TP12 - Local features: detection and description

Computer Vision, FCUP, 2019/20
Miguel Coimbra
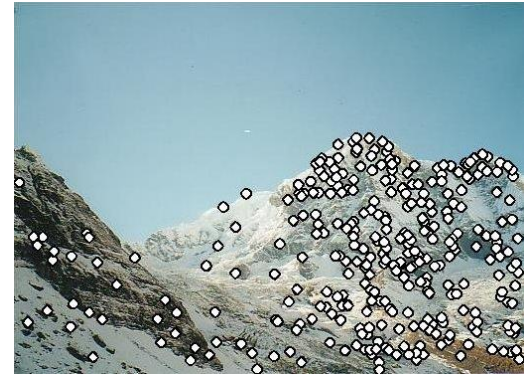Slides by Prof. Kristen Grauman

Today

- Local invariant features
  - Detection of interest points
    - (Harris corner detection)
    - Scale invariant blob detection: LoG
  - Description of local patches
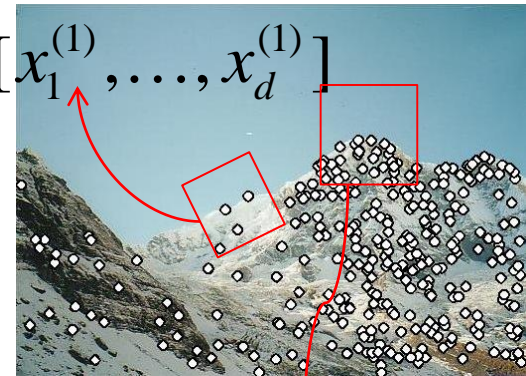   - SIFT : Histograms of oriented gradients

Local features: main components

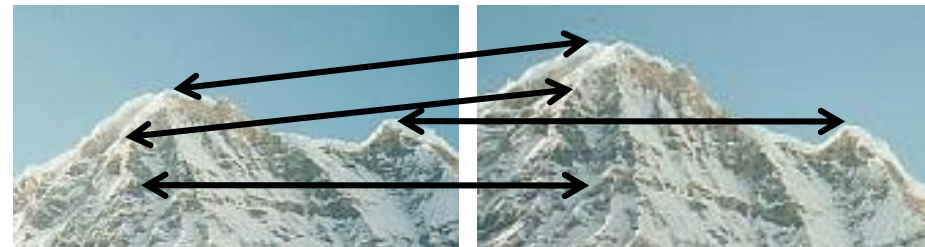1) **Detection:** Identify the interest points

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

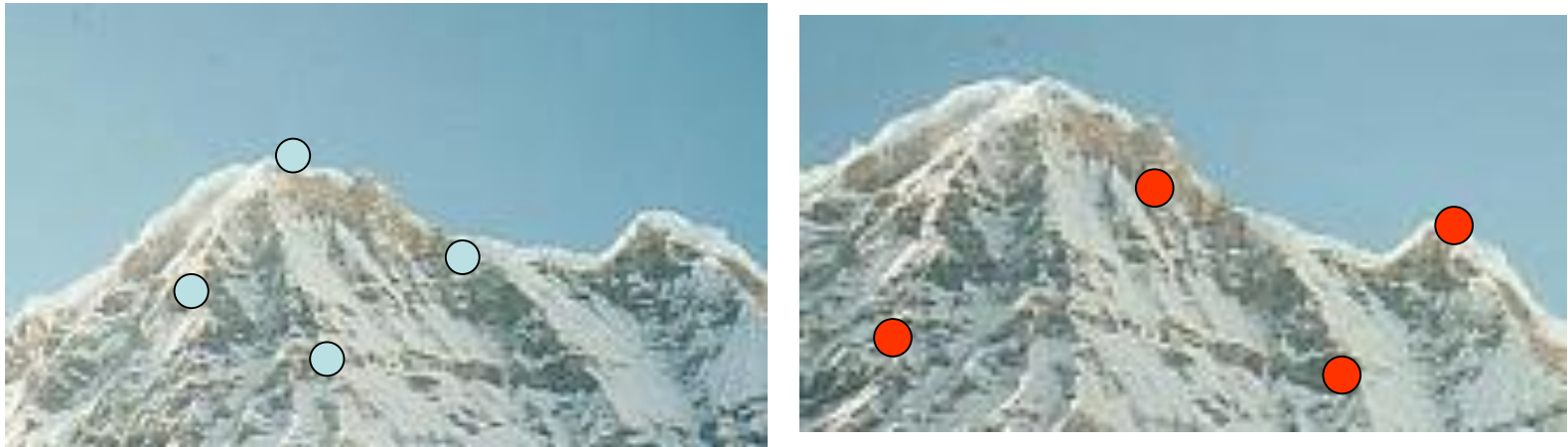$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

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

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

Kristen Grauman

Goal: interest operator repeatability

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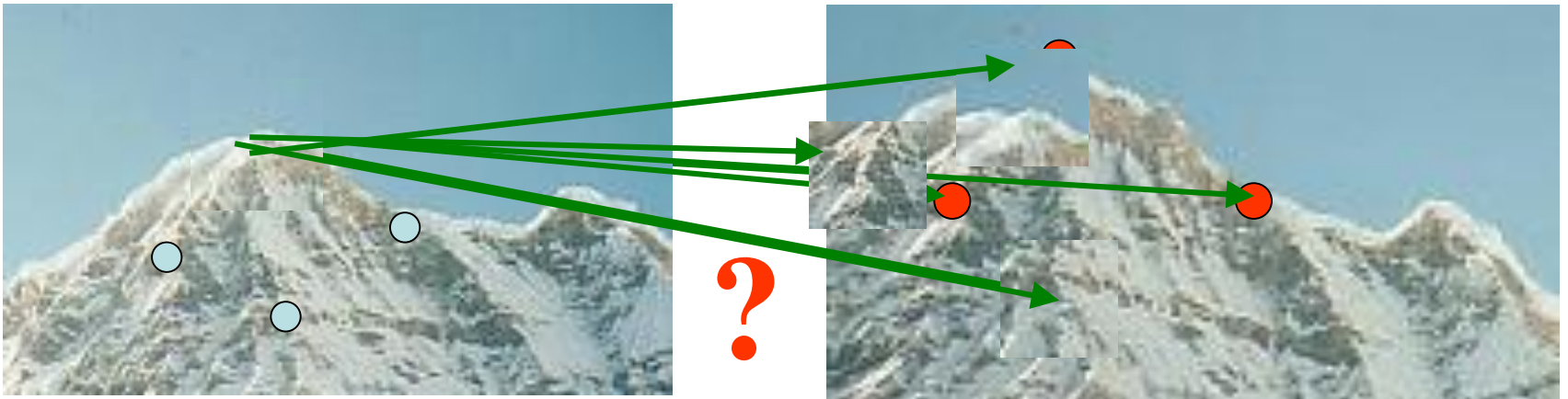

No chance to find true matches!

- 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.
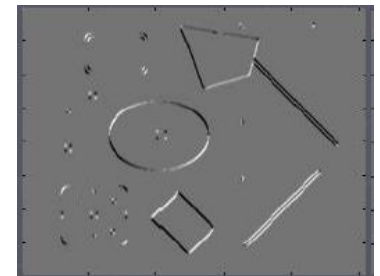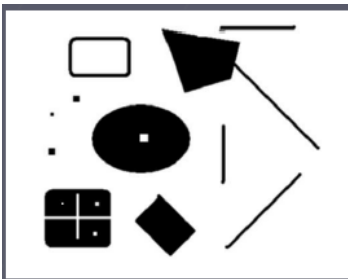
Local features: main components

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
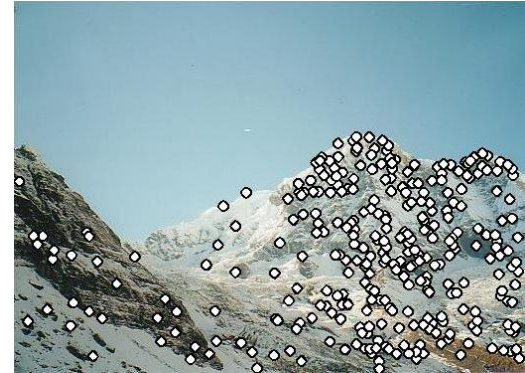
Recall: Corners as distinctive interest points

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

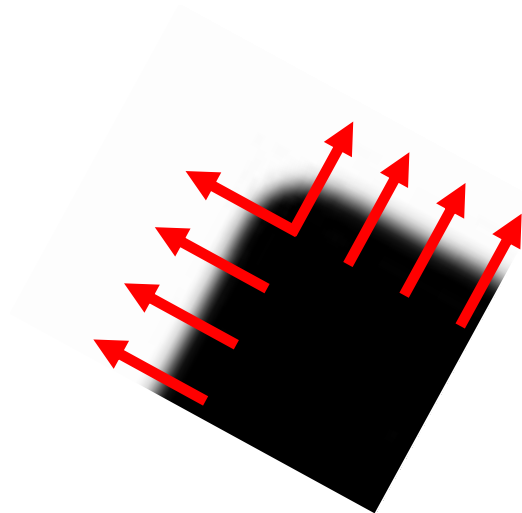2 x 2 matrix of image derivatives (averaged in neighborhood of a point).

Notation:

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

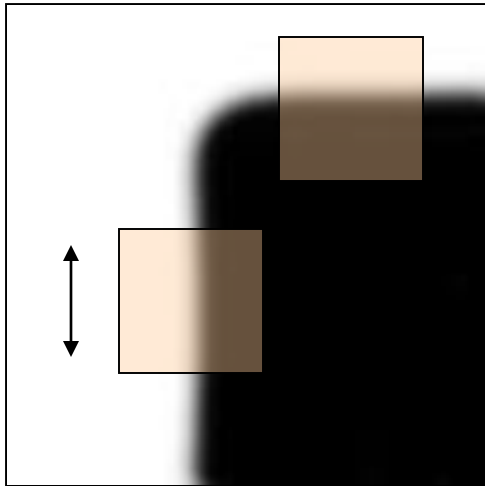# **Recall:** Corners as distinctive interest points

Since *M* is symmetric, we have $M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$

$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of *M* reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.
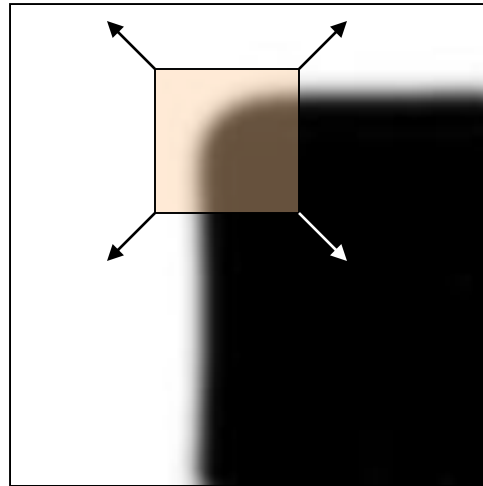
Recall: Corners as distinctive interest points
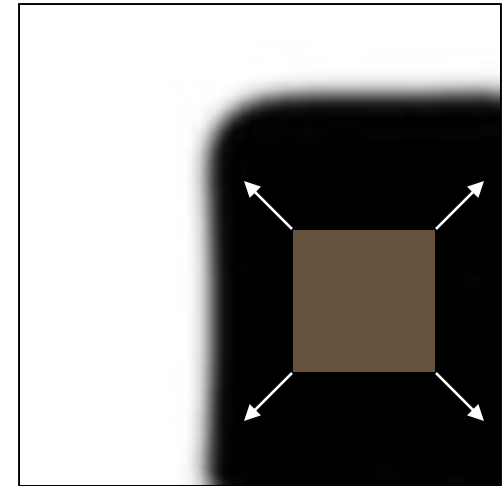

"edge":
$\lambda_1 >> \lambda_2$
$\lambda_2 >> \lambda_1$

"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:

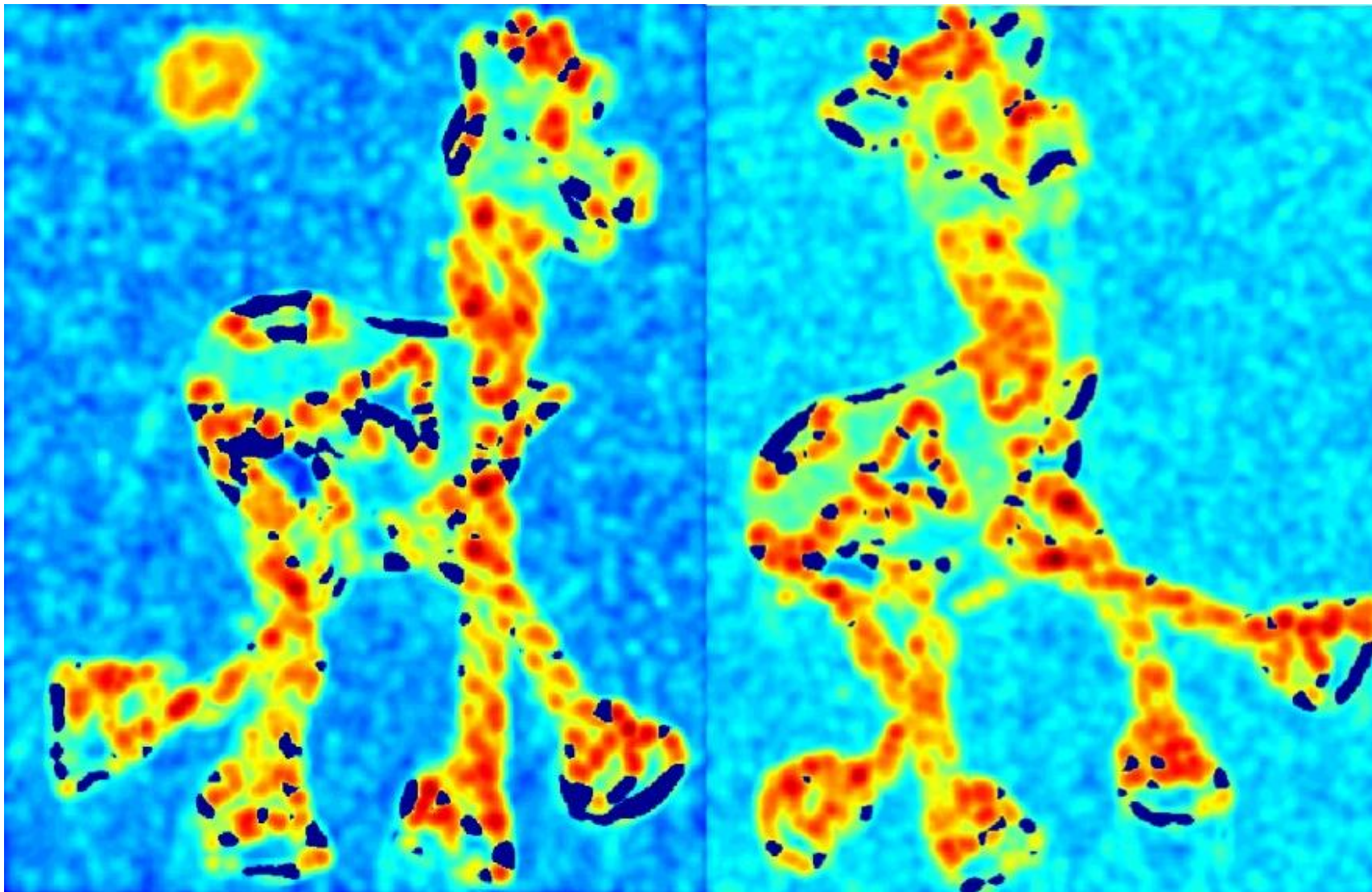$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

