Local features: detection and description

Kristen Grauman
UT Austin

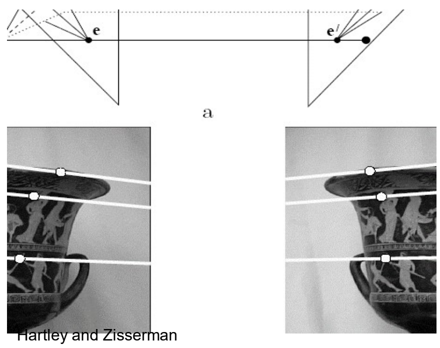
Tues Feb 27

Announcements

- Reminder: Slides posted on course webpage
- Midterm next Thursday Mar 9
  - Closed book
  - One 8.5x11” sheet of notes allowed

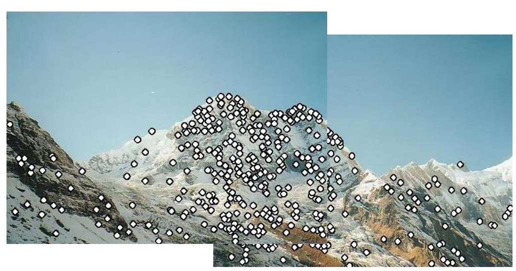
Multiple views

Matching, invariant features, stereo vision, instance recognition

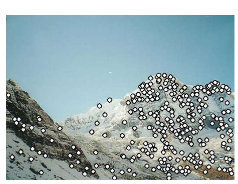


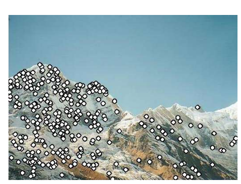
Slide credit: Kristen Grauman

Important tool for multiple views: Local features



Multi-view matching relies on **local feature** correspondences.





How to *detect* which local features to match?
How to *describe* the features we detect?

Review questions

- What properties should an interest operator have?
- What will determine how many interest points a given image has?
- What does it mean to have multiple local maxima at a single pixel during LoG scale space selection?

Outline

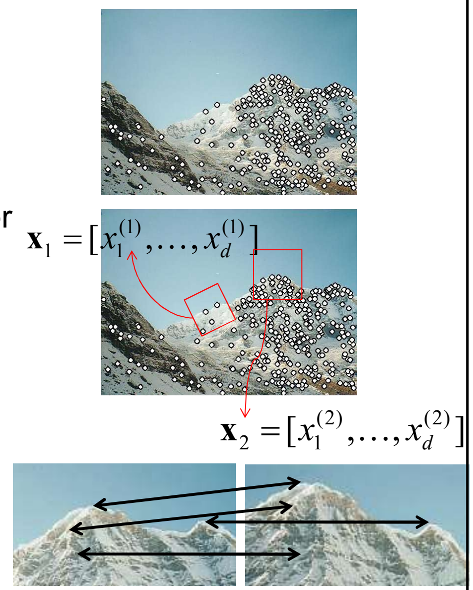
- **Last time**: Interest point detection
  - Harris corner detector
  - Laplacian of Gaussian, automatic scale selection
- **Today**: Local descriptors and matching
  - SIFT descriptors for image patches
  - Matching sets of features

Local features: main components

1) Detection: Identify the interest points

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

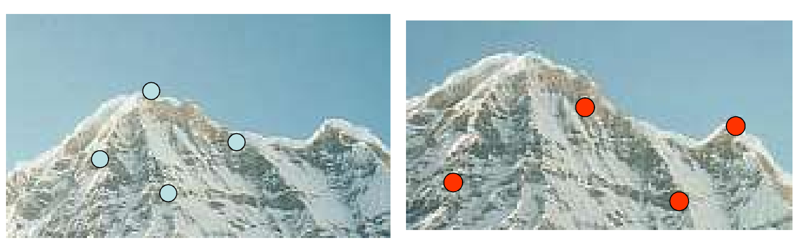
3) Matching: Determine correspondence between descriptors in two views



Slide credit: Kristen Grauman

Goal: interest operator repeatability

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

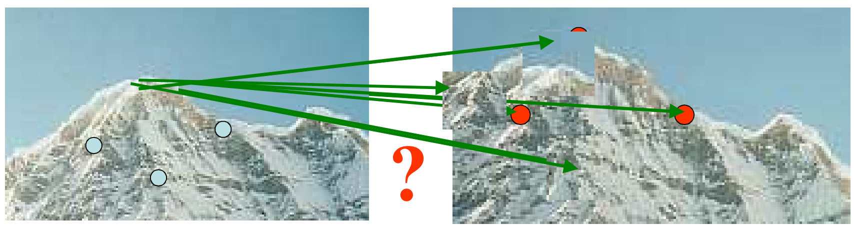


No chance to find true matches!

- 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.

Recall: Harris corner detector

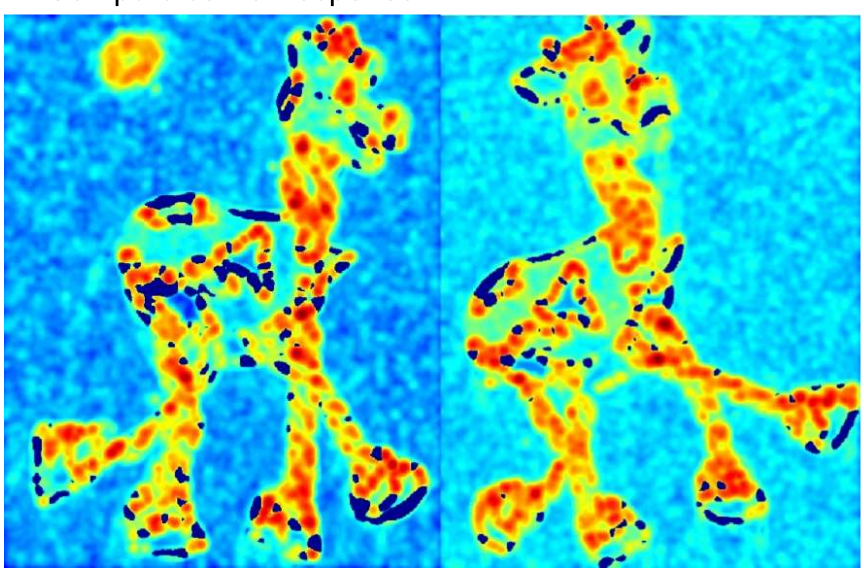
$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

