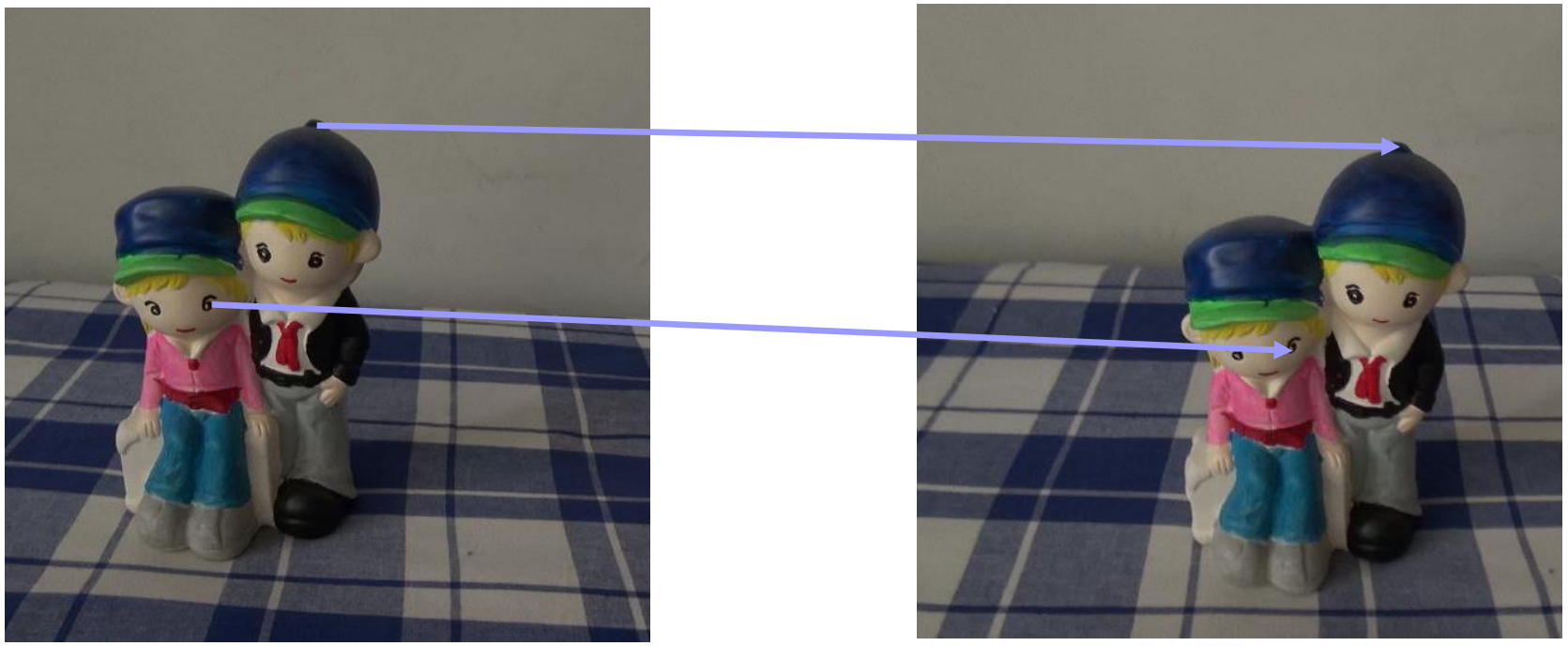
计算机视觉
Computer Vision

-- Matching 1

钟 凡
zhongfan@sdu.edu.cn

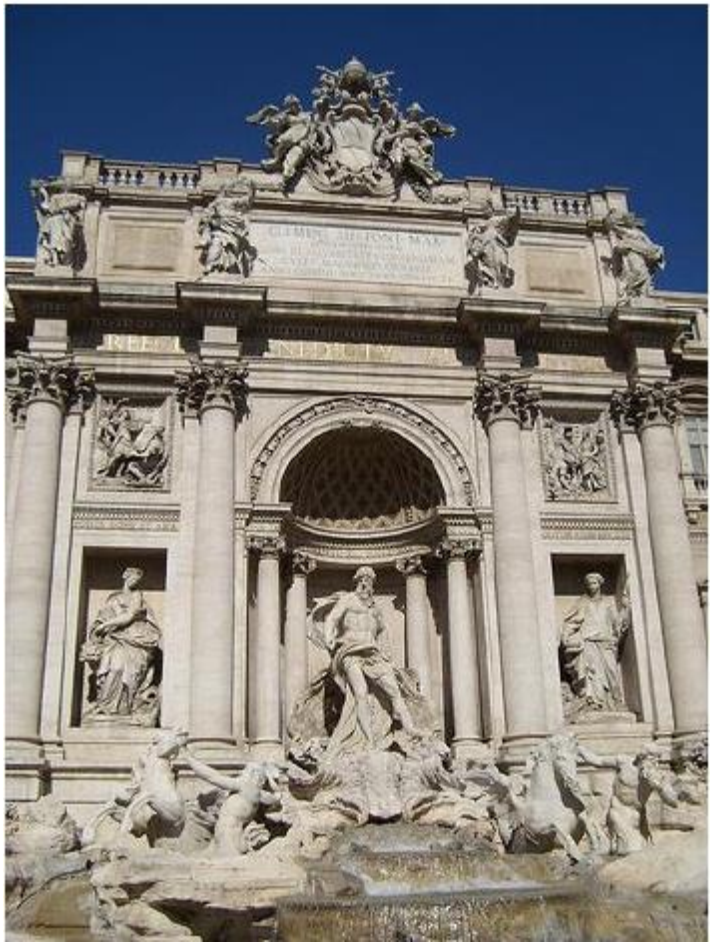
图像匹配

同一场景点在不同图像之间像素的对应关系



同一视频中两帧

图像匹配



不同人拍摄的同一场景

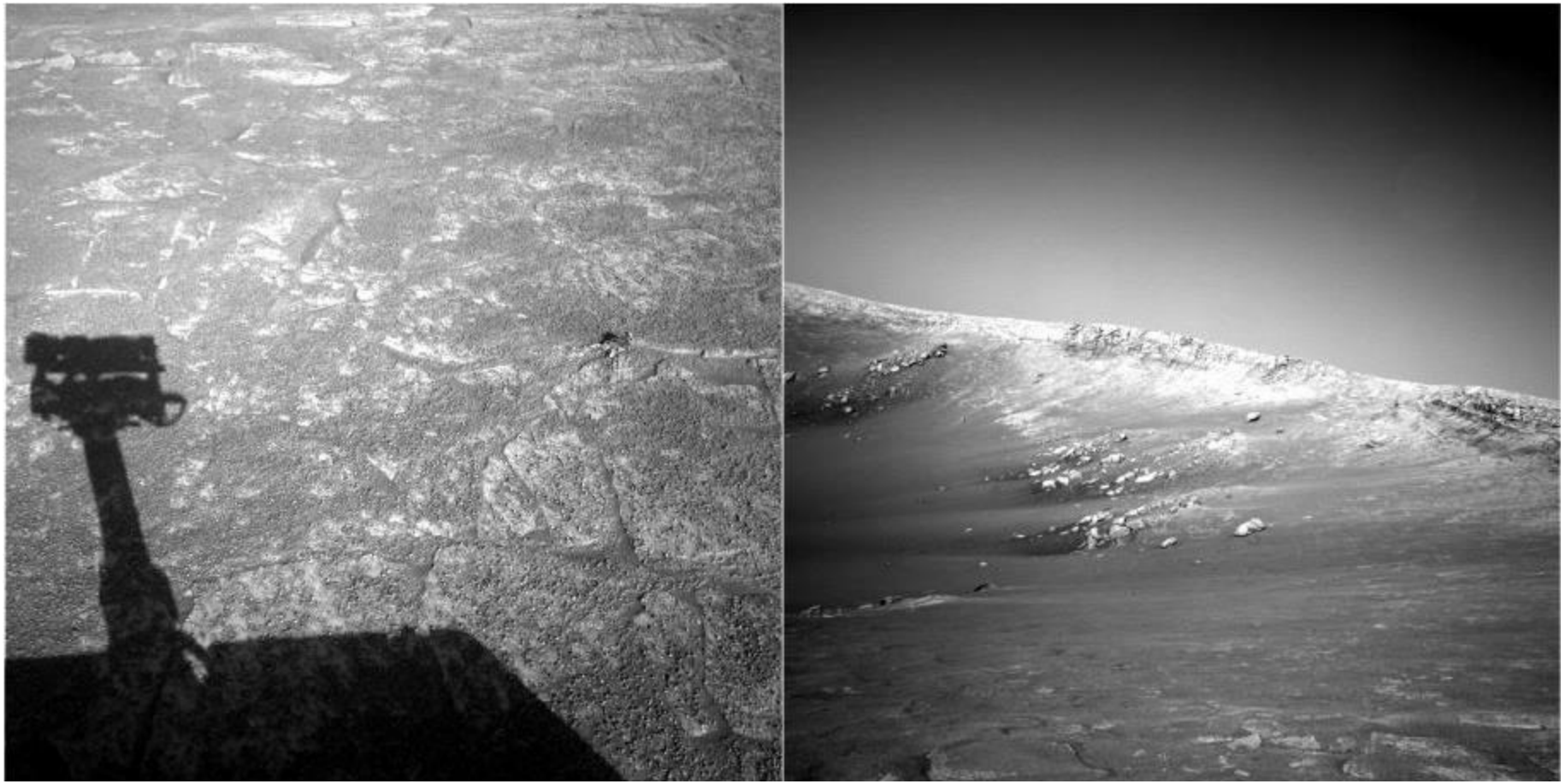
图像匹配

同一场景的不同角度

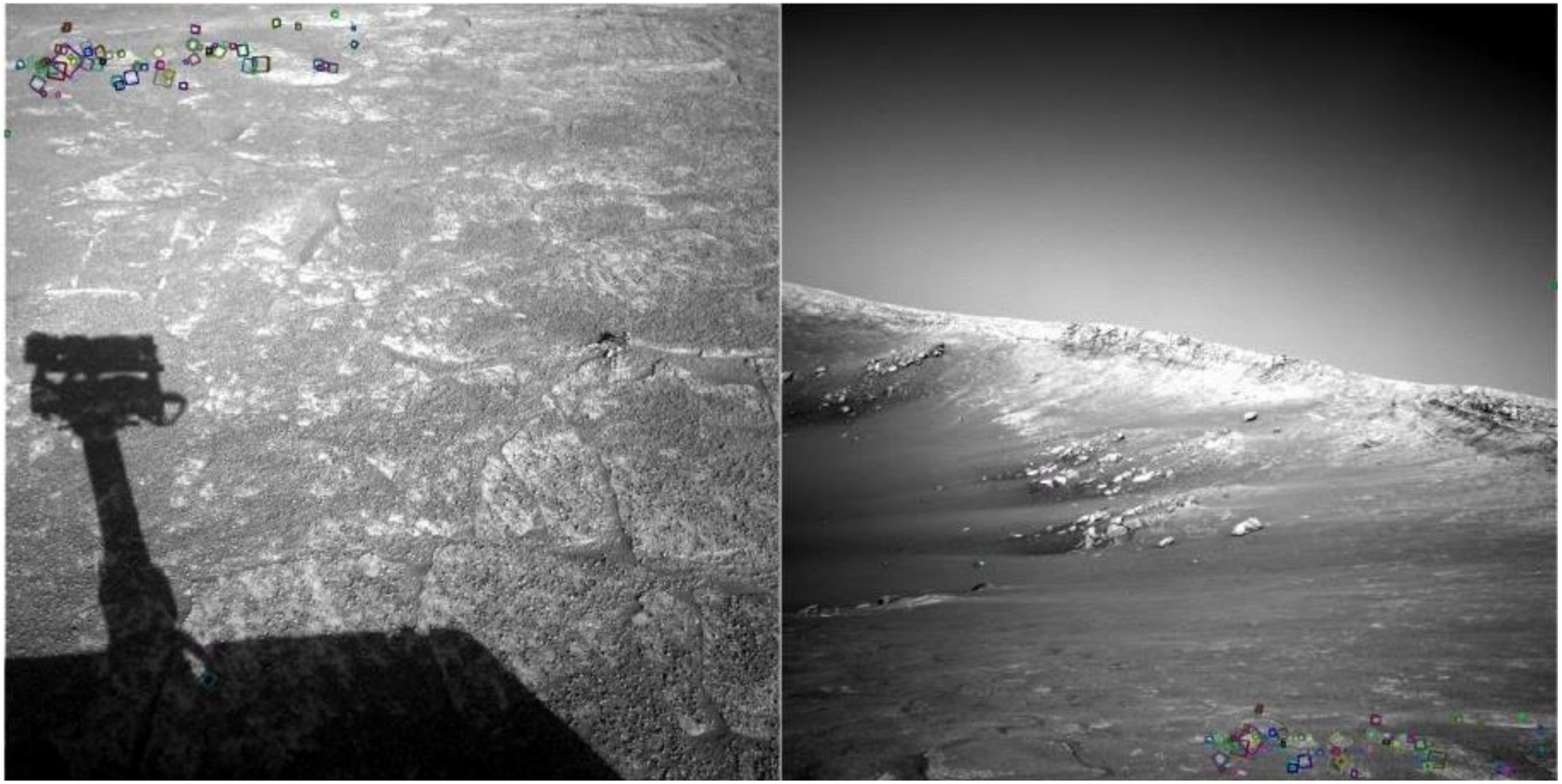
图像匹配

同一物体，不同场景

??



!!

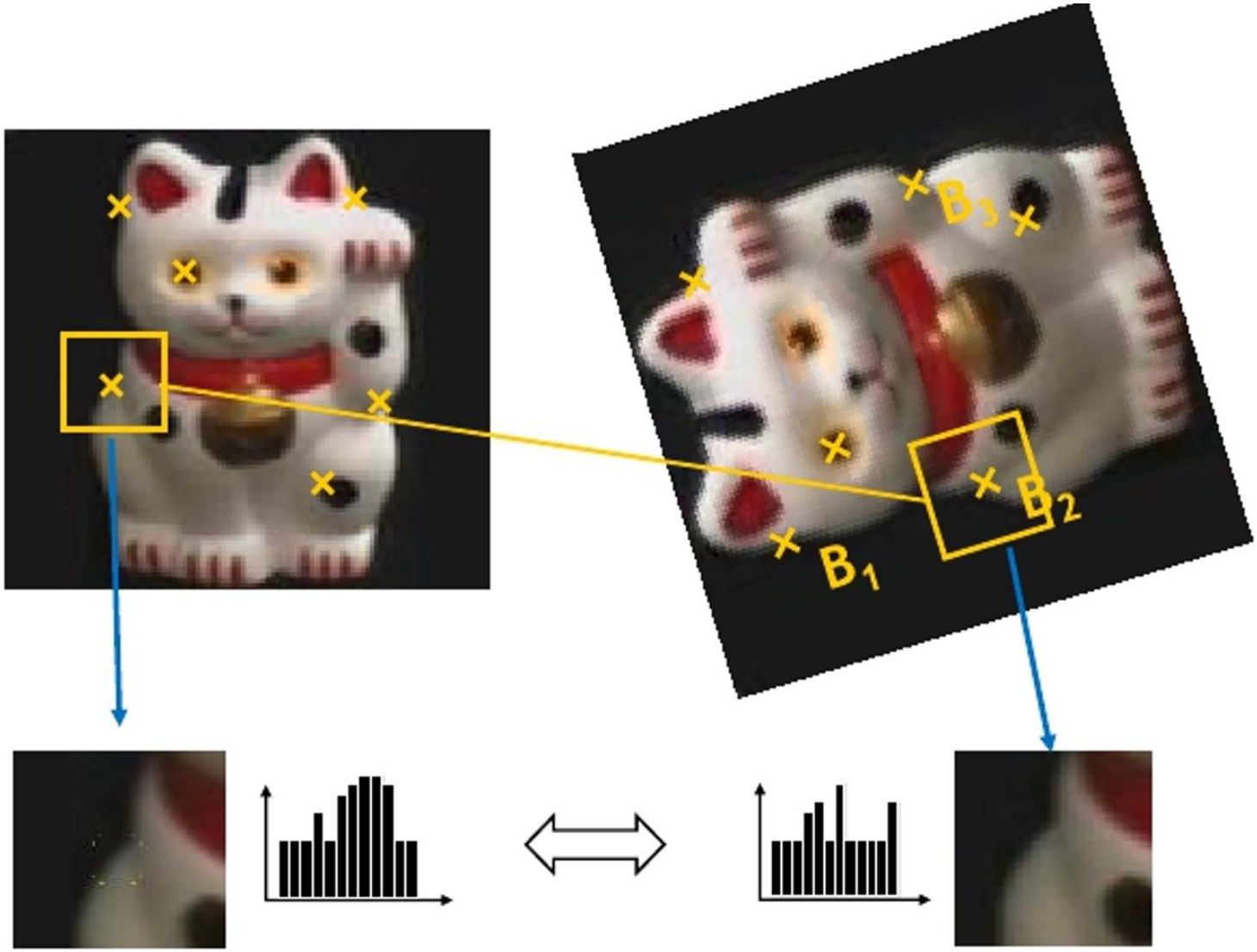


意义

- 视频处理与分析的基础
- 视觉三维重建的基础
- 早期物体识别、图像检索技术的基础
- ……
- 计算机视觉半壁江山的基础

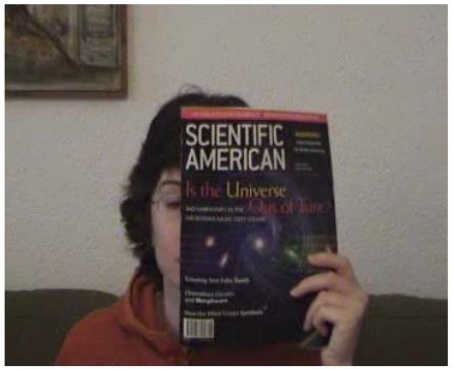
你会怎么办?

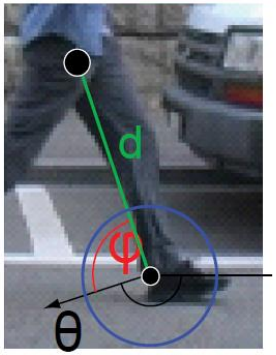
局部特征

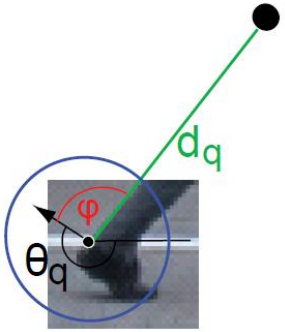

B3
B2
B1

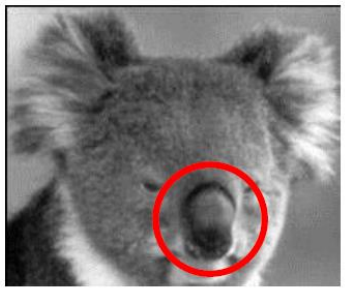
为什么要“局部”？

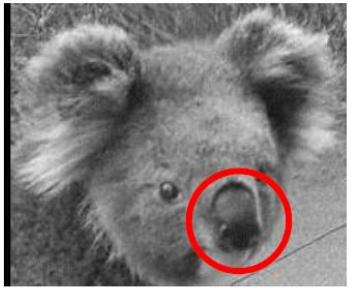
- 全局特征面临以下情况时会面临难以克服的困难：
  - 遮挡
  - 形变
  - 环境变化



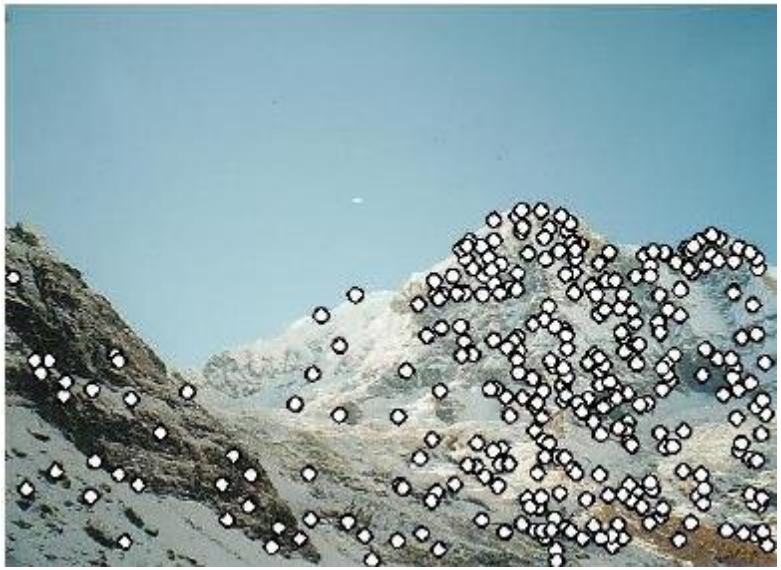


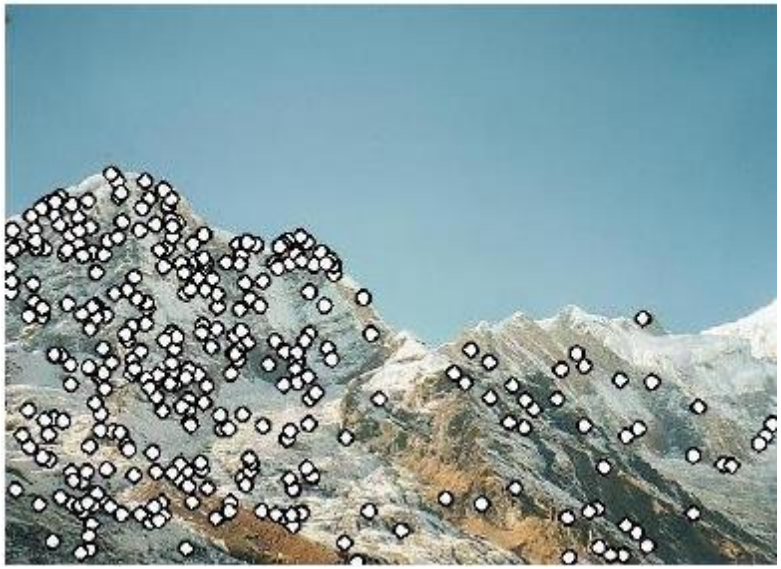




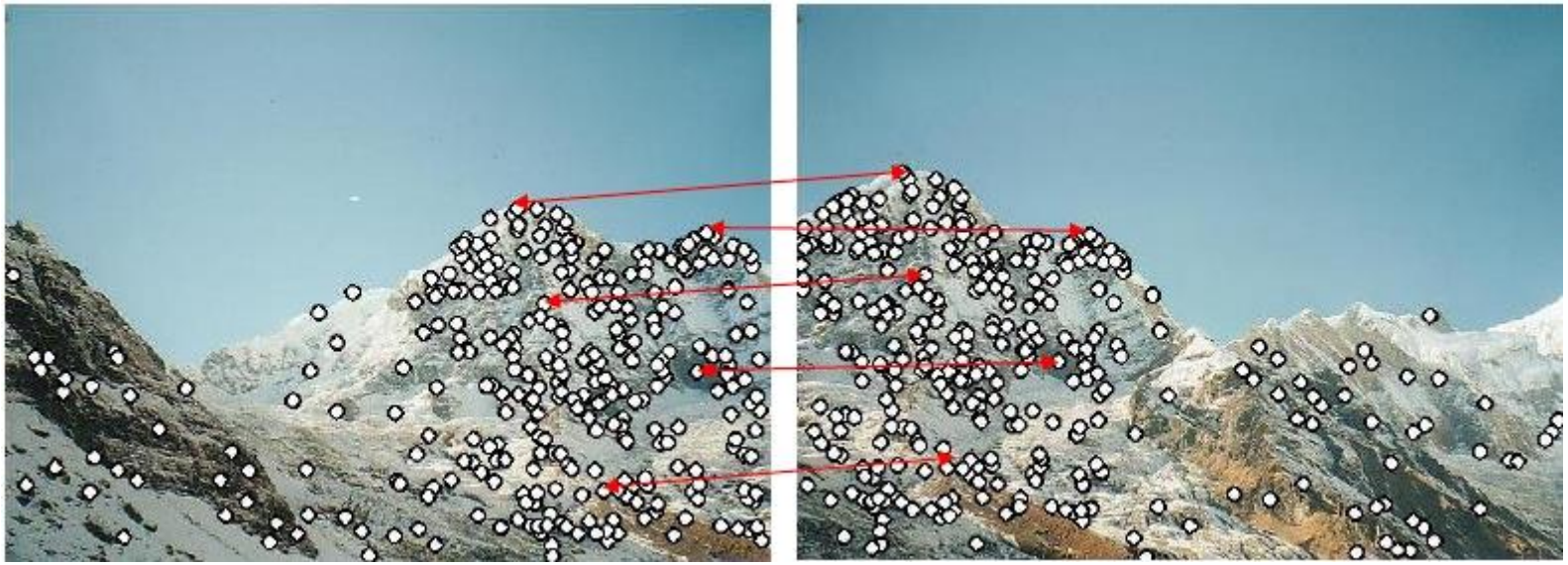


1. 特征检测





2. 特征匹配



3. 运动估计

特征检测

特征点

- 什么样的点才应该是特征点?

