Feature Matching and RANSAC

Recognising Panoramas.

[M. Brown and D. Lowe,ICCV 2003]

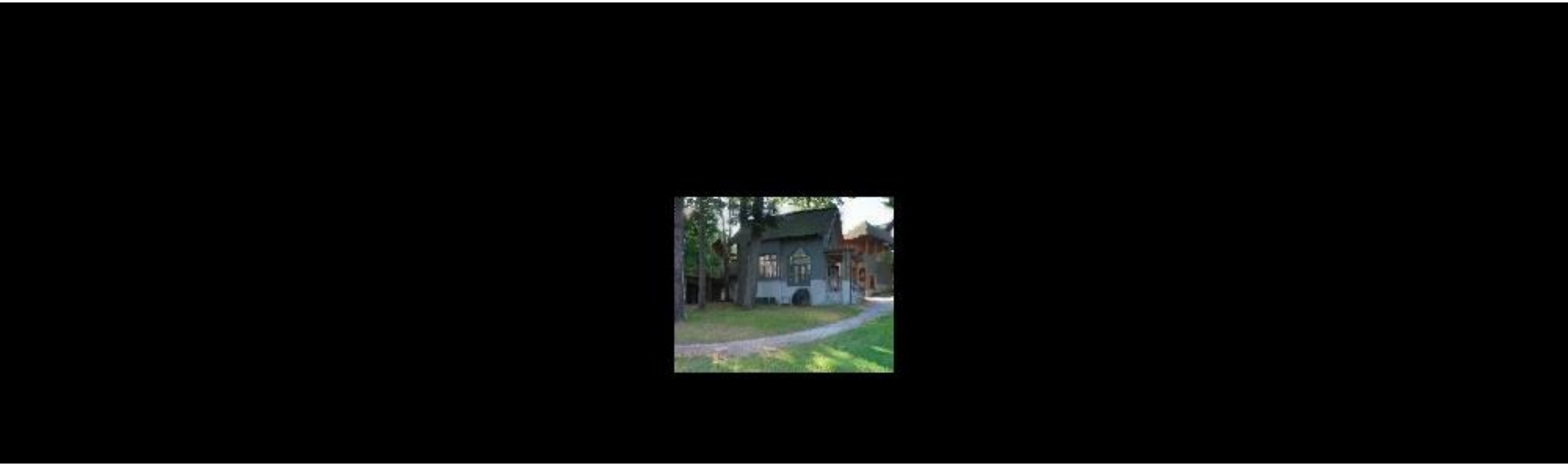
[Brown, Szeliski, Winder, CVPR' 2005]

with a lot of slides stolen from
Steve Seitz, Rick Szeliski, A. Efros

Introduction

Are you getting the whole picture?

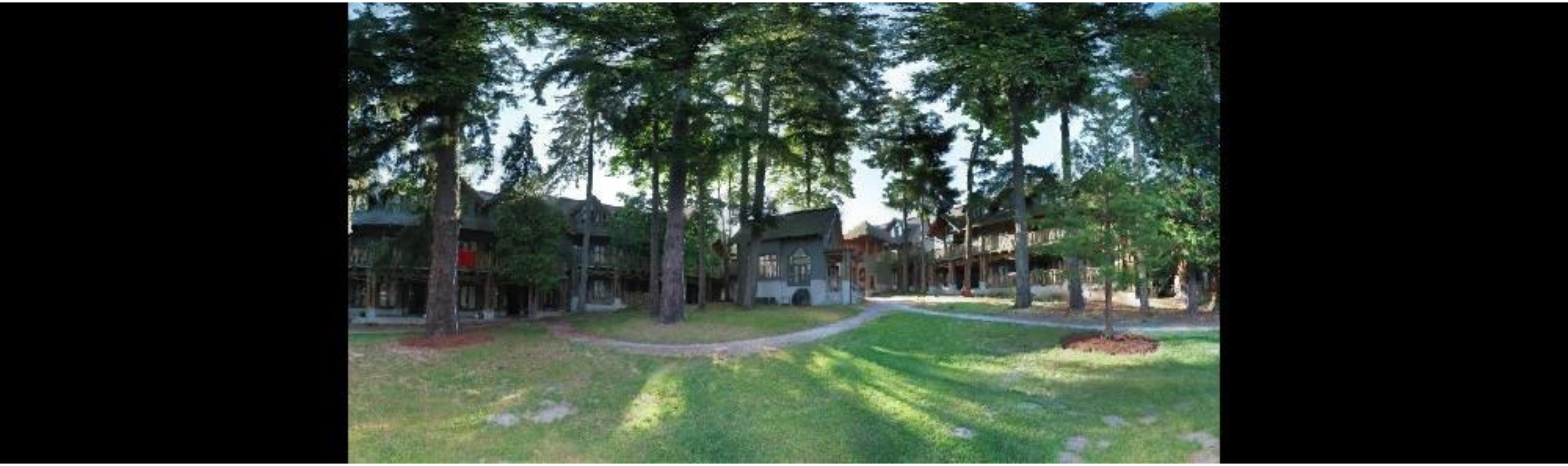
- Compact Camera FOV = 50 x 35°



Introduction

Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°



Introduction

Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°
- Panoramic Mosaic = 360 x 180°

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 $\nRightarrow$ matching images

Why "Recognising Panoramas"?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

- 2D Rotations (θ, ϕ)
  - Ordering $\nRightarrow$ matching images

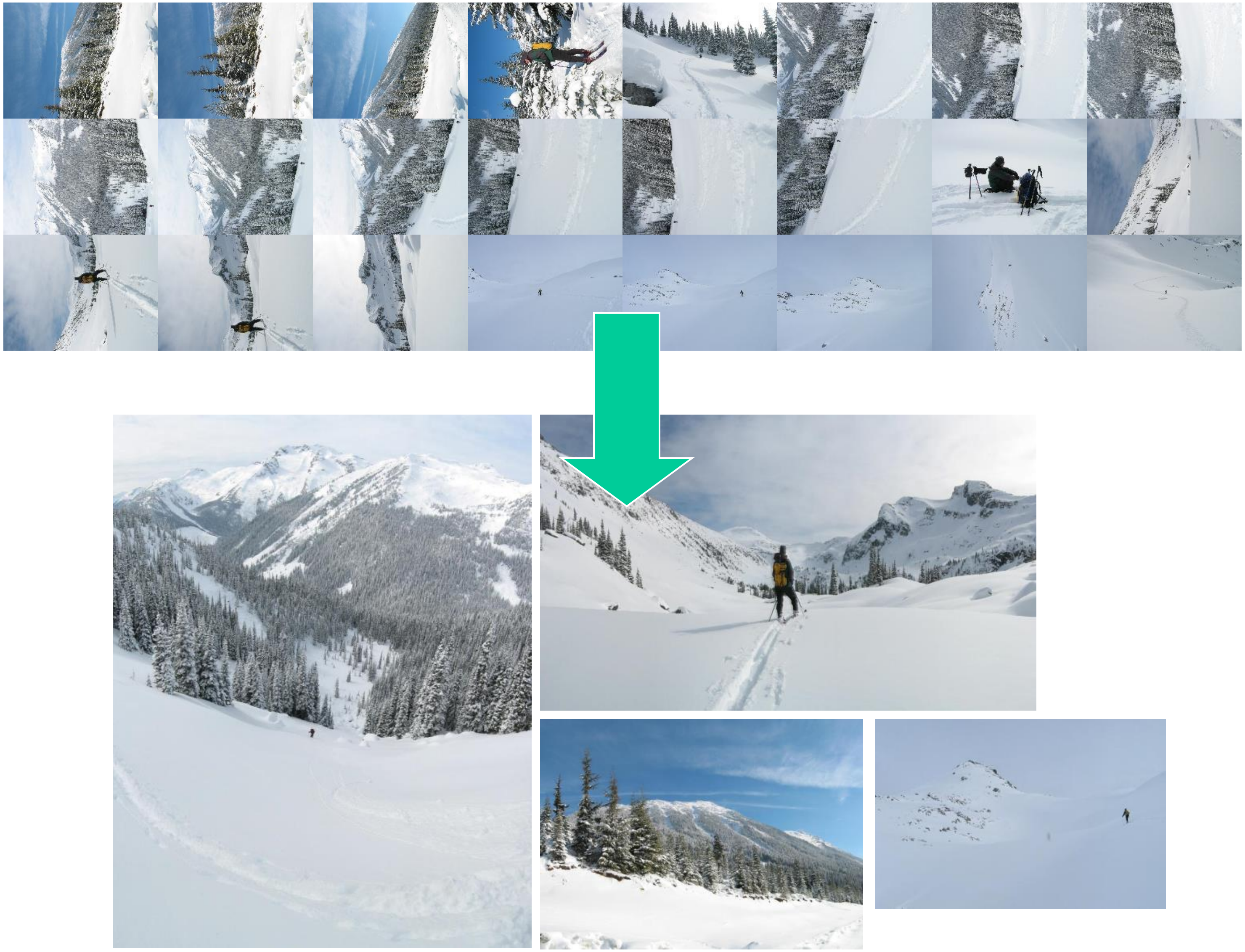
Why "Recognising Panoramas"?

1D Rotations (θ)

- Ordering $\Rightarrow$ matching images

- 2D Rotations (θ, ϕ)
  - Ordering $\nRightarrow$ matching images

Why “Recognising Panoramas”?



Overview

Feature Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

- Corner Features
- Nearest Neighbour Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

- Corner Features
- Nearest Neighbour Matching

Image Matching

Bundle Adjustment

Multi-band Blending

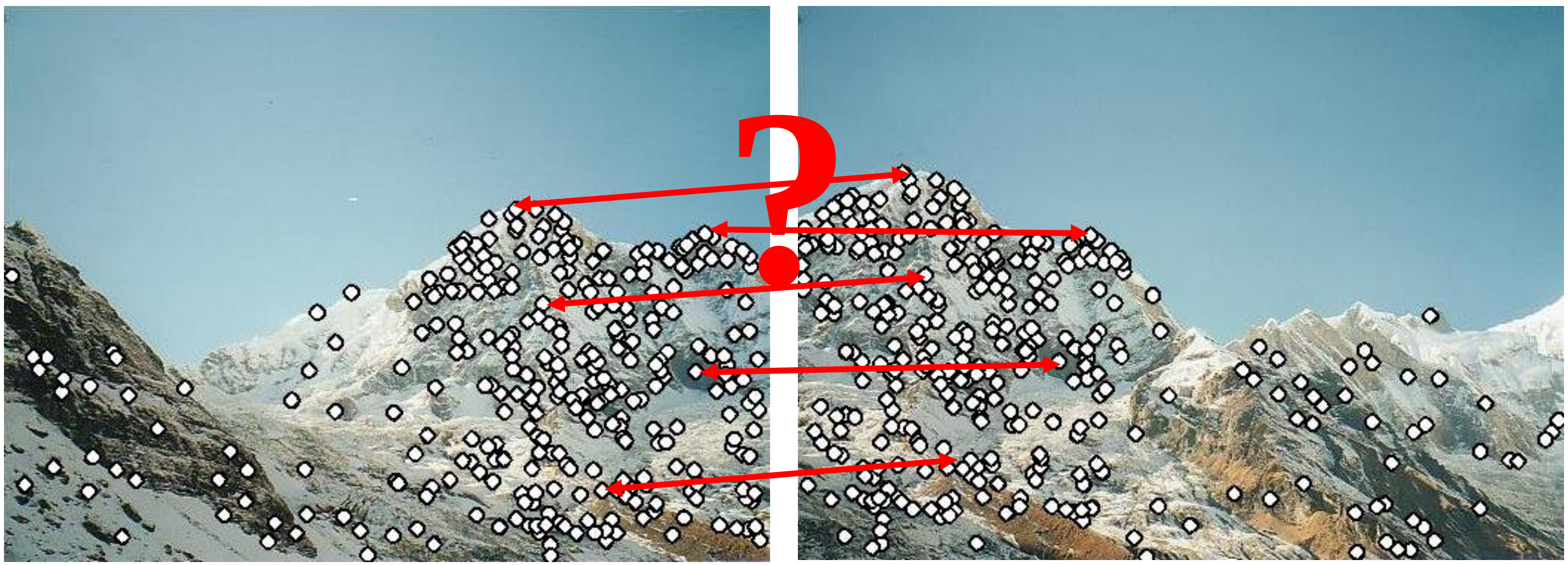
Results

Conclusions

Feature descriptors

We know how to detect points

Next question: **How to match them?**



Point descriptor should be:

1. Invariant
2. Distinctive

Descriptors Invariant to Rotation

Find local orientation

Dominant direction of gradient

- Extract image patches relative to this orientation

Multi-Scale Oriented Patches

Interest points

- Multi-scale Harris corners
- Orientation from blurred gradient
- Geometrically invariant to rotation

Descriptor vector

- Bias/gain normalized sampling of local patch (8x8)
- Photometrically invariant to affine changes in intensity

[Brown, Szeliski, Winder, CVPR’ 2005]

Descriptor Vector

Orientation = blurred gradient

Rotation Invariant Frame

- Scale-space position (x, y, s) + orientation (θ)

