SIFT

scale-invariant feature transform (SIFT)

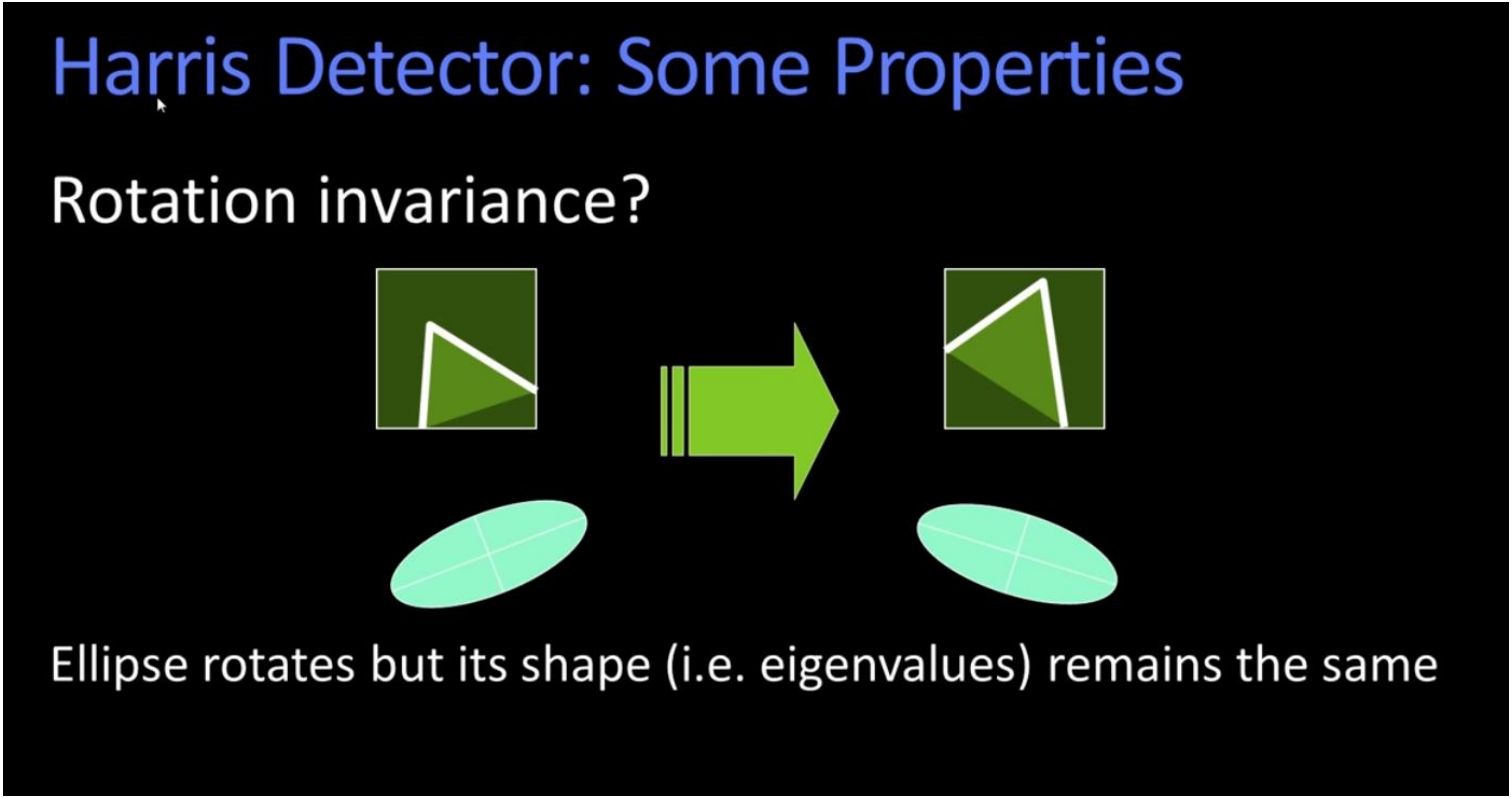
Distinctive image features from scale-invariant keypoints

DG **Lowe** - International journal of computer vision, **2004** - Springer

This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching …

☆ 99 Cited by 55116 Related articles All 155 versions

Harris: עמיד לשינוי סבוב



Harris: עמיד לשינוי בצבע

Harris Detector: Some Properties

- Mostly invariant to additive and multiplicative intensity changes (threshold issue for multiplicative)

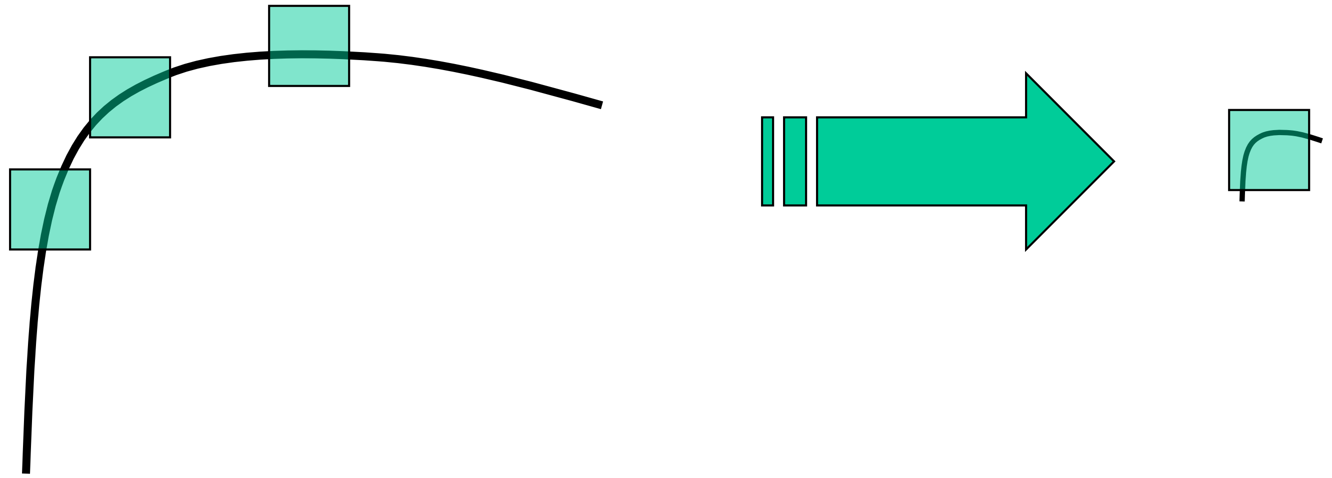
  ✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

Properties of the Harris corner detector

Translation invariant? Yes

Rotation invariant? Yes

Scale invariant? No



All points will be classified as edges

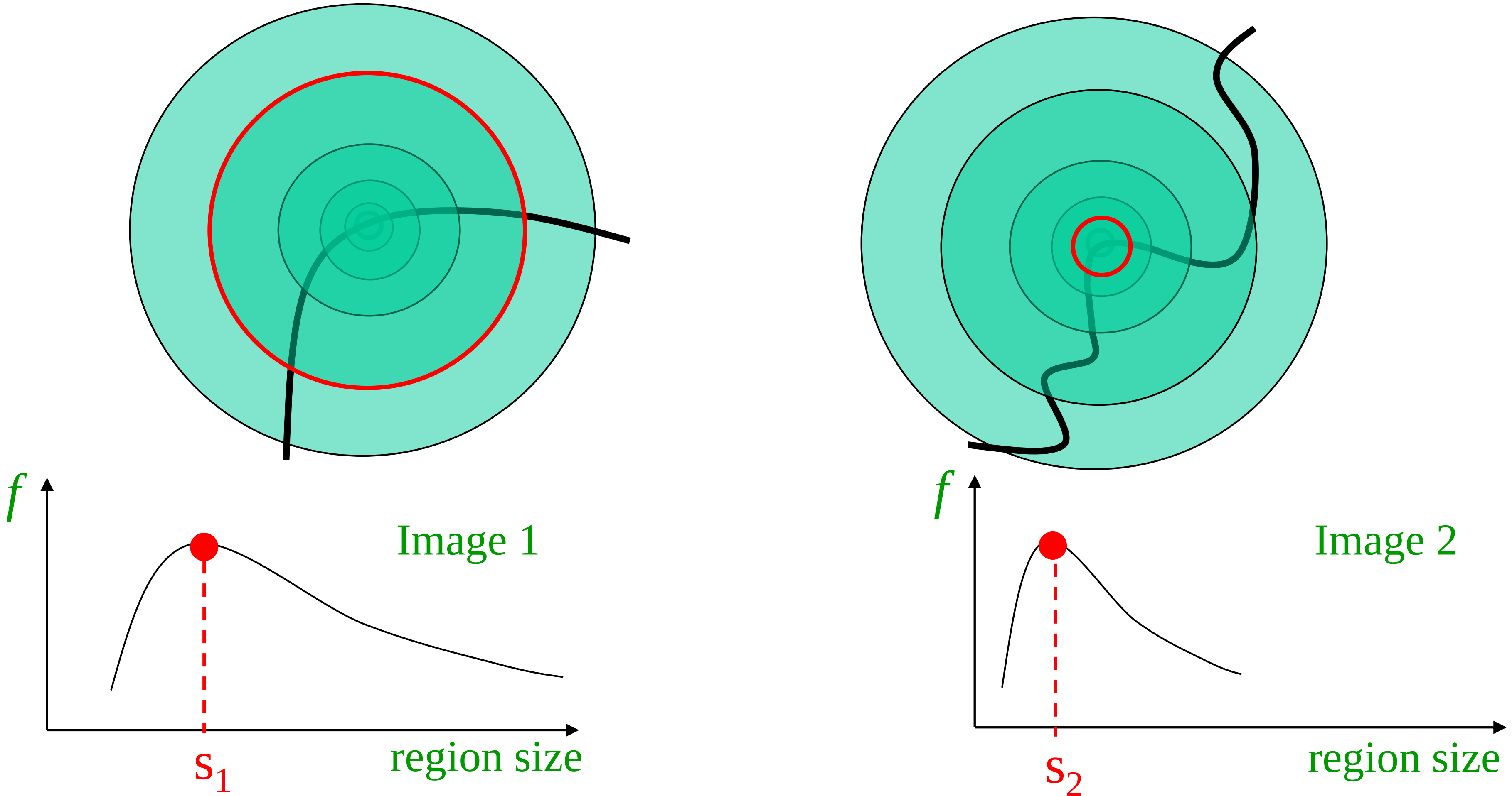
Corner !

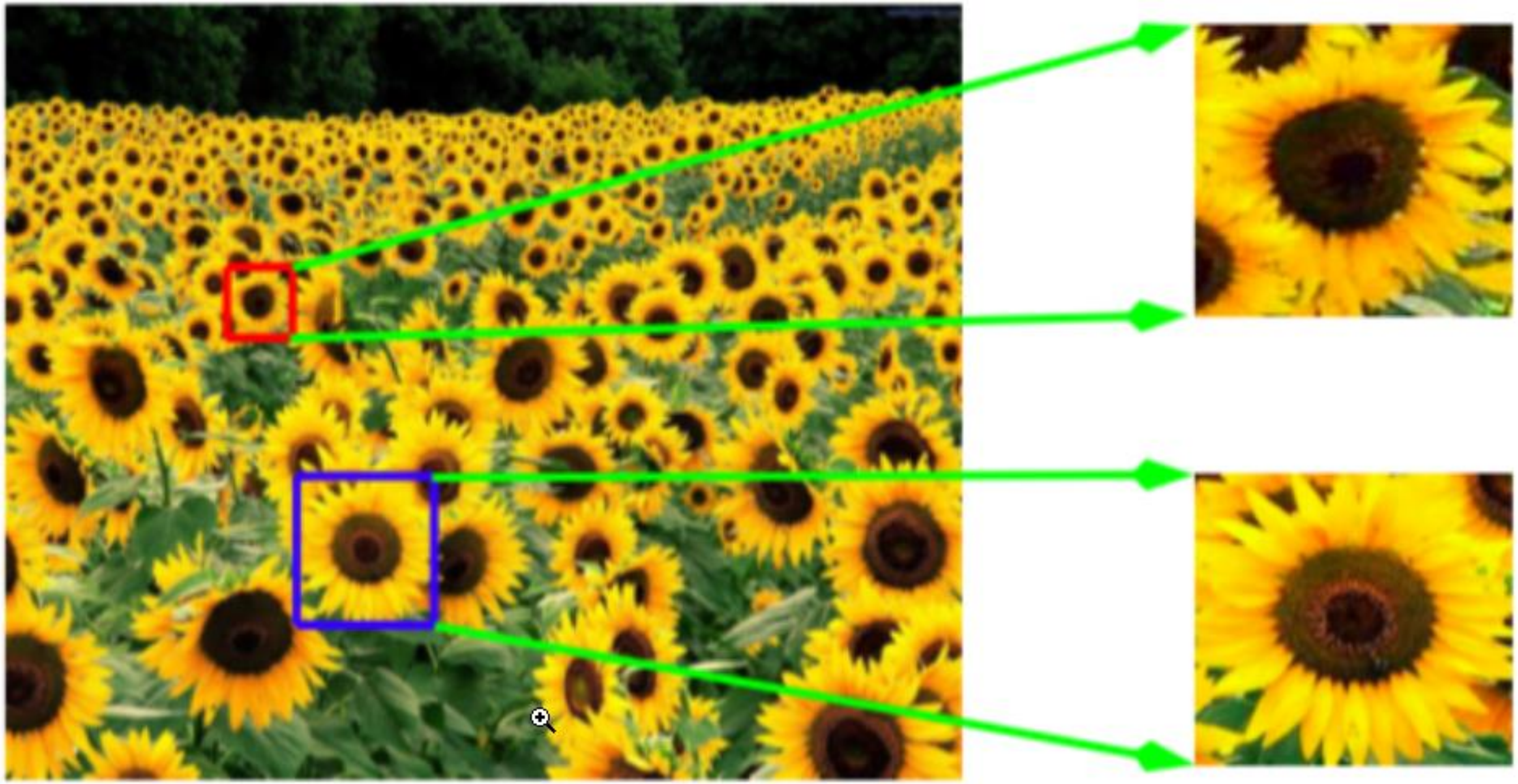
Automatic scale selection

פונקציה שעמידה לשינוי
בגודל -- מציאת מקסימום
לפונקציה מסוימת

Intuition:

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

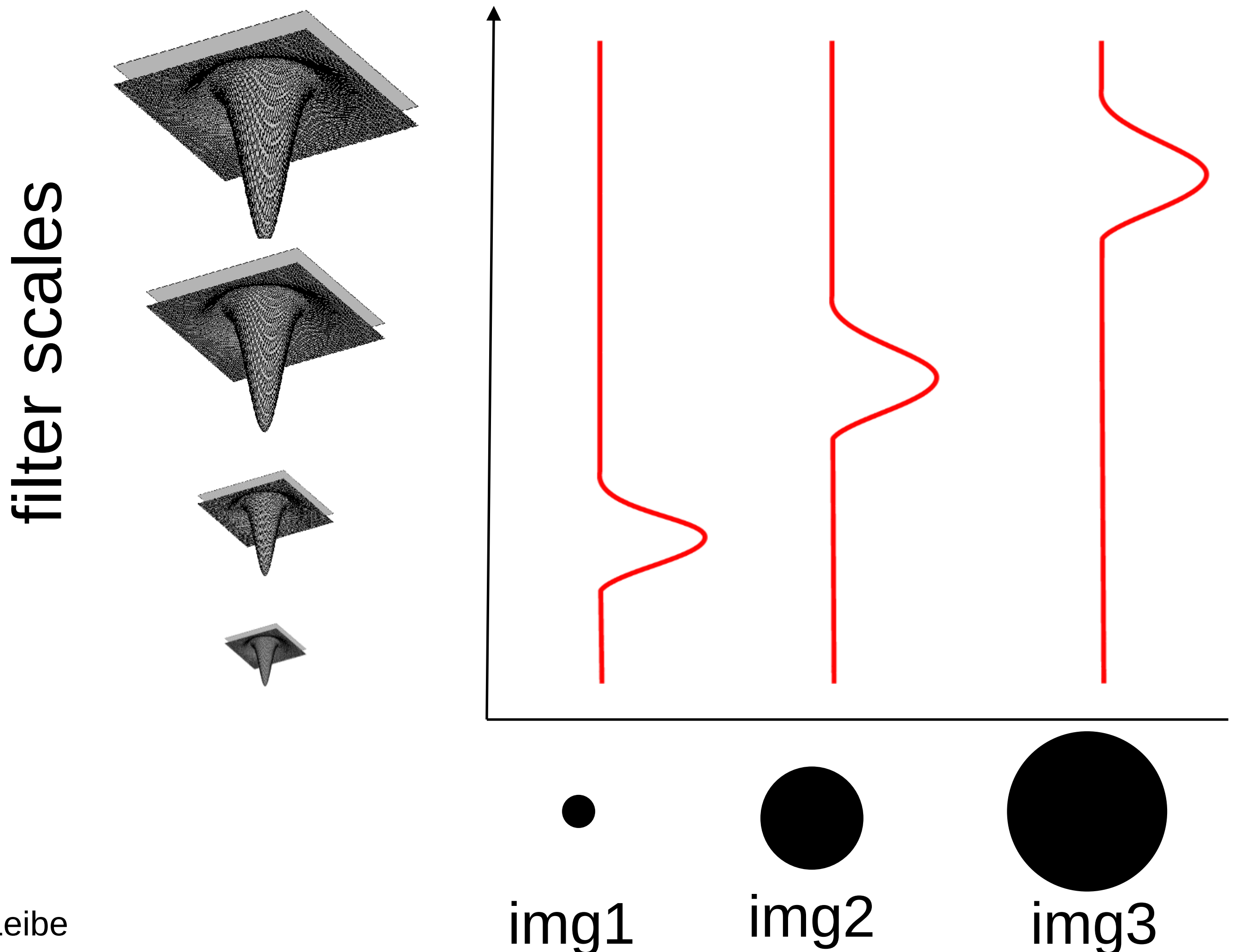


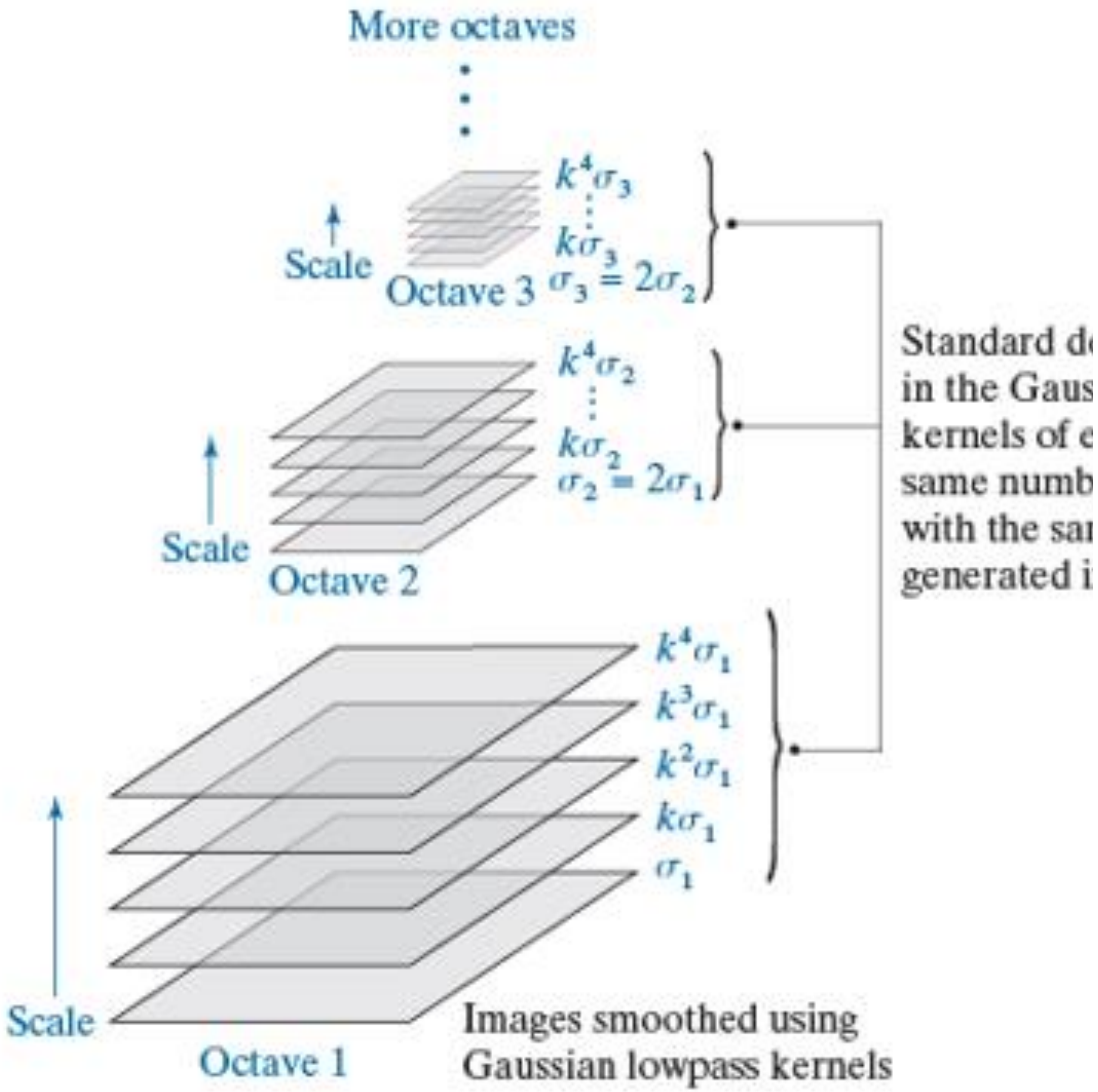


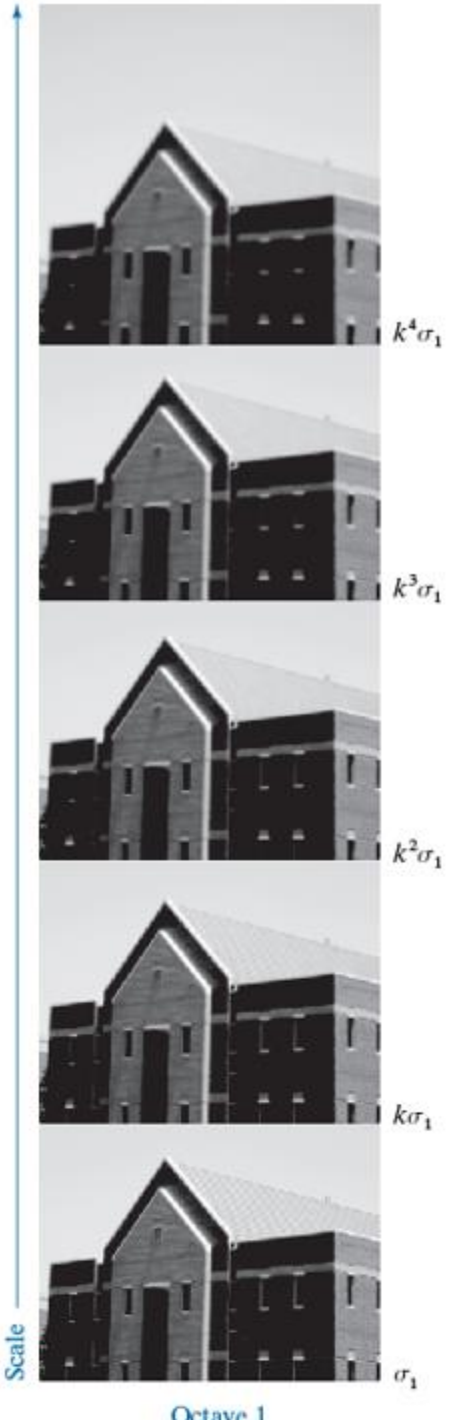
צלעות + פונק' עמידה לשינוי בגודל

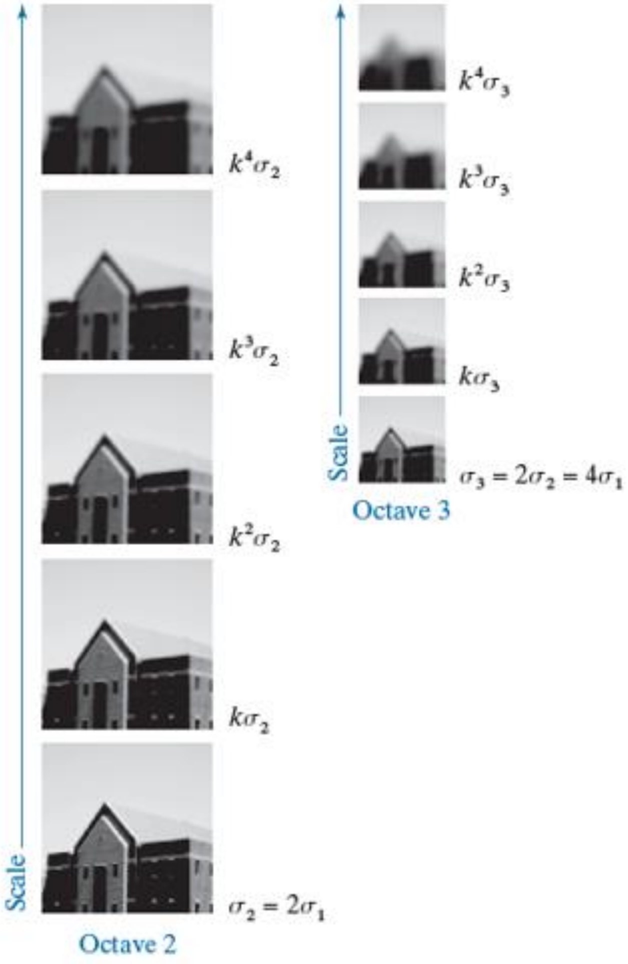
Blob detection in 2D: scale selection

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



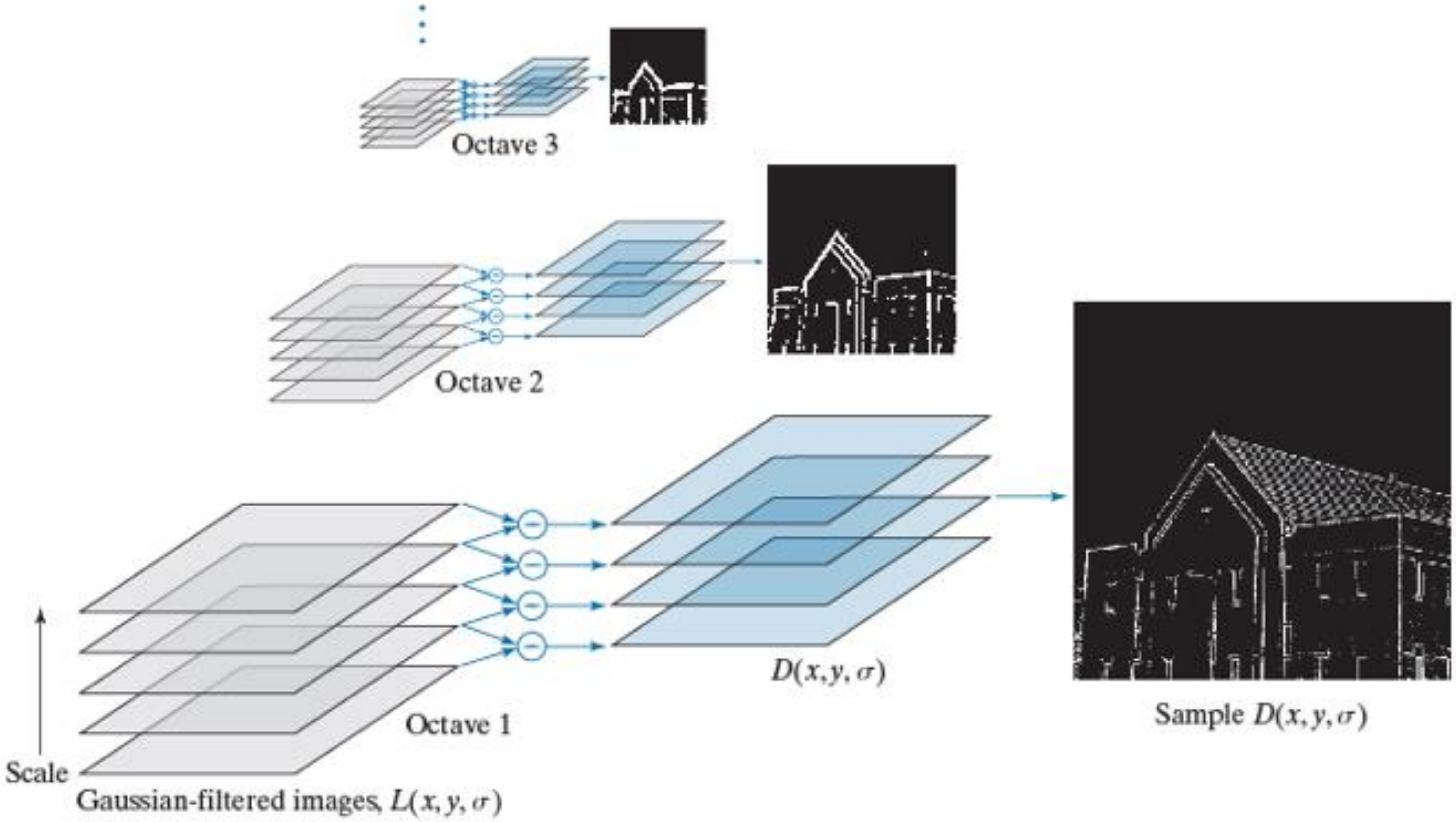




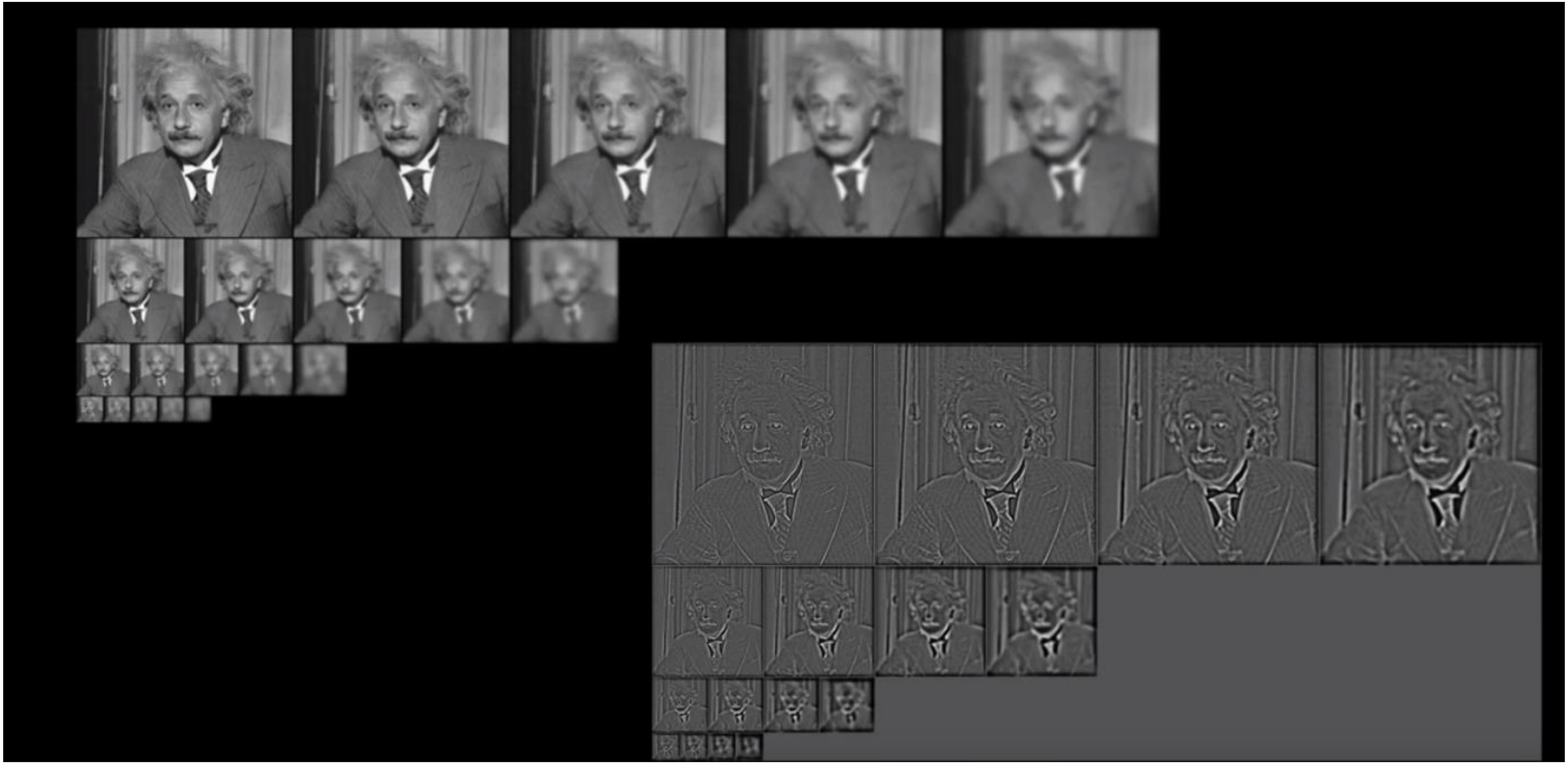