特征点

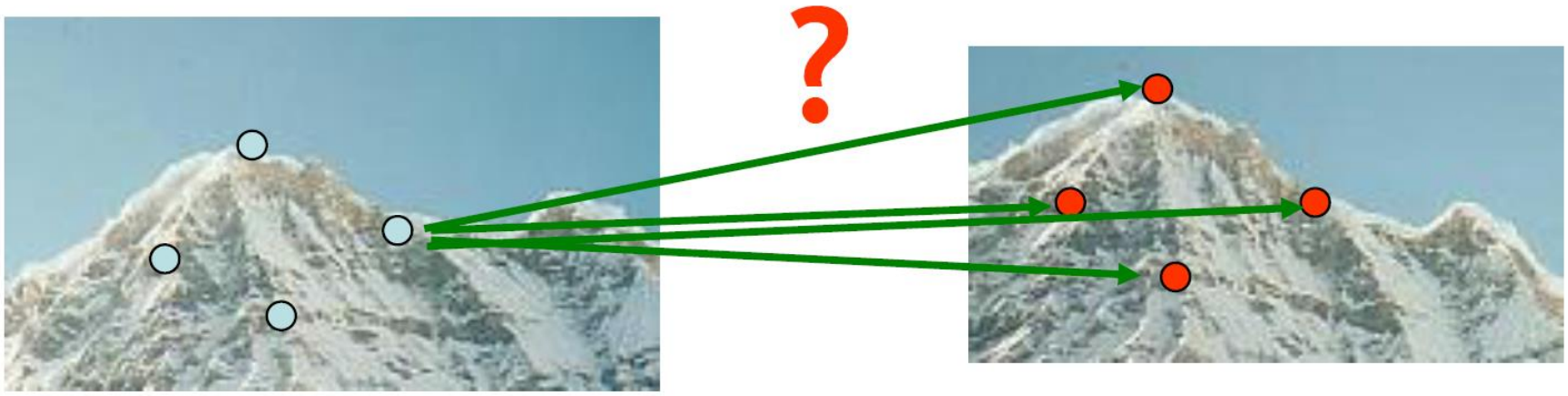
- 目标1：稳定检测

No chance to match!

We need a repeatable detector!

特征点

- 目标2：易于匹配

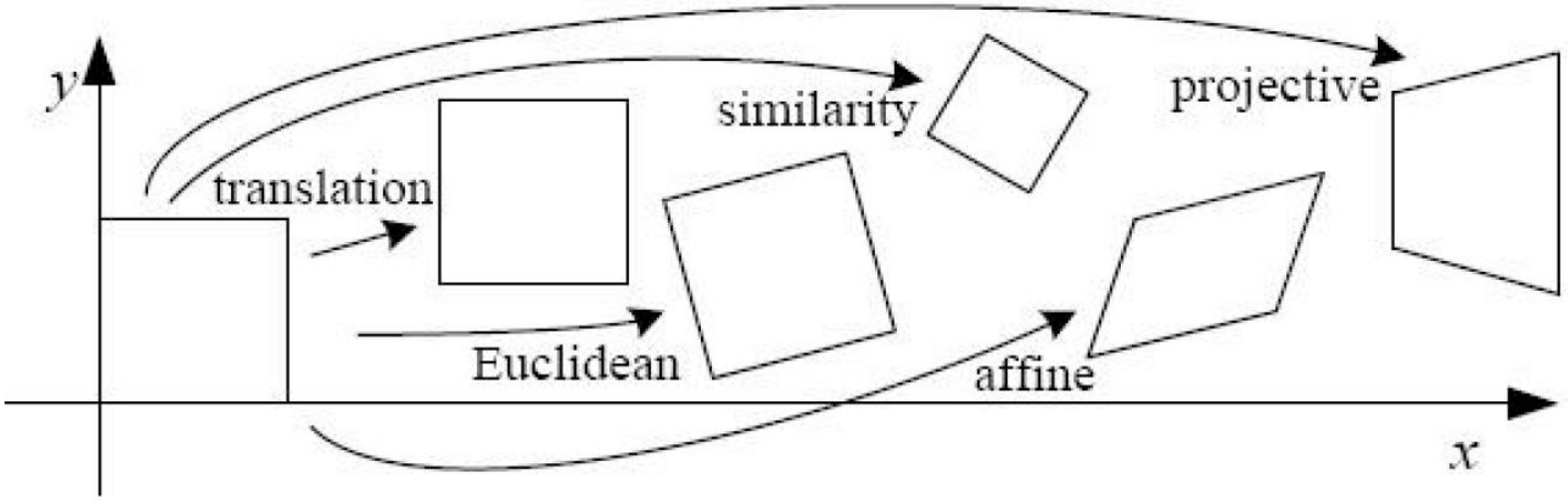


特征检测与匹配

- 挑战1：对图像几何变换的稳定性

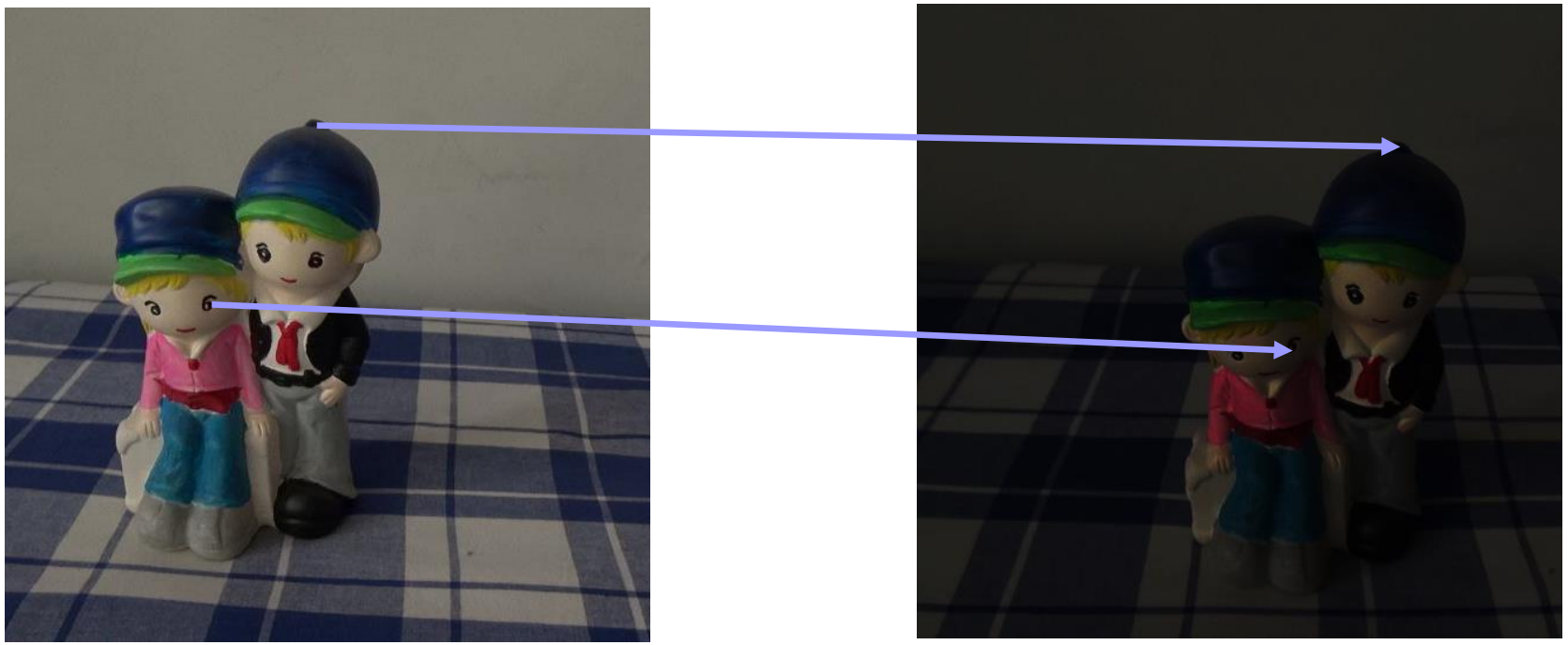
特征检测与匹配

- 挑战1：对图像几何变换的稳定性



特征检测与匹配

- 挑战2：对颜色、光照变化的稳定性

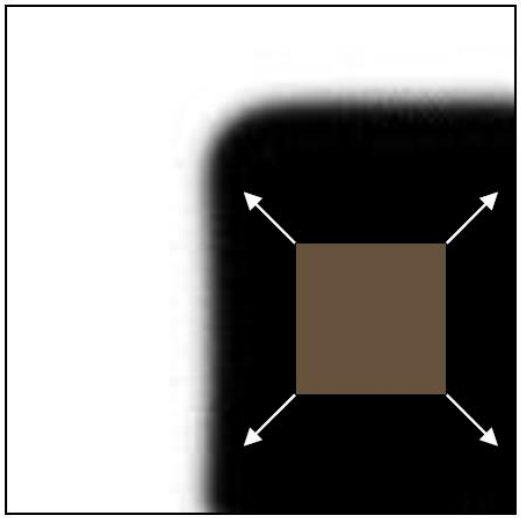


Requirements

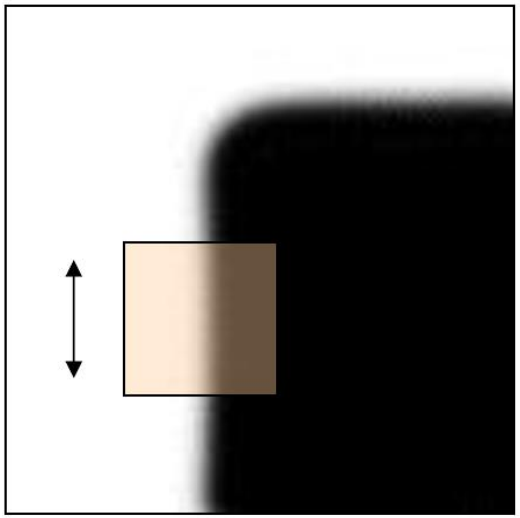
- Region extraction needs to be **repeatable** and **accurate**
  - **Invariant** to translation, rotation, scale changes
  - **Robust** or **covariant** to out-of-plane (≈affine) transformations
  - **Robust** to lighting variations, noise, blur, quantization
- **Locality**: Features are local, therefore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

角点检测（Corner Detection)

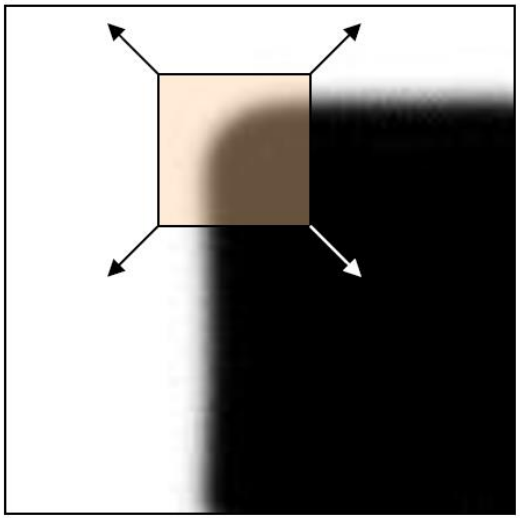
- 角点



“flat” region:
no change in all directions



“edge”:
no change along the edge direction

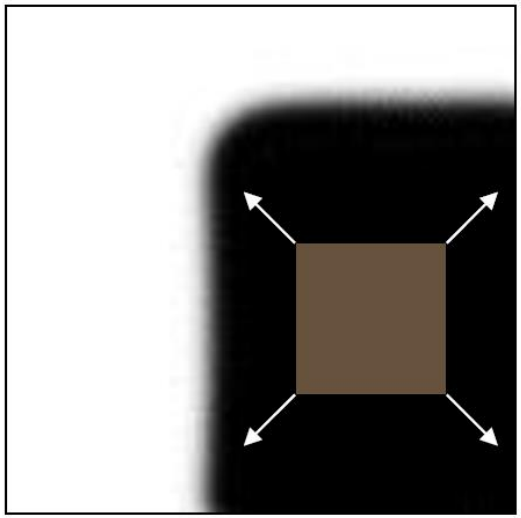


“corner”:
significant change in all directions

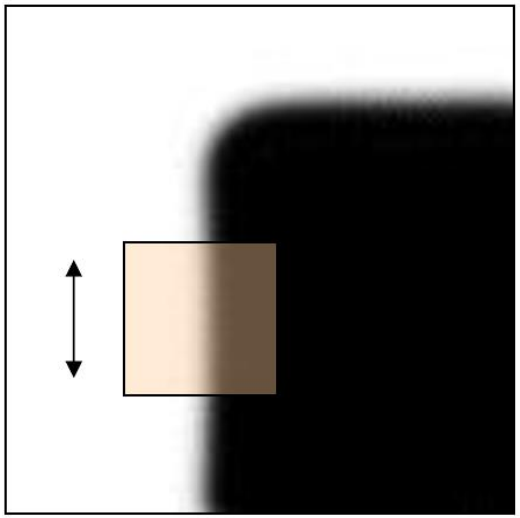
角点检测（Corner Detection)

- 角点

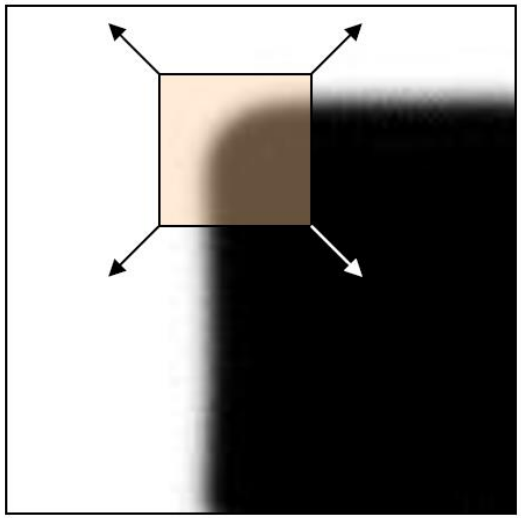
如何区分三种类型的点?



平坦



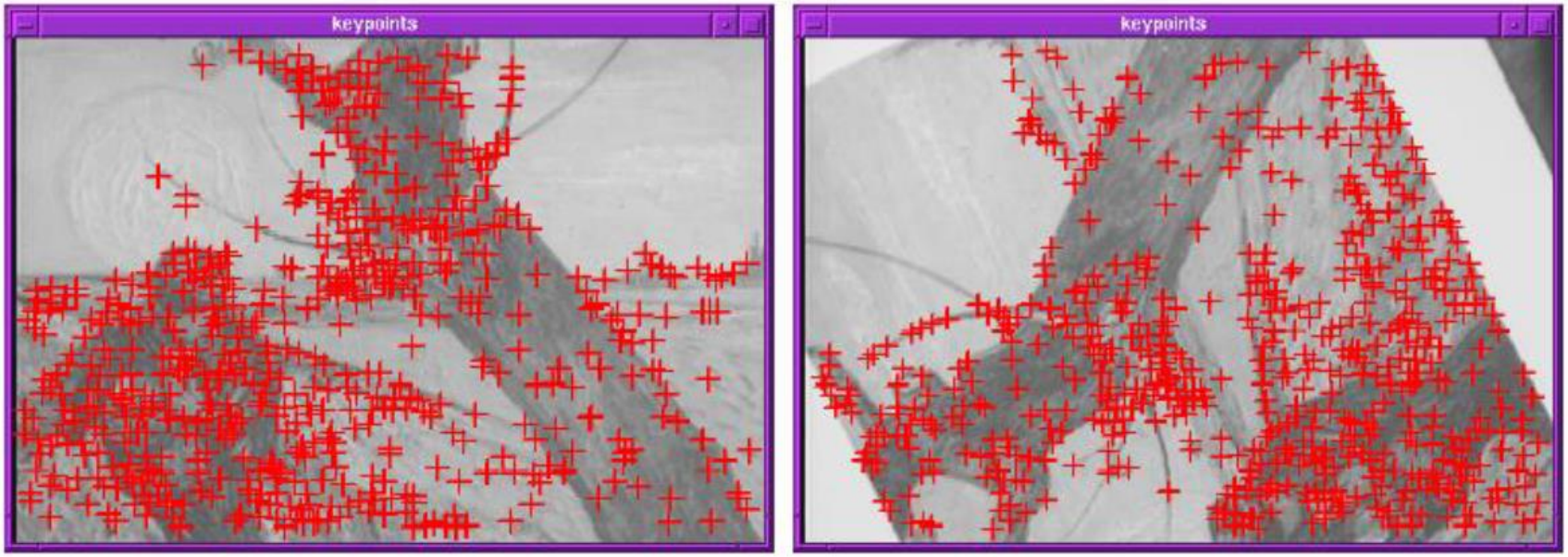
边



角点

Harris

- 在角点处，沿任意方向运动都会引起像素颜色的明显变化
- 等价于：在角点附近，图像梯度具有至少两个主方向



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

Harris Detector Formulation

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

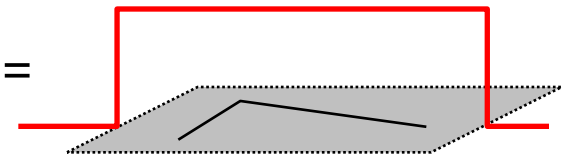
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

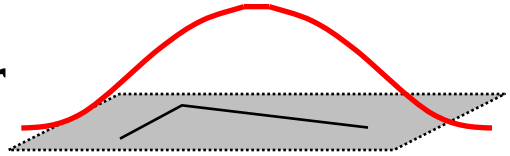
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector Formulation

- This measure of change can be approximated by:

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

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

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

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

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

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \;\; I_y]$$

Slide credit: Rick Szeliski

Harris Detector Formulation

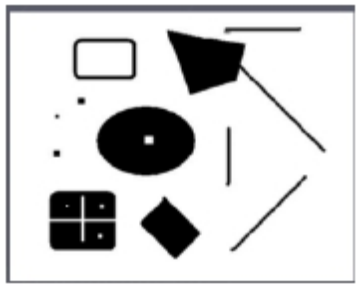
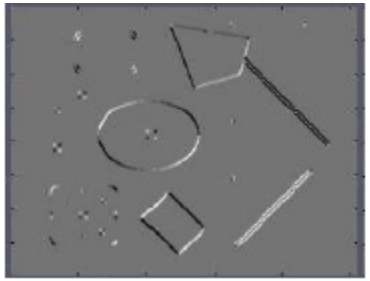


Image I

I_x

I_y



$I_x I_y$

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

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

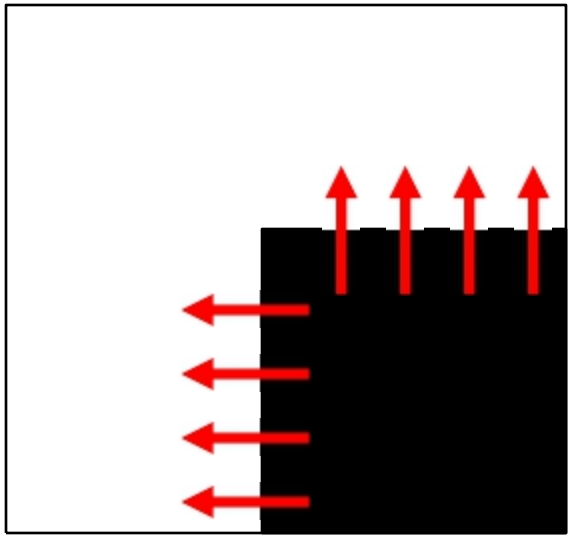
Gradient with respect to x, times gradient with respect to y

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

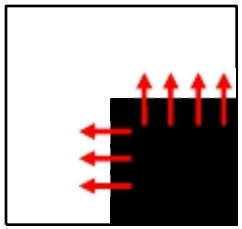
$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y]$$

What Does This Matrix Reveal?

- First, let's consider an axis-aligned corner:



What Does This Matrix Reveal?



- First, let's consider an axis-aligned corner:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

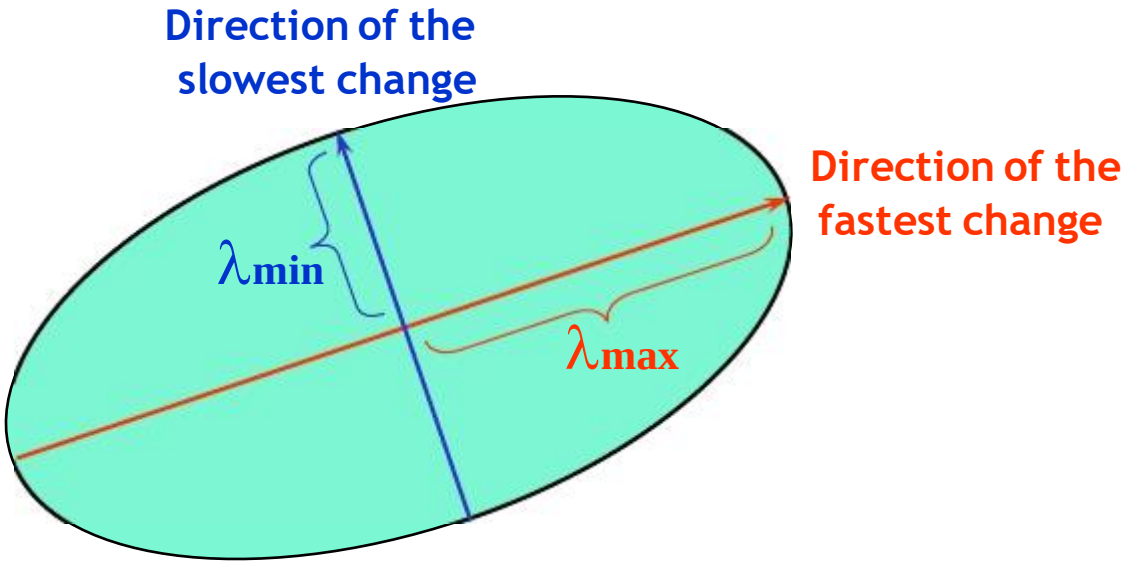
- This means:
  - Dominant gradient directions align with x or y axis
  - If either λ is close to 0, then this is not a corner, so look for locations where both are large.
- What if we have a corner that is not aligned with the image axes?

