Automatic Panoramic Image Stitching using Local Features

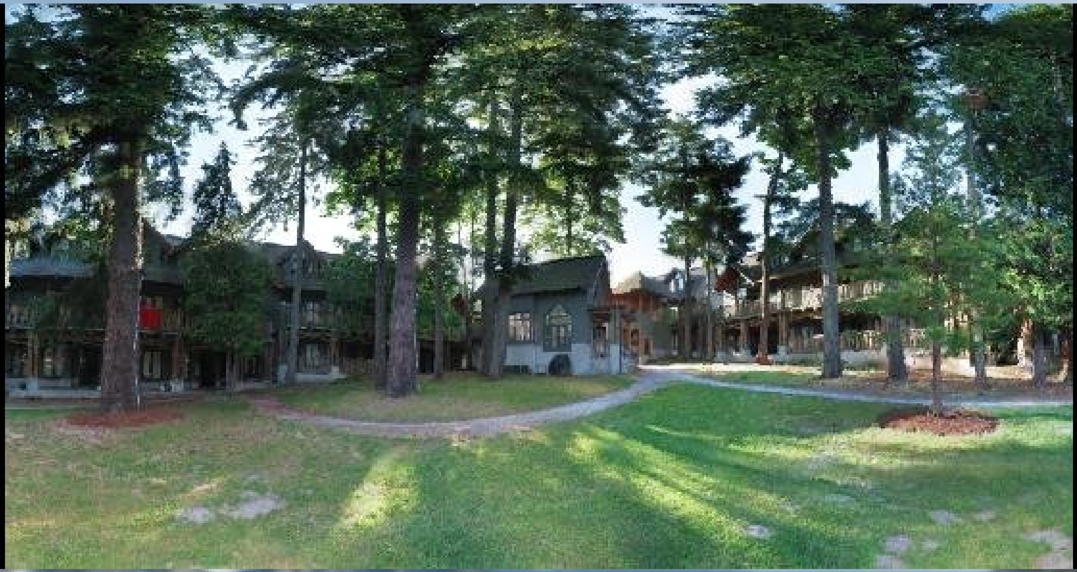
Matthew Brown and David Lowe,
University of British Columbia

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 Use Local Features?

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

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

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

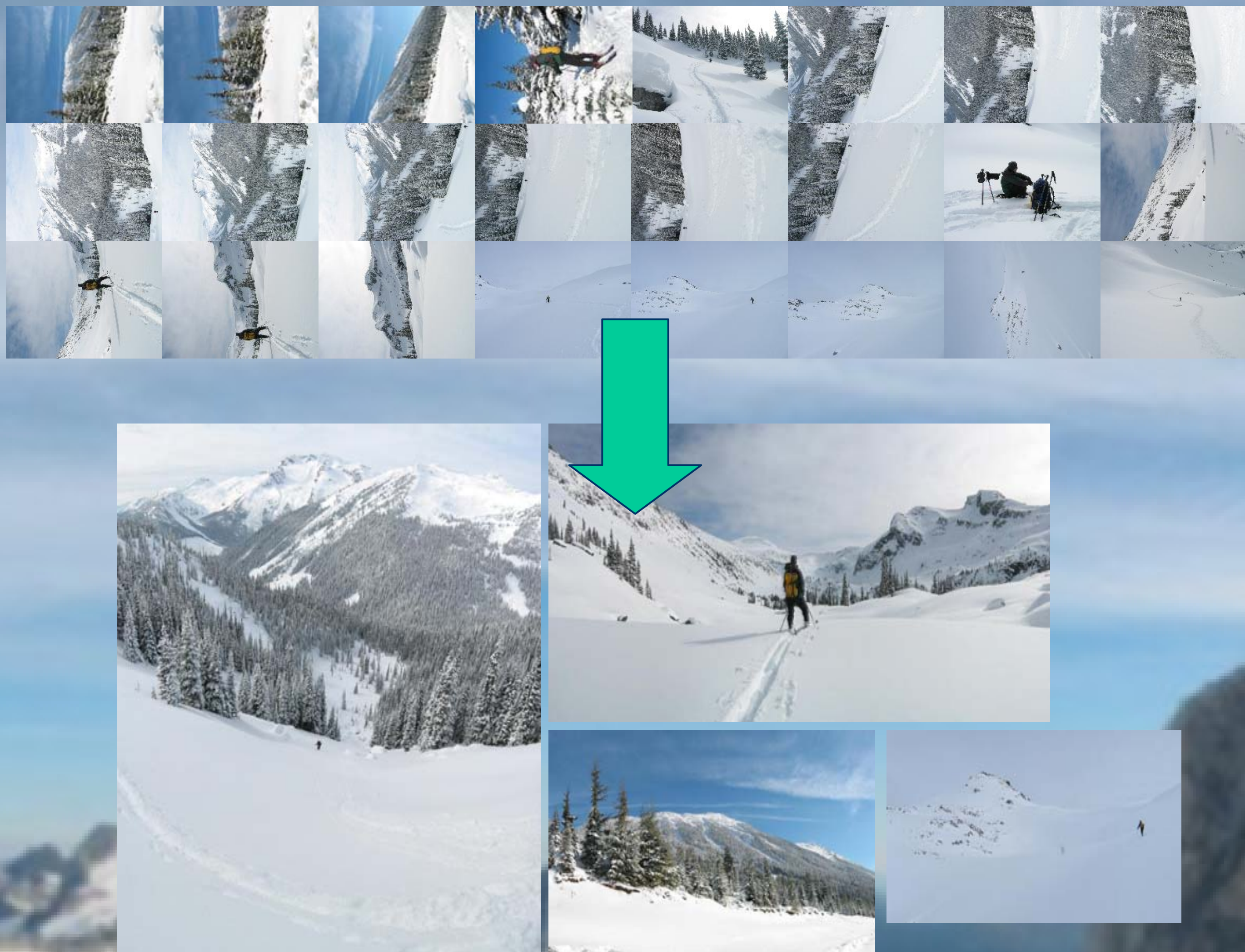
- 2D Rotations (θ, ϕ)
  - Ordering $\nRightarrow$ matching images

