Computer Vision - Features

Junjie Cao @ DLUT

Spring 2018

- 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

by Diva Sian

by swashford

Slightly different color and viewpoint, but still easy

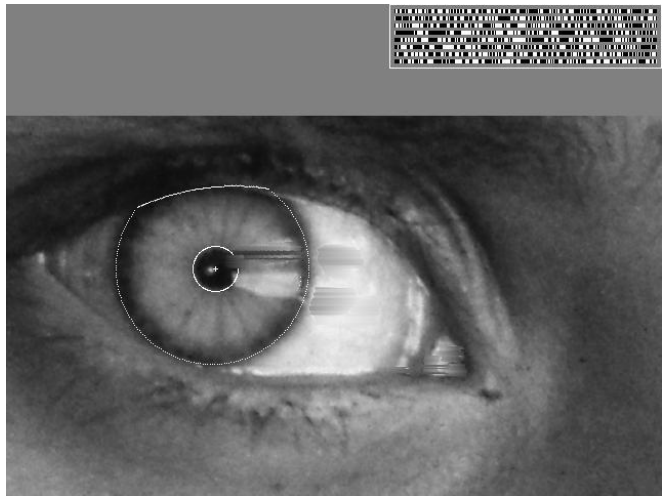
Harder case

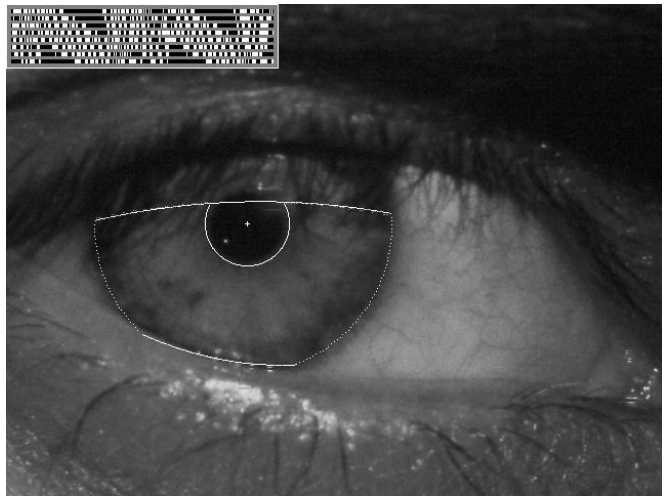
by Diva Sian

by scgbt

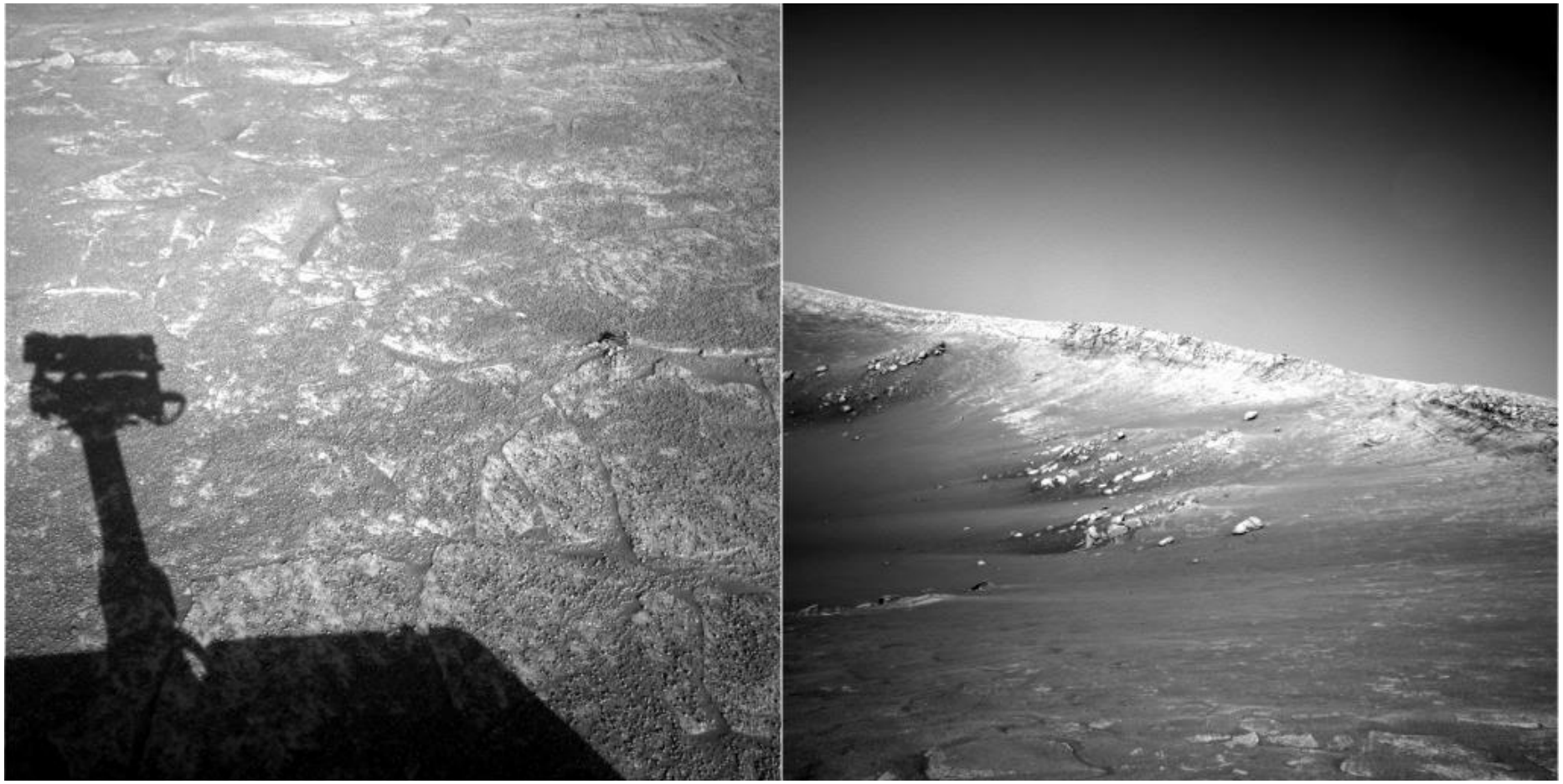
Even harder case

“*How the Afghan Girl was Identified by Her Iris Patterns*” Read the [story]



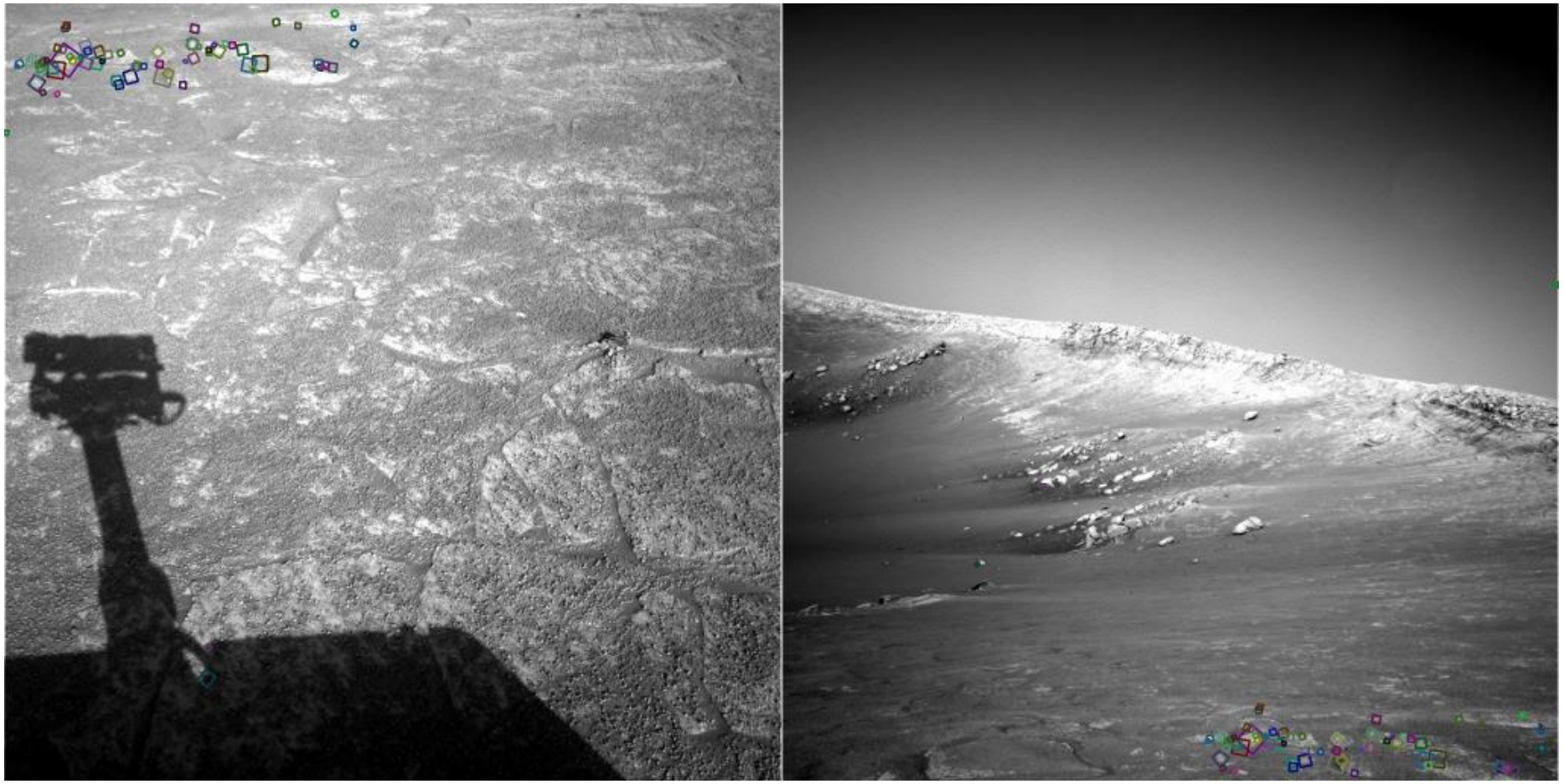


Harder still?



NASA Mars Rover images

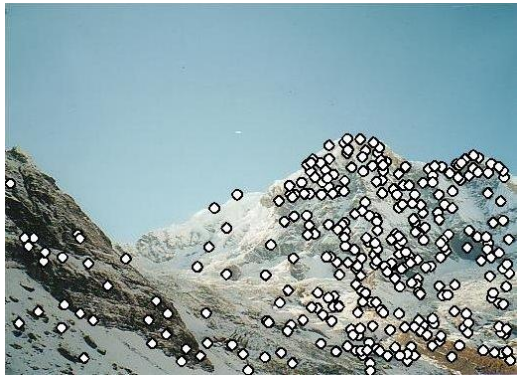
Answer below (look for tiny colored squares…)



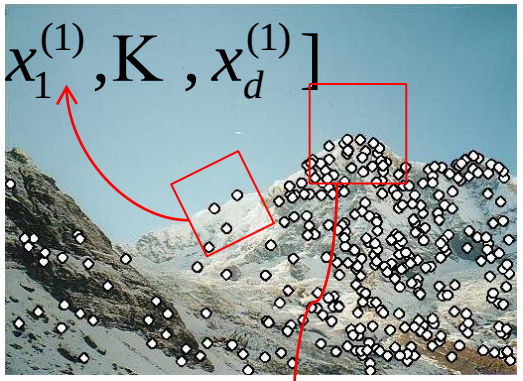
NASA Mars Rover images
with **SIFT** feature matches
Figure by Noah Snavely

Local features: main components

1) **Detection:** Identify the interest points



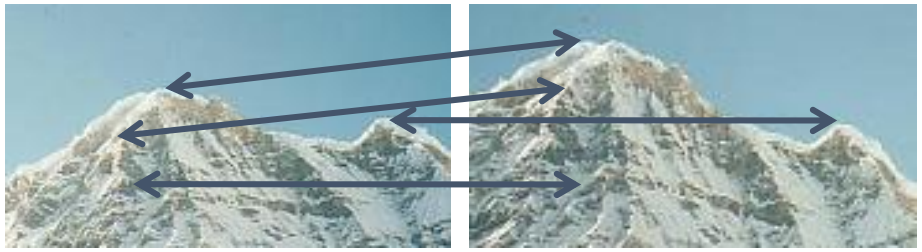
2) **Description**: Extract vector feature descriptor surrounding each interest point.



$$\mathbf{x}_1 = [x_1^{(1)}, \mathrm{K}, x_d^{(1)}]$$

$$\mathbf{x}_2 = [x_1^{(2)}, \mathrm{K}, x_d^{(2)}]$$

3) **Matching:** Determine correspondence between descriptors in two views



Features

All is Vanity, by C. Allan Gilbert, 1873-1929

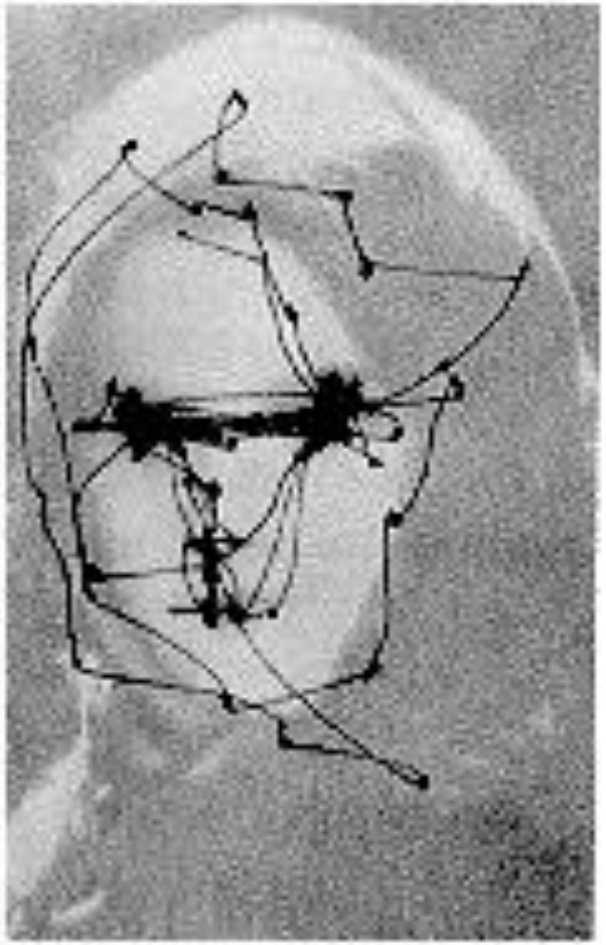
Readings

- Richard Szeliski, Ch 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_

Today

- Local invariant features
  - Detection of interest points
    - (Harris corner detection)
    - Scale invariant blob detection: LoG
  - Description of local patches
    - SIFT : Histograms of oriented gradients
    - Bag of Feature
- A brief introduction of deep learning

Human eye movements

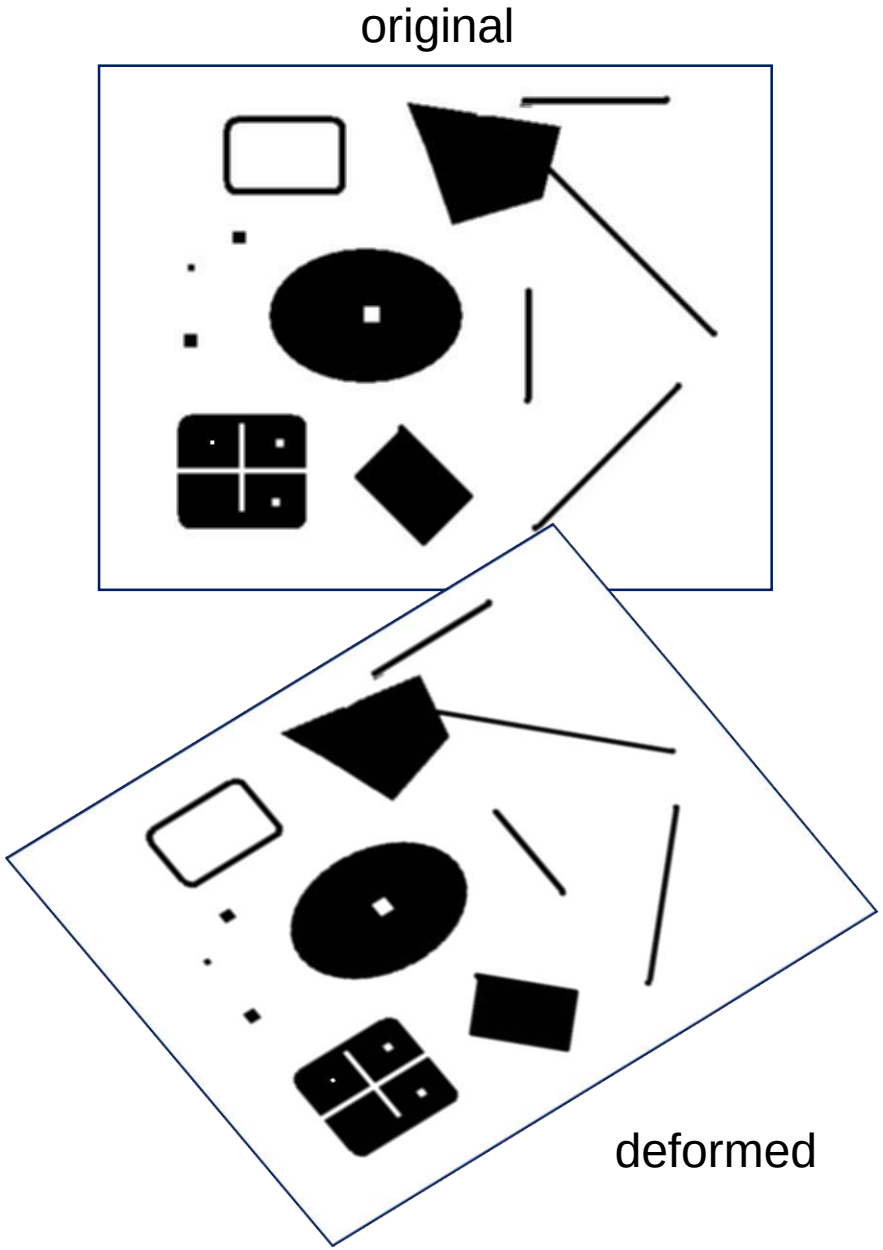


Yarbus eye tracking

What catches your interest?

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?**



Goal: interest operator repeatability

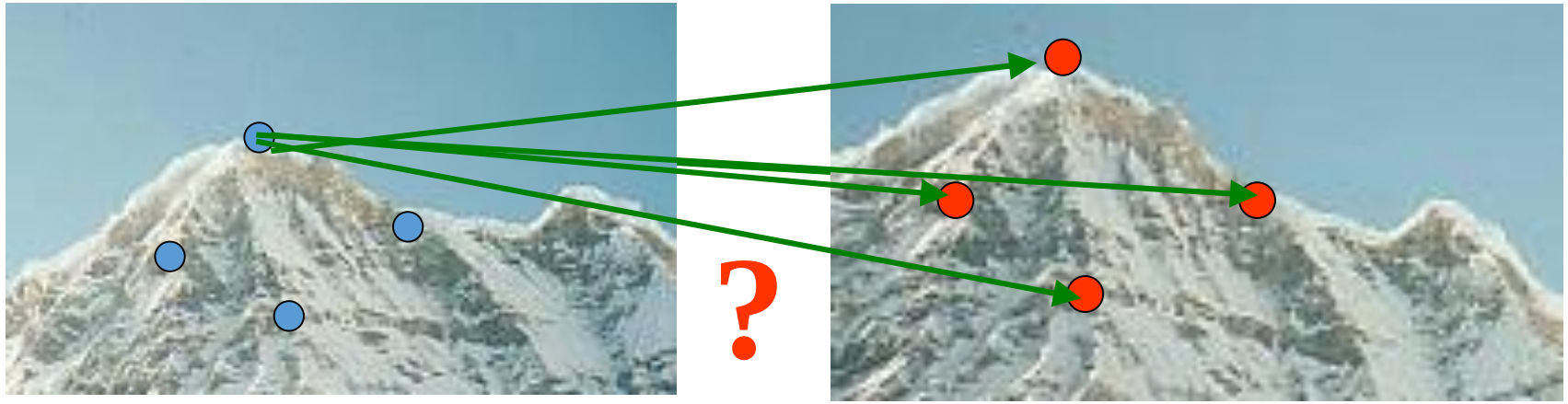
- **We want to detect (at least some of) the same points in both images.**

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.**



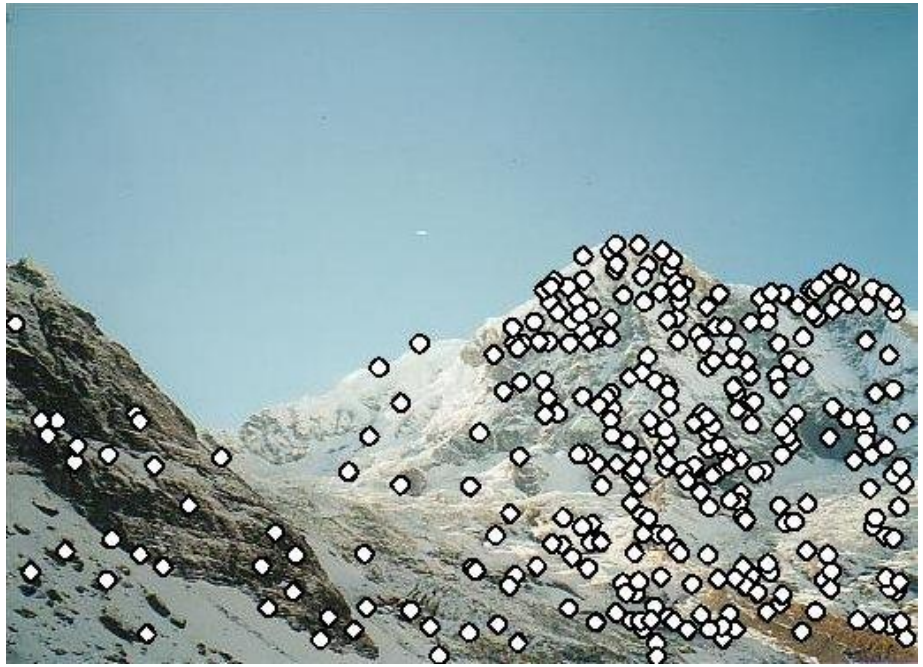
- **Must provide some invariance to geometric and photometric differences between the two views.**

Local features: main components

1) **Detection:** Identify the interest points

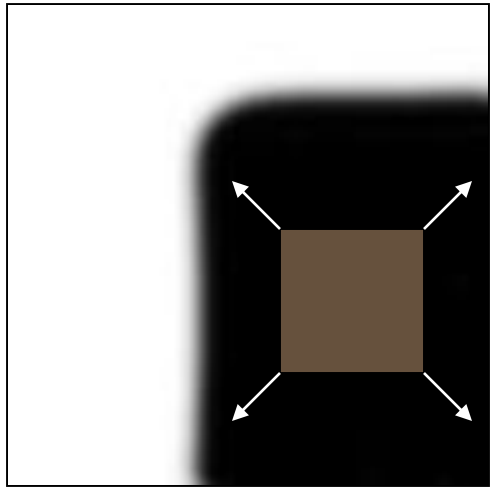
2) **Description:** Extract vector feature descriptor surrounding each interest point.

3) **Matching:** Determine correspondence between descriptors in two views

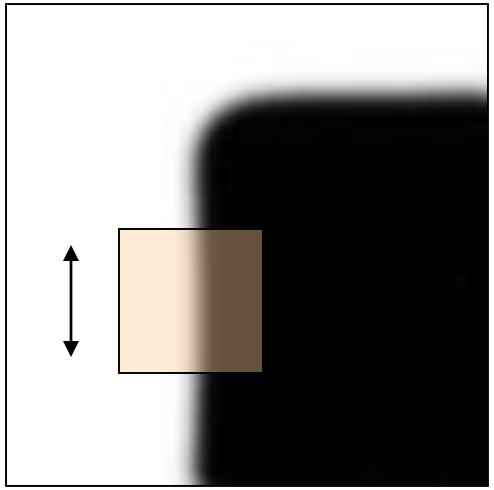


Corners

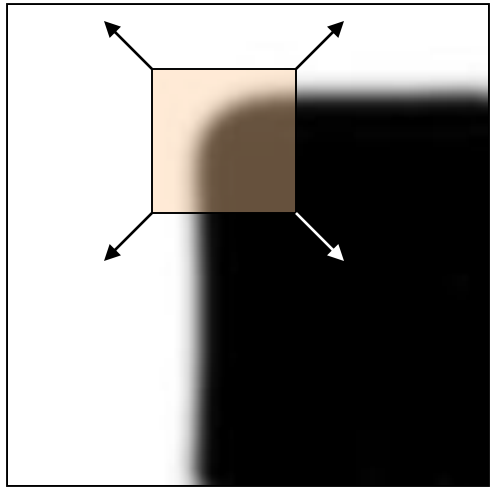
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region: no change in all directions



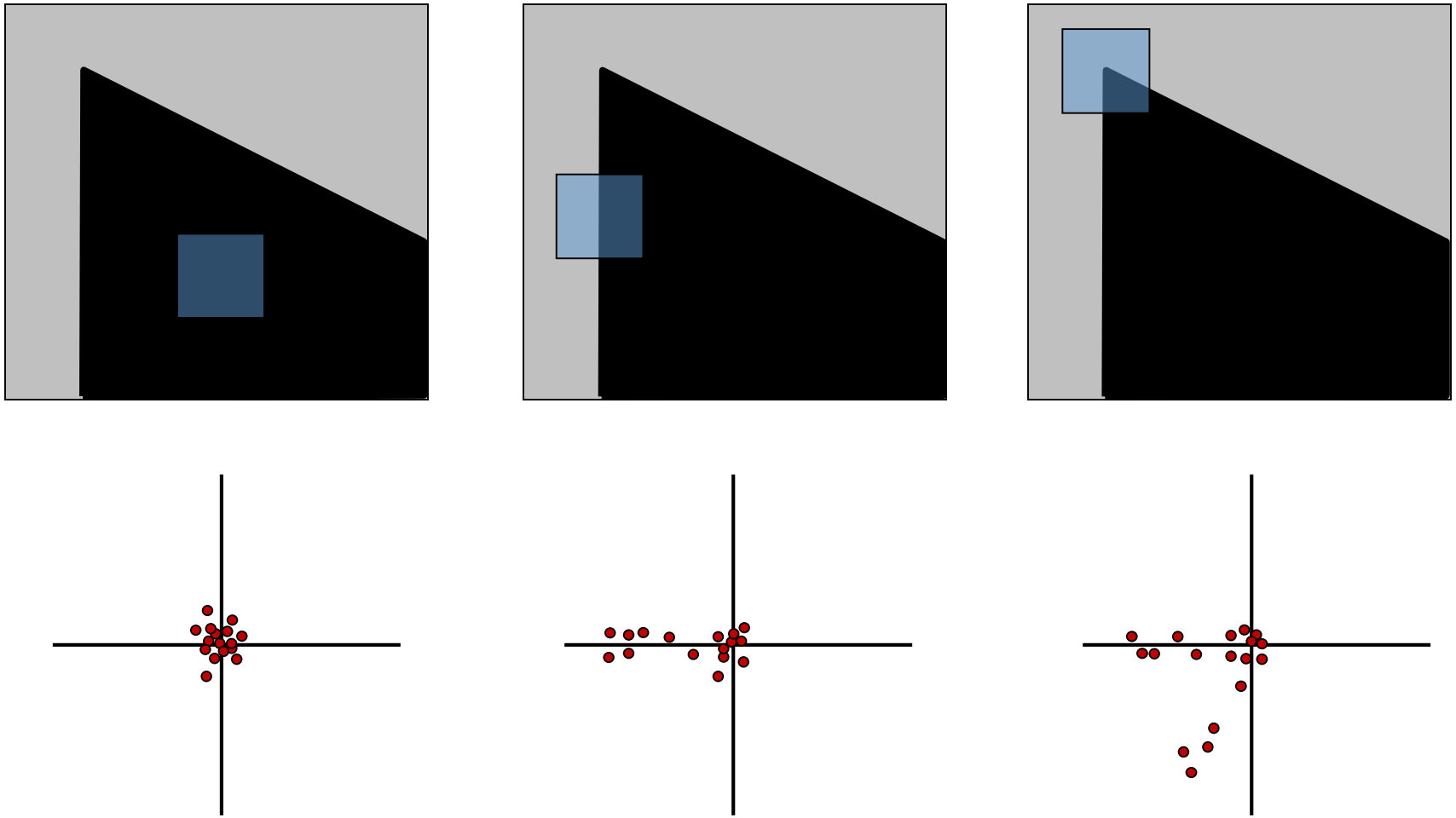
"edge": no change along the edge direction



"corner": significant change in all directions

Source: A. Efros

Let’s look at the gradient distributions



Principal Component Analysis

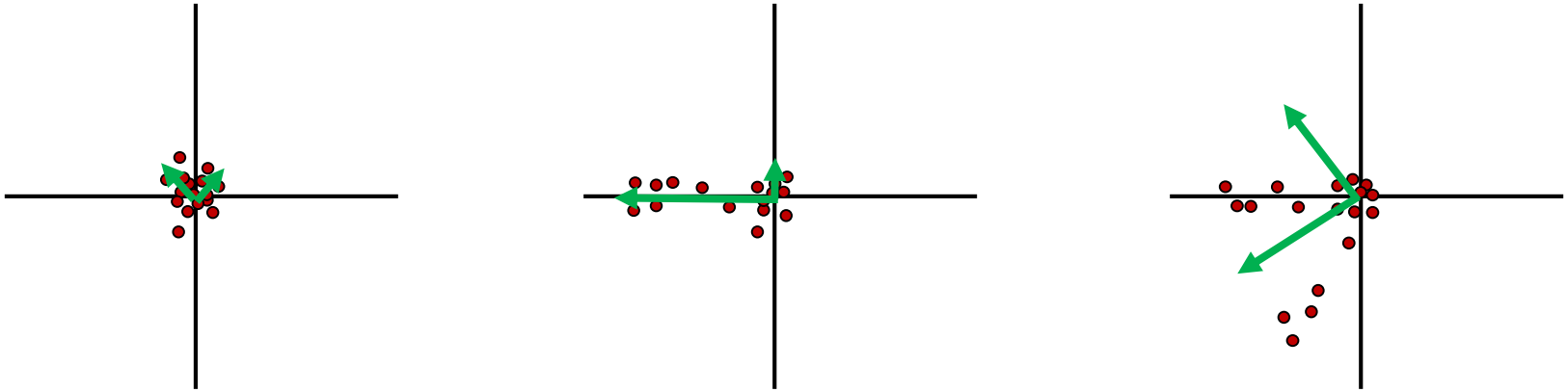
Principal component is the direction of highest variance.

