计算机视觉表征与识别
Chapter 7: Interest Points: Detector

王利民

媒体计算课题组

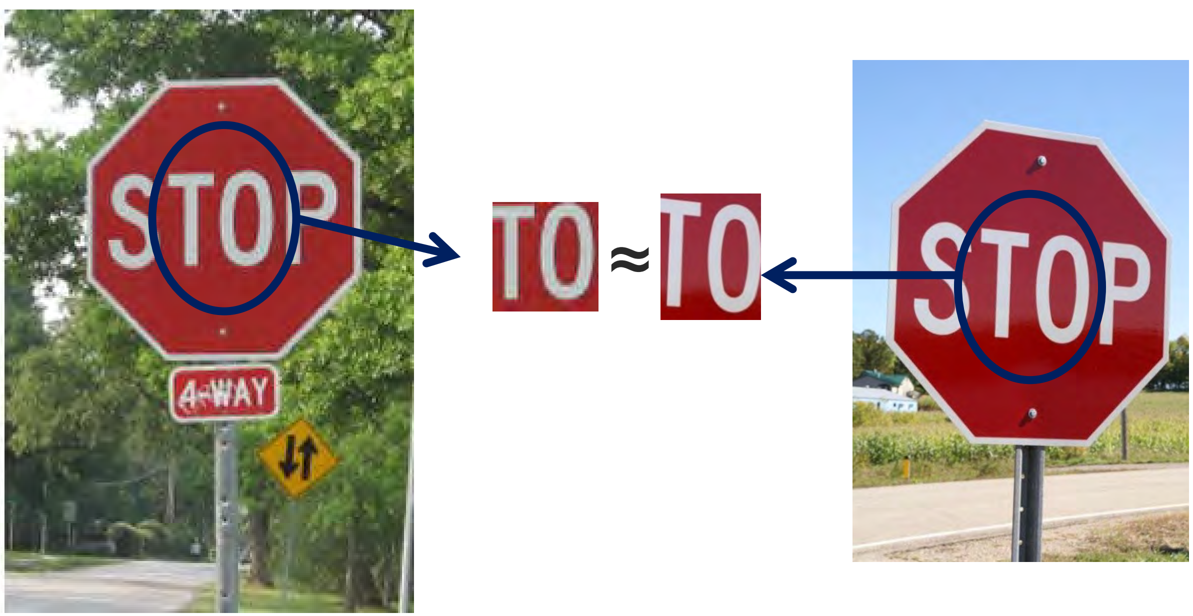
http://mcg.nju.edu.cn/

Today's Class

- **Introduction to correspondence and alignment**
- Overview of interest points
  - Matching pipeline
  - Repeatable & Distinctive
- Keypoint Localization
  - Harris detector
  - Hessian detector
- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gaussian (LoG) & Difference of Gaussian (DoG)
  - Combinations: Harris-Laplacian & Hessian-Laplacian

Correspondence and alignment

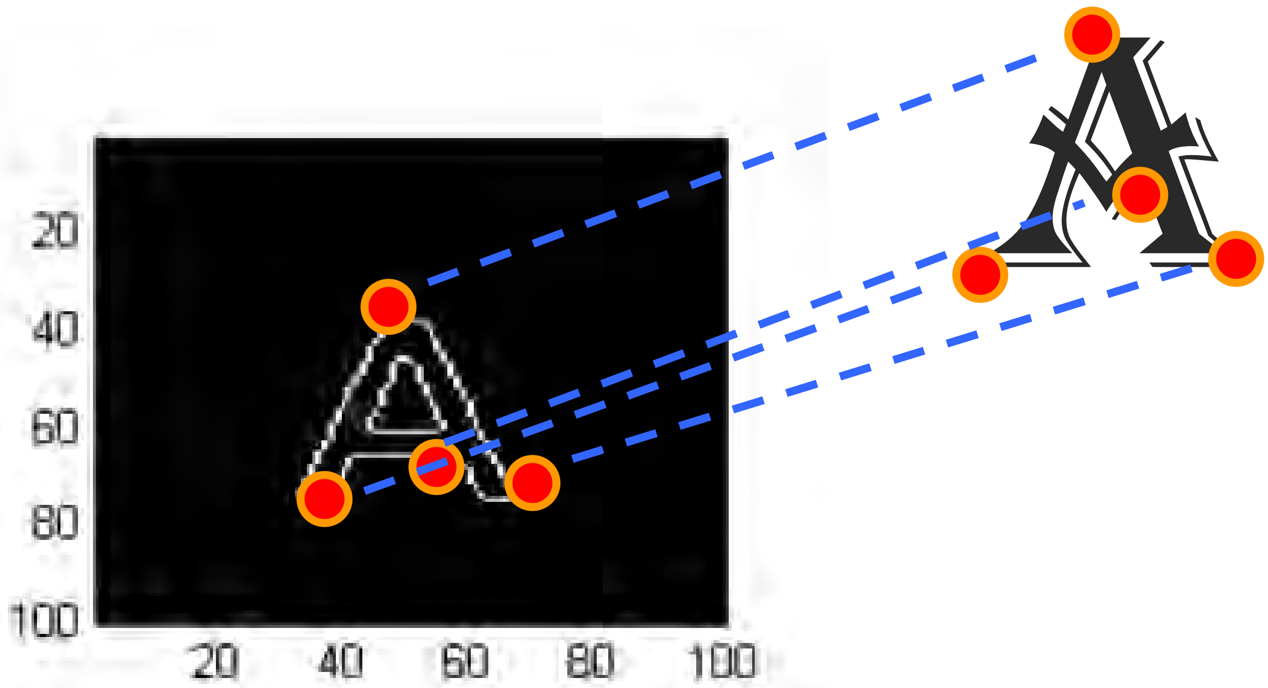
Correspondence: matching points, patches, edges, or regions across images



Correspondence and alignment

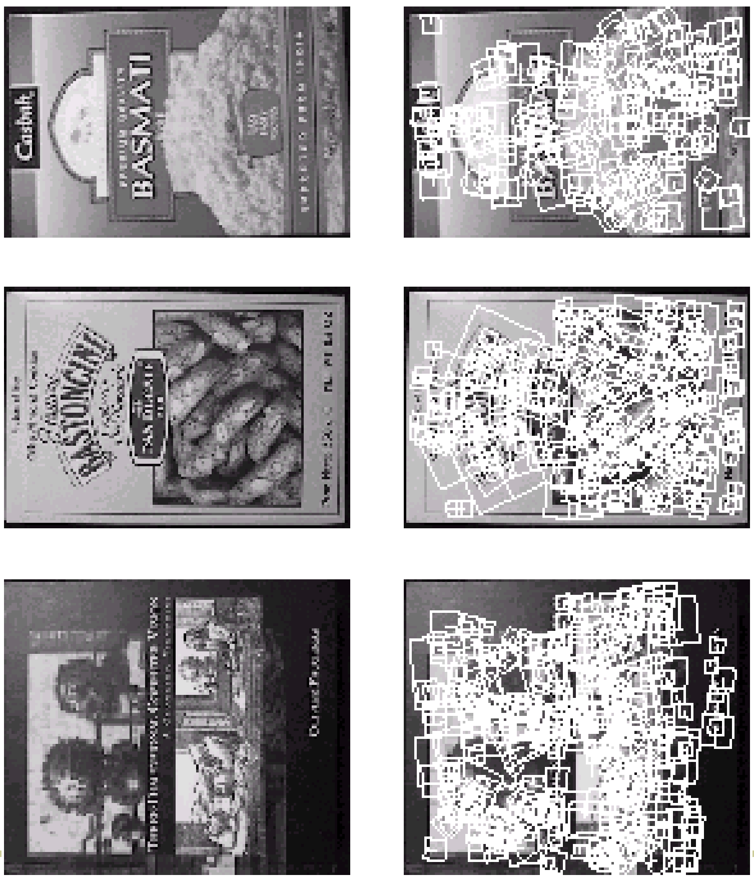
Alignment: solving the transformation that makes two things match better

Example: fitting an 2D shape template



Planar object instance recognition

Database of planar objects



Instance recognition

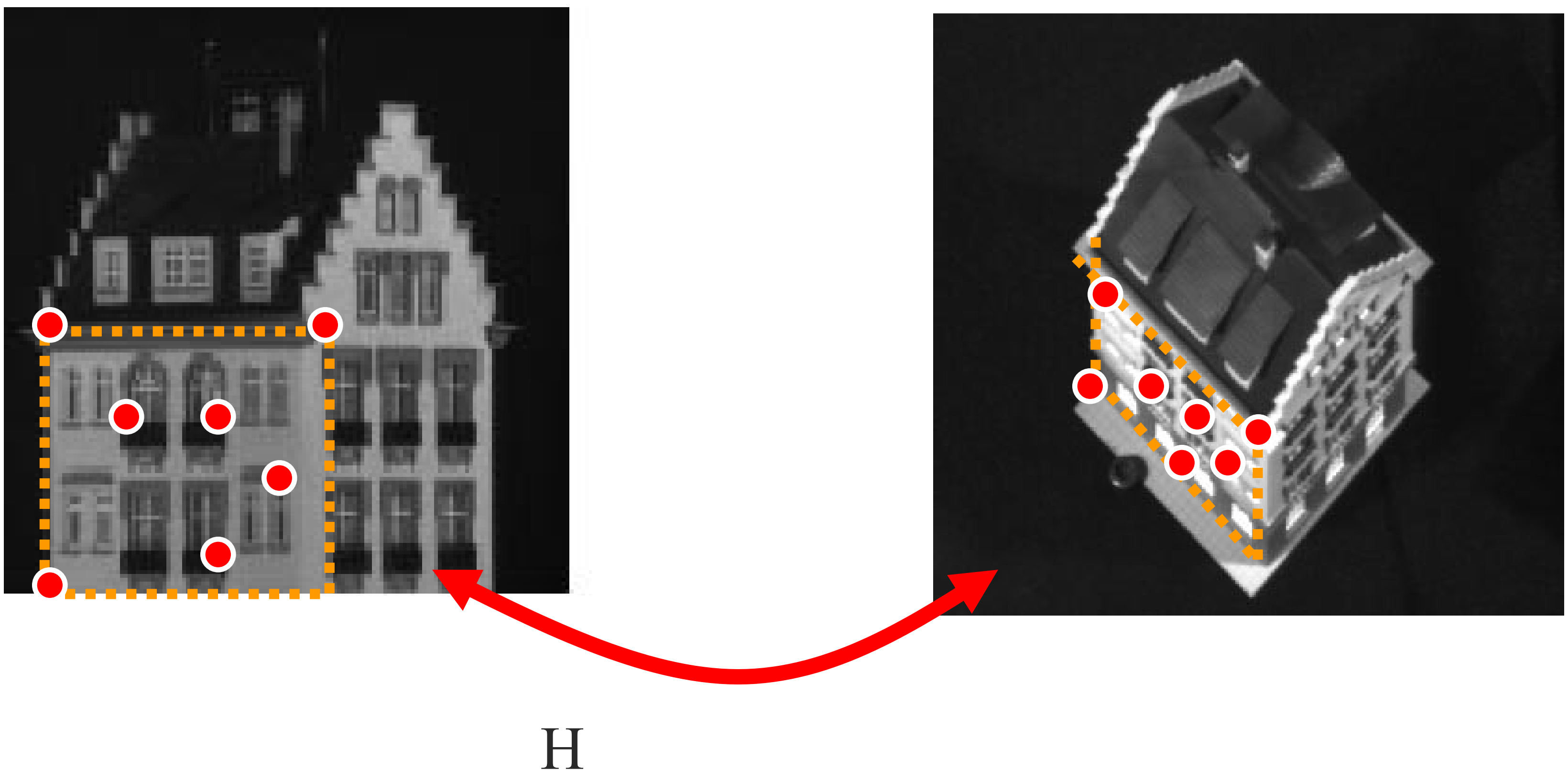
3D object recognition

Database of 3D objects

3D objects recognition

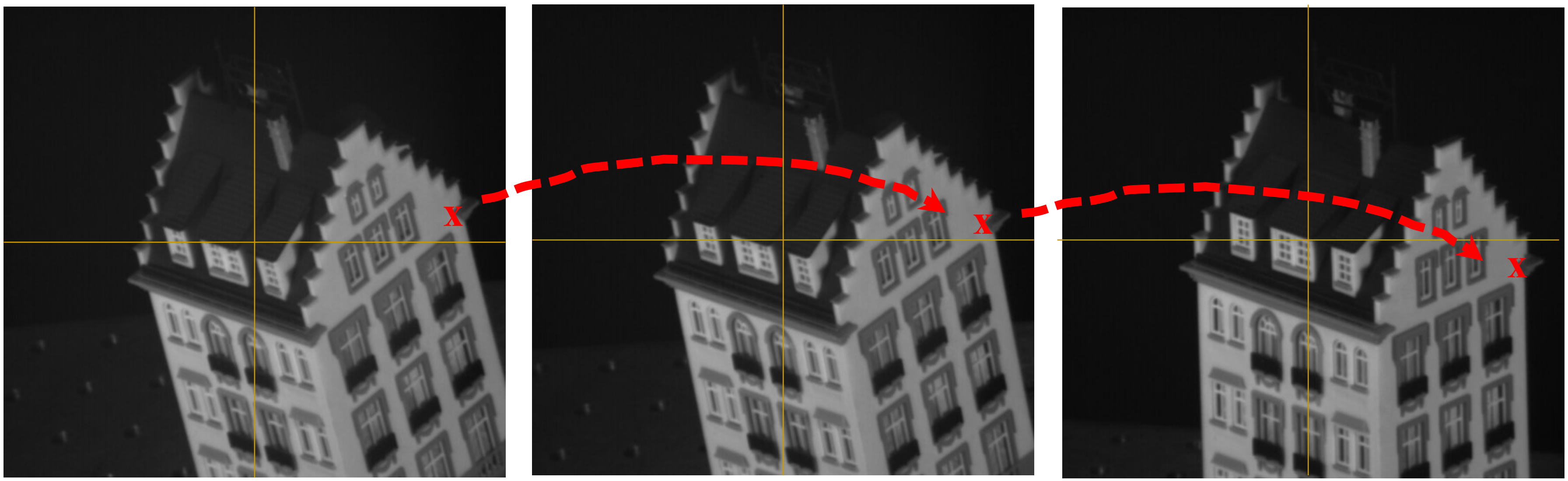
Example: Image matching

Example: Estimating an homographic transformation



Slide from Silvio Savarese

Example: tracking points



frame 0 frame 22 frame 49

Today's Class

- Introduction to correspondence and alignment
- **Overview of interest points**
  - Matching pipeline
  - Repeatable & Distinctive
- Keypoint Localization
  - Harris detector
  - Hessian detector
- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gaussian (LoG) & Difference of Gaussian (DoG)
  - Combinations: Harris-Laplacian & Hessian-Laplacian

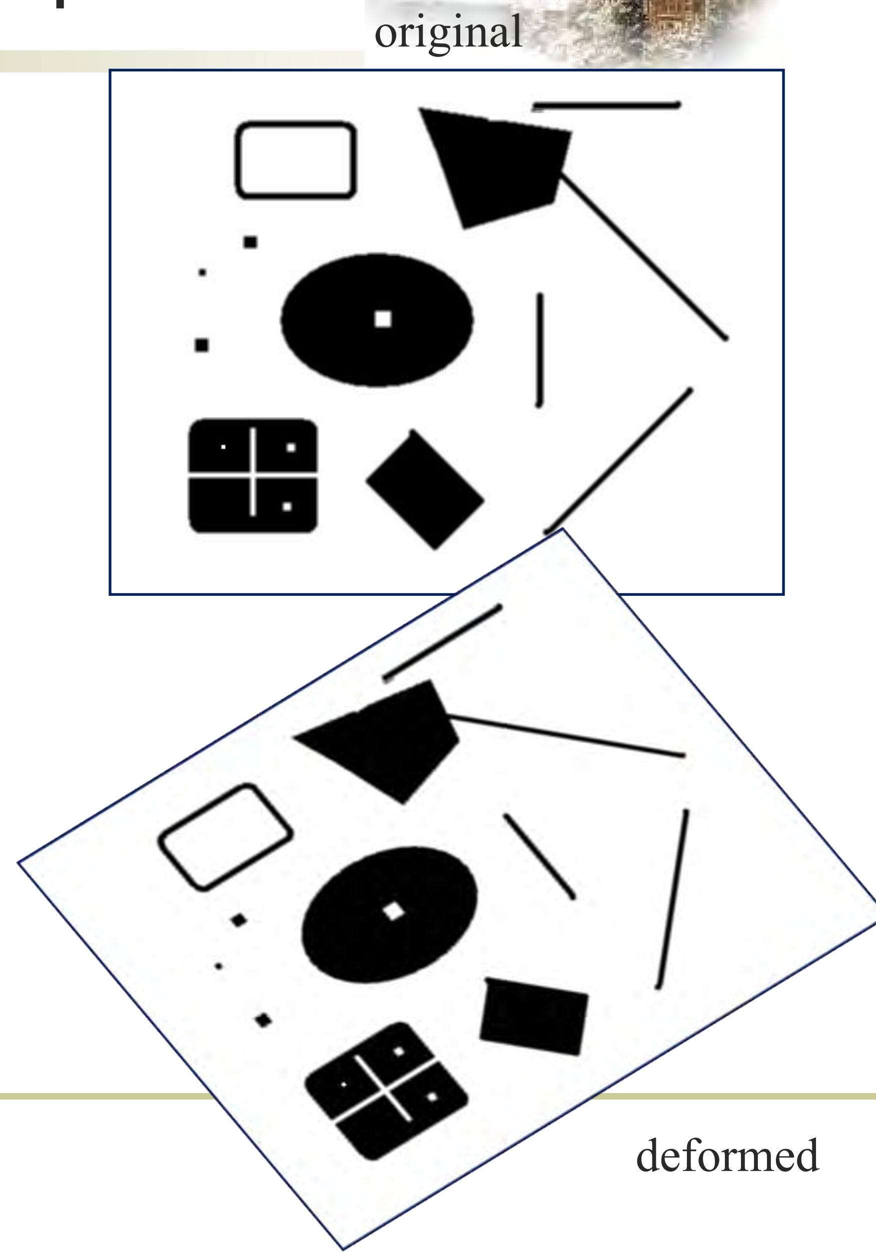
This class: interest points

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

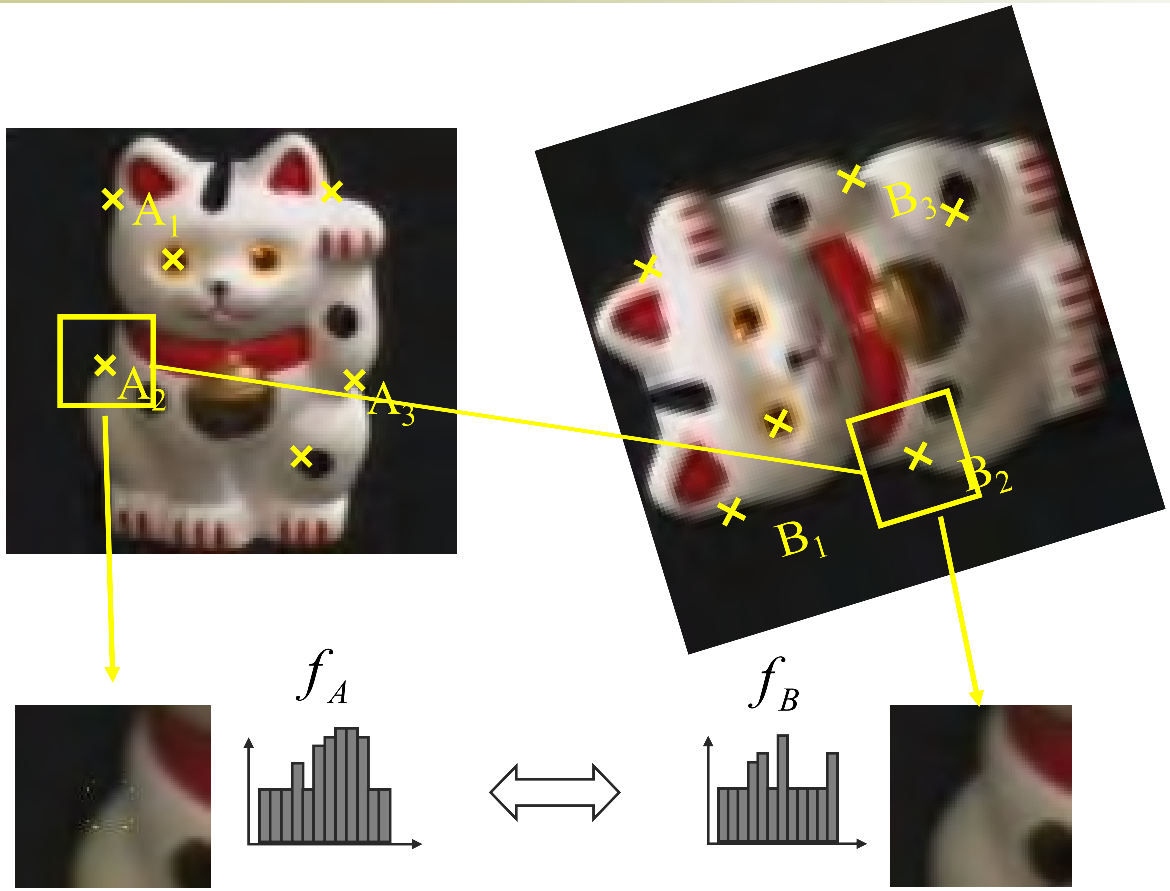
- Many applications
  - tracking: which points are good to track?
  - recognition: find patches likely to tell us something about object category
  - 3D reconstruction: find correspondences across different views

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



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

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

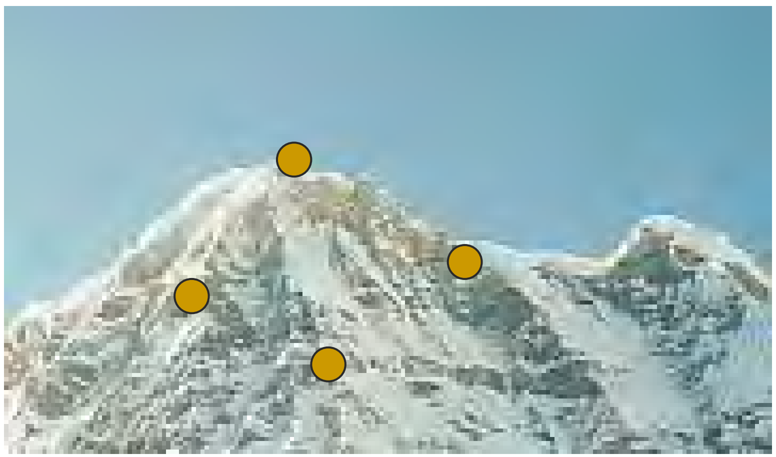
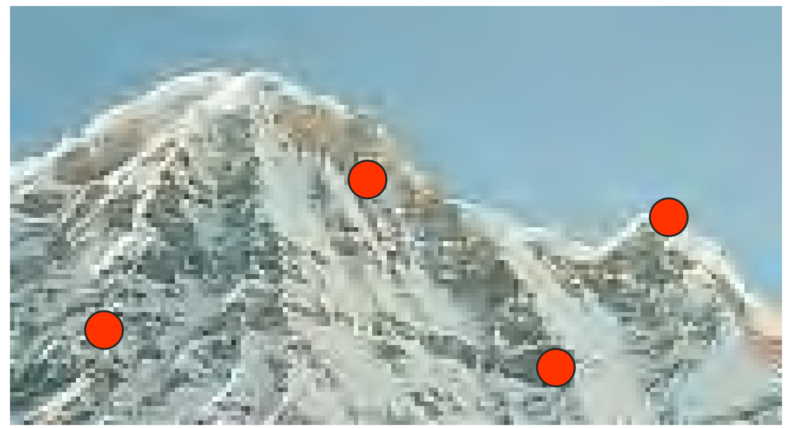
K. Grauman, B. Leibe

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Goal: interest operator repeatability

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

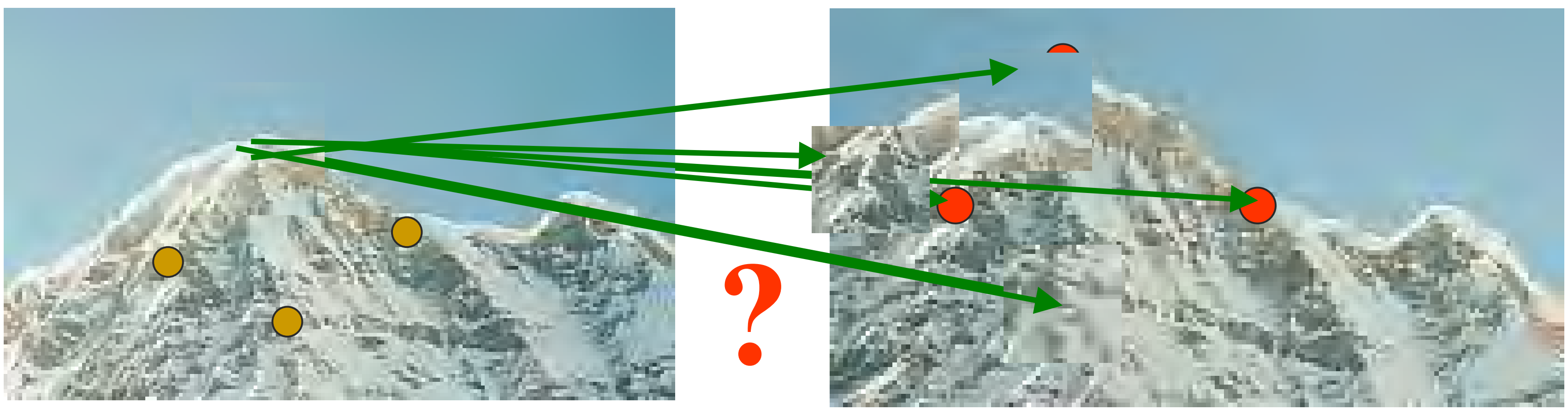



No chance to find true matches!

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

Goal: descriptor distinctiveness

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

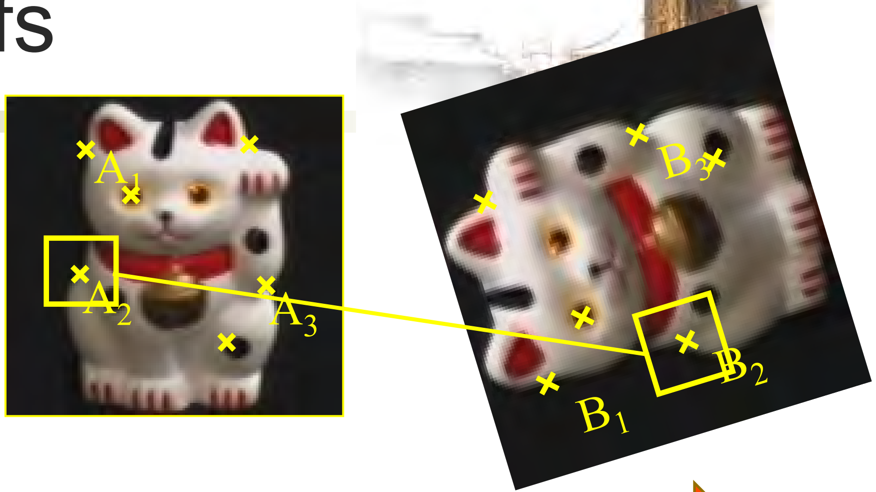


- Must provide some invariance to geometric and photometric differences between the two views.

