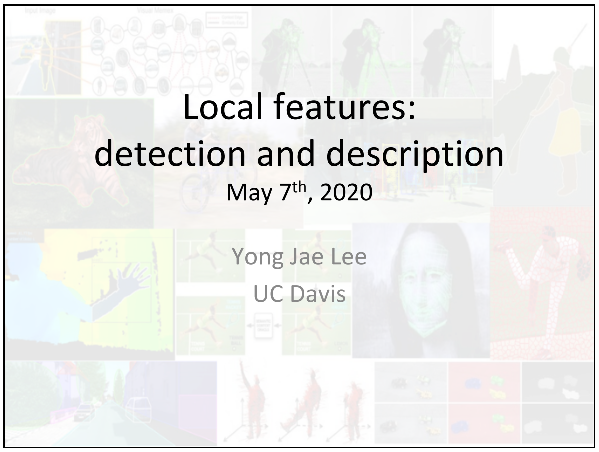


# Local features: detection and description

May 7th, 2020

Yong Jae Lee
UC Davis

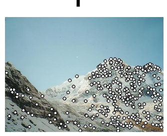
## Today

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

2

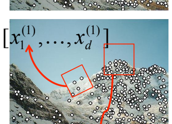
## Local features: main components

1) Detection: Identify the interest points



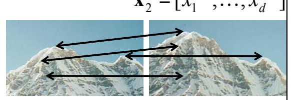
2) Description: Extract vector feature descriptor surrounding each interest point.

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



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

3) Matching: Determine correspondence between descriptors in two views

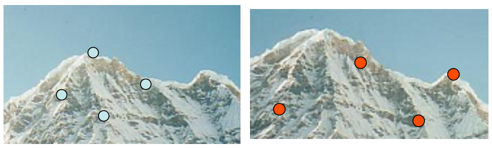


Kristen Grauman

3

## Goal: interest operator repeatability

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



No chance to find true matches!

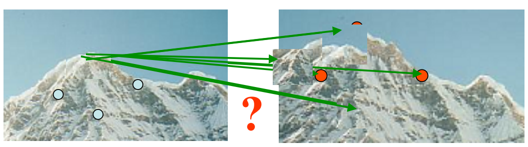
- Yet we have to be able to run the detection procedure *independently* per image.

4

Kristen Grauman

## Goal: descriptor distinctiveness

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



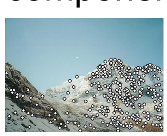

- Must provide some invariance to geometric and photometric differences between the two views.

5

Kristen Grauman

## 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

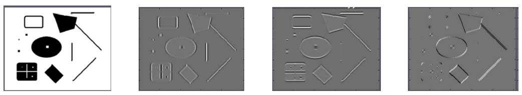
6

Kristen Grauman

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

$$M = \sum \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

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



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

Kristen Grauman

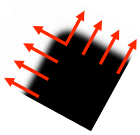
7

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

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

(Eigenvalue decomposition)

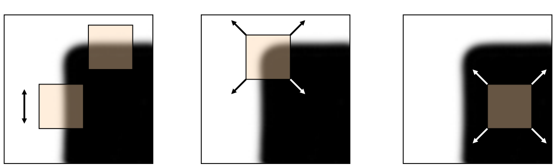
$$Mx_i = \lambda_i x_i$$



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

8

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



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

"corner": $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$;

"flat" region: $\lambda_1$ and $\lambda_2$ are small;

One way to score the cornerness: $$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