Detections at multiple scales

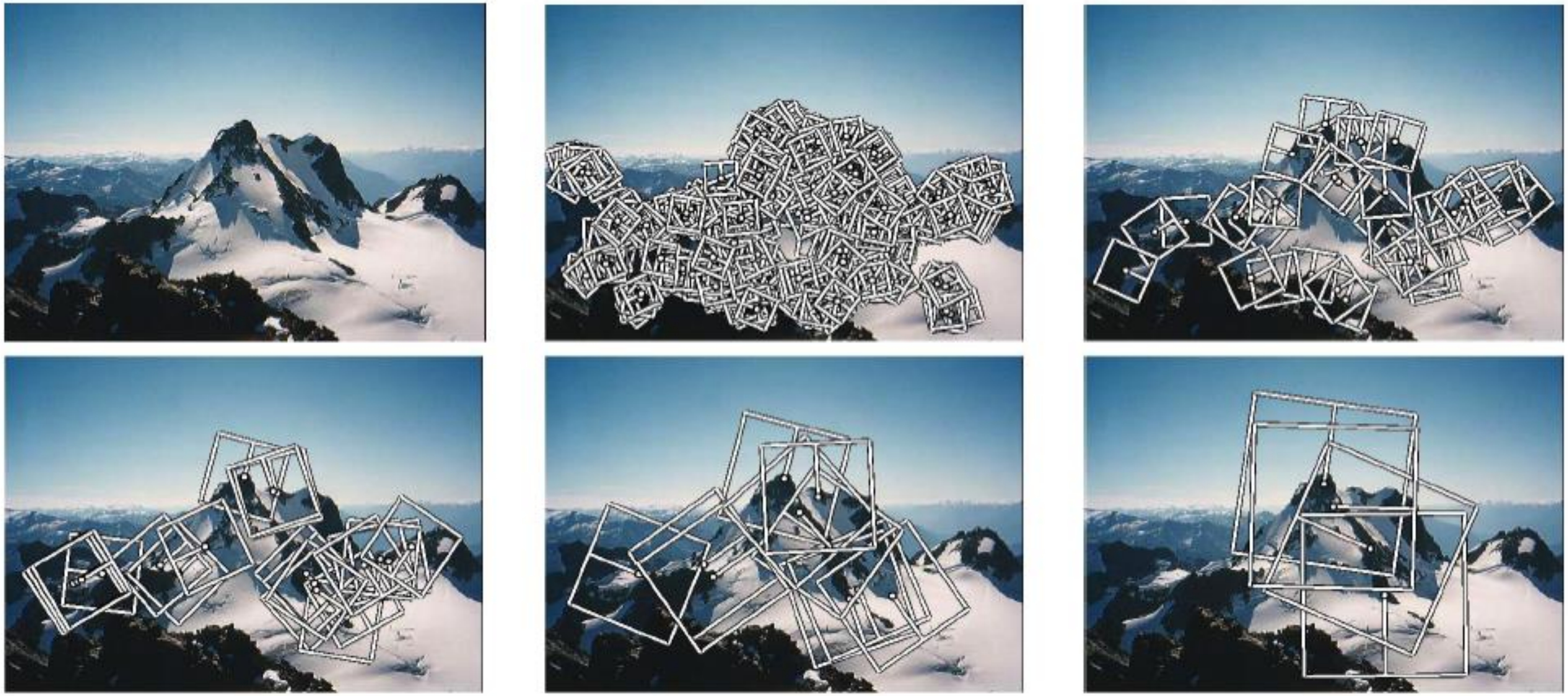


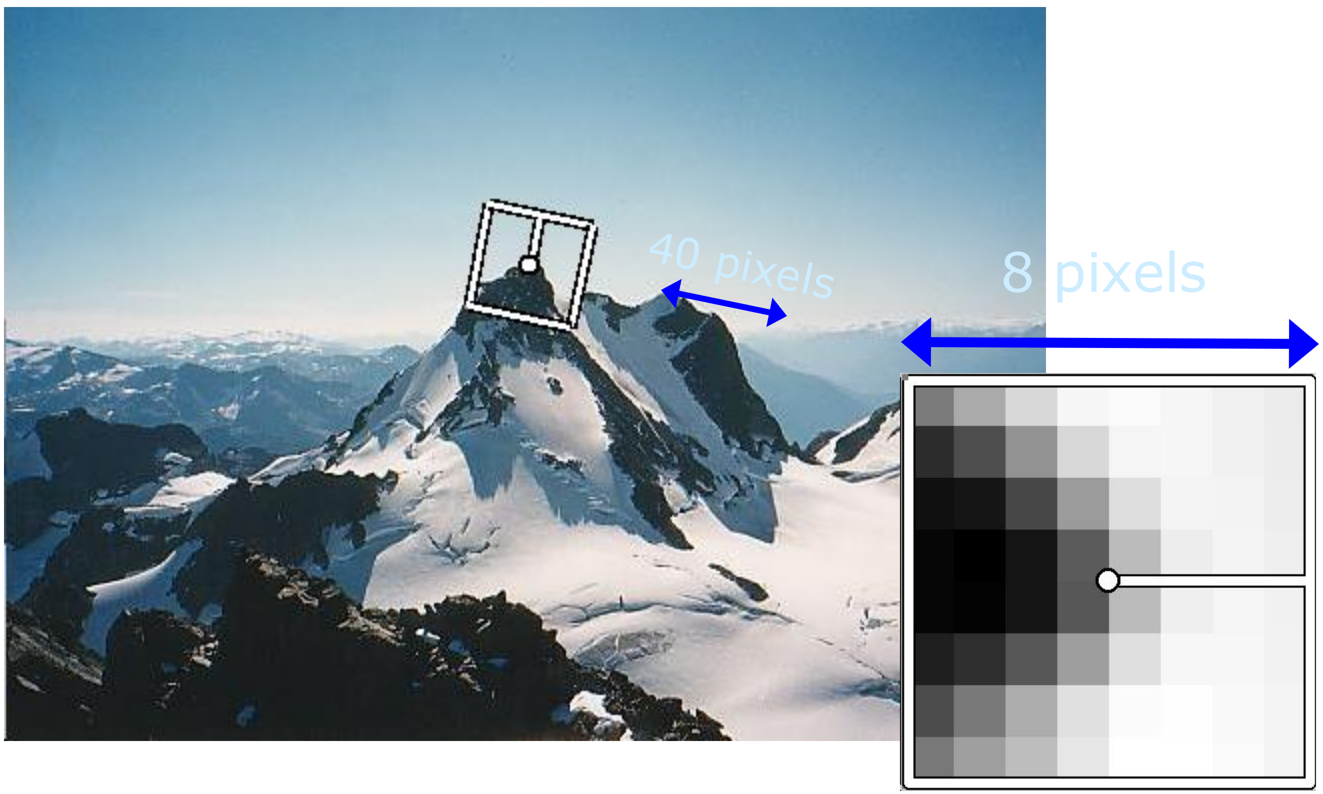
Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