Local features: desired properties

- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Distinctiveness
  - Each feature has a distinctive description
- 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

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?

Corner detection

Choosing interest points

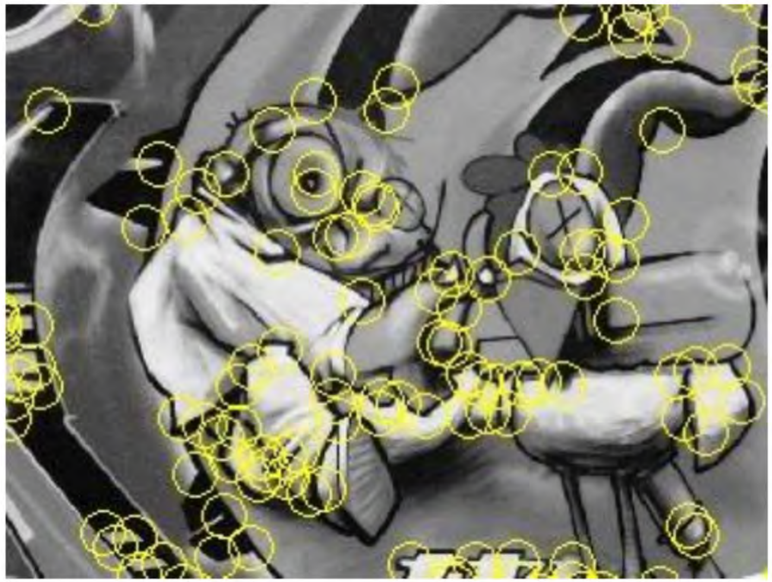
Where would you tell your friend to meet you?

Blob (valley/peak) detection

Today's Class

- Introduction to correspondence and alignment
- Overview of interest points
  - Matching pipeline
  - Repeatable & Distinctive
- **Keypoint Localization**
  - Harris detector
  - Hessian detector
- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gaussian (LoG) & Difference of Gaussian (DoG)
  - Combinations: Harris-Laplacian & Hessian-Laplacian

Keypoint Localization

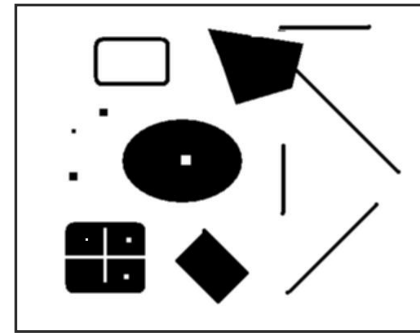


- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

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

Harris Detector [Harris88]

- Second moment matrix



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

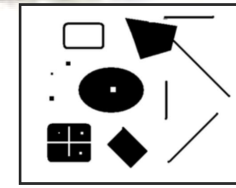
Intuition: **Search for local neighborhoods where the image content has two main directions (eigenvectors).**

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

K. Grauman, B. Leibe

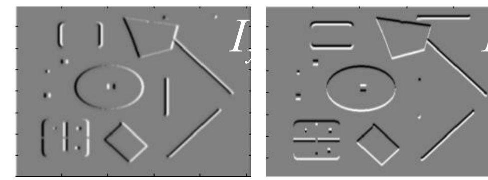
Harris Detector [Harris88]

- Second moment matrix



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

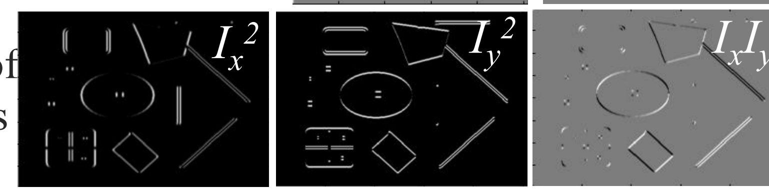
1. Image derivatives (optionally, blur first)



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

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

2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

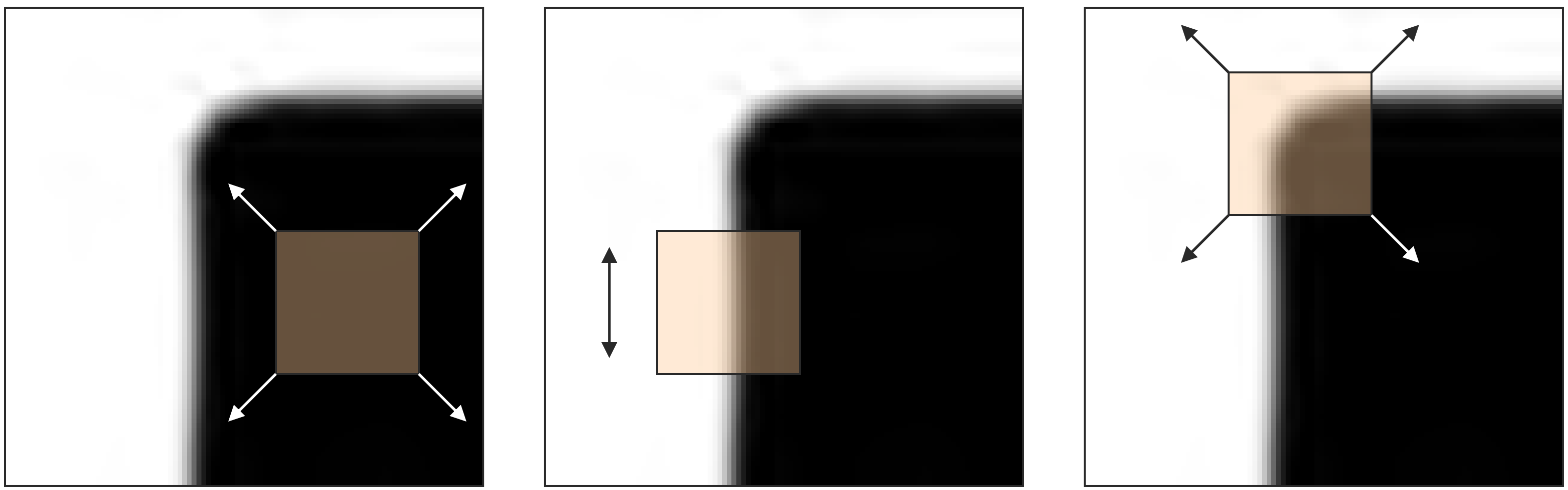
$$har=\det[\mu(\sigma_I,\sigma_D)]-\alpha[\operatorname{trace}(\mu(\sigma_I,\sigma_D))^2]=$$

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

5. Non-maxima suppression

Corners as distinctive interest points

- 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

Error function

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

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

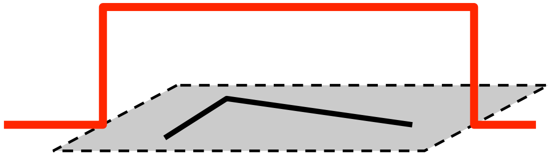
Error function

Window function

Shifted intensity

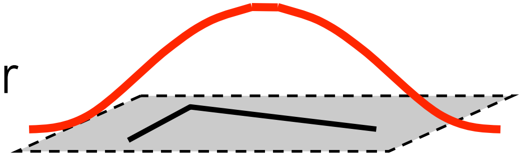
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Error function approximation

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

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

First-order Taylor expansion of I(x,y) about (0,0)
(bilinear approximation for small shifts)

Bilinear approximation

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

Change in appearance for a shift [u,v]

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

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

'second moment' matrix
'structure tensor'

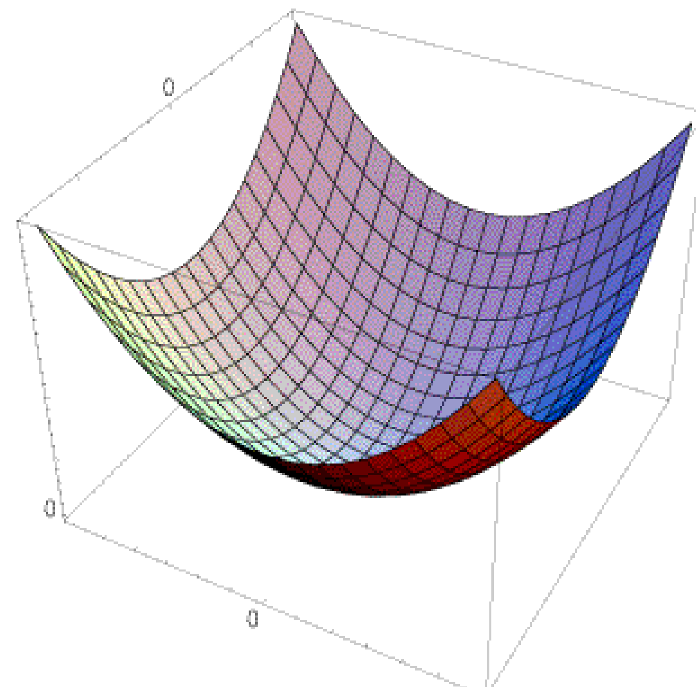
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Visualization of a quadratic

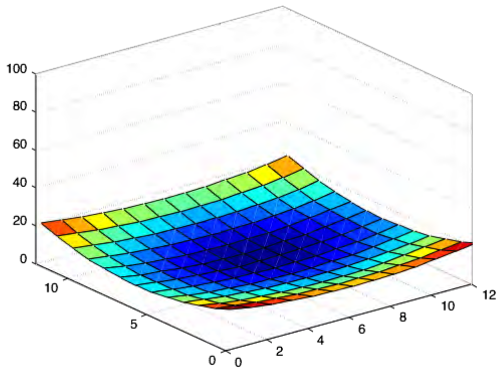
The surface *E*(*u*,*v*) is locally approximated by a quadratic form

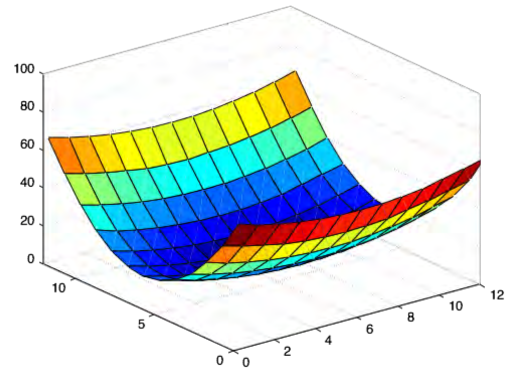
$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

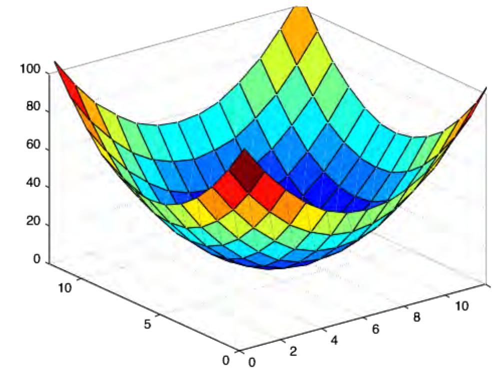
$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Which error surface indicates a good image feature?

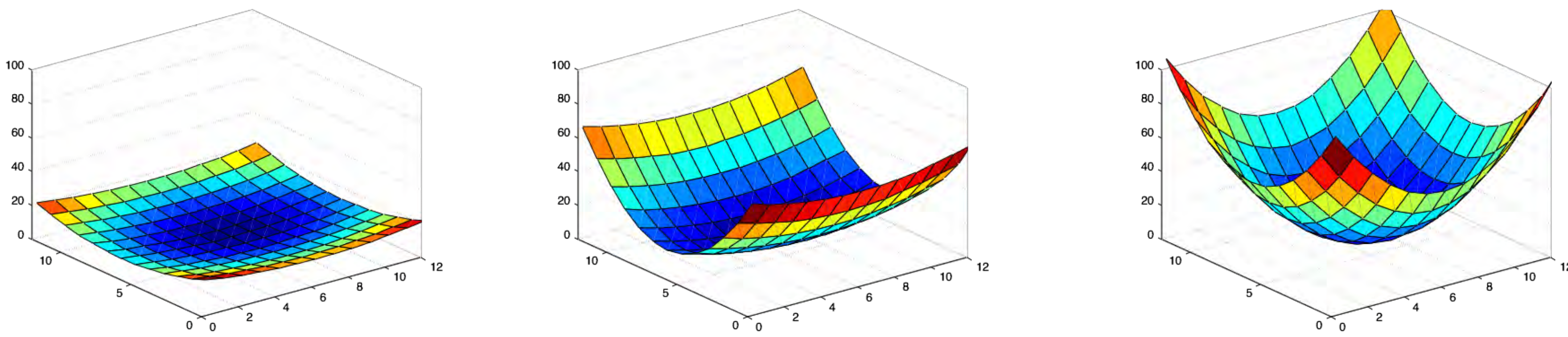






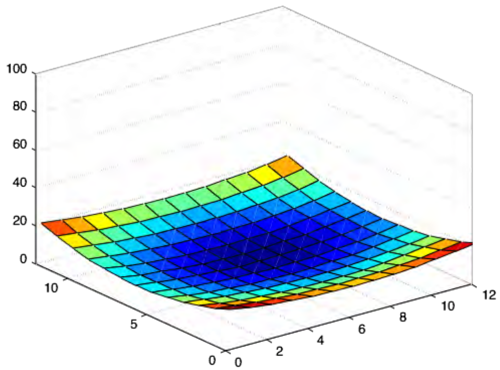
What kind of image patch do these surfaces represent?

Which error surface indicates a good image feature?

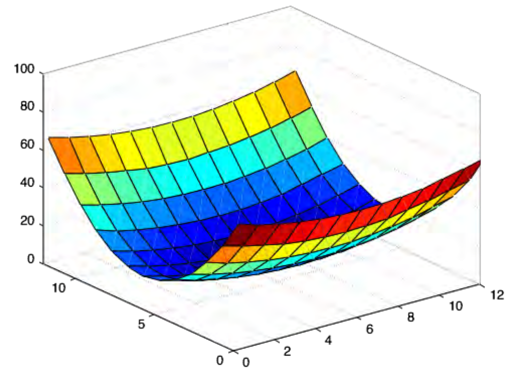


flat

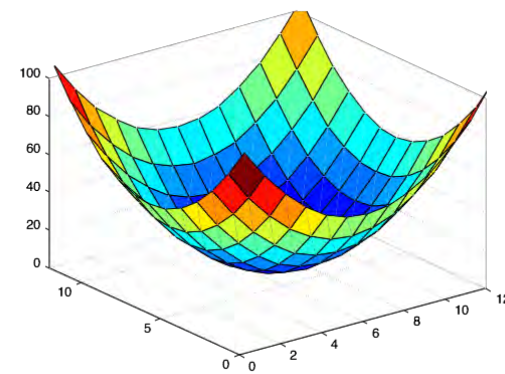
Which error surface indicates a good image feature?



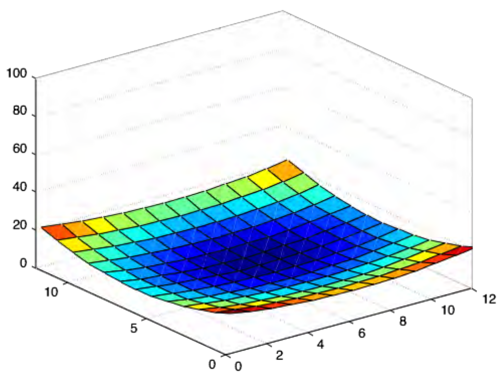
flat



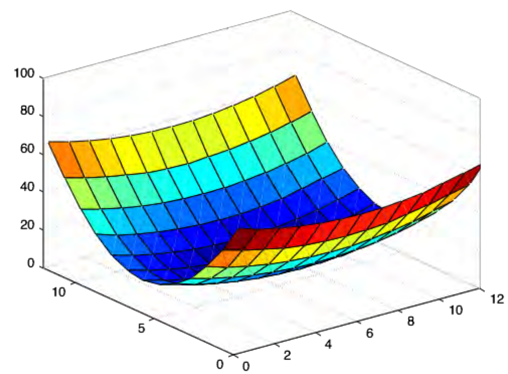
edge
‘line’



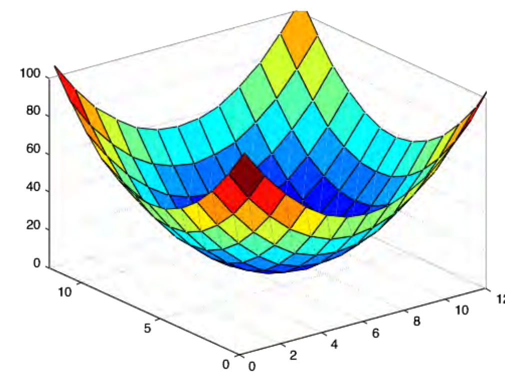
Which error surface indicates a good image feature?



flat



edge
‘line’



corner
‘dot’

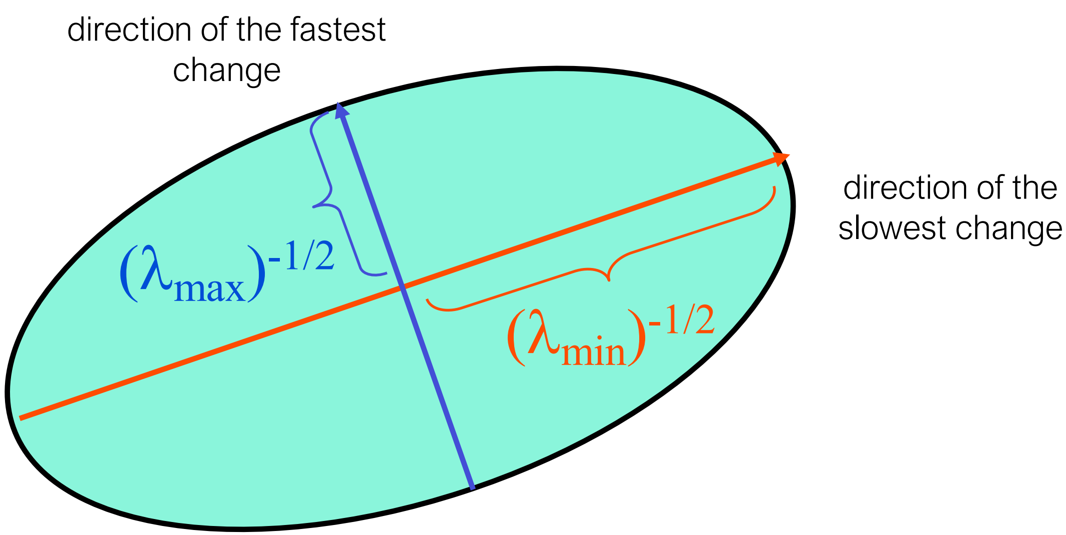
Visualization as an ellipse

Since M is symmetric, we have $$M = R^{-1}\begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}R$$

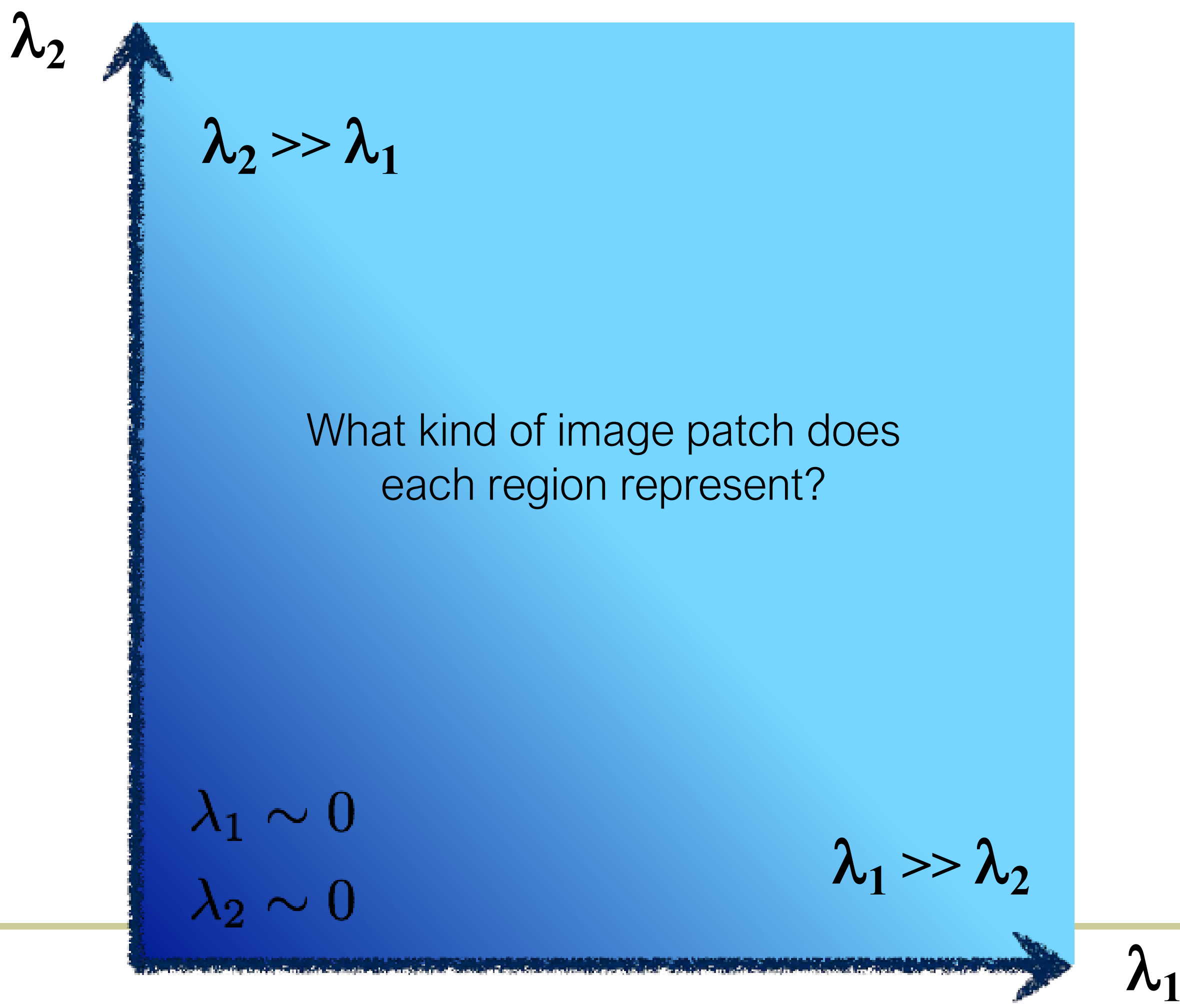
We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*

Ellipse equation:

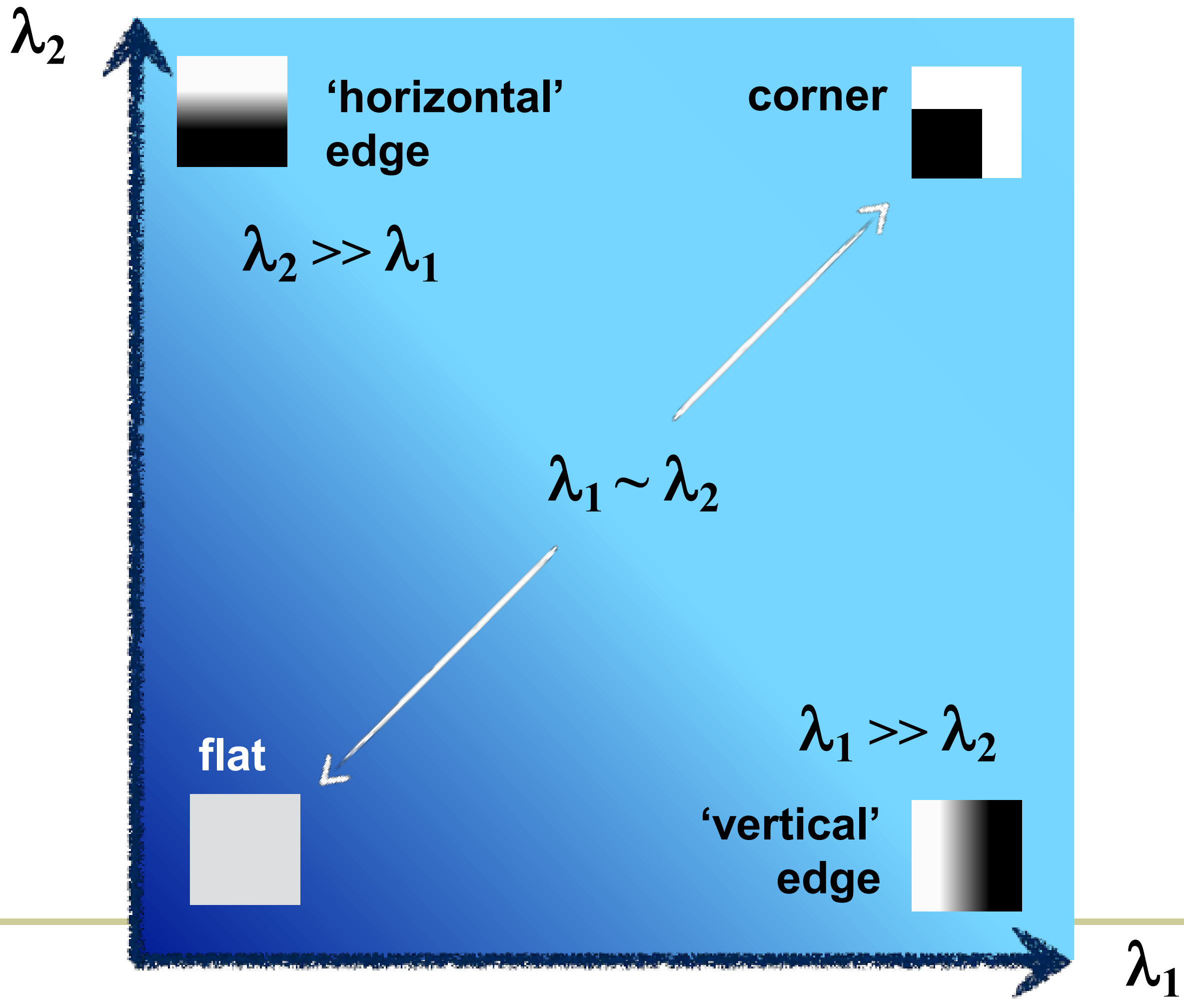
$$[u \;\; v] \;\; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