Next, highest component is the direction with highest variance *orthogonal* to the previous components.

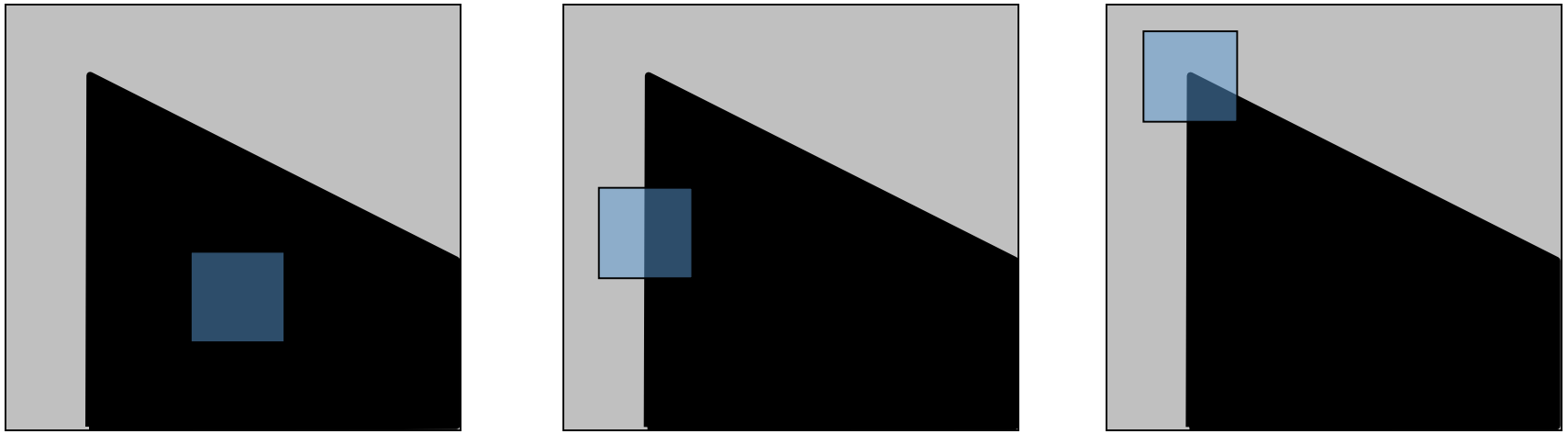
How to compute PCA components:

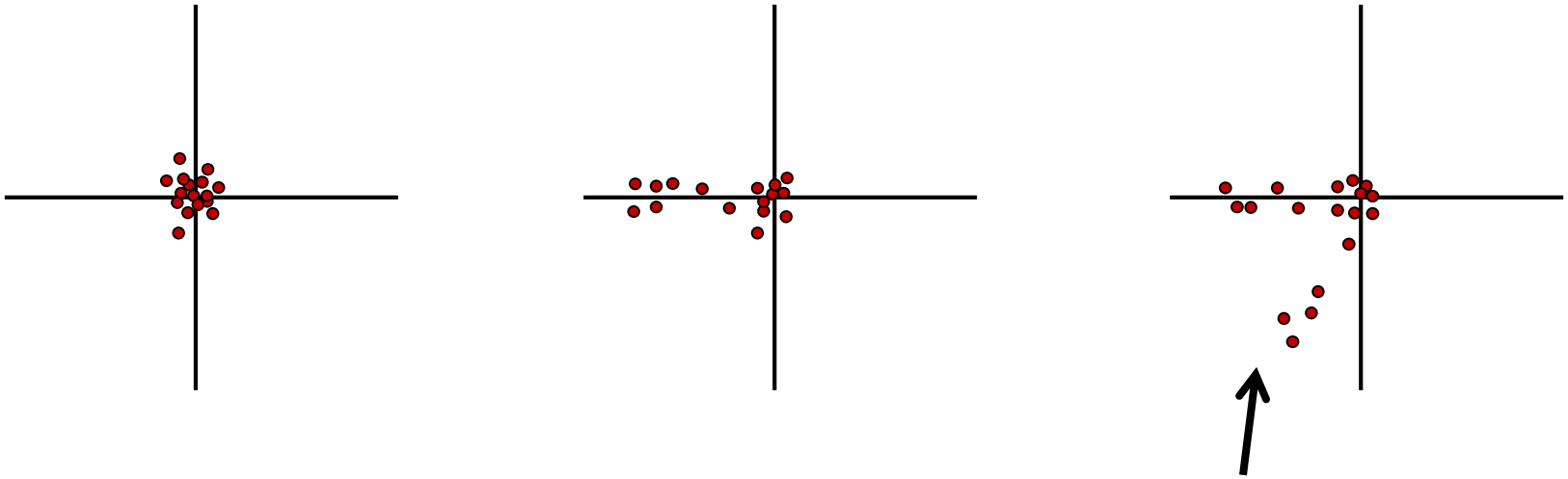
1.Subtract off the mean for each data point.
2.Compute the covariance matrix.
3.Compute eigenvectors and eigenvalues.
4.The components are the eigenvectors ranked by the eigenvalues.

$$Hx = \lambda x$$



Corners have …



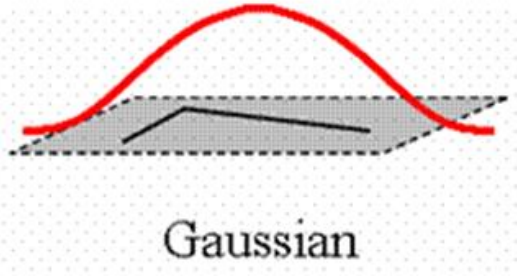


Both eigenvalues are large!

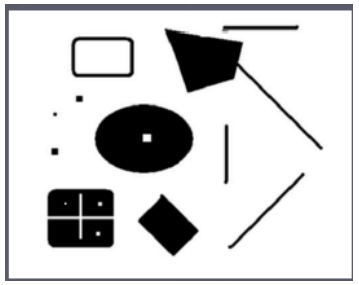
Recall: Second Moment Matrix or Harris Matrix

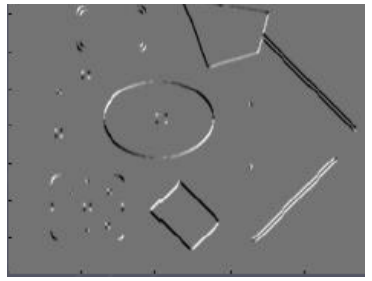
$$H = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Typically Gaussian weights



2 x 2 matrix of image derivatives smoothed by Gaussian weights.





Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

- First compute I_x, I_y, and $I_x I_y$ as 3 images; then apply Gaussian to each.
- OR, first apply the Gaussian and the compute the derivatives.

The math

To compute the eigenvalues:

1. Compute the Harris matrix over a window.

$$H = \sum_{(u,v)} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \qquad I_x = \frac{\partial f}{\partial x}, I_y = \frac{\partial f}{\partial y}$$

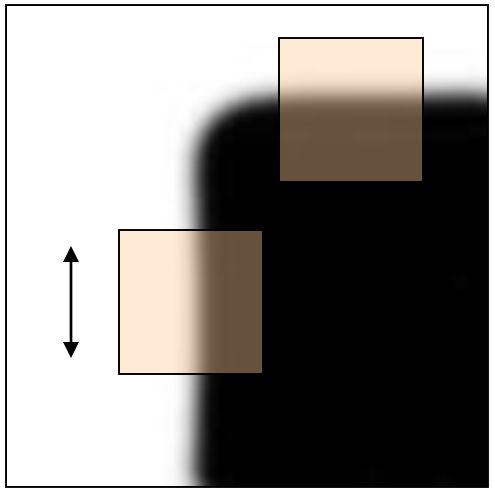
Typically Gaussian weights

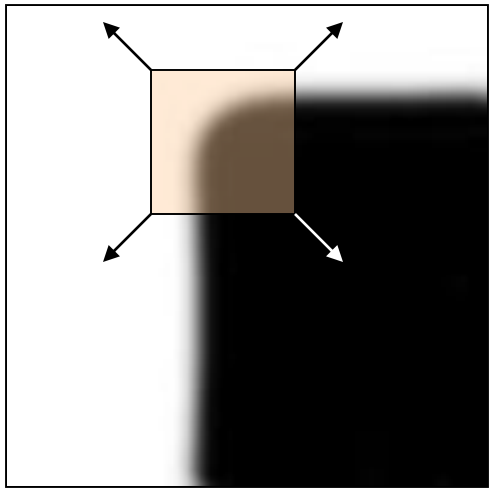
$$\begin{bmatrix} \Sigma\text{smoothed } I_x^2 & \Sigma\text{smoothed } I_x I_y \\ \Sigma\text{smoothed } I_x I_y & \Sigma\text{smoothed } I_y^2 \end{bmatrix}$$

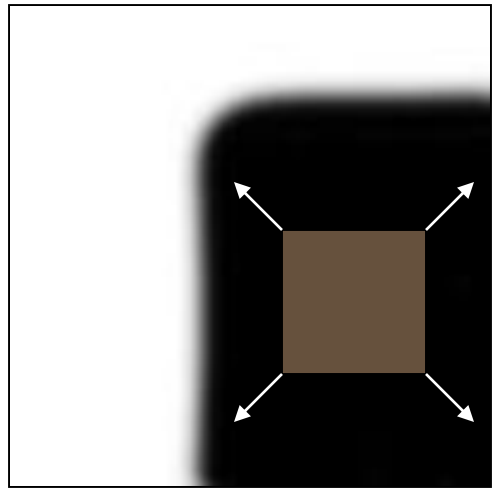
2. Compute eigenvalues from that.

$$H = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \lambda_{\pm} = \frac{1}{2}\left((a+d) \pm \sqrt{4bc + (a-d)^2}\right)$$

Corner Response Function







"edge":

$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$

"corner":

λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$;

"flat" region

λ_1 and λ_2 are small;

One way to score the cornerness:

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$\lambda_1 \lambda_2 \rightarrow \det(M)$

$\lambda_1 + \lambda_2 \rightarrow \text{trace}(M)$

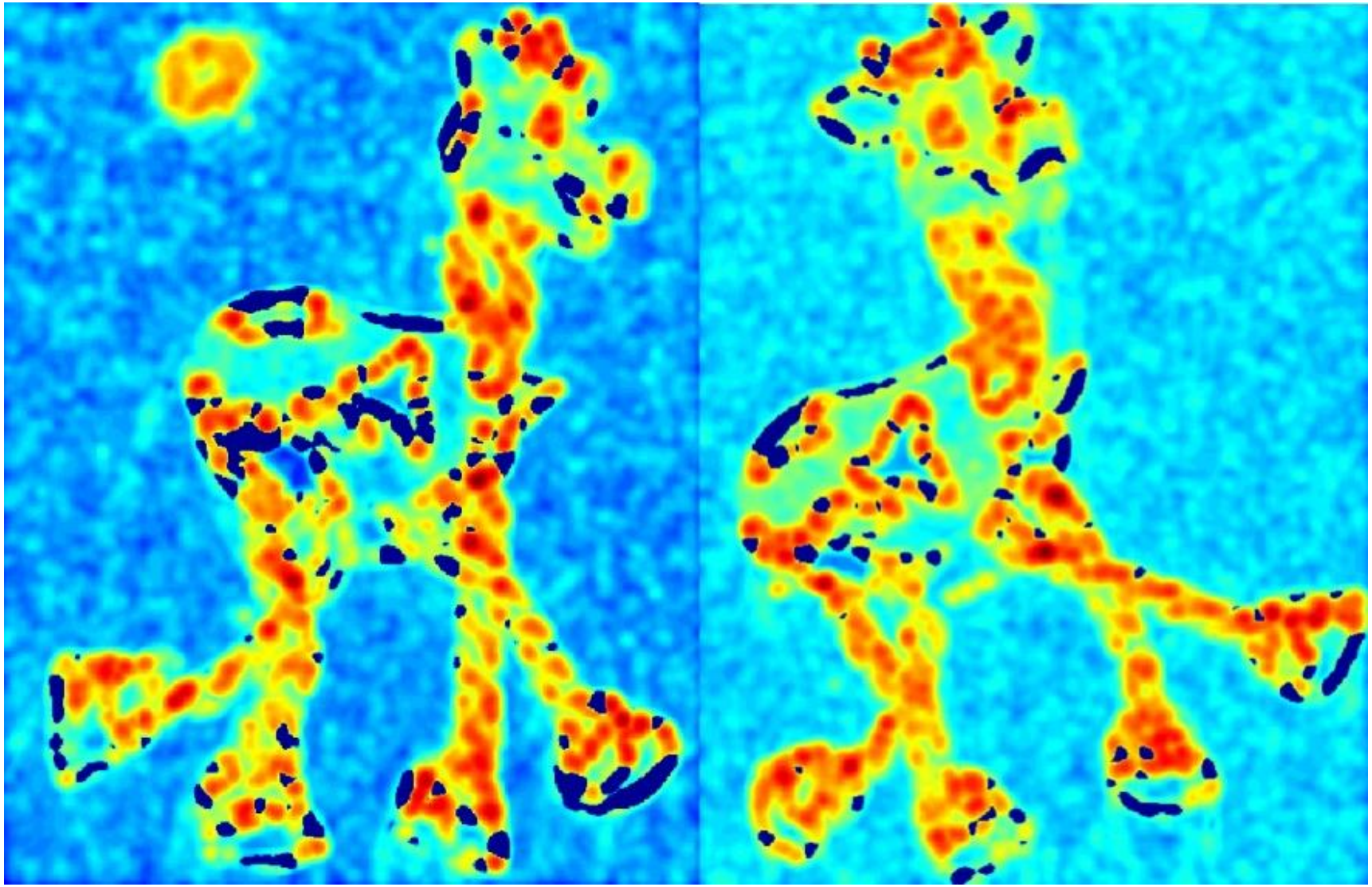
Harris corner detector

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

Harris Detector: Steps

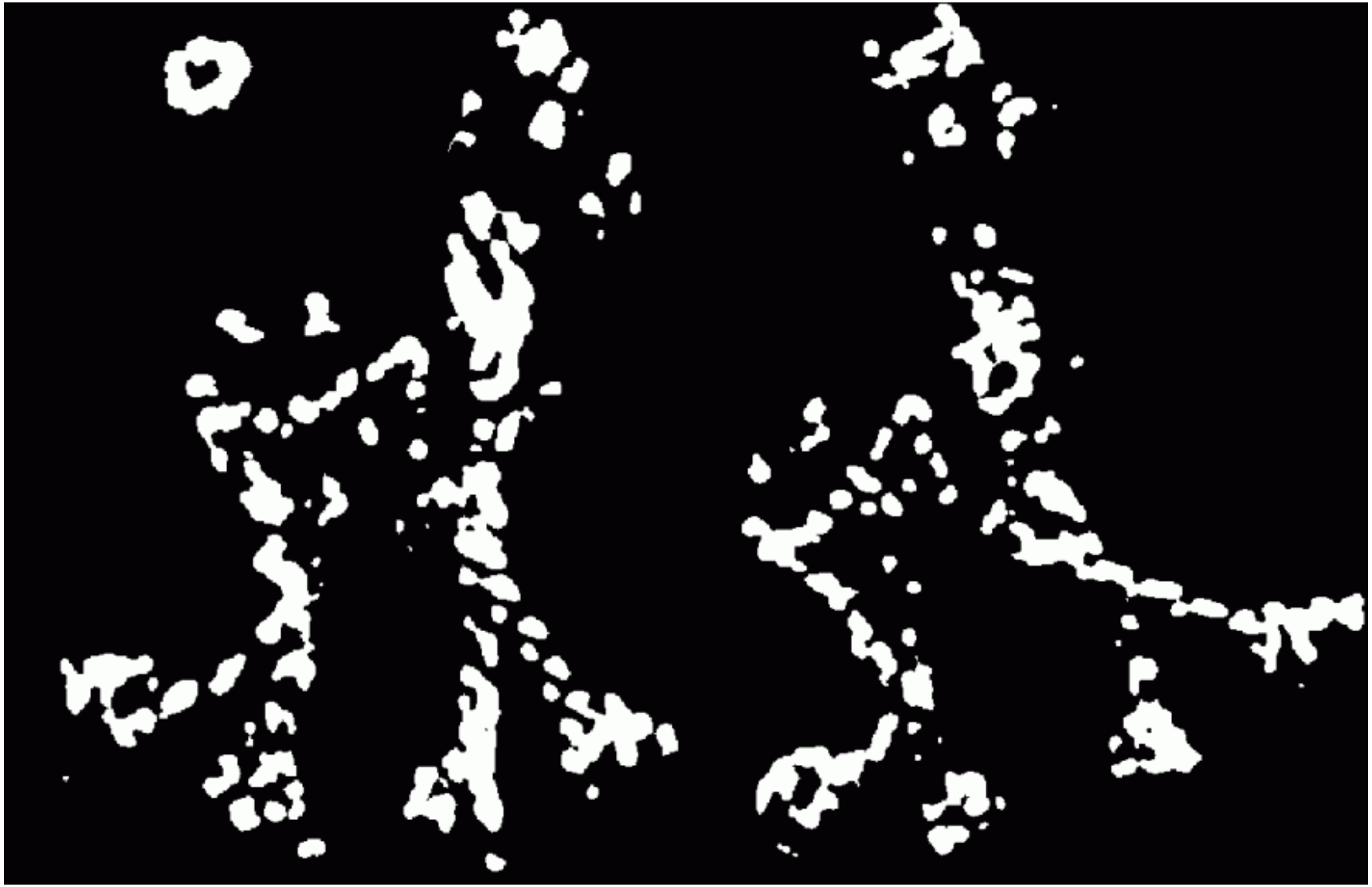
Harris Detector: Steps

Compute corner response *f*



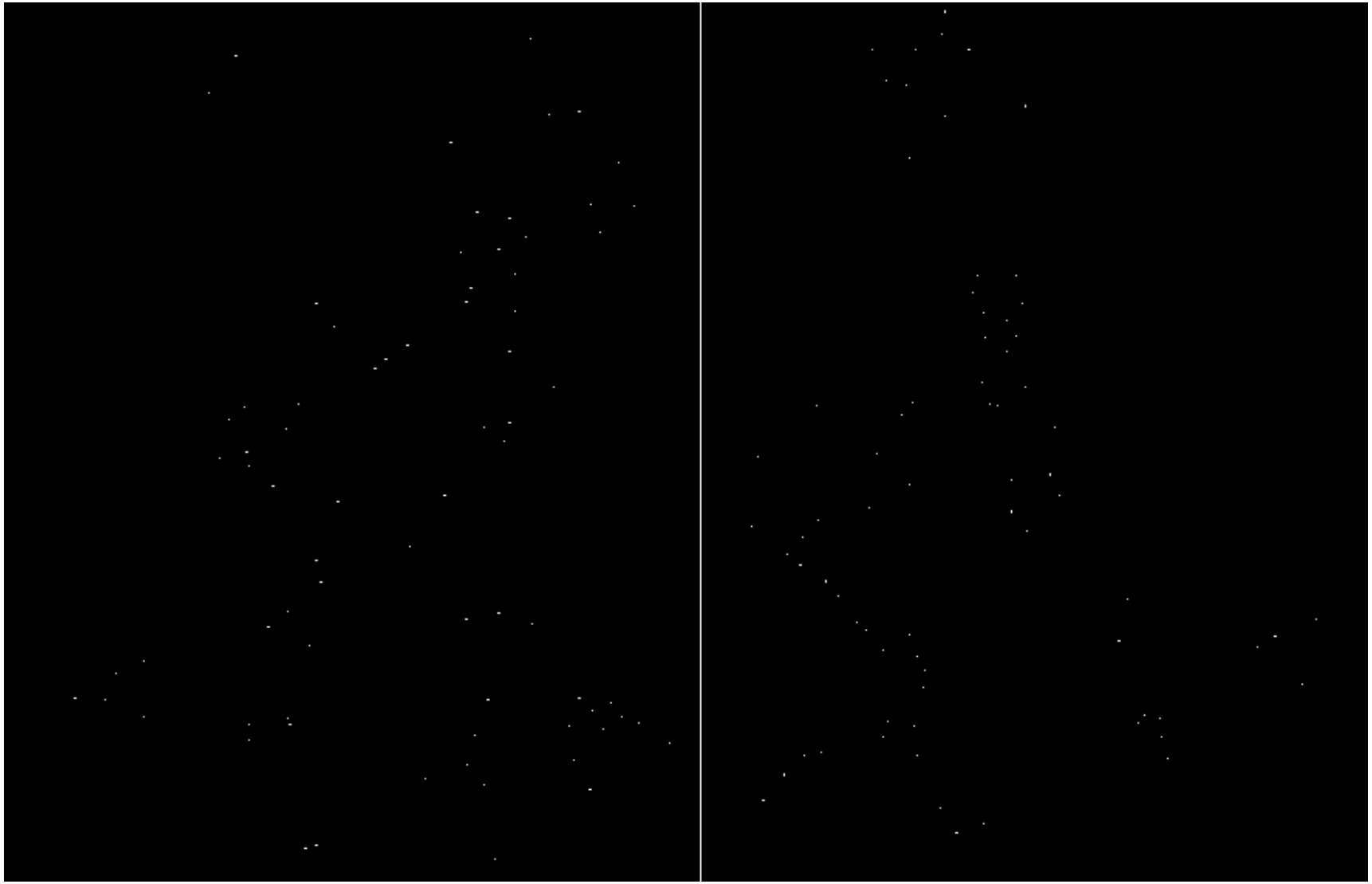
Harris Detector: Steps

Find points with large corner response: f > threshold

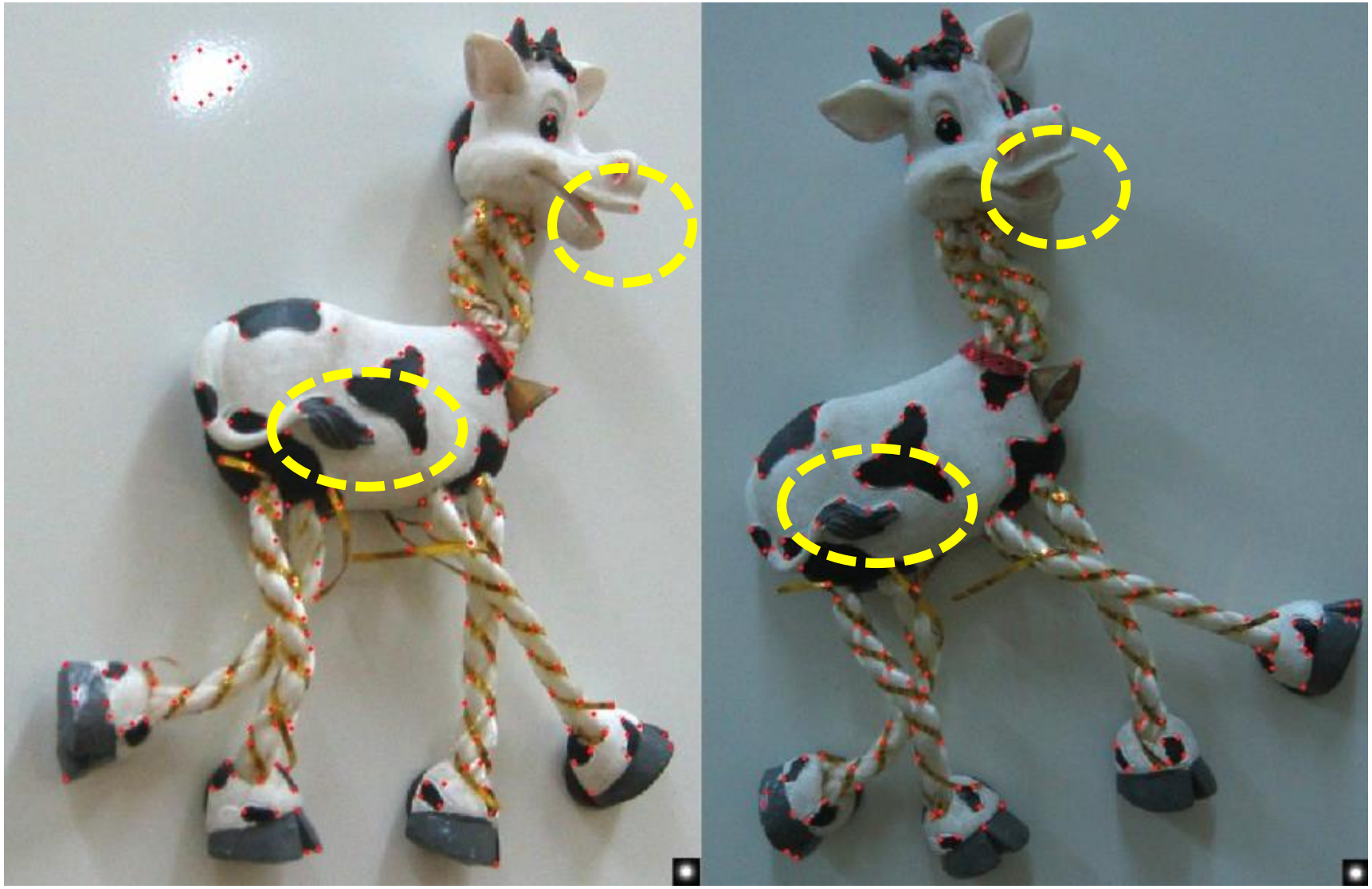


Harris Detector: Steps

Take only the points of local maxima of *f*



Harris Detector: Steps



Properties of the Harris corner detector

- Rotation invariant? Yes

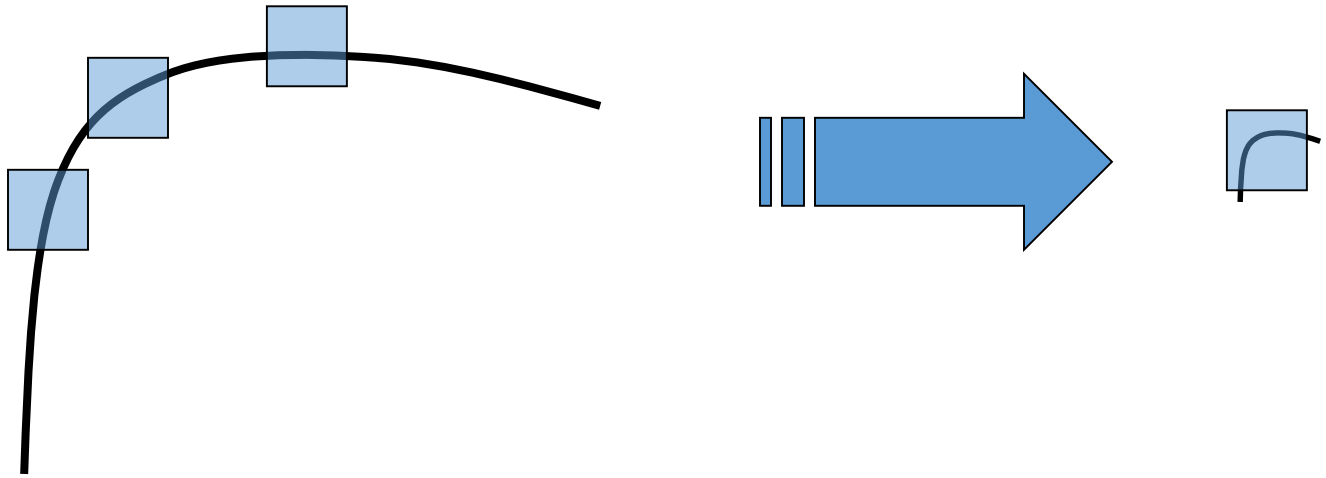
$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$$

- Scale invariant?