MOPS descriptor vector

8x8 oriented patch

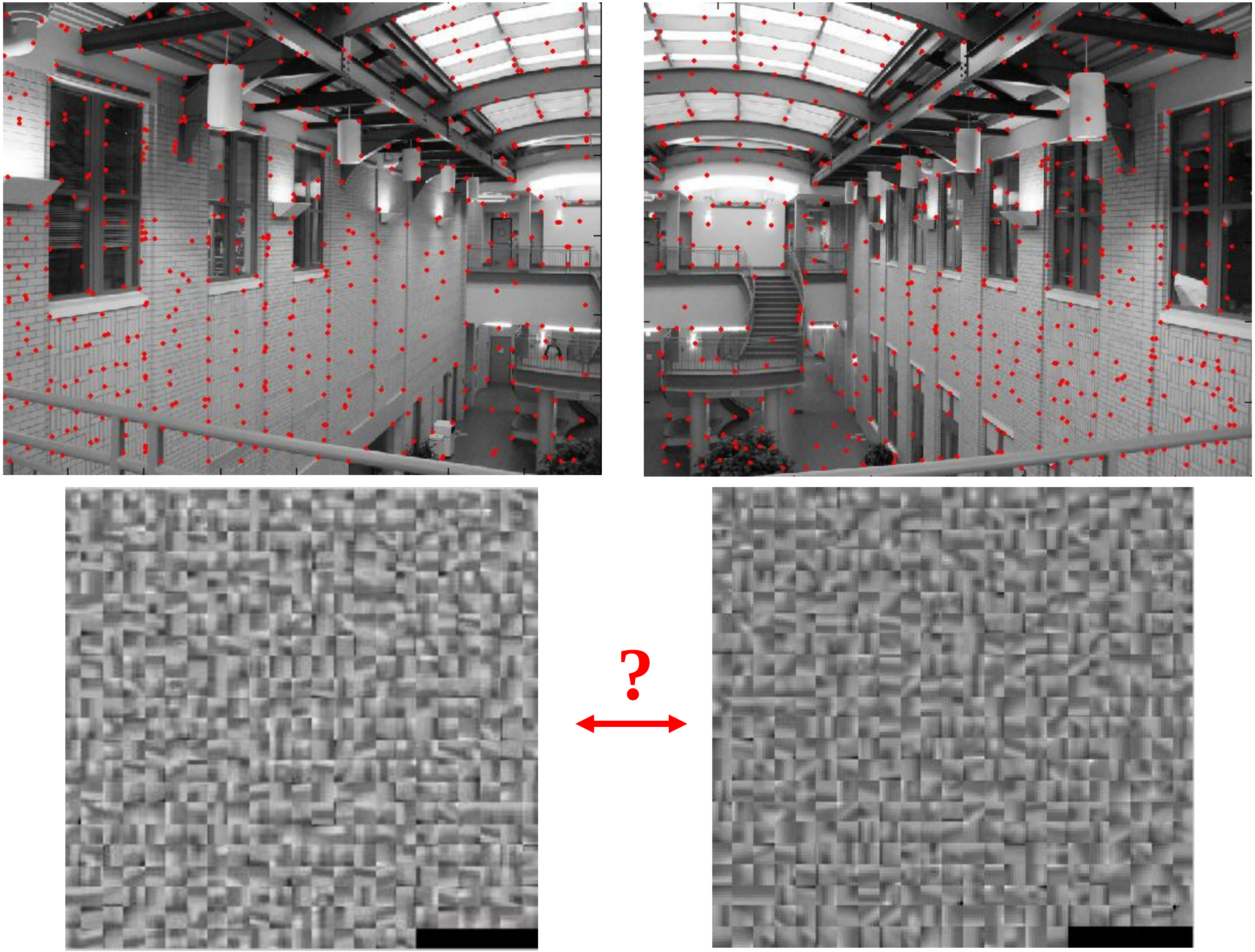
- Sampled at 5 x scale

Bias/gain normalisation: I’ = (I – μ)/σ



Invariant Features

Feature matching



Overview

Feature Matching

- corner Features
- Nearest Neighbour Matching

Image Matching

Bundle Adjustment

Multi-band Blending

Results

Conclusions

Nearest Neighbour Matching

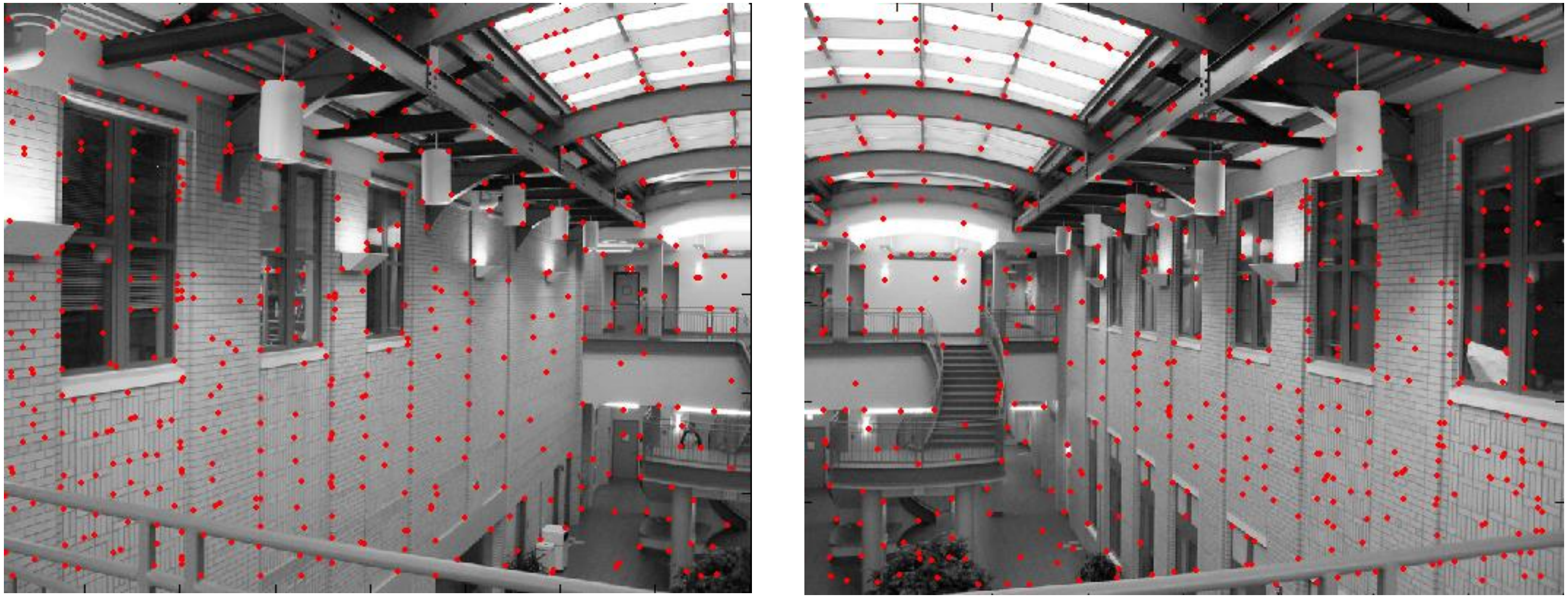
Find k-NN for each feature

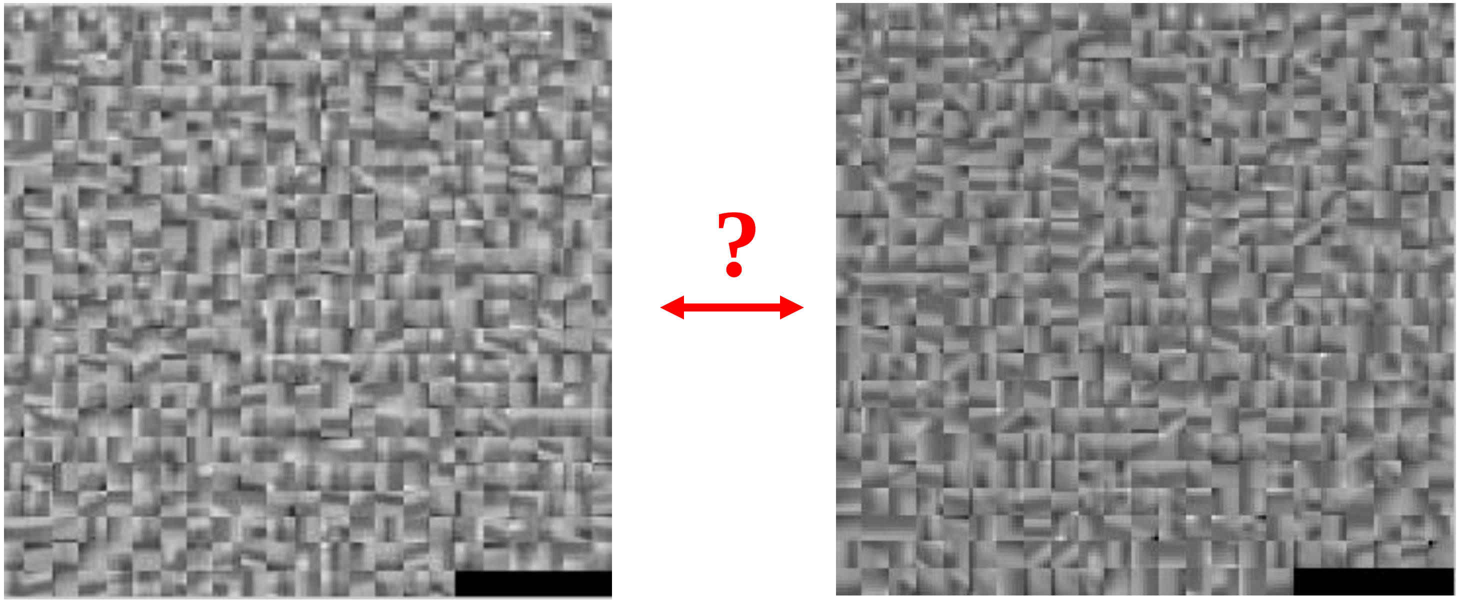
- k ≈ number of overlapping images (we use k = 4)

Use k-d tree

- k-d tree recursively bi-partitions data at mean in the dimension of maximum variance
- Approximate nearest neighbours found in O(nlogn)

What about outliers?



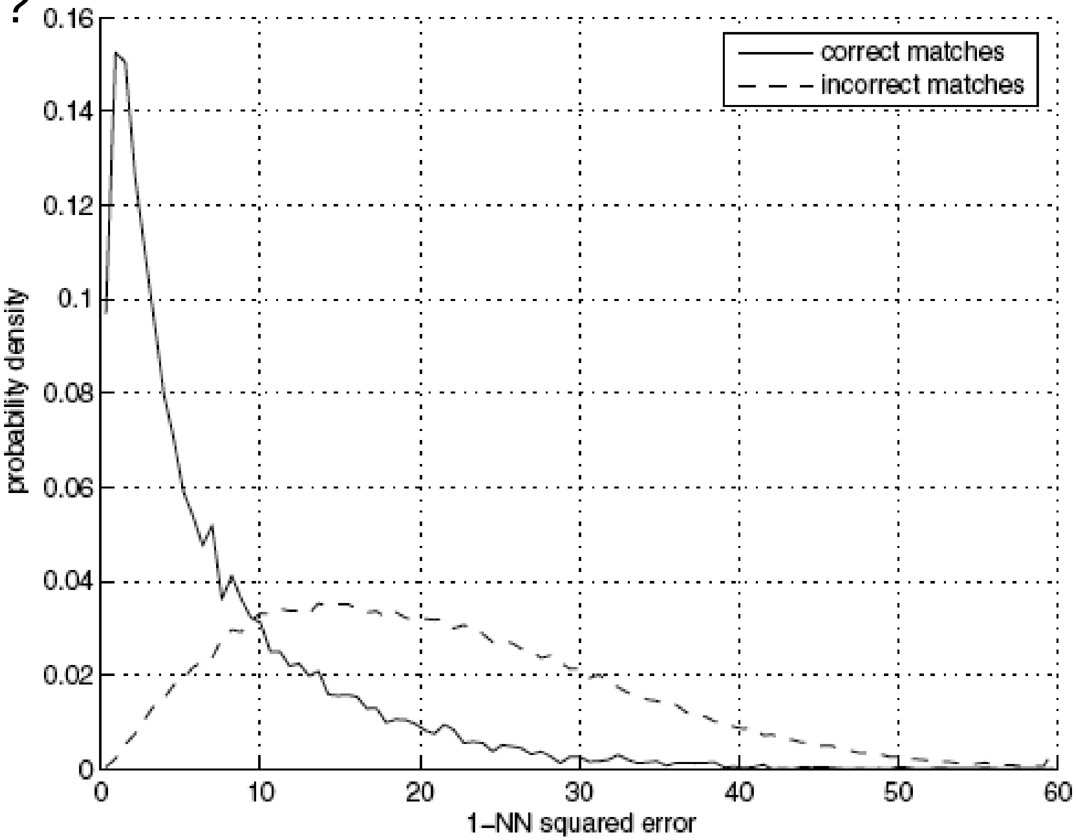


Feature-space outlier rejection

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

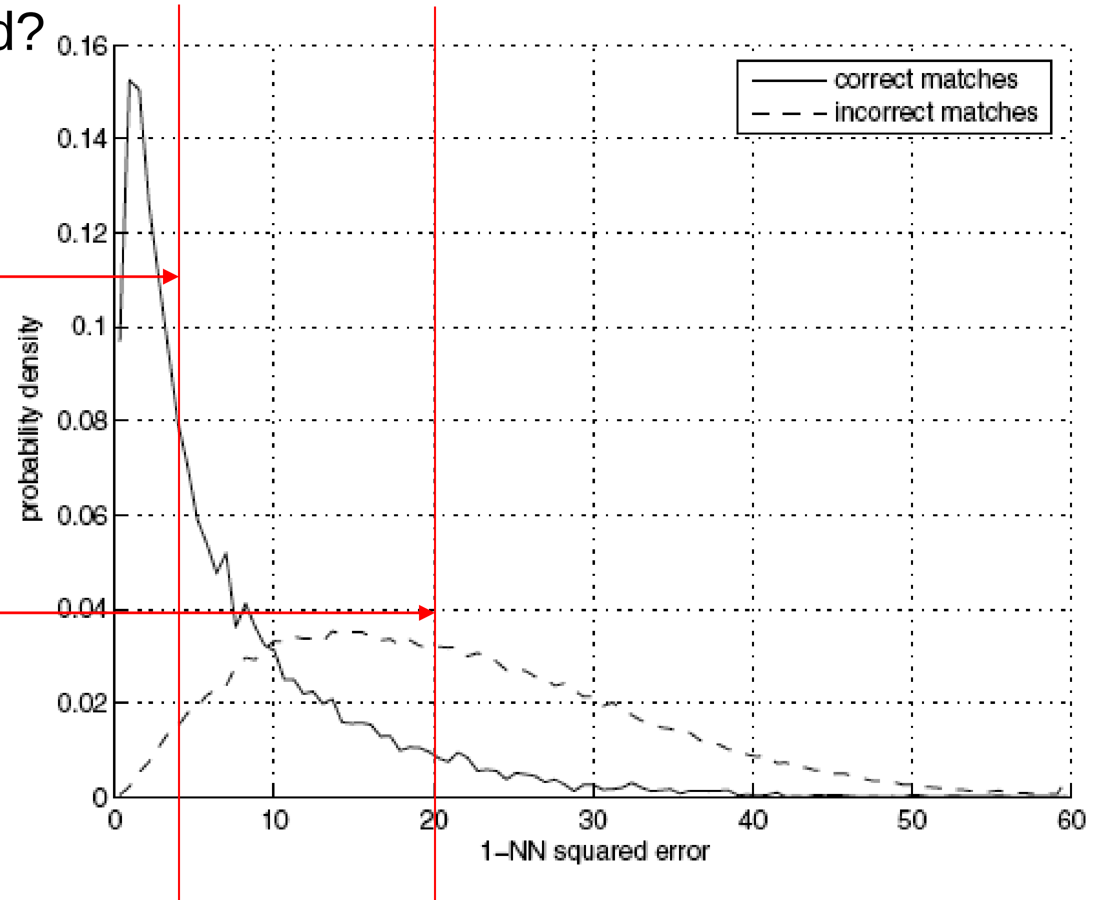
Let's not match all features, but only these that have "similar enough" matches?

How can we do it?