9

## 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

10

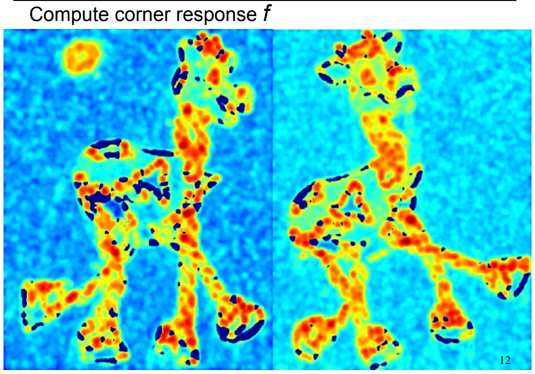
Kristen Grauman

## Harris Detector: Steps

11

## Harris Detector: Steps

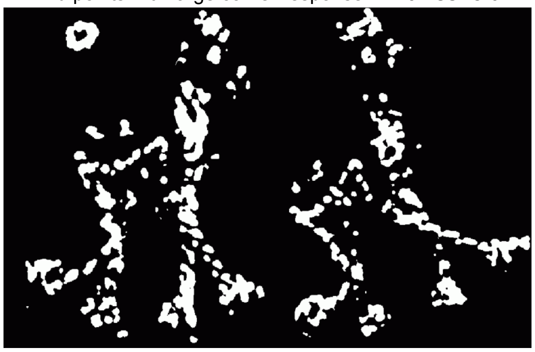
Compute corner response $f$



12

## Harris Detector: Steps

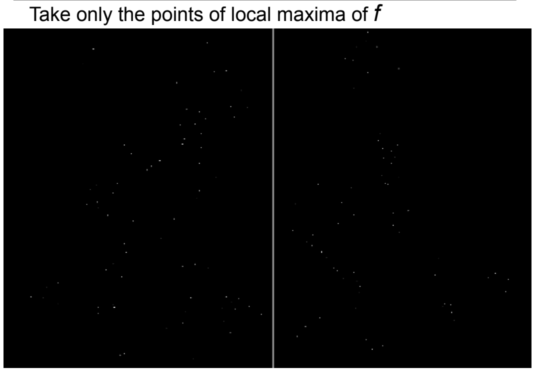
Find points with large corner response: $f$ > threshold



13

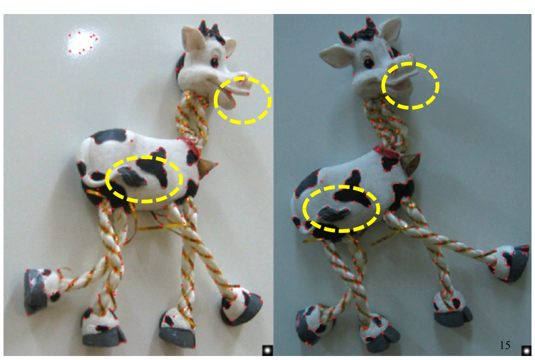
## Harris Detector: Steps

Take only the points of local maxima of $f$



14

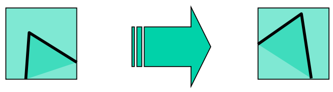
## Harris Detector: Steps



15

## Properties of the Harris corner detector

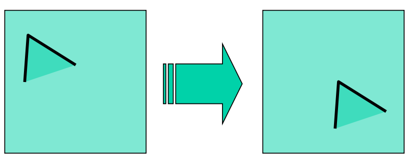
Rotation invariant? Yes





## Properties of the Harris corner detector

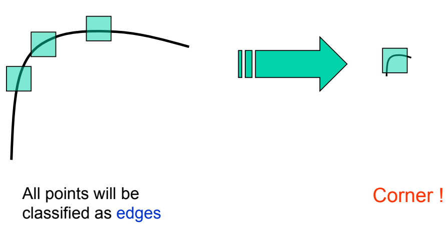
Rotation invariant? Yes
Translation invariant? Yes





