Local features and image matching

May 7th, 2019

Yong Jae Lee
UC Davis

Last time

- RANSAC for robust fitting
  - Lines, translation
- Image mosaics
  - Fitting a 2D transformation
    - Homography

2

Today

Mosaics recap:
How to warp one image to the other, given H?

How to detect *which features* to match?

3

How to stitch together a panorama (a.k.a. mosaic)?

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - (If there are more images, repeat)

4

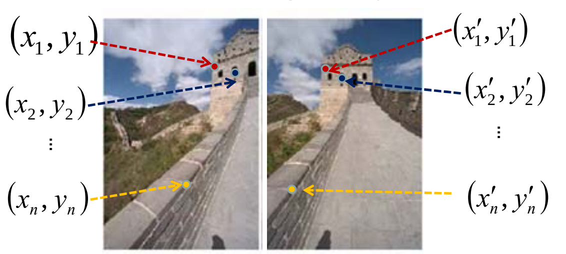
Source: Steve Seitz

Mosaics

Obtain a wider angle view by combining multiple images.

5

Homography



To **compute** the homography given pairs of corresponding points in the images, we need to set up an equation where the parameters of **H** are the unknowns…

6

Solving for homographies

p' = Hp

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Defined up to a scale factor.
Constrain Frobenius norm of H to be 1.

Problem to be solved:

$$\min \|Ah - b\|^2$$
$$s.t. \quad \|h\|^2 = 1$$

where vector of unknowns h = $[h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12}, h_{20}, h_{21}, h_{22}]^T$

7

Adapted from Devi Parikh

Solving for homographies

$$\begin{bmatrix} wx_i' \\ wy_i' \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$wx_i' = h_{00}x_i + h_{01}y_i + h_{02}$$
$$wy_i' = h_{10}x_i + h_{11}y_i + h_{12}$$
$$w = h_{20}x_i + h_{21}y_i + h_{22}$$

$$x_i' = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$
$$y_i' = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$\Updownarrow$$

$$x_i' = \frac{kh_{00}x_i + kh_{01}y_i + kh_{02}}{kh_{20}x_i + kh_{21}y_i + kh_{22}}$$
$$y_i' = \frac{kh_{10}x_i + kh_{11}y_i + kh_{12}}{kh_{20}x_i + kh_{21}y_i + kh_{22}}$$

There are 9 variables $h_{00},\ldots,h_{22}$.
Are there 9 degrees of freedom?

No. We can multiply all h_{ij} by nonzero scalar k without changing the equations:

8

Enforcing 8 DOF

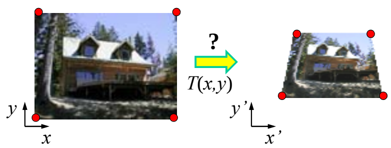
Impose unit vector constraint

$$x_i' = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \qquad y_i' = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

Subject to: $h_{00}^2 + h_{01}^2 + h_{02}^2 + h_{10}^2 + h_{11}^2 + h_{12}^2 + h_{20}^2 + h_{21}^2 + h_{22}^2 = 1$

9

Projective: # correspondences?



How many correspondences needed for projective?

10

Source: Alyosha Efros

RANSAC for estimating homography

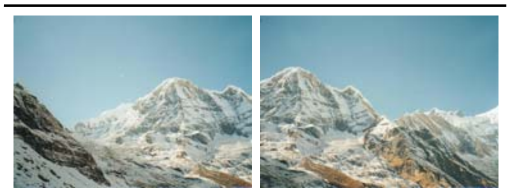
RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \mathbf{H} p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

