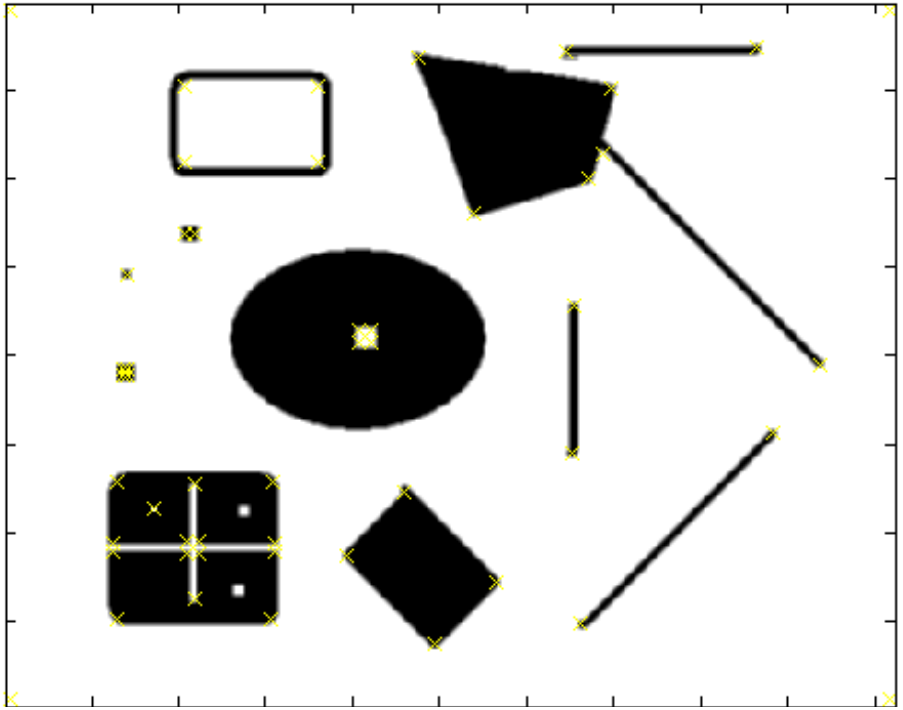
Take only the points of local maxima of C

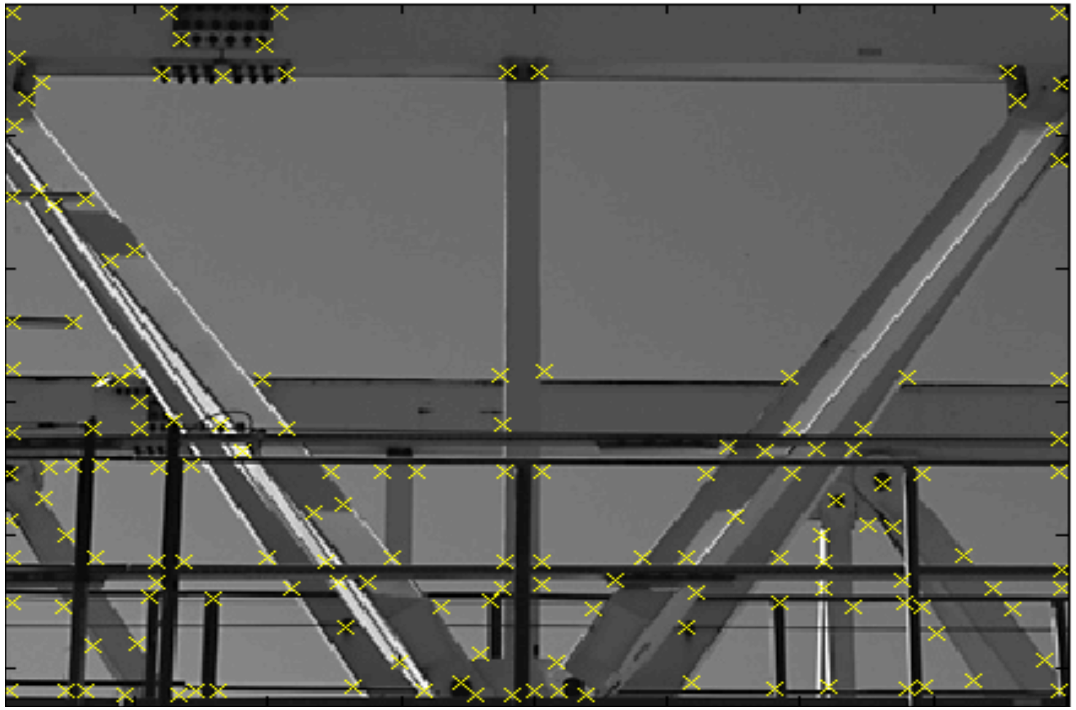


Harris Detector: Steps



Harris Detector – Responses [Harris88]

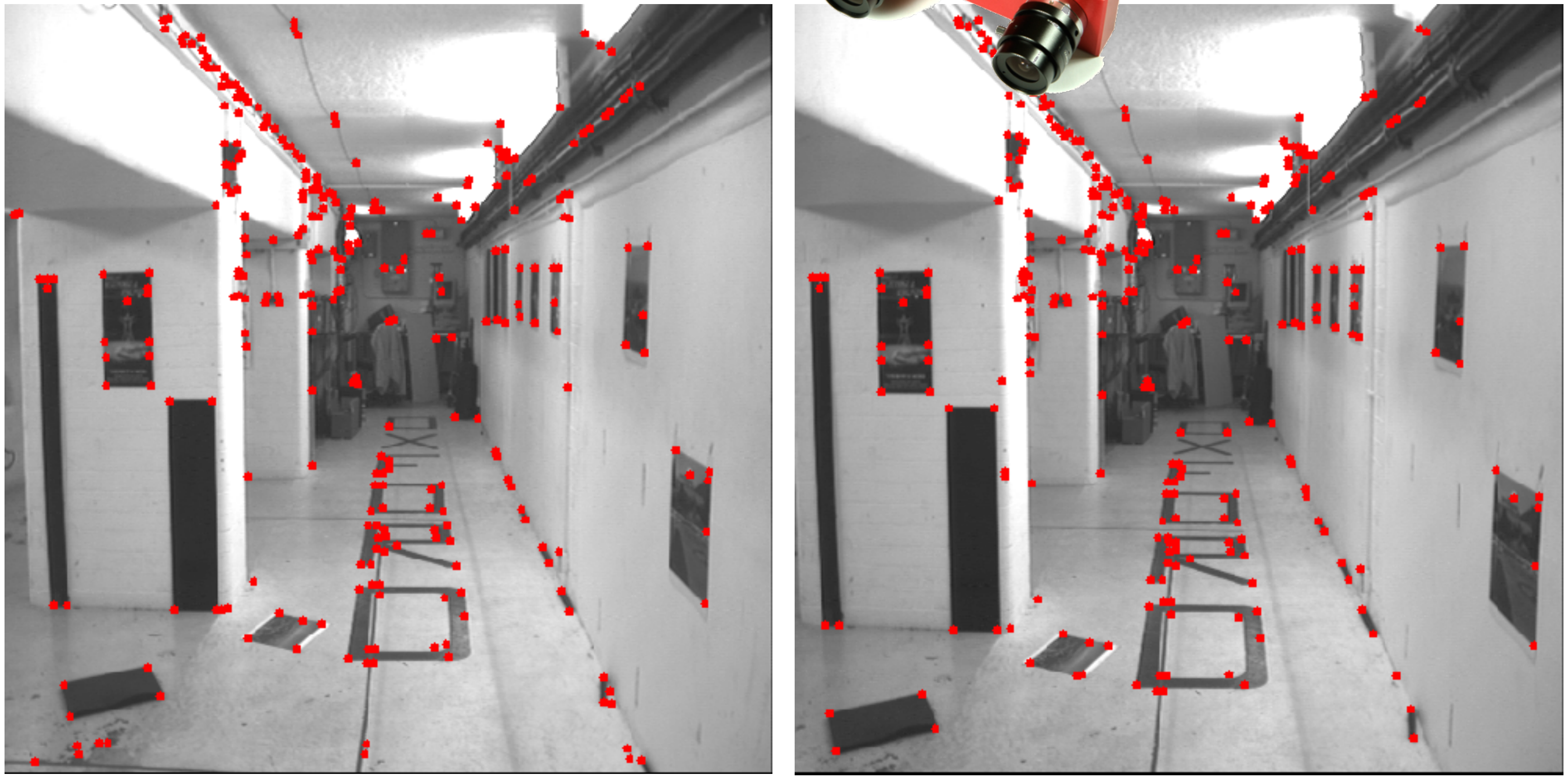




Effect: A very precise corner detector.

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]



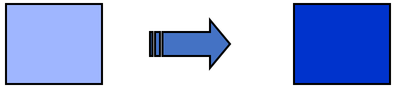
- Results are well suited for finding stereo correspondences

Slide credit: Kristen Grauman

Invariance and covariance

- Are locations *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

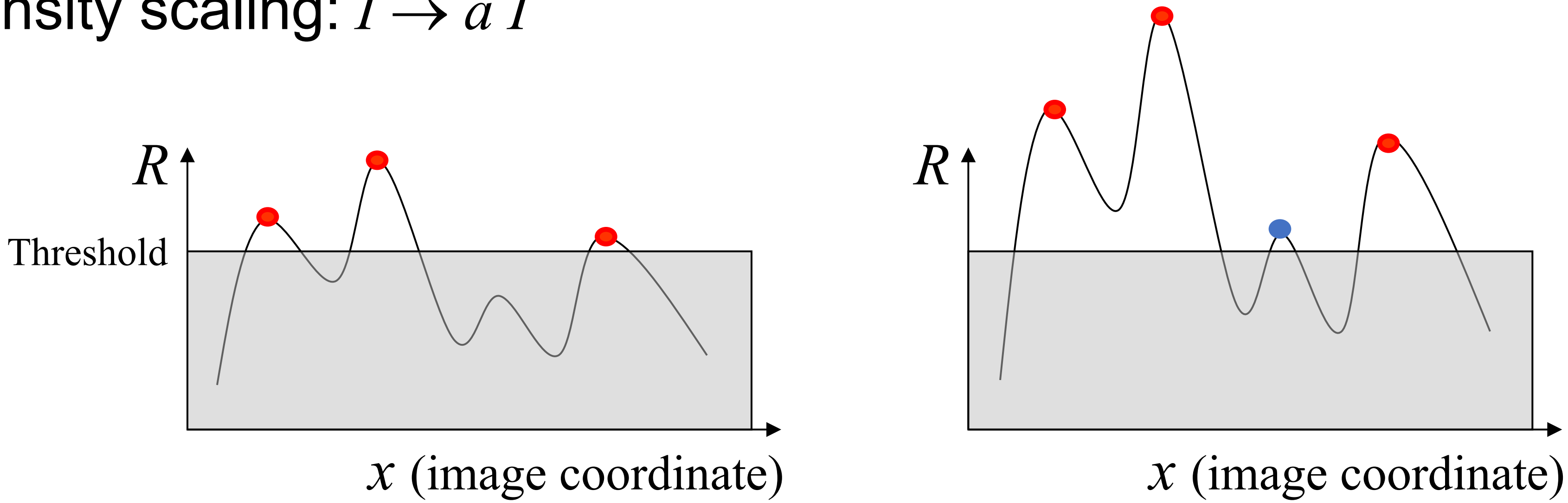
Affine intensity change



$$I \rightarrow a\,I + b$$

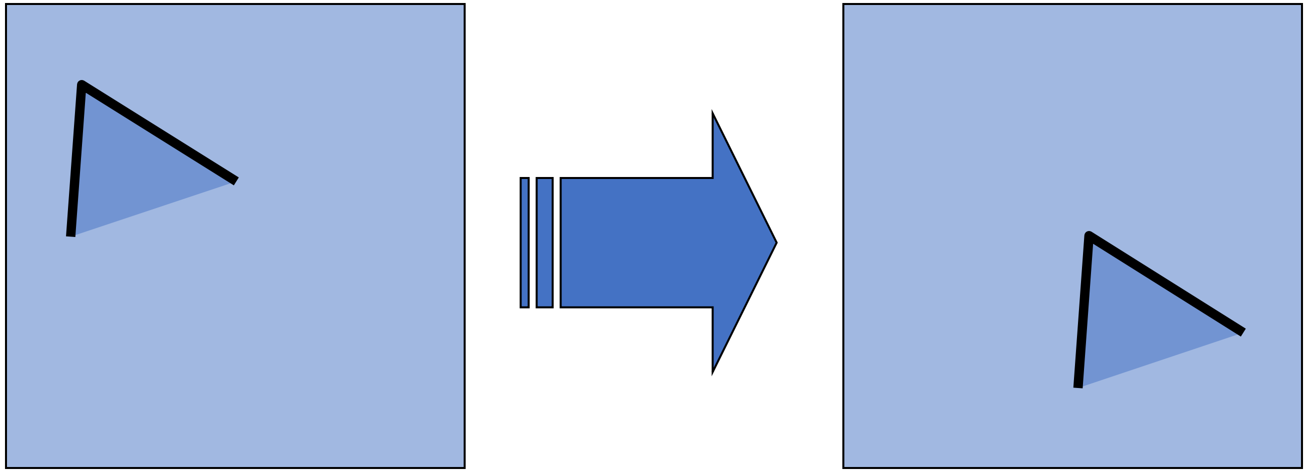
- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

- Intensity scaling: $I \rightarrow a\,I$



Partially invariant to affine intensity change

Image translation

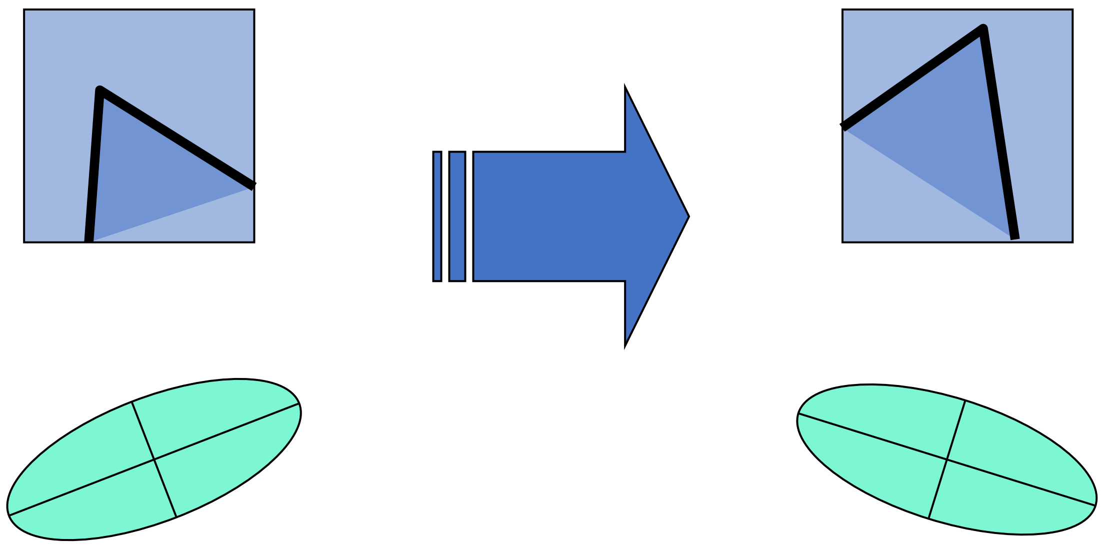


- Derivatives and window function are shift-invariant.

Corner location is covariant w.r.t. translation

James Hays

Image rotation

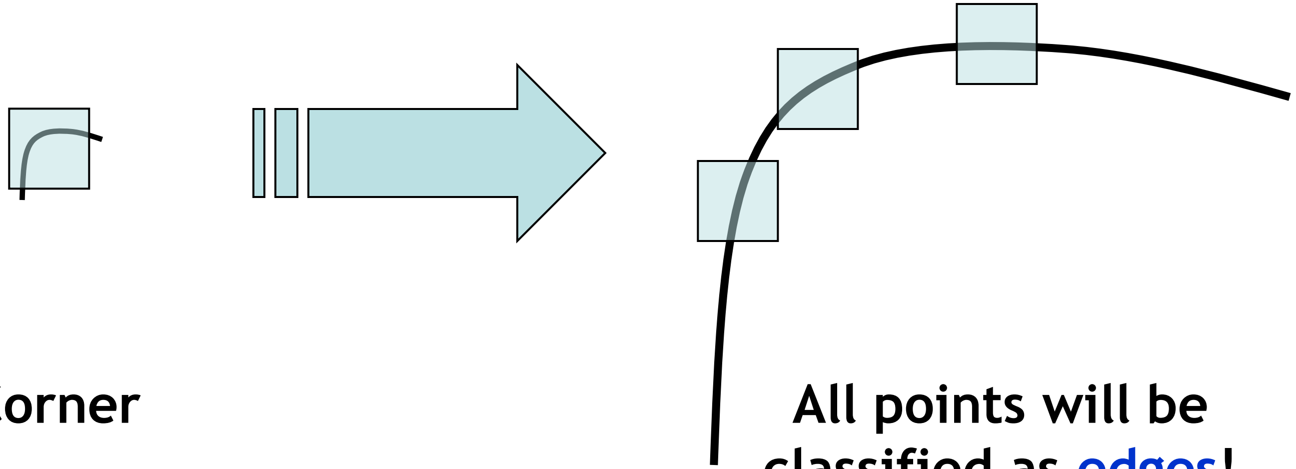


Second moment ellipse rotates but its shape (i.e., eigenvalues) remains the same.

Corner location is covariant w.r.t. rotation

Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?



Corner

All points will be classified as edges!

Not invariant to image scale!