General Case

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

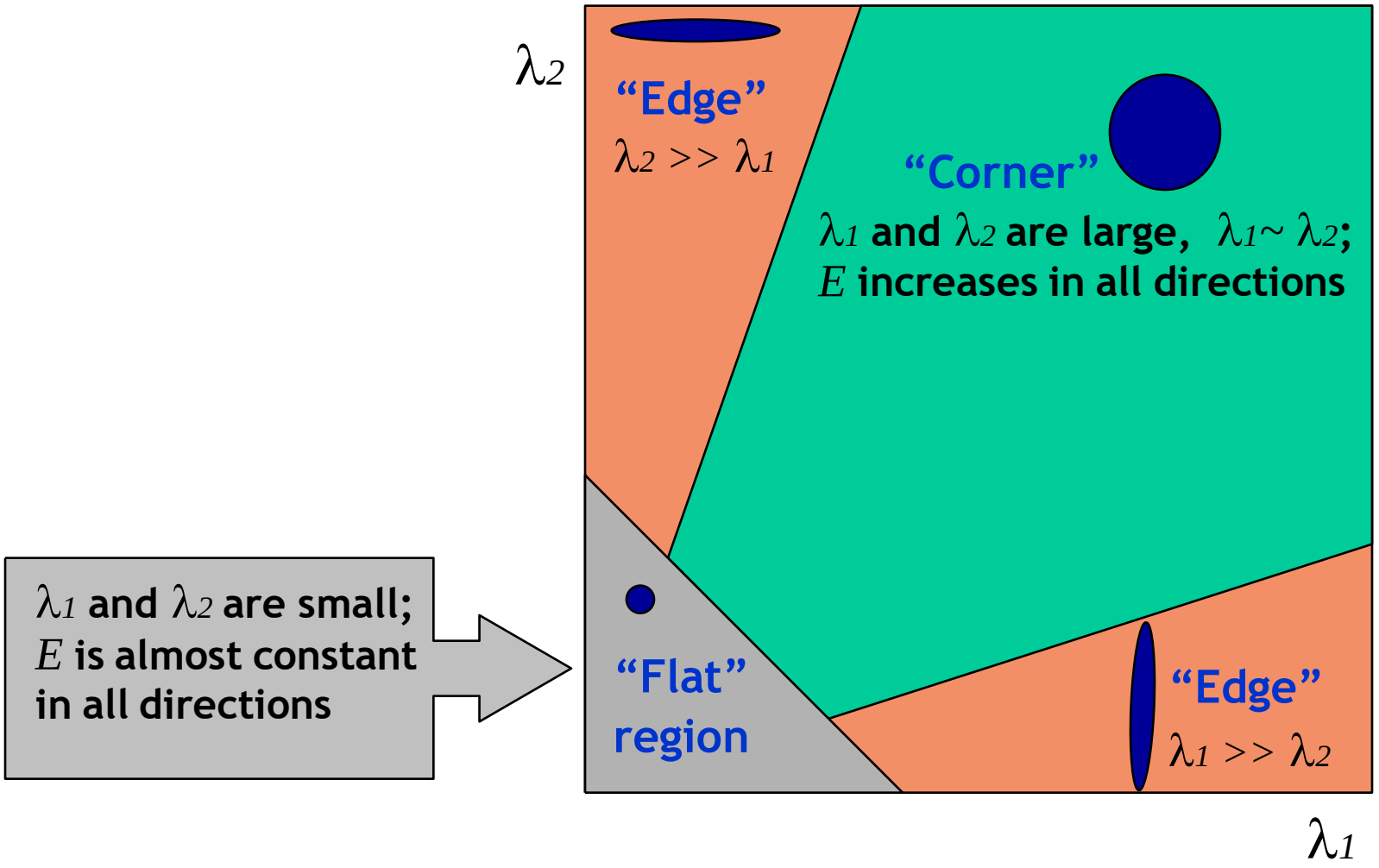
(Eigenvalue decomposition)

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



Interpreting the Eigenvalues

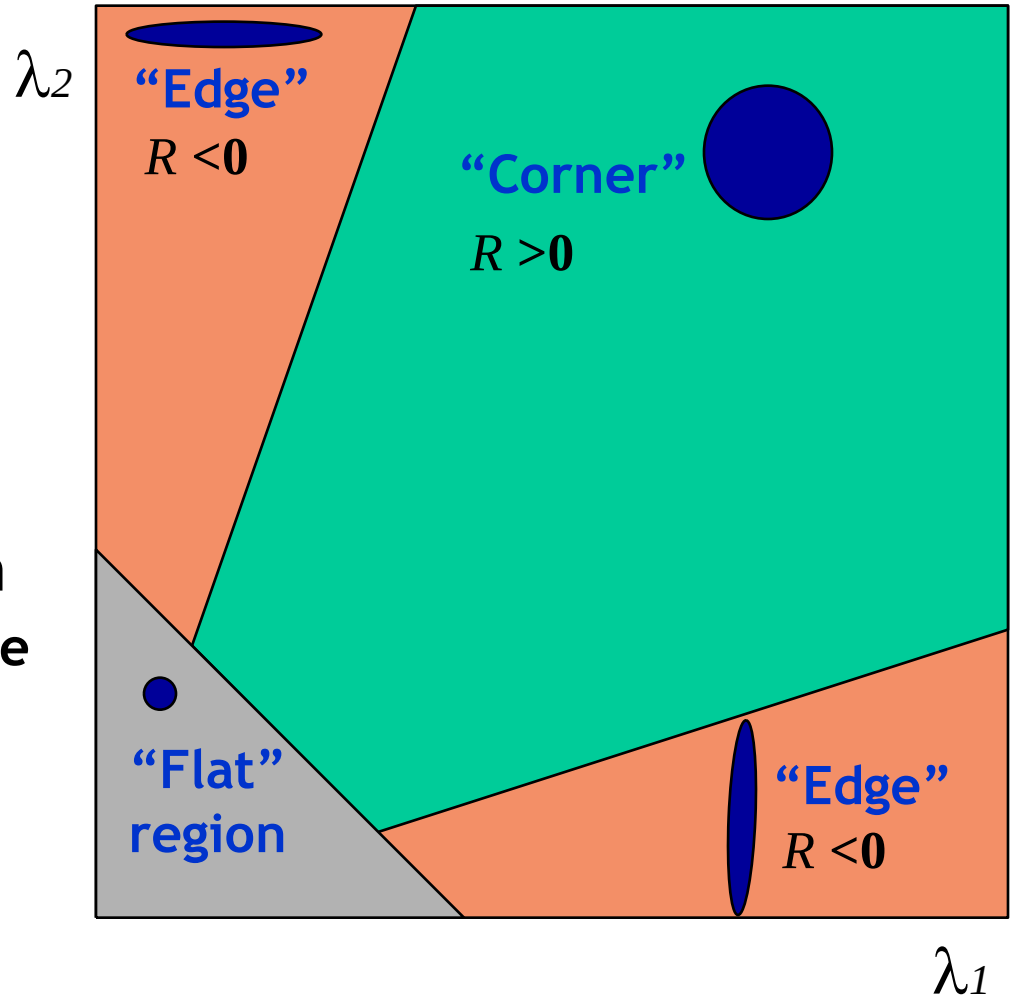
- Classification of image points using eigenvalues of M:



Slide credit: Kristen Grauman

Corner Response Function

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



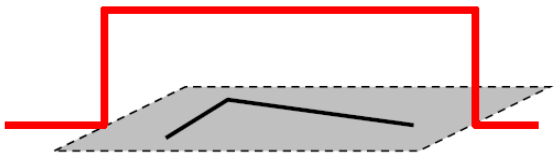
- **Fast approximation**
  - Avoid computing the eigenvalues
  - α: constant (0.04 to 0.06)

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

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

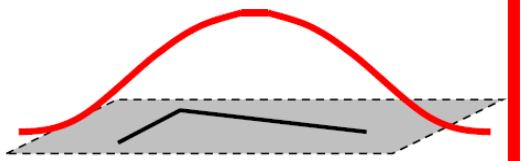


1 in window, 0 outside

  - Problem: not rotation invariant

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

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



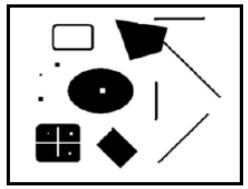
Gaussian

  - Result is rotation invariant

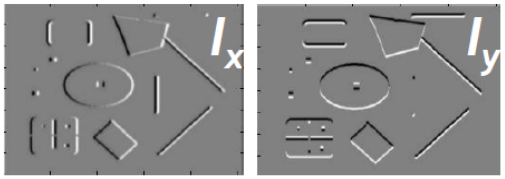
Summary: Harris Detector [Harris88]

- **Compute second moment matrix (autocorrelation matrix)**

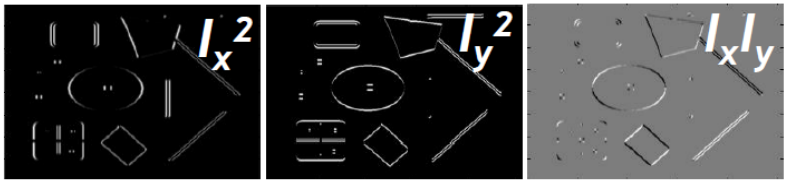
$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

4. **Cornerness function - two strong eigenvalues**

$$R = \det[M(\sigma_I,\sigma_D)] - \alpha[\text{trace}(M(\sigma_I,\sigma_D))]^2$$
$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

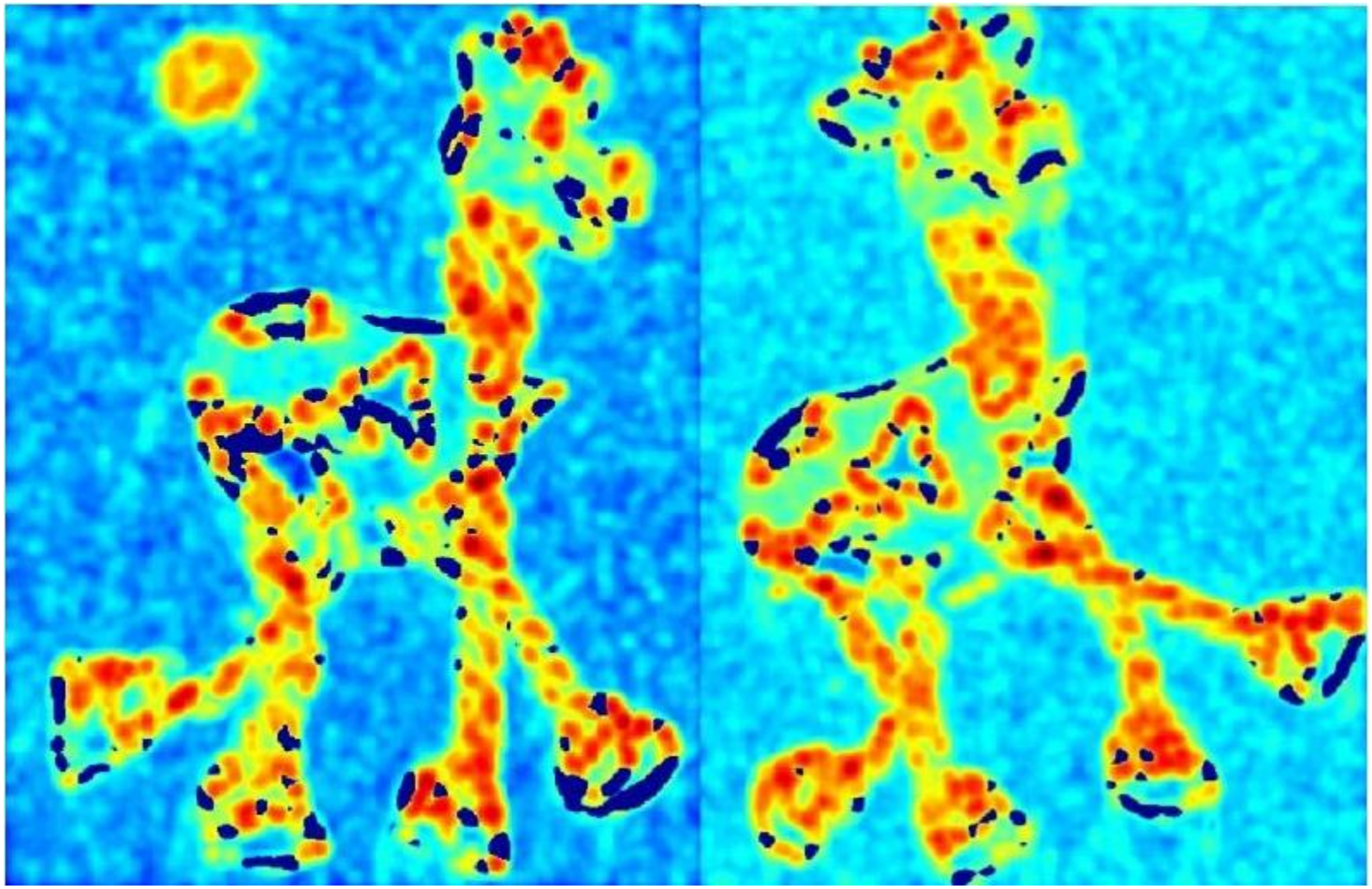
5. **Perform non-maximum suppression**

Slide credit: Krystian Mikolajczyk

Harris Detector: Workflow

Slide adapted from Darya Frolova, Denis Simakov

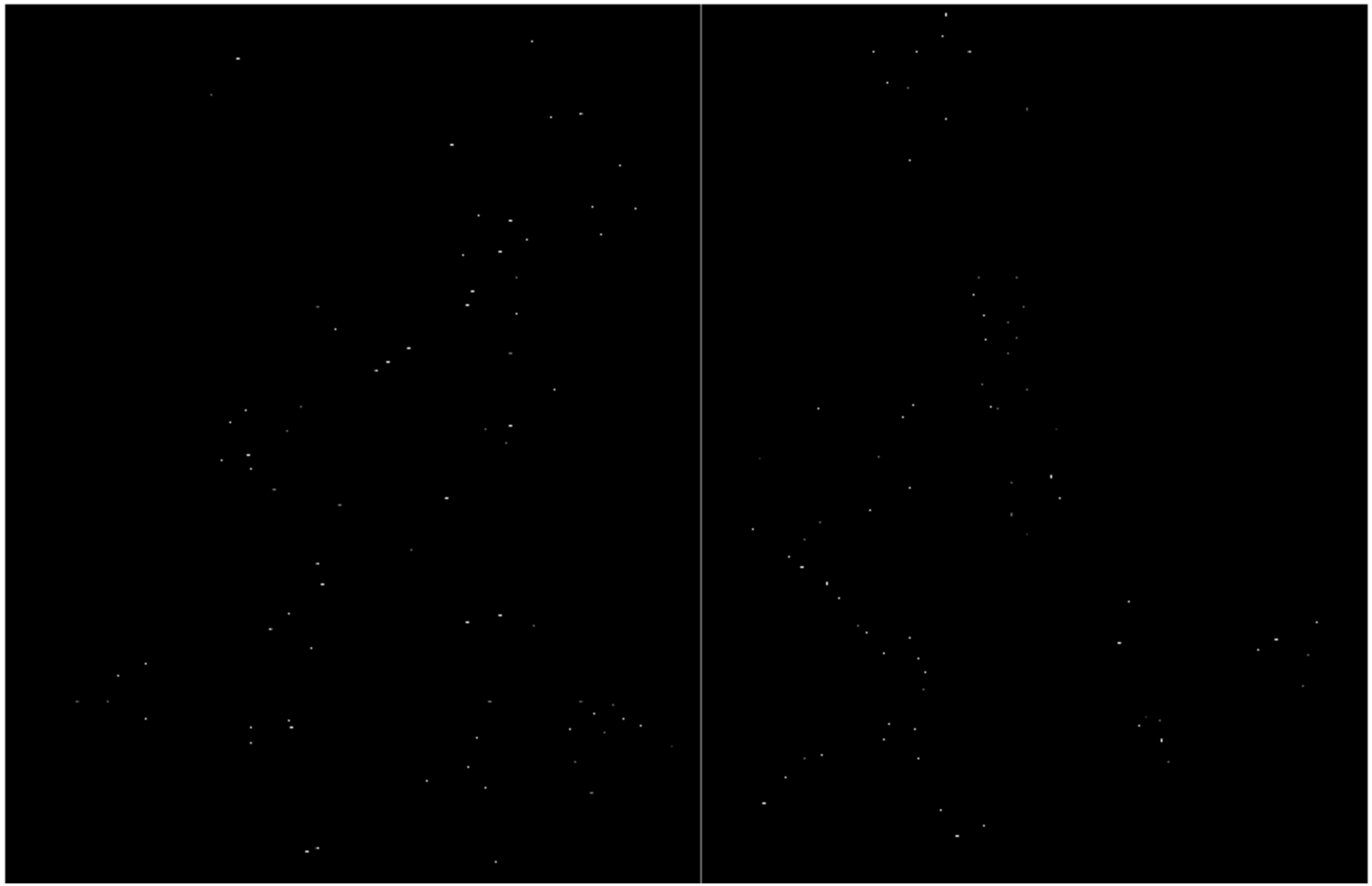
Harris Detector: Workflow



- **Compute corner responses *R***

Slide adapted from Darya Frolova, Denis Simakov

Harris Detector: Workflow



- Take only the local maxima of R, where R > threshold.

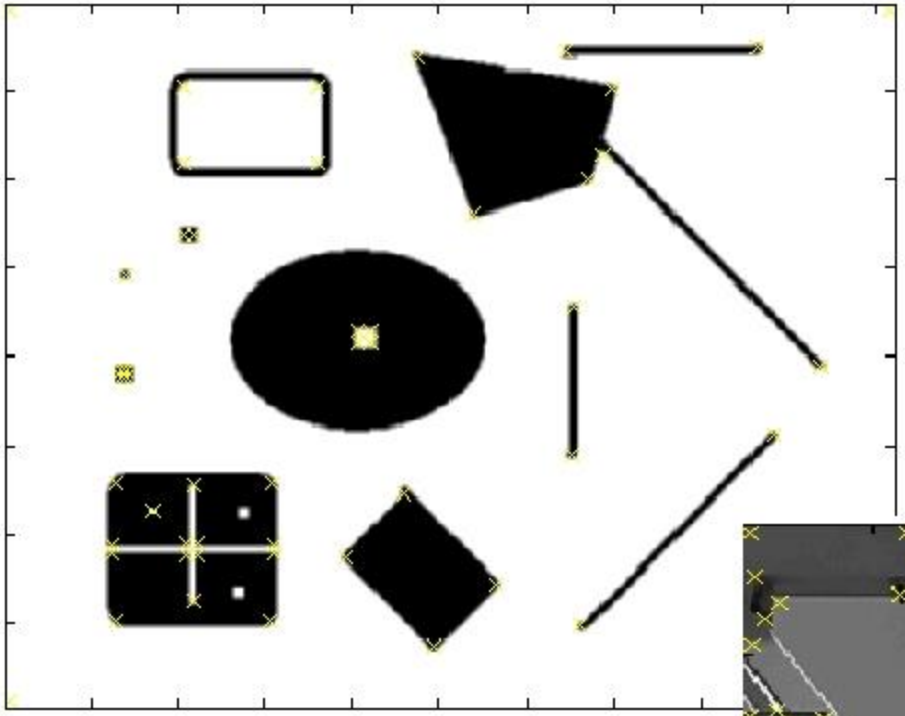
Slide adapted from Darya Frolova, Denis Simakov

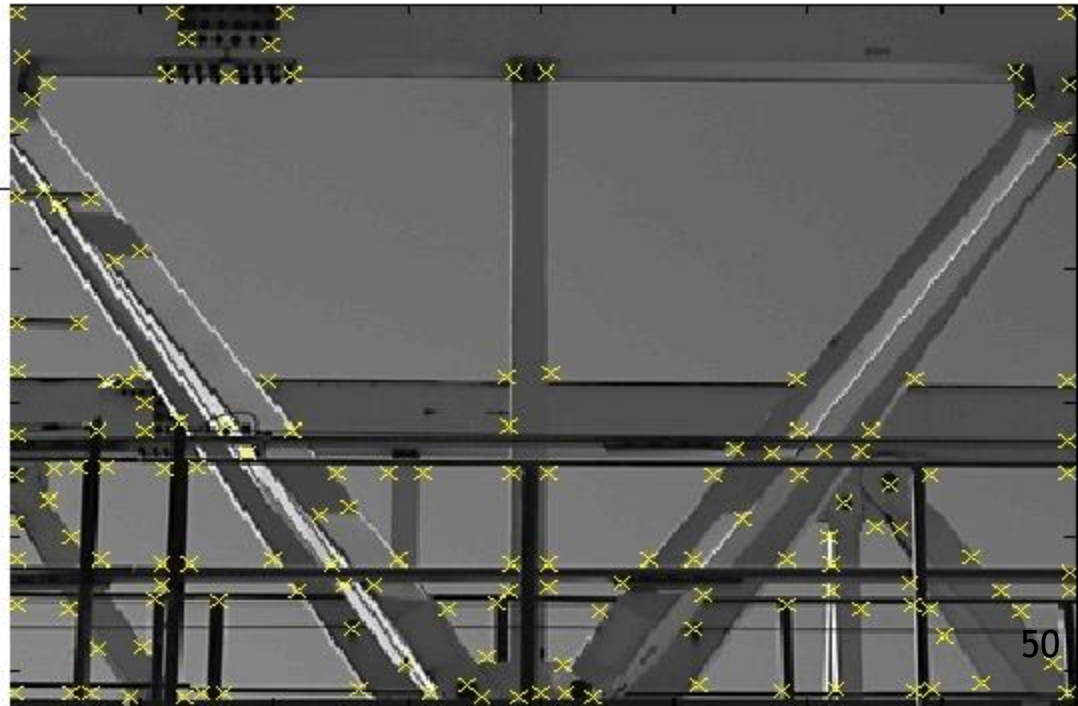
Harris Detector: Workflow

- **Resulting Harris points**

Slide adapted from Darya Frolova, Denis Simakov

Harris Detector – Responses [Harris88]





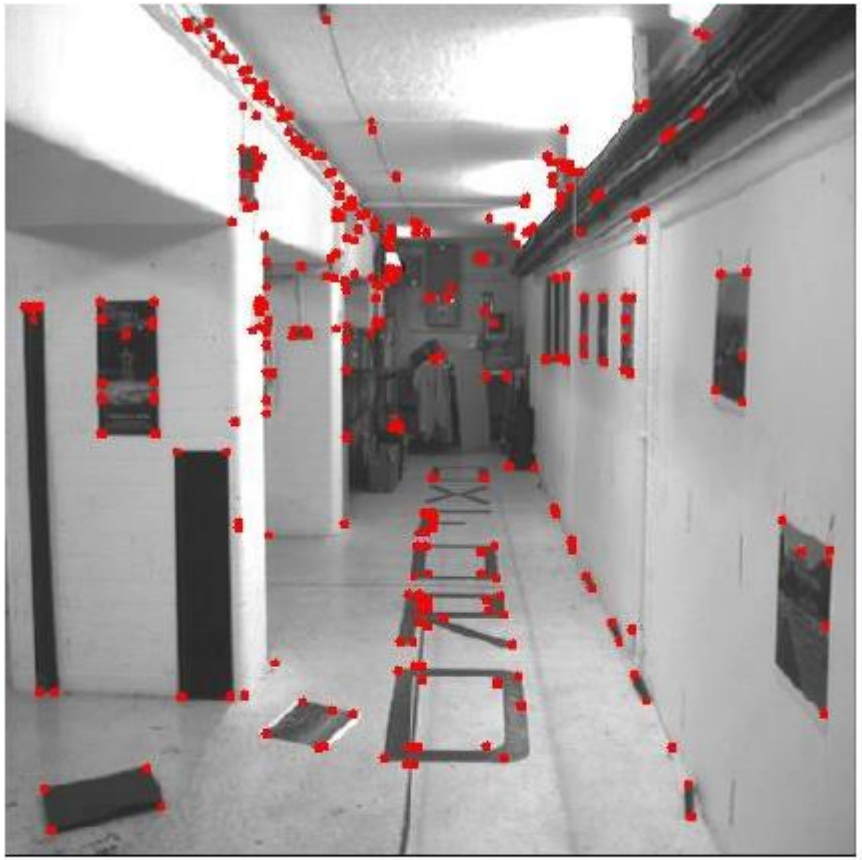
Effect: **A very precise corner detector.**

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]

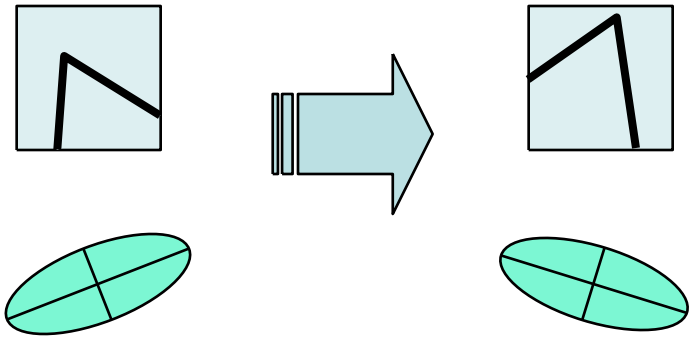


- **Results are well suited for finding stereo correspondences**

Slide credit: Kristen Grauman

Harris对光照变化和几何变换的稳定性?

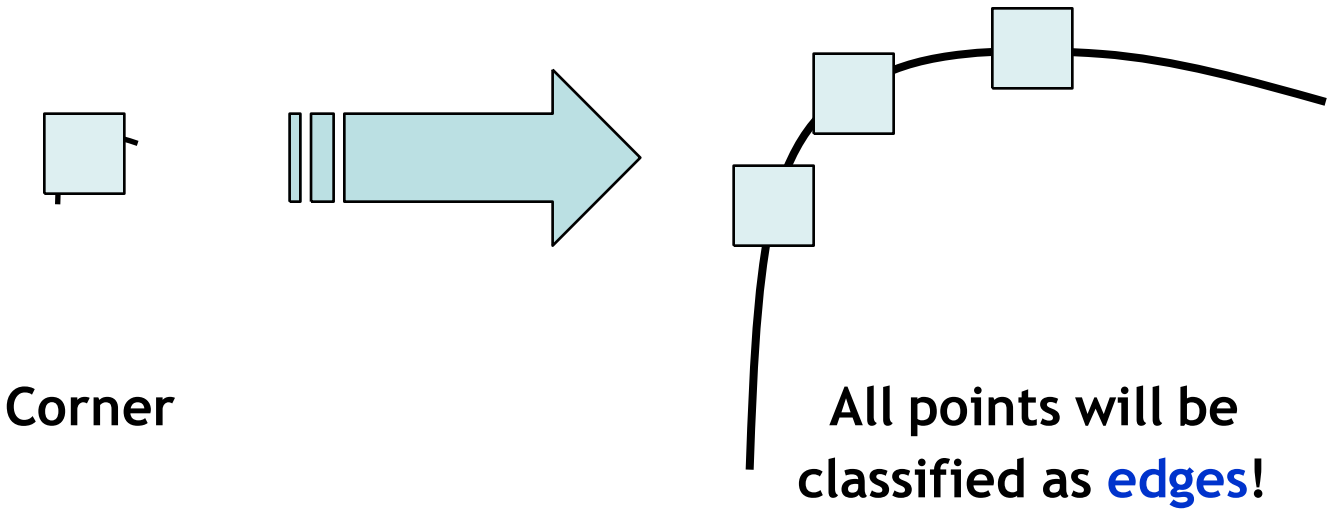
Rotation invariance?



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Scale invariance?



