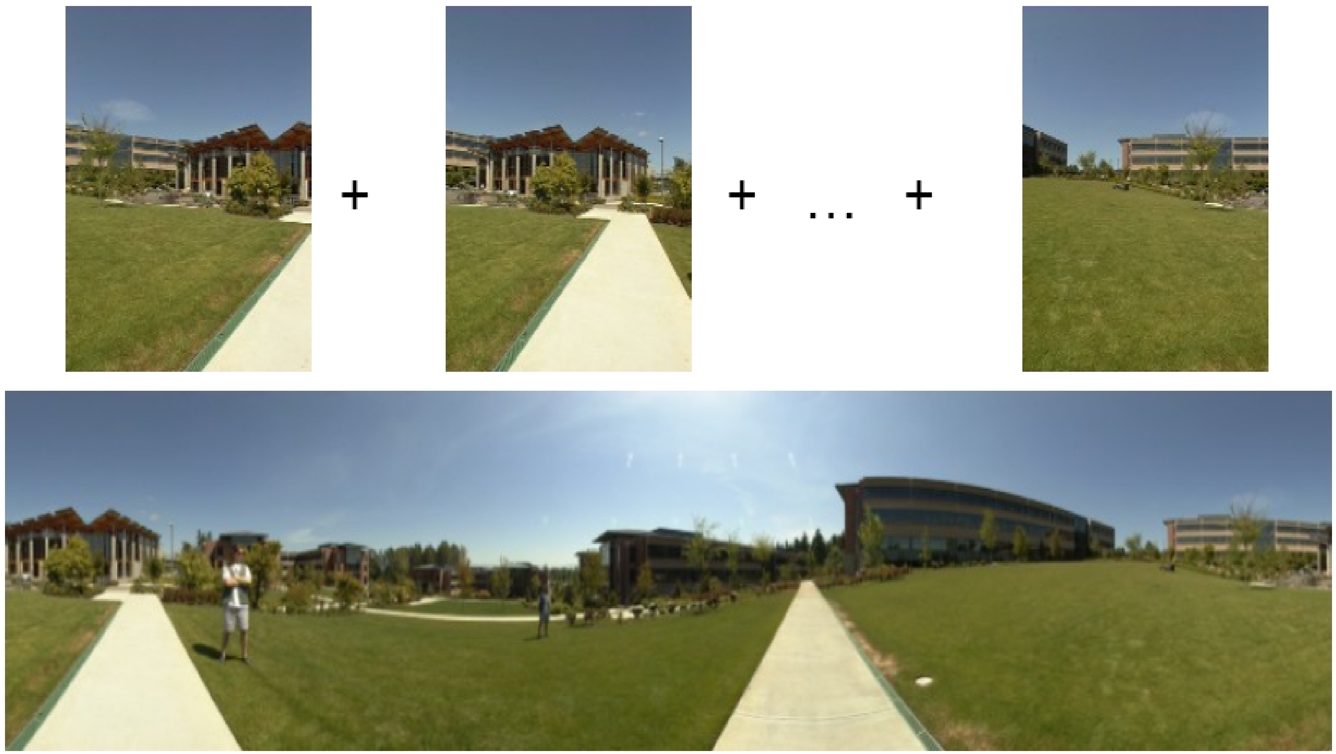
Mosaics

CS 554 – Computer Vision
Pinar Duygulu
Bilkent University

Image Mosaics



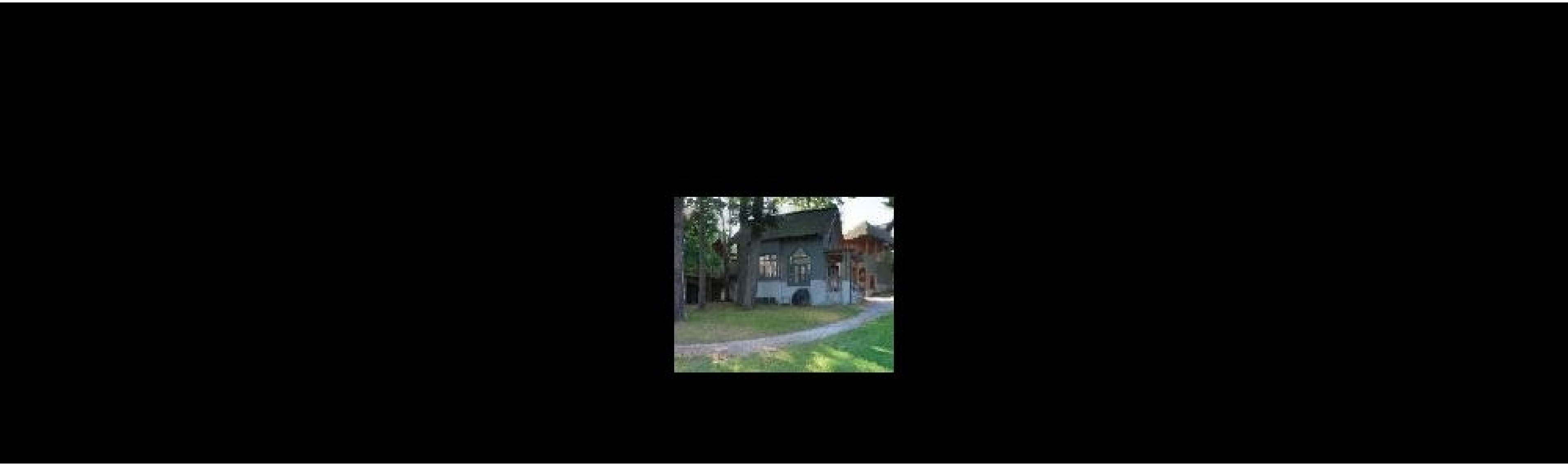
Goal

- Stitch together several images into a seamless composite

Adapted from Steven Seitz, University of Washington

Motivation

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°



Adapted from Brown & Lowe

Motivation

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°

Adapted from Brown & Lowe

Motivation

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Adapted from Brown & Lowe

How to create panoramas?

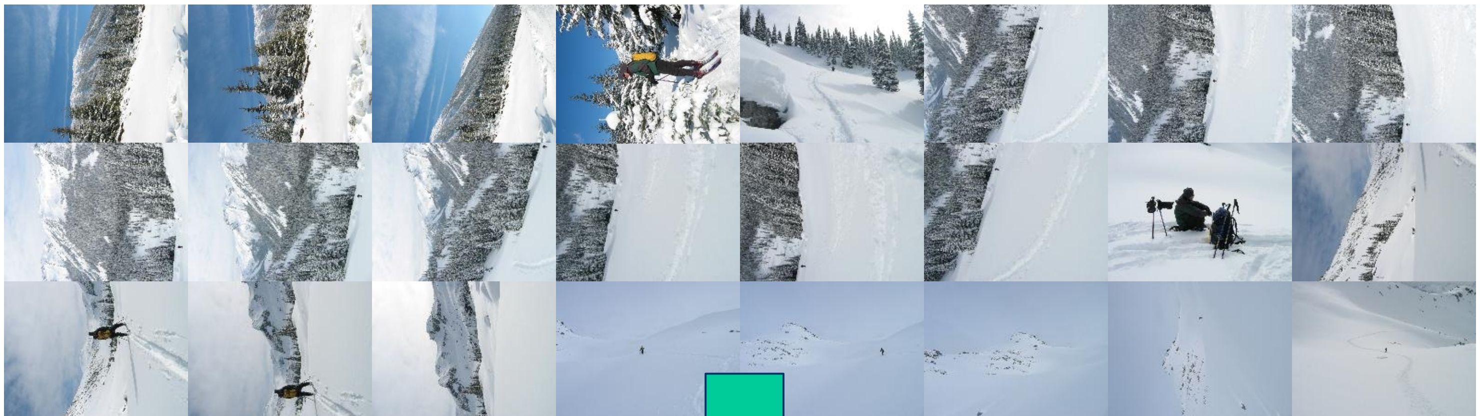
- Ordering = matching images

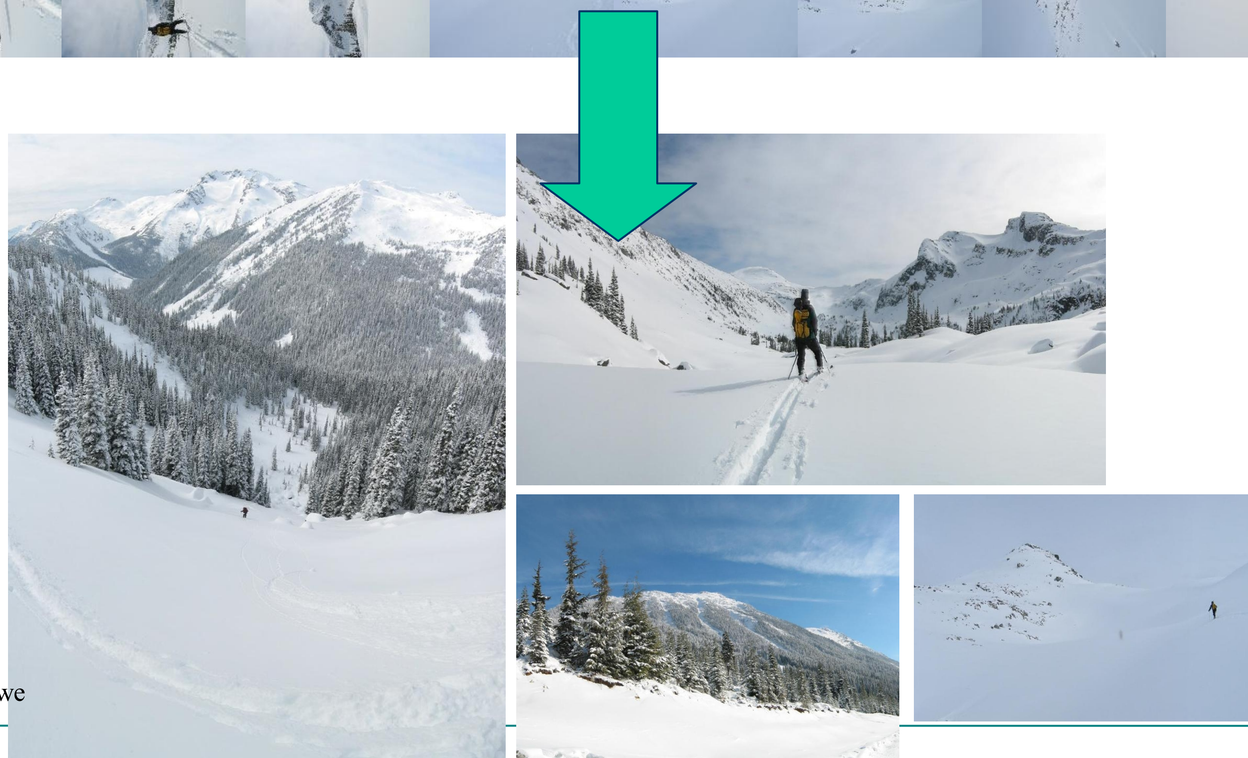
Adapted from Brown & Lowe

How to create panoramas?

- Ordering != matching images

Recognising Panoramas





Adapted from Brown & Lowe

Basic Procedure

Take a sequence of images from the same position

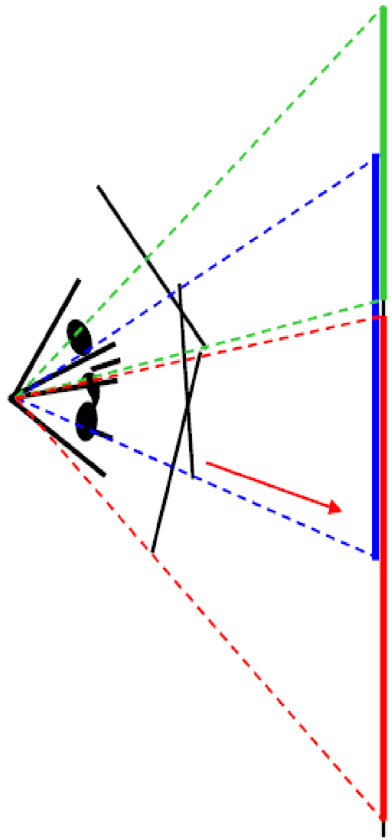
- Compute transformation between second image and first
- Shift the second image to overlap with the first
- Blend the two together to create a mosaic
- If there are more images, repeat