$\sigma_1 = \sqrt{2}/2 = 0.707 \quad k = \sqrt{2} = 1.414$

Octave	Scale 1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

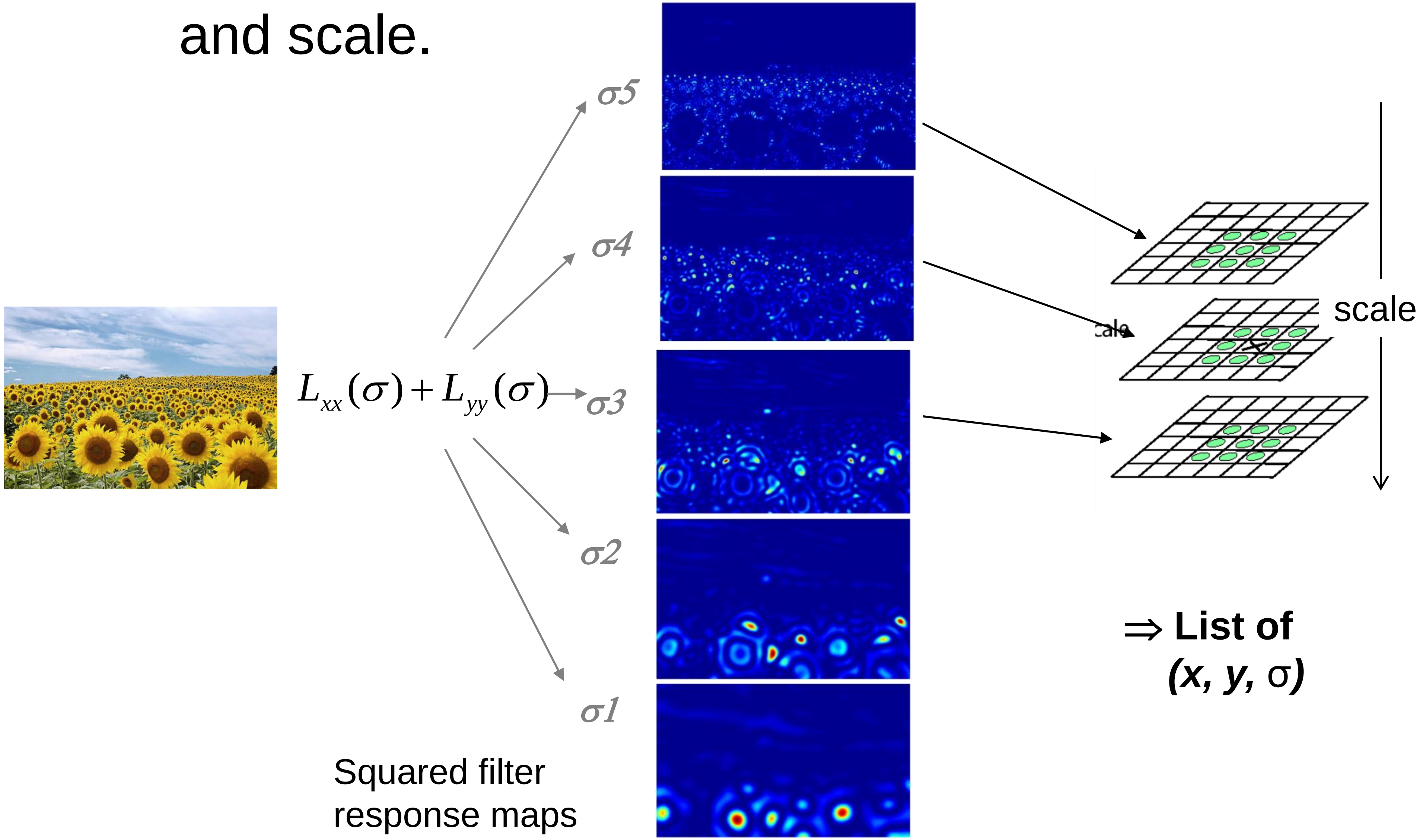

Octave 3
Octave 2
Octave 1
Scale
Gaussian-filtered images, $L(x, y, \sigma)$
$D(x, y, \sigma)$
Sample $D(x, y, \sigma)$

עוד דוגמא

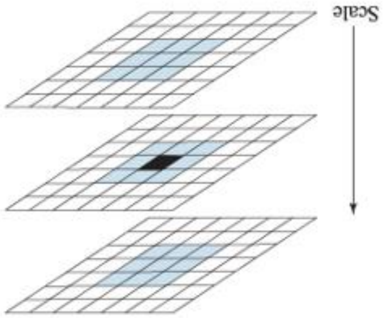


Scale invariant interest points

Interest points are local maxima in both position and scale.