Why Use Local Features?

- 1D Rotations (θ)
  - Ordering $\Rightarrow$ matching images

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

Recognising Panoramas



[Brown and Lowe ICCV 2003, IJCV 2007]

Automatic Stitching

- Feature Matching
- Image Matching
- Image Alignment
- Rendering
- Results
- Conclusions

Automatic Stitching

- Feature Matching
  - SIFT Features
  - Nearest Neighbour Matching
- Image Matching
- Image Alignment
- Rendering
- Results
- Conclusions

Invariant Features

- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

SIFT Features

- Invariant Features
  - Establish invariant frame
    - Maxima/minima of scale-space DOG $\Rightarrow$ x, y, s
    - Maximum of distribution of local gradients $\Rightarrow \theta$
  - Form descriptor vector
    - Histogram of smoothed local gradients
    - 128 dimensions
- SIFT features are…
  - Geometrically invariant to similarity transforms,
    - some robustness to affine change
  - Photometrically invariant to affine changes in intensity

Automatic Stitching

- Feature Matching
  - SIFT Features
  - Nearest Neighbour Matching
- Image Matching
- Image Alignment
- Rendering
- Results
- Conclusions

Nearest Neighbour Matching

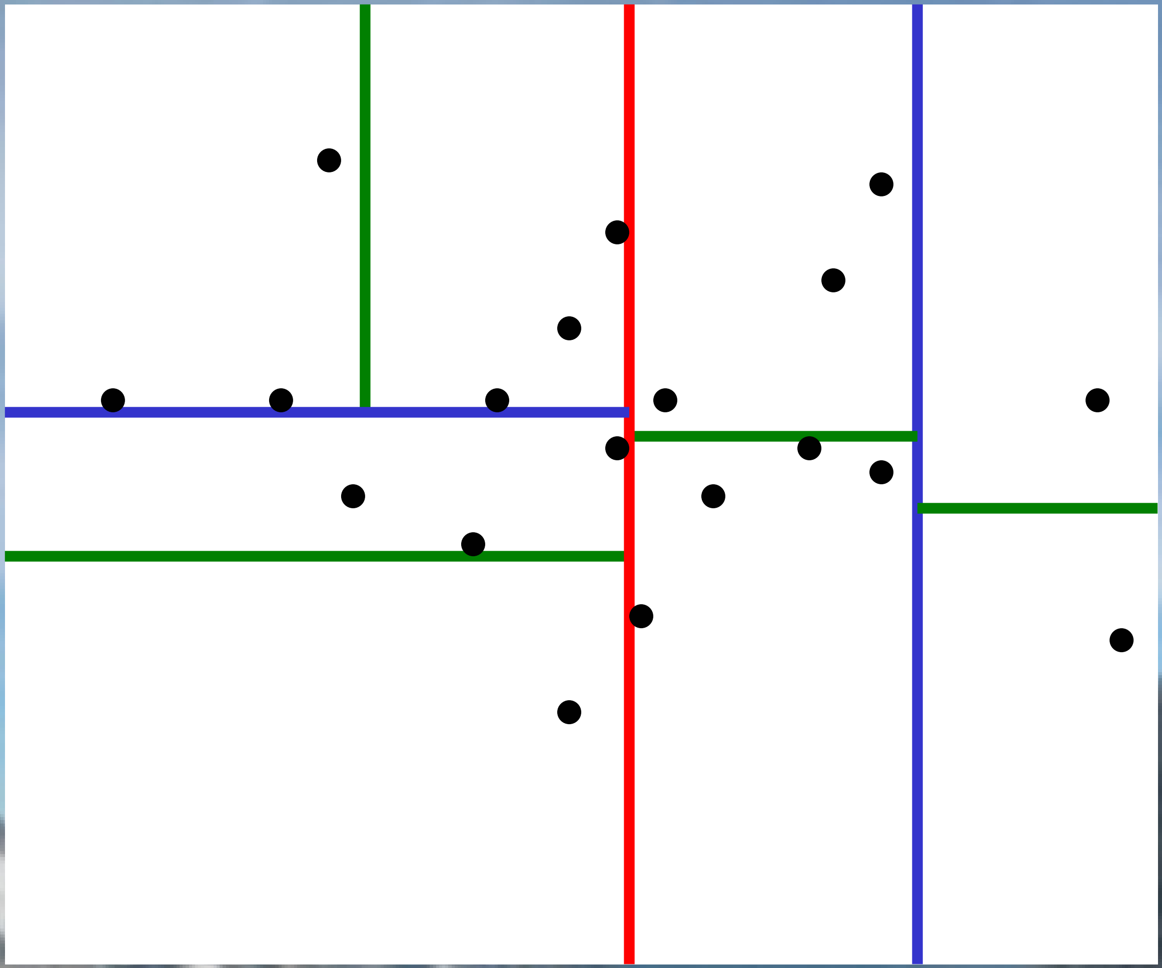
- Nearest neighbour matching

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

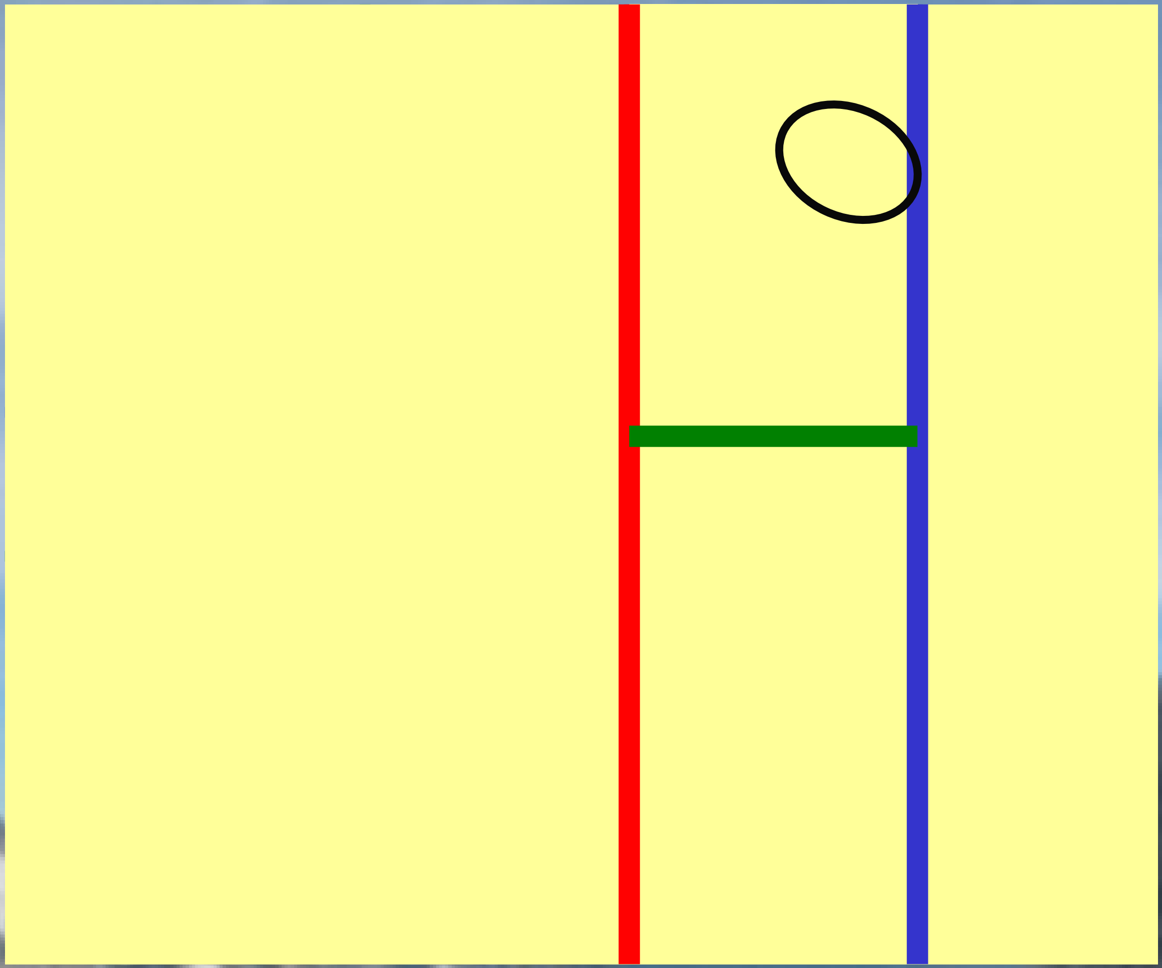
[Beis Lowe 1997, Nene Nayar 1997, Gray Moore 2000, Shakhnarovich 2003]

- 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(n log n)
- Find k-NN for each feature
  - k ≈ number of overlapping images (we use k = 4)

K-d tree



K-d tree



Automatic Stitching

- Feature Matching
- Image Matching
  - Motion Model
  - RANSAC
- Image Alignment
- Rendering
- Results
- Conclusions