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

Too low, miss many good matches

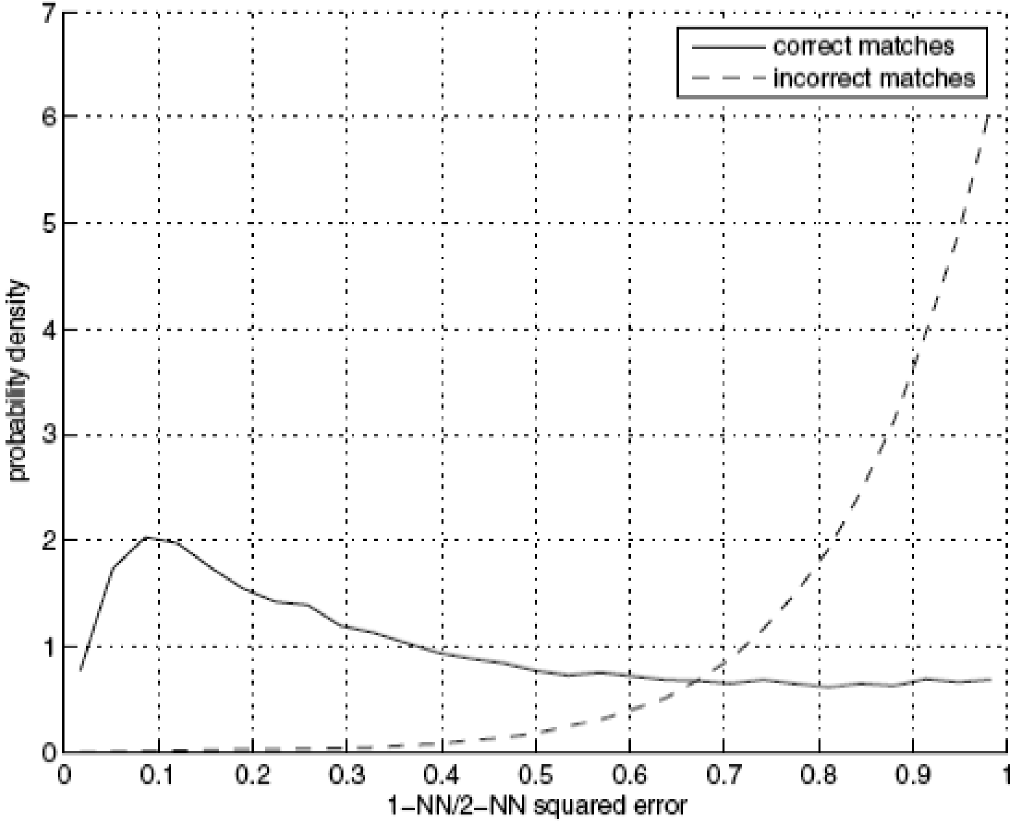
Too high, too many false matches



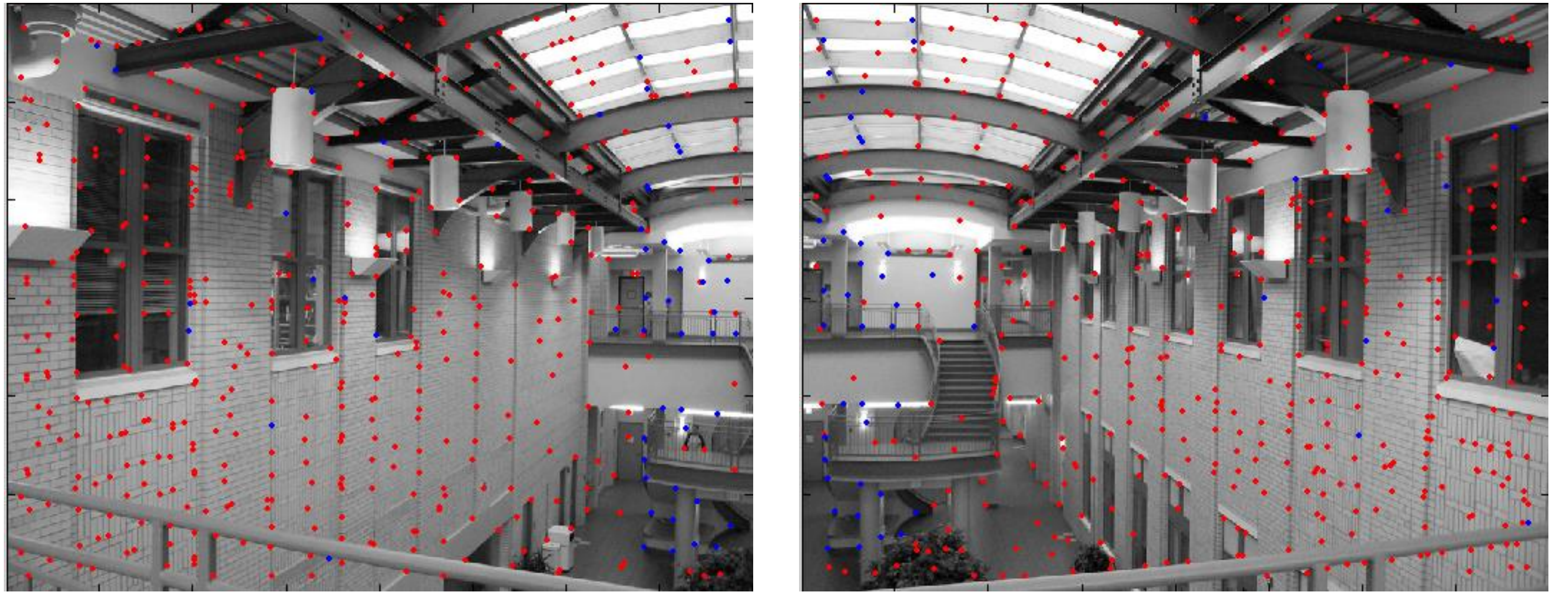
Feature-space outlier rejection

A better way [Lowe, 1999]:

- 1-NN: SSD of the closest match
- 2-NN: SSD of the <u>second-closest</u> 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?

Overview

Feature Matching

- Corner Features
- Nearest Neighbour Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

