2-view Alignment and RANSAC

CSE P576

Dr. Matthew Brown

AutoStitch iPhone

"Create gorgeous panoramic photos on your iPhone"
- Cult of Mac

"Raises the bar on iPhone panoramas"
- TUAW

"Magically combines the resulting shots"
- New York Times

Available on the iPhone App Store

4F12 class of '99

Case study – Image mosaicing

Any two images of a general scene with the same camera centre are related by a planar projective transformation given by:

$$\tilde{\mathbf{w}}' = \mathbf{K}\mathrm{R}\mathbf{K}^{-1}\tilde{\mathbf{w}}$$

where $\mathbf{K}$ represents the camera calibration matrix and R is the rotation between the views.

This projective transformation is also known as the homography induced by the plane at infinity. A minimum of four image correspondences can be used to estimate the homography and to warp the images onto a common image plane. This is known as **mosaicing**.

Scale Invariant Feature Transform

- Extract SIFT features from an image

- Each image might generate 100's or 1000's of SIFT descriptors

[A. Vedaldi]

Feature Matching

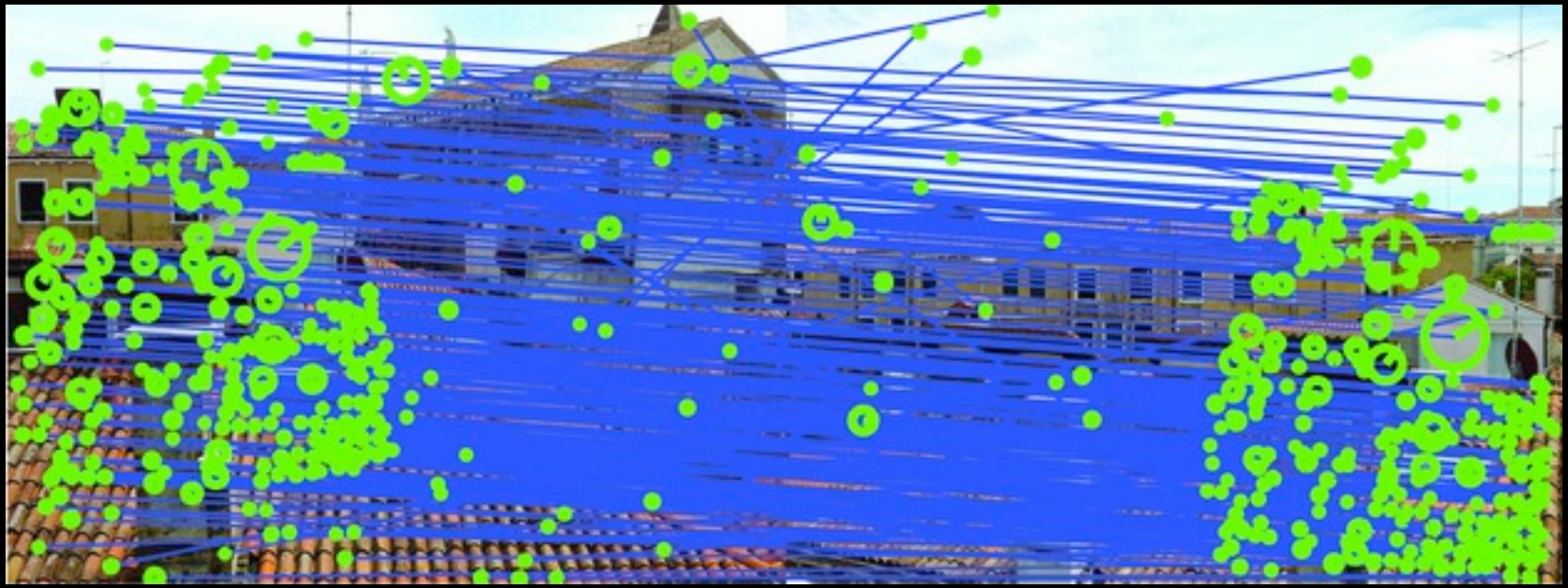
- Each SIFT feature is represented by 128 numbers
- Feature matching becomes task of finding a nearby 128-d vector
- All nearest neighbours:

$$\forall j \ NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||, \ i \neq j$$

- Solving this exactly is $O(n^2)$, but good approximate algorithms exist
- e.g., [Beis, Lowe ’97] Best-bin first k-d tree
- Construct a binary tree in 128-d, splitting on the coordinate dimensions
- Find approximate nearest neighbours by successively exploring nearby branches of the tree

2-view Rotational Geometry

- Feature matching returns a set of noisy correspondences
- To get further, we will have to understand something about the **geometry** of the setup



2-view Rotational Geometry

- Recall the projection equation for a pinhole camera

$$\tilde{\mathbf{u}} = \begin{bmatrix} & & \\ & \mathbf{K} & \\ & & \end{bmatrix} \left[\begin{array}{c|c} & \\ \mathbf{R} & \mathbf{t} \\ & \end{array} \right] \tilde{\mathbf{X}}$$

$\tilde{\mathbf{u}} \sim [u,\ v,\ 1]^T$: Homogeneous image position

$\tilde{\mathbf{X}} \sim [X,\ Y,\ Z,\ 1]^T$: Homogeneous world coordinates

$\mathbf{K}\ (3 \times 3)$: Intrinsic (calibration) matrix

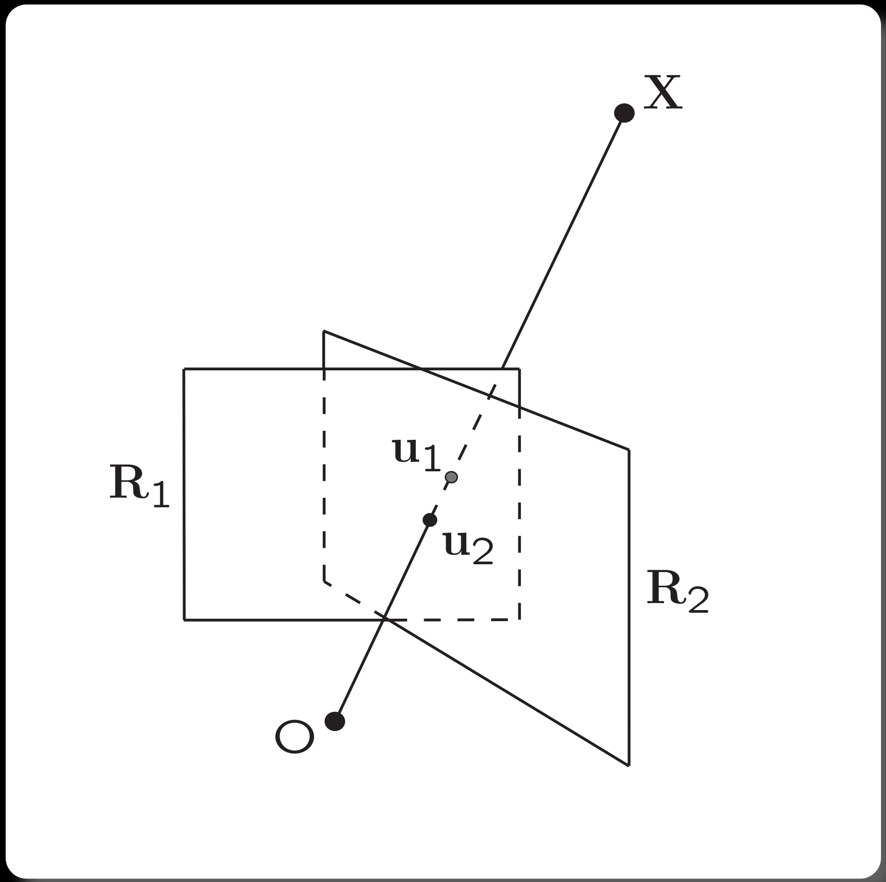
$\mathbf{R}\ (3 \times 3)$: Rotation matrix

$\mathbf{t}\ (3 \times 1)$: Translation vector

2-view Rotational Geometry

- Consider two cameras at the same position (translation)
- WLOG we can put the origin of coordinates there

$$\tilde{\mathbf{u}}_1 = \mathbf{K}_1[\,\mathbf{R}_1 \mid \mathbf{t}_1\,]\,\tilde{\mathbf{X}}$$



- Set translation to 0

$$\tilde{\mathbf{u}}_1 = \mathbf{K}_1[\,\mathbf{R}_1 \mid \mathbf{0}\,]\,\tilde{\mathbf{X}}$$

- Remember $\tilde{\mathbf{X}} \sim [X,\ Y,\ Z,\ 1]^T$ so

$$\underline{\tilde{\mathbf{u}}_1 = \mathbf{K}_1\mathbf{R}_1\mathbf{X}}$$

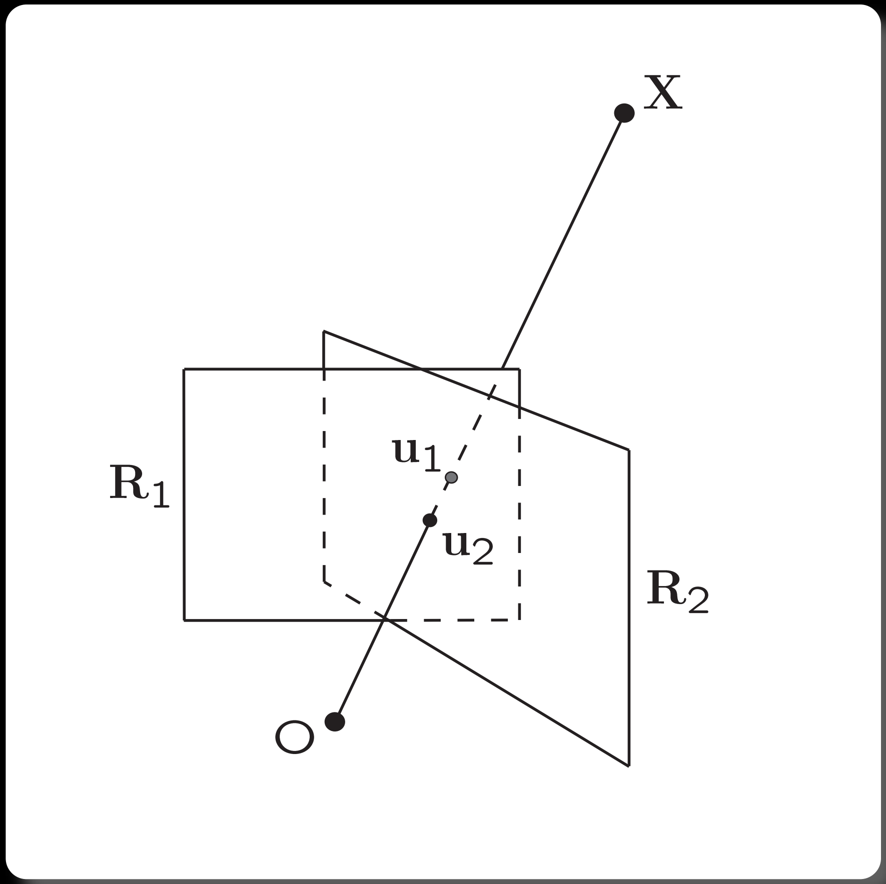
(where $\mathbf{X} = [X,\ Y,\ Z]^T$)

2-view Rotational Geometry

- Add a second camera (same translation but different rotation and intrinsic matrix)

$$\tilde{\mathbf{u}}_1 = \mathbf{K}_1 \mathbf{R}_1 \mathbf{X}$$

$$\tilde{\mathbf{u}}_2 = \mathbf{K}_2 \mathbf{R}_2 \mathbf{X}$$



- Now eliminate **X**

$$\mathbf{X} = \mathbf{R}_1^T \mathbf{K}_1^{-1} \tilde{\mathbf{u}}_1$$

- Substitute in equation 1

$$\underline{\tilde{\mathbf{u}}_2 = \mathbf{K}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{K}_1^{-1} \tilde{\mathbf{u}}_1}$$

This is a 3x3 matrix -- a (special form) of **homography**

Computing H: Quiz

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Each correspondence between 2 images generates _____ equations
- A homography has _____ degrees of freedom
- _____ point correspondences are needed to compute the homography
- Rearranging to make H the subject leads to an equation of the form

$$\mathbf{Mh} = \mathbf{0}$$

- This can be solved by _____

Computing H: Quiz

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

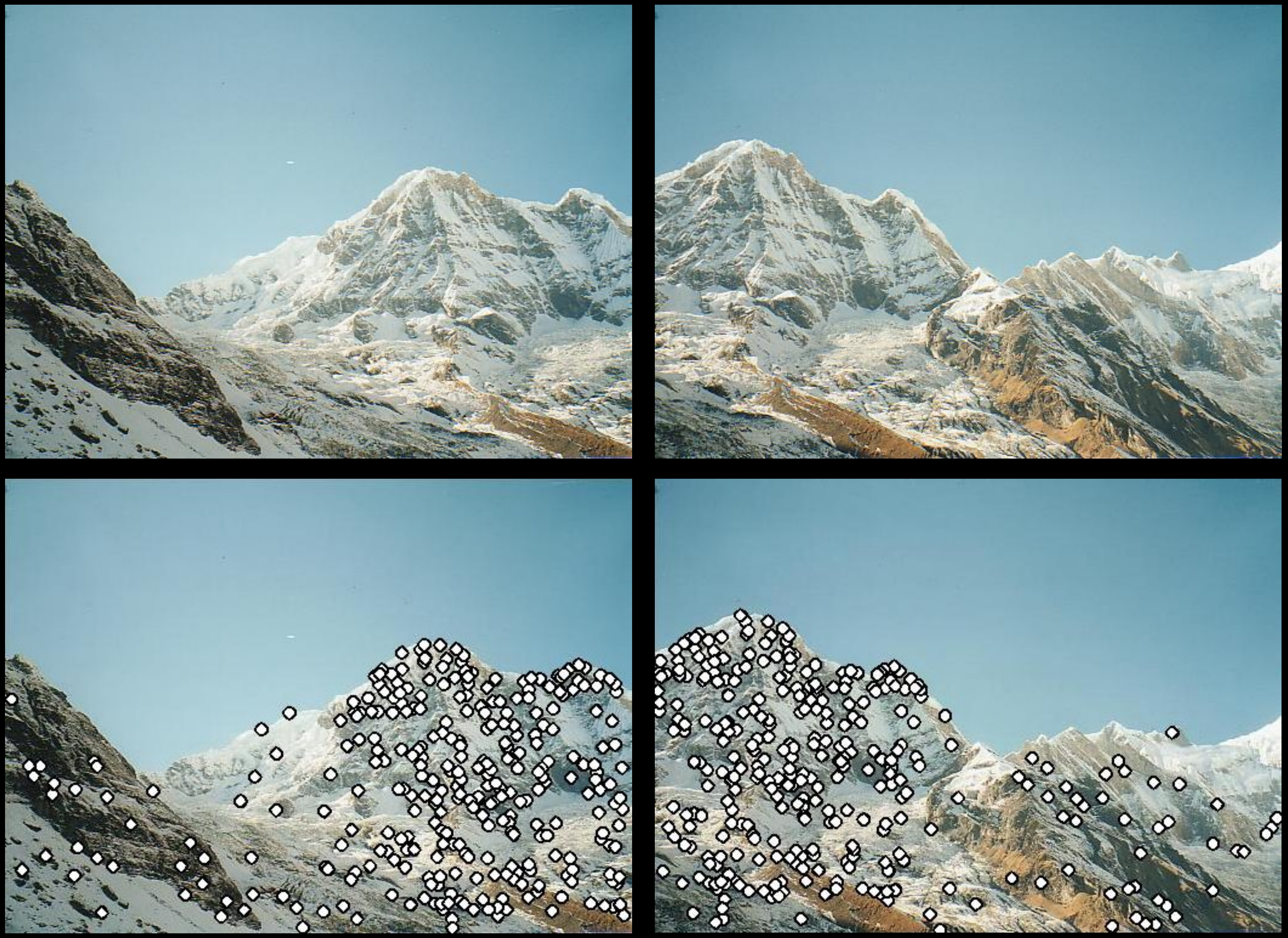
- Each correspondence between 2 images generates _____ equations
- A homography has _____ degrees of freedom
- _____ point correspondences are needed to compute the homography
- Rearranging to make H the subject leads to an equation of the form

$$\mathbf{Mh} = \mathbf{0}$$

- This can be solved by _____

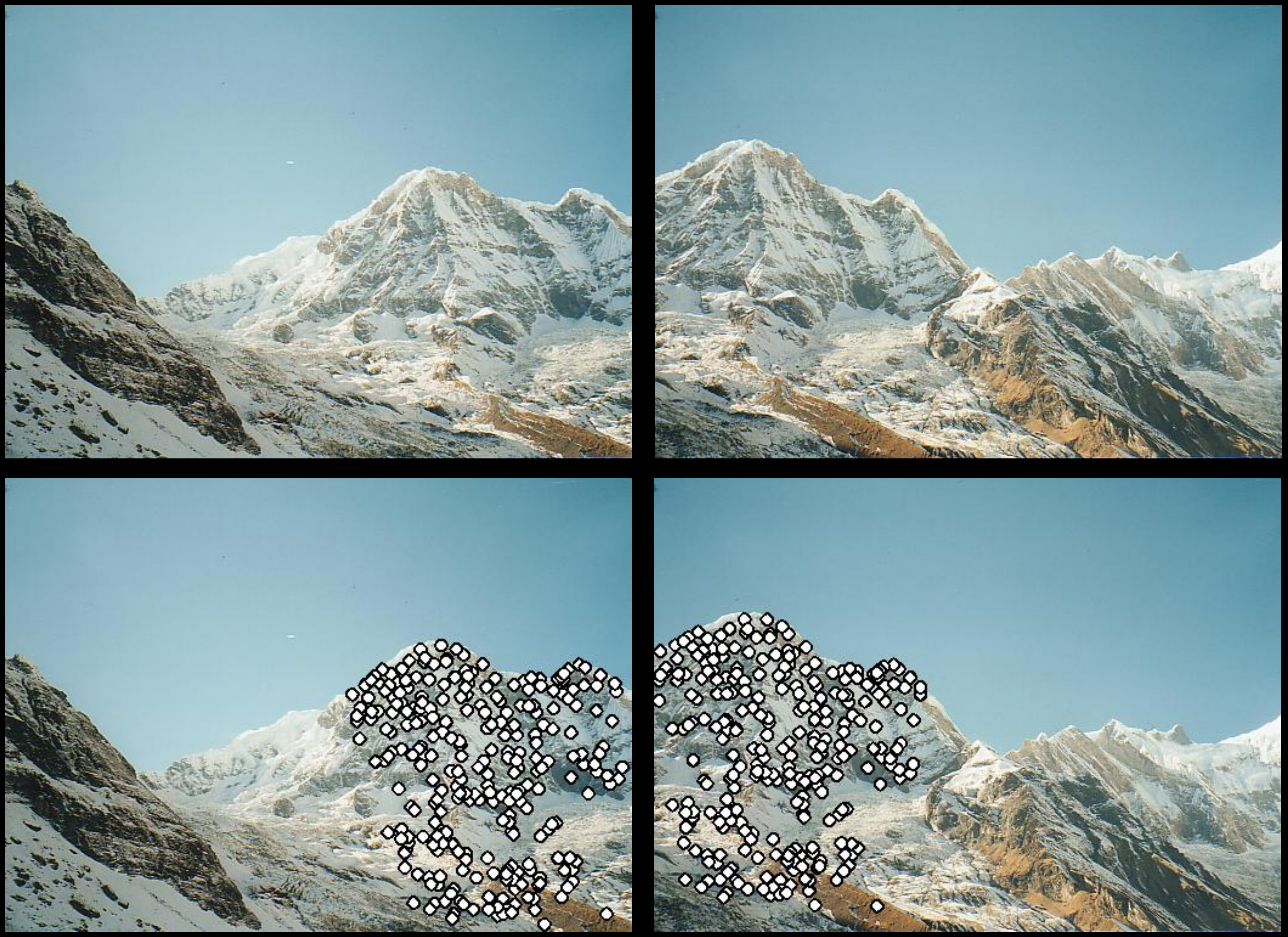
Finding Consistent Matches

- Raw SIFT correspondences (contains **outliers**)



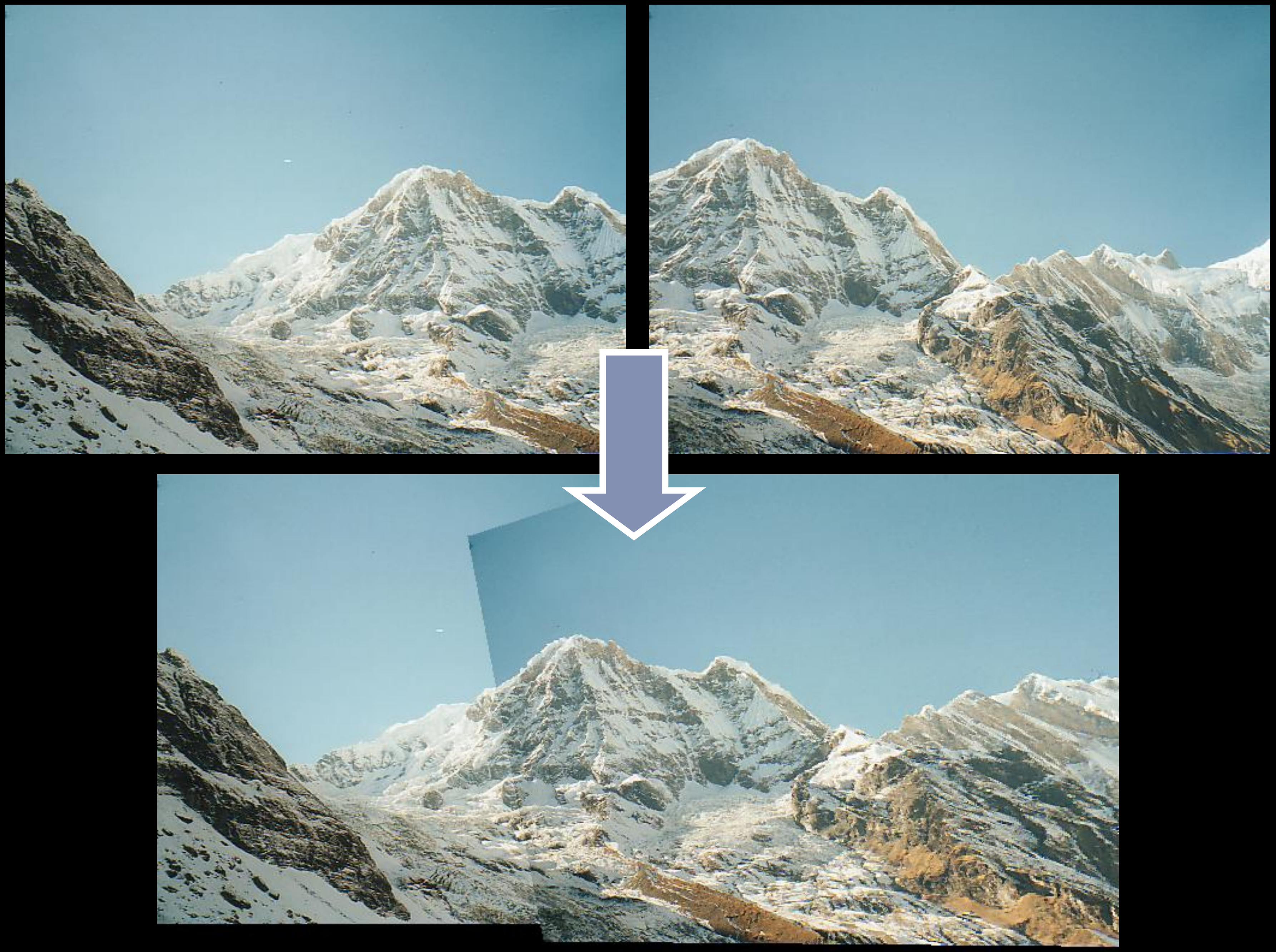
Finding Consistent Matches

- SIFT matches consistent with a rotational **homography**



Finding Consistent Matches

- Warp images to common coordinate frame



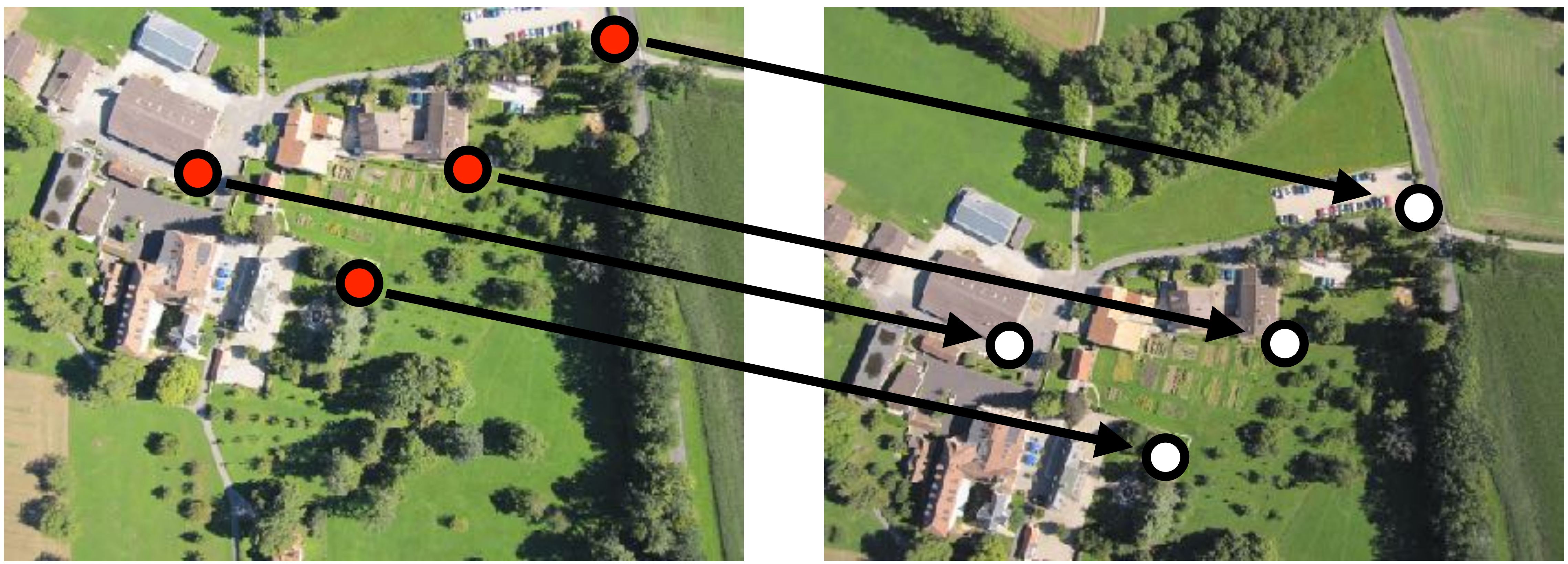
2-view Alignment + RANSAC

- 2-view alignment: linear equations
- Least squares and outliers
- Robust estimation via sampling

[Szeliski 6.1]

Image Alignment

- Find corresponding (matching) points between the images

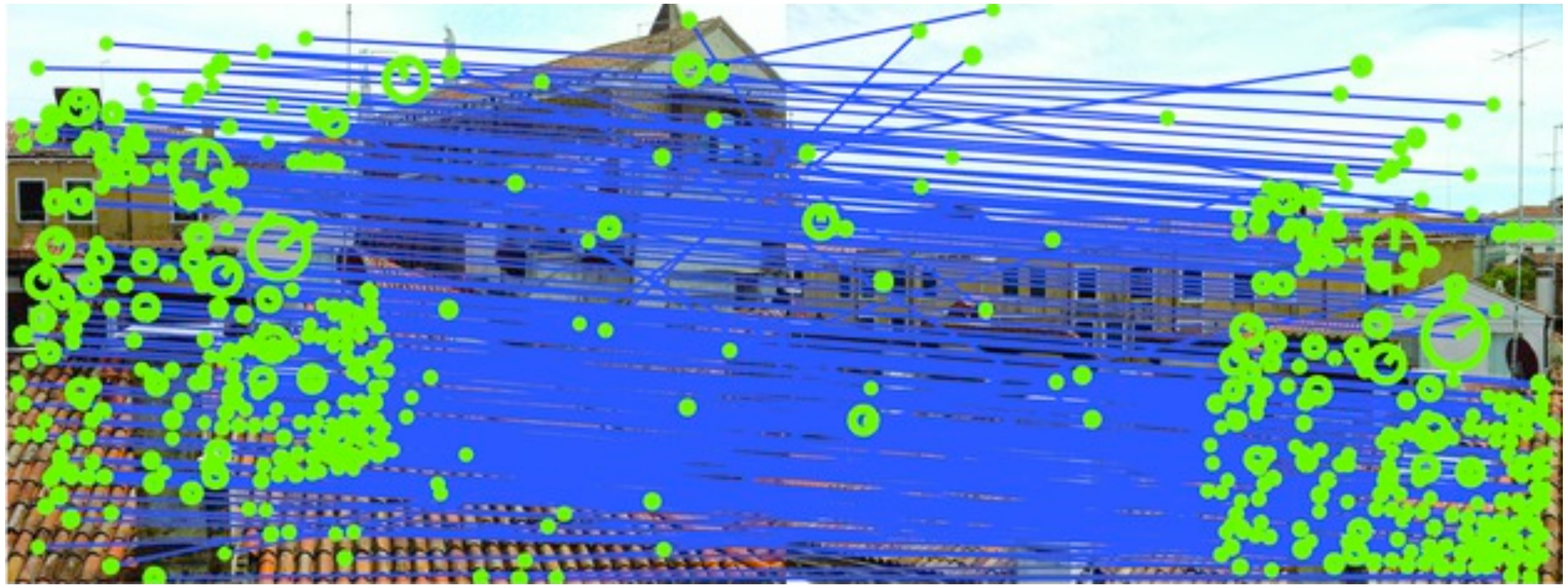


$$\mathbf{u} = \mathbf{H}\mathbf{x}$$

2 points for Similarity
3 for Affine
4 for Homography

Image Alignment

- In practice we have many noisy correspondences + **outliers**



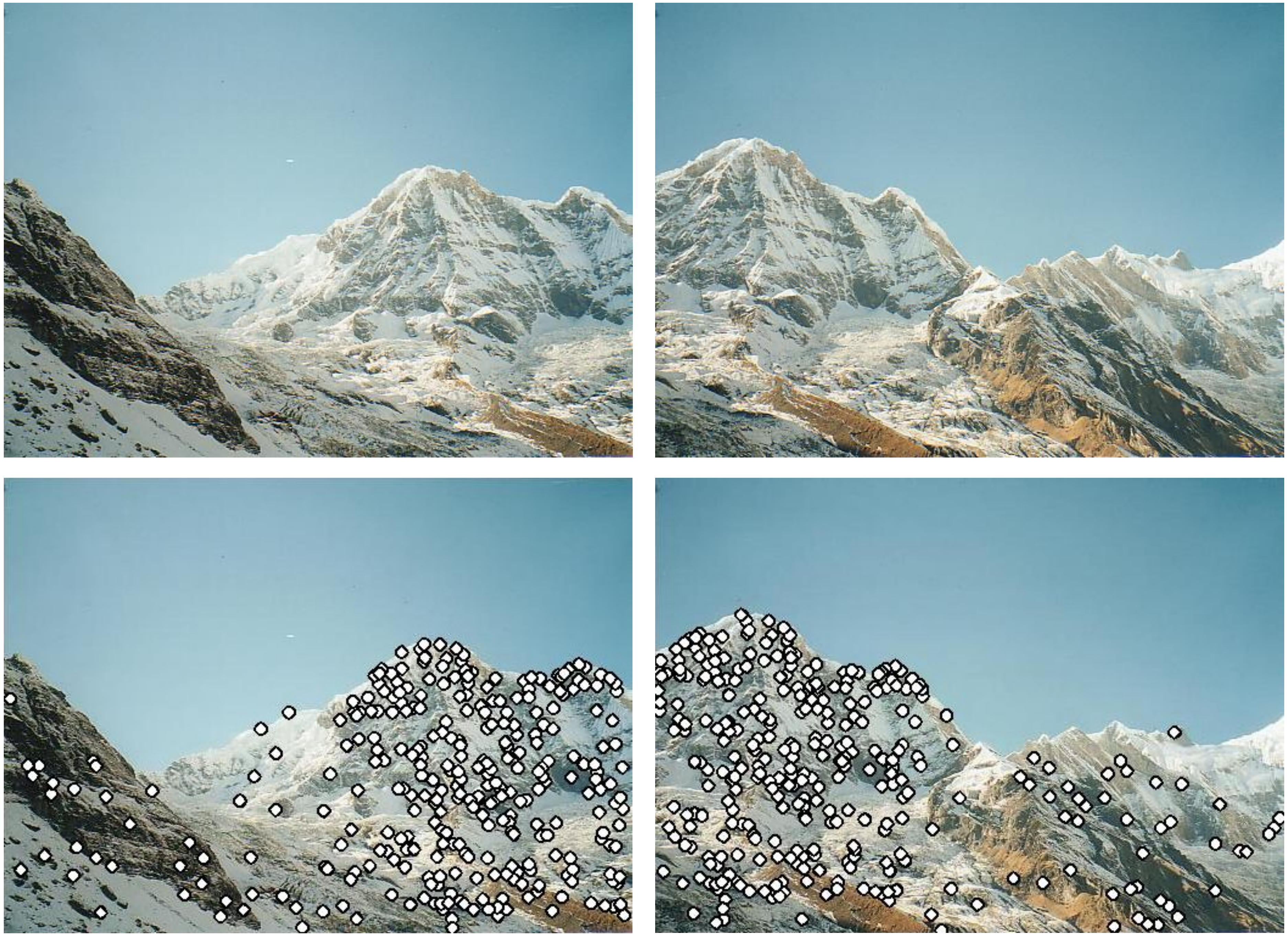
RANSAC algorithm

1. Match feature points between 2 views
2. Select minimal subset of matches*
3. Compute transformation T using minimal subset
4. Check consistency of all points with T — compute projected position and count #inliers with distance < threshold
5. Repeat steps 2-4 to maximise #inliers

* Similarity transform = 2 points, Affine = 3, Homography = 4

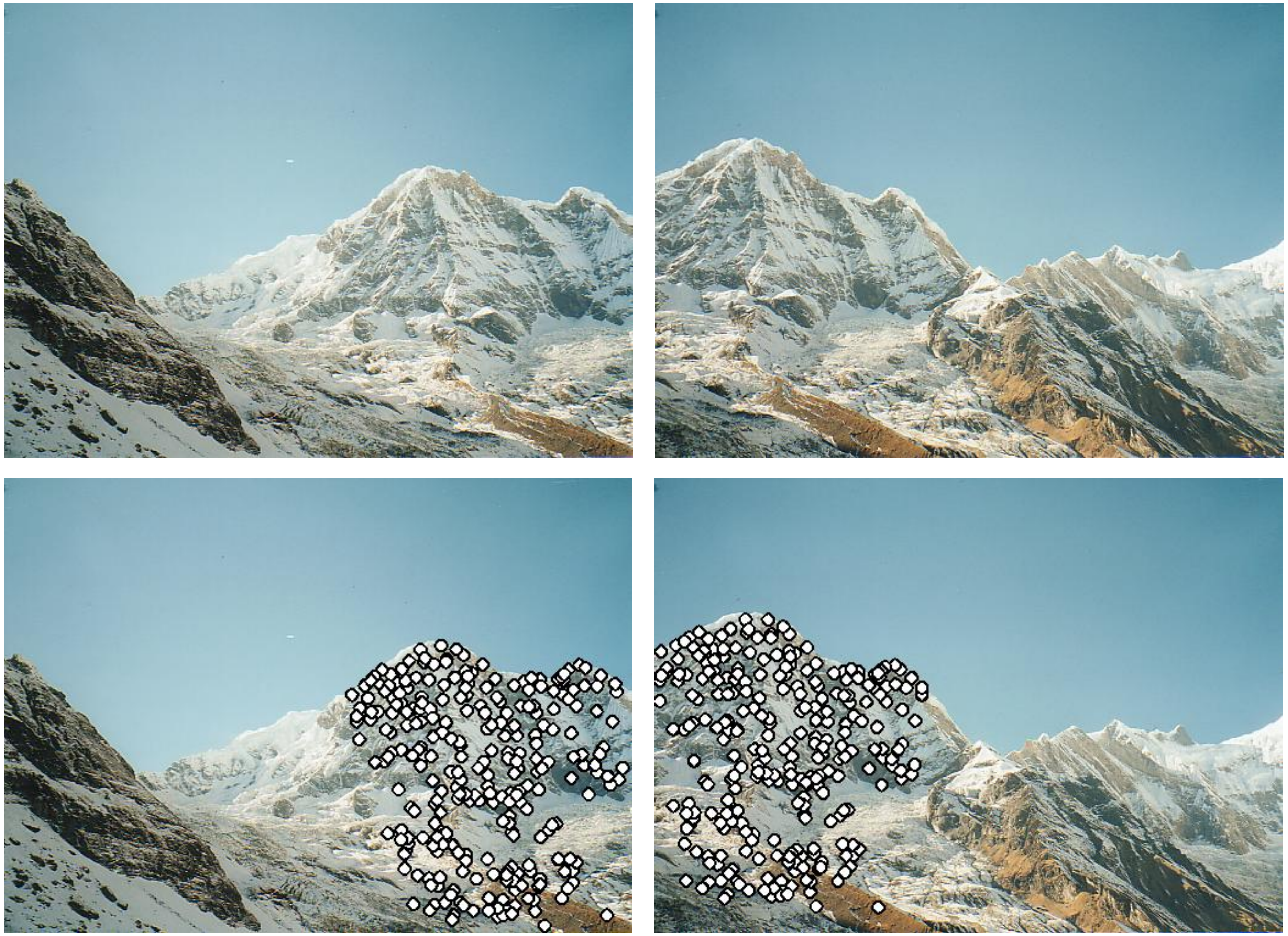
2-view Rotation Estimation

- Find features + raw matches, use RANSAC to find Similarity



2-view Rotation Estimation

- Remove outliers, can now solve for R using least squares



2-view Rotation Estimation

- Final rotation estimation