Aligning Images

Translations are not enough to align images

Adapted from Steven Seitz, University of Washington

Image Reprojection



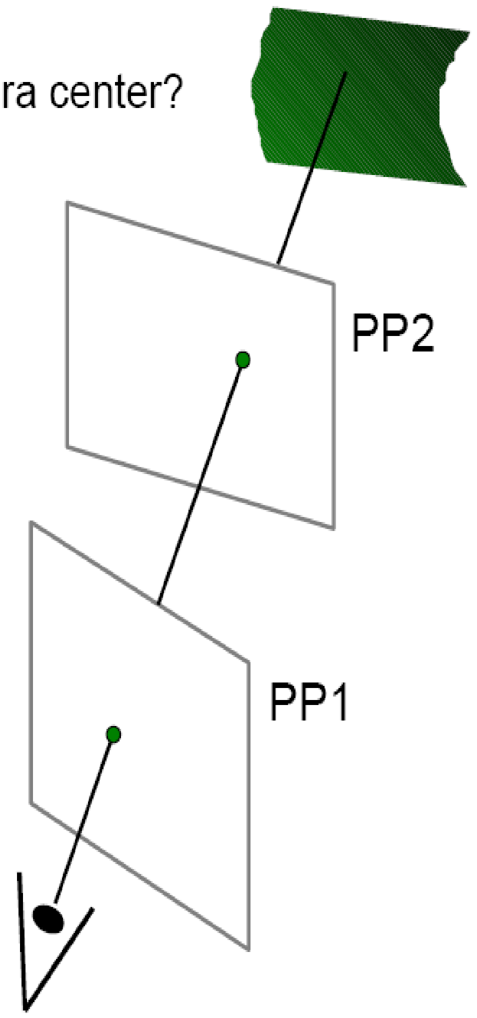
The mosaic has a natural interpretation in 3D

- The images are reprojected onto a common plane
- The mosaic is formed on this plane

Image Reprojection

Basic question

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

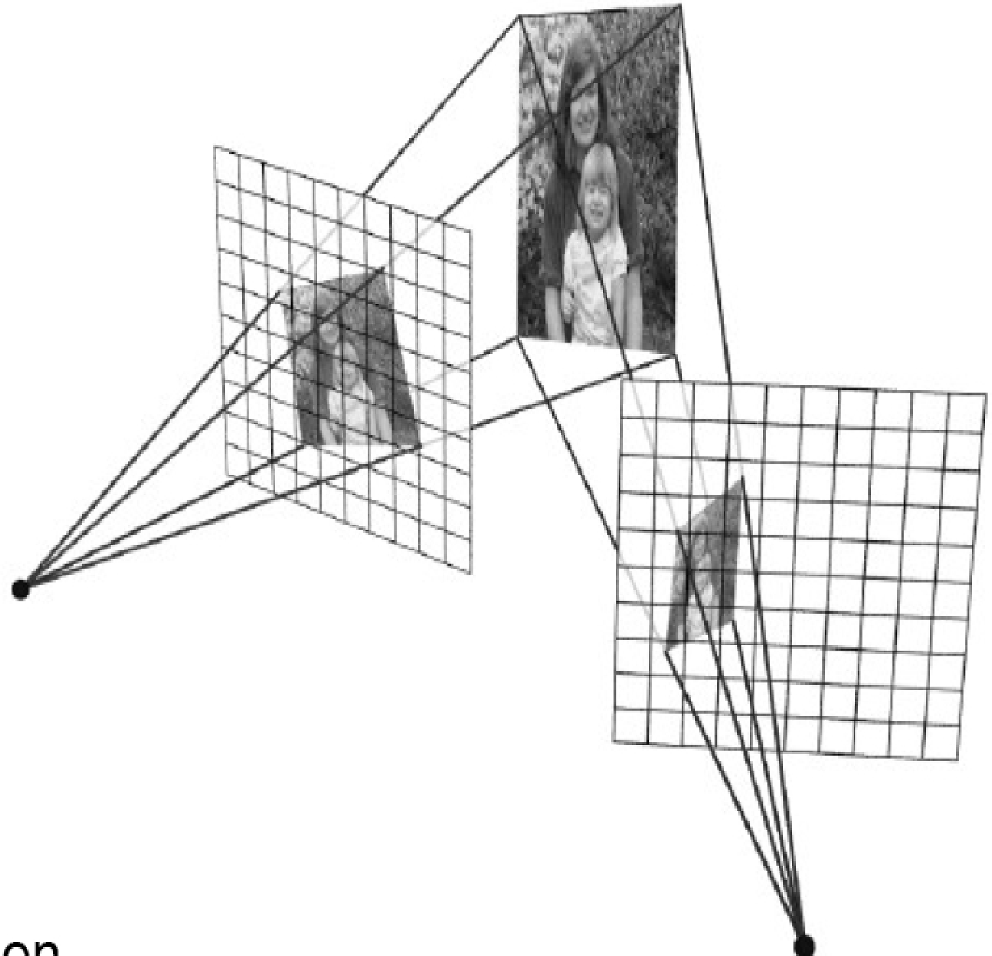


Answer

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

Adapted from Steven Seitz, University of Washington

Image Reprojection



Observation

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

Adapted from Steven Seitz, University of Washington

Homographies

Perspective projection of a plane
- Lots of names for this:
  homography, texture-map, colineation, planar projective map
- Modeled as a 2D warp using homogeneous coordinates

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

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

To apply a homography **H**
- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

