Image warping and stitching

May 2nd, 2019

Yong Jae Lee
UC Davis

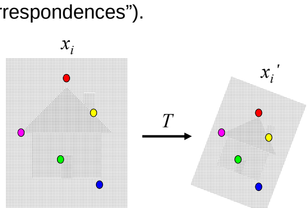
Last time

- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

2

Alignment problem

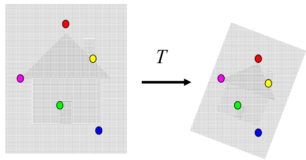
- In alignment, we will **fit the parameters of some transformation** according to a set of matching feature pairs ("correspondences").



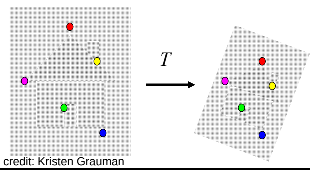
3

Slide credit: Adapted by Devi Parikh from Kristen Grauman

Main questions



Alignment: Given two images, what is the transformation between them?

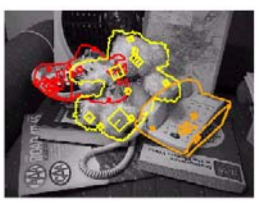


Warping: Given a source image and a transformation, what does the transformed output look like?

4

Slide credit: Kristen Grauman

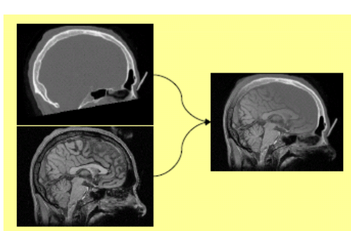
Motivation for feature-based alignment: Recognition



5

Figures from David Lowe

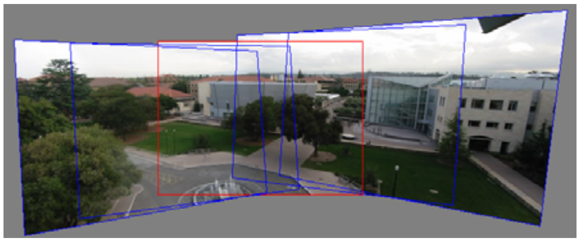
Motivation for feature-based alignment: Medical image registration



6

Slide credit: Kristen Grauman

Motivation for feature-based alignment: Image mosaics



7

Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fall/

Parametric (global) warping

Examples of parametric warps:


translation


rotation


aspect


affine


perspective

8

Source: Alyosha Efros

Parametric (global) warping

p = (x,y)

→ T →

p’ = (x’,y’)

Transformation T is a coordinate-changing machine:

p’ = *T*(p)

What does it mean that *T* is **global**?

- Is the same for any point p
- can be described by just a few numbers (parameters)

Let’s represent *T* as a matrix:

p’ = **M**p

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

9

Source: Alyosha Efros

Homogeneous coordinates

To convert to homogeneous coordinates:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

10

Slide credit: Kristen Grauman

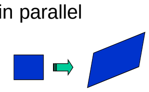
2D Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Affine transformations are combinations of …

- Linear transformations, and
- Translations

Parallel lines remain parallel



11

Slide credit: Kristen Grauman

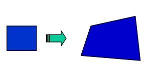
Projective Transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Projective transformations:

- Affine transformations, and
- Projective warps

Parallel lines do not necessarily remain parallel

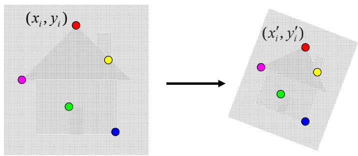


12

Slide credit: Kristen Grauman

Fitting an affine transformation

- Assuming we know the correspondences, how do we get the transformation?



$$\begin{bmatrix}x_i'\\ y_i'\end{bmatrix}=\begin{bmatrix}m_1 & m_2\\ m_3 & m_4\end{bmatrix}\begin{bmatrix}x_i\\ y_i\end{bmatrix}+\begin{bmatrix}t_1\\ t_2\end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0\\ 0 & 0 & x_i & y_i & 0 & 1\\ & & \cdots & & & \end{bmatrix}\begin{bmatrix}m_1\\ m_2\\ m_3\\ m_4\\ t_1\\ t_2\end{bmatrix}=\begin{bmatrix}\cdots\\ x_i'\\ y_i'\\ \cdots\end{bmatrix}$$