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

Recall: Harris Detector: Steps

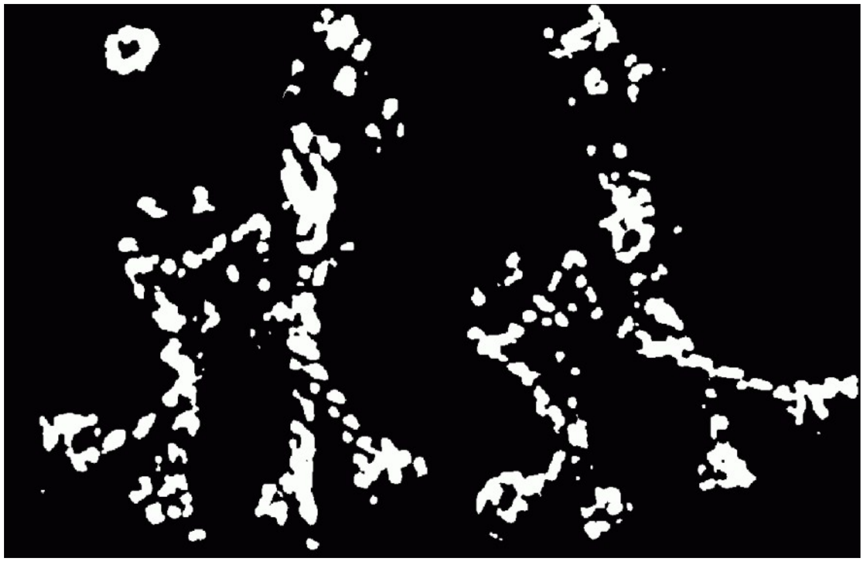
Recall: Harris Detector: Steps

Compute corner response f



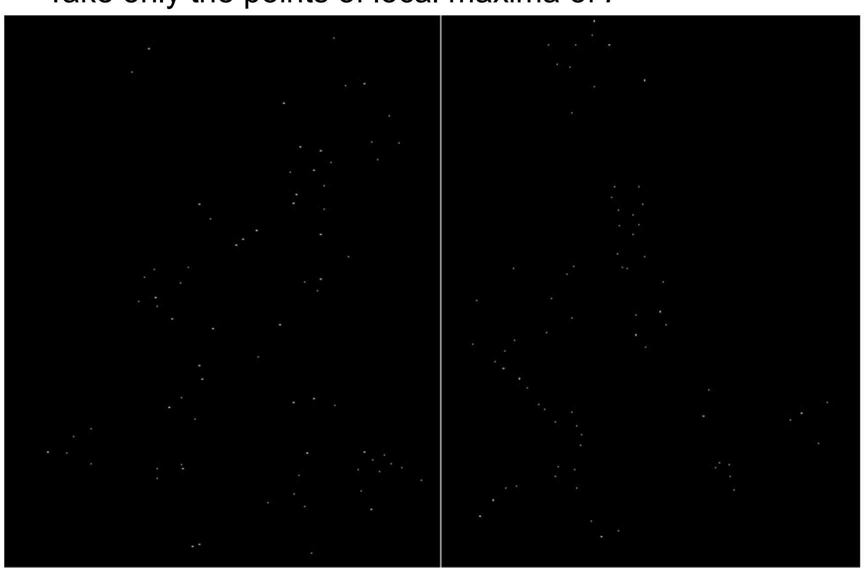
Recall: Harris Detector: Steps

Find points with large corner response: f > threshold

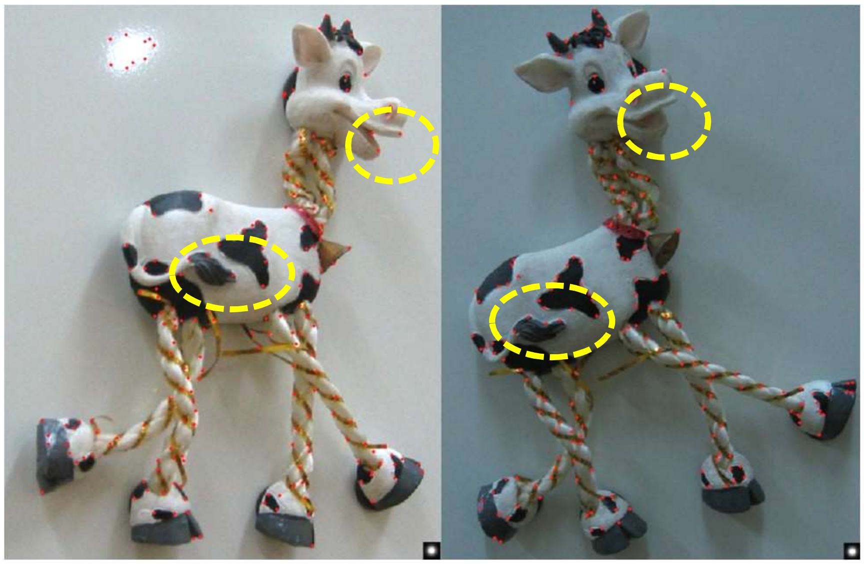


Recall: Harris Detector: Steps

Take only the points of local maxima of f



