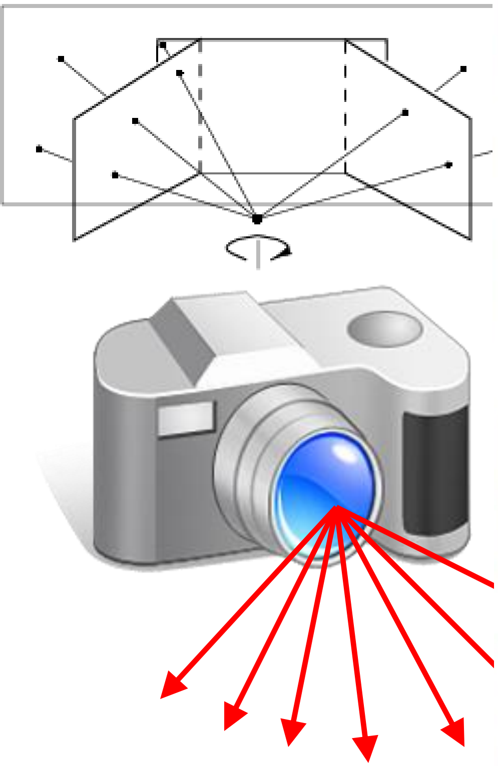
Image Mosaics

COS 429

Princeton University

Image Mosaics (Panoramas)

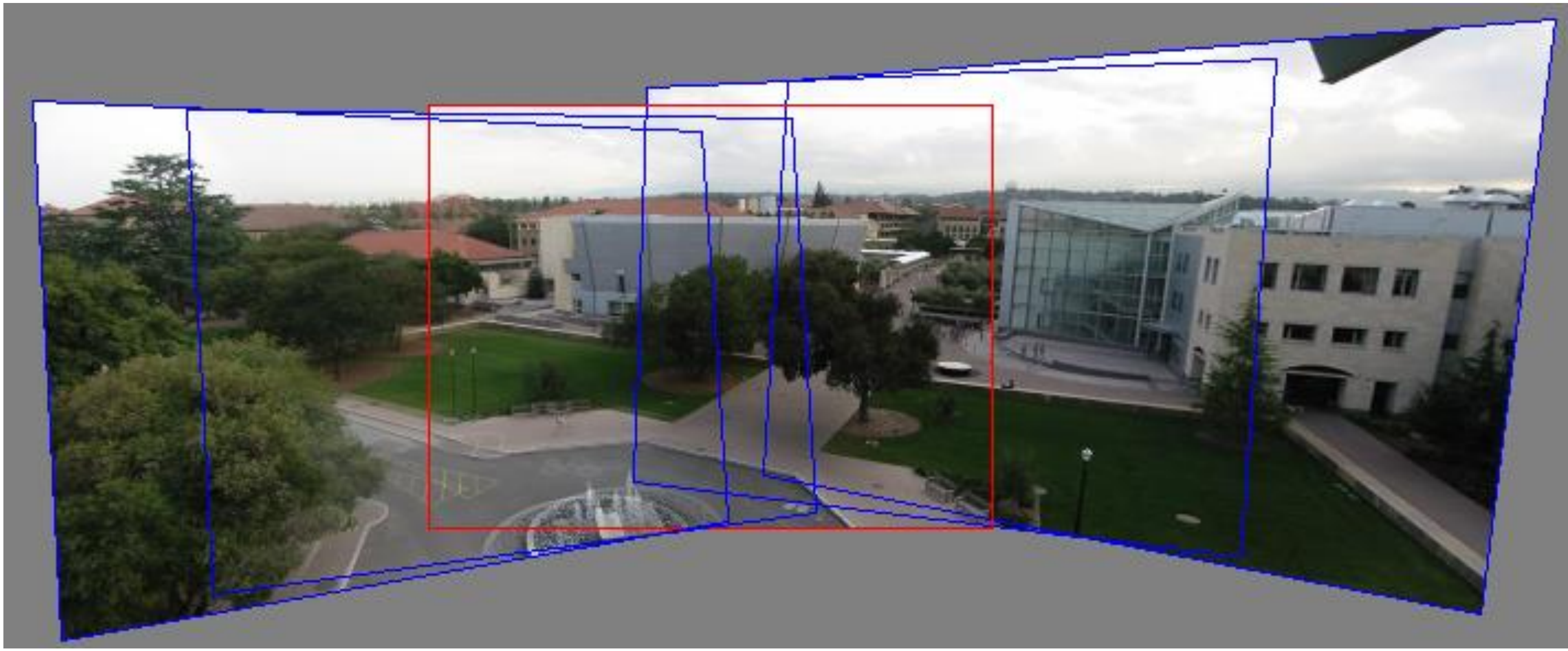


. . .

image from S. Seitz

Obtain a wider angle view by combining multiple images.

Image Mosaics (Panoramas)

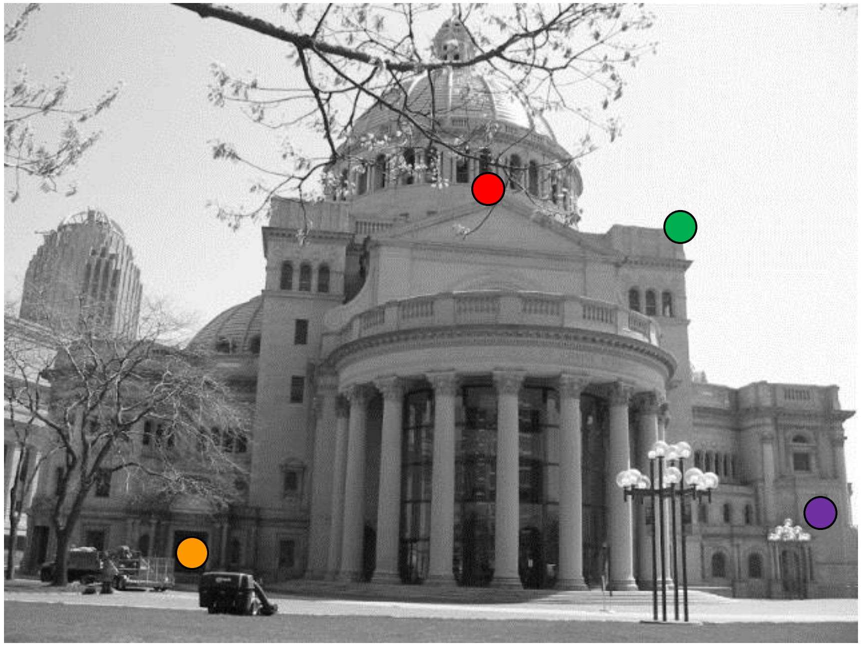


To construct an image mosaic, we need to find the homographies (projective transformations) that map image planes onto one

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

Image Mosaics (Panoramas)

Computing a mosaic (panorama) requires finding a set of >=4 point correspondences



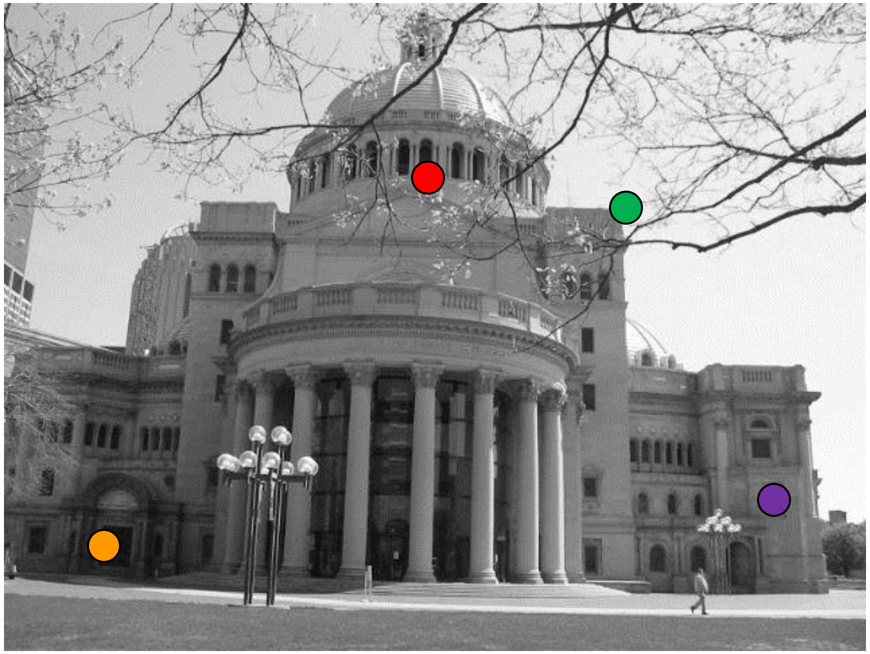


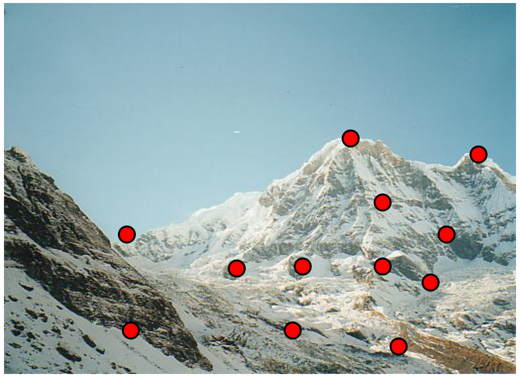
Image Mosaicing

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor
   for each feature
3) Feature Matching:
   Find candidate matches
   between features
4) Feature Correspondence:
   Find consistent set of
   (inlier) correspondences
   between features

Kristen Grauman

Image Mosaicing (last time)

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features



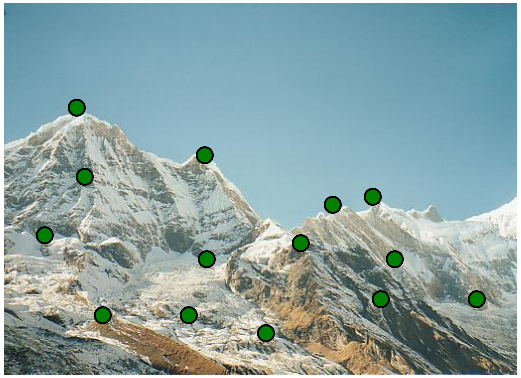
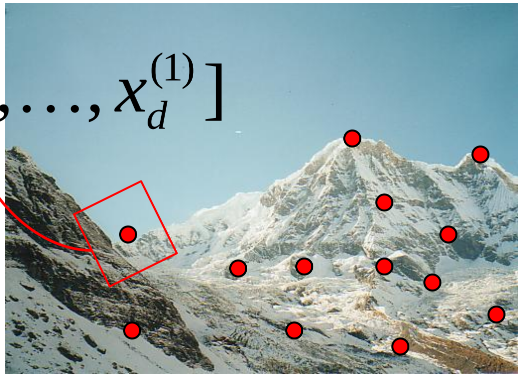
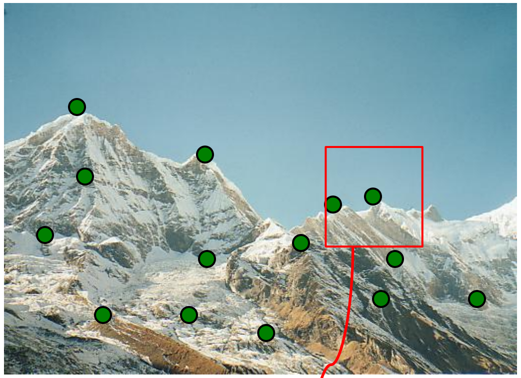


Image Mosaicing (last time)

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features

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

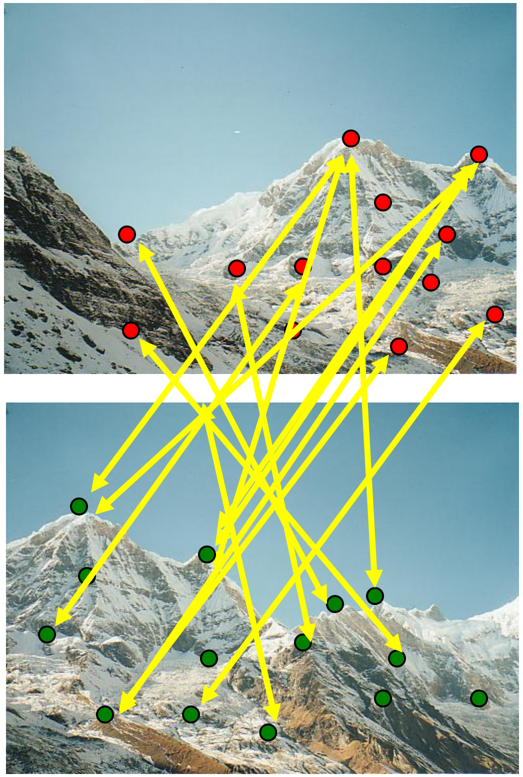




$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}$$

Image Mosaicing (this time)

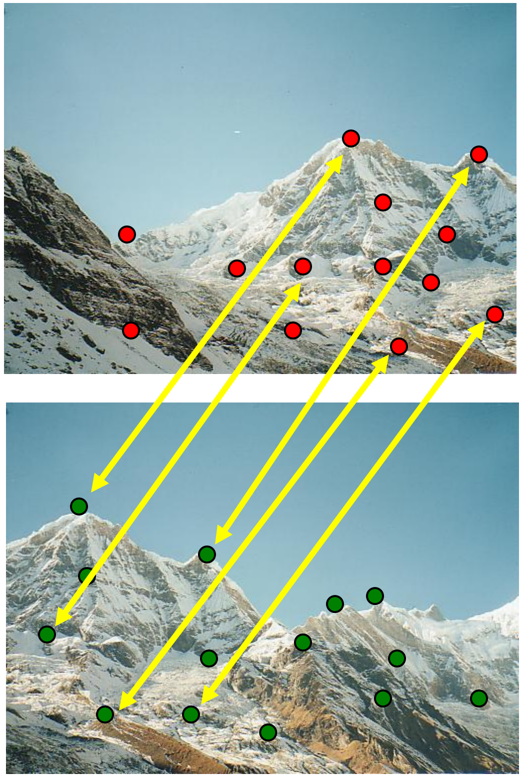
1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features



Kristen Grauman

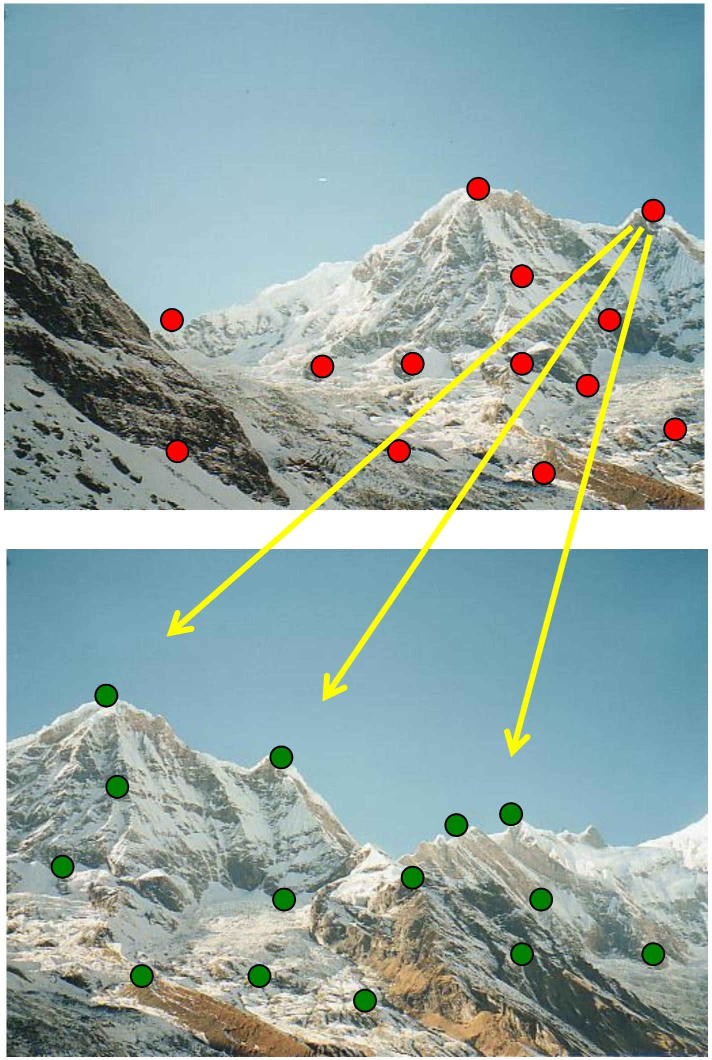
Image Mosaicing (this time)

