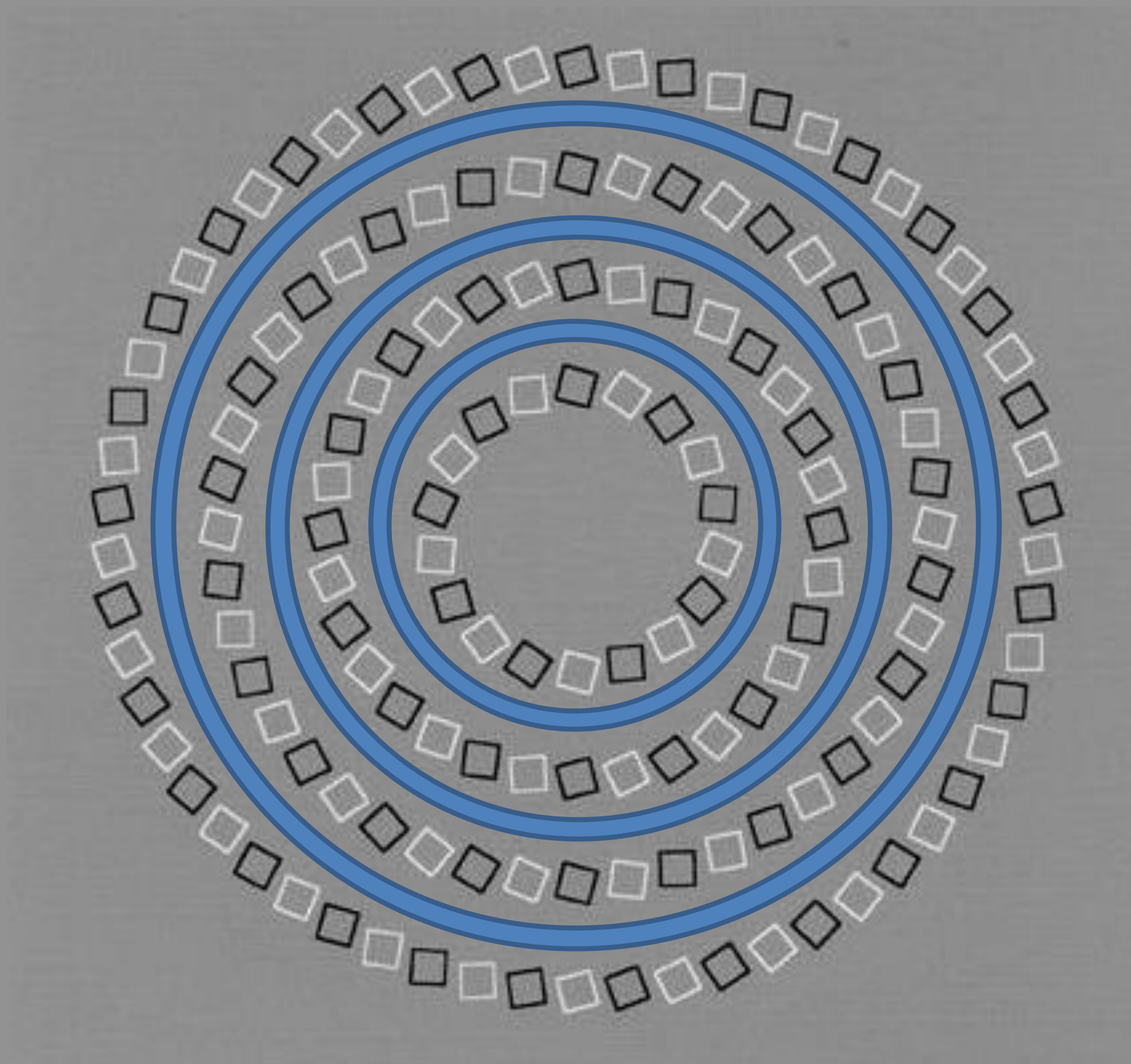


Project 3 questions?

Interest Points and Instance Recognition

Computer Vision

CS 143, Brown

James Hays

Many slides from Kristen Grauman and Derek Hoiem

Last time

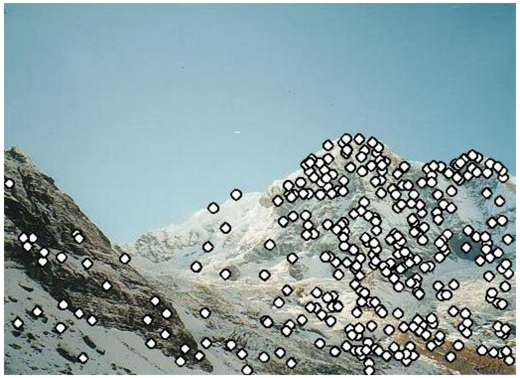
- Detecting corner-like points in an image

Today

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

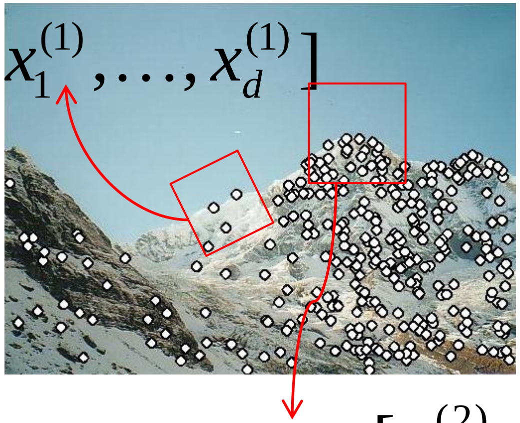
Local invariant features: outline

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



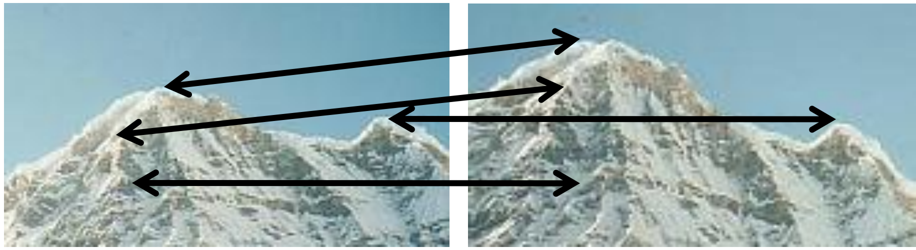
2) **Description**: Extract vector feature descriptor surrounding each interest point.

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



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

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.

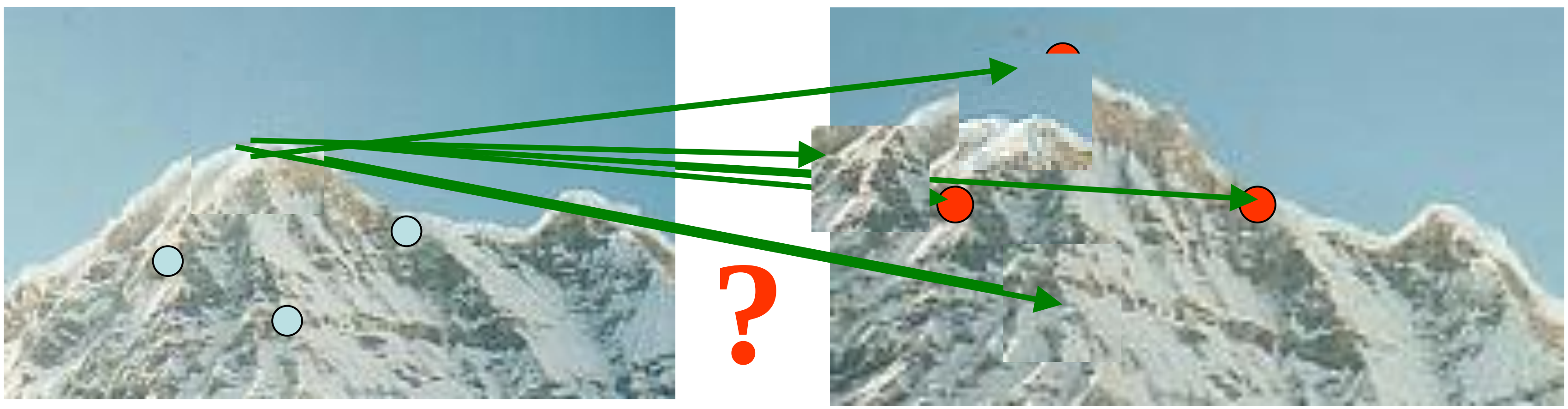
 

No chance to find true matches!

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

Goal: descriptor distinctiveness

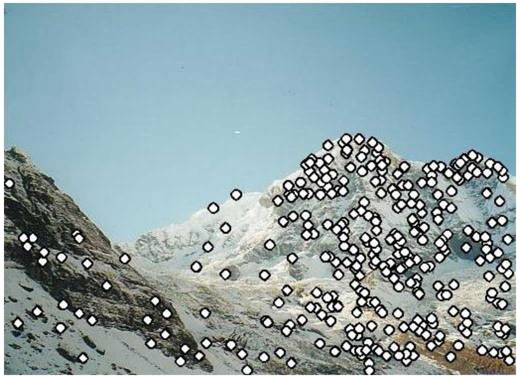
- We want to be able to reliably determine which point goes with which.



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

Local features: main components

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

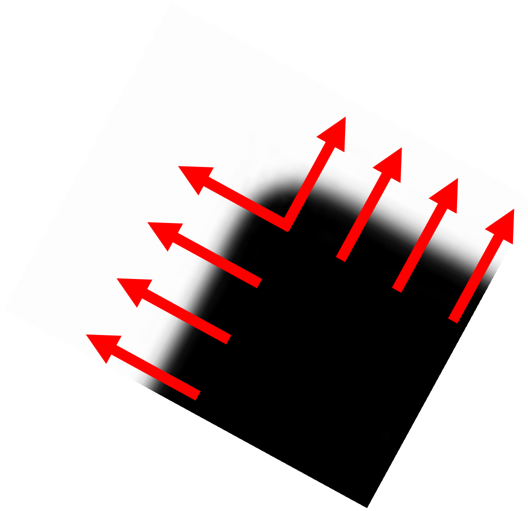


2) **Description:** Extract vector feature descriptor surrounding each interest point.

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

Recall: Corners as distinctive interest points

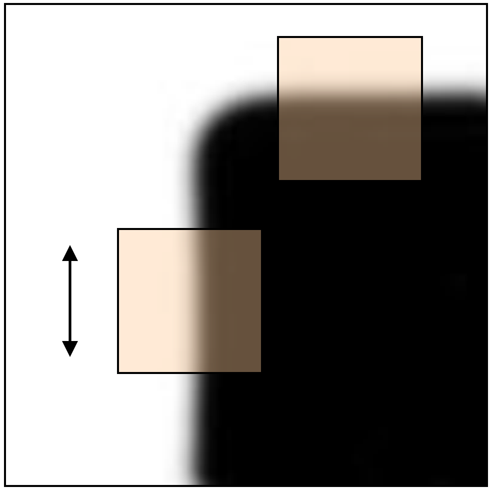
Since *M* is symmetric, we have $M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$



$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of *M* reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.

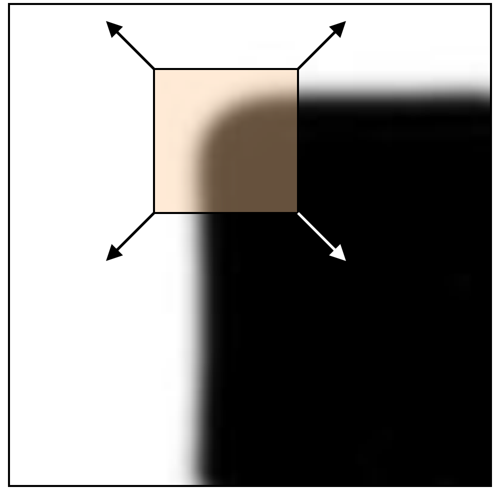
Recall: Corners as distinctive interest points



"edge":

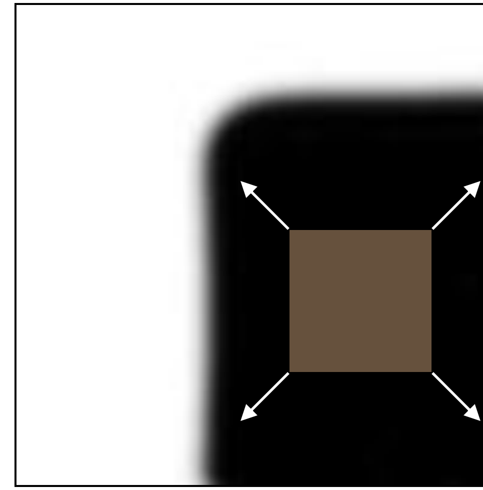
$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$



"corner":

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



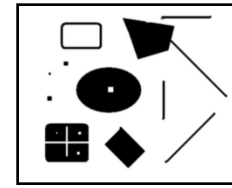
"flat" region

λ_1 and λ_2 are small;

One way to score the cornerness:

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

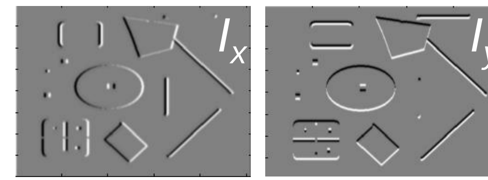
Harris Detector [Harris88]



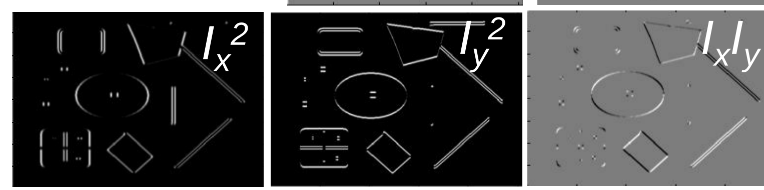
- Second moment matrix (autocorrelation matrix)