Image Matching

Multi-band Blending

Results

Overview

Feature Matching

Image Matching

- RANSAC for Homography
- Probabilistic model for verification

Multi-band Blending

Results

Overview

Feature Matching

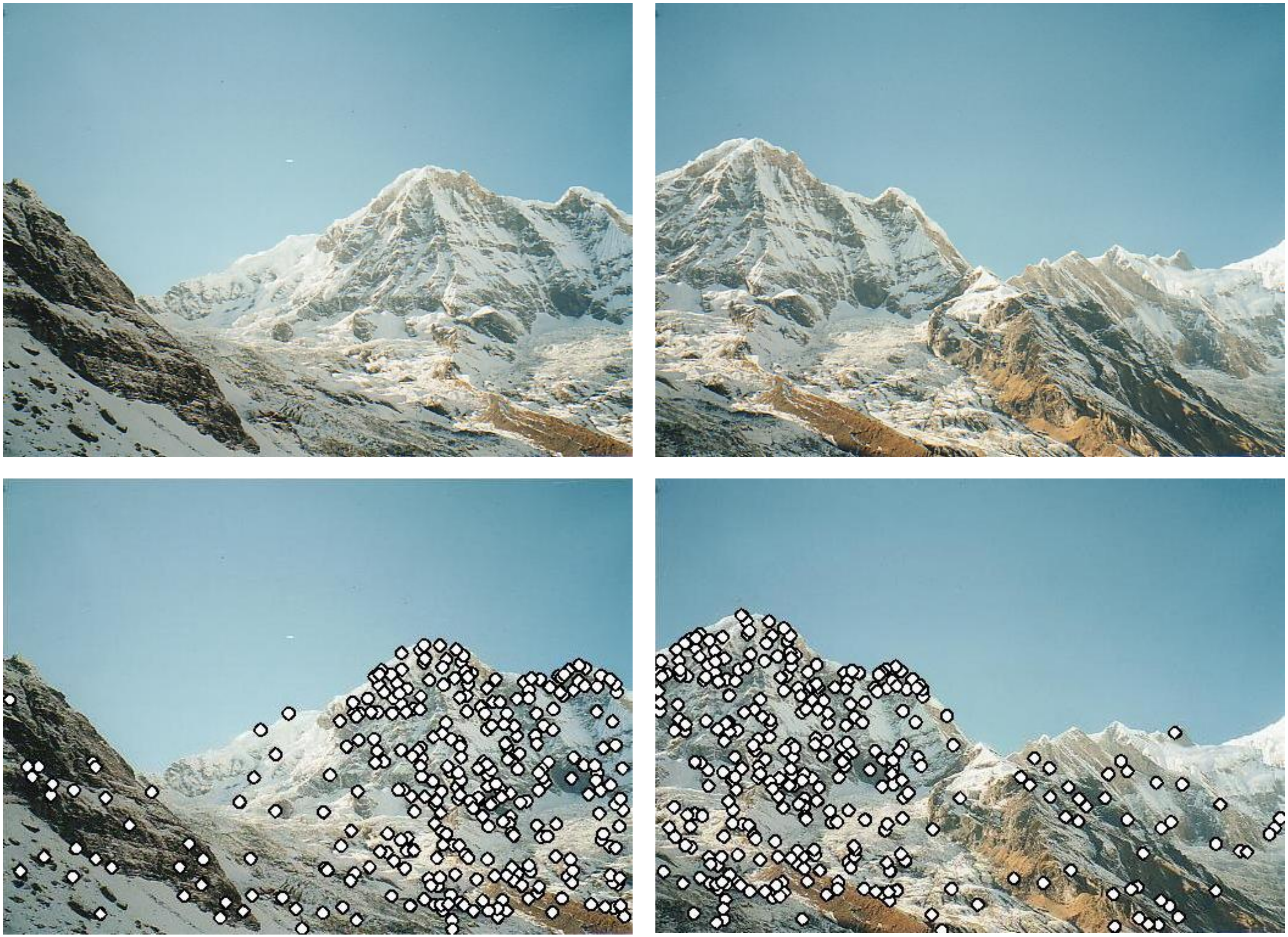
Image Matching

- RANSAC for Homography

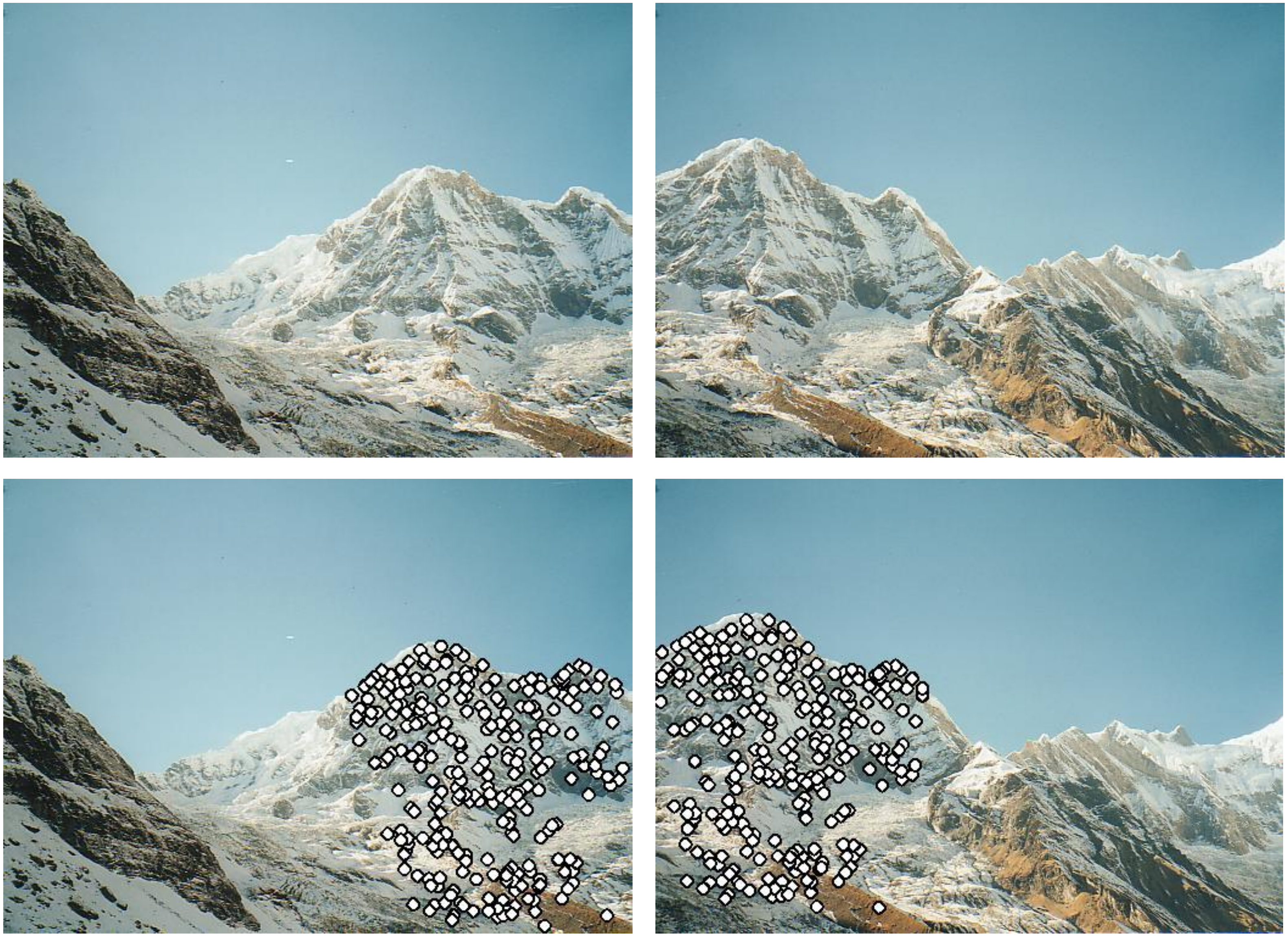
Multi-band Blending

Results

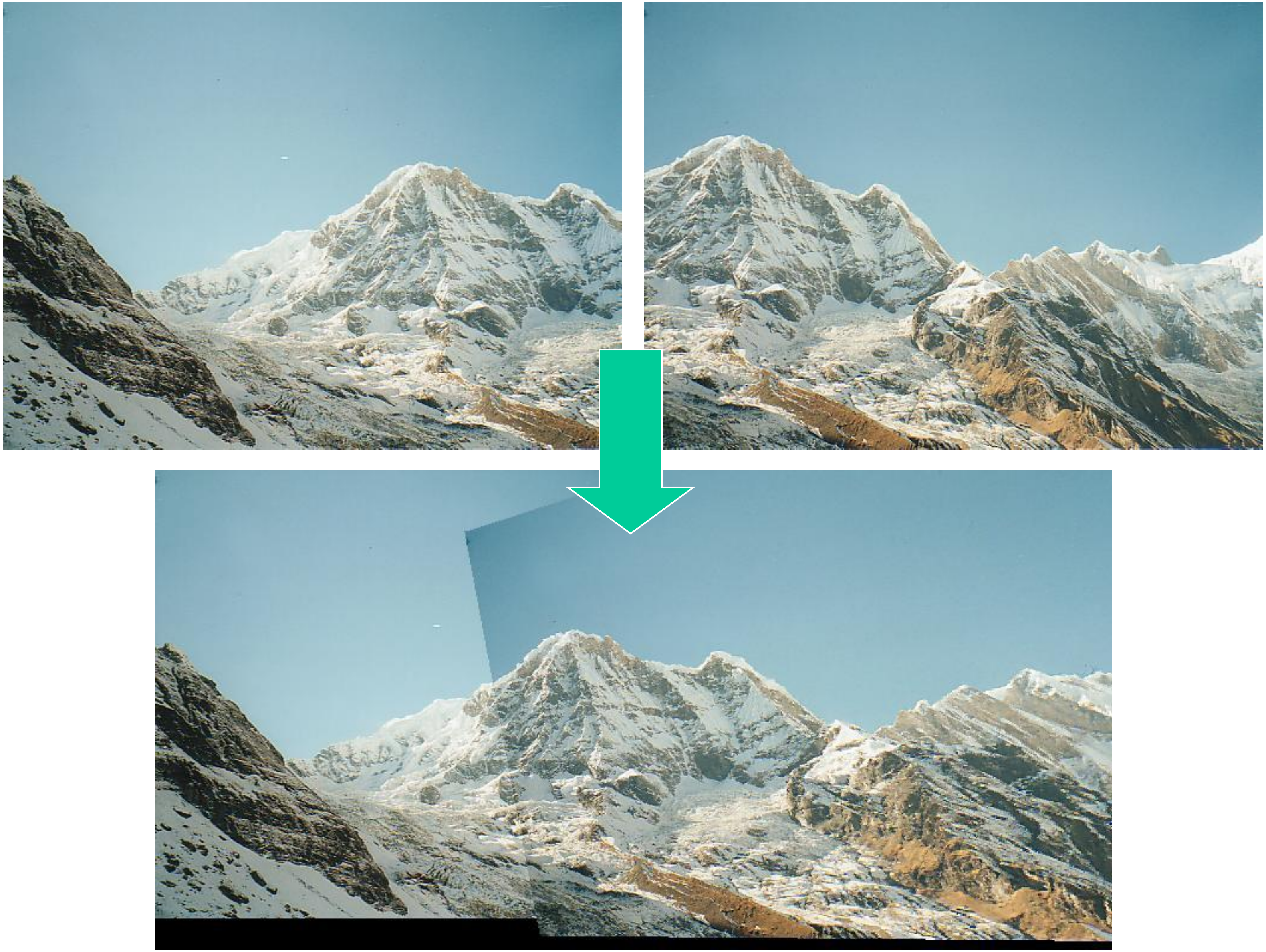
RANSAC for Homography



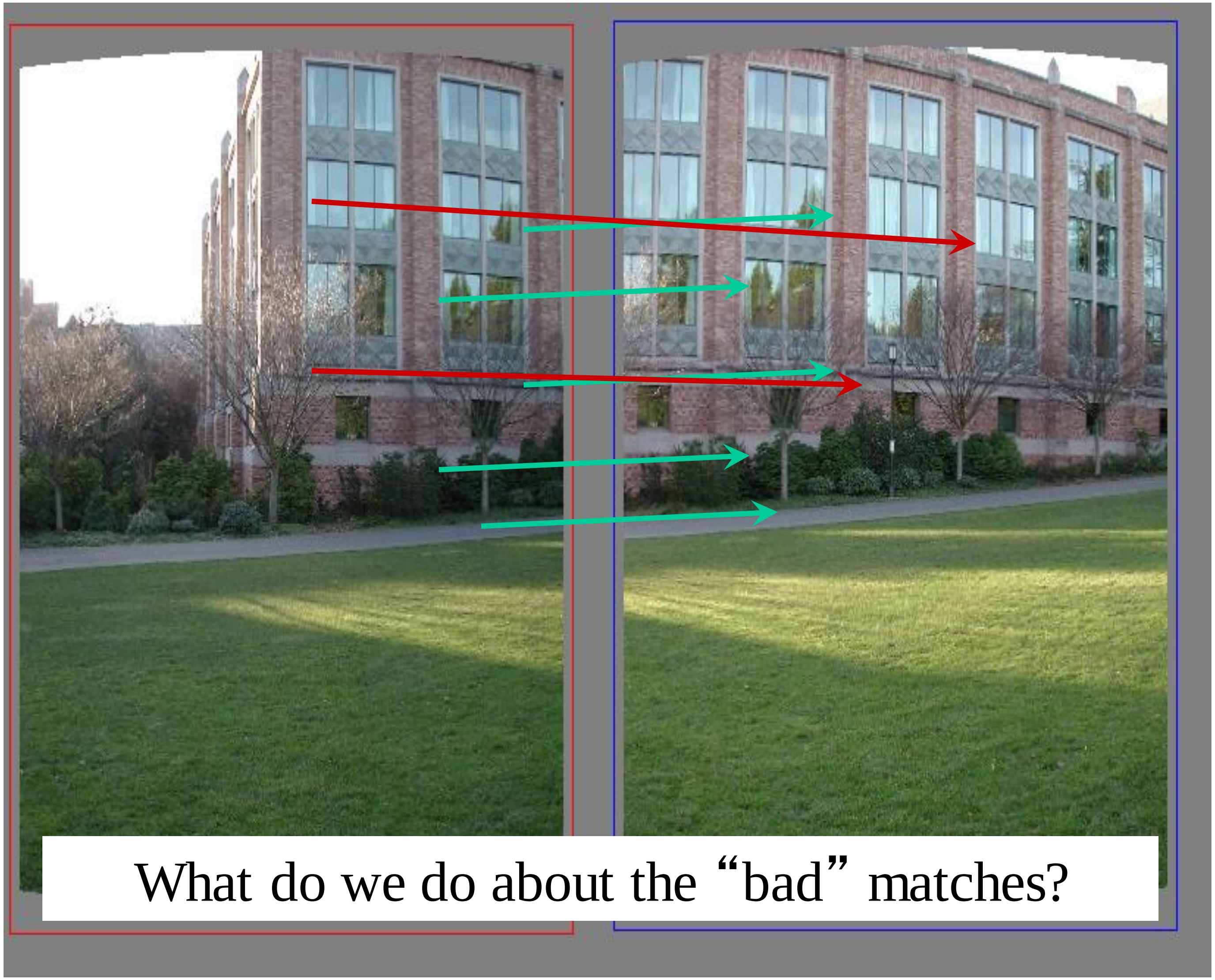
RANSAC for Homography



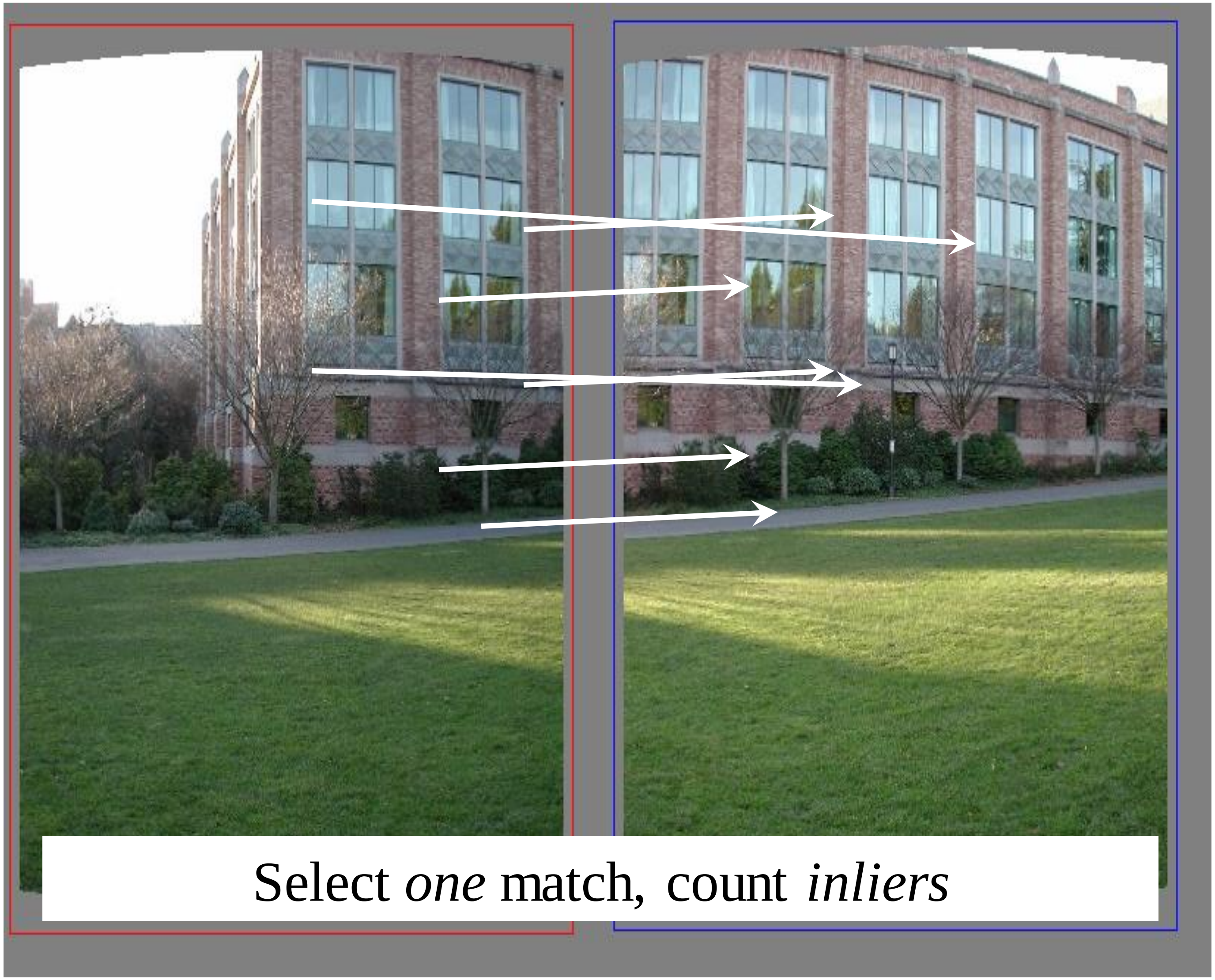
RANSAC for Homography



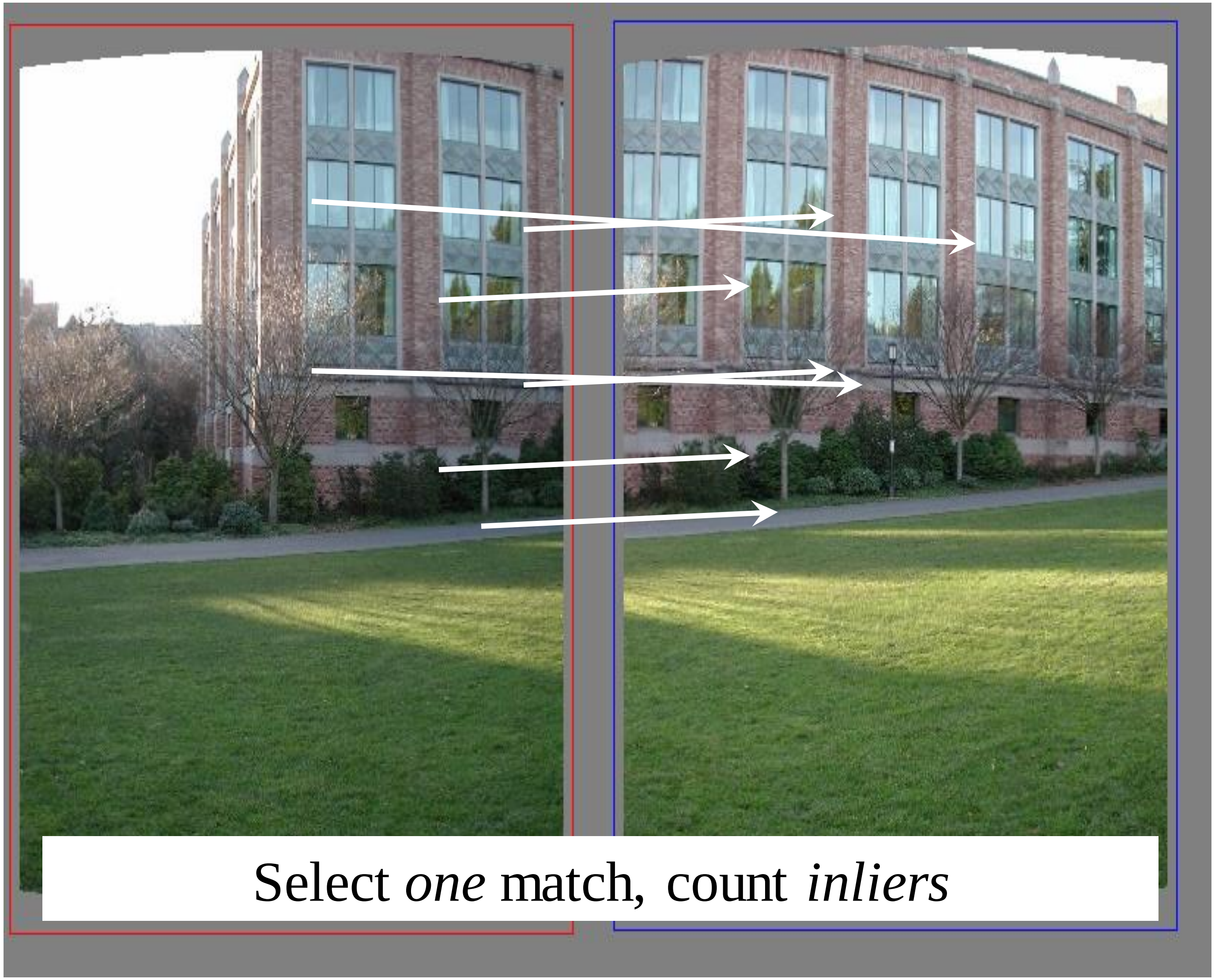
Matching features



RAndom SAmple Consensus