***Not* invariant to image scale!**

Illumination invariance?

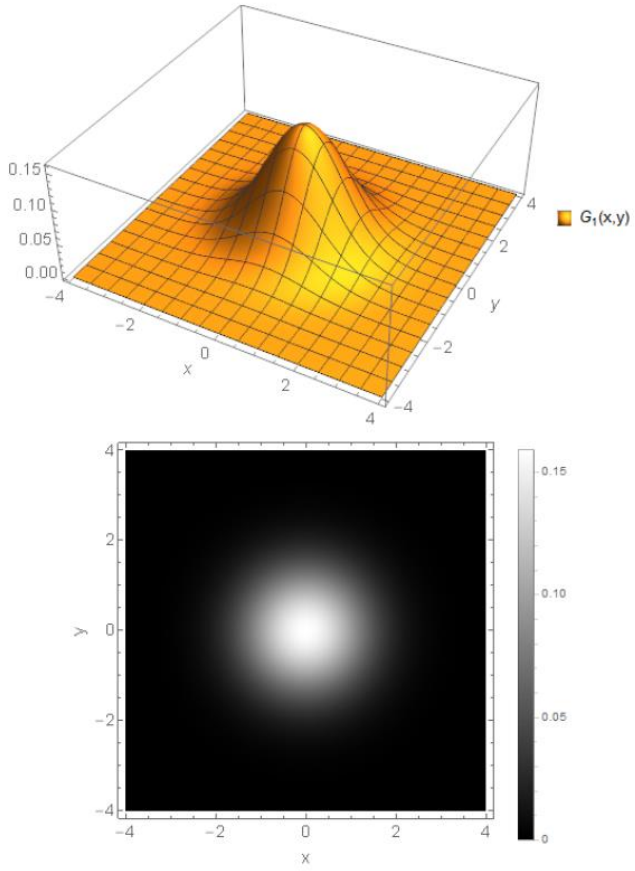
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Use only gradient

Invariant to additive changes!

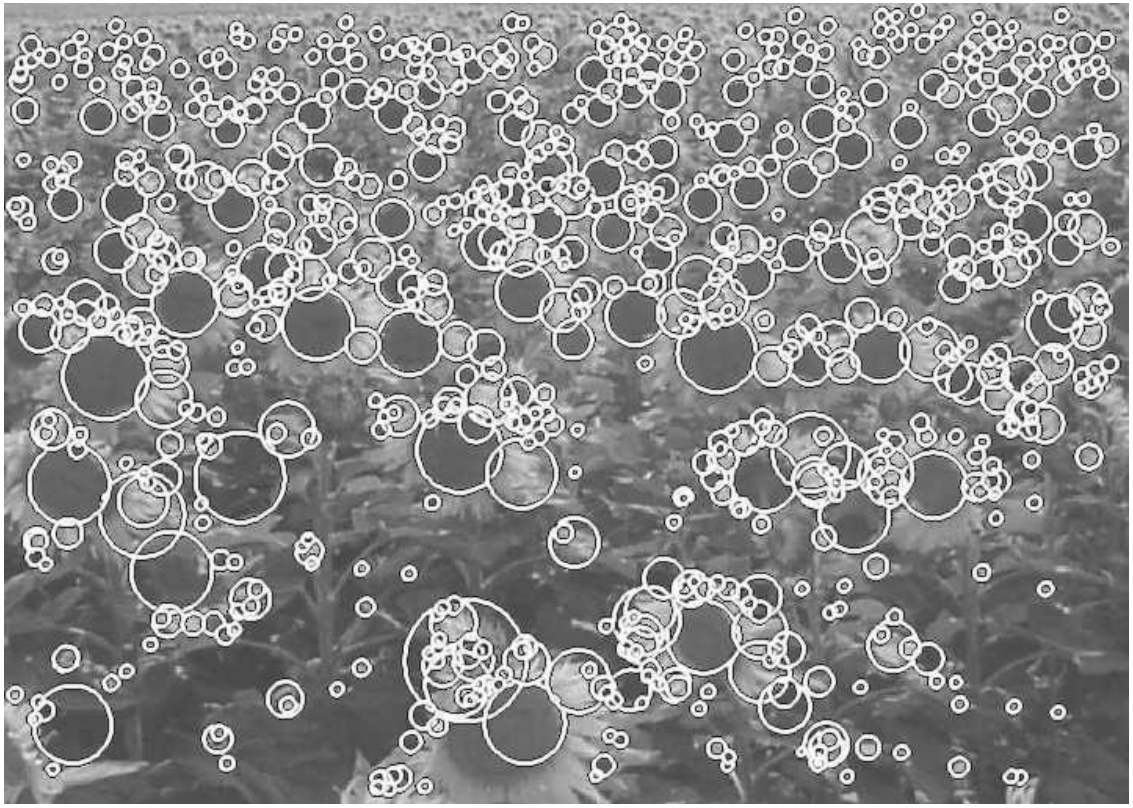
斑点检测（Blob Detection)

- 斑点



斑点检测 （Blob Detection)

- 斑点



斑点有何特征?

斑点有何特征?

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

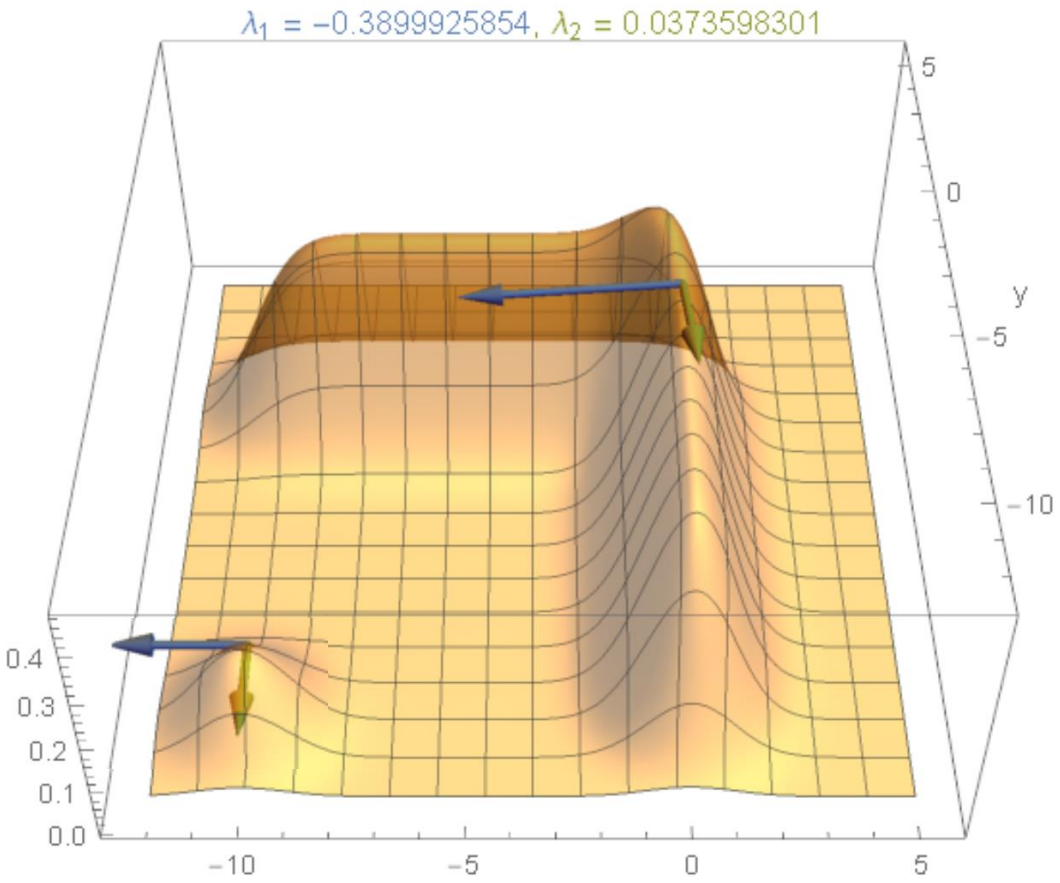
0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

图像（二维函数）的Hessian矩阵

$$H = \begin{pmatrix} \frac{\partial^2 I}{\partial^2 x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial^2 y^2} \end{pmatrix}$$

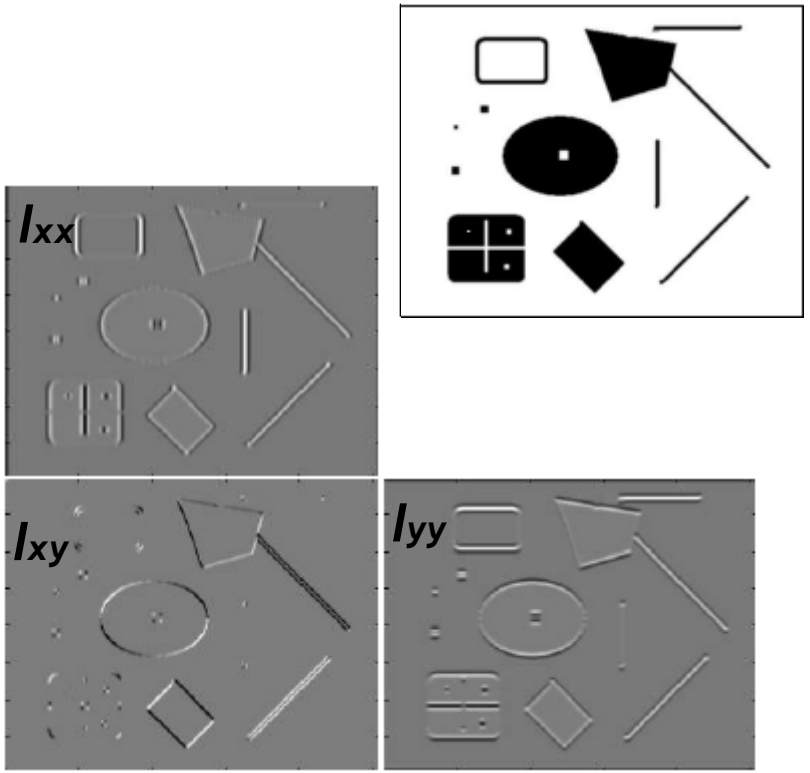
$$H = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$



Hessian Detector [Beaudet78]

- **Hessian determinant**

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



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

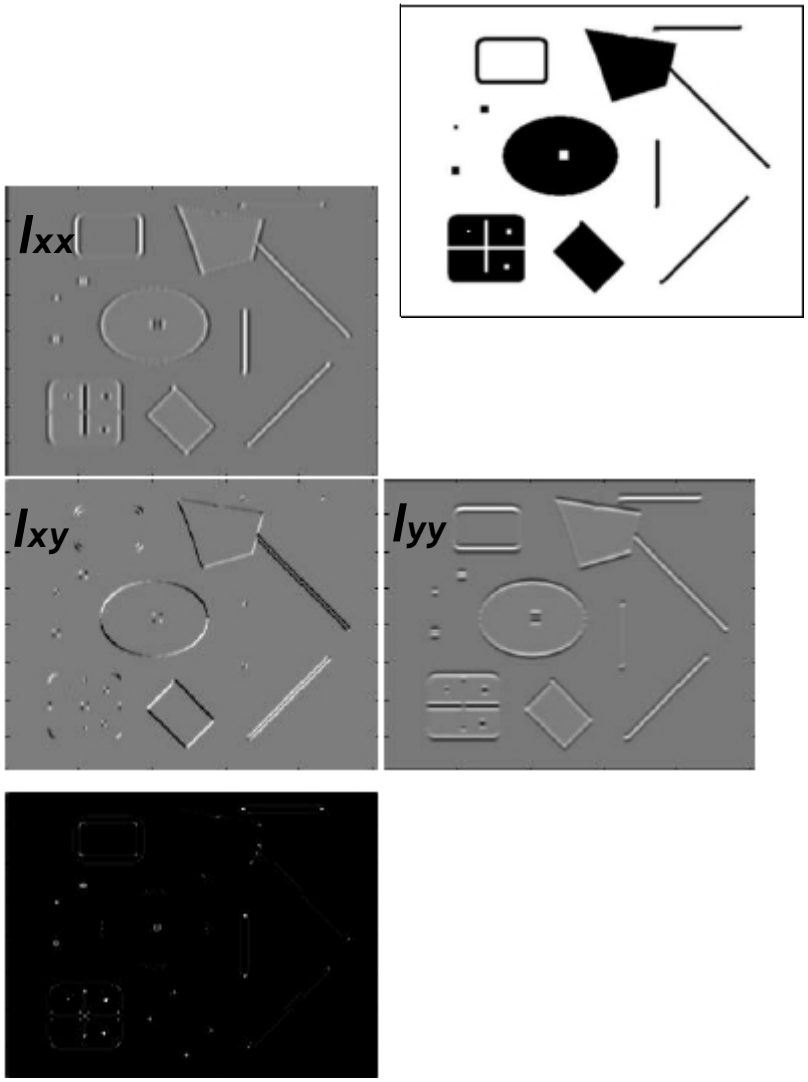
Slide credit: Krystian Mikolajczyk

Hessian Detector [Beaudet78]

- **Hessian determinant**

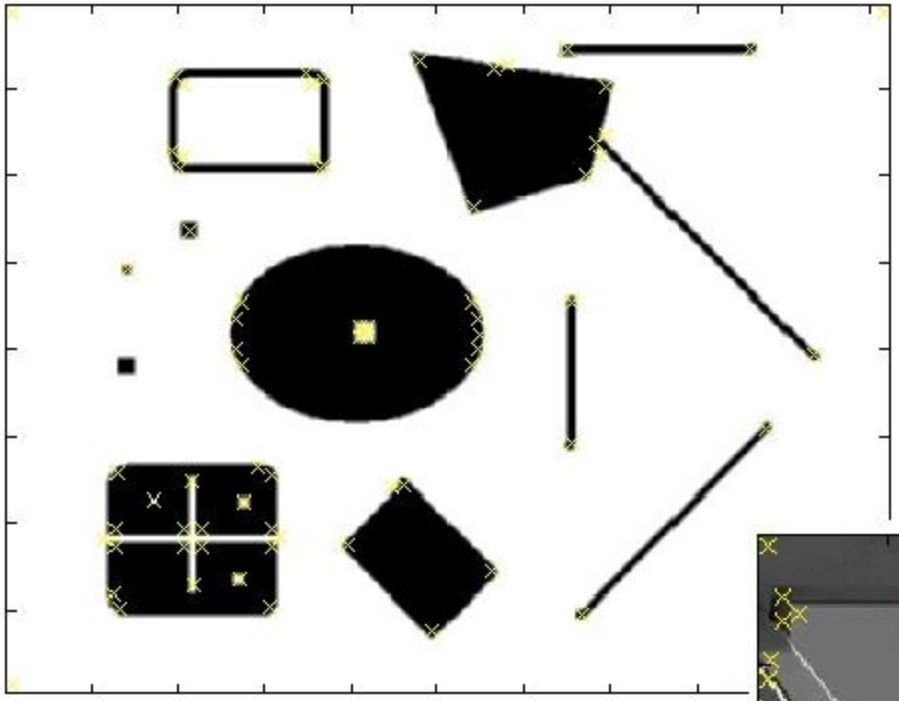
$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

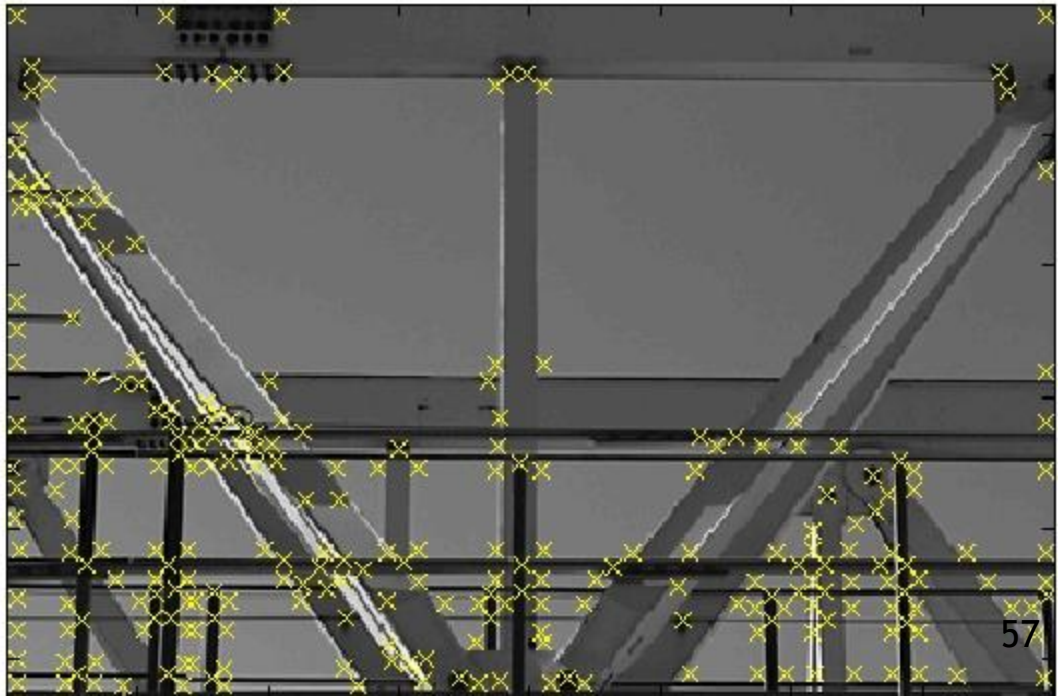
$$\det(H) = I_{xx} I_{yy} - I_{xy}^2 = \lambda_1 \lambda_2$$



Slide credit: Krystian Mikolajczyk

Hessian Detector – Responses [Beaudet78]





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

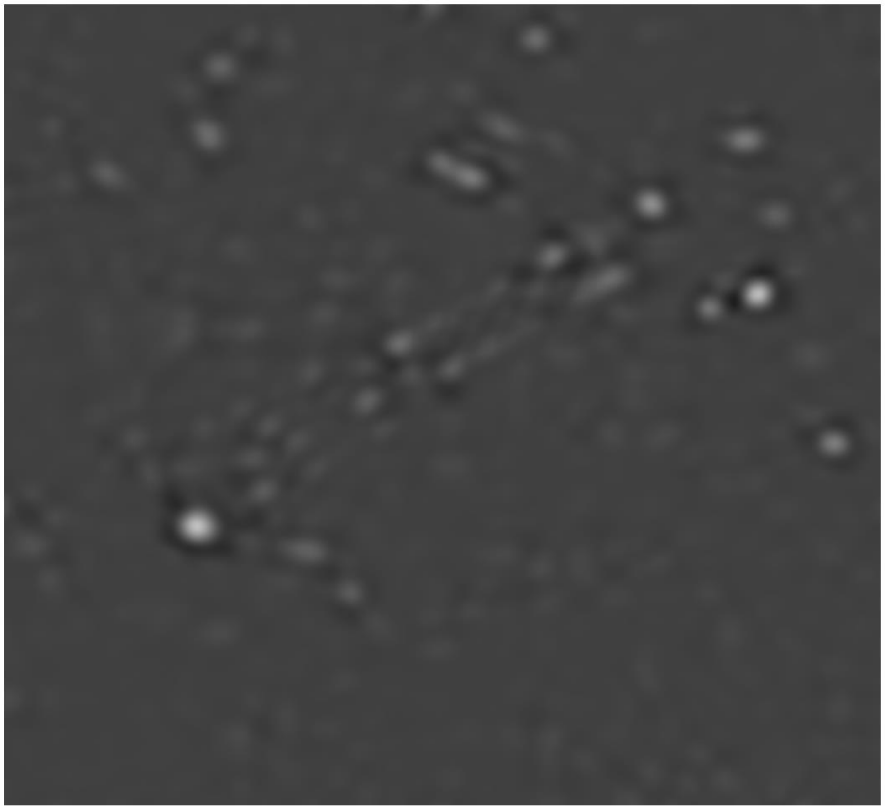
Slide credit: Krystian Mikolajczyk

Hessian Detector - Responses [Beaudet78]

$$\sigma = 2$$

Slide credit: Krystian Mikolajczyk

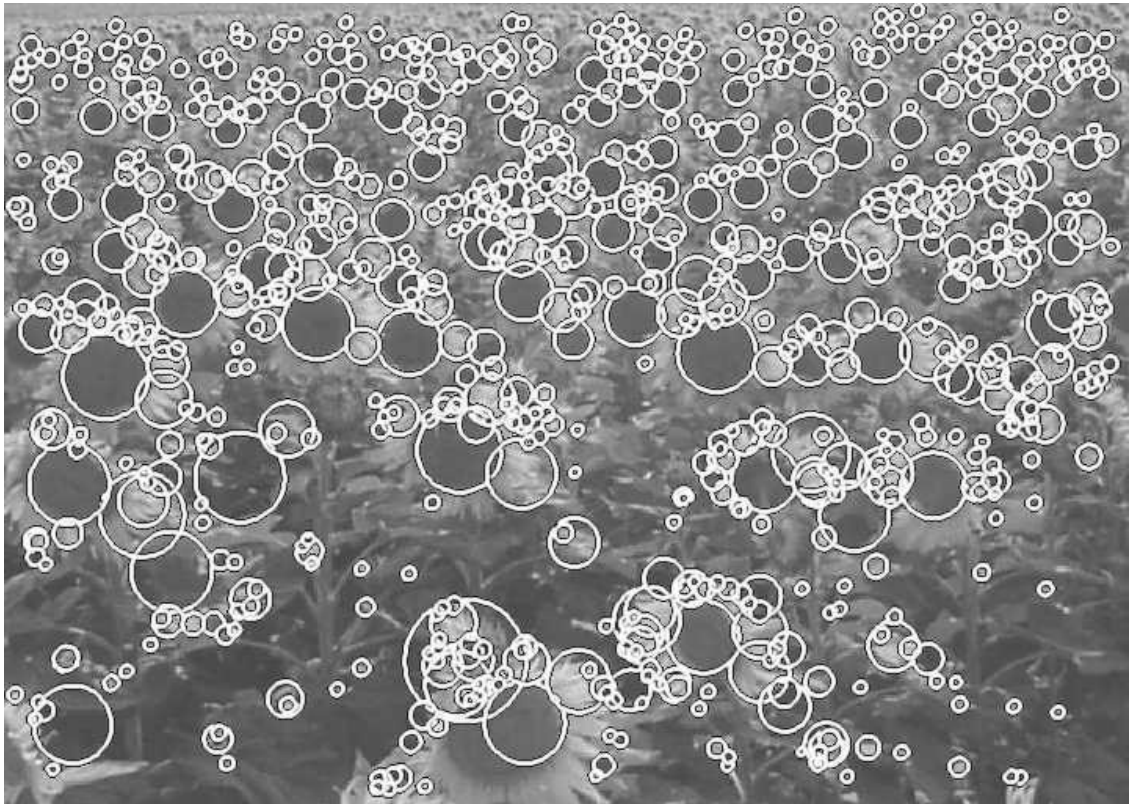
Hessian Detector - Responses [Beaudet78]



$$\sigma = 50$$

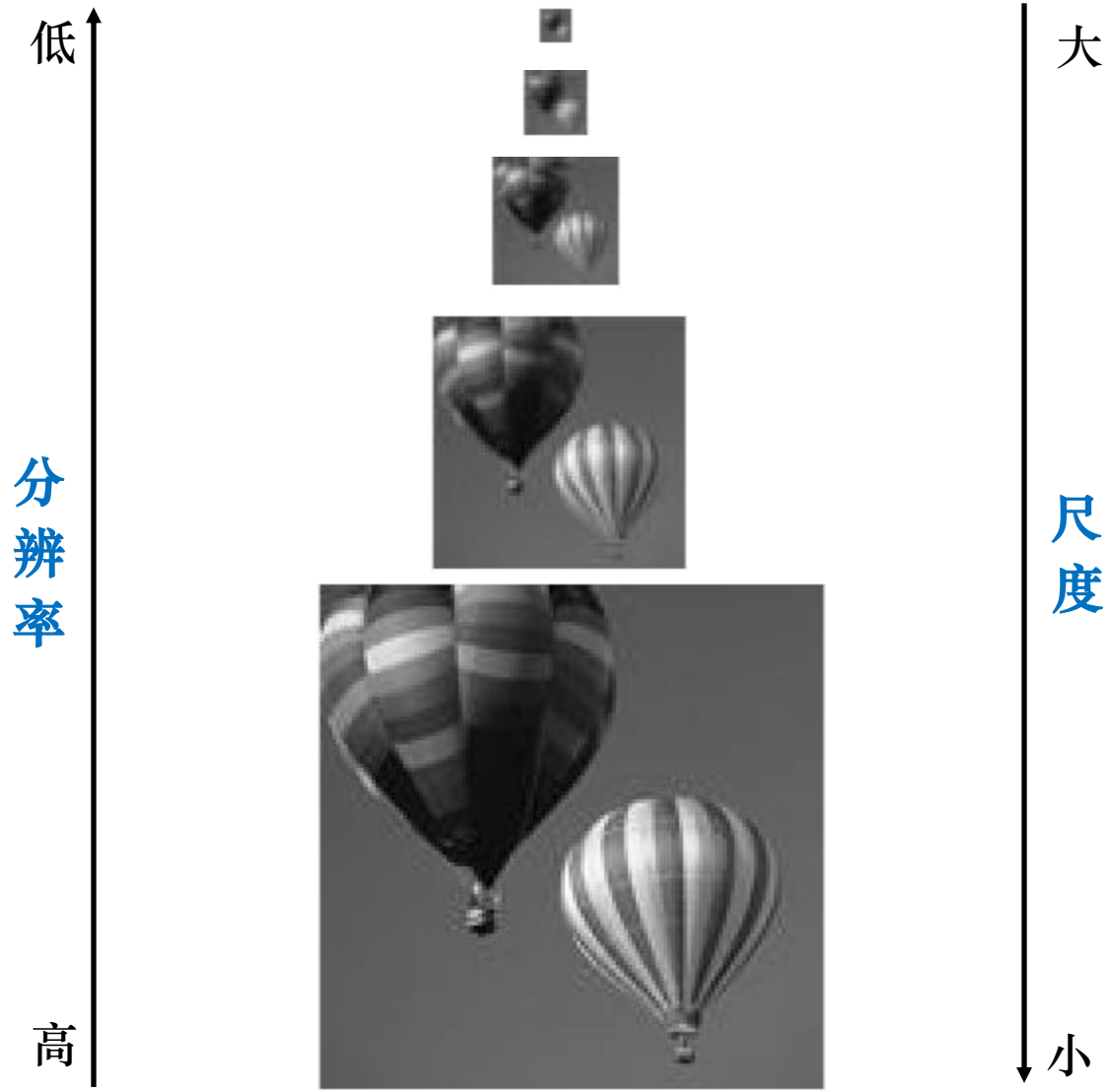
Slide credit: Krystian Mikolajczyk

Hessian对光照变化和几何变换的稳定性?