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

1. Image derivatives

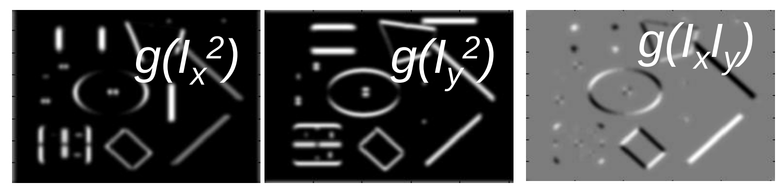


2. Square of derivatives



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

3. Gaussian filter $g(\sigma_I)$



4. Cornerness function – both eigenvalues are strong

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

5. Non-maxima suppression

Properties of the Harris corner detector

Rotation invariant? Yes

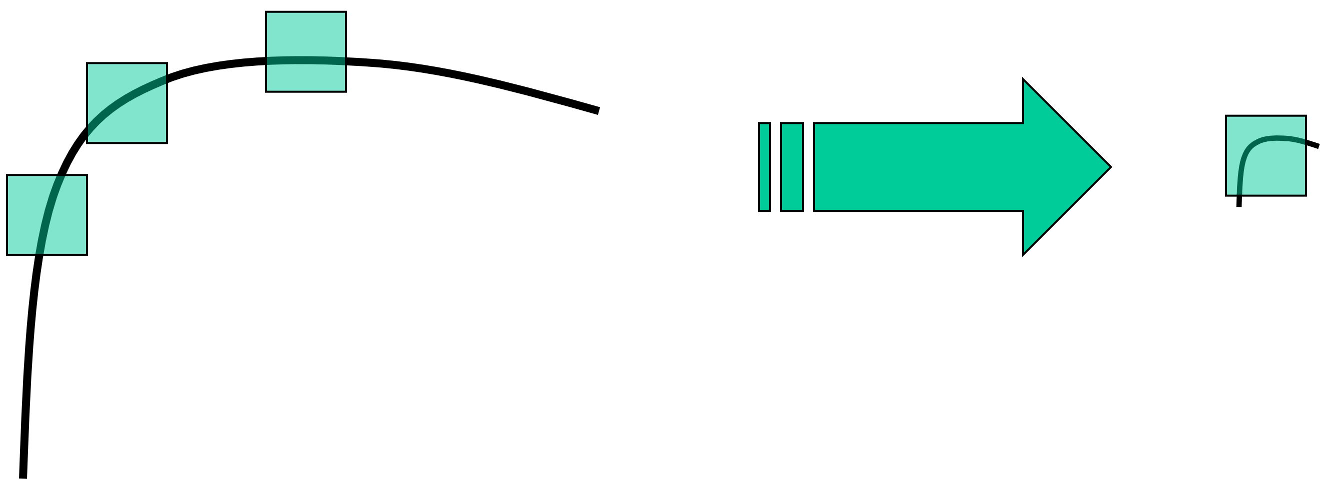
$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$$

Scale invariant?

Properties of the Harris corner detector

Rotation invariant? Yes

Scale invariant? No



All points will be classified as edges

Corner !

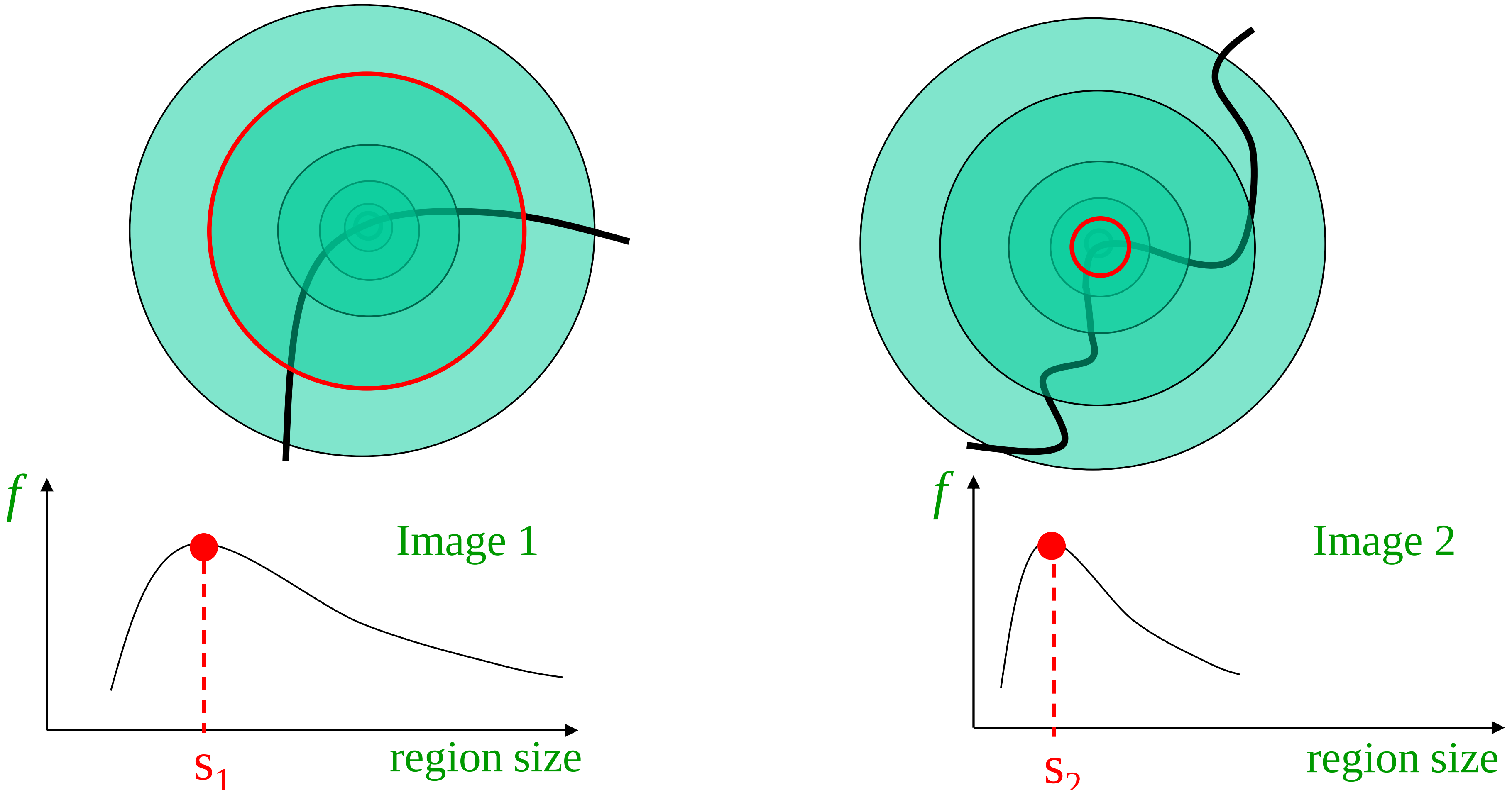
Scale invariant interest points

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

Automatic scale selection

Intuition:

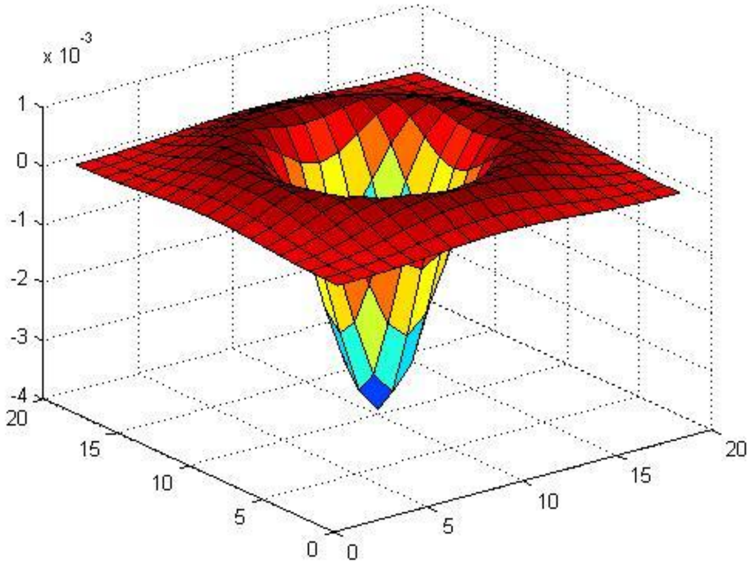
- Find scale that gives local maxima of some function *f* in both position and scale.

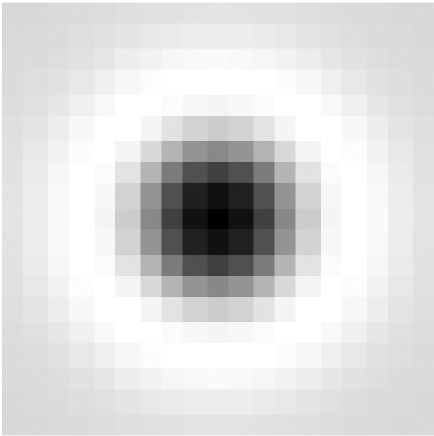


What can be the “signature” function?