Slide credit: Kristen Grauman

What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector

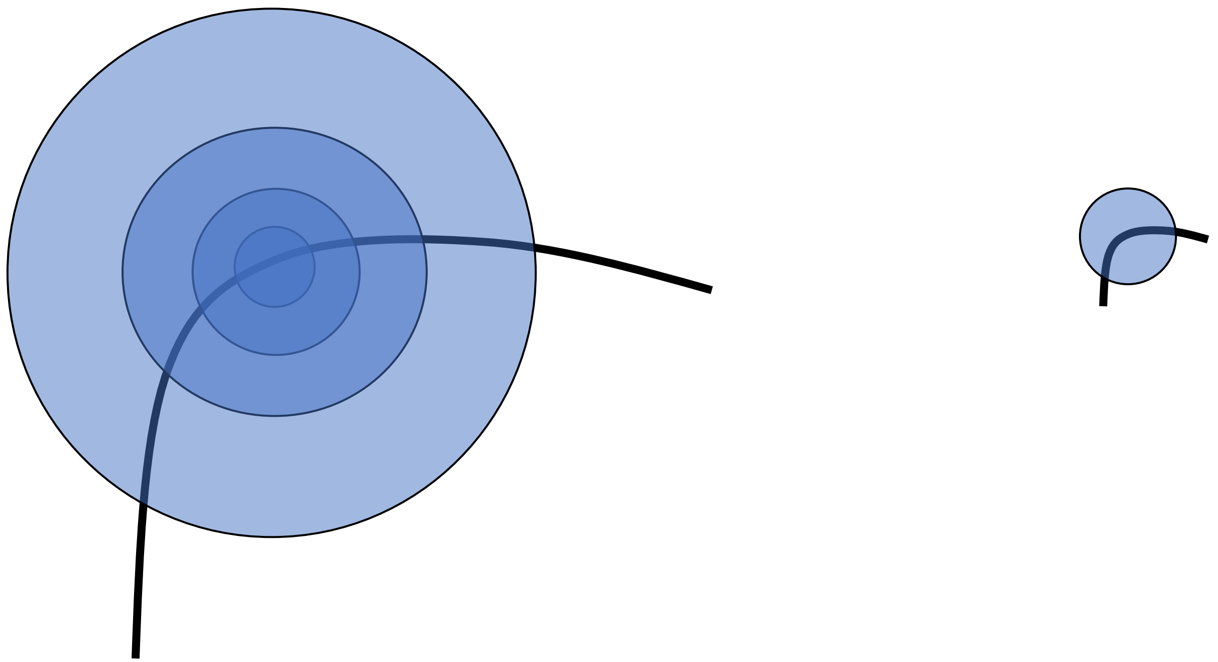
Some background reading:
Rick Szeliski, Chapter 4.1;

- (optional) K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10):1615-1630
  - http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_

What is the ‘scale’ of a feature point?

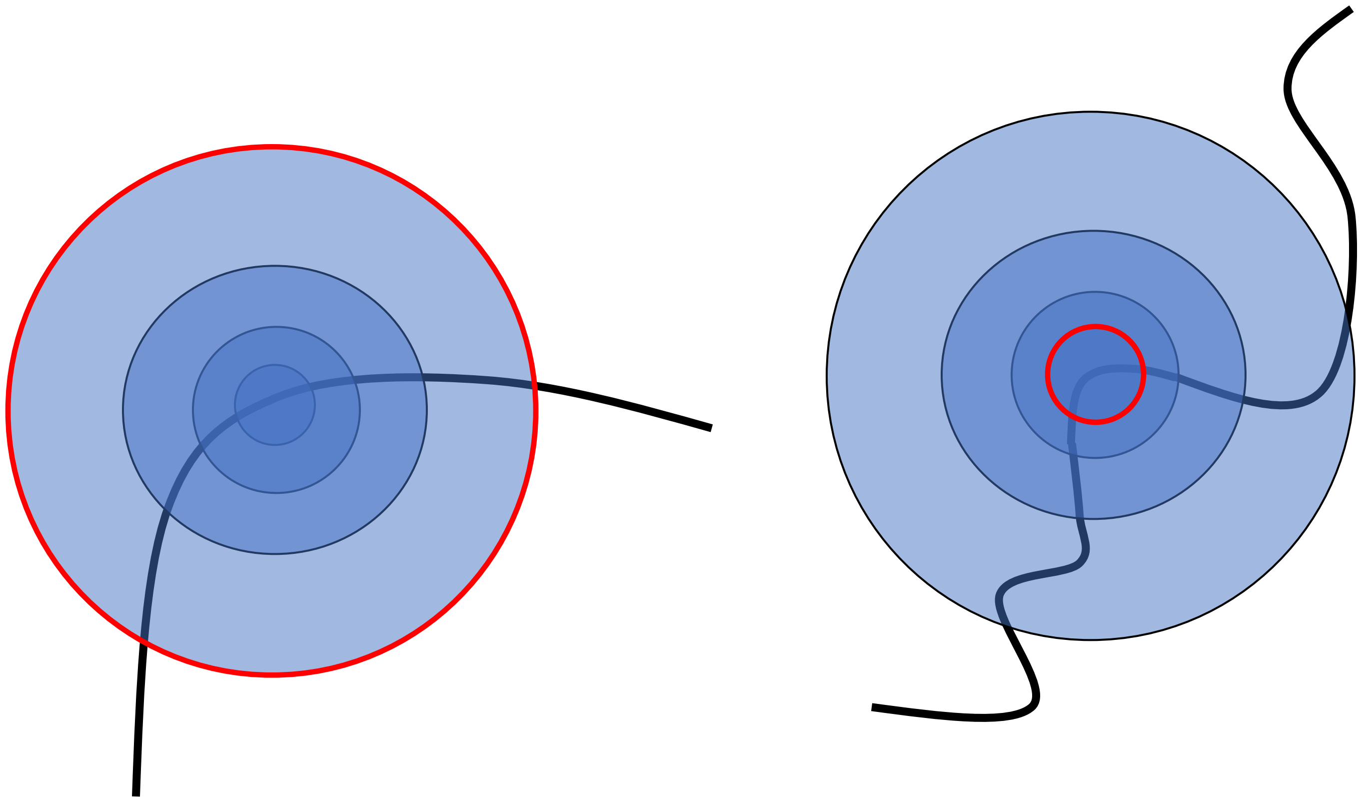
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Find regions of corresponding sizes that will look the same in both images?



Scale Invariant Detection

- The problem: how do we choose corresponding circles ***independently*** in each image?

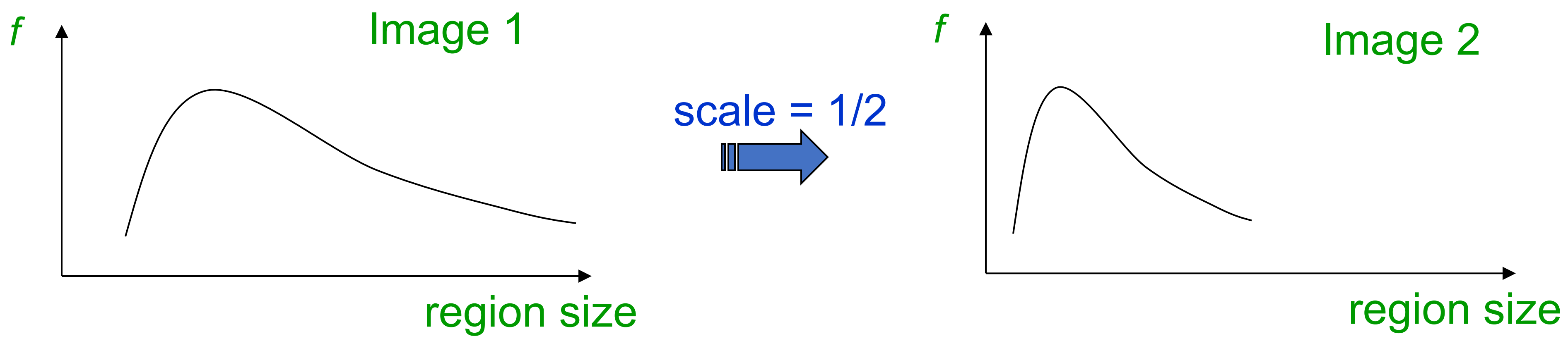


Scale Invariant Detection

- Solution:
  1. Design a function on the region (circle), 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.

  2. For a point in one image, we can consider it as a function of region size (circle radius)



Scale Invariant Detection

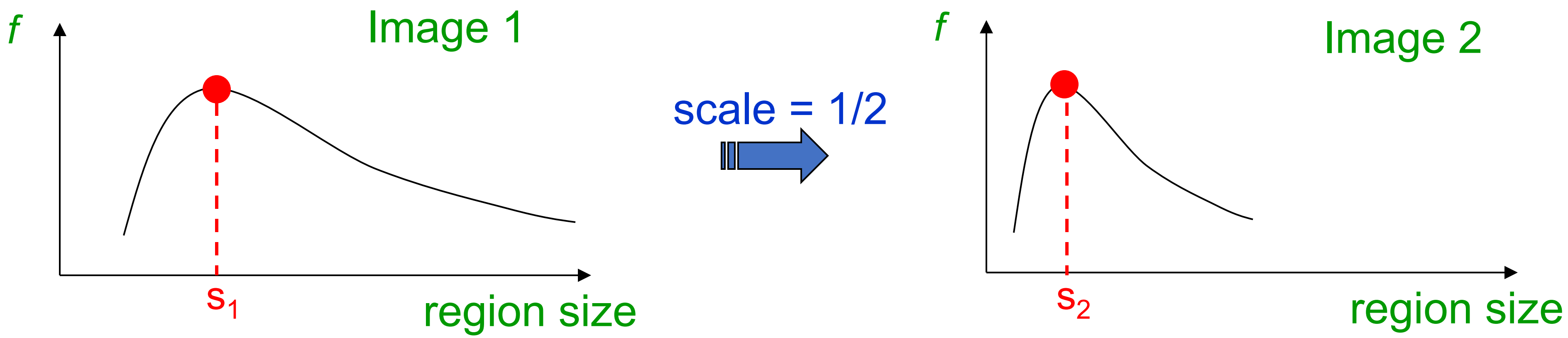
Scale invariant feature transform (SIFT, Lowe 2004)

- Common approach:

  Take a local maximum of this function

- Observation: region size, for which the maximum is achieved, should be *co-variant* with image scale.

Important: this scale invariant region size is found in each image independently!



D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

Automatic Scale Selection

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

How to find patch sizes at which f response is equal?

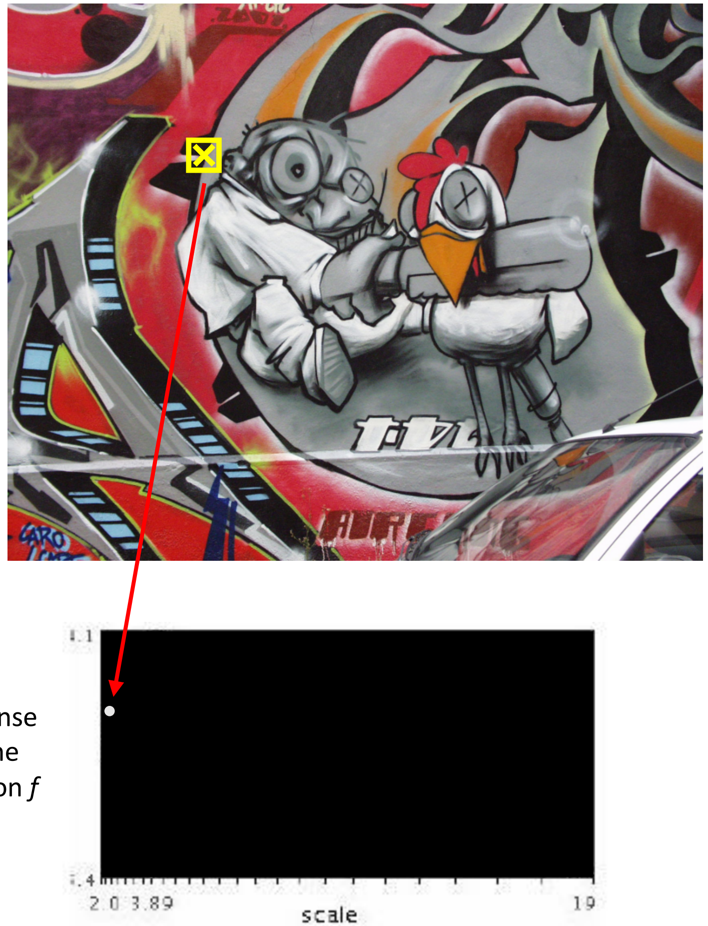
What is a good f ?

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Response
of some
function f



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

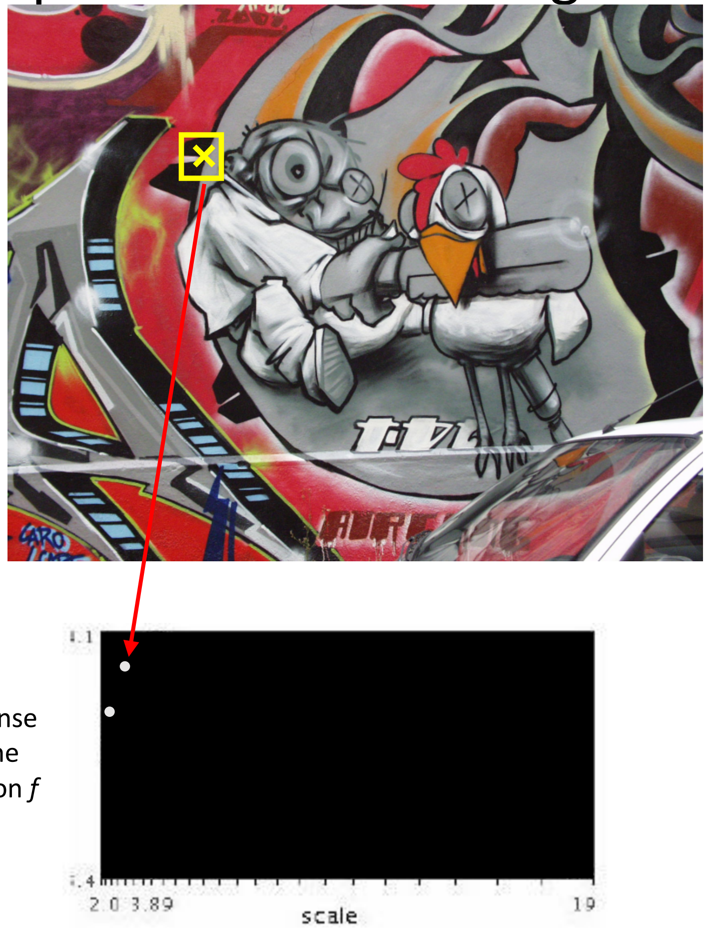
$f(I_{i_1 \ldots i_m}(x', \sigma'))$

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Response of some function *f*



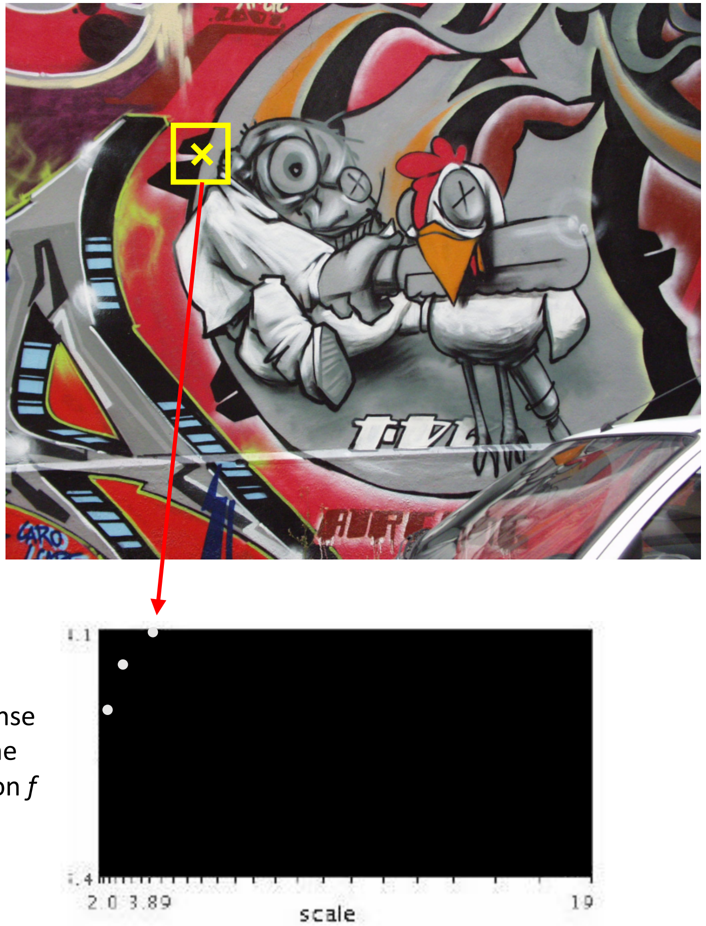
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



Response
of some
function *f*

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

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Response of some function *f*

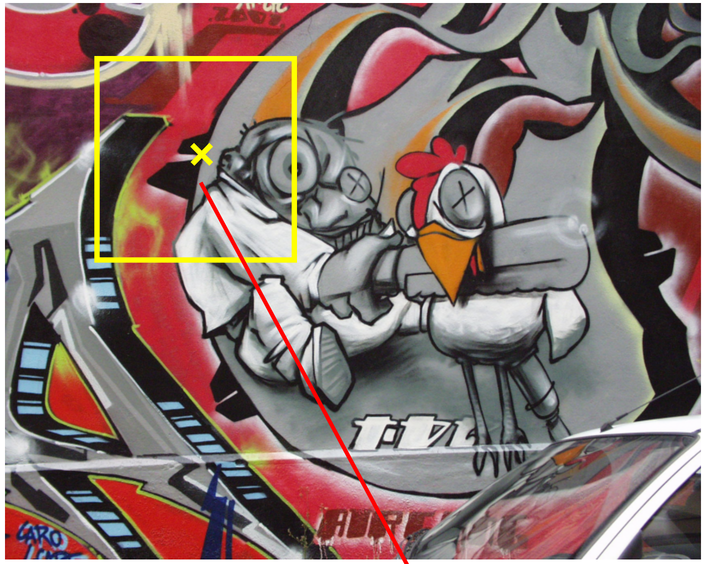
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

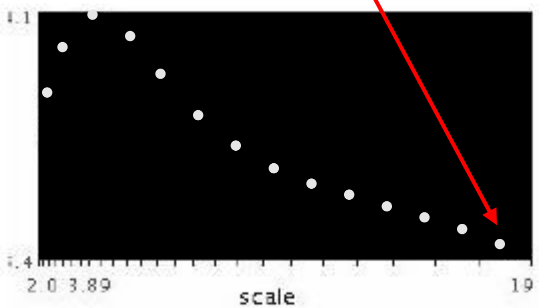
K. Grauman, B. Leibe

Automatic Scale Selection

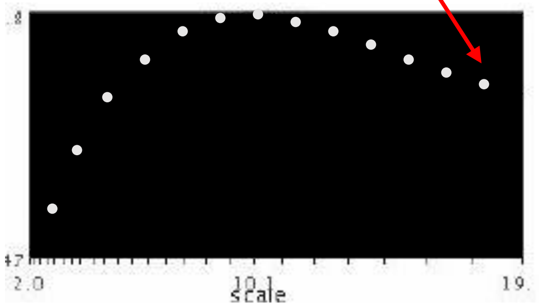
- Function responses for increasing scale (scale signature)