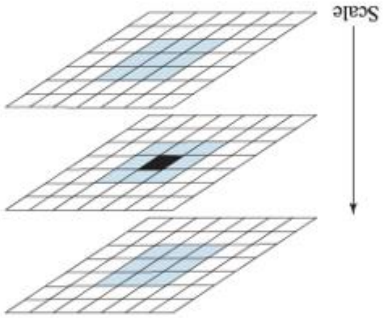


Example

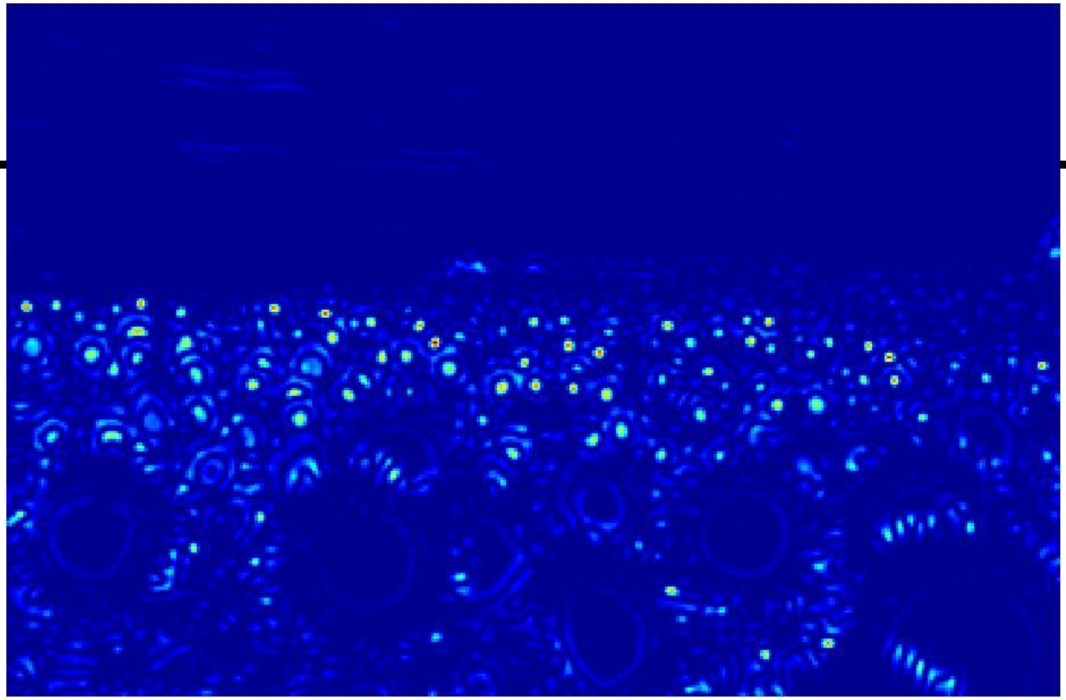


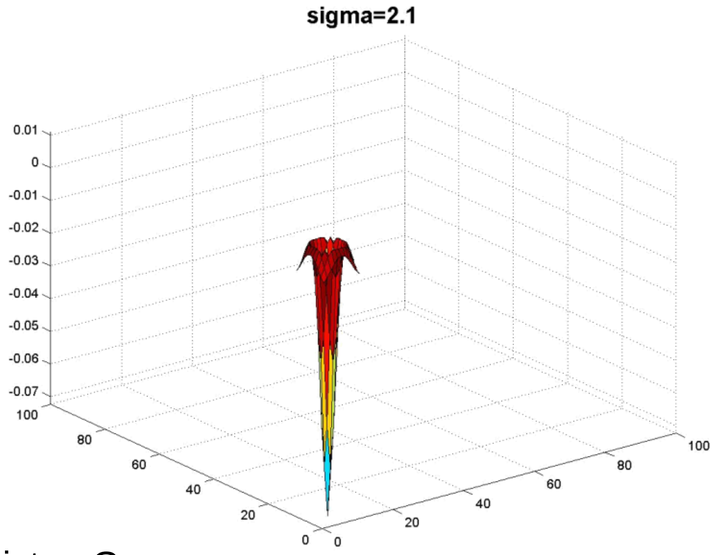
Original image
at ¾ the size

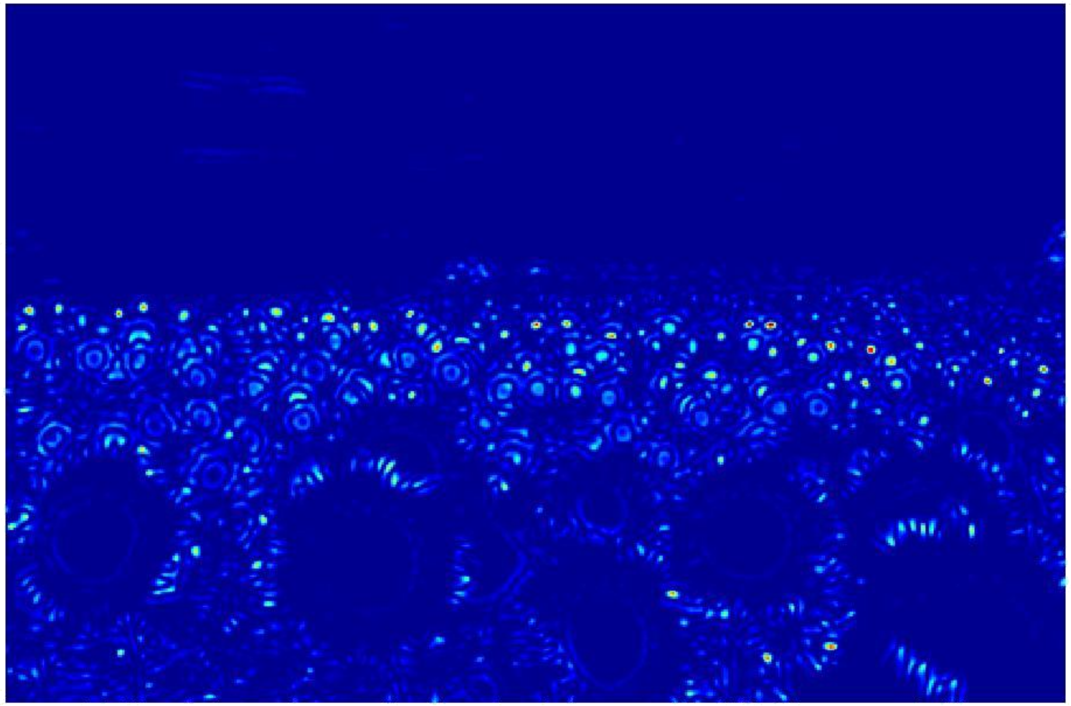
Kristen Grauman



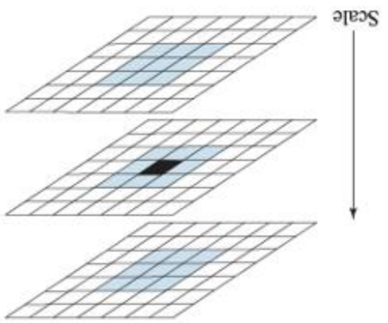
Original image
at ¾ the size

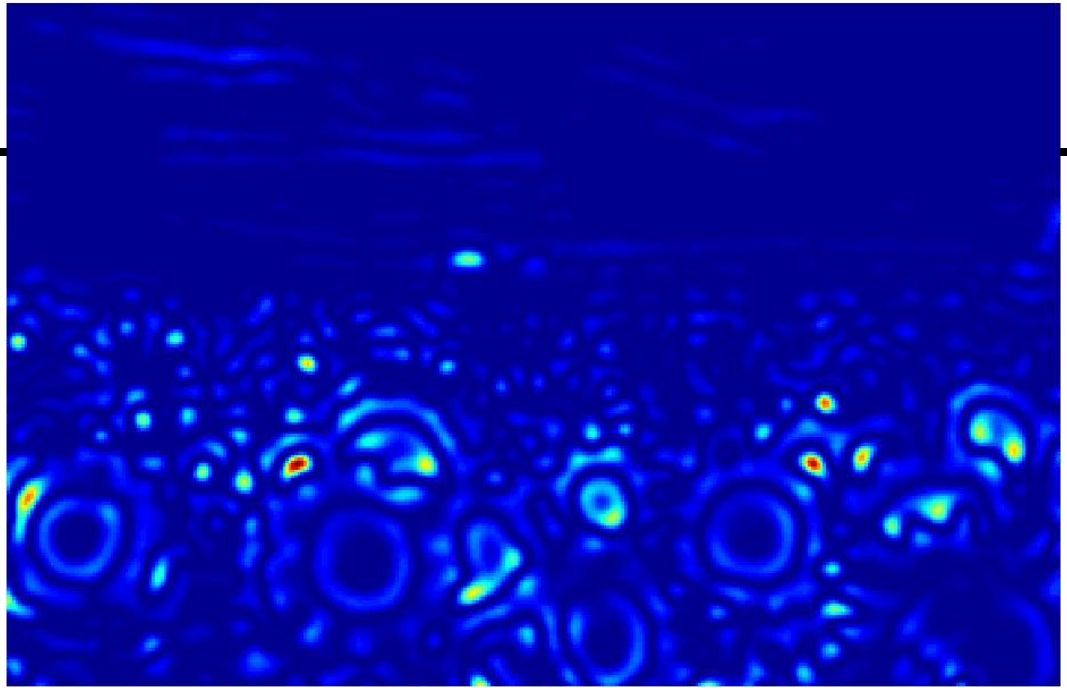


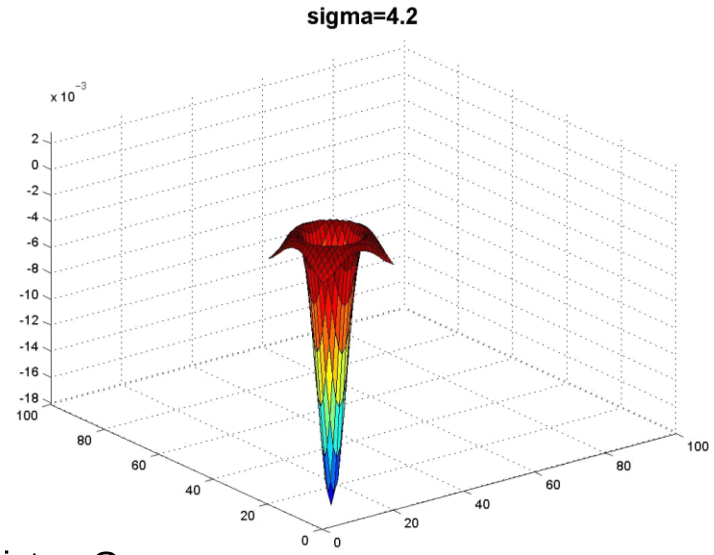


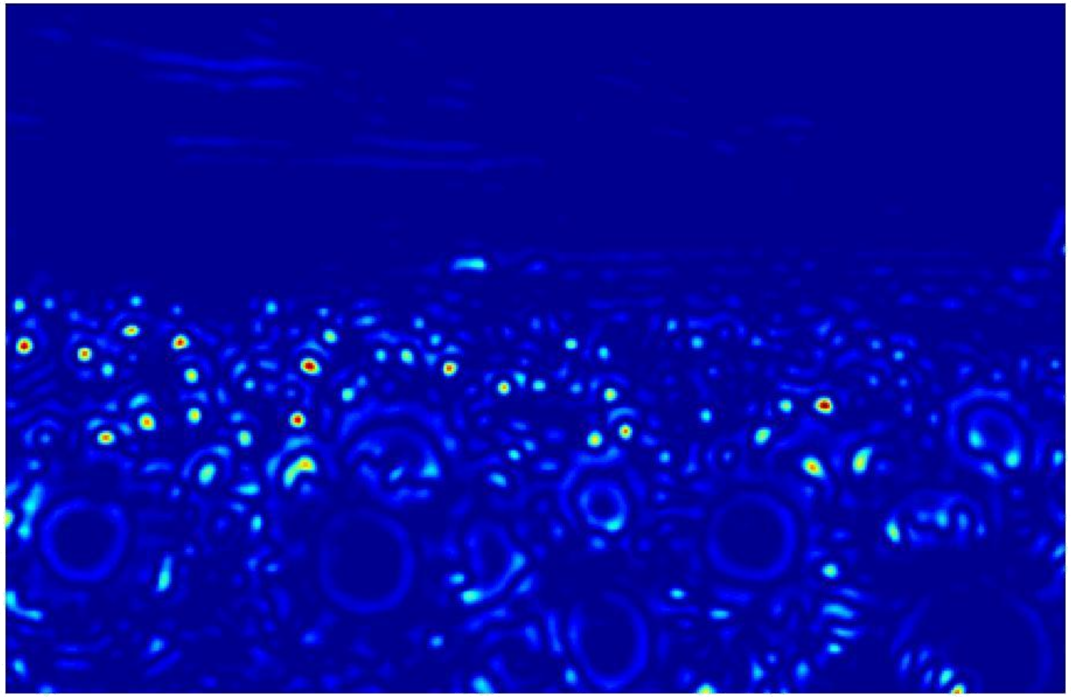


Kristen Grauman

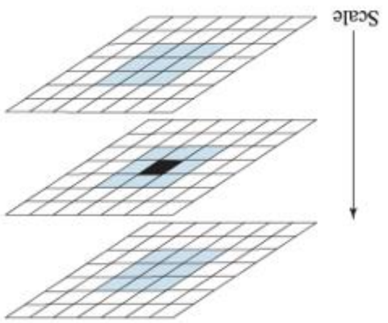


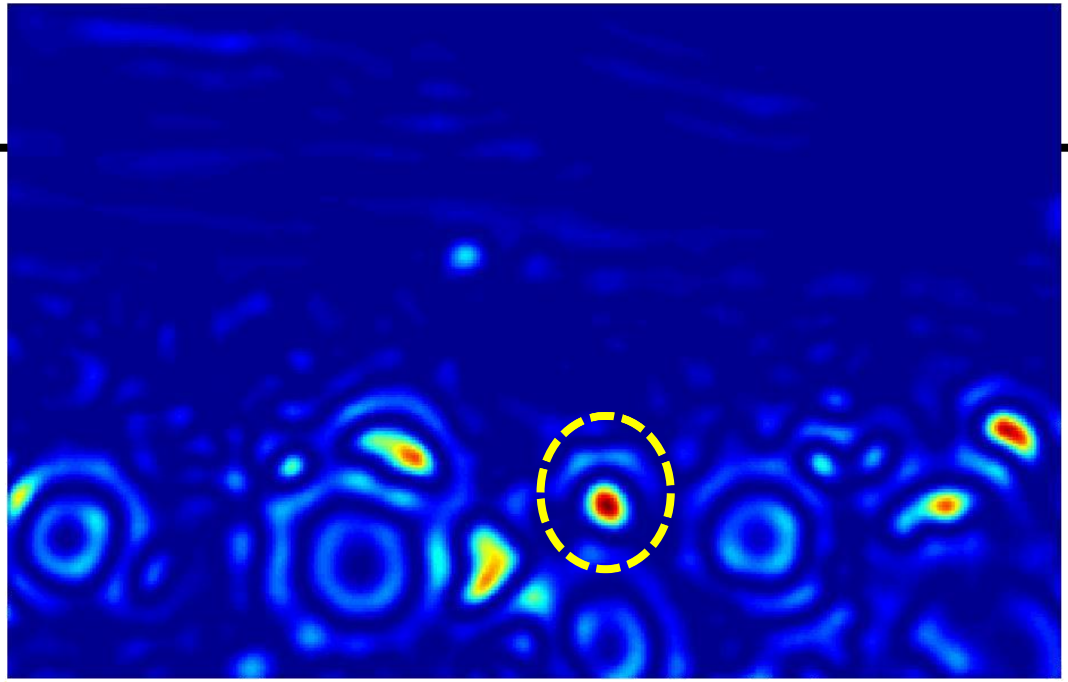


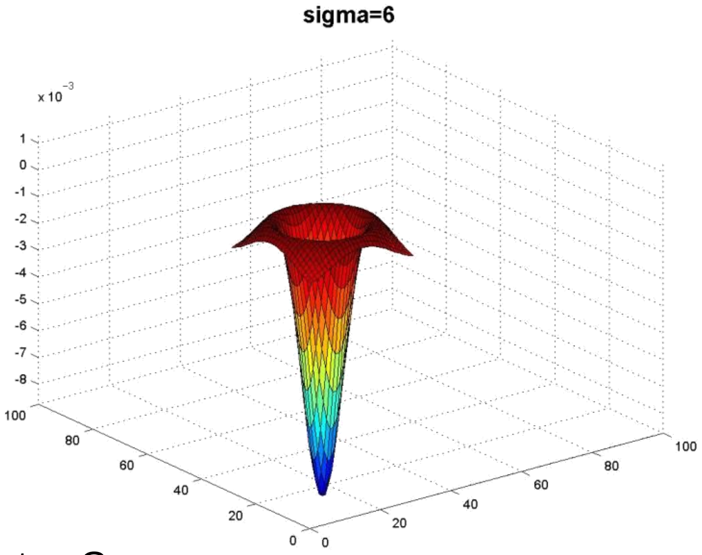




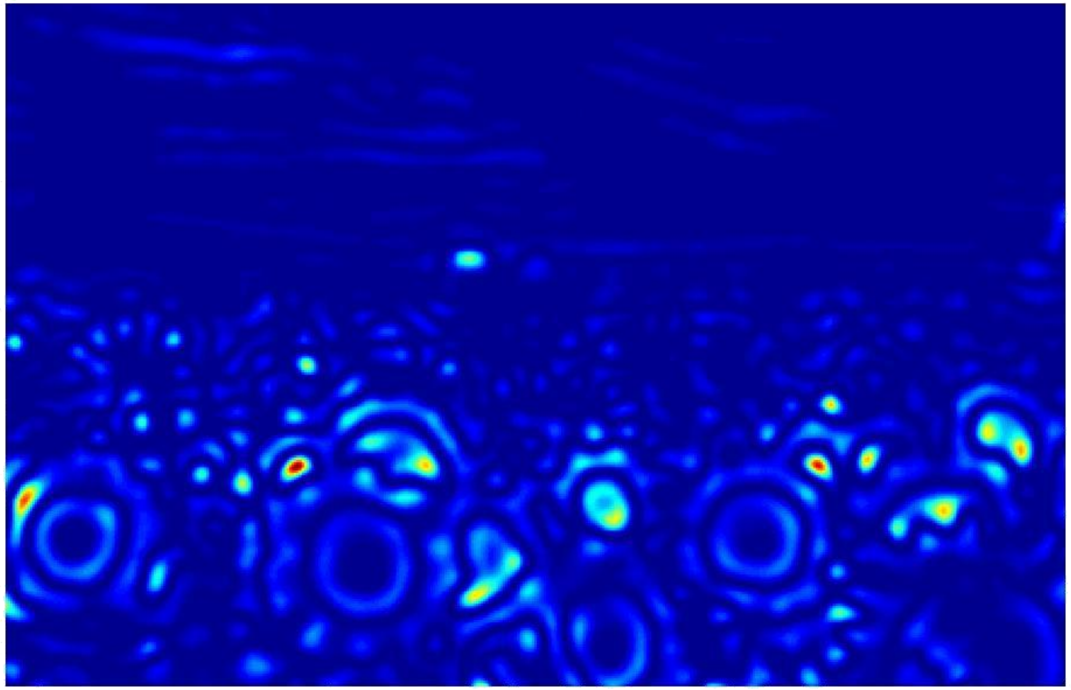
Kristen Grauman

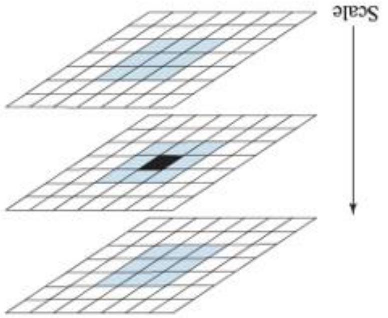


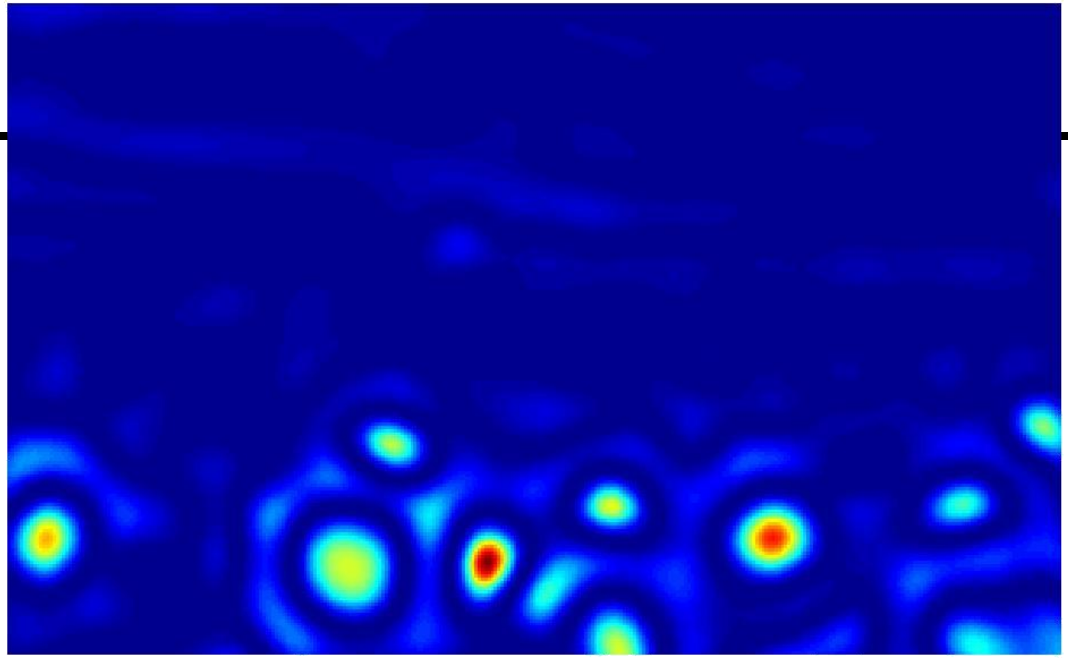