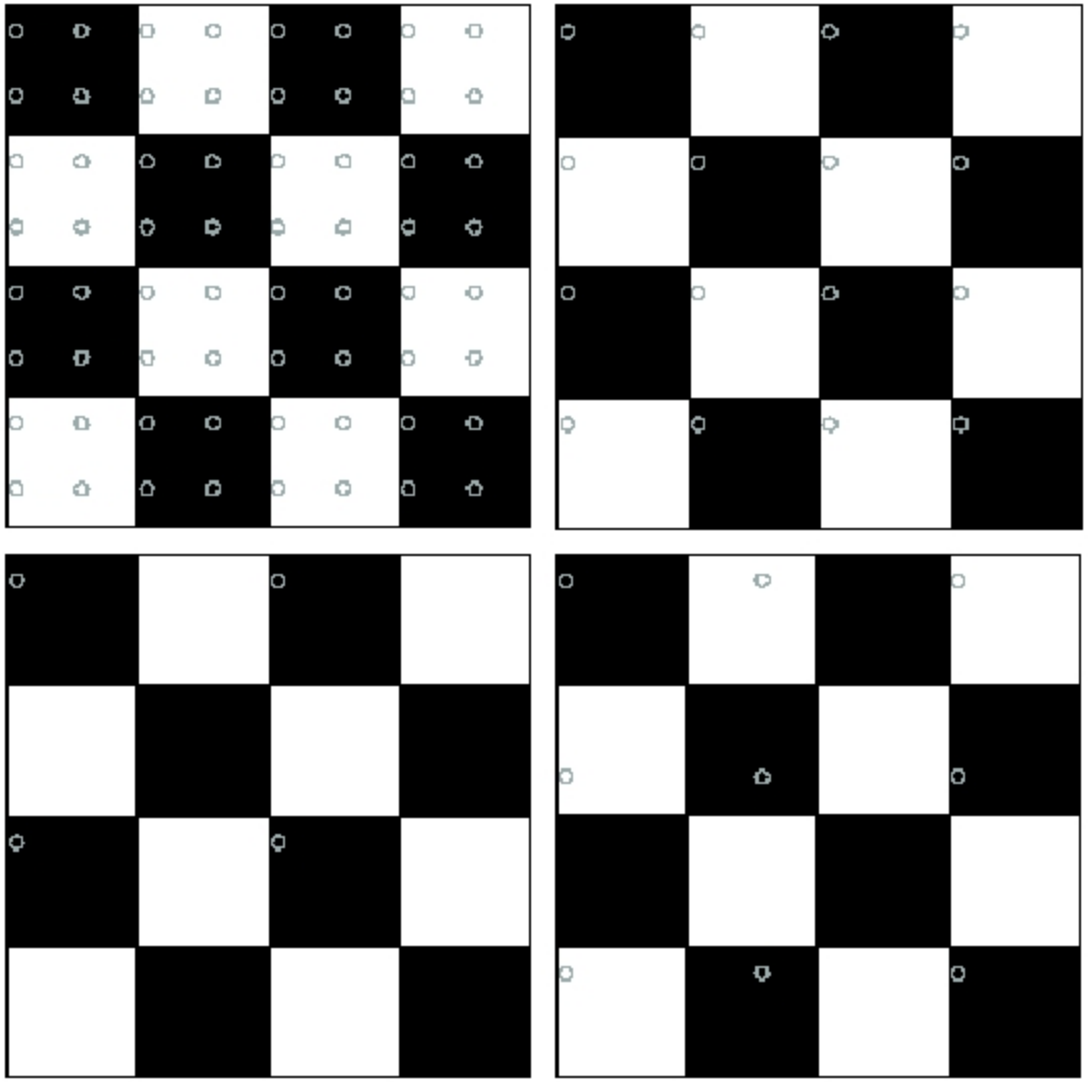


如何实现“尺度不变”？

尺度空间与图像金字塔 (Image Pyramid)



How Should We Go About Resampling?



Let's resample the checkerboard by taking one sample at each circle.

In the top left board, the new representation is reasonable. Top right also yields a reasonable representation.

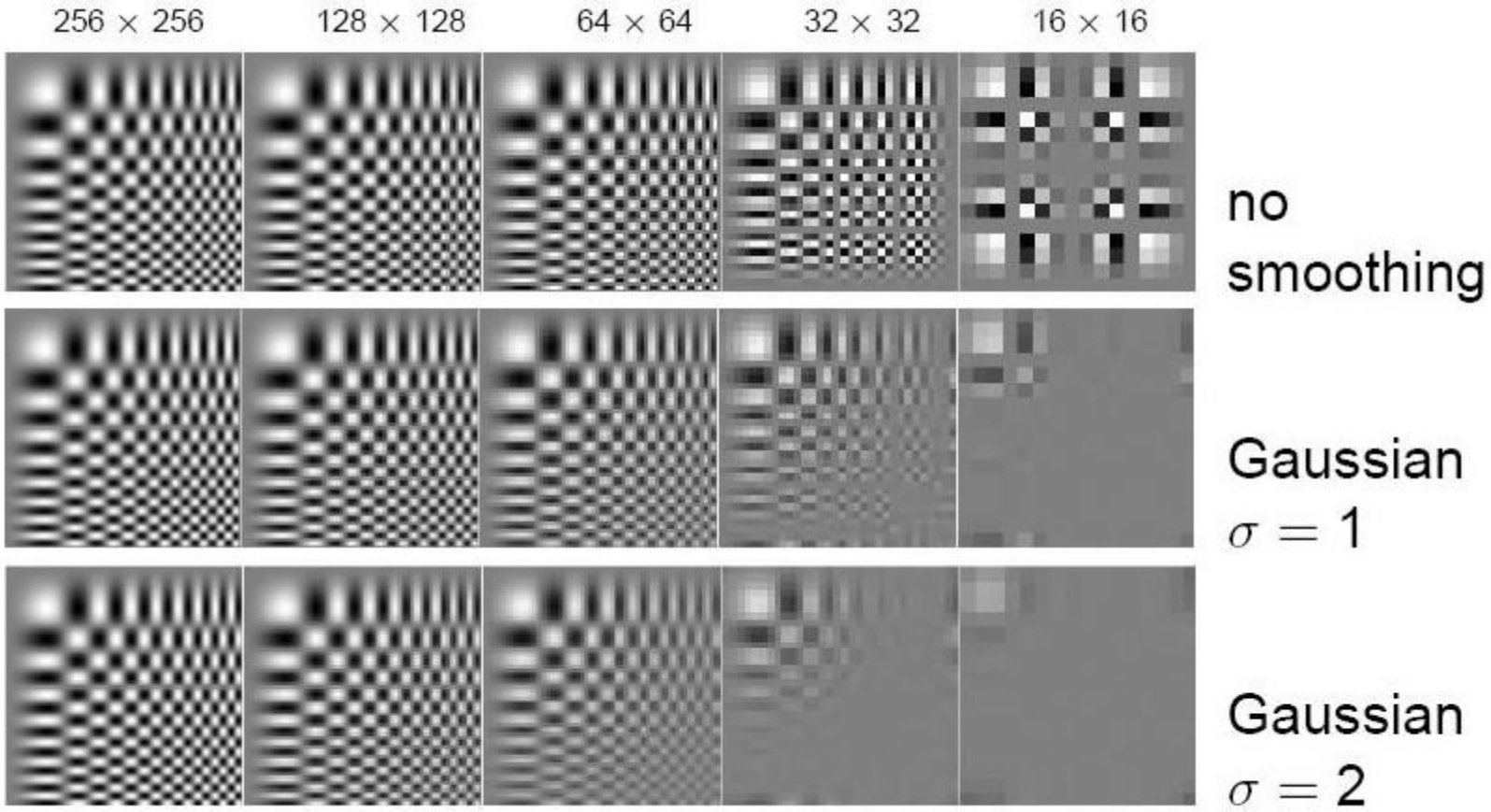
Bottom left is all black (dubious) and bottom right has checks that are too big.

Image Source: Forsyth & Ponce

Aliasing in Graphics

Image Source: Alexej Efros

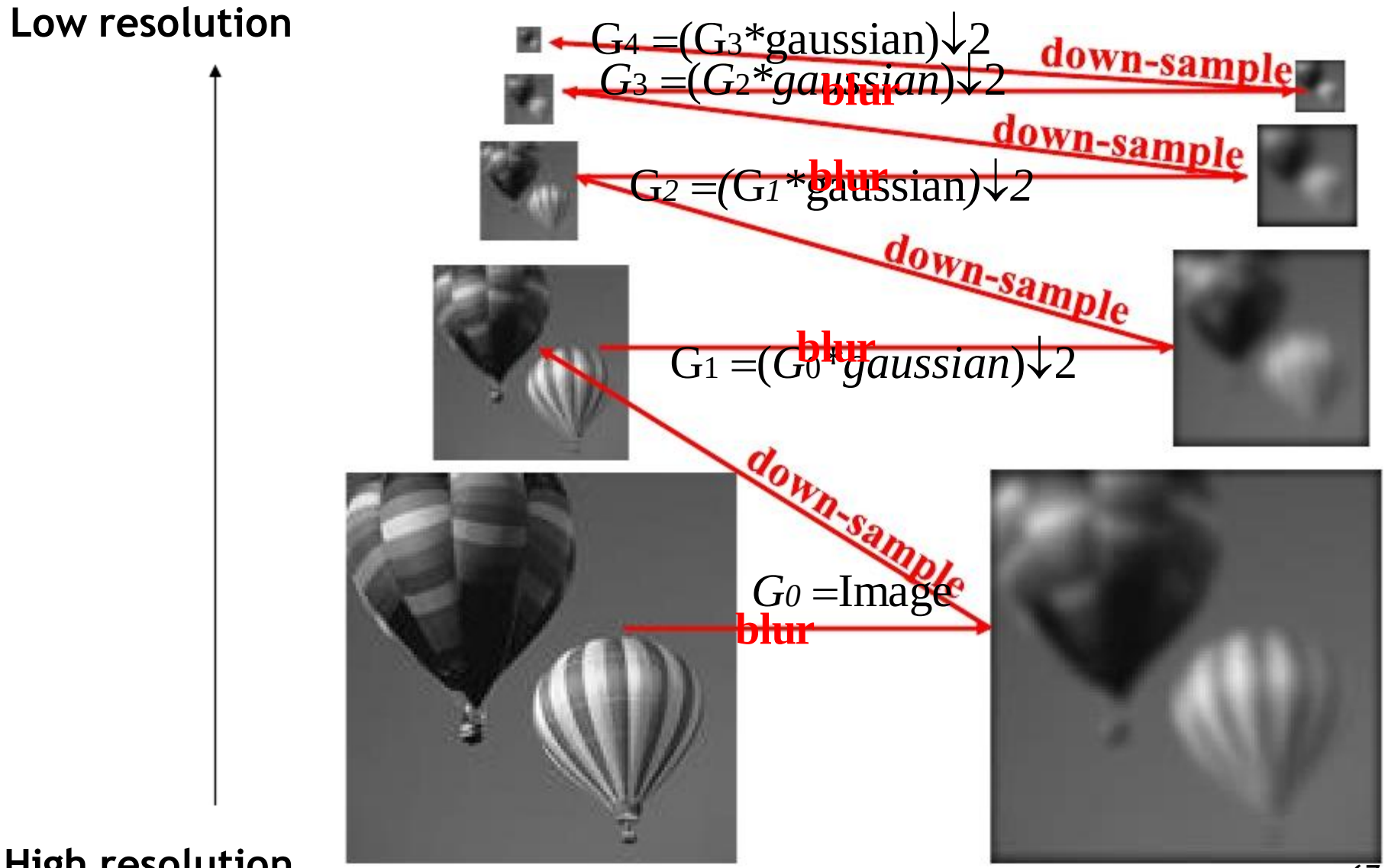
Resampling with Prior Smoothing



- **Note: We cannot recover the high frequencies, but we can avoid artifacts by smoothing before resampling.**

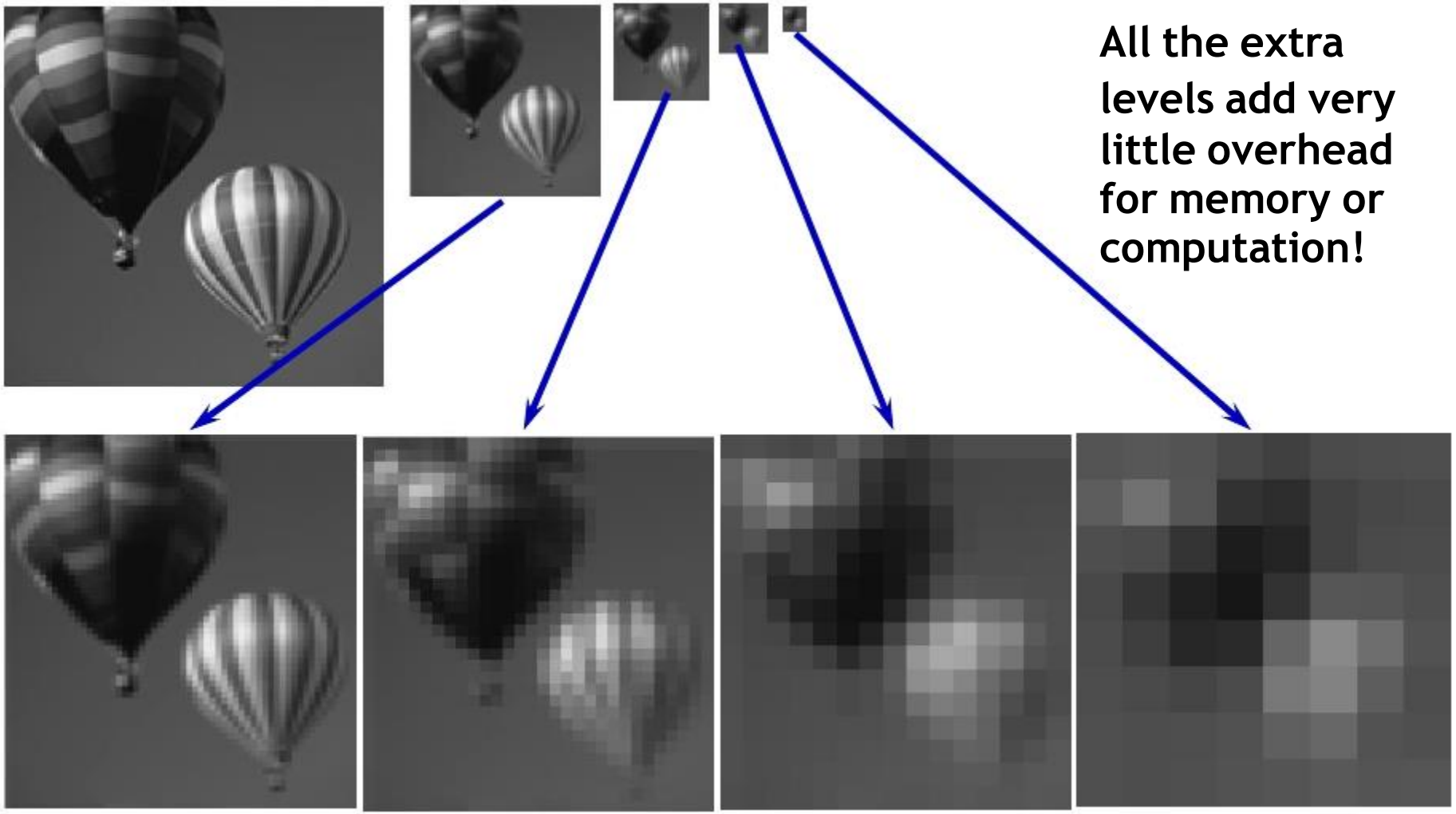
Image Source: Forsyth & Ponce

The Gaussian Pyramid



Source: Irani & Basri

Gaussian Pyramid – Stored Information



All the extra levels add very little overhead for memory or computation!

Source: Irani & Basri

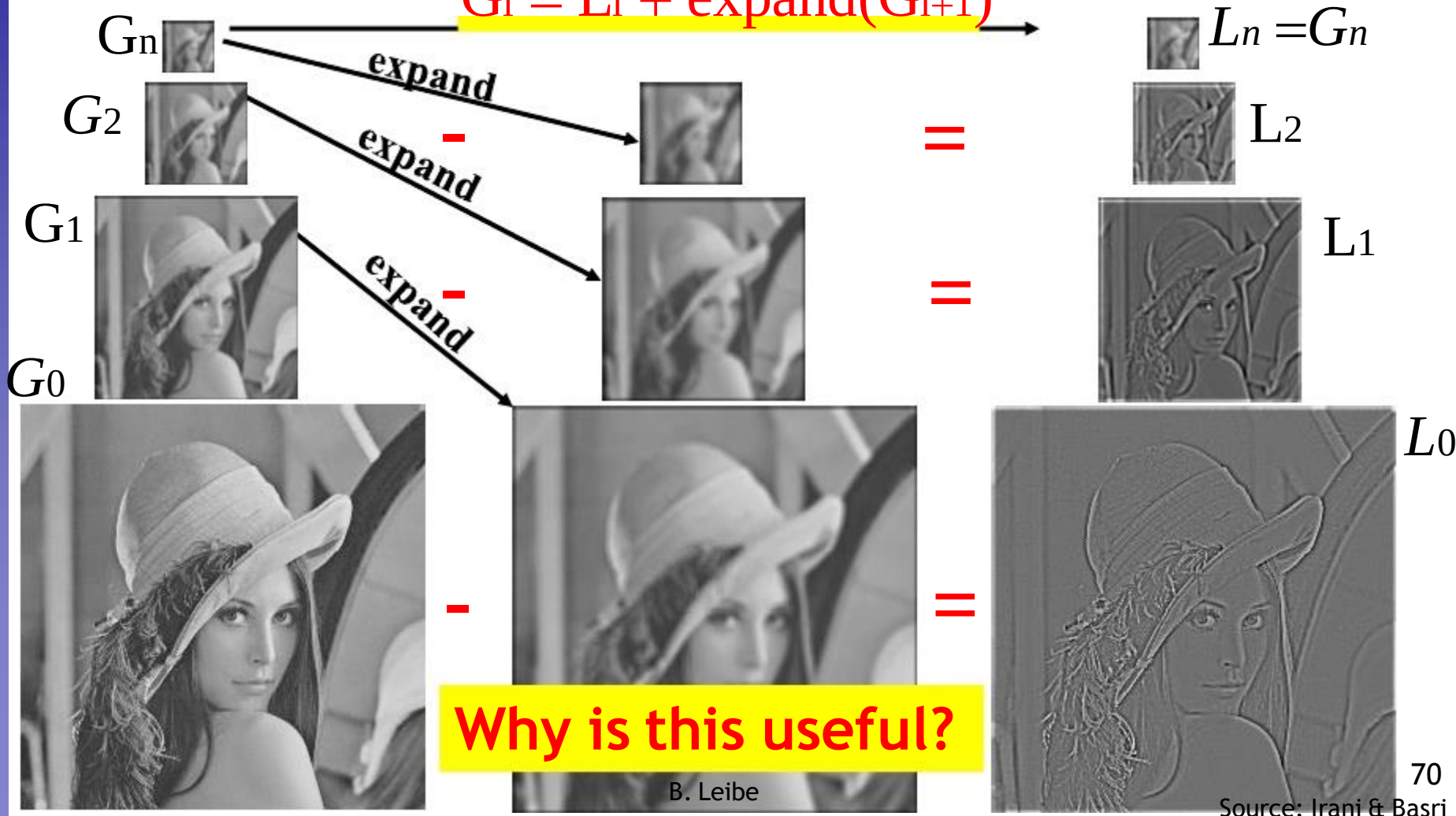
Gaussian Pyramid: Summary

- Construction: create each level from previous one
  - Smooth and sample
- Smooth with Gaussians, in part because
  - a Gaussian*Gaussian = another Gaussian
  - $G(\sigma_1) * G(\sigma_2) = G(\mathrm{sqrt}(\sigma_1^2 + \sigma_2^2))$
- Gaussians are low-pass filters, so the representation is redundant once smoothing has been performed.
  - $\Rightarrow$ There is no need to store smoothed images at the full original resolution.

The Laplacian Pyramid

$$L_i = G_i - \text{expand}(G_{i+1})$$

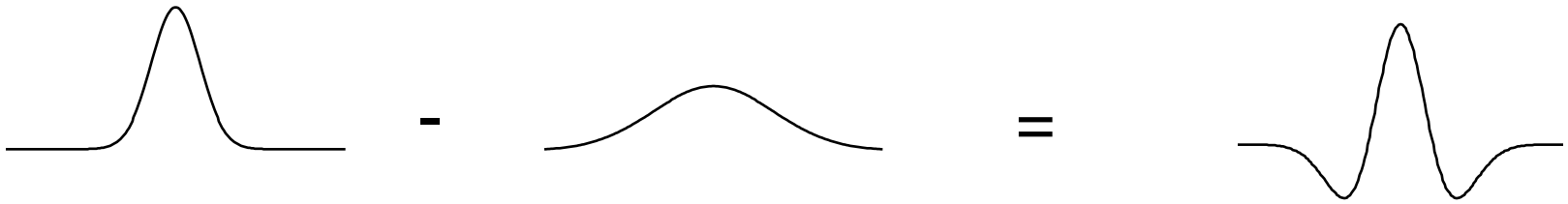
$$G_i = L_i + \text{expand}(G_{i+1})$$

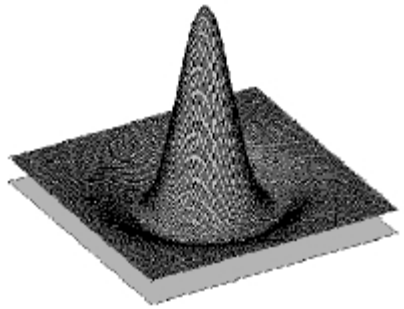


Why is this useful?