Response of some function *f*



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

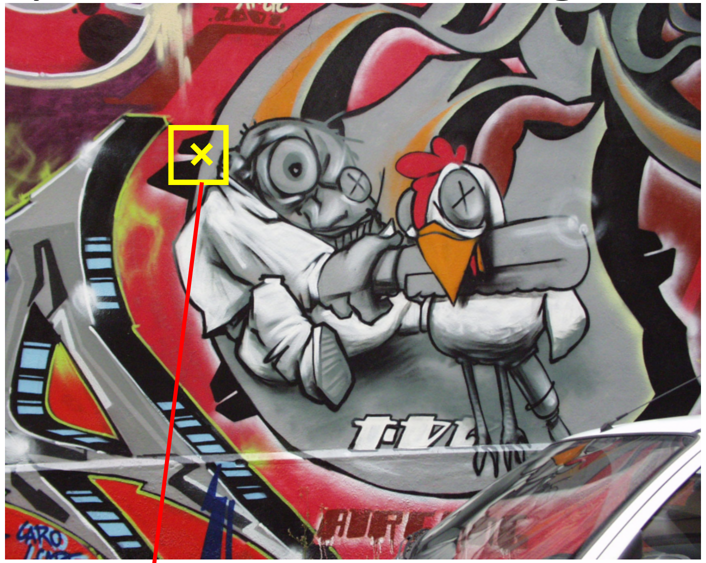


$f(I_{i_1 \ldots i_m}(x',\sigma'))$

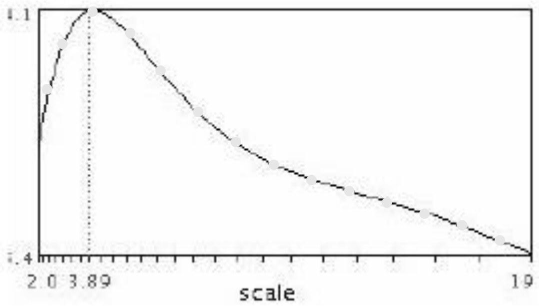
K. Grauman, B. Leibe

Automatic Scale Selection

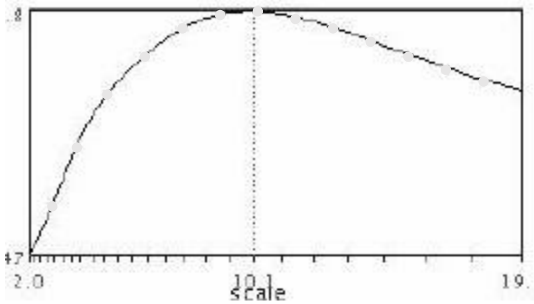
- Function responses for increasing scale (scale signature)



Response of some function *f*



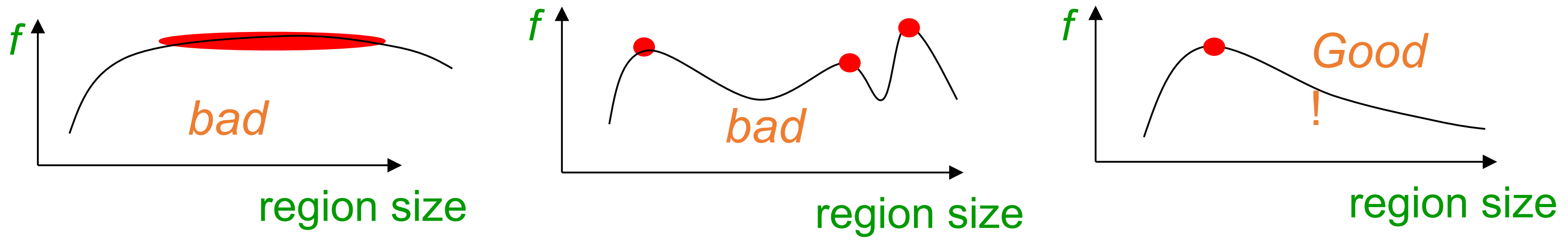
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$



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

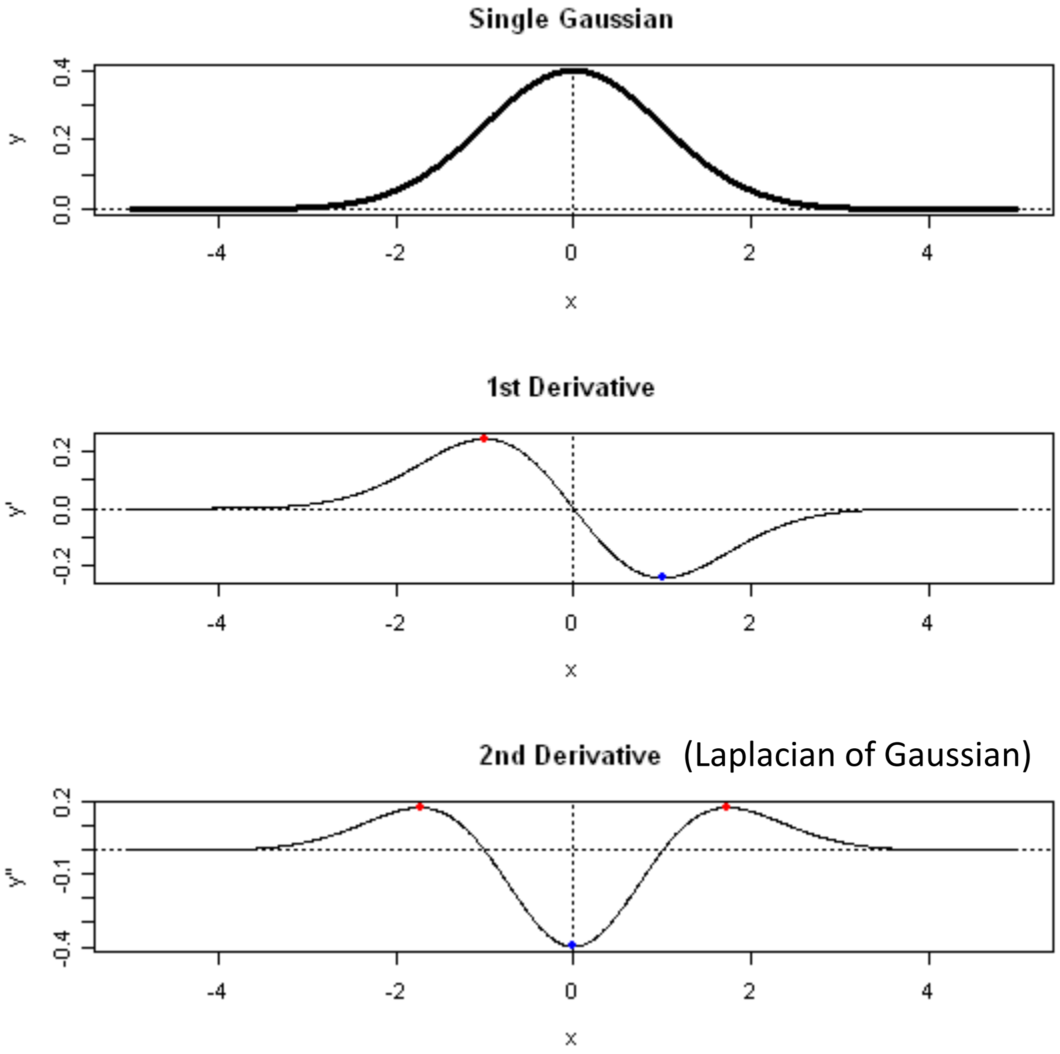
Scale Invariant Detection

- A "good" function for scale detection:
  has one stable sharp peak



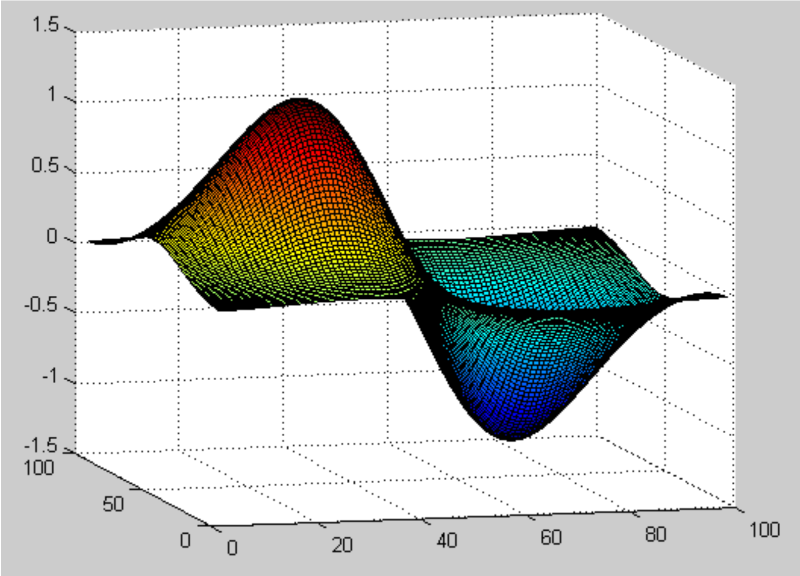
- For usual images: a good function would be one which responds to contrast (sharp local intensity change)

What Is A Useful Signature Function f ?



(Laplacian of Gaussian)

1st Derivative of Gaussian



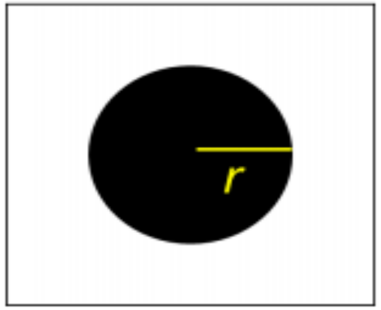
Earl F. Glynn

Scale selection

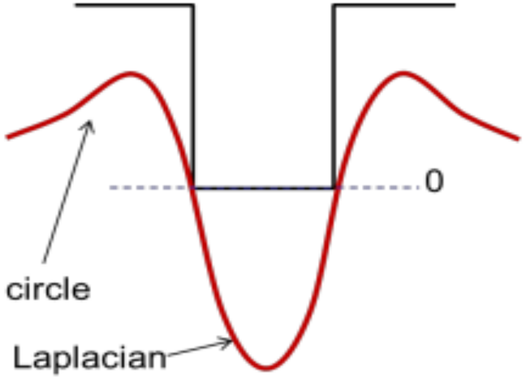
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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



image



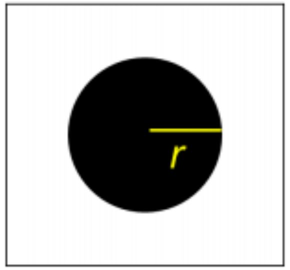
- usually results in **strong positive responses** for dark blobs of radius r=\sqrt {2t} (for a two-dimensional image, r=\sqrt {dt} for a d-dimensional image)
- and **strong negative responses for bright blobs** of similar size.

Scale selection

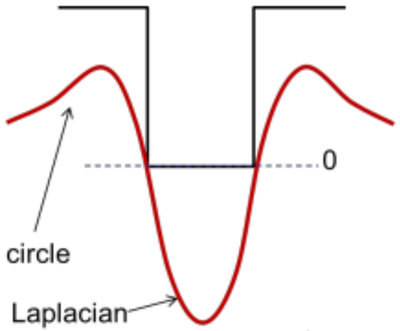
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

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

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

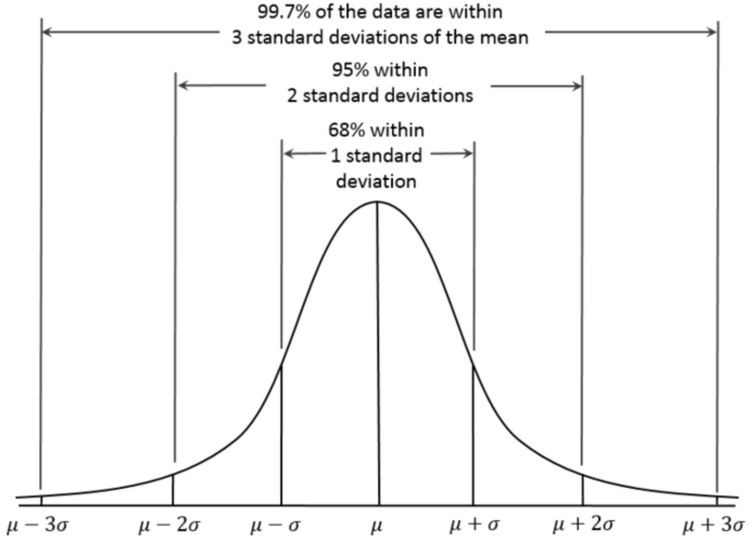


image



$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

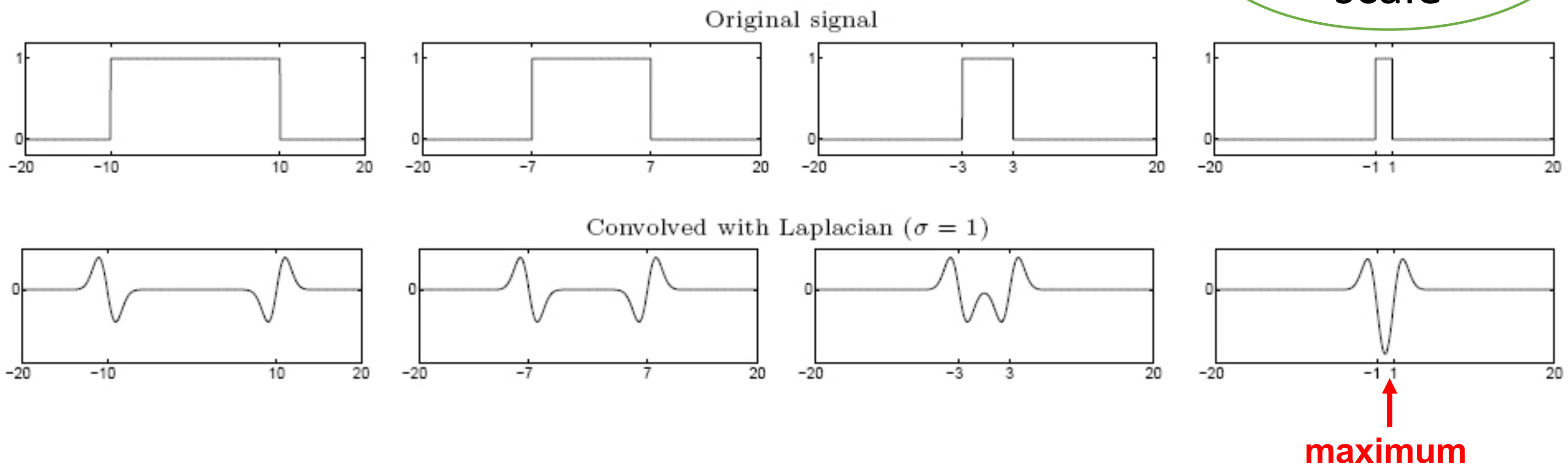
$$\nabla^2 G(x, y; \sigma) = \left(\frac{x^2 + y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G(x, y; \sigma)$$



From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

Small scale



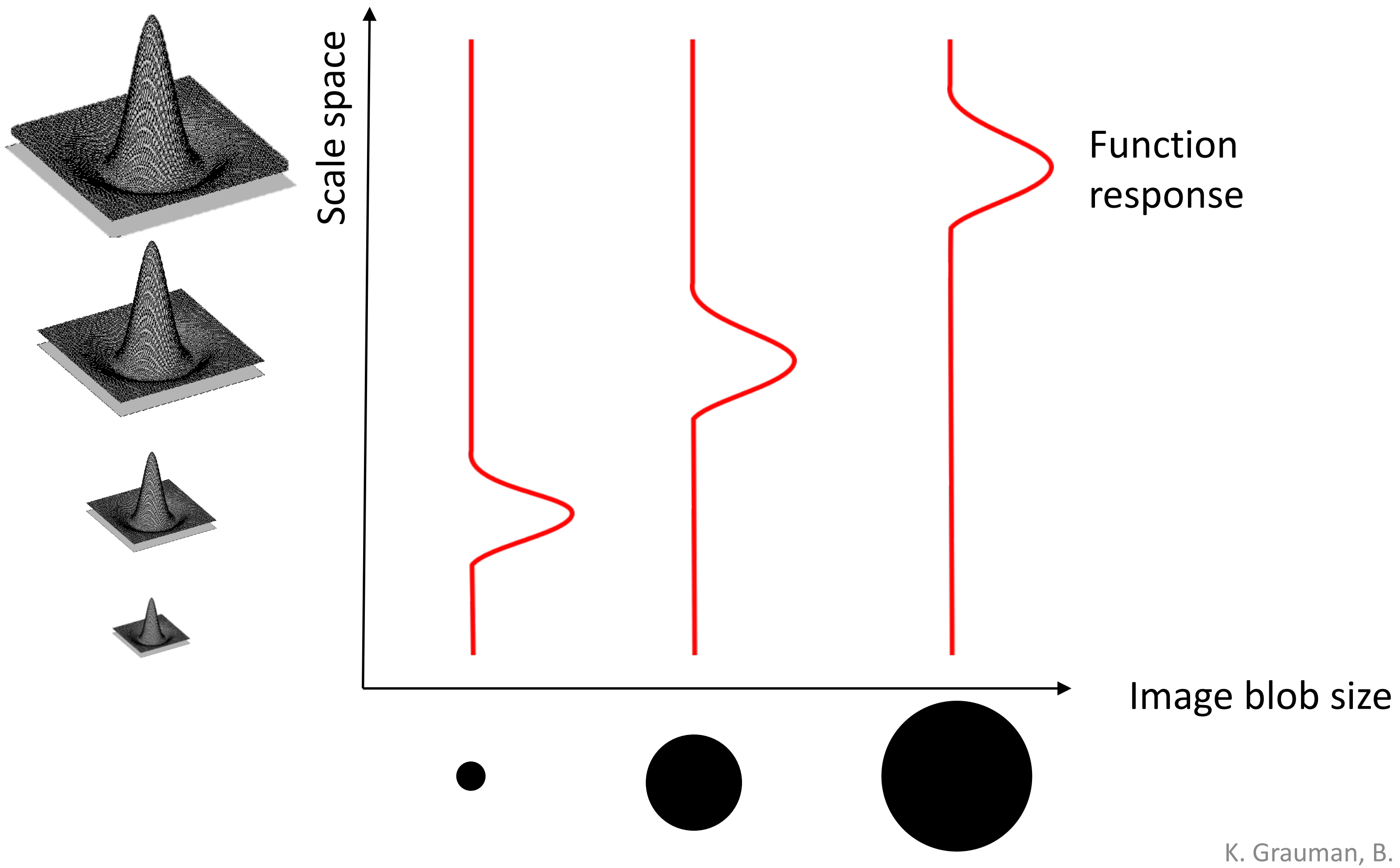
Scale selection: the **magnitude** of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

Slide credit: Lana Lazebnik

What Is A Useful Signature Function f ?