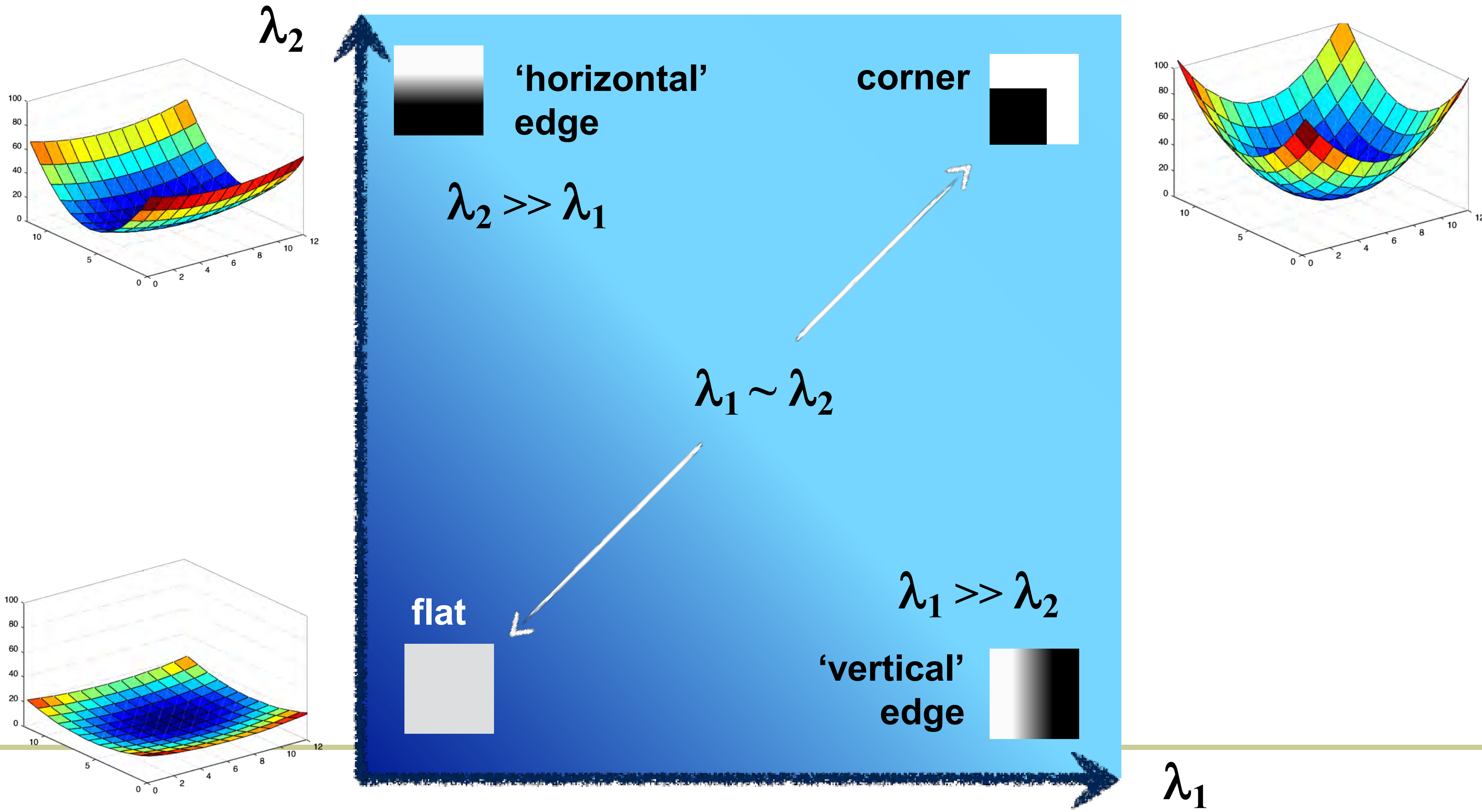
interpreting eigenvalues



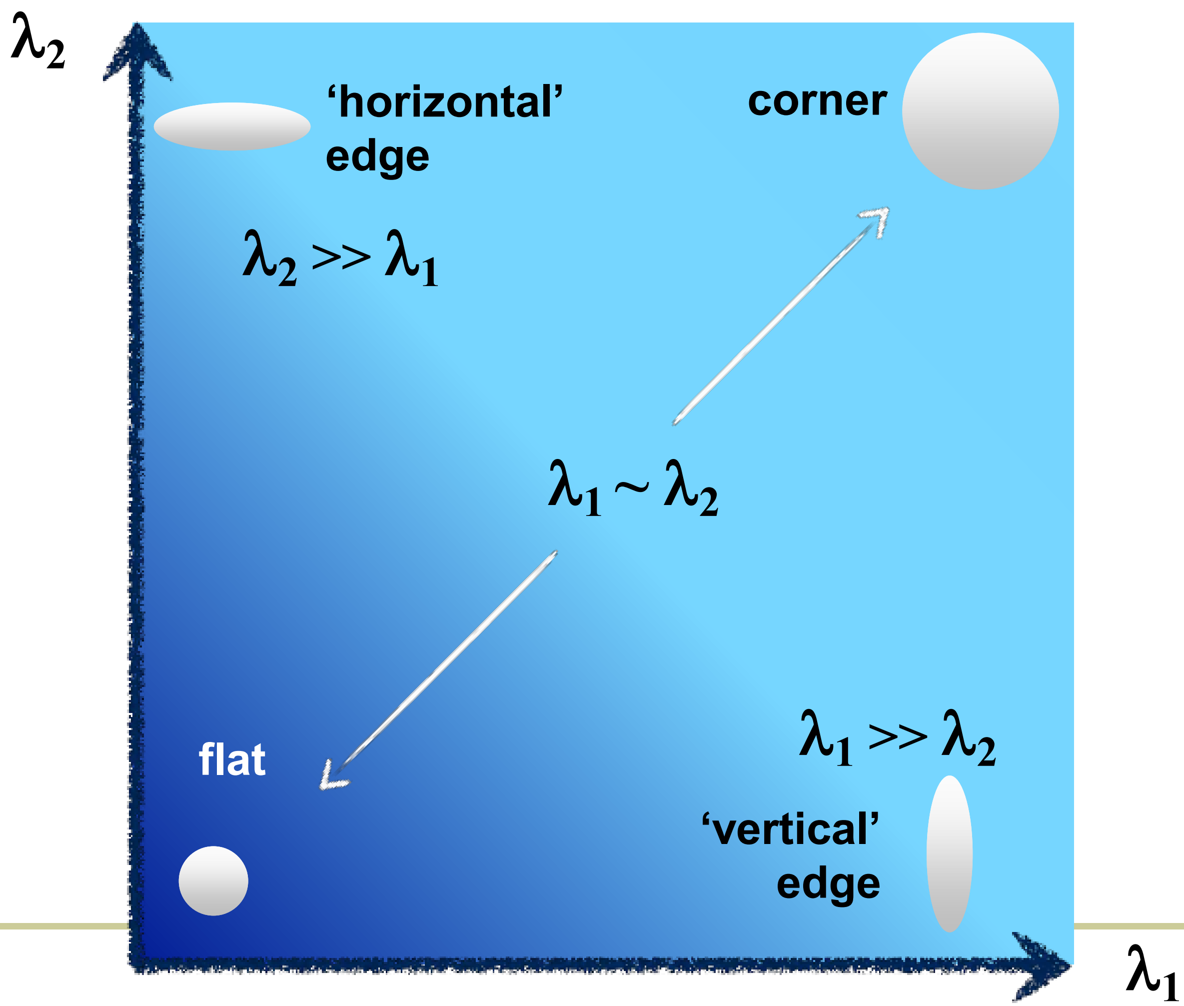
interpreting eigenvalues



interpreting eigenvalues

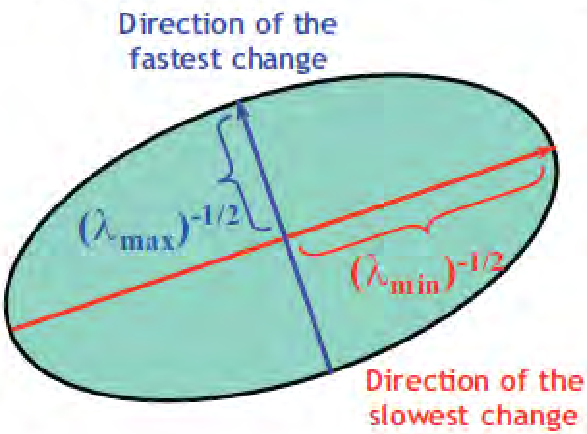


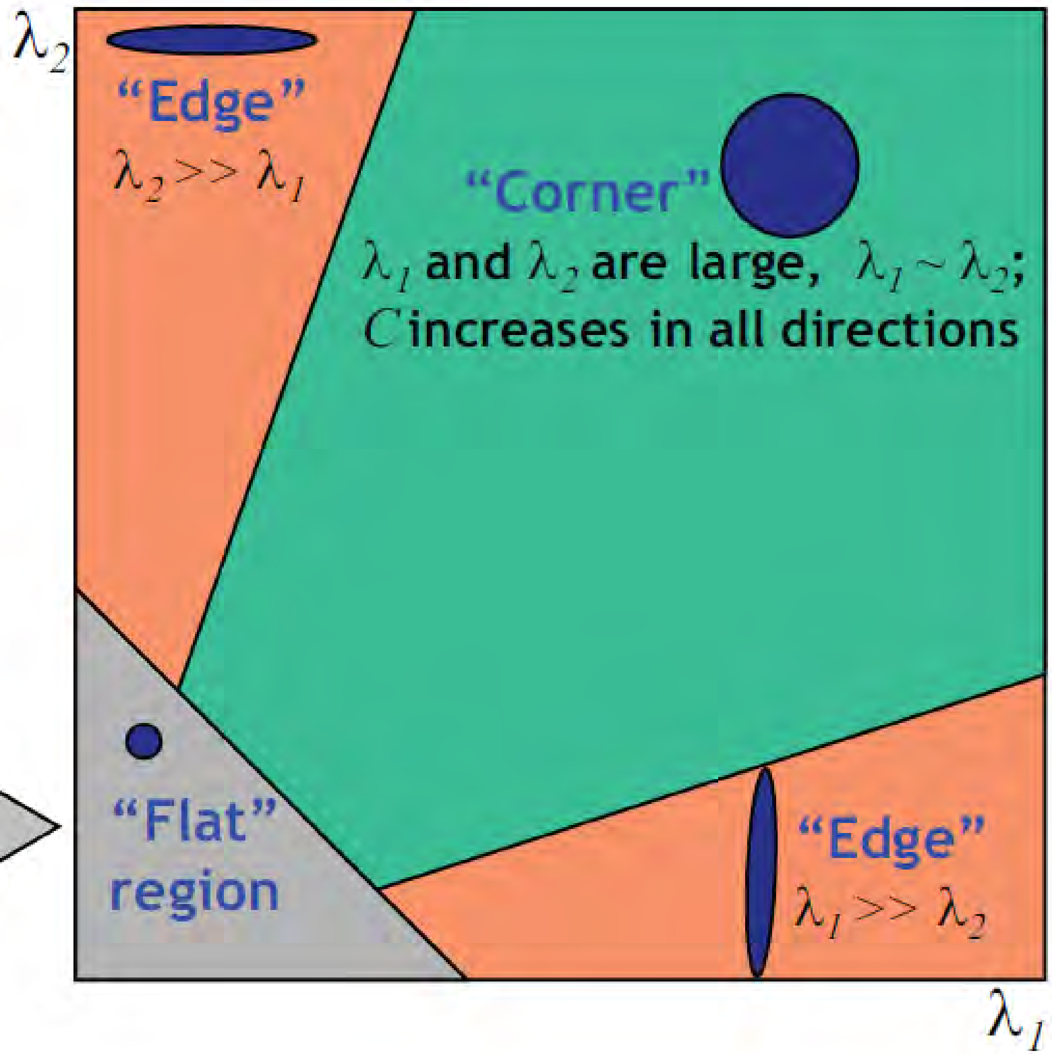
interpreting eigenvalues



Explanation of Harris Criterion

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$





From Grauman and Leibe

Harris Detector: Criteria

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

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

Harris Detector: Criteria

Harris & Stephens (1988)

$$R = \det(M) - \kappa \operatorname{trace}^2(M)$$

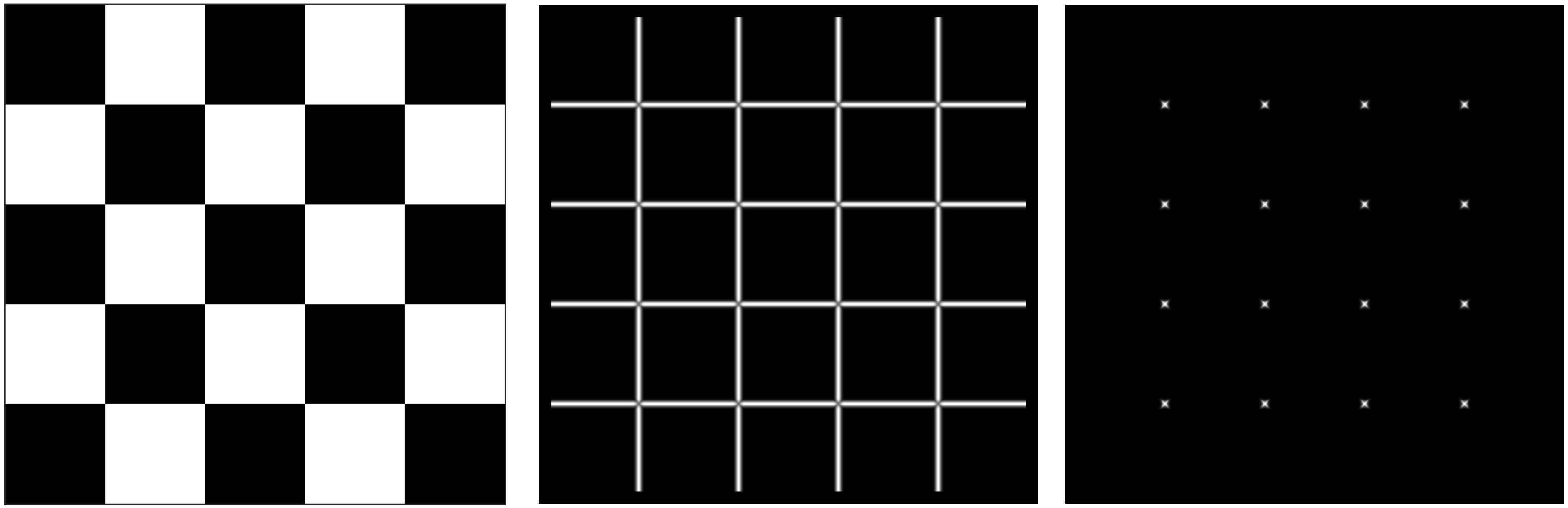
Kanade & Tomasi (1994)

$$R = \min(\lambda_1, \lambda_2)$$

Nobel (1998)

$$R = \frac{\det(M)}{\operatorname{trace}(M) + \epsilon}$$

Harris Detector: Criteria

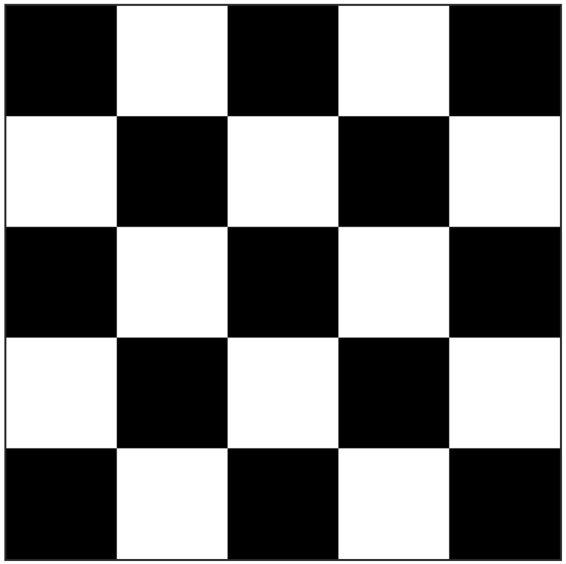


I $\quad$ $\lambda_{\max}$ $\quad$ $\lambda_{\min}$

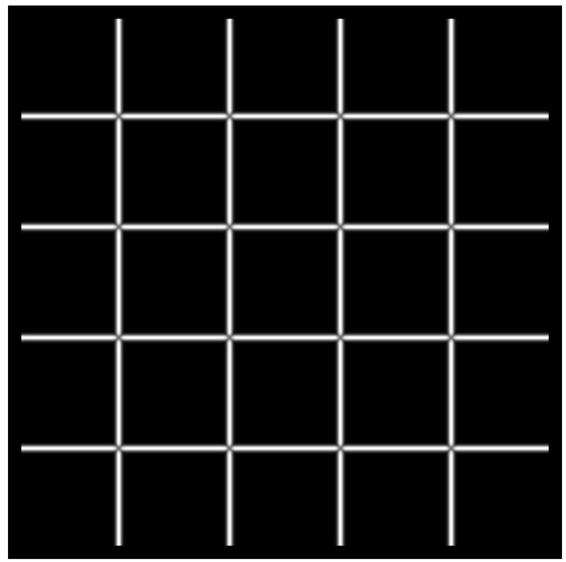
Yet another option: $$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

How do you write this equivalently using determinant and trace?

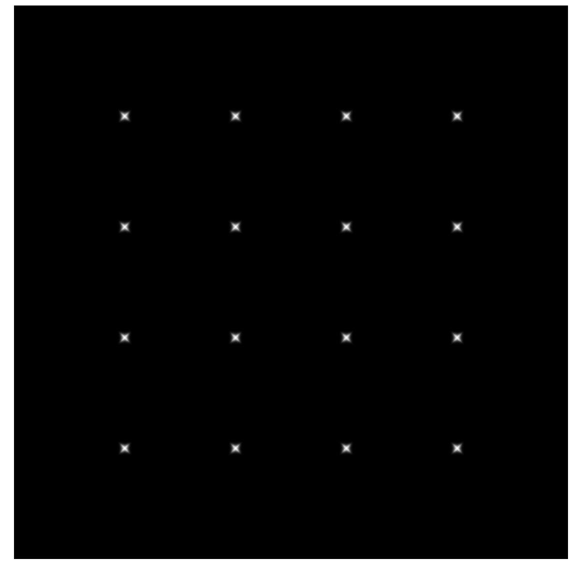
Harris Detector: Criteria



I



$\lambda_{\max}$

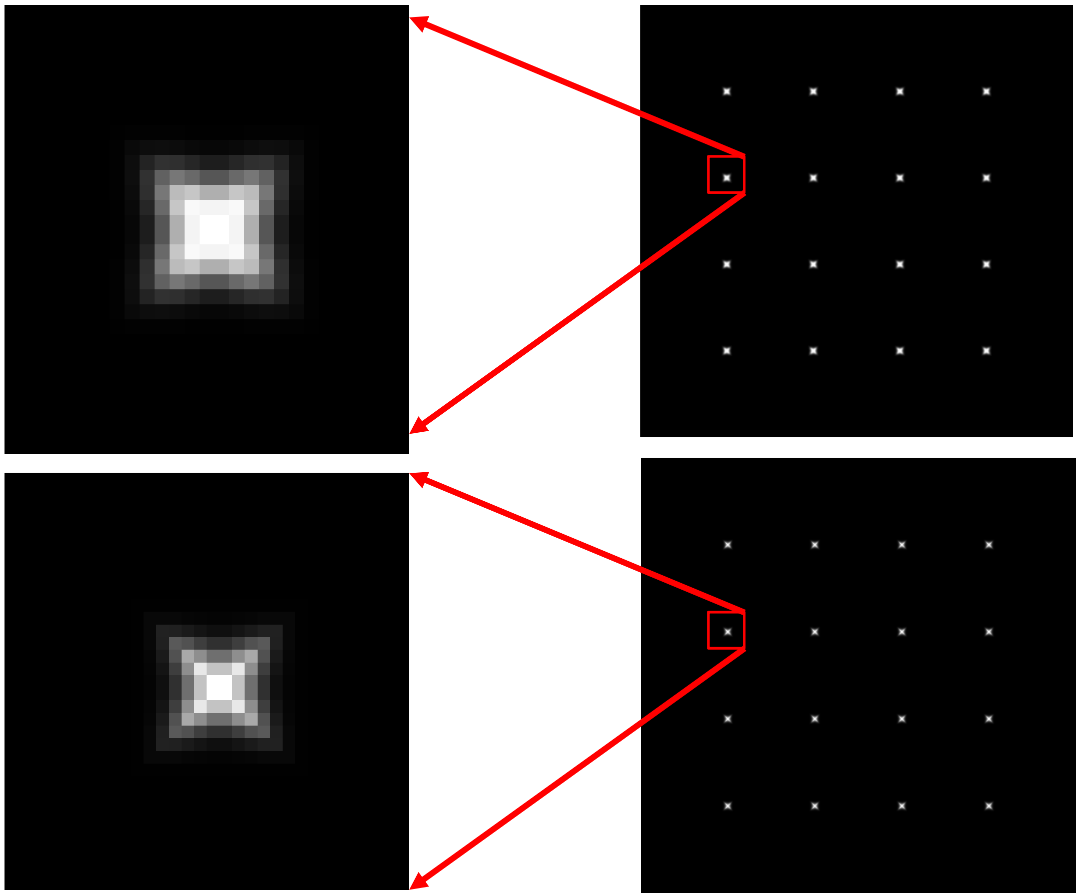


$\lambda_{\min}$

Yet another option:

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{determinant(H)}{trace(H)}$$

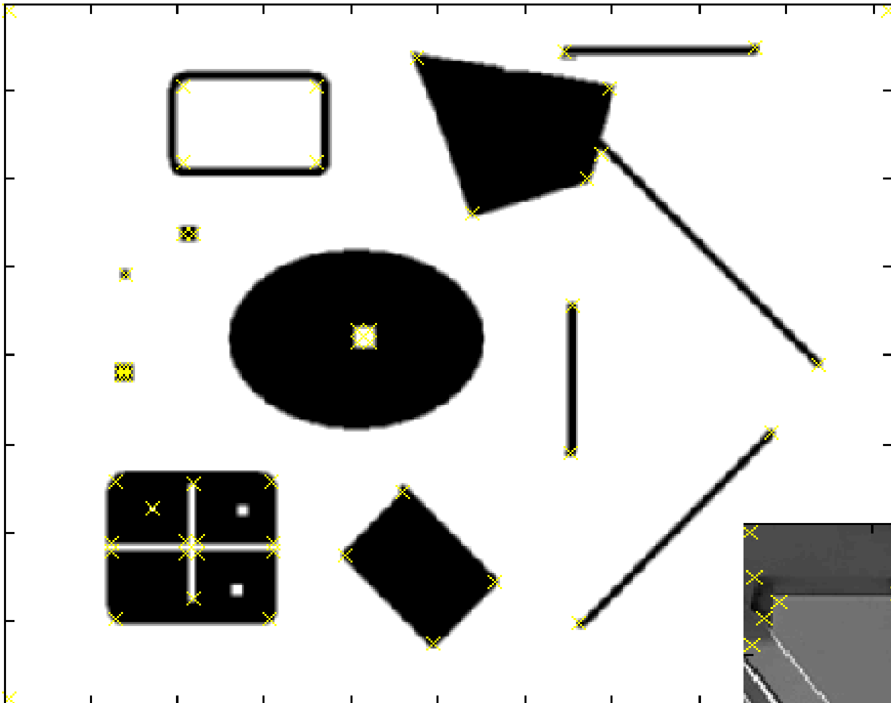
Different criteria

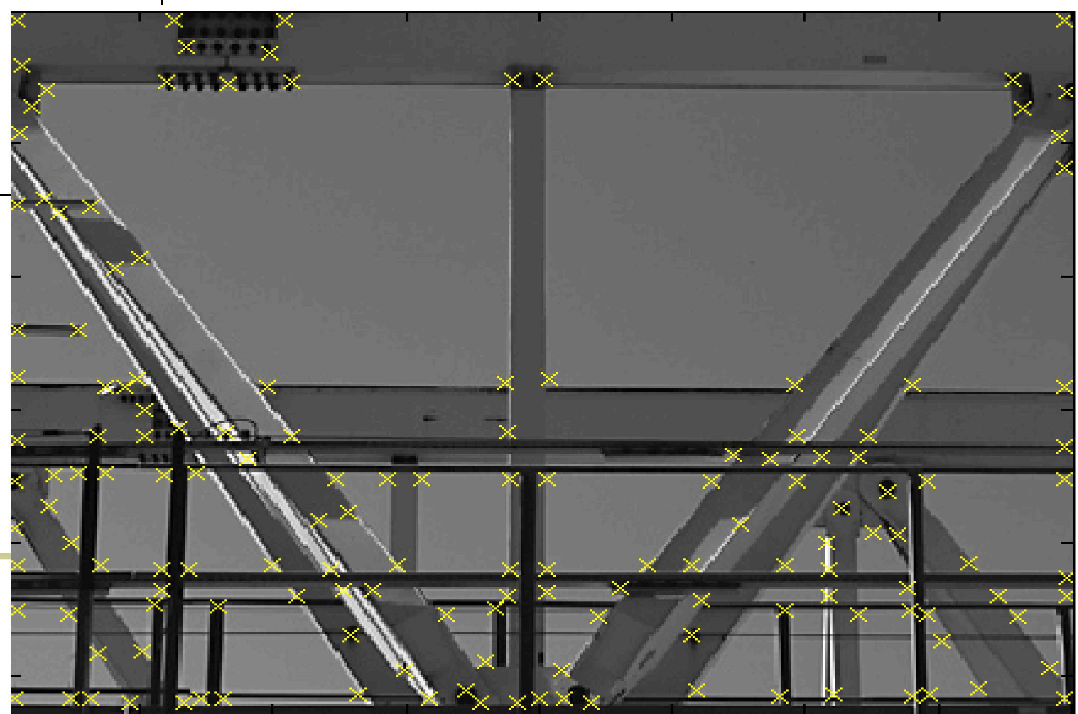


Harris criterion

$\lambda_{\min}$

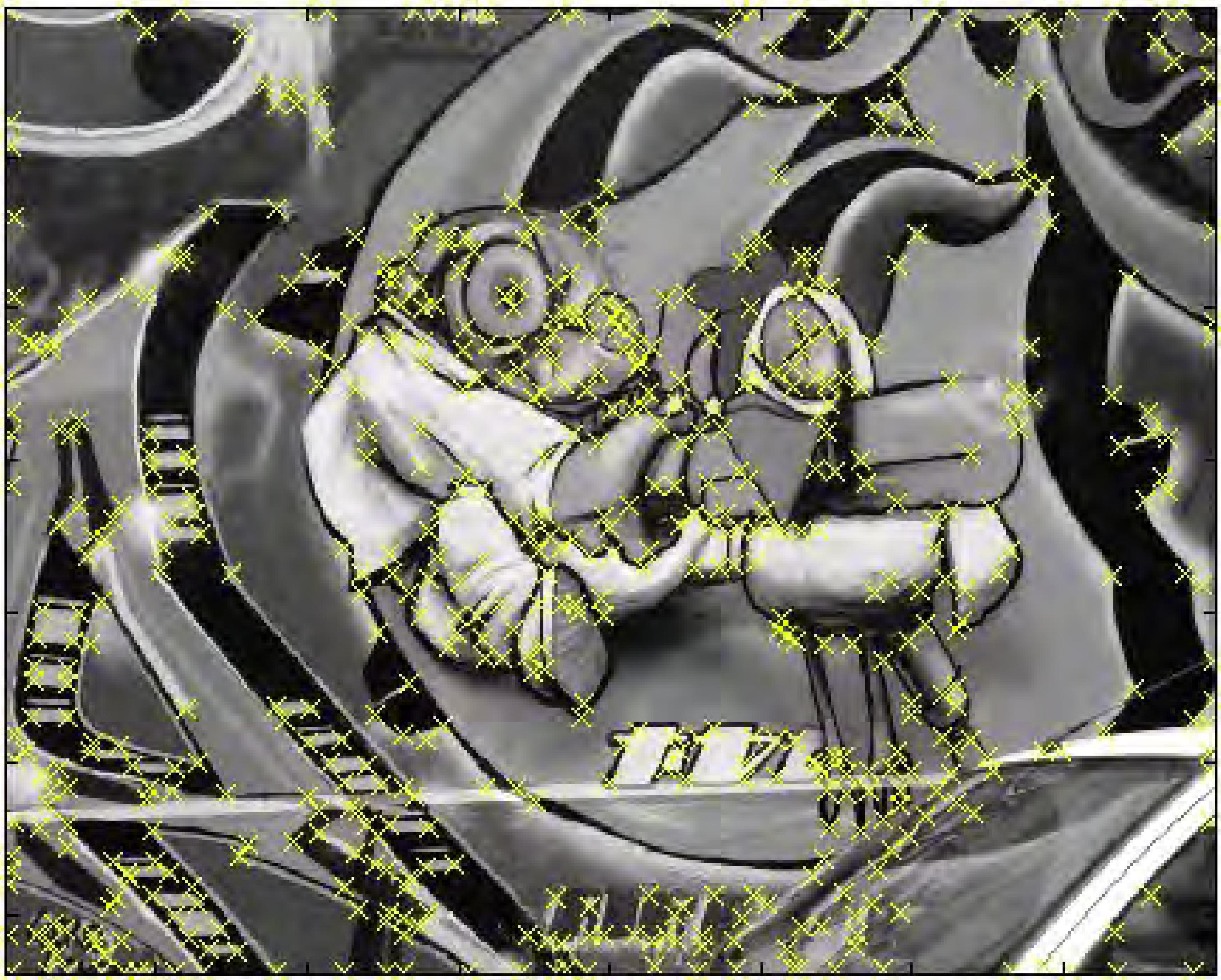
Harris Detector – Responses [Harris88]