11

Slide credit: Steve Seitz

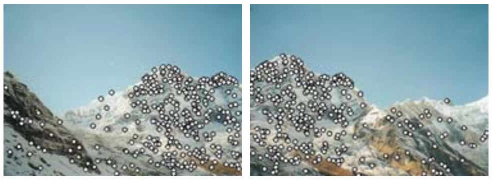
Robust feature-based alignment



12

Source: L. Lazebnik

Robust feature-based alignment

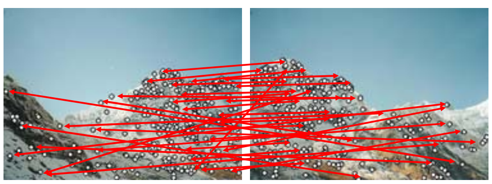


- Extract features

13

Source: L. Lazebnik

Robust feature-based alignment

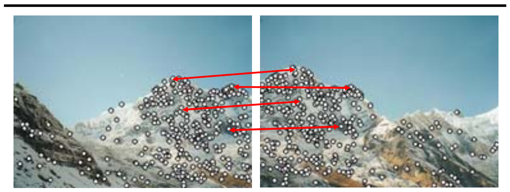


- Extract features
- Compute *putative matches*

14

Source: L. Lazebnik

Robust feature-based alignment

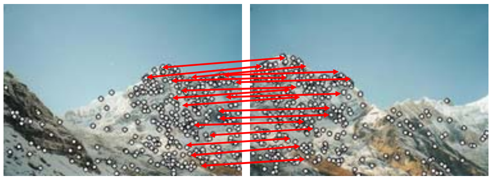


- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

15

Source: L. Lazebnik

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*)

16

Source: L. Lazebnik

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*)

17

Source: L. Lazebnik

Creating and Exploring a Large Photorealistic Virtual Space

Josef Sivic, Biliana Kaneva, Antonio Torralba, Shai Avidan and William T. Freeman, Internet Vision Workshop, CVPR 2008.
http://www.youtube.com/watch?v=E0rboU10rPo

18

Creating and Exploring a Large Photorealistic Virtual Space

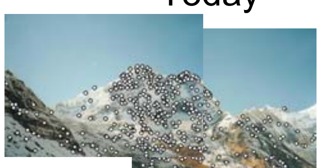
Current view, and desired view in green

Synthesized view from new camera

Induced camera motion

19

Today



Mosaics recap:
How to warp one image to the other, given H?

How to detect *which features* to match?

20

Detecting local invariant features

- Detection of interest points
  - Harris corner detection
  - (Scale invariant blob detection: LoG)
- (Next time: description of local patches)

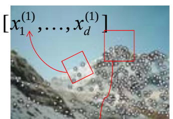
21

Local features: main components

1) Detection: Identify the interest points

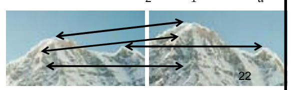
2) Description: Extract vector feature descriptor surrounding each interest point.

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



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

3) Matching: Determine correspondence between descriptors in two views



22

Kristen Grauman

Local features: desired properties

- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

23

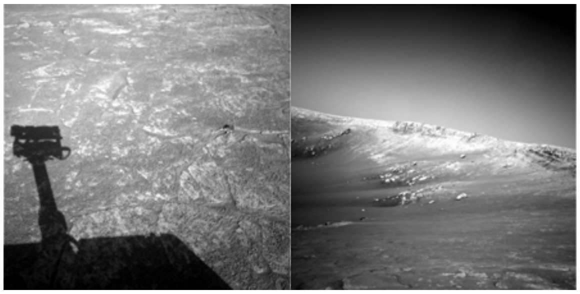
Applications

- Local features have be used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition

24

Lana Lazebnik

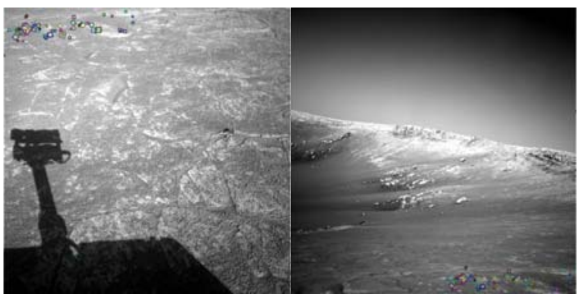
A hard feature matching problem



NASA Mars Rover images

25

Answer below (look for tiny colored squares…)

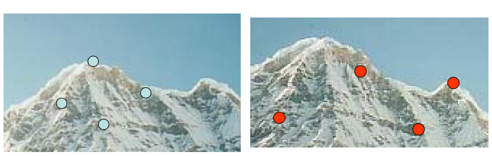


NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

26

Goal: interest operator repeatability

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



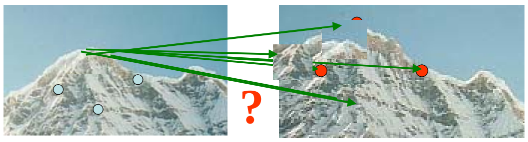
No chance to find true matches!