"Blob" detector is common for corners

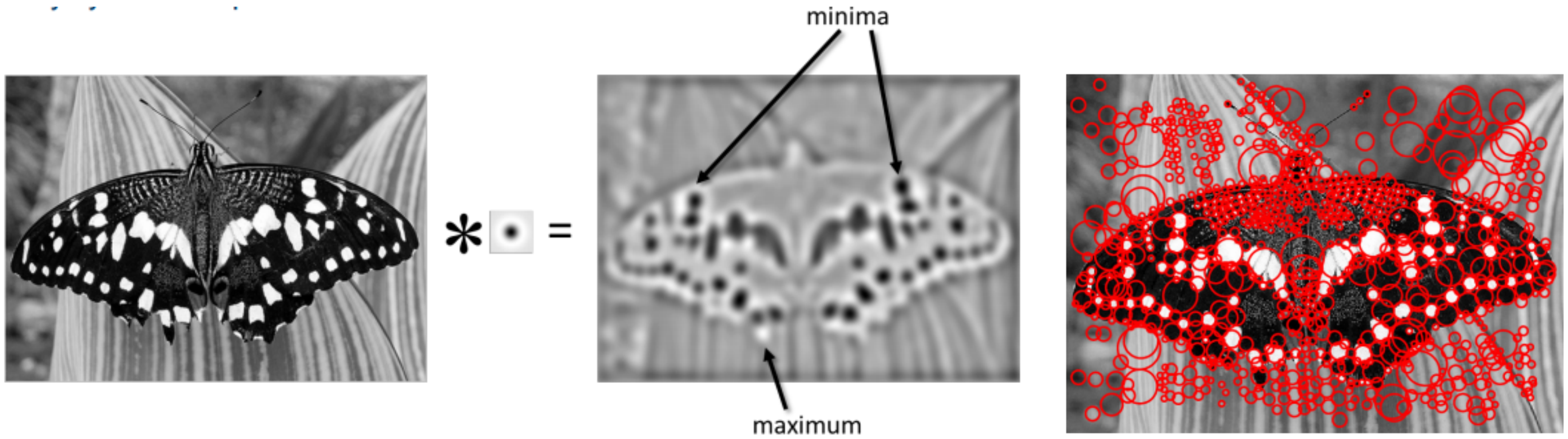
- Laplacian (2nd derivative) of Gaussian (LoG)



K. Grauman, B. Leibe

Laplacian of Gaussian

- Circularly symmetric operator for blob detection in 2D



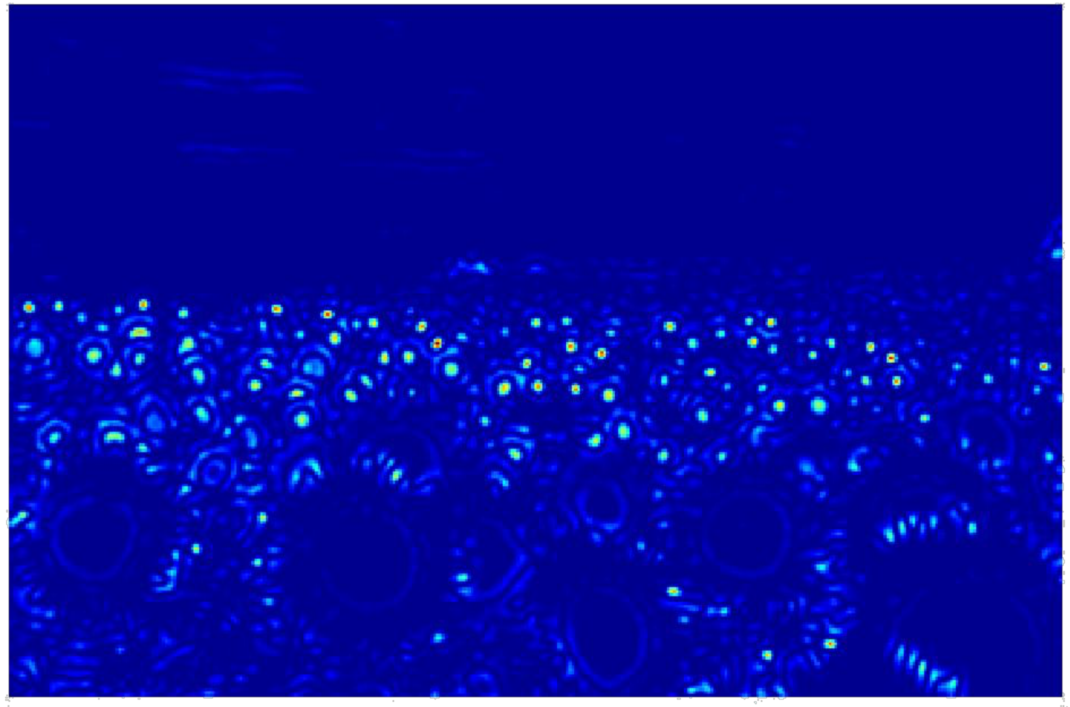
- Find maxima & minima of LoG operator in space and scale