– divide by w (third) coordinate

Adapted from Steven Seitz, University of Washington

Image Warping with Homographies

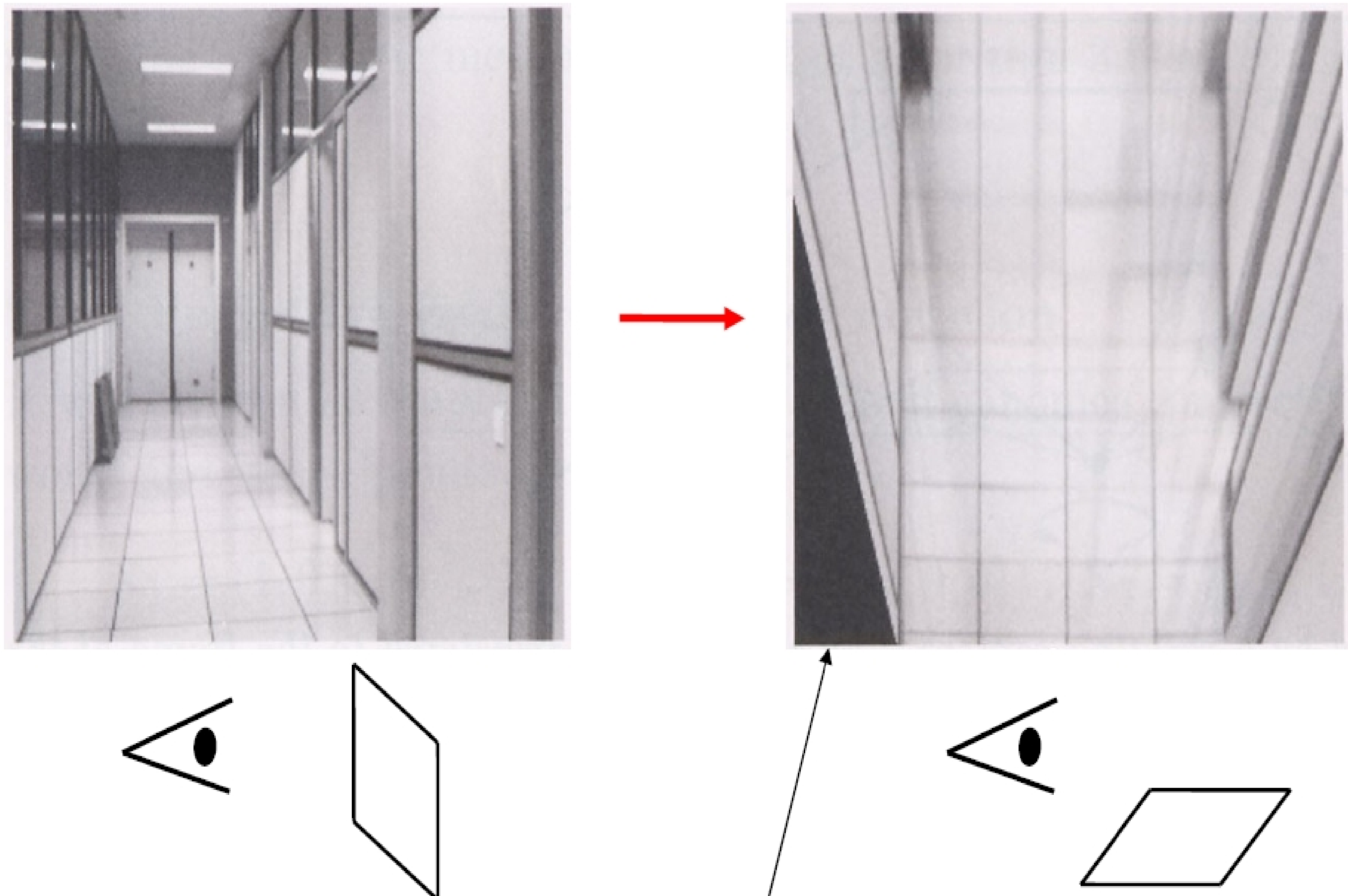


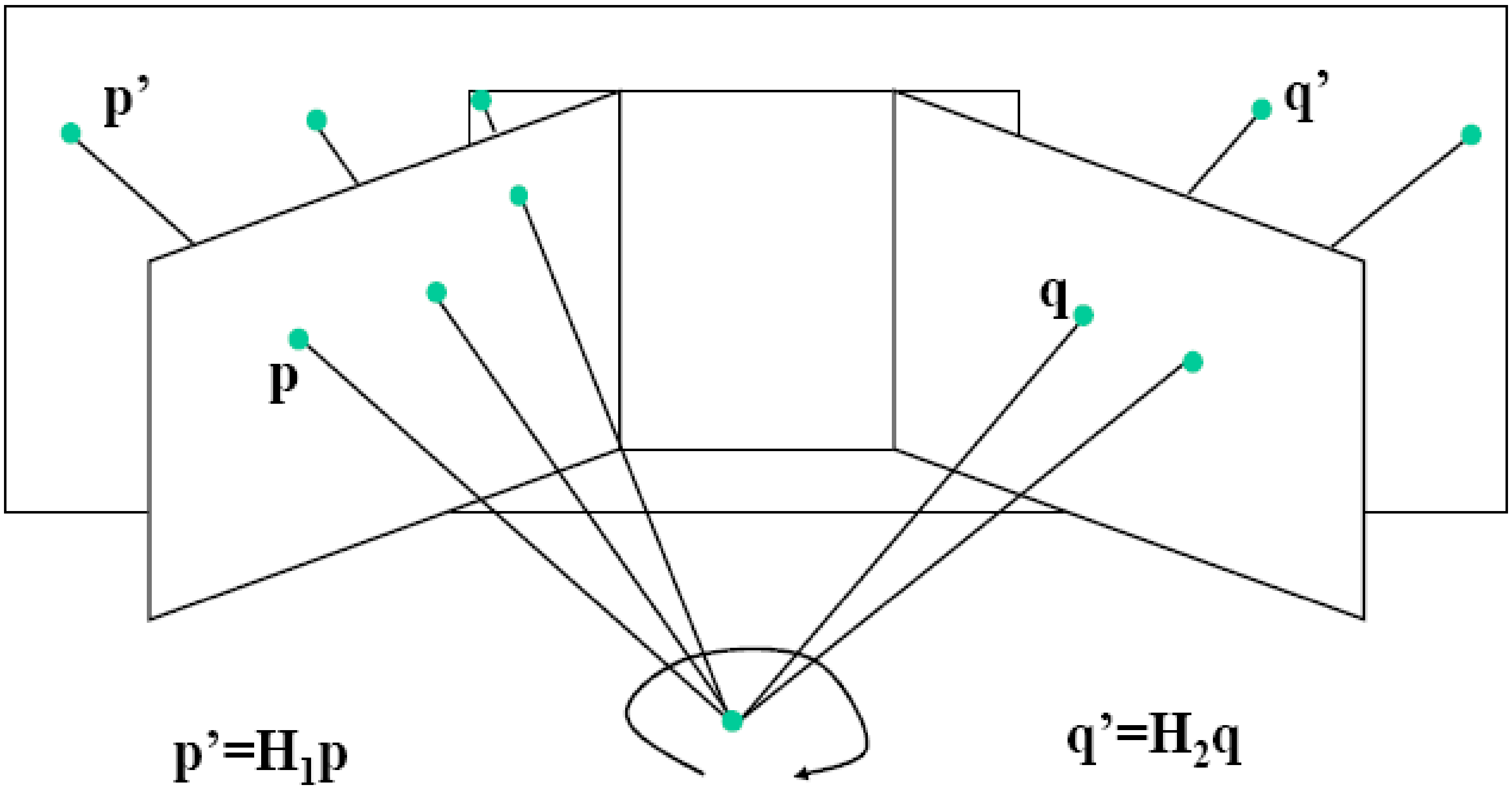
image plane in front

image plane below

black area
where no pixel
maps to

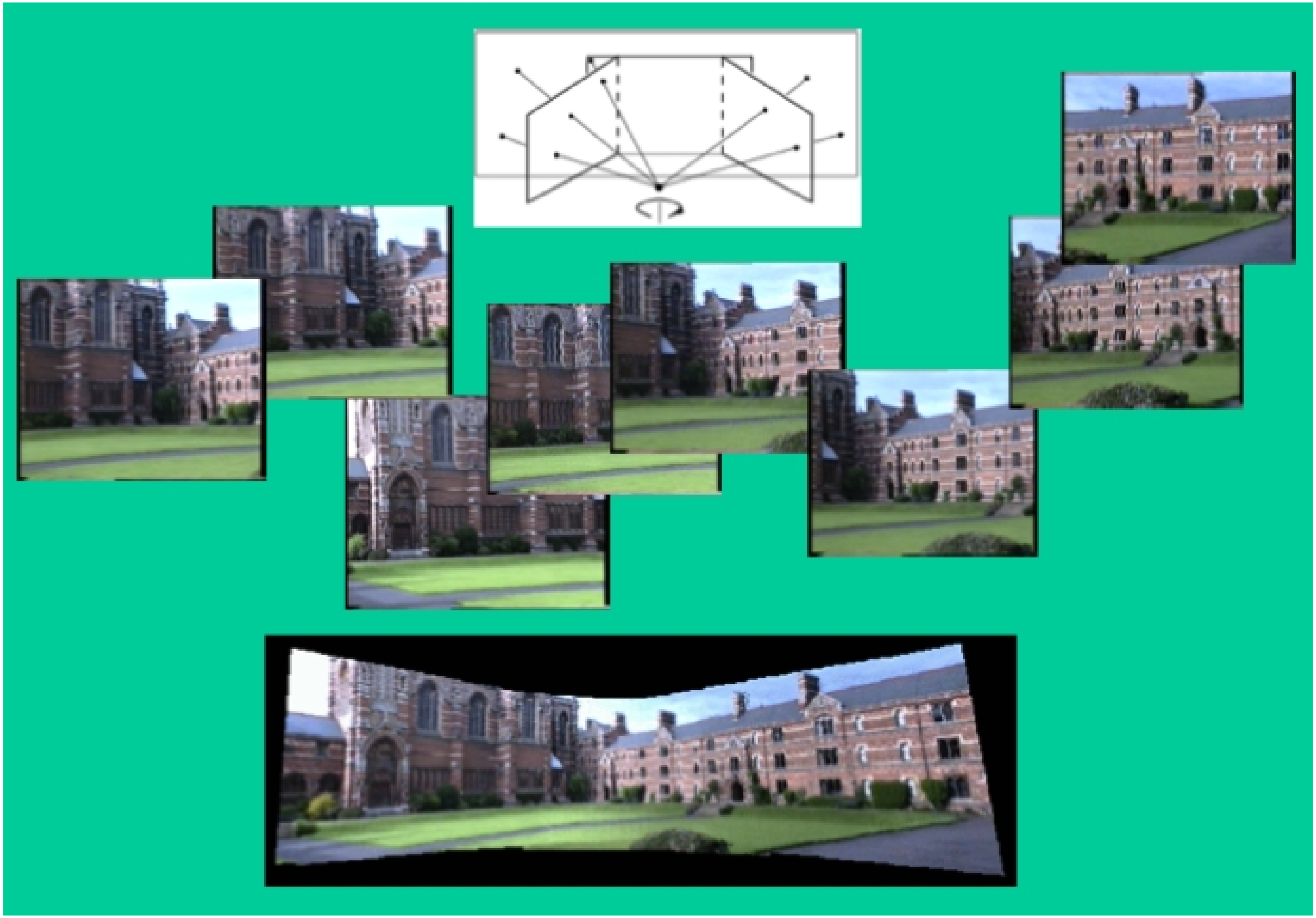
Adapted from Steven Seitz, University of Washington

Homographies



Adapted from Martial Hebert, CMU

Homographies



Adapted from Serge Belongie

Homographies

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \Leftrightarrow \boldsymbol{x}_2 = H\boldsymbol{x}_1$$

In inhomogenous coordinates ($x'_2 = x_2/z_2$ and $y'_2 = y_2/z_2$),

$$x'_2 = \frac{H_{11}x_1 + H_{12}y_1 + H_{13}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1}$$
$$y'_2 = \frac{H_{21}x_1 + H_{22}y_1 + H_{23}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1}$$

Without loss of generality, set $z_1 = 1$ and rearrange:

$$x'_2(H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{11}x_1 + H_{12}y_1 + H_{13}$$
$$y'_2(H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{21}x_1 + H_{22}y_1 + H_{23}$$