13

Slide credit: Kristen Grauman

RANSAC: General form

- RANSAC loop:

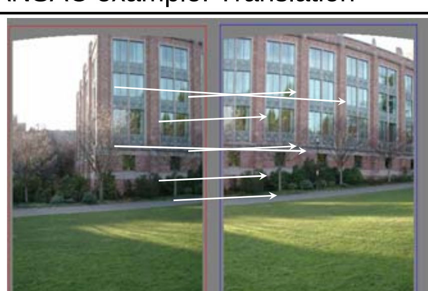
1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

14

Slide credit: Kristen Grauman

RANSAC example: Translation

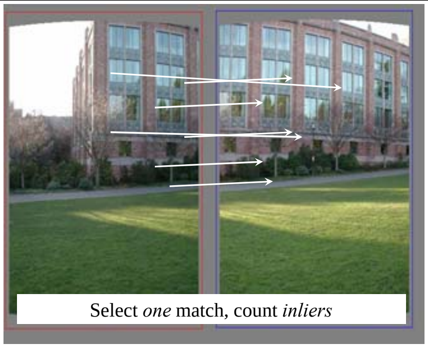


Select *one* match, count *inliers*

15

Source: Rick Szeliski

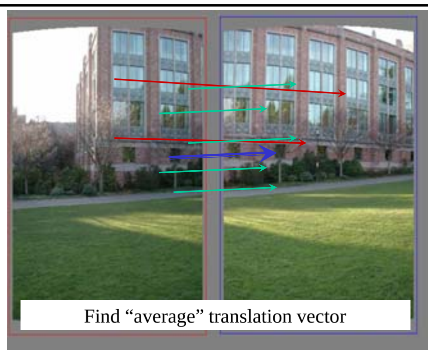
RANSAC example: Translation



Select *one* match, count *inliers*

16

RANSAC example: Translation



Find "average" translation vector

17

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - Doesn't work well for low inlier ratios (too many iterations, or can fail completely)

18

Lana Lazebnik

Today

- Image mosaics
  - Fitting a 2D transformation
    - Homography
  - 2D image warping
  - Computing an image mosaic

19

HP frames commercial

- http://www.youtube.com/watch?v=2RPl5vPEoQk

20

Mosaics

Obtain a wider angle view by combining multiple images.

21

Slide credit: Kristen Grauman

Panoramic Photos are old

- Sydney, 1875

Beirut, late 1800's

Slide credit: James Hays

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

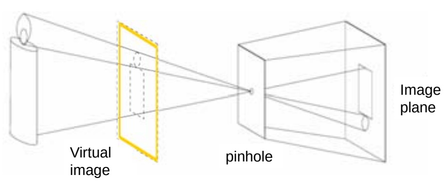
- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - (If there are more images, repeat)
- …but **wait**, why should this work at all?
  - What about the 3D geometry of the scene?
  - Why aren't we using it?

23

Source: Steve Seitz

Pinhole camera

- Pinhole camera is a simple model to approximate imaging process, perspective **projection**.

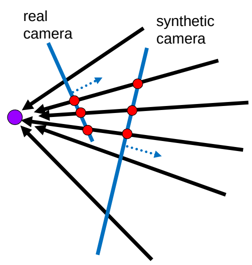


If we treat pinhole as a point, only one ray from any given point can enter the camera.

24

Slide credit: Kristen Grauman   Fig from Forsyth and Ponce

Mosaics: generating synthetic views



Can generate any synthetic camera view
as long as it has **the same center of projection**!

25

Source: Alyosha Efros

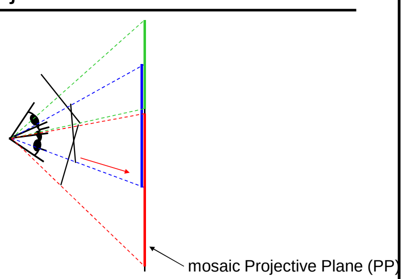
Mosaics

Obtain a wider angle view by combining multiple images.

26

Slide credit: Kristen Grauman

Image reprojection



The mosaic has a natural interpretation in 3D

- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*

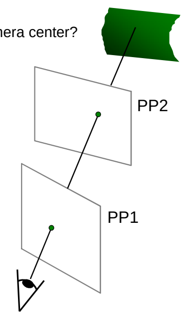
27

Source: Steve Seitz

Image reprojection

Basic question

- How to relate two images from the same camera center?
  - how to map a pixel from PP1 to PP2



Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2

Observation:
Rather than thinking of this as a 3D reprojection, think of it as a 2D **image warp** from one image to another.