Blob detection in 2D

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

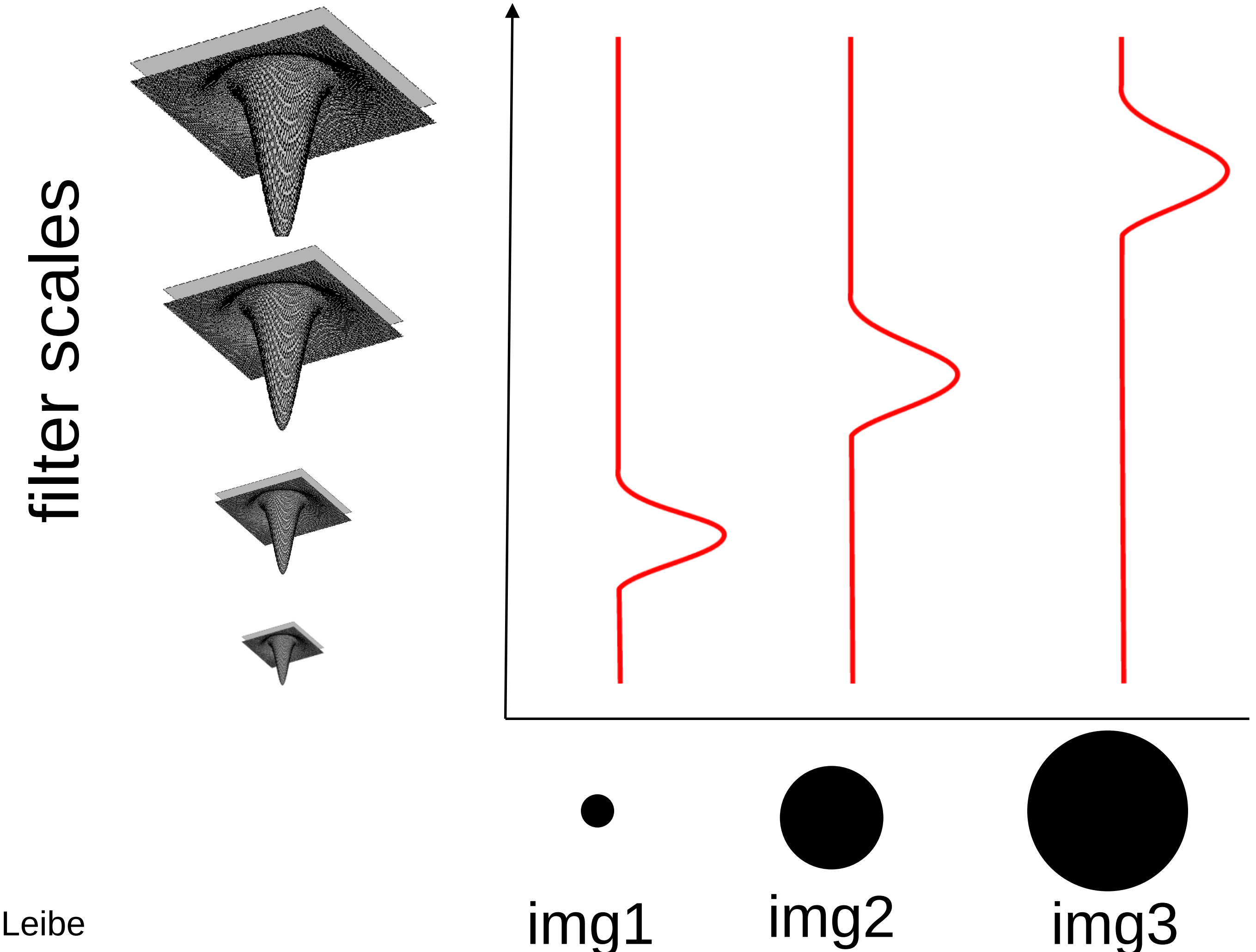




$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

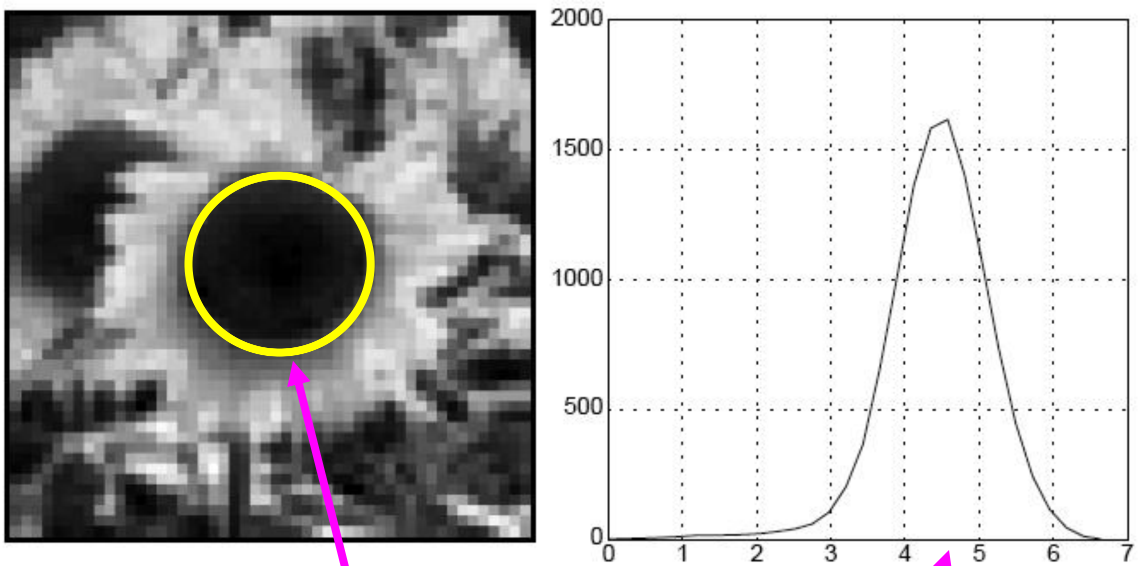
Blob detection in 2D: scale selection

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



Blob detection in 2D

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



characteristic scale

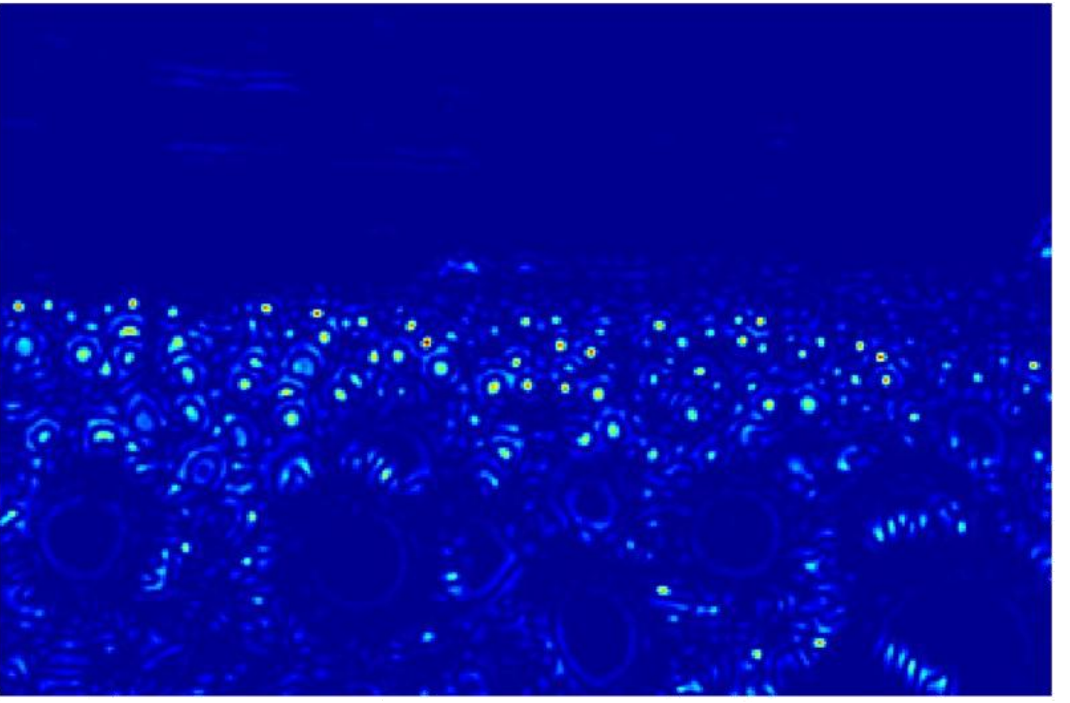
Slide credit: Lana Lazebnik

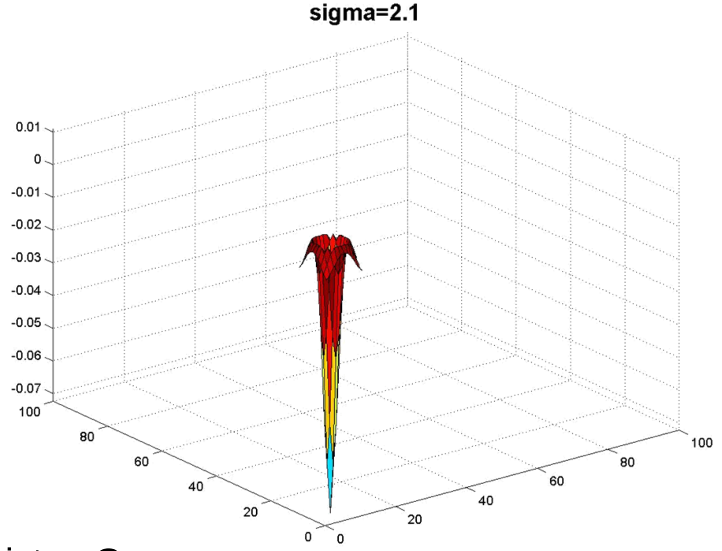
Example

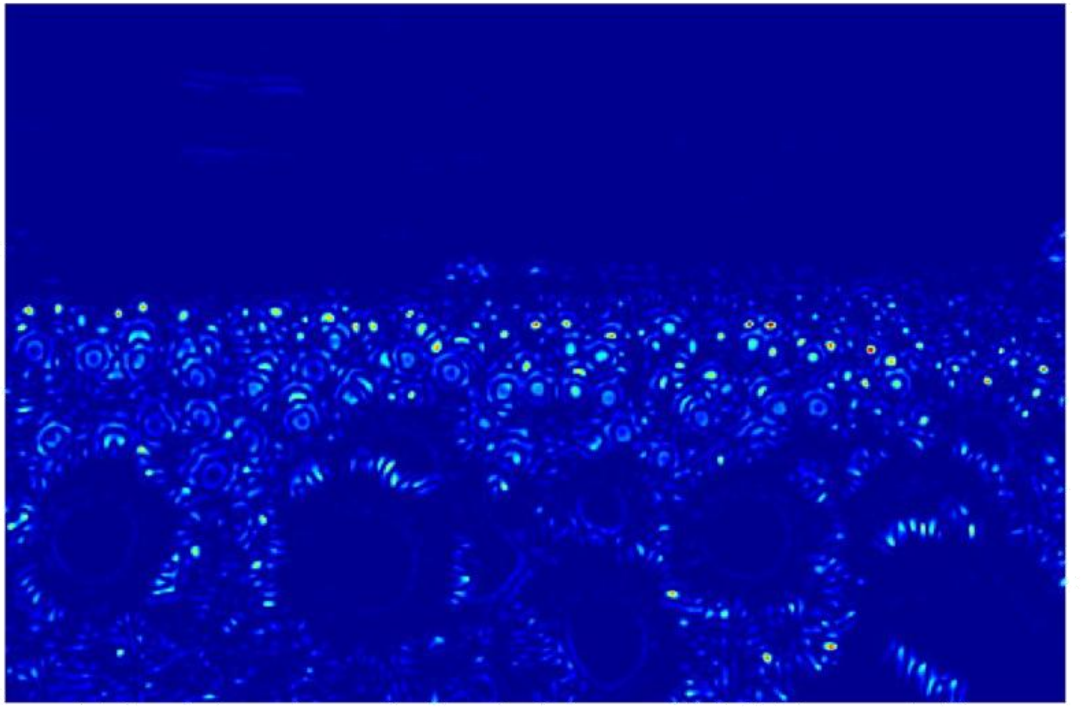
Original image at ¾ the size

Kristen Grauman