## Properties of the Harris corner detector

Rotation invariant? Yes
Translation invariant? Yes
Scale invariant? No



## Scale invariant interest points

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

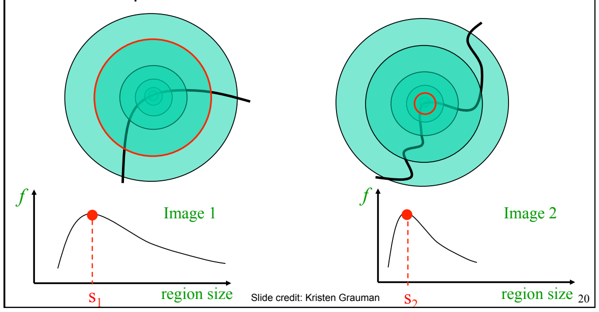
19

Kristen Grauman

## Automatic scale selection

**Intuition:**

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




Slide credit: Kristen Grauman

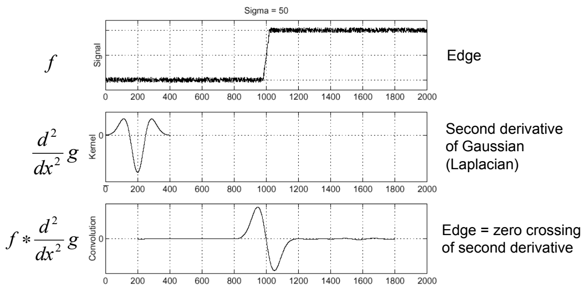
20

## What can be the "signature" function?

21

Kristen Grauman

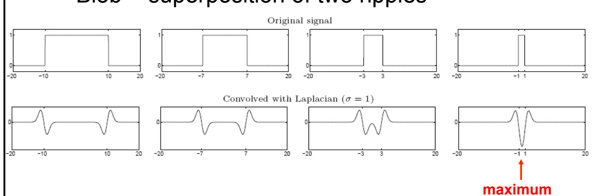
## Recall: Edge detection




22
Source: S. Seitz

## From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

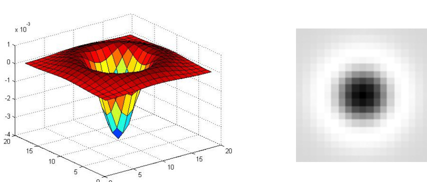



**Spatial selection**: the **magnitude** of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

23
Slide credit: Lana Lazebnik

## Blob detection in 2D

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

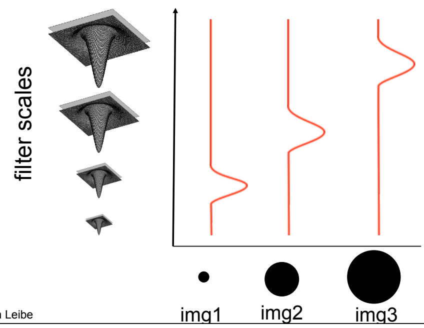


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