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

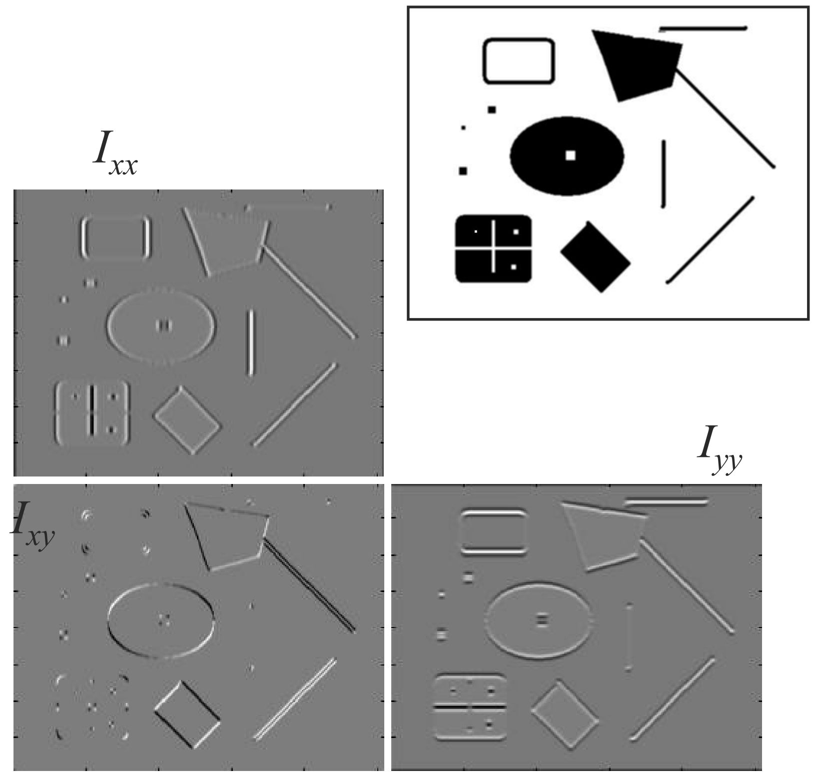
Harris Detector - Responses [Harris88]



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

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

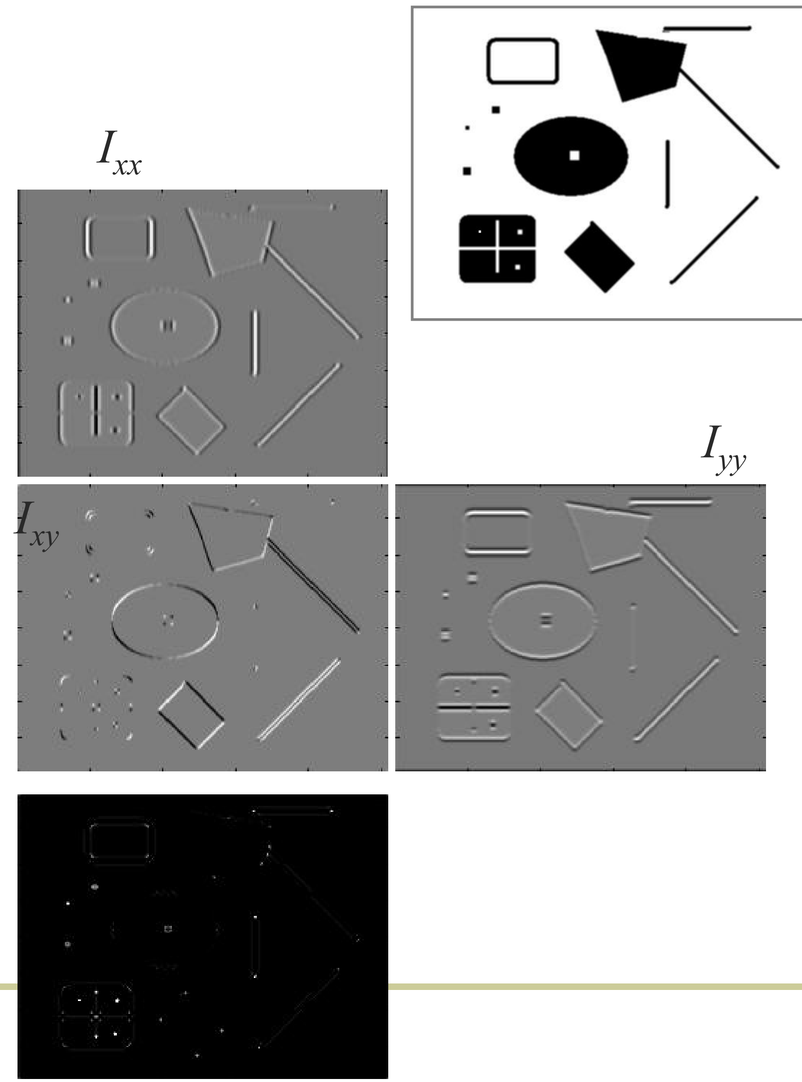
$$\det M = \lambda_1\lambda_2$$
$$\operatorname{trace} M = \lambda_1+\lambda_2$$

Find maxima of determinant

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

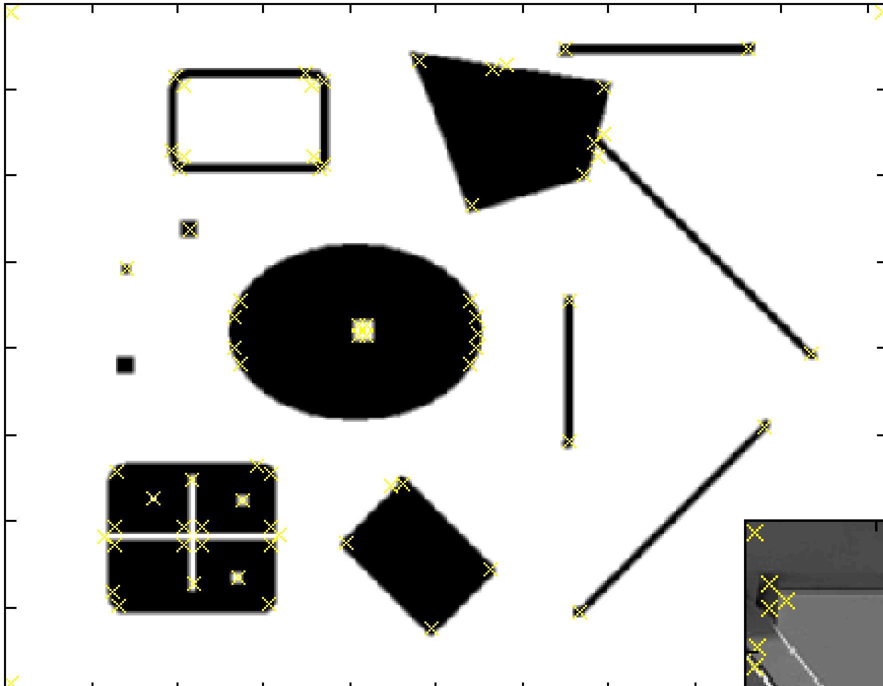
In Matlab:

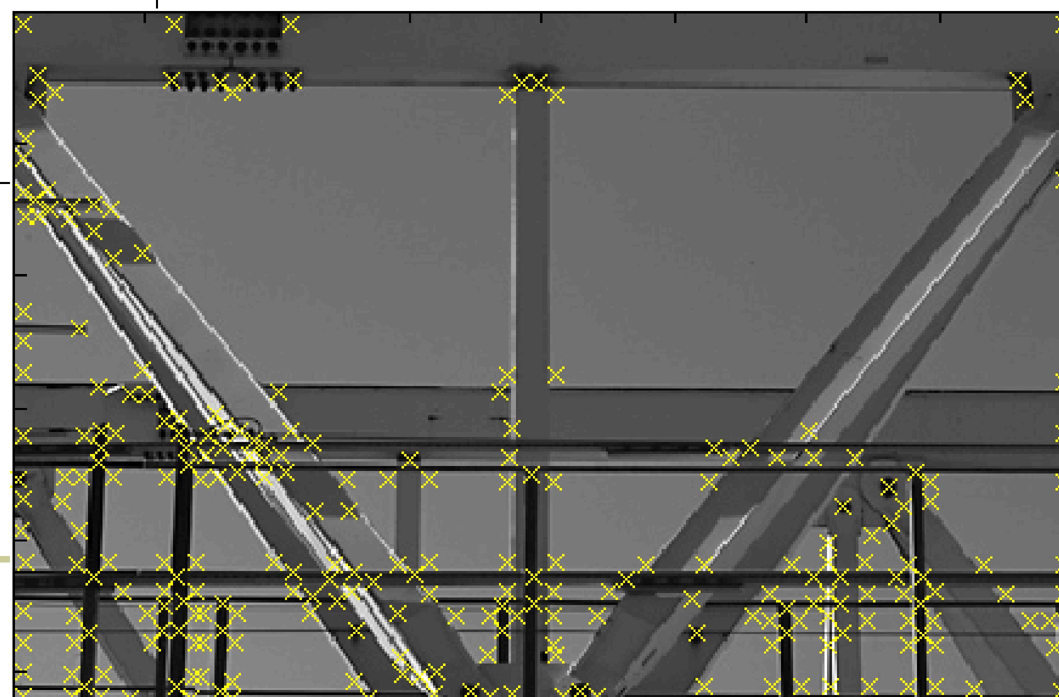
$$I_{xx}.*I_{yy}-(I_{xy})\verb|^|2$$



K. Grauman, B. Leibe

Hessian Detector – Responses [Beaudet78]

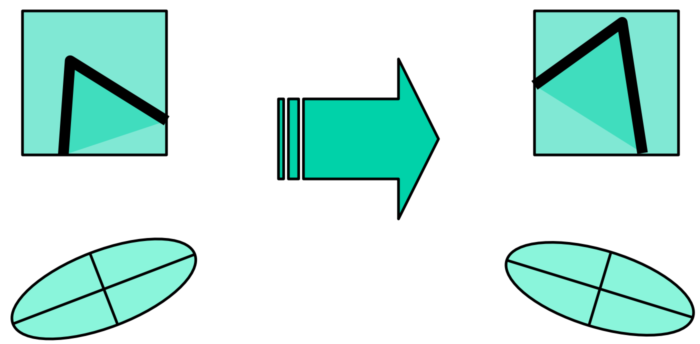




Effect: **Responses mainly on corners and strongly textured areas.**

Hessian Detector – Responses [Beaudet78]

Harris corner response is invariant to rotation



Ellipse rotates but its shape (**eigenvalues**) remains the same

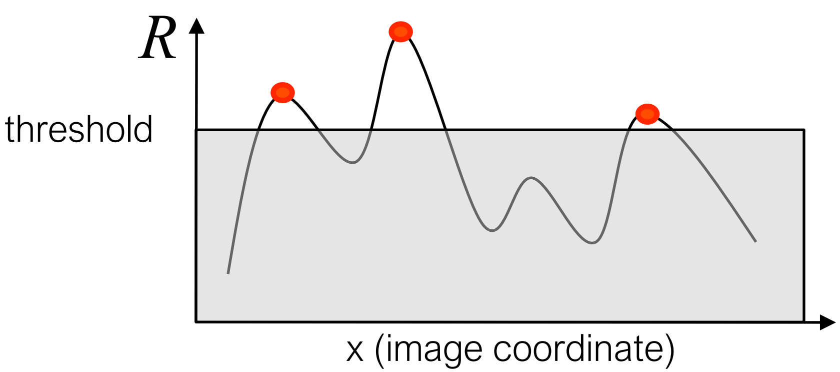
Corner response R is invariant to image rotation

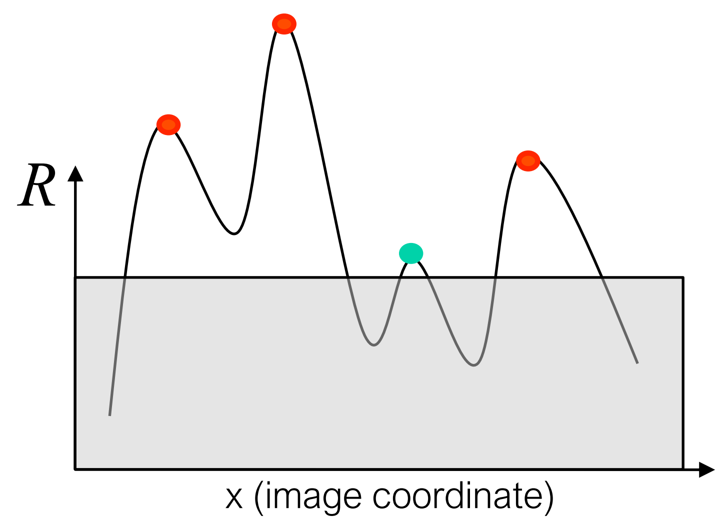
Harris corner response is invariant to intensity changes

Partial invariance to *affine intensity* change

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

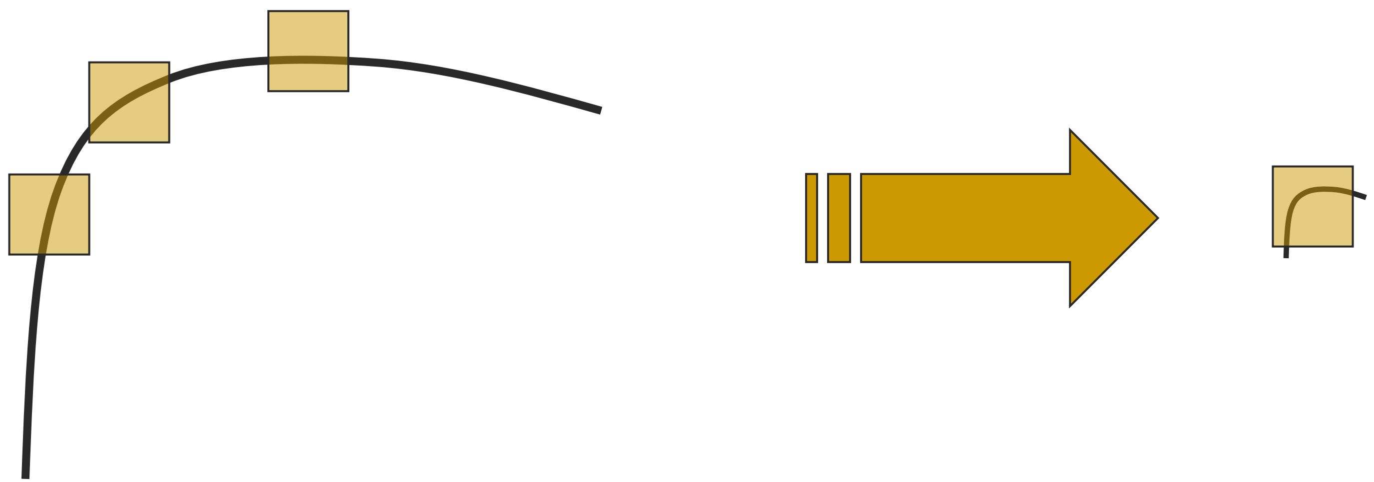
Intensity scale: $I \to a\, I$





Scale invariance?

- Scale invariant? No



All points will be classified as edges

Corner !

Today's Class

- Introduction to correspondence and alignment
- Overview of interest points
  - Matching pipeline
  - Repeatable & Distinctive
- Keypoint Localization
  - Harris detector
  - Hessian detector
- **Scale invariant region selection**
  - Automatic scale selection
  - Laplacian of Gaussian (LoG) & Difference of Gaussian (DoG)
  - Combinations: Harris-Laplacian & Hessian-Laplacian

From points to regions

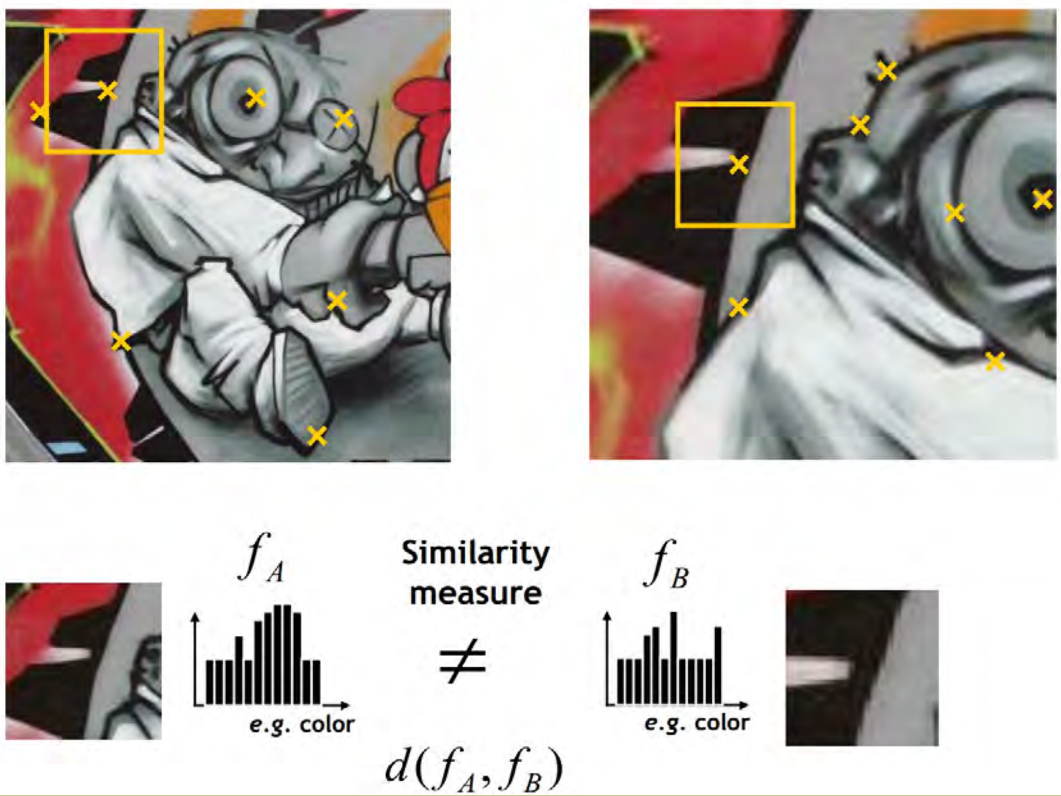
- **The Harris and Hessian operators define interest points.**
  - **Precise localization**
  - **High repeatability**