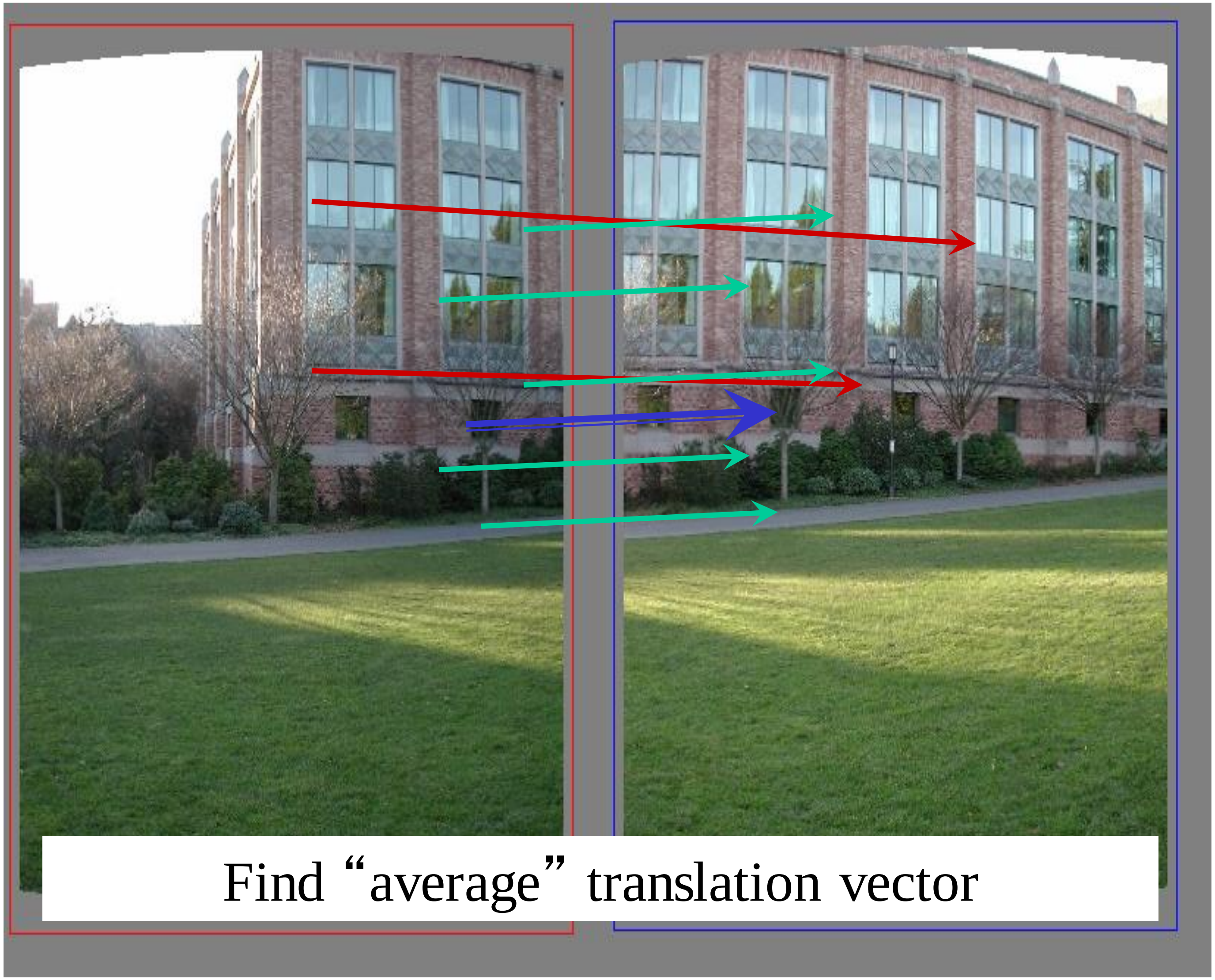


RAndom SAmple Consensus



Select *one* match, count *inliers*

Least squares fit

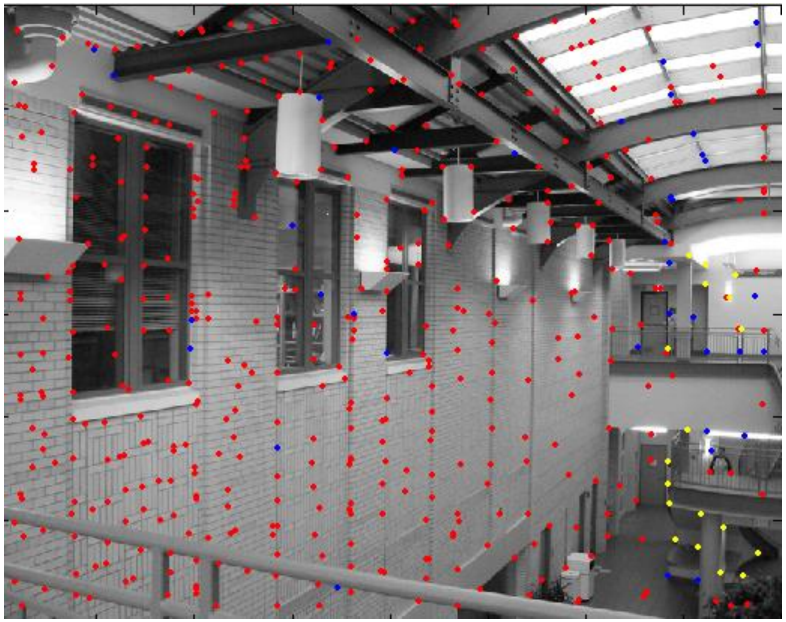


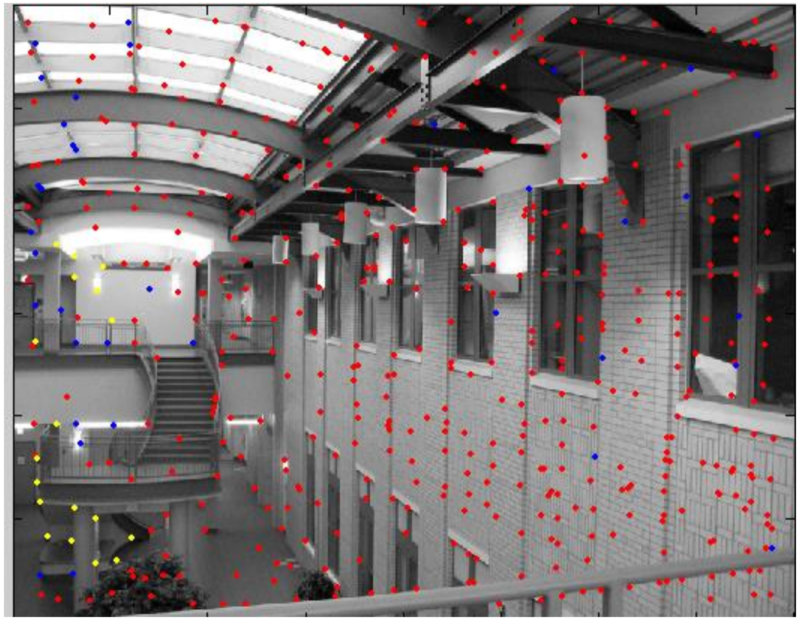
RANSAC for estimating homography

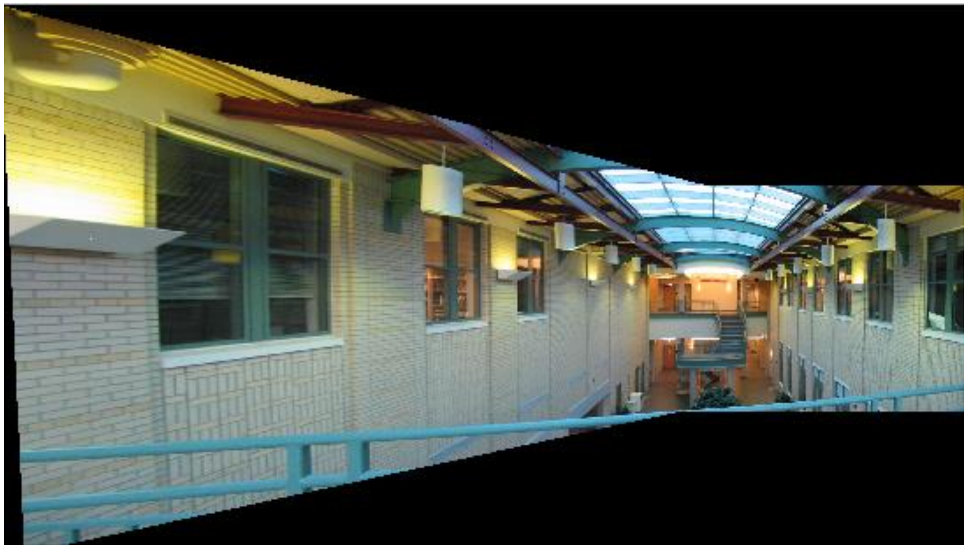
RANSAC loop:

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

RANSAC







RANSAC for estimating homography

RANSAC loop:

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

Image warping with homographies

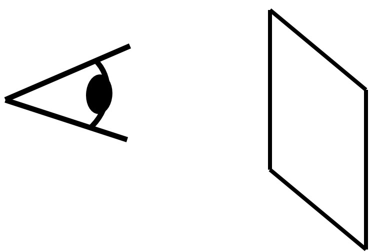
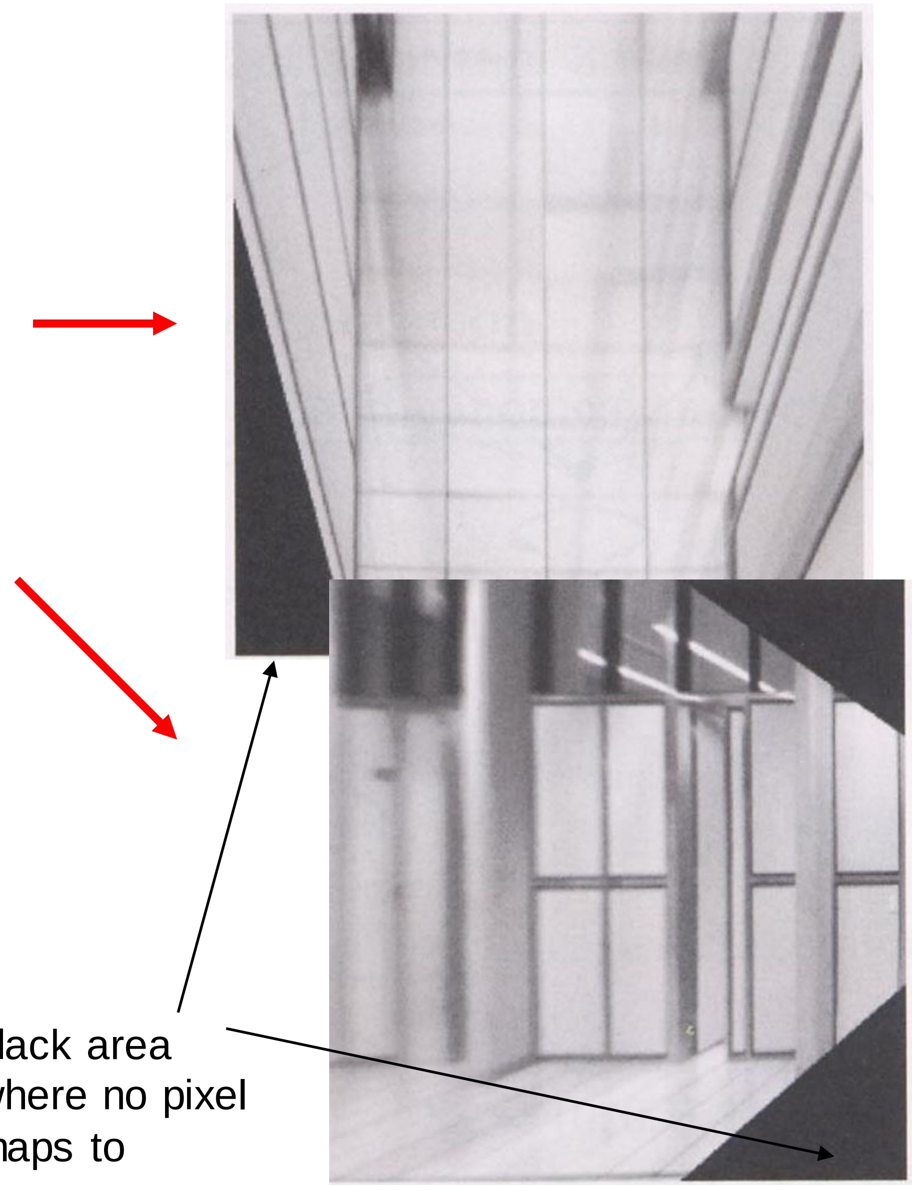
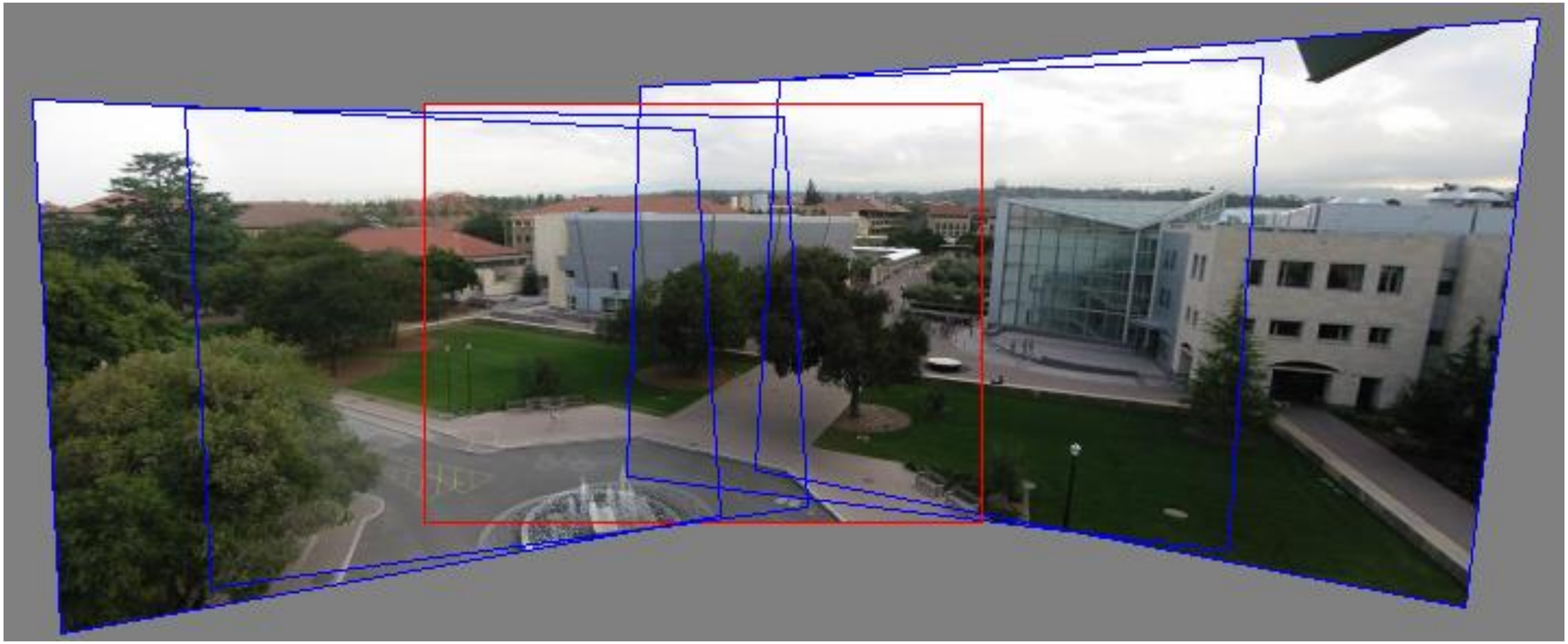