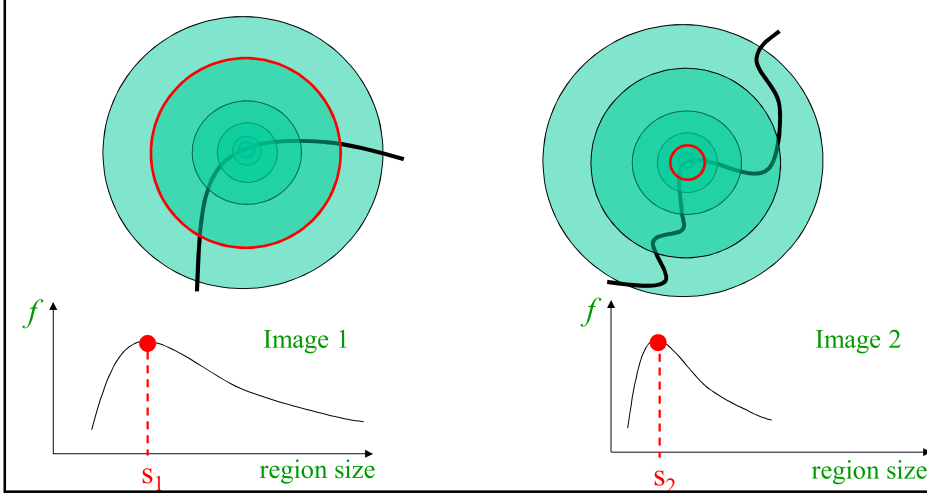
Recall: Harris Detector: Steps



Recall: Automatic scale selection

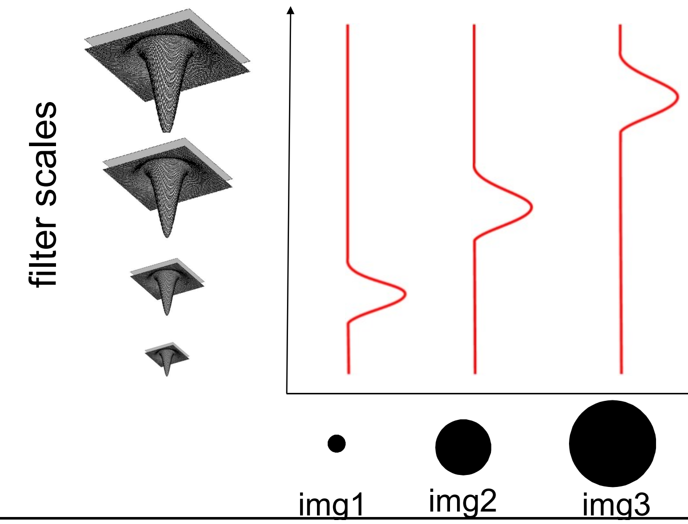
Intuition:

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



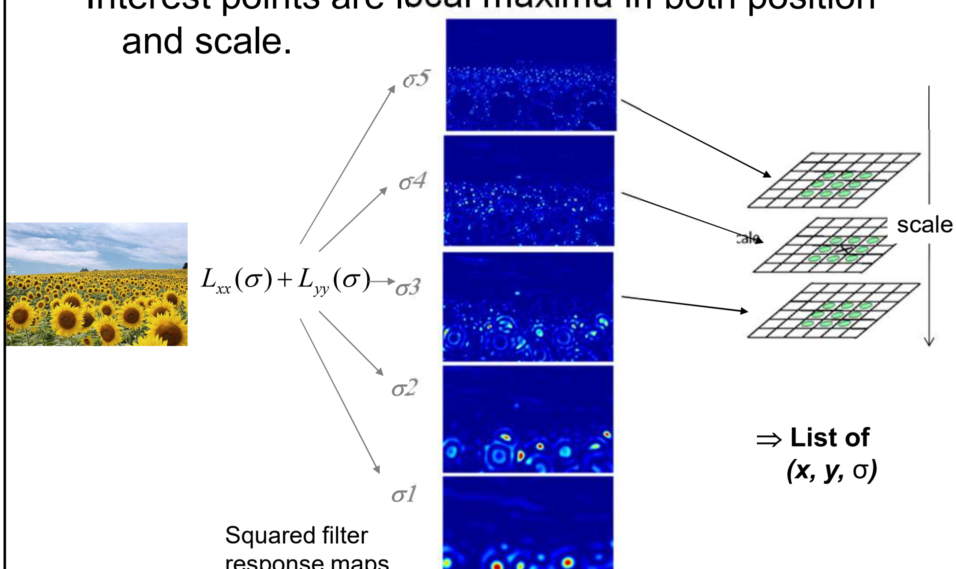
Recall: Blob detection for scale selection

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



Recall: Scale invariant interest points

Interest points are local maxima in both position and scale.