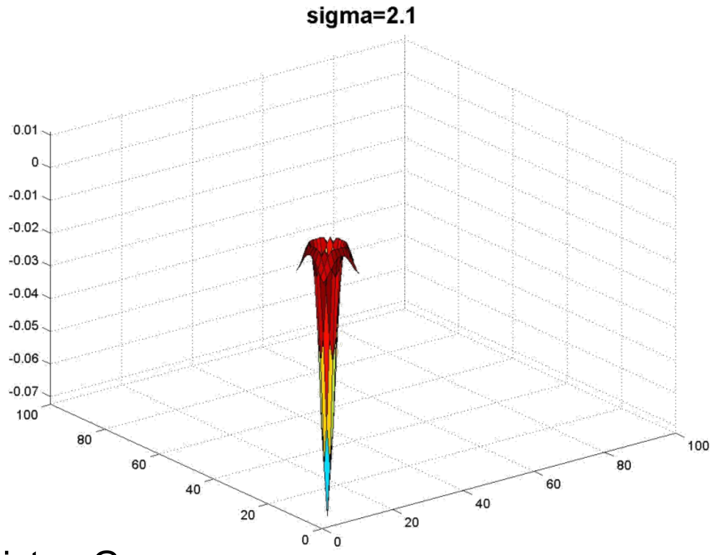
Example

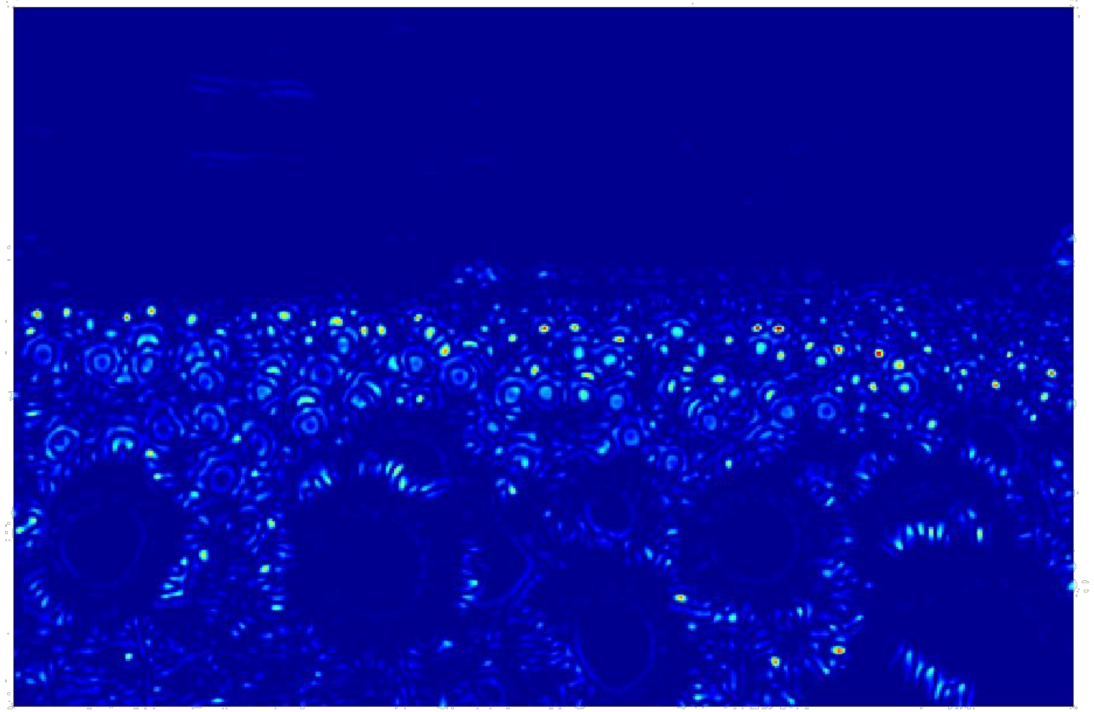
Original image
at ¾ the size

Kristen Grauman

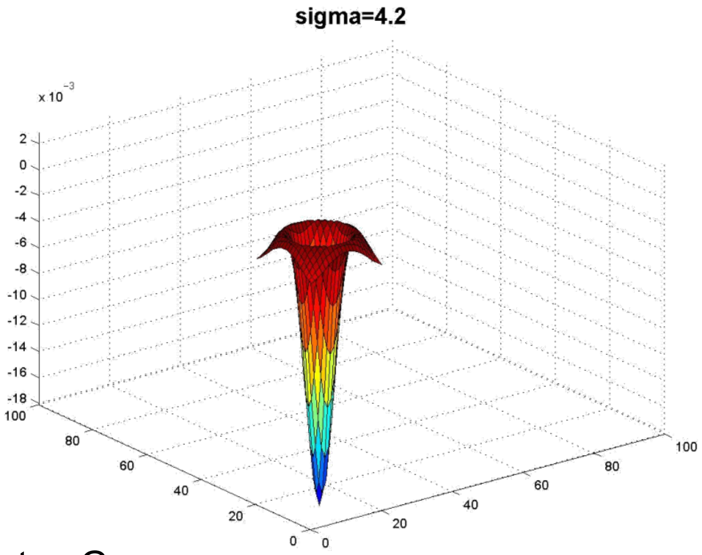
Original image
at ¾ the size

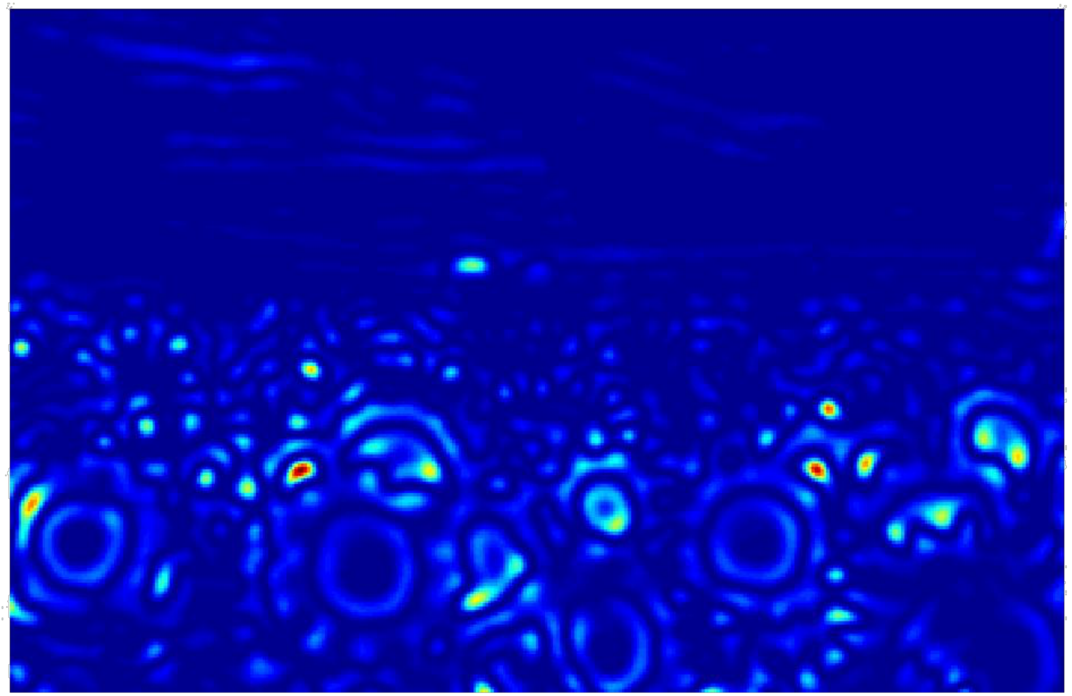


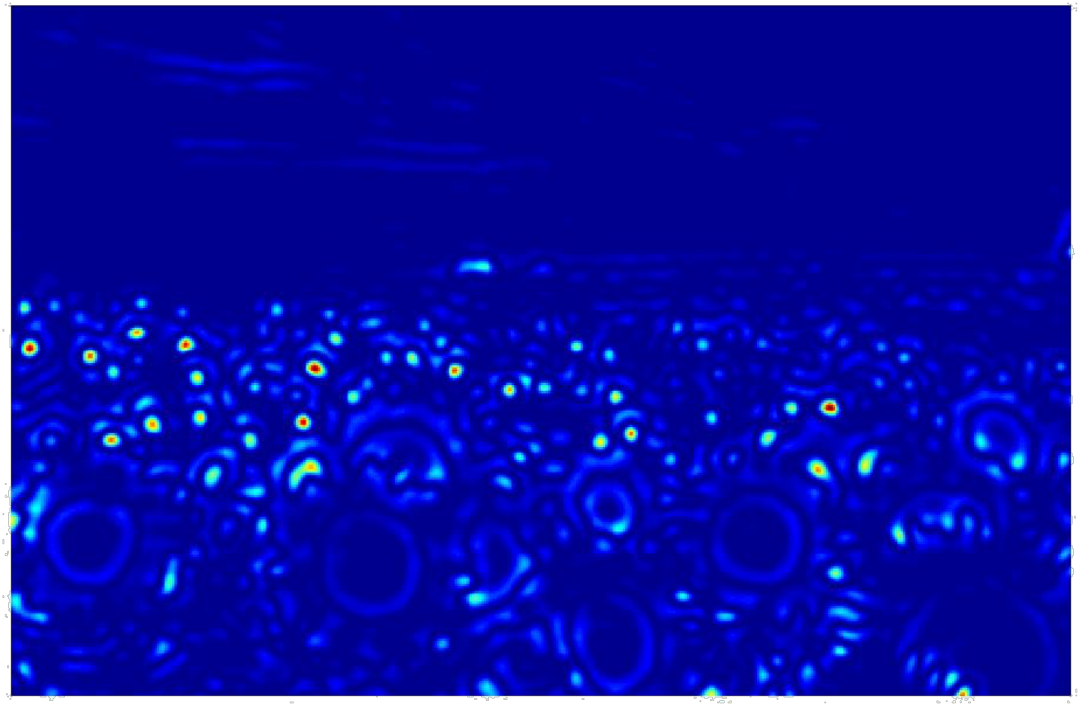




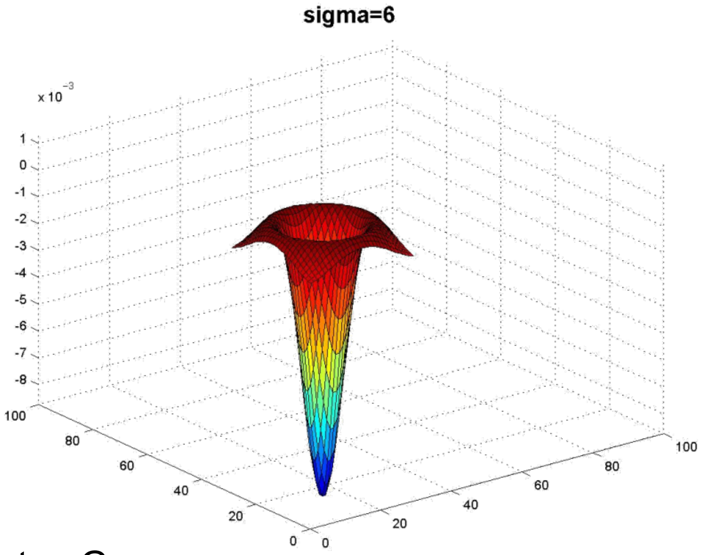
Kristen Grauman



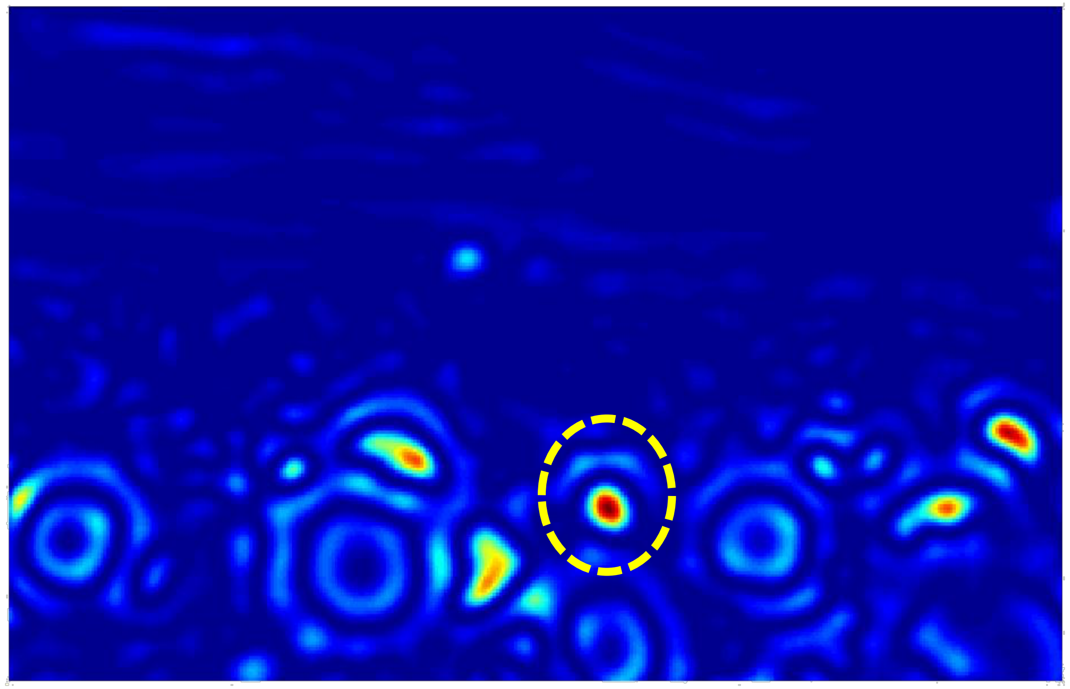


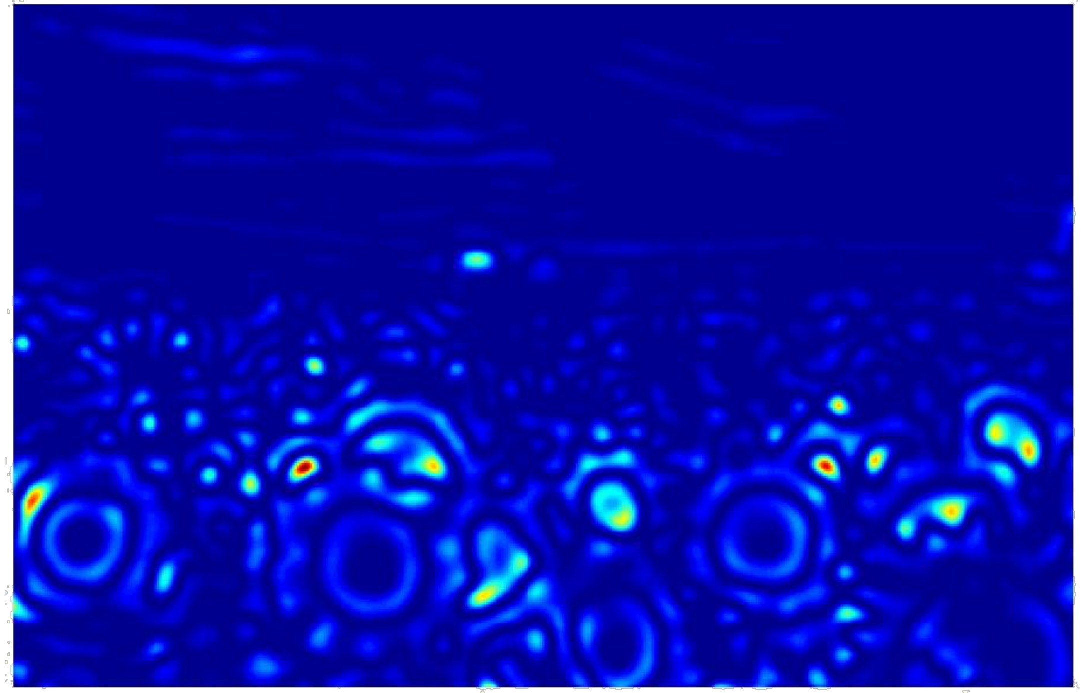


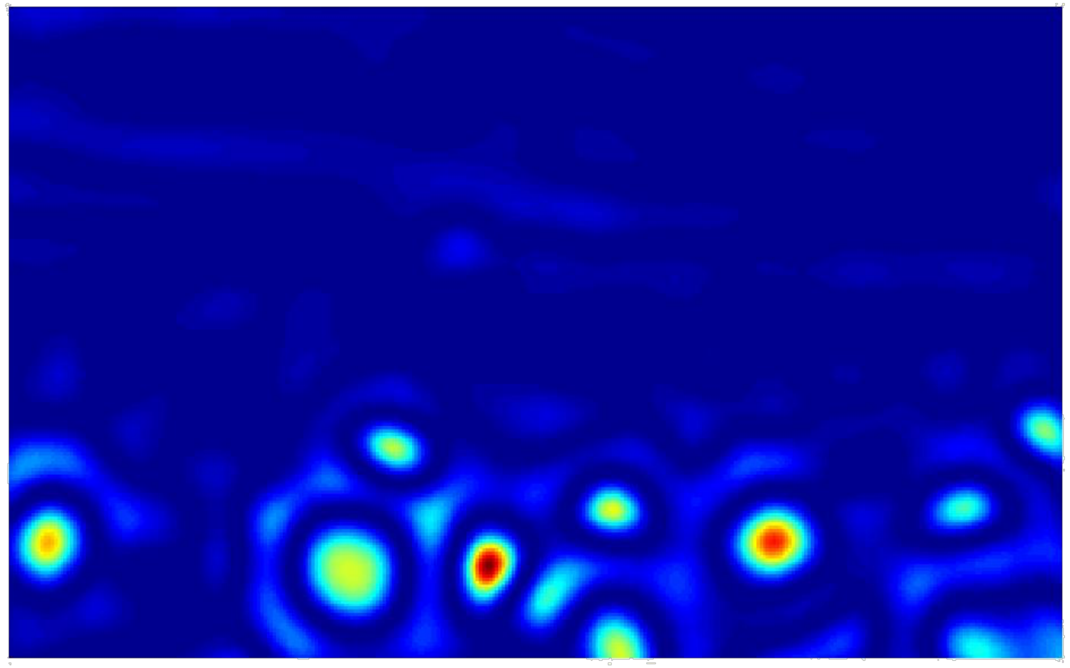
Kristen Grauman

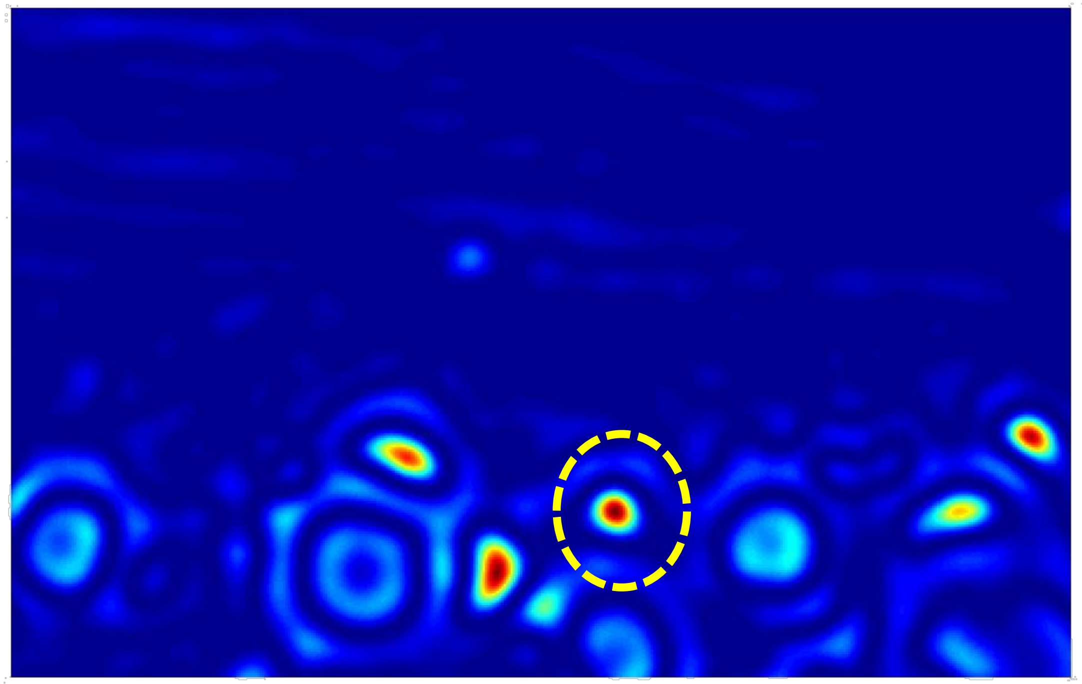


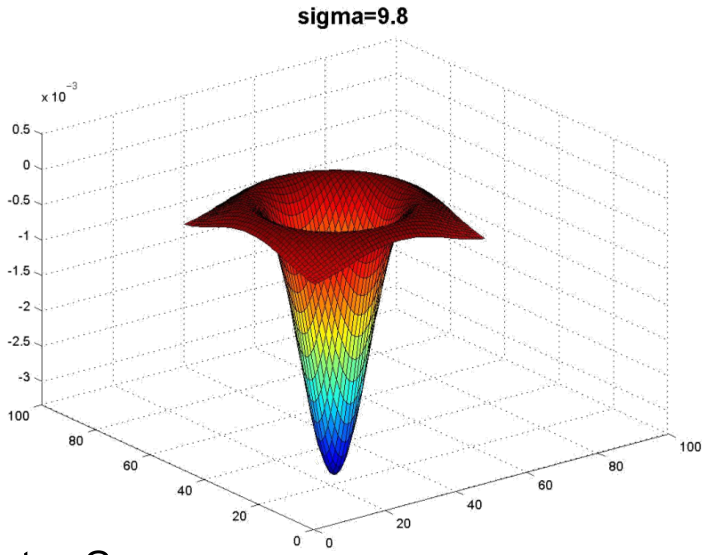
Kristen Grauman



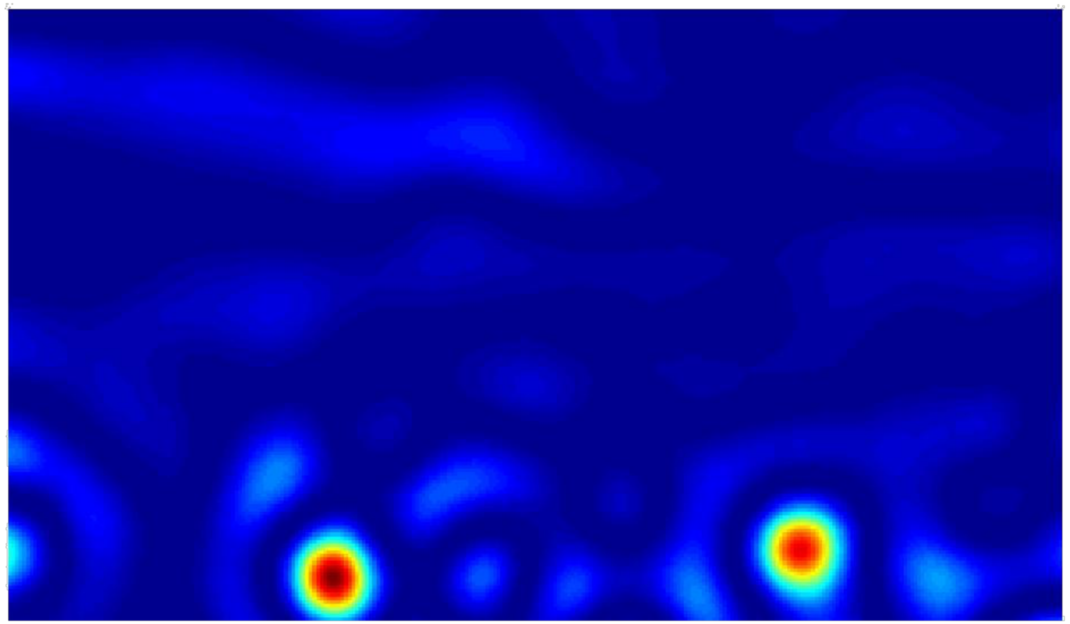


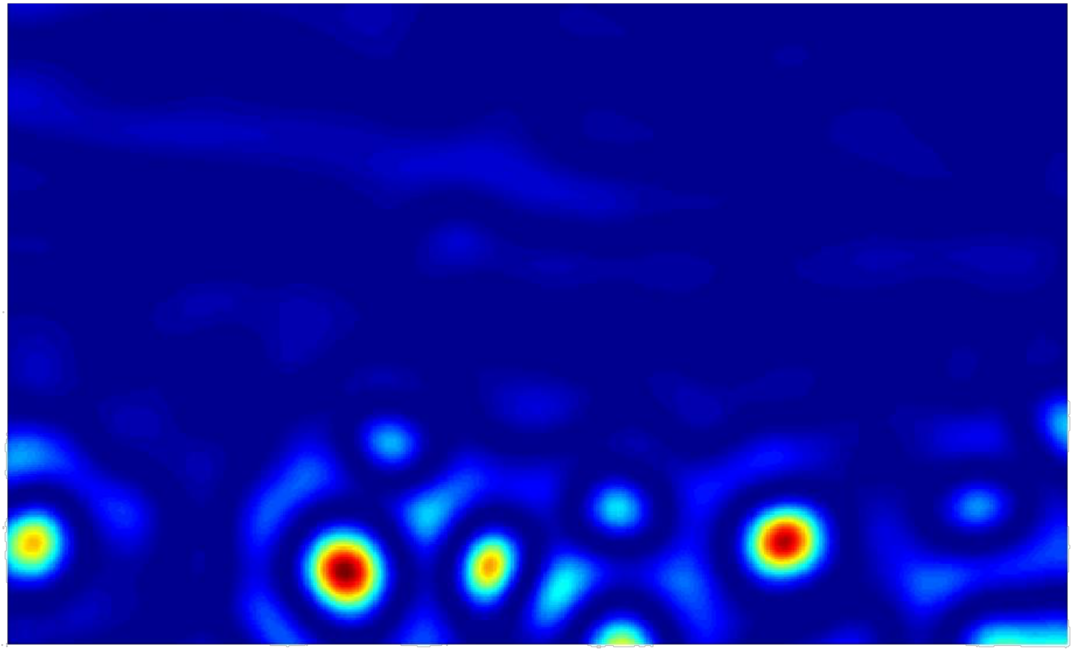


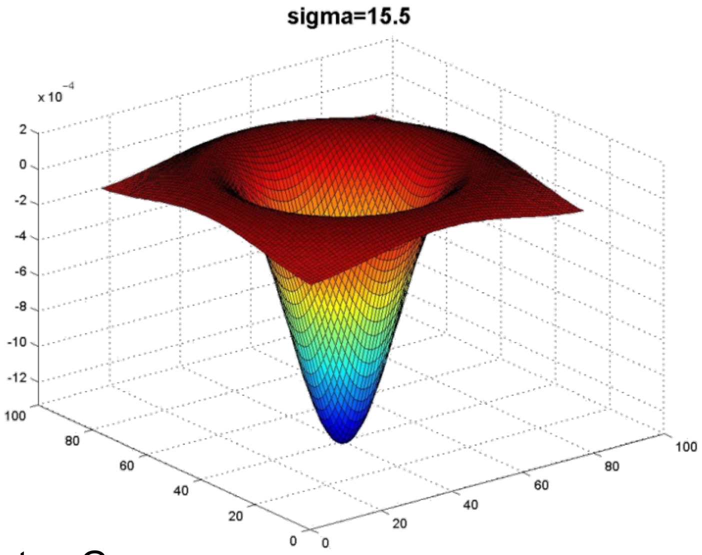




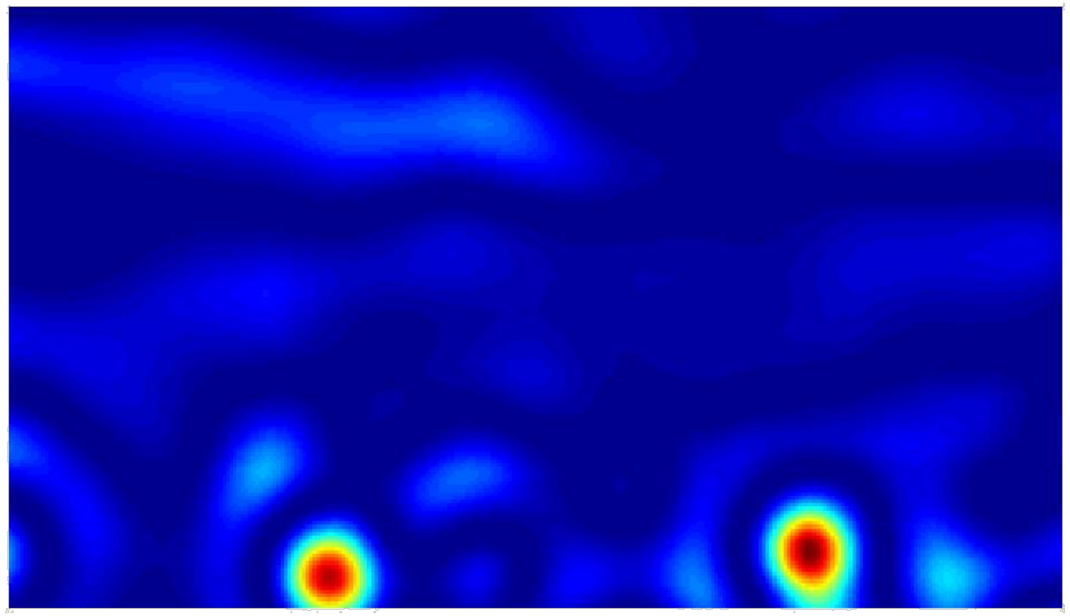
Kristen Grauman

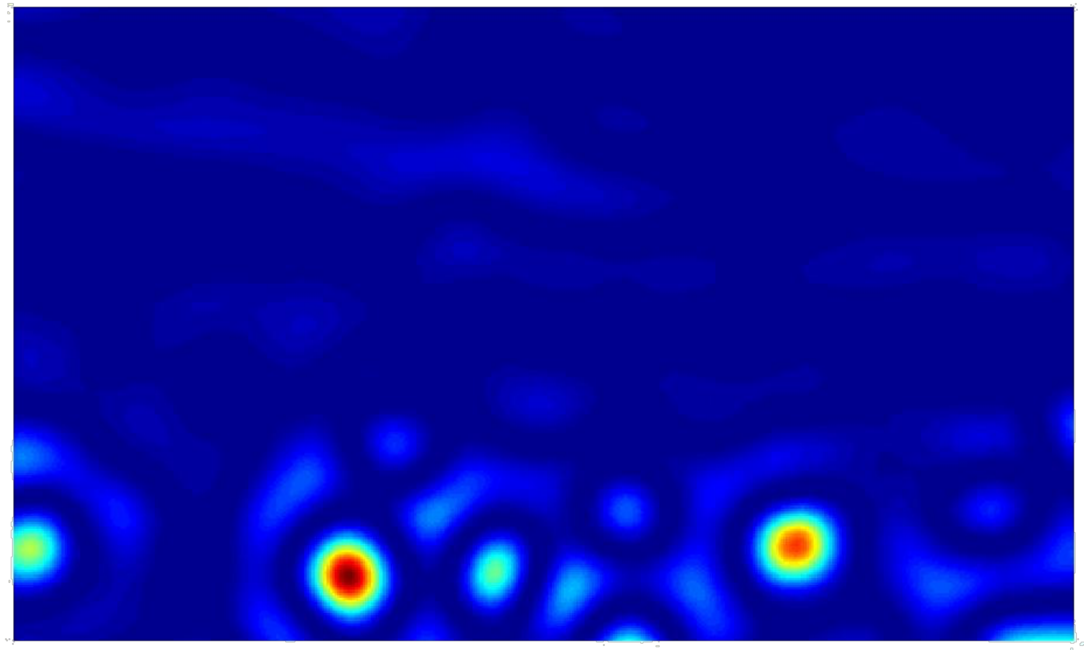


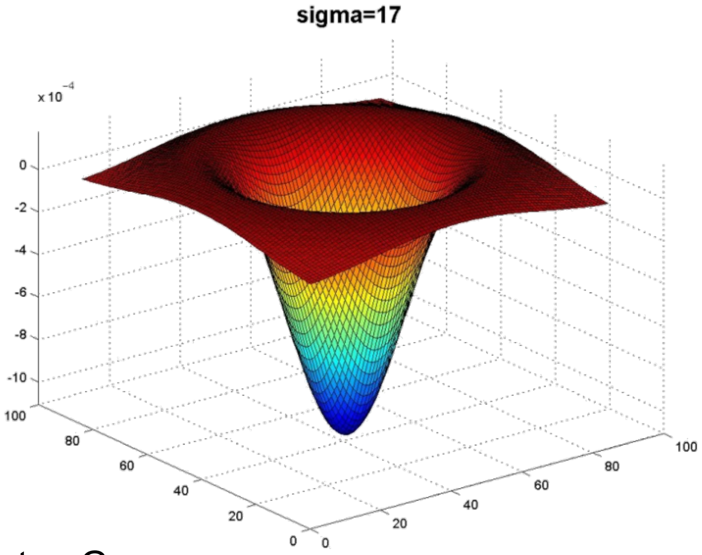




Kristen Grauman



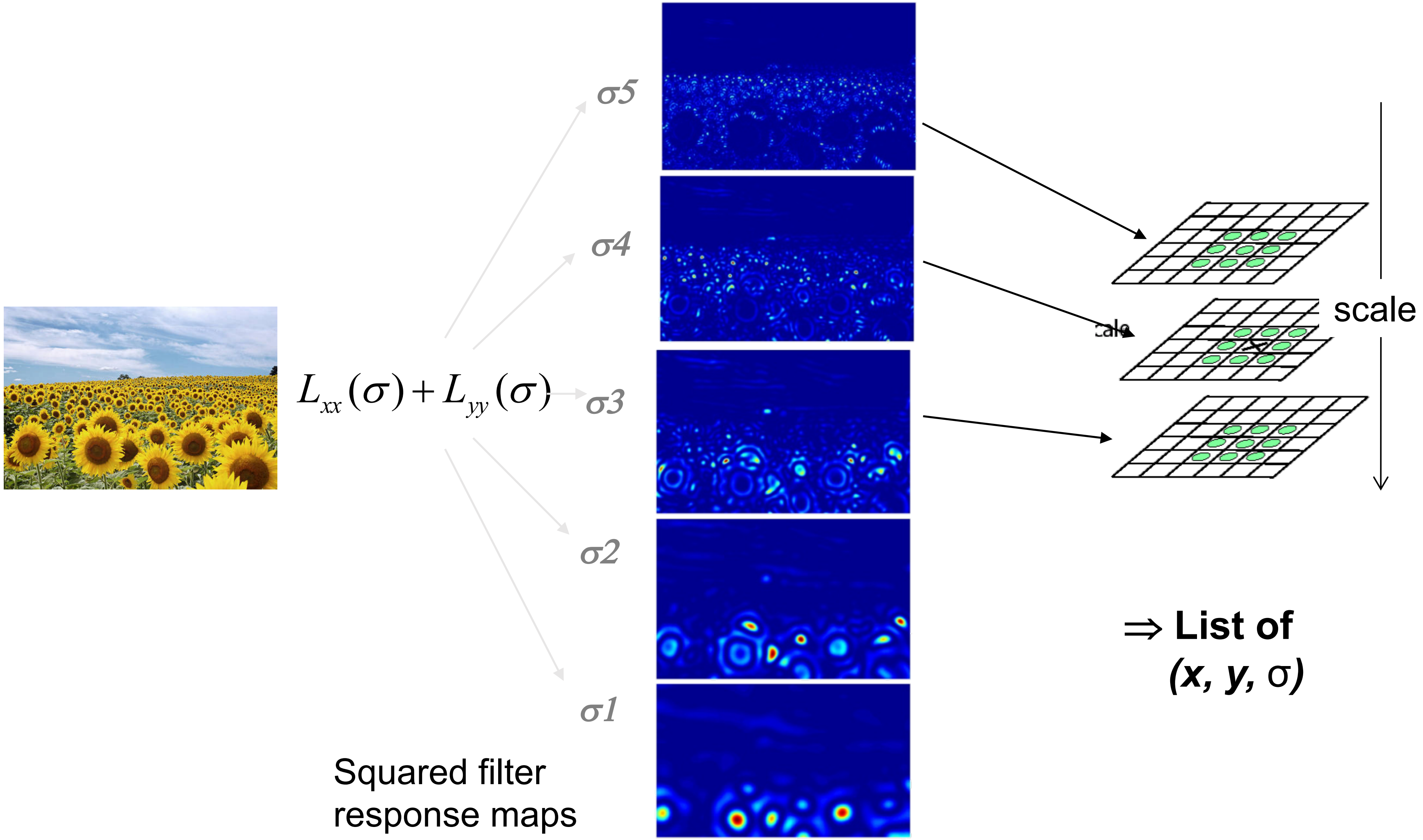




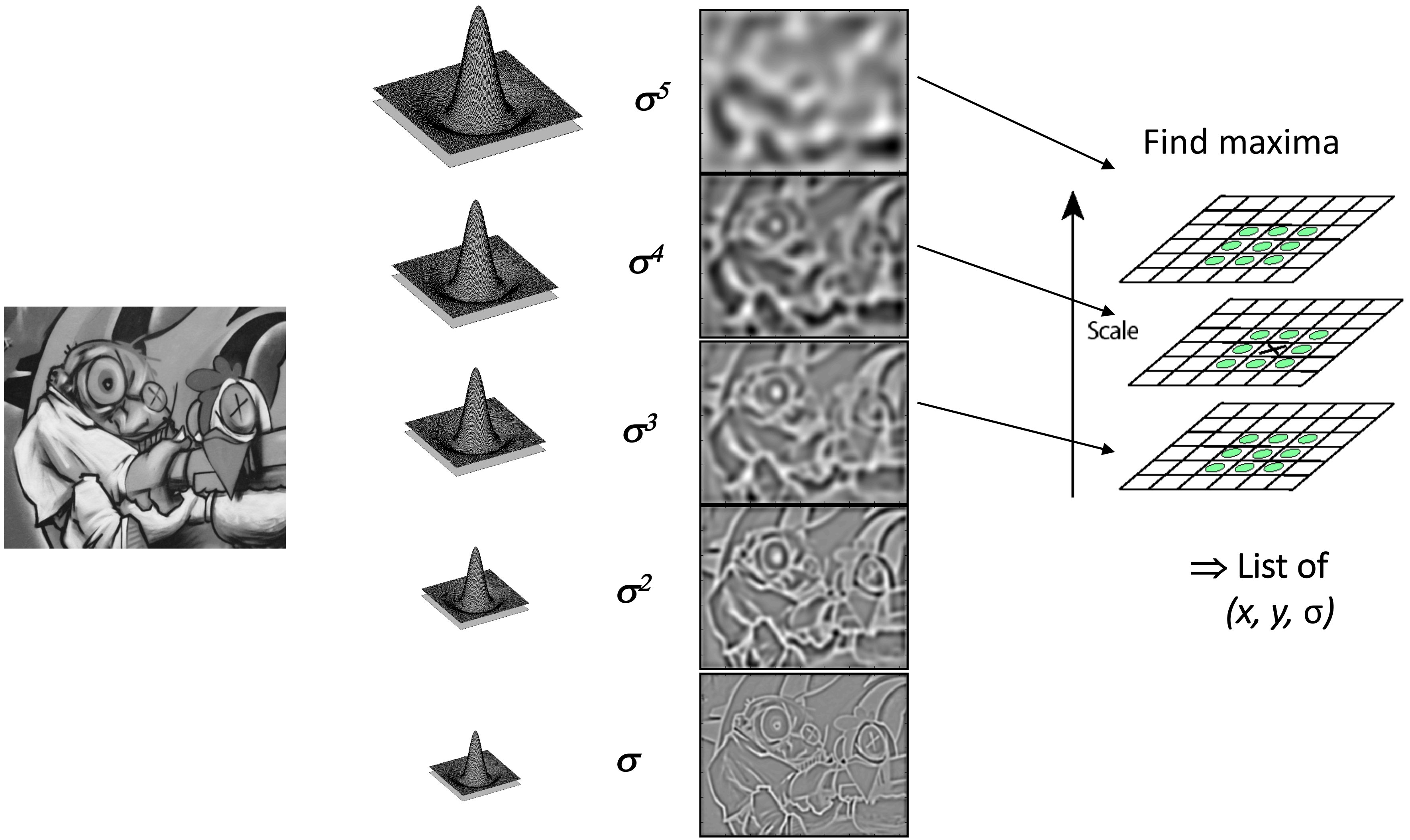
Kristen Grauman

Scale invariant interest points

Interest points are local maxima in both position and scale.

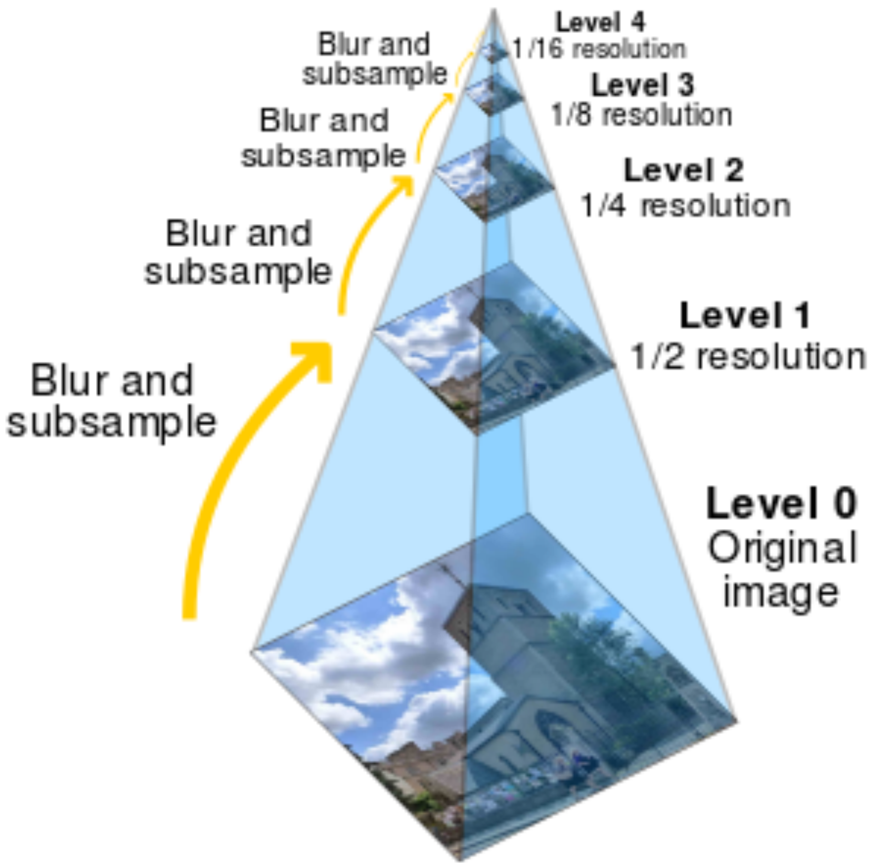


Find local maxima in position-scale space



K. Grauman, B. Leibe

Laplacian pyramid



Like a Gaussian pyramid (left), but subtract layers from each other before subsampling.