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

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

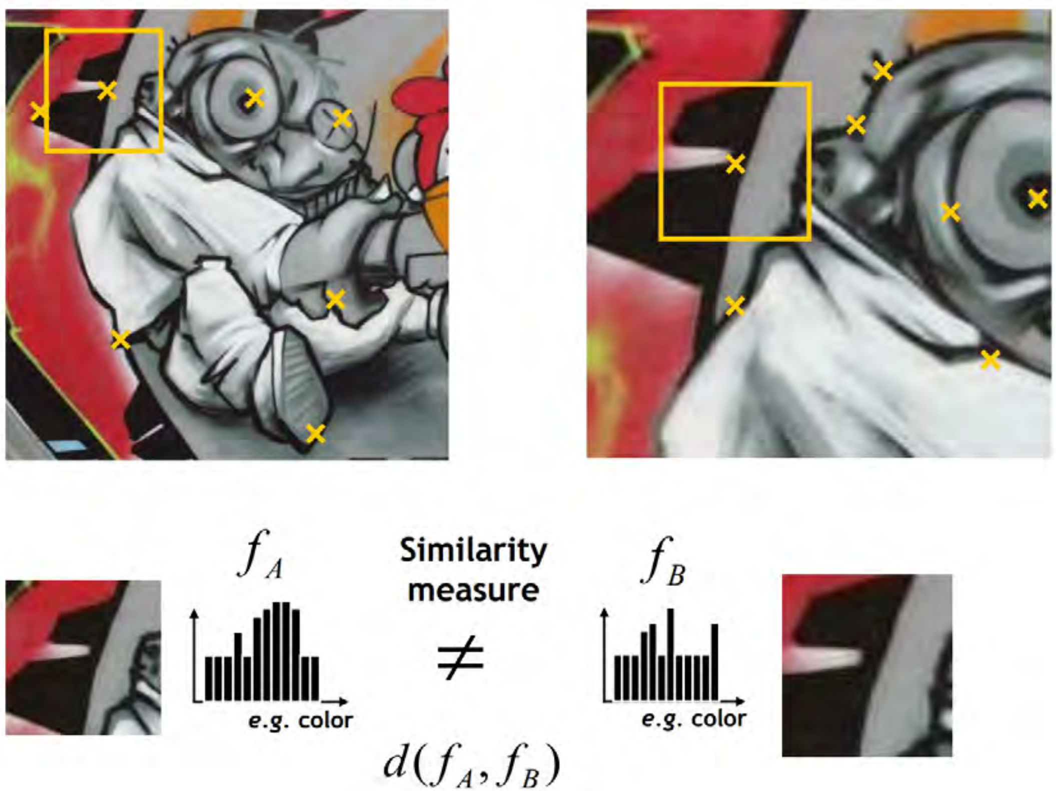
Naïve approach: exhaustive search

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



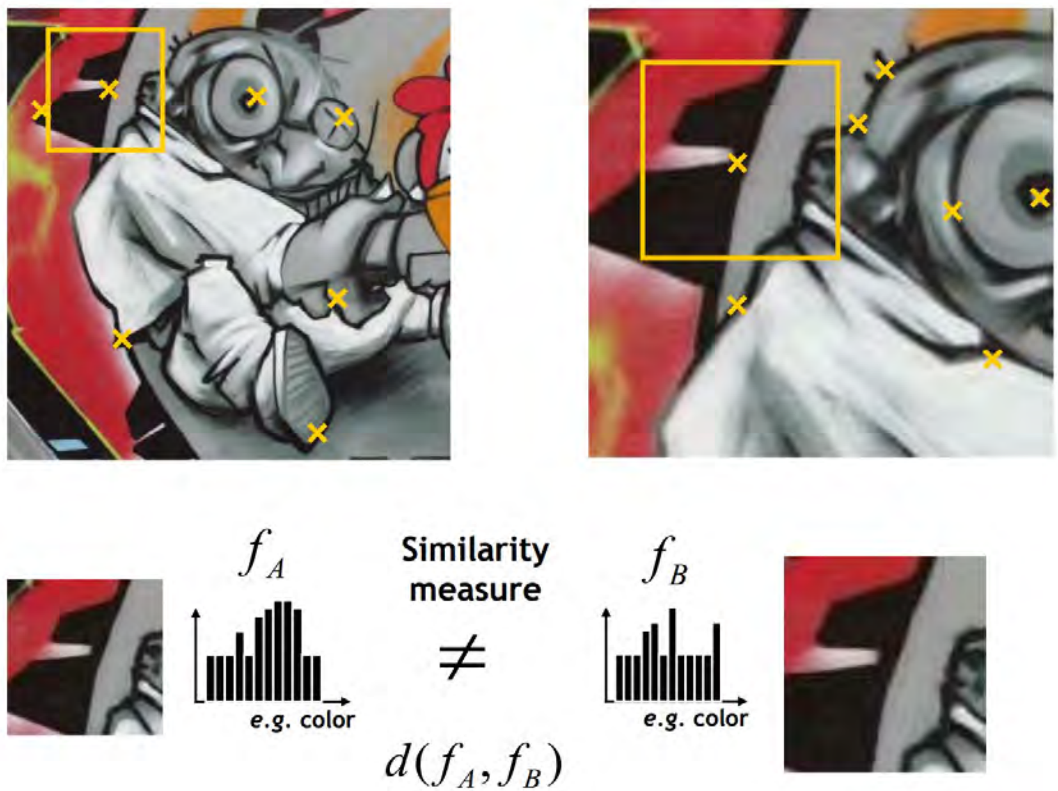
Naïve approach: exhaustive search

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



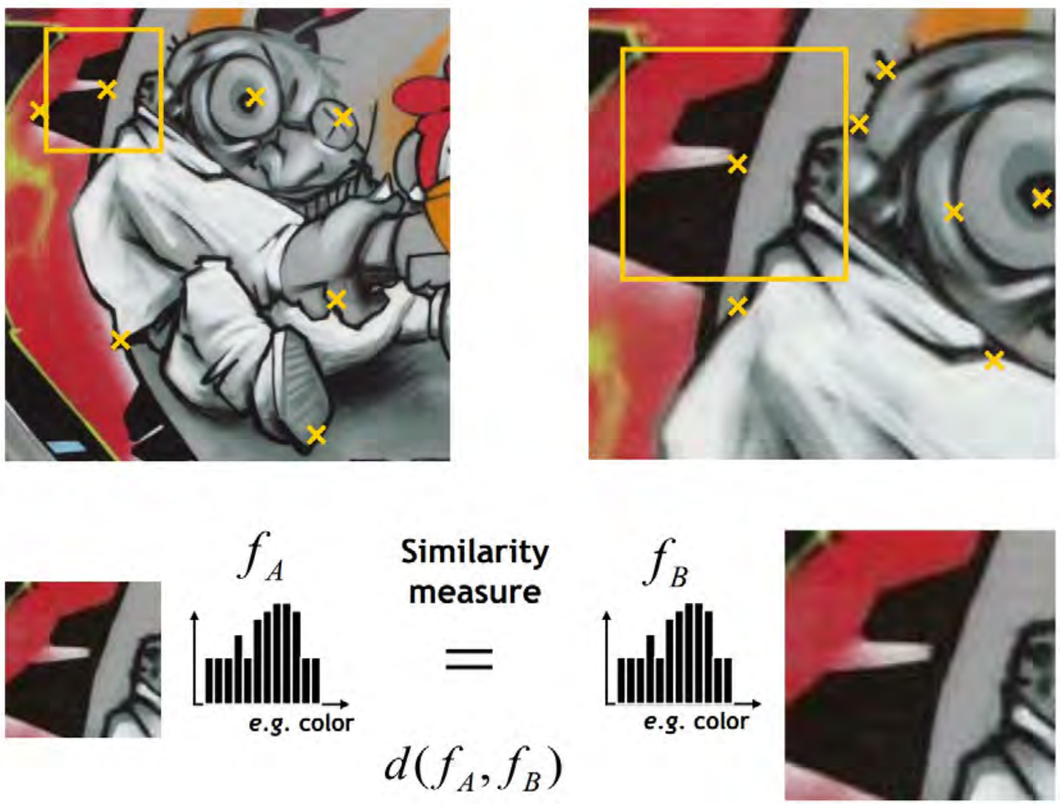
Naïve approach: exhaustive search

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



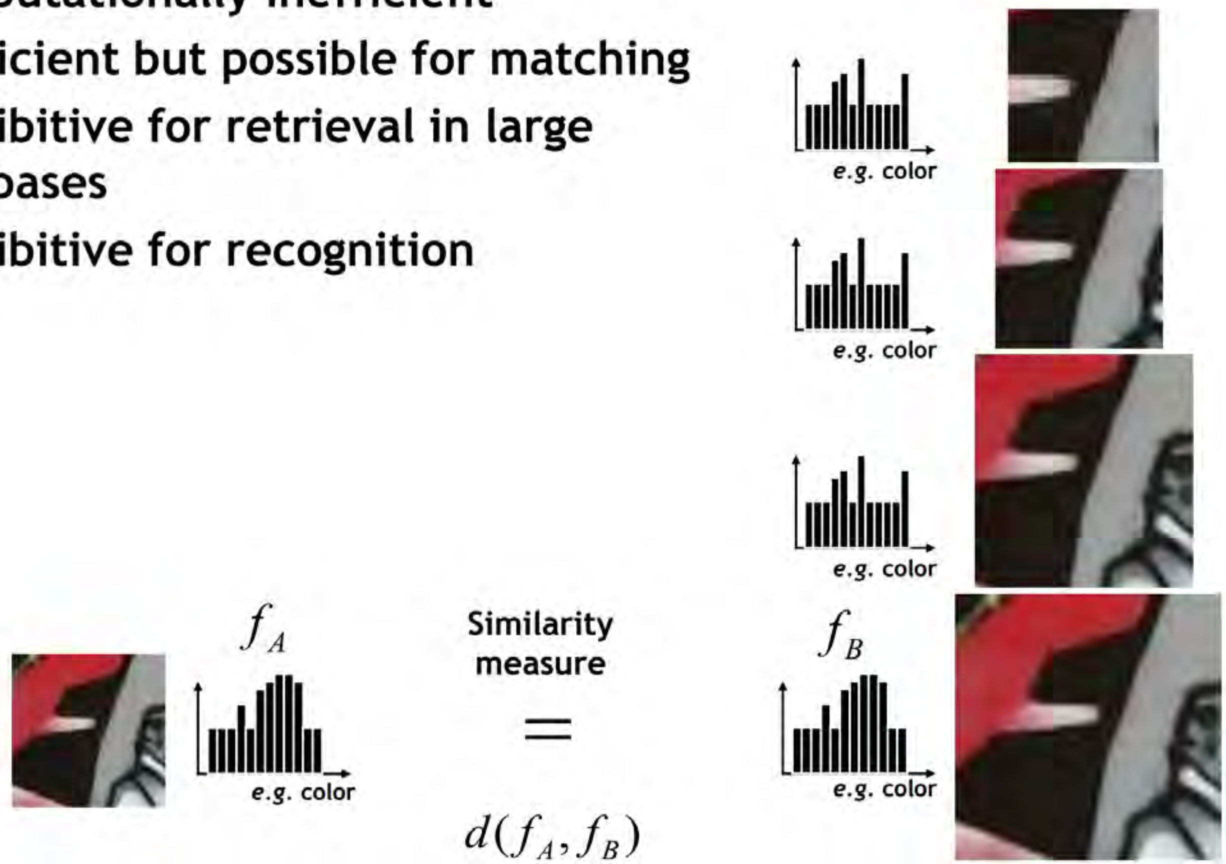
Naïve approach: exhaustive search

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



Naïve approach: exhaustive search

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

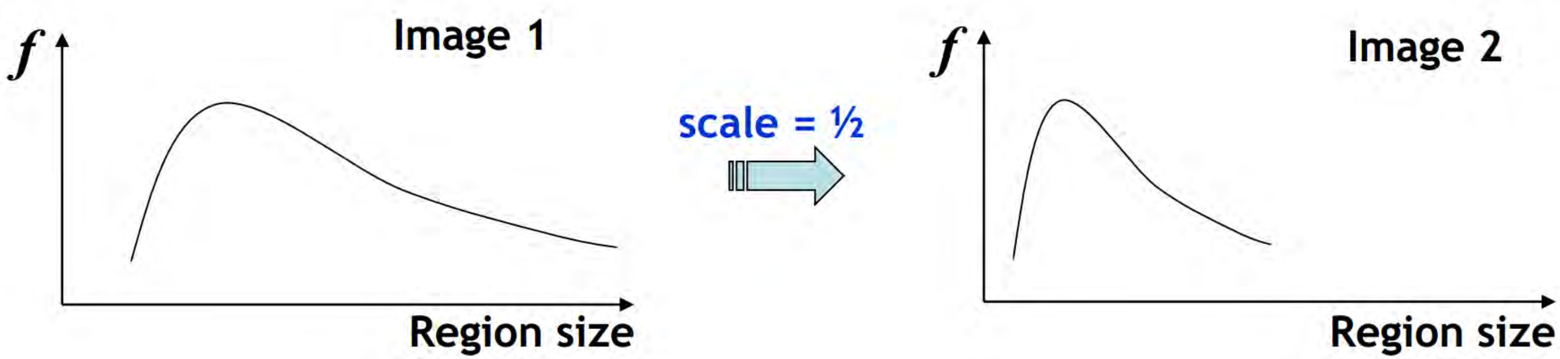


Automatic scale selection

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

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

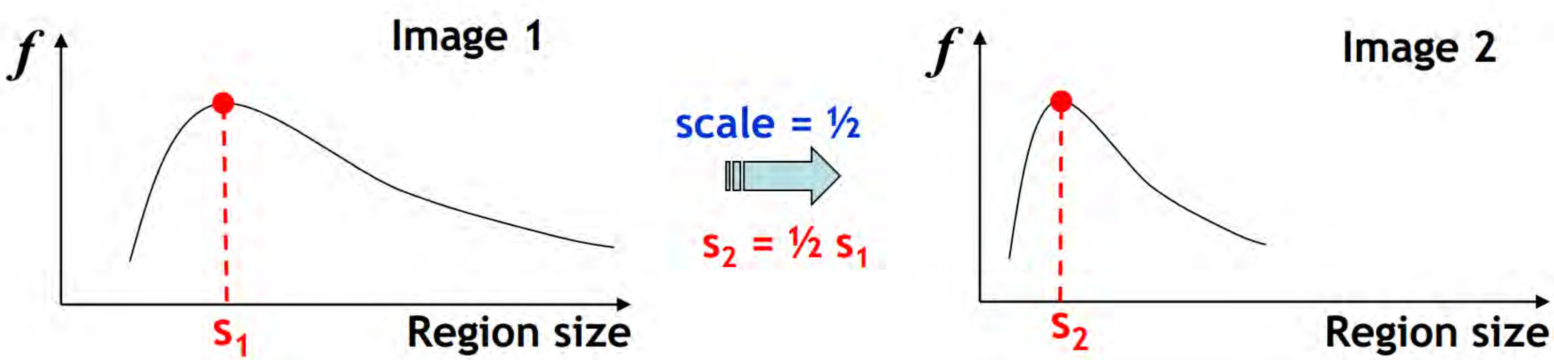
  - **For a point in one image, we can consider it as a function of region size (patch width)**



Automatic scale selection

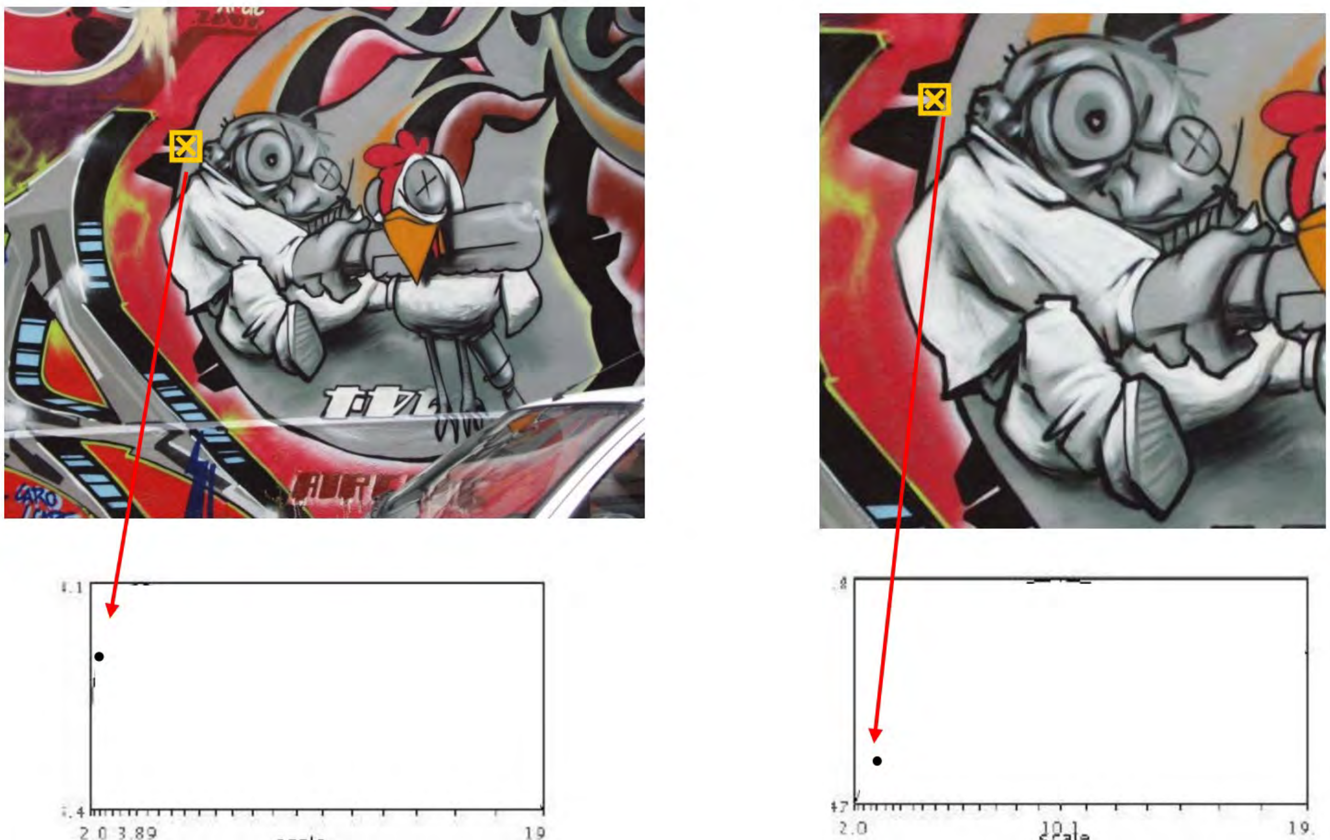
- **Common approach:**
  - Take a local maximum of this function.
  - Observation: region size for which the maximum is achieved should be *invariant* to image scale.

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



Automatic scale selection

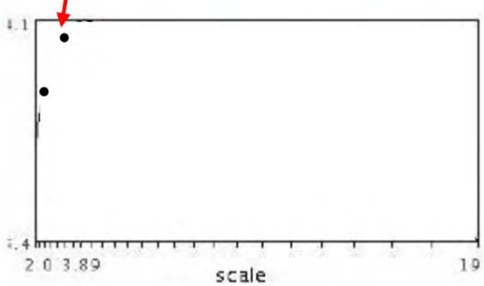
- **Function responses for increasing scale (scale signature)**



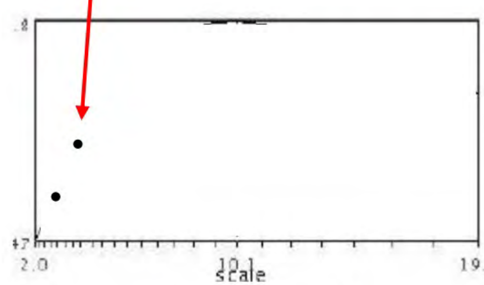
$f(I_{i_1 \dots i_m}(x, \sigma))$ $f(I_{i_1 \dots i_m}(x', \sigma))$

Automatic scale selection

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



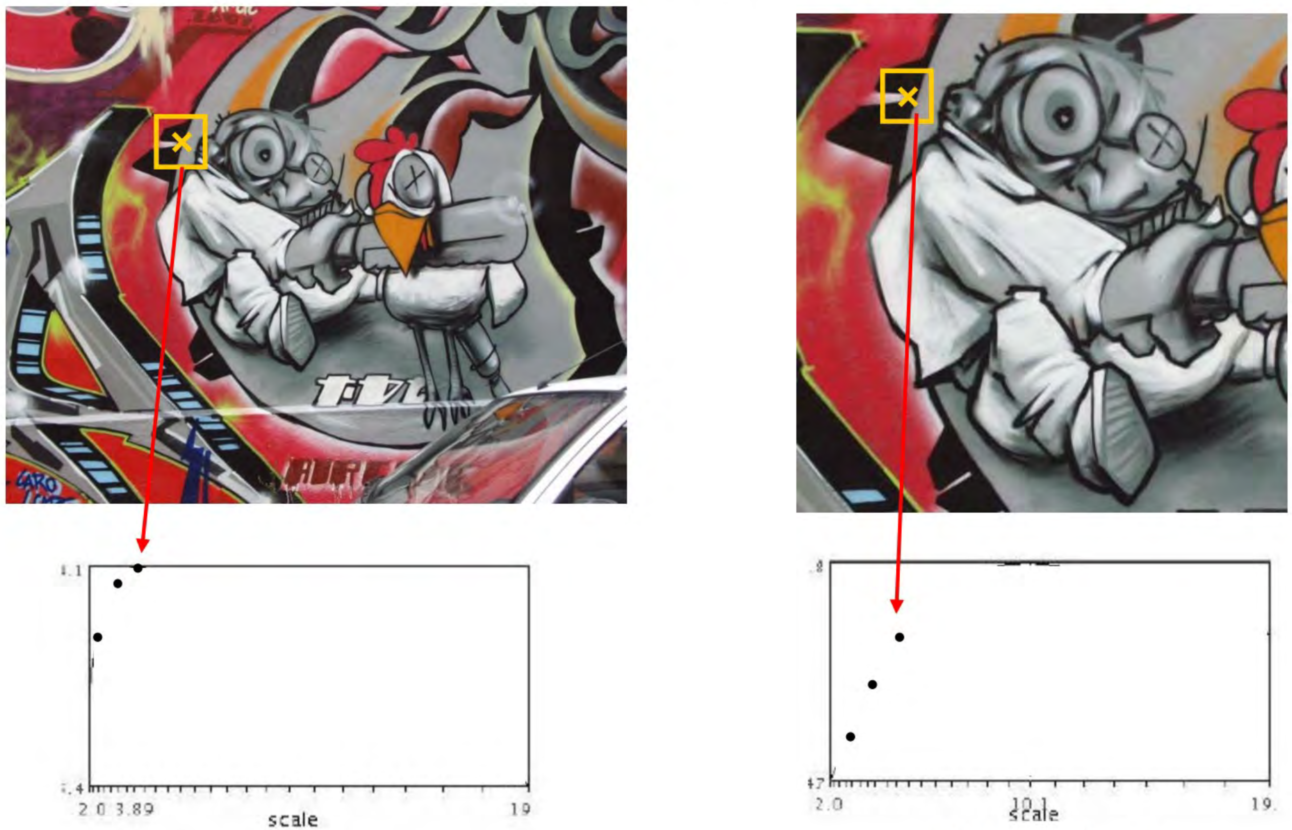
$f(I_{i_1 \ldots i_m}(x, \sigma))$



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

Automatic scale selection

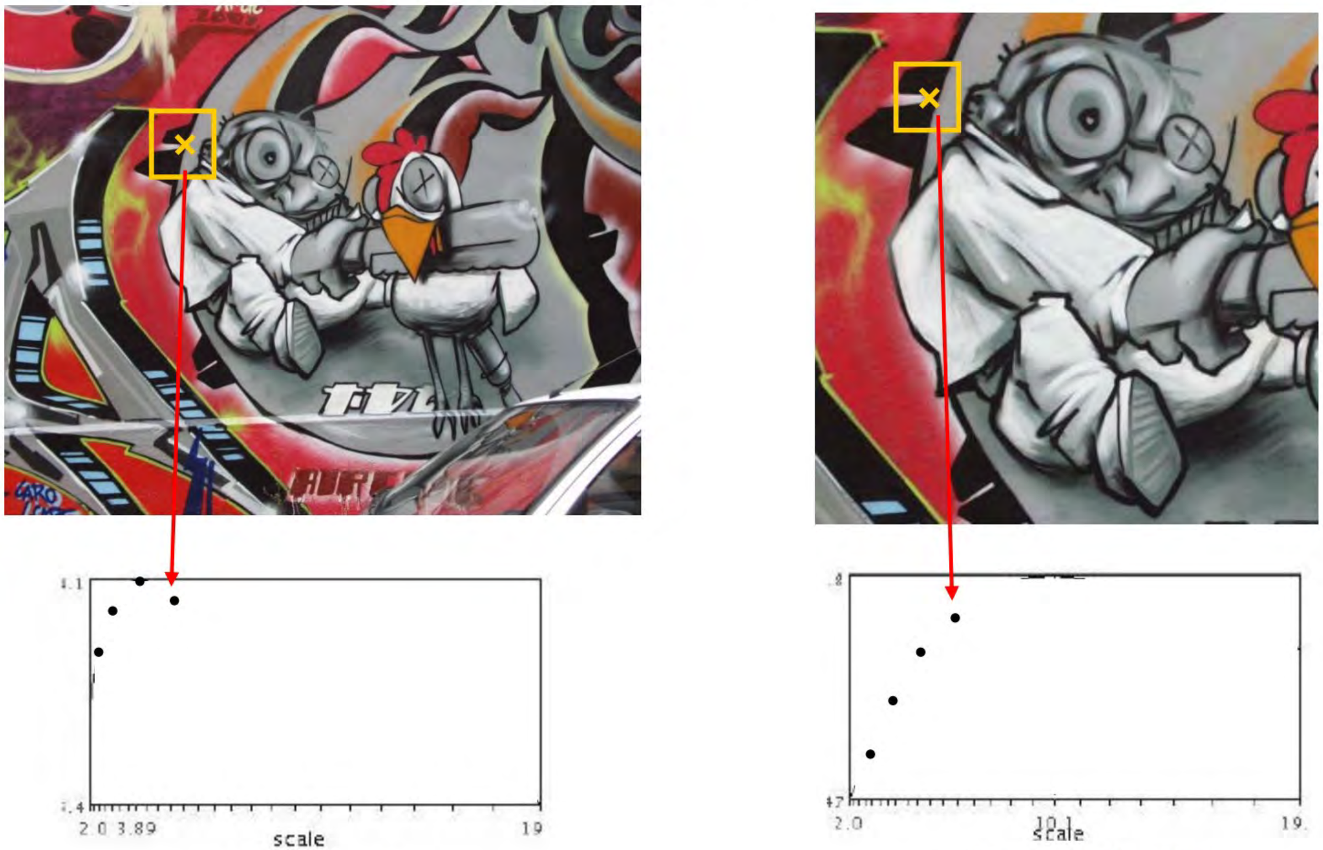
- **Function responses for increasing scale (scale signature)**



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

Automatic scale selection

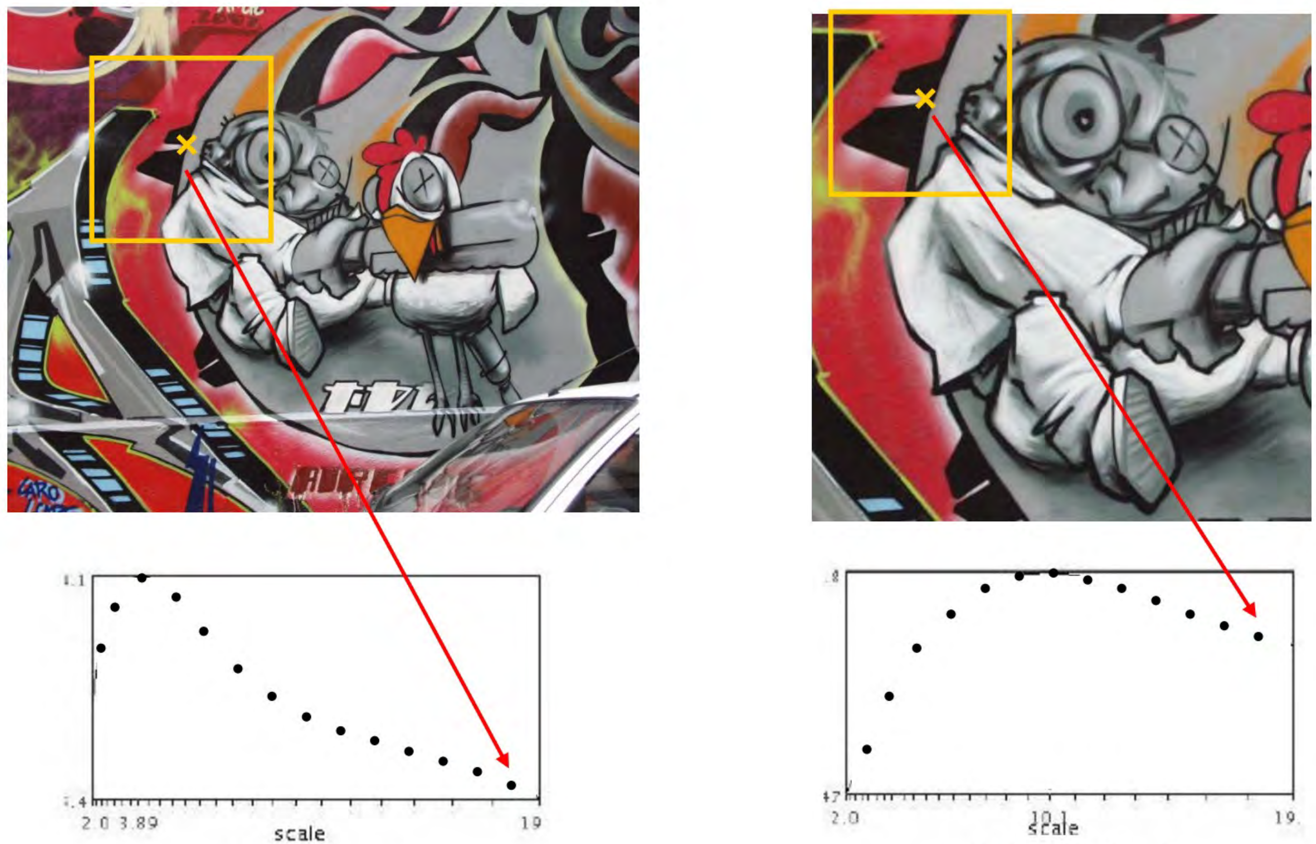
- **Function responses for increasing scale (scale signature)**



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

Automatic scale selection

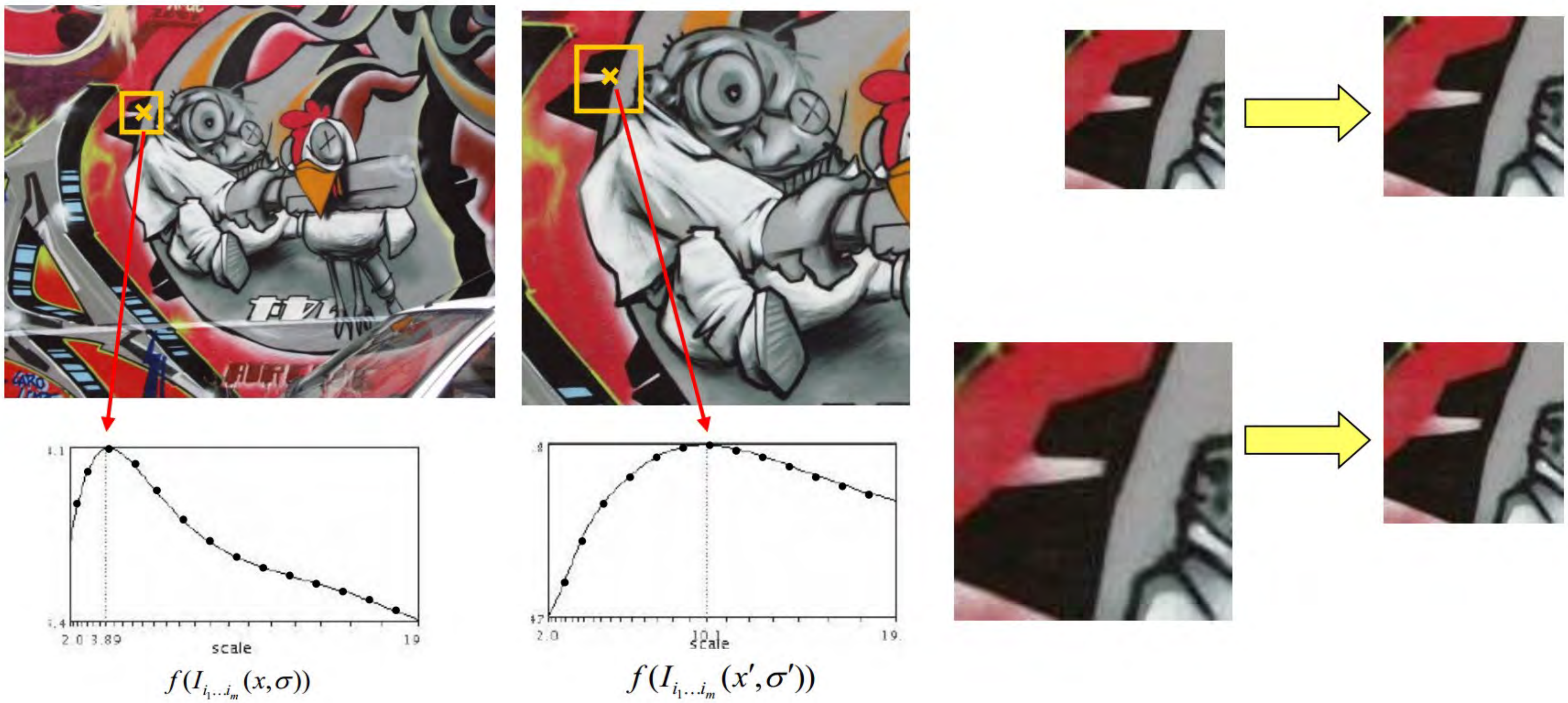
- **Function responses for increasing scale (scale signature)**



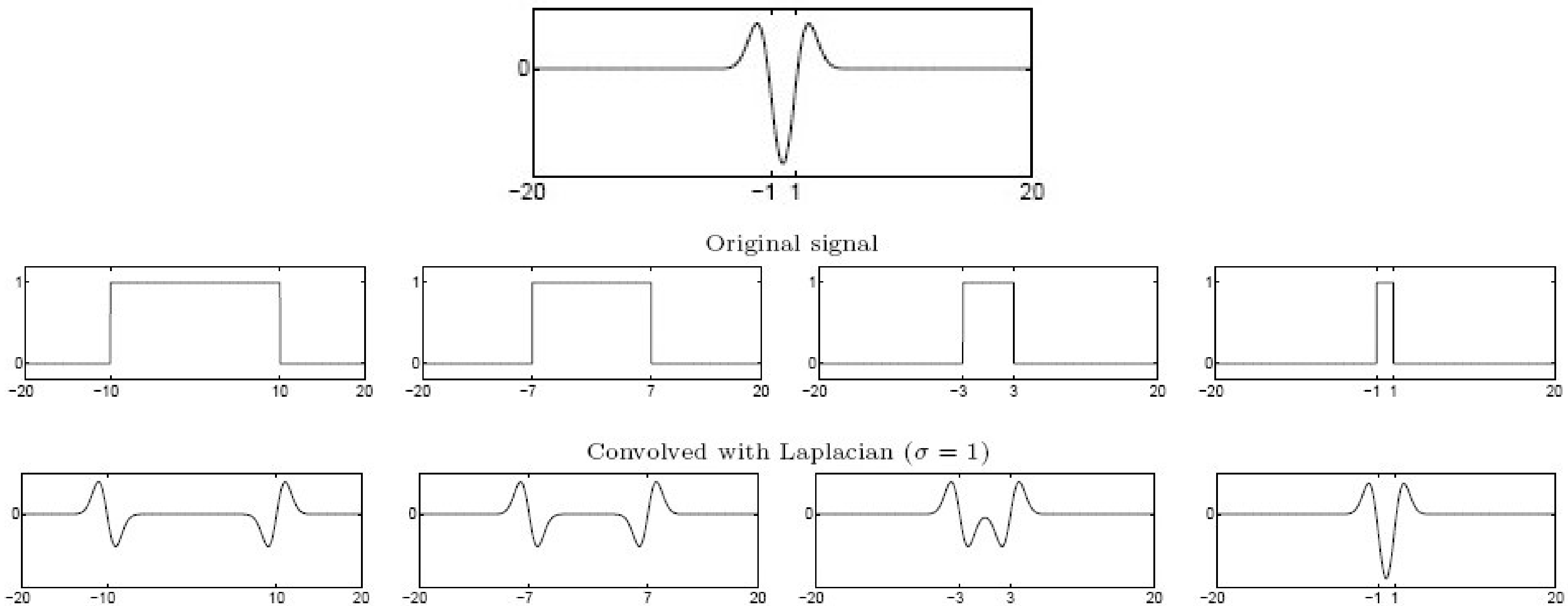
$f(I_{i_1 \ldots i_m}(x, \sigma))$ $\qquad$ $f(I_{i_1 \ldots i_m}(x', \sigma))$

Automatic scale selection

- **Normalize: Rescale to fixed size**



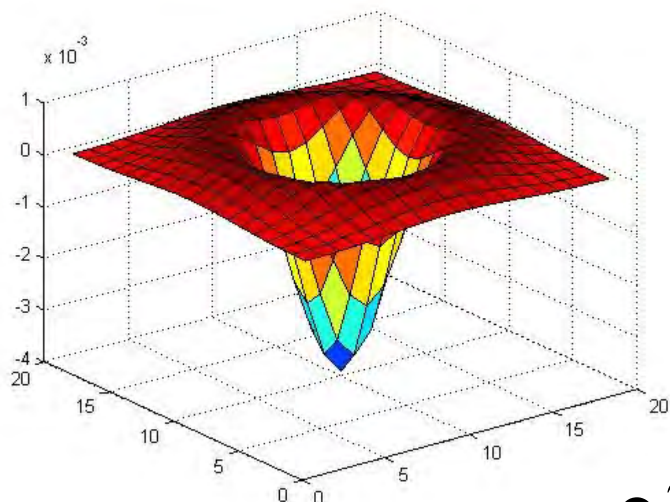
What can be the “signature” function?

