Mosaics review
Robust fitting

Kristen Grauman
UT-Austin

Thursday, Oct 8

Today

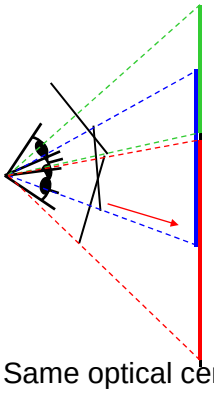
- Review mosaic construction
- Robust fitting with RANSAC
- Midterm questions

Mosaics

Obtain a wider angle view by combining multiple images.

Mosaics with image reprojection

- Applicable:



Same optical center, rotated camera

Planar scene

Very distant (~planar) scene

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

- Compute transformation between second image and first (homography)
- Transform the second image to overlap with the first
- Blend the two together to create a mosaic

- (If there are more images, repeat)

Source: Steve Seitz

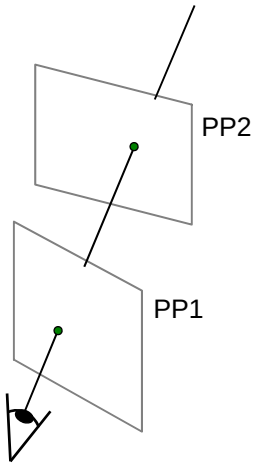
Image reprojection: Homography

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

called **Homography**

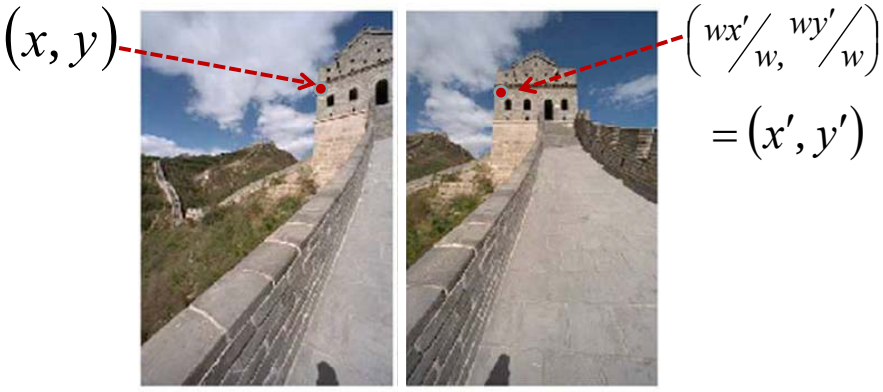
$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\mathbf{p'}$ $\quad$ $\mathbf{H}$ $\quad$ $\mathbf{p}$



Source: Alyosha Efros

Homography



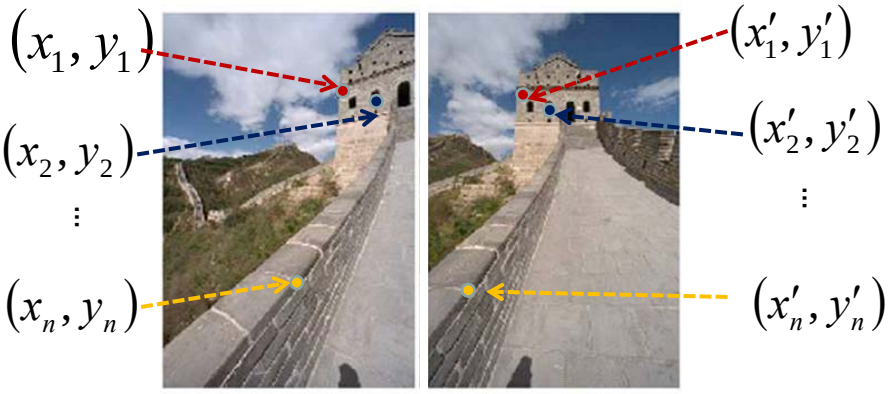
To **apply** a given homography **H**

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

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

p' **H** **p**

Homography



Given pairs of corresponding points in the images, can solve for the parameters of the homography matrix **H**.