2D Motion Models

- Linear (affine)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}$$

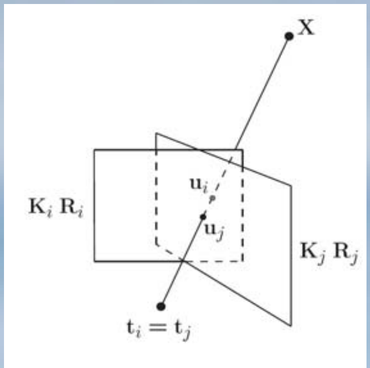
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Homography

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

Homography for Rotation

- Projection equation $\tilde{\mathbf{u}} = \mathbf{K}(\mathbf{R}|\mathbf{t})\tilde{\mathbf{X}}$



set t = 0 for a pair

$$\tilde{\mathbf{u}}_i = \mathbf{K}_i(\mathbf{R}_i|0)\tilde{\mathbf{X}} = \mathbf{K}_i\mathbf{R}_i\mathbf{X}$$
$$\tilde{\mathbf{u}}_j = \mathbf{K}_j(\mathbf{R}_j|0)\tilde{\mathbf{X}} = \mathbf{K}_j\mathbf{R}_j\mathbf{X}$$

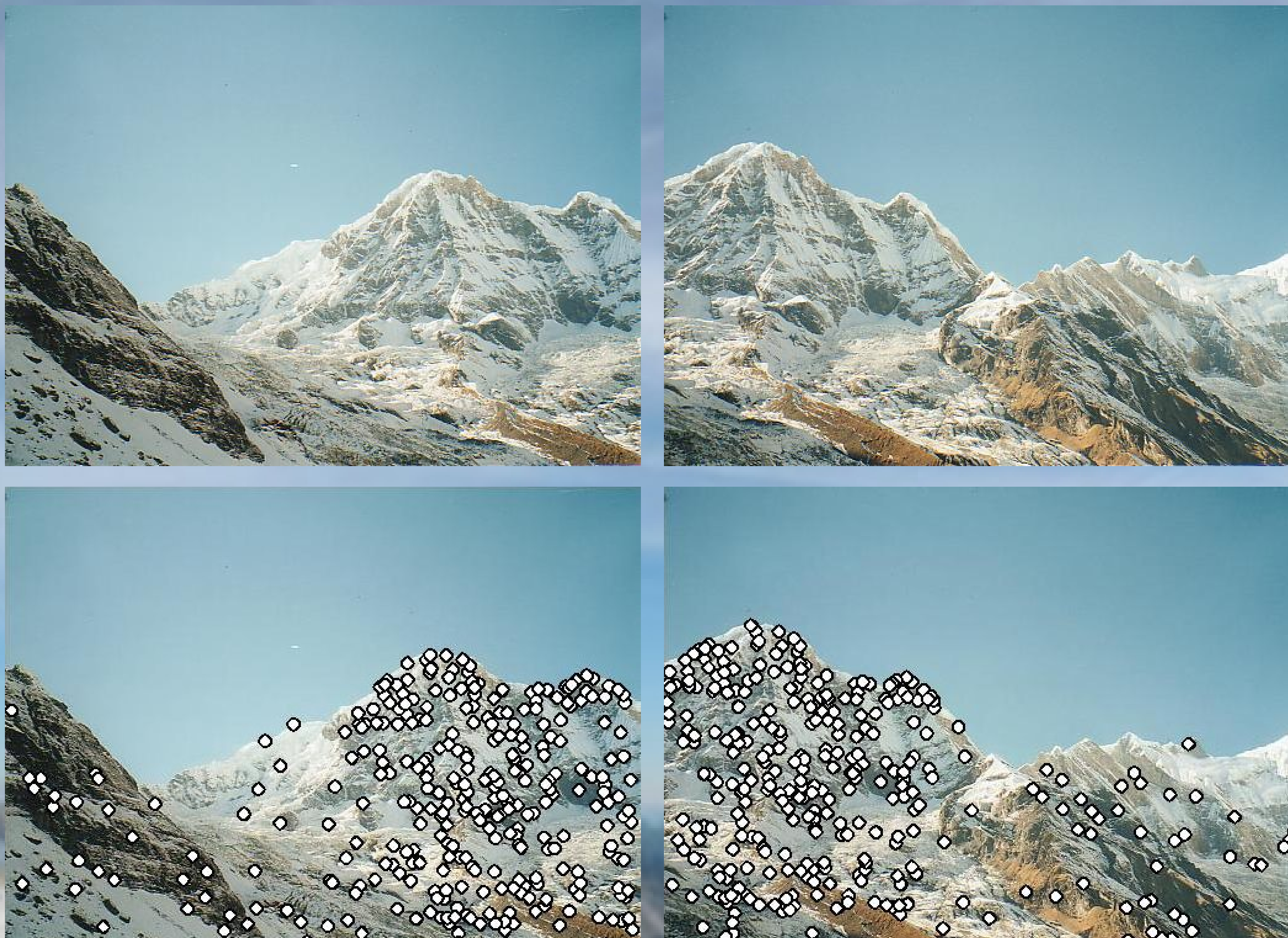
- $\rightarrow$ pairwise homographies $\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j$

where $\mathbf{H}_{ij} = \mathbf{K}_i\mathbf{R}_i\mathbf{R}_j^T\mathbf{K}_j^{-1}$