24

## Blob detection in 2D: scale selection

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

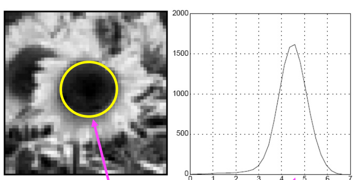



25

Bastian Leibe

## Blob detection in 2D

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




characteristic scale

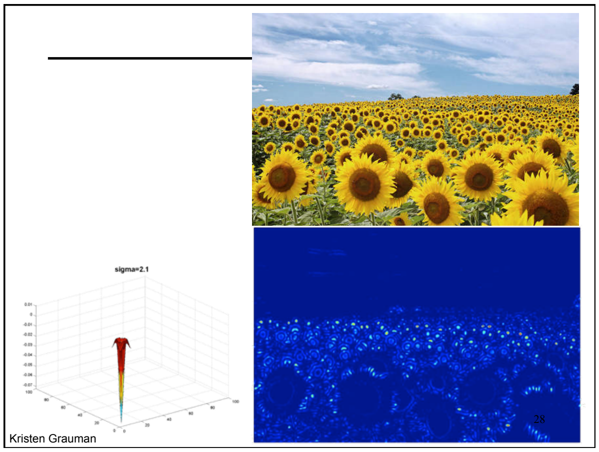
26

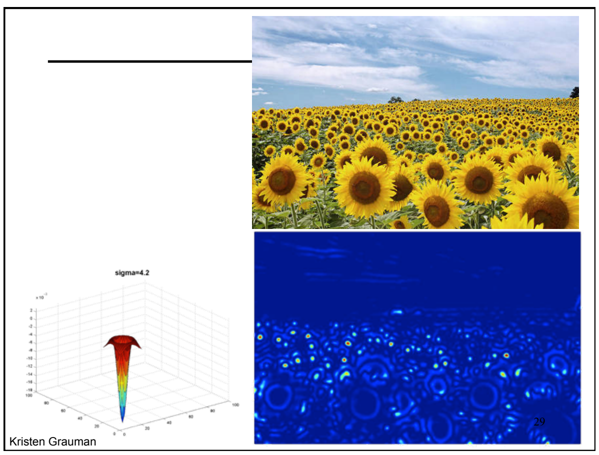
Slide credit: Lana Lazebnik

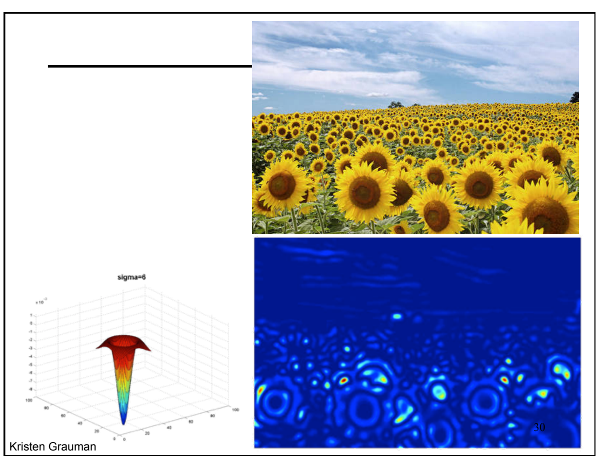
## Example

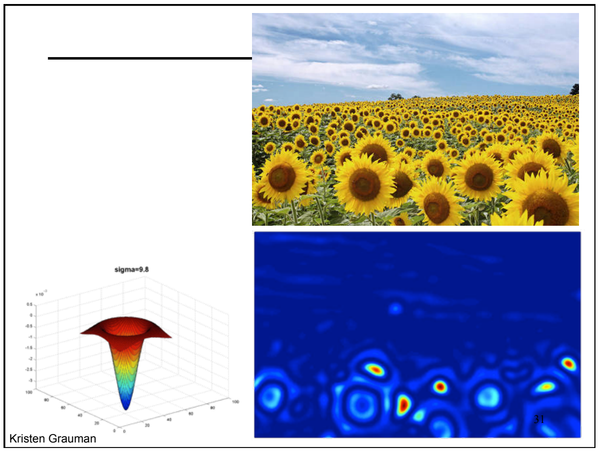
Kristen Grauman

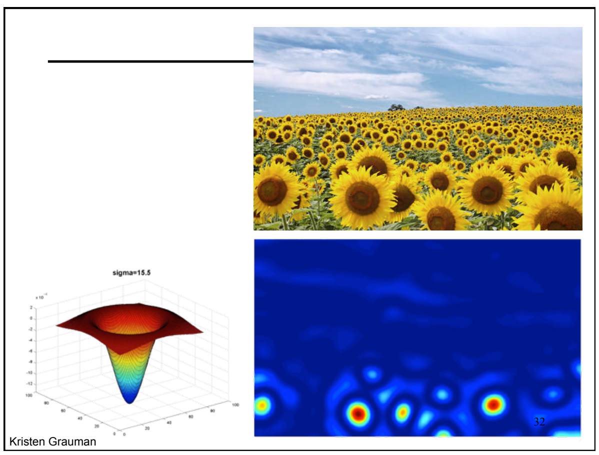
27

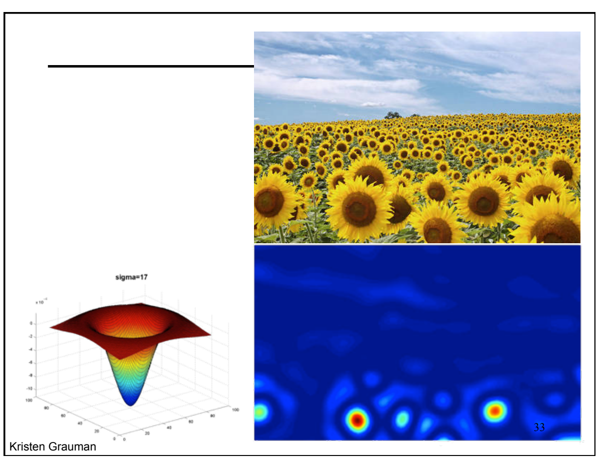

sigma=2.1
Kristen Grauman
28


sigma=4.2
Kristen Grauman
29


sigma=6
Kristen Grauman
30

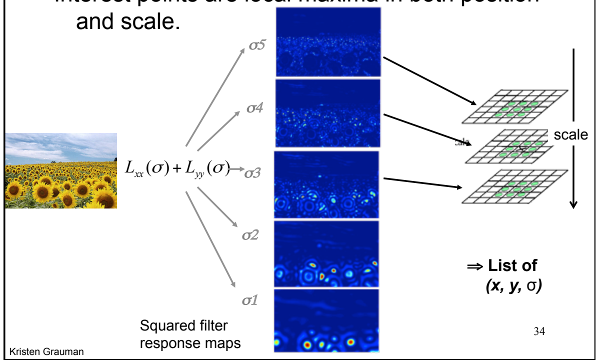

sigma=9.8
Kristen Grauman
31


sigma=15.5
Kristen Grauman
32


sigma=17
Kristen Grauman
33

## Scale invariant interest points

Interest points are local maxima in both position and scale.




Kristen Grauman

34

## Scale-space blob detector: Example

35

Image credit: Lana Lazebnik