Technical detail - Efficient implementation

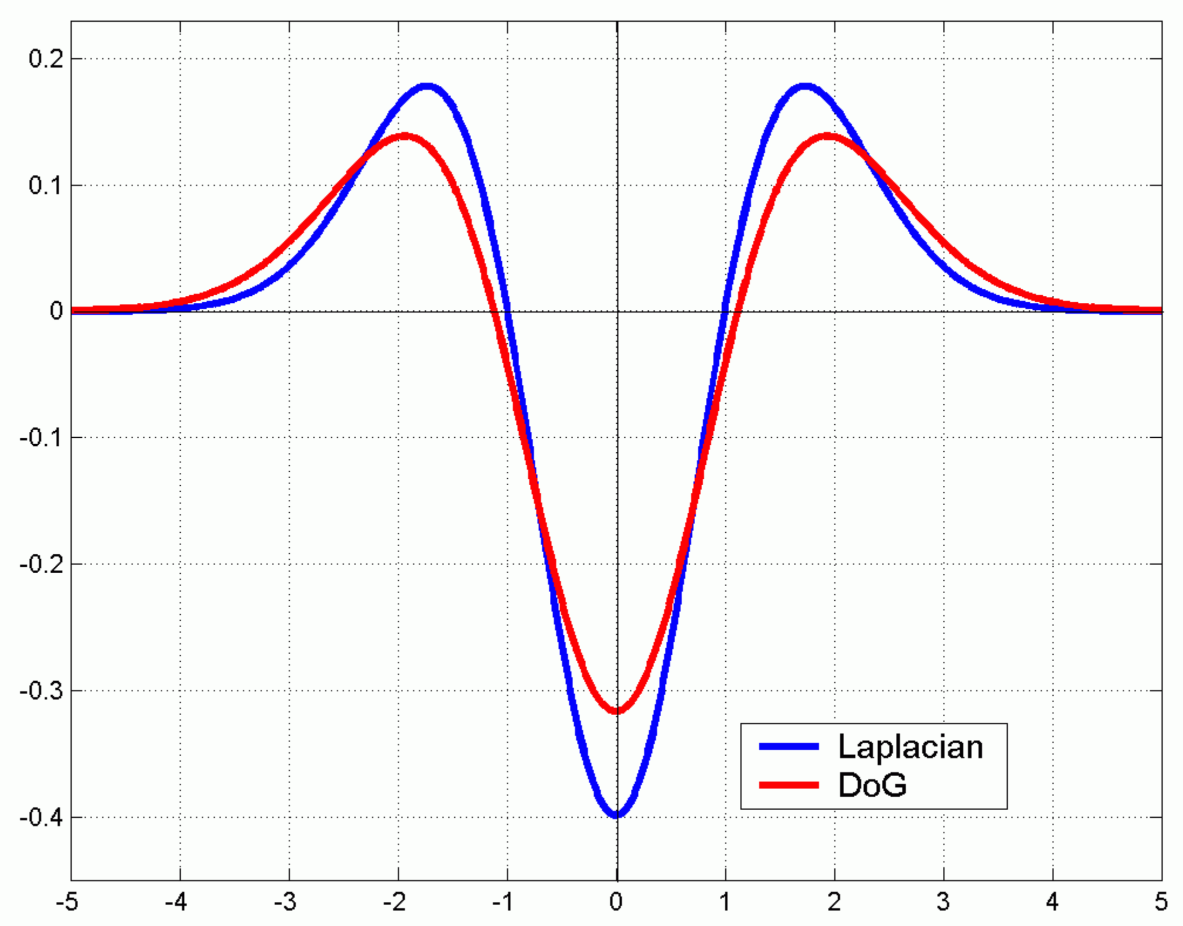
- We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

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

(LOG: Laplacian)

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

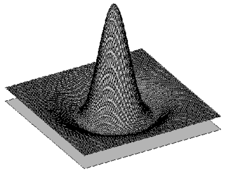
(DOG: Difference of Gaussians)



Alternative kernel

Approximate LoG with Difference-of-Gaussian (DoG).

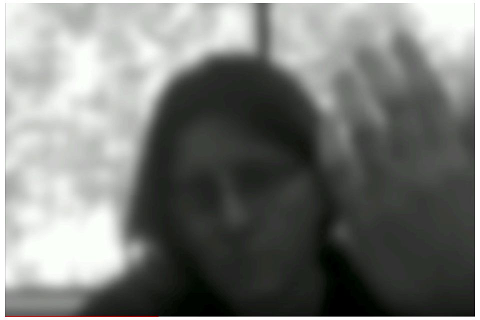
1. Blur image with   σ Gaussian kernel
2. Blur image with *k*σ Gaussian kernel
3. Subtract 2. from 1.



K. Grauman, B. Leibe

Example of gaussian kernel

Original video



Blurred with a gaussian kernel

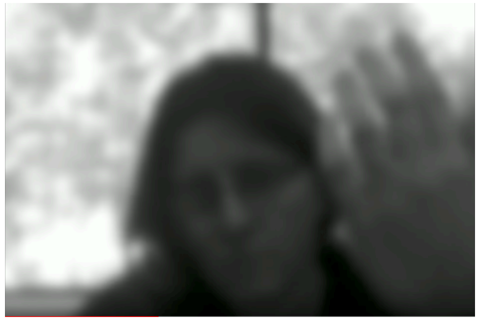
Blurred with a different gaussian kernel

What happens if you subtract one blurred image from another?

source

Difference of Gaussians (DoG)

Original video

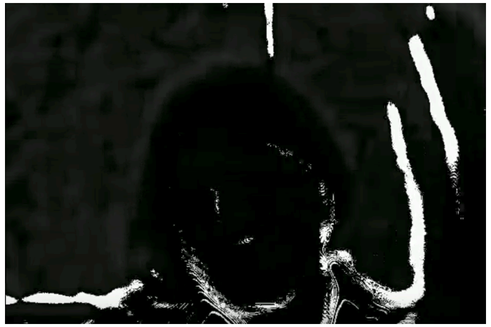


Blurred with a gaussian kernel: k_1

Blurred with a different gaussian kernel: k_2

DoG: $k_1 - k_2$

DoG: $k_1 - k_3$

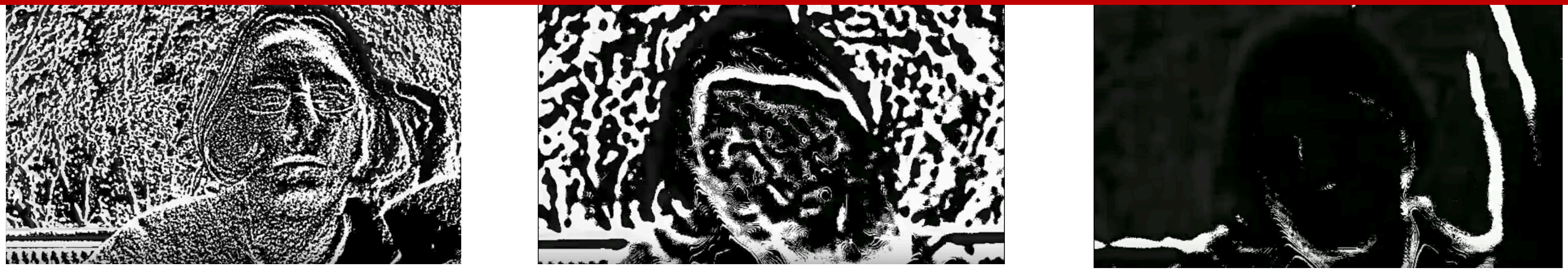


DoG: $k_1 - k_4$

source

Difference of Gaussians (DoG)

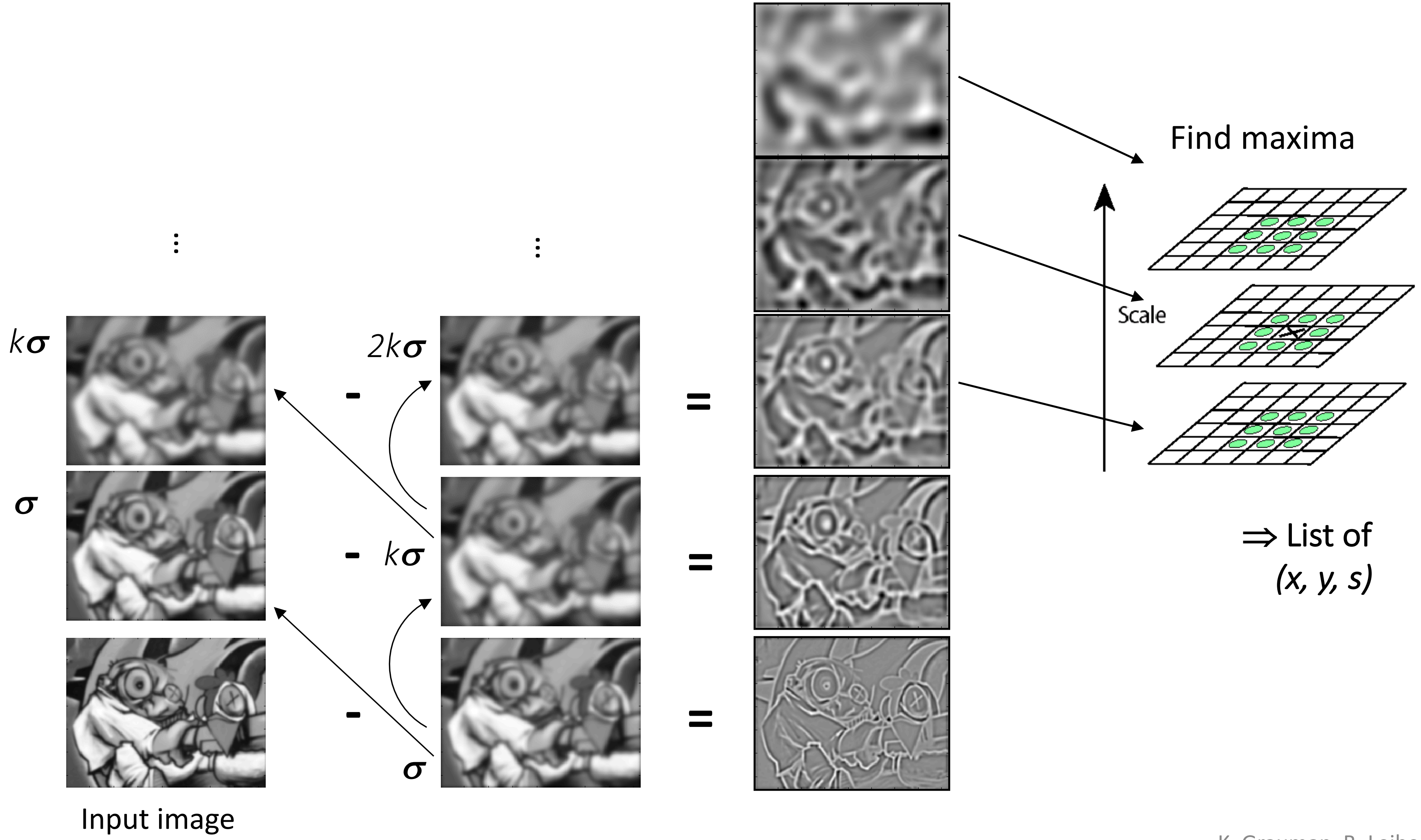
At different resolutions of kernel size, we see different fine details of the image. In other words, we can capture keypoints at varying scales.