Panoramas: main steps

- **1. Collect correspondences (manually for now)**
- **2. Solve for homography matrix H**
  - Least squares solution
- **3. Warp content from one image frame to the other to combine: say im1 into im2 reference frame**
- **4. Overlay im2 content onto the warped im1 content.**

Panoramas: main steps

- **1. Collect correspondences (manually for now)**
- **2. Solve for homography matrix H**
  - Least squares solution
- **3. Warp content from one image frame to the other to combine: say im1 into im2 reference frame**
  - Determine bounds of the new combined image:
    - Where will the corners of im1 fall in im2's coordinate frame?
    - We will attempt to lookup colors for any of these positions we can get from im1.
- **4. Overlay im2 content onto the warped im1 content.**

im1

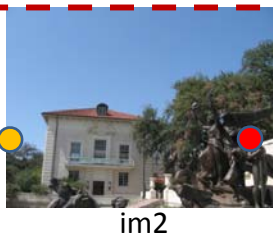
H

im2

(Assuming we have solved for the H that maps points from im1 to im2.)

$$\begin{bmatrix} wx_2 \\ wy_2 \\ w \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

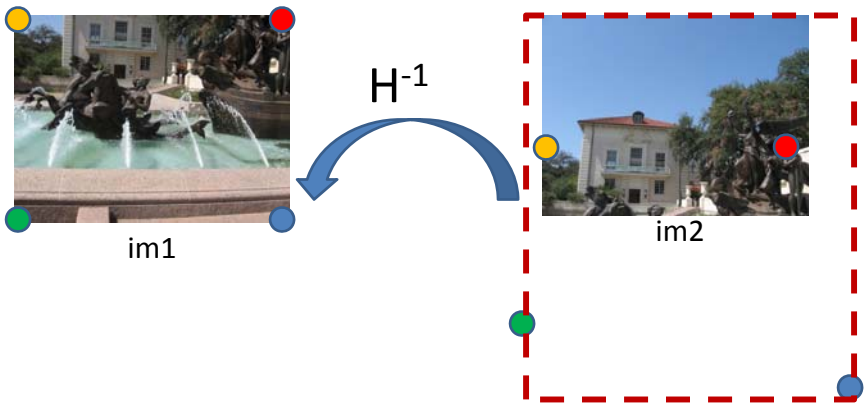
im1



im2

Panoramas: main steps

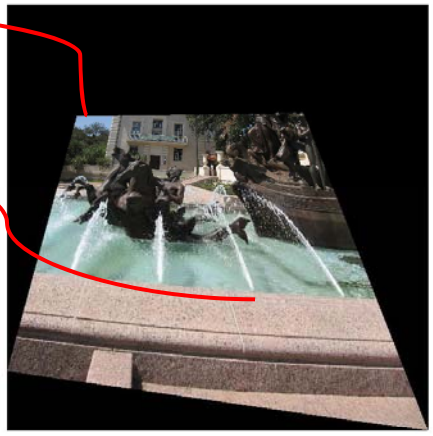
- **1. Collect correspondences (manually for now)**
- **2. Solve for homography matrix H**
  - Least squares solution
- **3. Warp content from one image frame to the other to combine: say im1 into im2 reference frame**
  - Determine bounds of the new combined image:
    - Where will the corners of im1 fall in im2's coordinate frame?
    - We will attempt to lookup colors for any of these positions we can get from im1.
  - Inverse warp:
    - Compute coordinates in im1's reference frame (via homography) for all points in that range.
    - Lookup all colors for all these positions from im1 (interp2)
- **4. Overlay im2 content onto the warped im1 content.**



(Assuming we have solved for the H that maps points from im1 to im2.)