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

27

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.

28

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

29

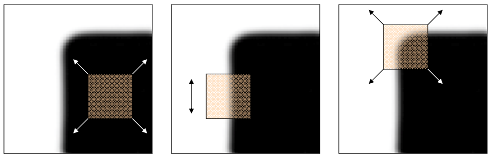
- What points would you choose (for repeatability, distinctiveness)?

30

Corners as distinctive interest points

We should easily recognize the point by looking through a small window

Shifting a window in *any direction* should give *a large change* in intensity



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Slide credit: Alyosha Efros, Darya Frolova, Denis Simakov

31

Corners as distinctive interest points

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

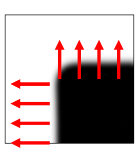
2 x 2 matrix of image derivatives (averaged in neighborhood of a point).

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

32

What does this matrix reveal?

First, consider an axis-aligned corner:



33

What does this matrix reveal?

First, consider an axis-aligned corner:

$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

This means dominant gradient directions align with x or y axis

Look for locations where **both** λ's are large.

If either λ is close to 0, then this is **not** corner-like.

What if we have a corner that is not aligned with the image axes?

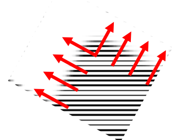
34

What does this matrix reveal?

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

(Eigenvalue decomposition)

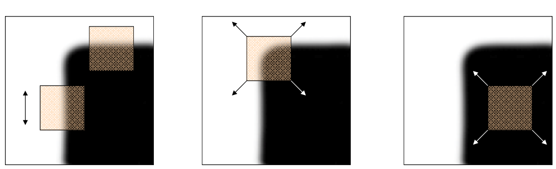
$$Mx_i = \lambda_i x_i$$



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

35

Corner response function



"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;

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

36

Harris corner detector

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

37

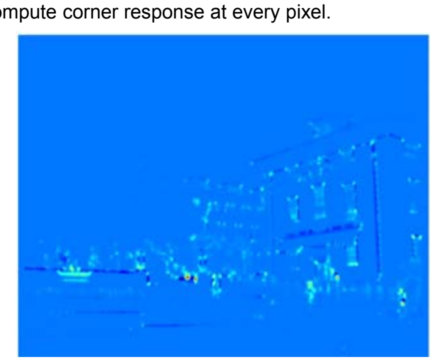
Example of Harris application

38

Kristen Grauman

Example of Harris application

Compute corner response at every pixel.



39

Kristen Grauman

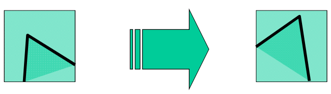
Example of Harris application

40

Kristen Grauman

Properties of the Harris corner detector

Rotation invariant? Yes

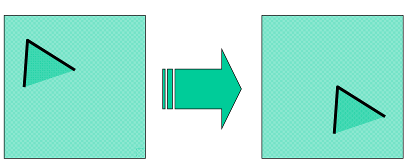


41

Properties of the Harris corner detector

Rotation invariant? Yes

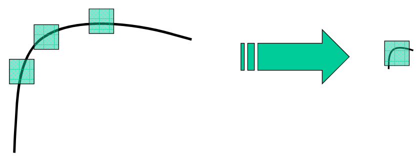
Translation invariant? Yes



42

Properties of the Harris corner detector

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



All points will be classified as edges

Corner !

43

Summary

- Image warping to create mosaic, given homography
- Interest point detection
  - Harris corner detector
  - Next time:
    - Laplacian of Gaussian, automatic scale selection

44