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features



Kristen Grauman

Feature Matching

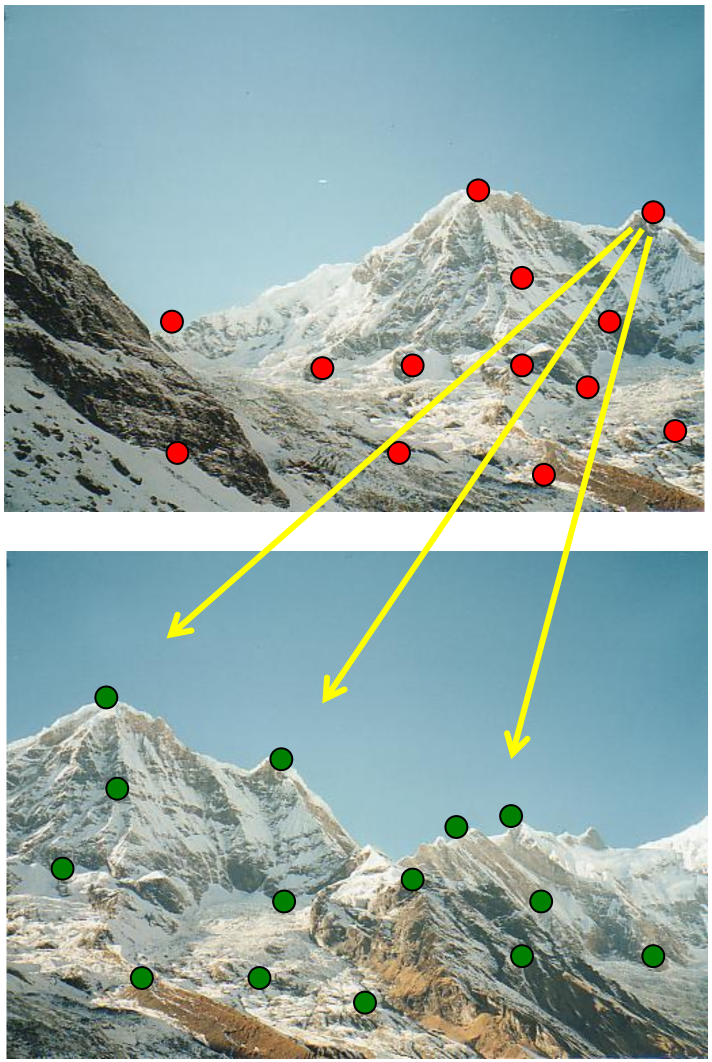


Goal: produce a set of candidate matches that contains…

- Many "correct" matches
- As few "wrong" matches as possible

We will consider only these candidate matches when searching for correspondences

Feature Matching

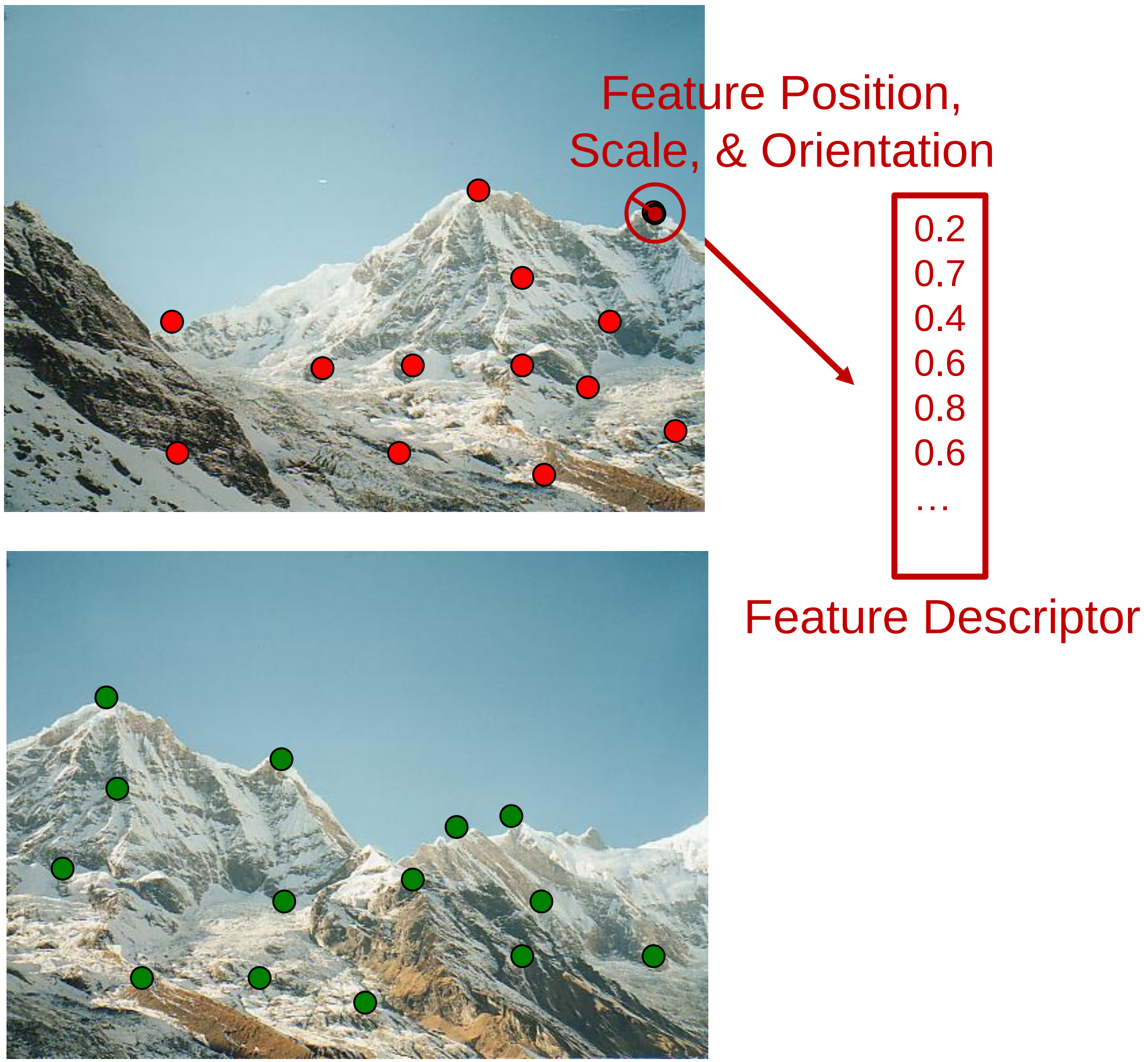


How?