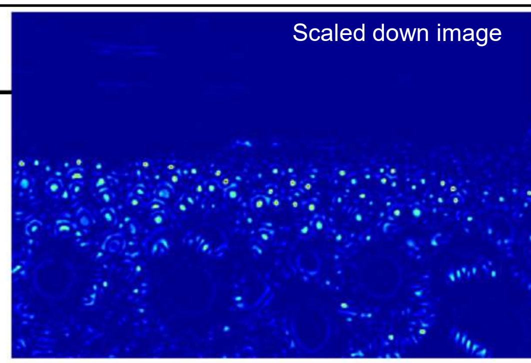


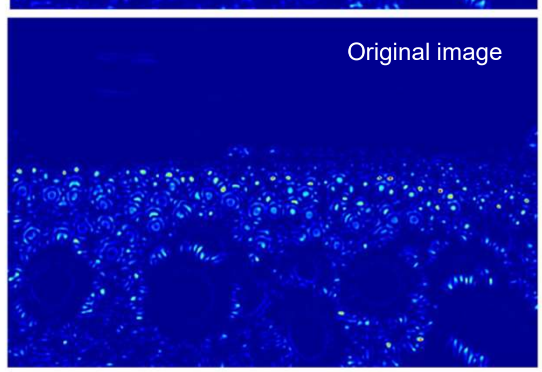
Example

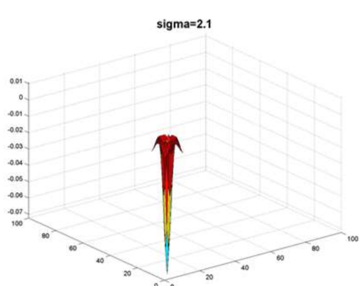
Original image
at ¾ the size

Slide credit: Kristen Grauman

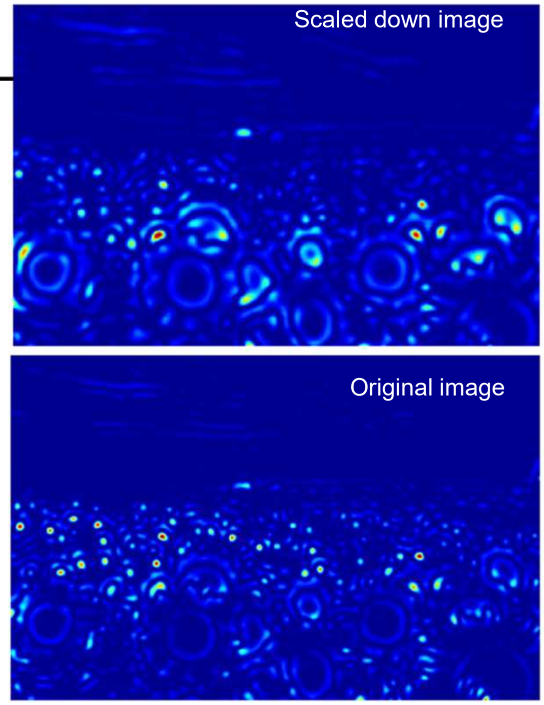
Original image
at ¾ the size

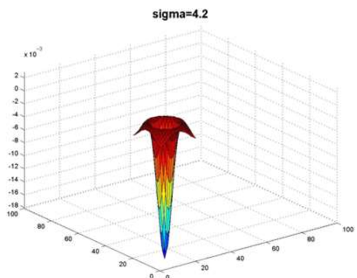




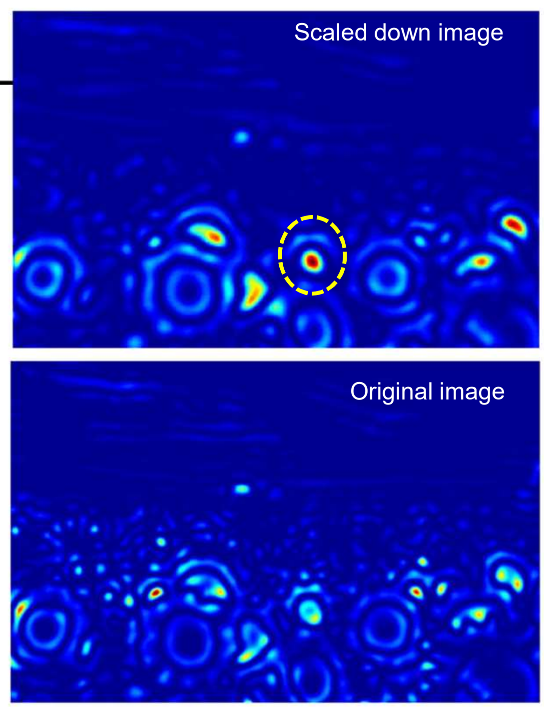


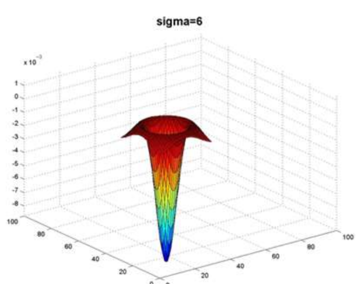
Slide credit: Kristen Grauman



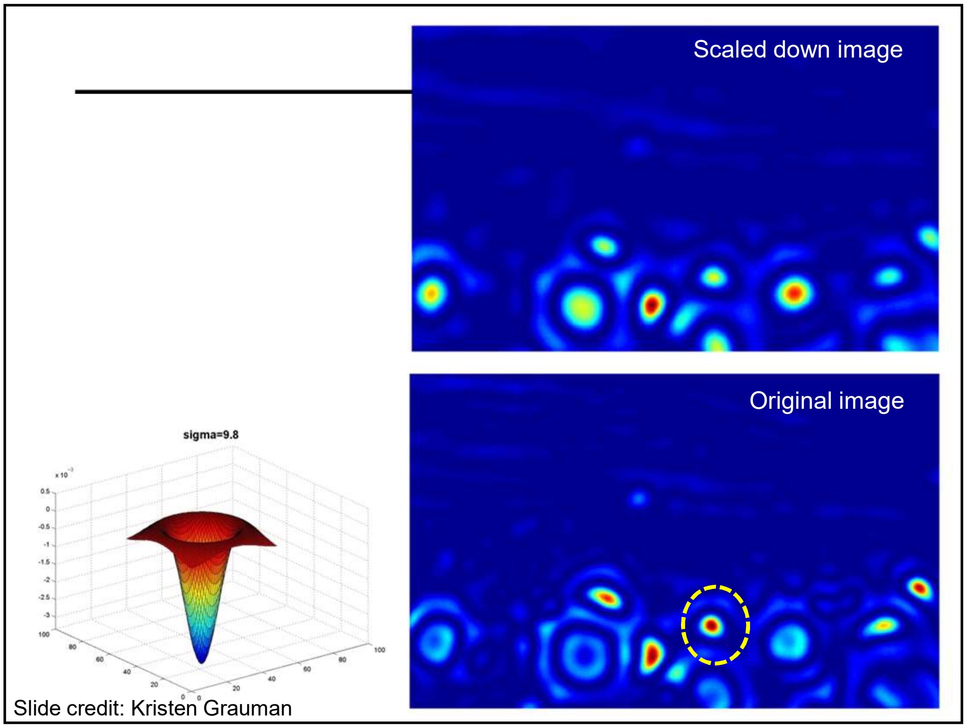


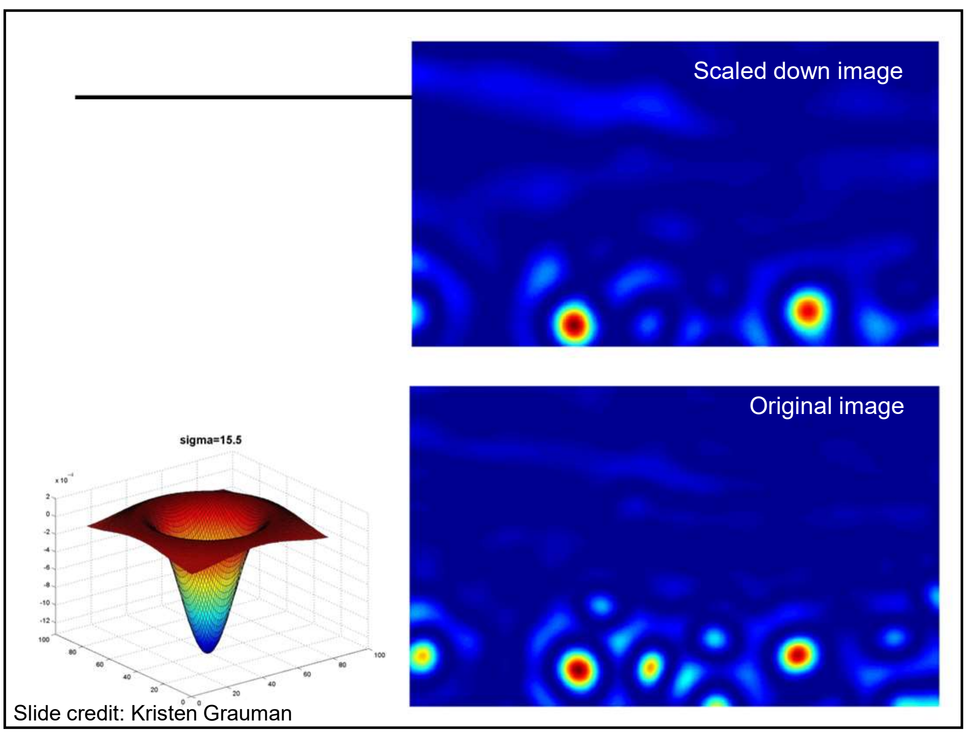
Slide credit: Kristen Grauman

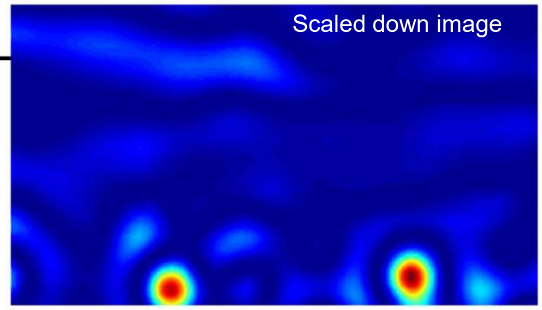


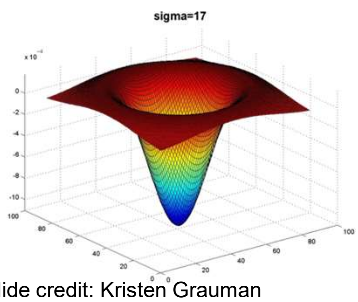