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
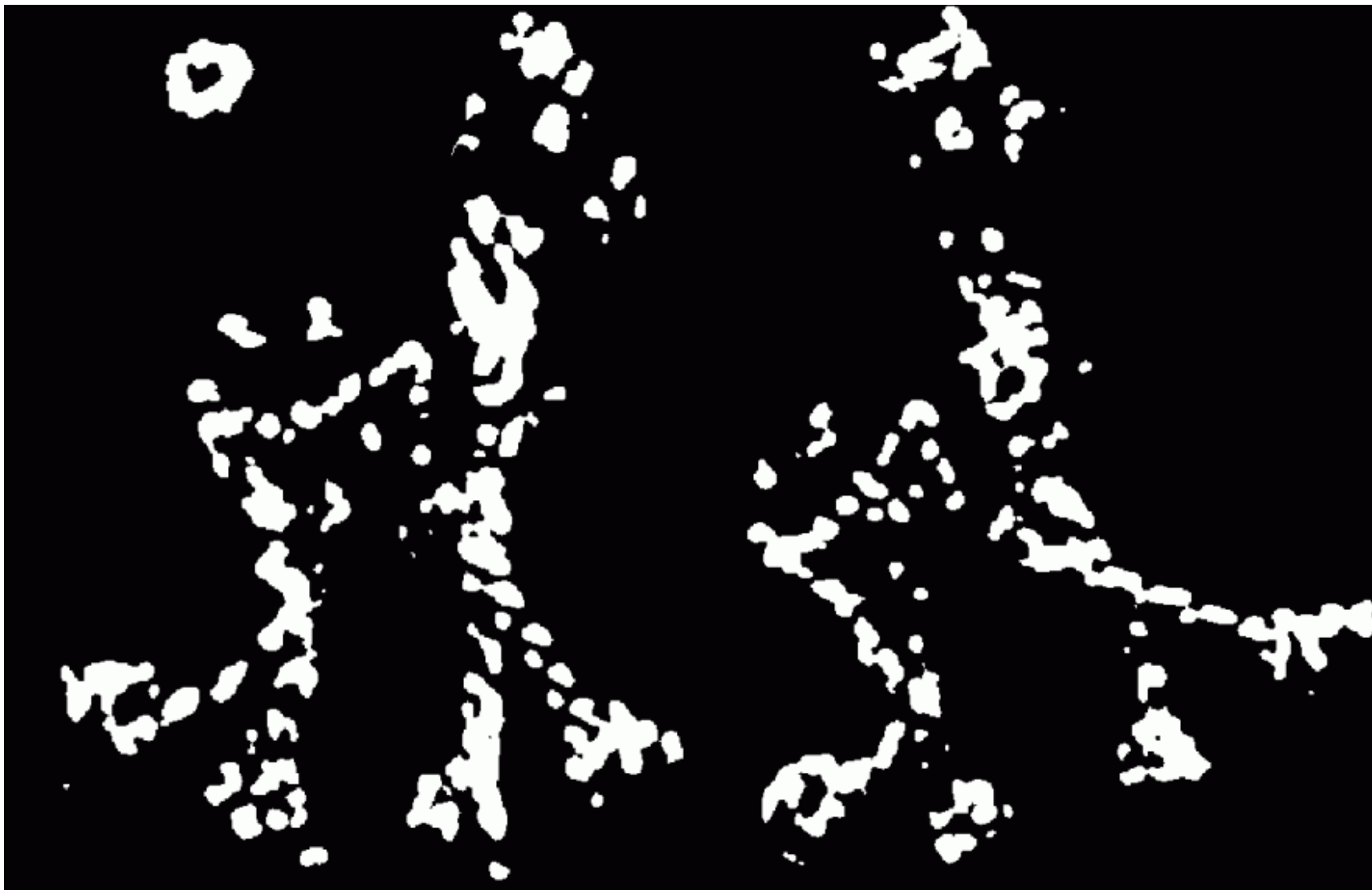
Harris Detector: Steps

Harris Detector: Steps

Compute corner response *f*

Harris Detector: Steps
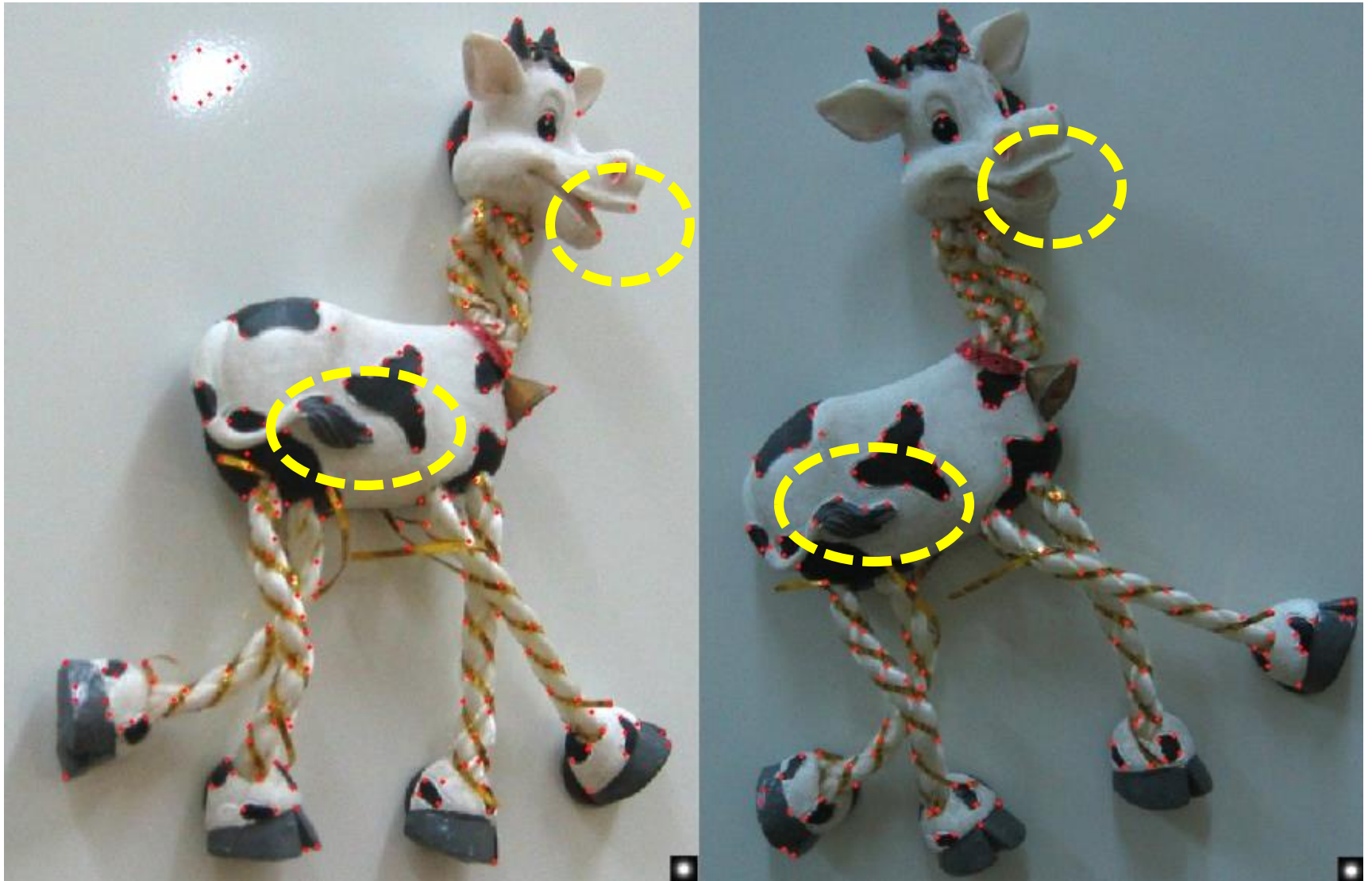
Find points with large corner response: f > threshold

Harris Detector: Steps

Take only the points of local maxima of f

Harris Detector: Steps

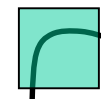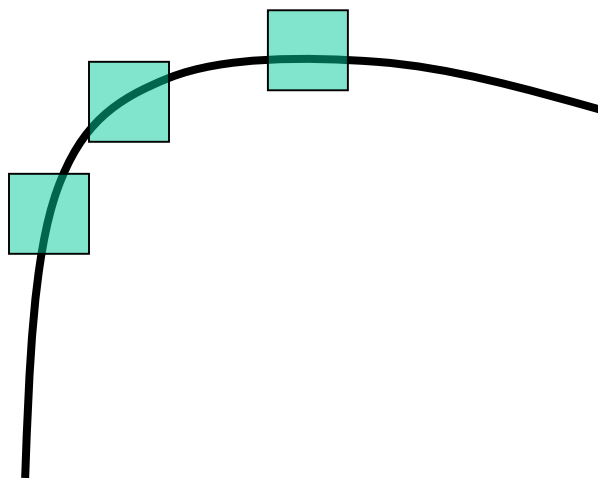# Properties of the Harris corner detector

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

Scale invariant?   No

All points will be
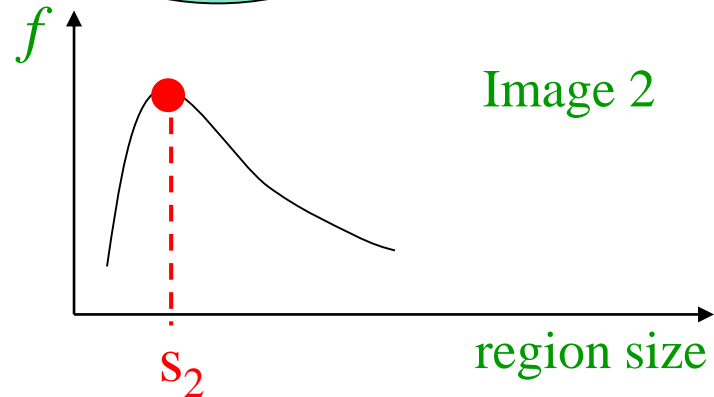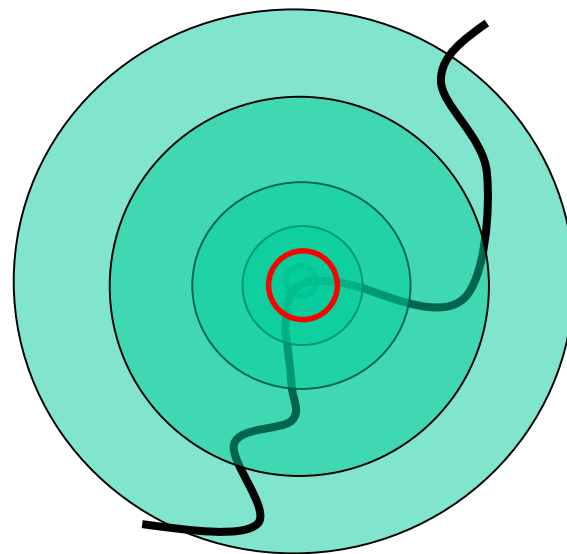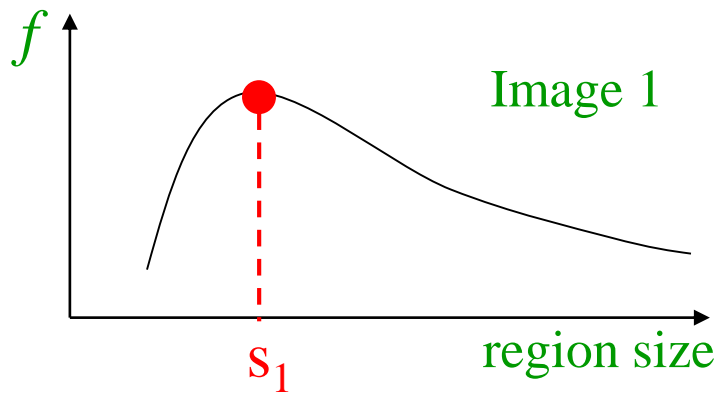classified as edges

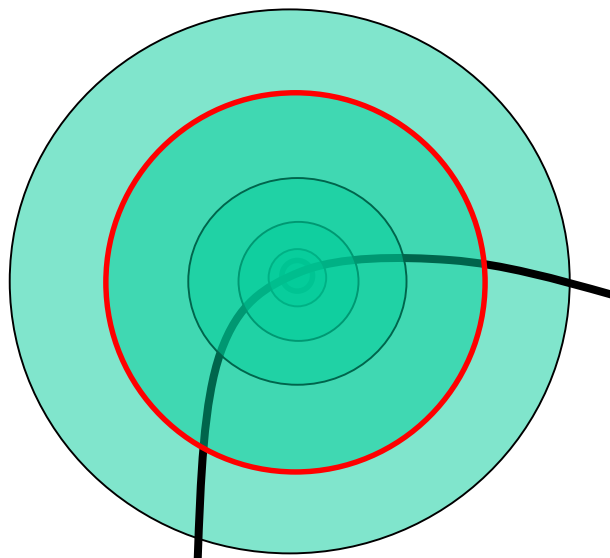Corner !

Scale invariant interest points

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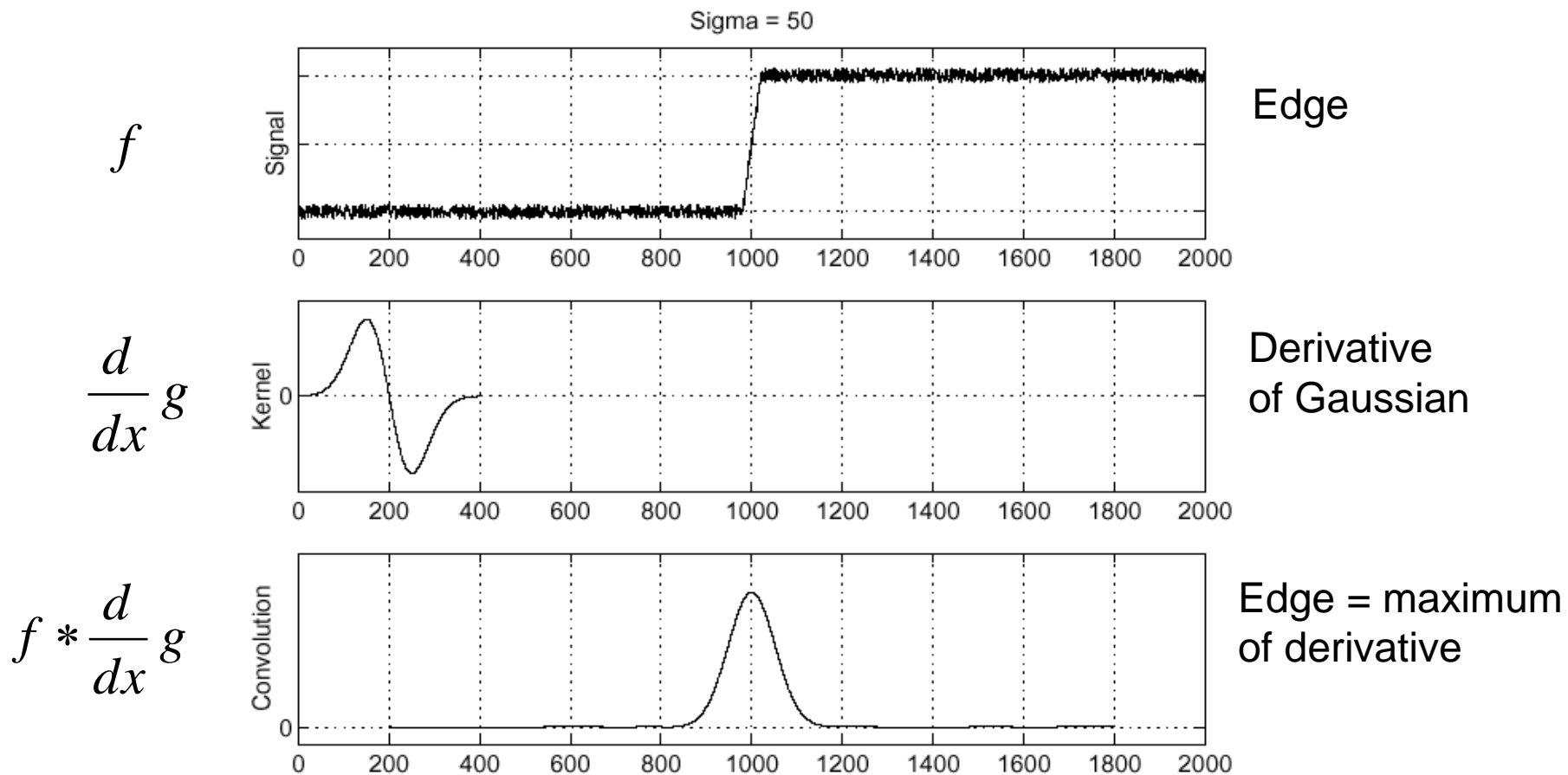
Automatic scale selection

Intuition:

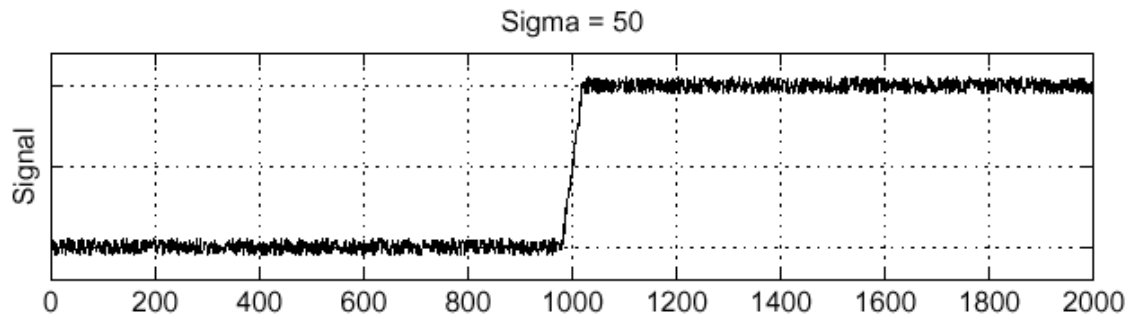- Find scale that gives local maxima of some function f in both position and scale.