28

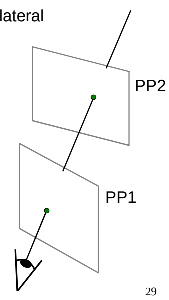
Source: Alyosha Efros

Image reprojection: Homography

A projective transform is a mapping between any two PPs with the same center of projection

- rectangle should map to arbitrary quadrilateral
- parallel lines aren't preserved
- but must preserve straight lines

called **Homography**



$$\underset{\mathbf{p'}}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{\mathbf{H}}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

29

Source: Alyosha Efros

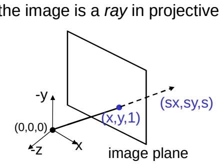
The projective plane

Why do we need homogeneous coordinates?

- represent points at infinity, homographies, perspective projection, multi-view relationships

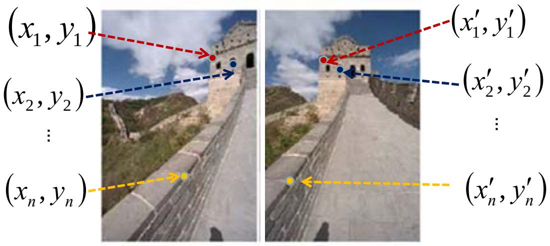
What is the geometric intuition?

- a point in the image is a *ray* in projective space



- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  – all points on the ray are equivalent: (x, y, 1) ≅ (sx, sy, s)

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…

31

Slide credit: Kristen Grauman

Solving for homographies

p' = Hp

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

Upto a scale factor.

Constraint 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 = $[h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12}, h_{20}, h_{21}, h_{22}]^T$

32

Solving for homographies

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

$$wx'_i = h_{00}x_i + h_{01}y_i + h_{02}$$
$$wy'_i = h_{10}x_i + h_{11}y_i + h_{12}$$
$$w = h_{20}x_i + h_{21}y_i + h_{22}$$

$$x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}$$
$$y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Solving for homographies

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

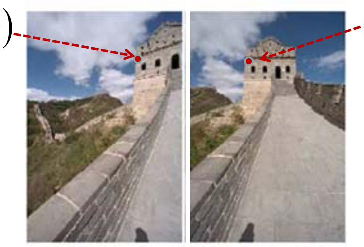
A (2n × 9) **h** (9) **0** (2n)

Defines a least squares problem: $\text{minimize } \|\mathbf{A}\mathbf{h} - \mathbf{0}\|^2$

- Since **h** is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$ $(i.e., \|h\|^2 = 1)$
- Solution: $\hat{\mathbf{h}}$ = eigenvector of $\mathbf{A}^T\mathbf{A}$ with smallest eigenvalue
- Works with 4 or more points

Homography

(x, y)



$\left(\frac{wx'}{w}, \frac{wy'}{w} \right) = (x', y')$

To **apply** a given homography **H**

- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

Slide credit: Kristen Grauman

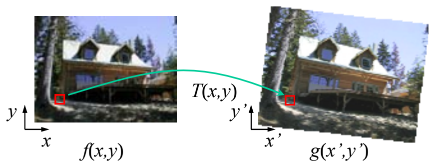
35

Today

- RANSAC for robust fitting
  - Lines, translation
- Image mosaics
  - Fitting a 2D transformation
    - Homography
  - 2D image warping
  - Computing an image mosaic

36

Image warping

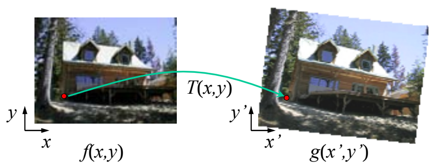


Given a coordinate transform and a source image *f*(*x*,*y*), how do we compute a transformed image *g*(*x'*,*y'*) = *f*(*T*(*x*,*y*))?

37

Slide from Alyosha Efros

Forward warping



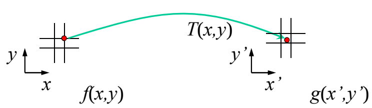
Send each pixel *f*(*x*,*y*) to its corresponding location (*x'*,*y'*) = *T*(*x*,*y*) in the second image

Q: what if pixel lands "between" two pixels?