Adapted from Serge Belongie

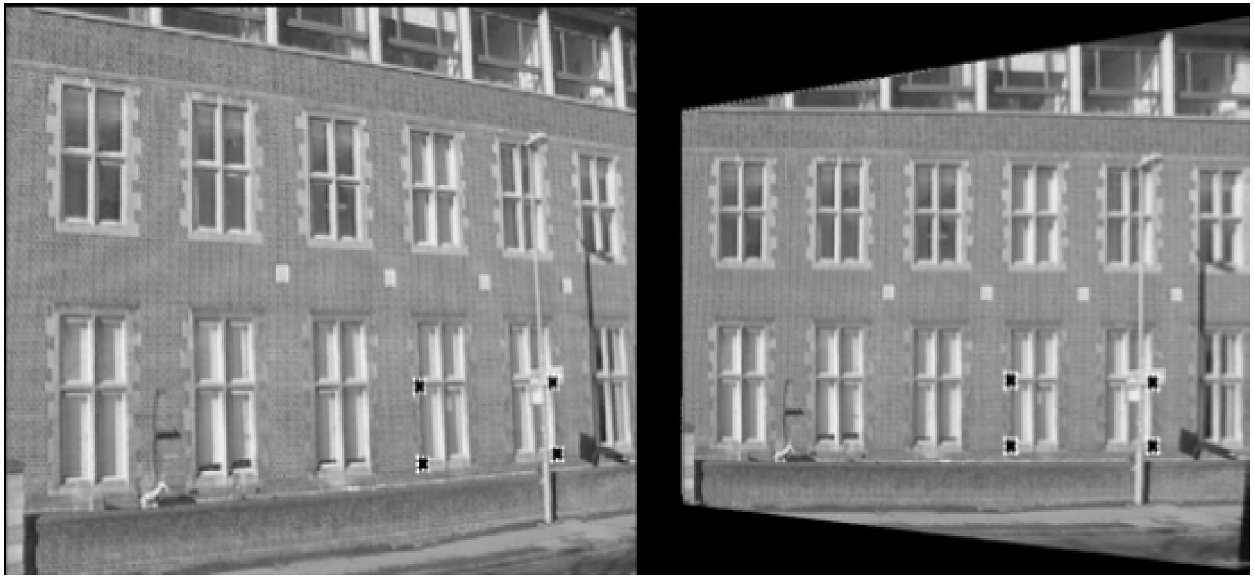
Homographies

$$x_1^i \overset{corresponds}{\longleftrightarrow} x_2^i$$

Each corresponding point gives two equations

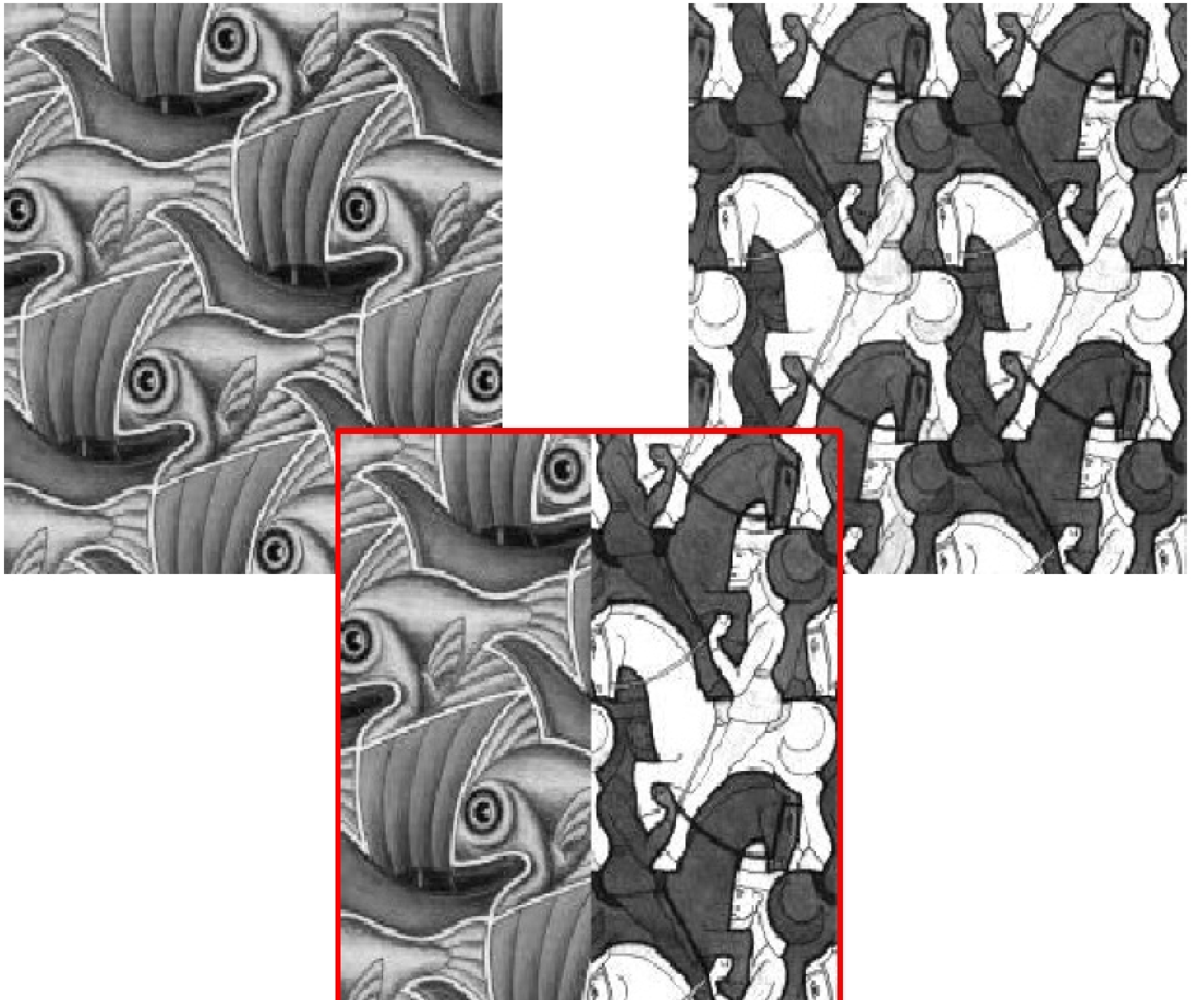
H has 8 degrees of freedom

so we need 8 equations = 4 correspondences to get H



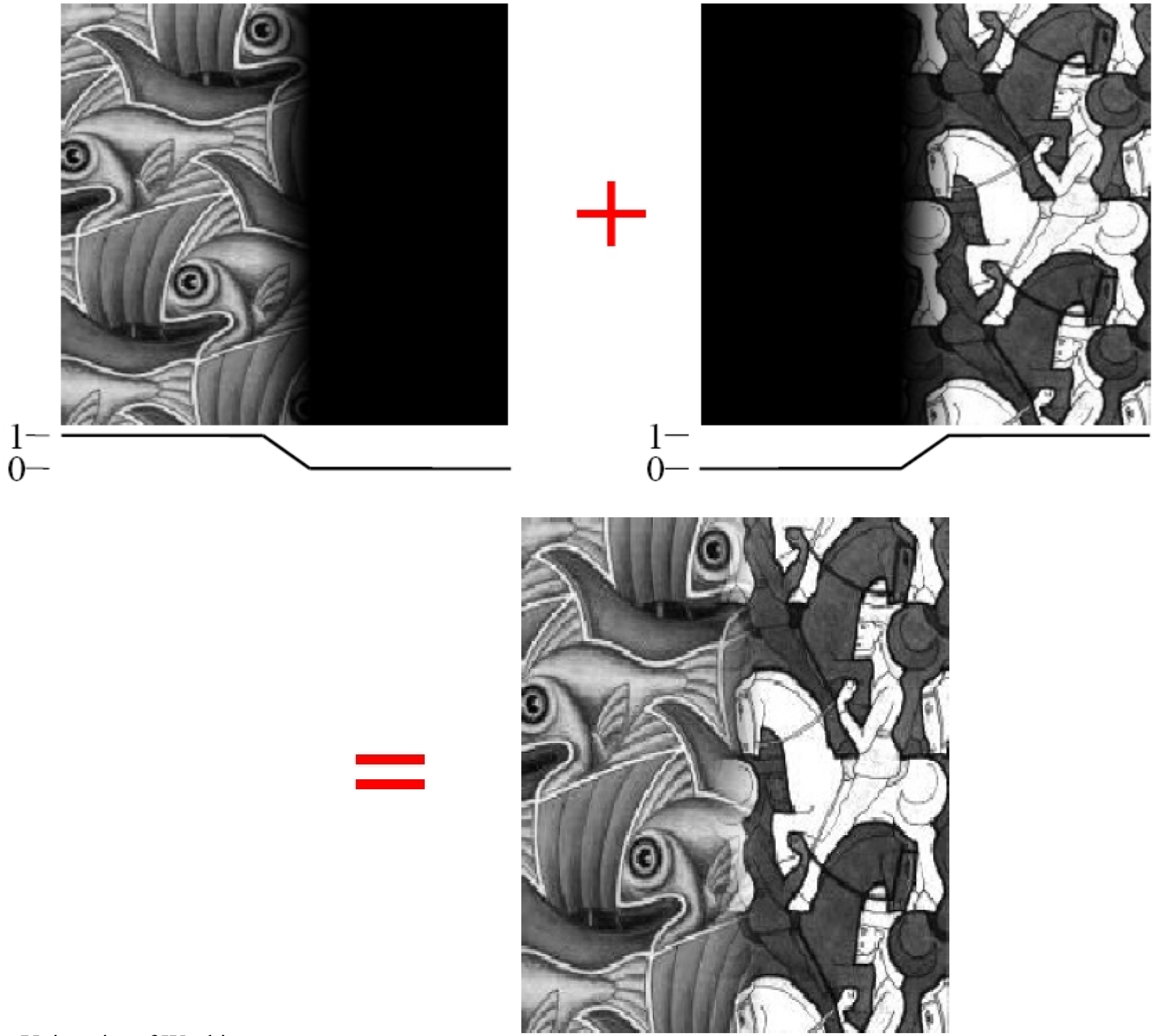
Adapted from Serge Belongie

Image Blending



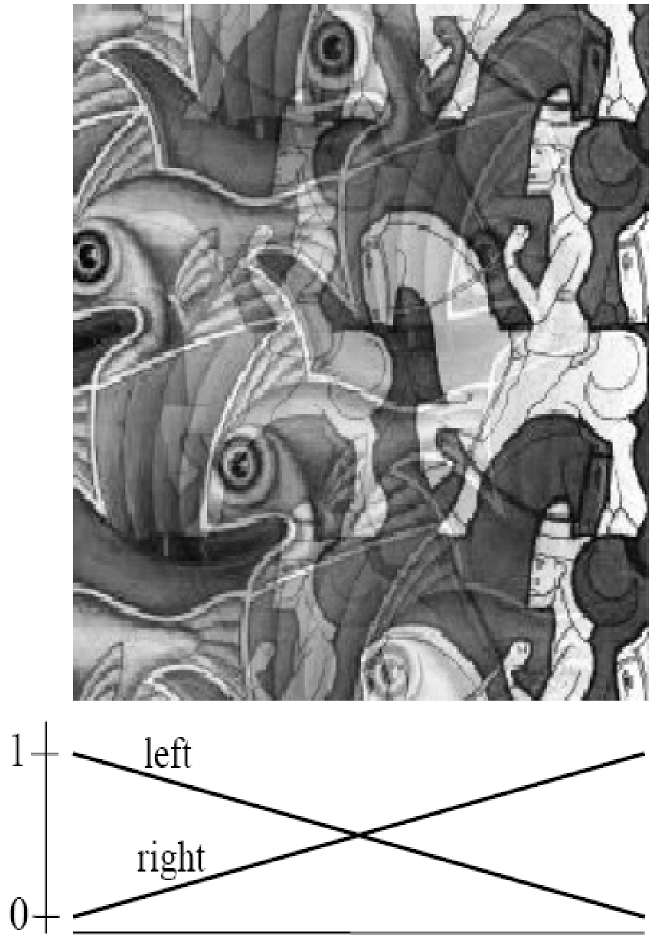
Adapted from Steven Seitz, University of Washington

Image Blending : Feathering



Adapted from Steven Seitz, University of Washington

Image Blending : Effect of window size

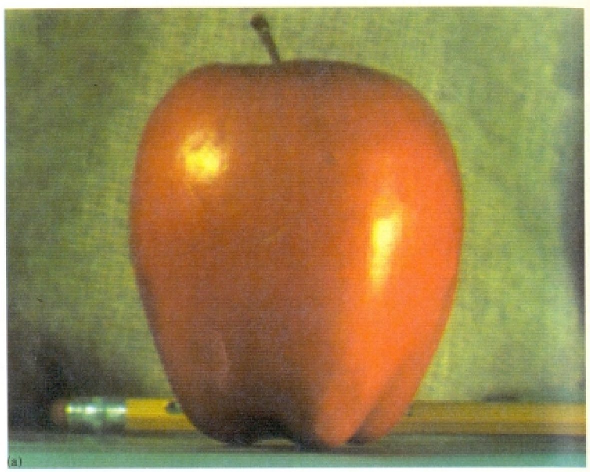


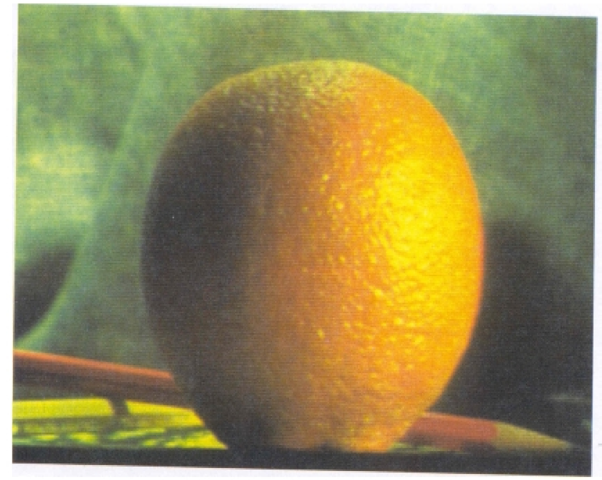
Adapted from Steven Seitz, University of Washington

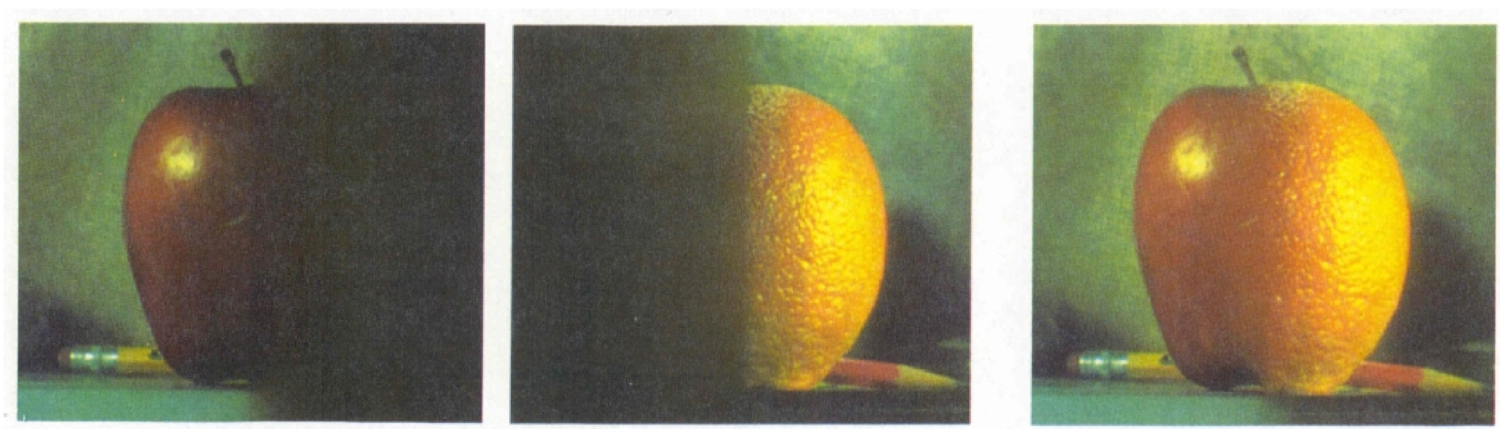
Image Blending : Effect of window size

Adapted from Steven Seitz, University of Washington

Pyramid Blending







(d) (h) (l)

Create a Laplacian pyramid, blend each level

- Burt, P. J. and Adelson, E. H., A multiresolution spline with applications to image mosaics, ACM Transactions on Graphics, 42(4), October 1983, 217-236.

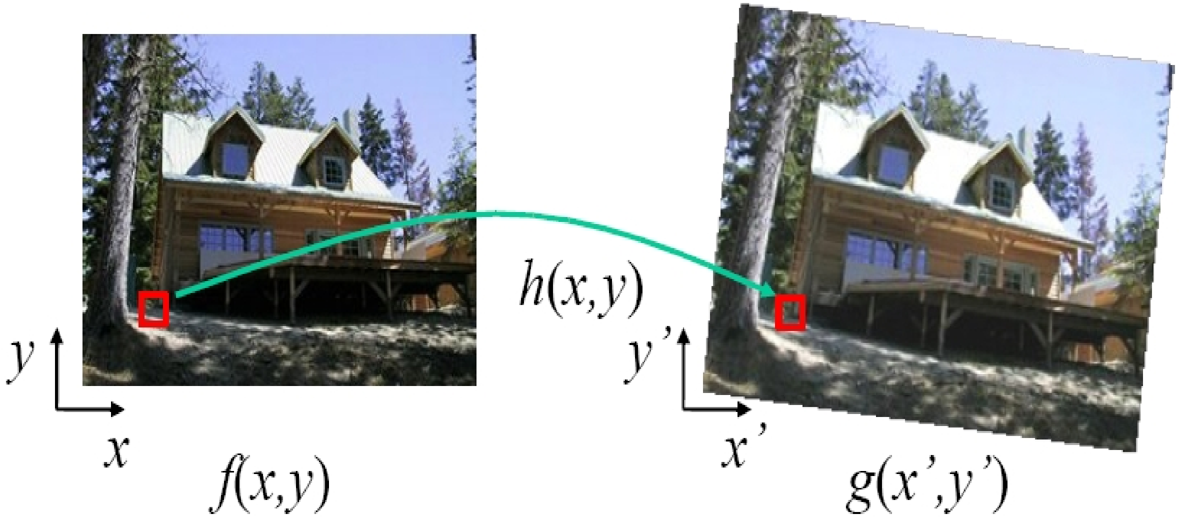
Image Editting

For more info: Perez et al, SIGGRAPH 2003

- http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf

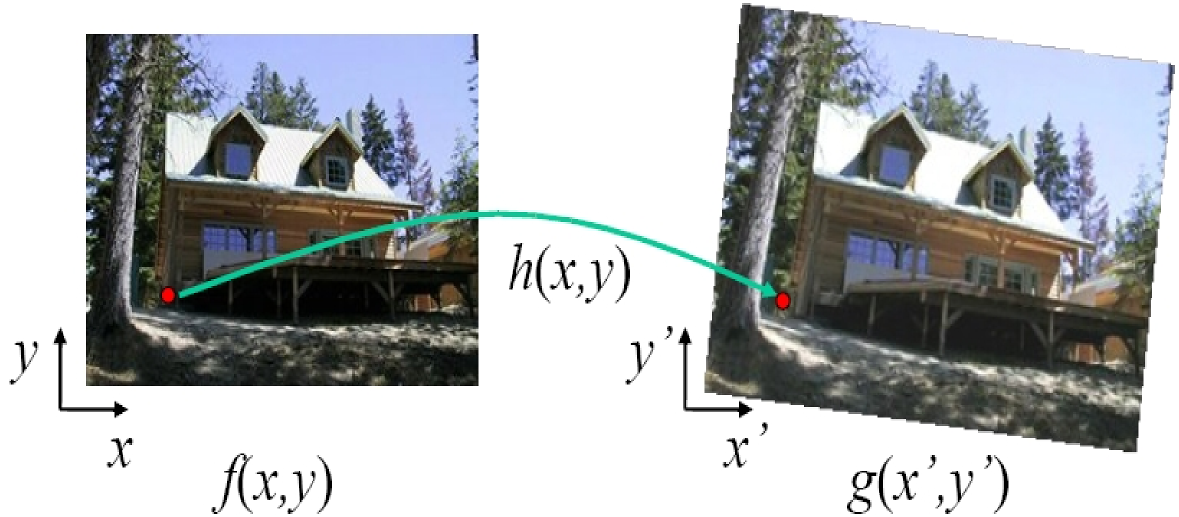
Adapted from Steven Seitz, University of Washington

Image warping



Given a coordinate transform $(x',y') = h(x,y)$ and a source image $f(x,y)$, how do we compute a transformed image $g(x',y') = f(h(x,y))$?