## Technical detail

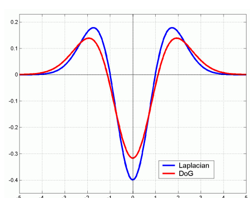
We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

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

(Laplacian)

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

(Difference of Gaussians)




36

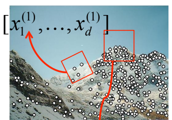
Slide credit: Kristen Grauman

## Local features: main components

1) Detection: Identify the interest points

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

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



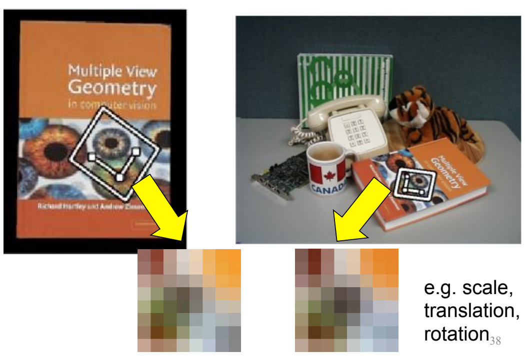
$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$

3) Matching: Determine correspondence between descriptors in two views

37

Slide credit: Kristen Grauman

## Geometric transformations

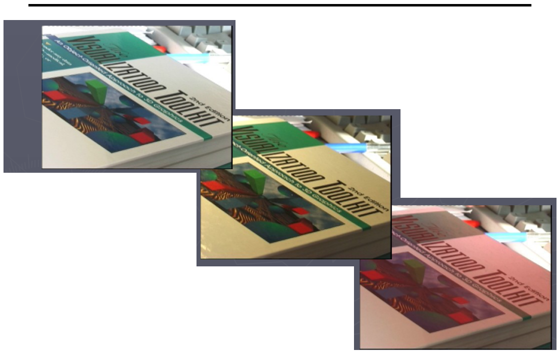



e.g. scale, translation, rotation

38

Slide credit: Kristen Grauman

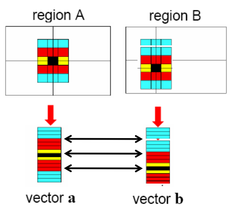
## Photometric transformations



39

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.