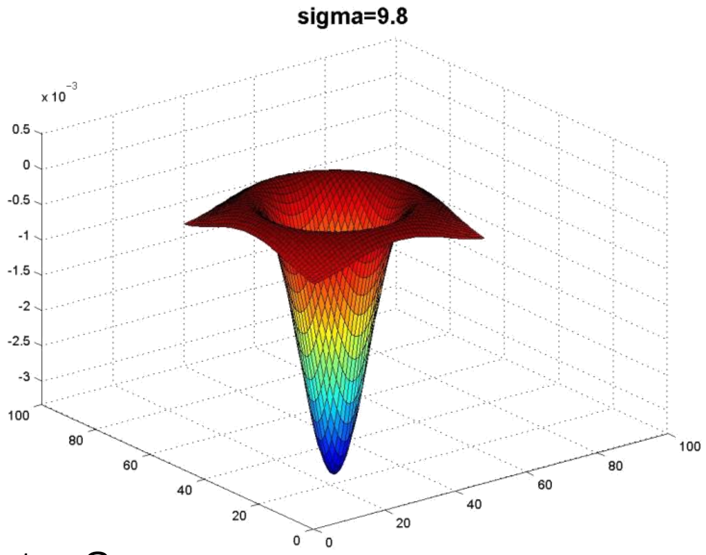


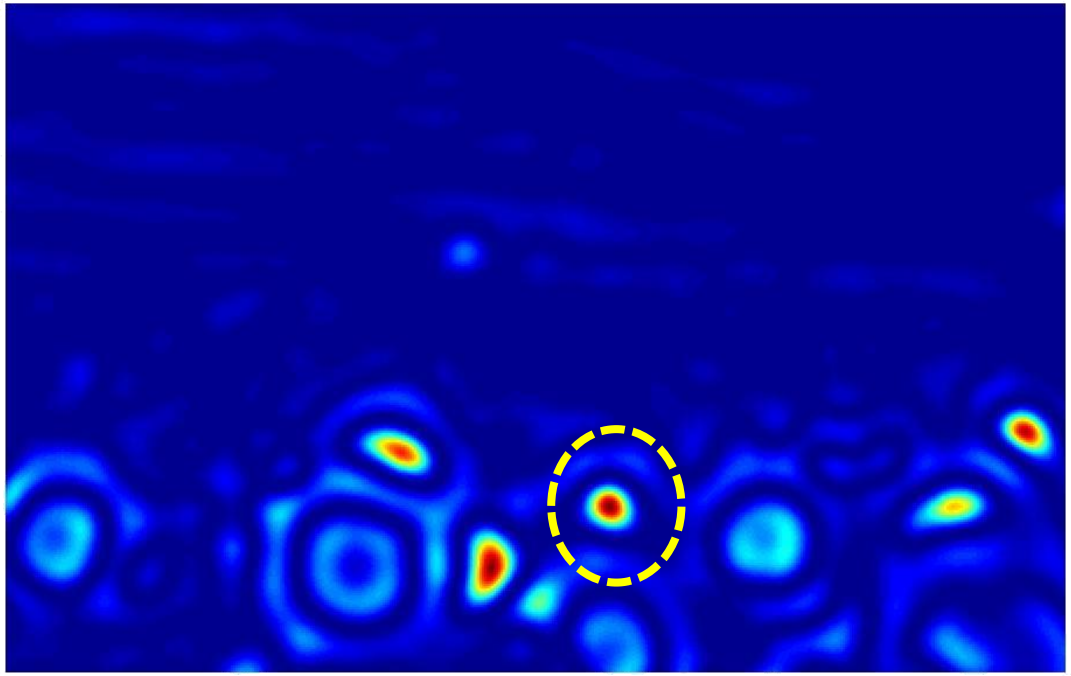
Kristen Grauman



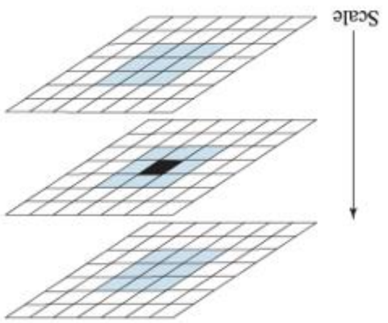


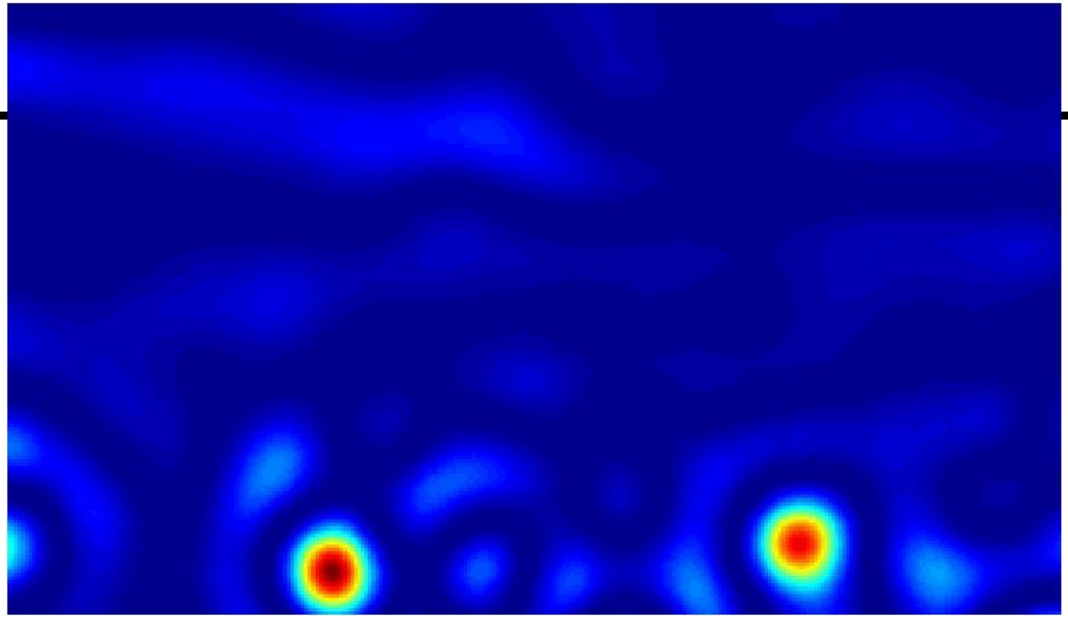


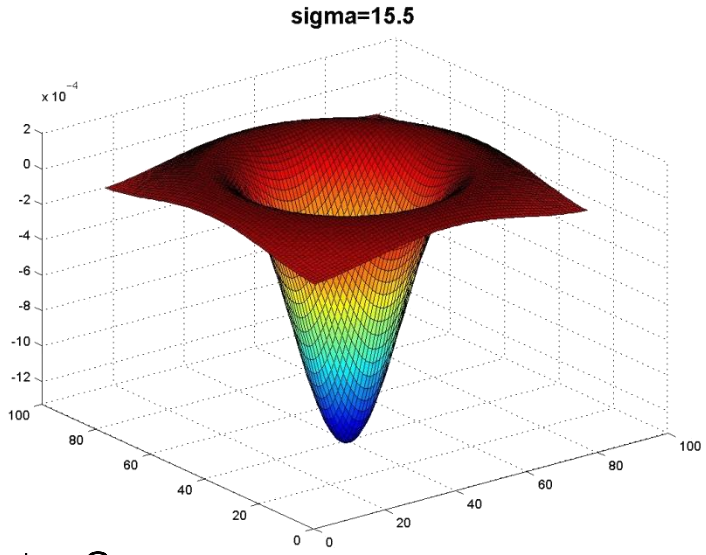


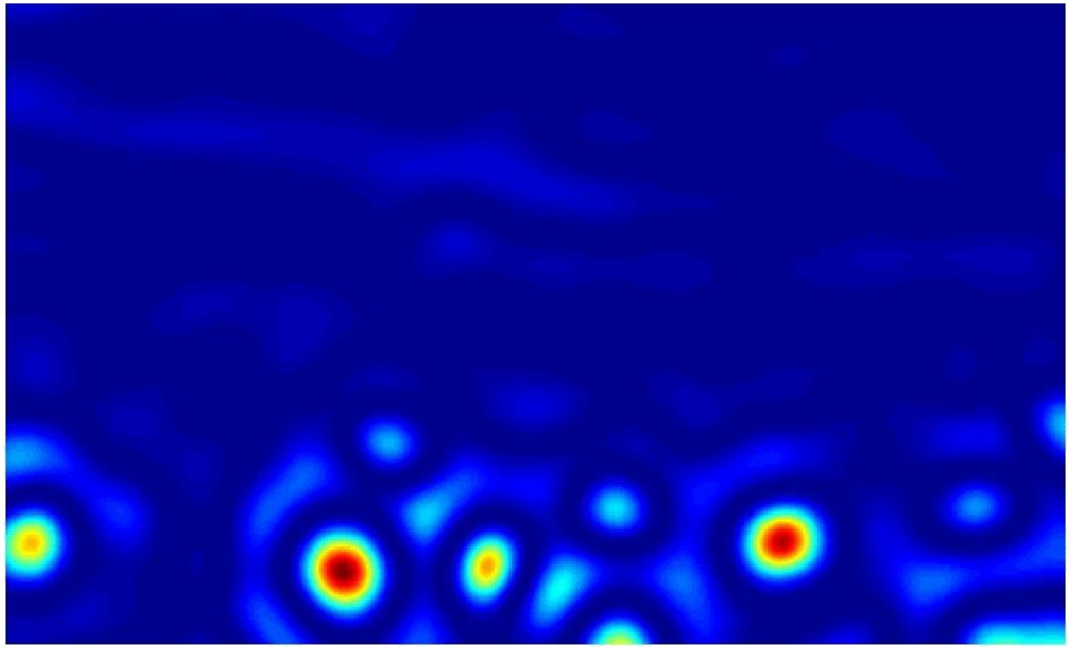


Kristen Grauman





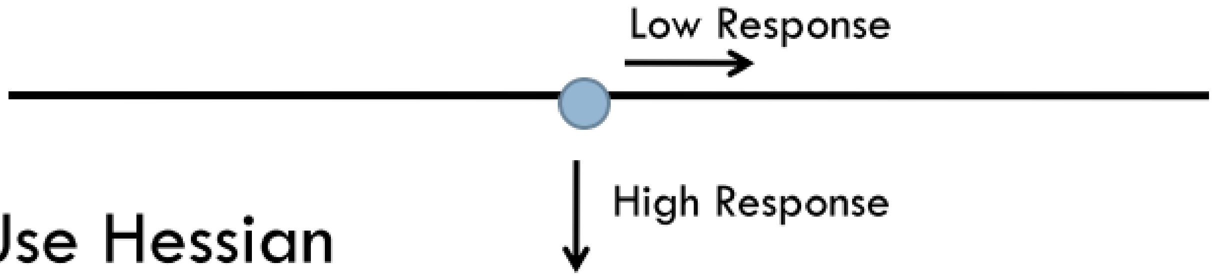




Kristen Grauman

Edge Response Elimination

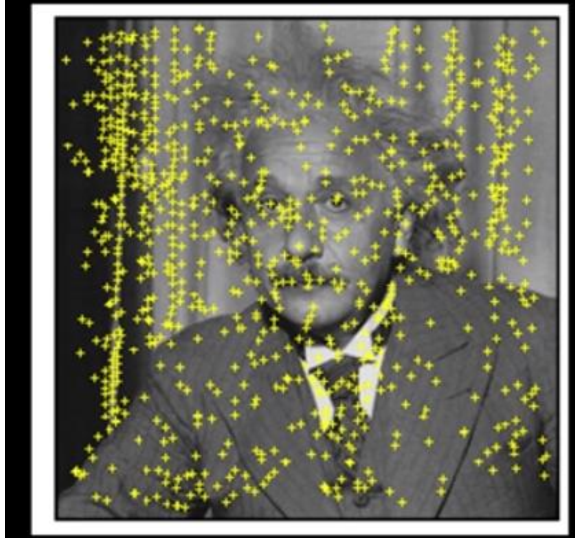
- Peak has high response along edge, poor other direction



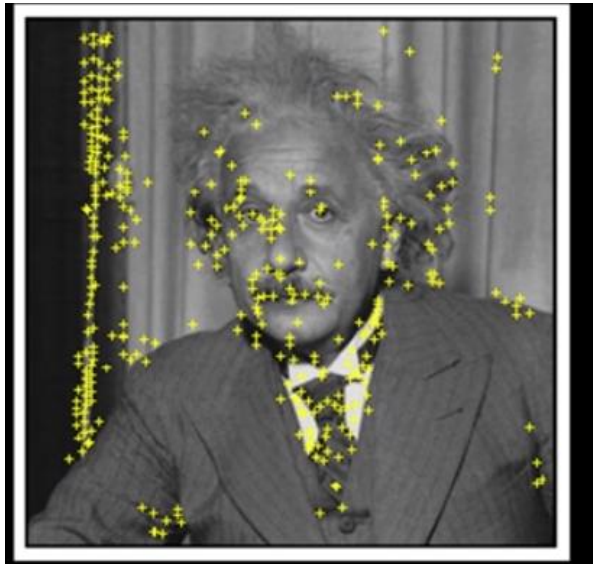
- Use Hessian
  - Eigenvalues Proportional to principle Curvatures
  - Use Trace and Determinant

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta, Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