Properties of the Harris corner detector

- Rotation invariant? Yes
- Brightness invariant? Probably

- Scale invariant? No



All points will be classified as edges

Corner !

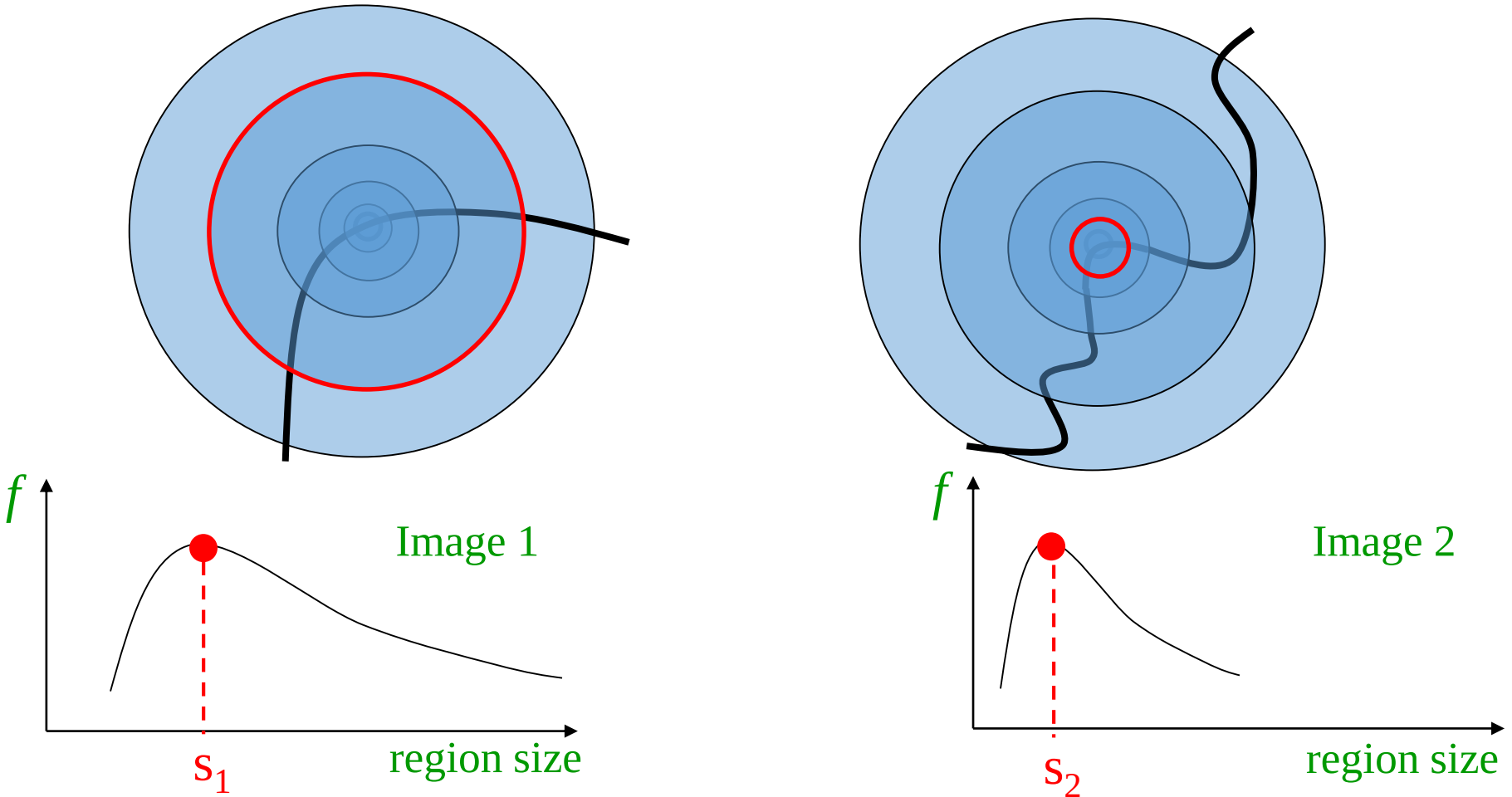
Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

Automatic scale selection

Intuition:

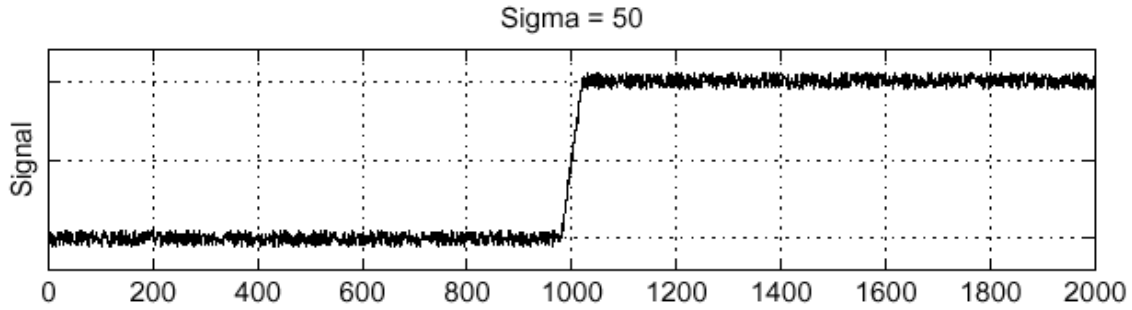
- Find scale that gives local maxima of some function *f* in both **position and scale**.



- What can be the “signature” function?

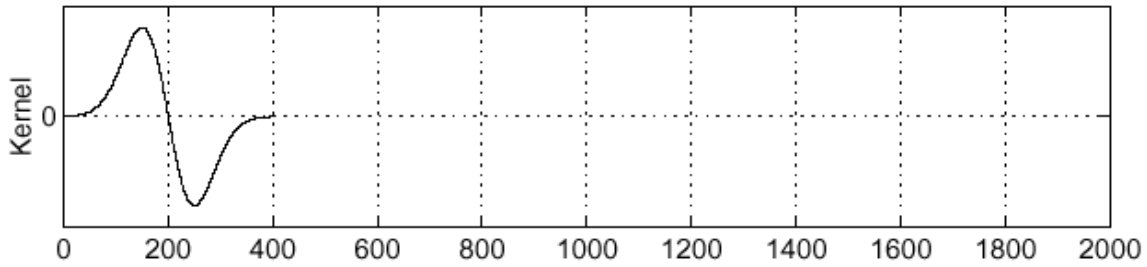
Recall: Edge detection

f



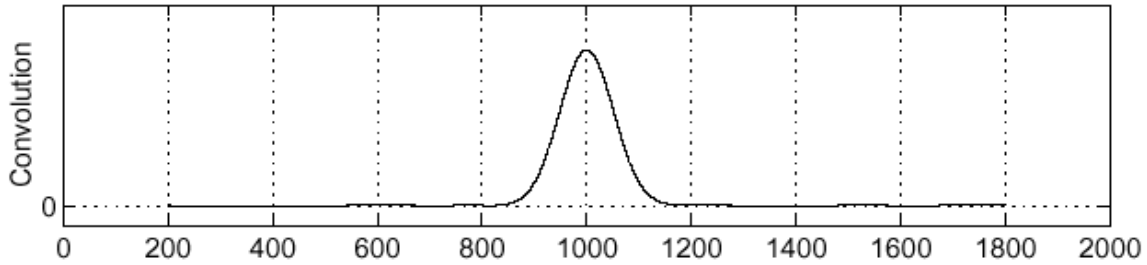
Edge

$\frac{d}{dx} g$



Derivative
of Gaussian

$f * \frac{d}{dx} g$

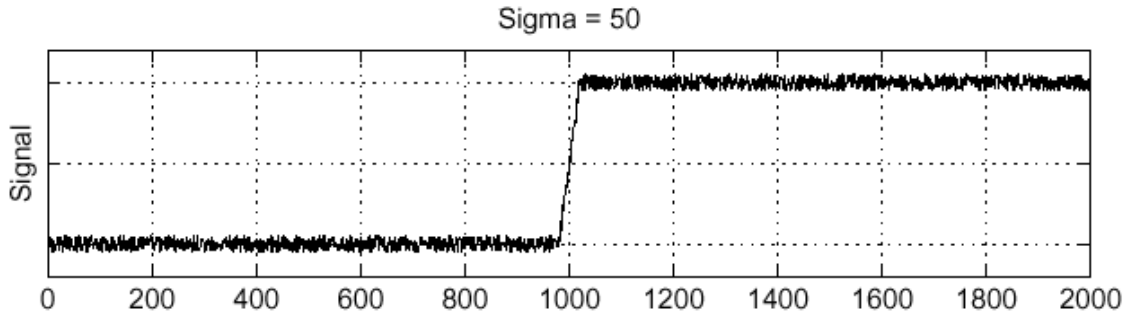


Edge = maximum
of derivative

Source: S. Seitz

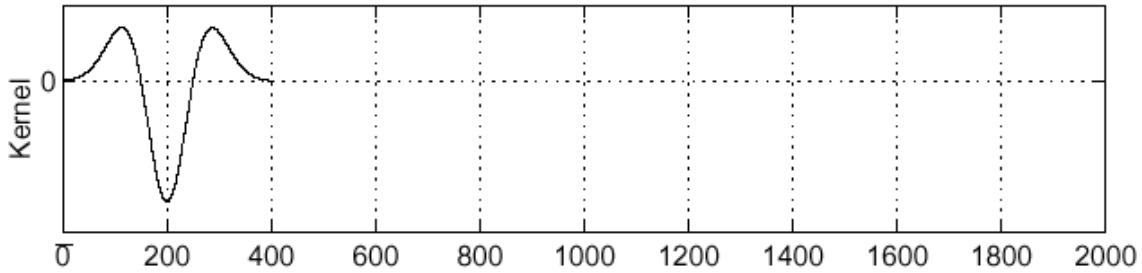
Recall: Edge detection

f



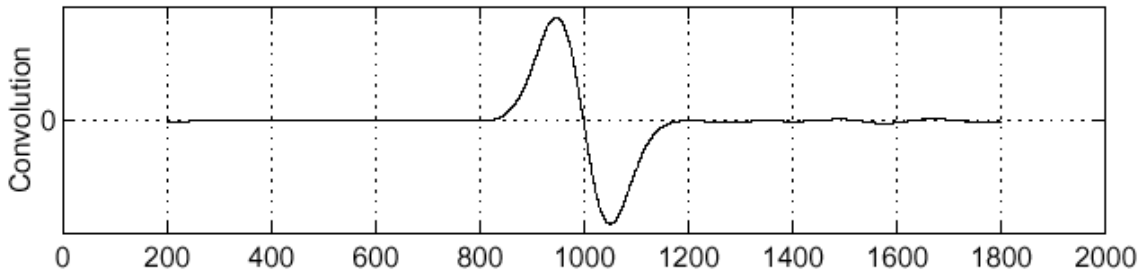
Edge

$\frac{d^2}{dx^2} g$



Second derivative of Gaussian (Laplacian)

$f * \frac{d^2}{dx^2} g$

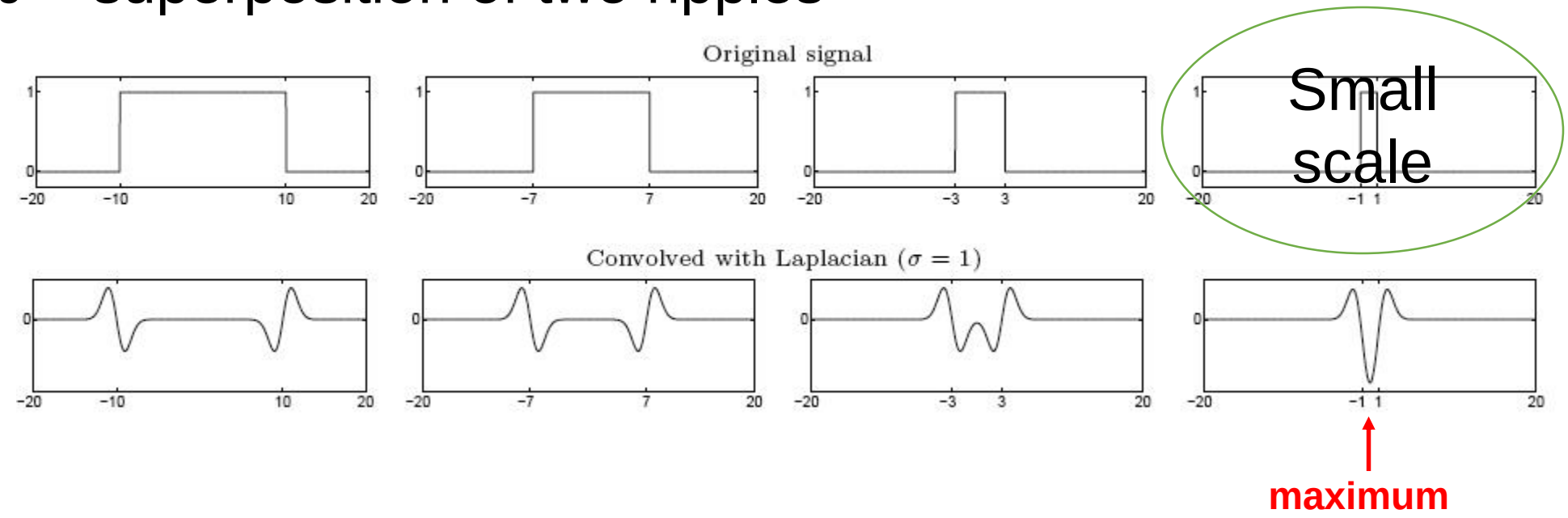


Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

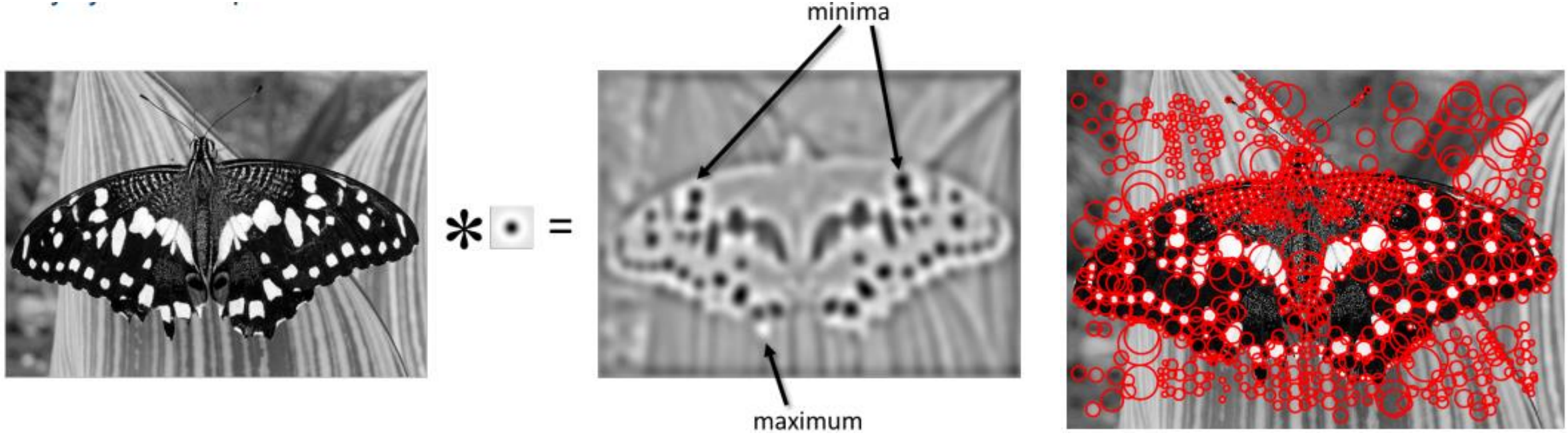


Spatial selection: the **magnitude** of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

Slide credit: Lana Lazebnik

Laplacian of Gaussian

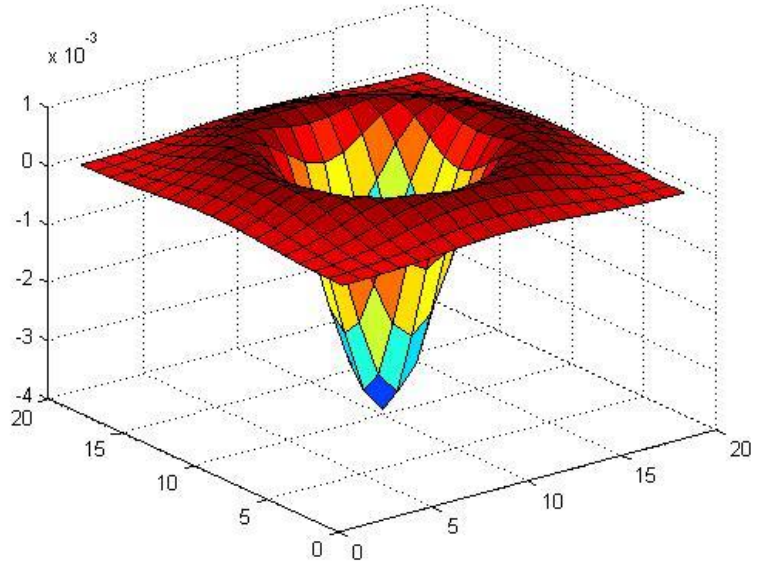
- Circularly symmetric operator for blob detection in 2D

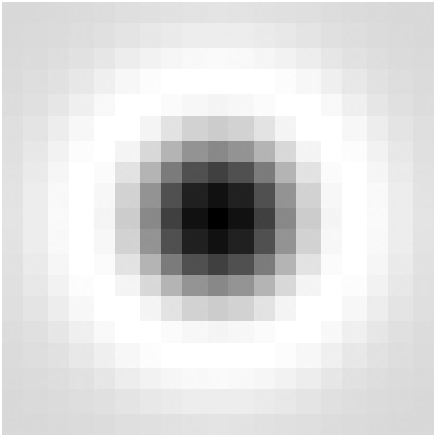


- Find maxima and minima of LoG operator in space and scale

Blob detection in 2D

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

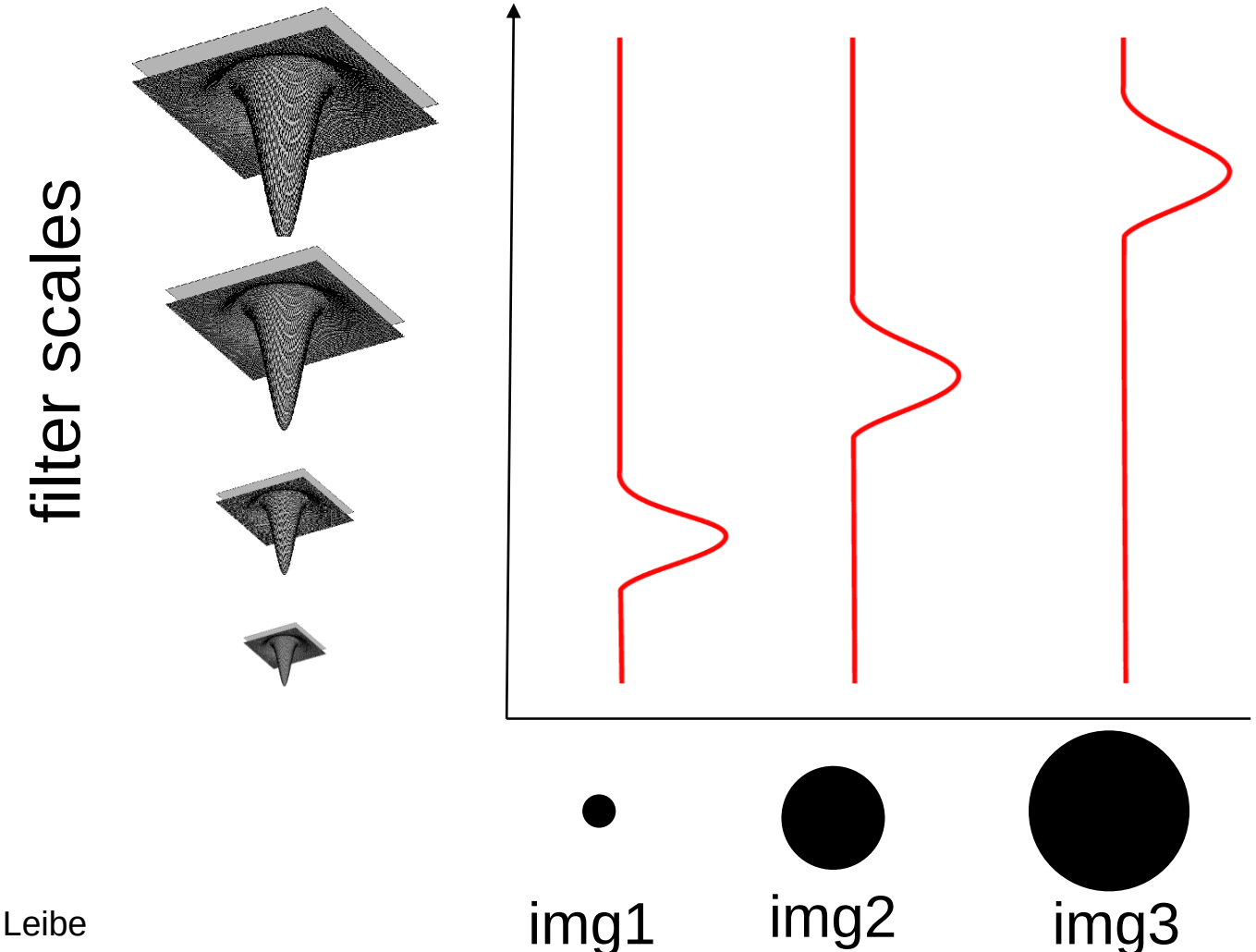




$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection in 2D: scale selection

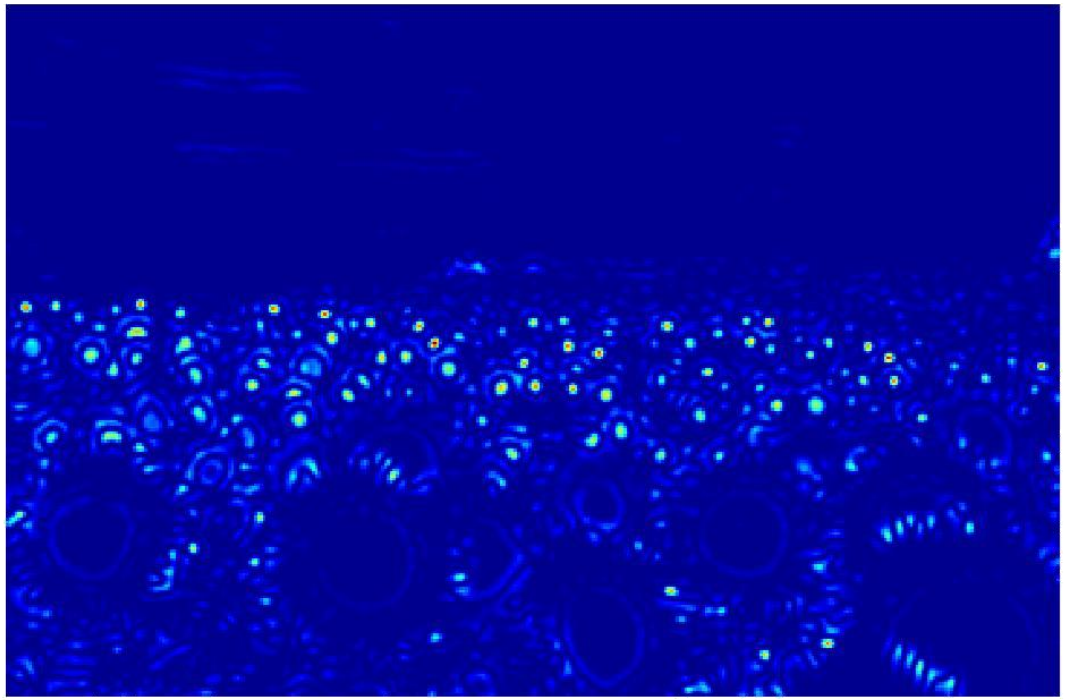
- Laplacian-of-Gaussian = “blob” detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$