Image 1

Image 2

What can be the "signature" function?

Recall: Edge detection

f

$\dfrac{d}{dx}g$

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

Sigma = 50

Edge

Derivative
of Gaussian

Edge = maximum
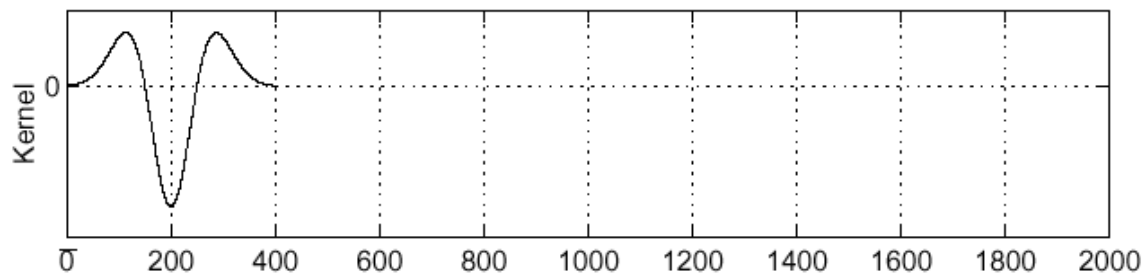of derivative

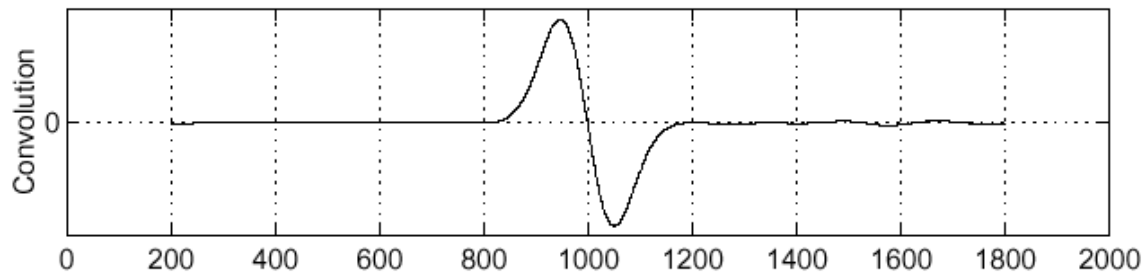Source: S. Seitz

Recall: Edge detection

f

Edge

$\dfrac{d^2}{dx^2}\,g$
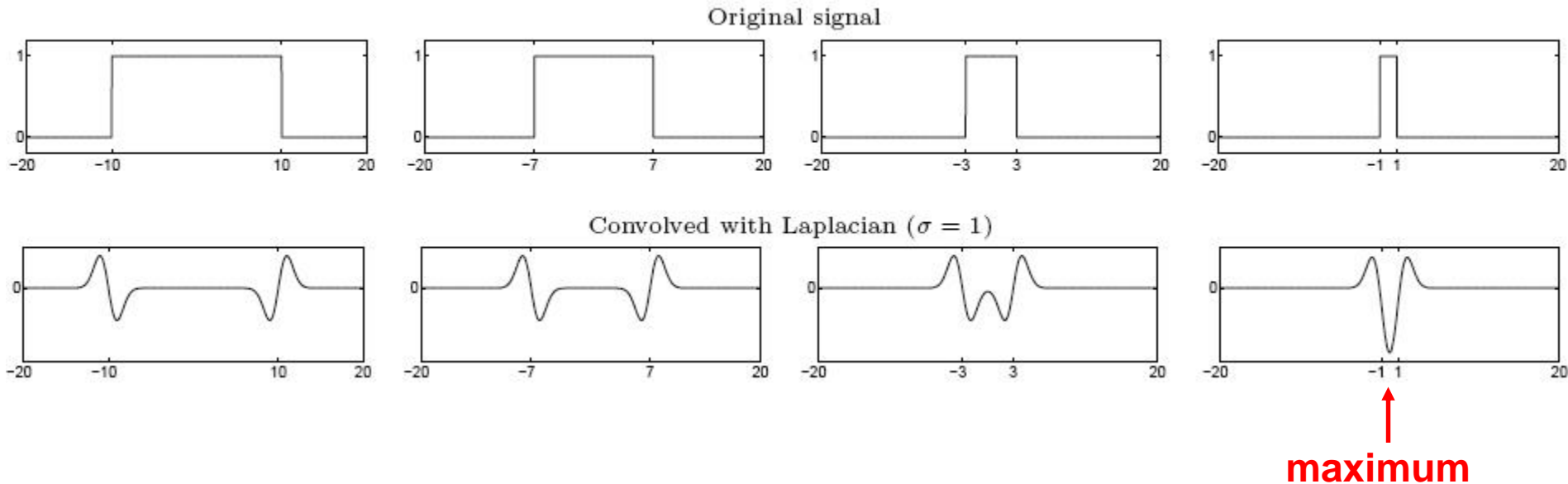
Second derivative
of Gaussian
(Laplacian)

$f * \dfrac{d^2}{dx^2}\,g$

Edge = zero crossing
of second derivative

From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

Original signal

Convolved with Laplacian ($\sigma = 1$)

maximum

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
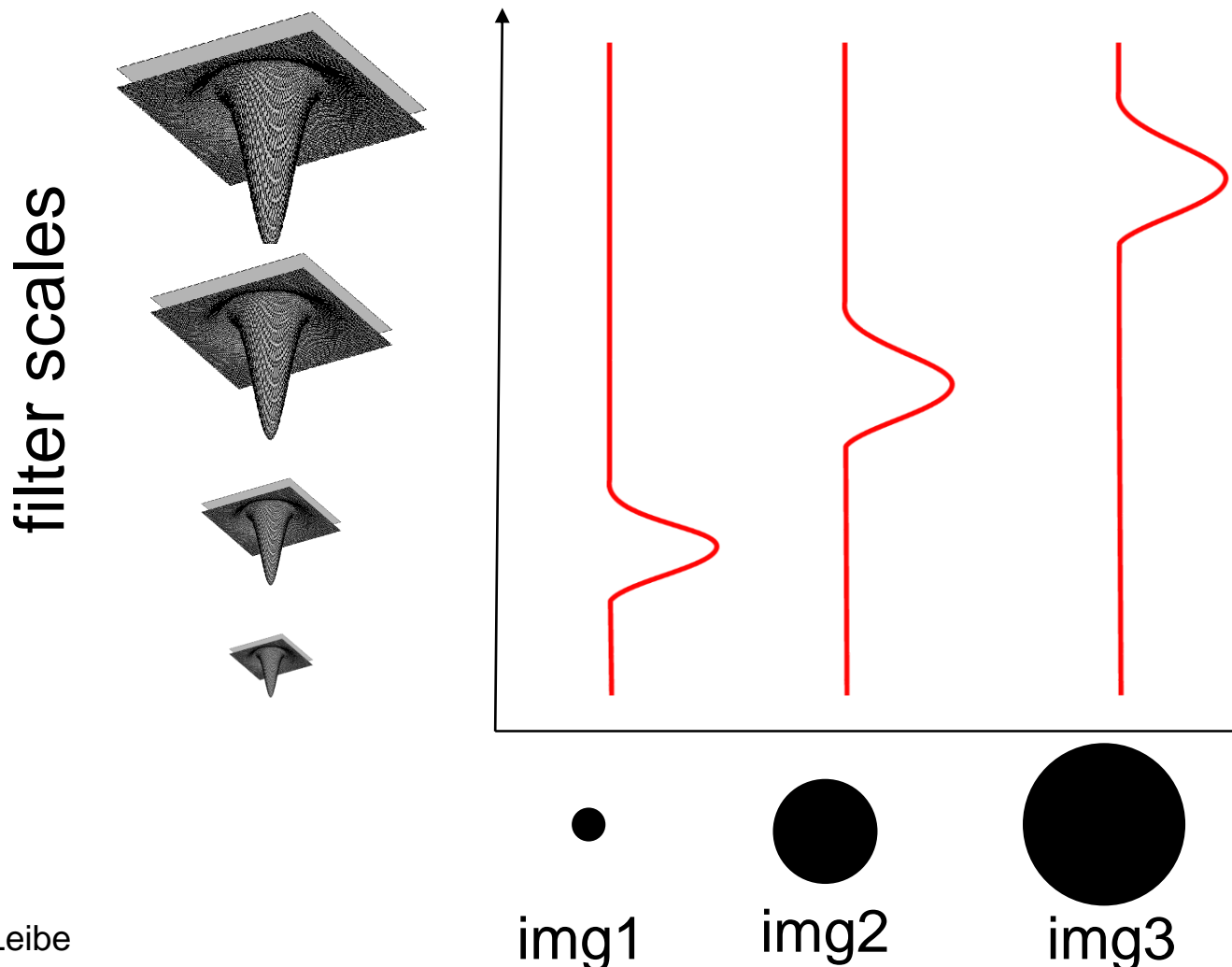
Blob detection in 2D

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D
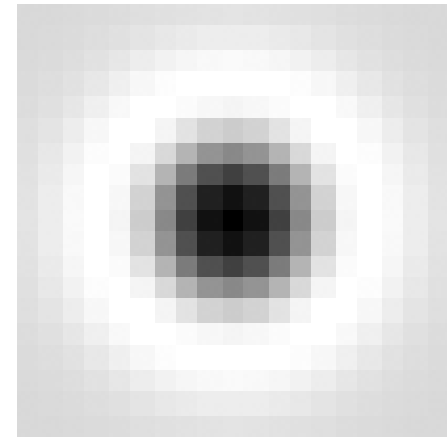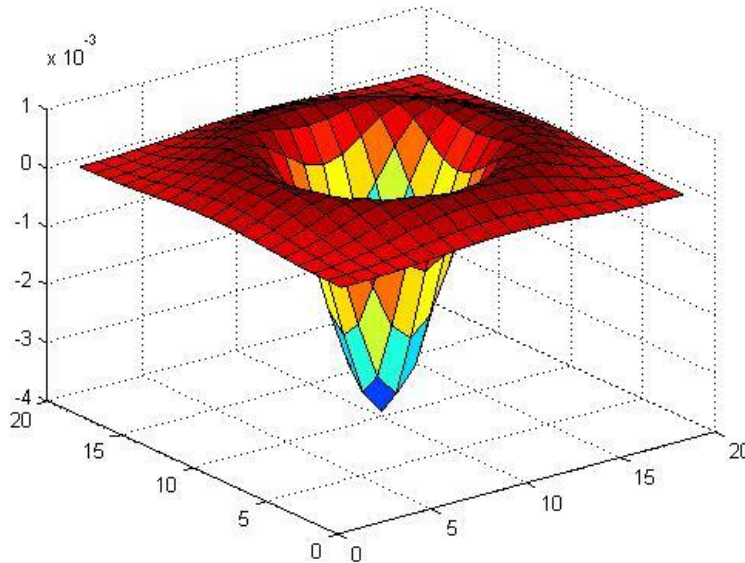
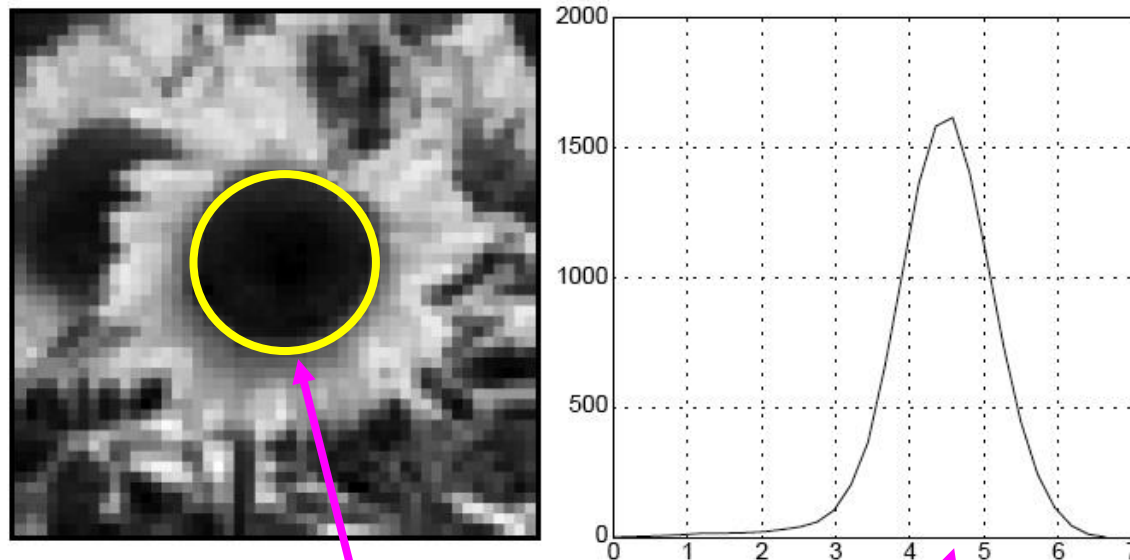$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection in 2D: scale selection