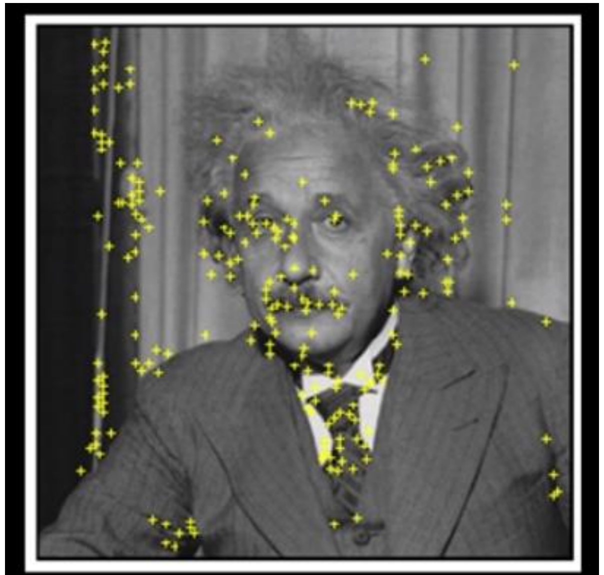
$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$



Extrema points



Contrast > C



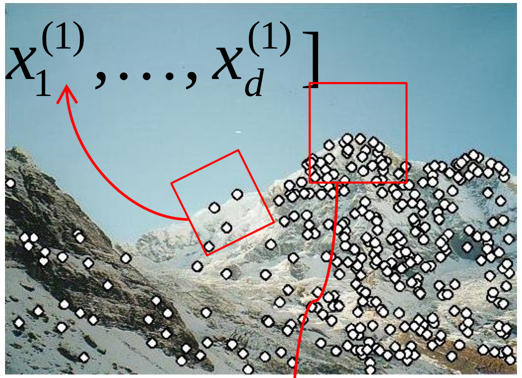
Not on edge

Local features: main components

1) Detection: Identify the interest points

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

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



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

3) Matching: Determine correspondence between descriptors in two views

Overall SIFT Procedure

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment

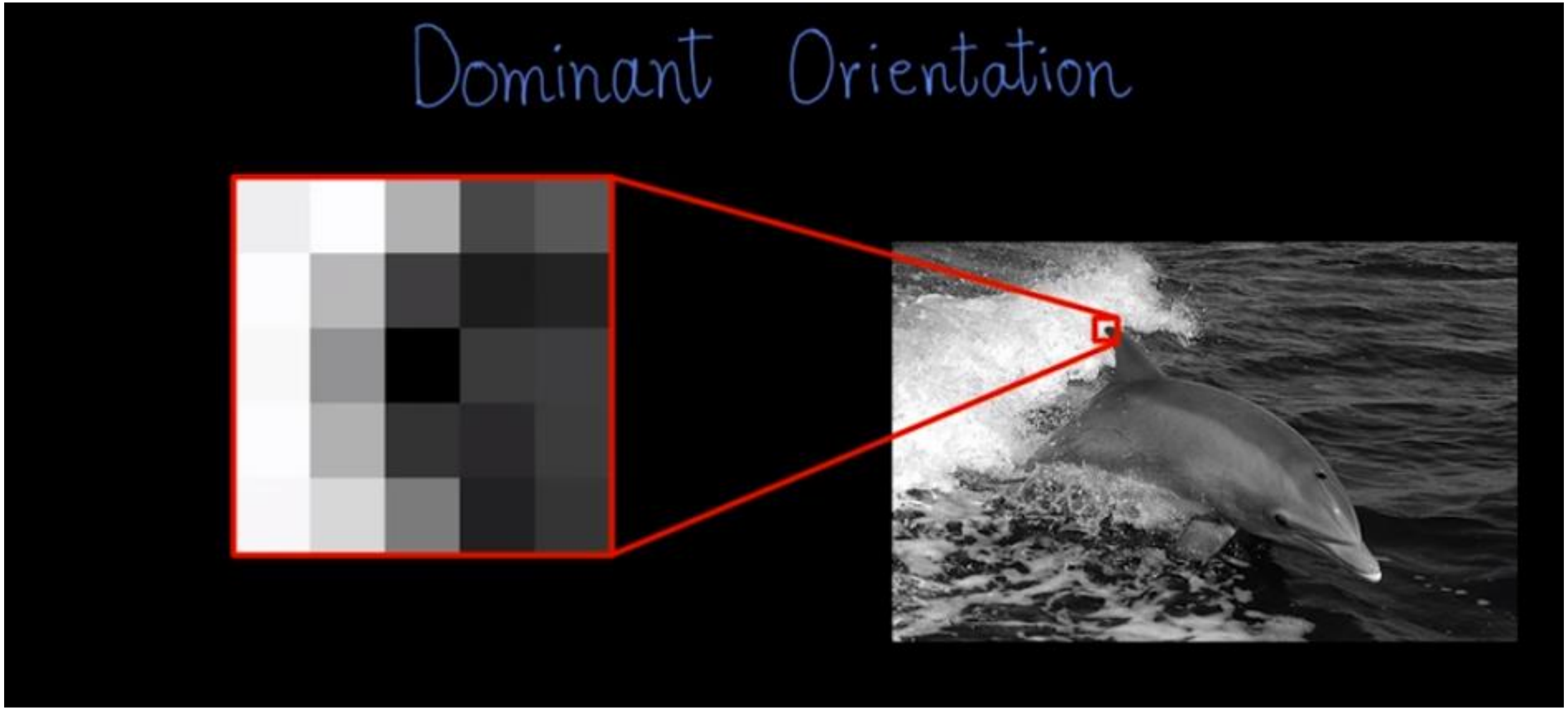
   Compute best orientation(s) for each keypoint region.