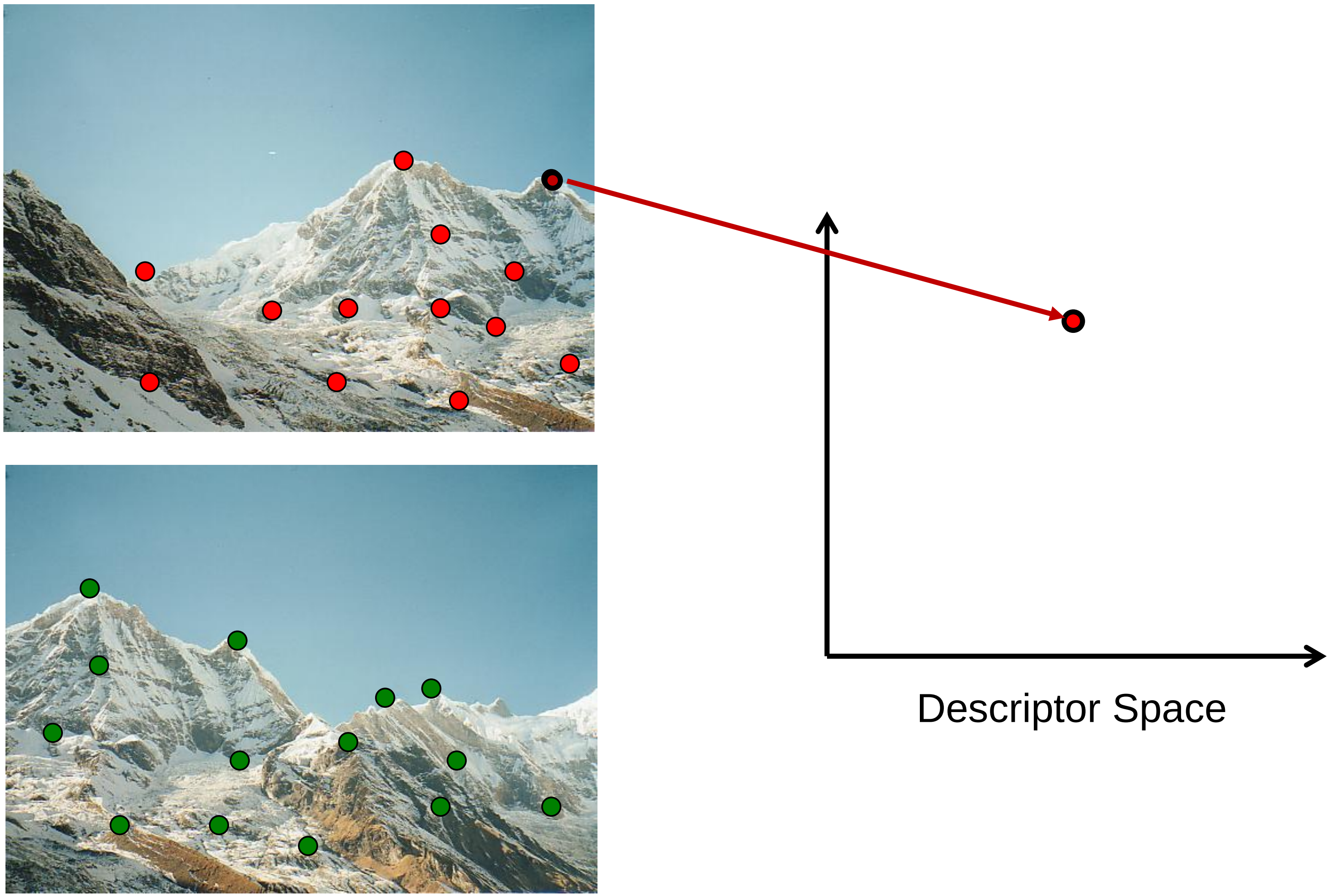
Feature Matching

Feature Position,
Scale, & Orientation

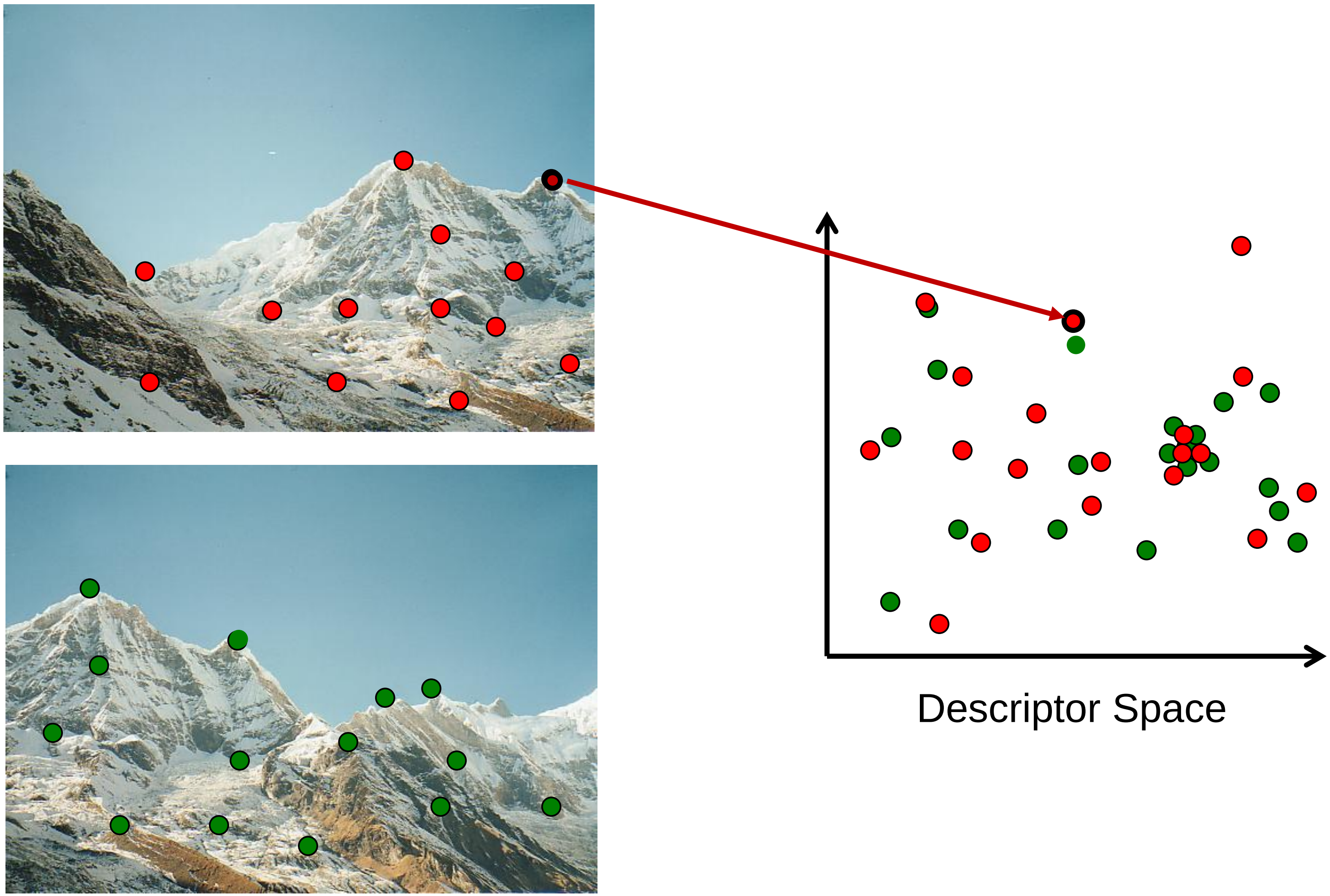
Feature Matching



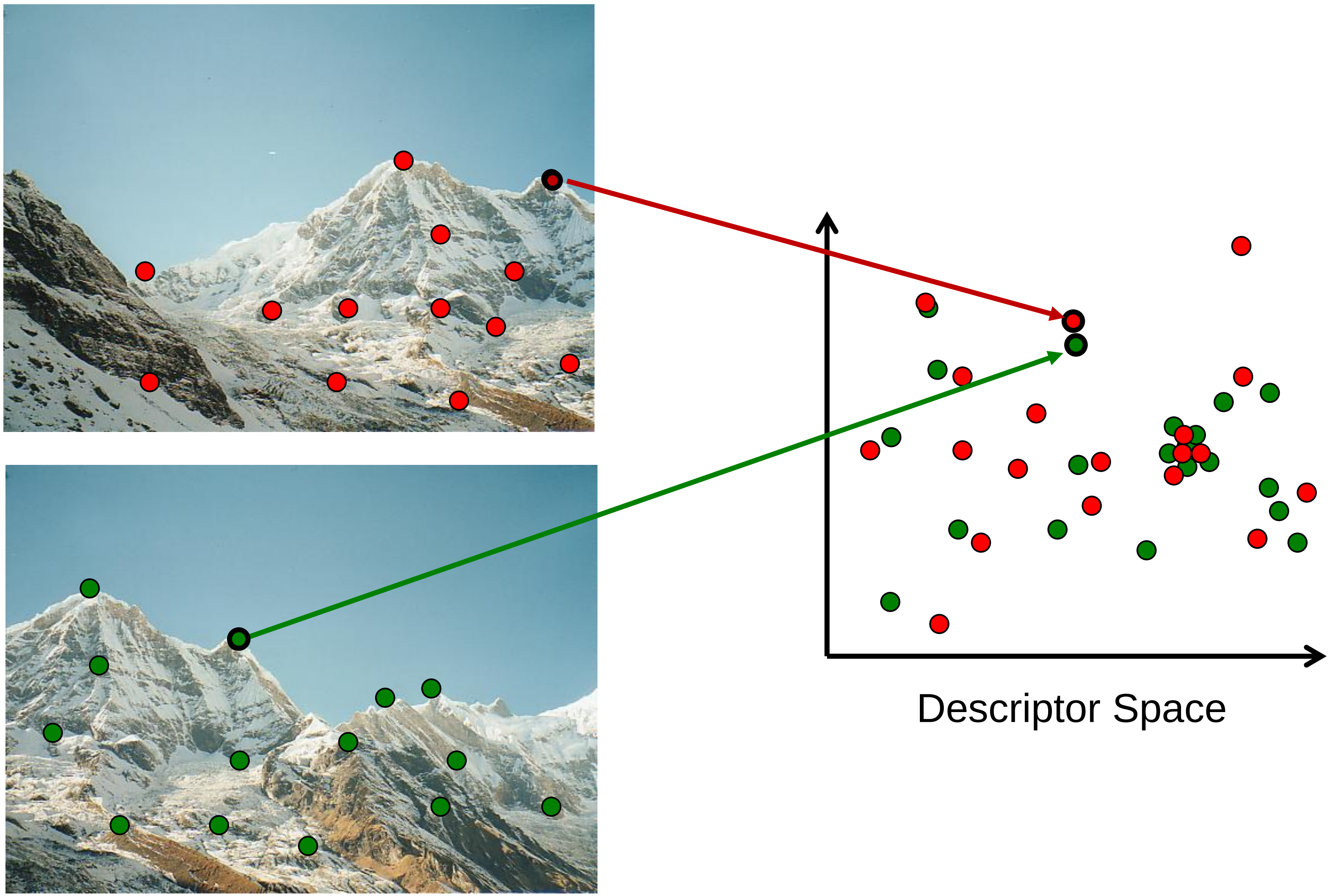
Feature Matching



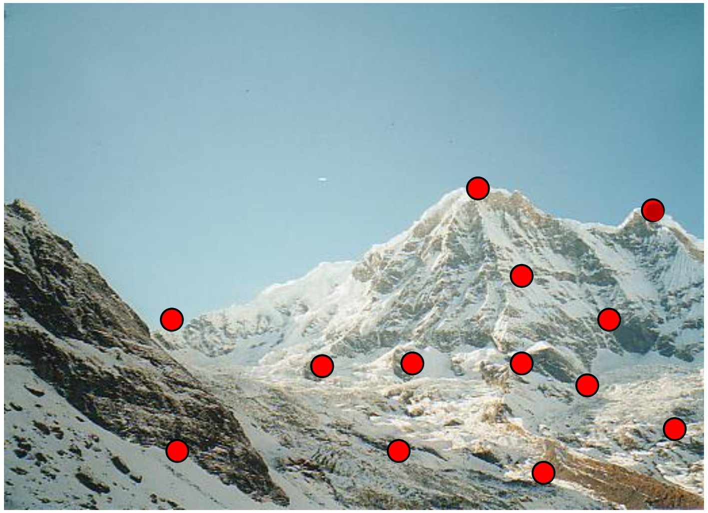
Feature Matching



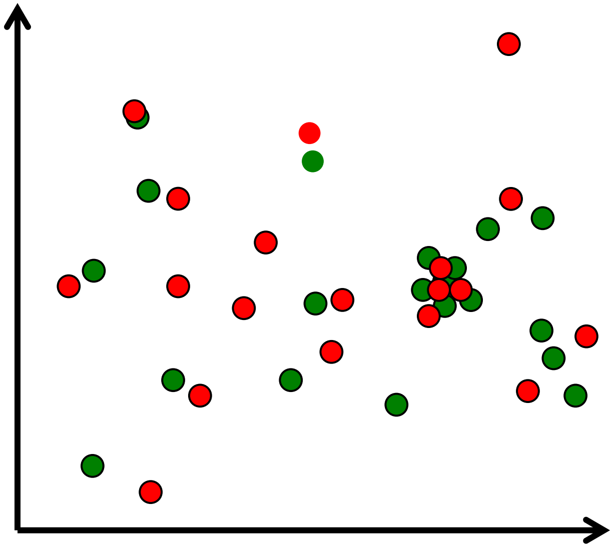
Feature Matching



Feature Matching



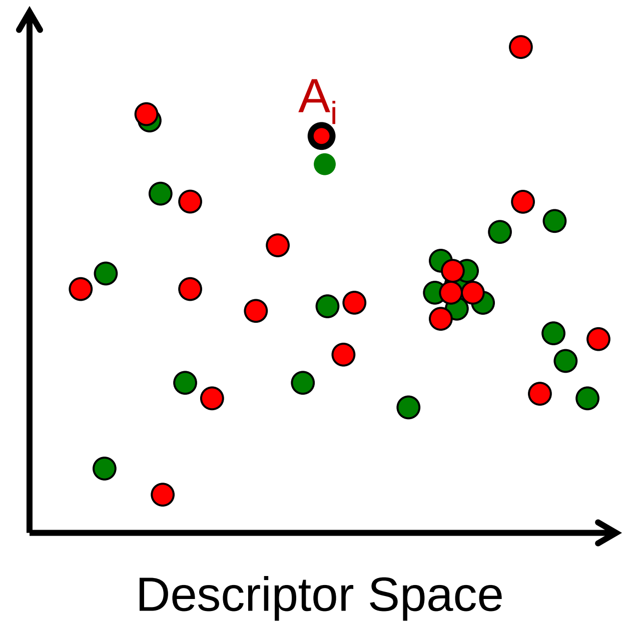
What is a good way to choose candidate matches?



Descriptor Space

Feature Matching

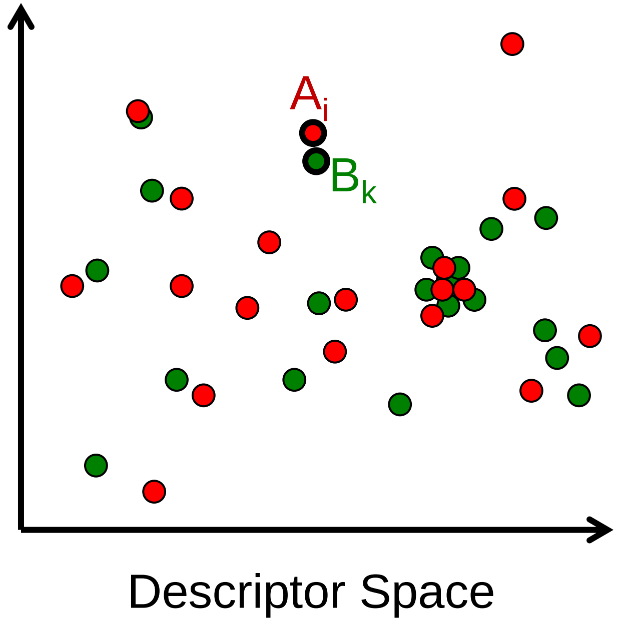
Simple method 1: create matches from each feature A_i to feature B_k in B with minimum $d(A_i,B_k)$



$d(A_i,B_j)$ is Euclidean distance in descriptor space

Feature Matching

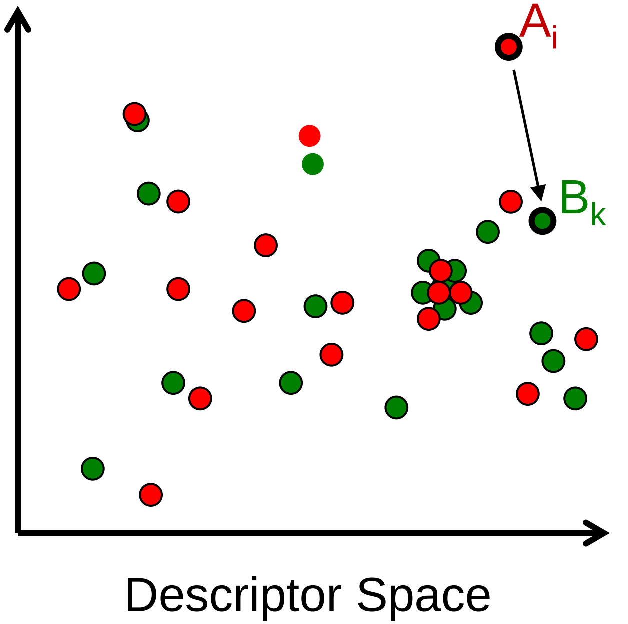
Simple method 1: create matches from each feature A_i to feature B_k in B with minimum $d(A_i, B_k)$



$d(A_i, B_j)$ is Euclidean distance in descriptor space

Feature Matching

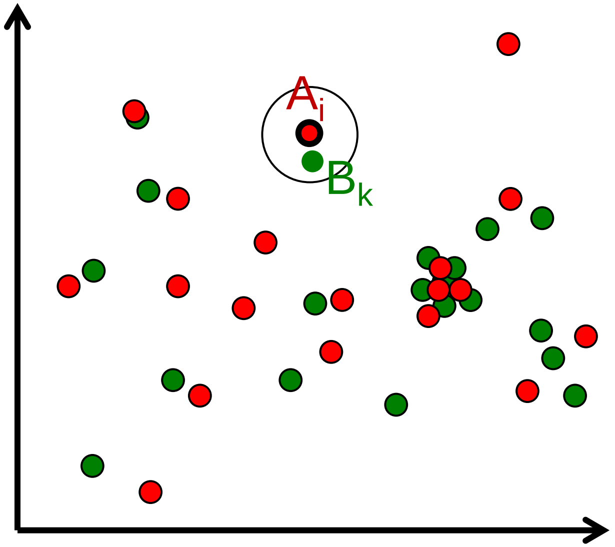
Simple method 1: create matches from each feature A_i to feature B_k in B with minimum $d(A_i,B_k)$



$d(A_i,B_j)$ is Euclidean distance in descriptor space

Feature Matching

Simple method 2: create matches from each feature A_i to feature B_k in B with minimum $d(A_i,B_k)$ iff $d(A_i,B_k)$ < threshold

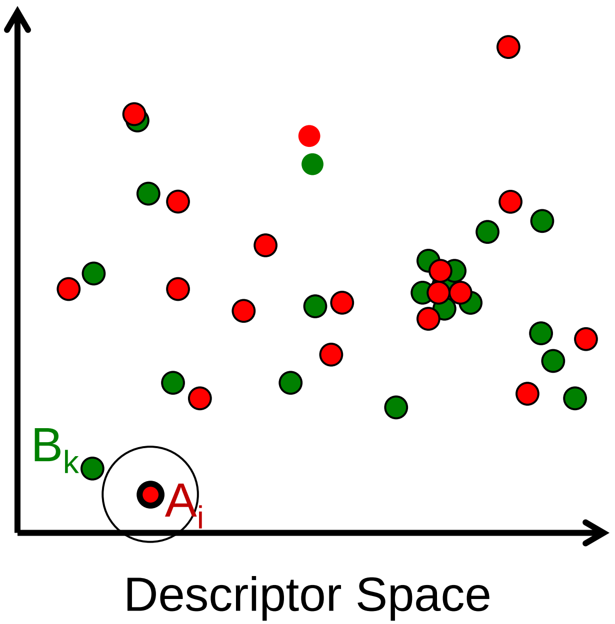


Descriptor Space

$d(A_i,B_j)$ is Euclidean distance in descriptor space

Feature Matching

Simple method 2: create matches from each feature A_i to feature B_k in B with minimum $d(A_i, B_k)$ iff $d(A_i, B_k) <$ threshold



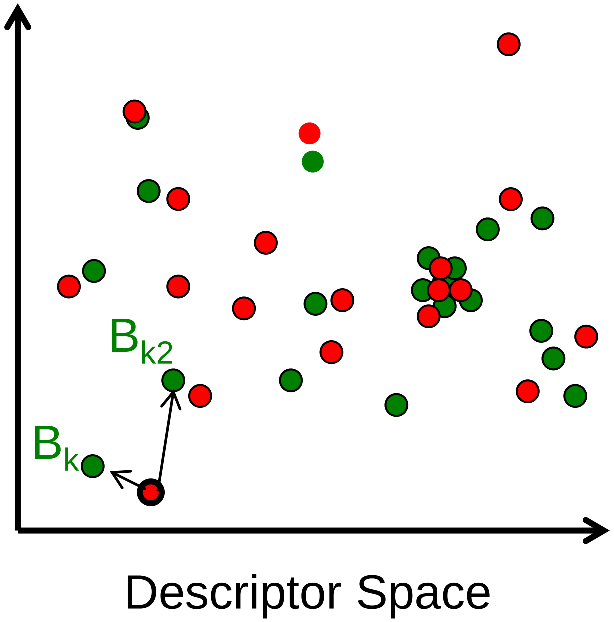
$d(A_i, B_j)$ is Euclidean distance in descriptor space

Feature Matching

Ratio method: create matches from each feature A_i to the closest feature in B with minimum $d(A_i,B_k)$ iff $d(A_i,B_k) / d(A_i,B_{k2}) < \text{threshold}$