Source: Irani & Basri

Laplacian ~ Difference of Gaussian

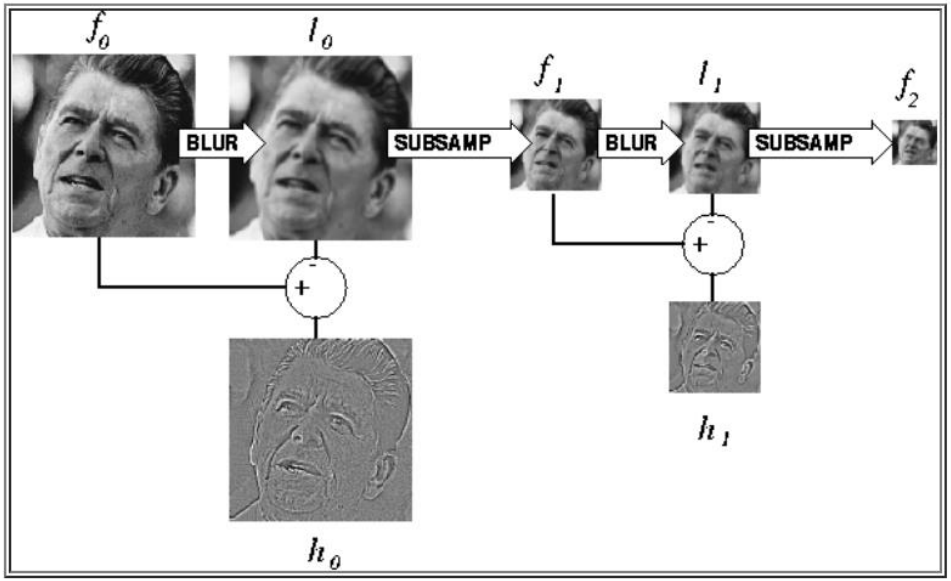




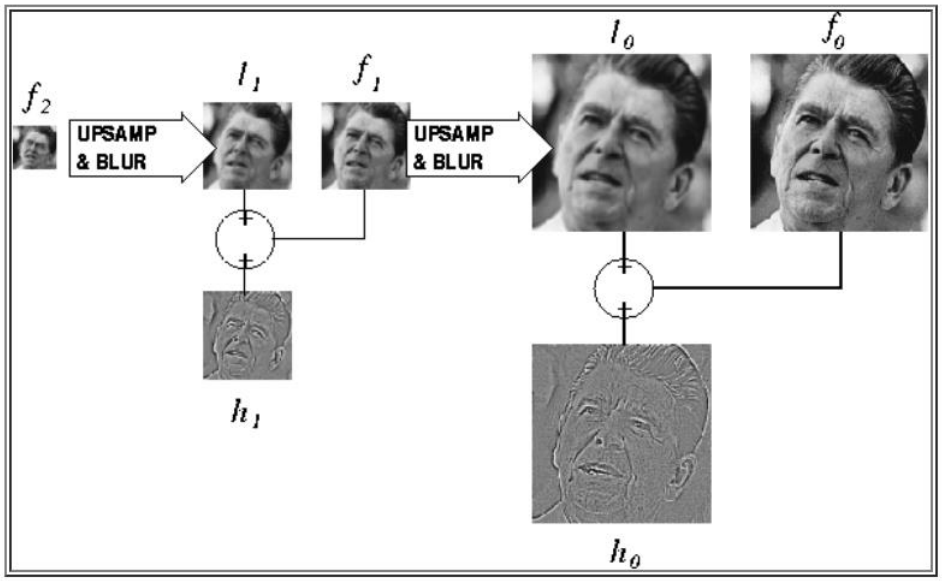
DoG = Difference of Gaussians

Cheap approximation - no derivatives needed.

Laplacian 图像分解&重建



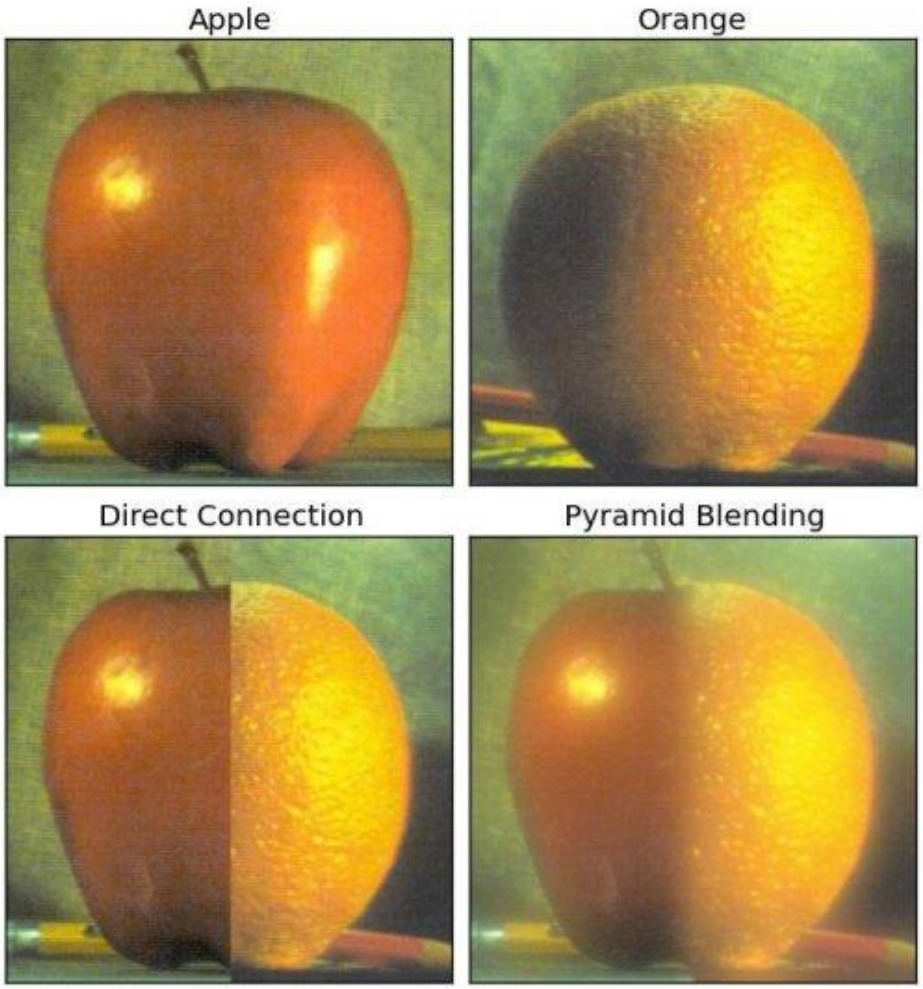
分解



重建

Laplacian图像融合

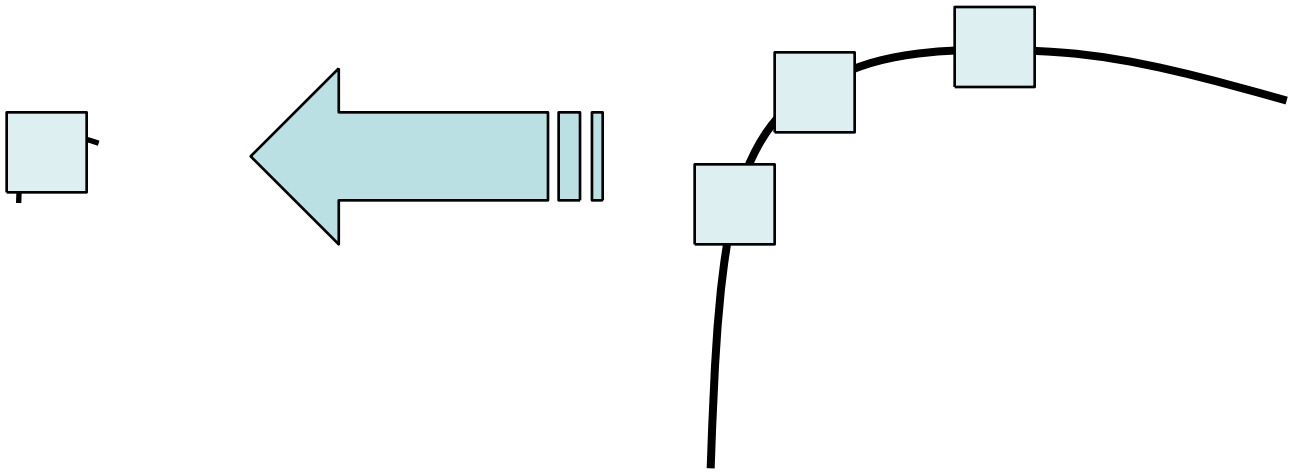
1. 构造Laplacian金字塔
2. 在金字塔的每一层进行融合
3. 用融合后的金字塔进行重建



https://docs.opencv.org/3.1.0/dc/dff/tutorial_py_pyramids.html

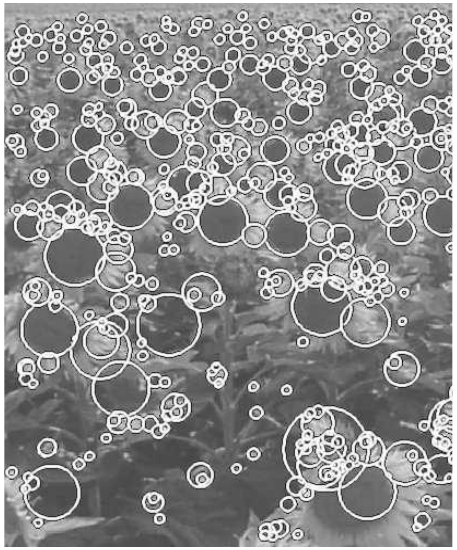
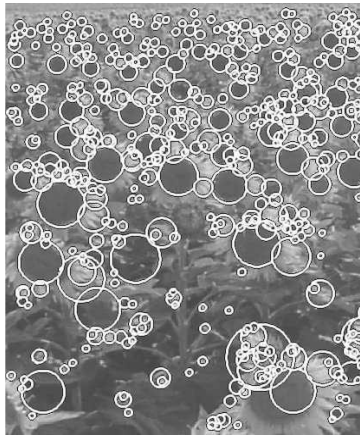
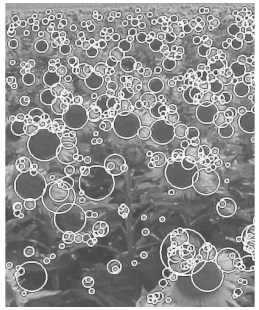
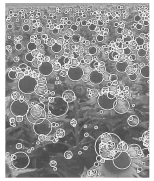
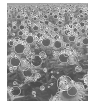
实现“尺度不变”？

- **尺度相关：**在某一尺度上不是特征点的位置，在其它尺度上可能是特征点



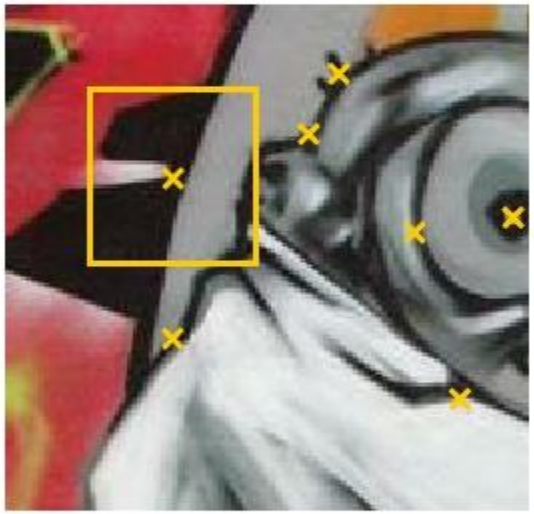
实现“尺度不变”？

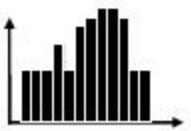
- 在不同尺度同时进行检测与匹配

Naïve Approach: Exhaustive Search

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





Similarity measure

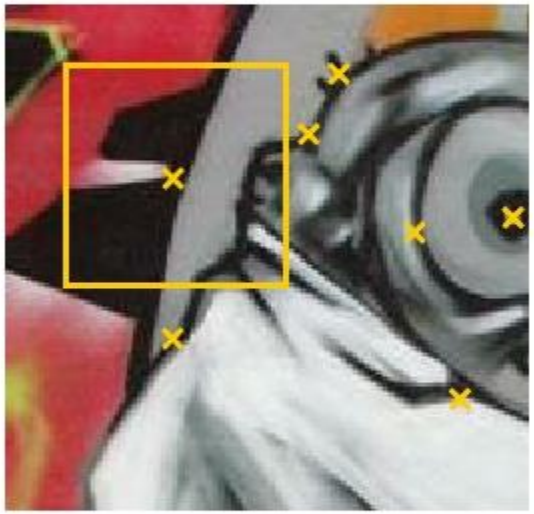
$\neq$

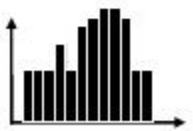
$d(f_A, f_B)$

Slide credit: Krystian Mikolajczyk

Naïve Approach: Exhaustive Search

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





Similarity measure

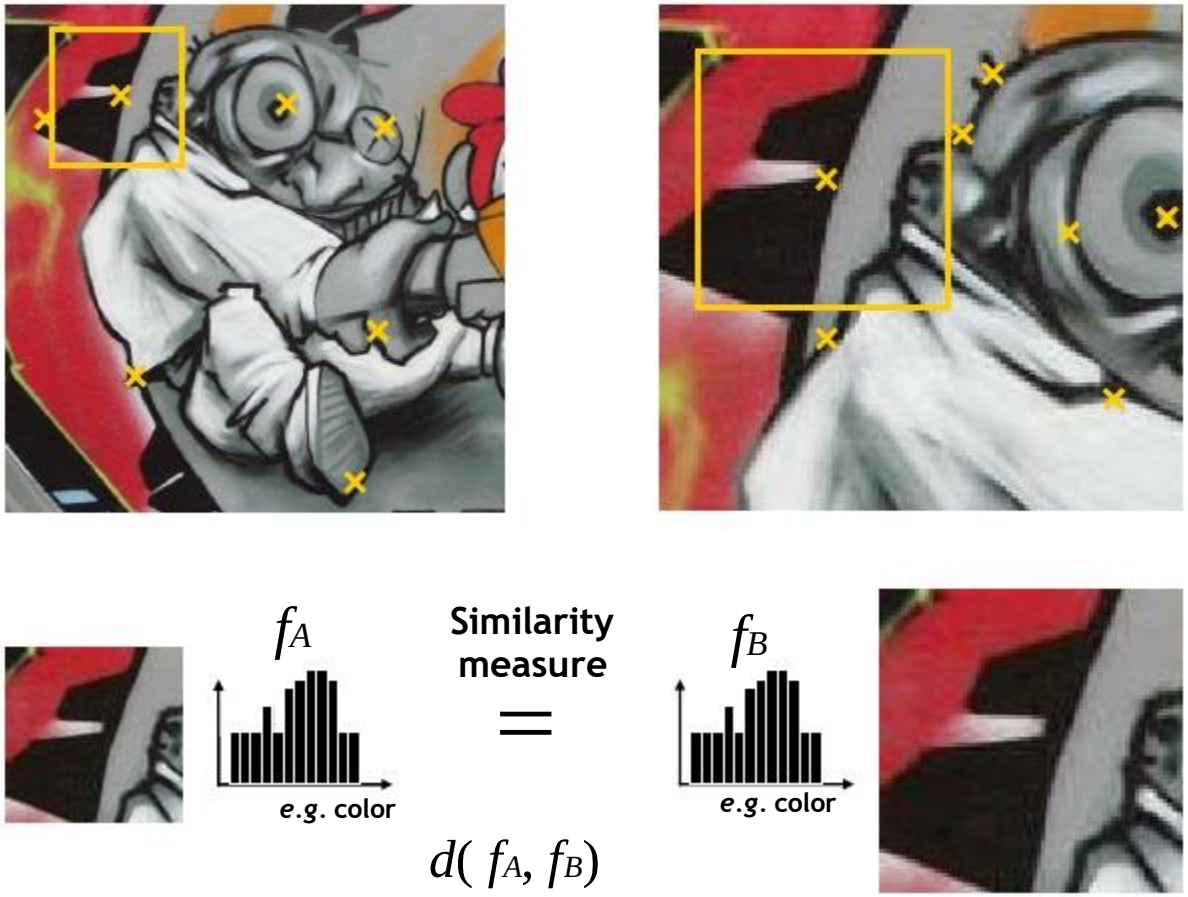
$\neq$

$d(f_A, f_B)$

Slide credit: Krystian Mikolajczyk

Naïve Approach: Exhaustive Search

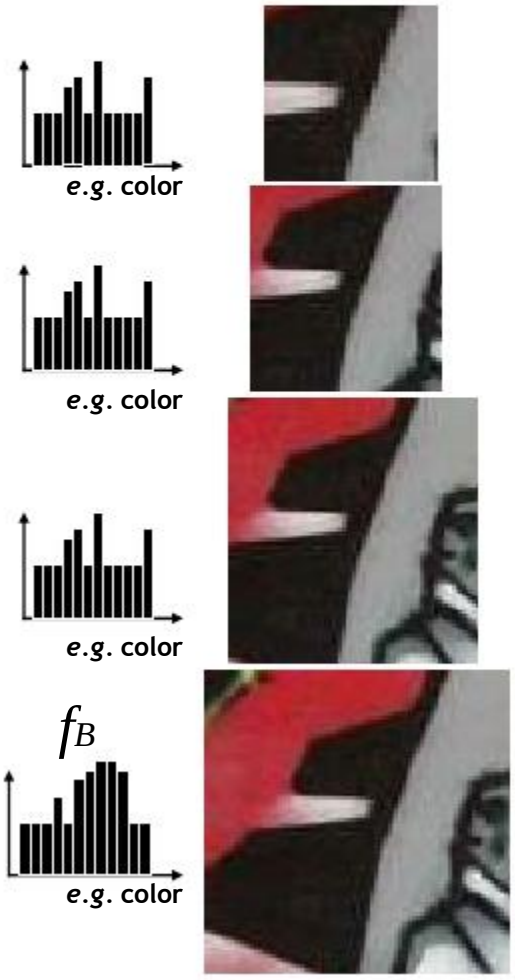
- **Multi-scale procedure**
  - Compare descriptors while varying the patch size

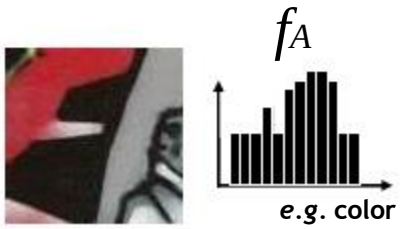


Slide credit: Krystian Mikolajczyk

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





Similarity measure

=

$d(f_A, f_B)$

Slide credit: Krystian Mikolajczyk

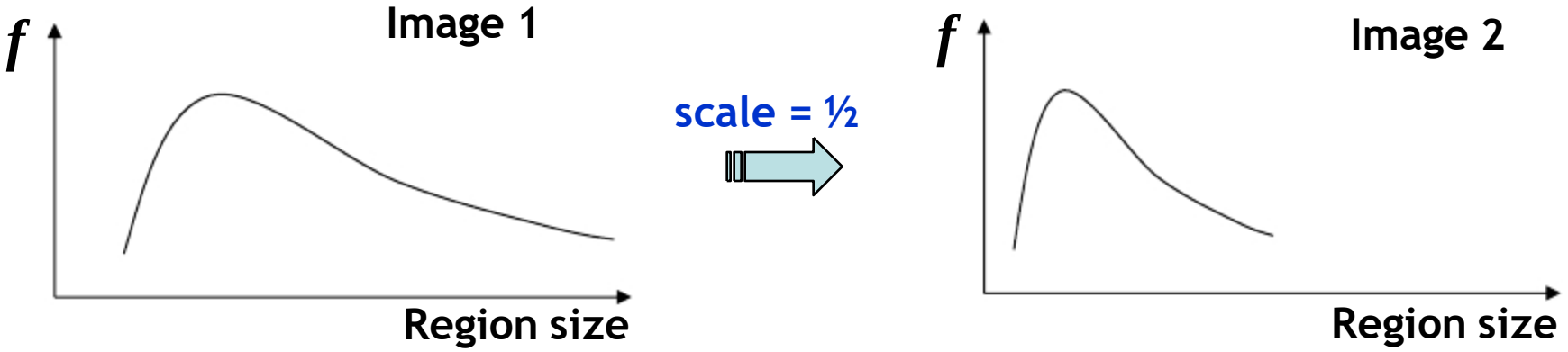
B. Leibe

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)

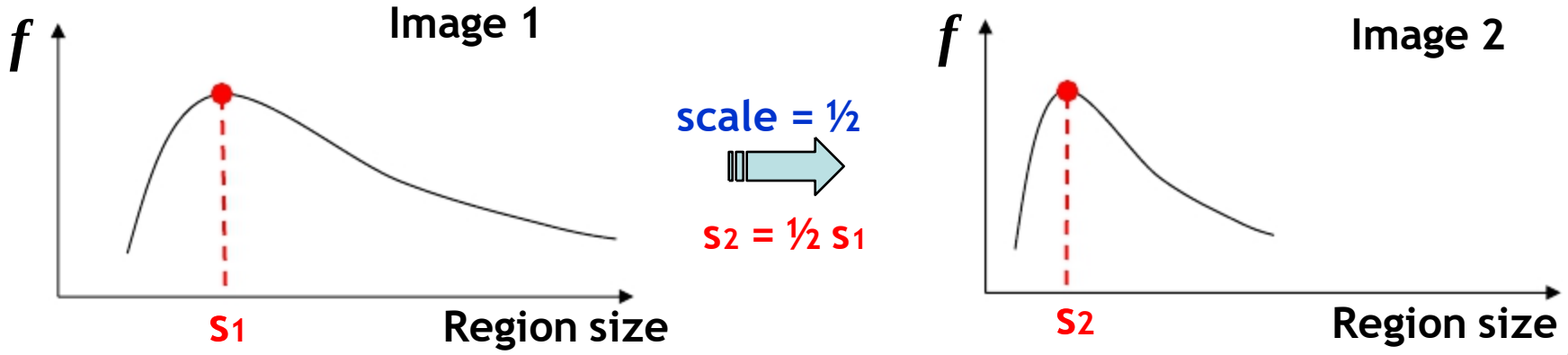


Slide credit: Kristen Grauman

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.

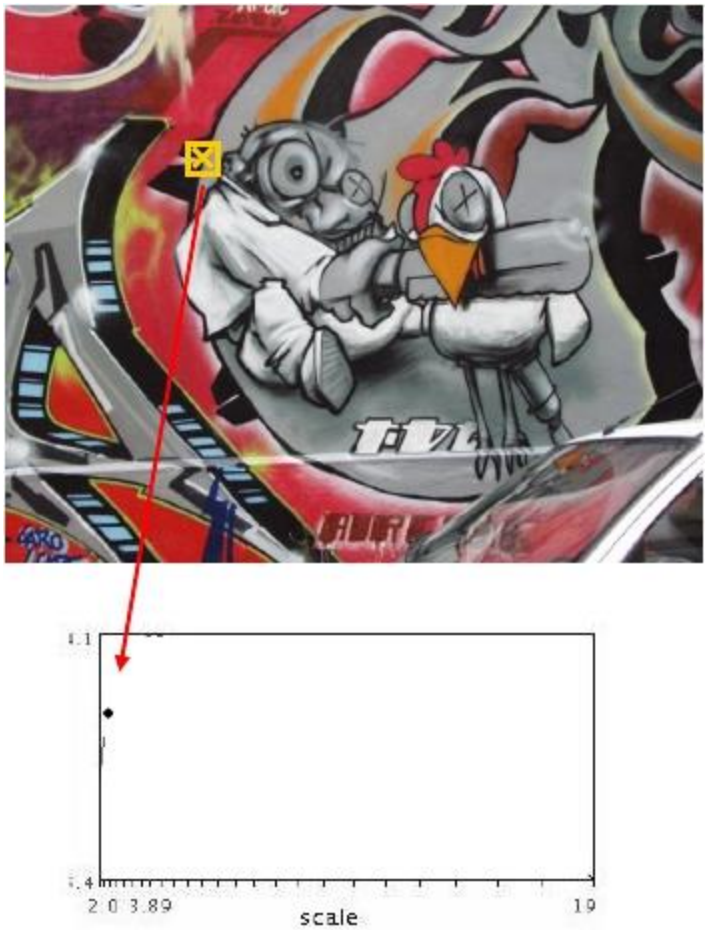
Important: this scale invariant region size is found in each image independently!