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

filter scales

img1   img2   img3

Blob detection in 2D

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

characteristic scale

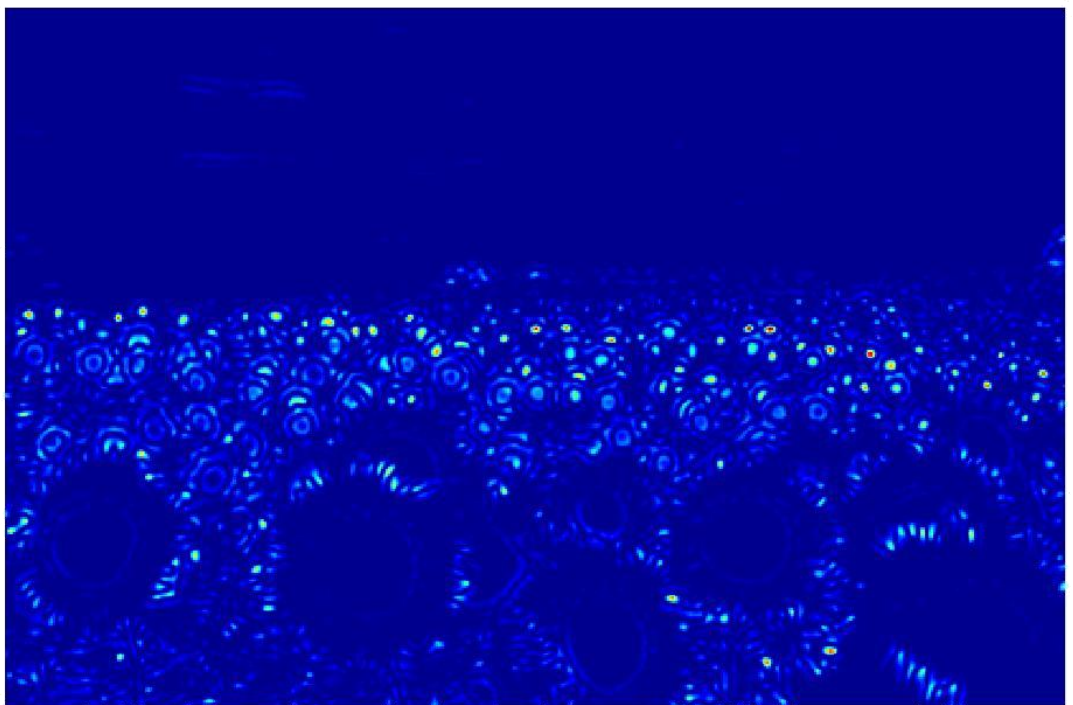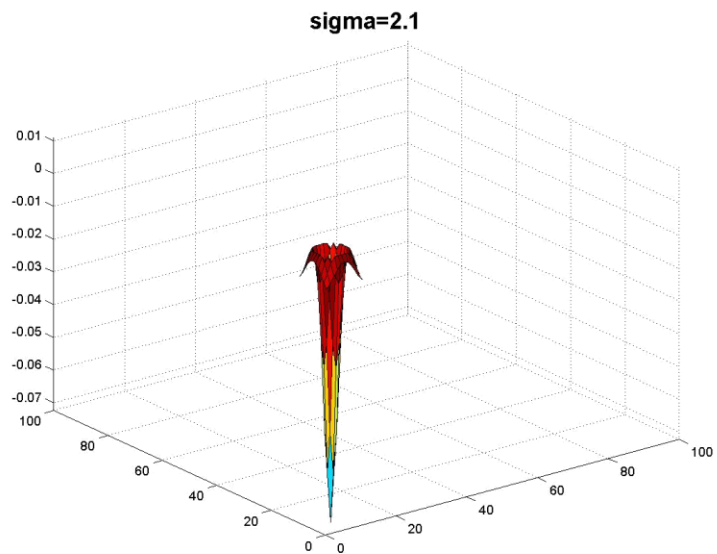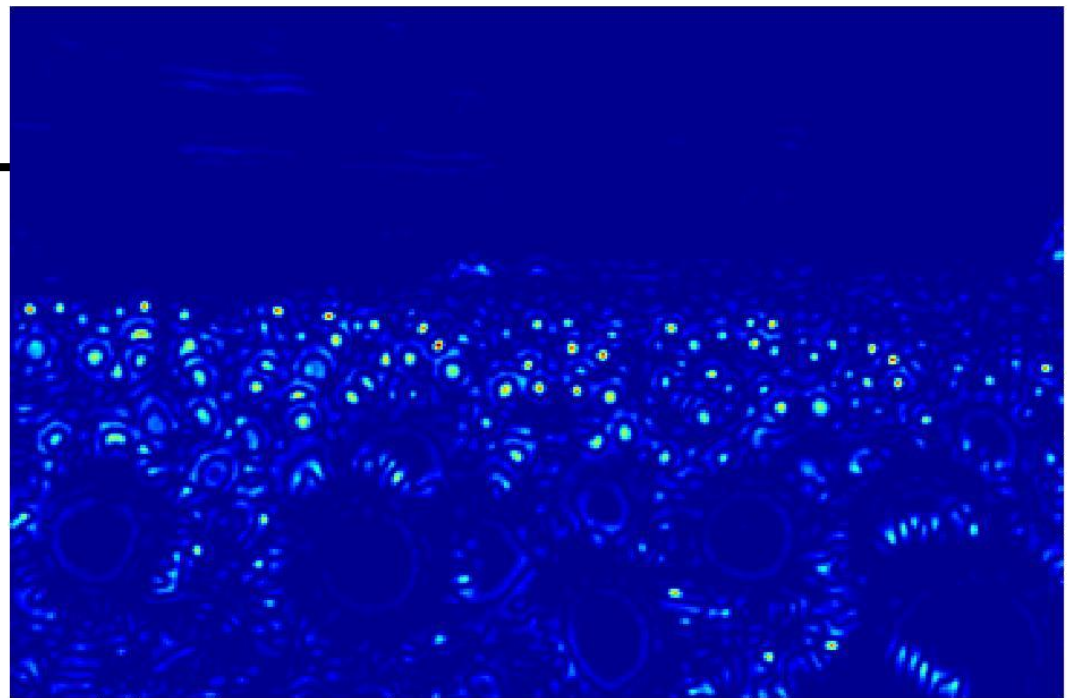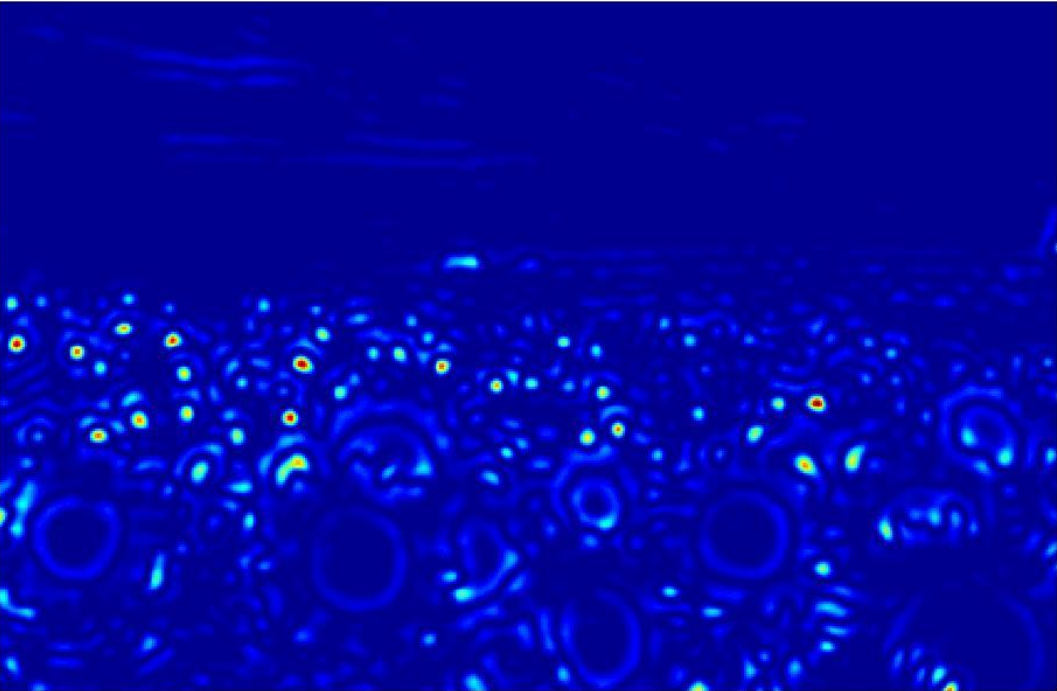# Example