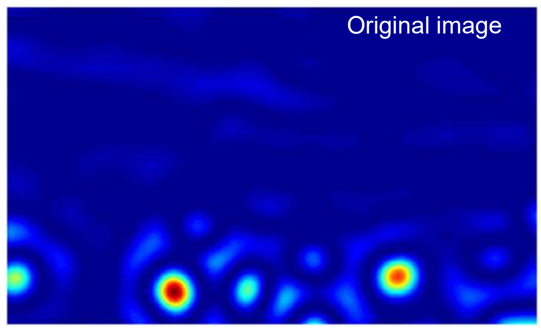
Slide credit: Kristen Grauman


Scaled down image
Original image
sigma=9.8
Slide credit: Kristen Grauman


Scaled down image
Original image
sigma=15.5
Slide credit: Kristen Grauman







Slide credit: Kristen Grauman

Scale-space blob detector: Example

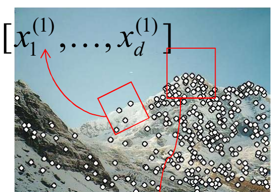
Image credit: Lana Lazebnik

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

Slide credit: Kristen Grauman

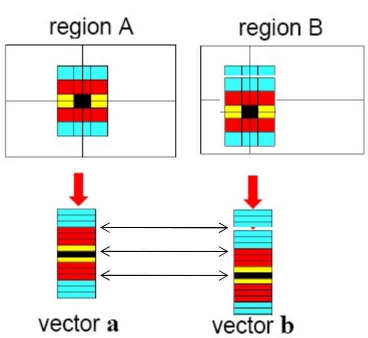
Geometric transformations

e.g. scale, translation, rotation

Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Raw patches as local descriptors