image plane in front



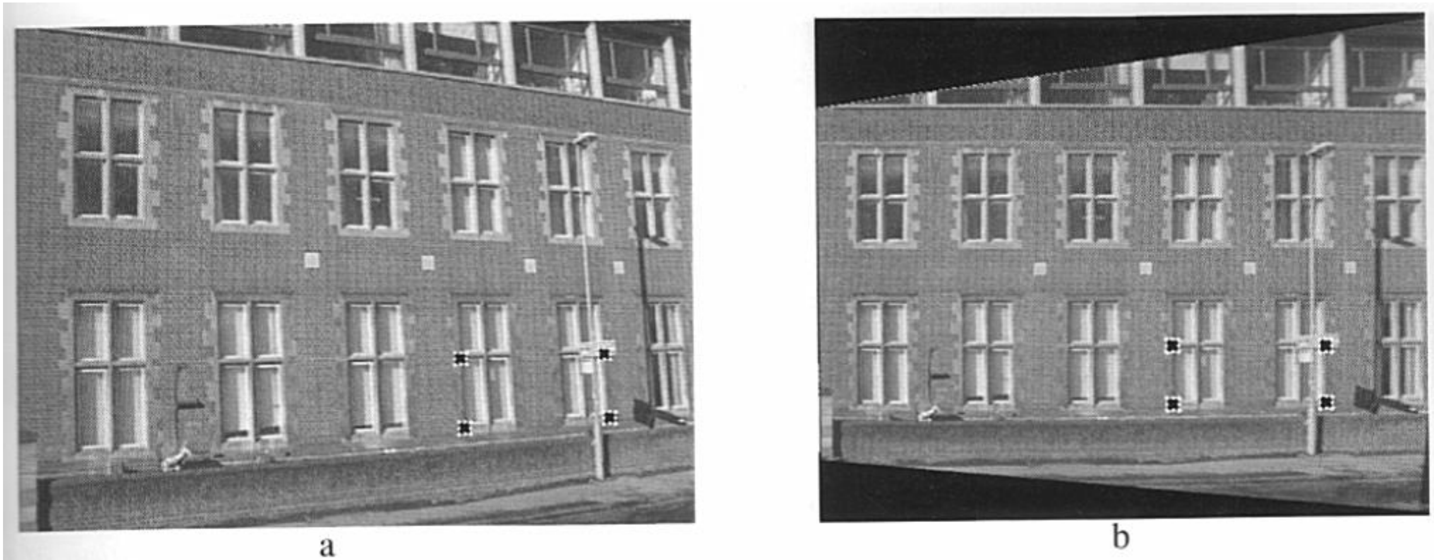
black area
where no pixel
maps to

Panoramas



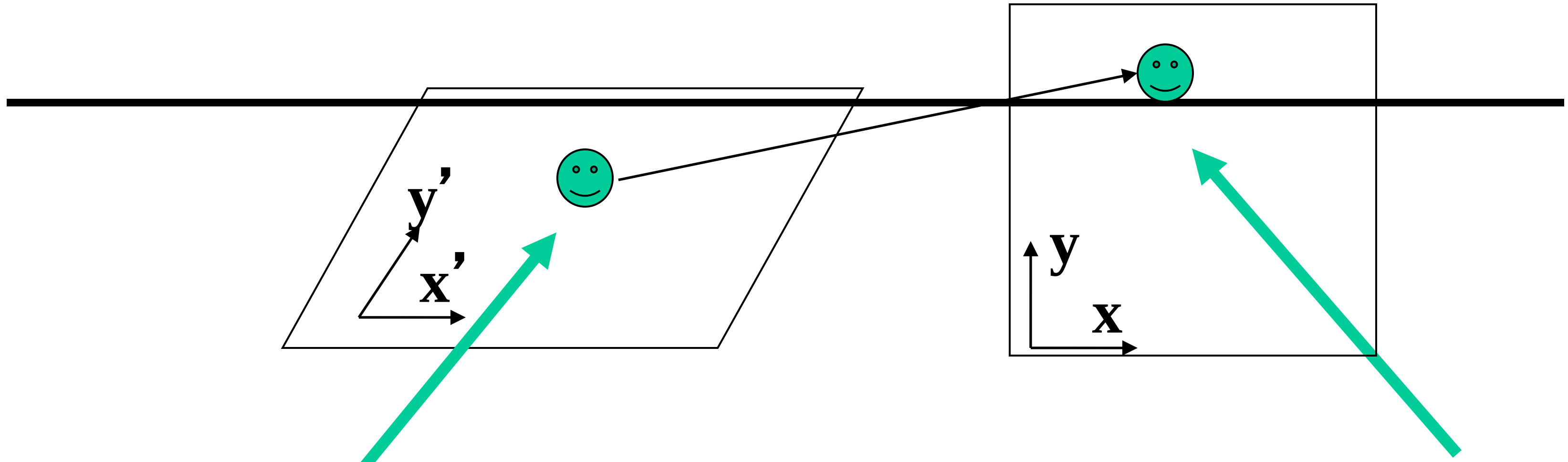
Pick one image (red)
Warp the other images towards it (usually, one by one)
blend

4 point algorithm



$$H = \begin{bmatrix} h11 & h12 & h13 \\ h21 & h22 & h23 \\ h31 & h32 & h33 \end{bmatrix}$$

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



$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} \mathrm{h11} & \mathrm{h12} & \mathrm{h13} \\ \mathrm{h21} & \mathrm{h22} & \mathrm{h23} \\ \mathrm{h31} & \mathrm{h32} & \mathrm{h33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$x' = \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3}$$

How many independent para? Can we always set h33 = 1?

4 points direct solution

For each point x_i, we have

$$\mathrm{H}\mathrm{x}_i = \begin{bmatrix} h^{1T}\mathrm{x}_i \\ h^{2T}\mathrm{x}_i \\ h^{3T}\mathrm{x}_i \end{bmatrix}$$

Since $\mathrm{x}'_i = (x'_i, y'_i, w'_i) = \mathrm{H}\mathrm{x}_i$ **Satisfies:**

$$\mathrm{x}'_i \times \mathrm{H}\mathrm{x}_i = 0$$

and

$$\mathrm{x}'_i \times \mathrm{H}\mathrm{x}_i = \begin{bmatrix} y'_i h^{3T}\mathrm{x}_i - w'_i h^{2T}\mathrm{x}_i \\ w'_i h^{1T}\mathrm{x}_i - x'_i h^{3T}\mathrm{x}_i \\ x'_i h^{2T}\mathrm{x}_i - y'_i h^{1T}\mathrm{x}_i \end{bmatrix}$$

4 points algorithm

Rewrite the equation as:

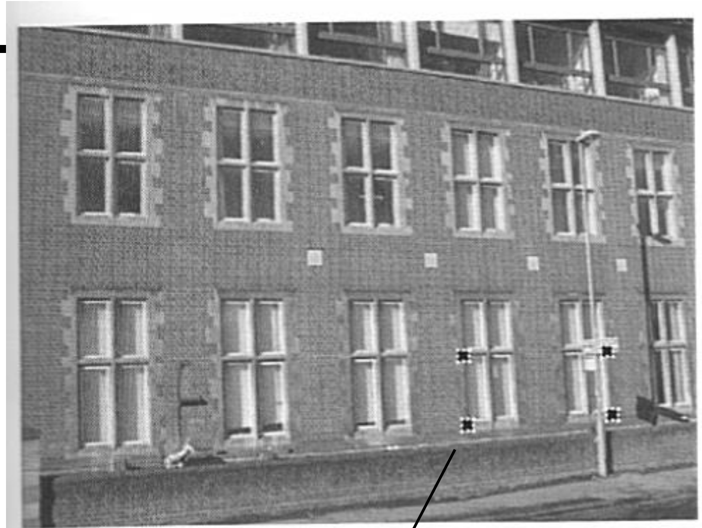
$$\begin{bmatrix} 0^T & -w'_i \mathrm{x}_i^T & y'_i \mathrm{x}_i^T \\ -w'_i \mathrm{x}_i^T & 0^T & -x'_i \mathrm{x}_i^T \\ -y'_i \mathrm{x}_i^T & x'_i \mathrm{x}_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

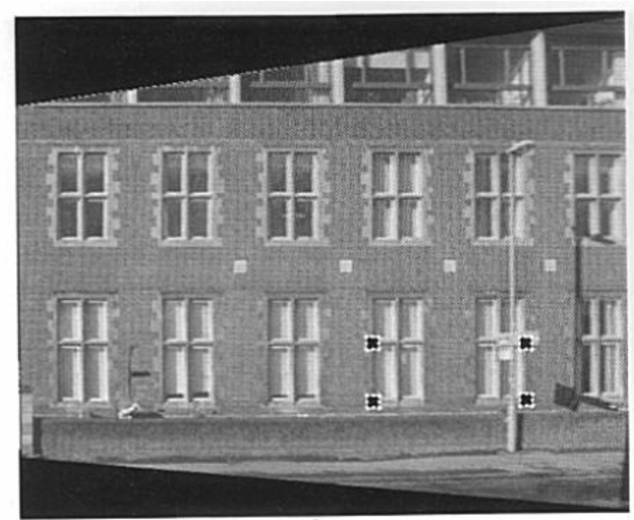
$$A_i \mathrm{h} = 0$$

1 point gives two independent equations,

H has 8 independent parameters => need 4 points

4 point algorithm


a


b

$$\mathbf{A} = \left(\begin{bmatrix} 0^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ -w'_i \mathbf{x}_i^T & 0^T & -x'_i \mathbf{x}_i^T \\ -y'_i \mathbf{x}_i^T & x'_i \mathbf{x}_i^T & 0^T \end{bmatrix} \right) \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix}$$