Original image
at ¾ the size

Original image
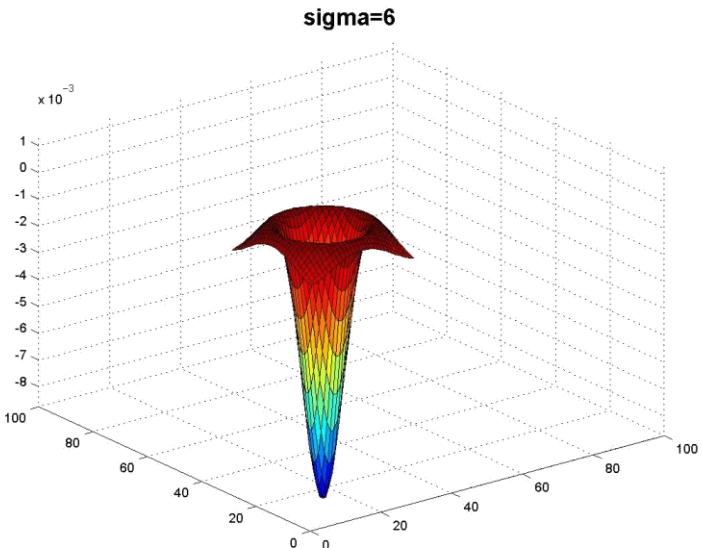at ¾ the size

sigma=2.1

Kristen Grauman

sigma=4.2

Kristen Grauman

sigma=6

Kristen Grauman

sigma=9.8

Kristen Grauman

sigma=15.5

sigma=17

Kristen Grauman
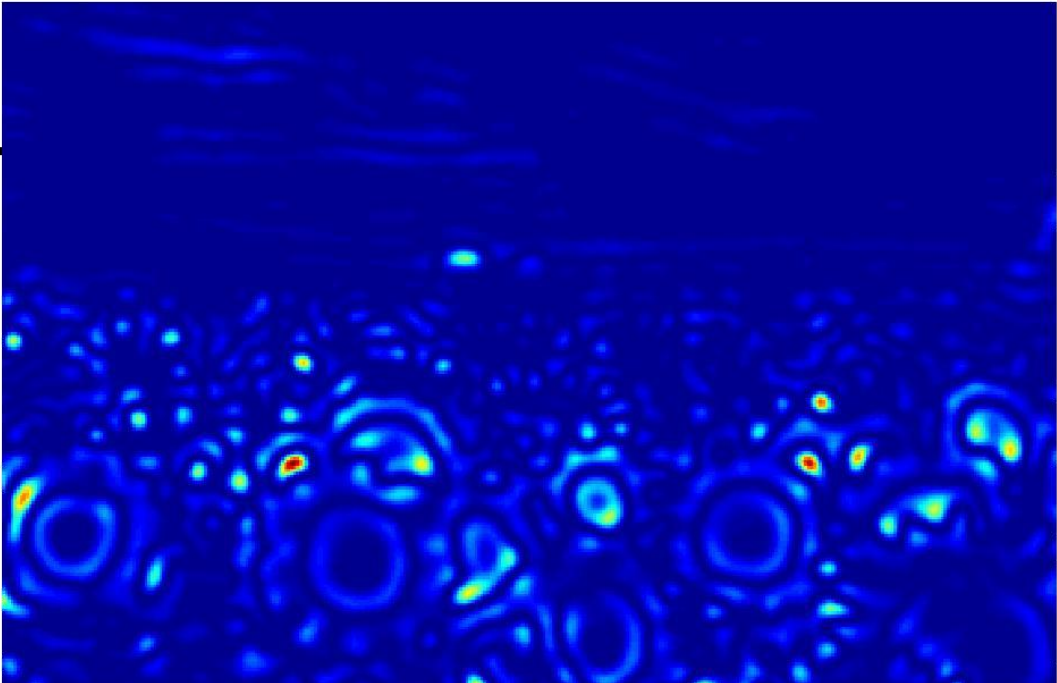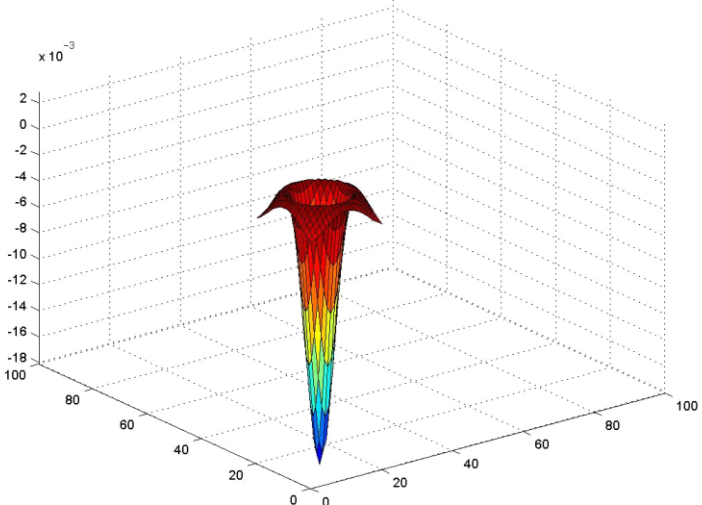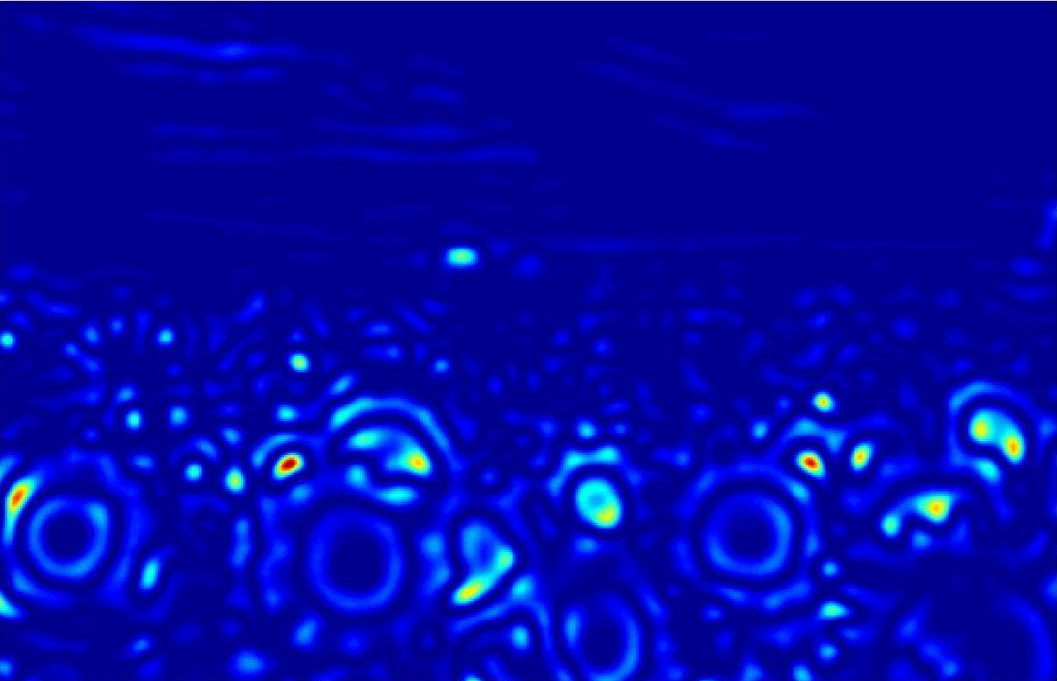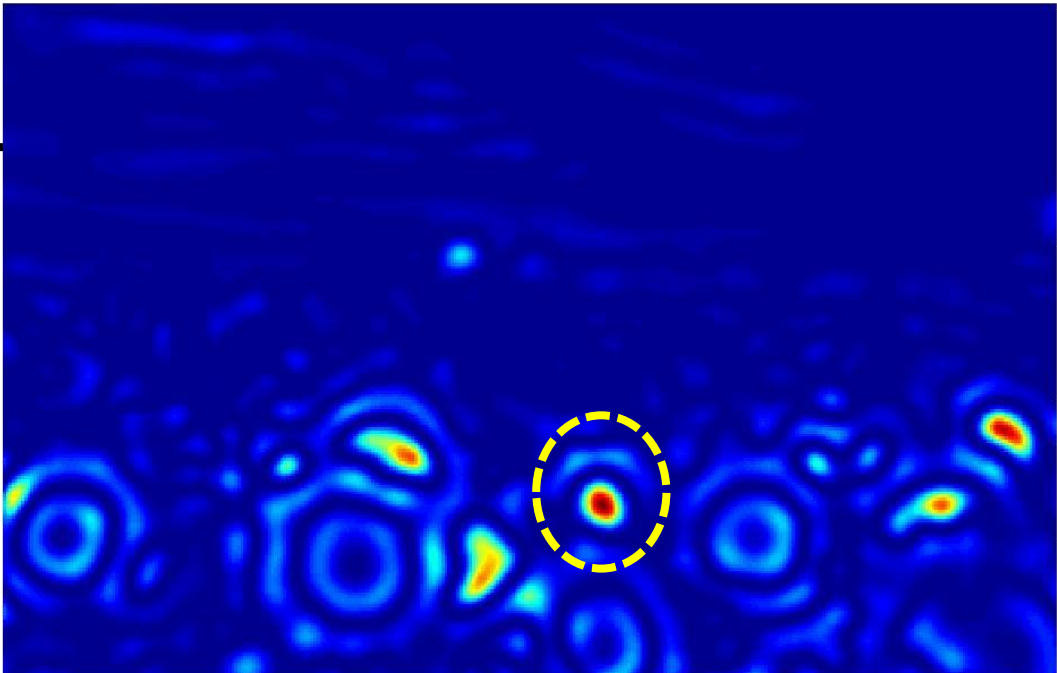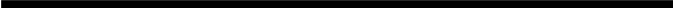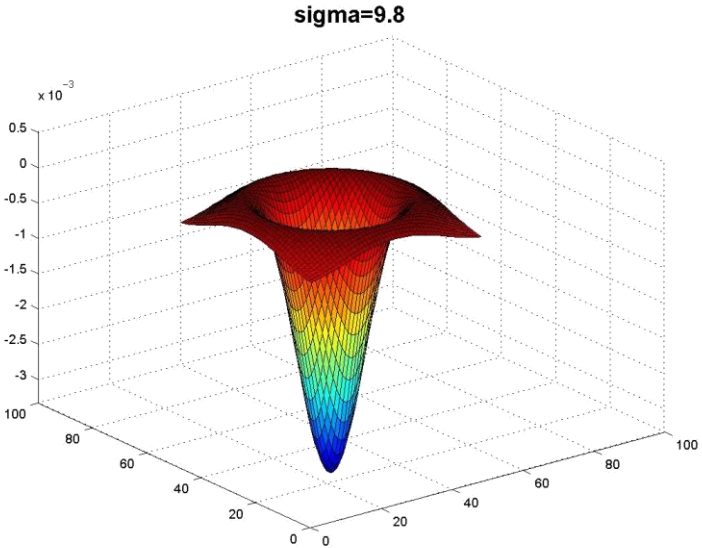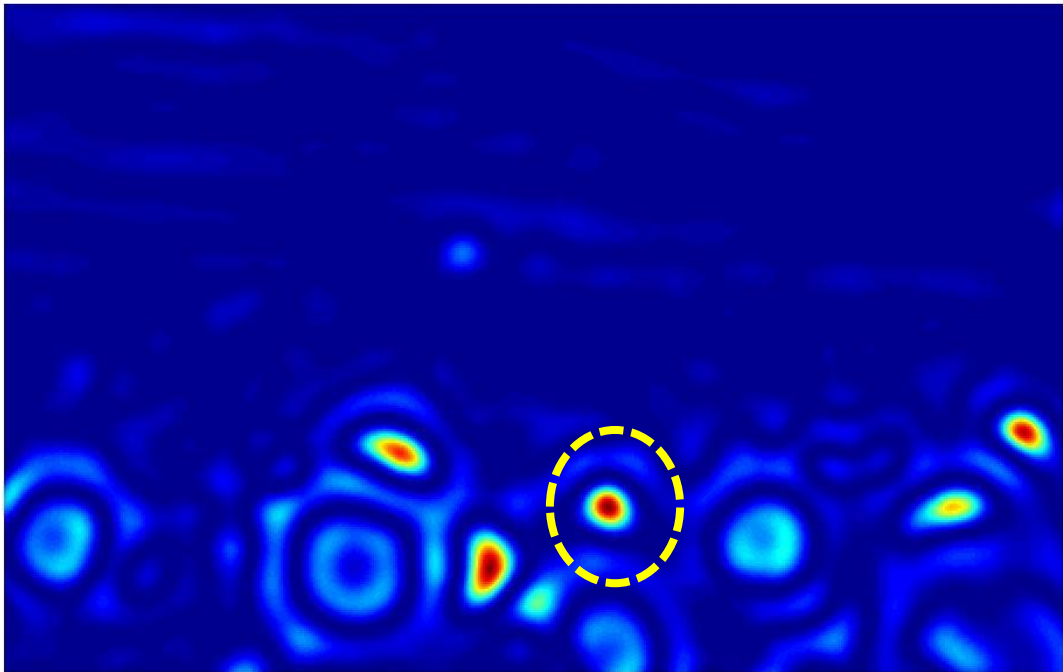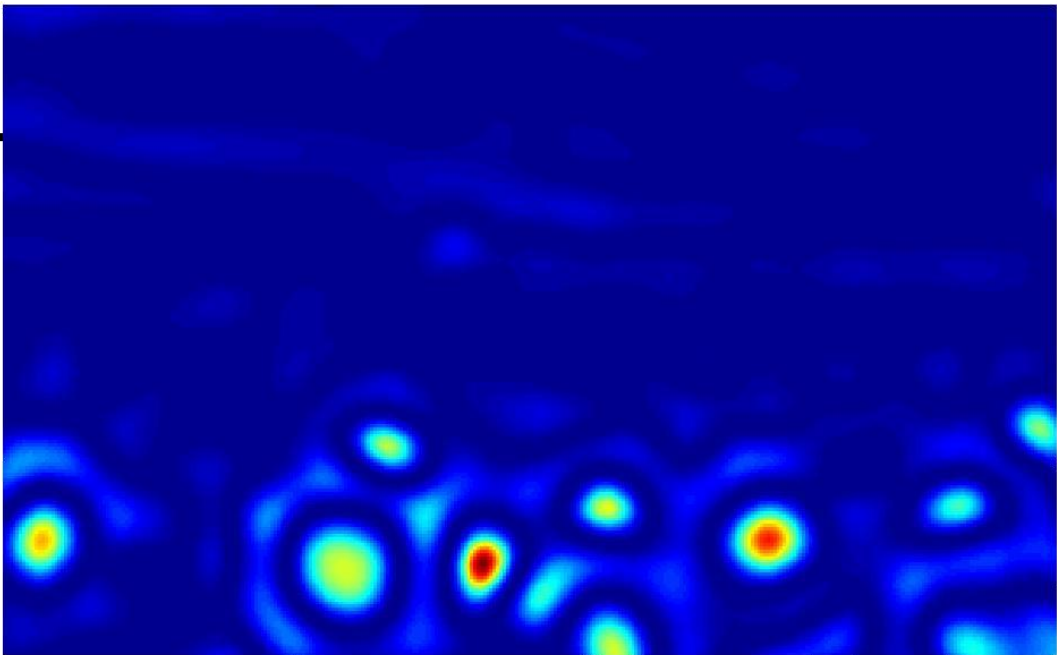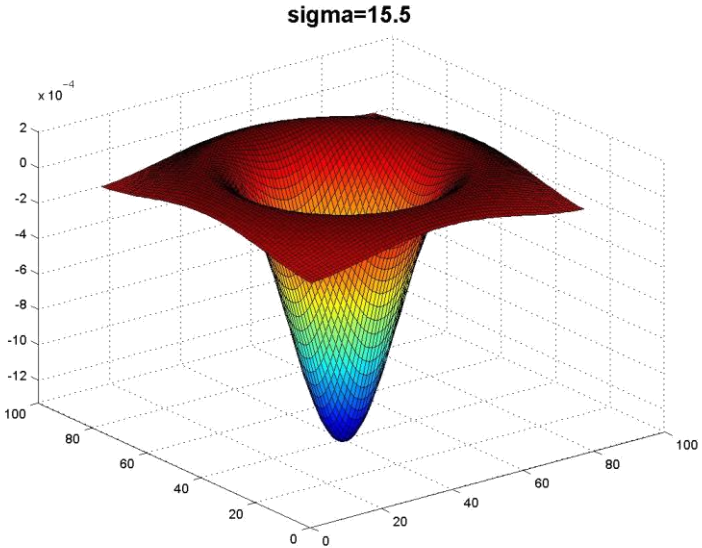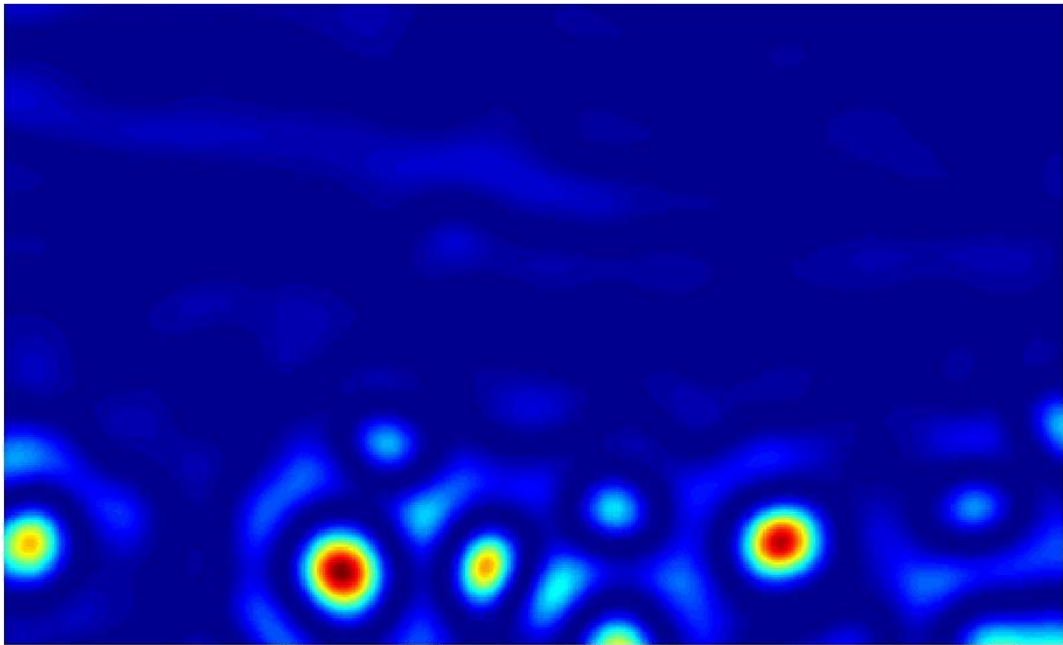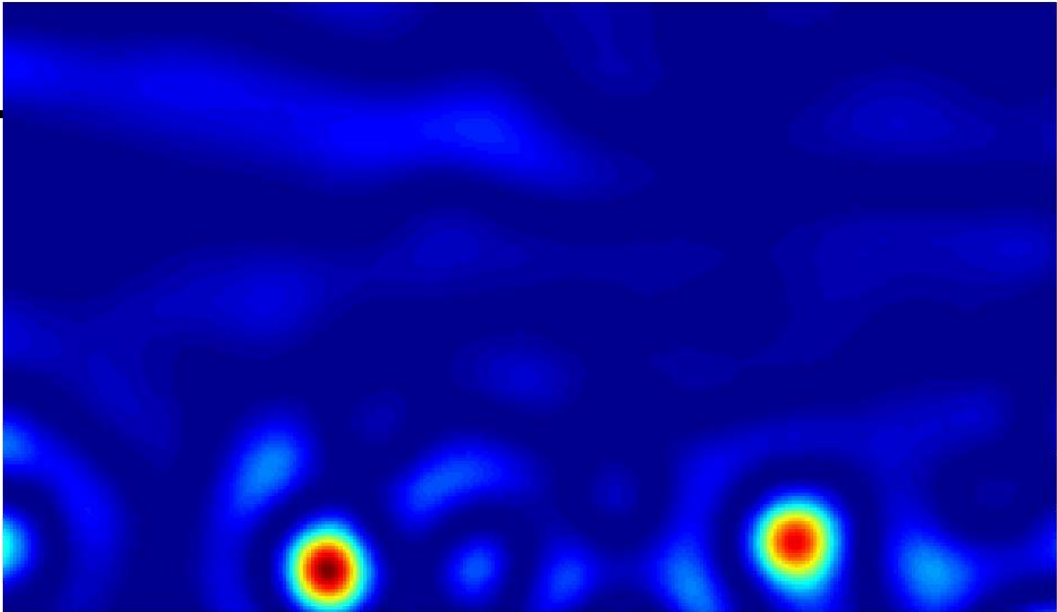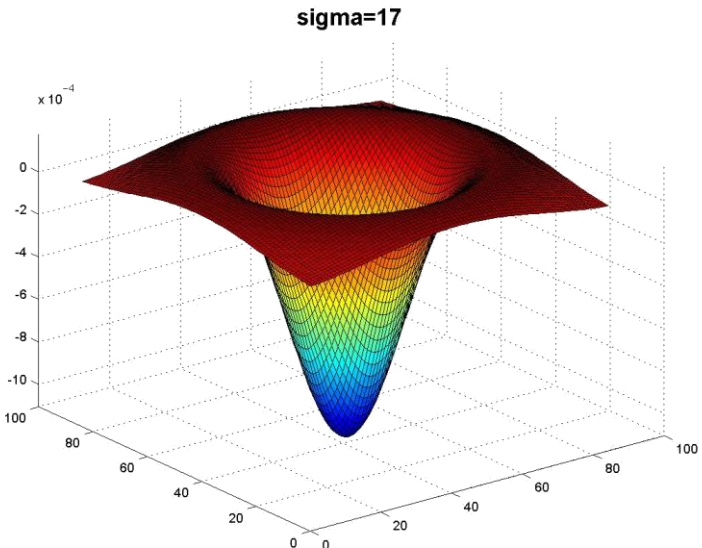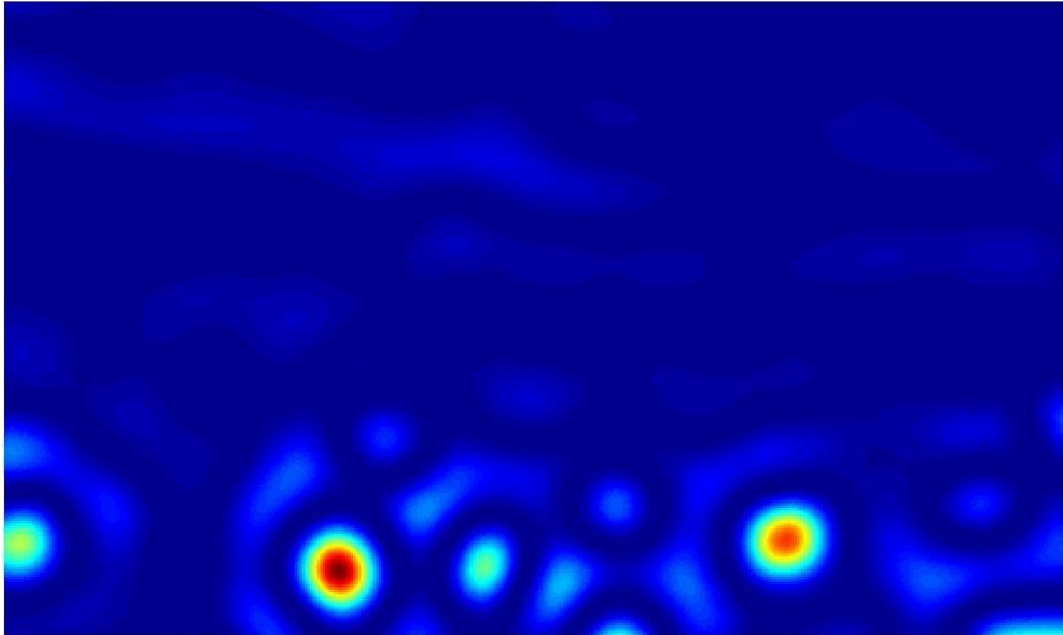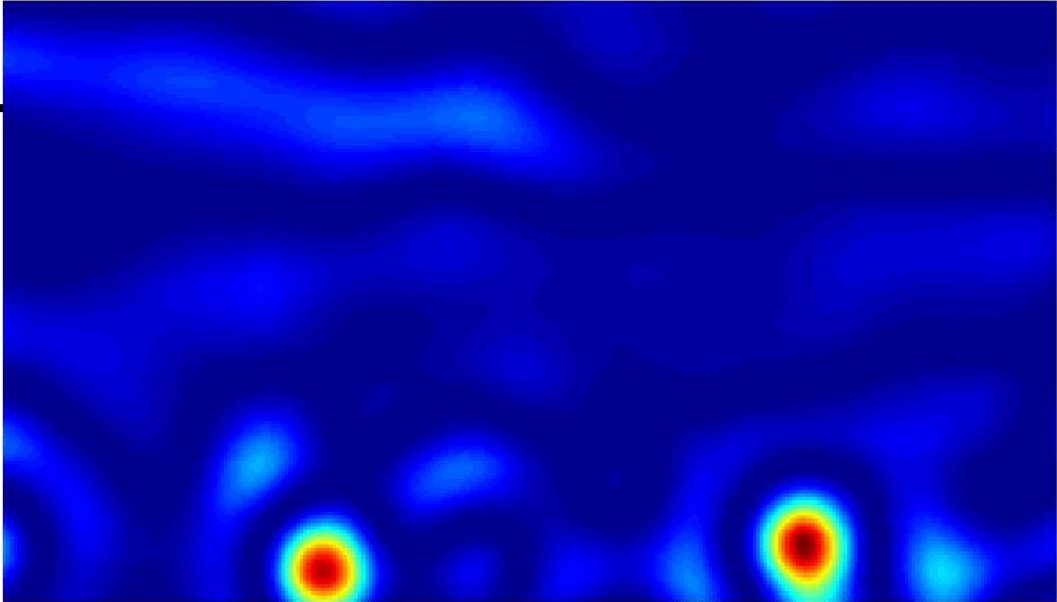
Scale invariant interest points

Interest points are local maxima in both position and scale.

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

σ5

σ4

σ3

σ2

σ1

Squared filter response maps

scale

scale

⇒ **List of (x, y, σ)**
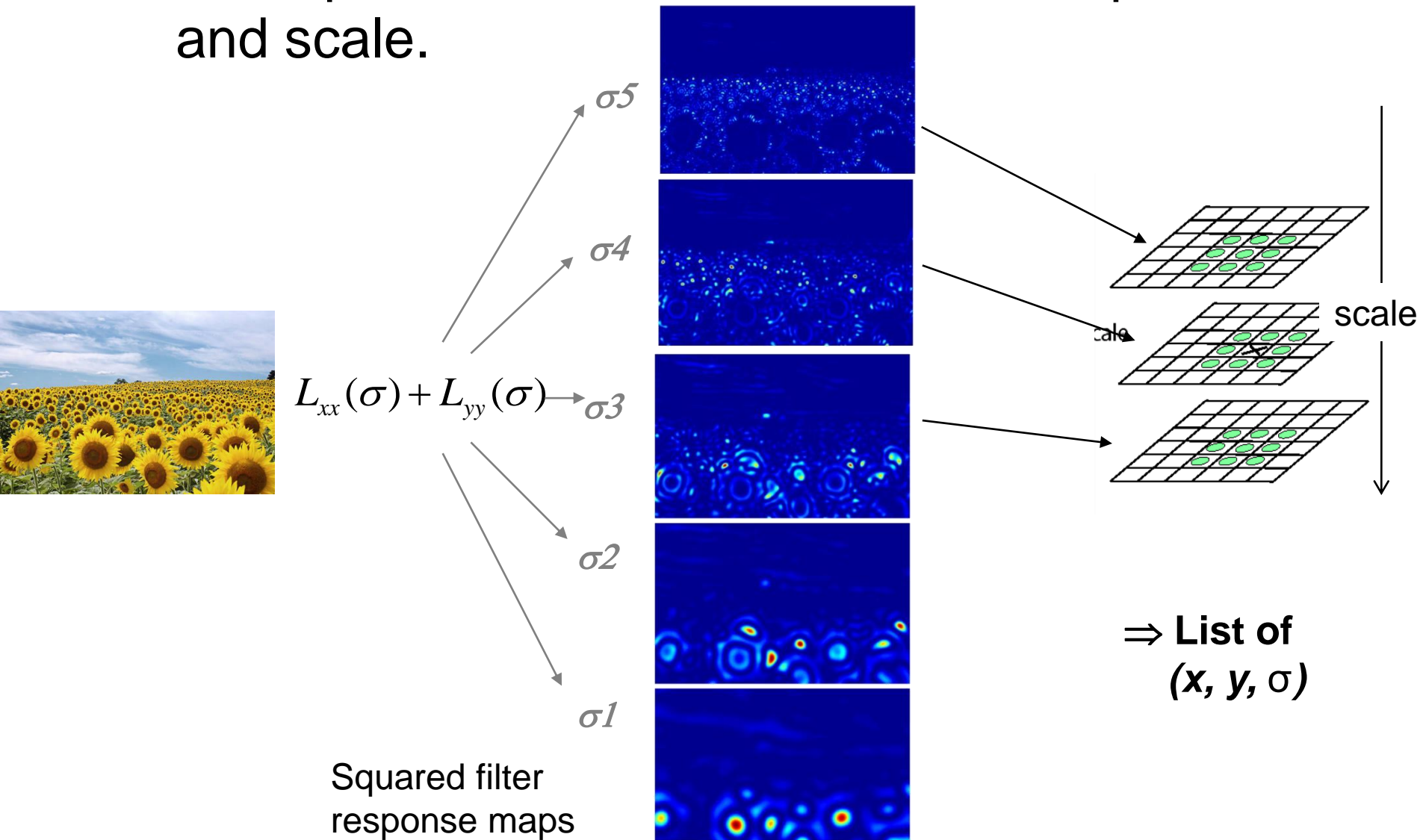
Scale-space blob detector: Example
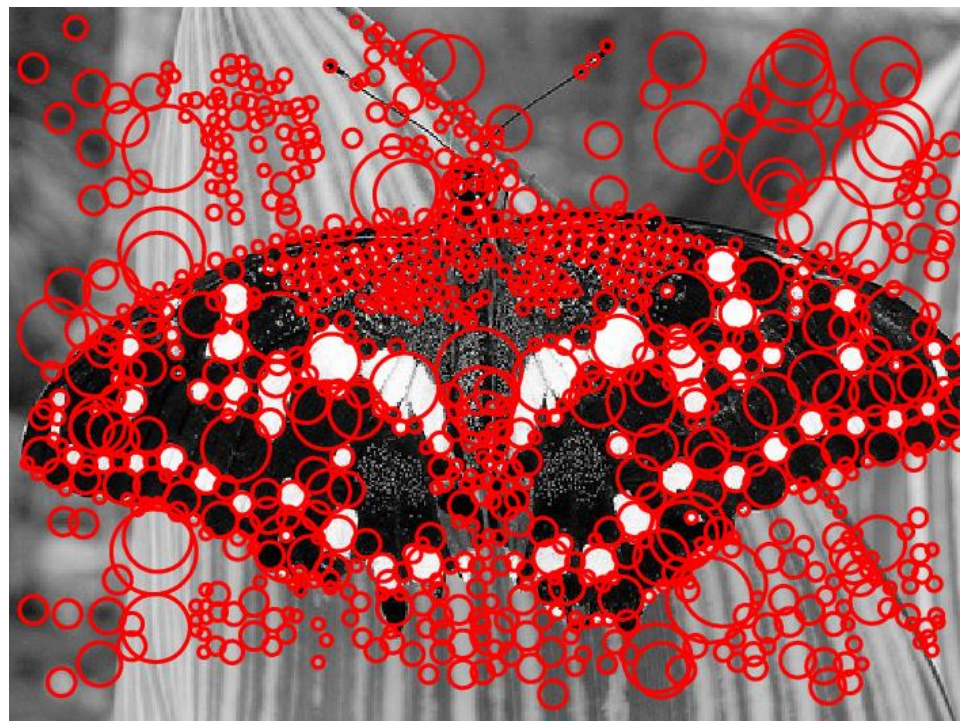
Technical detail

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)

$I(k\sigma)$ - $I(\sigma)$ = $I(k\sigma) - I(\sigma)$
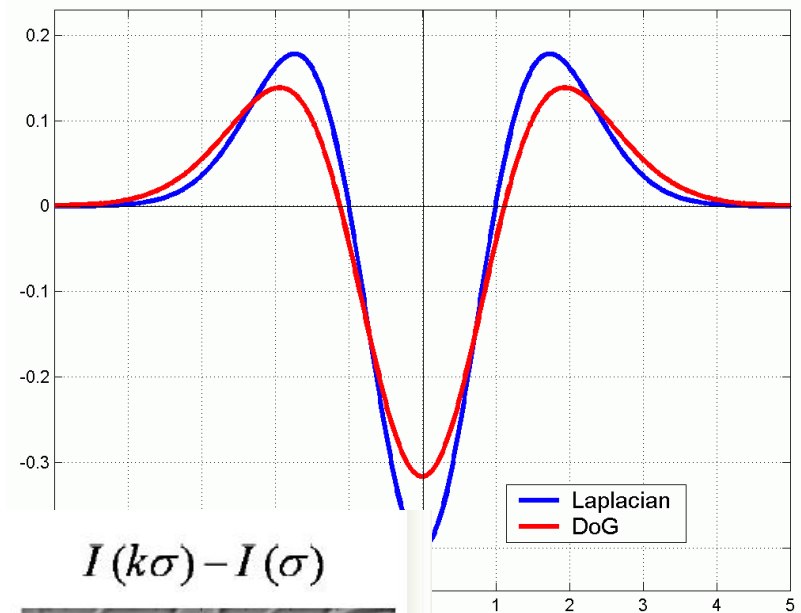
Local features: main components

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

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

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

$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$
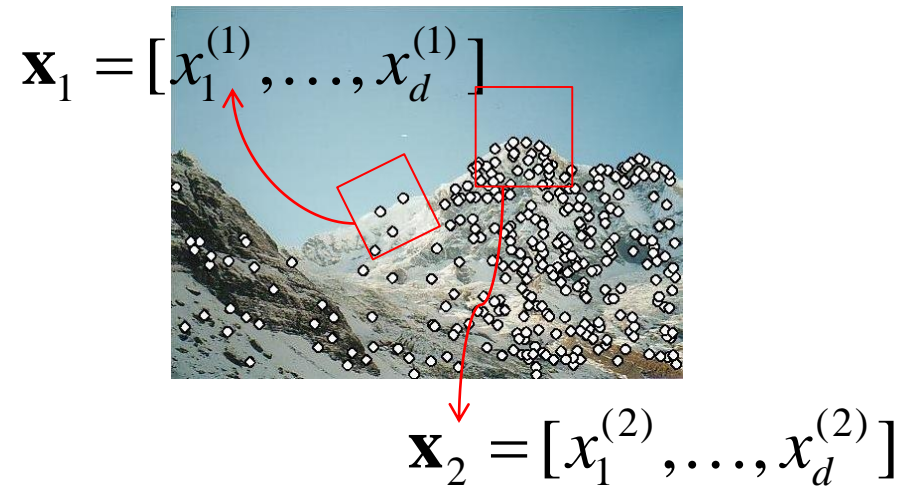
3) **Matching:** Determine correspondence between descriptors in two views

Geometric transformations
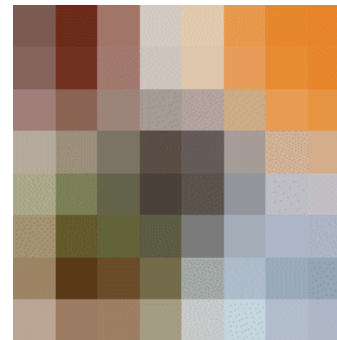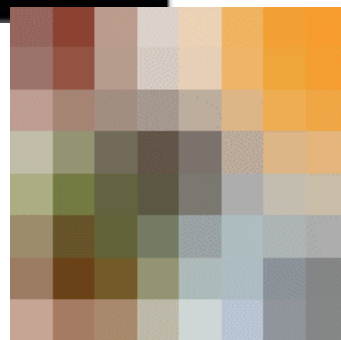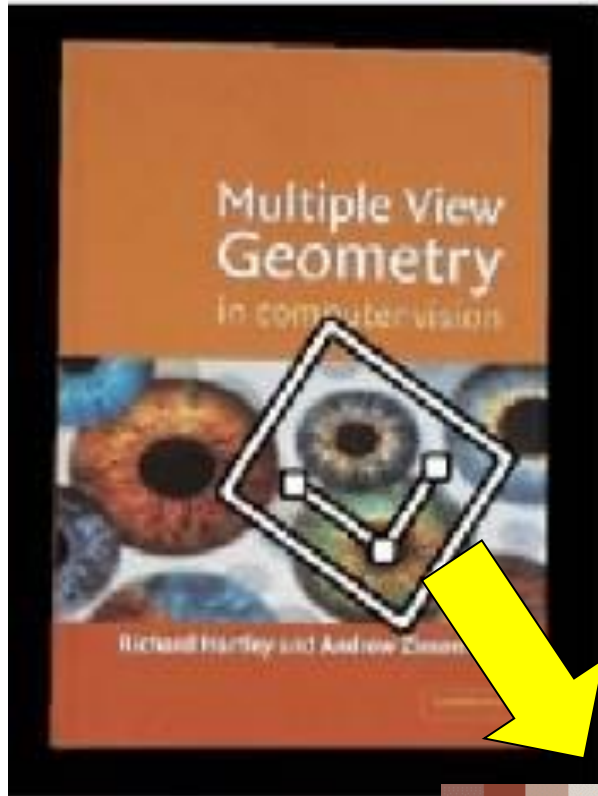
e.g. scale,
translation,
rotation

Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Raw patches as local descriptors

region A   region B
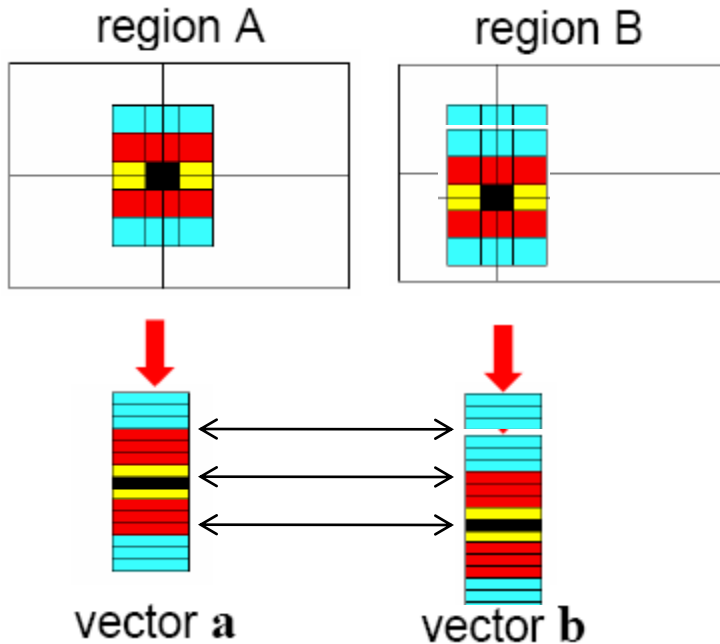
vector **a**   vector **b**

The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.
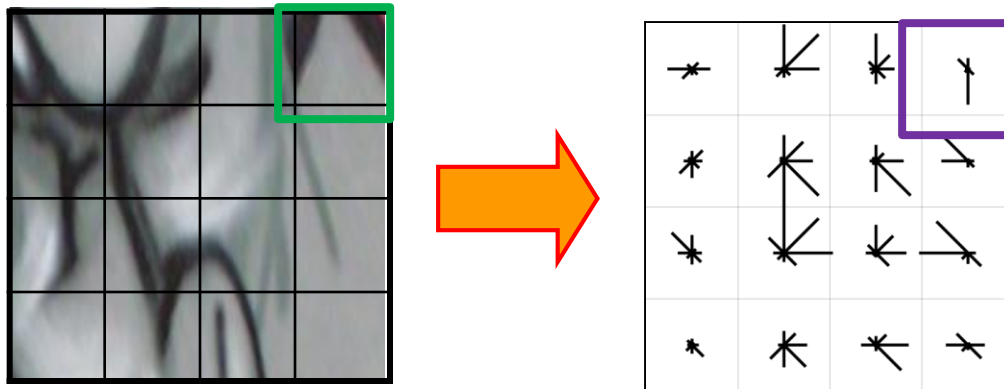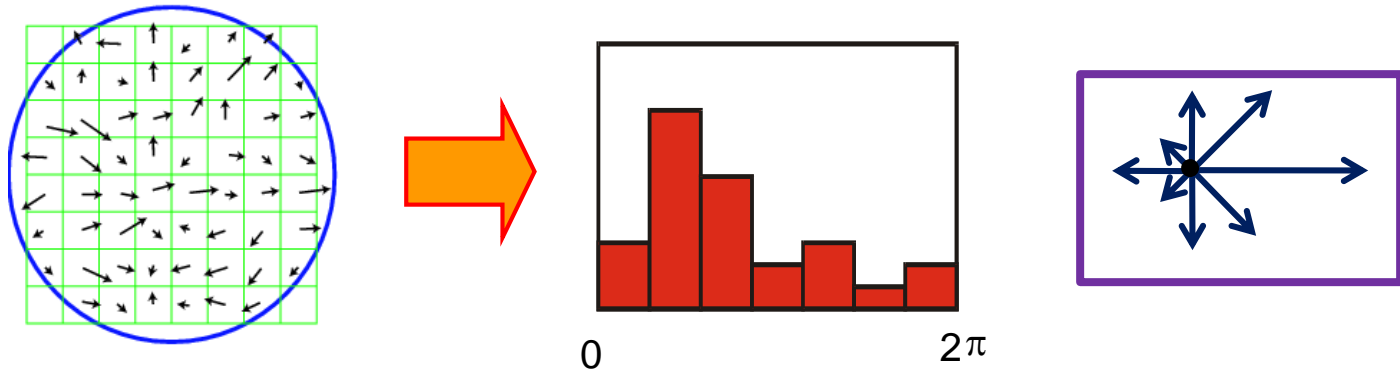
Why subpatches?

Why does SIFT have some illumination invariance?

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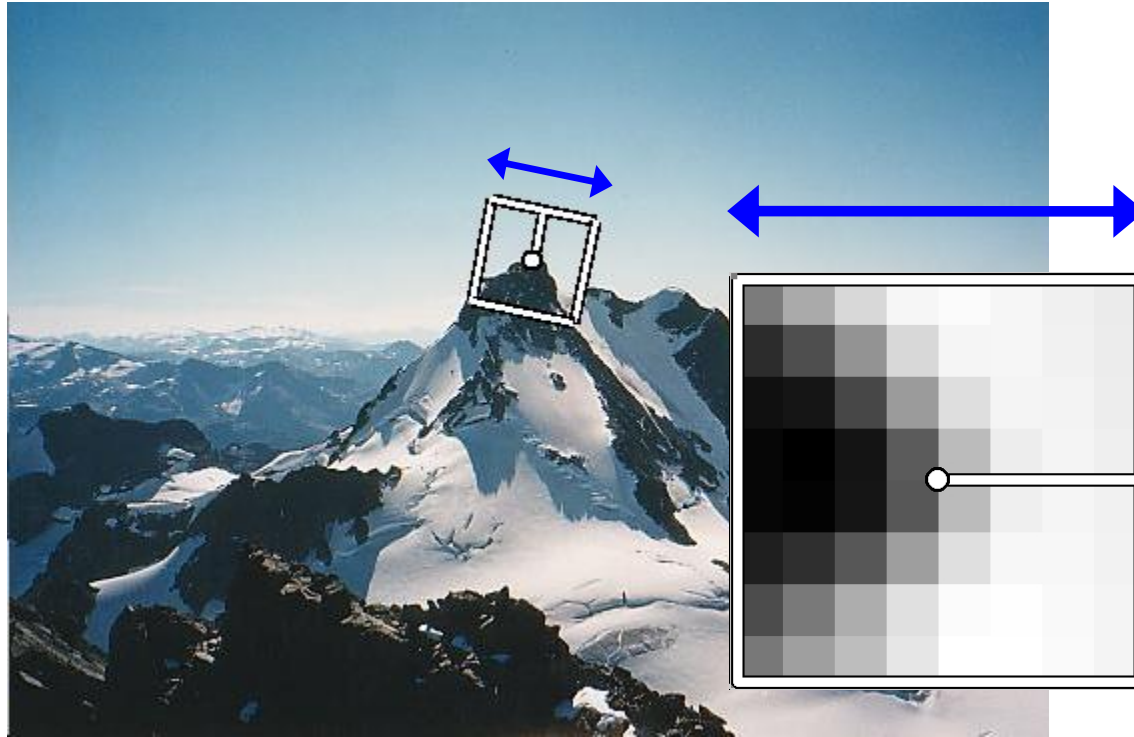
- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

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

Example
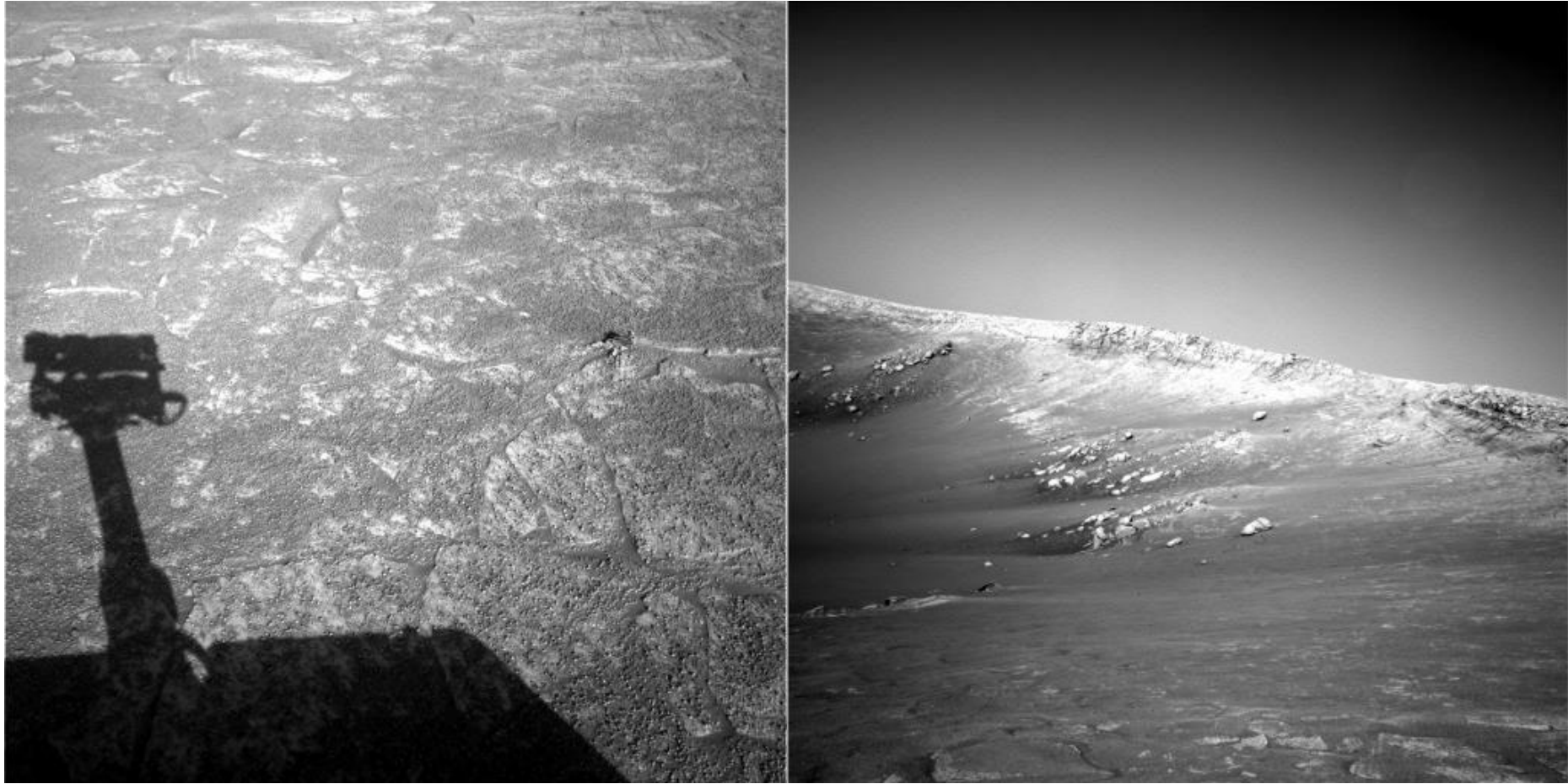
NASA Mars Rover images

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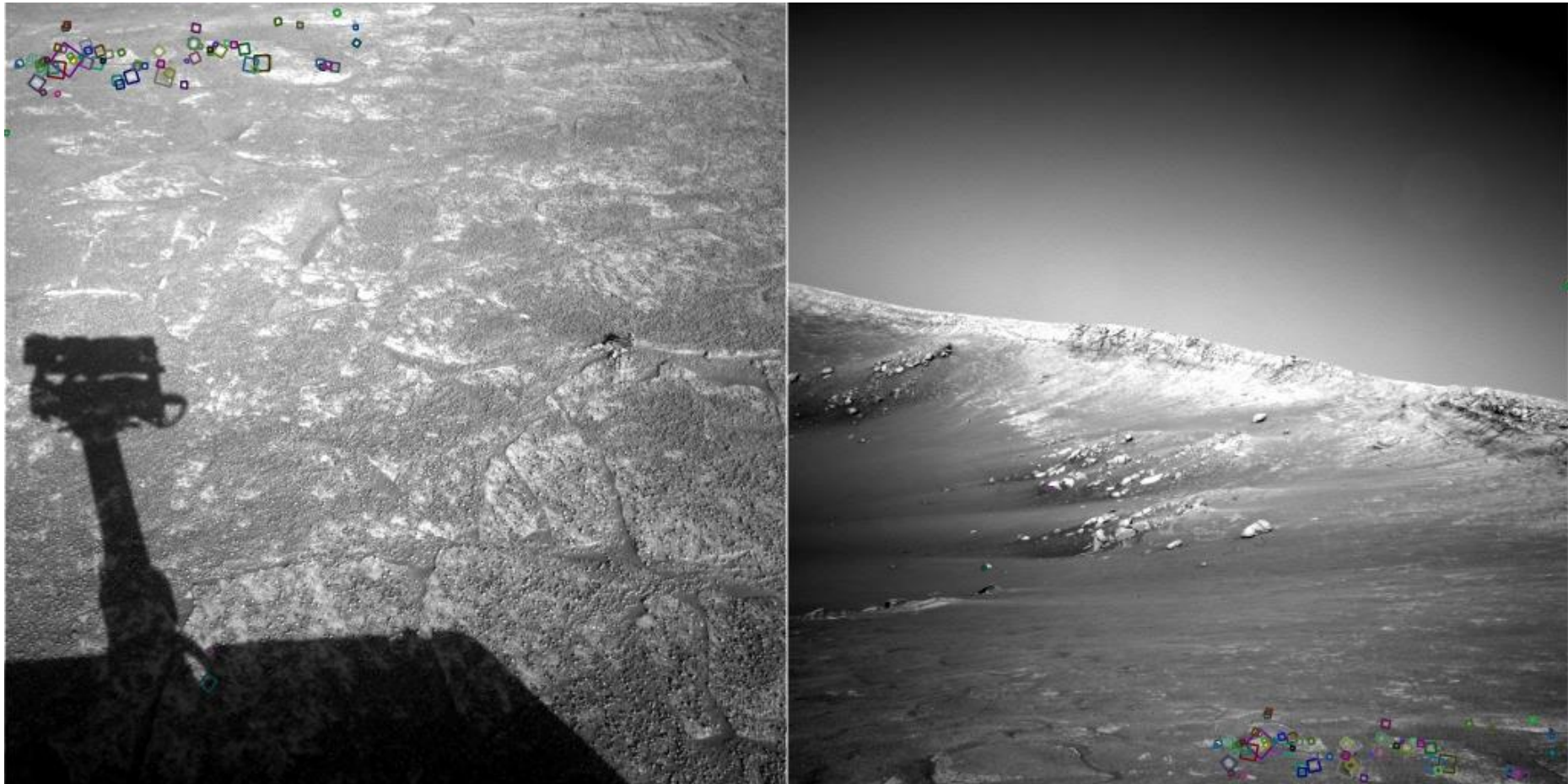
NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

SIFT properties

- Invariant to
  - Scale
  - Rotation

- Partially invariant to
  - Illumination changes
  - Camera viewpoint
 - Occlusion, clutter

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
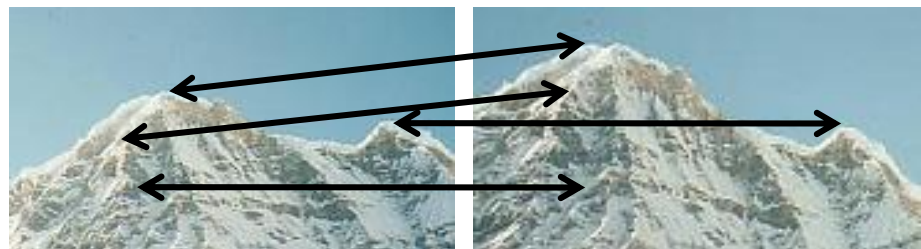
Matching local features

Matching local features
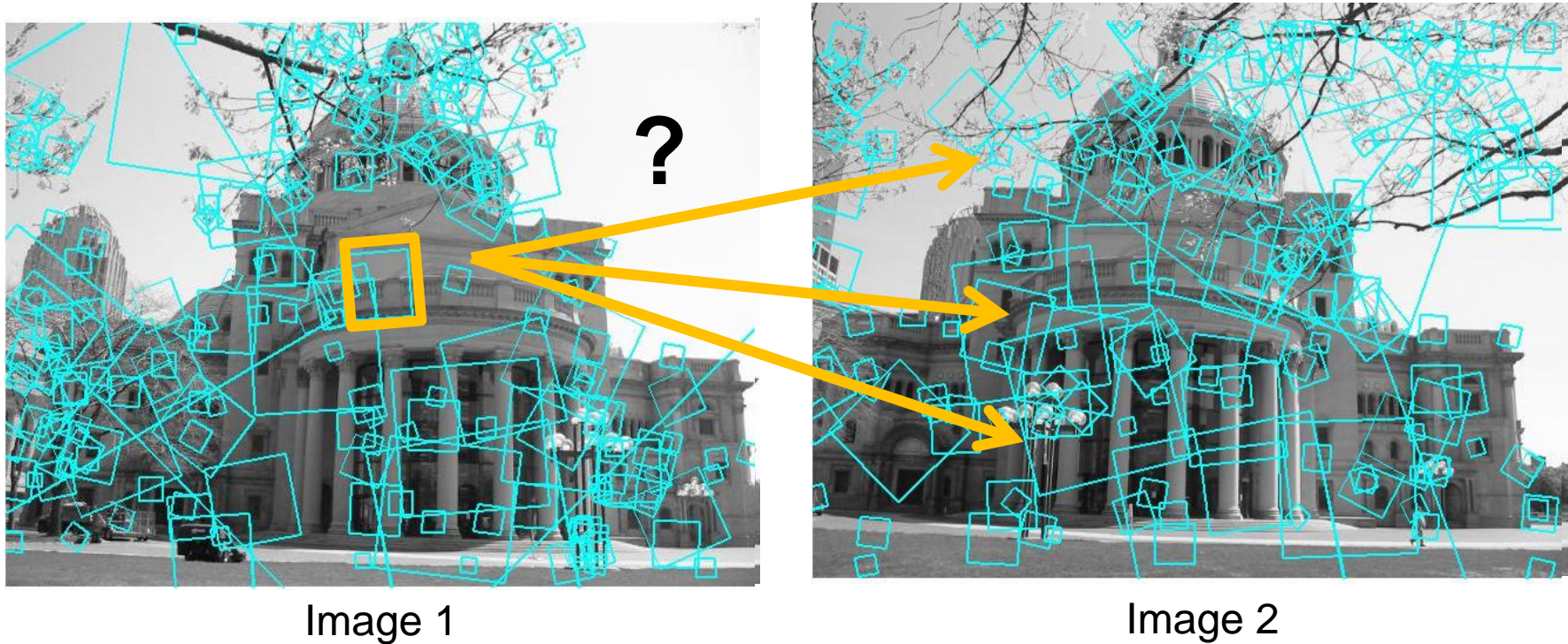
Image 1   Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Ambiguous matches

Image 1

Image 2

At what SSD value do we have a good match?

To add robustness to matching, can consider **ratio** :
distance to best match / distance to second best match

If low, first match looks good.

If high, could be ambiguous match.

Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor

Lowe IJCV 2004
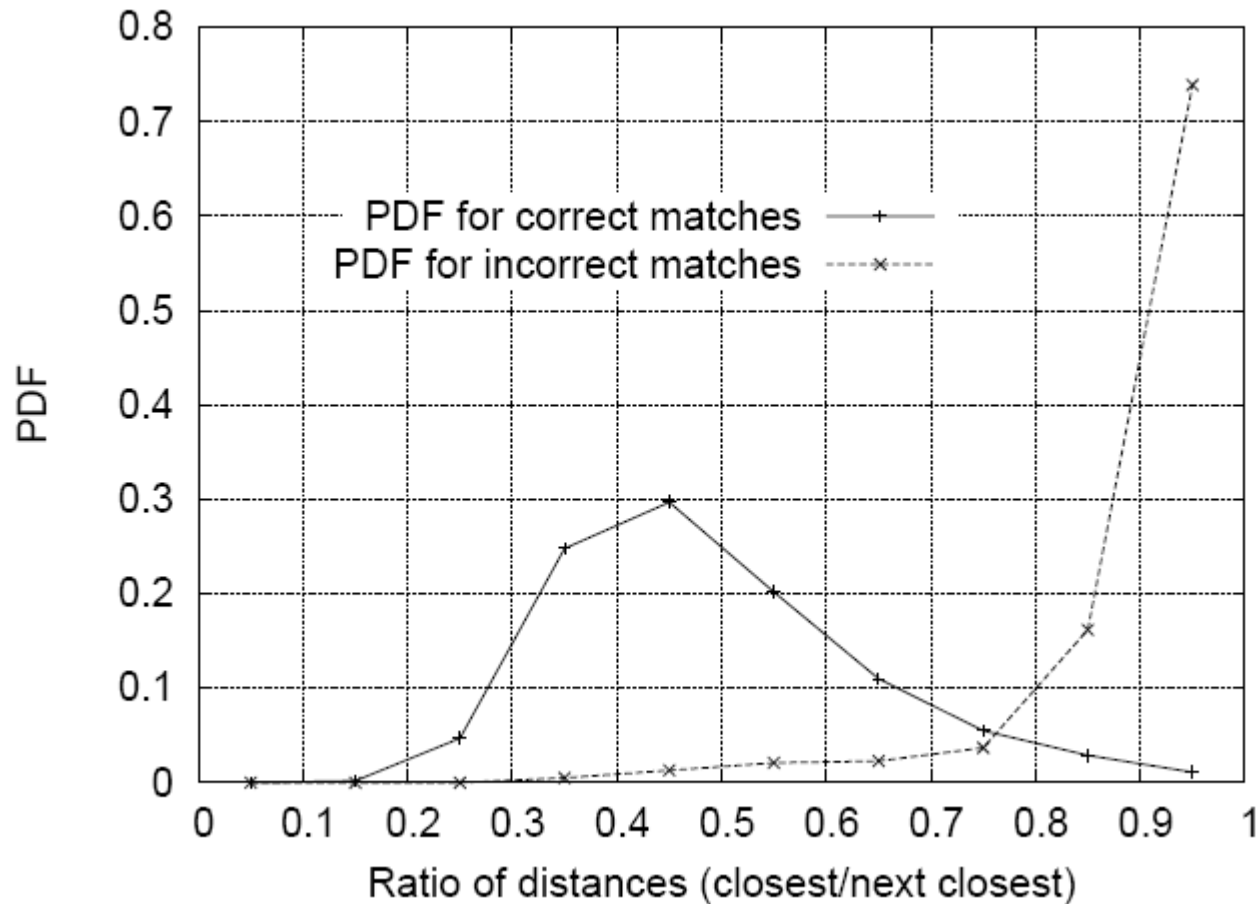
Recap: robust feature-based alignment

Recap: robust feature-based alignment
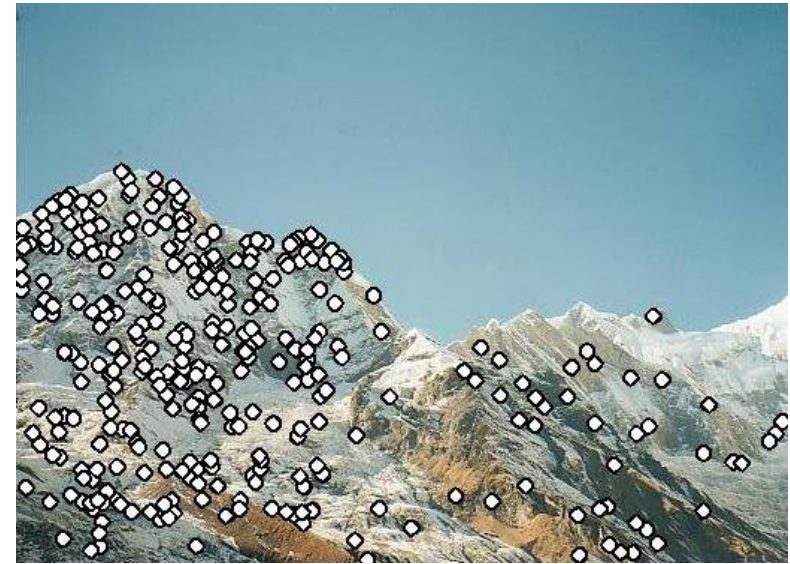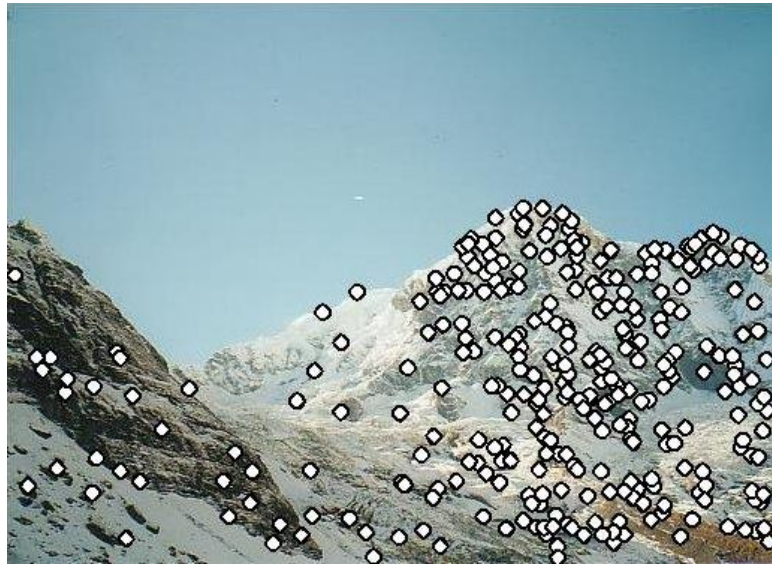
- Extract features

Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*

Recap: robust feature-based alignment
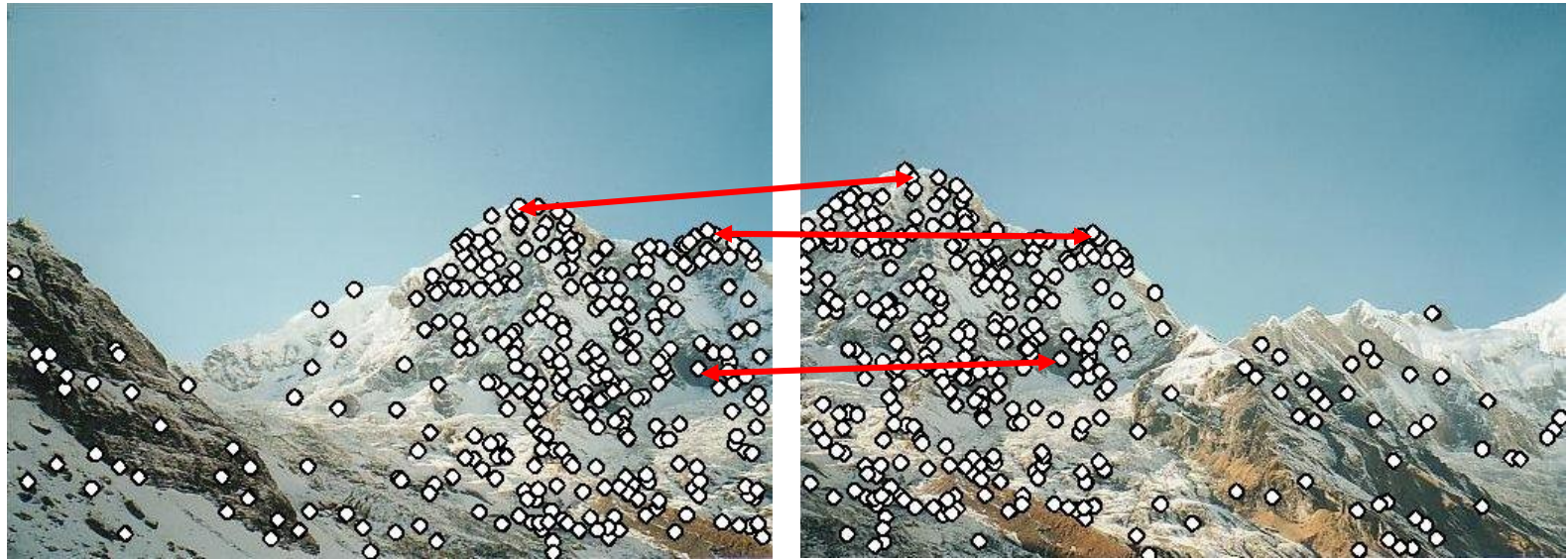
- Extract features

- Compute *putative matches*

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

Recap: robust feature-based alignment
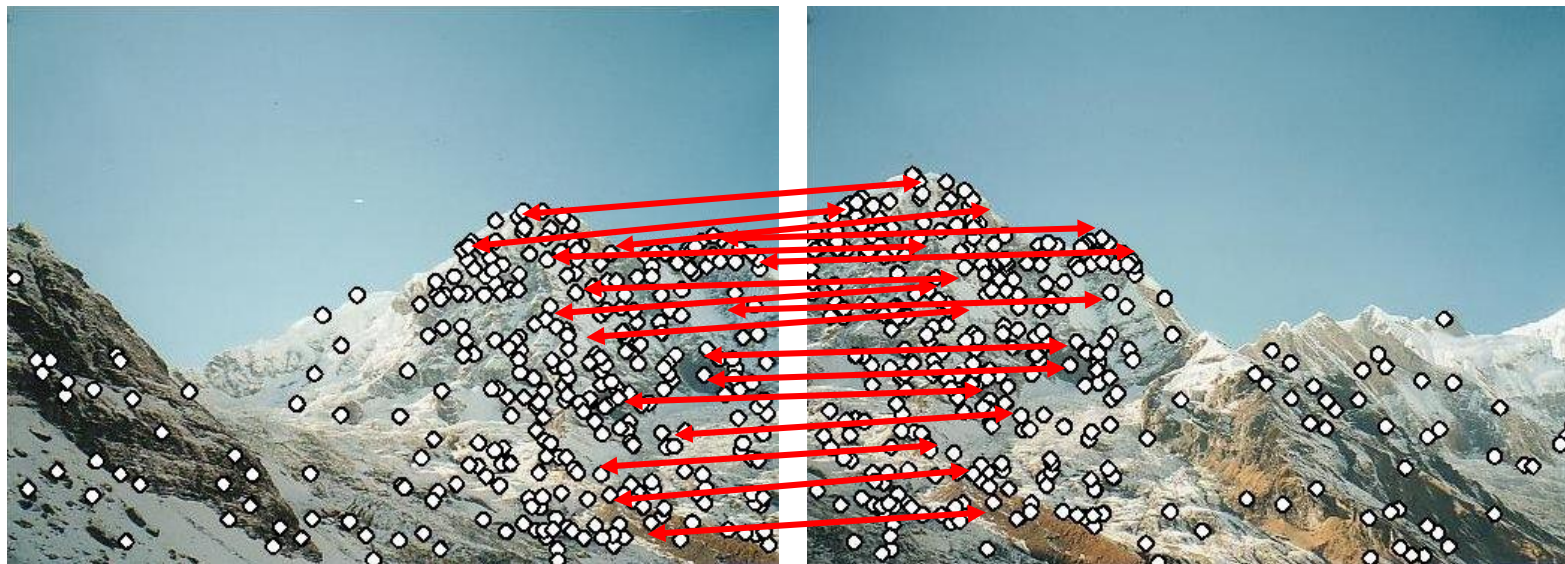
- Extract features

- Compute *putative matches*

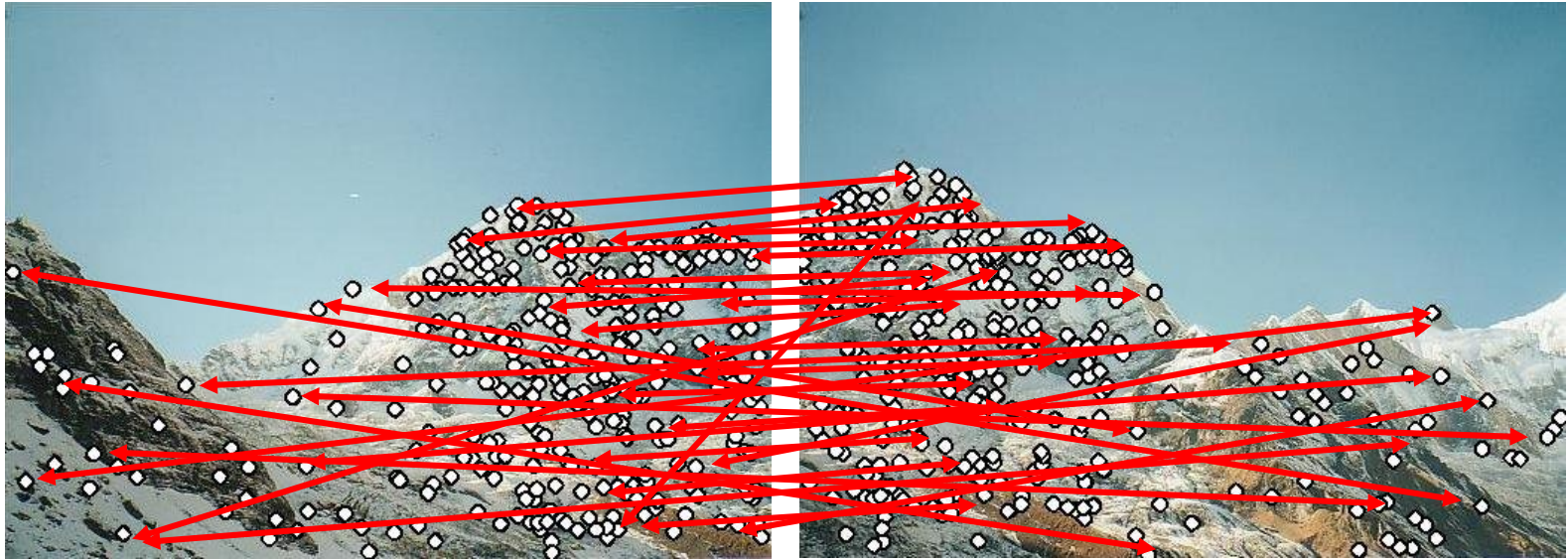- Loop:

  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

  - *Verify* transformation (search for other matches consistent with *T*)

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*)

Applications of local invariant features

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

Automatic mosaicing

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

Kristen Grauman

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)