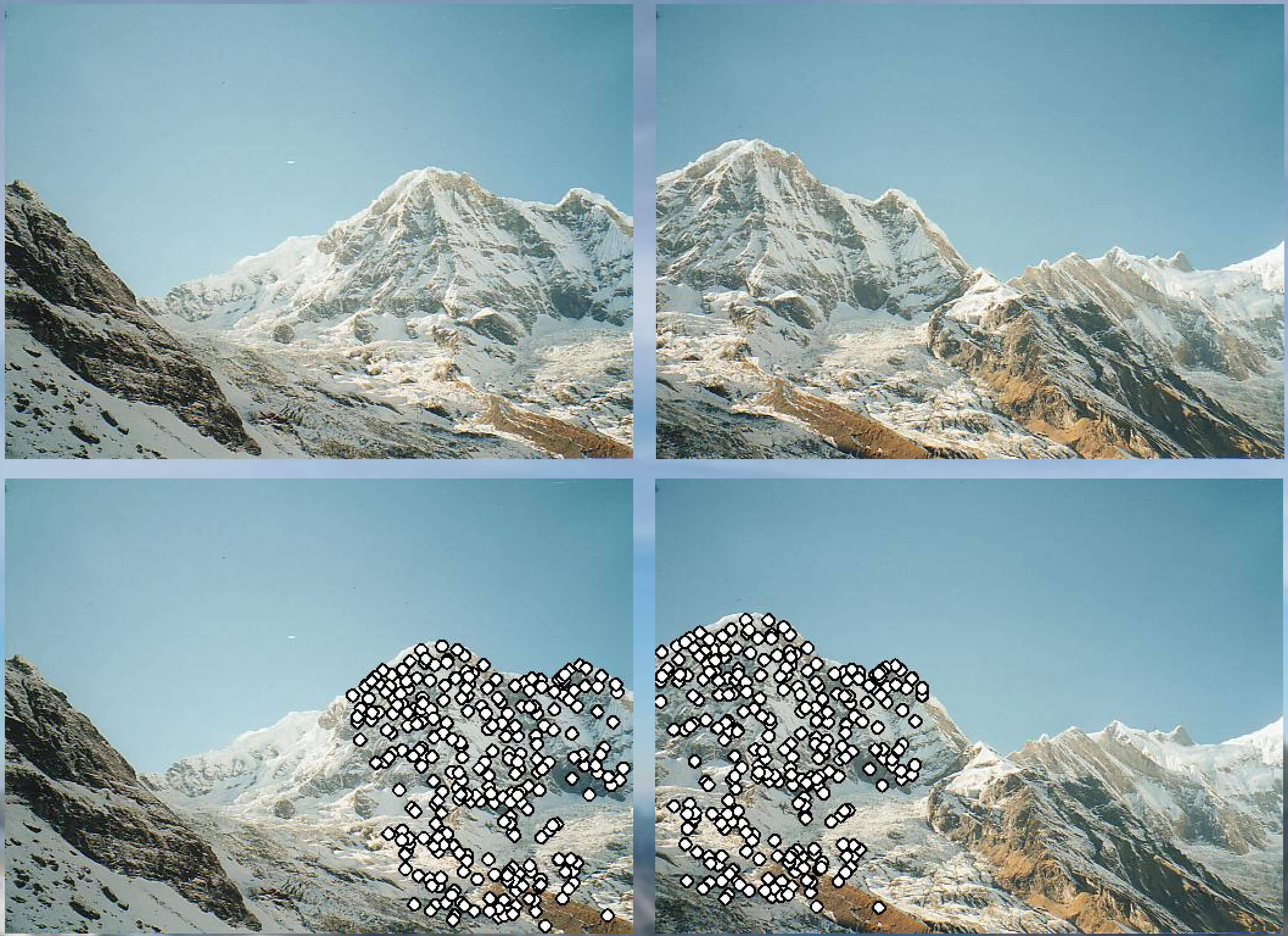
Automatic Stitching

- Feature Matching
- Image Matching
  - Motion Model
  - RANSAC
- Image Alignment
- Rendering
- Results
- Conclusions