38

Slide from Alyosha Efros

Forward warping



Send each pixel *f*(*x*,*y*) to its corresponding location (*x'*,*y'*) = *T*(*x*,*y*) in the second image

Q: what if pixel lands "between" two pixels?

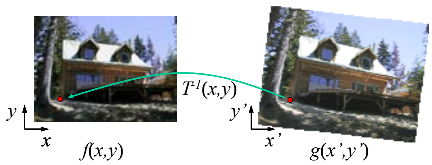
A: distribute color among neighboring pixels (x',y')

- Known as "splatting"

39

Slide from Alyosha Efros

Inverse warping



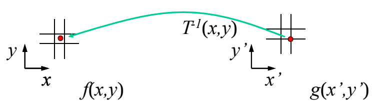
Get each pixel *g*(*x'*,*y'*) from its corresponding location
(*x*,*y*) = *T*$^{-1}$(*x'*,*y'*) in the first image

Q: what if pixel comes from "between" two pixels?

40

Slide from Alyosha Efros

Inverse warping



Get each pixel *g*(*x'*,*y'*) from its corresponding location
(*x*,*y*) = *T*$^{-1}$(*x'*,*y'*) in the first image

Q: what if pixel comes from "between" two pixels?

A: *Interpolate* color value from neighbors

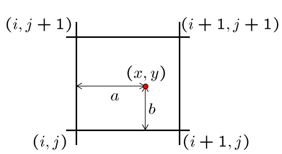
– nearest neighbor, bilinear… **`>> help interp2`**

41

Slide from Alyosha Efros

Bilinear interpolation

Sampling at *f(x,y):*



$$
\begin{aligned}
f(x,y) = \ & (1-a)(1-b) \quad f[i,j] \\
& +a(1-b) \quad f[i+1,j] \\
& +ab \quad f[i+1,j+1] \\
& +(1-a)b \quad f[i,j+1]
\end{aligned}
$$

42

Slide from Alyosha Efros

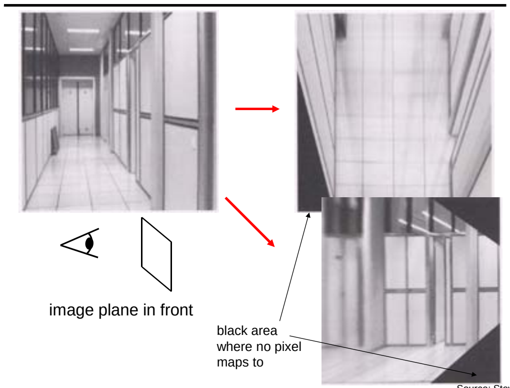
Recap: How to stitch together a panorama (a.k.a. mosaic)?

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

43

Source: Steve Seitz

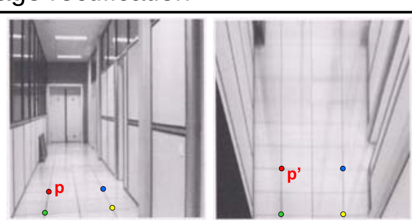
Image warping with homographies



44

Source: Steve Seitz

Image rectification

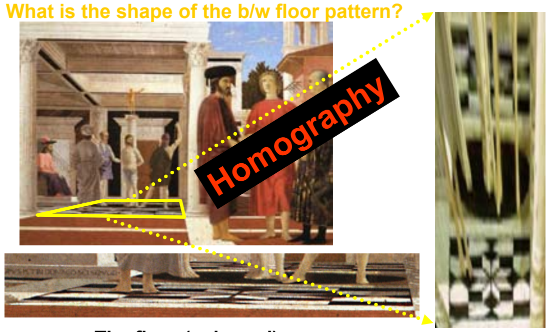


45

Slide credit: Kristen Grauman

Analysing patterns and shapes

What is the shape of the b/w floor pattern?



The floor (enlarged)

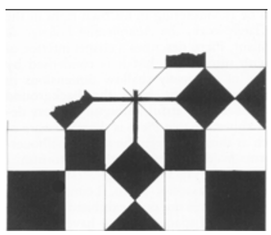
Automatically rectified floor

Slide from Antonio Criminisi

46

Analysing patterns and shapes

Automatic rectification



From Martin Kemp *The Science of Art* *(manual reconstruction)*

Slide from Antonio Criminisi

47

Analysing patterns and shapes

What is the (complicated) shape of the floor pattern?