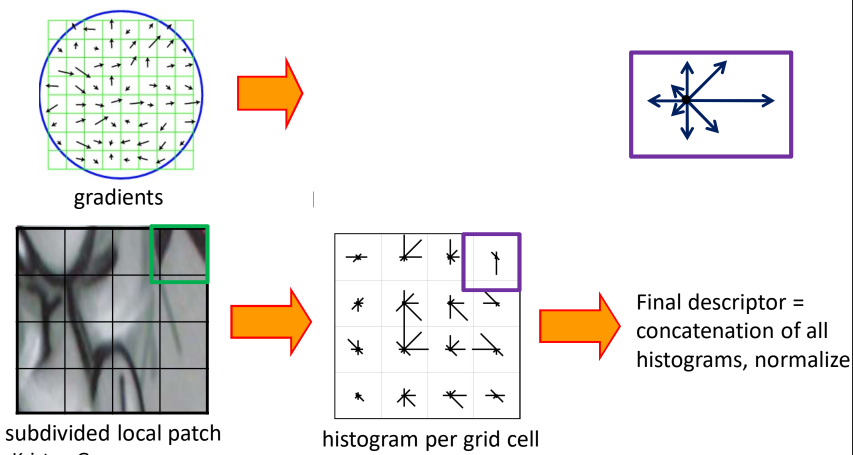
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

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

Figure: Andrew Zisserman

Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]

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



Slide credit: Kristen Grauman

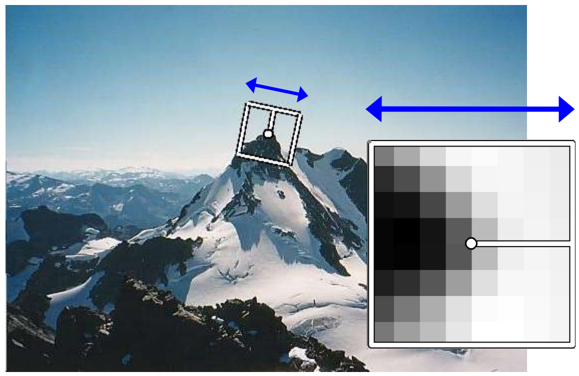
Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]

Interest points and their scales and orientations (random subset of 50)

SIFT descriptors

Slide credit: Kristen Grauman http://www.vlfeat.org/overview/sift.html

Making descriptor rotation invariant

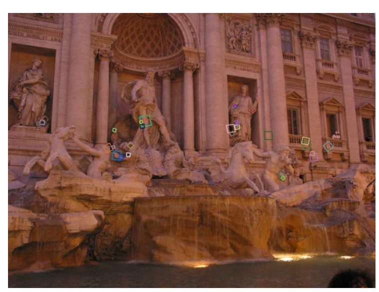


- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

Image from Matthew Brown

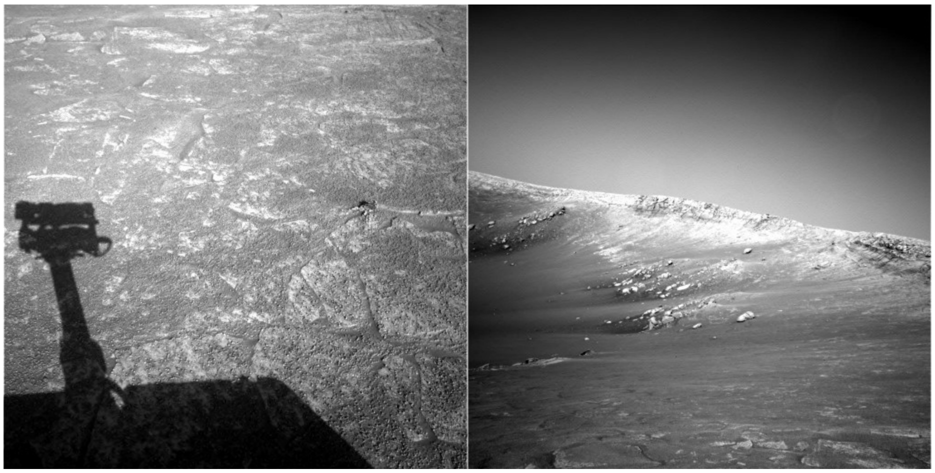
SIFT descriptor [Lowe 2004]

- 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, e.g. `http://www.vlfeat.org/overview/sift.html`



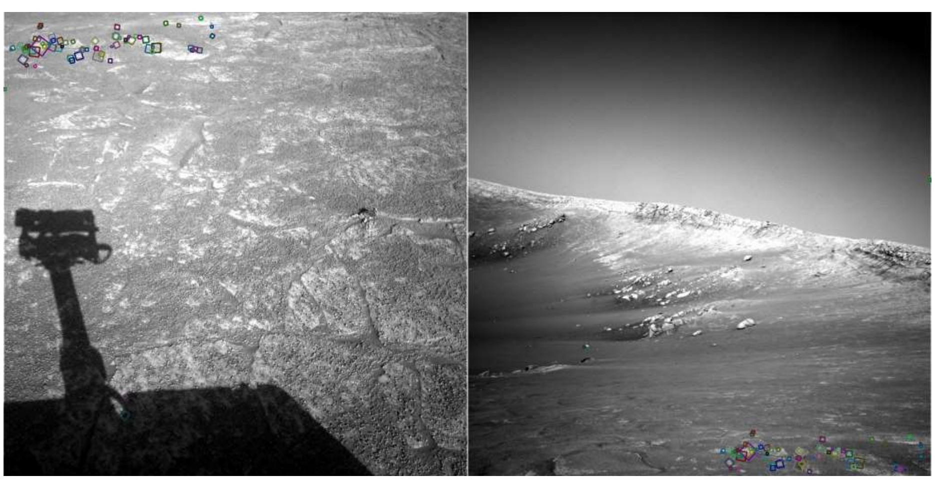
Slide credit: Steve Seitz

Example



NASA Mars Rover images

Example



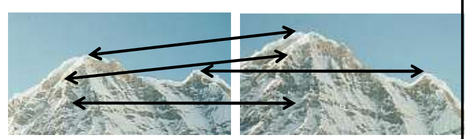
NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