Forward warping

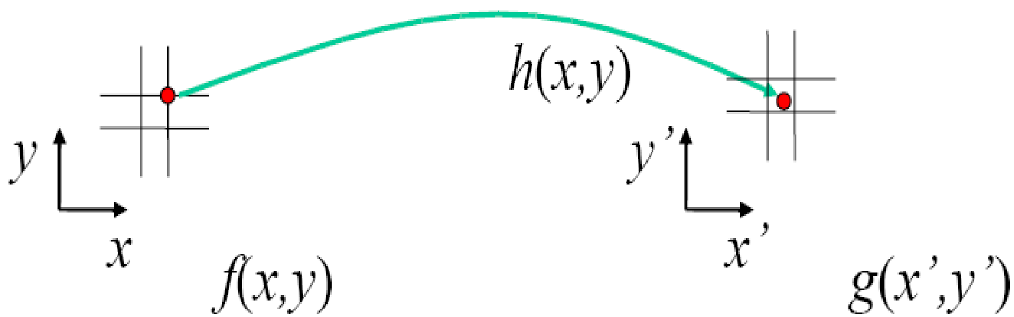


Send each pixel $f(x,y)$ to its corresponding location

$(x',y') = h(x,y)$ in the second image

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

Forward warping



Send each pixel $f(x,y)$ to its corresponding location

$(x',y') = h(x,y)$ in the second image

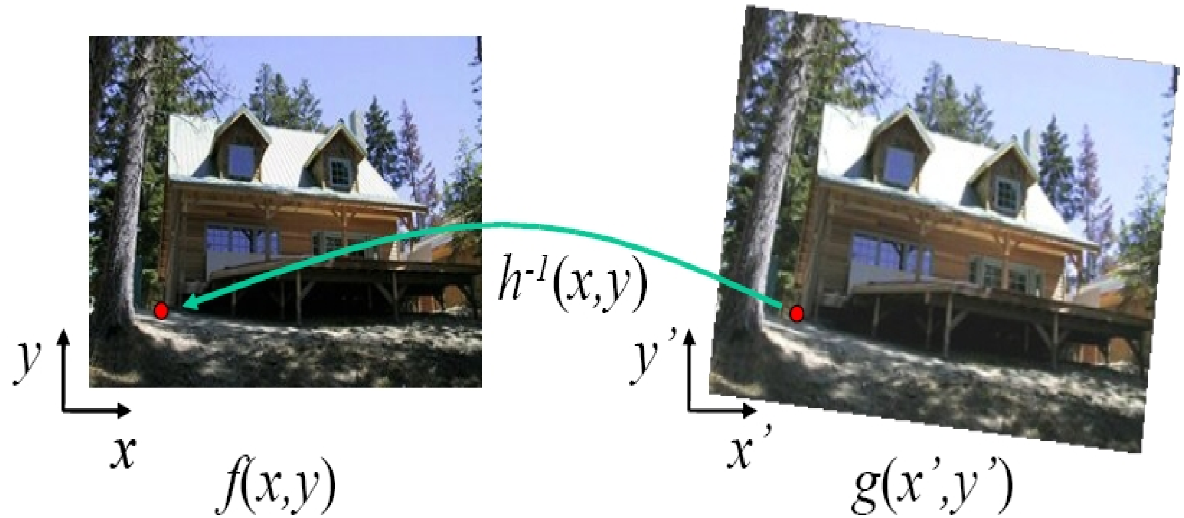
Q: what if pixel lands "between" two pixels?

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

- Known as "splatting"

Adapted from Steven Seitz, University of Washington

Inverse warping



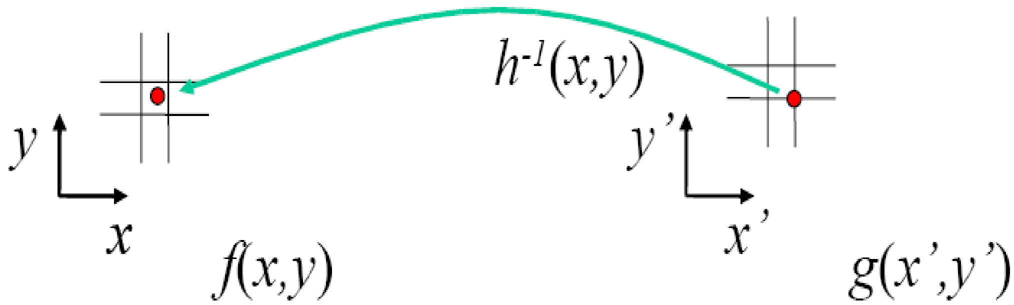
Get each pixel $g(x',y')$ from its corresponding location

$(x,y) = h^{-1}(x',y')$ in the first image

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

Adapted from Steven Seitz, University of Washington

Inverse warping



Get each pixel $g(x',y')$ from its corresponding location

$(x,y) = h^{-1}(x',y')$ in the first image

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

A: *resample* color value

Adapted from Steven Seitz, University of Washington

Autostitching

Method so far is not completely automatic
- -need to know which pairs fit together

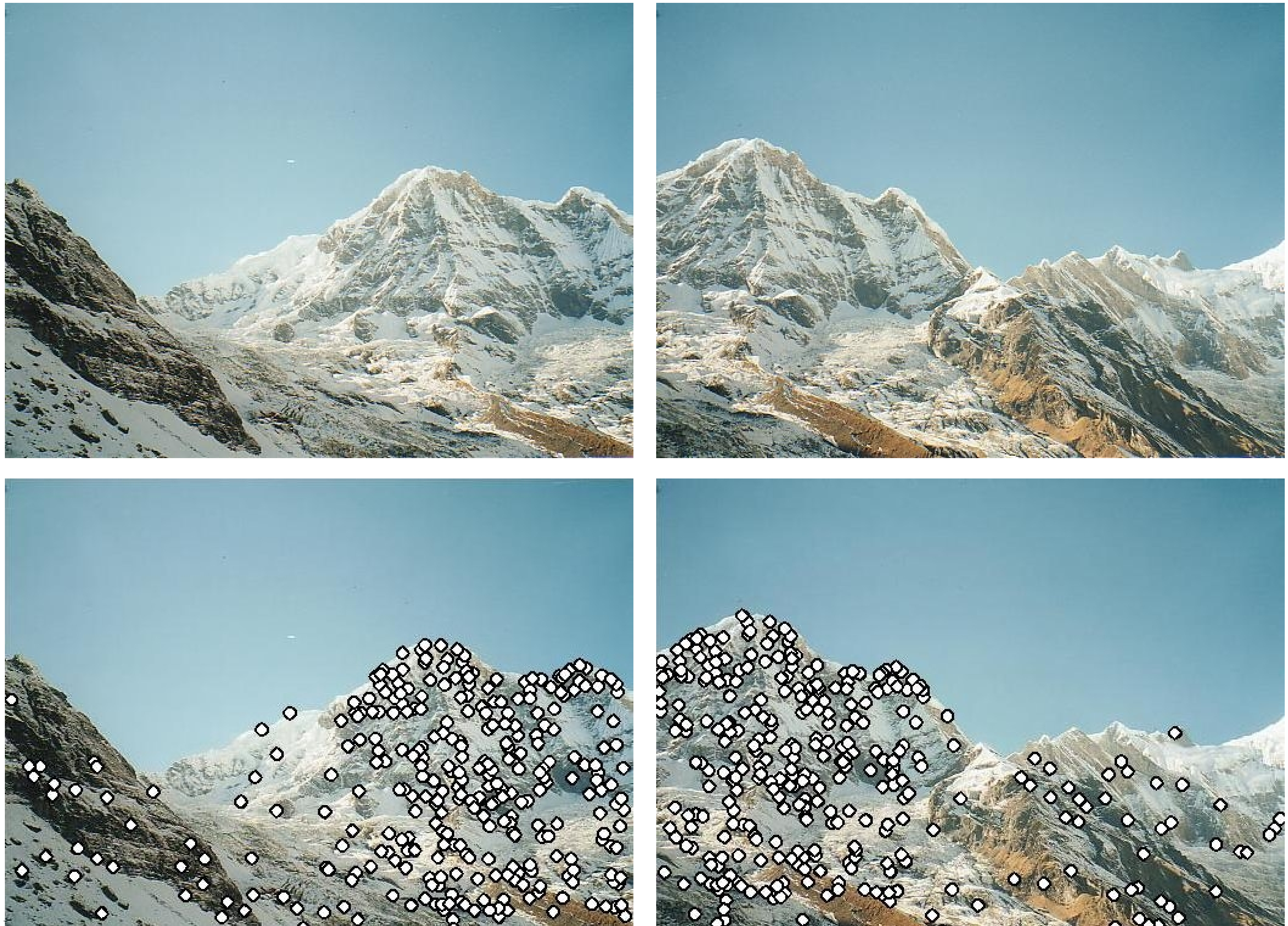
AutoStitch, by Matthew Brow and David Lowe
- -based on feature matching techniques

Autostitching – Feature Matching

- SIFT Features
  - Geometrically invariant to similarity transforms,
    - » some robustness to affine change
  - Photometrically invariant to affine changes in intensity
- Nearest Neighbor Matching