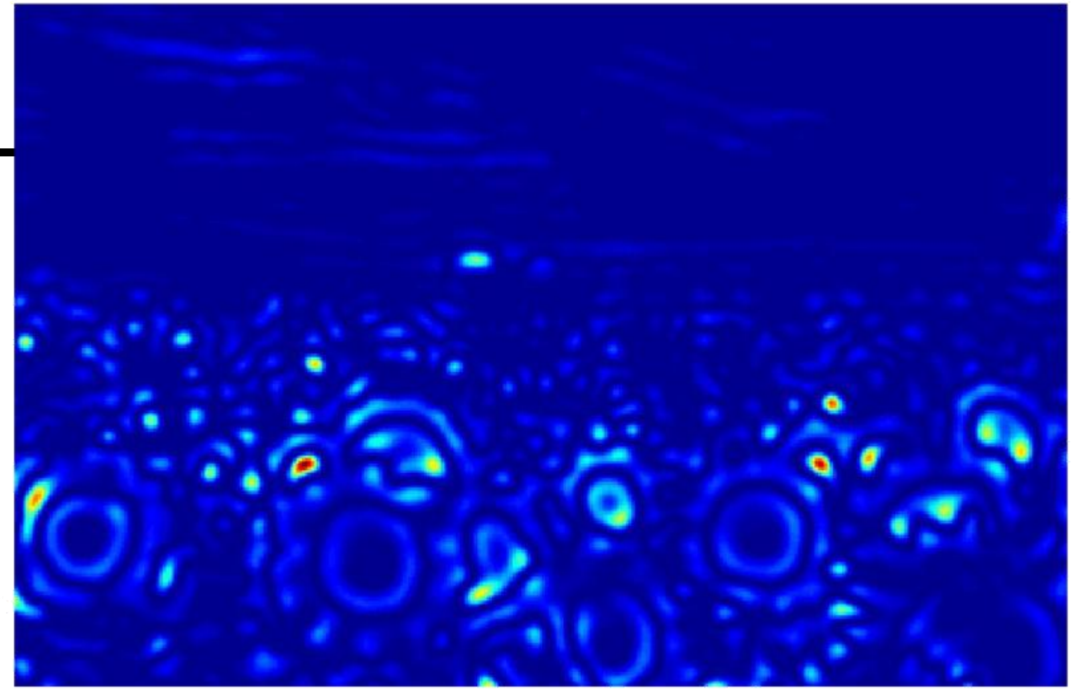
Original image
at ¾ the size

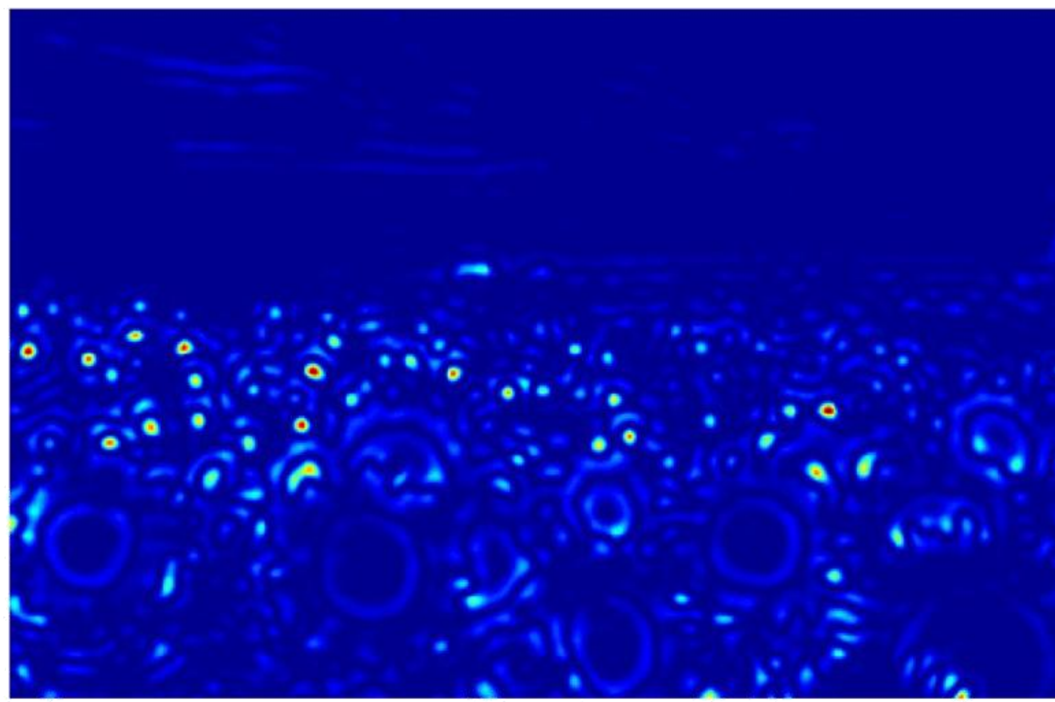


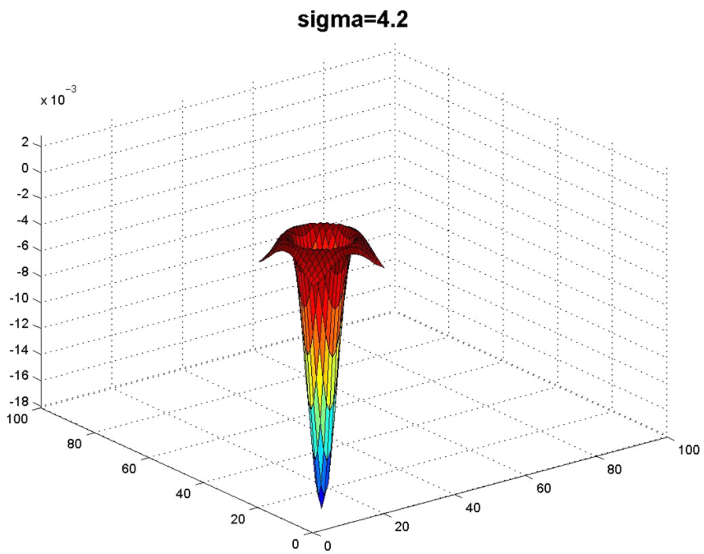




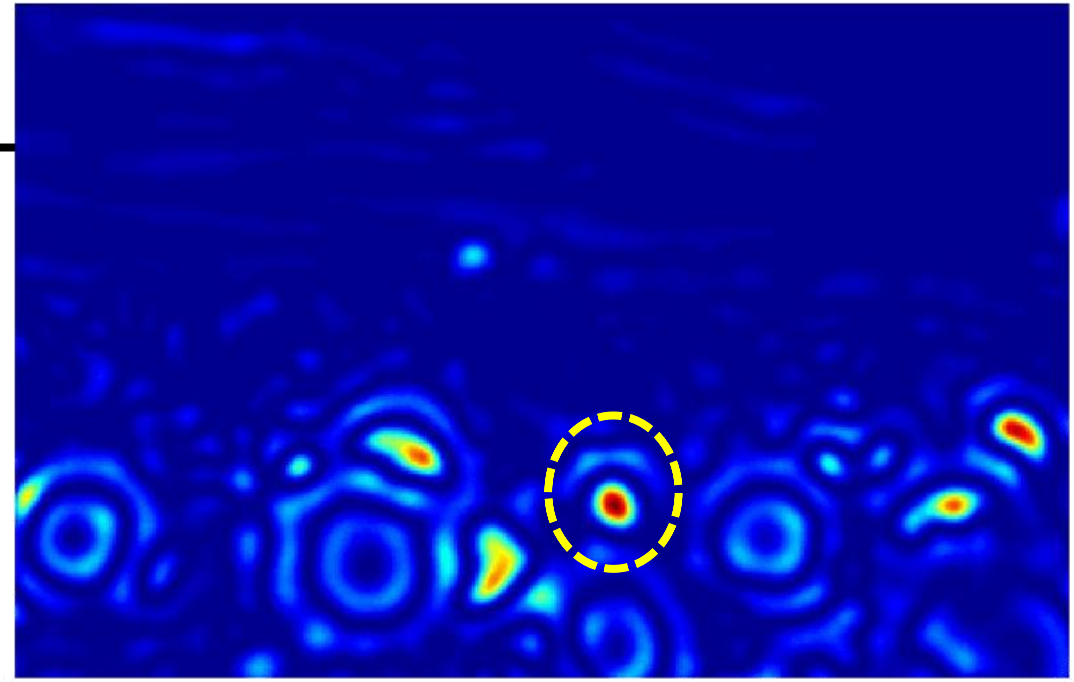
Kristen Grauman

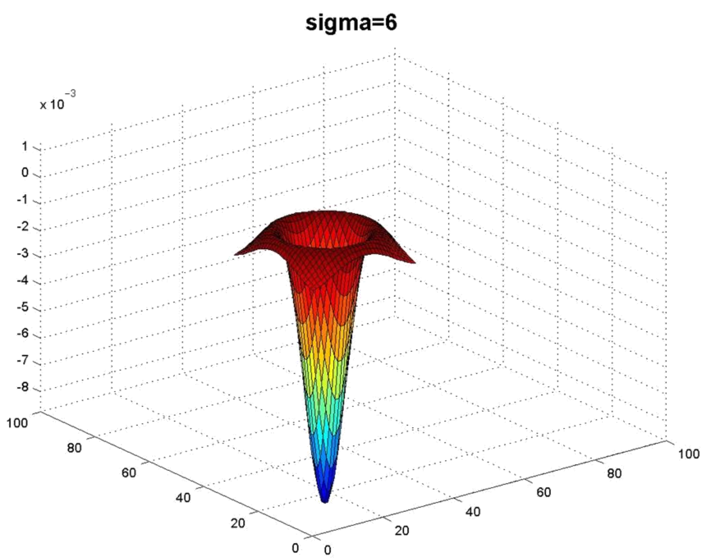


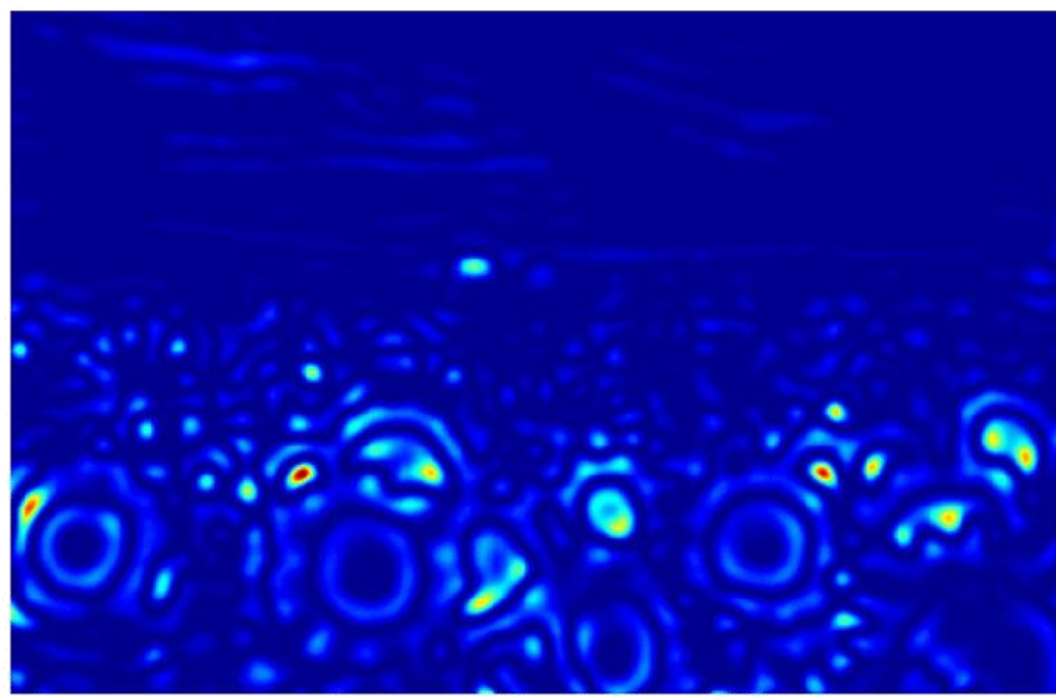




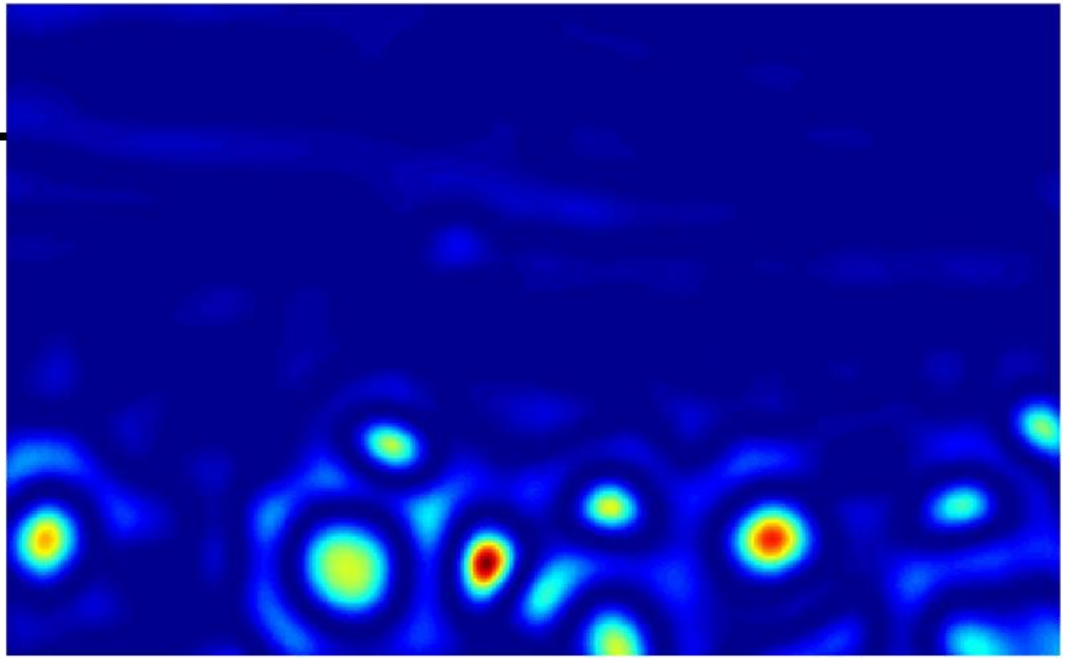
Kristen Grauman

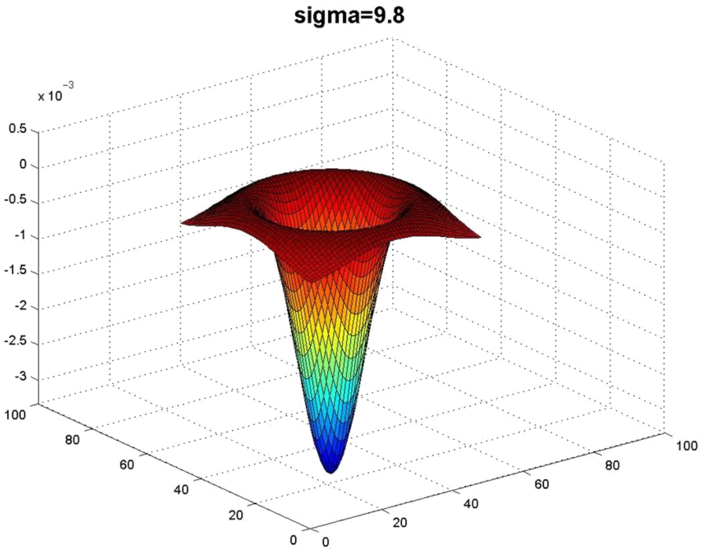


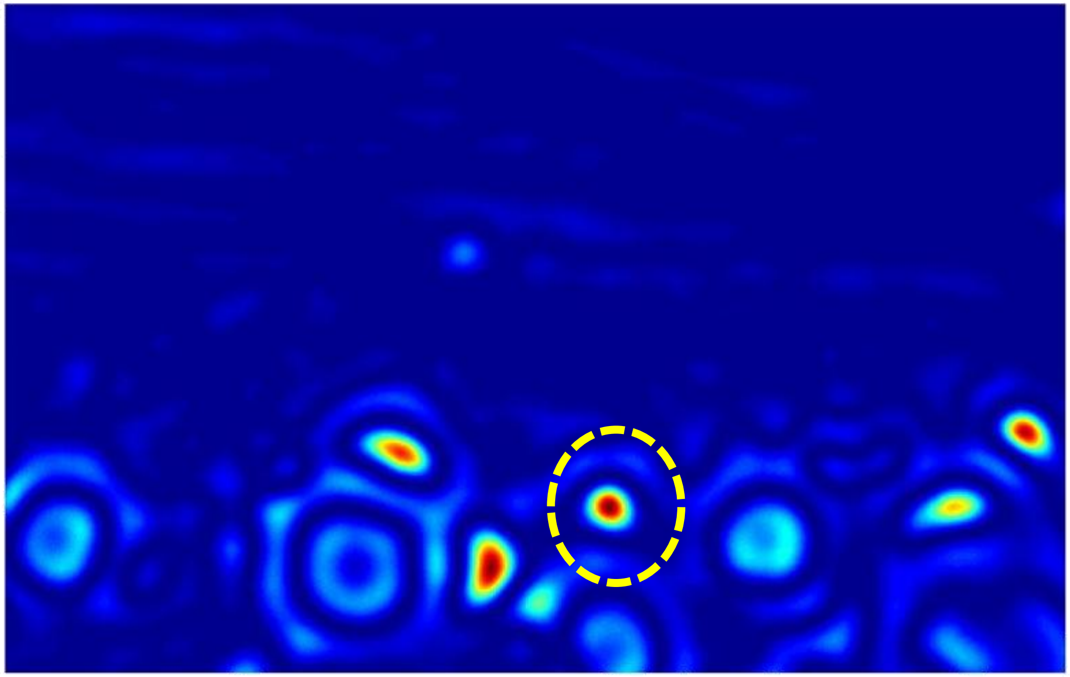




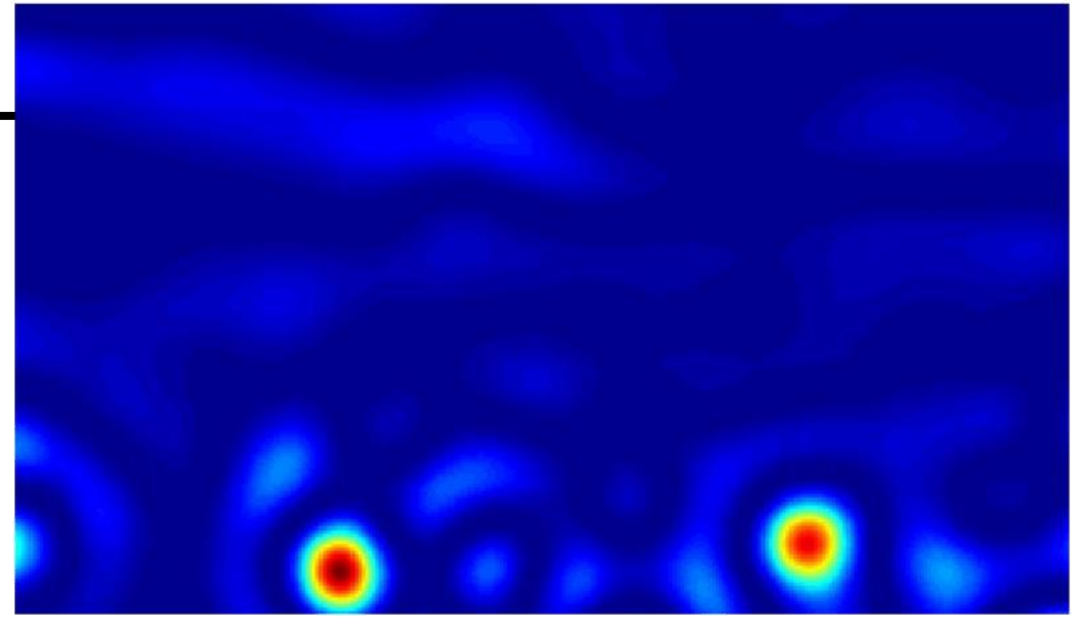
Kristen Grauman

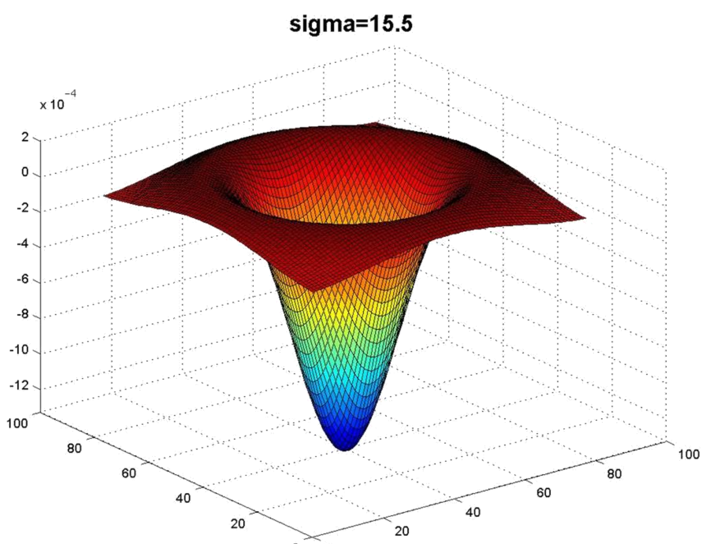


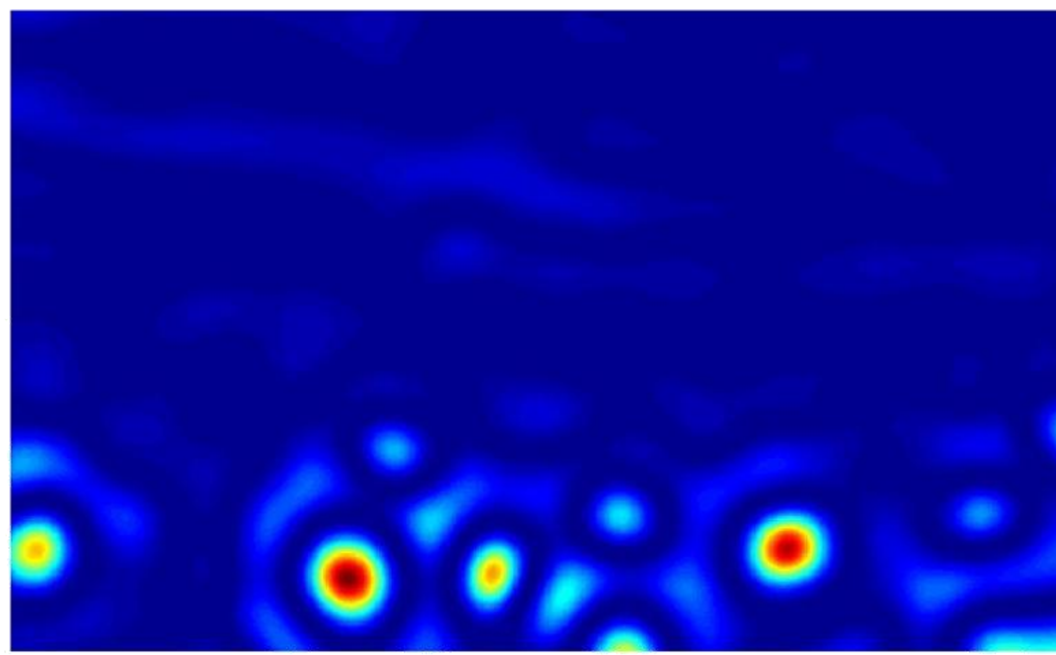




Kristen Grauman



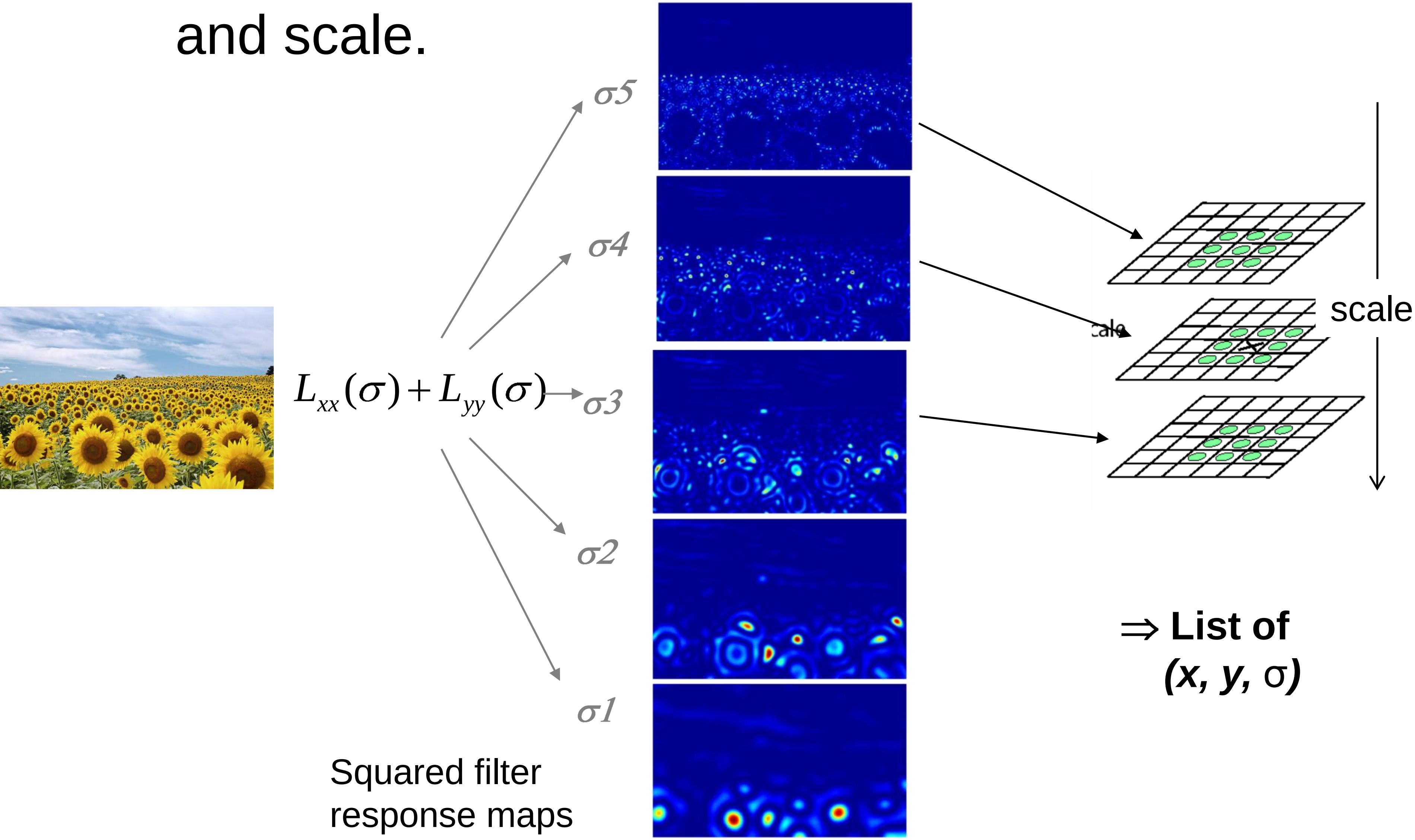