Local features: main components

1) Detection: Identify the interest points

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

3) Matching: Determine correspondence between descriptors in two views



Matching local features

Slide credit: Kristen Grauman

Matching local features

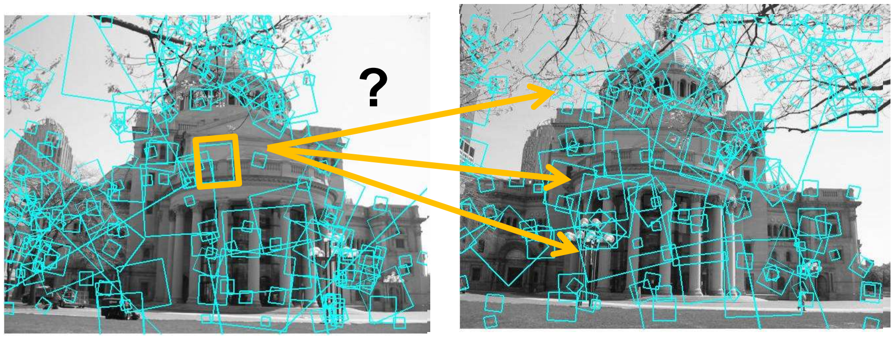


Image 1 Image 2

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)

Slide credit: Kristen Grauman

Ambiguous matches

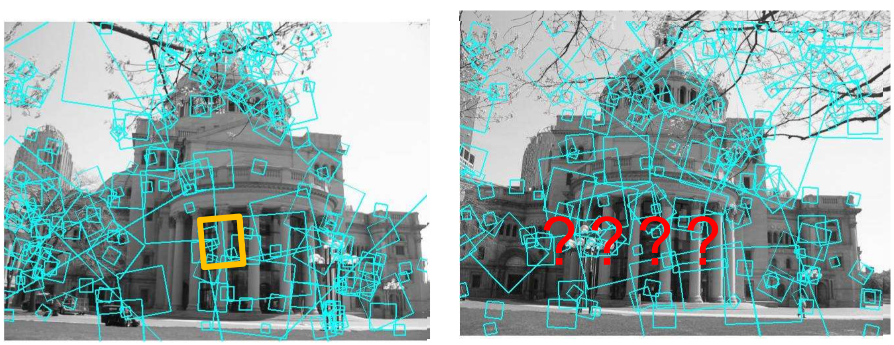


Image 1 Image 2

At what SSD value do we have a good match?

To add robustness to matching, consider **ratio** :

dist to best match / dist to second best match

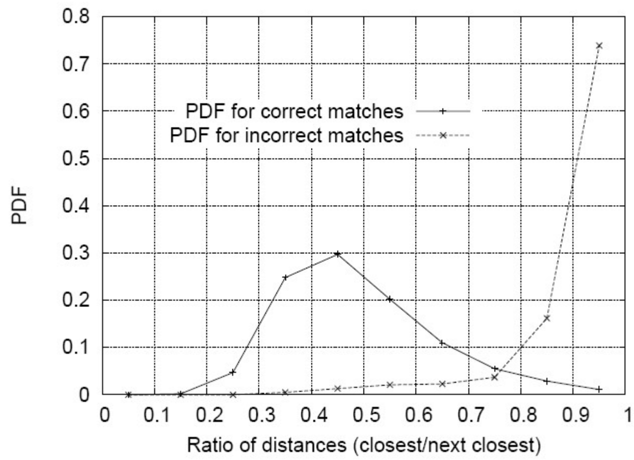
If low, first match looks good.

If high, could be ambiguous match.

Slide credit: Kristen Grauman

Matching SIFT Descriptors

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



Lowe IJCV 2004

Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]

Interest points and their scales and orientations (random subset of 50)

SIFT descriptors

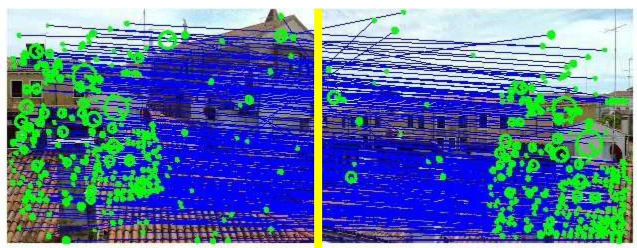
Slide credit: Kristen Grauman http://www.vlfeat.org/overview/sift.html

SIFT (preliminary) matches

img1

img2

img1



img2

Slide credit: Kristen Grauman http://www.vlfeat.org/overview/sift.html

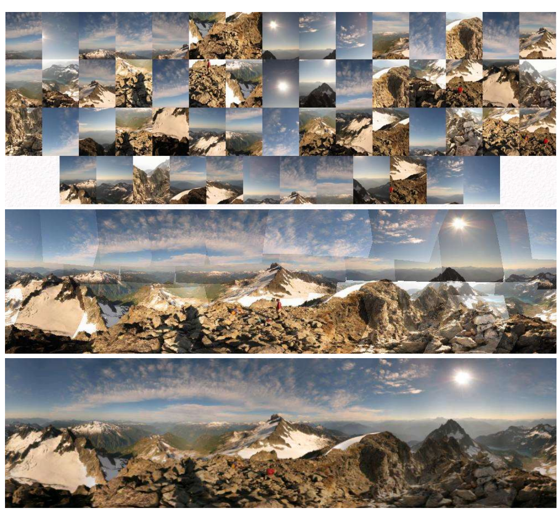
Value of local (invariant) features

- Complexity reduction via selection of distinctive points
- Describe images, objects, parts without requiring segmentation
  – Local character means robustness to clutter, occlusion
- Robustness: similar descriptors in spite of noise, blur, etc.