im1

im2

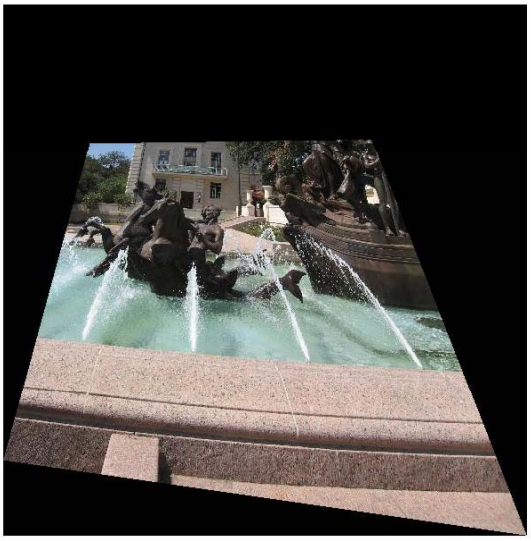


im1 warped into reference frame of im2.

Use interp2 to ask for the colors (possibly interpolated) from im1 at all the positions needed in im2's reference frame.

Panoramas: main steps

- **1. Collect correspondences (manually for now)**
- **2. Solve for homography matrix H**
  - Least squares solution
- **3. Warp content from one image frame to the other to combine: say im1 into im2 reference frame**
  - Determine bounds of the new combined image:
    - Where will the corners of im1 fall in im2's coordinate frame?
    - We will attempt to lookup colors for any of these positions we can get from im1.
  - Inverse warp:
    - Compute coordinates in im1's reference frame (via homography) for all points in that range.
    - Lookup all colors for all these positions from im1 (interp2)
- **4. Overlay im2 content onto the warped im1 content.**
  - Careful about new bounds of the output image



HP “Frames” commercials

- http://www.youtube.com/watch?v=UirmvNktkBc
- http://www.youtube.com/watch?v=2RPl5vPEoQk
- http://www.youtube.com/watch?v=lde77E4PY4Q

Summary: alignment & warping

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

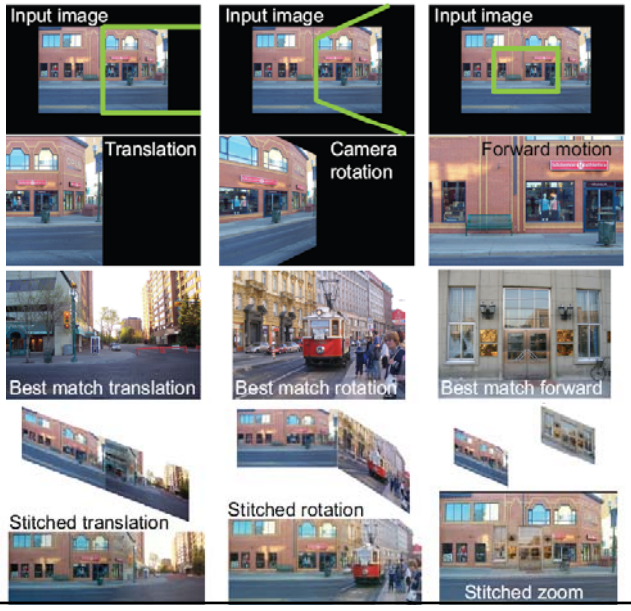
Boundary extension

- Wide-Angle Memories of Close-Up Scenes, Helene Intraub and Michael Richardson, Journal of Experimental Psychology: Learning, Memory, and Cognition, 1989, Vol. 15, No. 2, 179-187

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

Creating and Exploring a Large Photorealistic Virtual Space



Current view, and desired view in green

Synthesized view from new camera

Induced camera motion

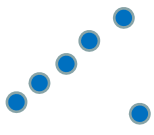
Today

- Review mosaic construction
- Robust fitting with RANSAC
- Midterm questions

Outliers

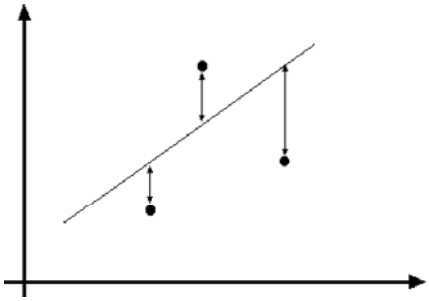
- **Outliers** can hurt the quality of our parameter estimates, e.g.,
  - an erroneous pair of matching points from two images
  - an edge point that is noise, or doesn't belong to the line we are fitting.