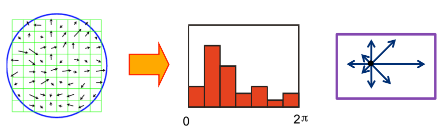
But this is very sensitive to even small shifts, rotations.

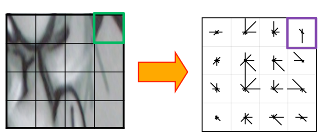
40

Slide credit: Kristen Grauman

## SIFT descriptor [Lowe 2004]

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



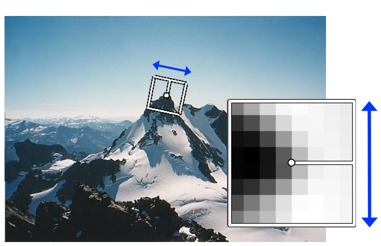



*Why subpatches?*
*Why does SIFT have some illumination invariance?*

41

Slide credit: Kristen Grauman

## Making descriptor rotation invariant



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

42

Image from Matthew Brown

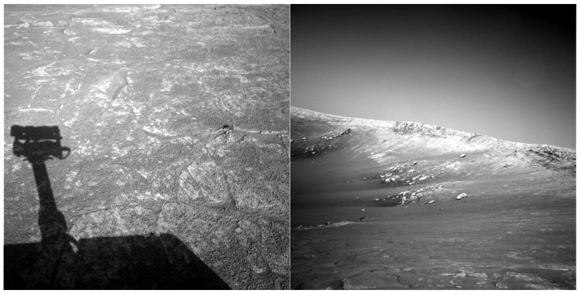
## SIFT descriptor [Lowe 2004]

- 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

43

Steve Seitz

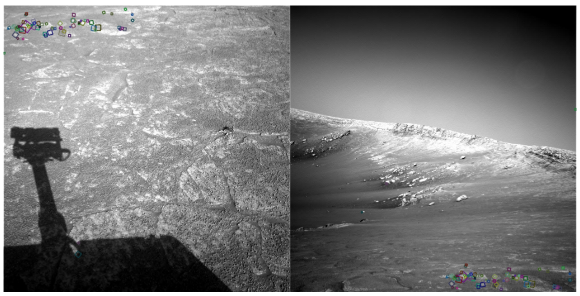
## Example



NASA Mars Rover images

44

## Example



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

45

## SIFT descriptor properties

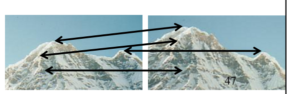
Invariant to
- Scale
- Rotation

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

46

## 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

47

## Matching local features

48

Kristen Grauman

## Matching local features

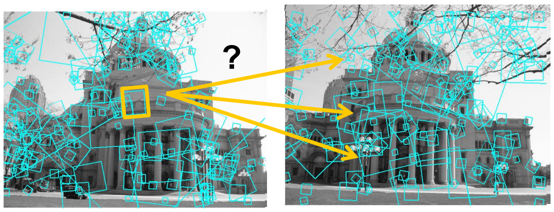



Image 1 Image 2

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)

49

Kristen Grauman

## Ambiguous matches

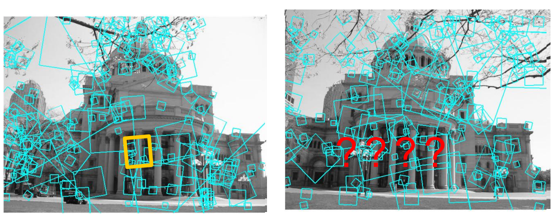



Image 1 Image 2

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.