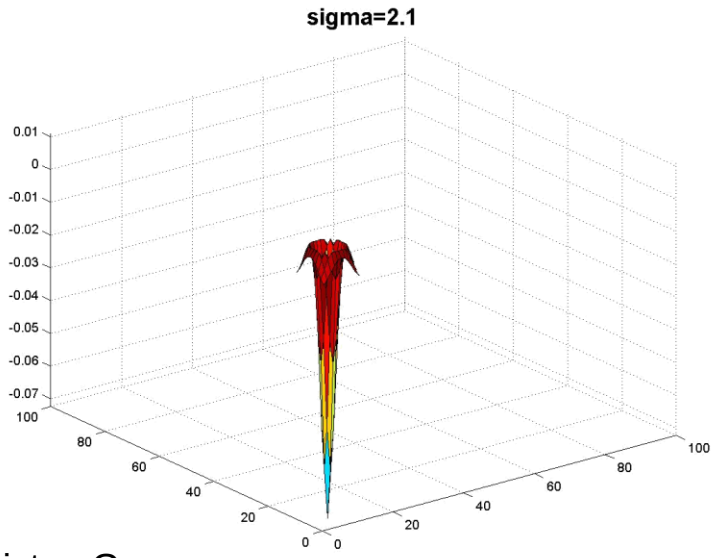


Example

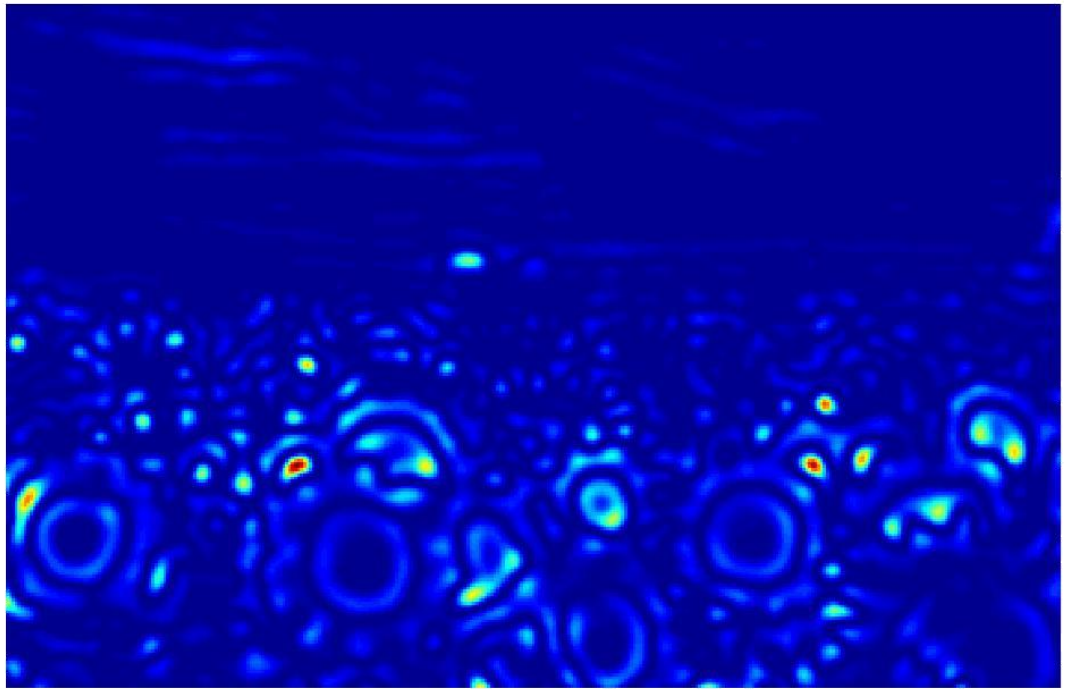
Original image
at ¾ the size

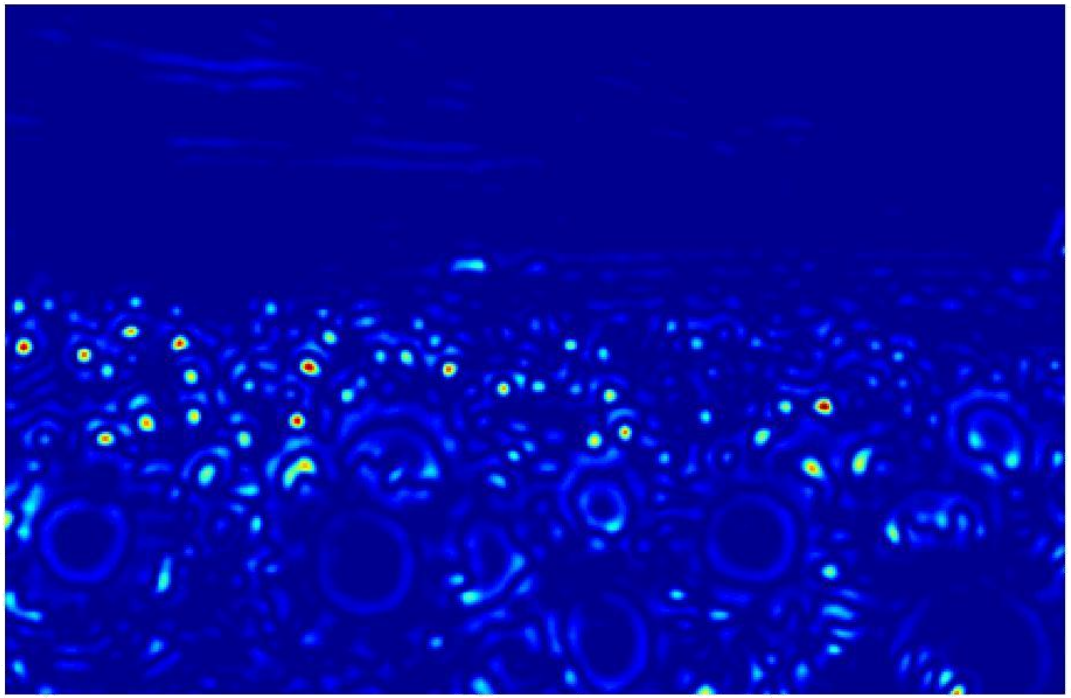
Original image
at ¾ the size

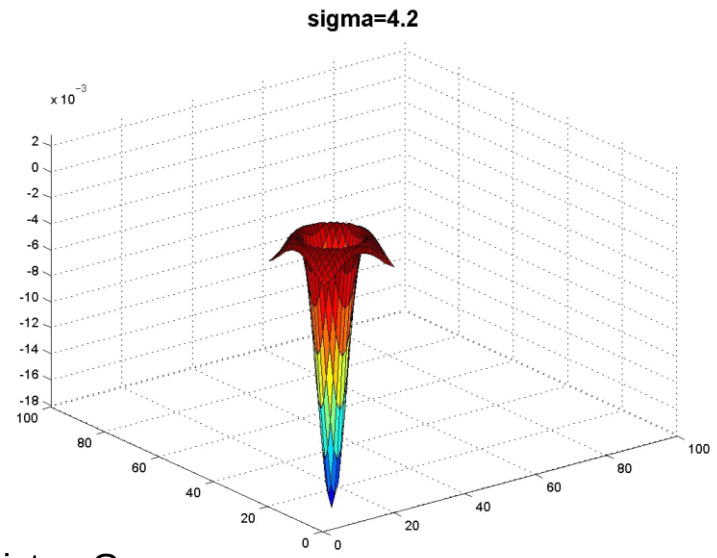




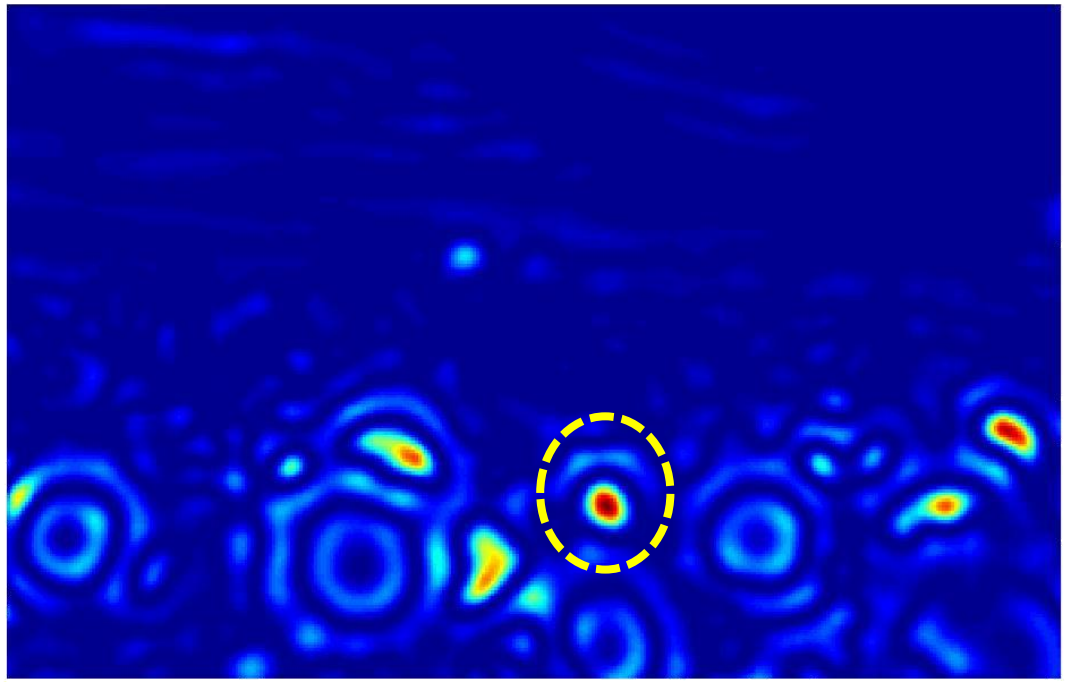
Kristen Grauman

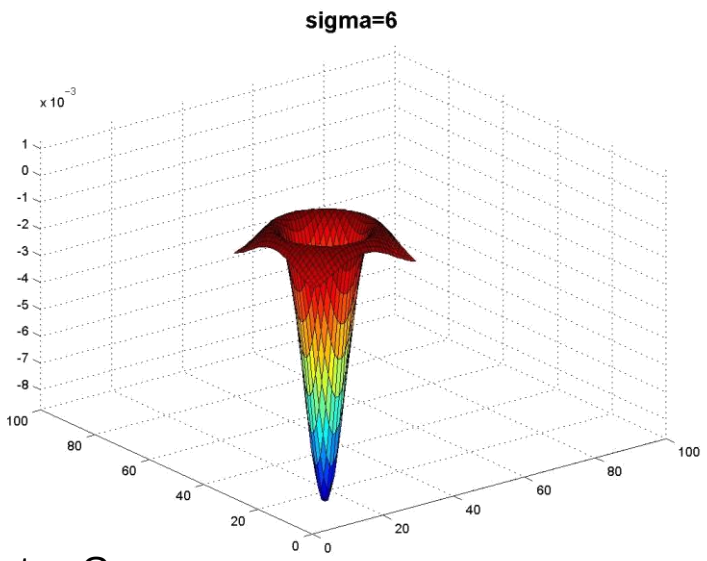


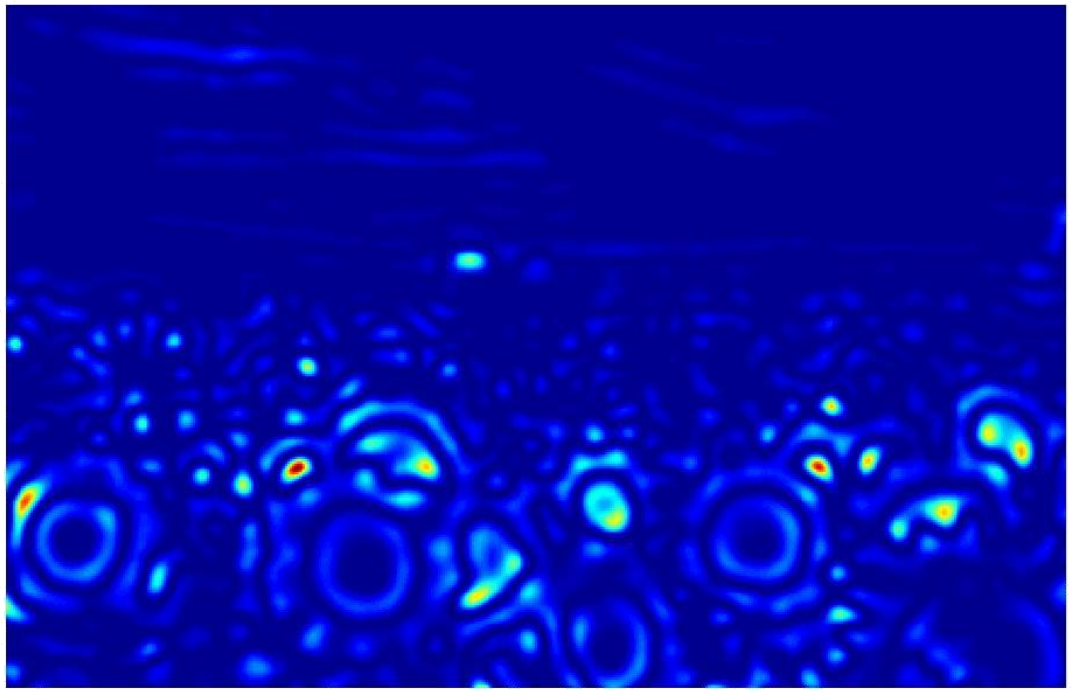




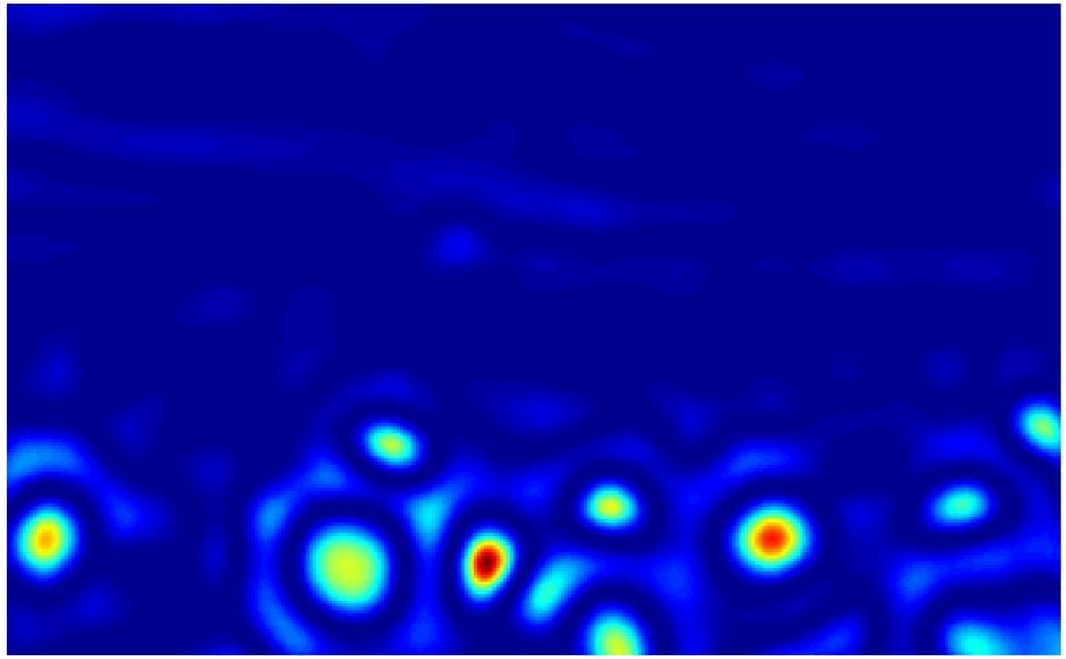
Kristen Grauman

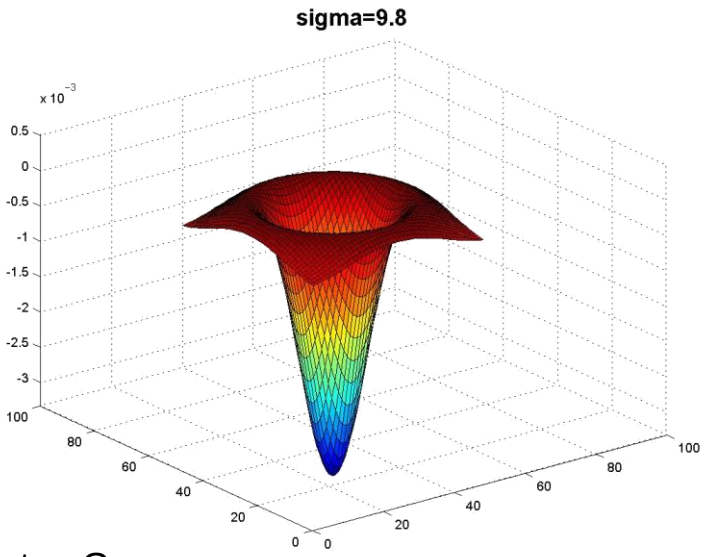


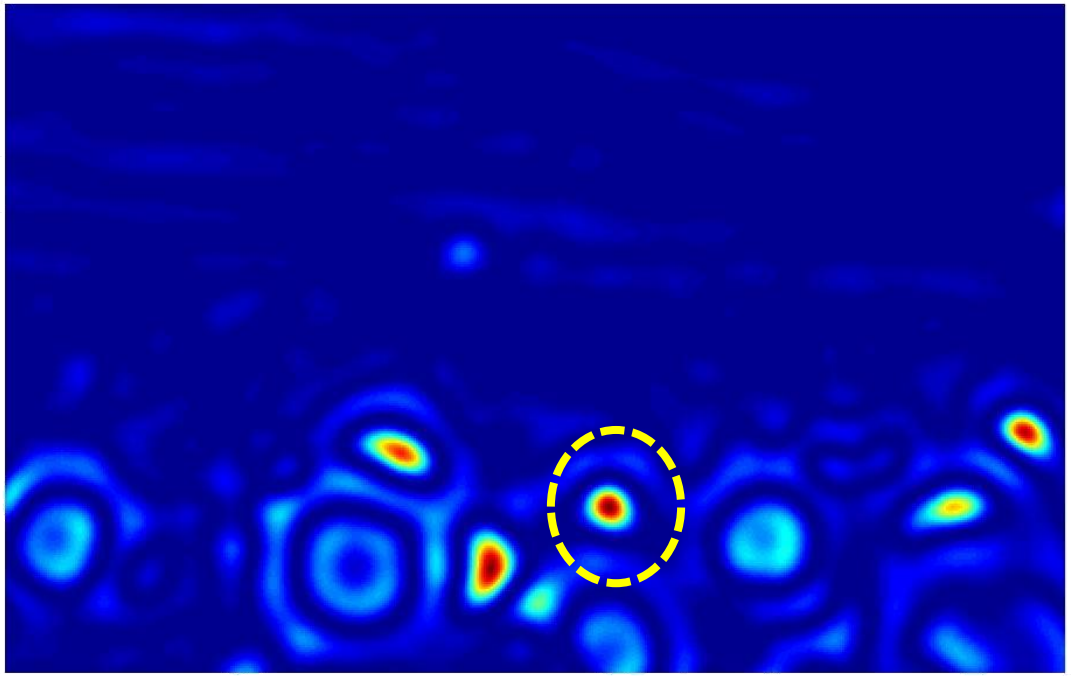




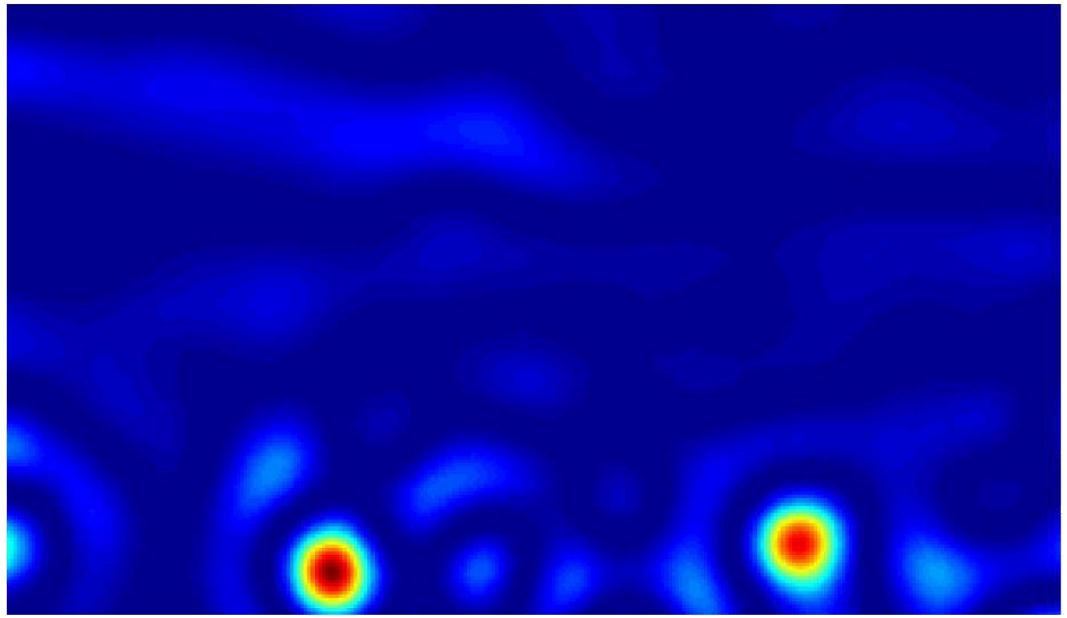
Kristen Grauman

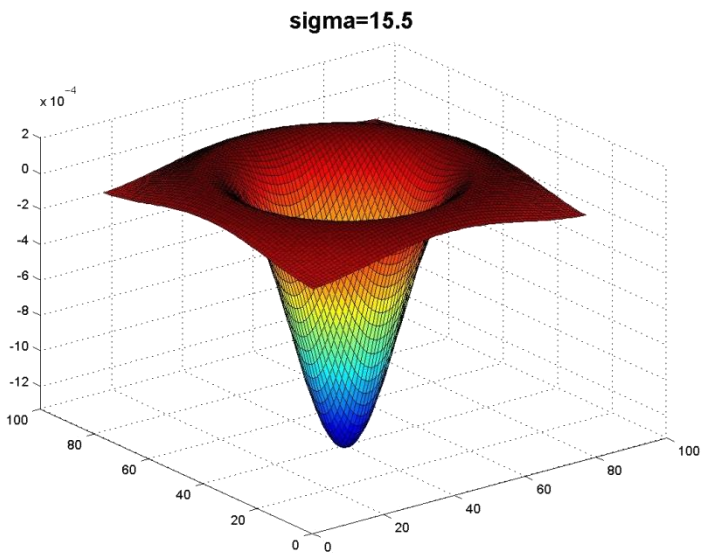


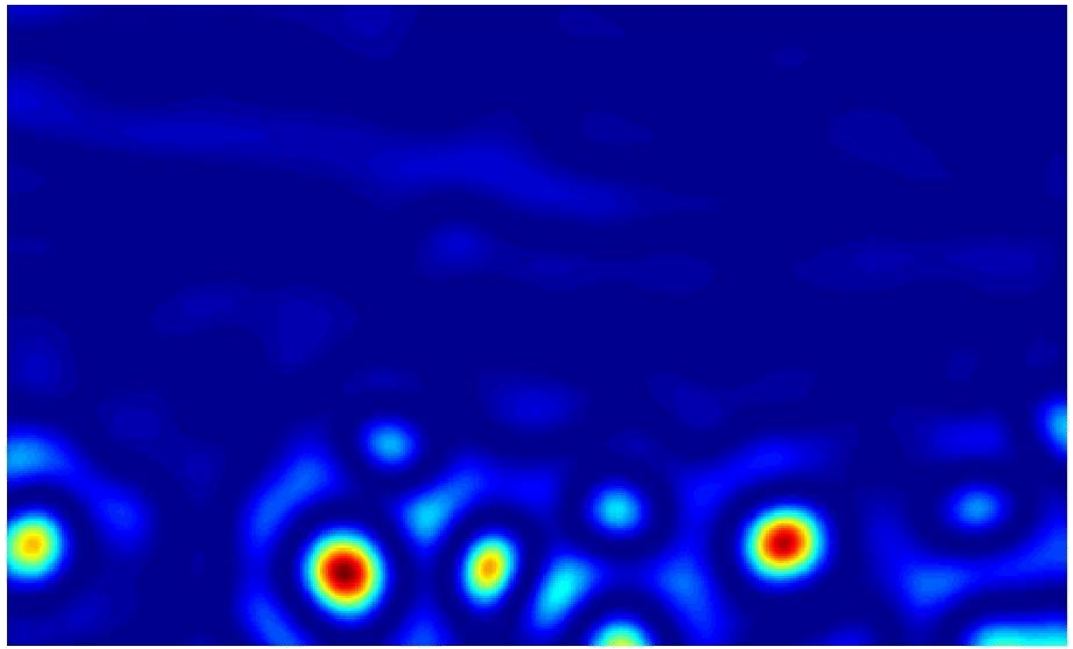




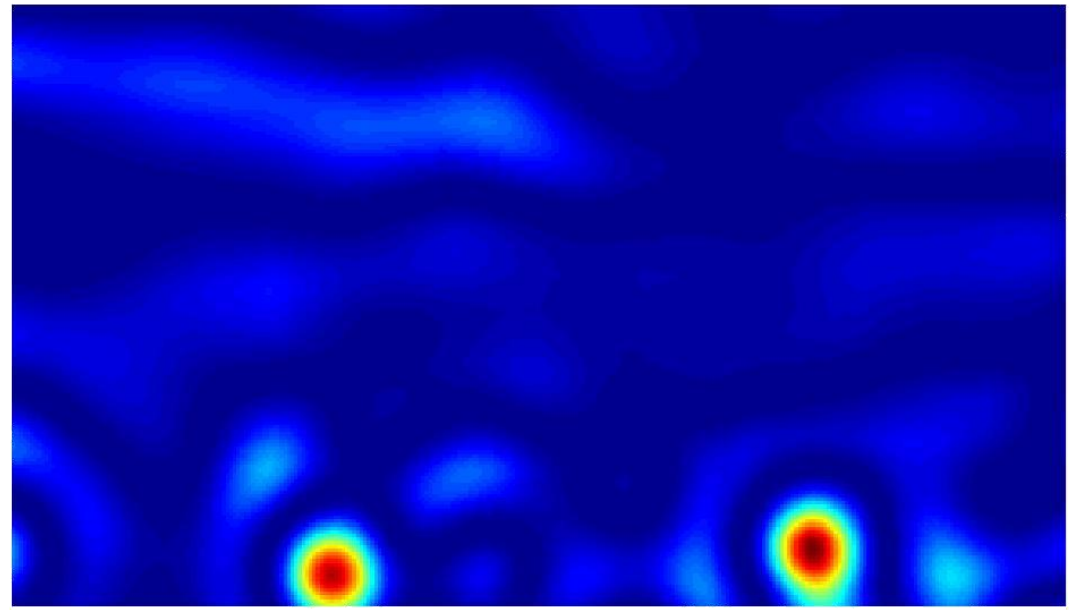
Kristen Grauman

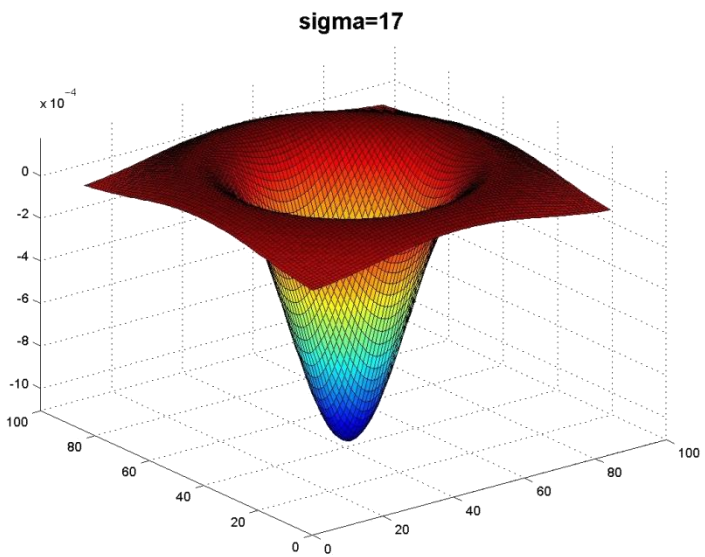


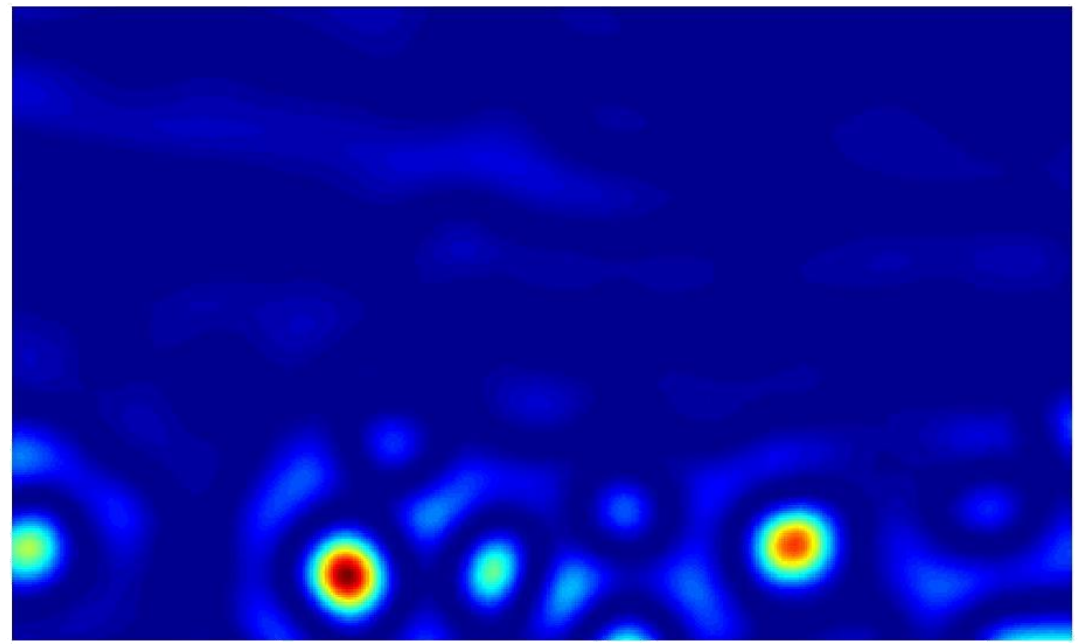




Kristen Grauman





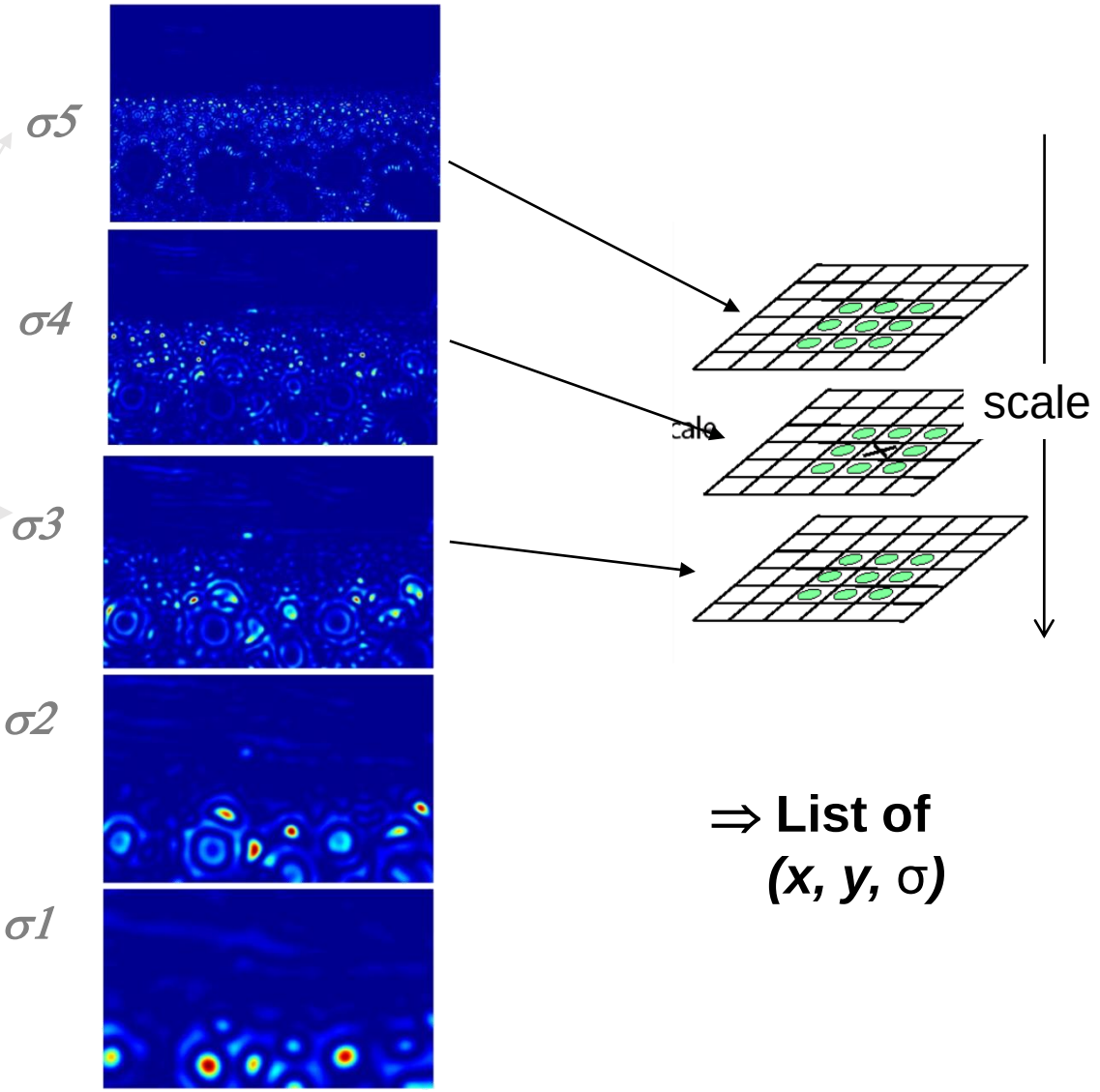


Kristen Grauman

Scale invariant interest points

Interest points are local maxima in both position and scale.

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



Squared filter response maps

⇒ **List of** ***(x, y, σ)***

Technical detail - Efficient implementation

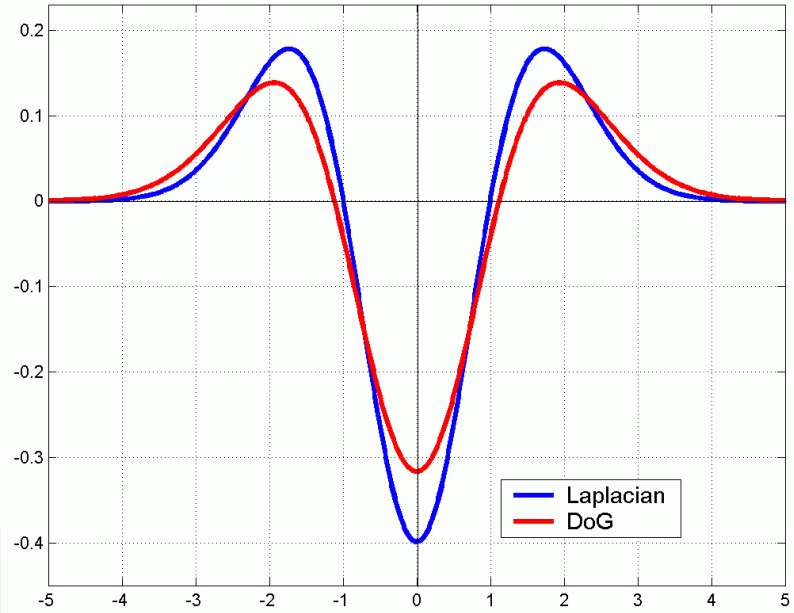
- We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

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

(Laplacian)

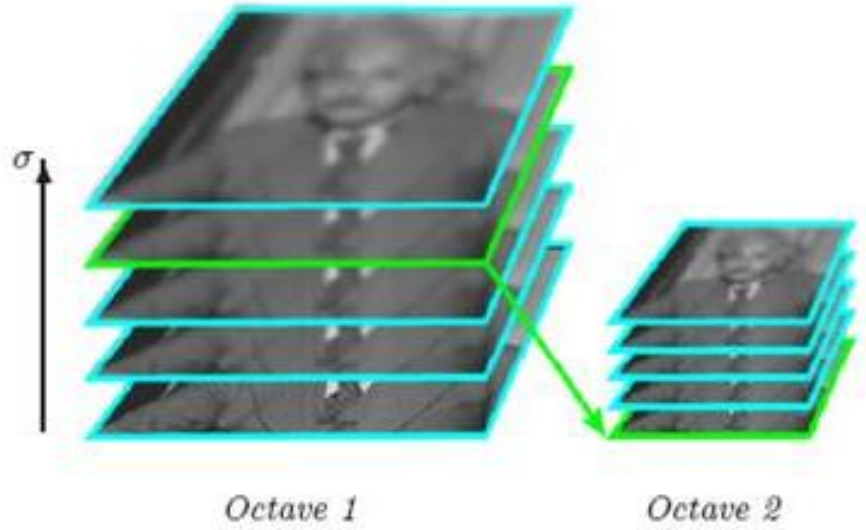