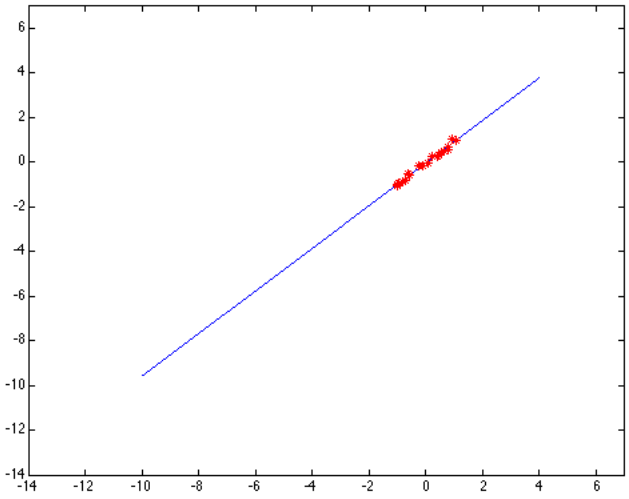


Example: least squares line fitting

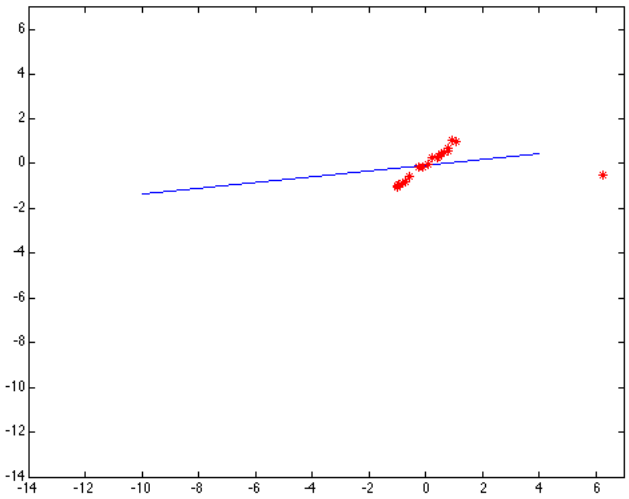
- Assuming all the points that belong to a particular line are known



Outliers affect least squares fit



Outliers affect least squares fit



RANSAC

- RANdom Sample Consensus
- Approach: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.
- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

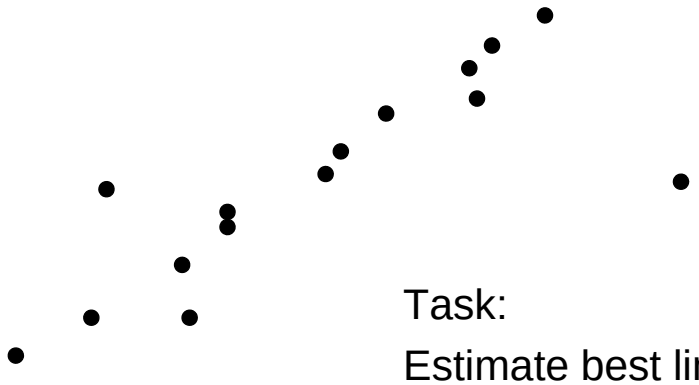
RANSAC

- 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 least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