B_k is the closest feature

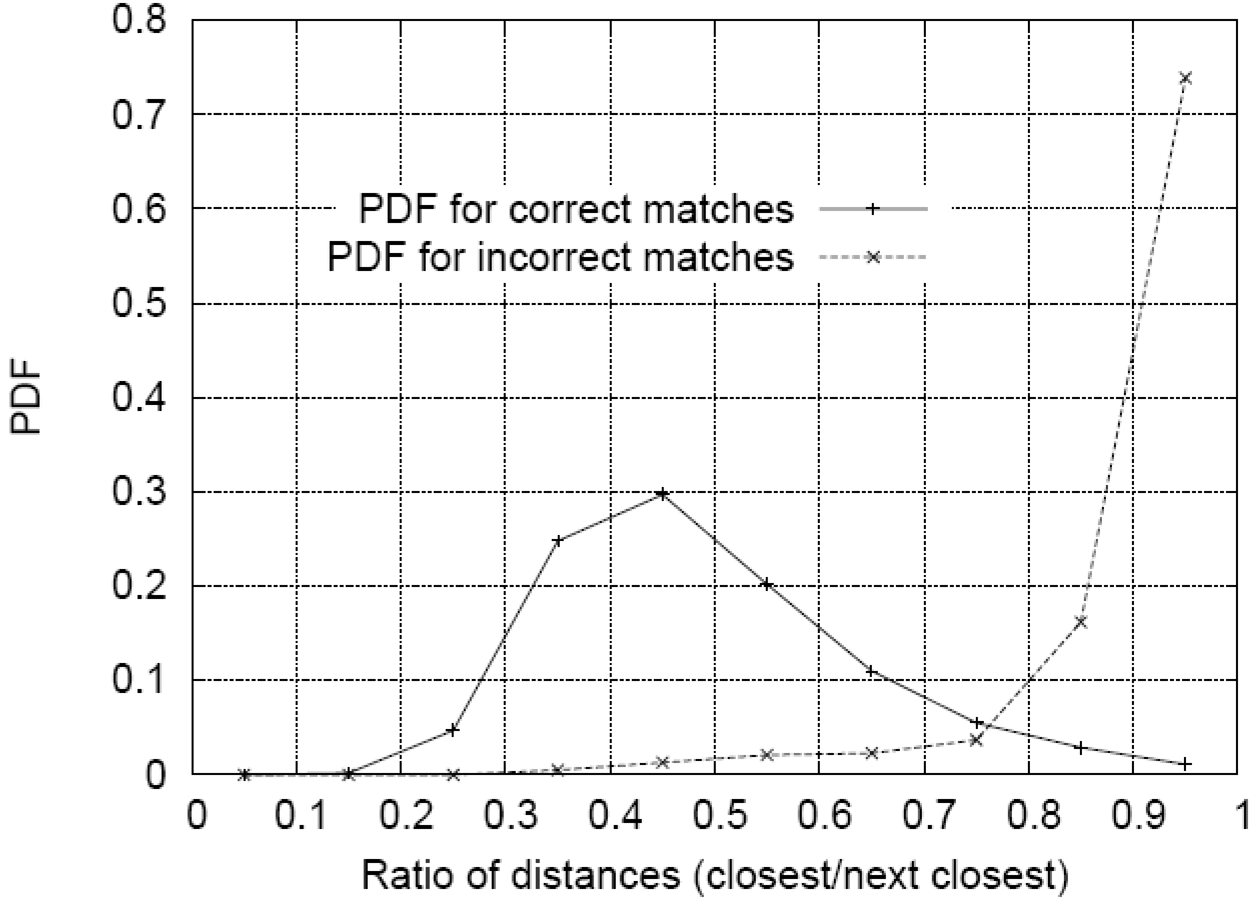
B_{k2} is the second closest feature



$d(A_i,B_j)$ is Euclidean distance in descriptor space

Feature Matching

Ratio method: threshold ratio of L2 distance to best match divided by L2 distance to 2nd best match

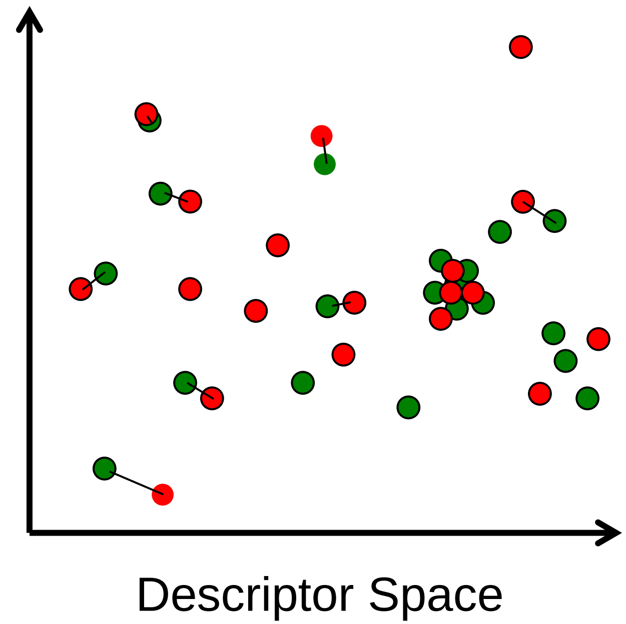


Typically threshold at ratio < 0.6

Lowe IJCV 2004

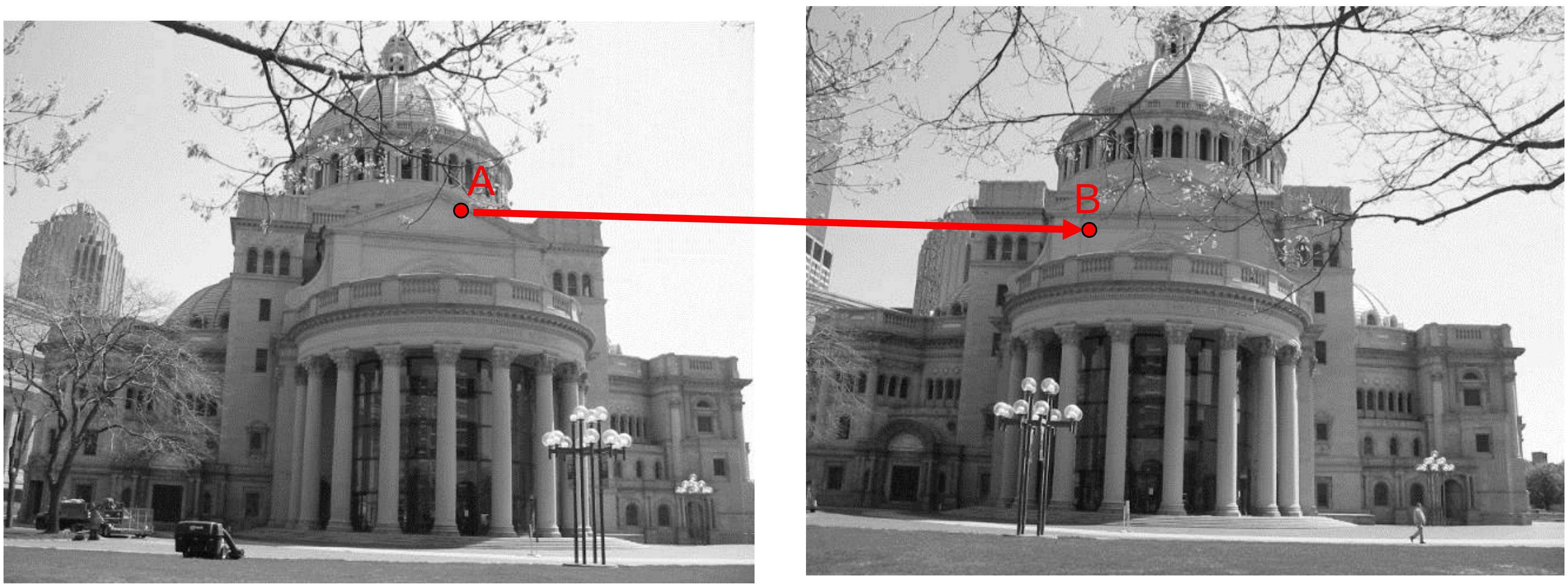
Feature Matching

Mutual closest method: create matches between features A_i and B_k iff they are mutually closest among all pairs of features



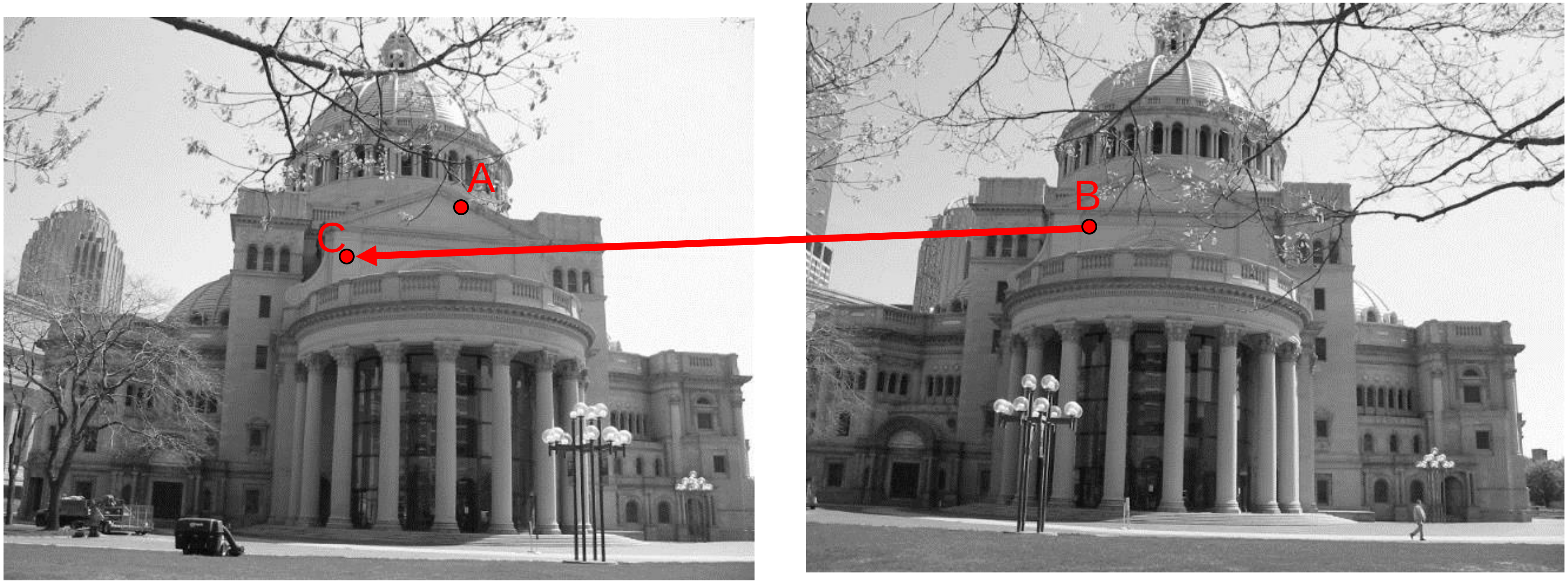
Feature Matching

Mutual closest method: create matches between features A_i and B_k iff they are mutually closest



Feature Matching

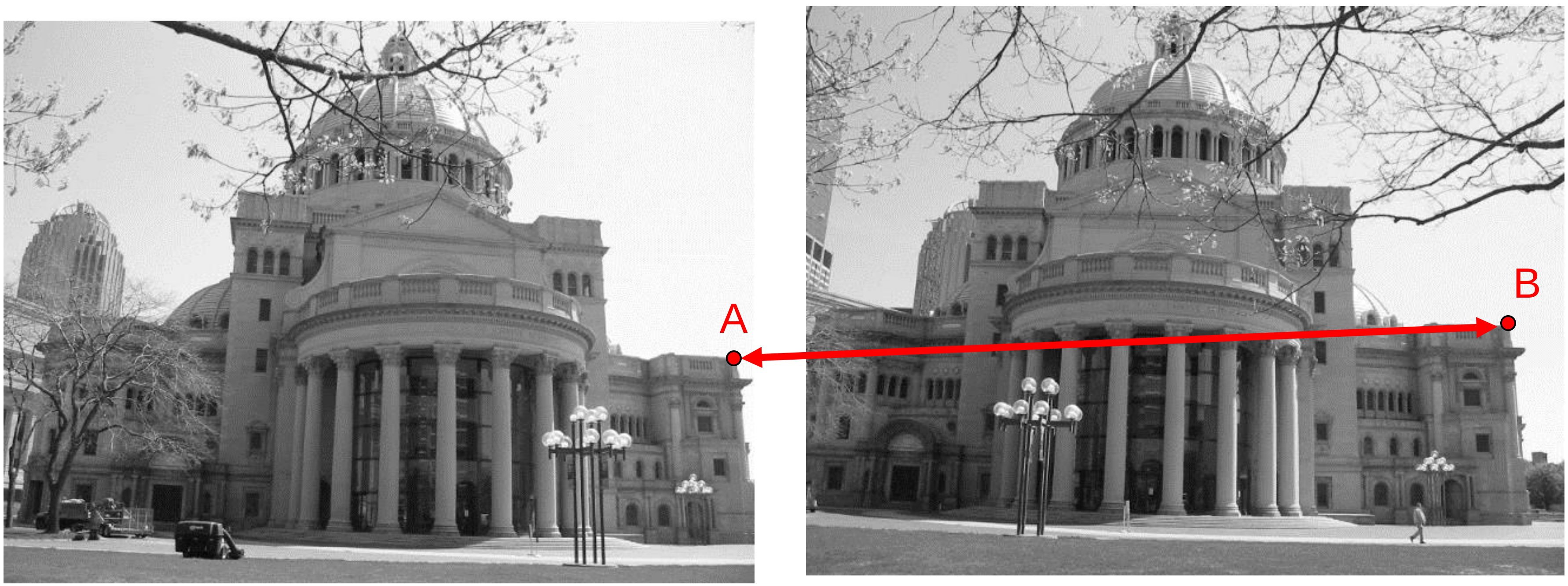
Mutual closest method: create matches between features A_i and B_k iff they are mutually closest



(A,B) is a bad match

Feature Matching

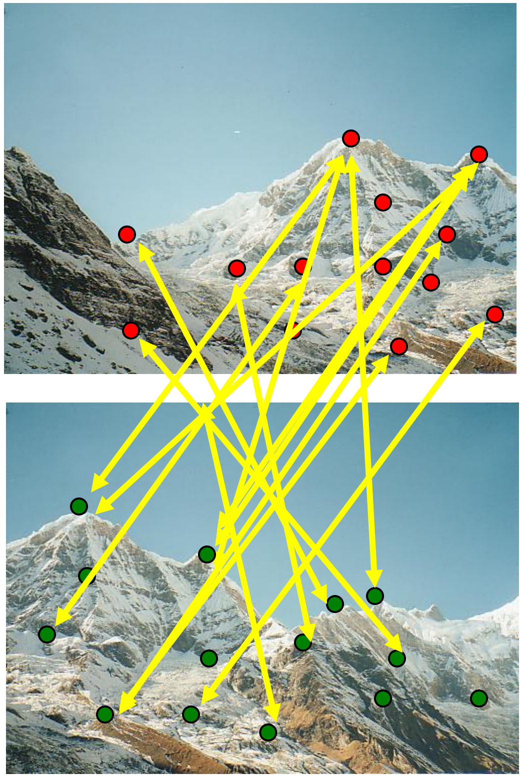
Mutual closest method: create matches between features A_i and B_k iff they are mutually closest



(A,B) is a good match

Image Mosaicing

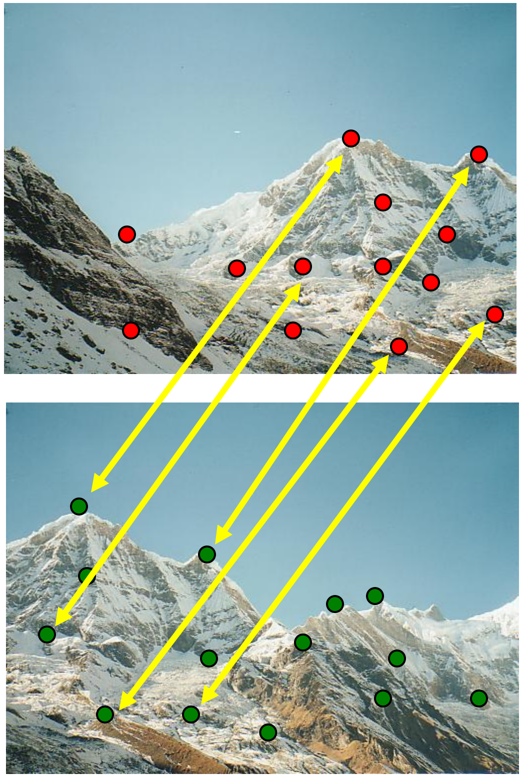
1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features



Kristen Grauman

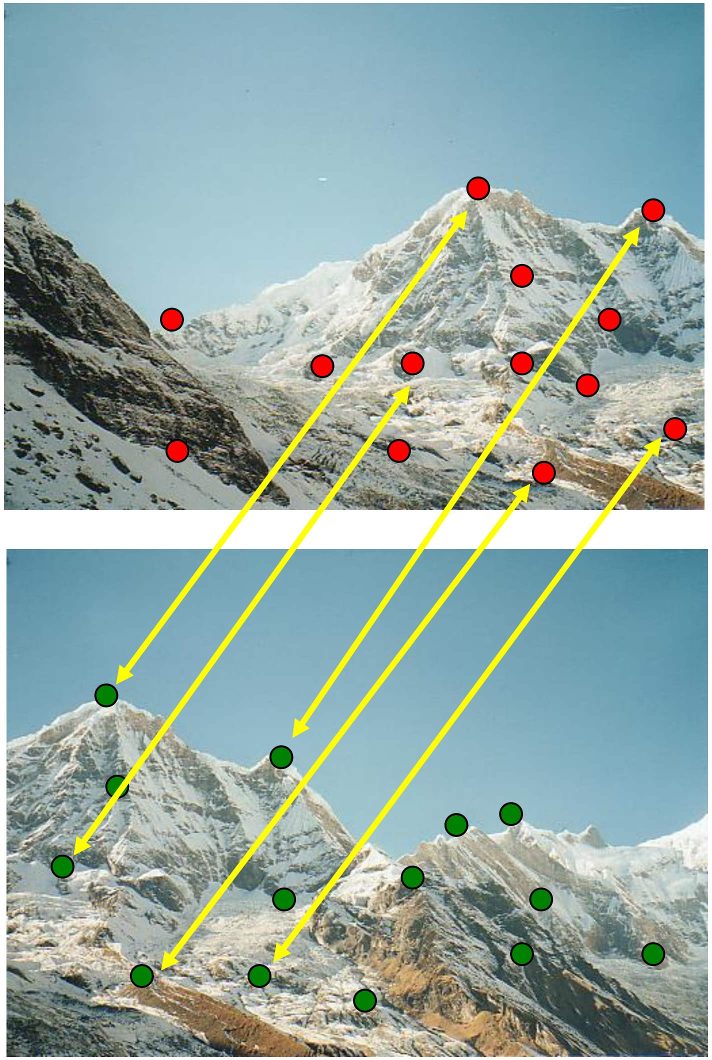
Image Mosaicing

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor for each feature
3) Feature Matching:
   Find candidate matches between features
4) Feature Correspondence:
   Find consistent set of (inlier) correspondences between features



Kristen Grauman

Feature Correspondence



Goal: produce a set of correspondences that …

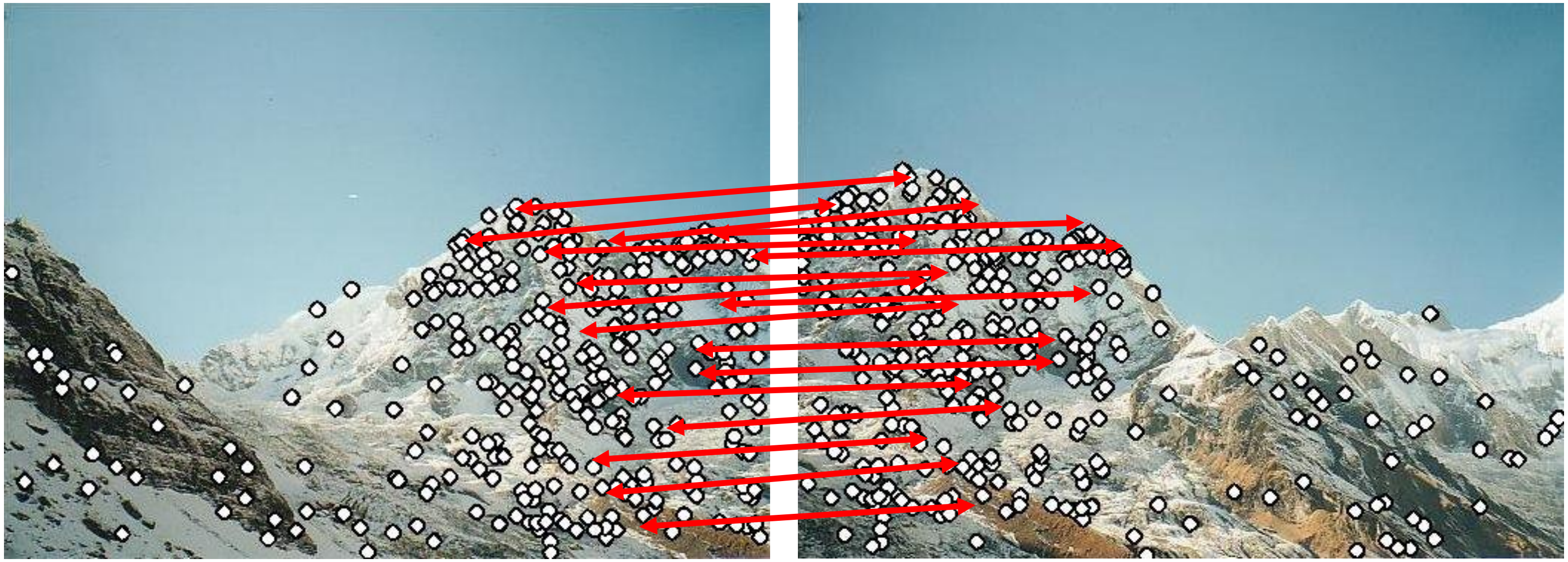
- Contains >=4 matches to define a homography
- Contains only matches that agree on the same homography
- Aligns as many matching features as possible

Feature Correspondence

Candidate matches

Source: L. Lazebnik

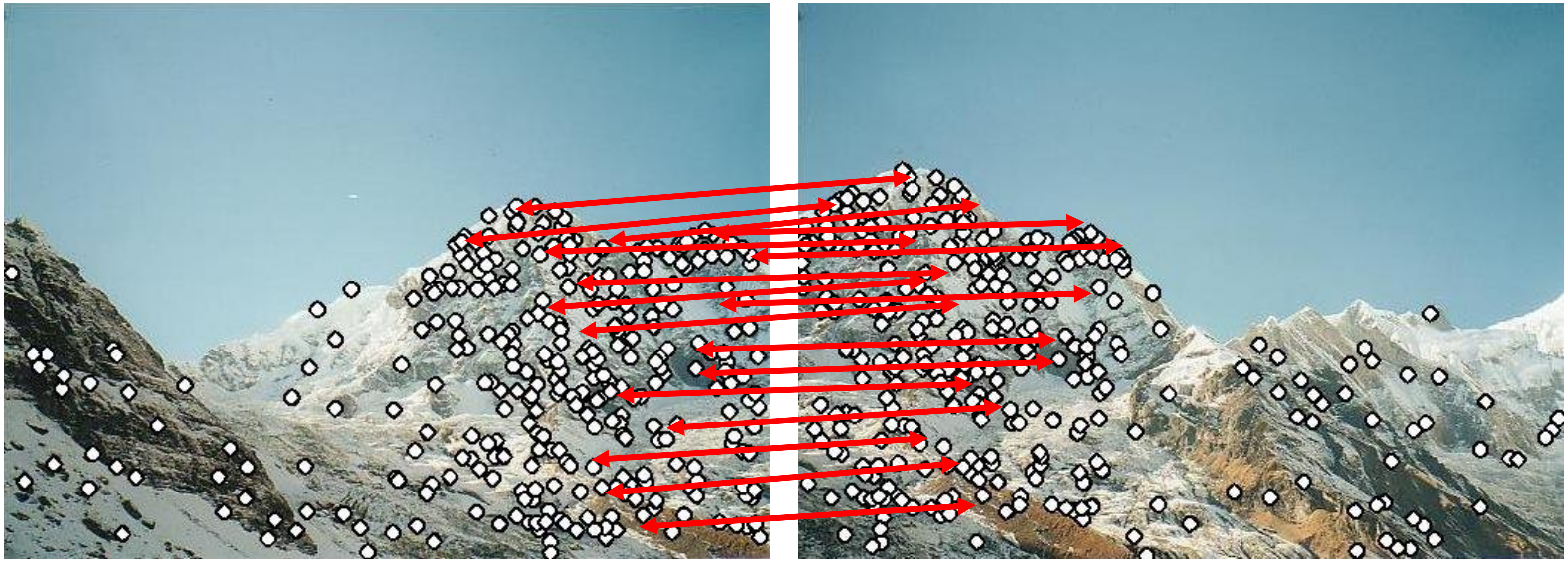
Feature Correspondence



Correspondences

Source: L. Lazebnik

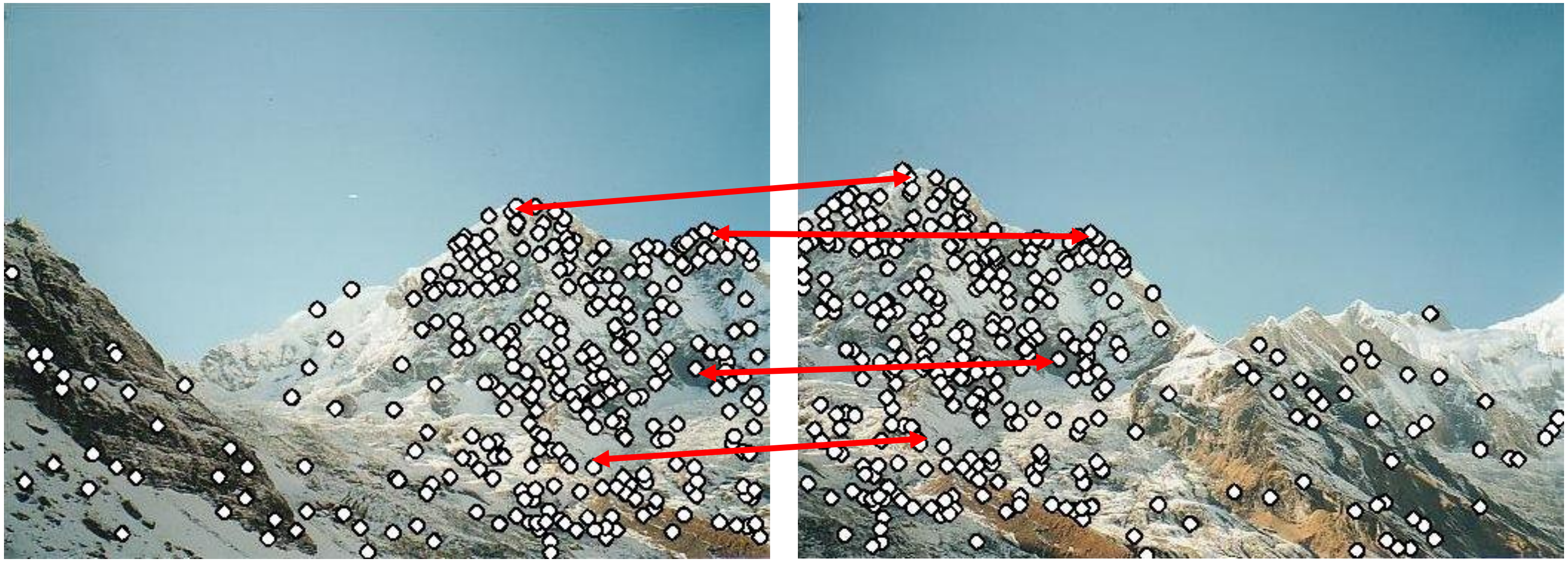
Feature Correspondence



How?

Source: L. Lazebnik

Feature Correspondence



Observation 1: *any combination of >= 4 matches defines a homography*

Source: L. Lazebnik

Feature Correspondence

Observation 1: *any combination of >= 4 matches defines a homography*

Source: L. Lazebnik

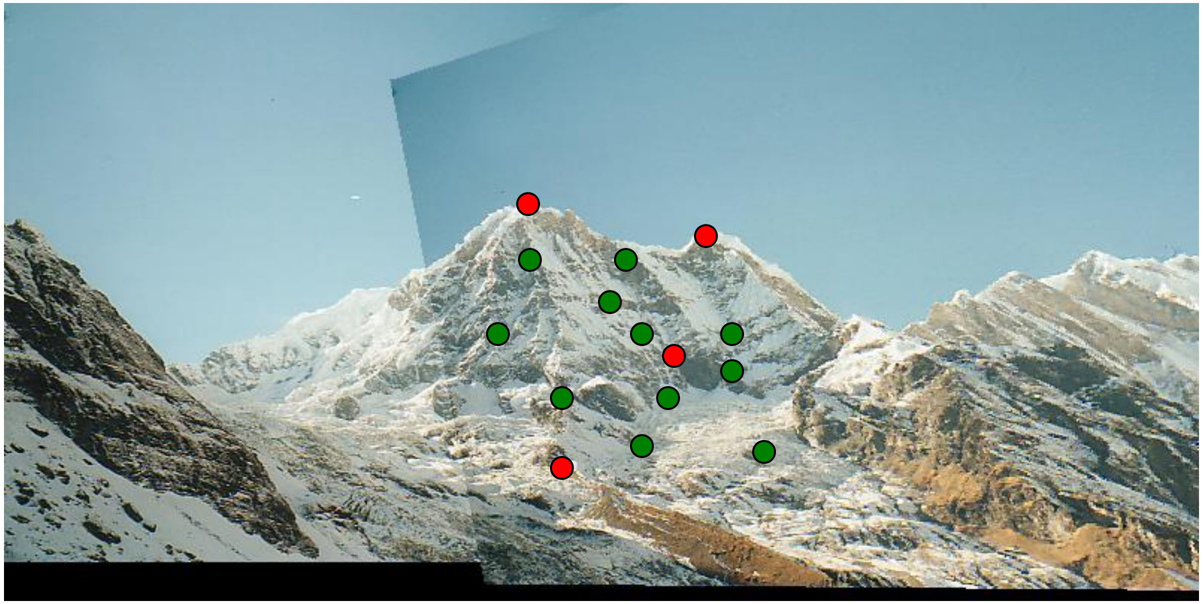
Feature Correspondence

Observation 2: every homography H provides a map

$$P' = H(p)$$

Source: L. Lazebnik

Feature Correspondence



Observation 3: can measure how “good” a map is based on how many matches are aligned by it

- generator matches: features defining the homography
- inlier matches: other features aligned by homography
- outlier matches: others (not shown)

Source: L. Lazebnik

RANSAC for Image Mosaics

RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H*

Composite images

Slide credit: Steve Seitz

RANSAC for Image Mosaics

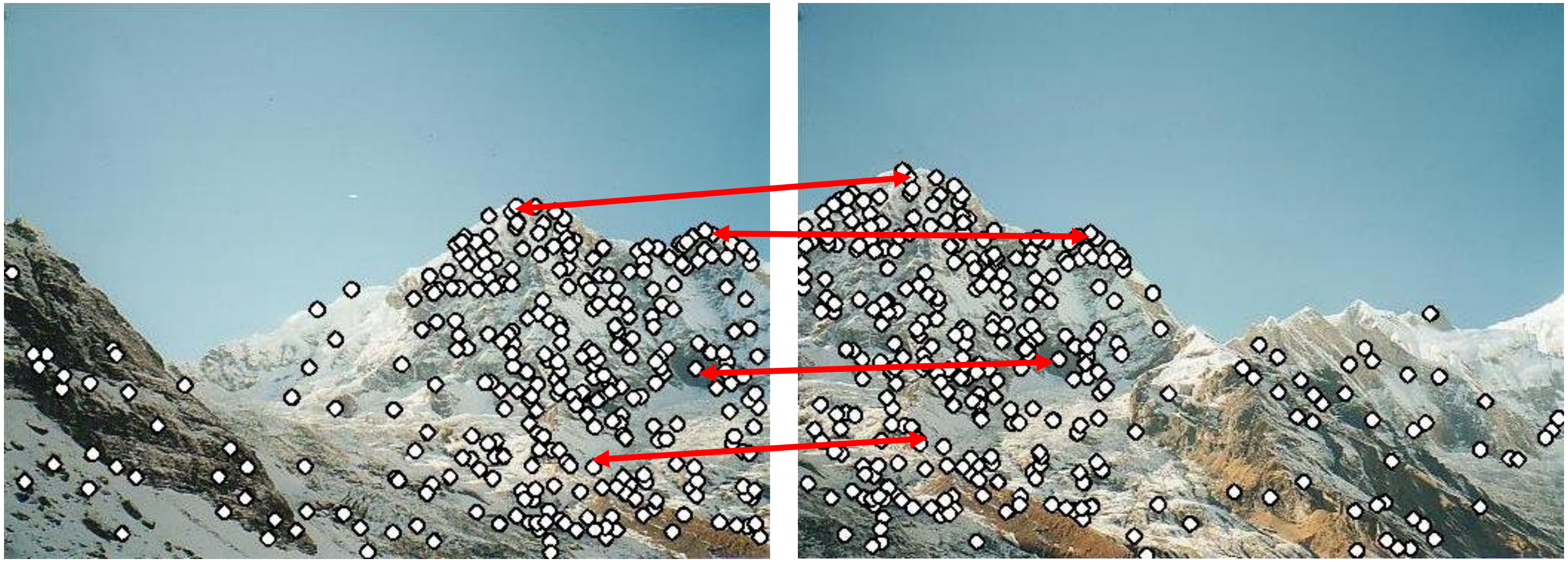
RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H} p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H} p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H* and composite images

Source: L. Lazebnik

RANSAC for Image Mosaics



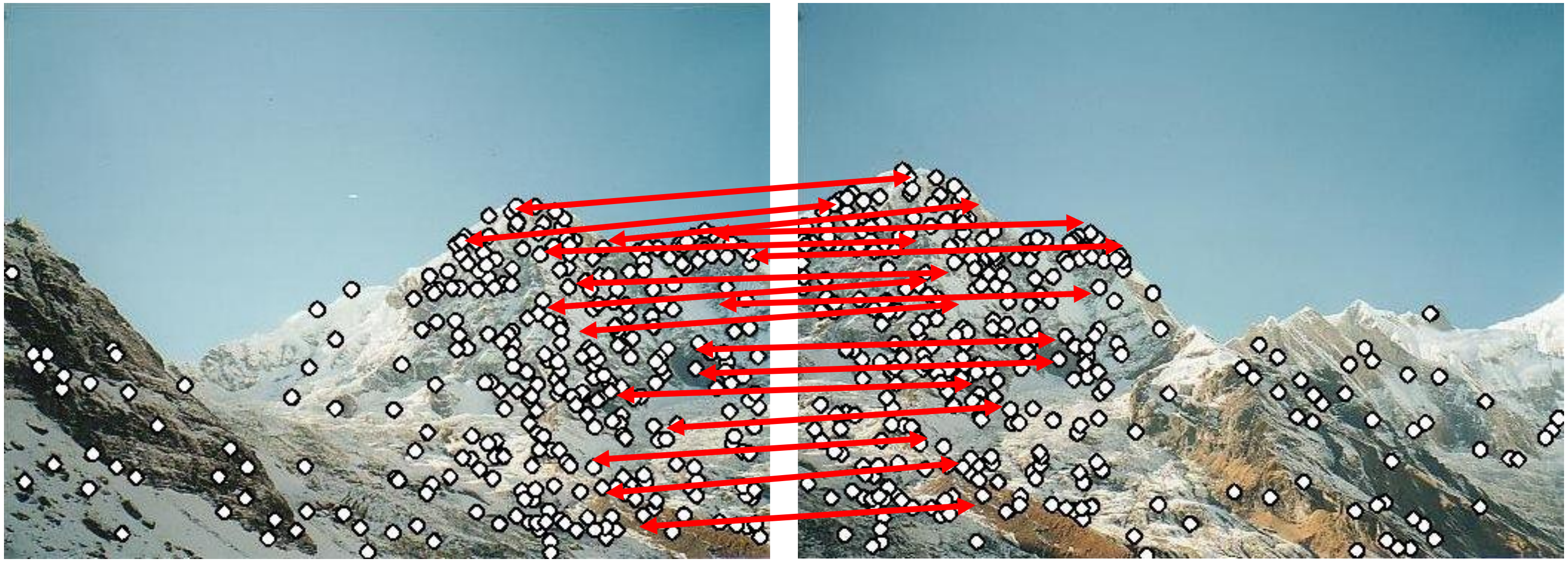
RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H} p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H} p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H* and composite images

Source: L. Lazebnik

RANSAC for Image Mosaics



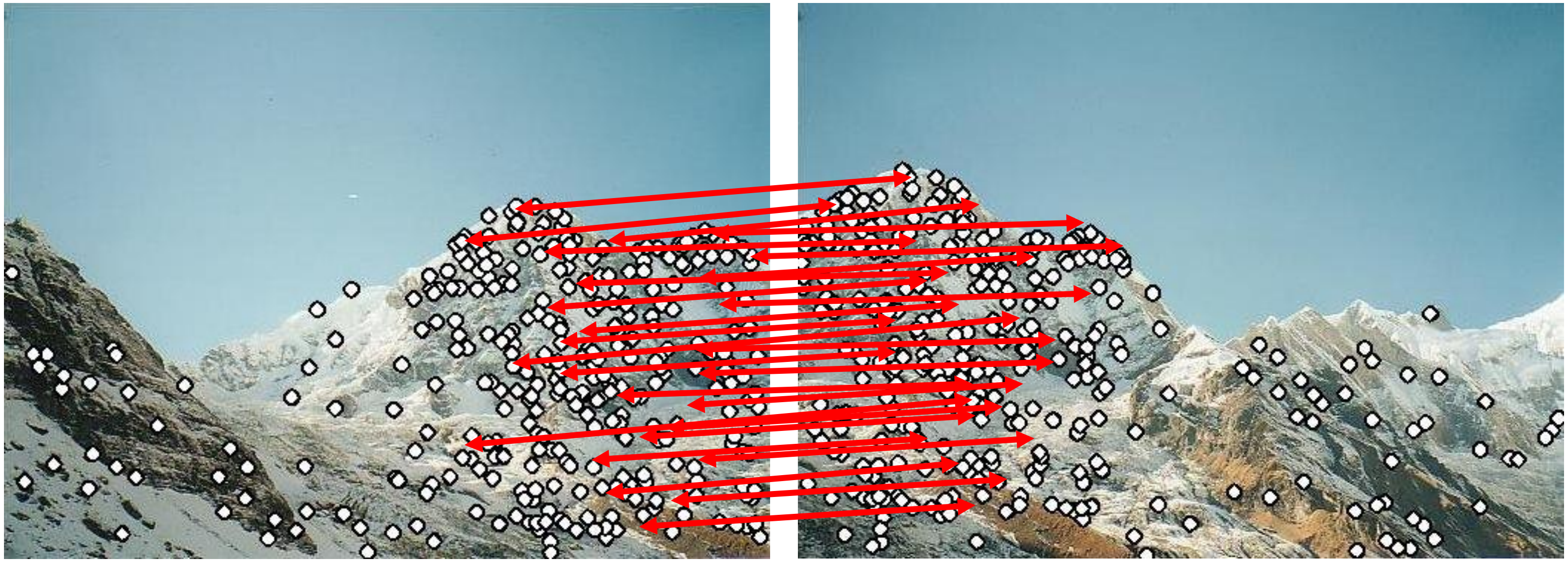
RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H* and composite images

Source: L. Lazebnik

RANSAC for Image Mosaics



RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H* and composite images

Source: L. Lazebnik

RANSAC for Image Mosaics

RANSAC loop:

1. Select four matches (at random)
2. Compute homography H aligning those matches
3. Find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
4. Re-compute H to align on all of its inliers (least squares)
5. Re-find *inlier matches* where $d(p_i', \mathbf{H}p_i) < \varepsilon$
6. H*=H if has H largest set of inliers seen so far

Warp image by H* and composite images

Source: L. Lazebnik

Image Mosaicing (summary, so far)

1) Feature Detection:
   Identify image features
2) Feature Description:
   Extract feature descriptor
   for each feature
3) Feature Matching:
   Find candidate matches
   between features
4) Feature Correspondence:
   Find consistent set of
   (inlier) correspondences
   between features

Kristen Grauman

Image Mosaicing (rest of the story)

4) Estimate homography:
    Solve linear system of equations

5) Warp one image to the other
    Apply homography transformation to every pixel

6) Composite images:
    Blend pixels in overlap area

Kristen Grauman

Image Mosaicing (rest of the story)

4) Estimate homography:

   Solve linear system of equations

5) Warp one image to the other

   Apply homography transformation to every pixel

6) Composite images:

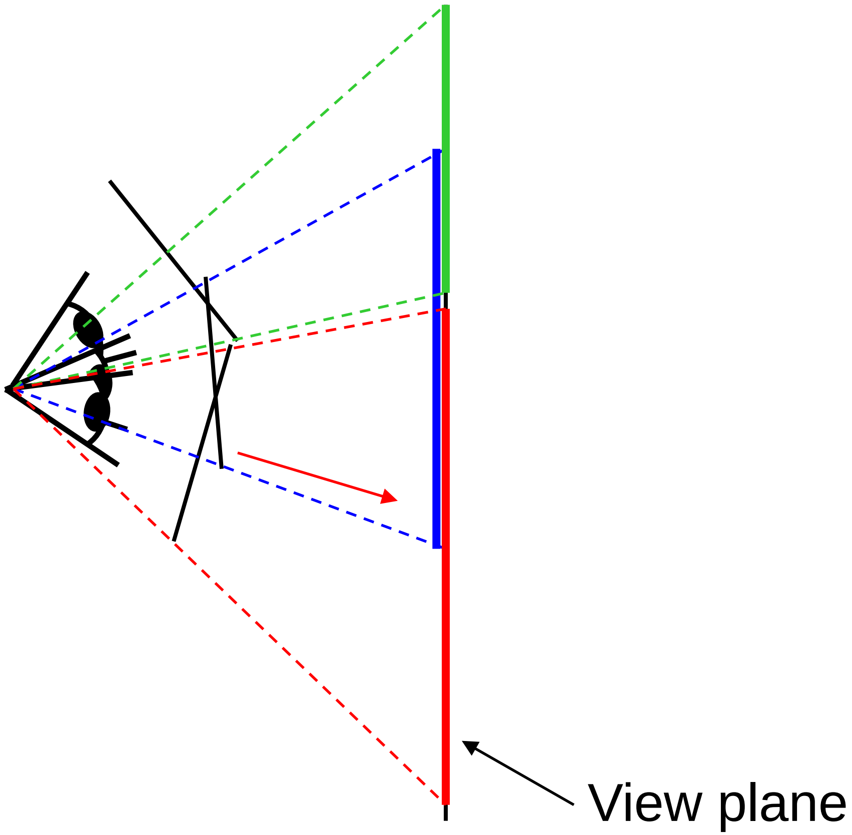
   Blend pixels in overlap area

Kristen Grauman

Estimating the Homography

A mosaic has a natural interpretation in 3D

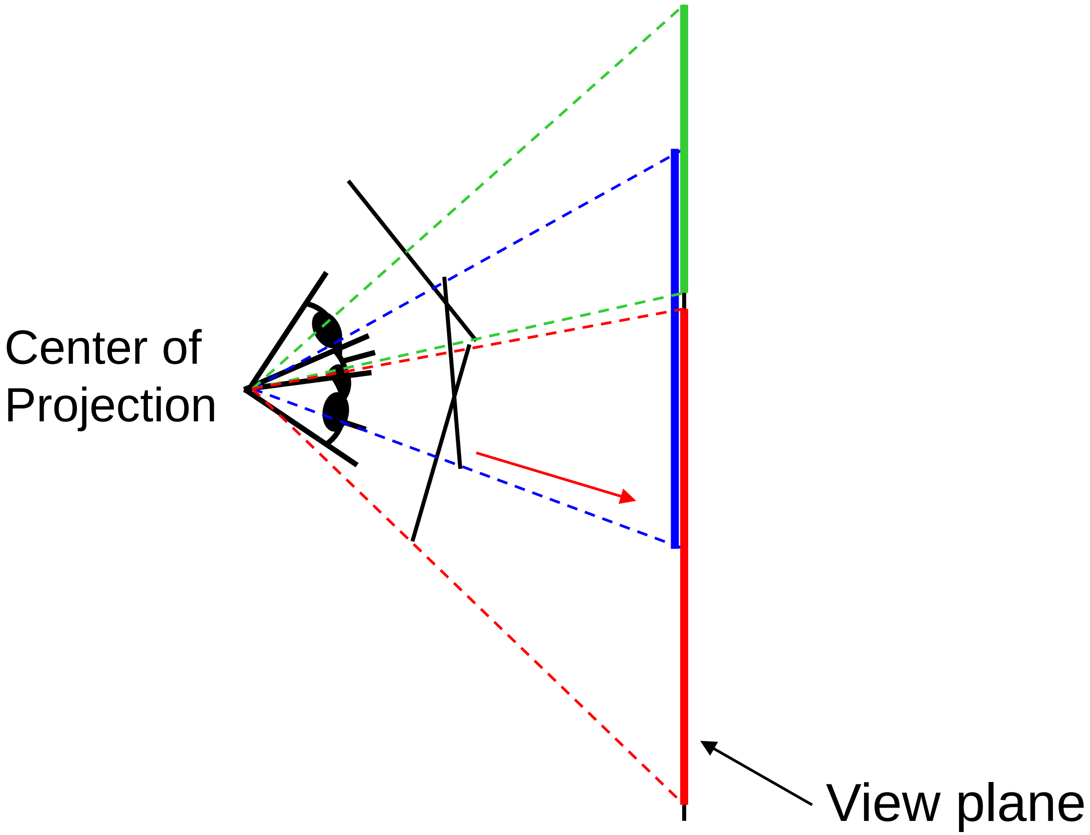
- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*



Source: Steve Seitz

Estimating the Homography

A homography is the transformation that projects an image onto a new view plane from the same viewpoint (center of projection)

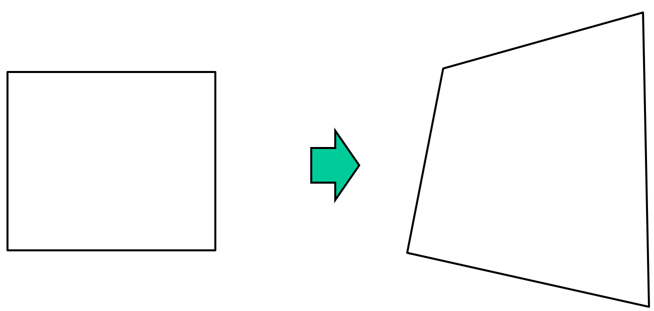


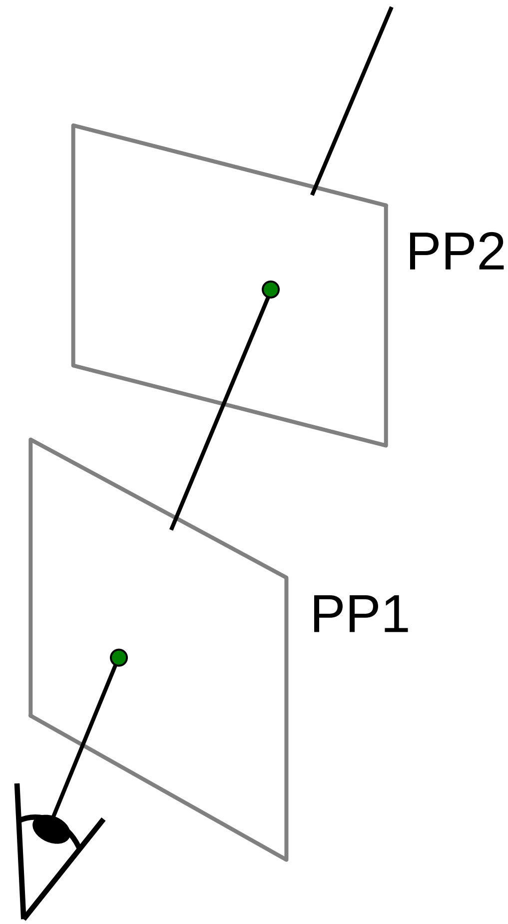
Source: Steve Seitz

What is a Homography?

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

- rectangle should map to arbitrary quadrilateral
- lines stay straight
- but don’t stay parallel





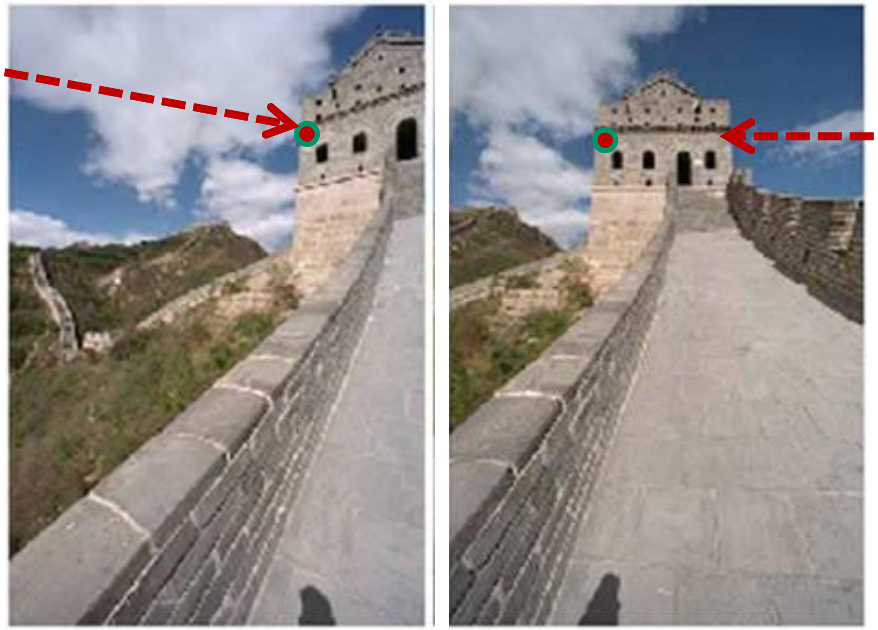
Source: Alyosha Efros

What is a Homography?

A projective transform of 2D points in homogeneous coordinates can be represented by a 3x3 matrix

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

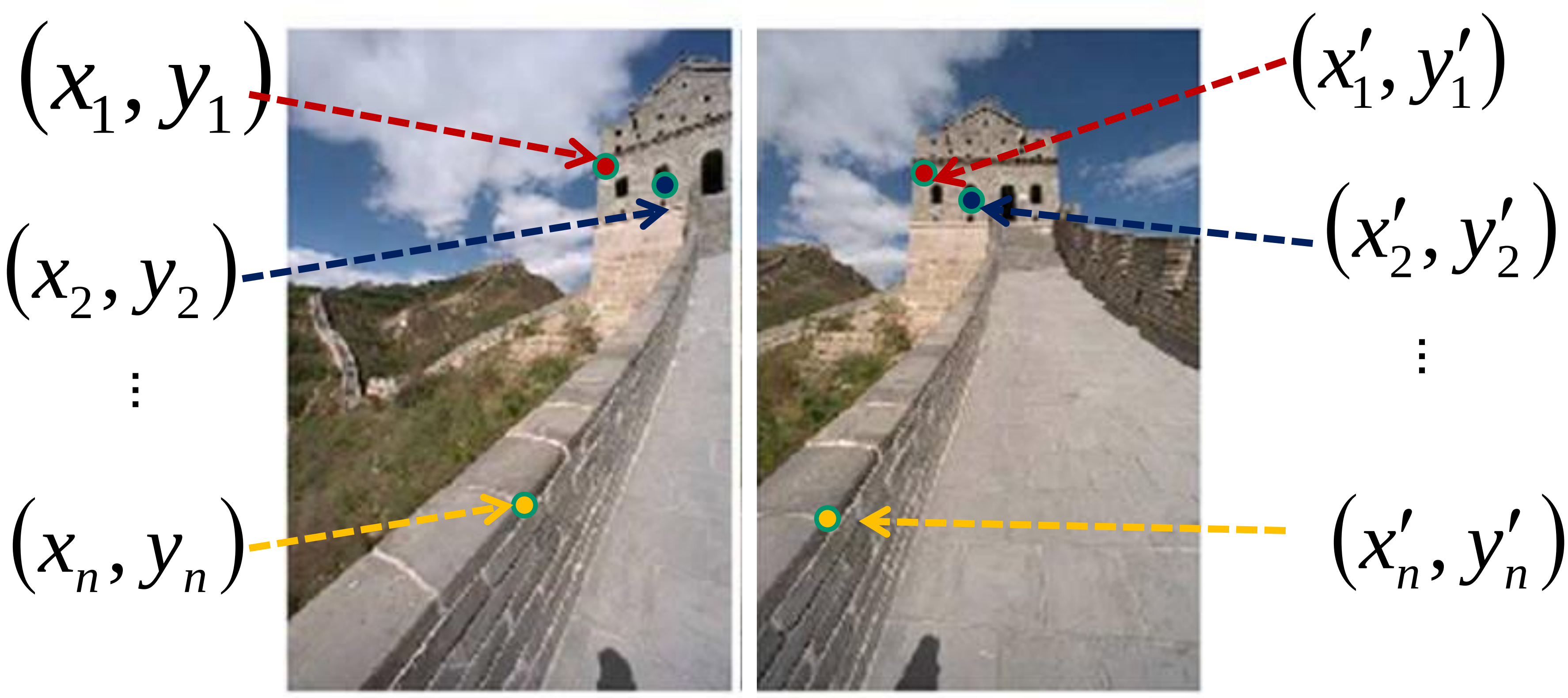
(x, y)



Homogeneous Coordinates

$$\left({}^{wx'}\!/_{w}, {}^{wy'}\!/_{w} \right) = (x', y')$$

Estimating the Homography



To **compute** the homography given pairs of corresponding points in the images, we solve a system of equations

$$\min \; \| Hp - p' \|^2$$

Image Mosaicing

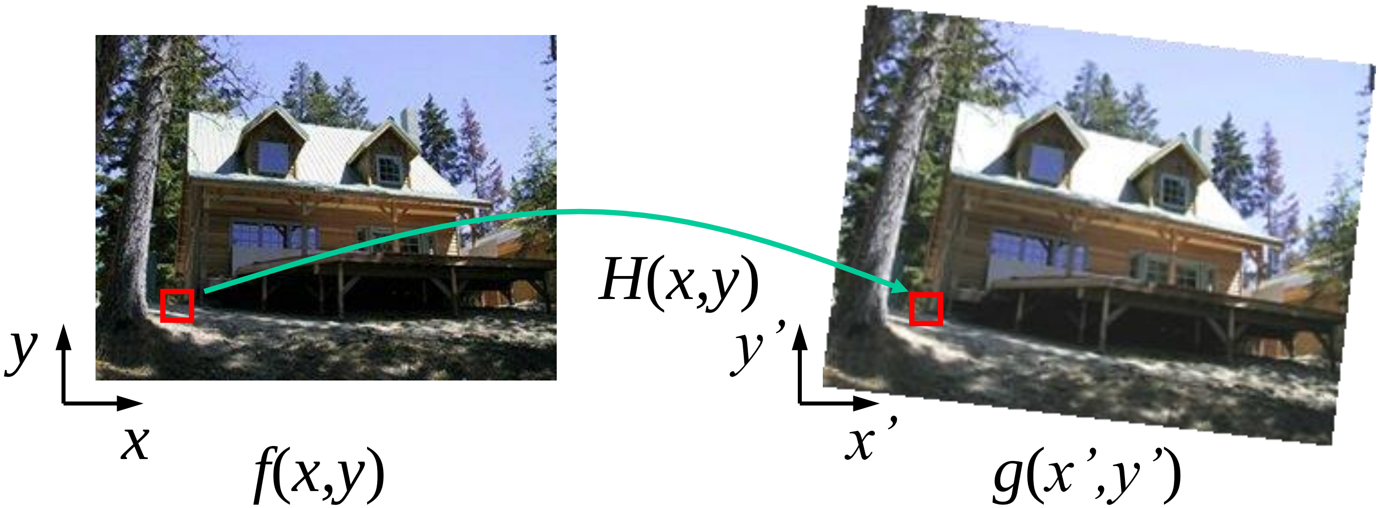
4) Estimate homography:
   Solve linear system of equations

5) Warp one image to the other
   Apply homography transformation to every pixel

6) Composite images:
   Blend pixels in overlap area

Kristen Grauman

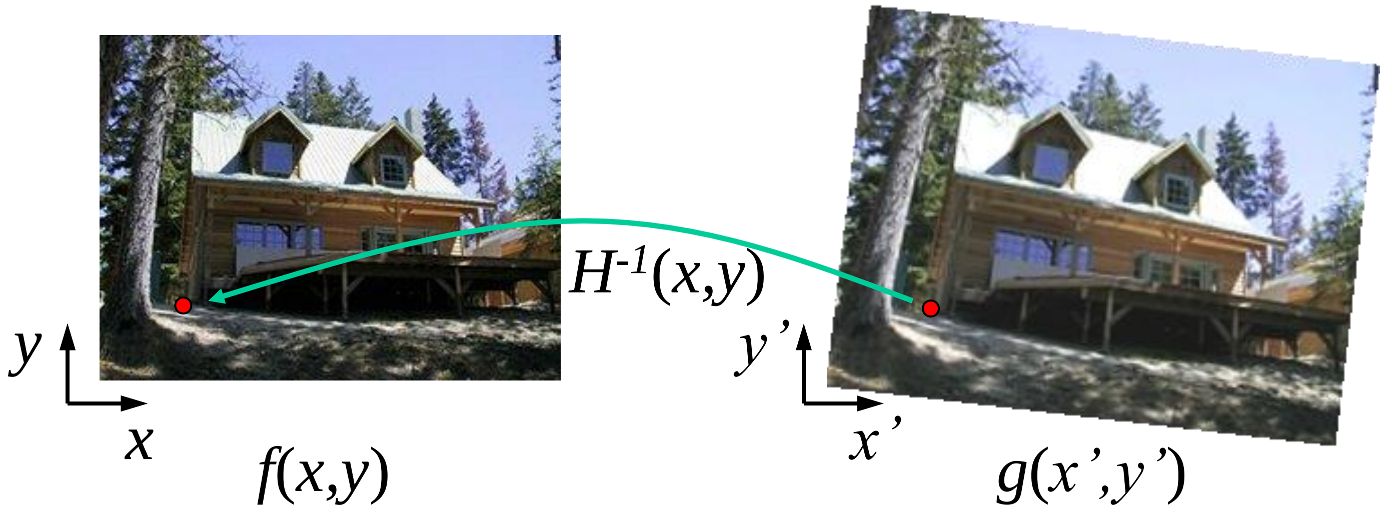
Image Warping



Given a coordinate transform H and a source image *f*(*x,y*), compute a transformed image *g(H(x,y))* = *f*(*x,y*)?

Slide from Alyosha Efros, CMU

Image warping



Reverse mapping:

Get each pixel $g(x',y')$ from its corresponding location $(x,y) = H^{-1}(x',y')$ in the source image $f(x,y)$

Slide from Alyosha Efros, CMU

Image Mosaicing

4) Estimate homography:
   Solve linear system of equations

5) Warp one image to the other
   Apply homography transformation to every pixel

6) Composite images:
   Blend pixels in overlap area

Kristen Grauman

Compositing Images

Goal: merge two overlapping images

How?

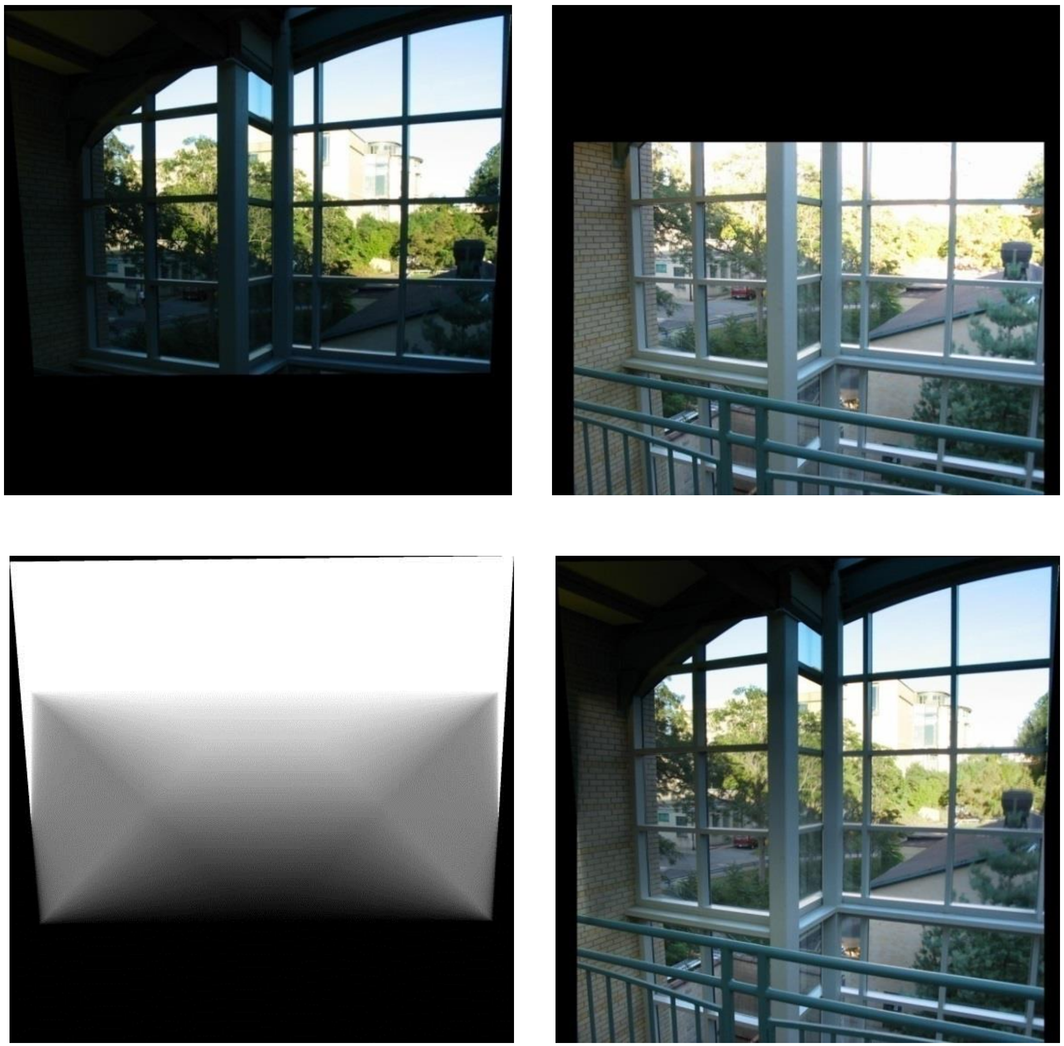
Compositing Images

Simplest method: overlay one image over the other

Slide credit: A. Efros

Compositing Images

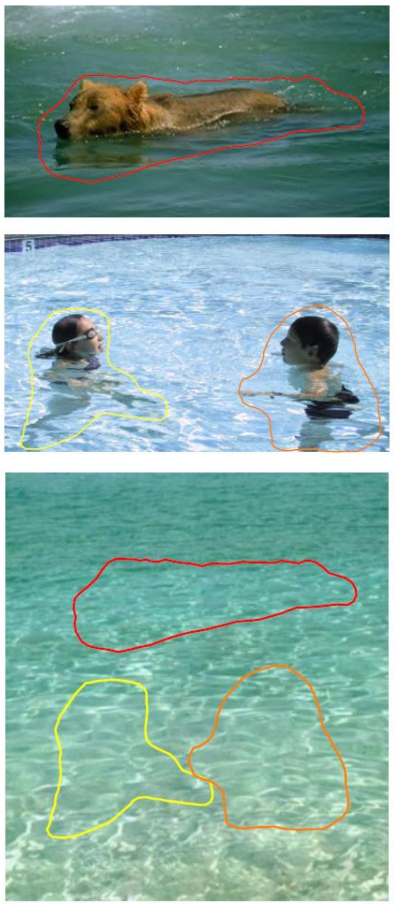
Small improvement: blend in overlap area



Slide credit: A. Efros

Compositing Images

Better improvement: blend gradients rather than colors



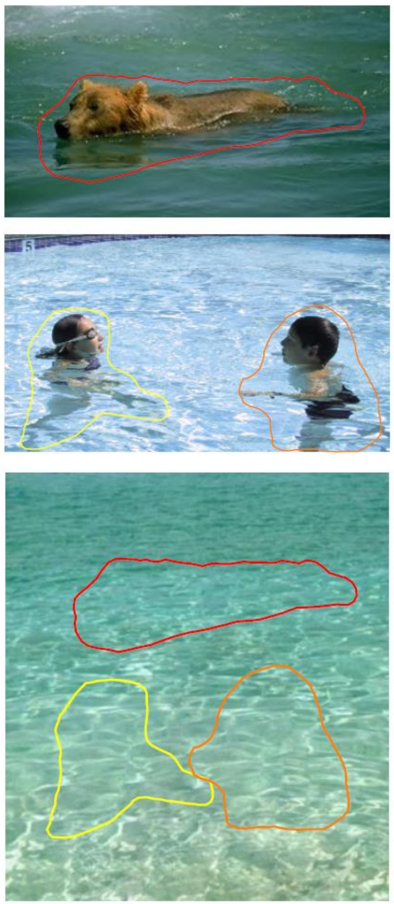
sources/destinations

Composition

Slide credit: F. Durand

Compositing Images

Better improvement: blend gradients rather than colors



sources/destinations

Composition

Gradient-domain Composition

Slide credit: F. Durand

Compositing Images

Even better improvement: use graph cut to find seam blending across in gradient domain

Slide credit: A. Efros

Compositing Images

Even better improvement: use graph cut to find seam blending across in gradient domain

Slide credit: A. Efros

Compositing Images

That’s a whole lecture for another time …

Slide credit: A. Efros

Summary for Assignment 2

Feature detection

- Harris corner or SIFT

Feature description

- Window or SIFT

Feature matching

- Ratio

Feature correspondence

- RANSAC

Homography estimation

- cp2tform

Image warping

- imtransform

Image composition

- Overlay

Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros

SIGGRAPH 2007

Slides by J. Hays and A. Efros

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Texture synthesis result

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Image Completion

?

Hays et al. SIGGRAPH 07

Image Completion

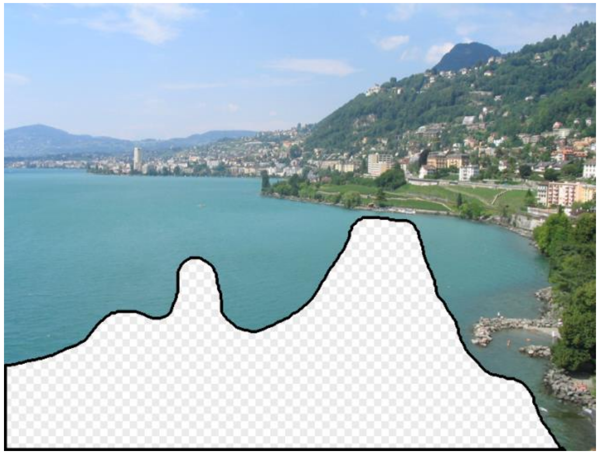
<u>2.3 Million</u> unique images from Flickr

Hays et al. SIGGRAPH 07

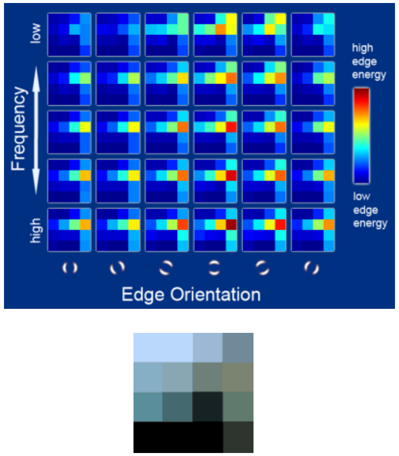
Scene Completion Result

Hays et al. SIGGRAPH 07

Image Completion Algorithm



Input image



Scene Descriptor

Image Collection

…

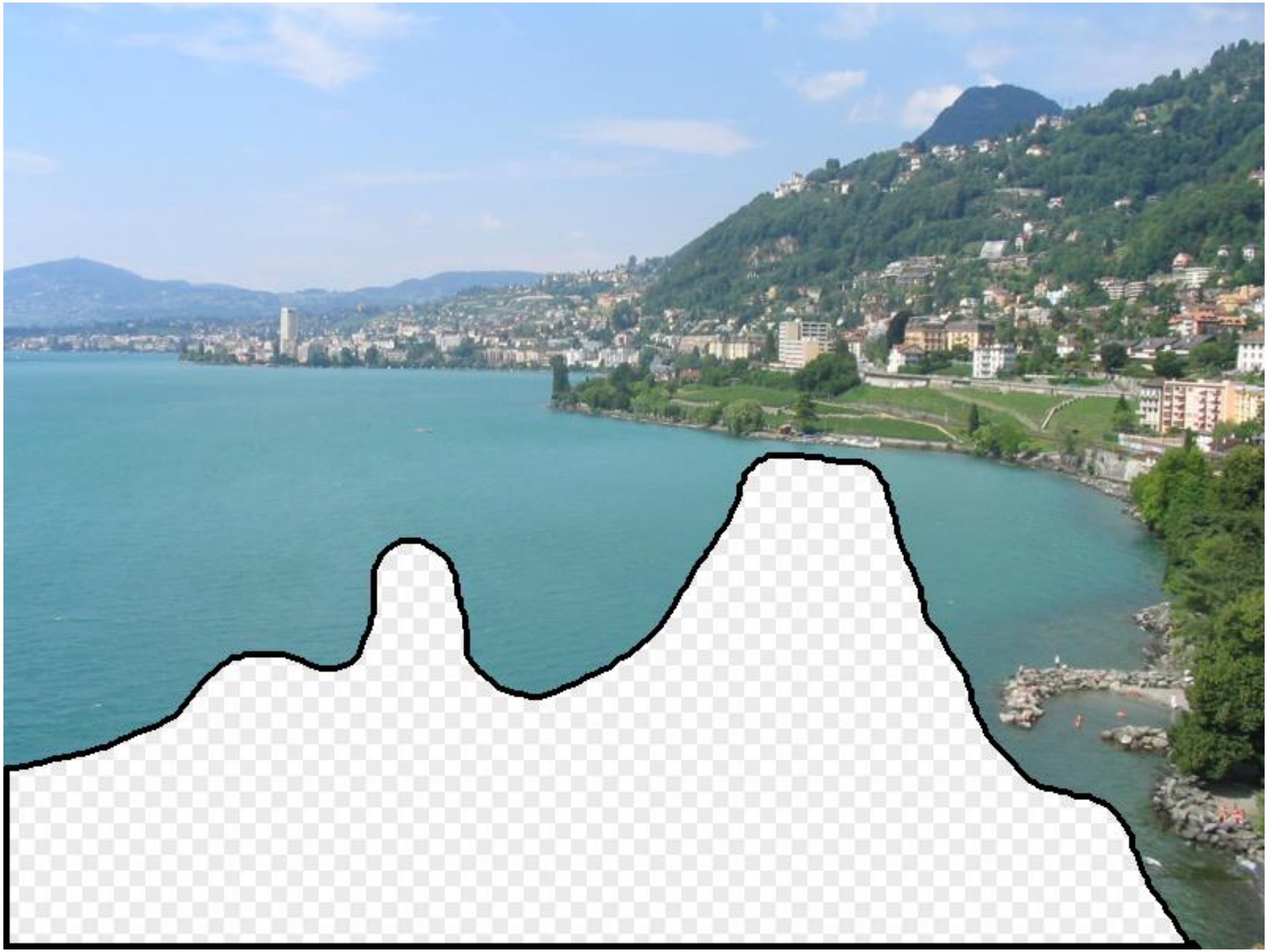
200 matches

…

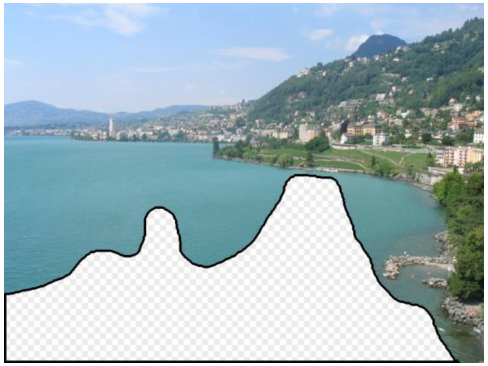
Mosaicing

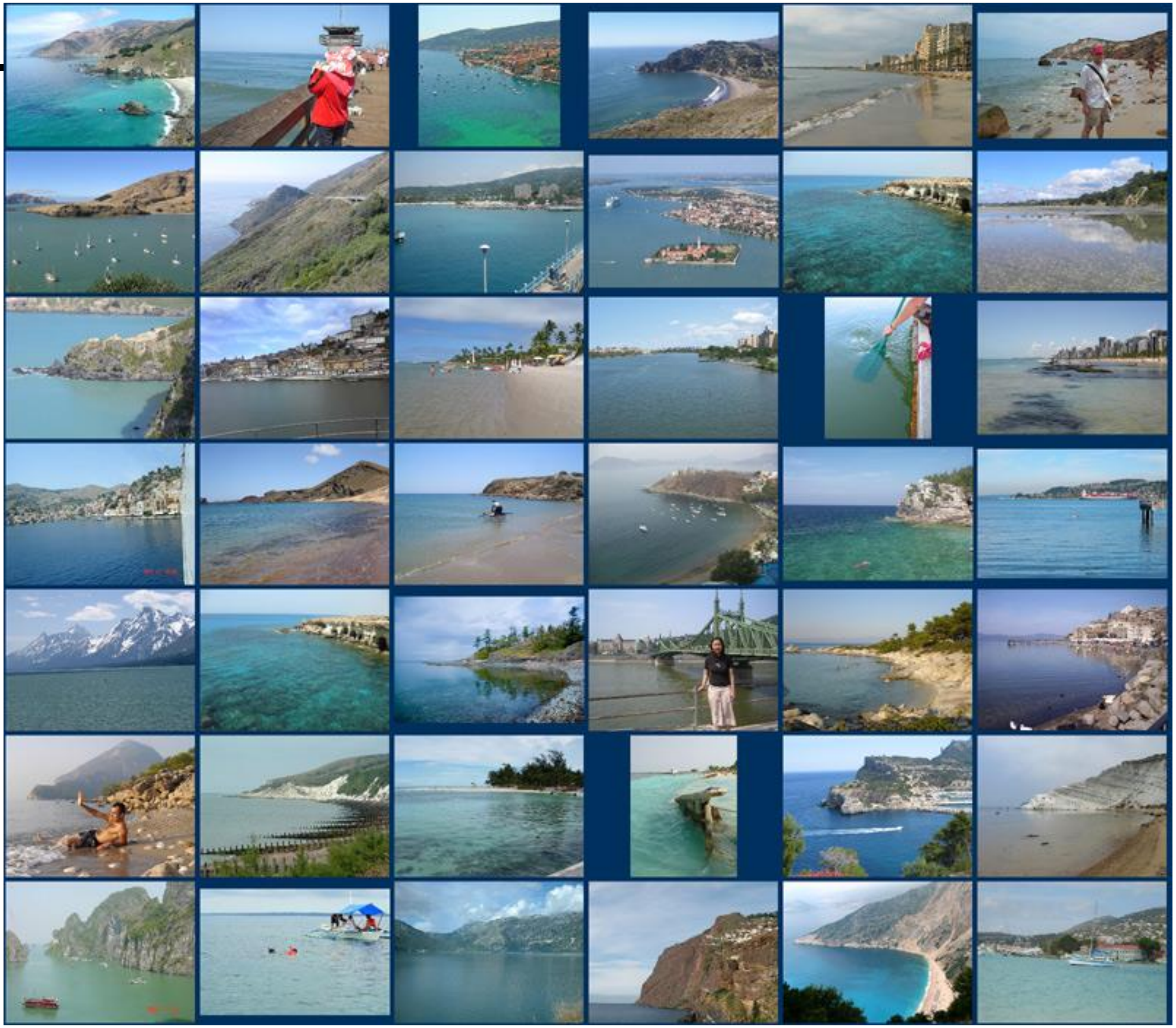
20 completions

Image Completion



Hays et al. SIGGRAPH 07

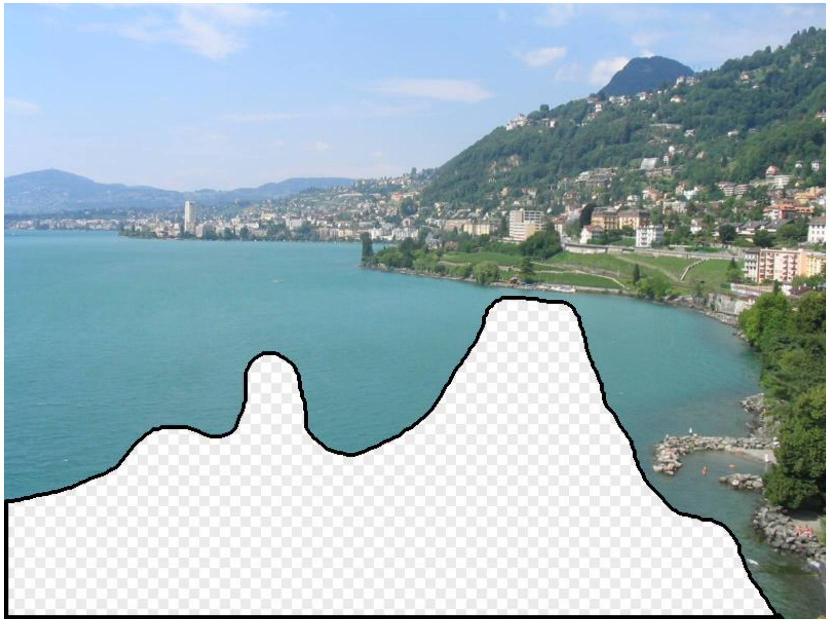




… 200 best matches

Hays et al. SIGGRAPH 07

Image Completion



Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Image Completion Result

Hays et al. SIGGRAPH 07

Image Completion Results

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07