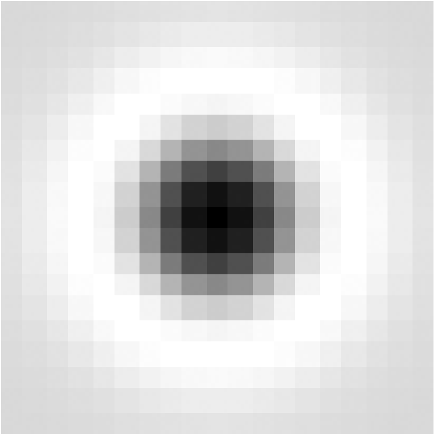


Highest response when the signal has the same **characteristic scale** as the filter

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

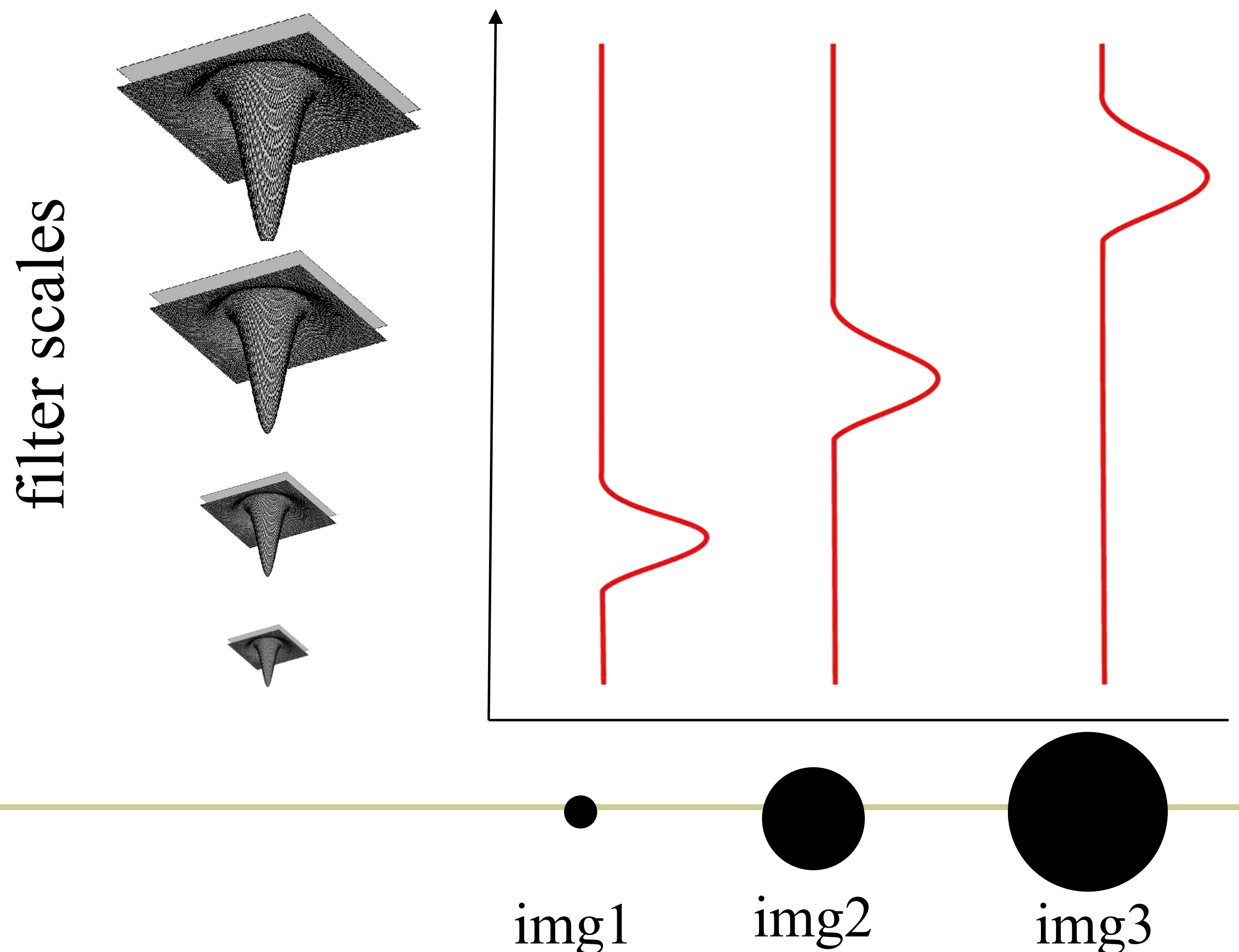




$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

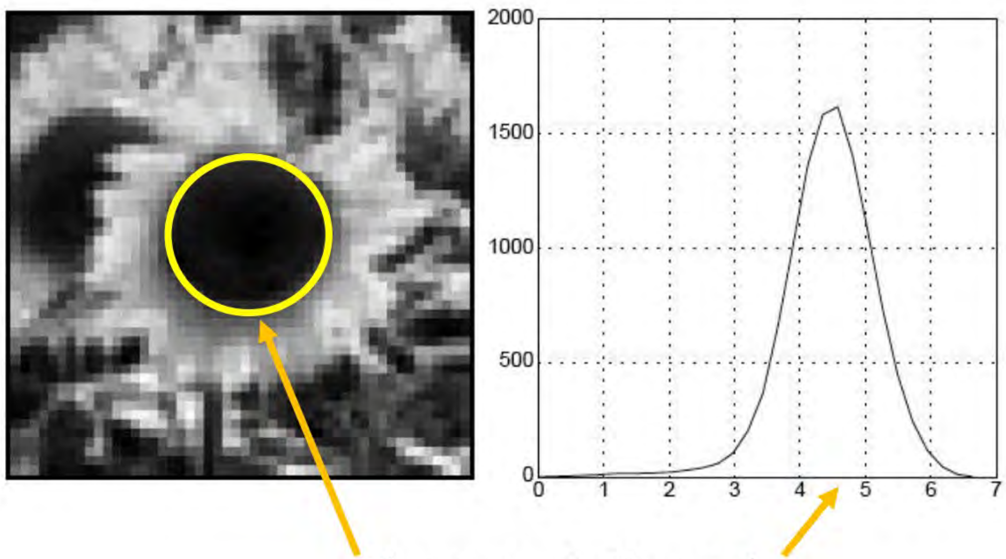
Blob detection in 2D

- Laplacian-of-Gaussian = “blob” detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$



Characteristic scale

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



Characteristic scale

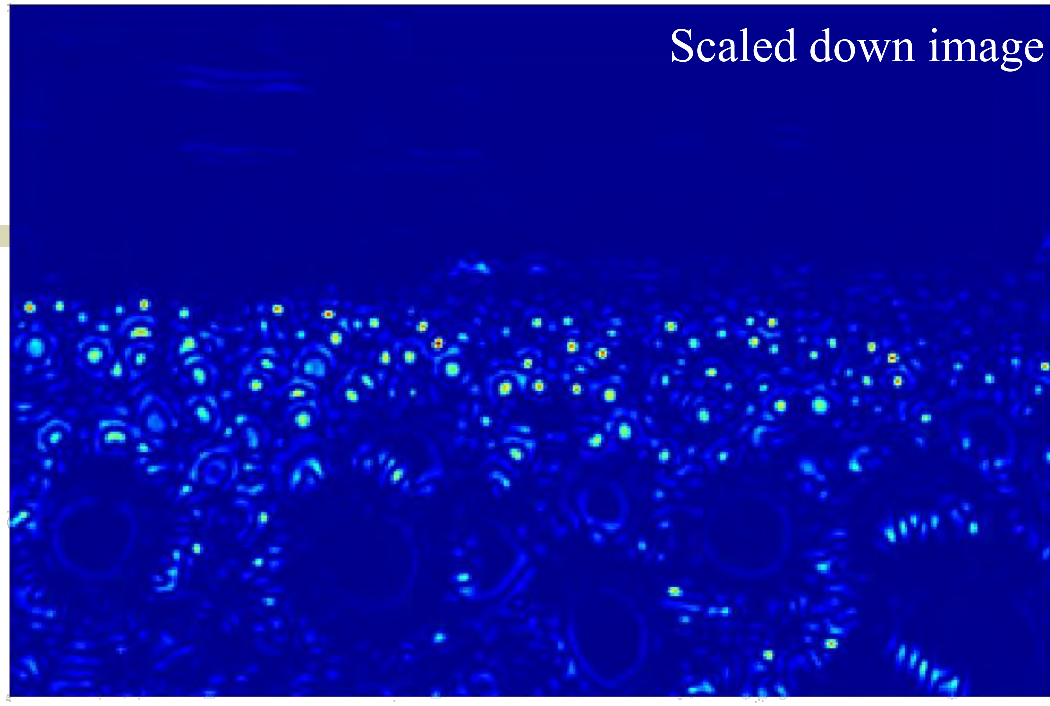
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

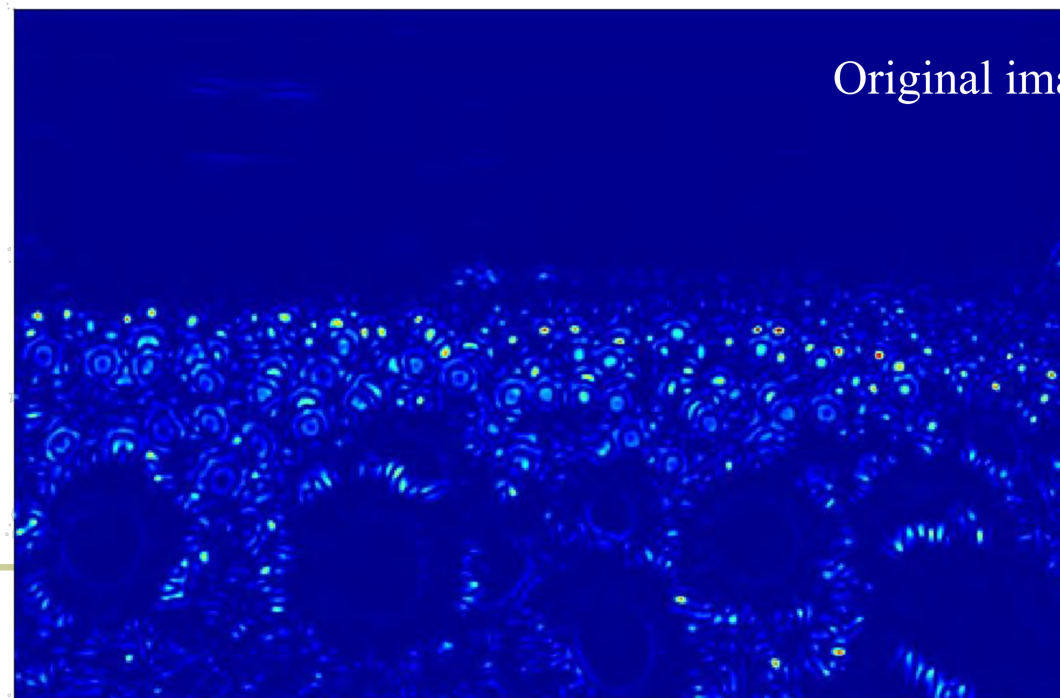
Example

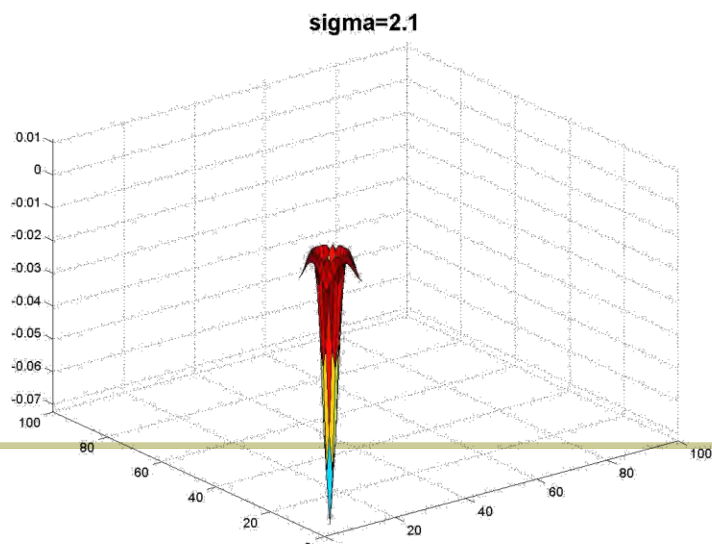
Original image at
¾ the size

What happened when you applied different Laplacian filters?