RANSAC Line Fitting Example

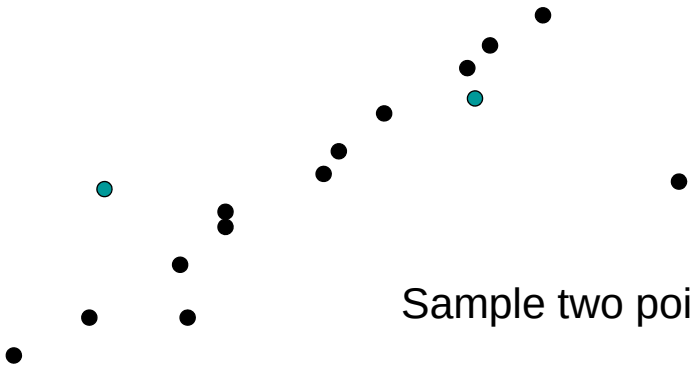


Task:

Estimate best line

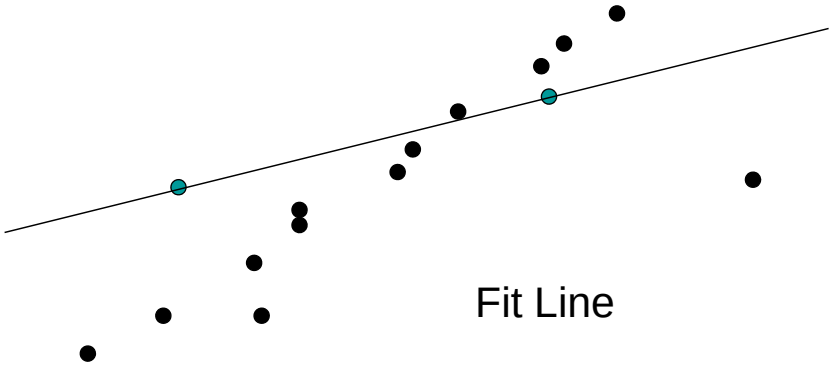
Slide credit: Jinxiang Chai, CMU

RANSAC Line Fitting Example



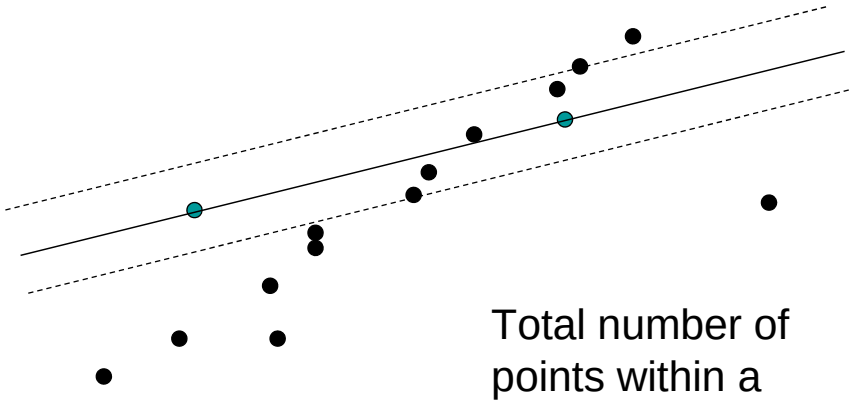
Sample two points

RANSAC Line Fitting Example



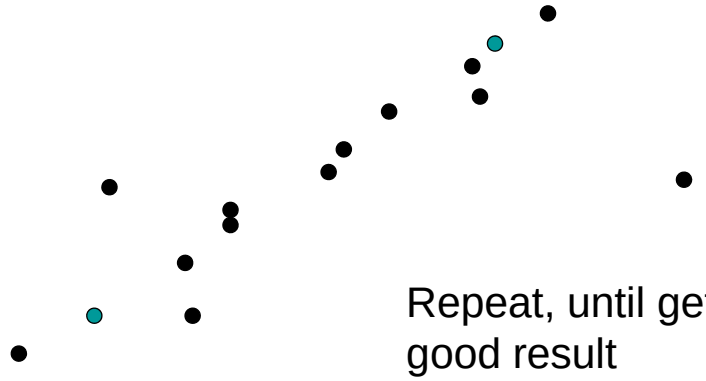
Fit Line

RANSAC Line Fitting Example



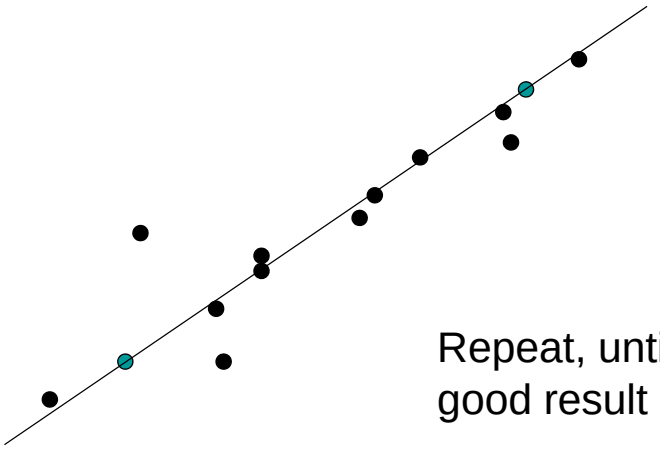
Total number of points within a threshold of line.

RANSAC Line Fitting Example



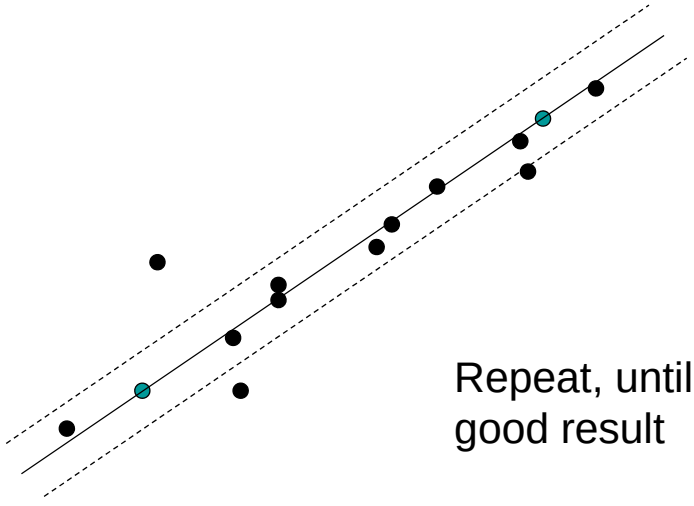
Repeat, until get a good result

RANSAC Line Fitting Example



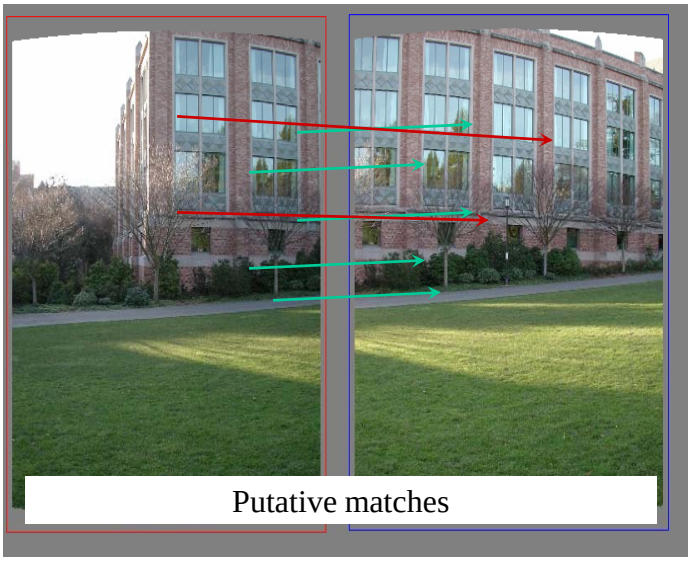
Repeat, until get a good result

RANSAC Line Fitting Example



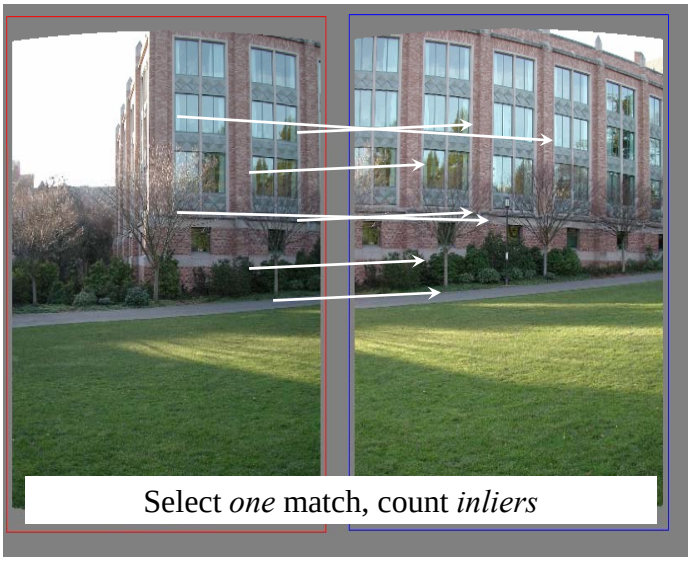
Repeat, until get a good result

RANSAC example: Translation

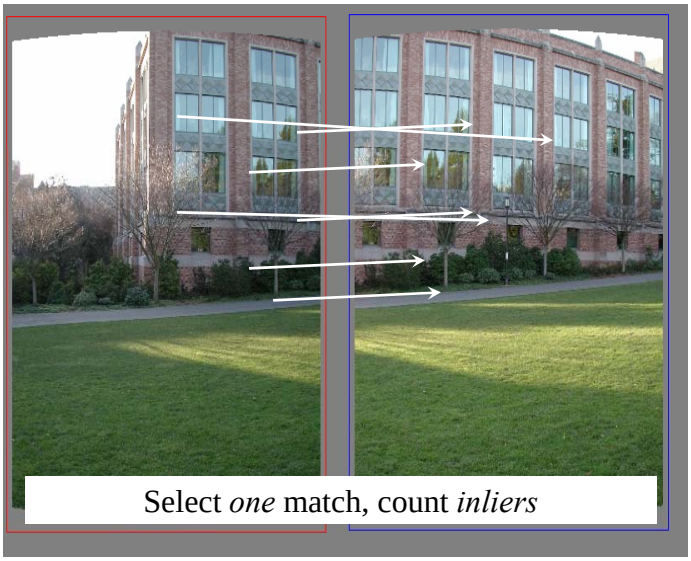


Source: Rick Szeliski

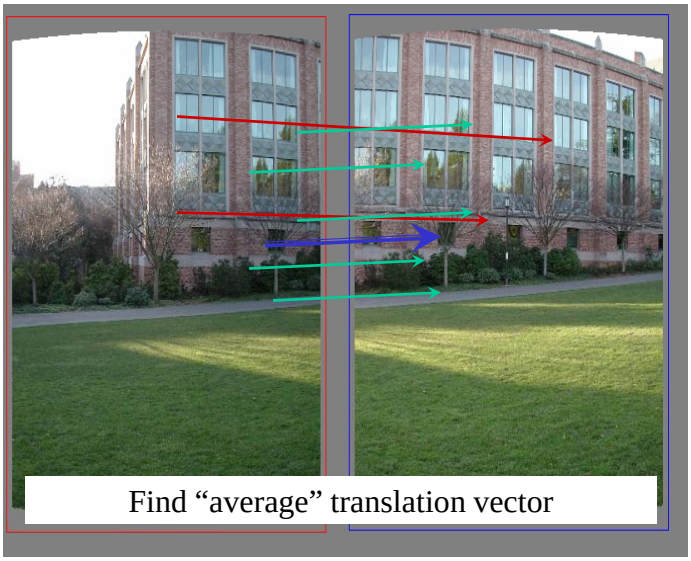
RANSAC example: Translation



RANSAC example: Translation



RANSAC example: Translation



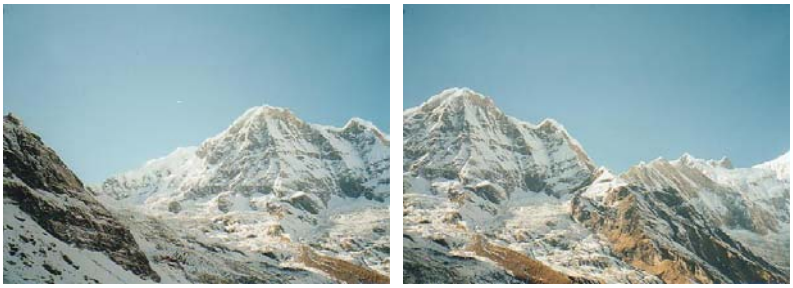
RANSAC for estimating homography

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

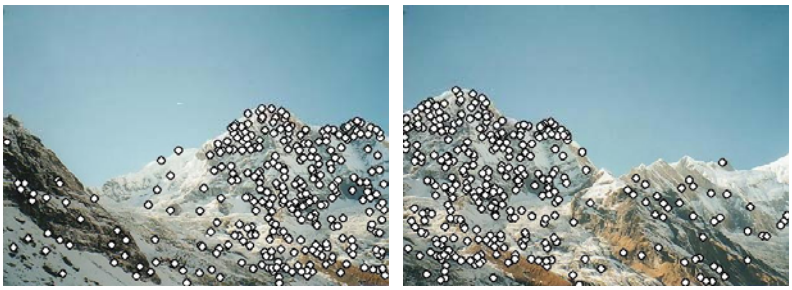
Slide credit: Steve Seitz

Robust feature-based alignment



Source: L. Lazebnik

Robust feature-based alignment



- Extract features

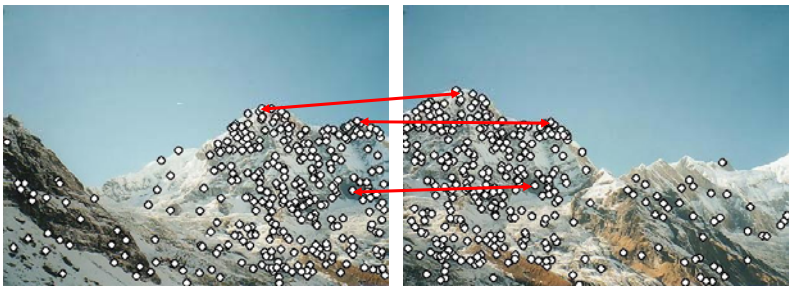
Source: L. Lazebnik

Robust feature-based alignment

- Extract features
- Compute *putative matches*

Source: L. Lazebnik

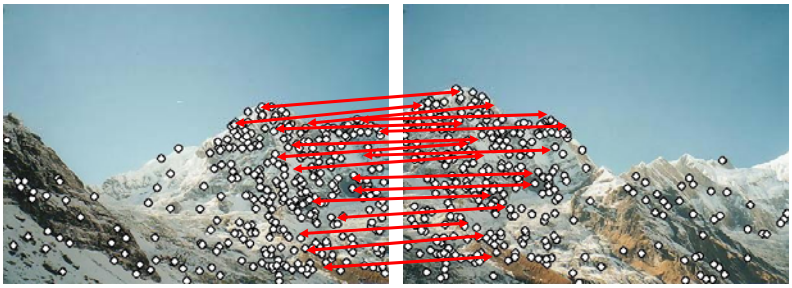
Robust feature-based alignment



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

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*)

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*)

Source: L. Lazebnik

Today

- Review mosaic construction
- Robust fitting with RANSAC
- Midterm questions