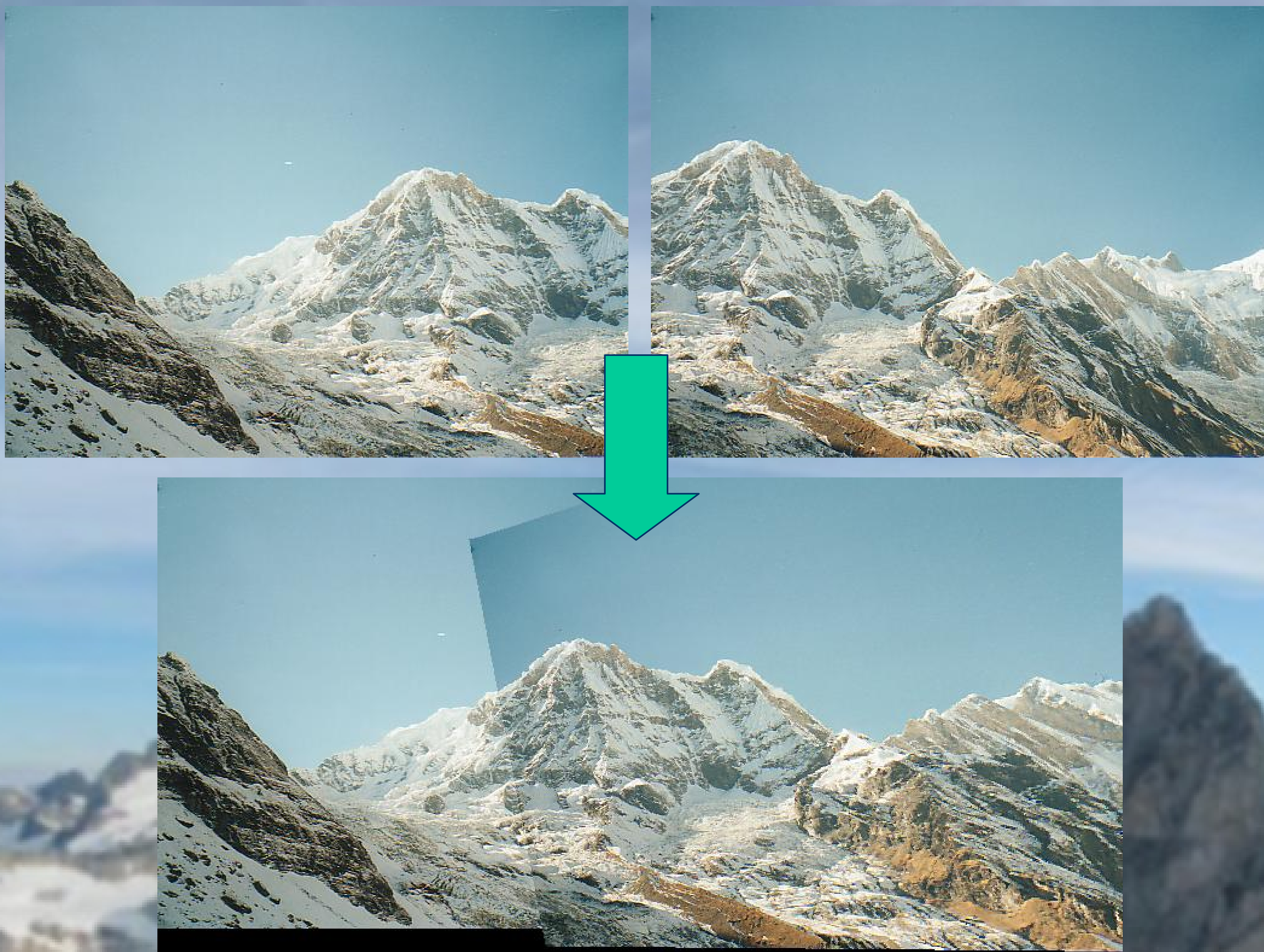
RANSAC for Homography



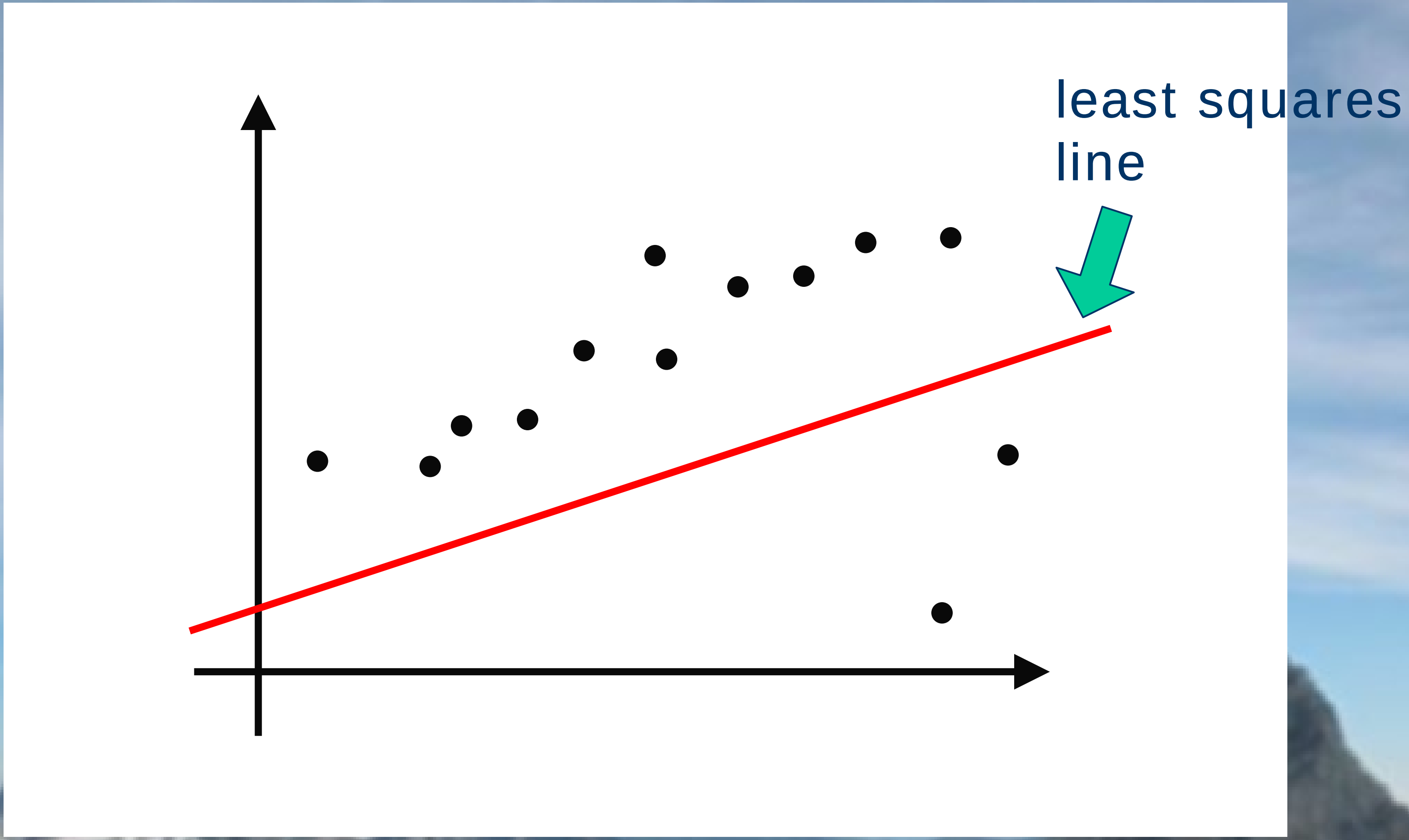