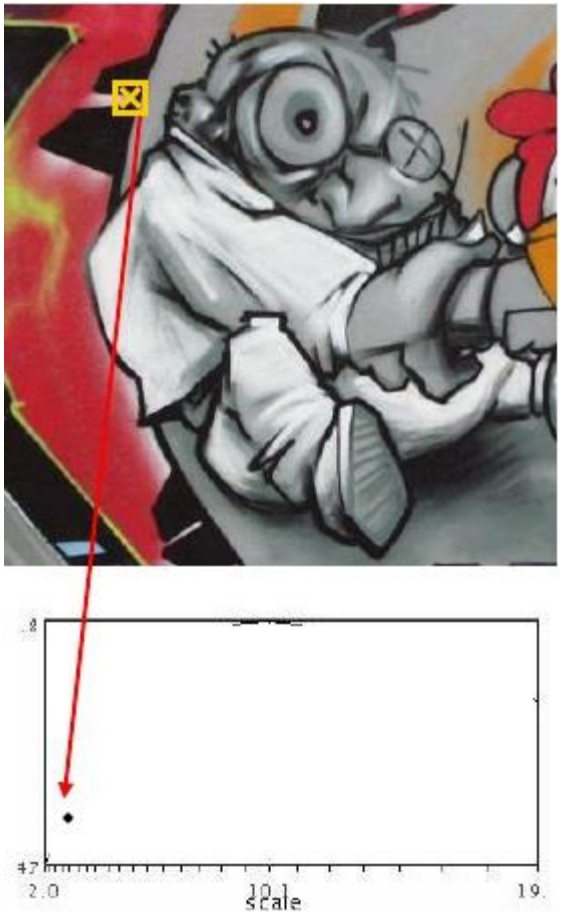
Slide credit: Kristen Grauman

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f\,(I_{i1\ldots im}(x,\sigma))$

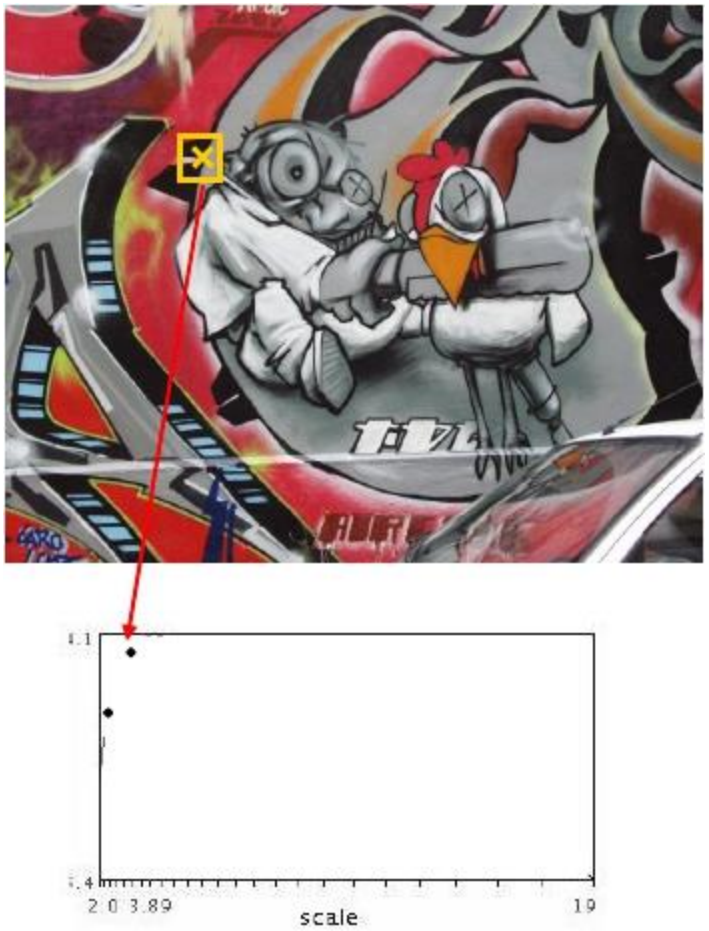


$f\,(I_{i1\ldots im}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



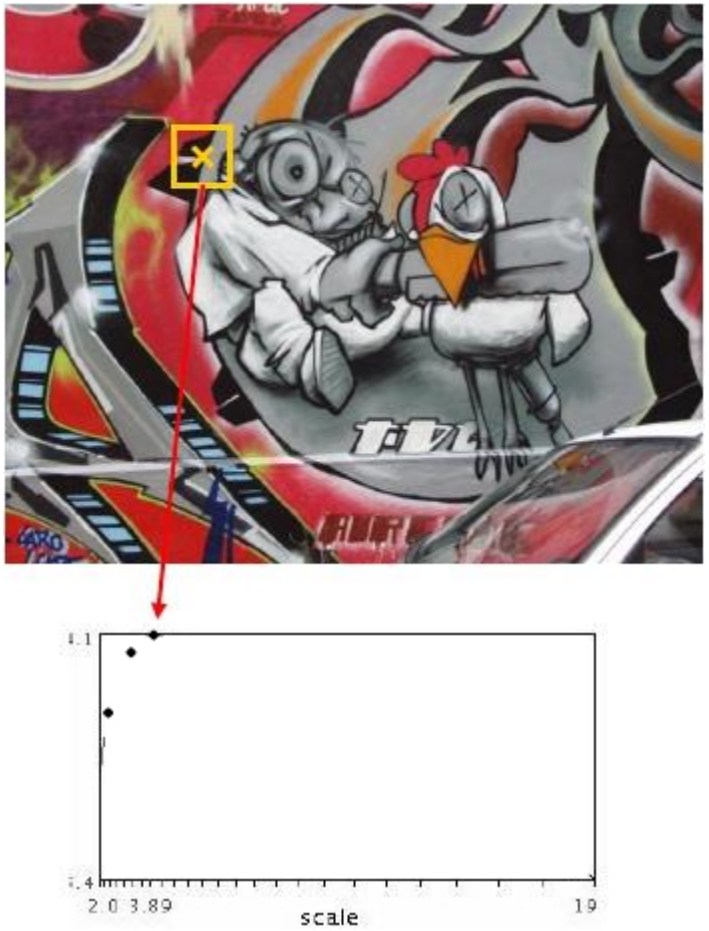
$f\,(I_{i1\ldots im}(x,\sigma))$

$f\,(I_{i1\ldots im}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



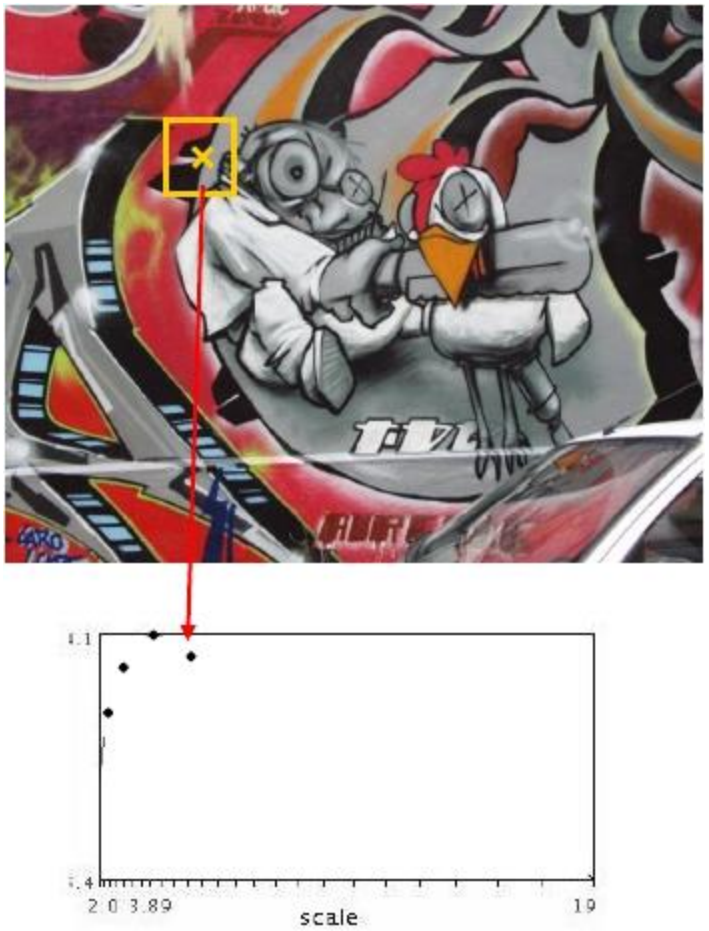
$f\,(I_{i1\ldots im}(x,\sigma))$

$f\,(I_{i1\ldots im}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



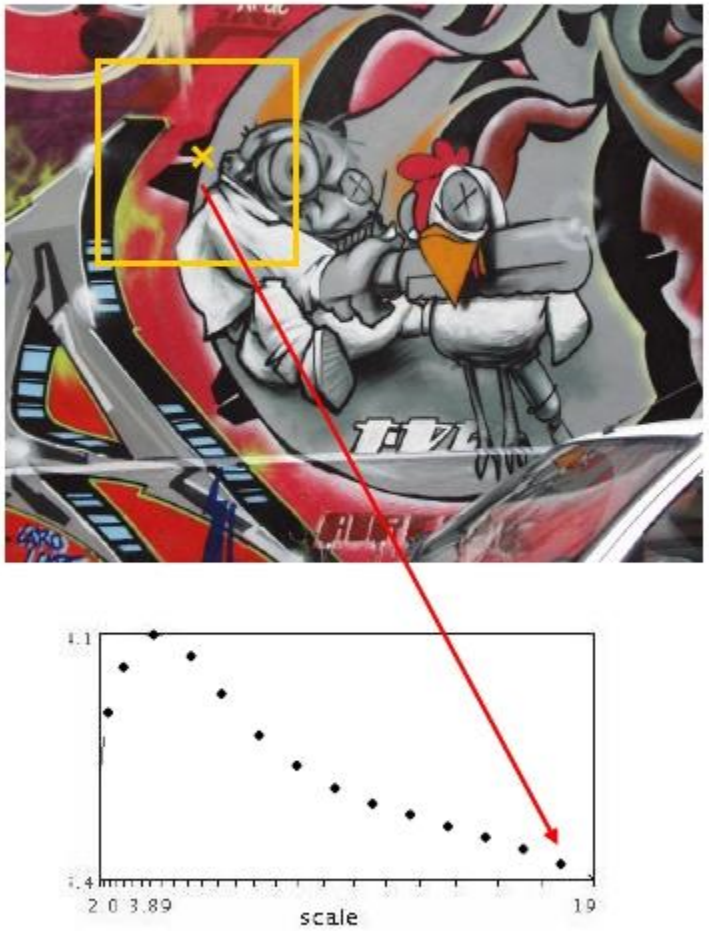
$f\,(I_{i1\ldots im}(x,\sigma))$

$f\,(I_{i1\ldots im}(x',\sigma))$

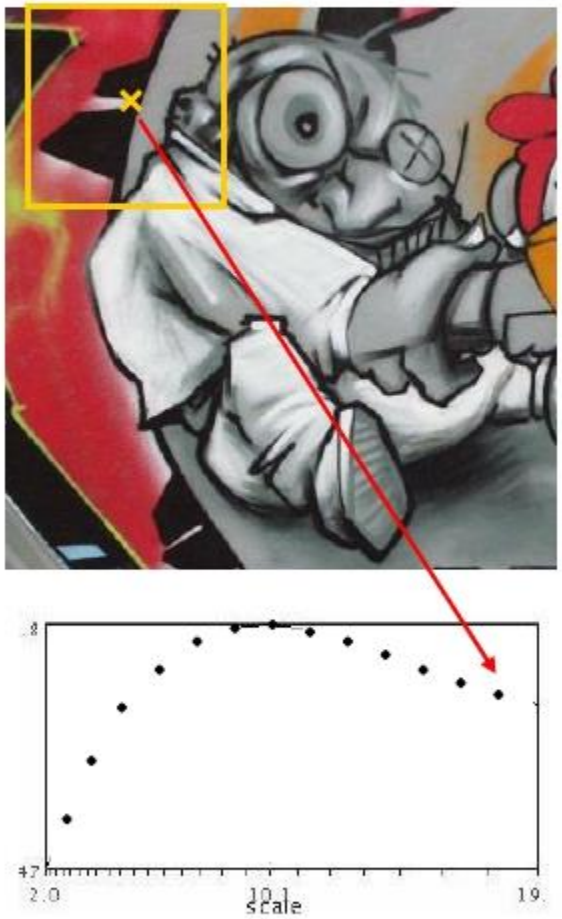
Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f\,(I_{i1\ldots im}(x,\sigma))$

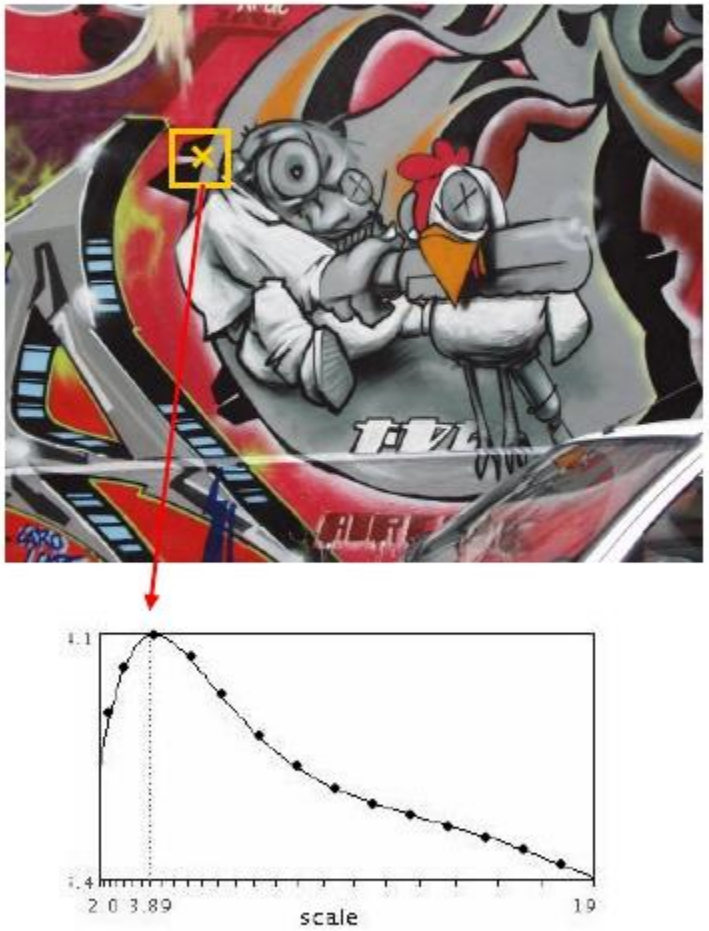


$f\,(I_{i1\ldots im}(x',\sigma))$

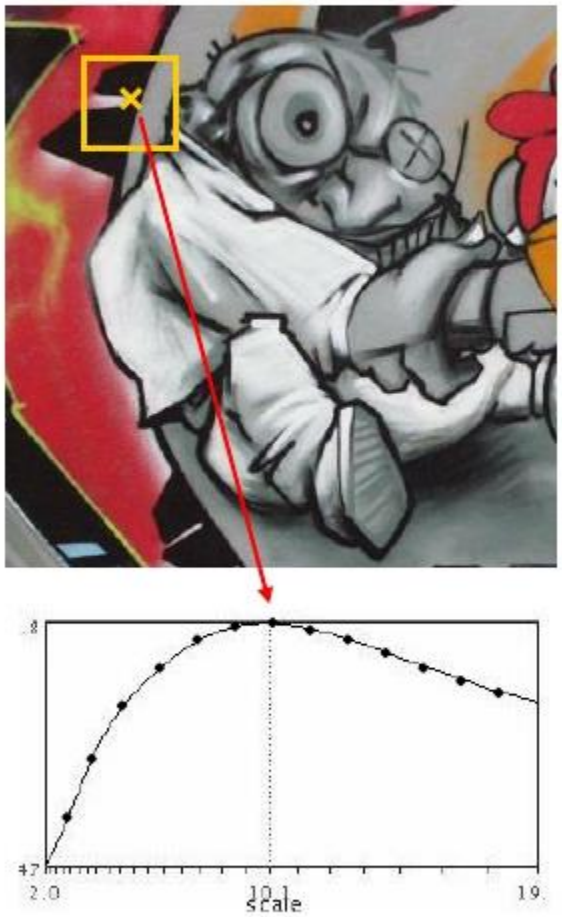
Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f\,(I_{i1\ldots im}(x,\sigma))$

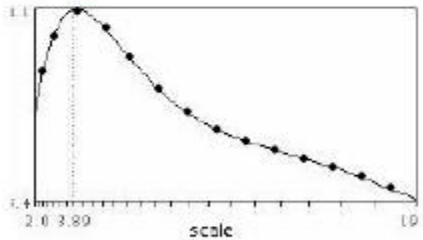


$f\,(I_{i1\ldots im}(x',\sigma'))$

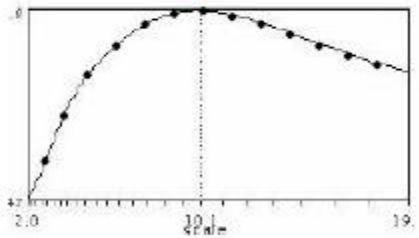
Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

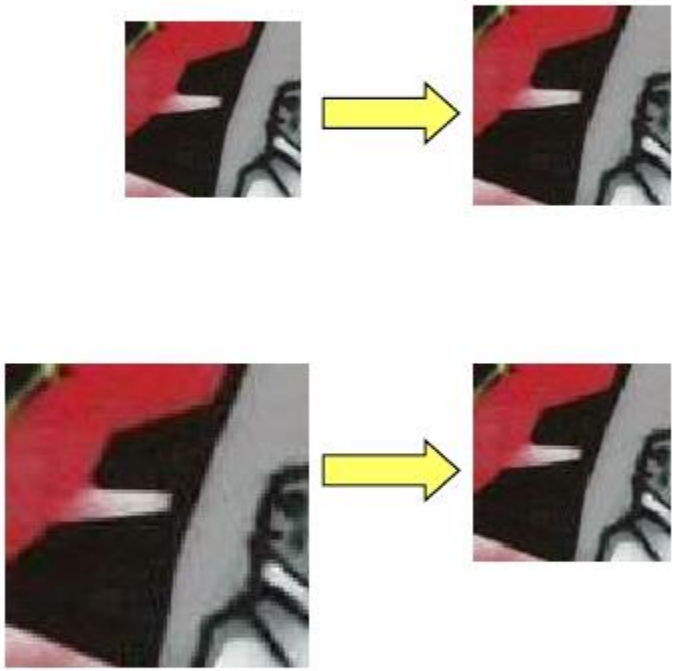
- **Normalize: Rescale to fixed size**



$f(I_{i1\ldots im}(x,\sigma))$

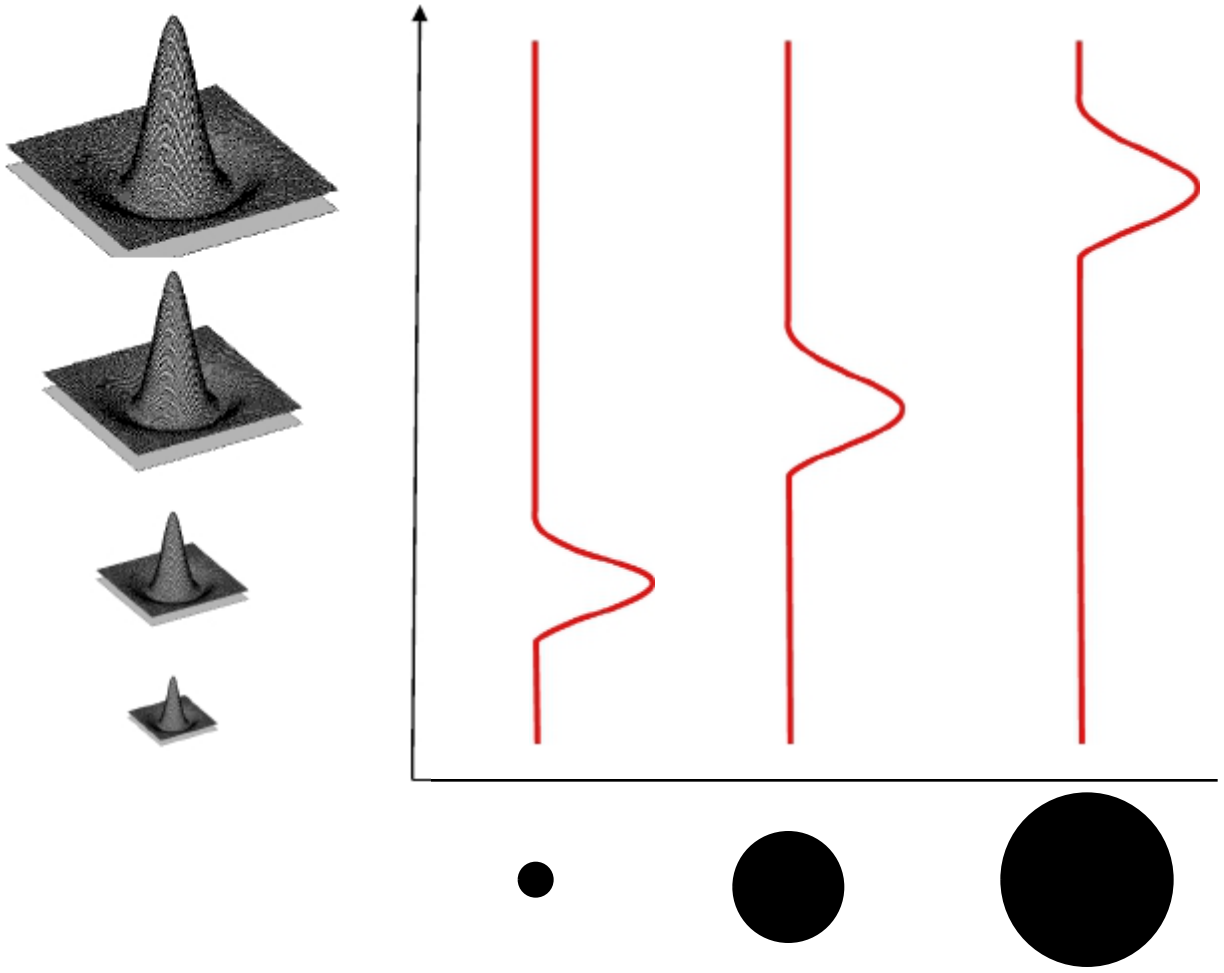


$f(I_{i1\ldots im}(x',\sigma'))$



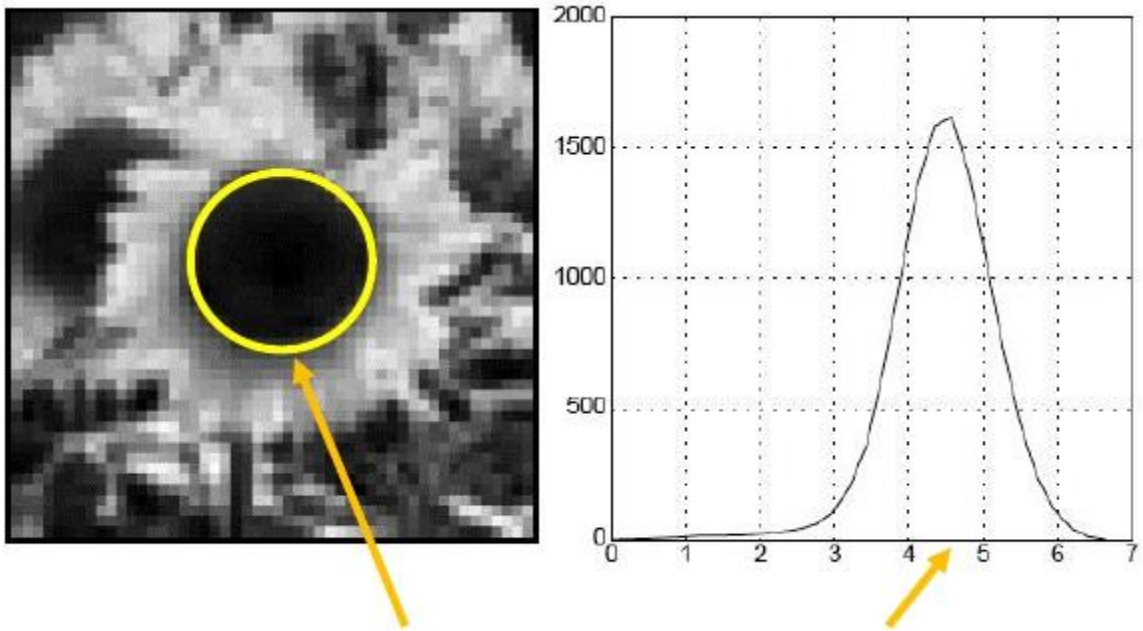
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = "blob" detector



Characteristic Scale

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

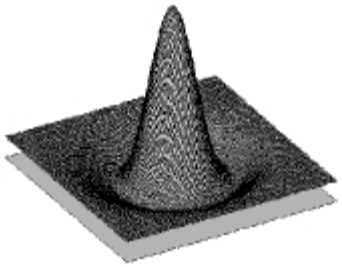


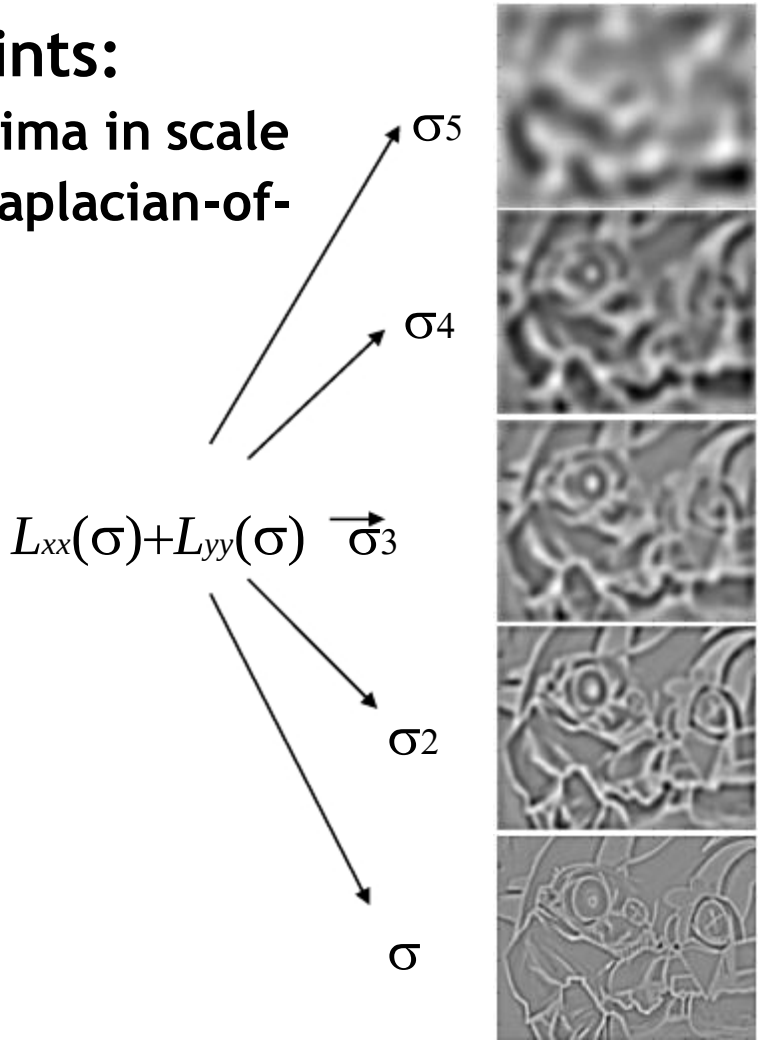
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