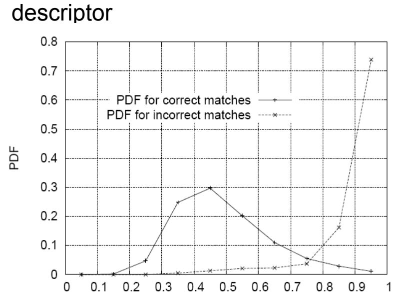
50

Kristen Grauman

## Matching SIFT Descriptors

Nearest neighbor (Euclidean distance)

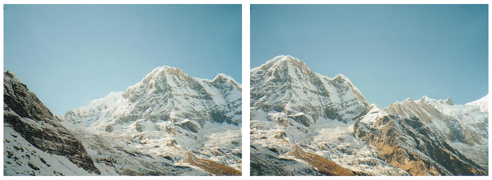
Threshold ratio of nearest to 2nd nearest descriptor




51

Lowe IJCV 2004

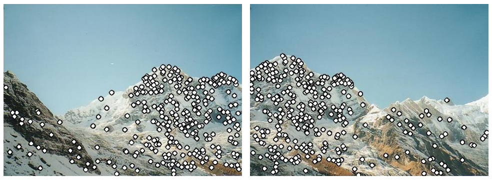
## Recap: robust feature-based alignment



52

Source: L. Lazebnik

## Recap: robust feature-based alignment

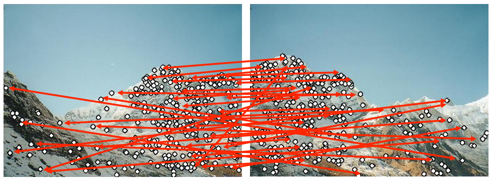


- Extract features

53

Source: L. Lazebnik

## Recap: robust feature-based alignment

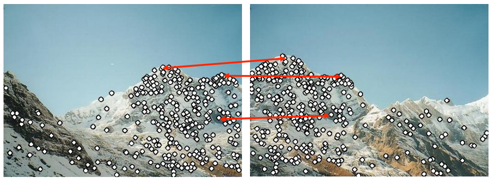


- Extract features
- Compute *putative matches*

54

Source: L. Lazebnik

## Recap: robust feature-based alignment

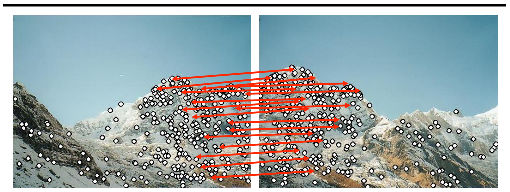


- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

55

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*)

56

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*)

57

Source: L. Lazebnik

## Applications of local invariant features

Wide baseline stereo
Motion tracking
Panoramas
3D reconstruction
Recognition (better for instance matching)
…

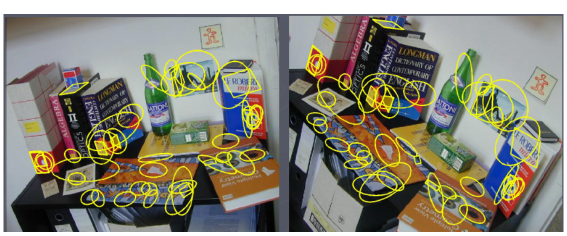
58

## Automatic mosaicing

AutoStitch

59

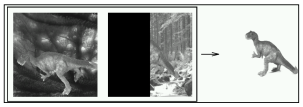
## Wide baseline stereo



[Image from T. Tuytelaars ECCV 2006 tutorial]

60

## Recognition of specific objects, scenes



Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

61

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)

62

Questions?

63