RANSAC for Homography



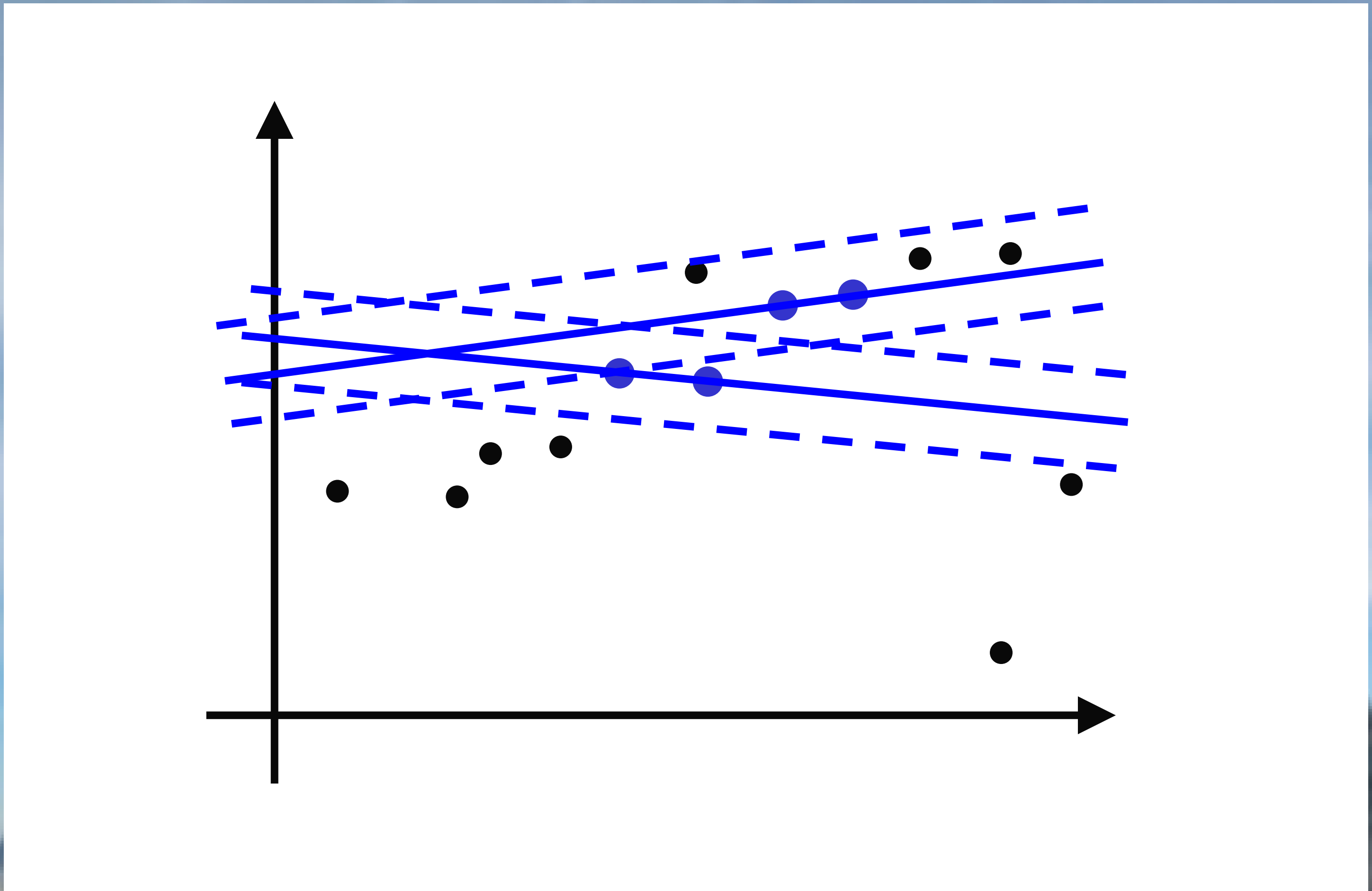