Compute

[v,d] = Eig(A^T A),

set h = eigenvector with the smallest eigenvalue

Overview

Feature Matching

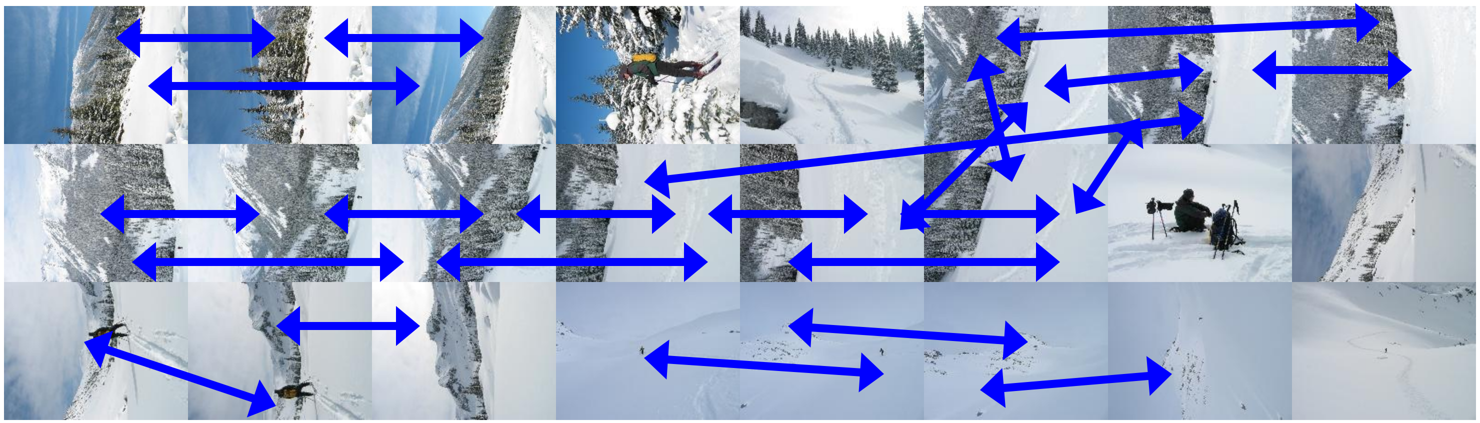
Image Matching

- RANSAC for Homography

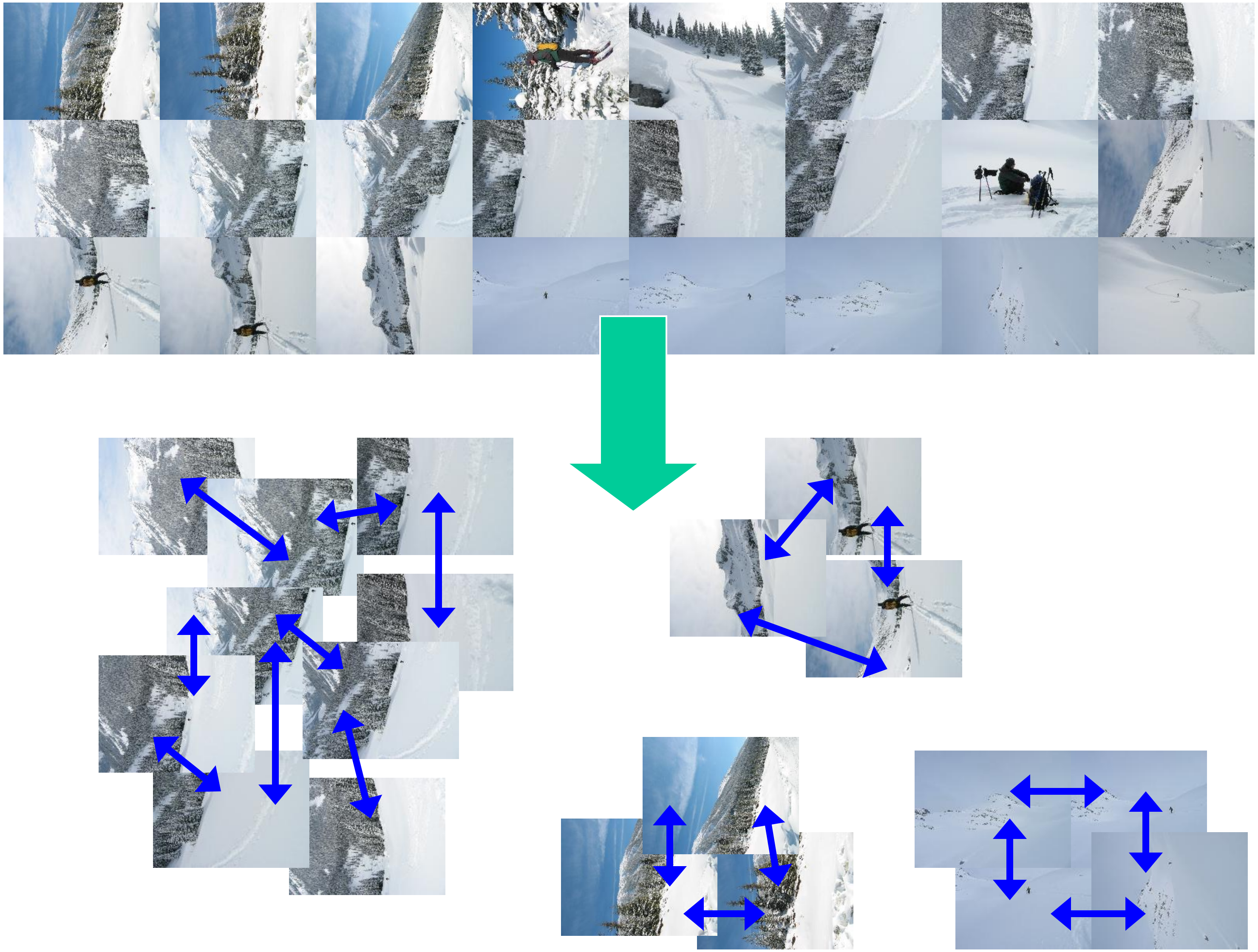
Multi-band Blending

Results

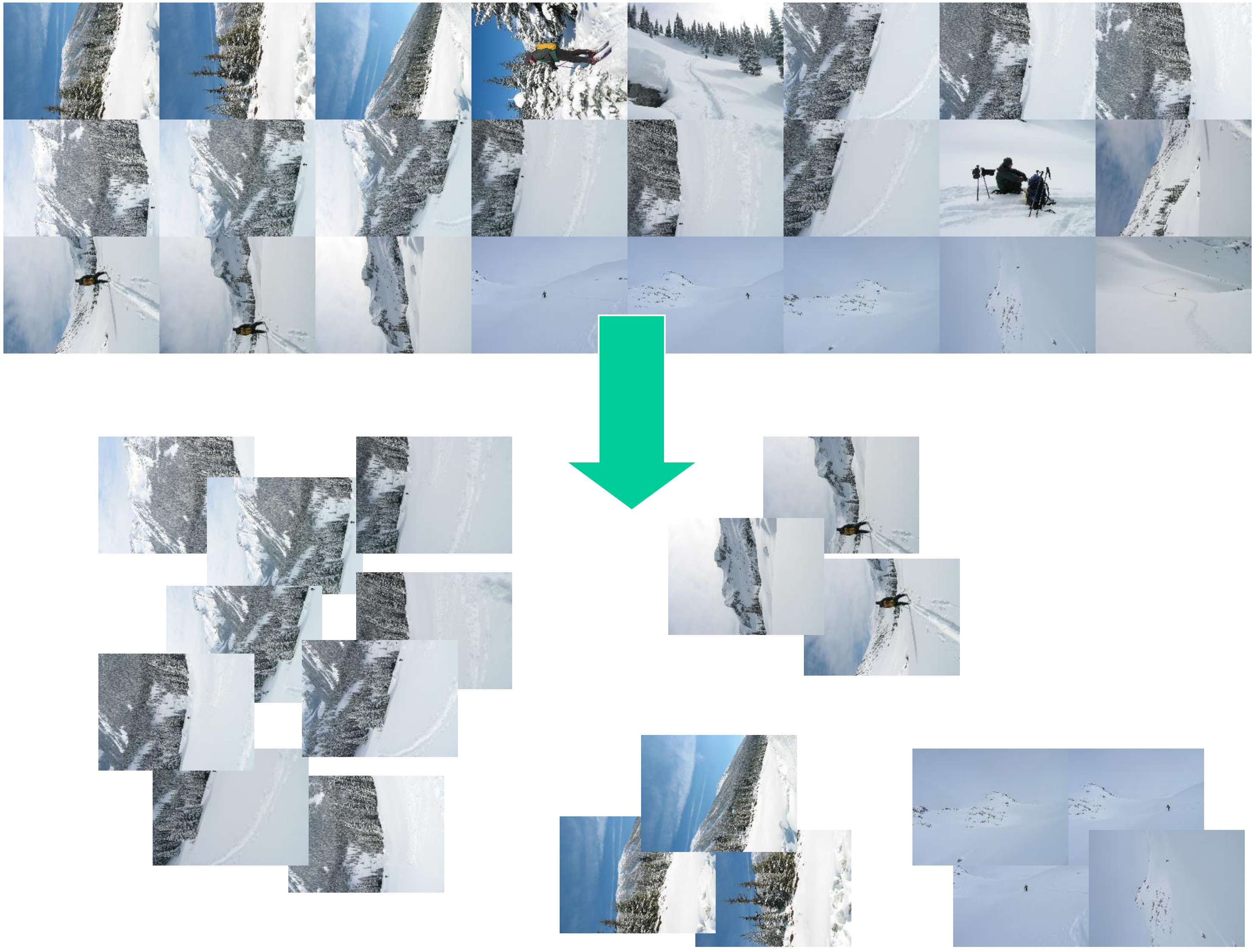
Finding the panoramas



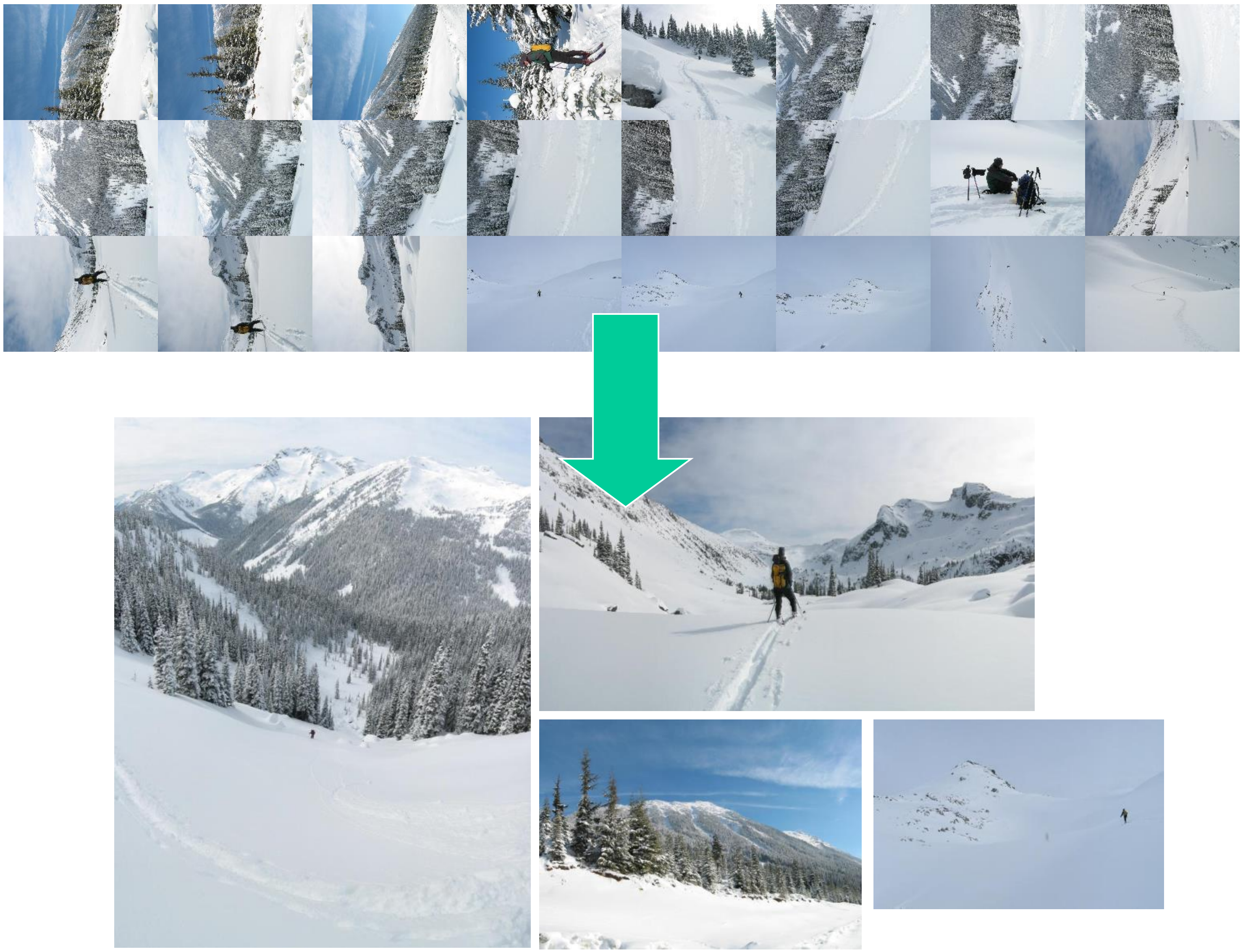
Finding the panoramas



Finding the panoramas



Finding the panoramas



Multi-band Blending

Burt & Adelson 1983

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

2-band Blending

Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

Overview

Feature Matching

Image Matching

Bundle Adjustment

Multi-band Blending

Results

Conclusions

Results