Applications of local invariant features

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

Automatic mosaicing

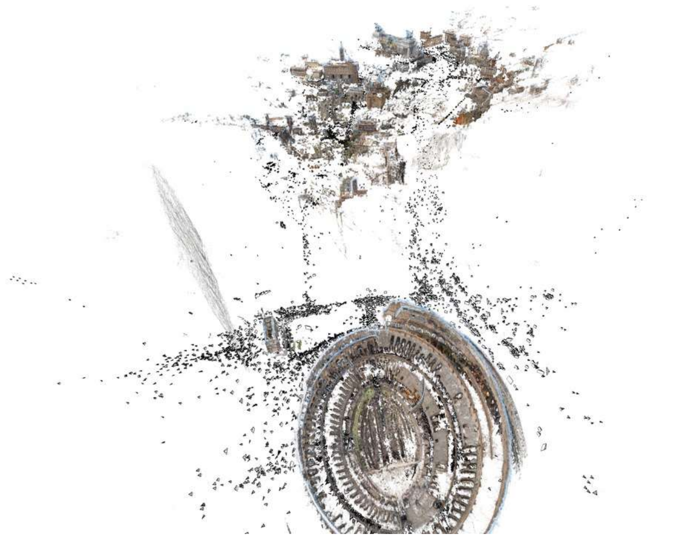


Matthew Brown
http://matthewalunbrown.com/autostitch/autostitch.html

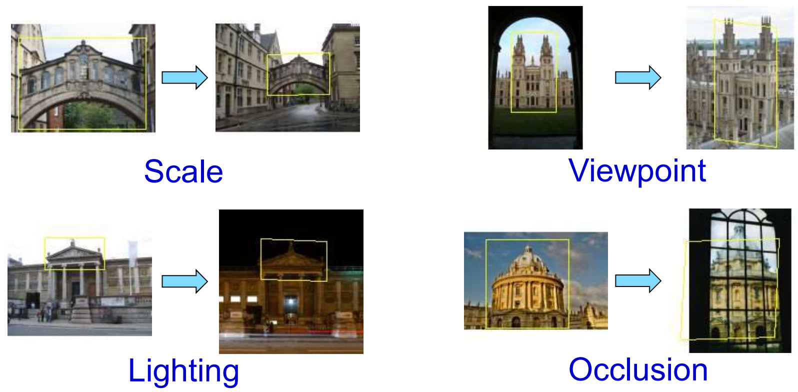
Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Photo tourism [Snavely et al.]



Recognition of specific objects, scenes



Slide credit: J. Sivic

Google Goggles

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)

Coming up

Additional questions we need to address to achieve these applications:

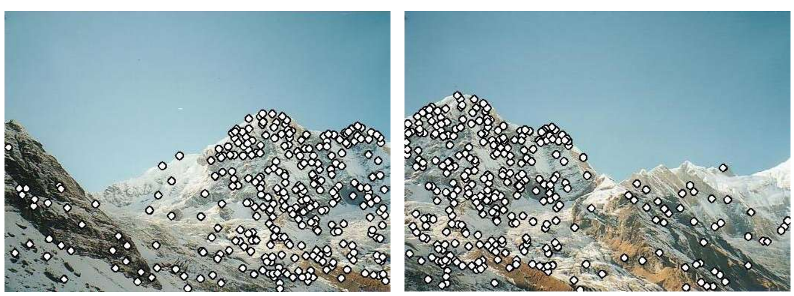
- Fitting a parametric transformation given putative matches
- Dealing with outlier correspondences
- Exploiting geometry to restrict locations of possible matches
- Triangulation, reconstruction
- Efficiency when indexing so many keypoints

Slide credit: Kristen Grauman

Coming up: robust feature-based alignment

Source: L. Lazebnik

Coming up: robust feature-based alignment



- Extract features

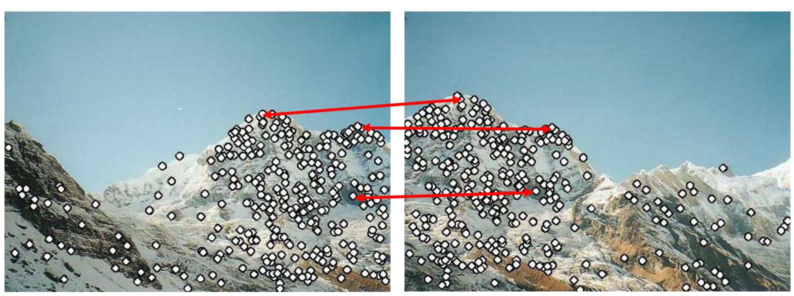
Source: L. Lazebnik

Coming up: robust feature-based alignment

- Extract features
- Compute *putative matches*

Source: L. Lazebnik

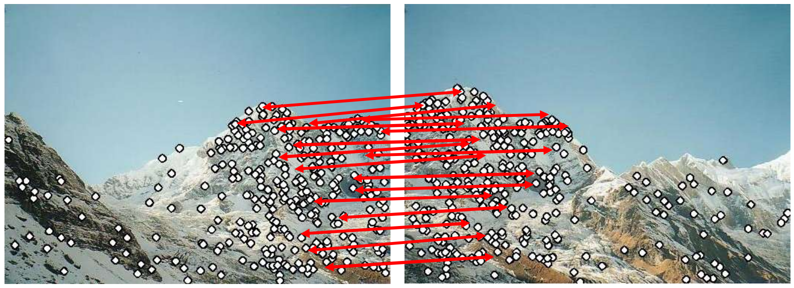
Coming up: robust feature-based alignment



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

Source: L. Lazebnik

Coming up: 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

Coming up: 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