DoG: $k_1 - k_2$ DoG: $k_1 - k_3$ DoG: $k_1 - k_4$

source

Find local maxima in position-scale space of DoG



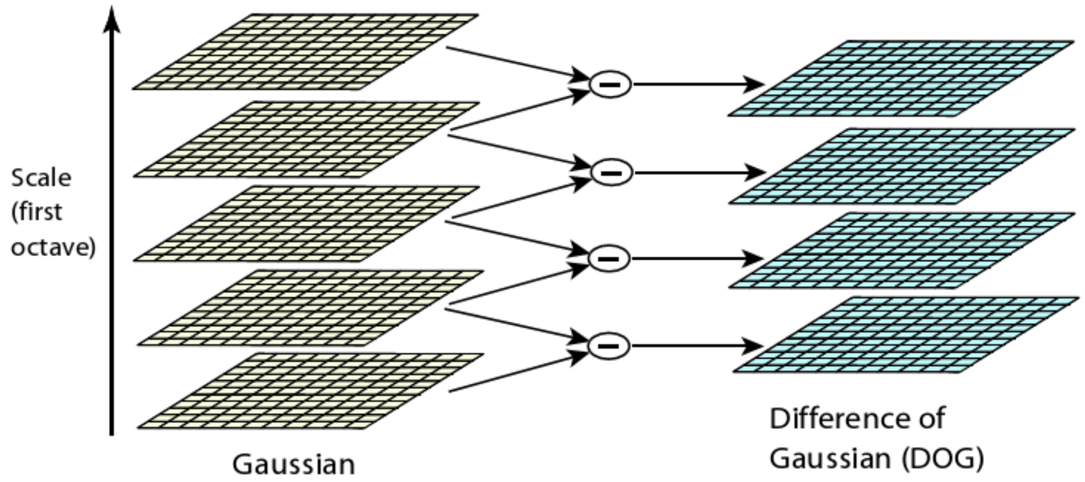
K. Grauman, B. Leibe

Difference-of-Gaussians

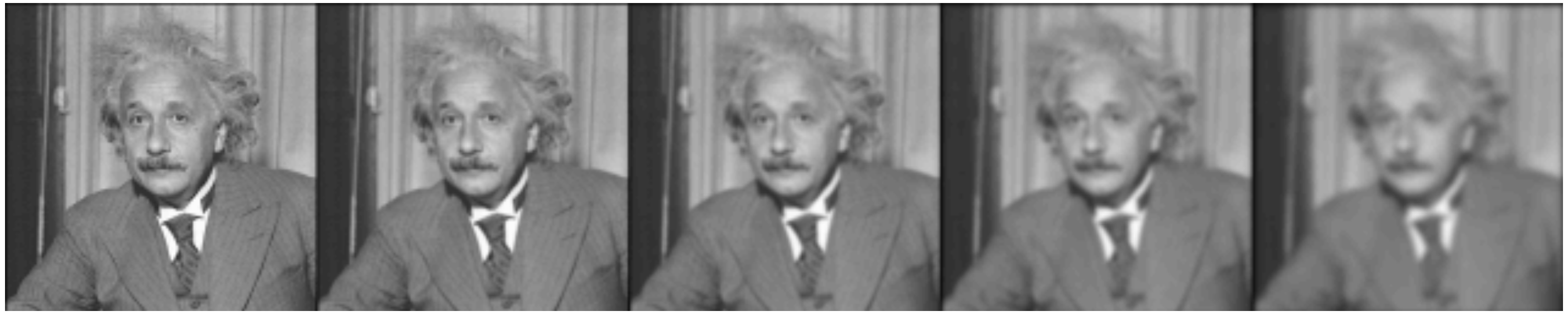
$G(k^2\sigma)*$

$G(k\sigma)*I$

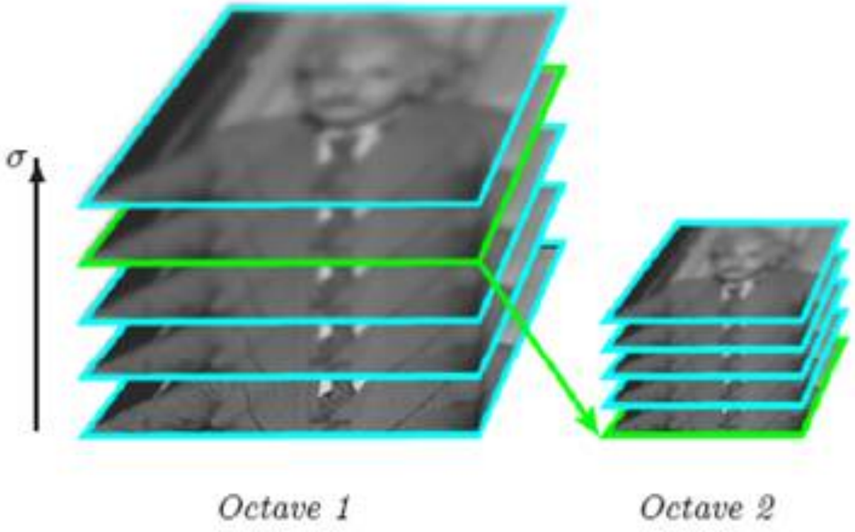
$G(\sigma)*I$

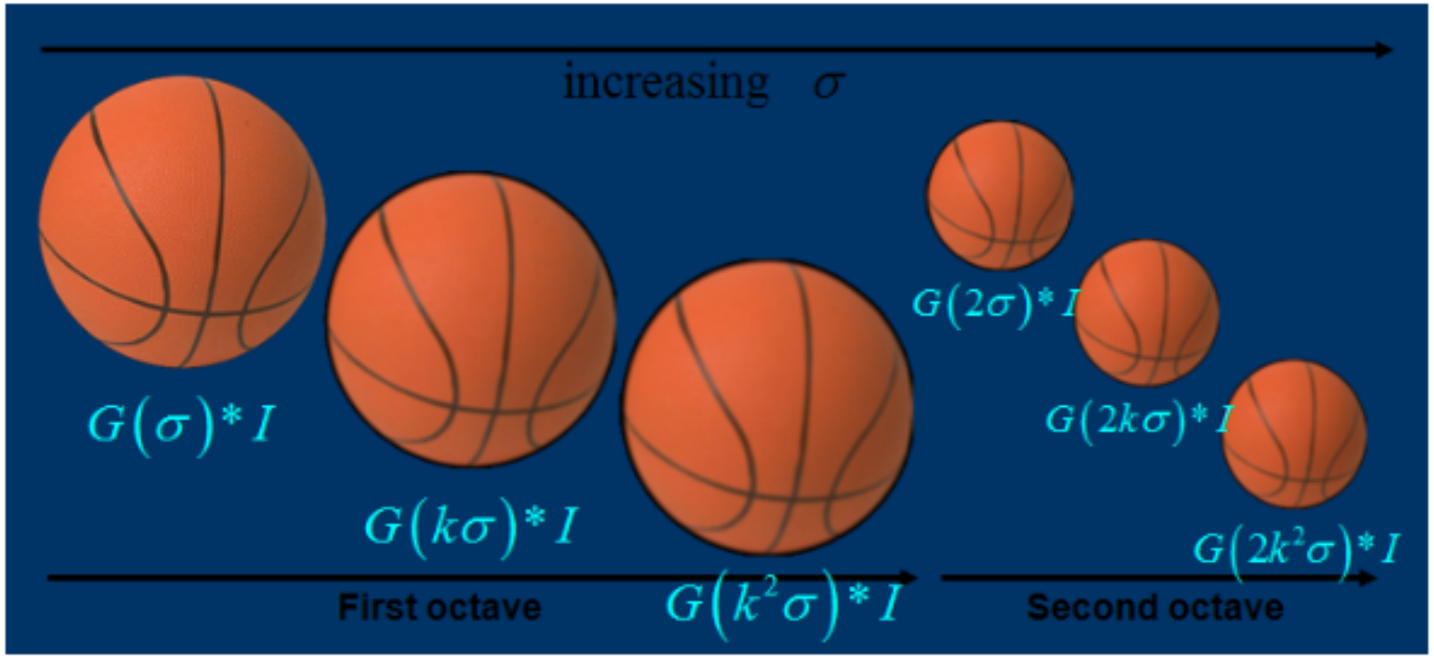


$D(\sigma) \equiv (G(k\sigma) - G(\sigma)) * I$

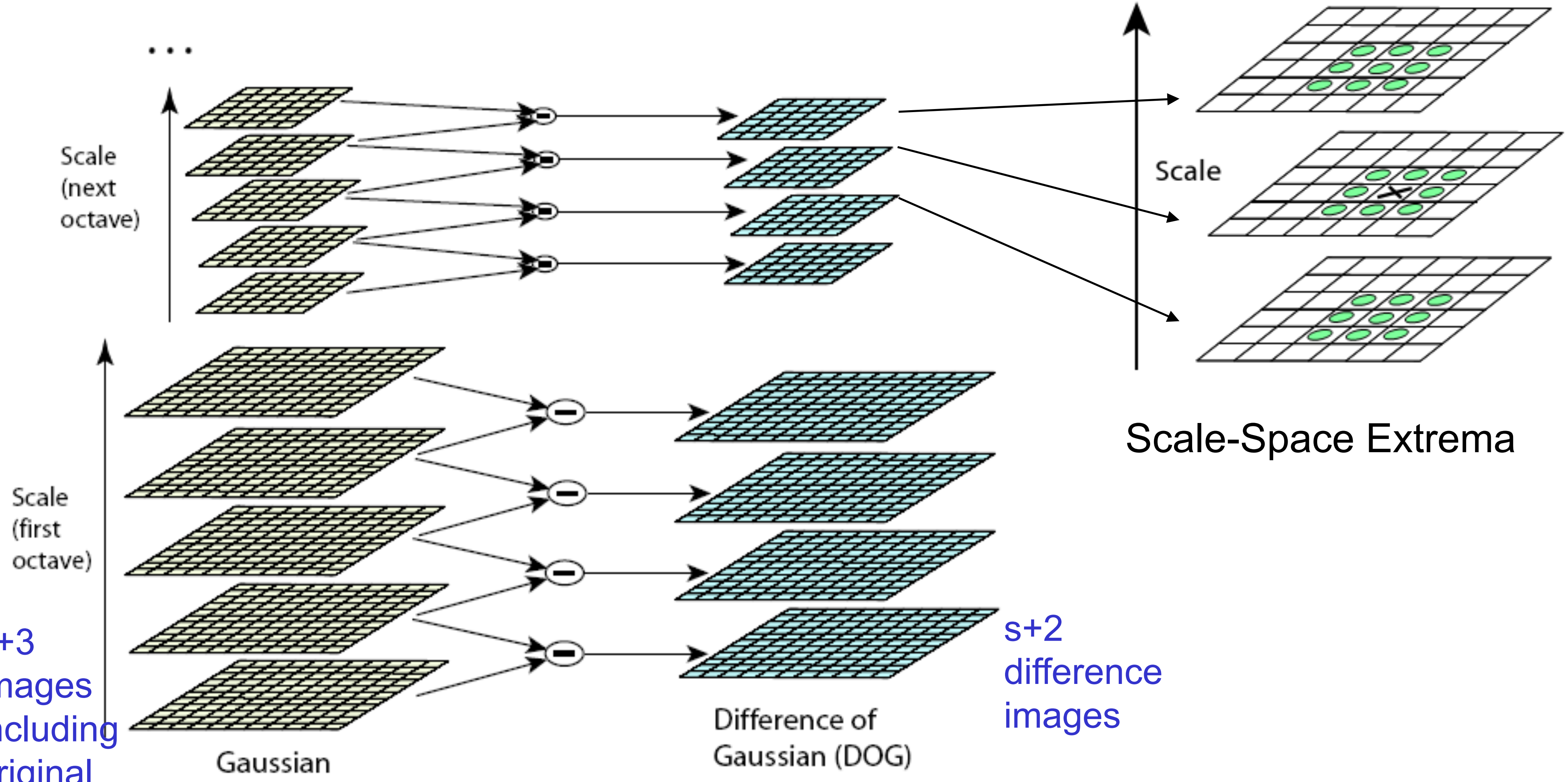


Construct scale space





Construct scale space - Lowe's Pyramid Scheme



s+2 filters

$\sigma_{s+1}=2^{(s+1)/s}\sigma_0$

.

.

$\sigma_i=2^{i/s}\sigma_0$

.

.

$\sigma_2=2^{2/s}\sigma_0$

$\sigma_1=2^{1/s}\sigma_0$

σ_0

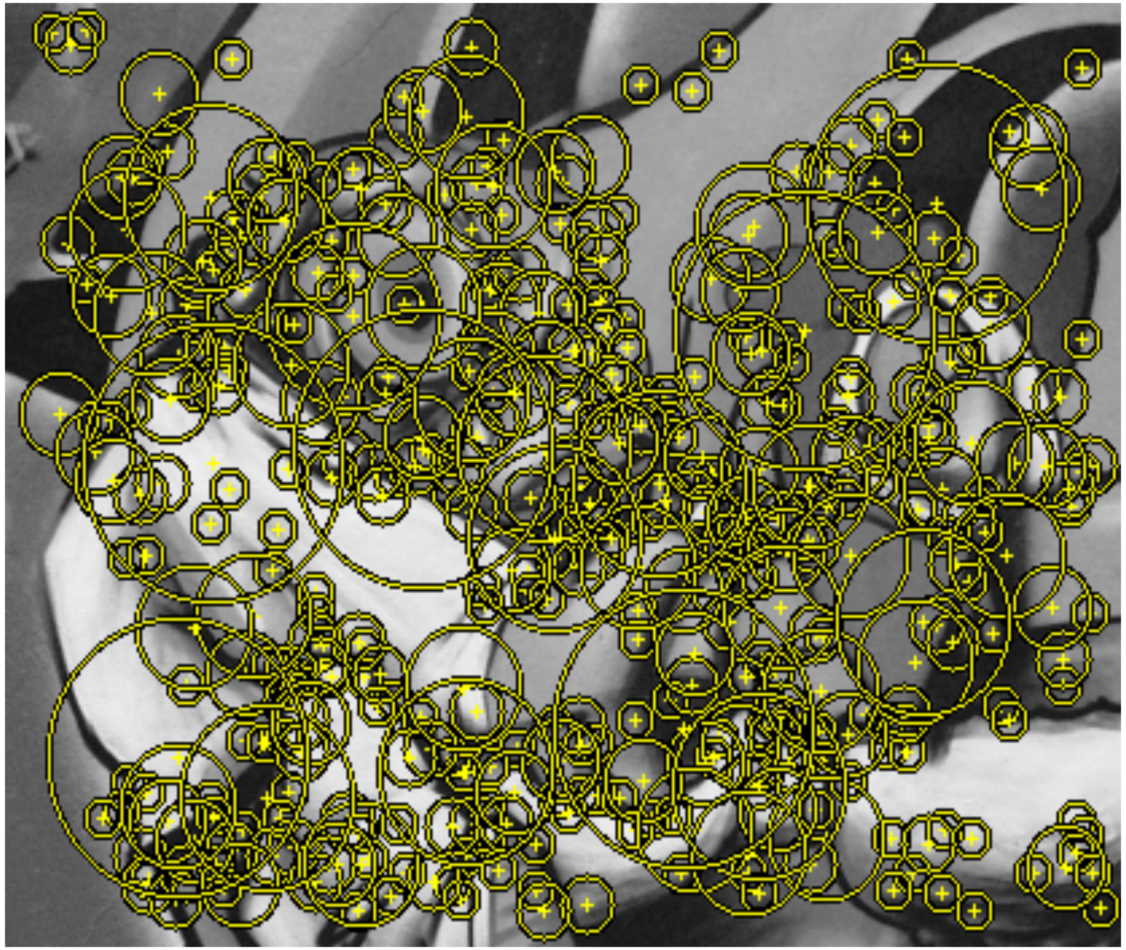
s+3 images including original

s+2 difference images

The parameter **s** determines the number of images per octave.

Results: Difference-of-Gaussian

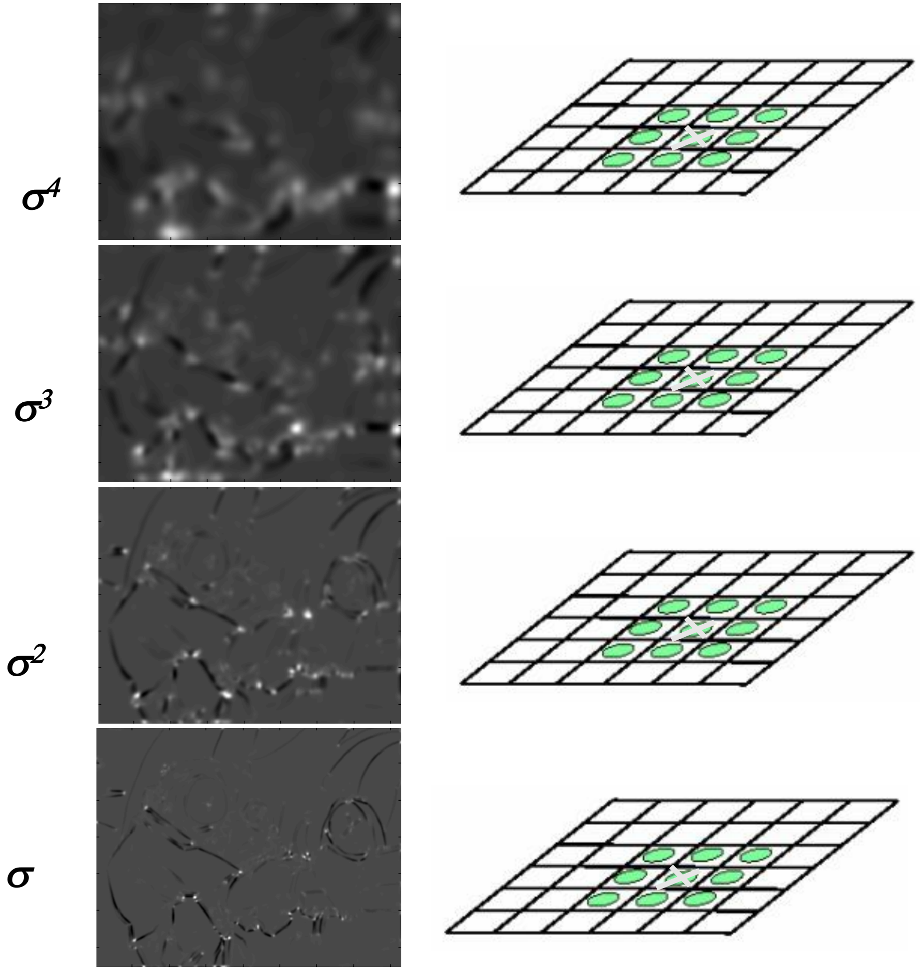
- Larger circles = larger scale
- Descriptors with maximal scale response



K. Grauman, B. Leibe

Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection

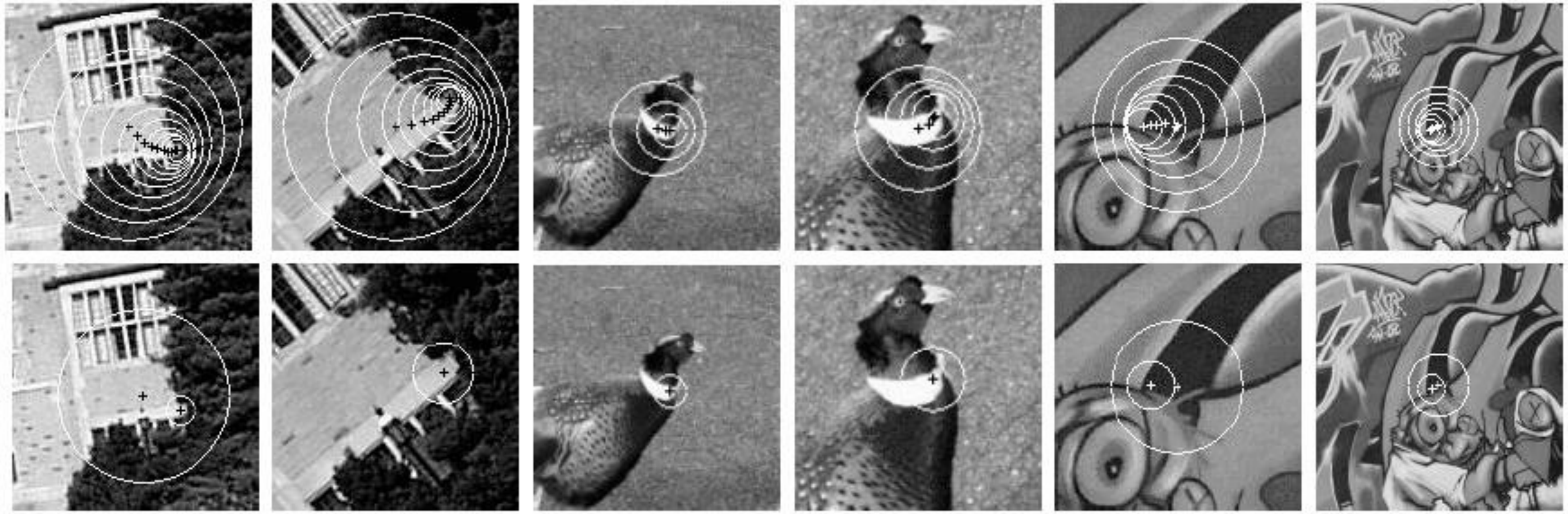


Computing Harris function **Detecting local maxima**

Harris-Laplace [Mikolajczyk '01]

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

Harris points



Harris-Laplace points

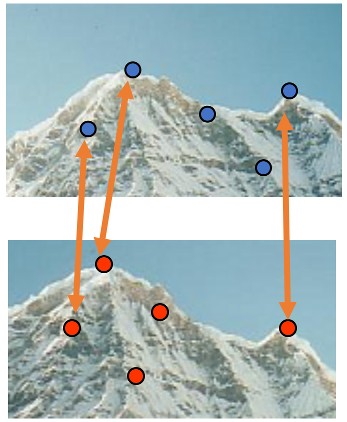
K. Grauman, B. Leibe

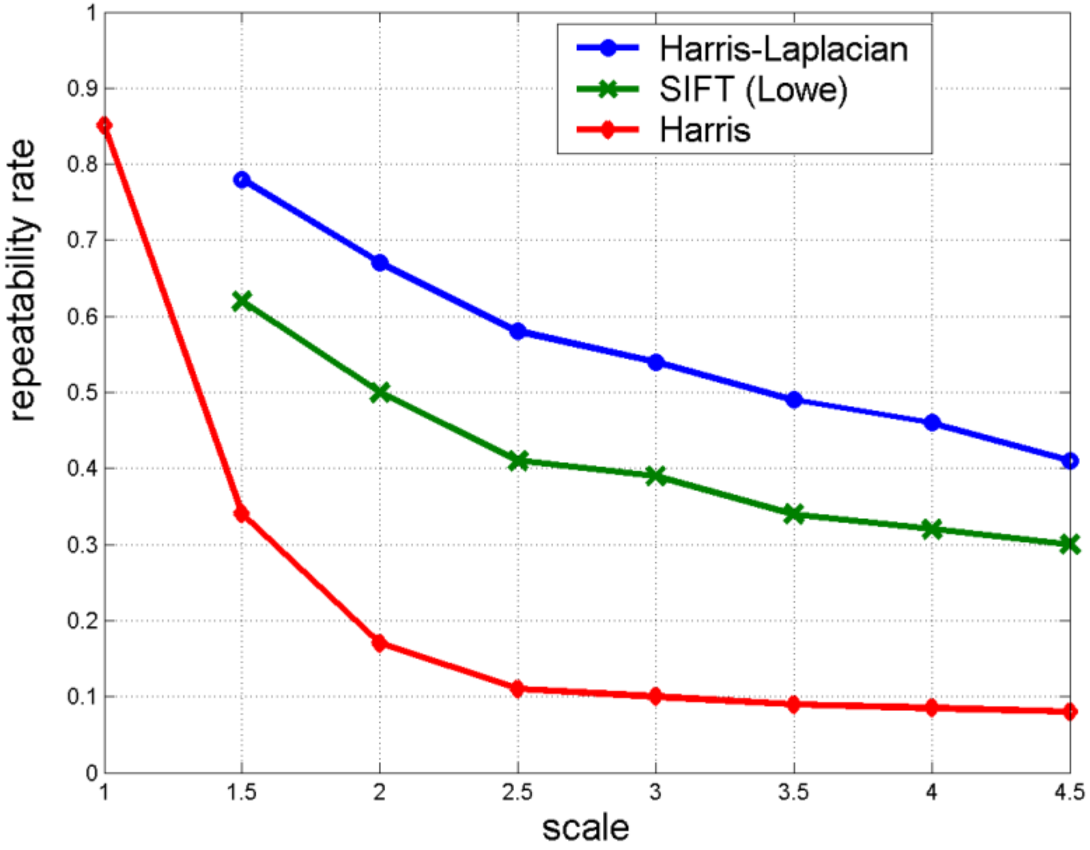
Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$

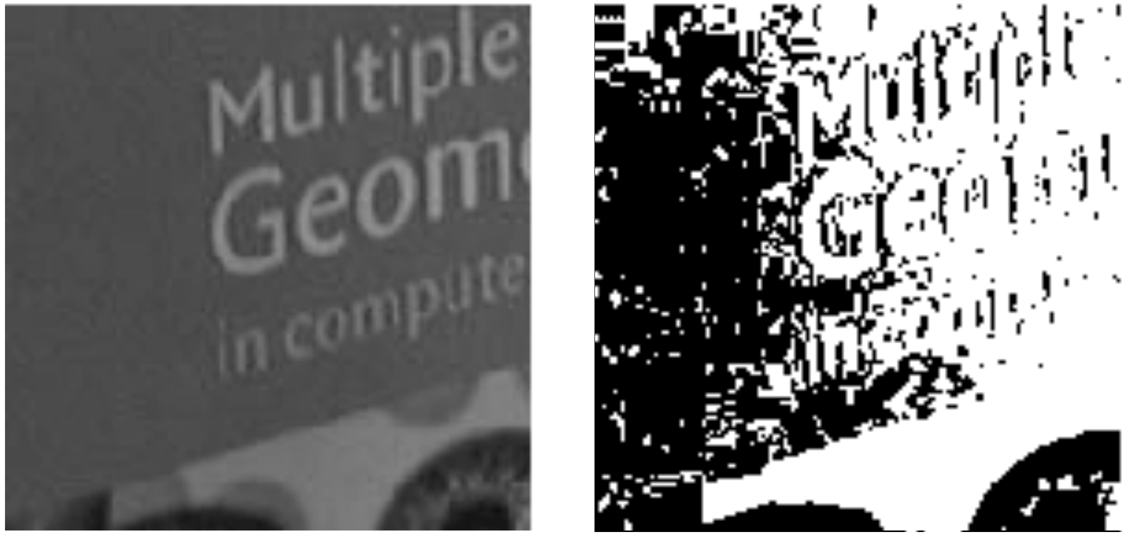




K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

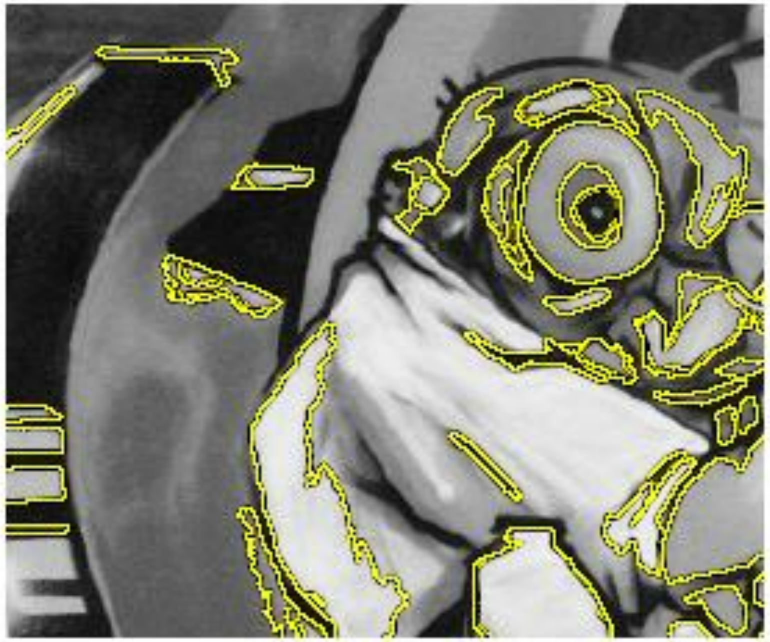
Maximally Stable Extremal Regions [Matas ‘02]

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



[Additional information]

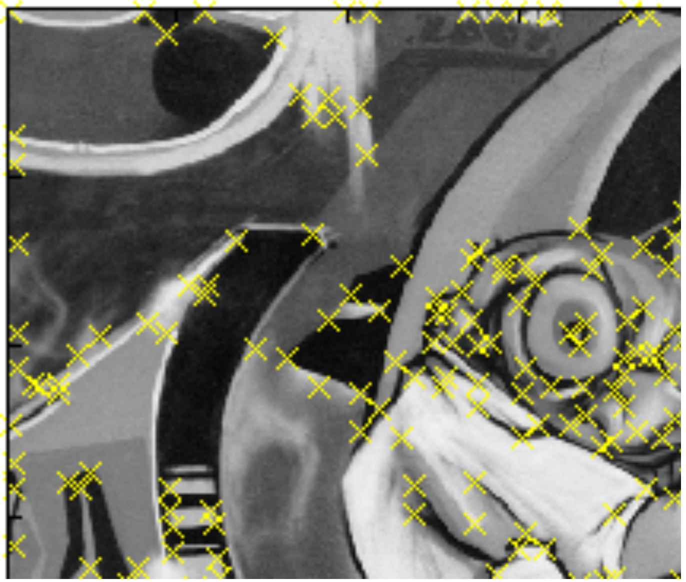
Example Results: MSER



[Additional information]

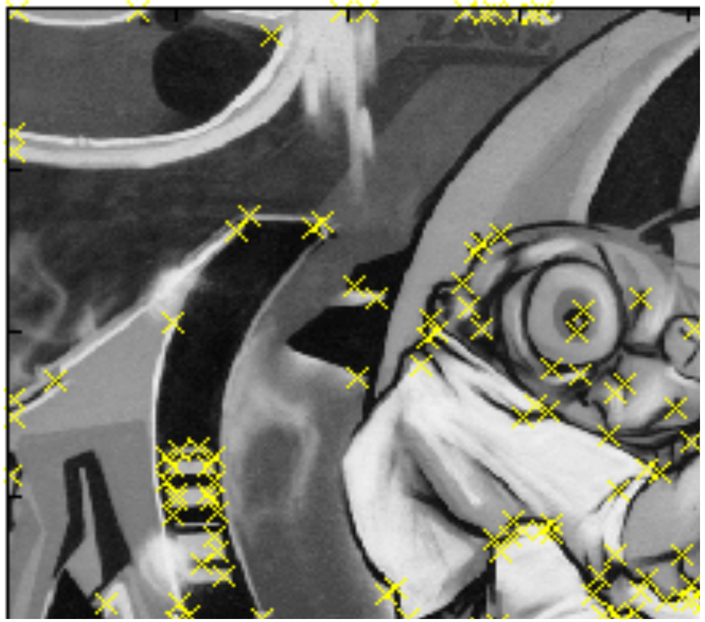
K. Grauman, B. Leibe

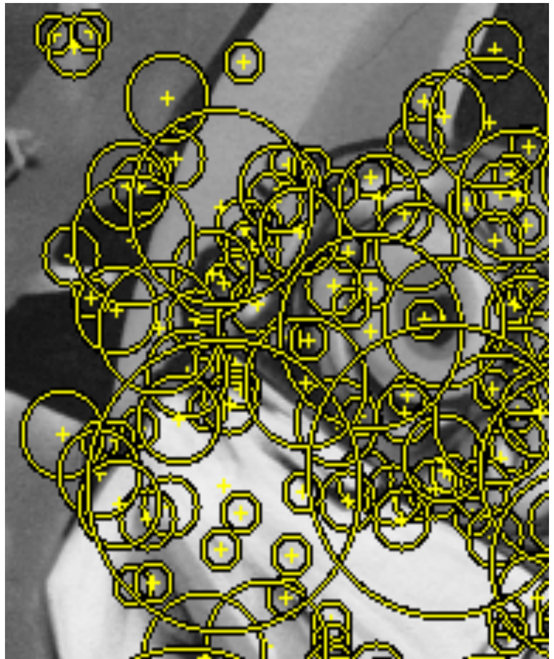
Comparison



Harris

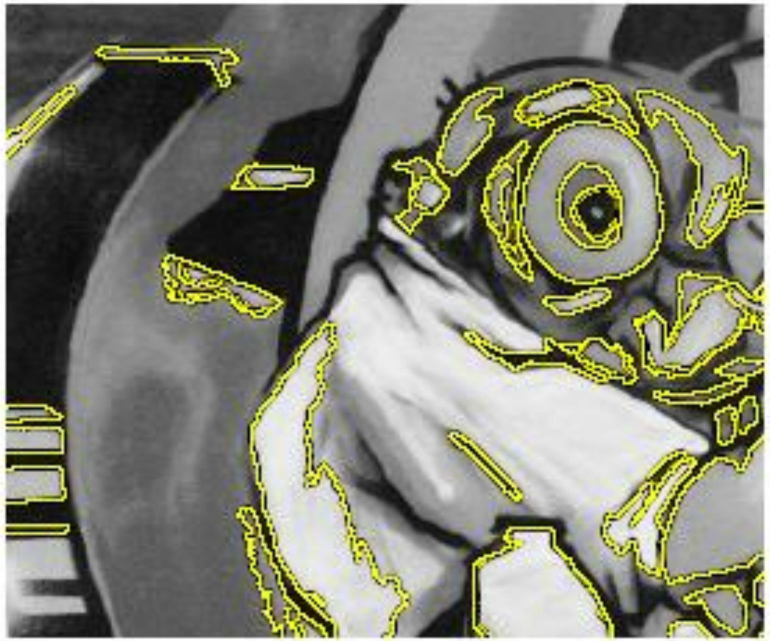
Hessian-based





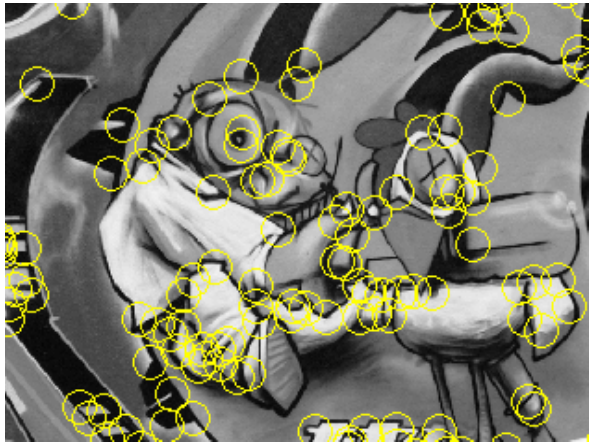
LoG

MSER



Review: Interest points

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



(a) Gray scale input image (b) Detected MSERs

Review: Choosing an interest point detector

- Why choose?
  - Collect more points with more detectors, for more possible matches
- 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
- 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

What we are learned today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector

Thanks