Automatically rectified floor

St. Lucy Altarpiece, **D. Veneziano**

Slide from Criminisi

48

Analysing patterns and shapes

Automatic rectification

From Martin Kemp, *The Science of Art* *(manual reconstruction)*

49

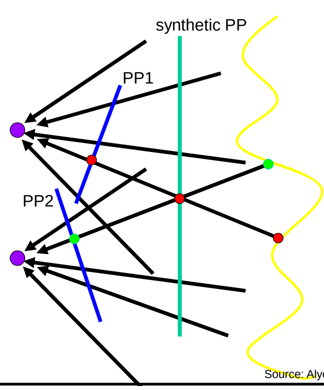
Slide from Criminisi

Julian Beever: Manual Homographies

http://users.skynet.be/J.Beever/pave.htm

Changing camera center

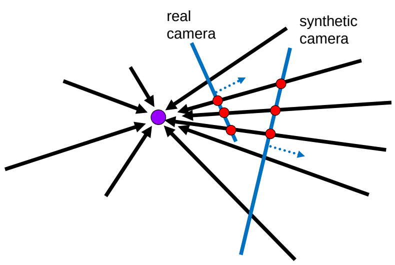
Does it still work?



51

Source: Alyosha Efros

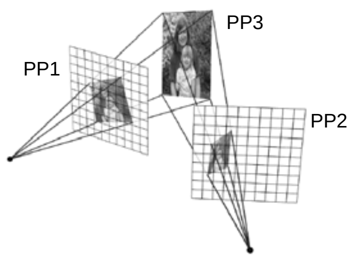
Recall: same camera center



Can generate synthetic camera view
as long as it has **the same center of projection**. 52

Source: Alyosha Efros

…Or: Planar scene (or far away)



PP3 is a projection plane of both centers of projection, so we are OK!

This is how big aerial photographs are made

53

Source: Alyosha Efros

54

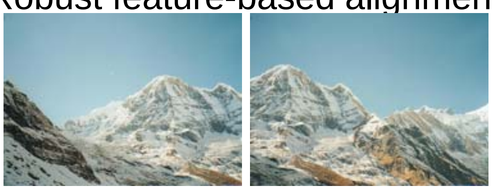
RANSAC for estimating homography

- 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

55

Slide credit: Steve Seitz

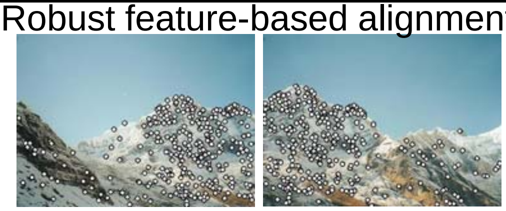
Robust feature-based alignment



56

Source: L. Lazebnik

Robust feature-based alignment



- Extract features

57

Source: L. Lazebnik

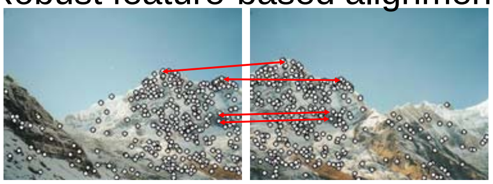
Robust feature-based alignment

- Extract features
- Compute *putative matches*

58

Source: L. Lazebnik

Robust feature-based alignment

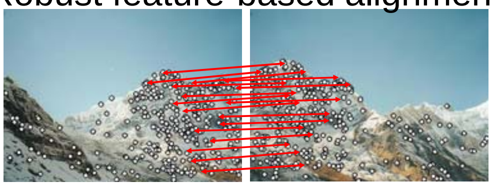


- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

59

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*)

60

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*)

61

Source: L. Lazebnik

Summary: alignment & warping

- Write **2d transformations** as matrix-vector multiplication (including translation when we use homogeneous coordinates)
- **Fitting transformations**: solve for unknown parameters given corresponding points from two views (affine, projective (homography)).
- Perform **image warping** (inverse)
- **Mosaics**: uses homography and image warping to merge views taken from same center of projection.

62

Slide credit: Kristen Grauman

Next time: which features should we match?

63

Slide credit: Kristen Grauman

Questions?

64