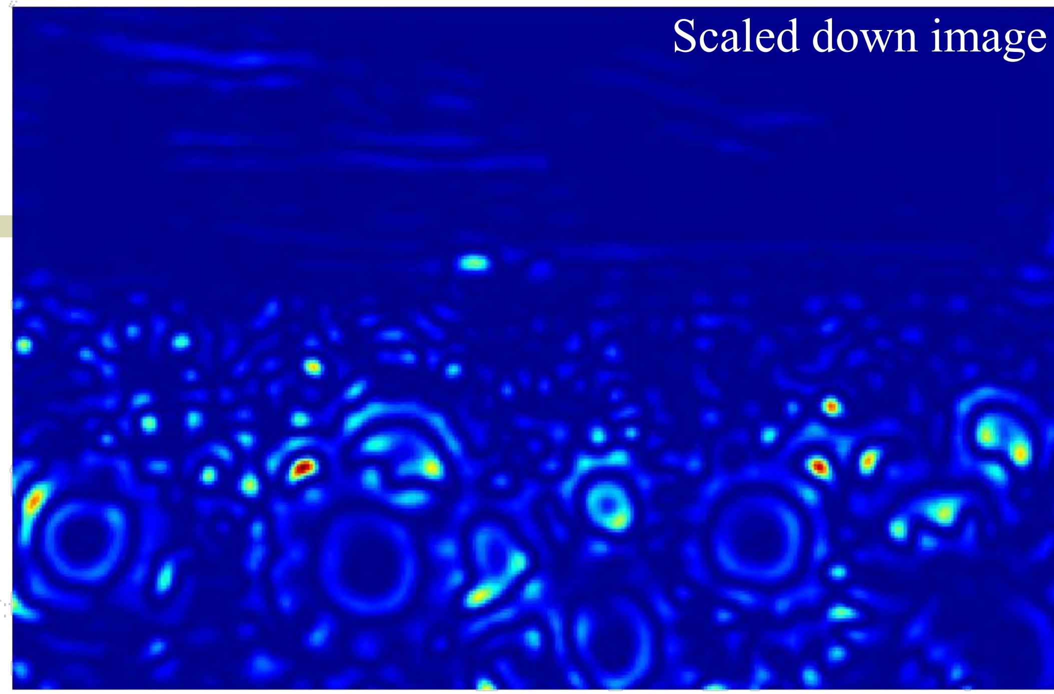
Original image at
¾ the size

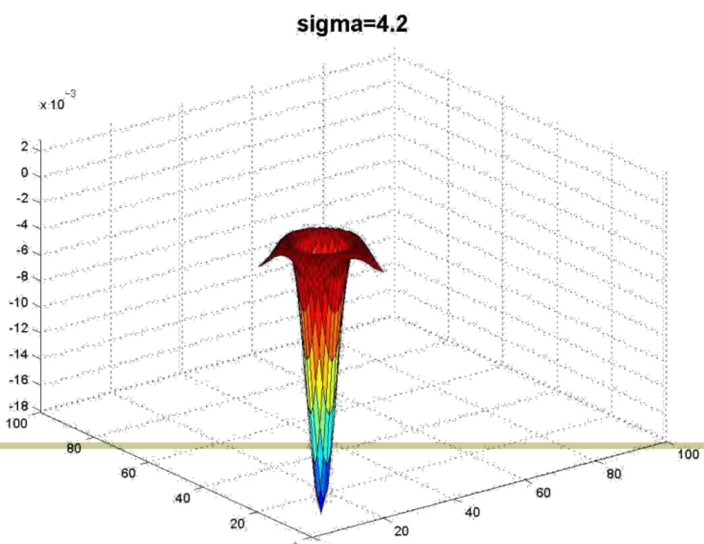


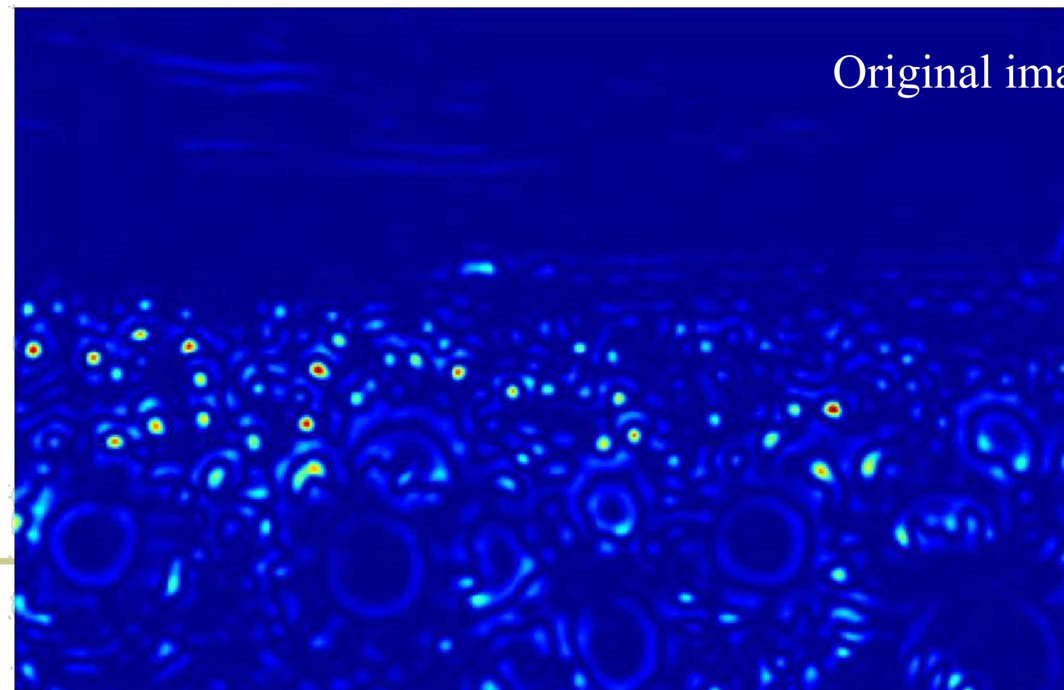




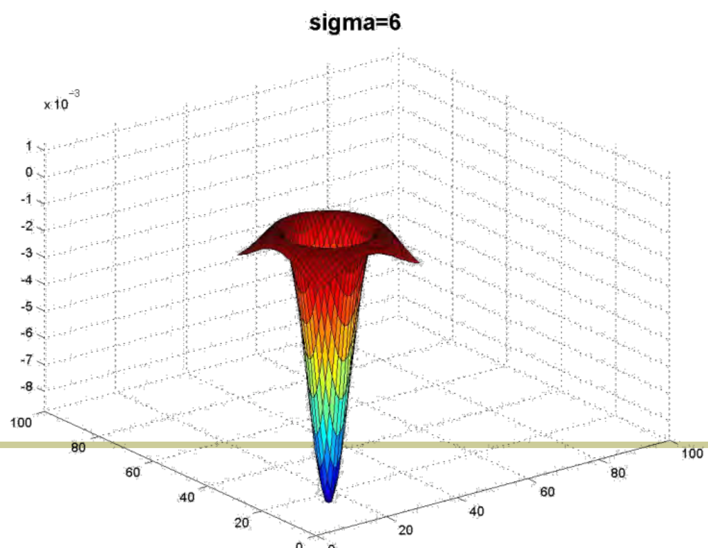
Slide credit: Kristen Grauman



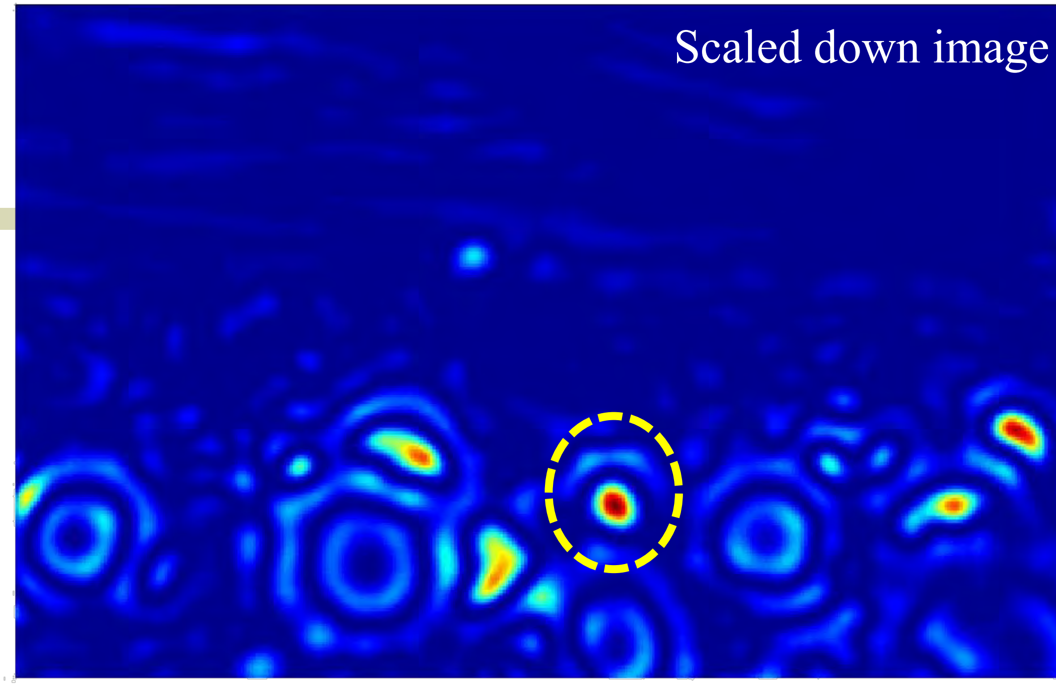


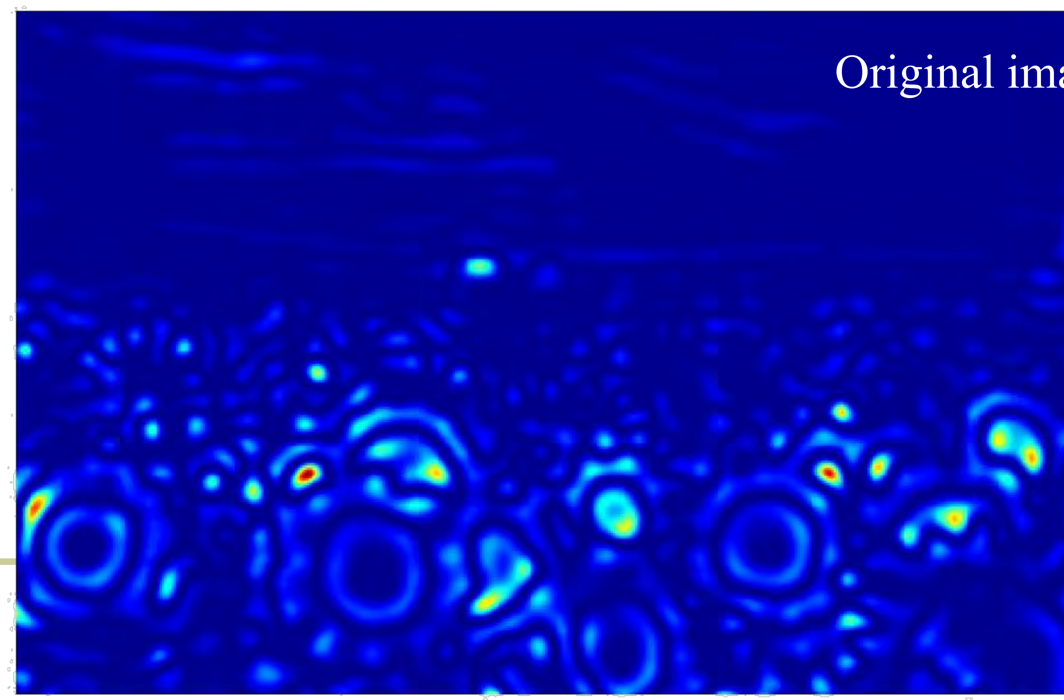


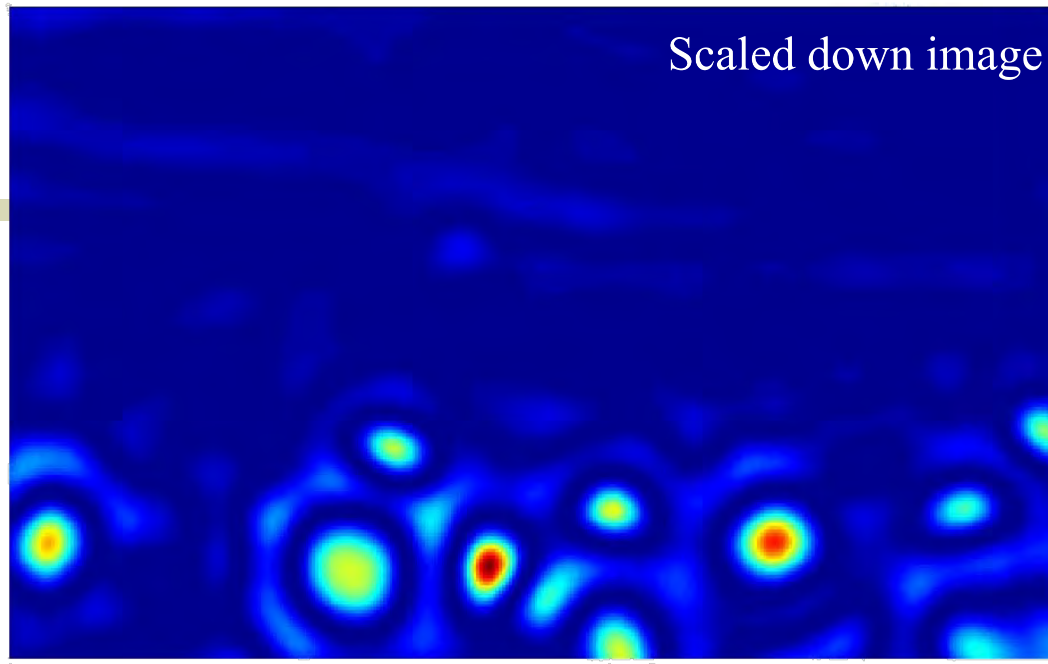
Slide credit: Kristen Grauman

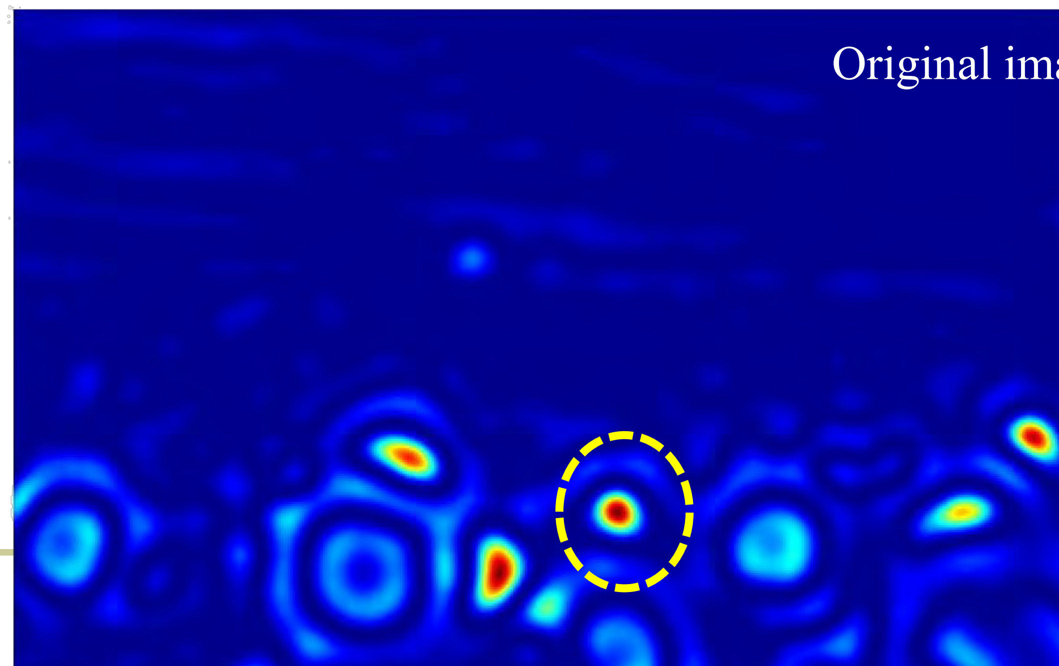


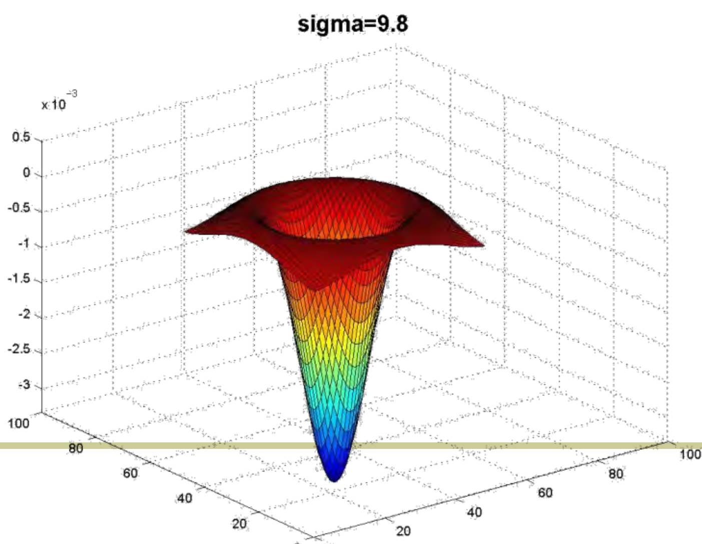
Slide credit: Kristen Grauman



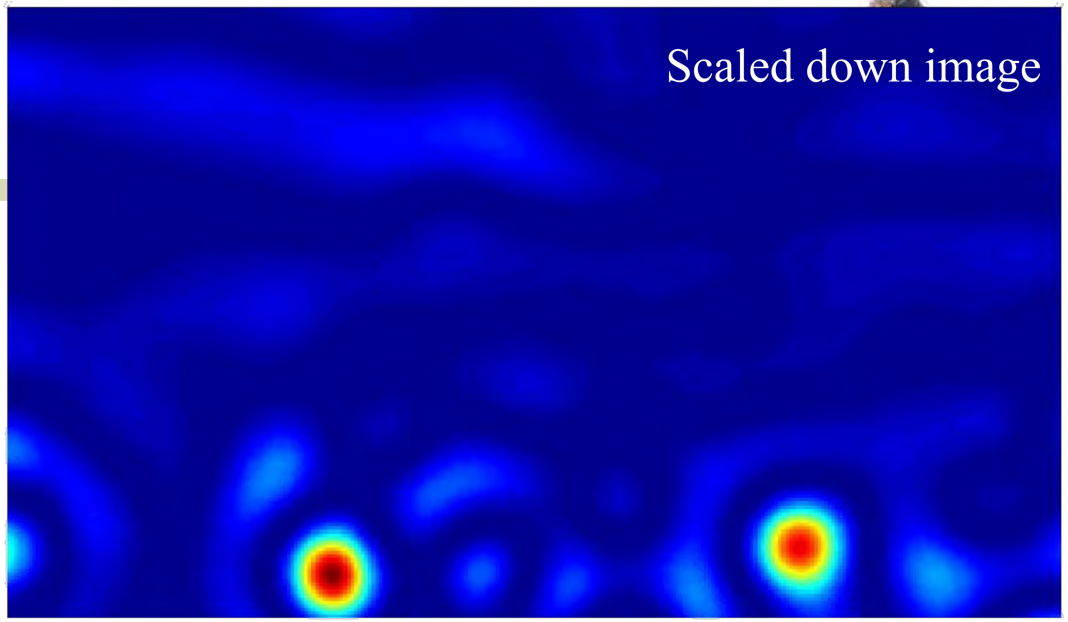


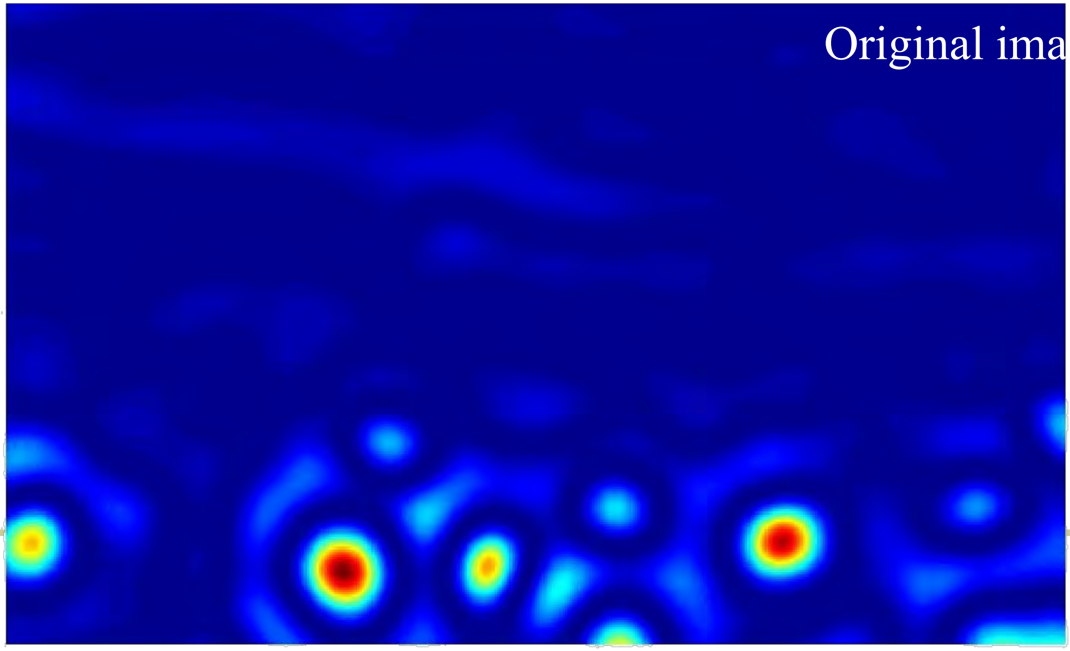


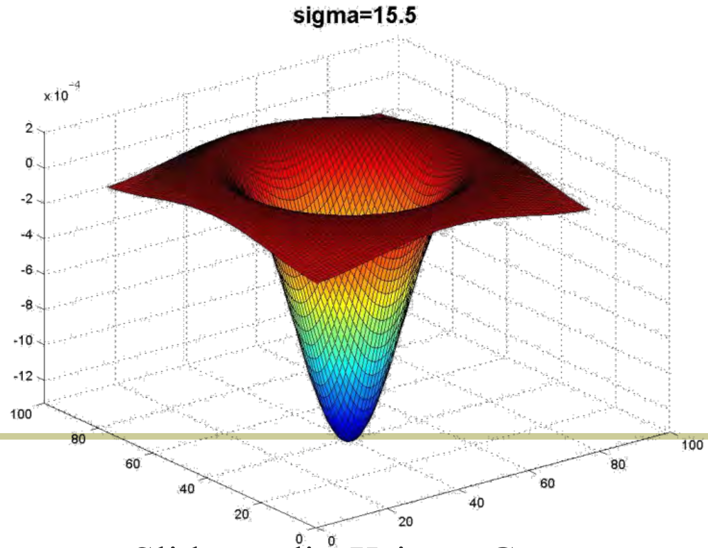




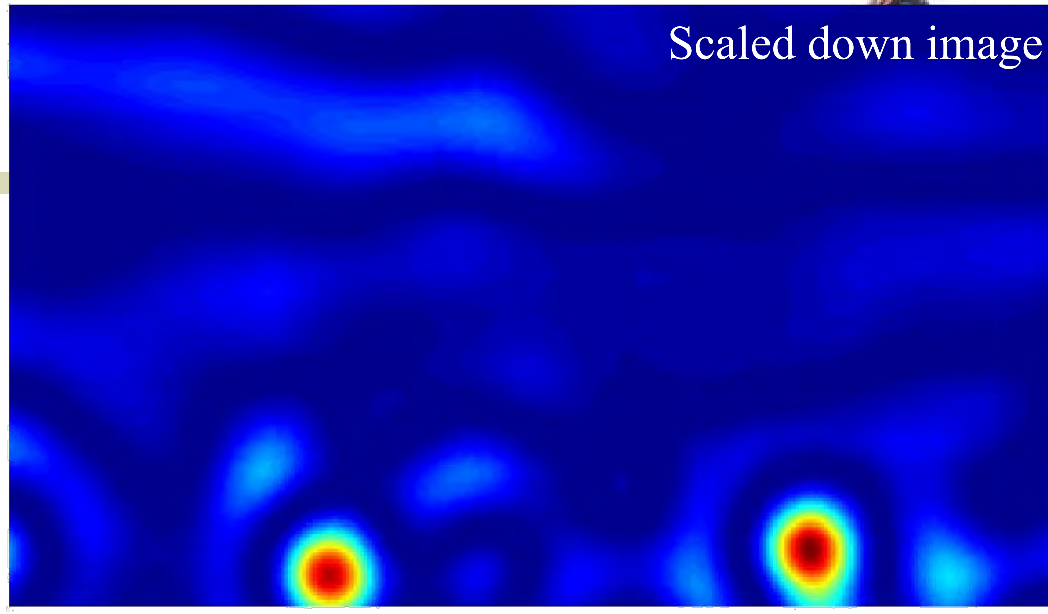
Slide credit: Kristen Grauman

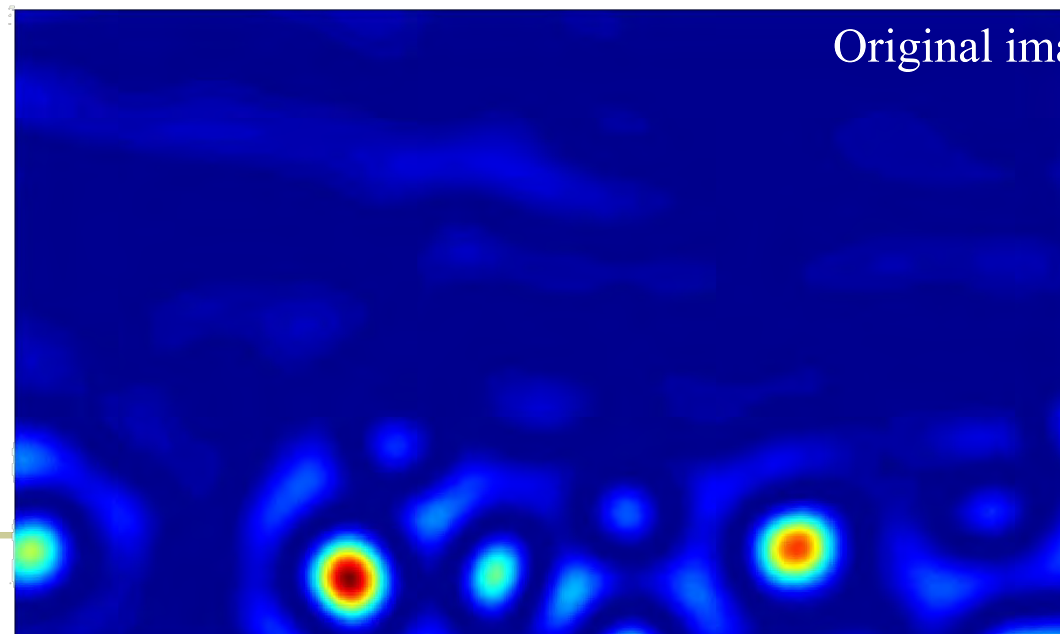


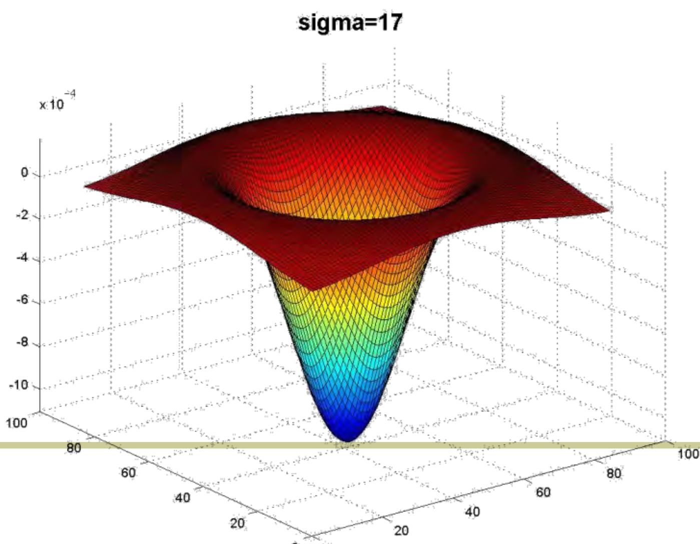




Slide credit: Kristen Grauman



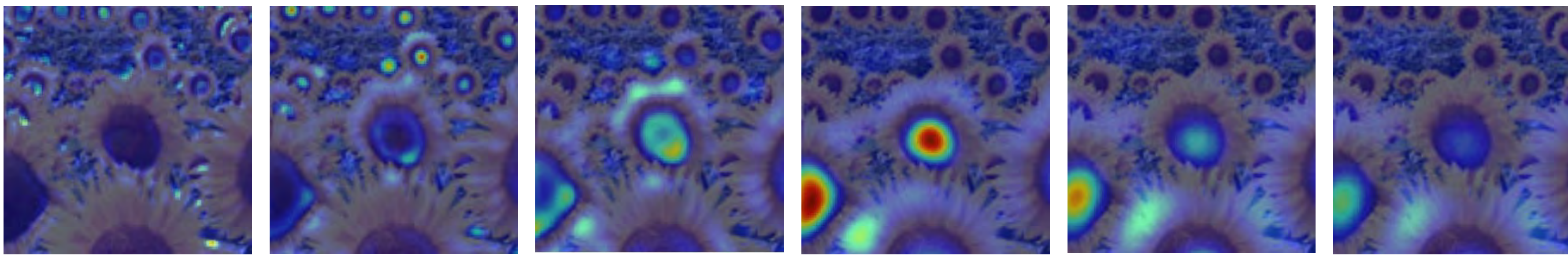




Slide credit: Kristen Grauman

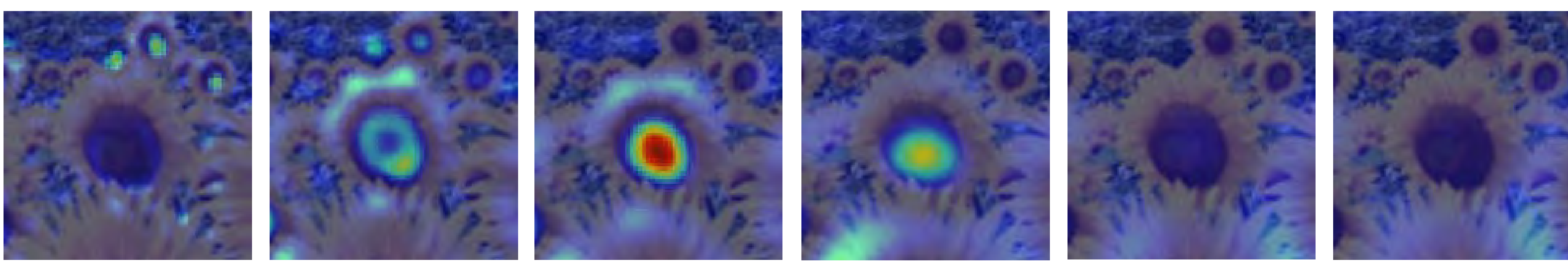
Optimal scale

2.1 4.2 6.0 9.8 15.5 17.0



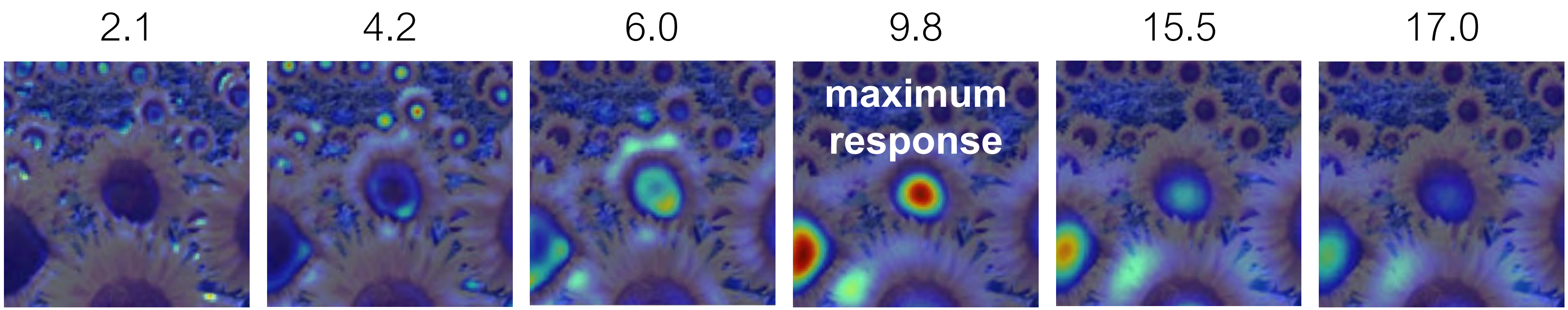
Full size image

2.1 4.2 6.0 9.8 15.5 17.0

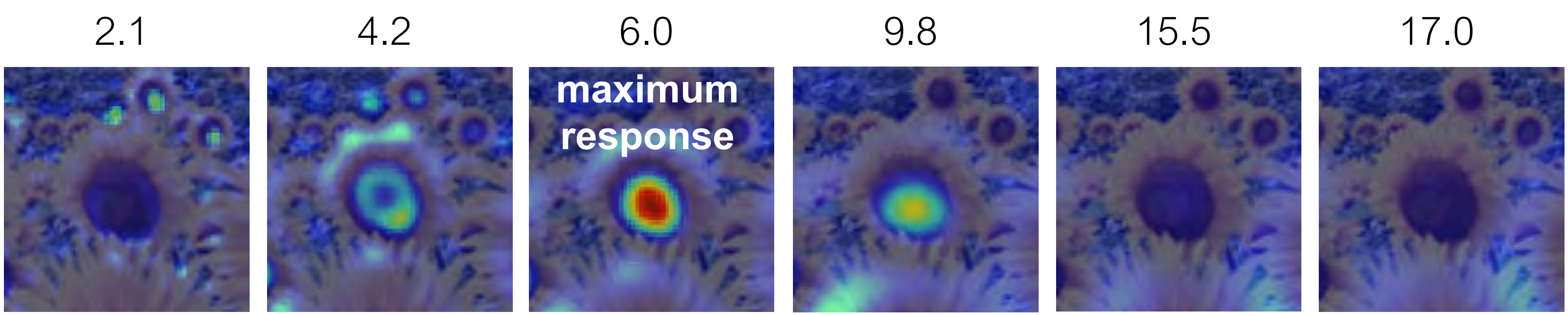


3/4 size image

Optimal scale



Full size image

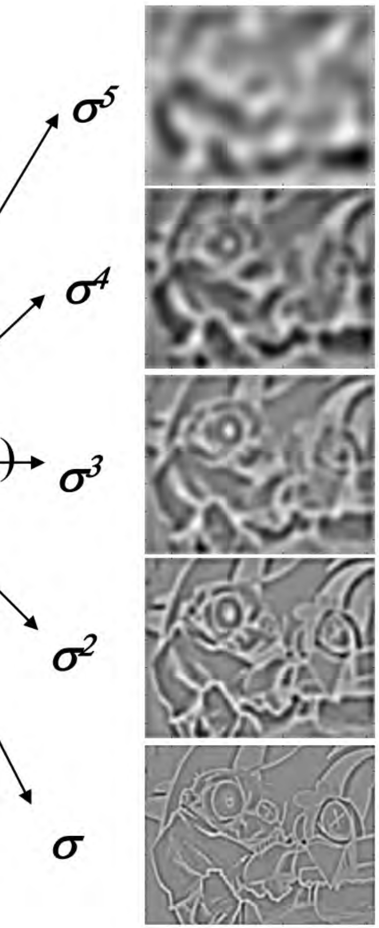


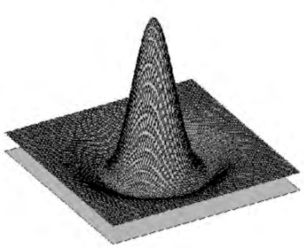
3/4 size image

Laplacian-of-Gaussian (LoG)

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

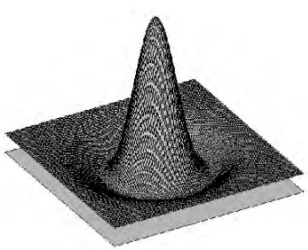
$$L_{xx}(\sigma)+L_{yy}(\sigma)$$

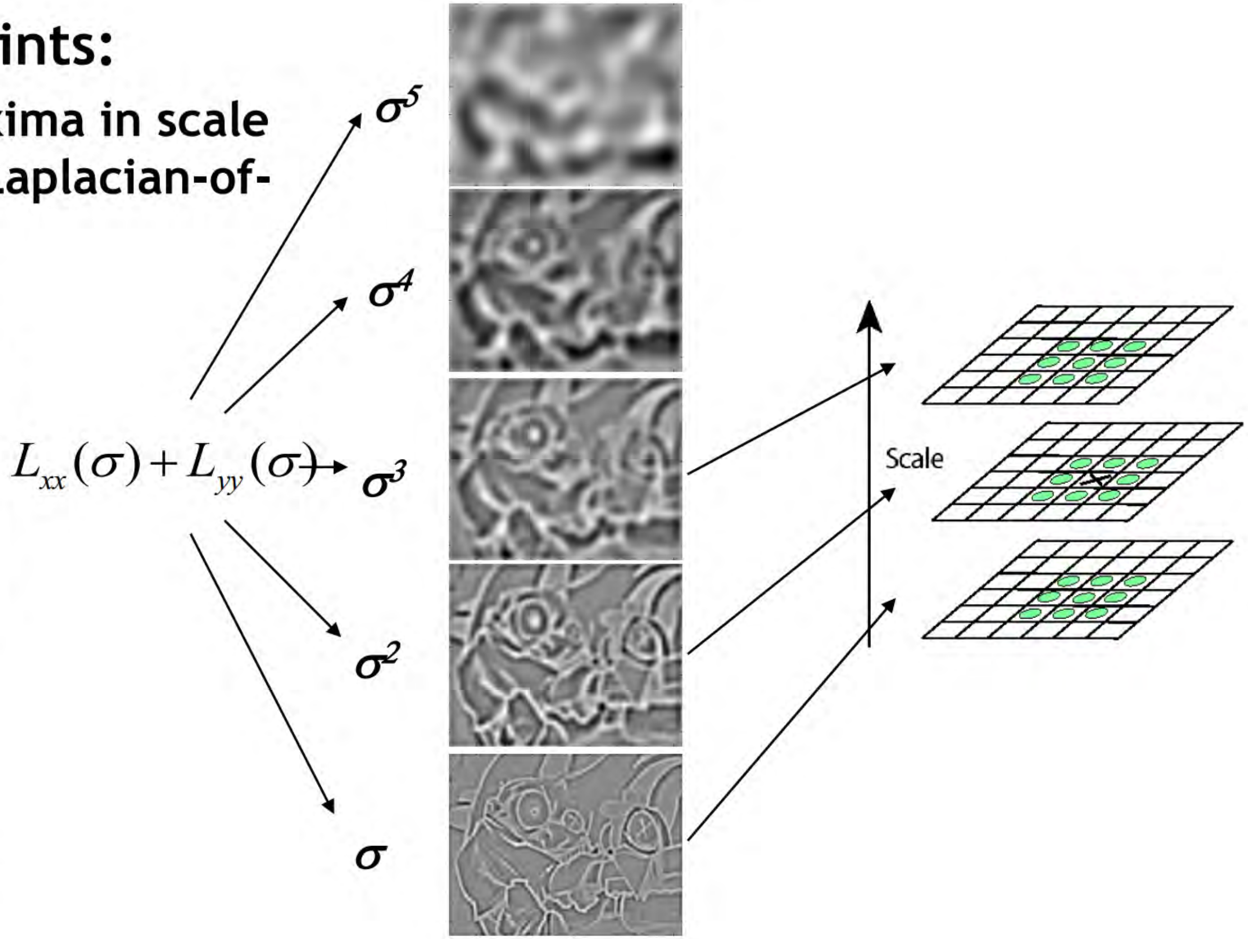




Laplacian-of-Gaussian (LoG)