4. Keypoint description

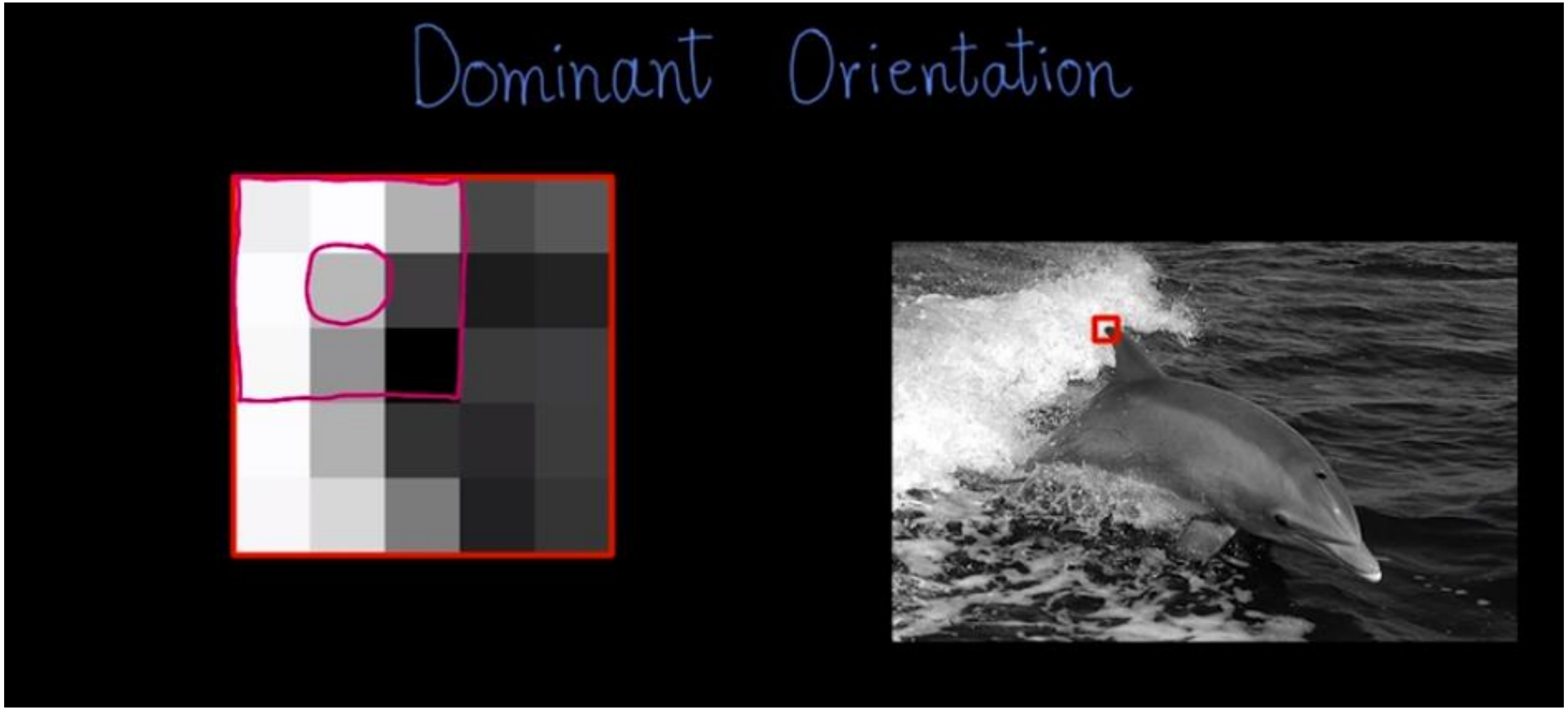
   Use local image gradients at selected scale and rotation to describe each keypoint region.