Kristen Grauman

Scale invariant interest points

Interest points are local maxima in both position and scale.



Scale-space blob detector: Example

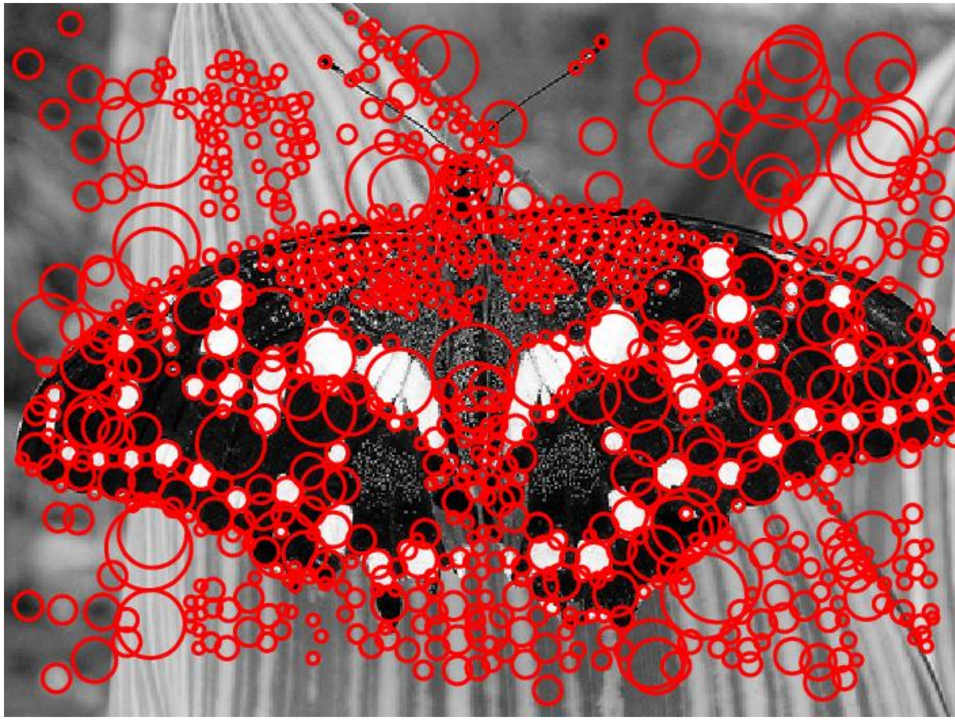


Image credit: Lana Lazebnik

Technical detail

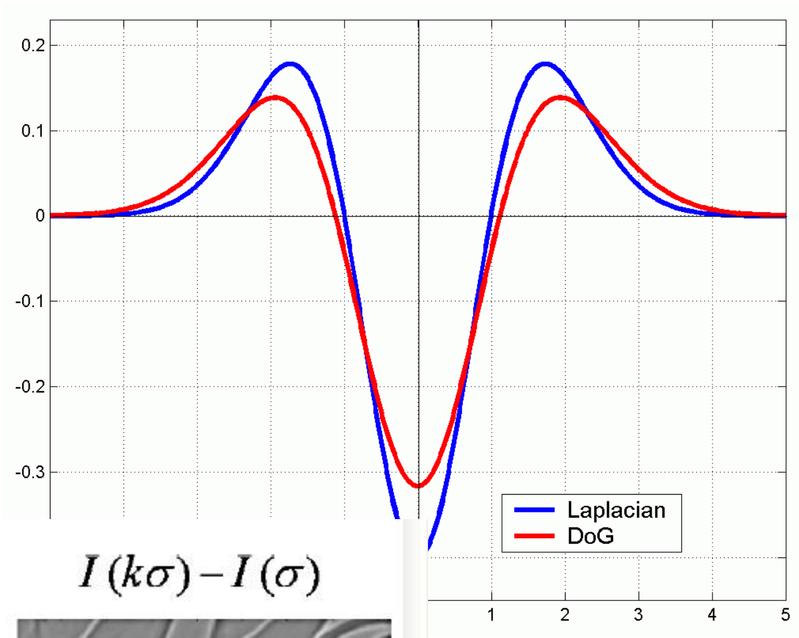
We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

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

(Laplacian)

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

(Difference of Gaussians)



$I(k\sigma)$

-

$I(\sigma)$

=

$I(k\sigma) - I(\sigma)$

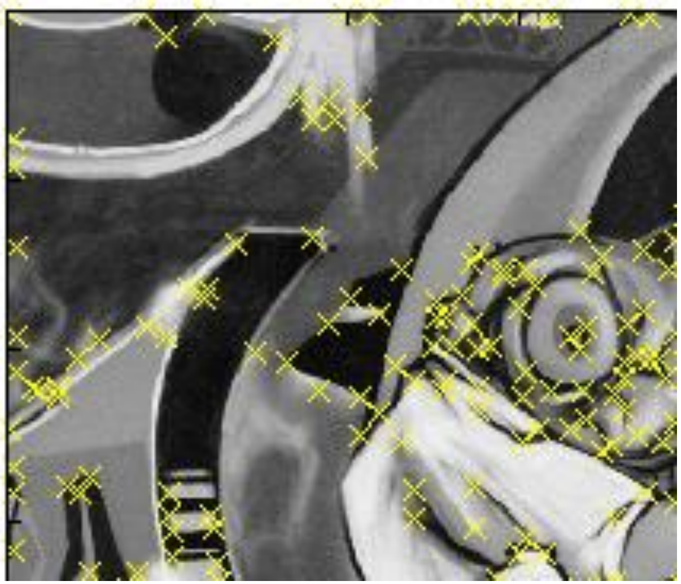
Maximally Stable Extremal Regions [Matas '02]

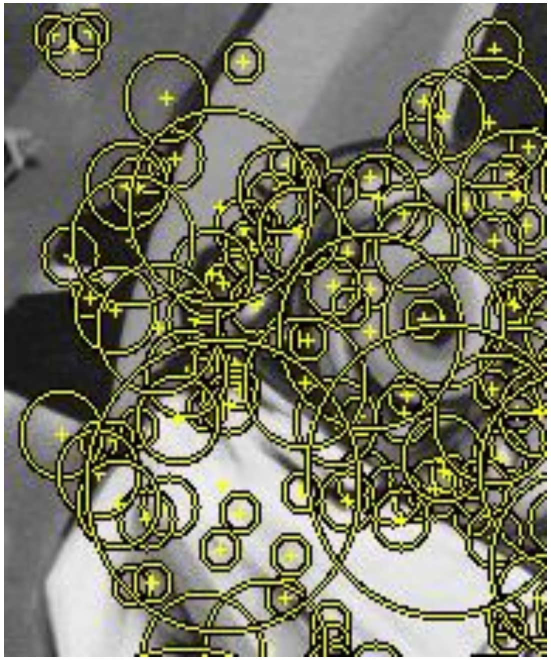
- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range

Example Results: MSER

Comparison

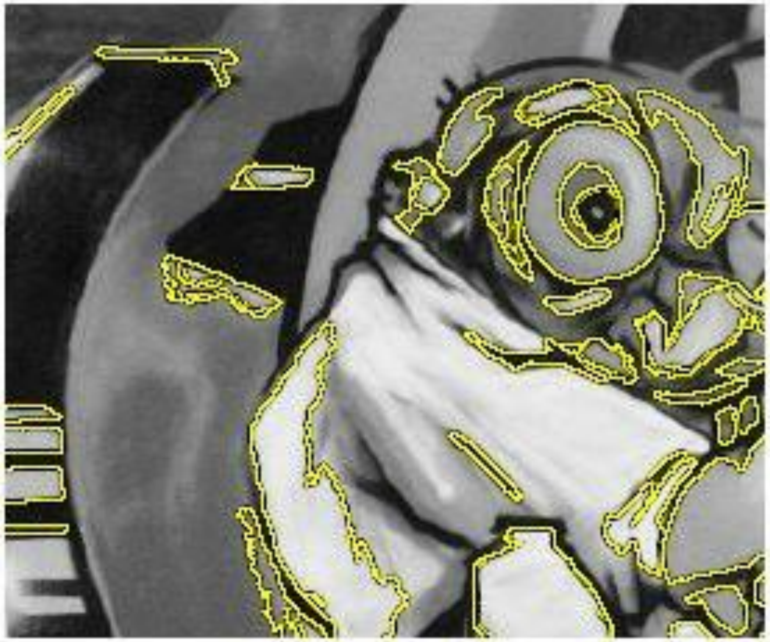
Harris





LoG

MSER



Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Tuytelaars Mikolajczyk 2008

Choosing a detector

- What do you want it for?
  - Precise localization in x-y: Harris
  - Good localization in scale: Difference of Gaussian
  - Flexible region shape: MSER
- Best choice often application dependent
  - Harris-/Hessian-Laplace/DoG work well for many natural categories
  - MSER works well for buildings and printed things
- Why choose?
  - Get more points with more detectors
- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV’05, PAMI’05]
  - All detectors/descriptors shown here work well

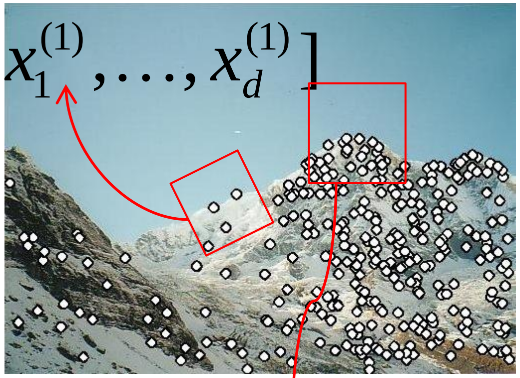
- For most local feature detectors, executables are available online:
  - http://robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.vision.ee.ethz.ch/~surf

Local features: main components

1) Detection: Identify the interest points

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

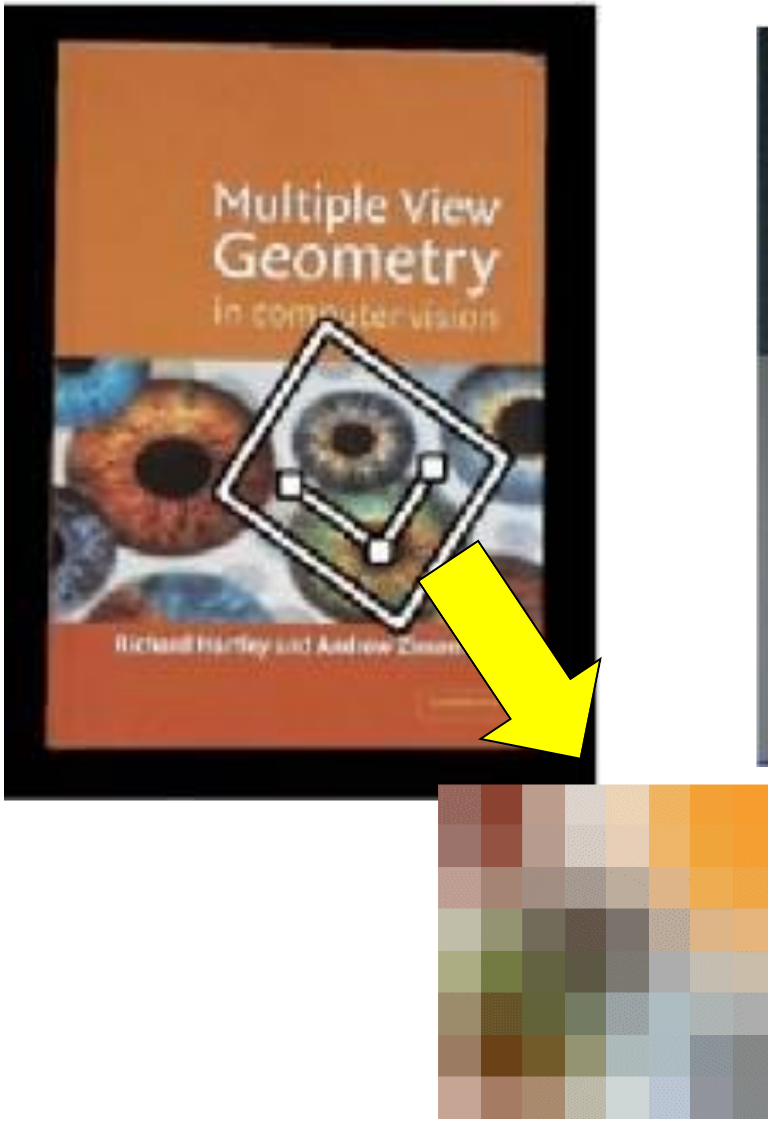
$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

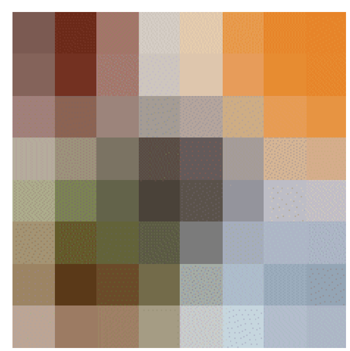


$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views

Geometric transformations



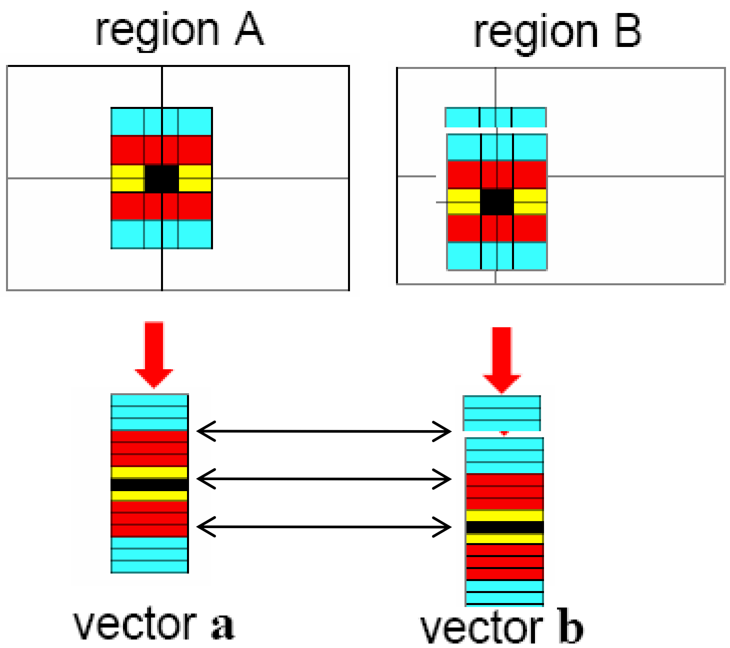


e.g. scale,
translation,
rotation

Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Raw patches as local descriptors

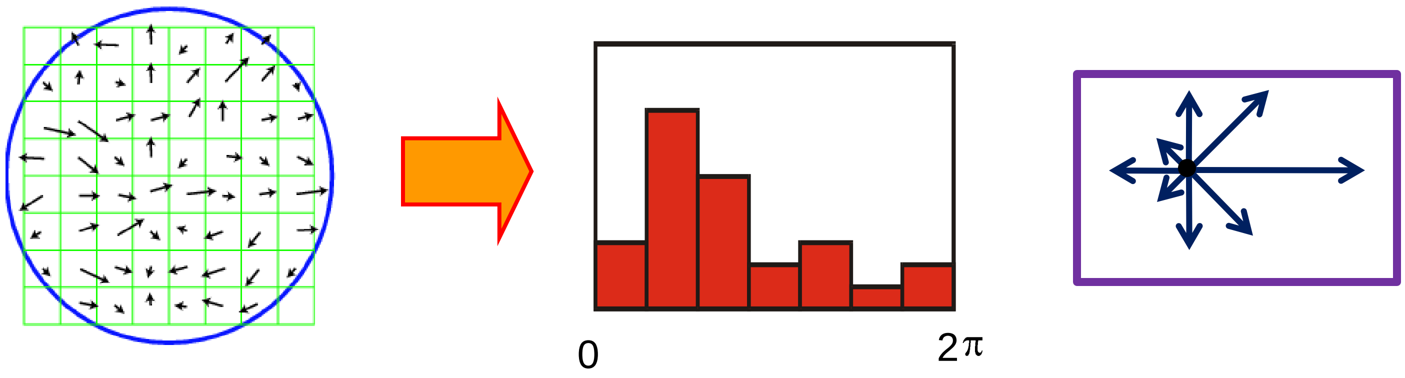


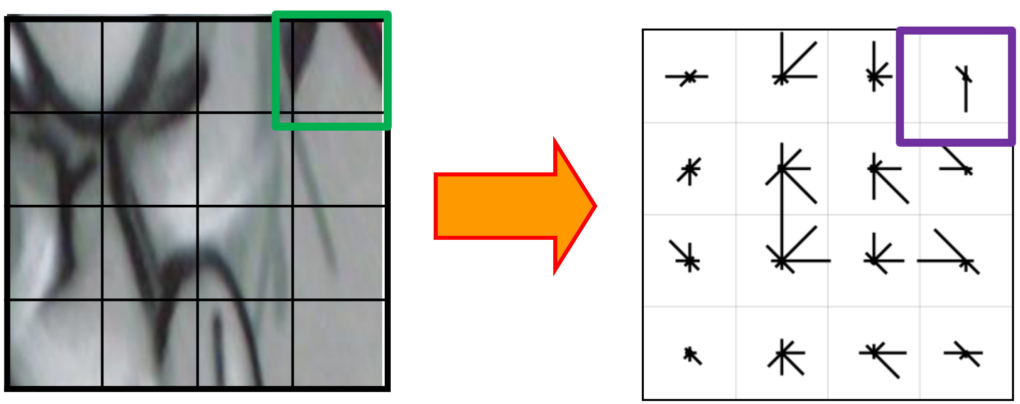
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

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.

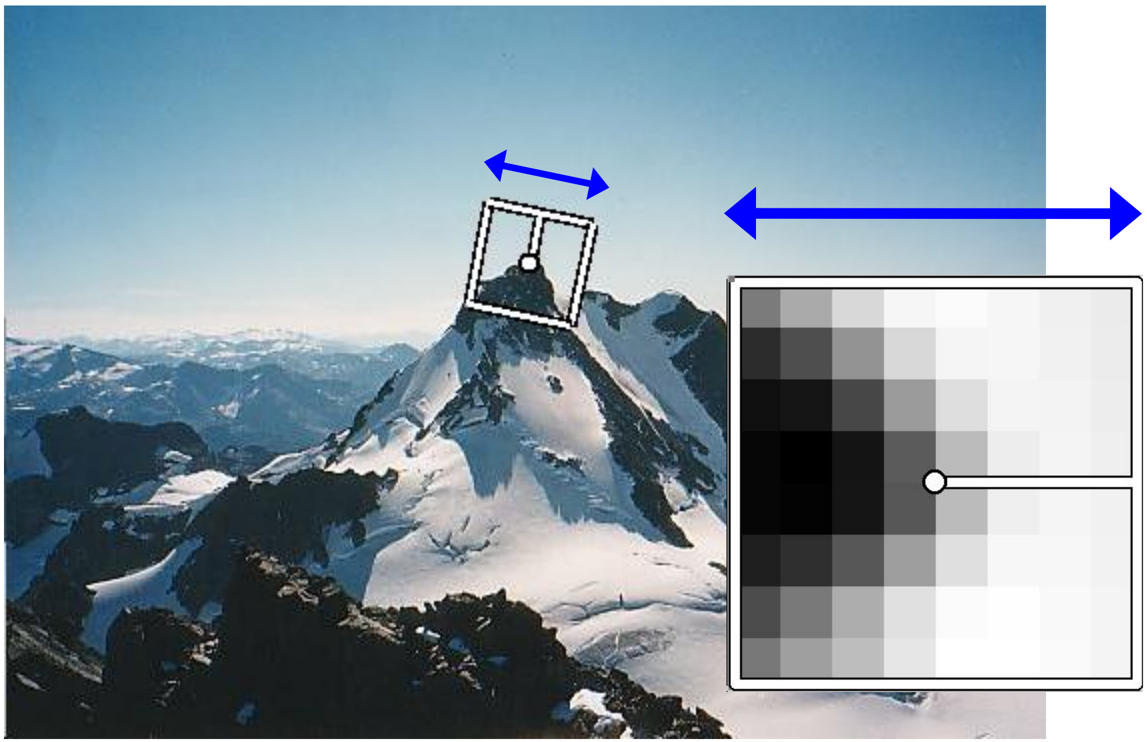




Why subpatches?

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.

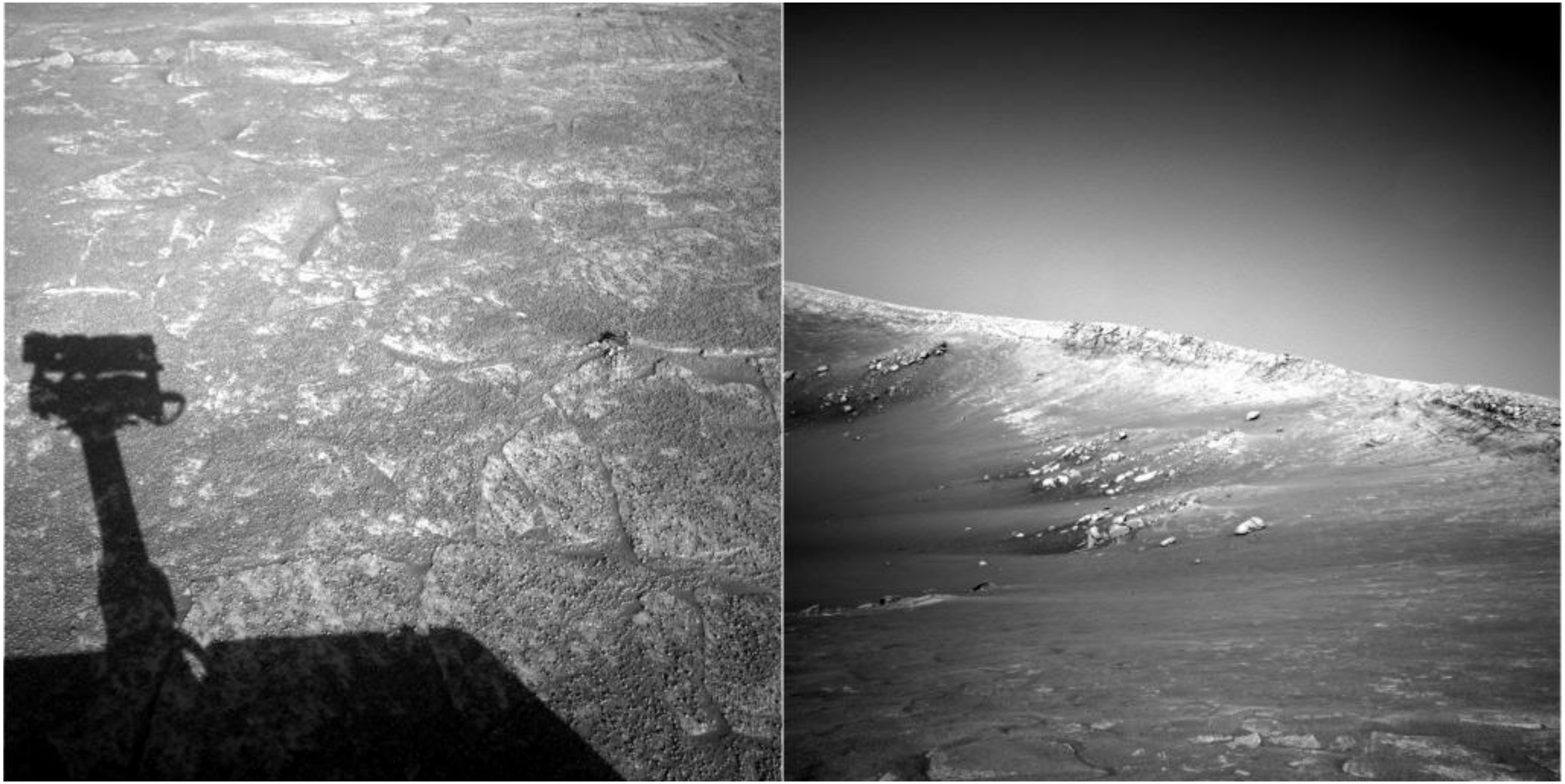
Image from Matthew Brown

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

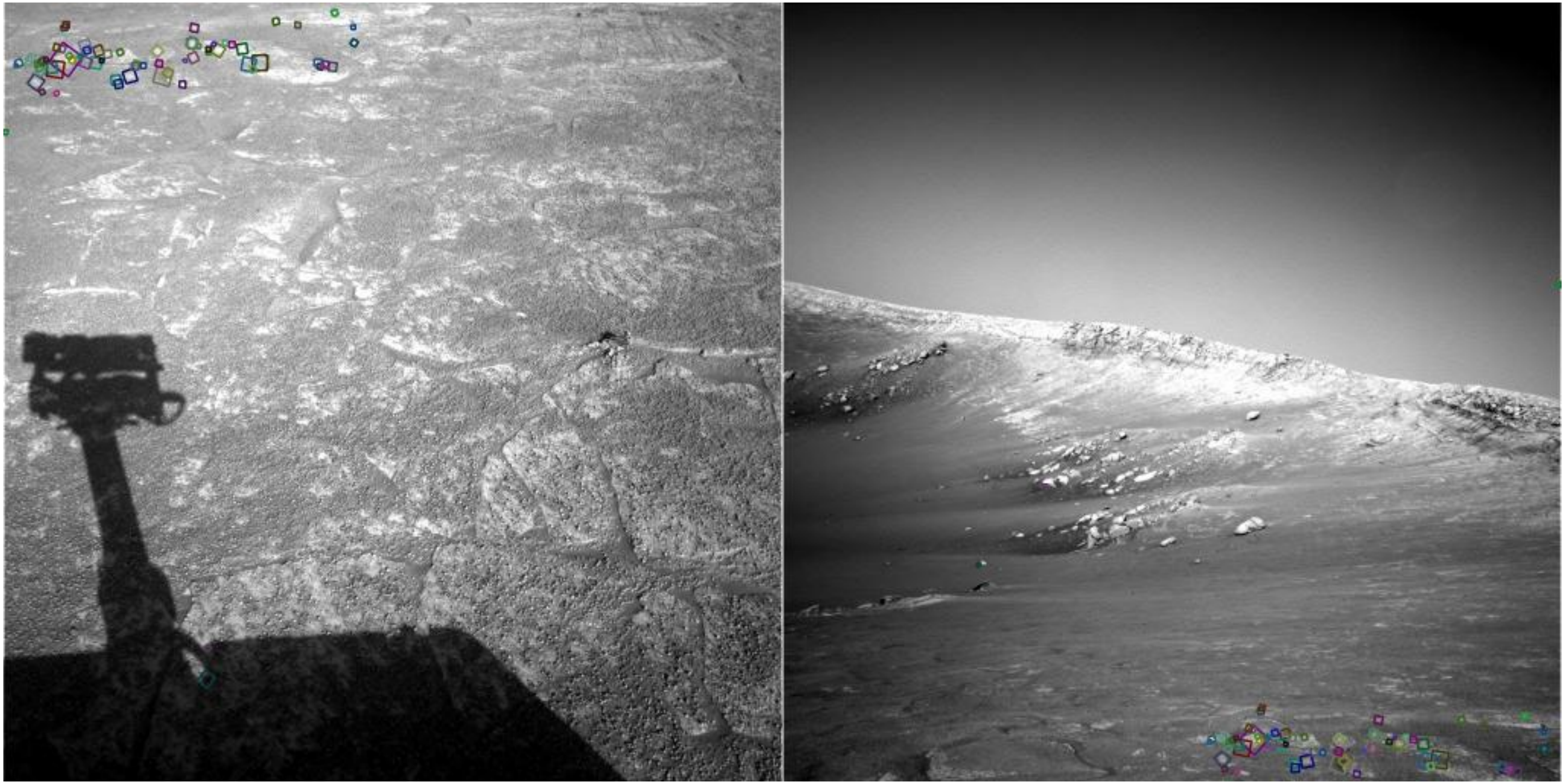
Steve Seitz

Example



NASA Mars Rover images

Example



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

SIFT properties

- Invariant to
  - Scale
  - Rotation

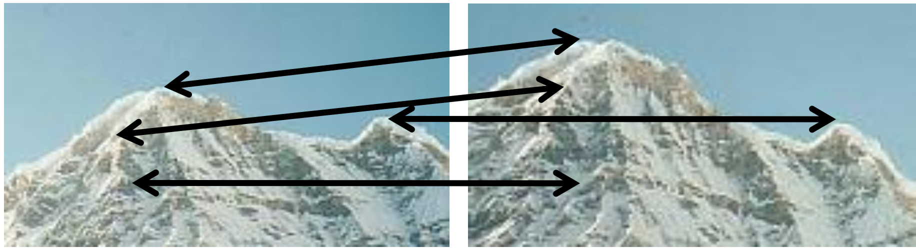
- Partially invariant to
  - Illumination changes
  - Camera viewpoint
  - Occlusion, clutter

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

Kristen Grauman

Matching local features

Image 1

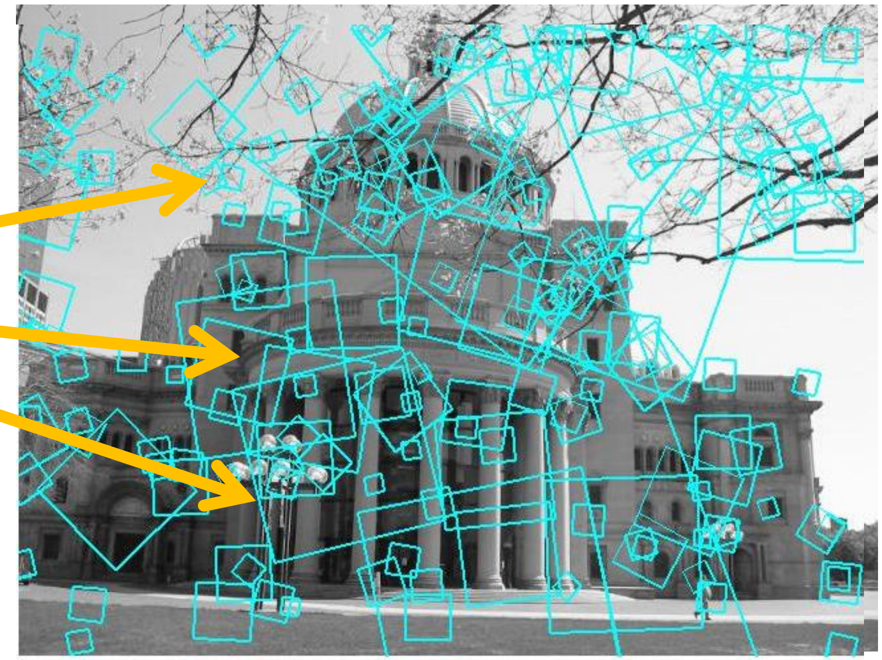


Image 2

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)

Kristen Grauman

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

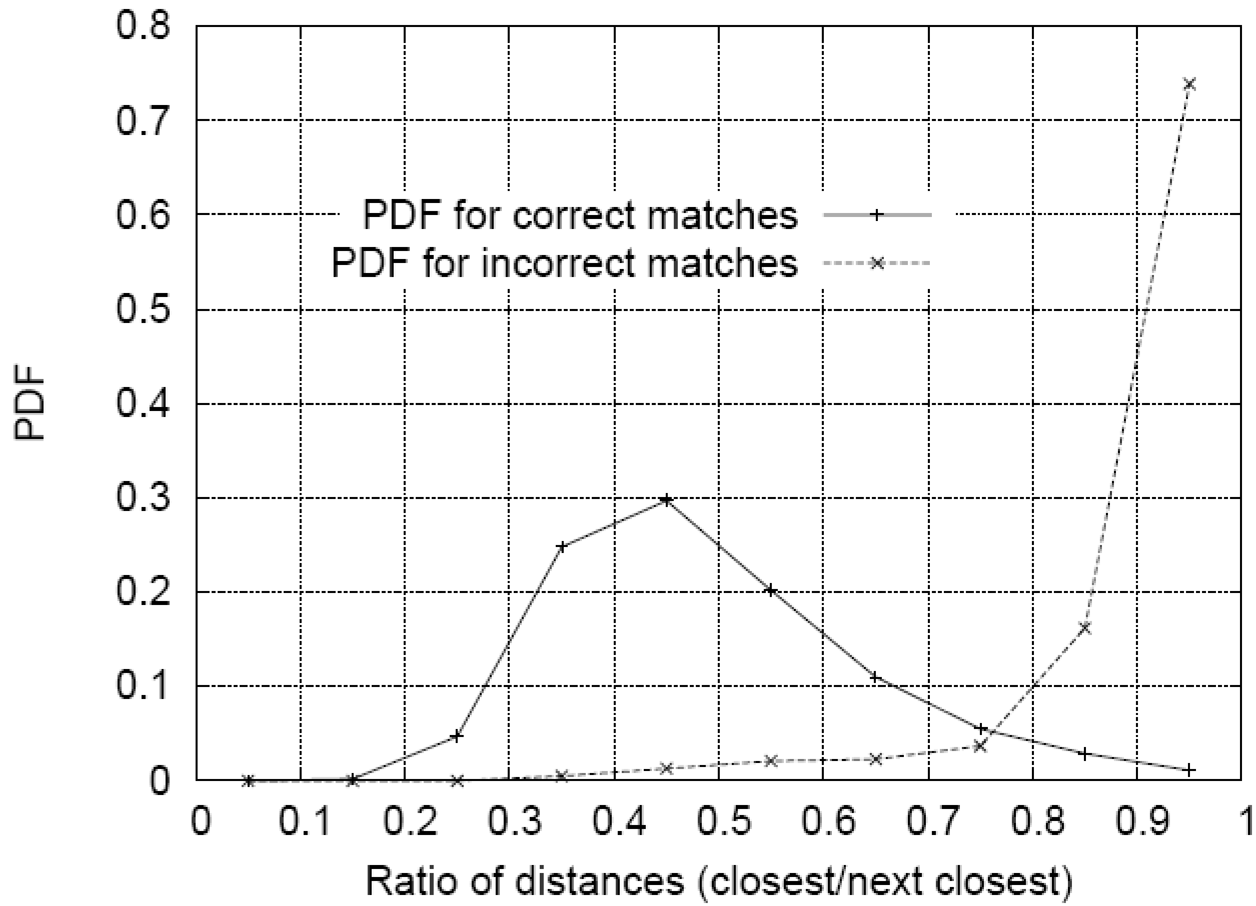
If low, first match looks good.

If high, could be ambiguous match.

Kristen Grauman

Matching SIFT Descriptors

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

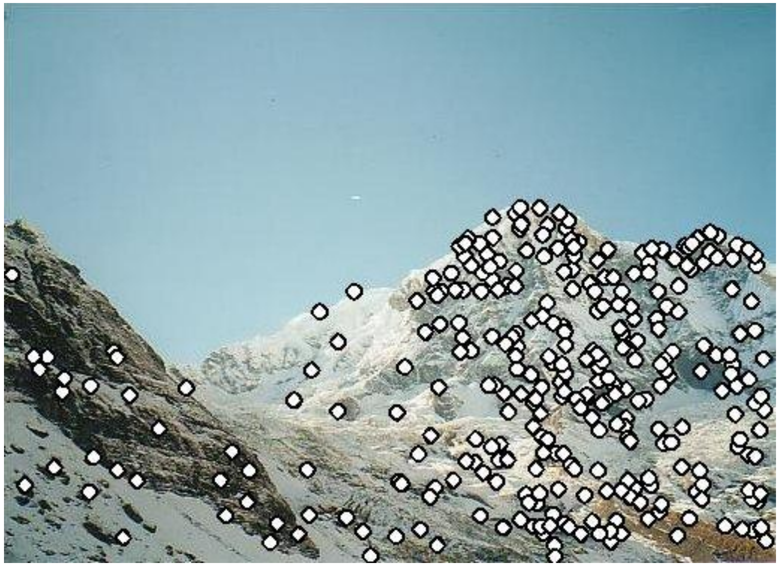


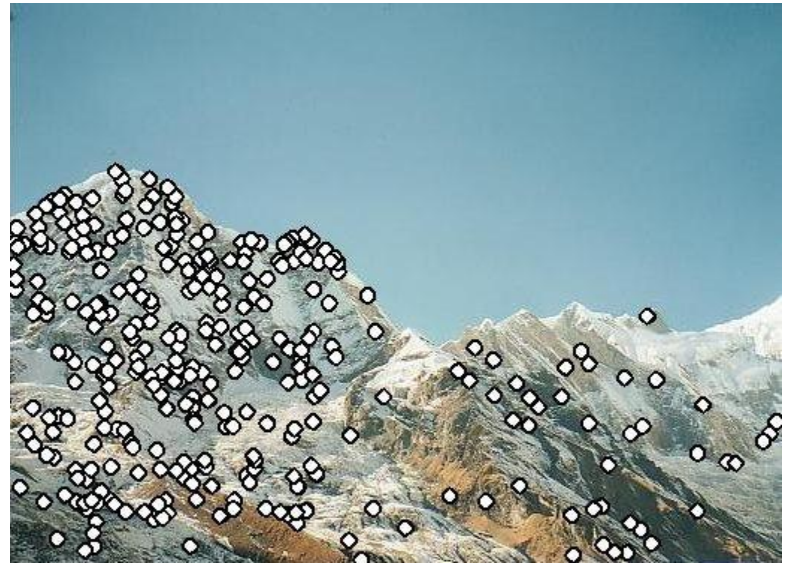
Lowe IJCV 2004

Recap: robust feature-based alignment

Source: L. Lazebnik

Recap: robust feature-based alignment





- Extract features

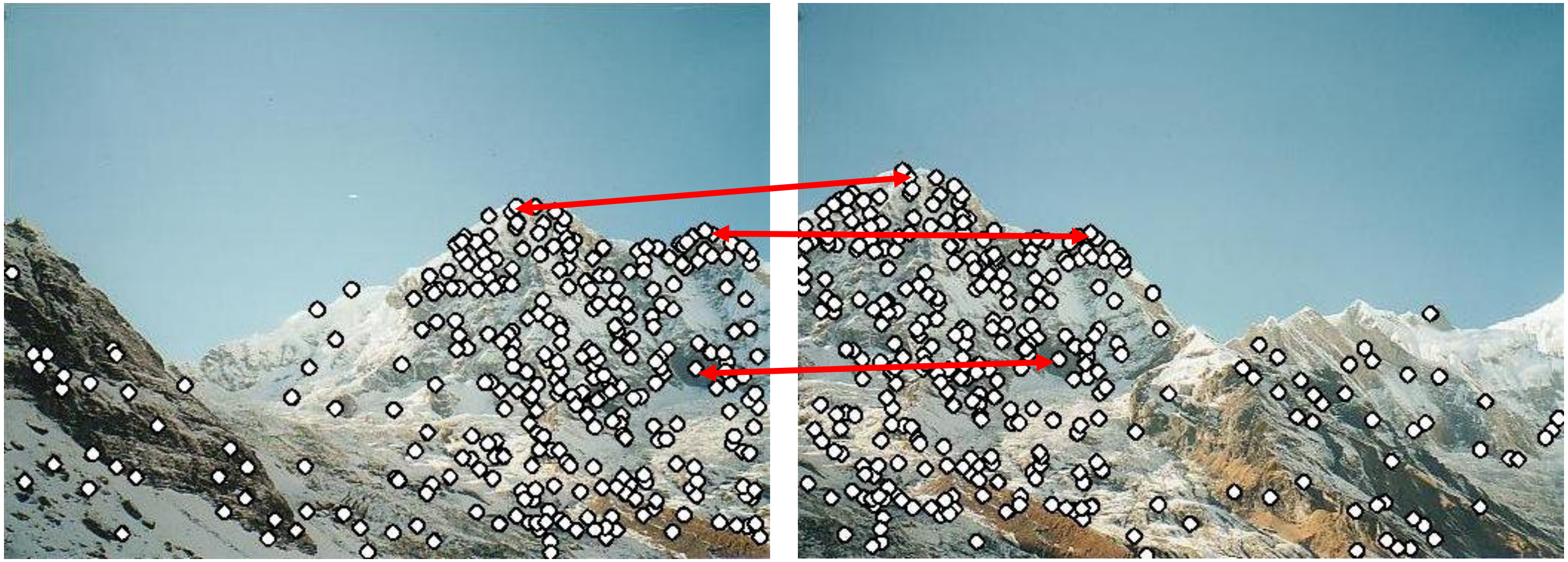
Source: L. Lazebnik

Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*

Source: L. Lazebnik

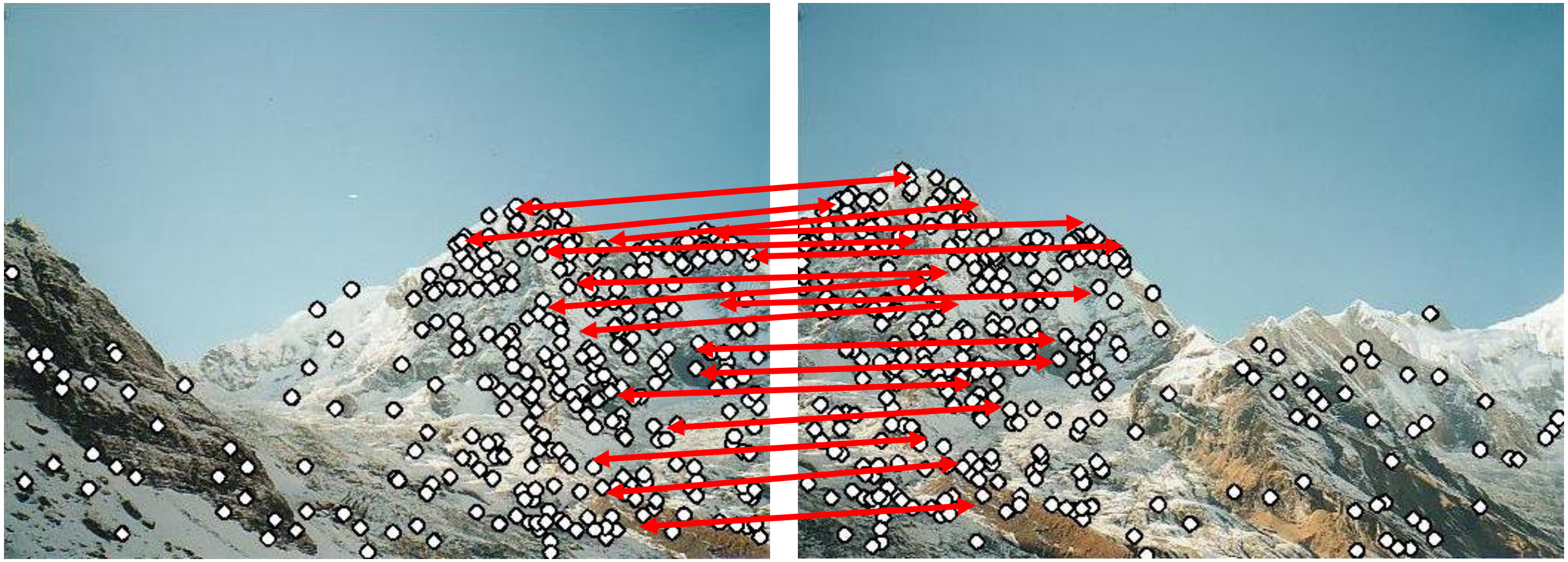
Recap: robust feature-based alignment



- 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

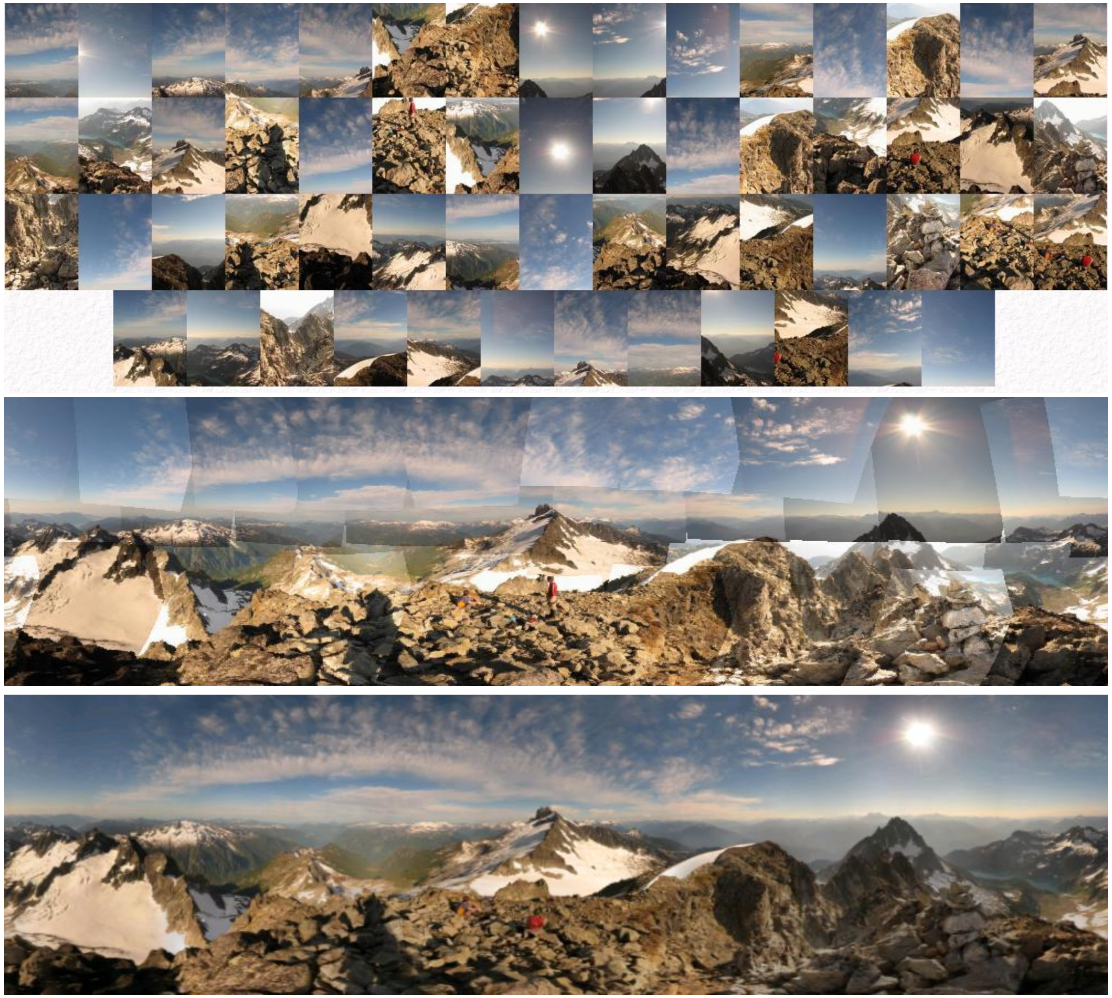
- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

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

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)