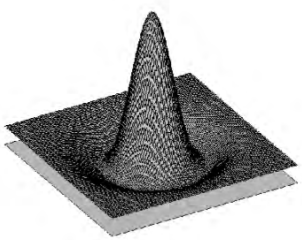
- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**

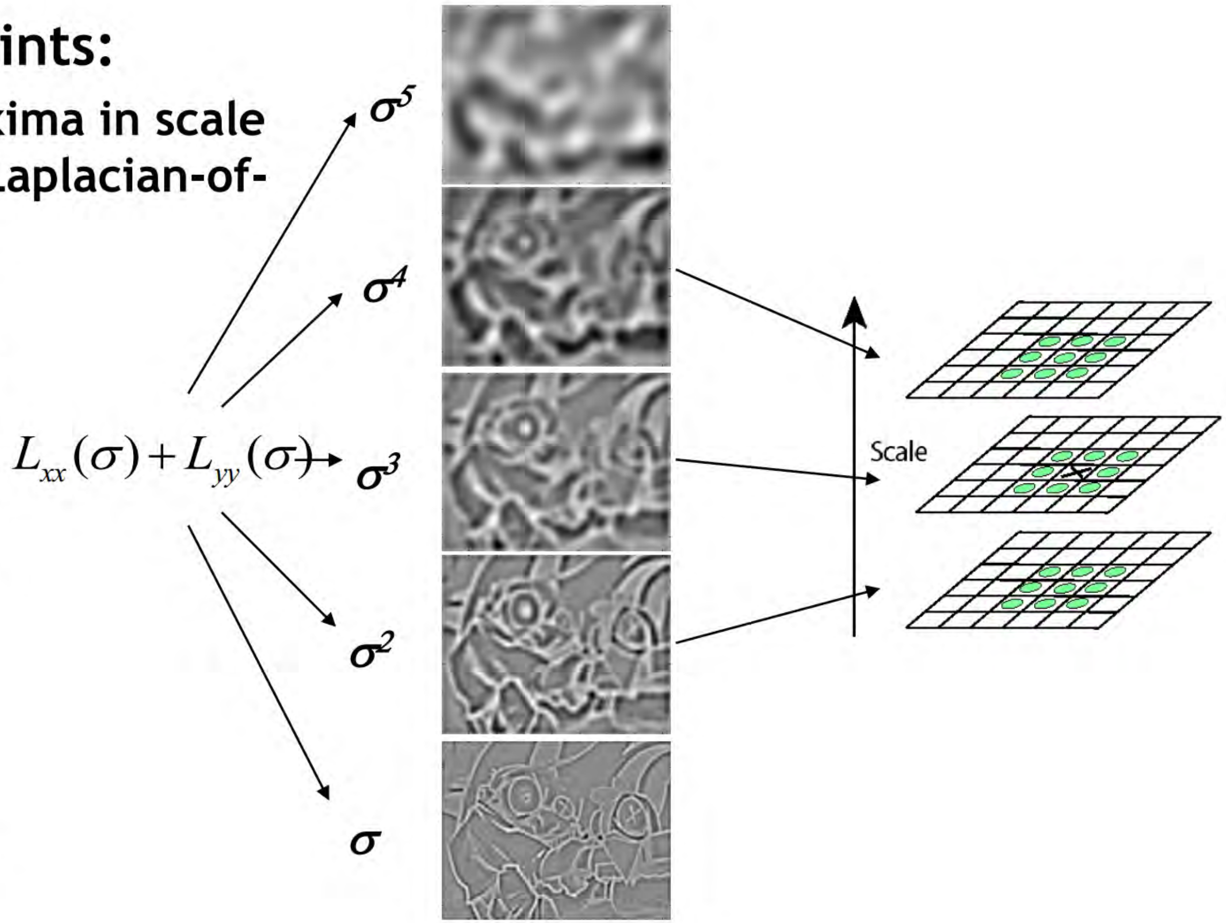




Laplacian-of-Gaussian (LoG)

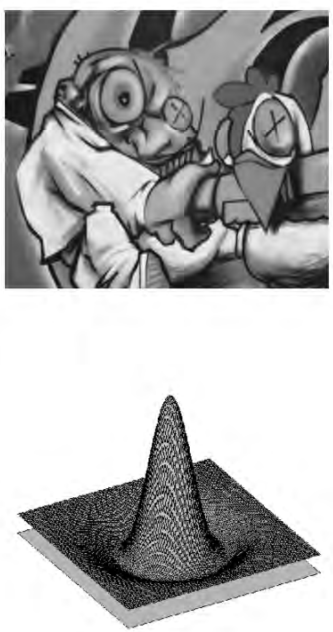
- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**

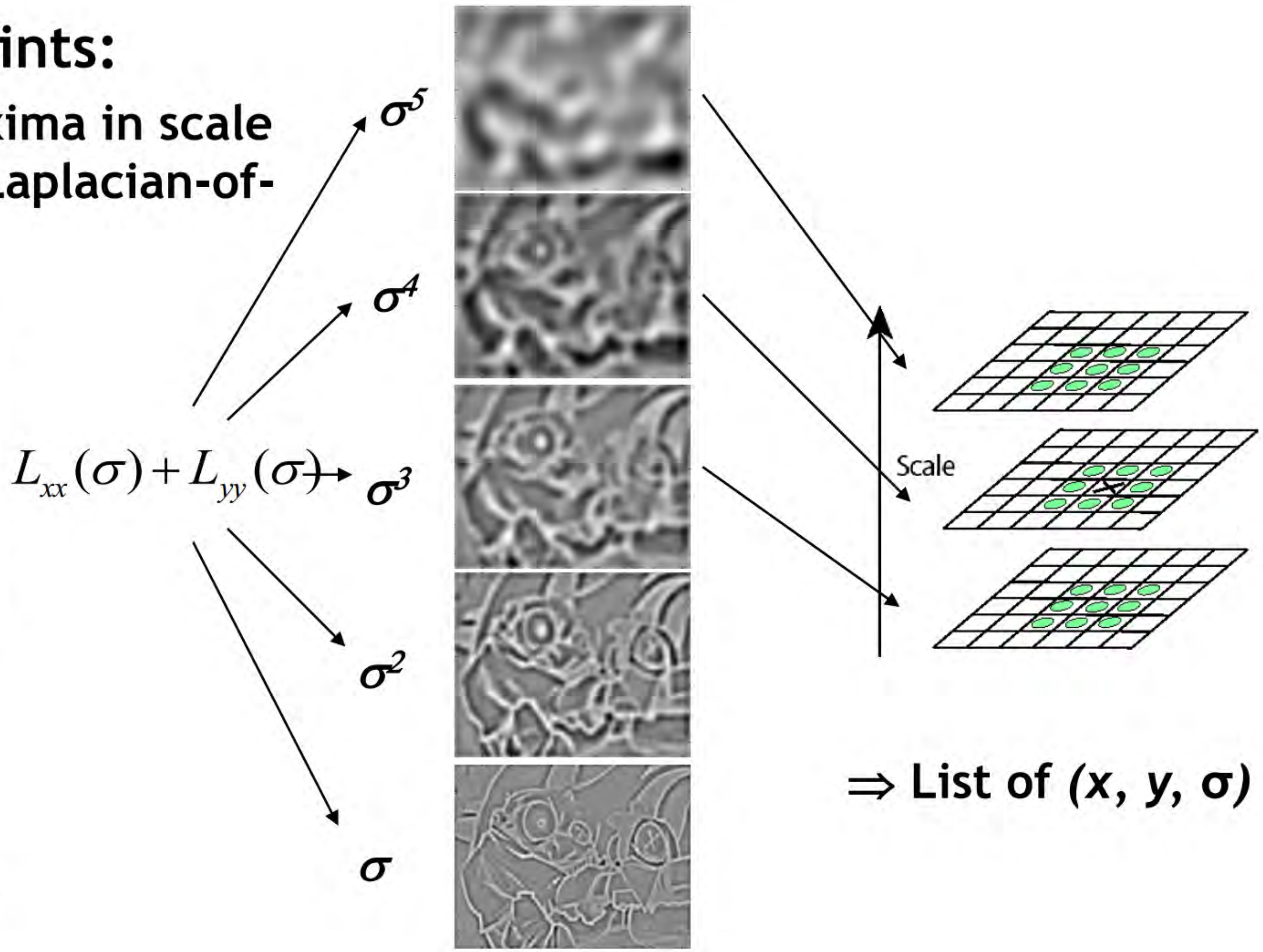




Laplacian-of-Gaussian (LoG)

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





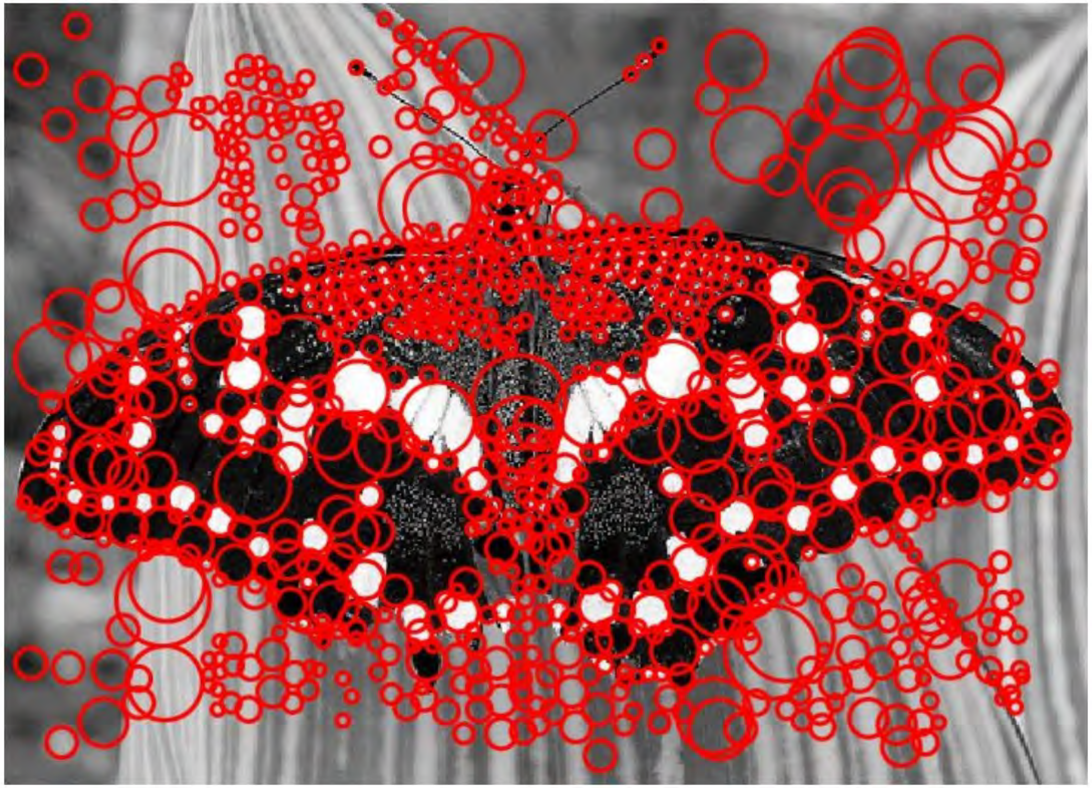
⇒ List of *(x, y, σ)*

LoG detector: workflow

LoG detector: workflow

sigma = 11.9912

LoG detector: workflow



Technical detail

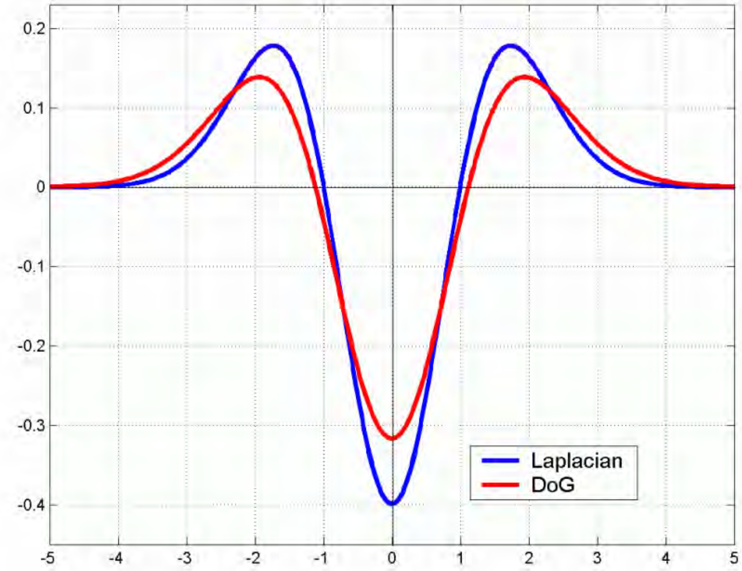
- **We can efficiently approximate the Laplacian with a difference of Gaussians:**

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

(Laplacian)

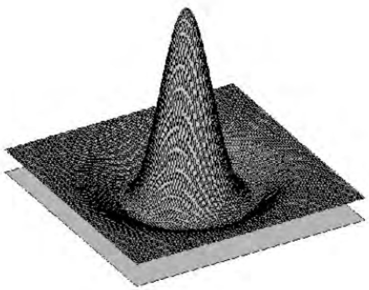
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



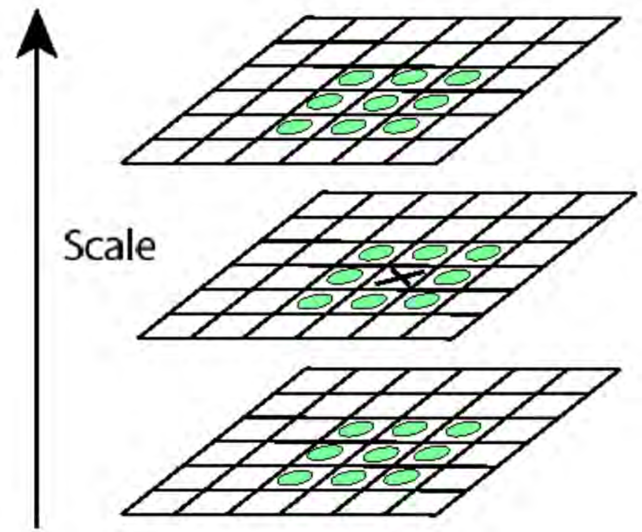
Difference-of-Gaussian(DoG)

- **Difference of Gaussians as approximation of the LoG**
  - This is used e.g. in Lowe's SIFT pipeline for feature detection.
- **Advantages**
  - No need to compute 2nd derivatives
  - Gaussians are computed anyway, e.g. in a Gaussian pyramid.



Keypoint localization with DoG

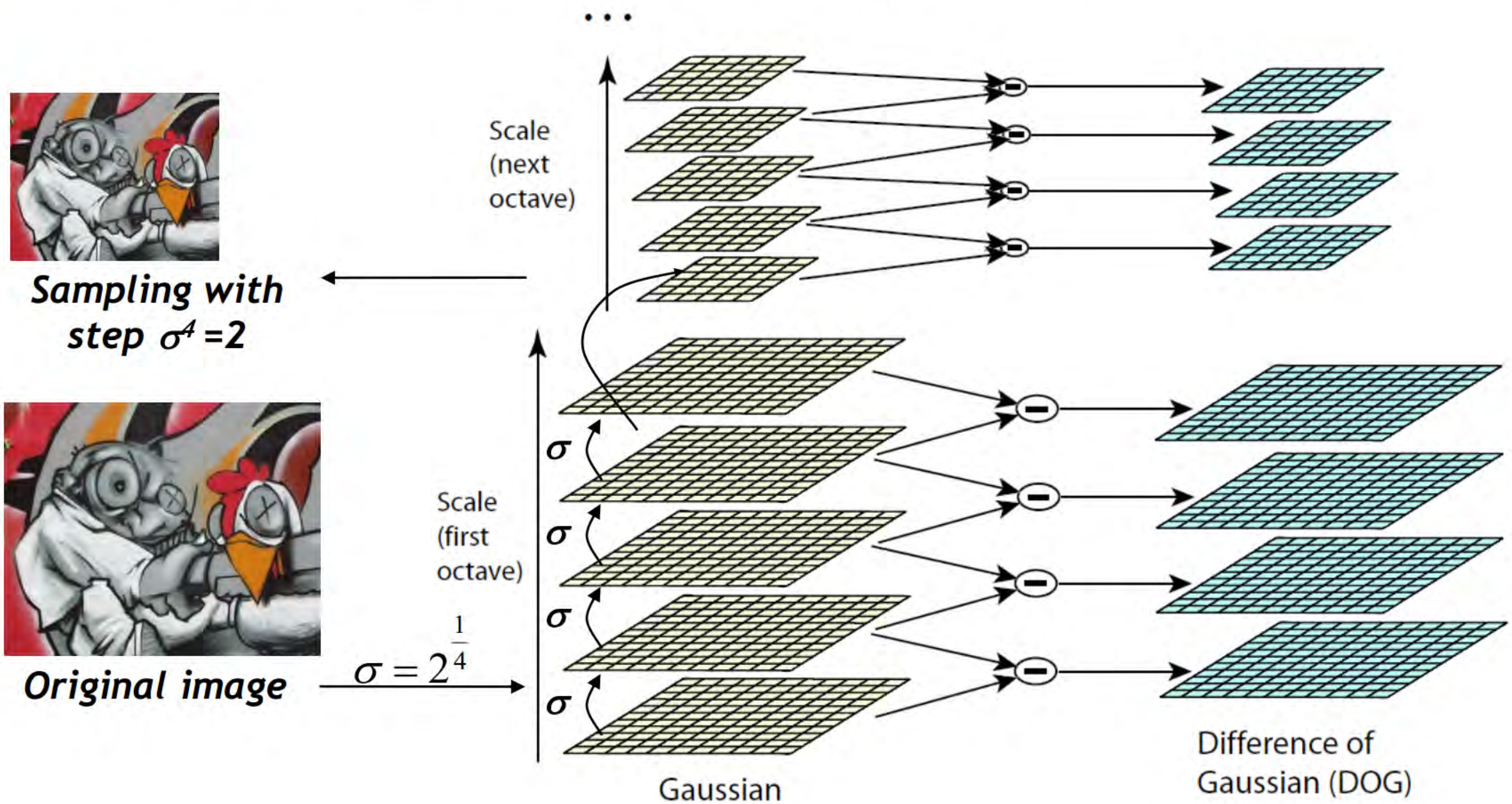
- **Detect maxima of difference-of-Gaussian (DoG) in scale space**
- **Then reject points with low contrast (threshold)**
- **Eliminate edge responses**



Candidate keypoints: list of (x,y,σ)

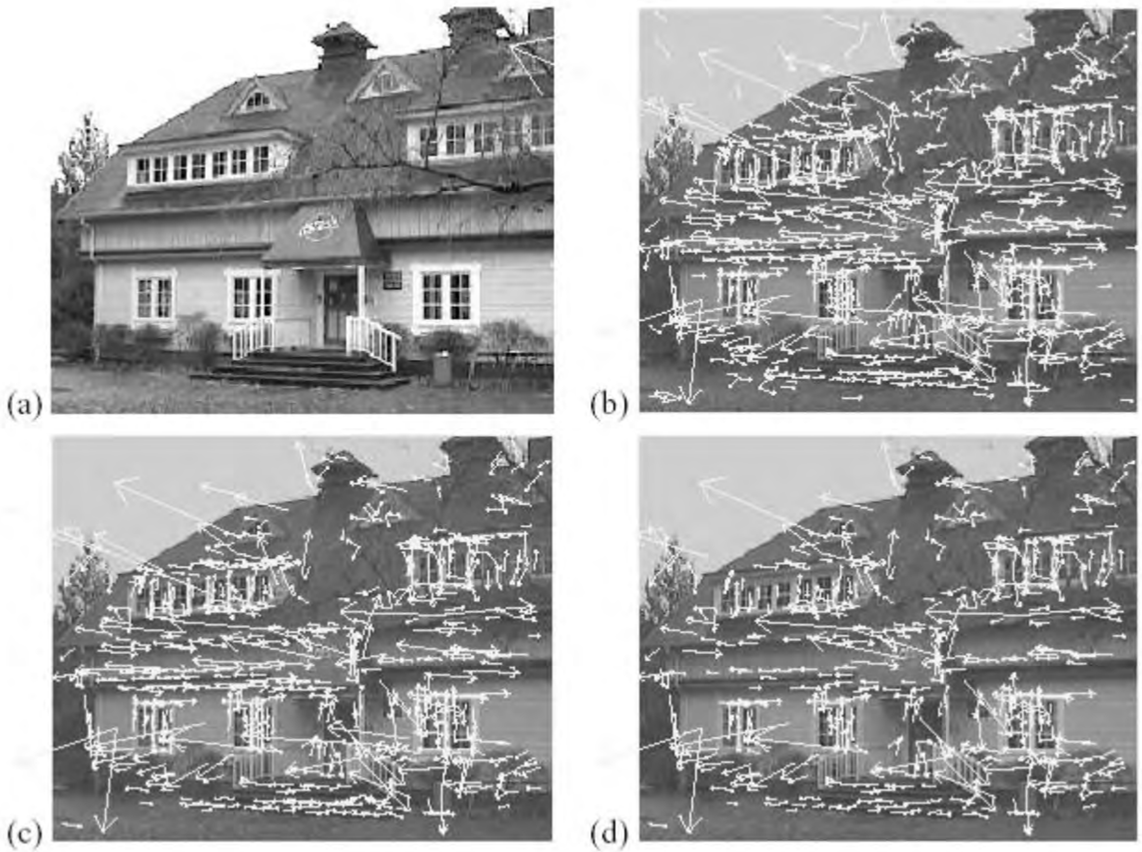
DoG: Efficient implementation

- **Computation in Gaussian scale pyramid**



Results: Lowe's DoG

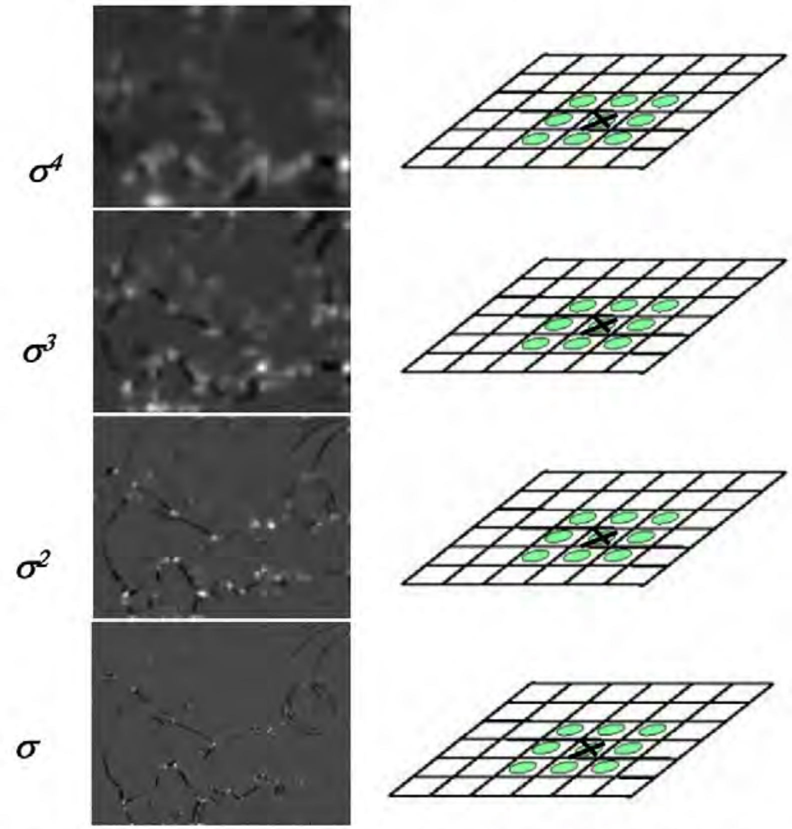
Example of Keypoint Detection



- (a) 233x189 image
- (b) 832 DoG extrema
- (c) 729 left after peak value threshold
- (d) 536 left after testing ratio of principle curvatures (removing edge responses)

Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection



Computing Harris function   Detecting local maxima

Slide adapted from Krystian Mikolajczyk

Harris-Laplace [Mikolajczyk '01]

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

Harris points

Harris-Laplace points

Summary: Scale Invariant Detection

- **Given:** Two images of the same scene with a large *scale difference* between them.
- **Goal:** Find *the same* interest points *independently* in each image.
- **Solution:** Search for *maxima* of suitable functions in *scale* and in *space* (over the image).

- Two strategies
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation

  - *These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).*

Summary

- Introduction to correspondence and alignment
- Overview of interest points
  - Matching pipeline
  - Repeatable & Distinctive
- Keypoint Localization
  - Harris detector
  - Hessian detector
- Scale invariant region selection
  - Automatic scale selection
  - Laplacian of Gaussian (LoG) & Difference of Gaussian (DoG)
  - Combinations: Harris-Laplacian & Hessian-Laplacian

You Can Try It at Home

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

Affine Covariant Features

Collaborative work between the Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception.

Affine Covariant Region Detectors

Detector output

Input image

Region detector

format:

1.0

m

$u_1\ v_1\ a_1\ b_1\ c_1$

:

:

$u_m\ v_m\ a_m\ b_m\ c_m$

output example:
img1.haraff

display_features.m

Image with displayed regions

Parameters defining an affine region

`u,v,a,b,c` in `a(x-u)(x-u)+2b(x-u)(y-v)+c(y-v)(y-v)=1`

with `(0,0)` at image top left corner

Code

- provided by the authors, see publications for details and links to authors web sites.

Linux binaries	Example of use	Displaying r
Harris-Affine & Hessian-Affine	`prompt>./h_affine.ln -haraff -i img1.ppm -o img1.haraff -thres 1000`	`matlab>> d`
	`prompt>./h_affine.ln -hesaff -i img1.ppm -o img1.hesaff -thres 500`	`matlab>> d`
MSER - Maximaly stable extremal regions (also Windows)	`prompt>./mser.ln -t 2 -es 2 -i img1.ppm -o img1.mser`	`matlab>> d`
IBR - Intensity extrema based detector	`prompt>./ibr.ln img1.ppm img1.ibr -scalefactor 1.0`	`matlab>> d`
EBR - Edge based detector	`prompt> ./ebr.ln img1.ppm img1.ebr`	`matlab>> d`
Salient region detector	`prompt>./salient.ln img1.ppm img1.sal`	`matlab>> d`

http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries

References

- Read David Lowe's SIFT paper
  - D. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60(2), pp. 91-110, 2004
- Good survey paper on Int. Pt. detectors and descriptors
  - T. Tuytelaars, K. Mikolajczyk, *Local Invariant Feature Detectors: A Survey*, Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 3, pp 177-280, 2008.
- Try the example code, binaries, and Matlab wrappers
  - Good starting point: Oxford interest point page http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries