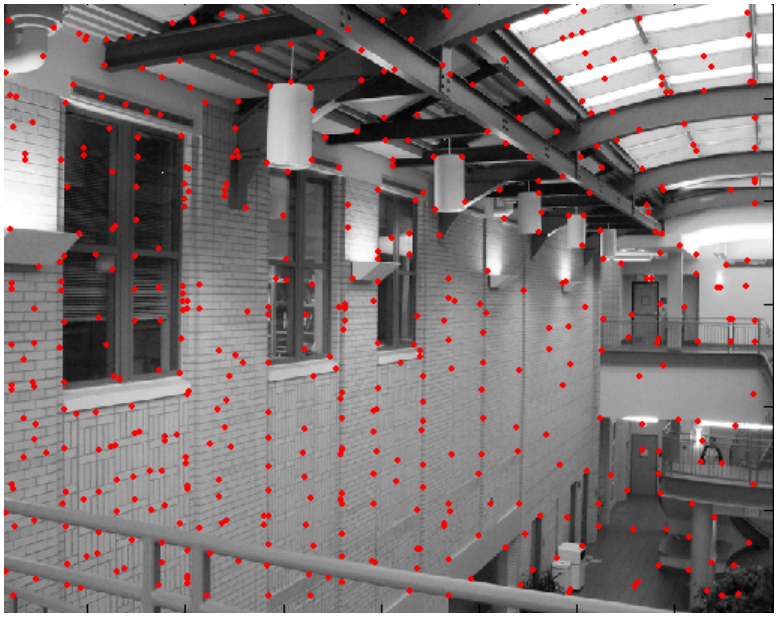
Feature Matching and RANSAC

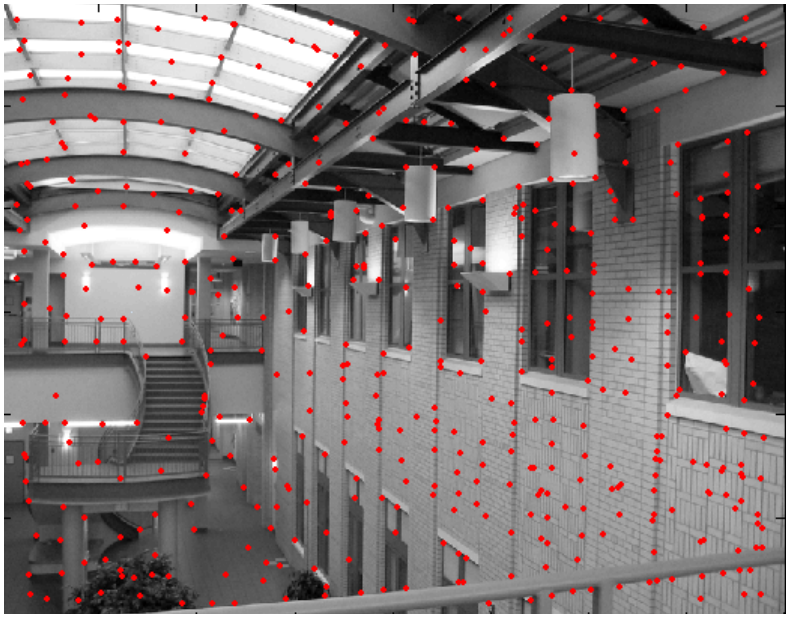
© **Krister Parmstrand**

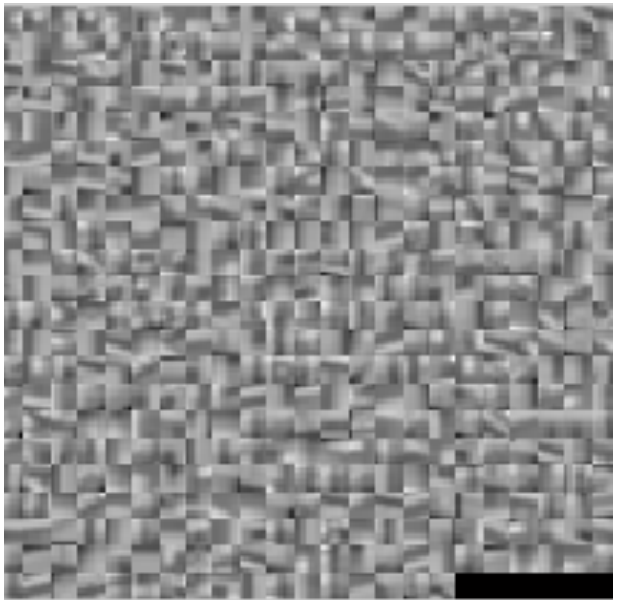
with a lot of slides stolen from
Steve Seitz and Rick Szeliski

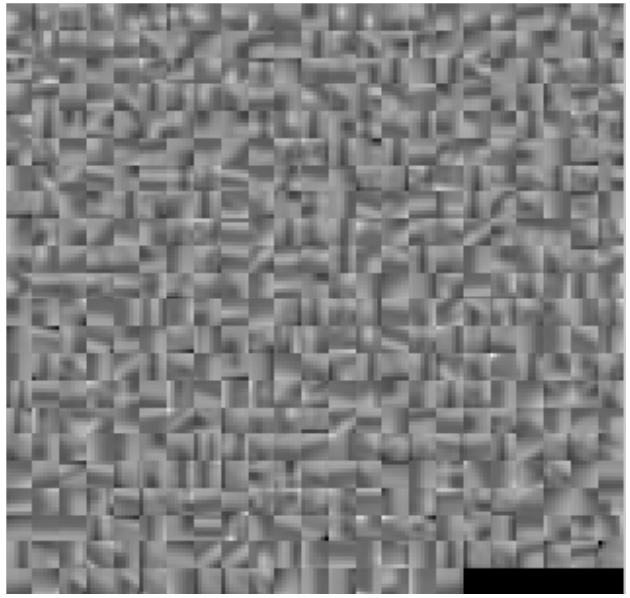
15-463: Computational Photography
Alexei Efros, CMU, Fall 2005

Feature matching





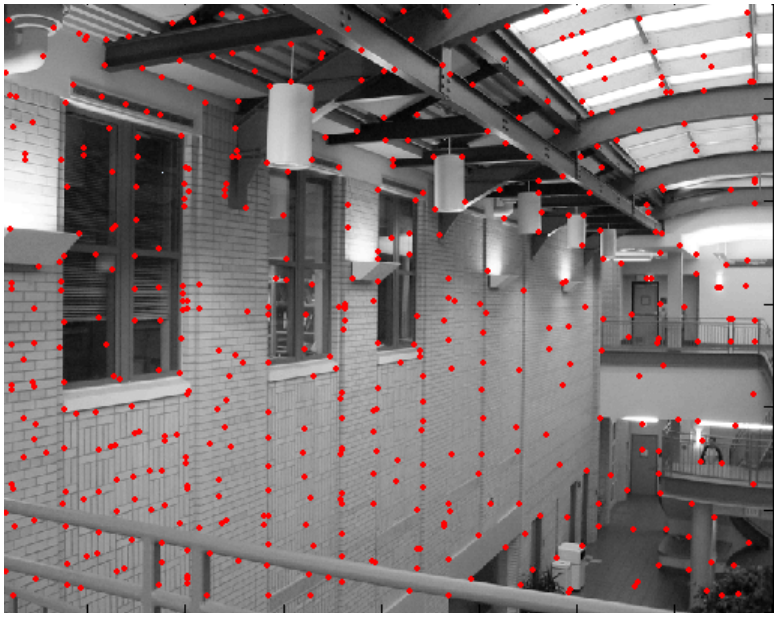


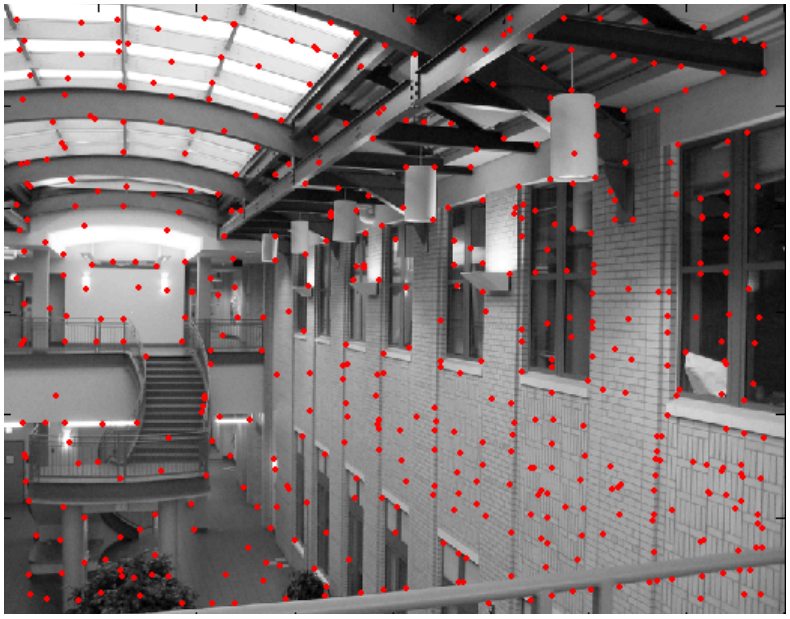


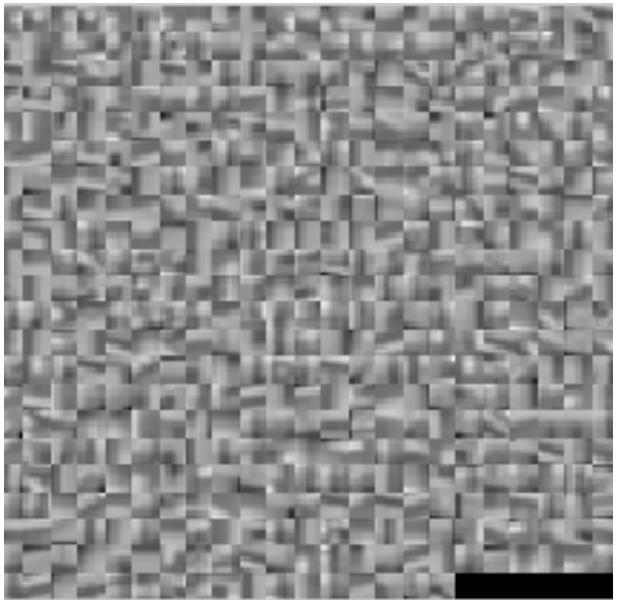
SIFT keypoints

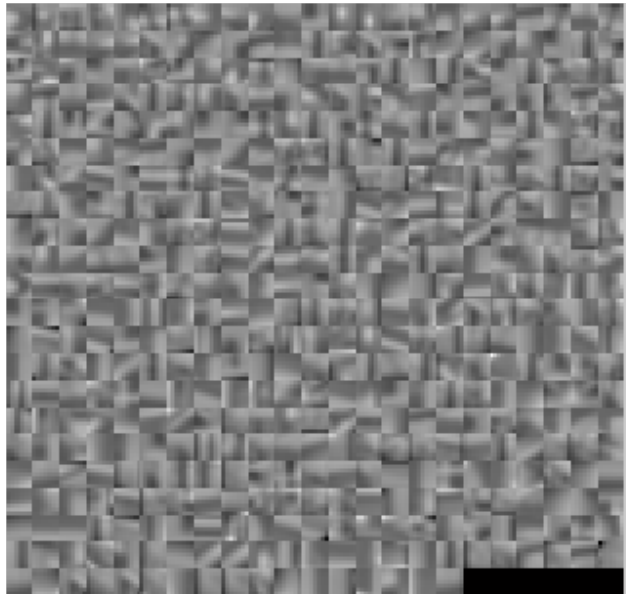
On the previous slide, the red points are all of the SIFT keypoints.

Feature matching

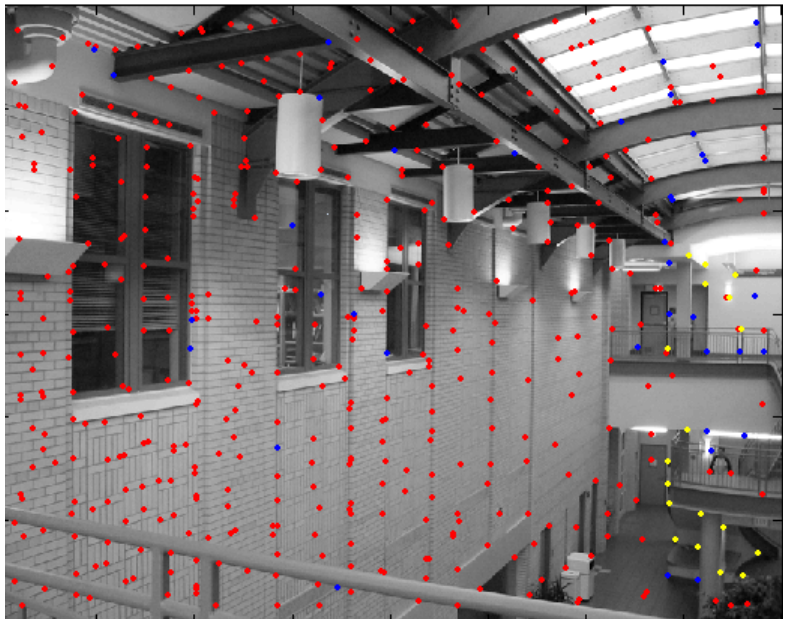


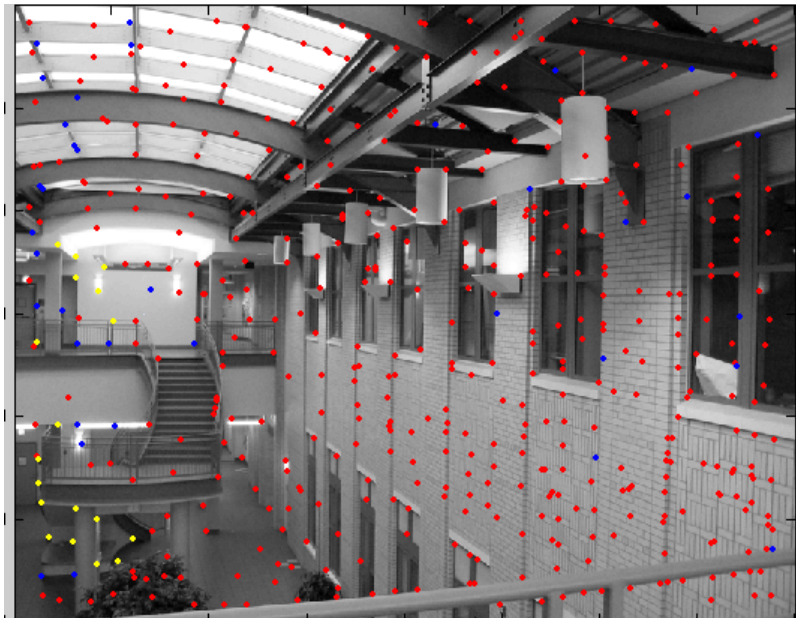






RANSAC





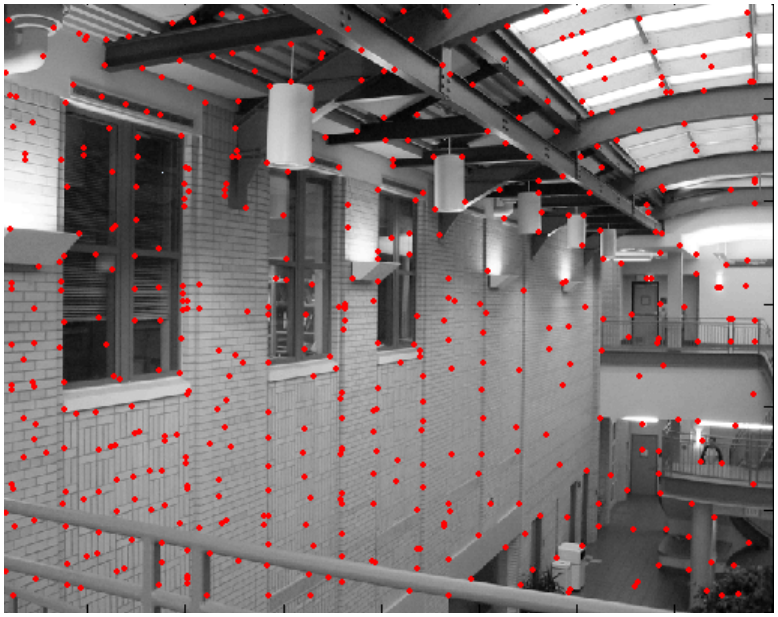
Color coding of points on previous slide

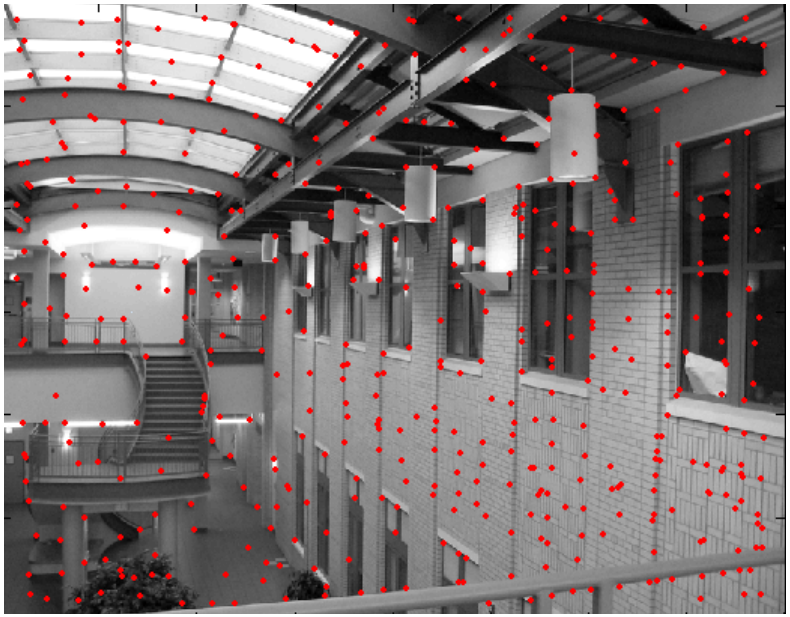
- **Red points**
  - **points without a “good” match in the other image**
  - **In this image, the goodness of the match is decided by looking at the ratio of the distances to the second nearest neighbor and first nearest neighbor. If this ration is high (above some threshold), it is considered a “good” match.**
- **Blue points**
  - **These are points with a “good” match in which the match was wrong, meaning it connected two points that did not actually correspond in the world.**
- **Yellow points**
  - **These are correct matches. We need to run RANSAC until it randomly picked 4 yellow points from among the blue and yellow points (the matches estimated to be “good”).**

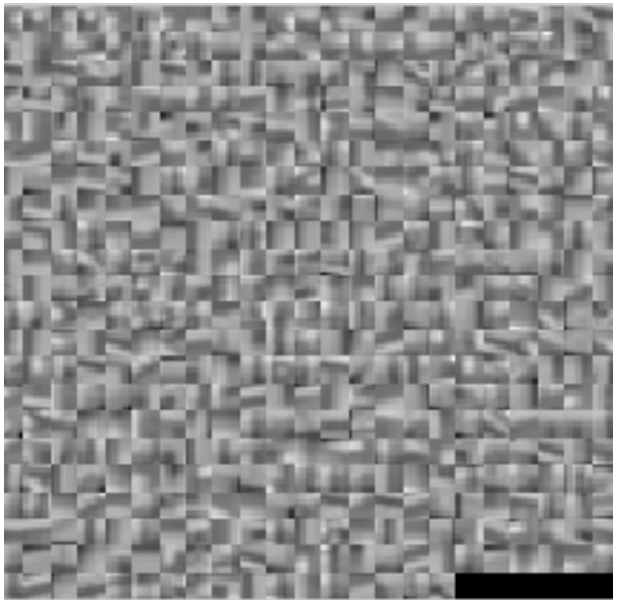
Feature matching

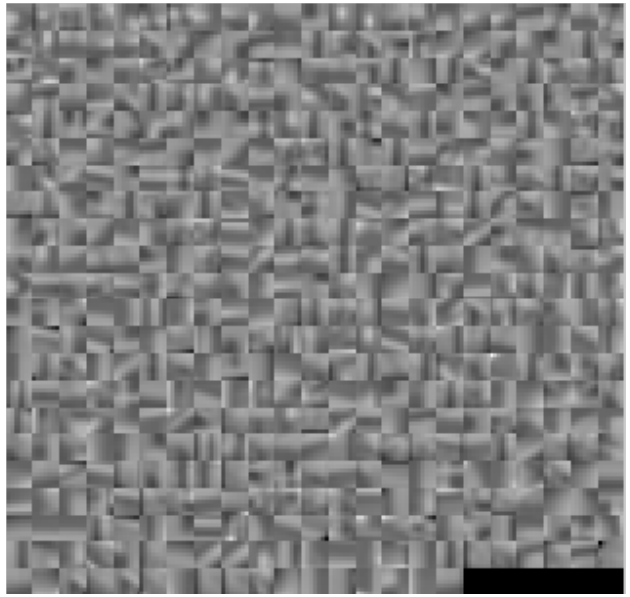
- Exhaustive search
  - for each feature in one image, look at *all* the other features in the other image(s)
- Hashing
  - compute a short descriptor from each feature vector, or hash longer descriptors (randomly)
- Nearest neighbor techniques
  - *kd*-trees and their variants

What about outliers?







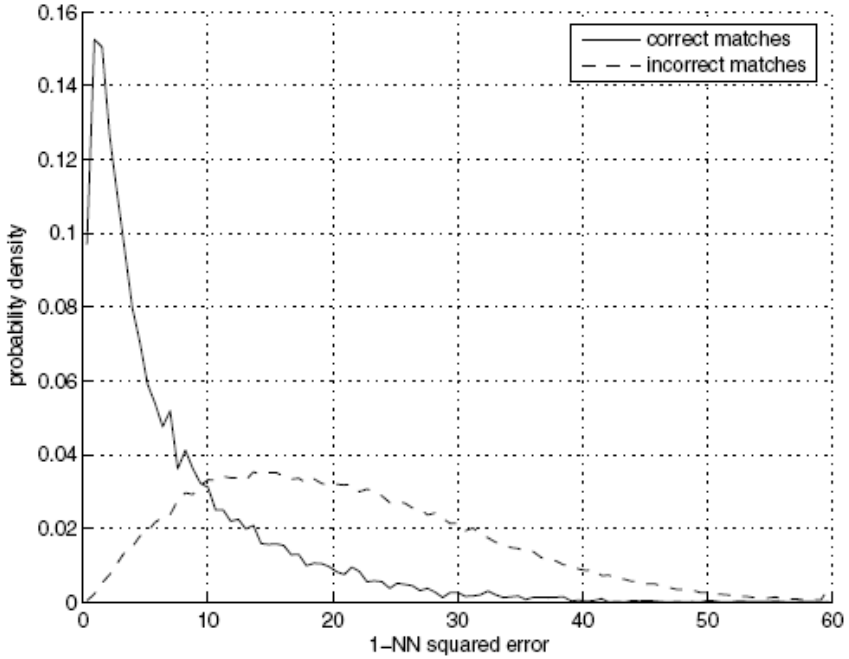


Feature-space outlier rejection

Let's not match all features, but only these that have "similar enough" matches?

How can we do it?

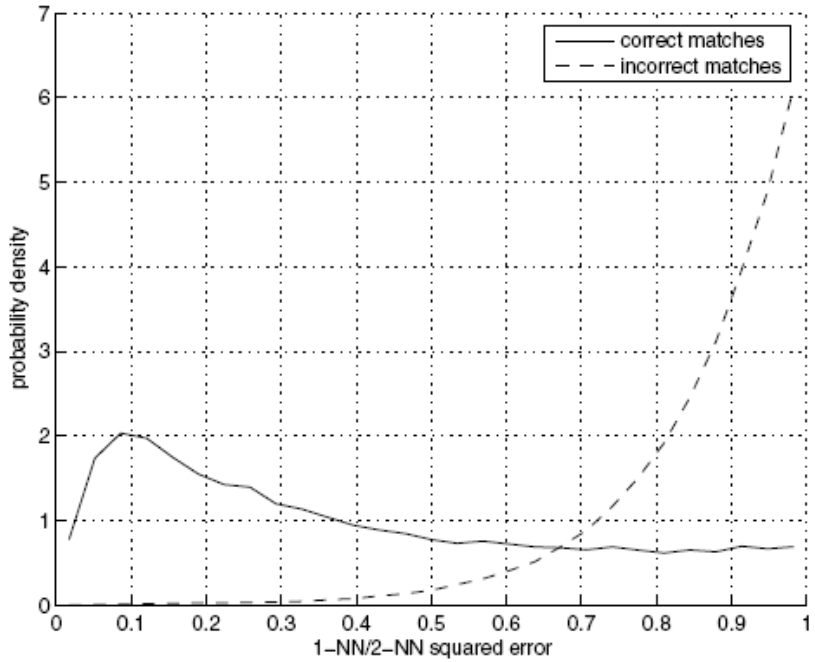
- SSD(patch1,patch2) < threshold
- How to set threshold?



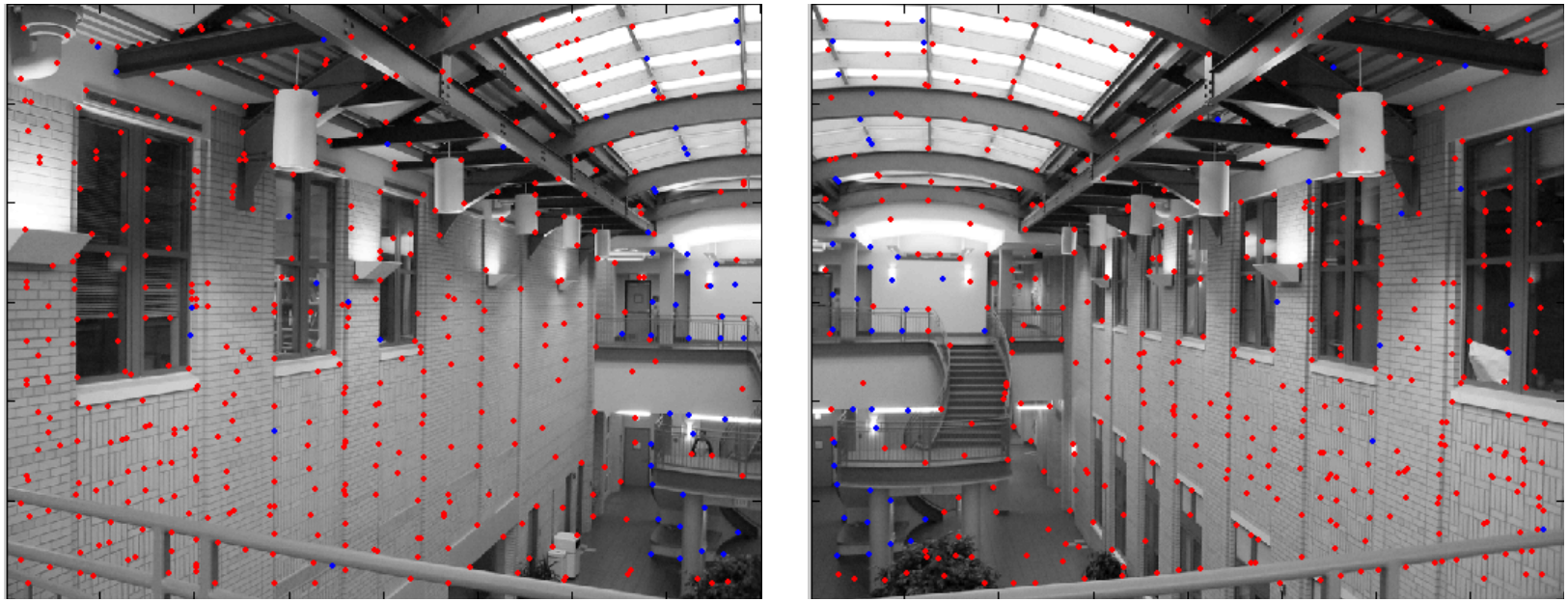
Feature-space outlier rejection

A better way [Lowe, 1999]:

- 1-NN: SSD of the closest match
- 2-NN: SSD of the second-closest match
- Look at how much better 1-NN is than 2-NN, e.g. 1-NN/2-NN
- That is, is our best match so much better than the rest?



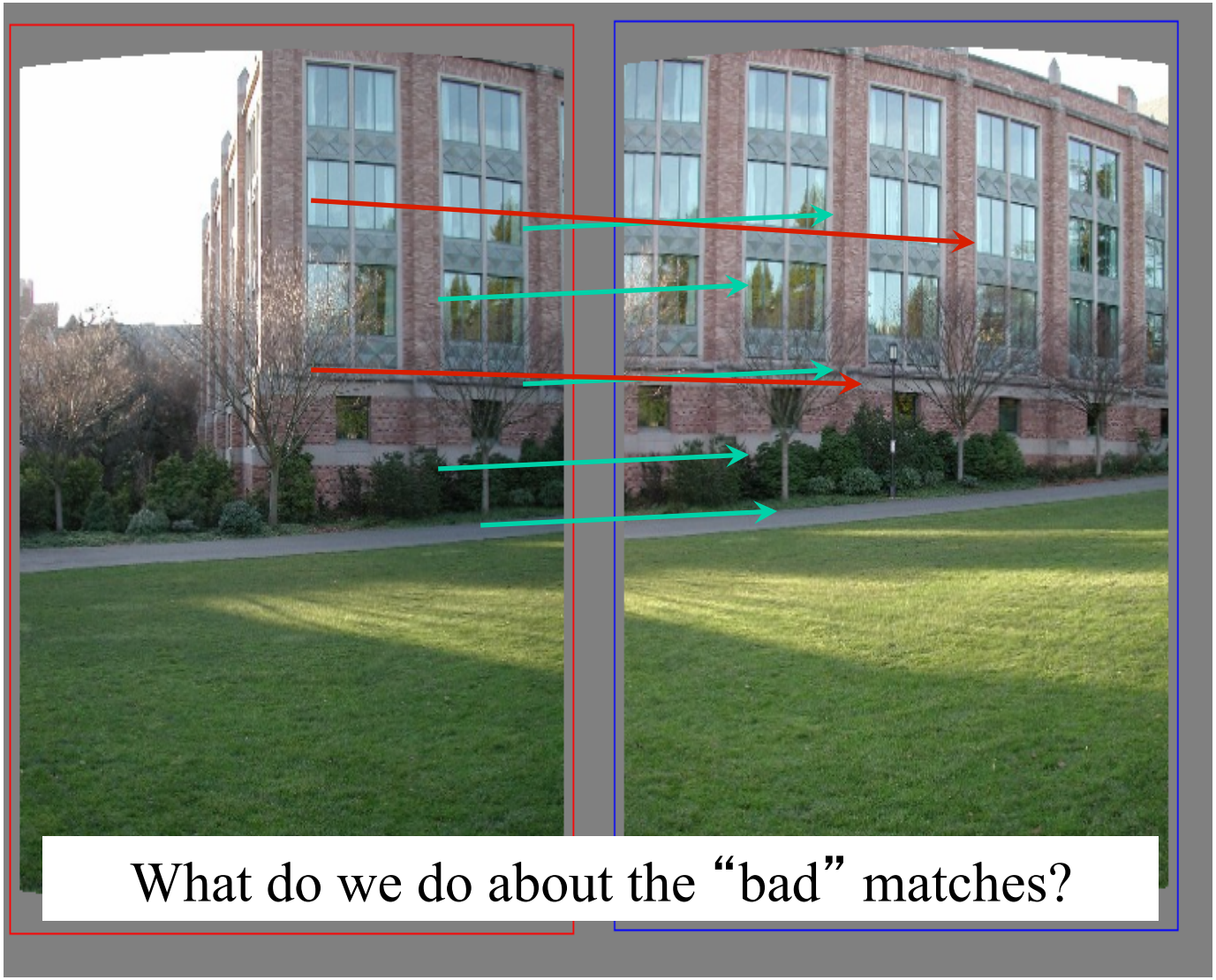
Feature-space outliner rejection



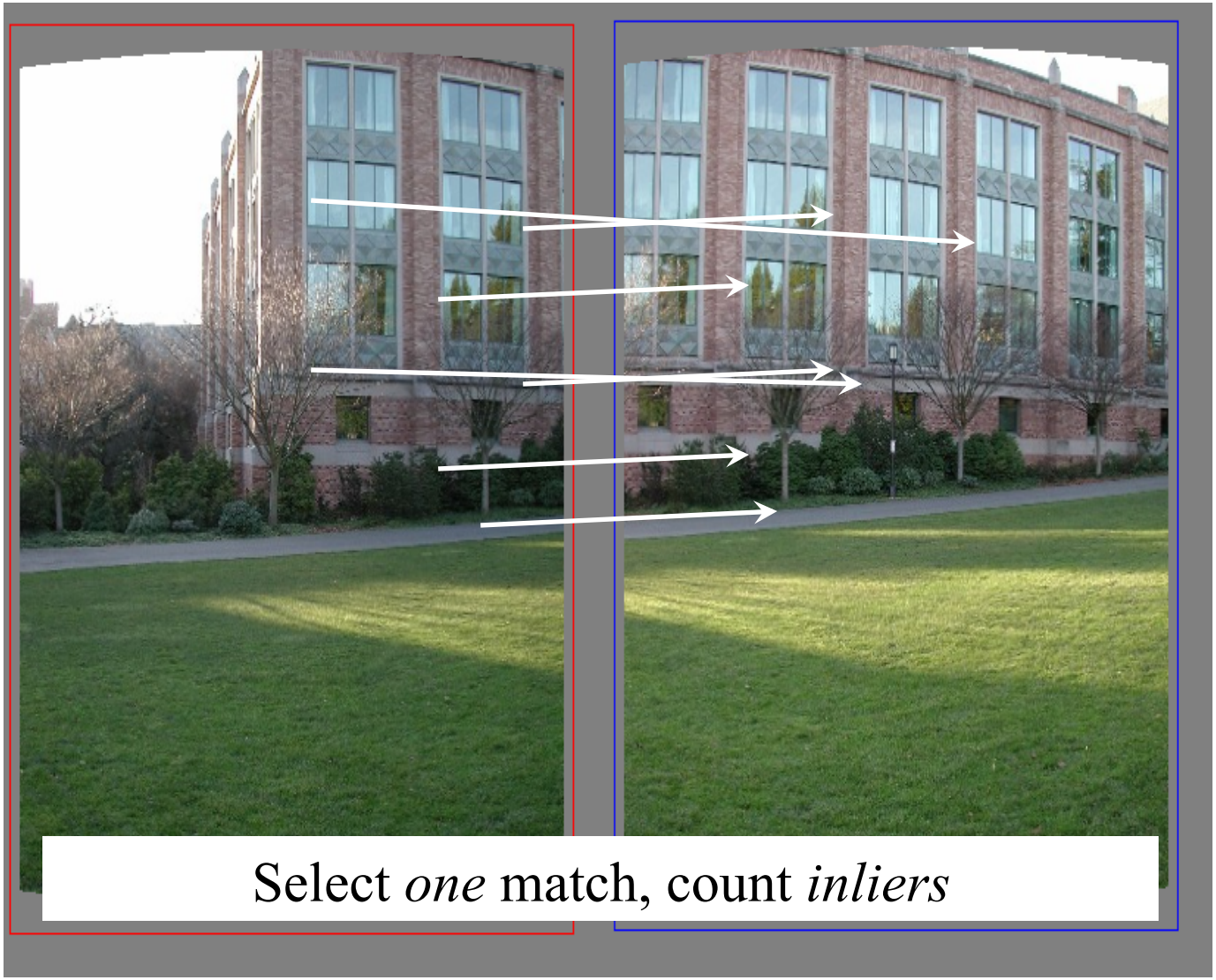
Can we now compute H from the blue points?

- No! Still too many outliers…
- What can we do?

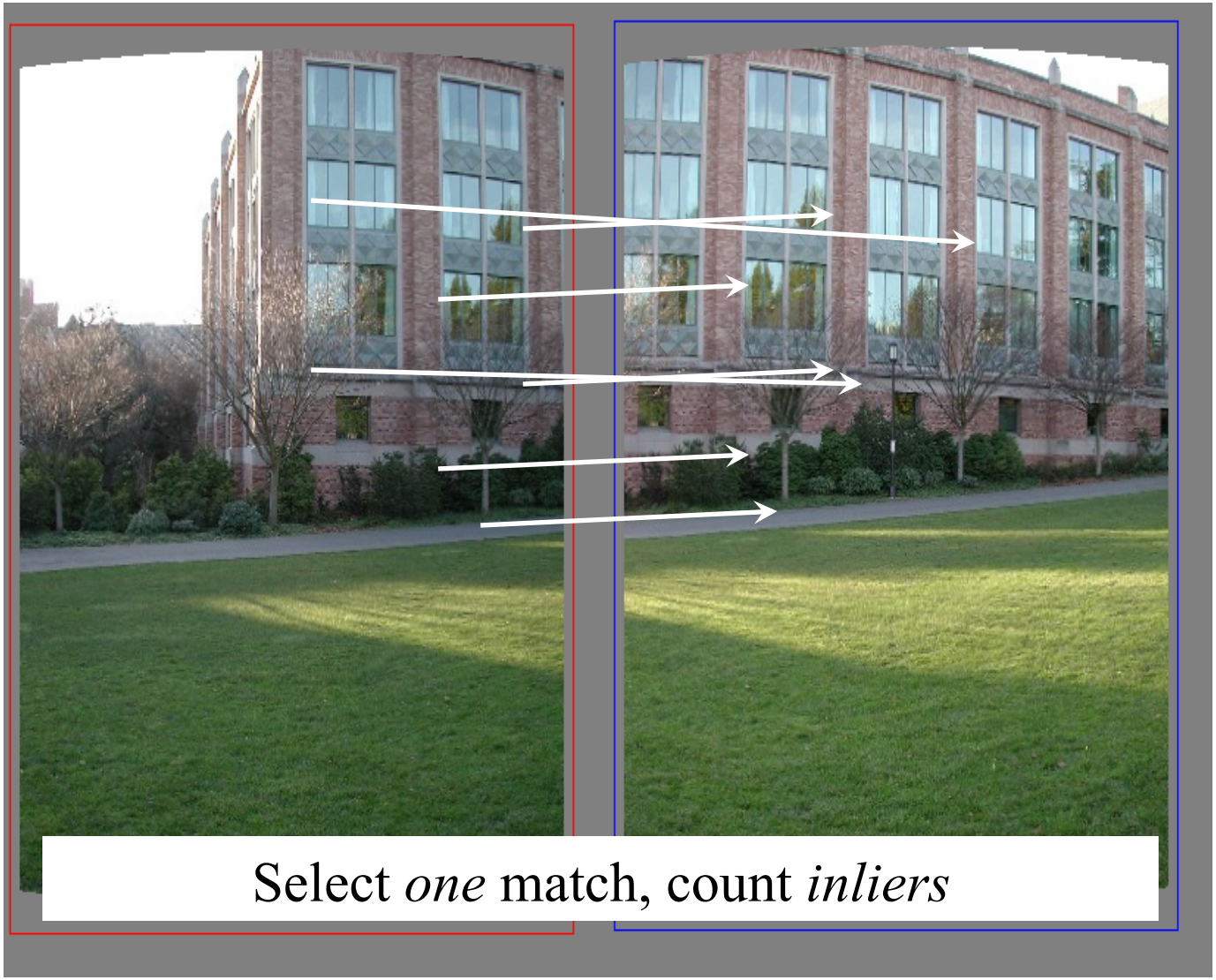
Matching features



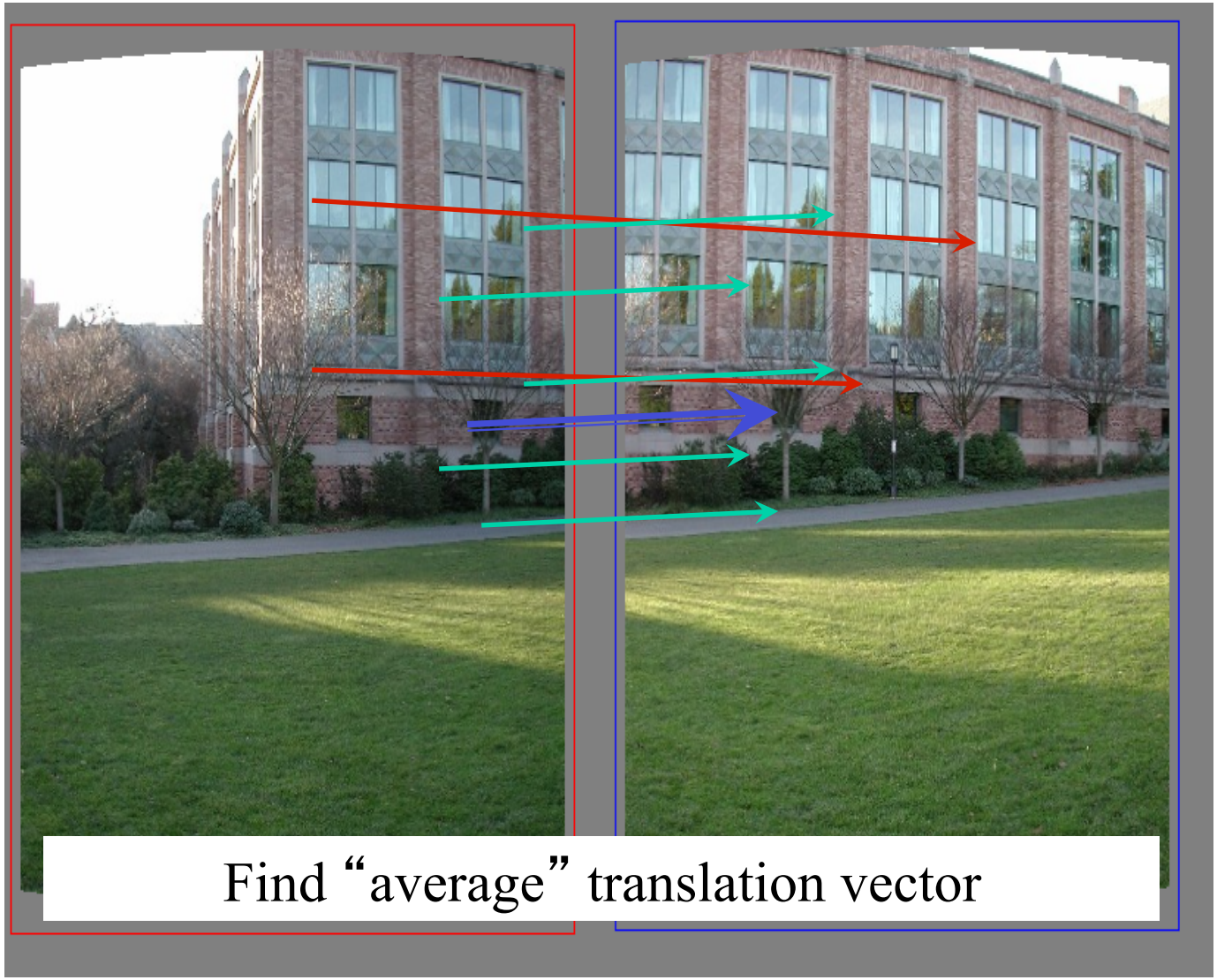
RAndom SAmple Consensus



RAndom SAmple Consensus



Least squares fit

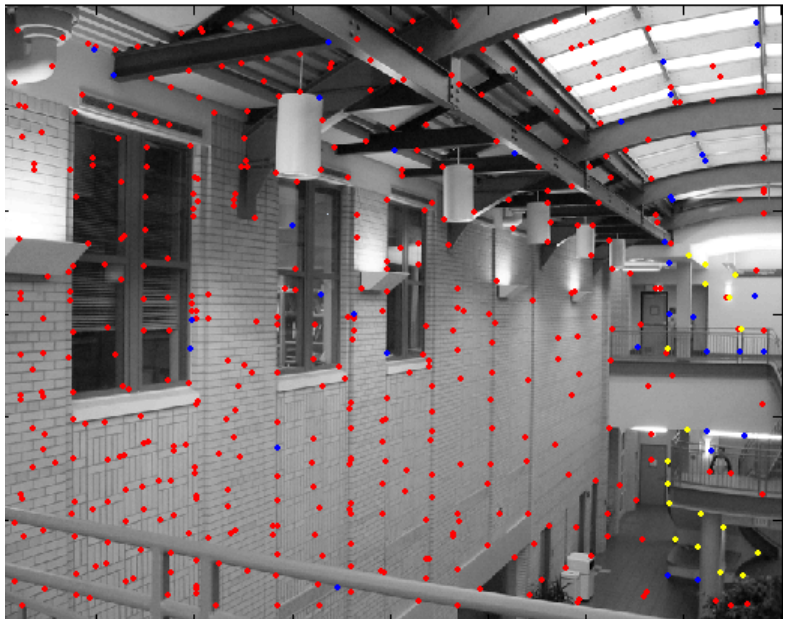


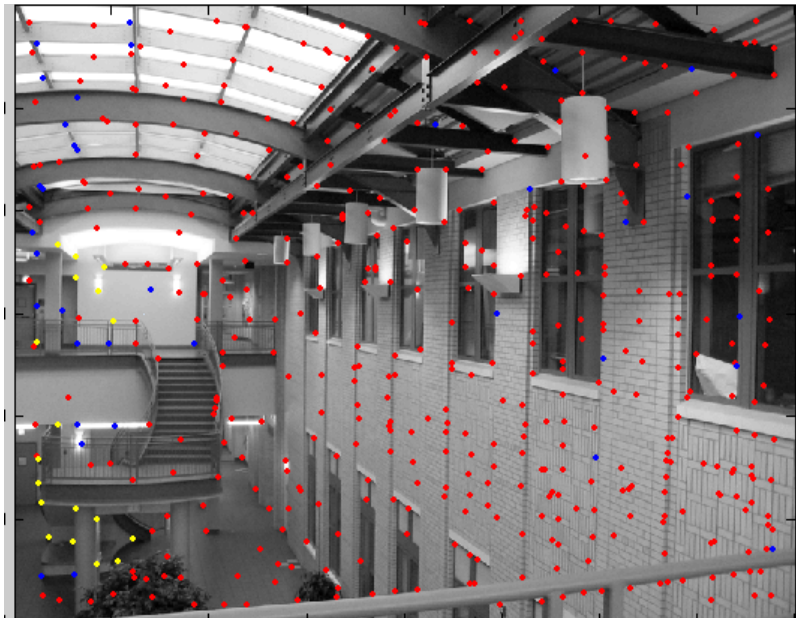
RANSAC for estimating homography

RANSAC loop:

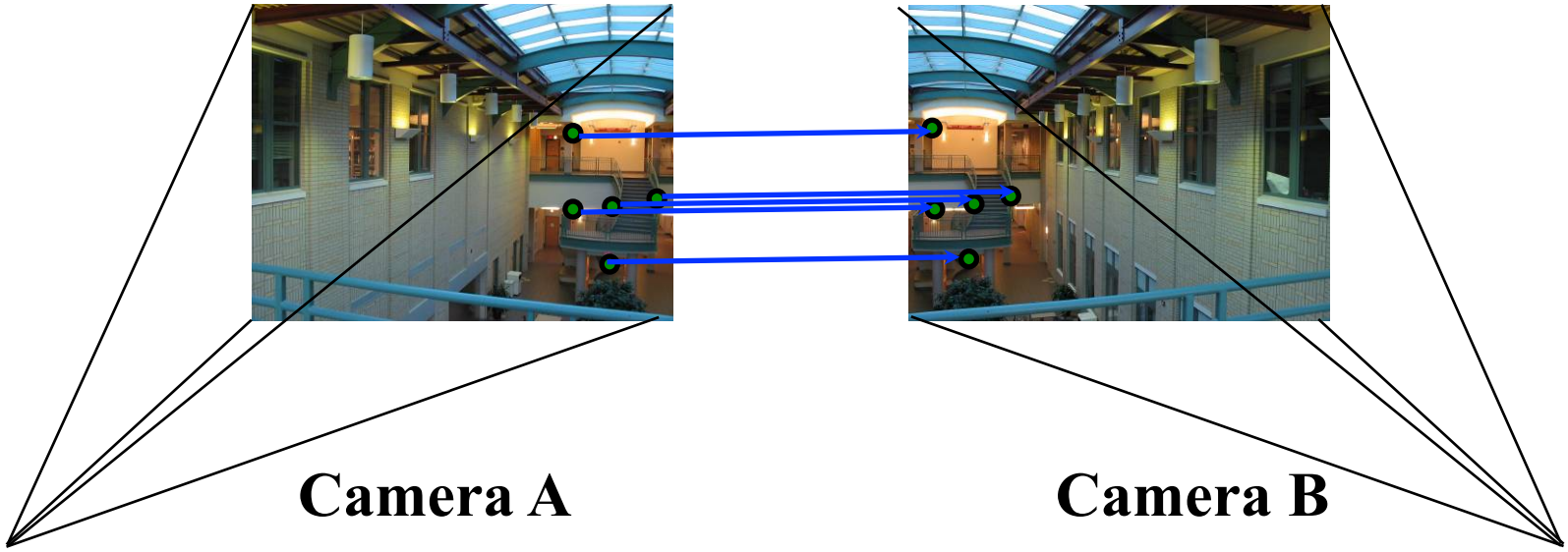
1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \boldsymbol{H}\, p_{i)} < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

RANSAC

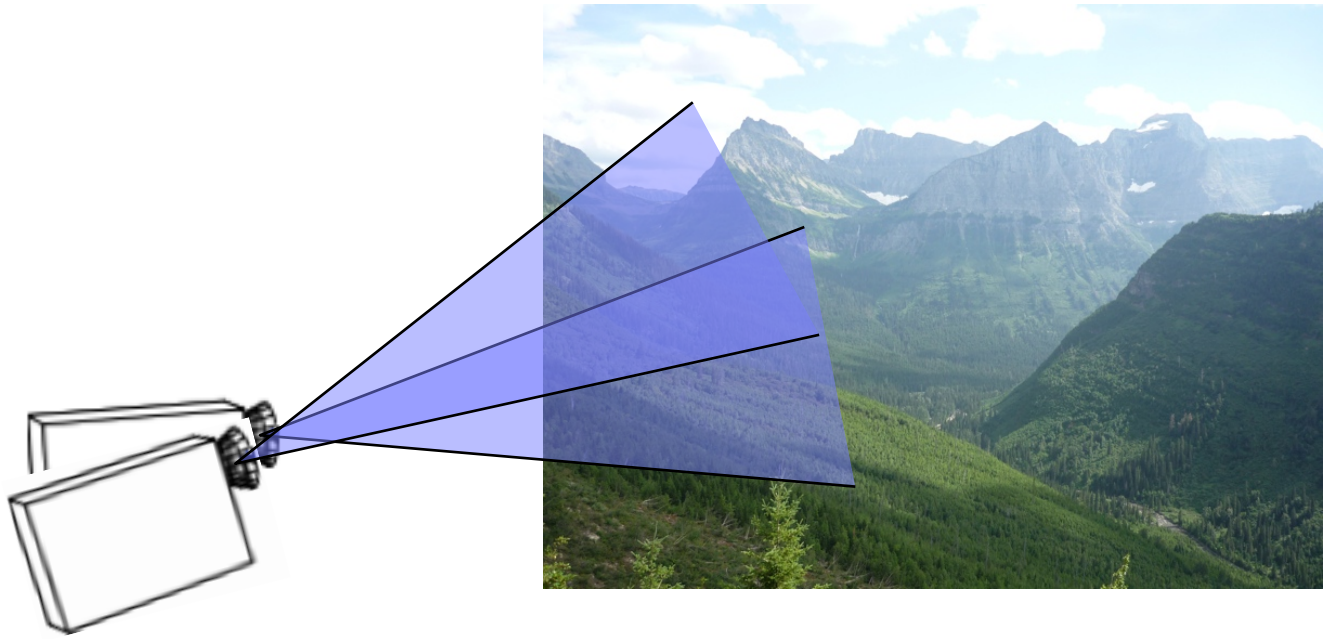




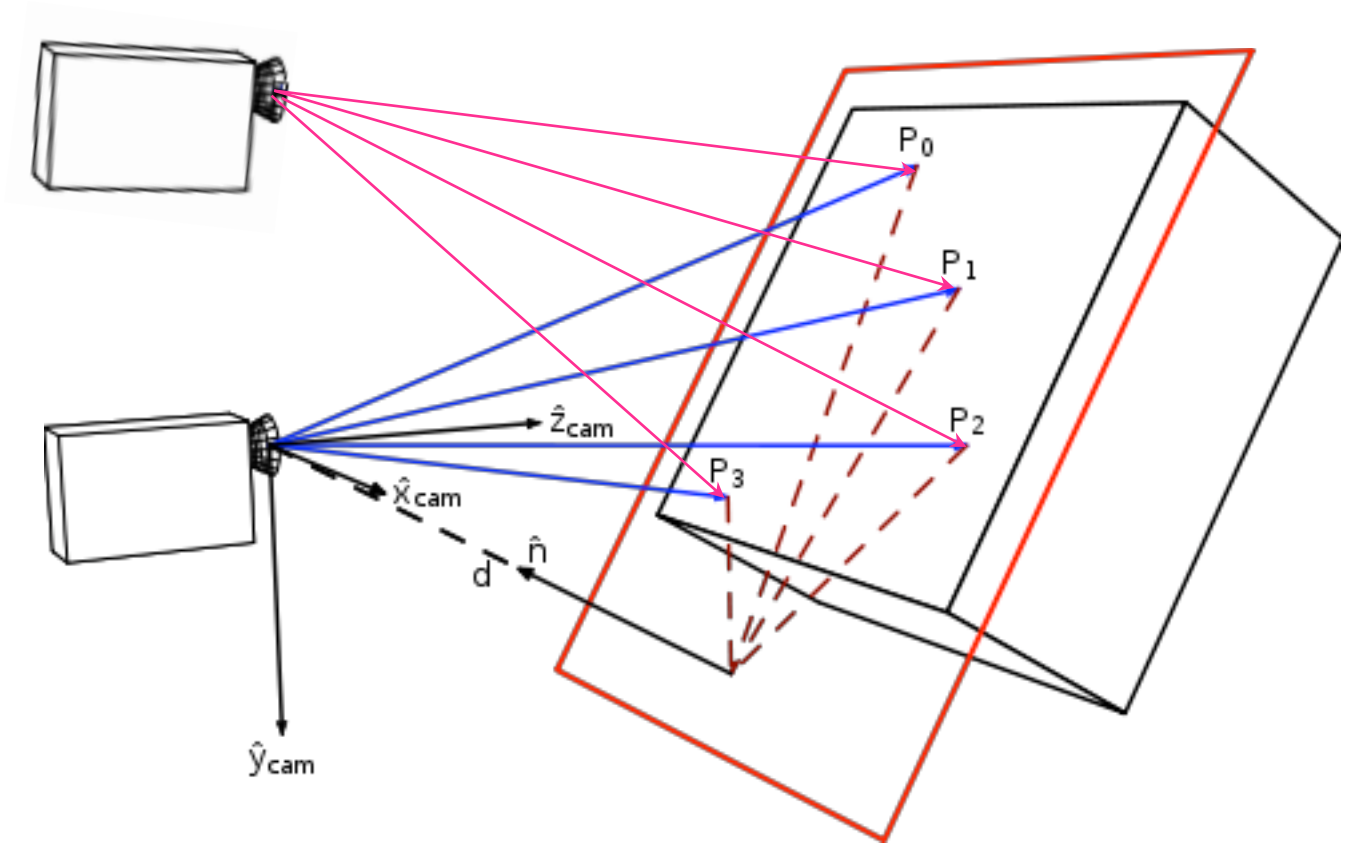
Under what conditions can you know where to translate each point of image A to where it would appear in camera B (with calibrated cameras), knowing nothing about image depths?



(a) camera rotation



and (b) imaging a planar surface



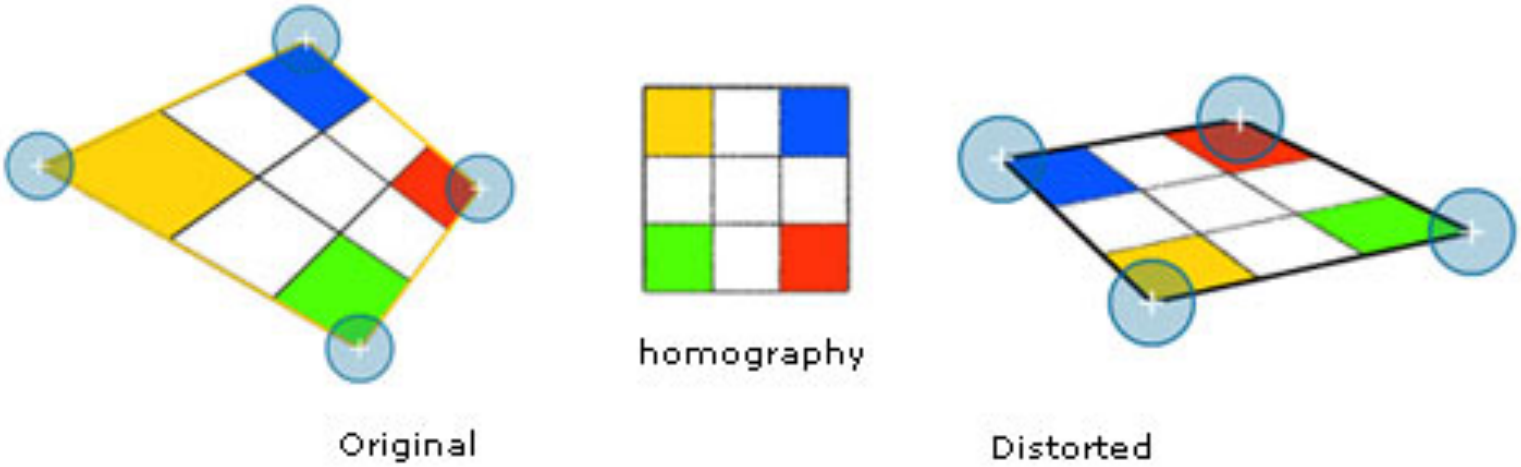
Aligning images: translation?

left on top

right on top

Translations are not enough to align the images

Homography example



UMass Lab for Perceptual Robotics

Example: Recognising Panoramas

M. Brown and D. Lowe,
University of British Columbia

Why “Recognising Panoramas”?

Why “Recognising Panoramas”?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

Why “Recognising Panoramas”?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

Why “Recognising Panoramas”?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

Why "Recognising Panoramas"?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

- 2D Rotations (θ, ϕ)
  - Ordering $\not\Rightarrow$ matching images

Why "Recognising Panoramas"?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

- 2D Rotations (θ, ϕ)
  - Ordering $\not\Rightarrow$ matching images

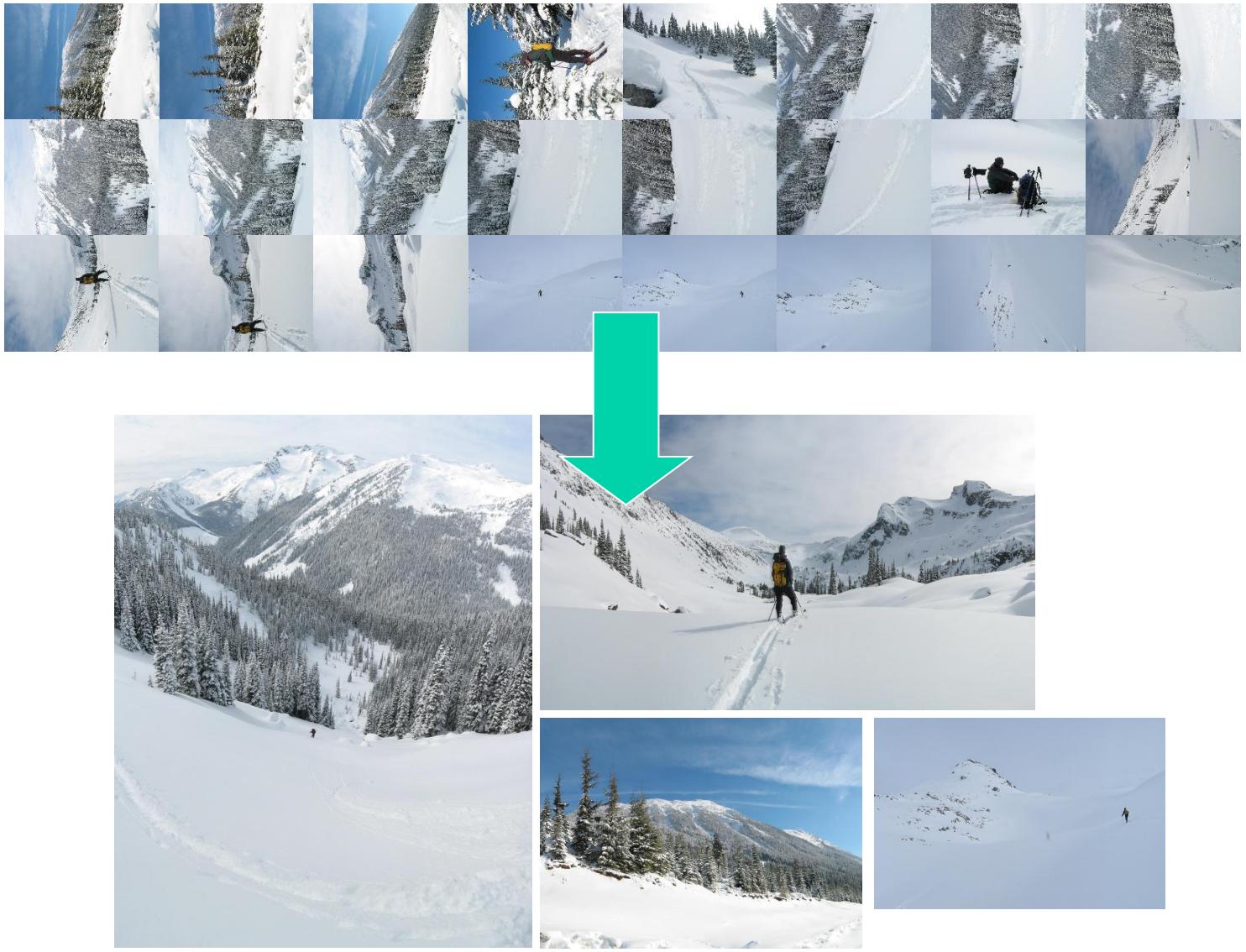
Why "Recognising Panoramas"?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

- 2D Rotations (θ, ϕ)
  - Ordering $\not\Rightarrow$ matching images

Why “Recognising Panoramas”?



Overview

Feature Matching

Image Matching

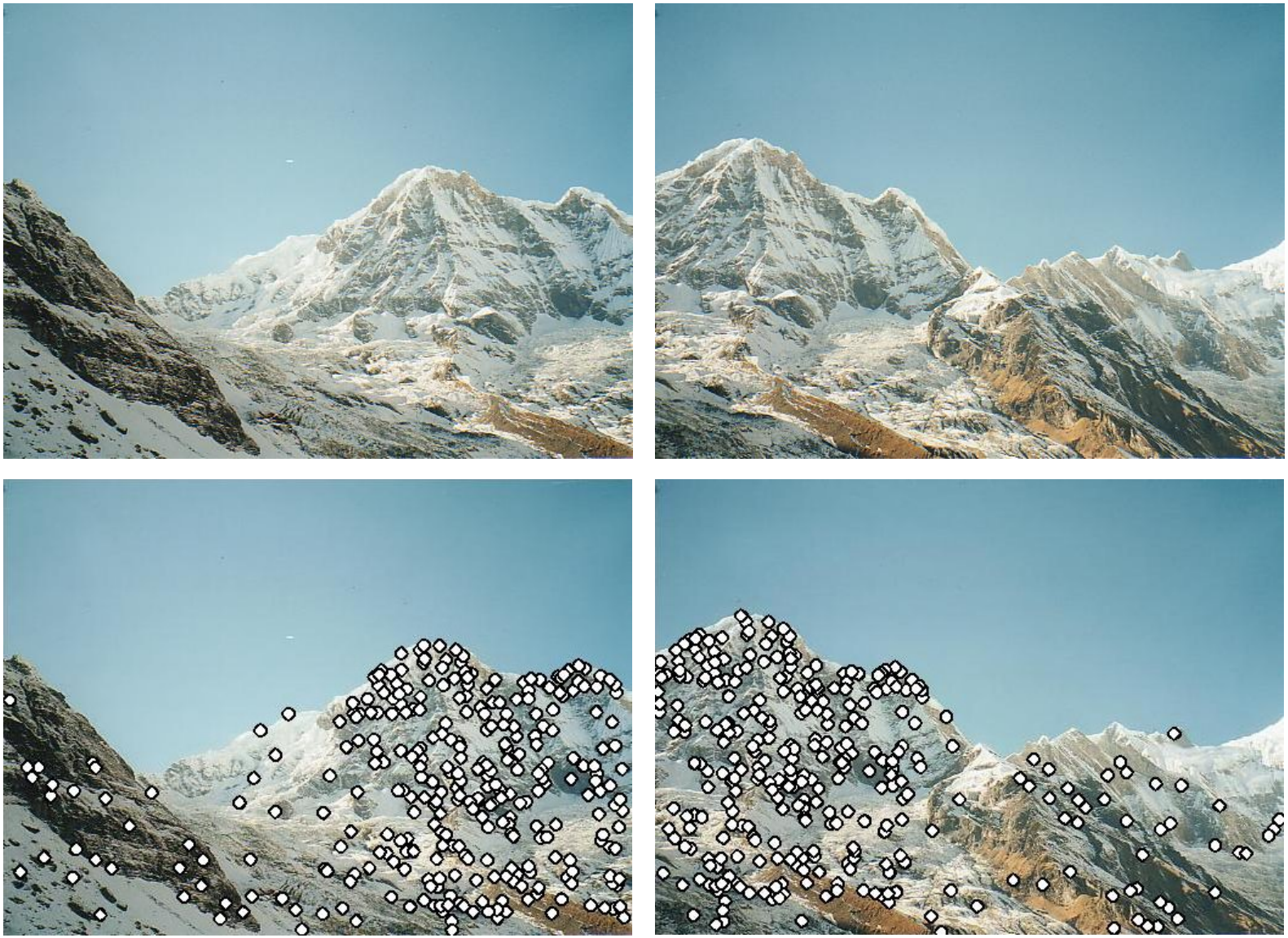
Bundle Adjustment

Multi-band Blending

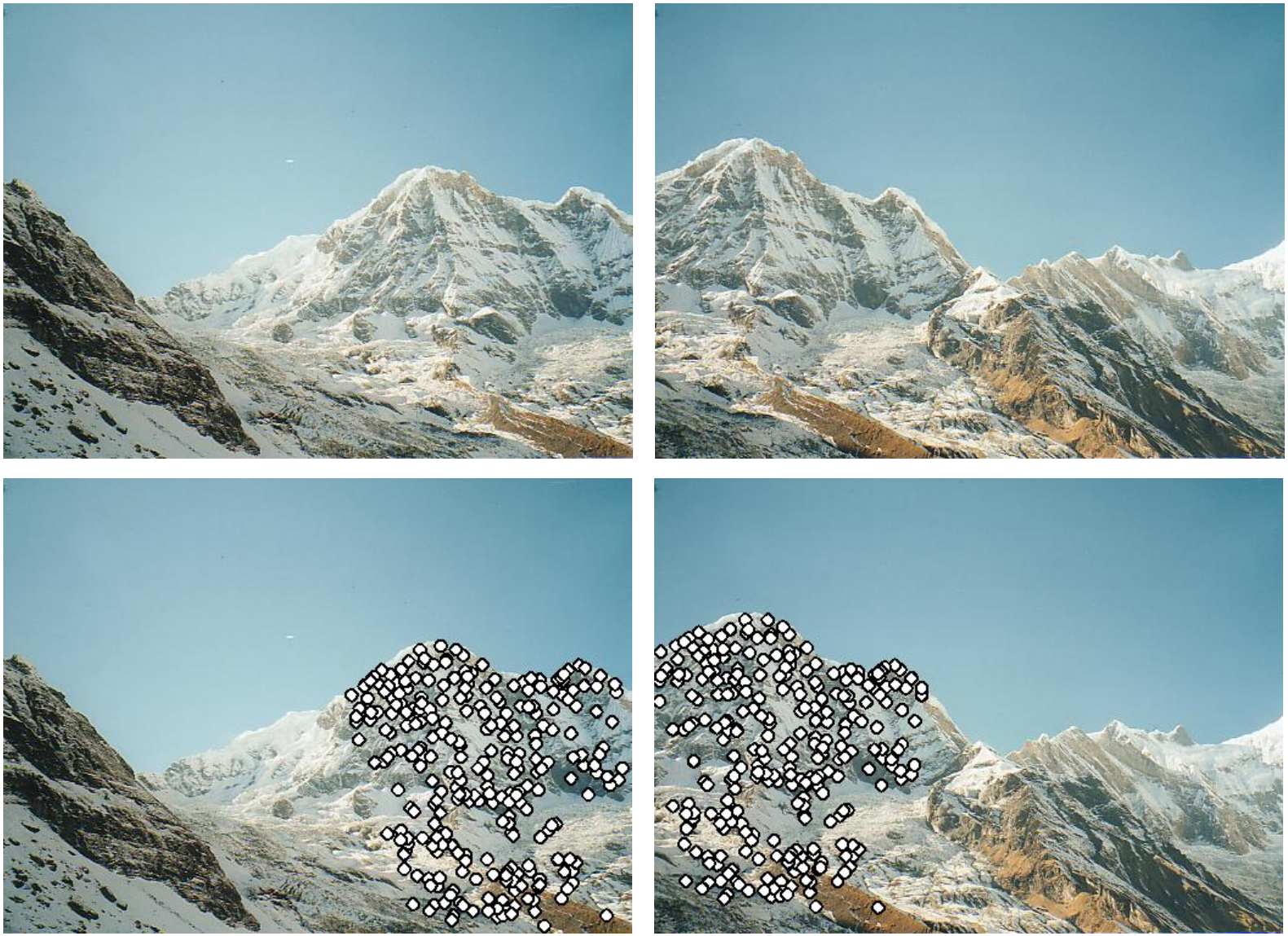
Results

Conclusions

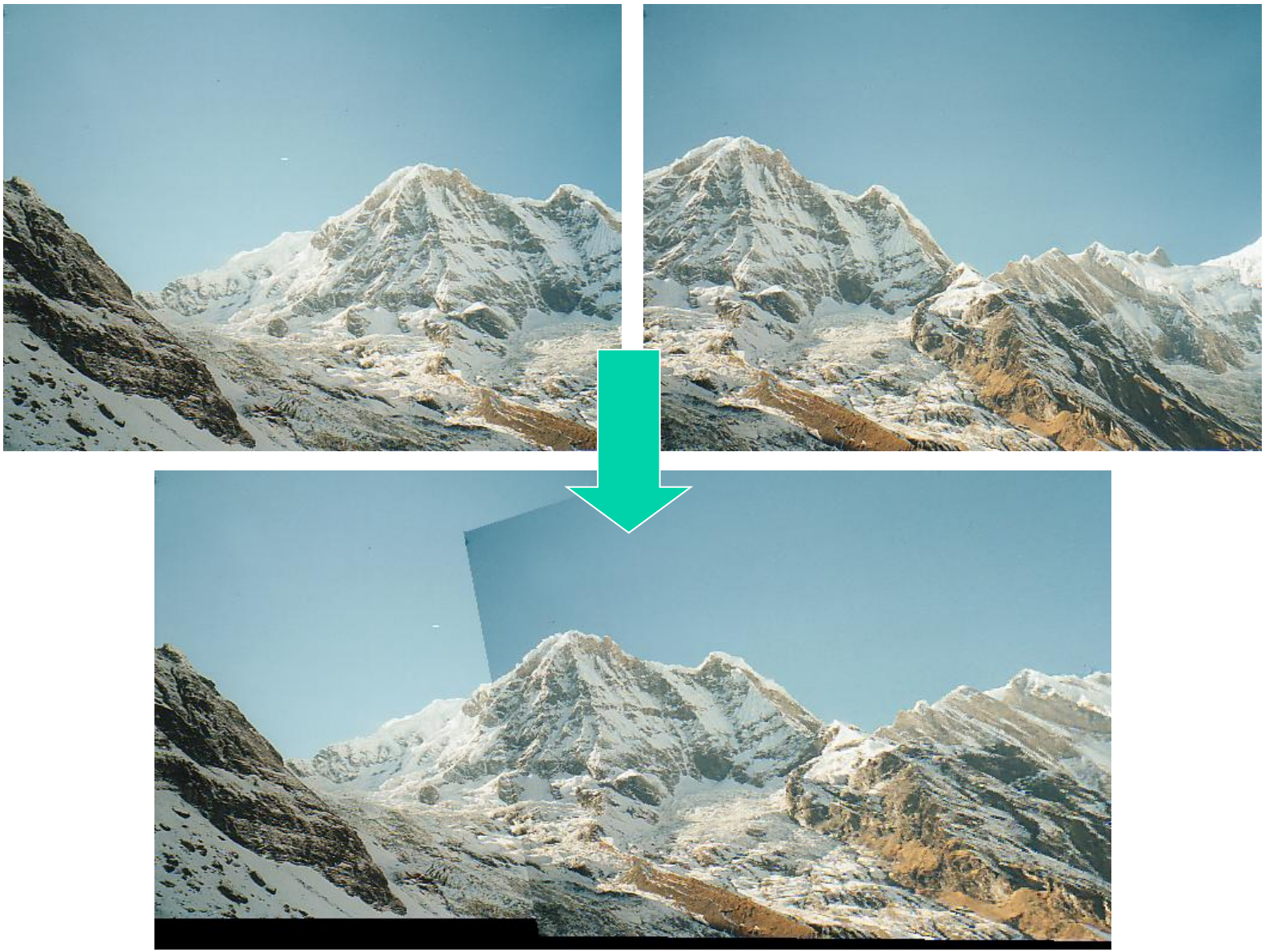
RANSAC for Homography



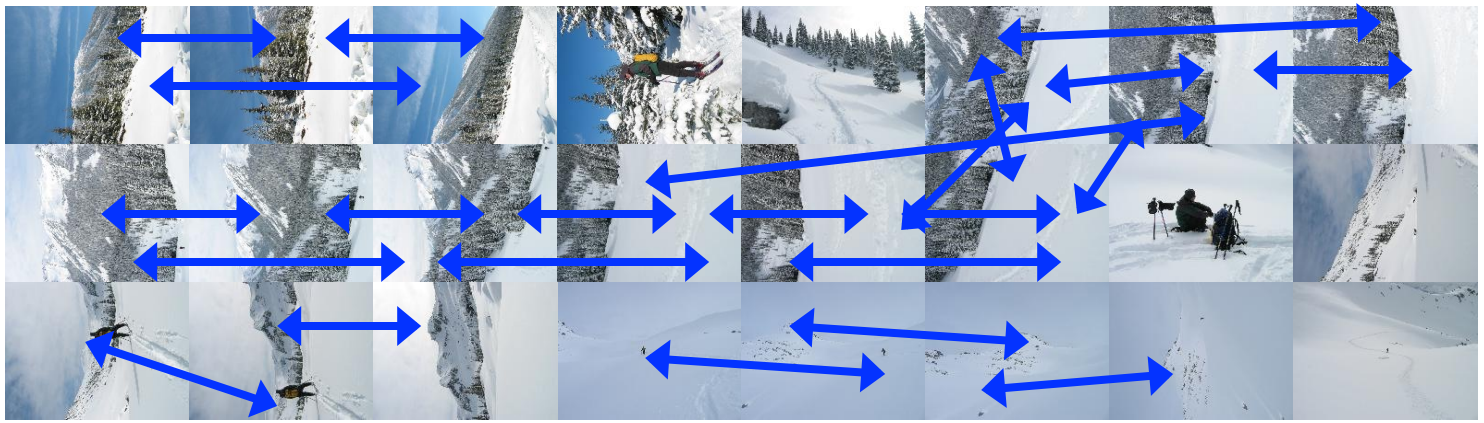
RANSAC for Homography



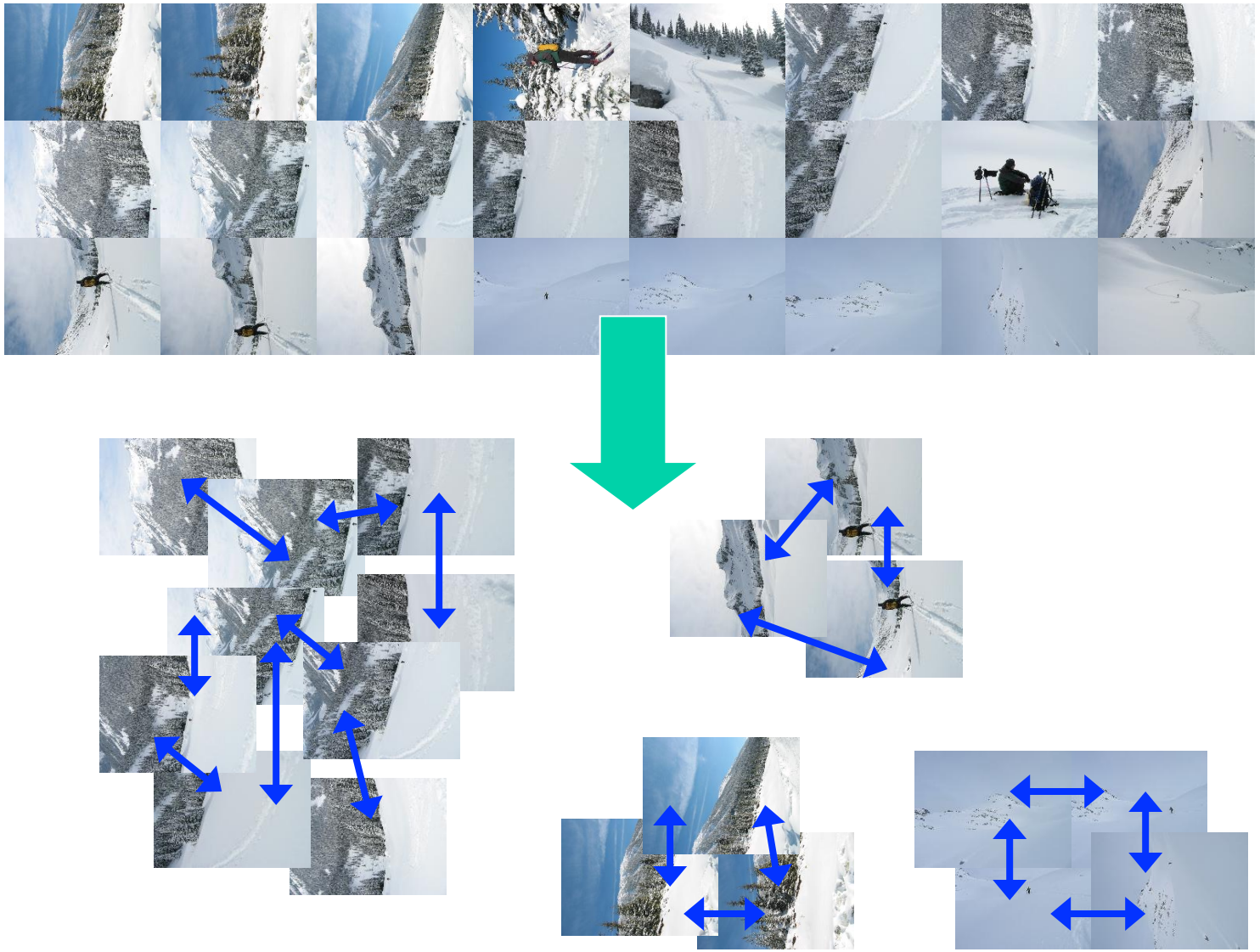
RANSAC for Homography



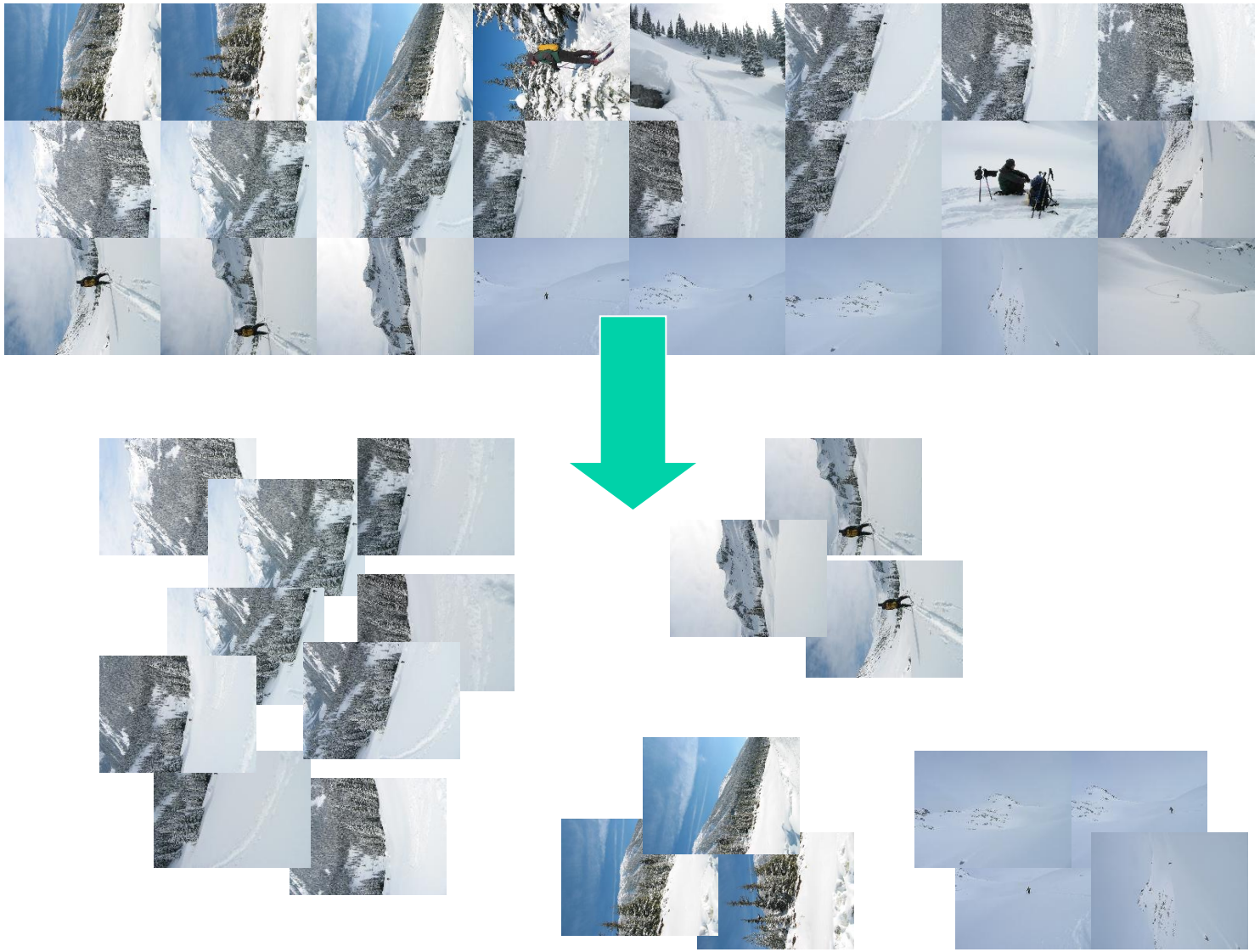
Finding the panoramas



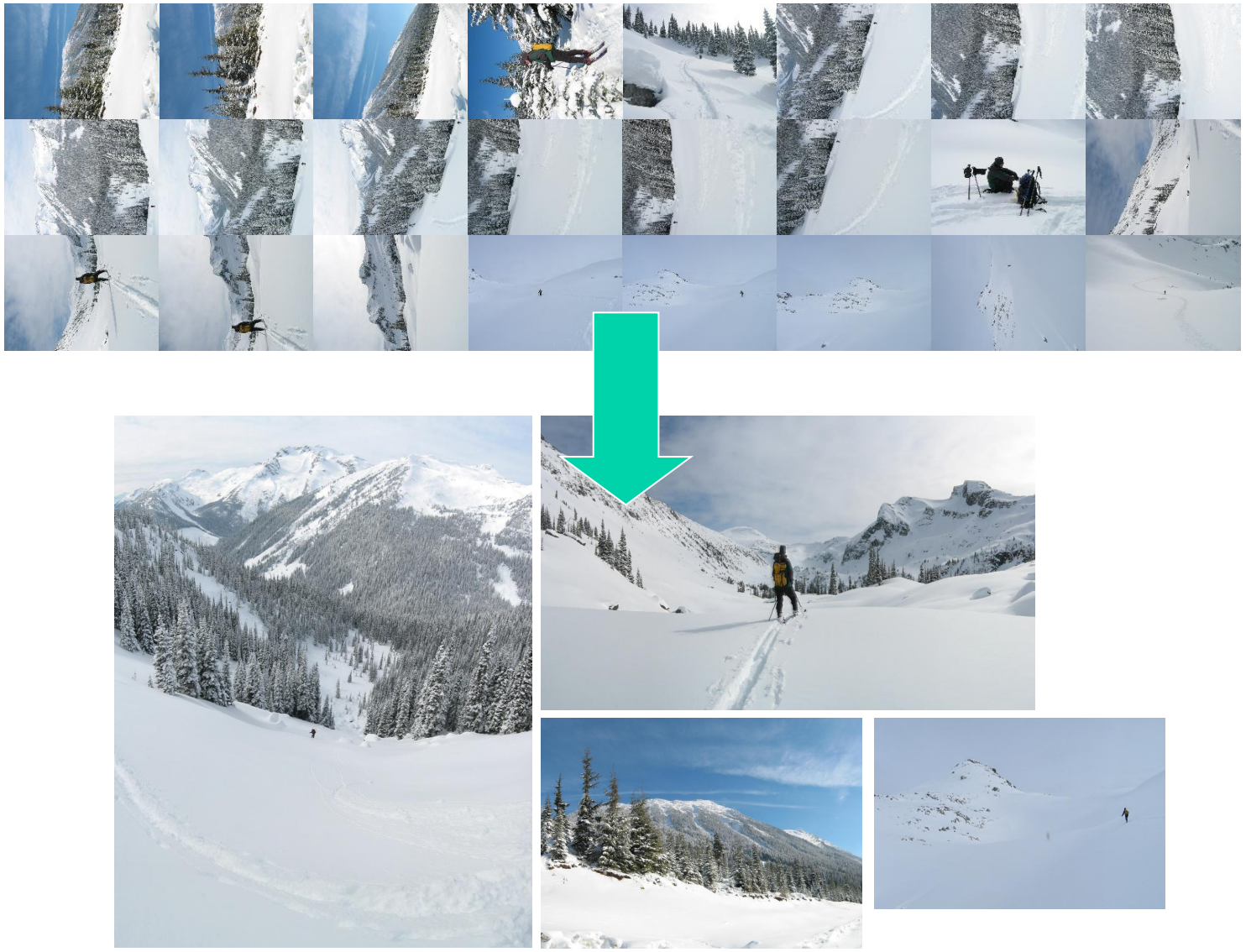
Finding the panoramas



Finding the panoramas



Finding the panoramas



Bundle Adjustment

New images initialised with rotation, focal length of best matching image

Multi-band Blending

Burt & Adelson 1983

- Blend frequency bands over range $\propto \lambda$

Results