Slide credit: Svetlana Lazebnik

Laplacian-of-Gaussian (LoG)

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

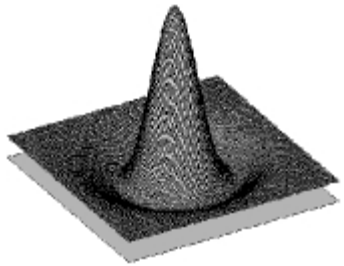


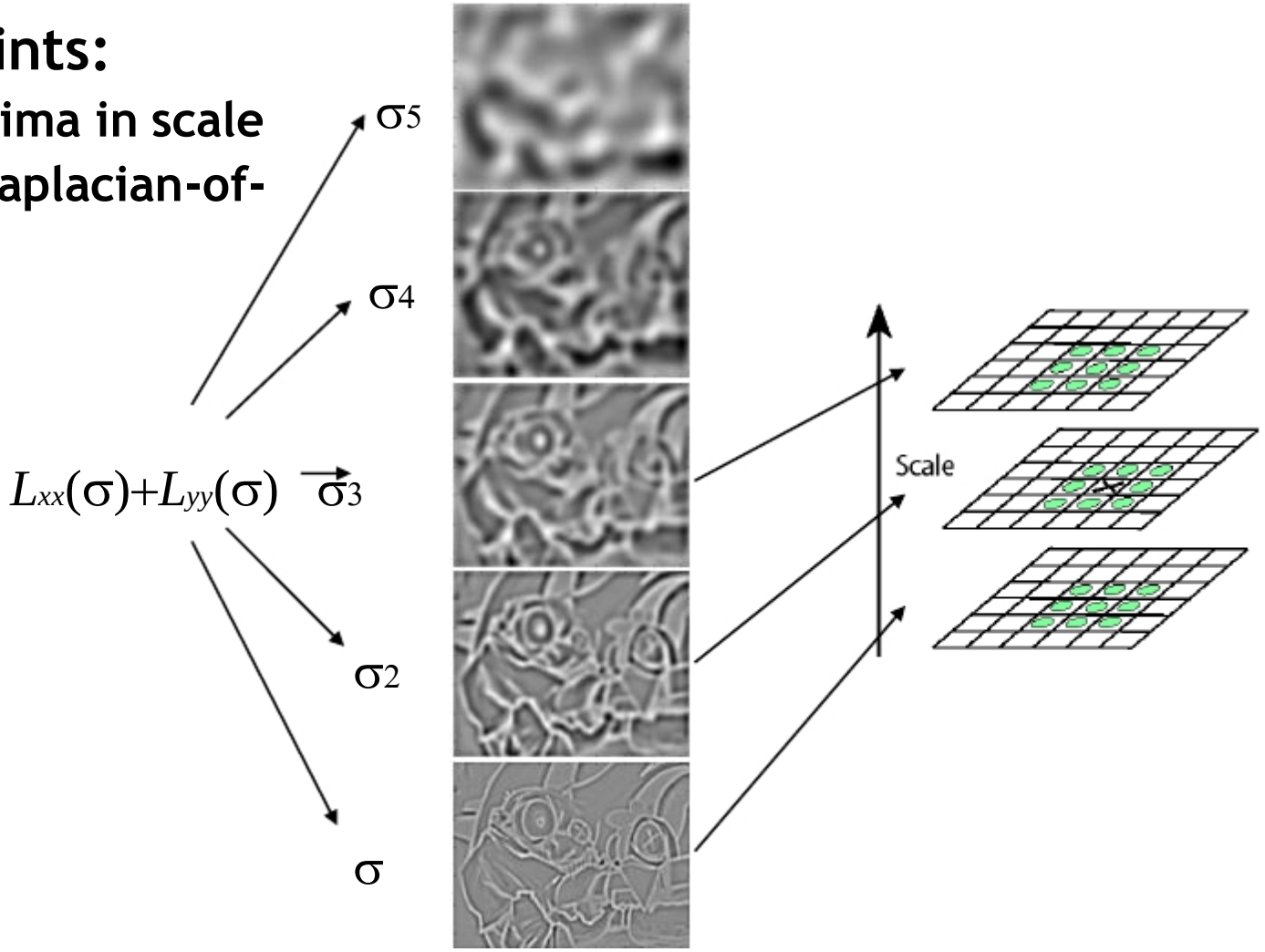


Slide adapted from Krystian Mikolajczyk

Laplacian-of-Gaussian (LoG)

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

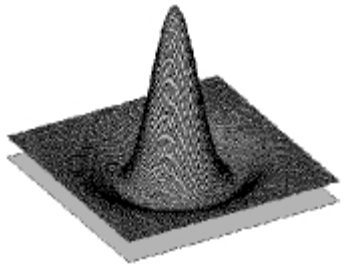


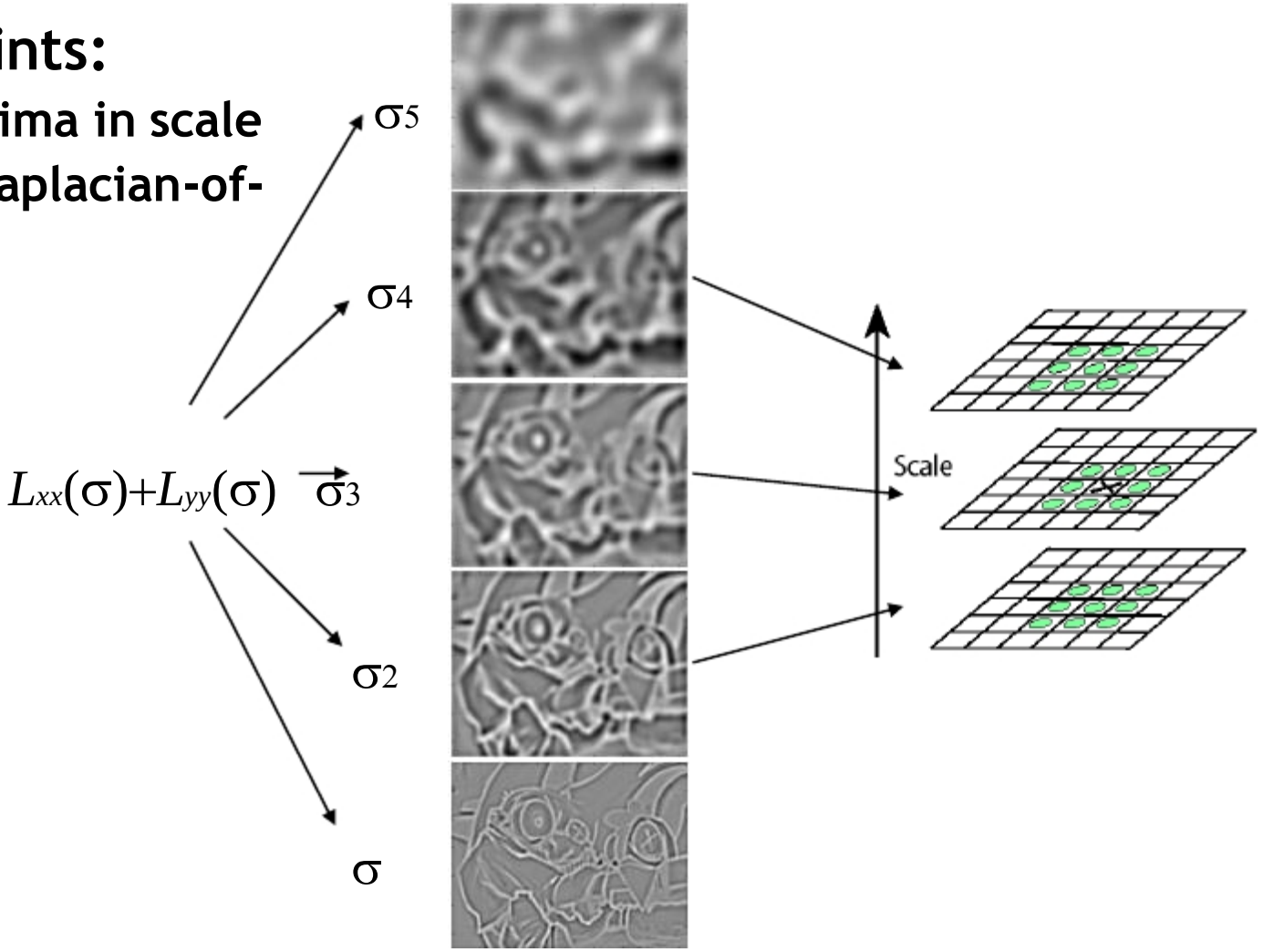


Slide adapted from Krystian Mikolajczyk

Laplacian-of-Gaussian (LoG)

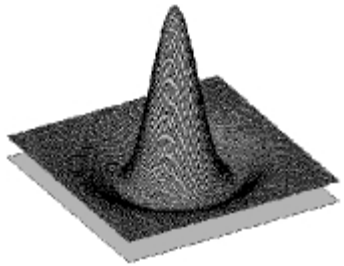
- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**

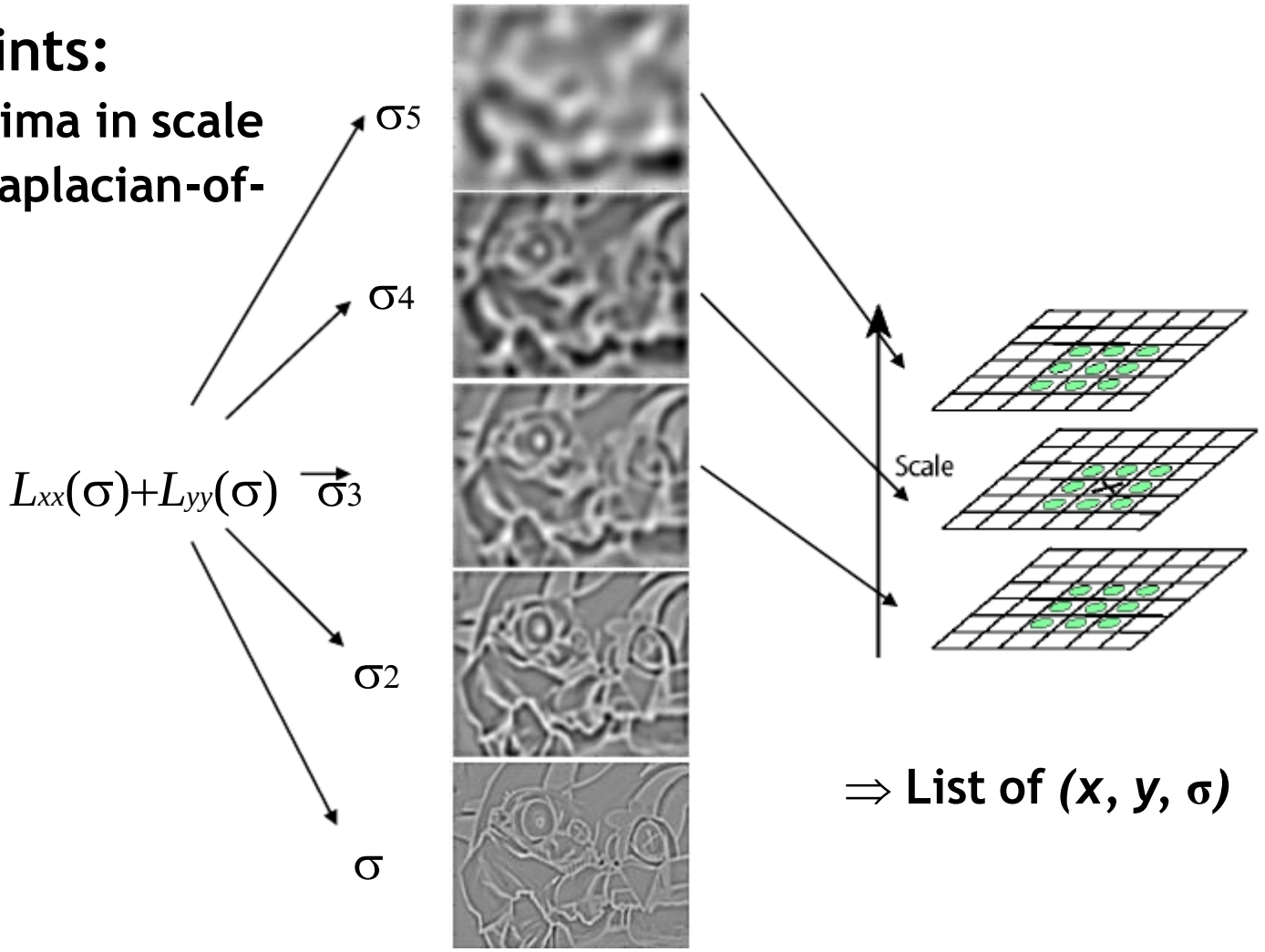




Laplacian-of-Gaussian (LoG)

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





⇒ **List of *(x, y, σ)***

Slide adapted from Krystian Mikolajczyk

Difference-of-Gaussian (DoG)

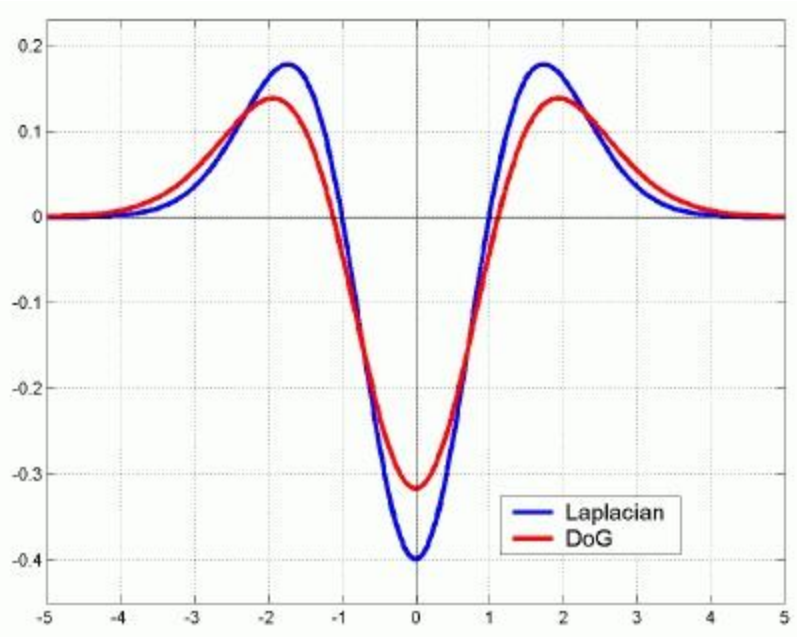
- **We can efficiently approximate the Laplacian with a difference of Gaussians:**

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

(Laplacian)

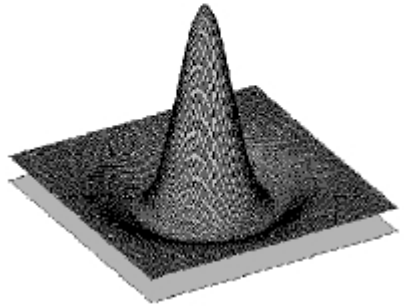
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



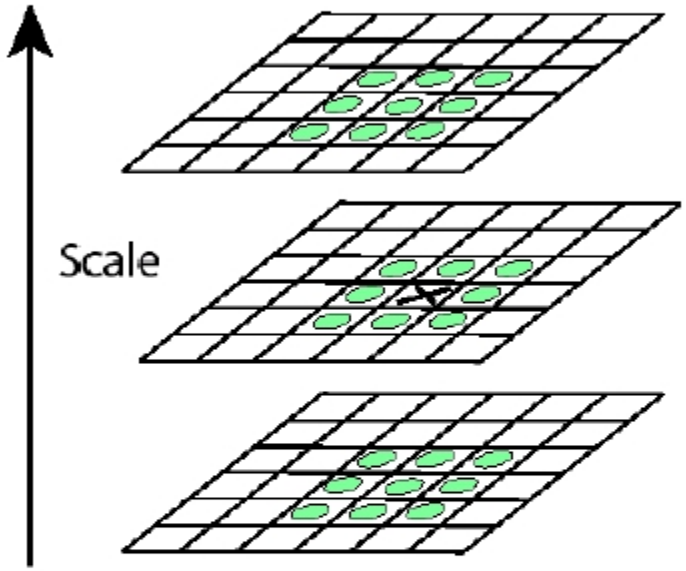
Difference-of-Gaussian (DoG)

- 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.



Key point localization with DoG

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

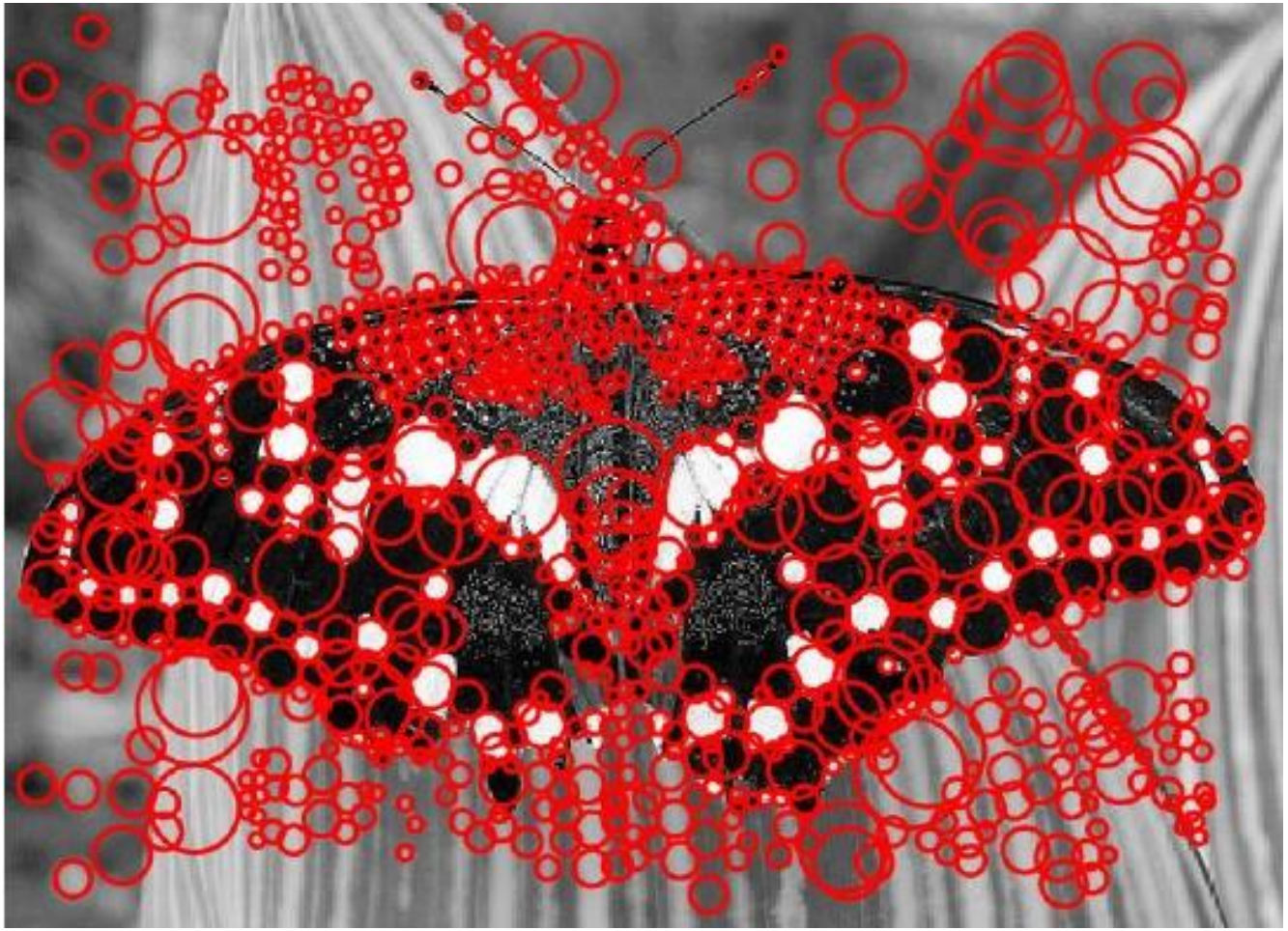


Candidate keypoints:
list of (x,y,σ)

Slide credit: David Lowe

Results

Results

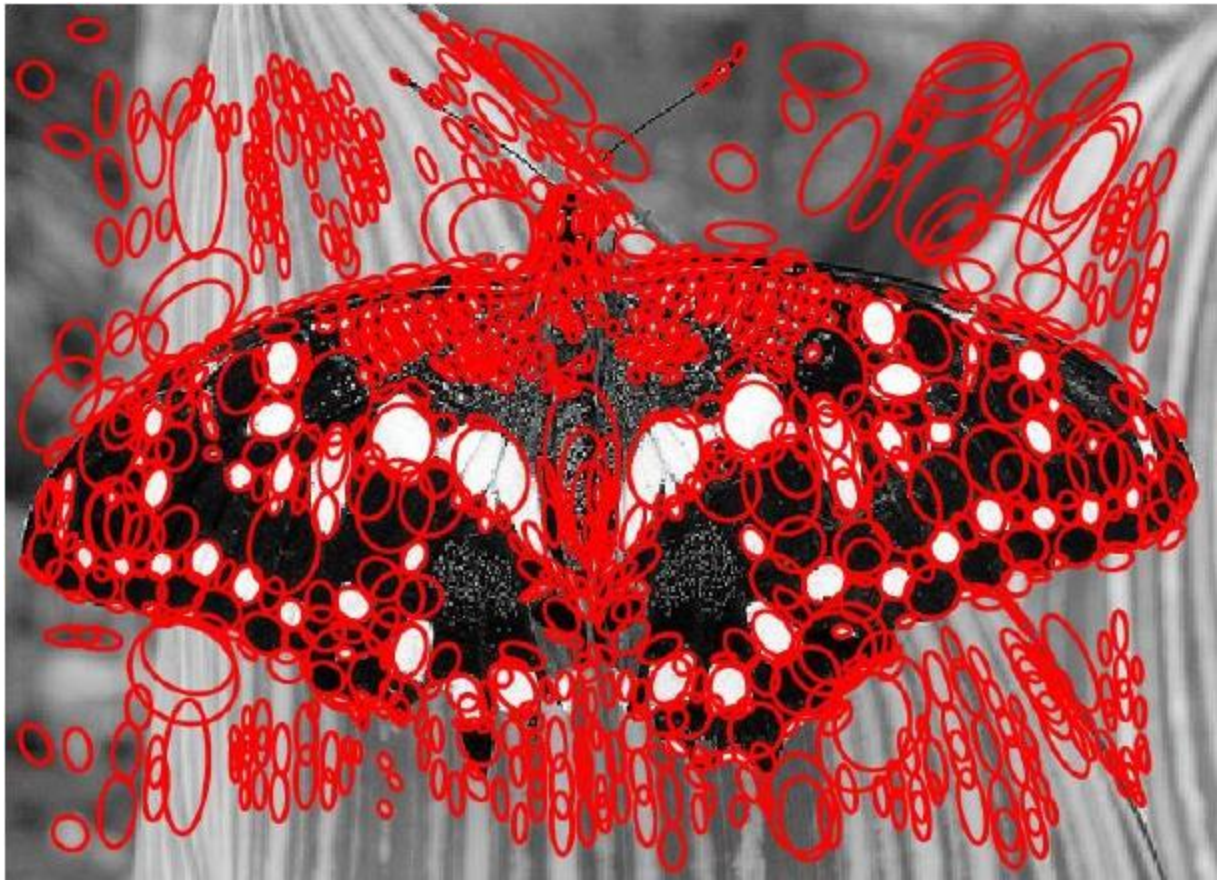


阅读文献

- **David G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, IJCV2004. 第3、4章**

实现“仿射不变”？

- …



实现“仿射不变”？

- …


ALGEBRA
LONGMAN
DICTIONARY OF
CONTEMPORARY
ENGLISH
ATTONI