Dominant Orientation

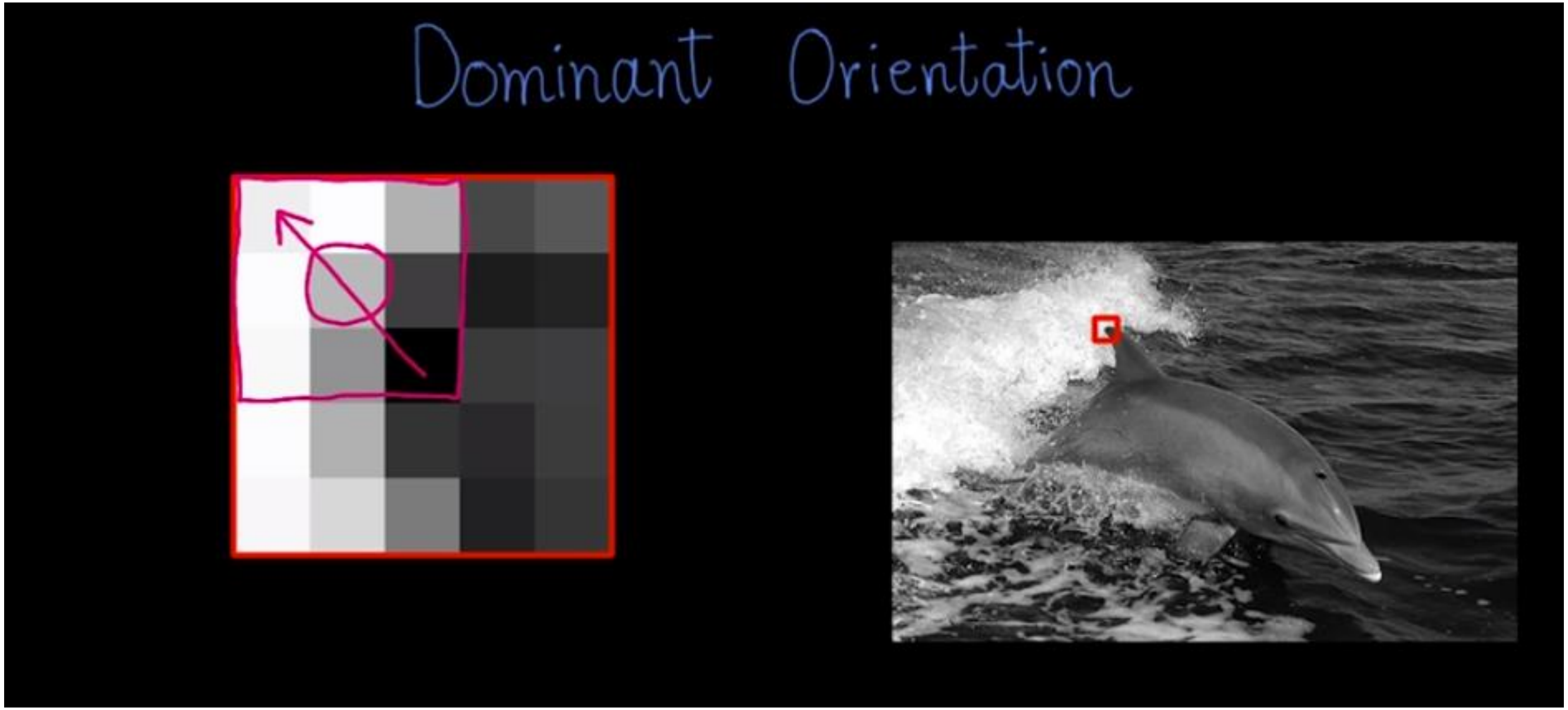
Dominant Orientation



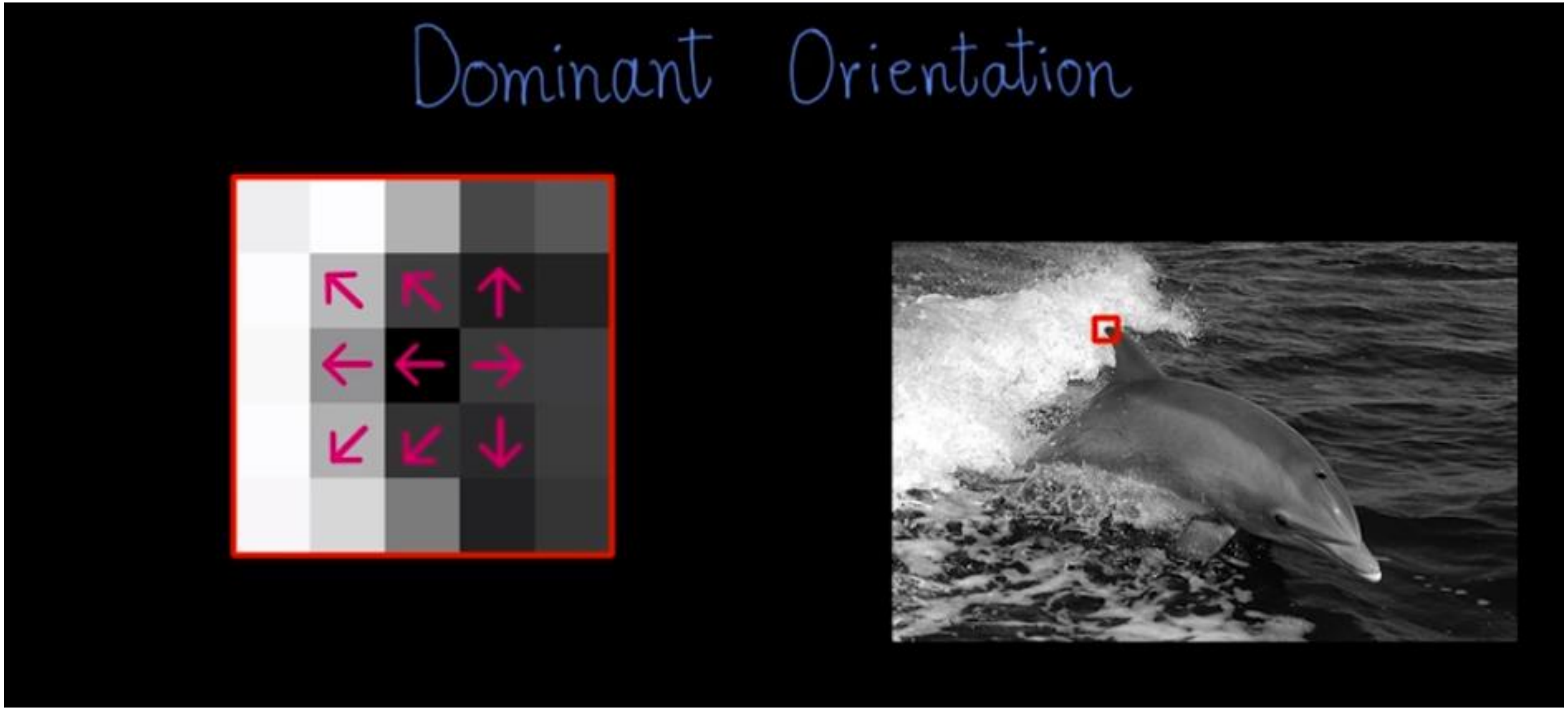
Dominant Orientation



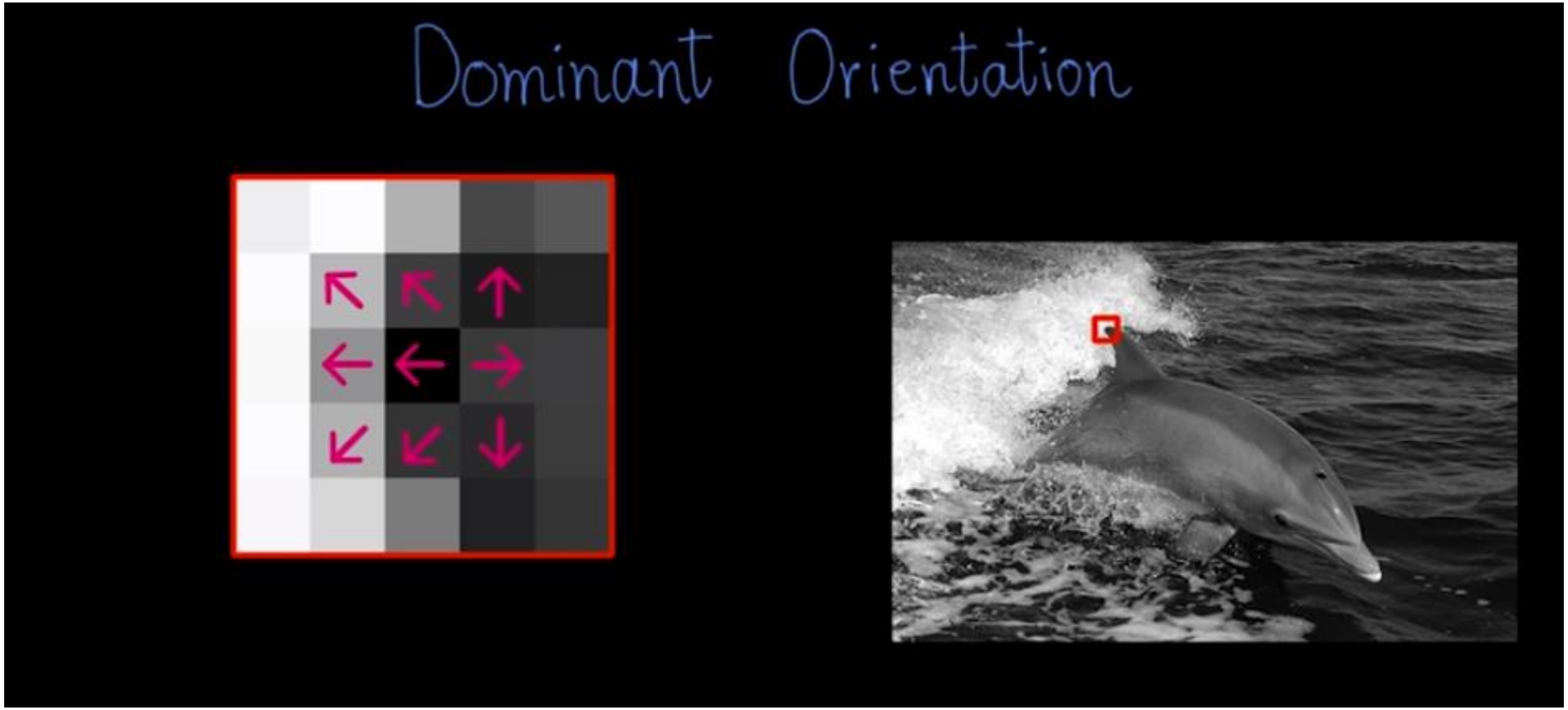
Dominant Orientation

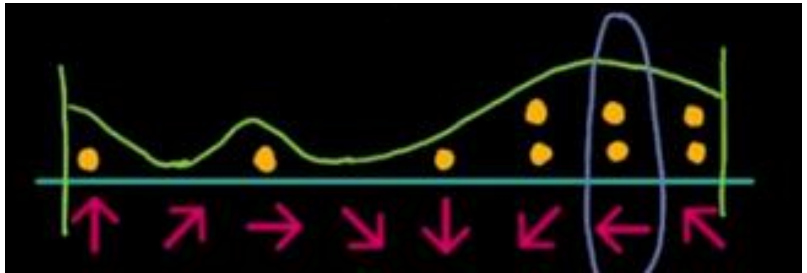


Dominant Orientation

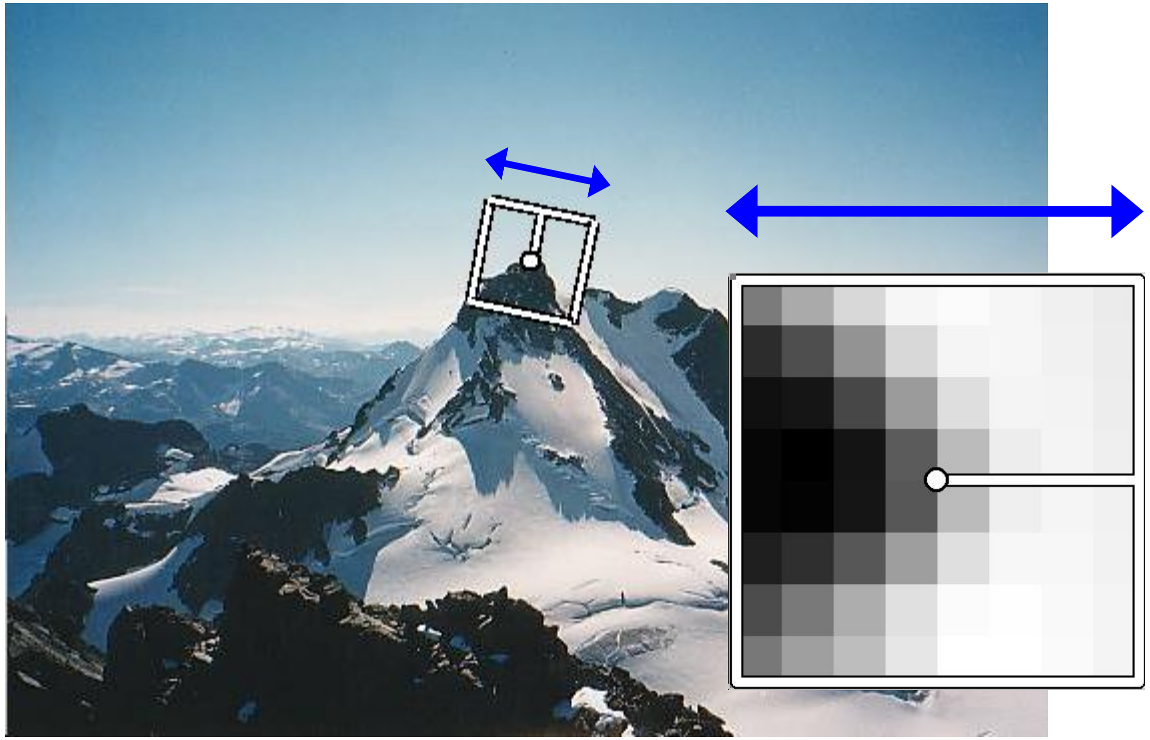


Dominant Orientation





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 descriptor

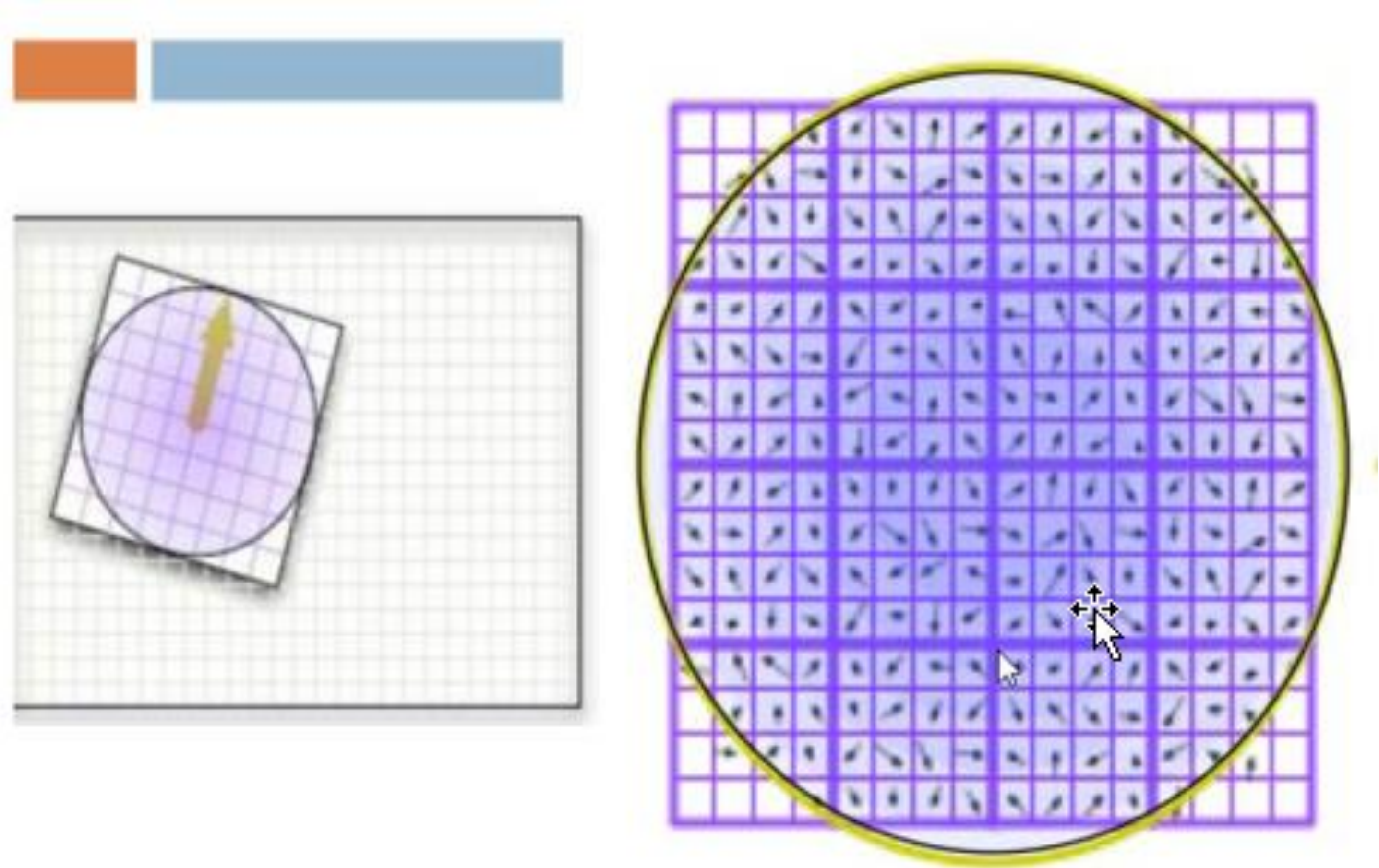
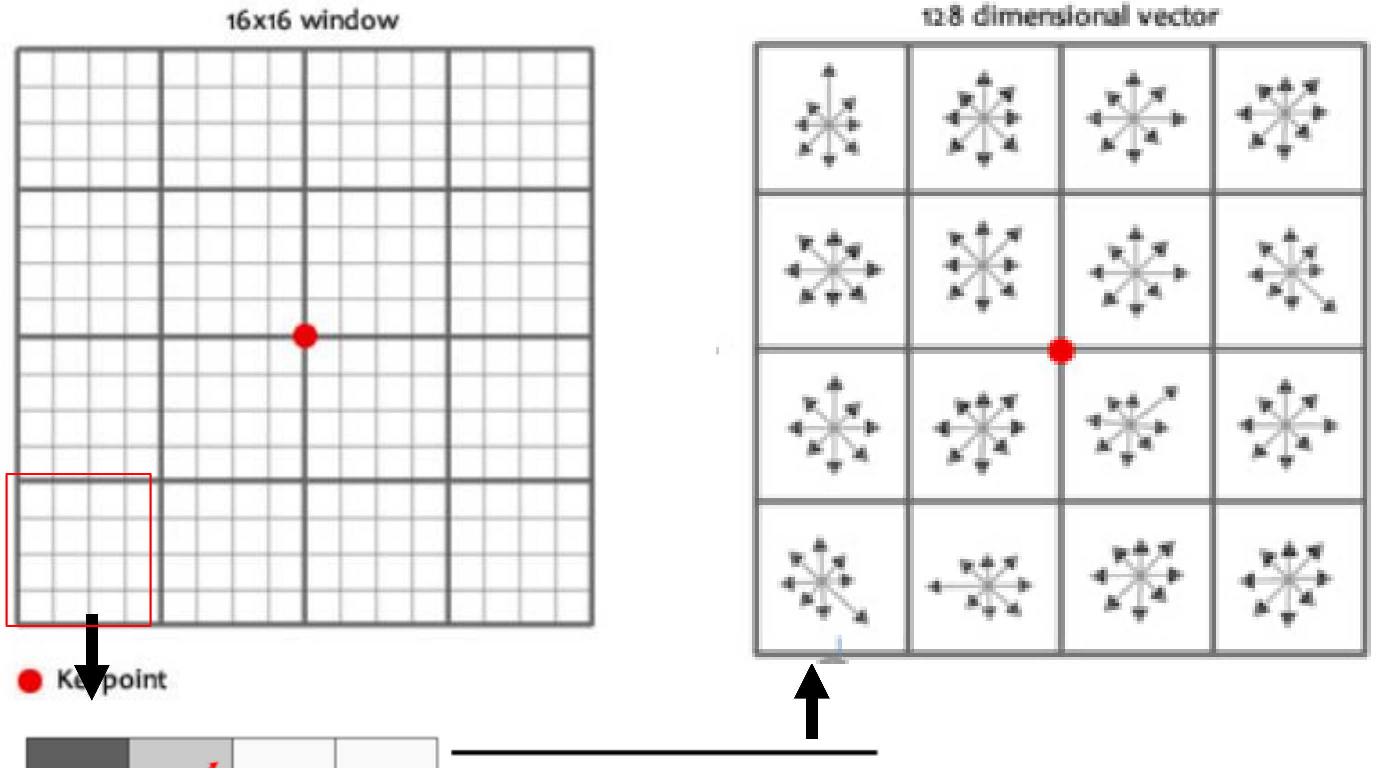
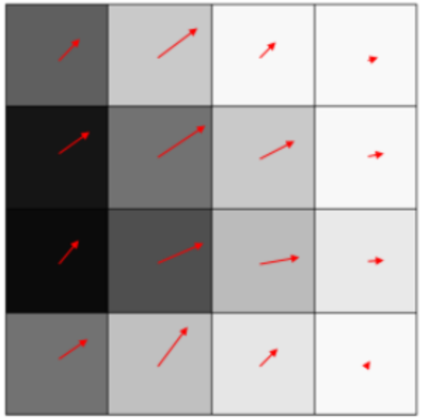


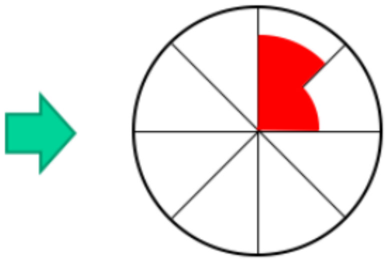
Image gradients

SIFT descriptor



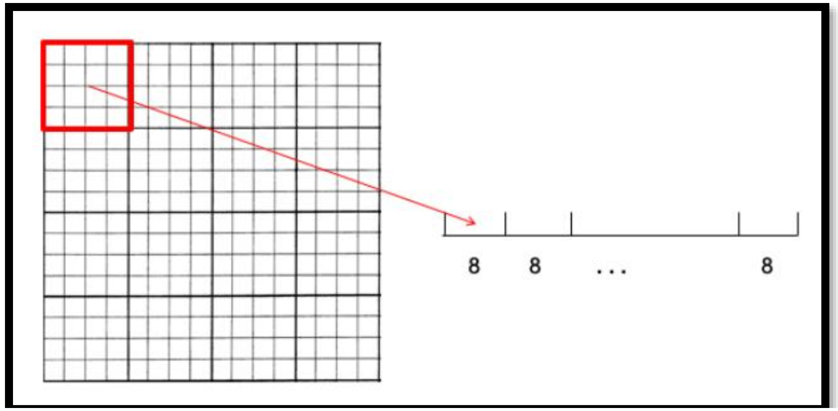


Orientations in each of the 16 pixels of the cell



The orientations all ended up in two bins: 11 in one bin, 5 in the other. (rough count)

5 11 0 0 0 0 0 0



- Actual implementation uses 4x4 descriptors from 16x16 which leads to a 4x4x8=128 element vector