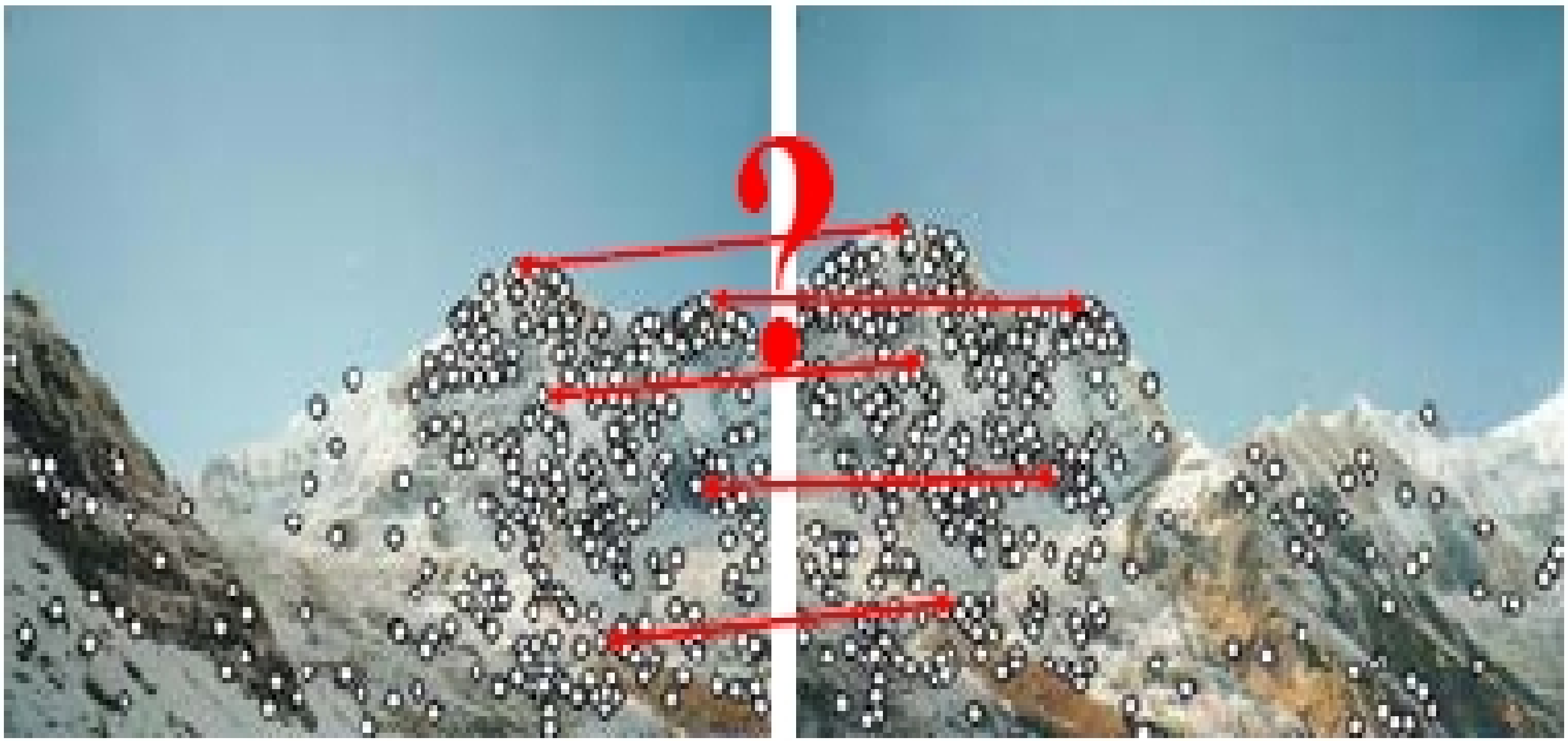
Adapted from Brown & Lowe

Autostitching



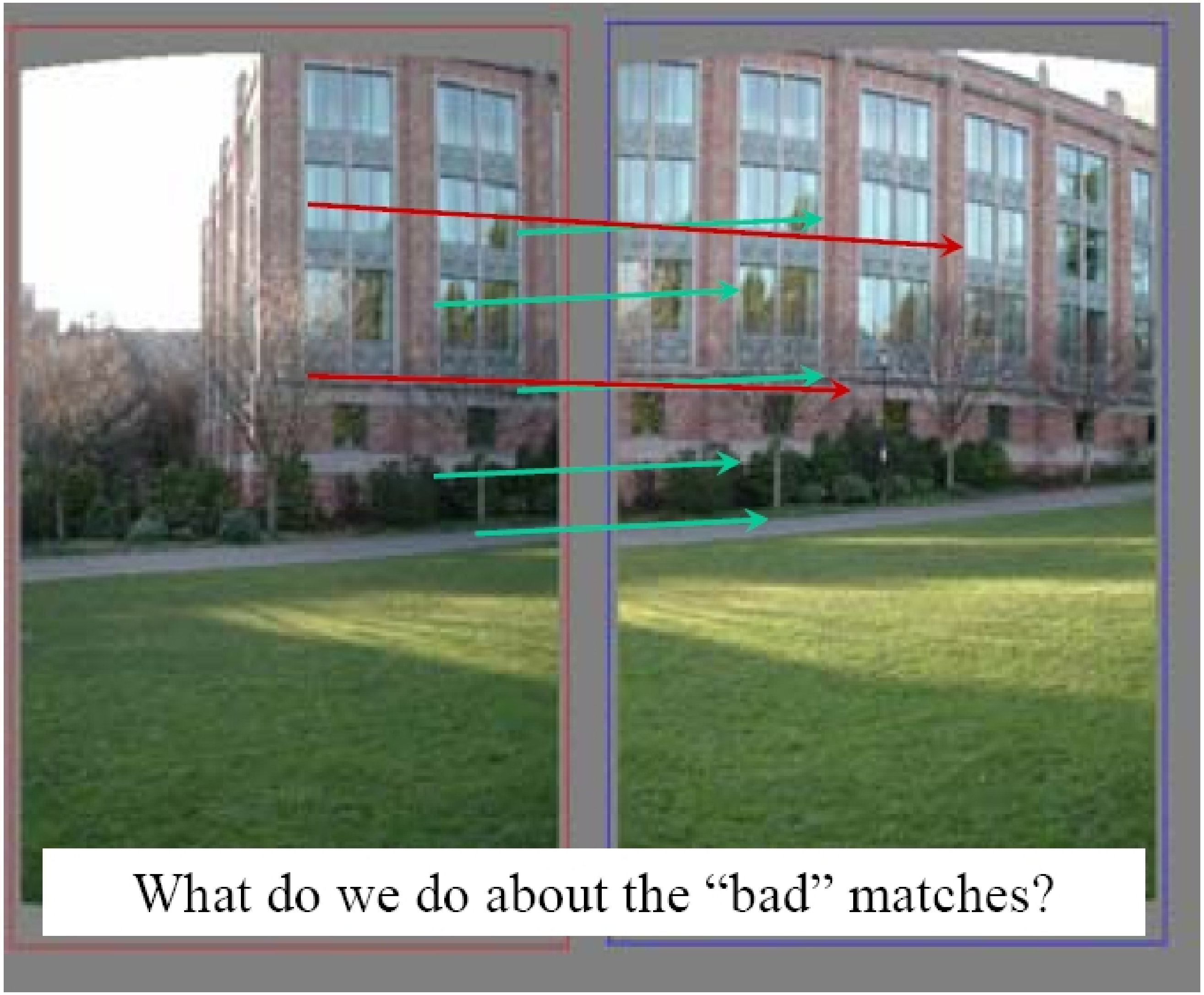
Adapted from

Autostiching



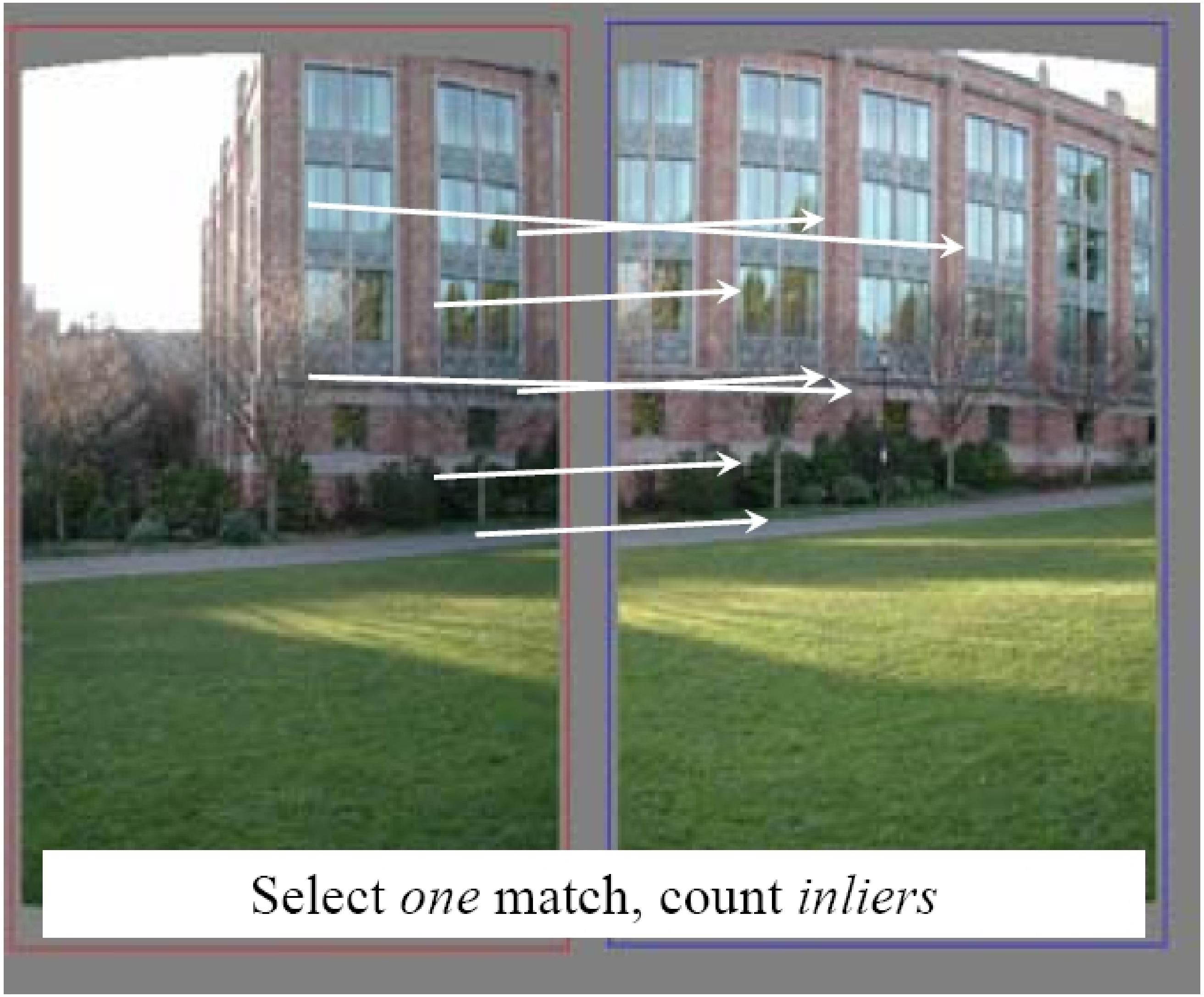
Adapted from Steven Seitz, University of Washington

Matching Features



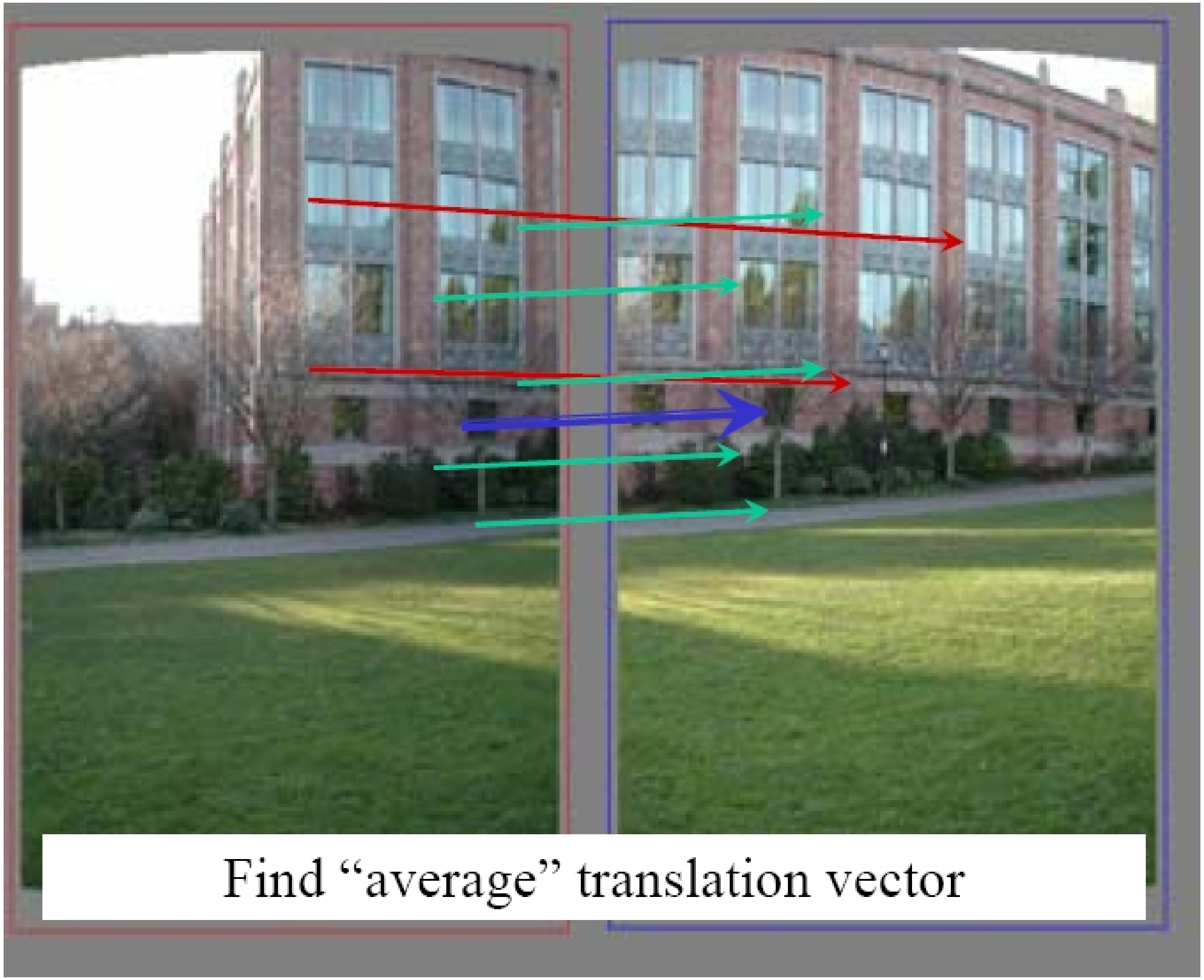
Adapted from Steven Seitz, University of Washington

RAndom SAmple Consensus (RANSAC)



Adapted from Steven Seitz, University of Washington

Least Squares Fit



Adapted from Steven Seitz, University of Washington

RAndom SAmple Consensus (RANSAC)

Popular approach for robust model fitting with outliers

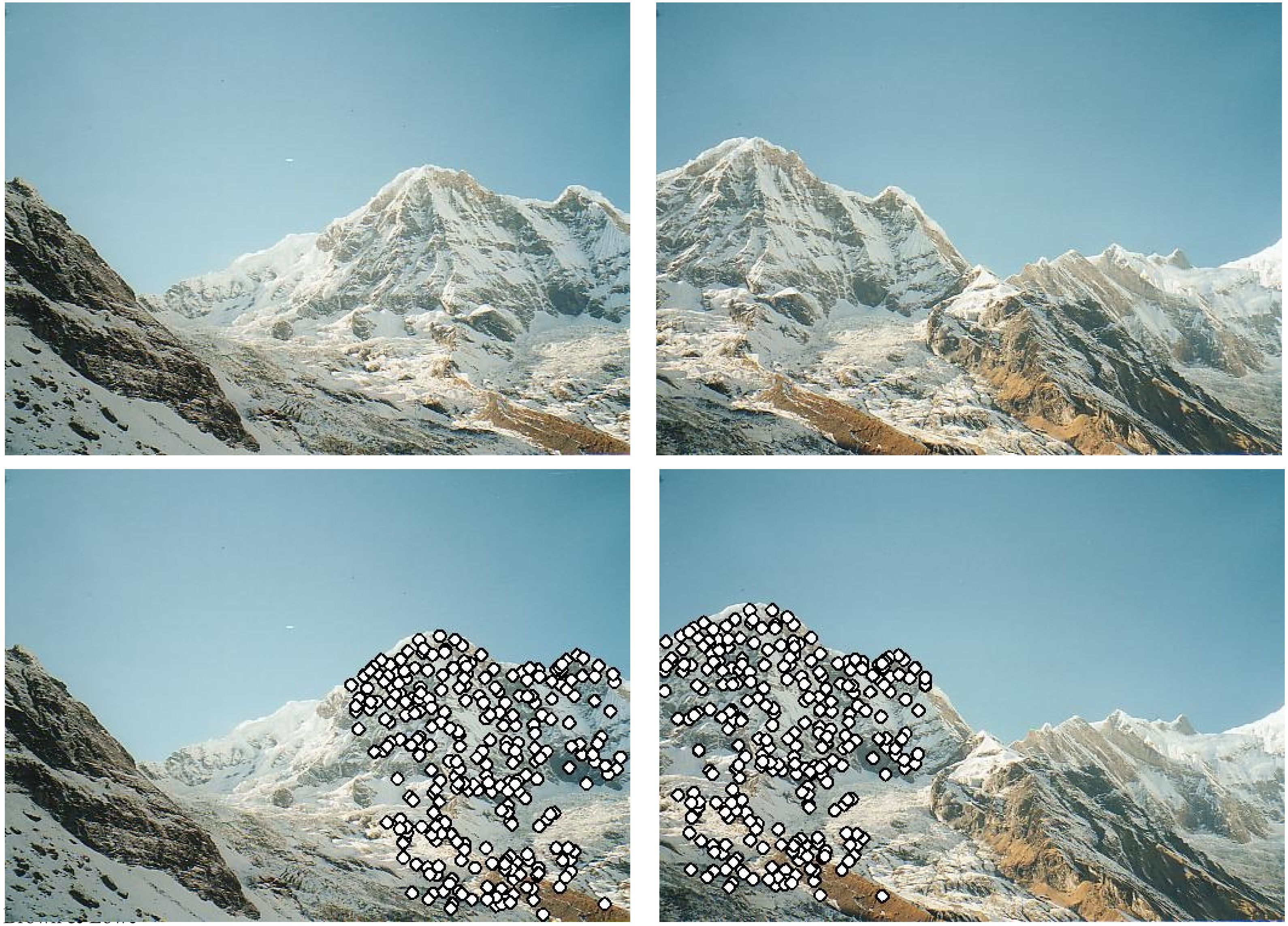
RANSAC loop:

1. Select K feature matches (at random)
2. Fit model (e.g., homography) based on these features
3. Count *inliers*:
   — number of other features that fit the model to within some specified threshold
4. The model with the largest number of inliers wins
5. Re-fit the model based on all of these inliers

More info:

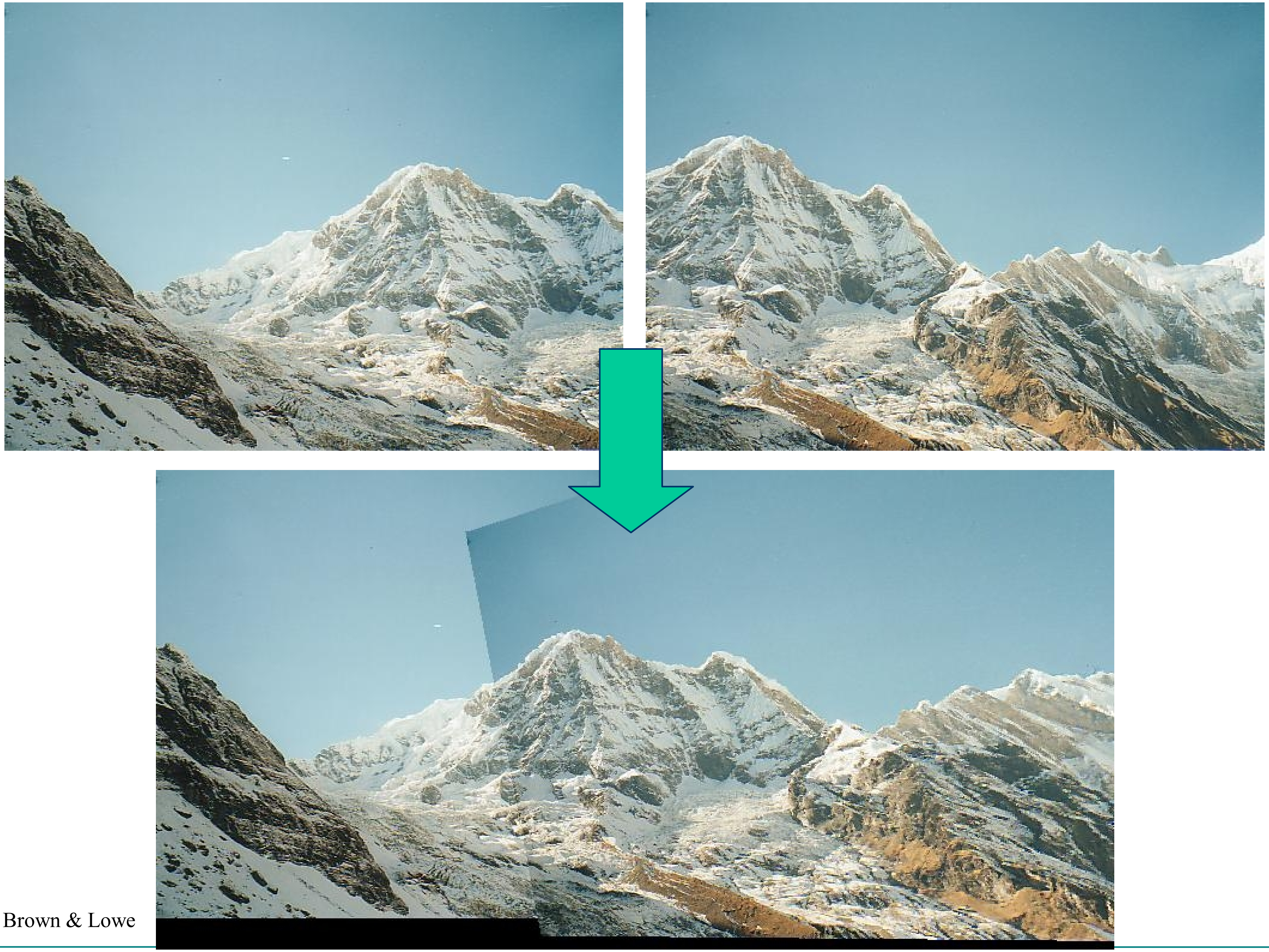
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FISHER/RANSAC/

Autostitching – RANSAC for Homography



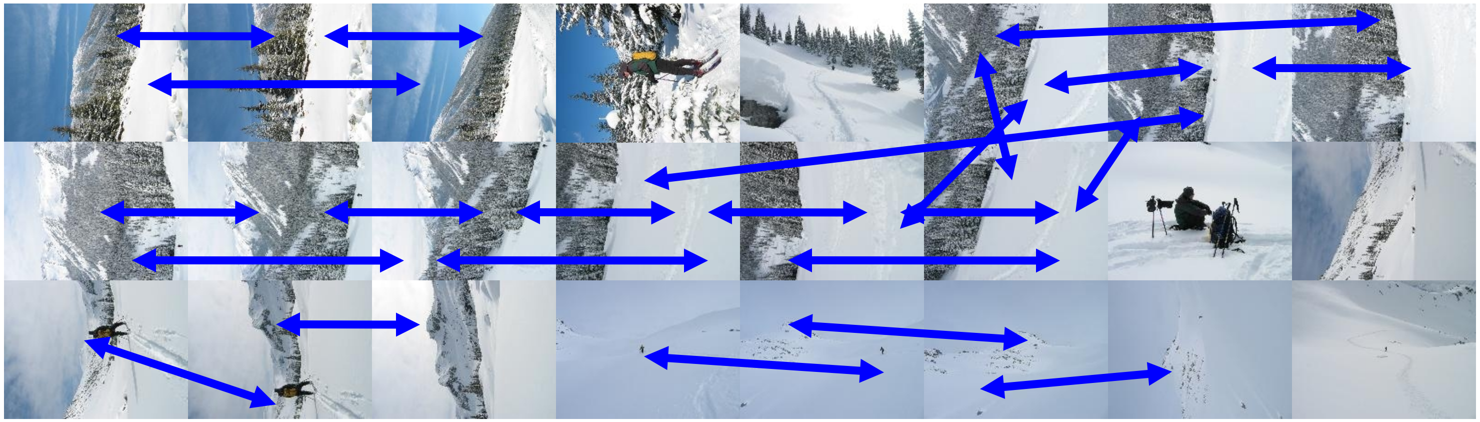
Adapted from

Autostitching – RANSAC for Homography



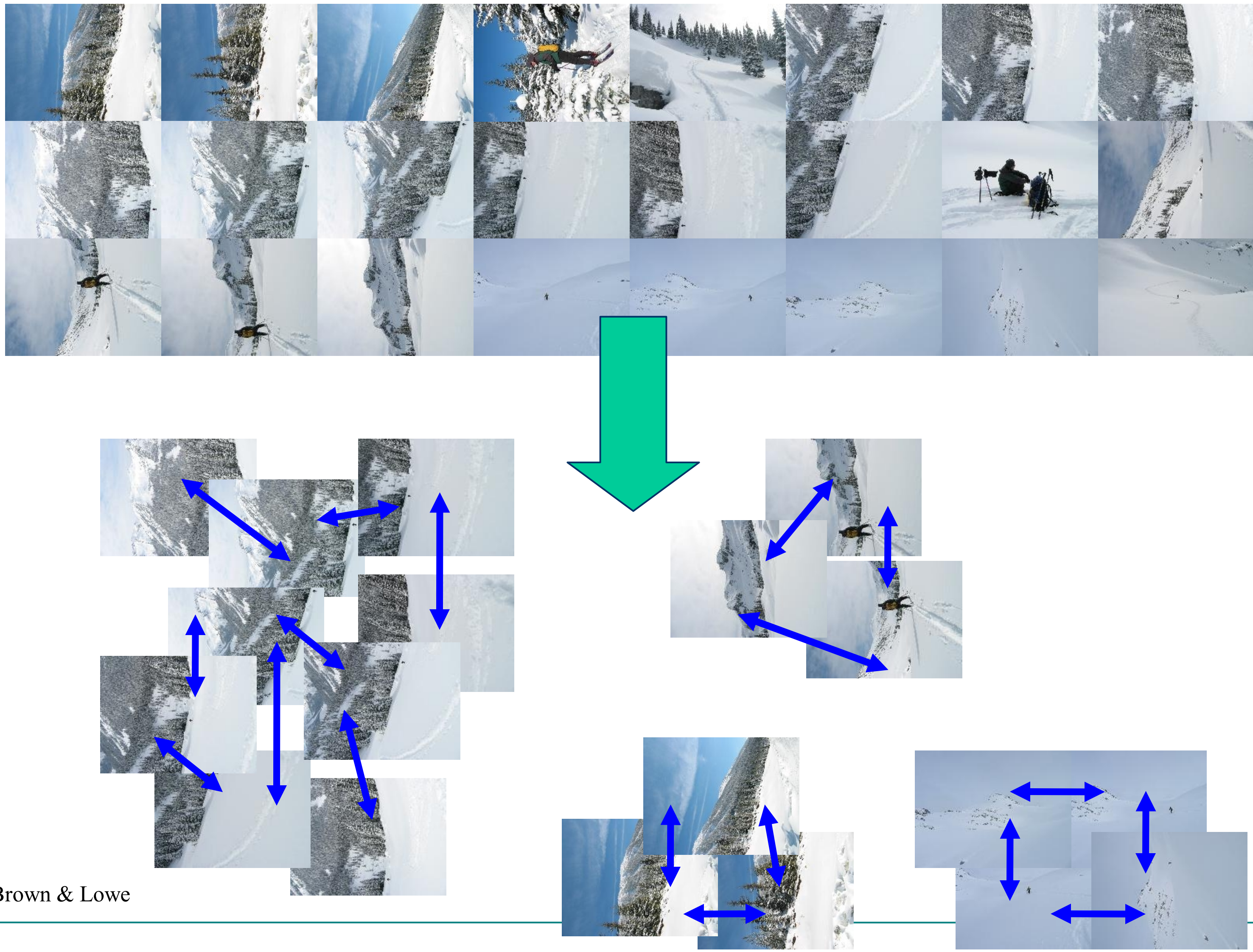
Adapted from Brown & Lowe

Autostitching – Finding the Panoramas



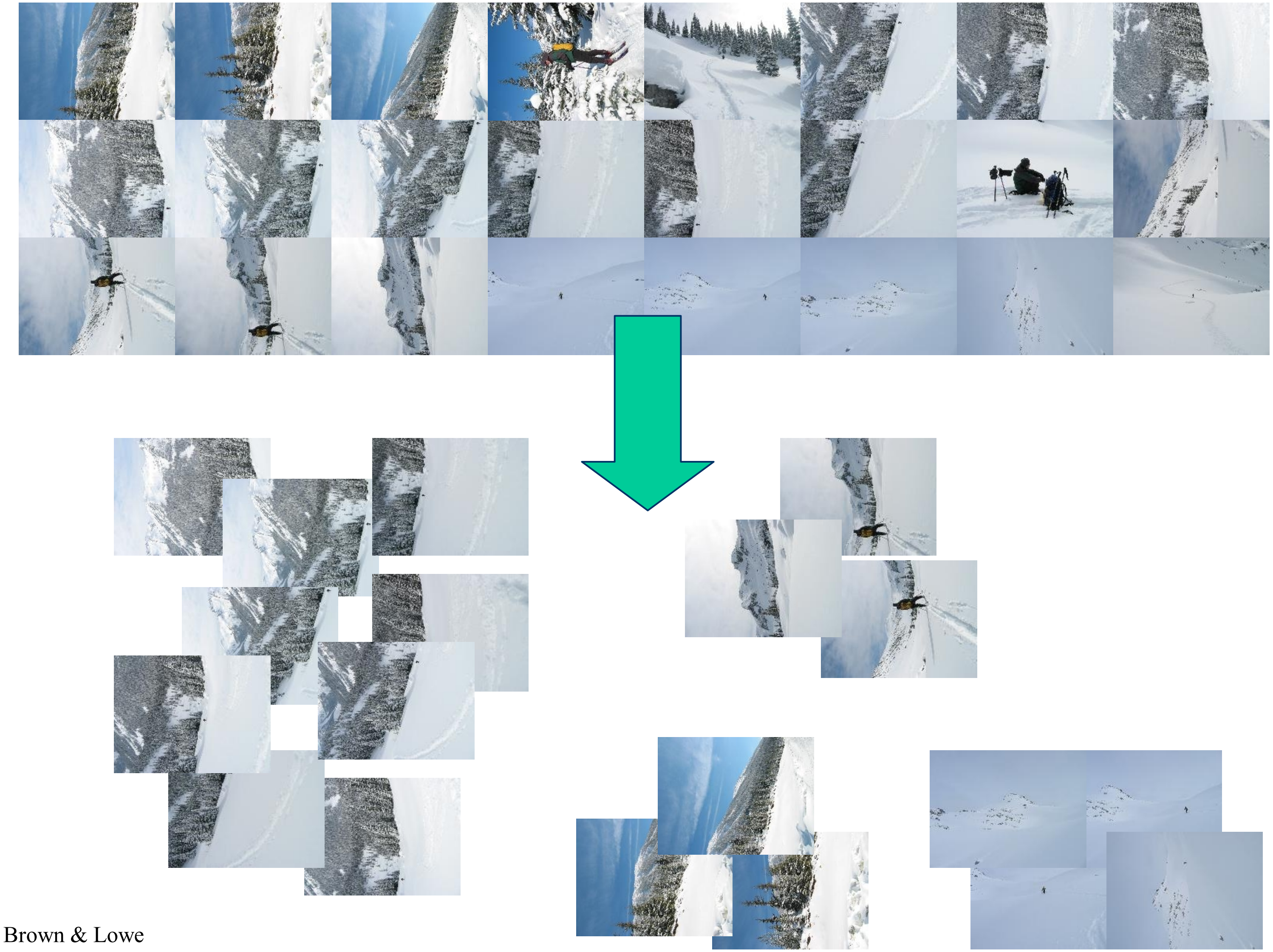
Adapted from Brown & Lowe

Autostitching – Finding the Panoramas



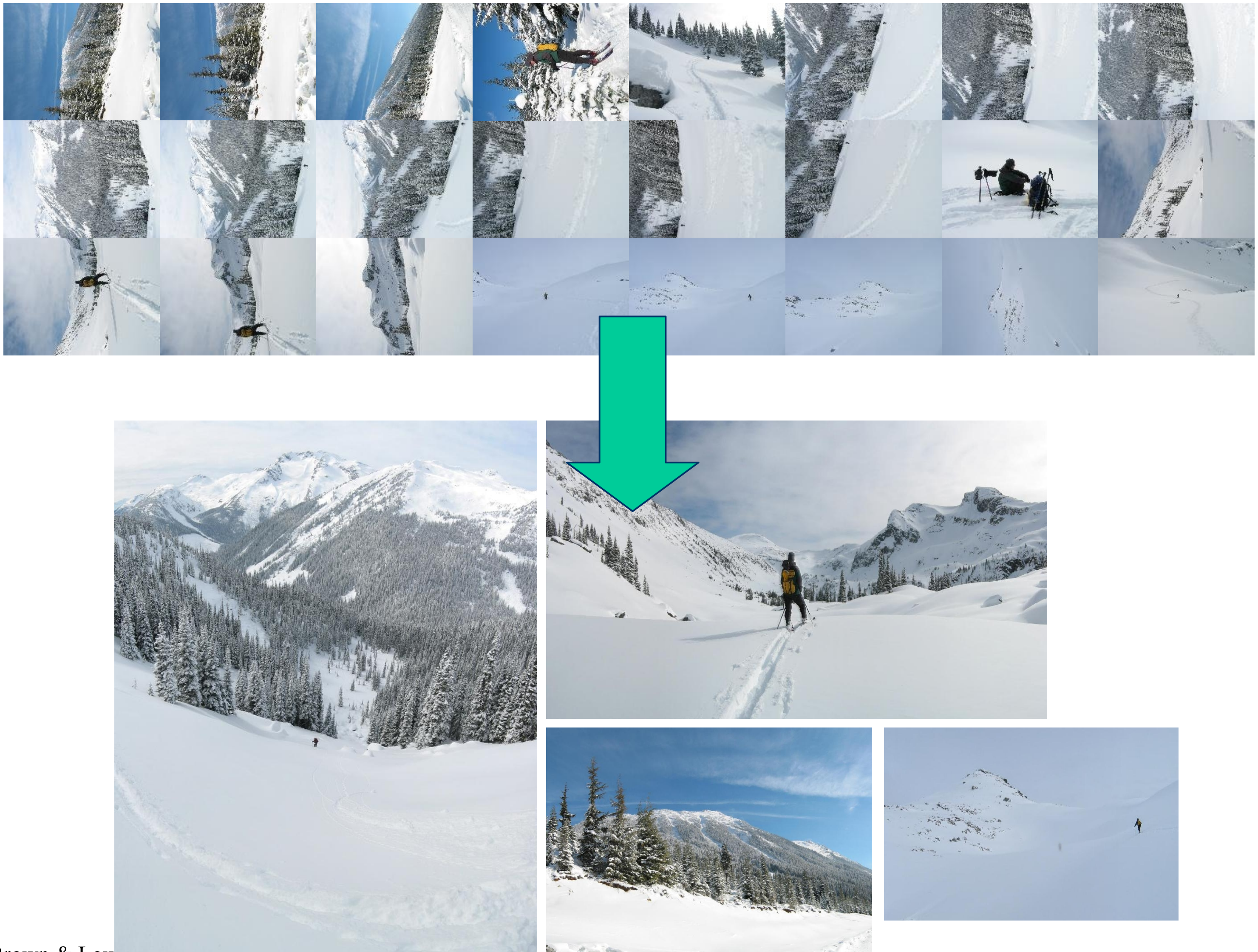
Adapted from Brown & Lowe

Autostitching – Finding the Panoramas



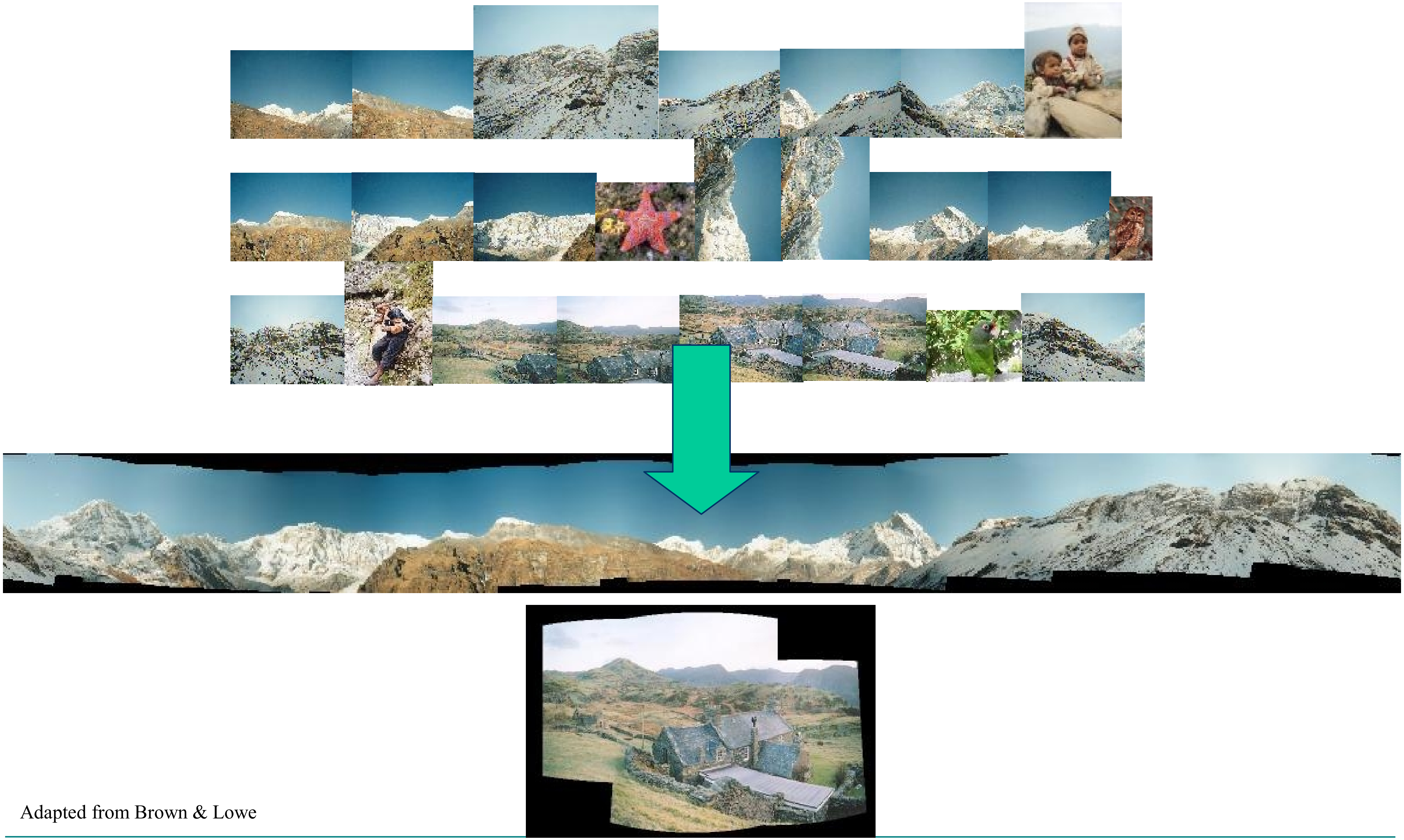
Adapted from Brown & Lowe

Autostitching – Finding the Panoramas



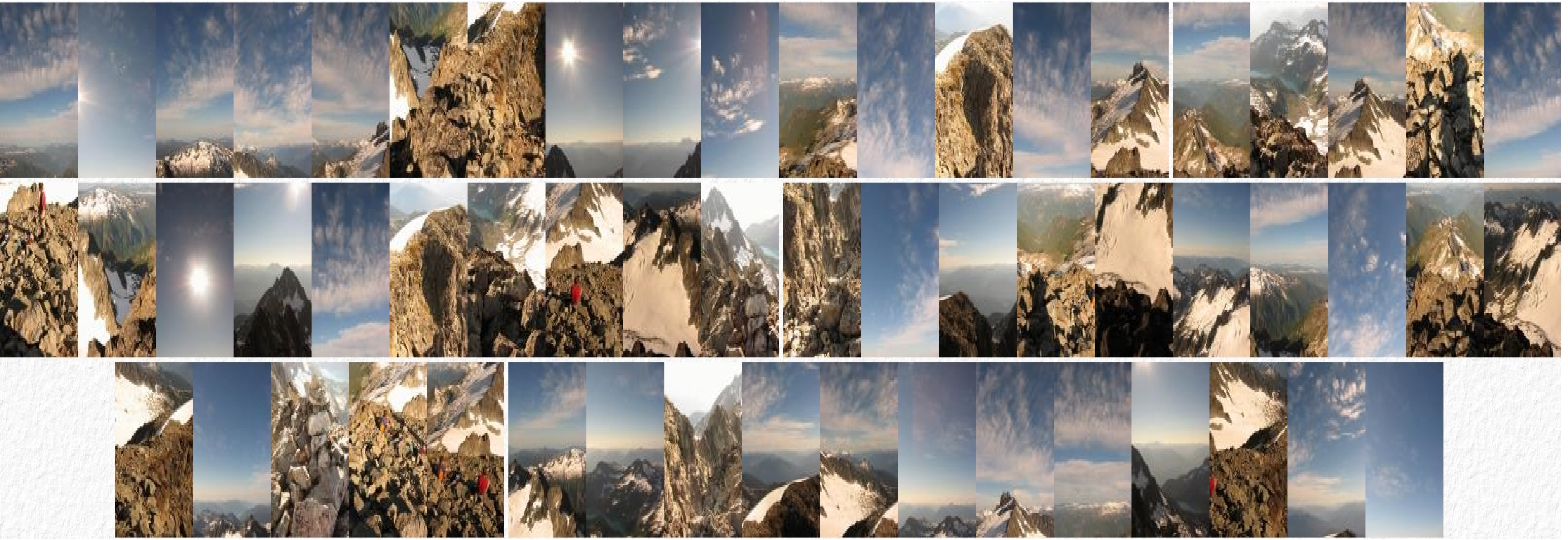
Adapted from Brown & Lowe

Autostitching – More Examples



Adapted from Brown & Lowe

Autostitch



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Autostitch

25 of 57 images aligned

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Autostitch

All 57 images aligned

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Autostitch

Final result

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html