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

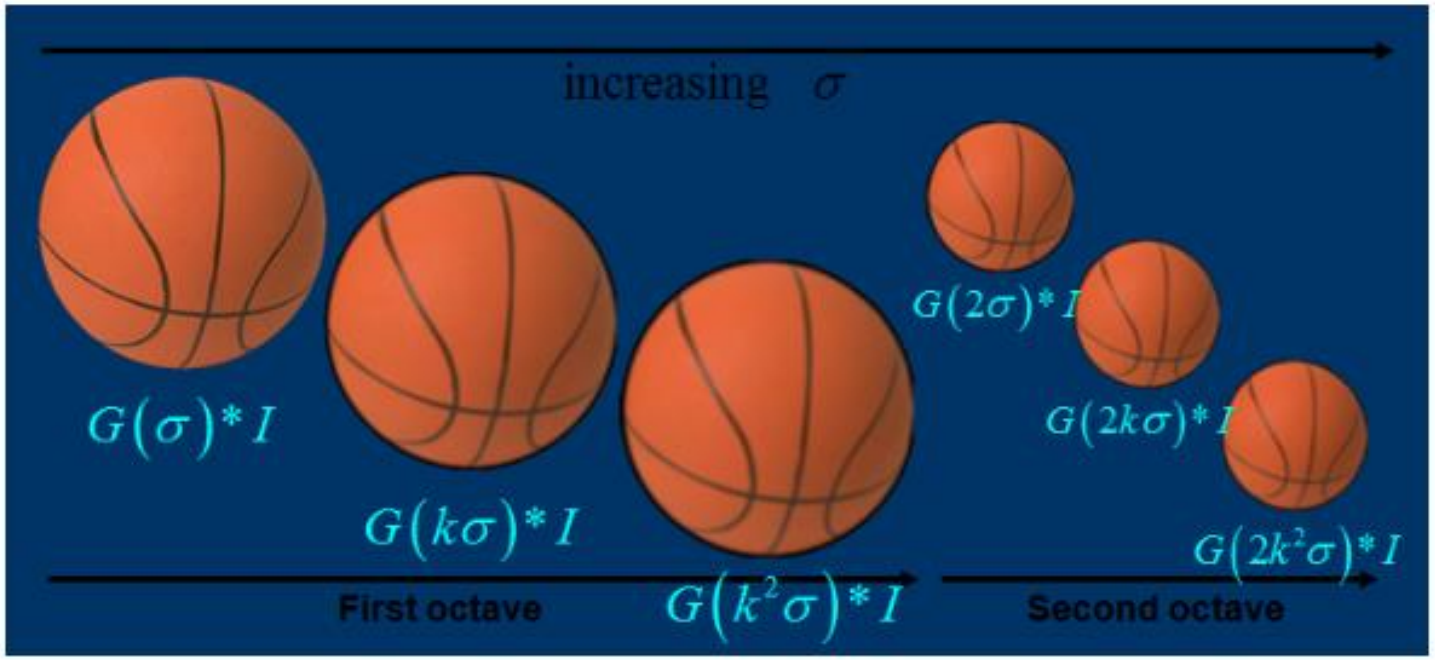
(Difference of Gaussians)



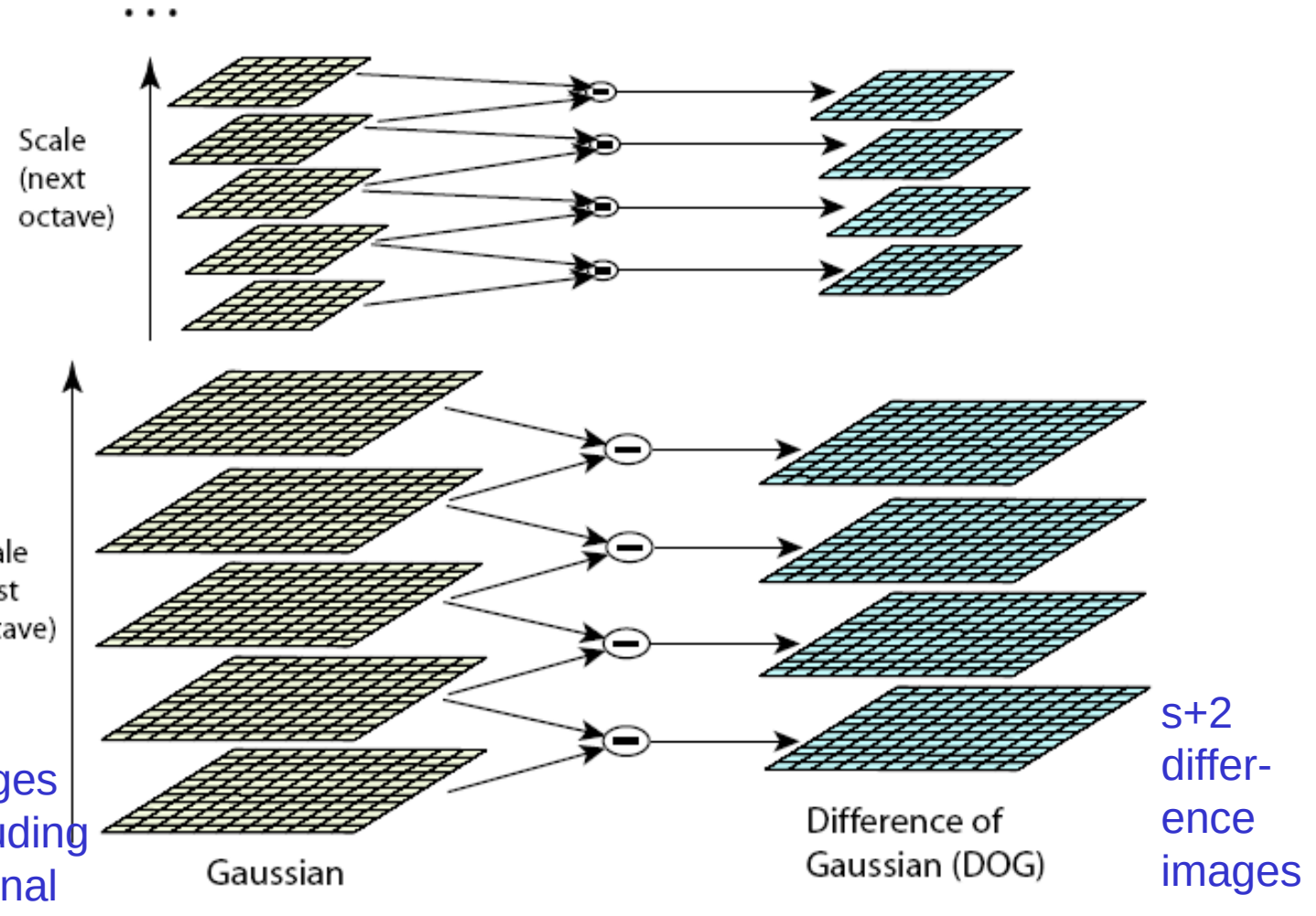
David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, IJCV, 2004

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

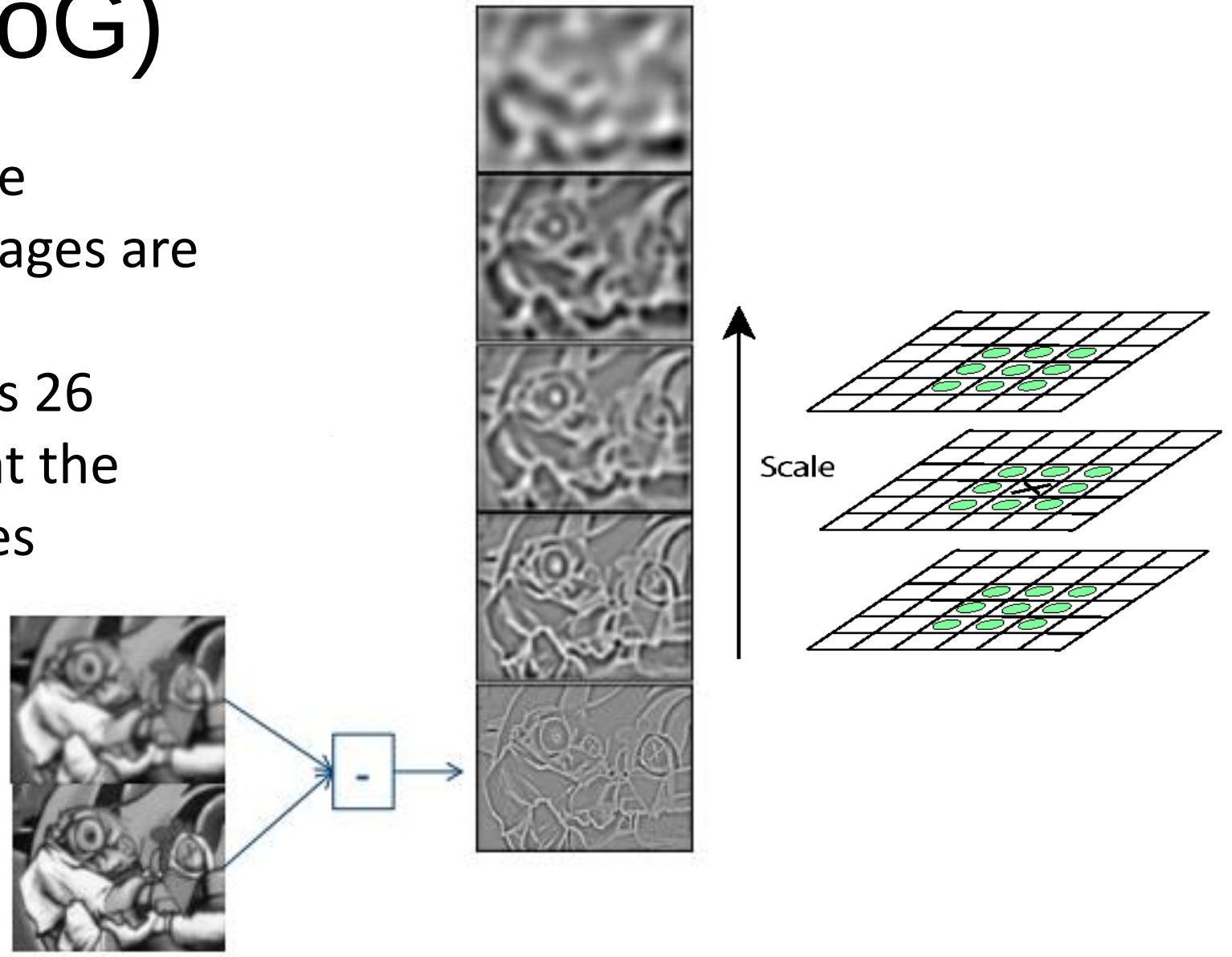
s+3 images including original

s+2 differ-ence images

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

Key point localization & Difference of Gaussians (DoG)

Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).



Summary

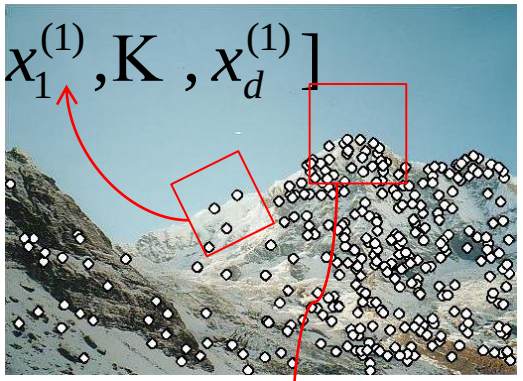
- Corner detectors
  - Stable in space
  - Harris
- Blob detectors
  - Stable in scale and space
  - LoG, DoG

Local features: main components

1) Detection: Identify the interest points

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

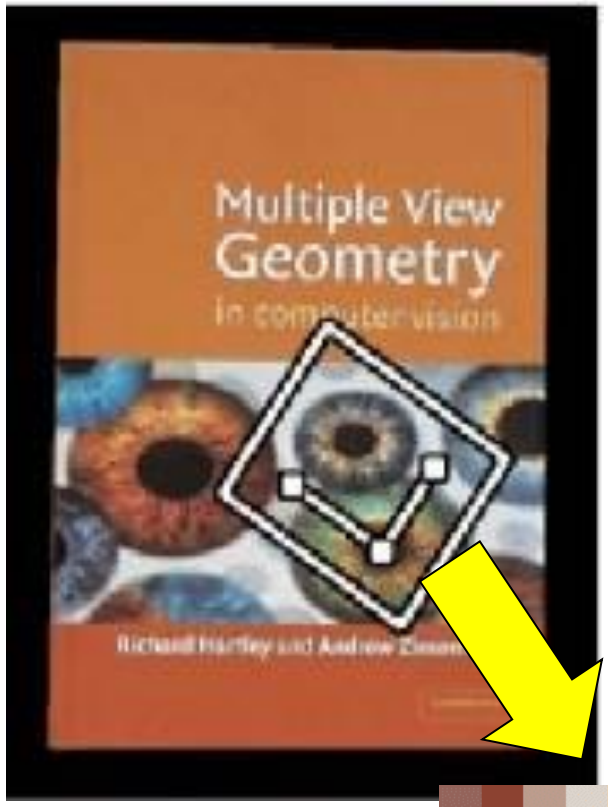
$$\mathbf{x}_1 = [x_1^{(1)}, \mathrm{K}, x_d^{(1)}]$$

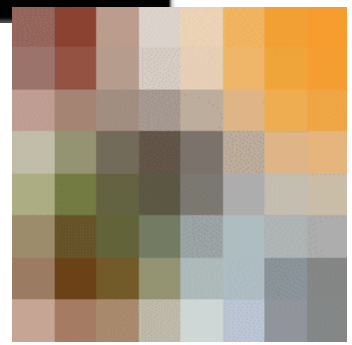


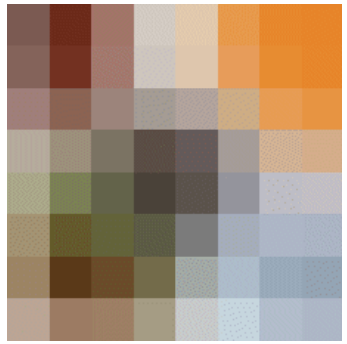
$$\mathbf{x}_2 = [x_1^{(2)}, \mathrm{K}, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views

Geometric transformations





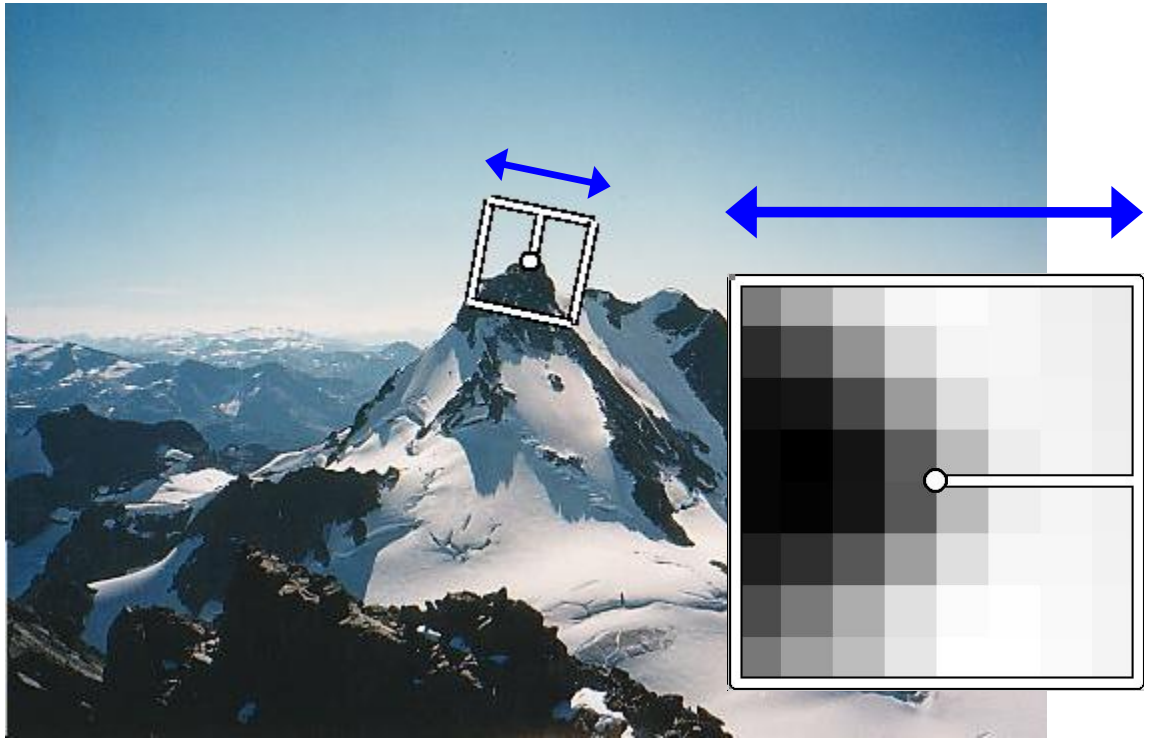


e.g. scale,
translation,
rotation

Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Making descriptor rotation invariant



- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

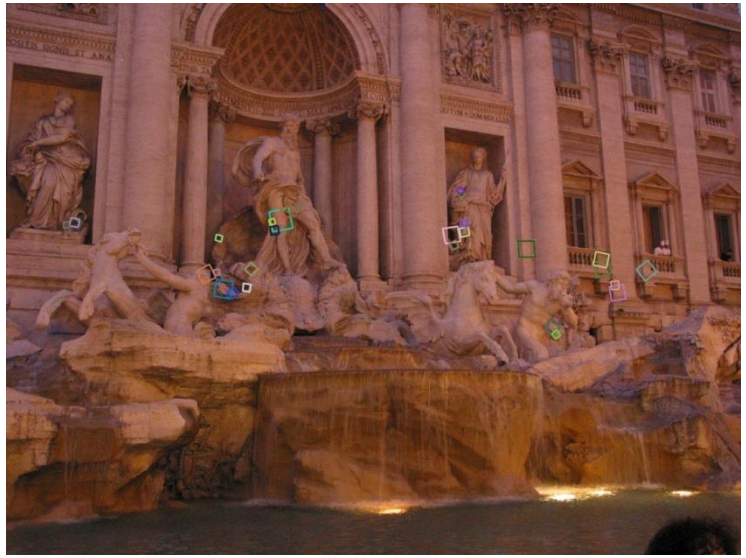
Image from Matthew Brown

SIFT Description

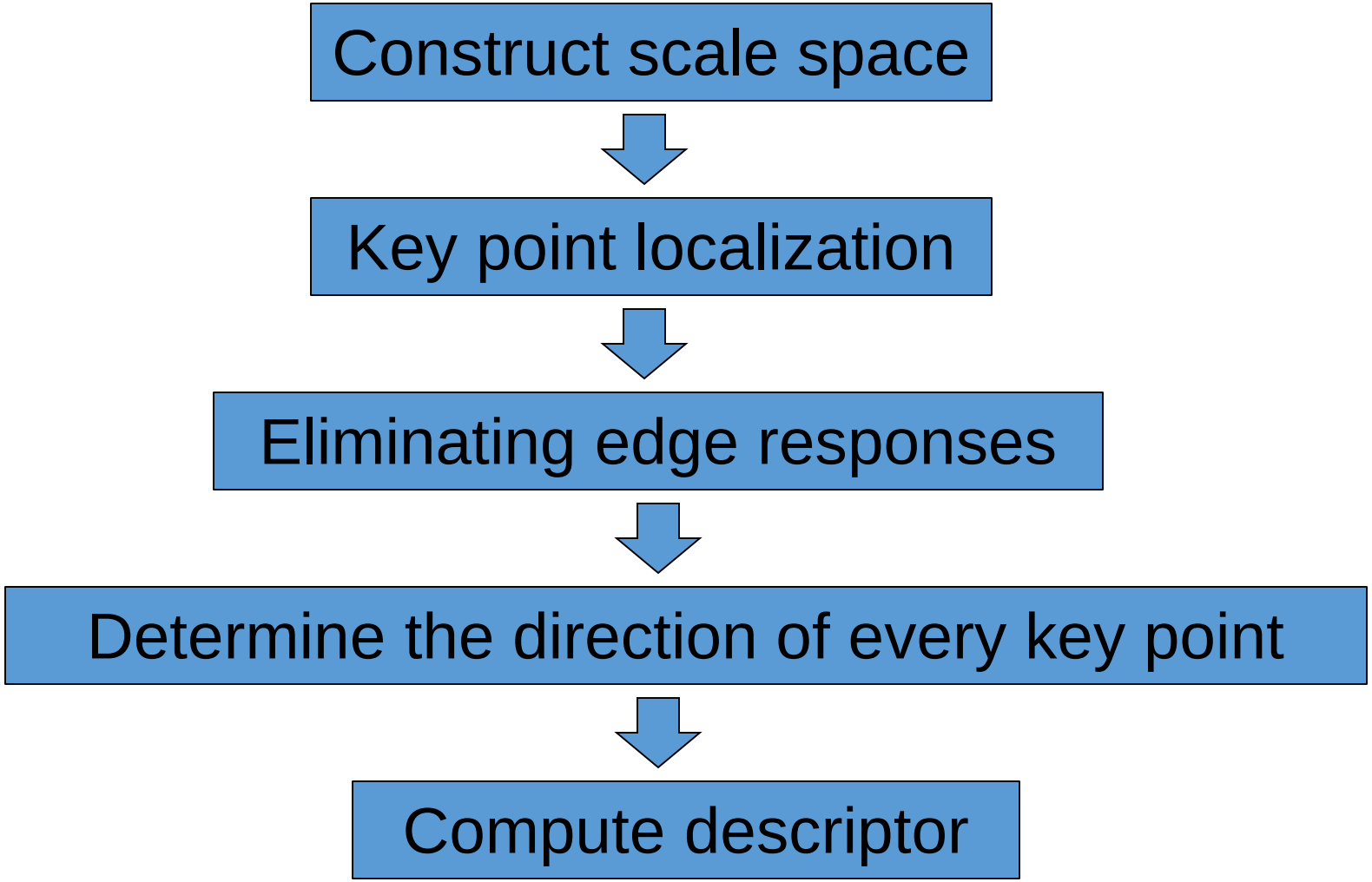
1999年British Columbia大学大卫.劳伊（David G.Lowe）教授总结了现有的基于不变量技术的特征检测方法，并正式提出了一种基于尺度空间的、对图像缩放、旋转甚至仿射变换保持不变性的图像局部特征描述算子－SIFT（尺度不变特征变换, Scale Invariant Feature Transform），这种算法在2004年被加以完善。

SIFT descriptor [Lowe 2004]

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



SIFT flowchat

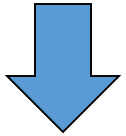


Key point localization

在计算过程中，分别对图像的行、列及尺度三个量进行了修正， 其修正结果如下：

$$D(X+\Delta X)=D(X)+(\frac{\partial D}{\partial X})^T(\Delta X)+\frac{1}{2}X^T\frac{\partial^2 D}{\partial X^2}\Delta X,\quad X=(x,y,\sigma)$$

$$\min\{D(X+\Delta X)-D(X)\}=\min\{(\frac{\partial D}{\partial X})^T\Delta X+\frac{1}{2}(\Delta X)^T\frac{\partial^2 D}{\partial X^2}(\Delta X)\}$$



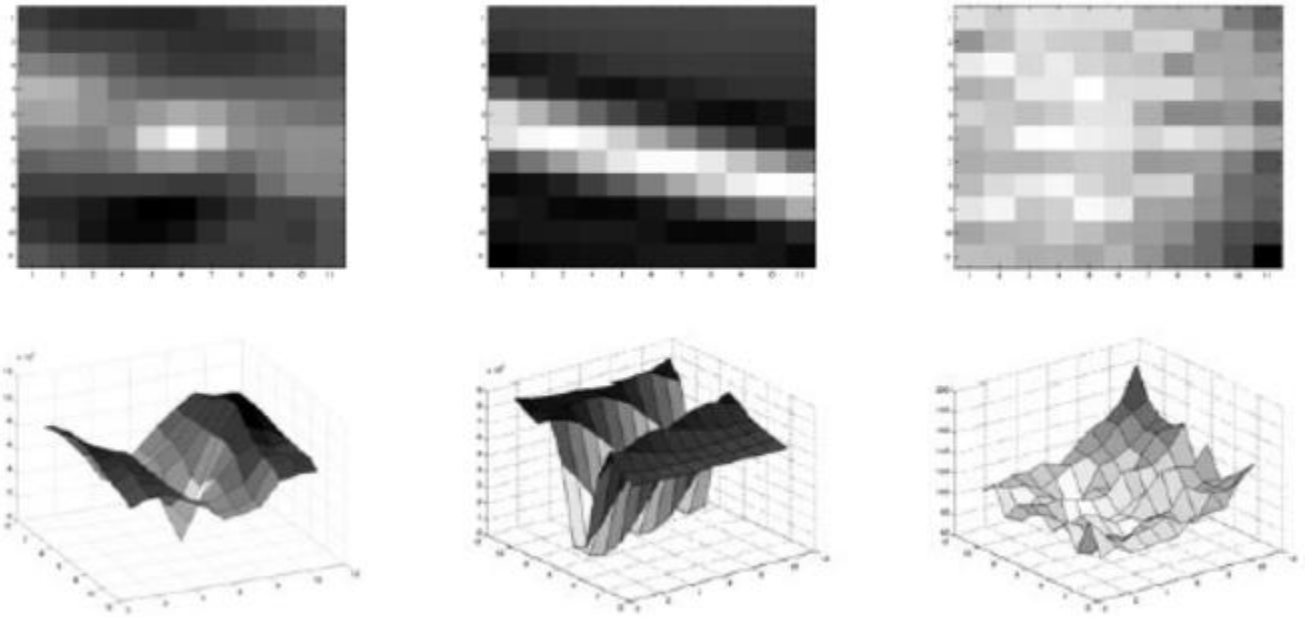
$$\Delta X=-(\frac{\partial D}{\partial X})^T(\frac{\partial^2 D}{\partial X^2})^{-1}$$

Compute: $D(X+\Delta X)=D(X)+\frac{1}{2}(\frac{\partial D}{\partial X})^T\Delta X$

$D(X+\Delta X)>t$  extreme point

Eliminating edge responses

- For stability, it is not sufficient to reject key points with low contrast.
- The DOG function will have a strong response along edges.
- So we need eliminating edge responses



- The principal curvatures can be computed from a 2x2 Hessian matrix, computed at the location and scale of the key point: $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$

The derivatives are estimated by taking differences of neighboring sample points.

Eliminating edge responses

The eigenvalues of H are proportional to the principal curvatures of D, we can avoid explicitly computing the eigenvalues, as we are only concerned with their ratio. Let α be the eigenvalue with the largest magnitude and β be the smaller one

$$Tr(H) = D_{xx} + D_{yy} \qquad Det(H) = D_{xx} \times D_{yy} - D_{xy} \times D_{xy}$$

$$\alpha = r\beta \qquad \frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r+1)^2}{r}$$

rejecting keypoints for which $\frac{Tr(H)^2}{Det(H)} > \frac{(r+1)^2}{r}$

(r=10 in Lowe)

Determine the direction of every key point

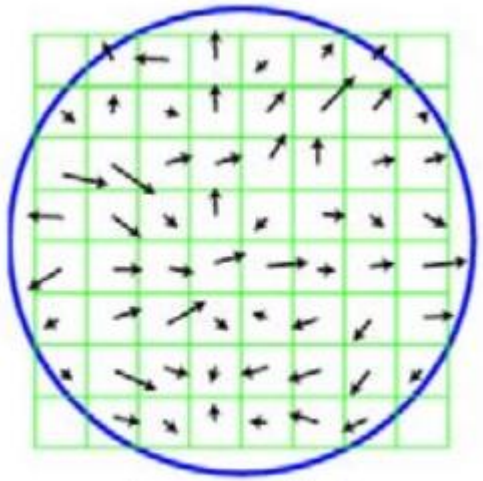


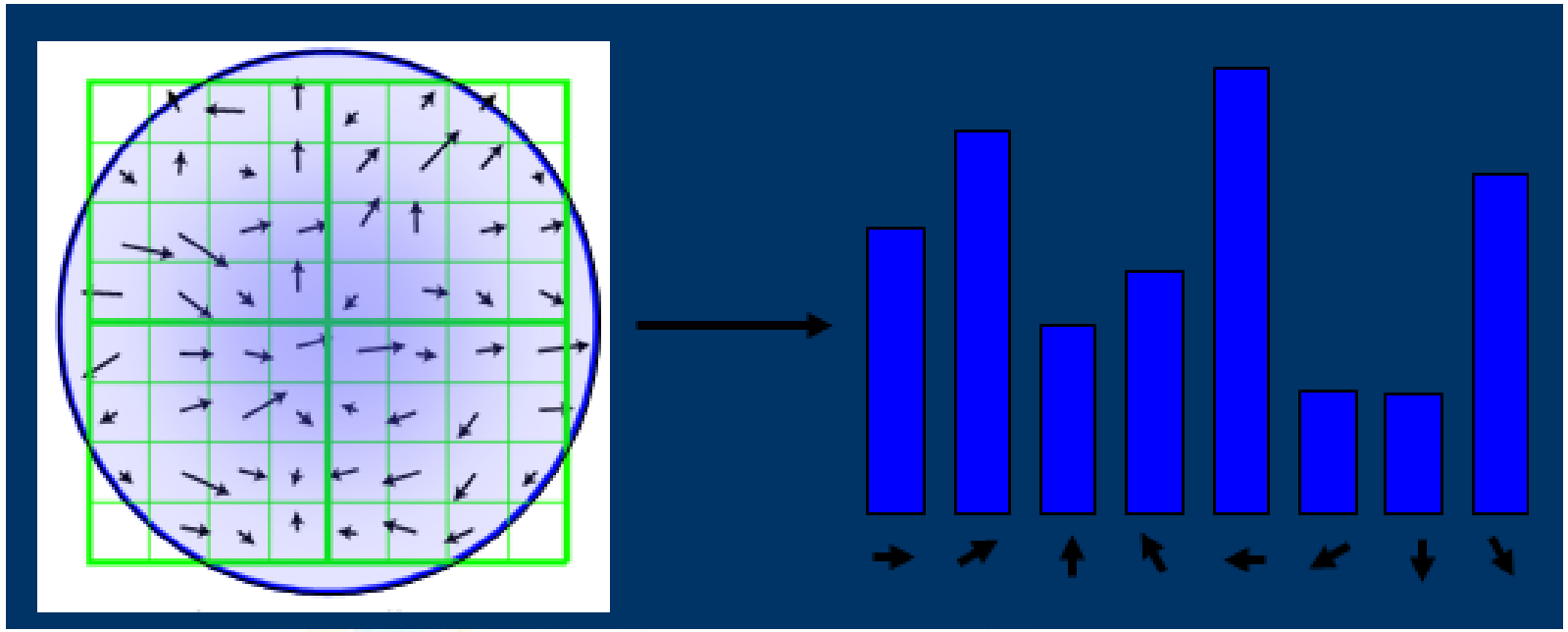
Image gradients

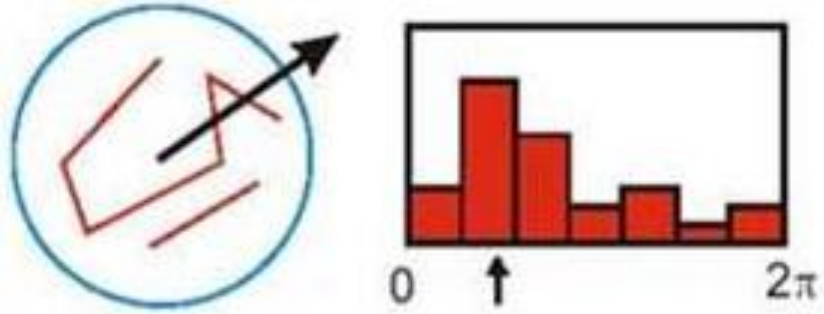
$$radius = \frac{3\sigma_{oct} \times \sqrt{2} \times (d+1) + 1}{2} \qquad d = 4$$

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Determine the direction of every key point

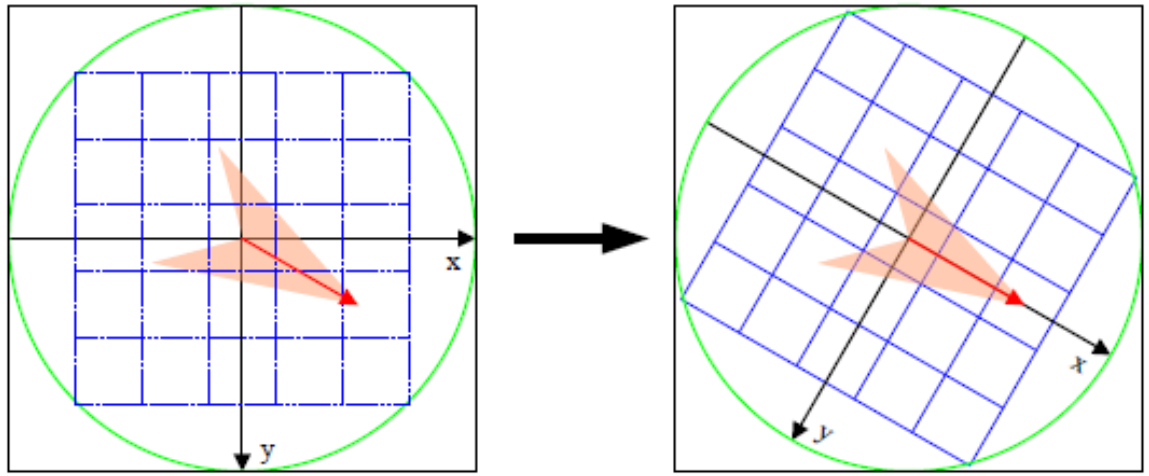




Identify peak and assign orientation and sum of magnitude to key point
The user may choose a threshold to exclude key points based on their assigned sum of magnitudes.

Compute descriptor

In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation



New position

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix}$$

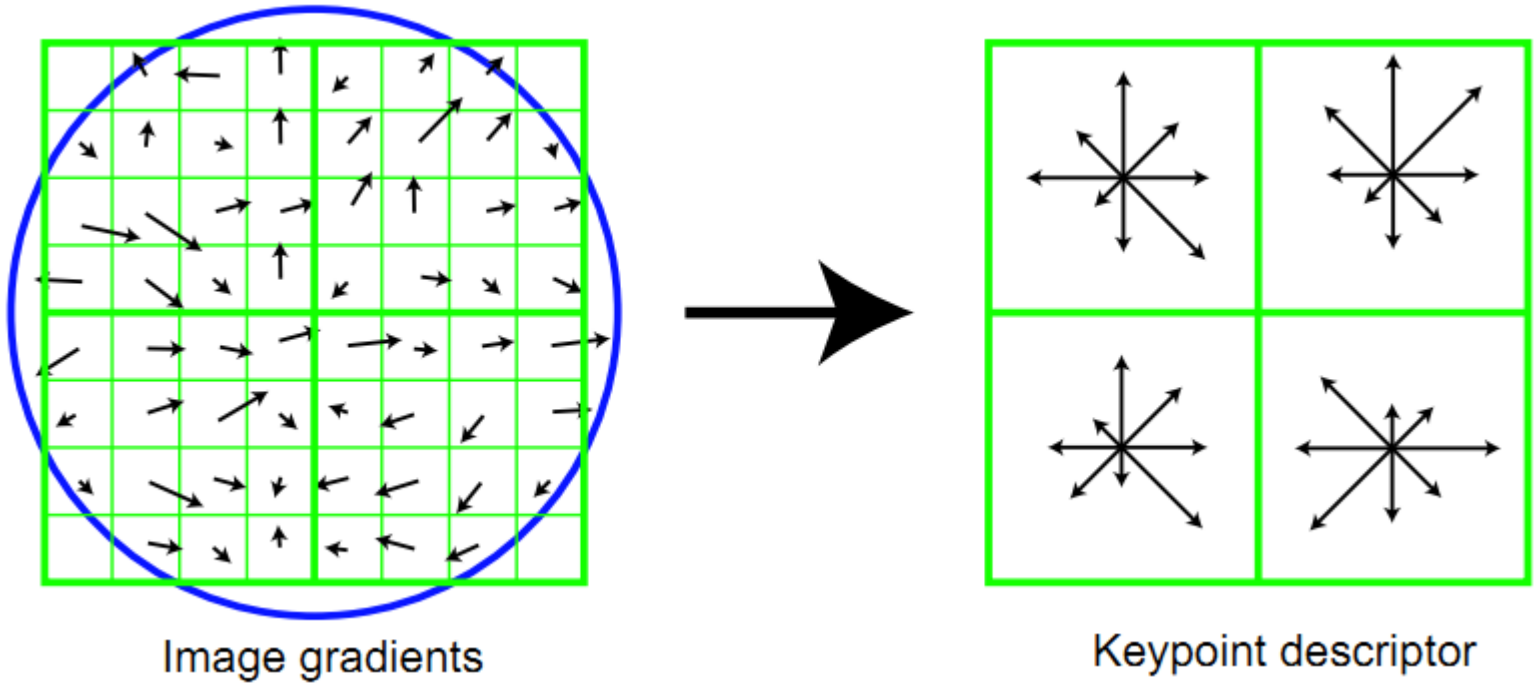


Figure 4.18 A schematic representation of Lowe's (2004) scale invariant feature transform (SIFT): (a) Gradient orientations and magnitudes are computed at each pixel and weighted by a Gaussian fall-off function (blue circle). (b) A weighted gradient orientation histogram is then computed in each subregion, using trilinear interpolation. While this figure shows an 8×8 pixel patch and a 2×2 descriptor array, Lowe's actual implementation uses 16×16 patches and a 4×4 array of eight-bin histograms.

Reduce the effect of Illumination

- Threshold the descriptors and renormalize.

  This strategy can reduce the effect of Illumination to some extent.

Reference: David G. Lowe, **"Distinctive image features from scale-invariant keypoints,"** *International Journal of Computer Vision,* 60, 2 (2004), pp. 91-110

Feature descriptors inspired by SIFT

PCA-SIFT Ke and Sukthankar (2004)

SURF Bay, Tuytelaars, and Van Gool (2006)

GLOH Mikolajczyk and Schmid (2005)

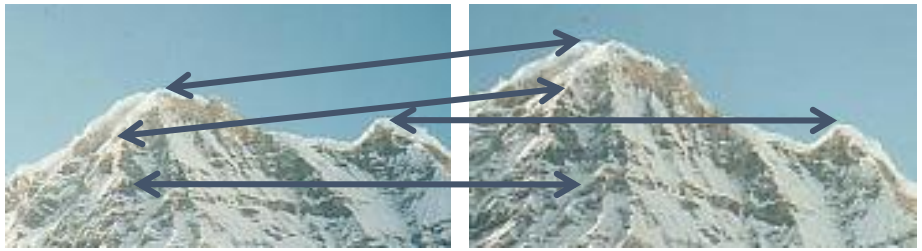
… …

Local features: main components

1) Detection: Identify the interest points

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

3) Matching: Determine correspondence between descriptors in two views

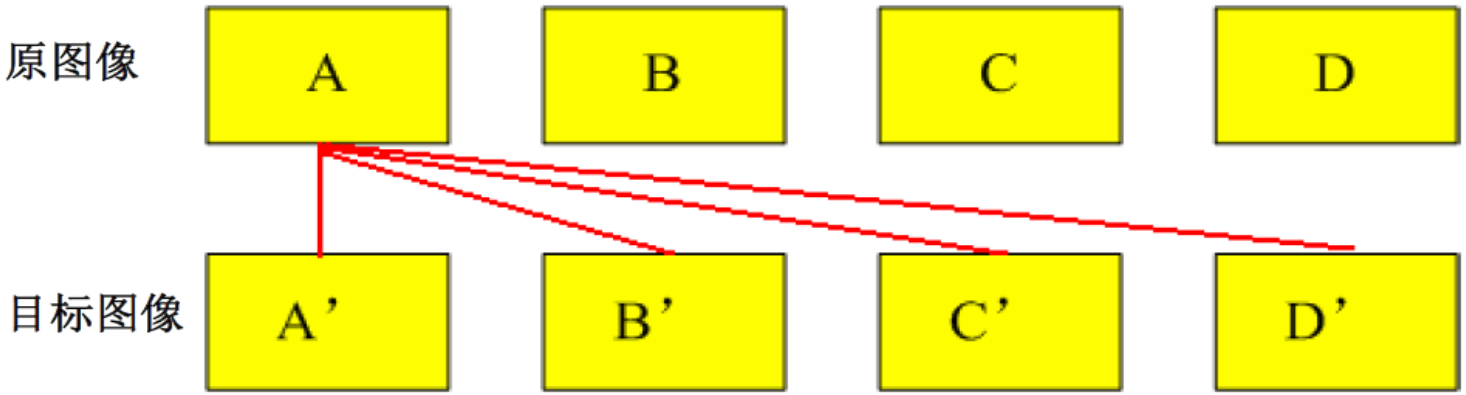


Key Point Matching

- **分别对模板图（参考图，reference image）和实时图（观测图，observation image）建立关键点描述子集合。**
- **目标的识别是通过两点集内关键点描述子的比对来完成。**

Key Point Matching

穷举匹配



模板图中关键点描述子： $R_i = (r_{i1}, r_{i2}, \cdots, r_{i128})$

实时图中关键点描述子： $S_i = (s_{i1}, s_{i2}, \cdots, s_{i128})$

任意两描述子相似性度量： $d(R_i, S_i) = \sqrt{\sum_{j=1}^{128} (r_{ij} - s_{ij})^2}$

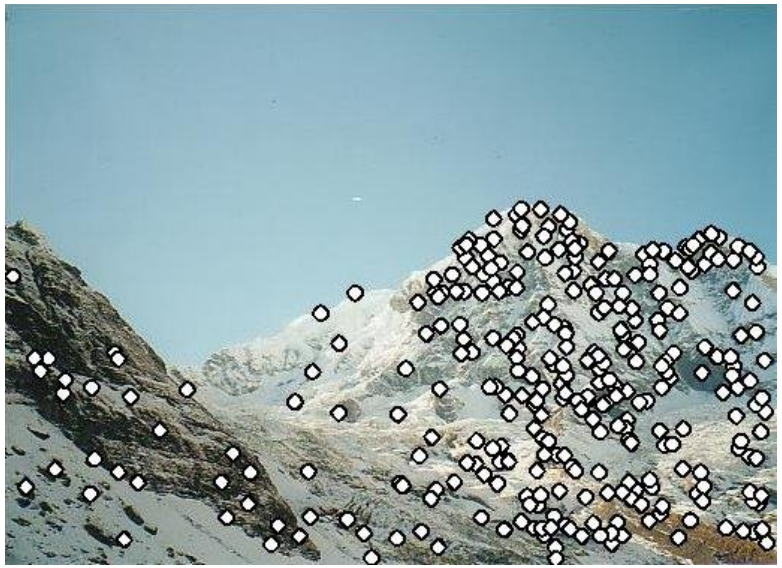
Key Point Matching

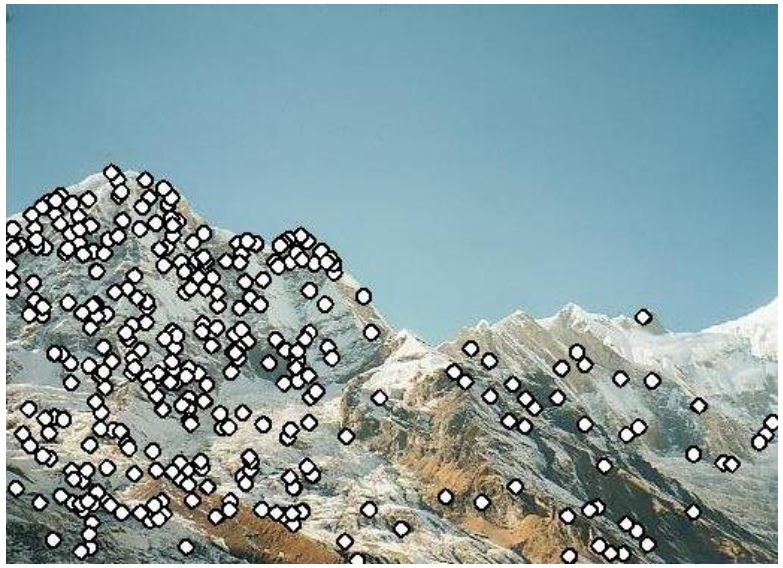
- 关键点的匹配可以采用穷举法来完成，但是这样耗费的时间太多，一般都采用一种叫kd树的数据结构来完成搜索。
- 搜索的内容是以目标图像的关键点为基准，搜索与目标图像的特征点最邻近的原图像特征点和次邻近的原图像特征点。

Recap: robust feature-based alignment

Source: L. Lazebnik

Recap: robust feature-based alignment





- Extract features

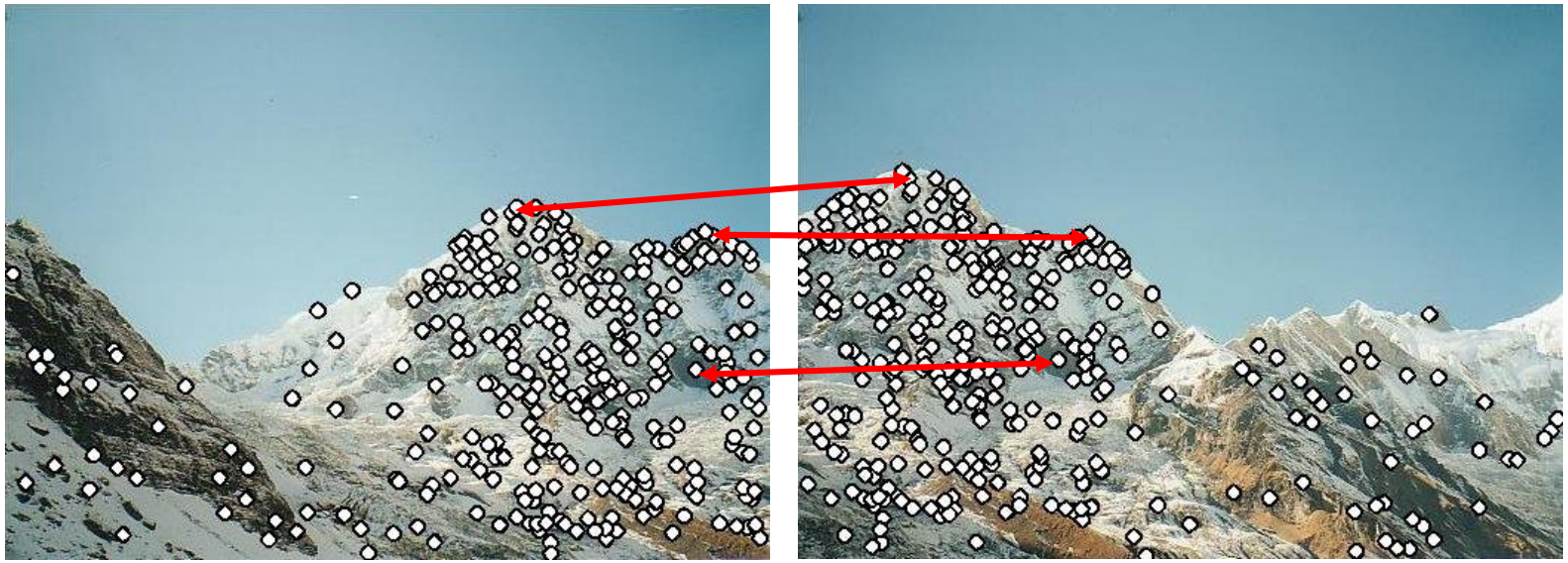
Source: L. Lazebnik

Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*

Source: L. Lazebnik

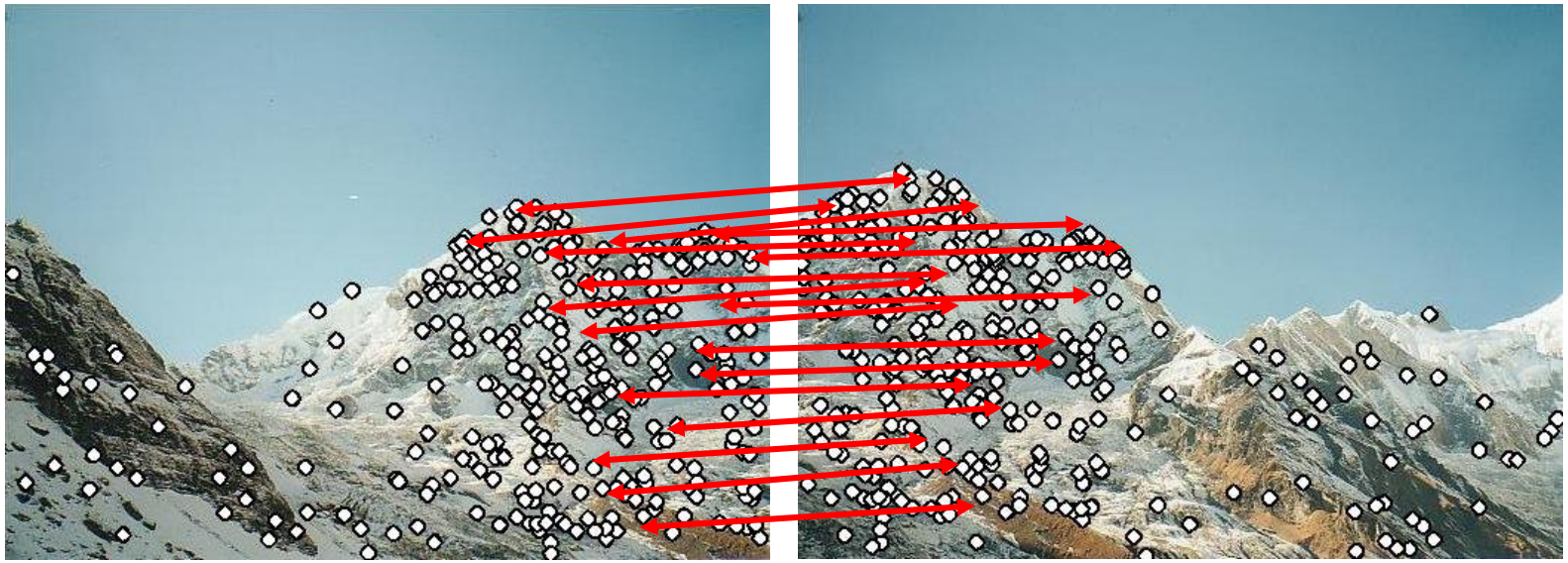
Recap: robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

Source: L. Lazebnik

Recap: robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Recap: robust feature-based alignment

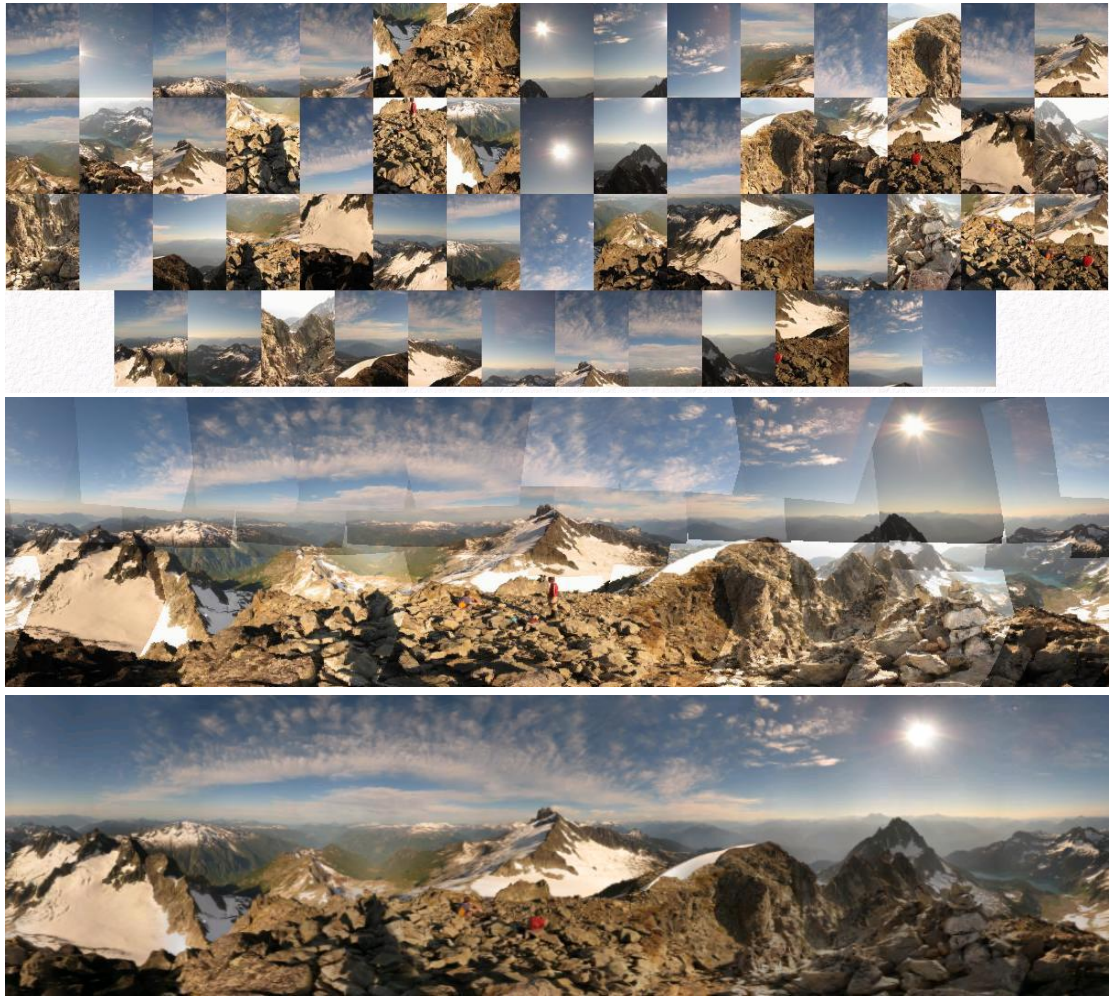
- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

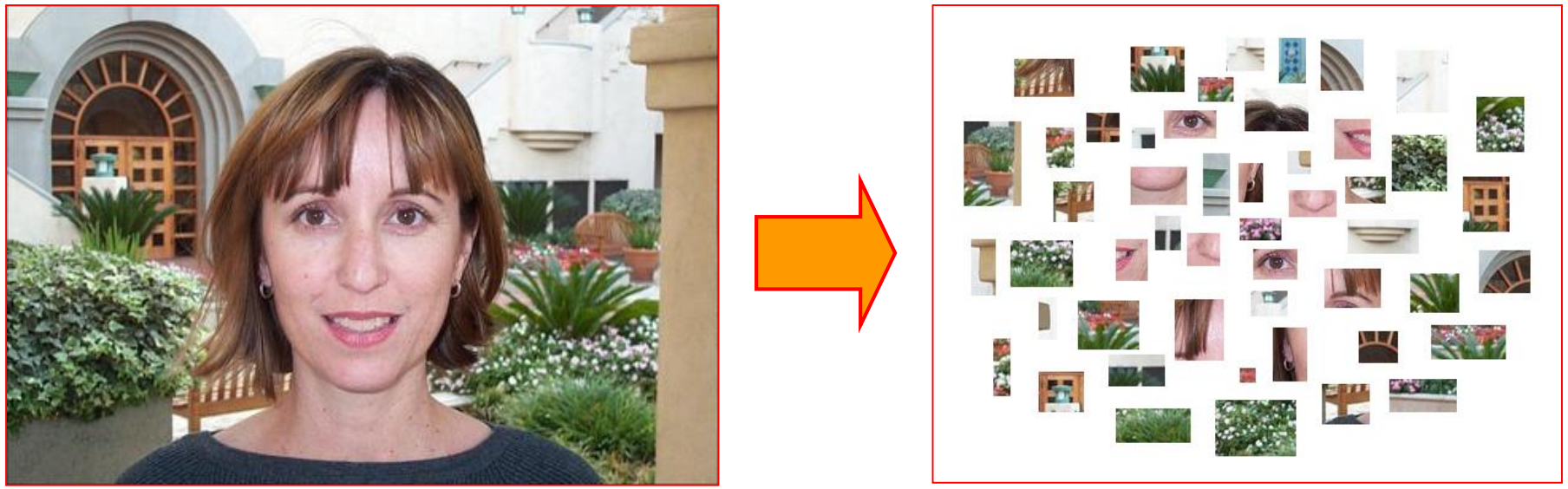
Lowe 2002

Summary

- Interest point detection
  - Harris corner detector
  - Laplacian of Gaussian, automatic scale selection
- Invariant descriptors
  - Rotation according to dominant gradient direction
  - Histograms for robustness to small shifts and translations (SIFT descriptor)

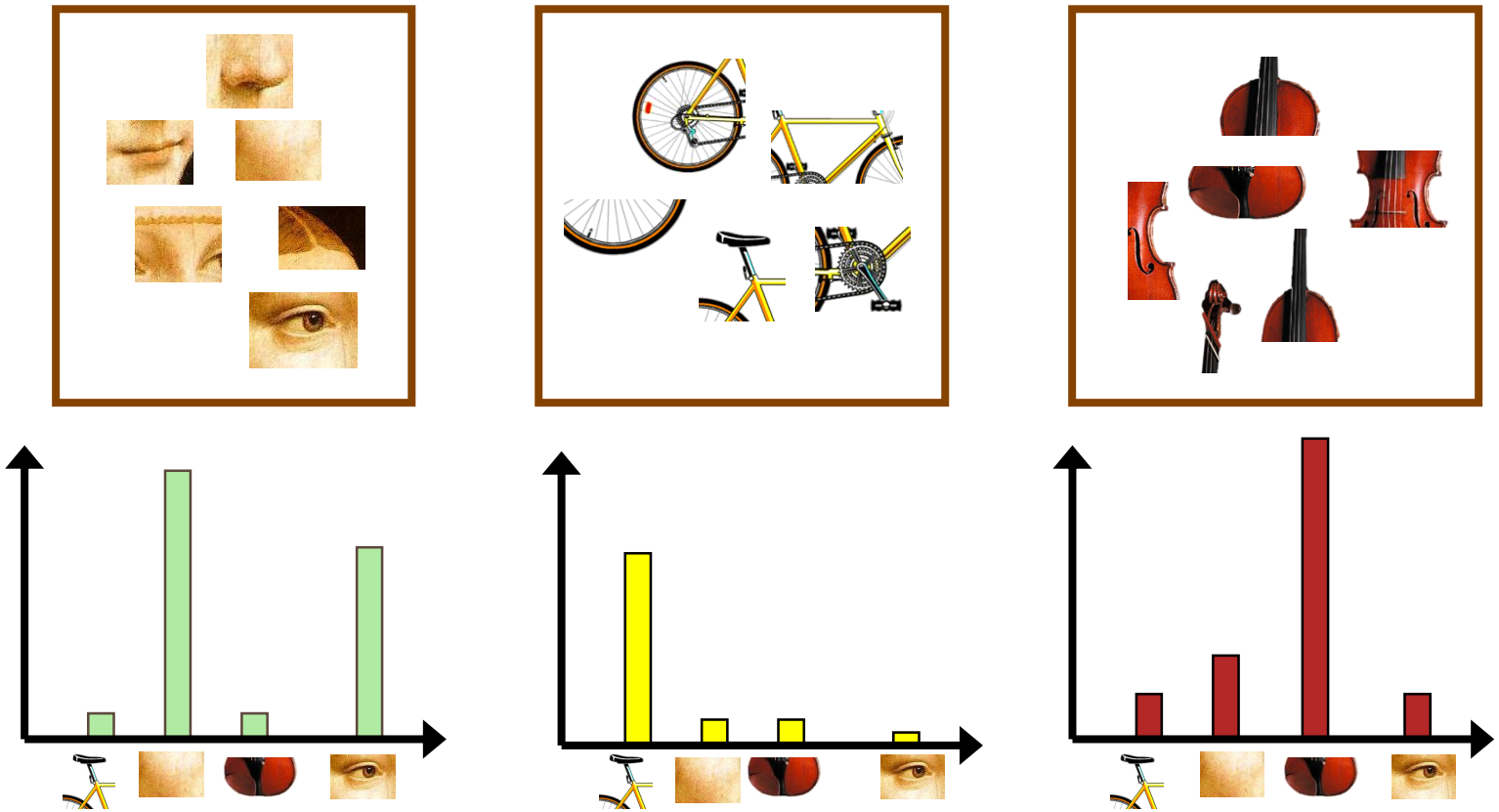
Bag-of-features

Bag-of-features models



Bag-of-features steps

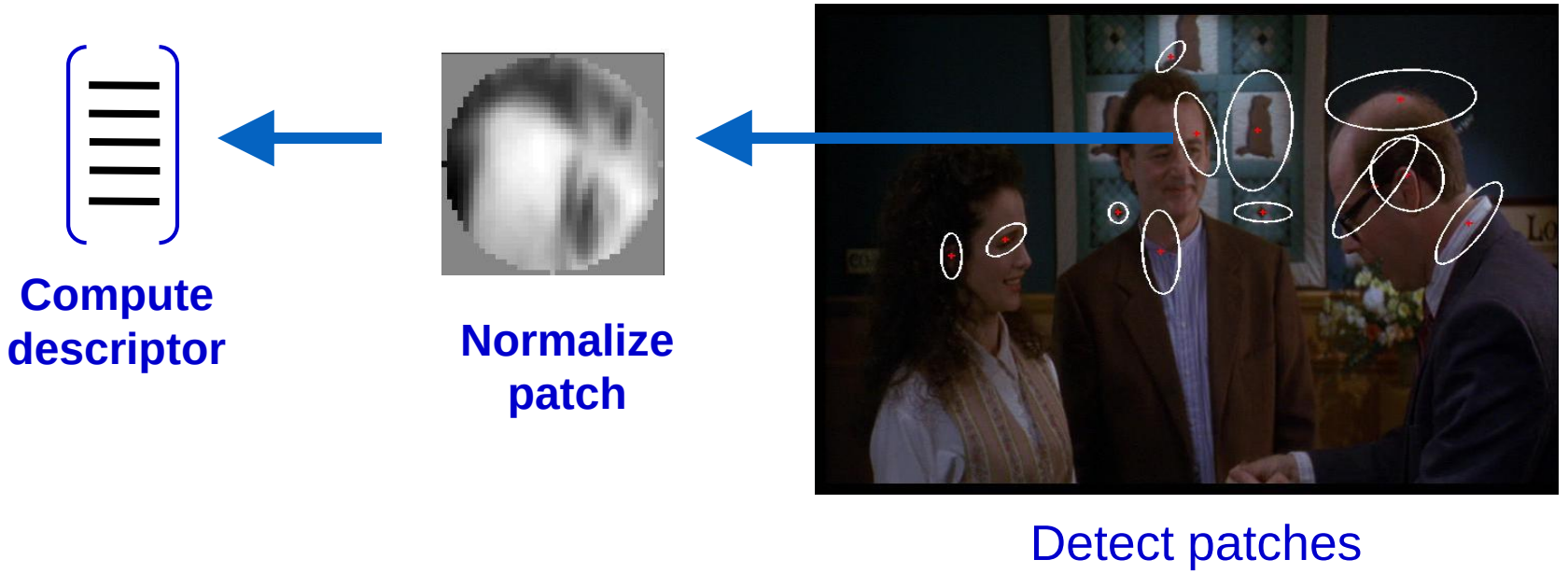
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”



1. Feature extraction

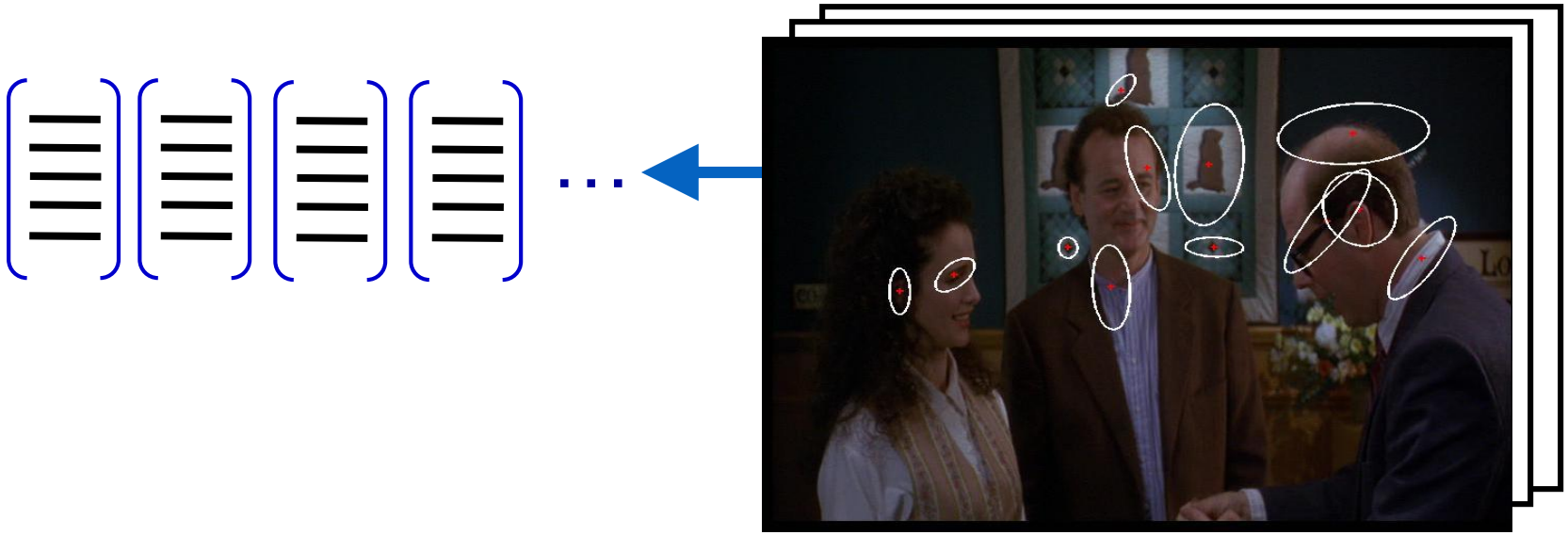
- Regular grid or interest regions

1. Feature extraction



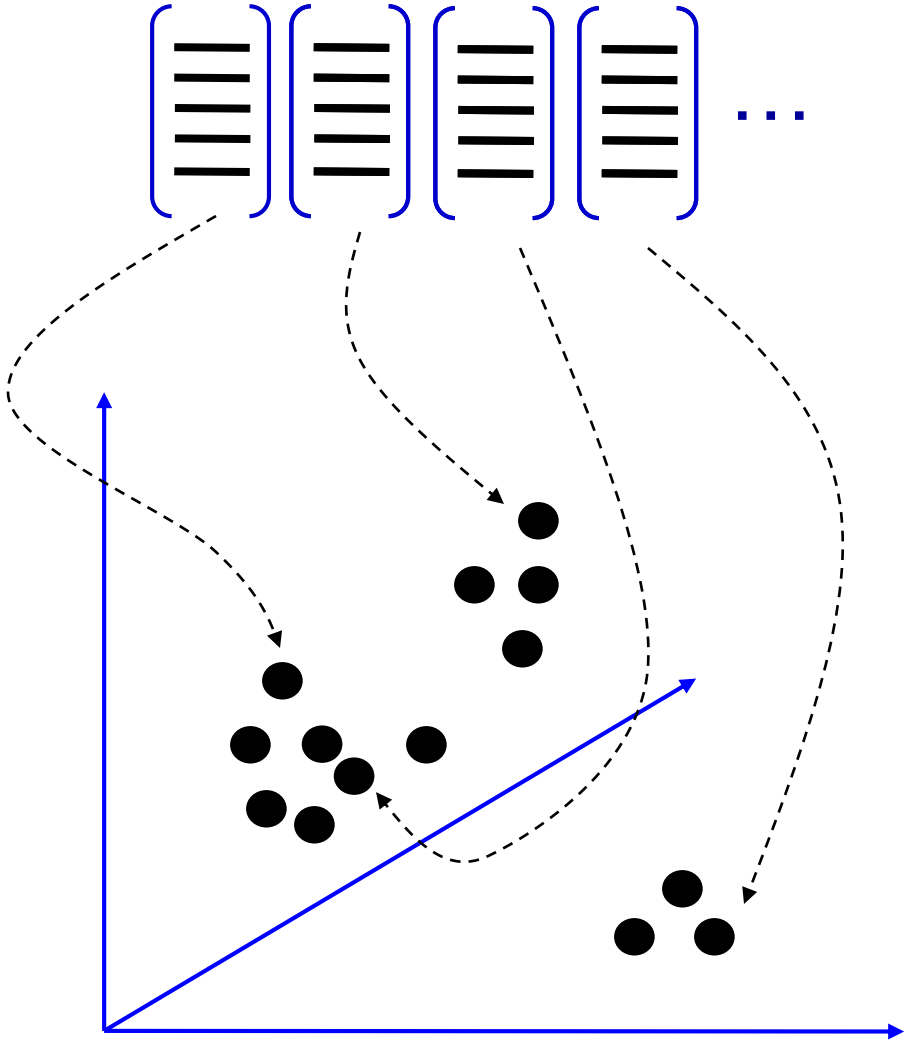
Slide credit: Josef Sivic

1. Feature extraction



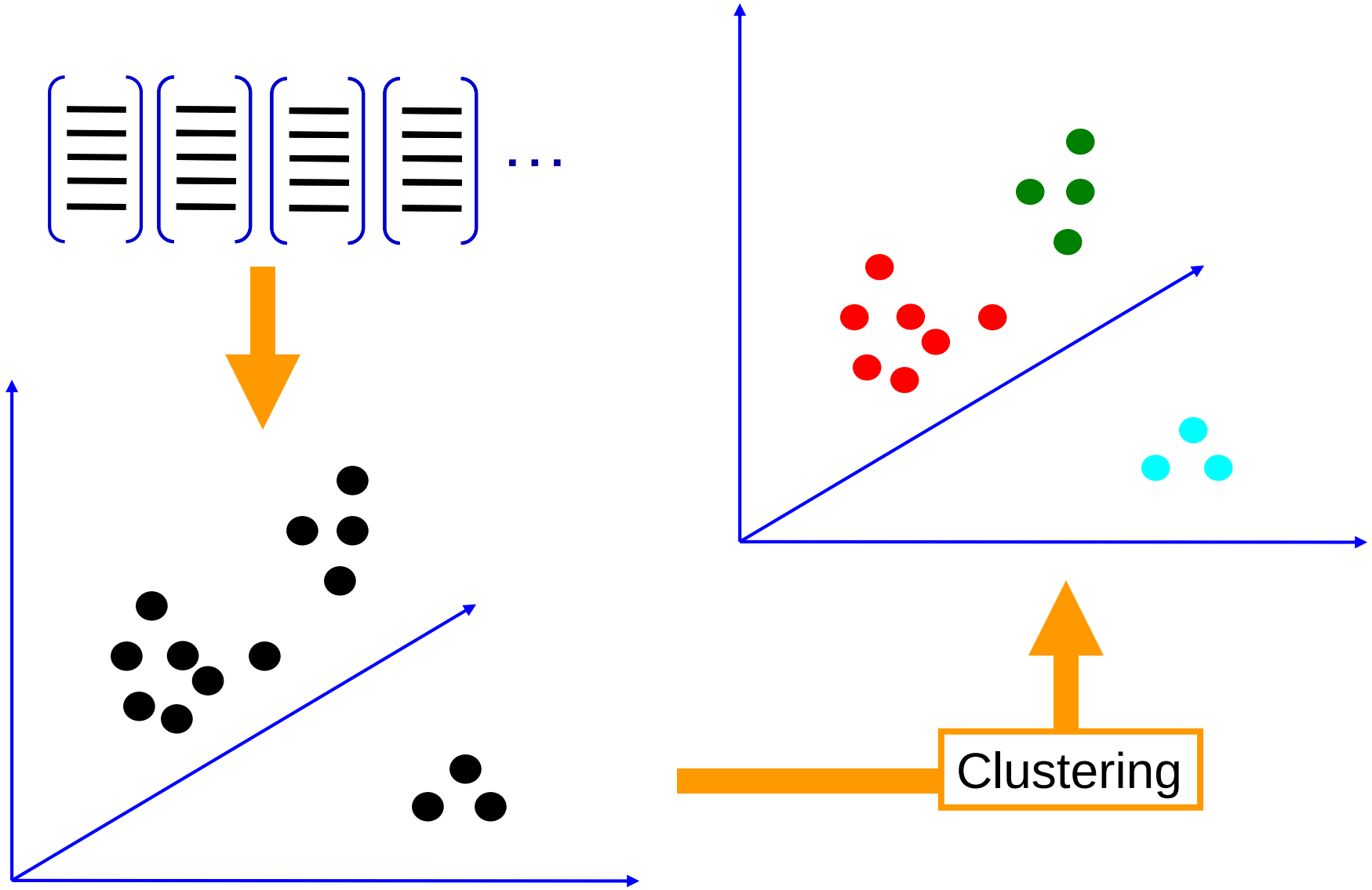
Slide credit: Josef Sivic

2. Learning the visual vocabulary



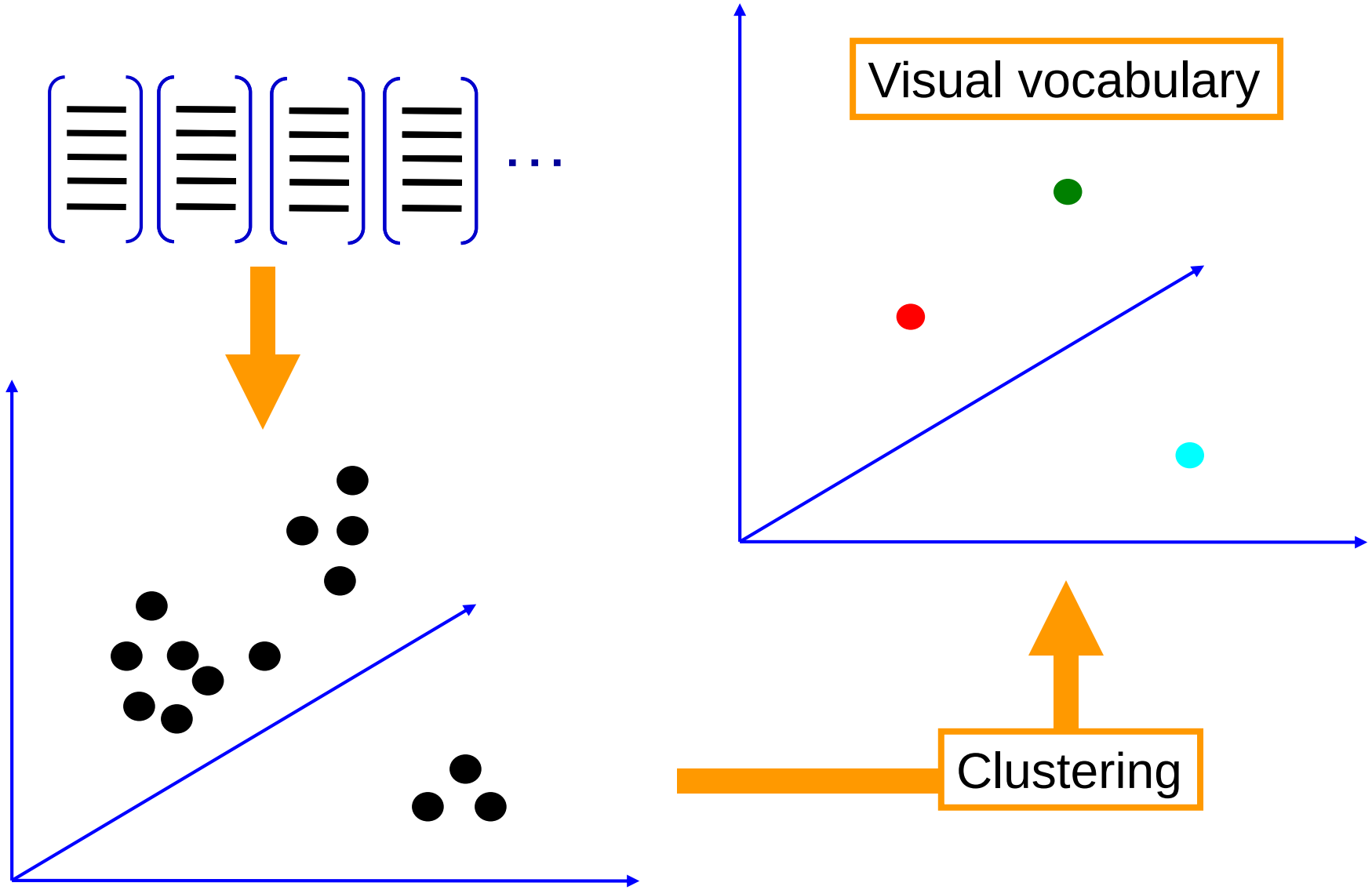
Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

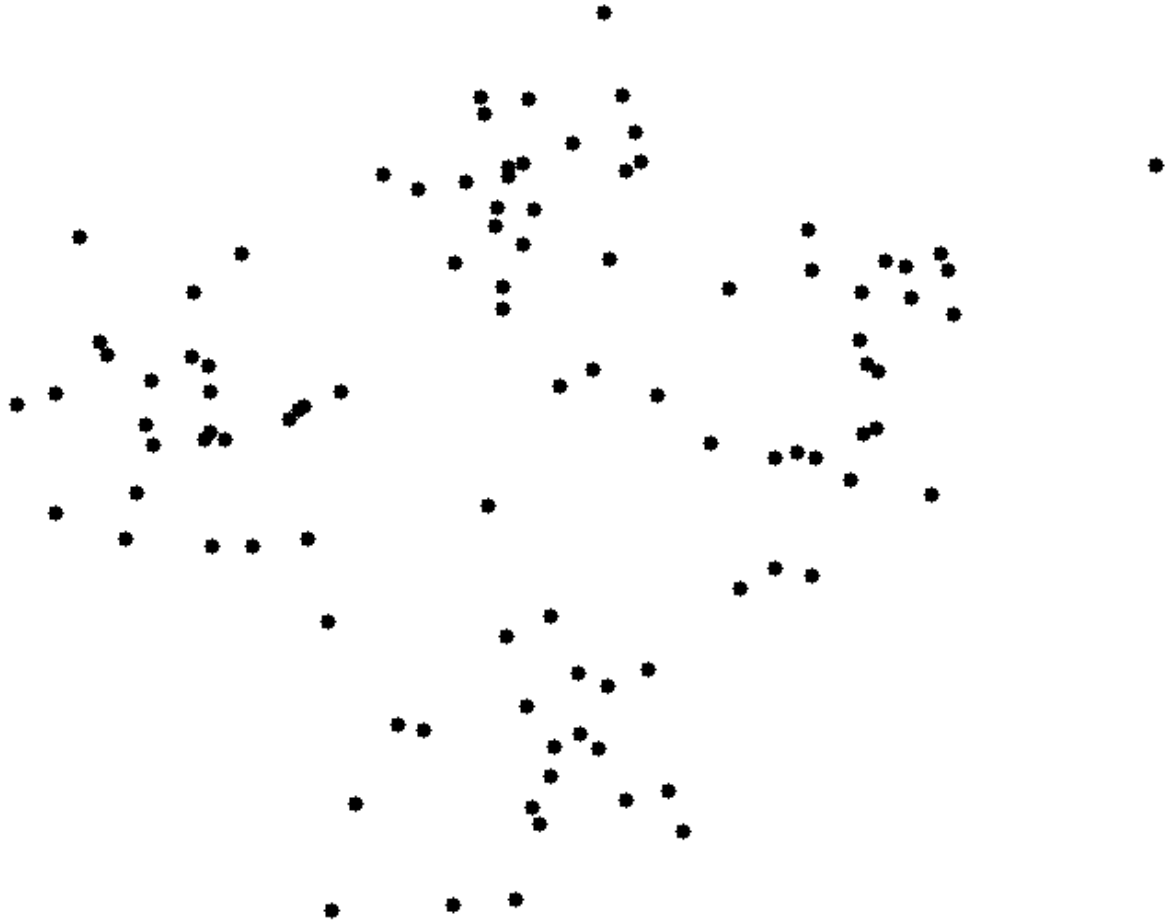
K-means clustering

- Want to minimize sum of squared Euclidean distances between points $\mathbf{x}_i$ and their nearest cluster centers $\mathbf{m}_k$

$$D(X, M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$

- Algorithm:
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it

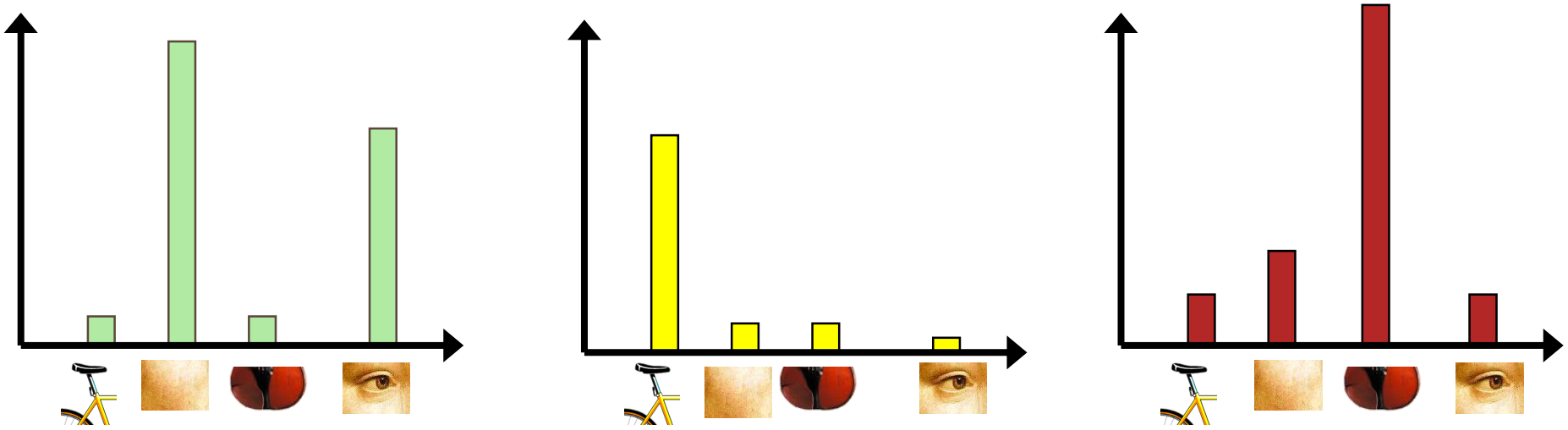
K-means demo



Source: http://shabal.in/visuals/kmeans/1.html
Another demo: http://www.kovan.ceng.metu.edu.tr/~maya/kmeans/

Image Representation

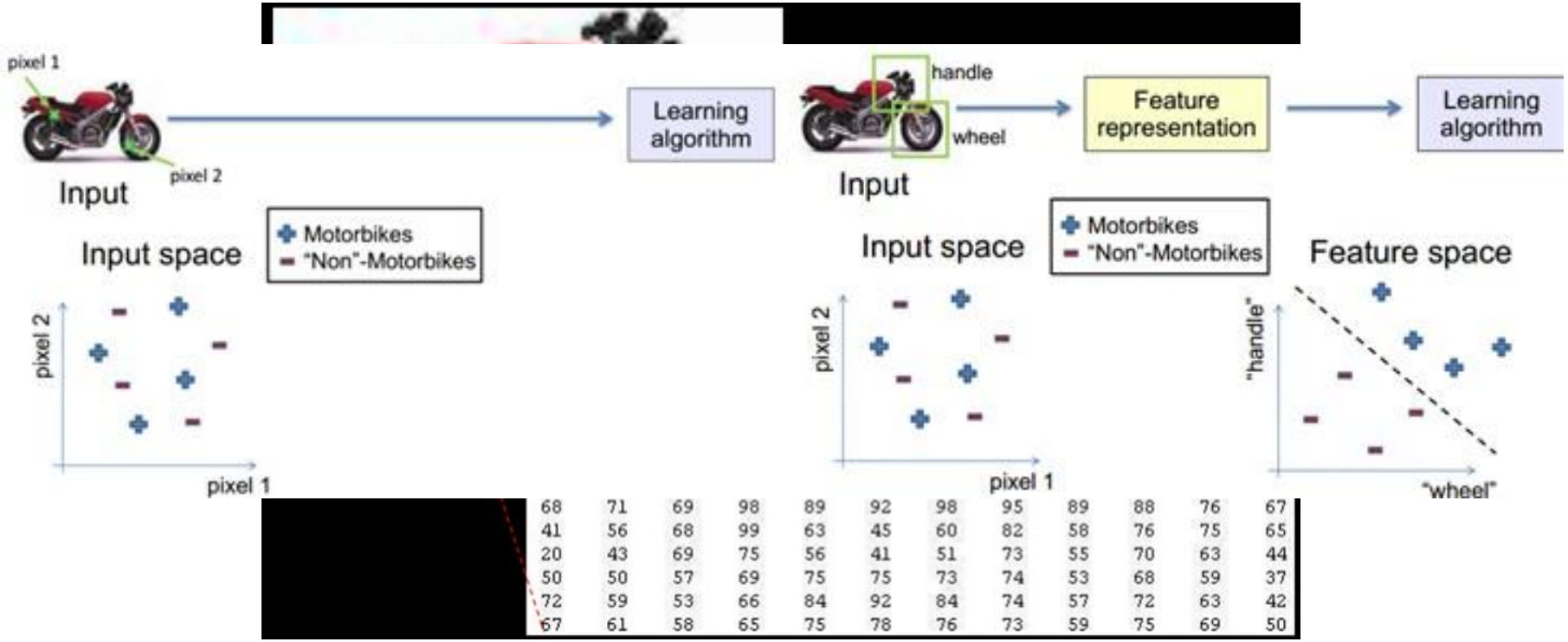
- 词典构建完了怎么用呢？
- 先初始化一个100个bin (也就是100个聚类中心, 可自己指定大小) 的初始值为0的直方图h。
- 每一幅图像不是有很多patch么？我们就再次计算这些patch和和每一个质心的距离，看看每一个patch离哪一个质心最近，那么直方图h中相对应的bin就加1
- 然后计算完这幅图像所有的patches之后，就得到了一个bin=100的直方图，
- 然后进行归一化，用这个100维的向量来表示这幅图像。
- 对所有图像计算完成之后，就可以进行分类聚类训练预测之类的了。



Computer Vision

Deep Learning

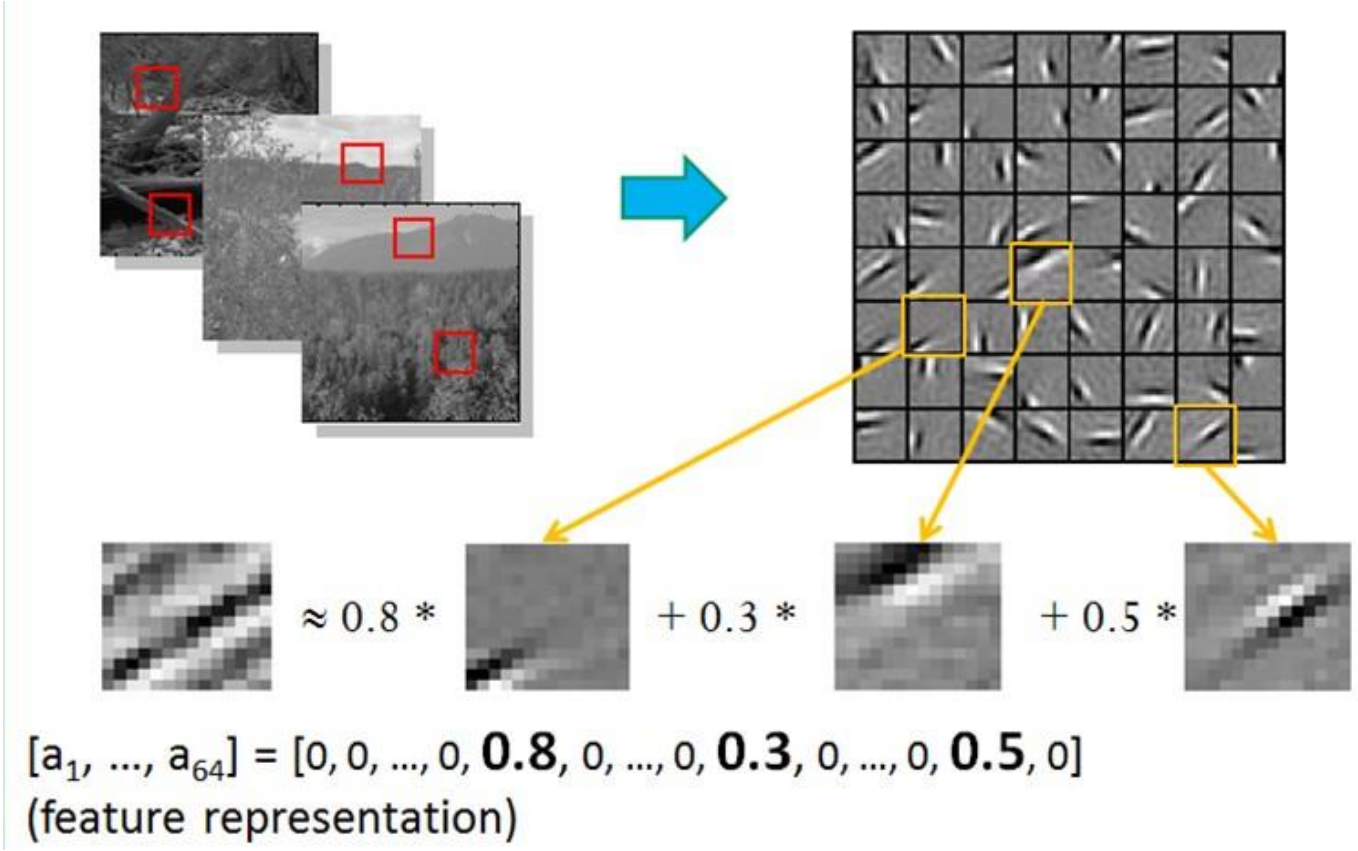
特征表示的粒度



从像素级别，根本得不到任何信息，其无法进行摩托车和非摩托车的区分。而如果特征是一个具有结构性（或者说有含义）的时候，比如是否具有车把手（handle），是否具有车轮（wheel），就很容易把摩托车和非摩托车区分，学习算法才能发挥作用。

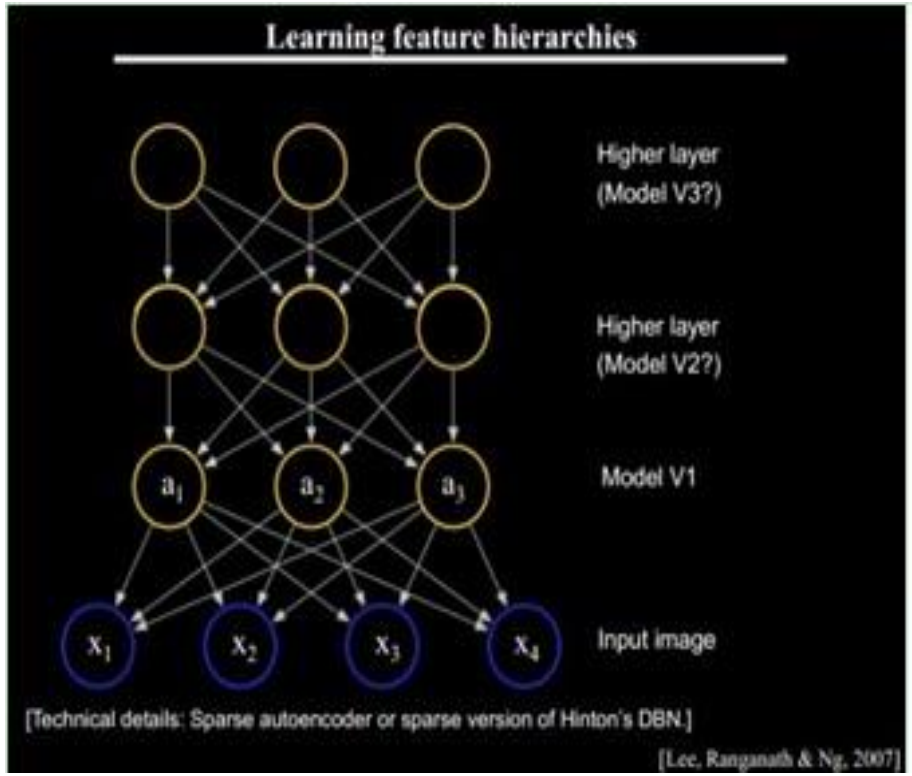
初级（浅层）特征表示

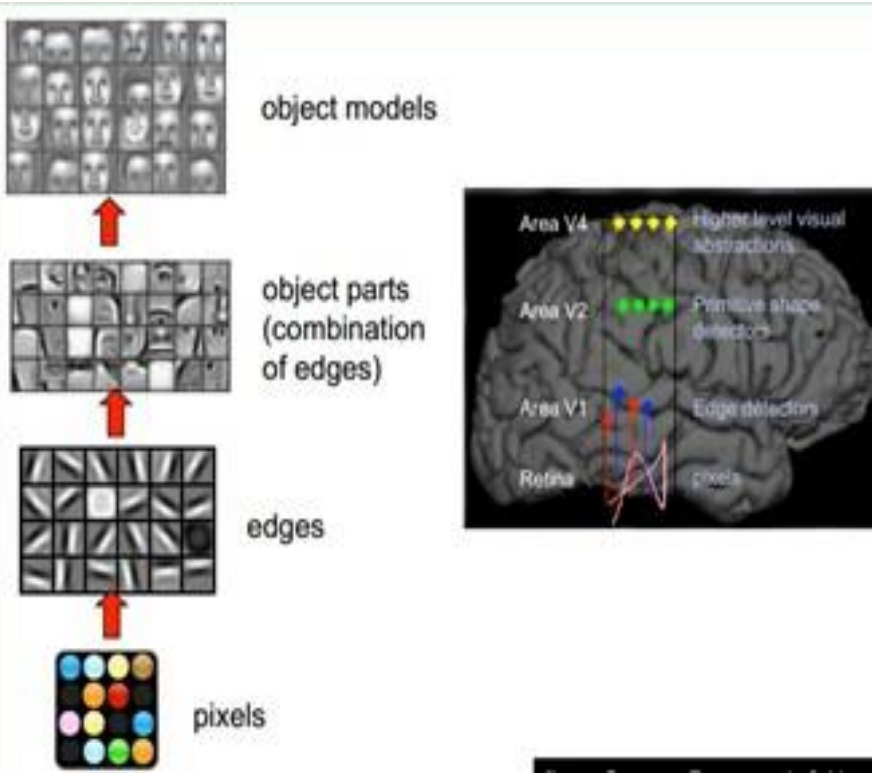
- Sparse coding[Bruno Olshausen& David Field]



结构性特征表示

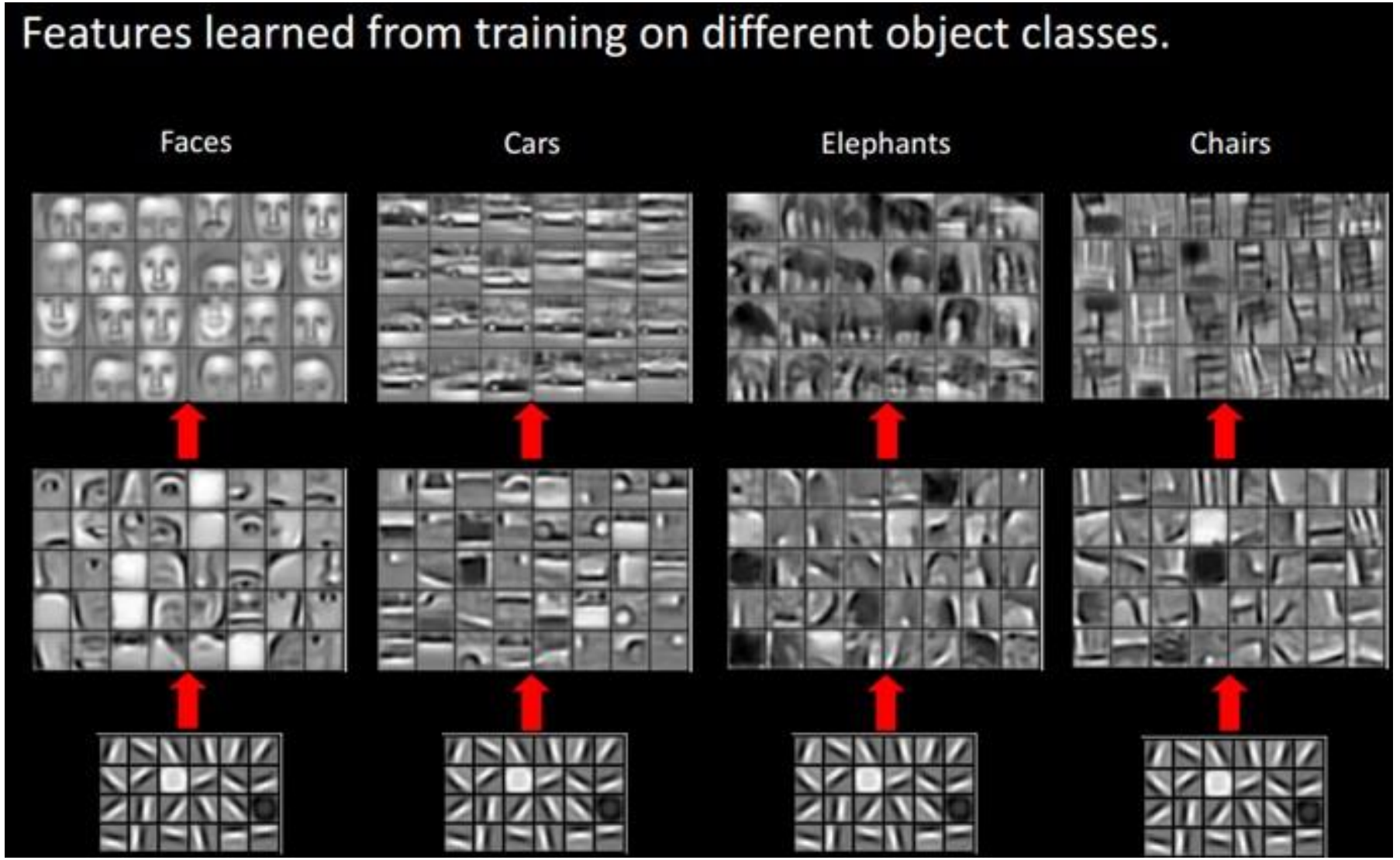
- Deep Learning





V1取提出的basis是边缘，然后V2层是V1层这些basis的组合，这时候V2区得到的又是高一层的basis。即上一层的basis组合的结果，上上层又是上一层的组合basis……

直观上说，就是找到make sense的小patch再将其进行combine，就得到了上一层的feature，递归地向上learning feature。



在不同object上做training是，所得的edge basis 是非常相似的，但object parts和models就会不同了

Thanks