RANSAC for Homography



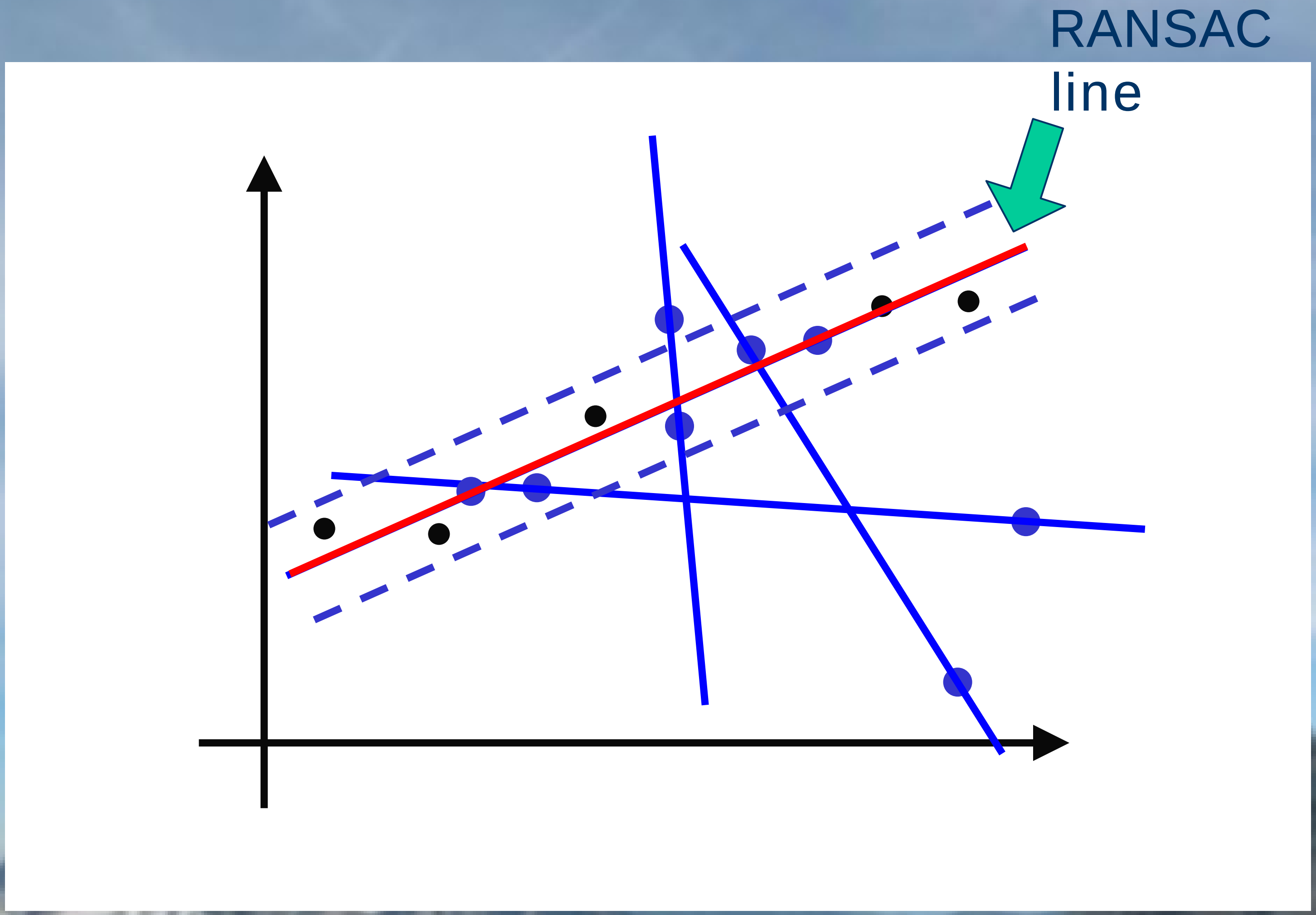
RANSAC: 1D Line Fitting



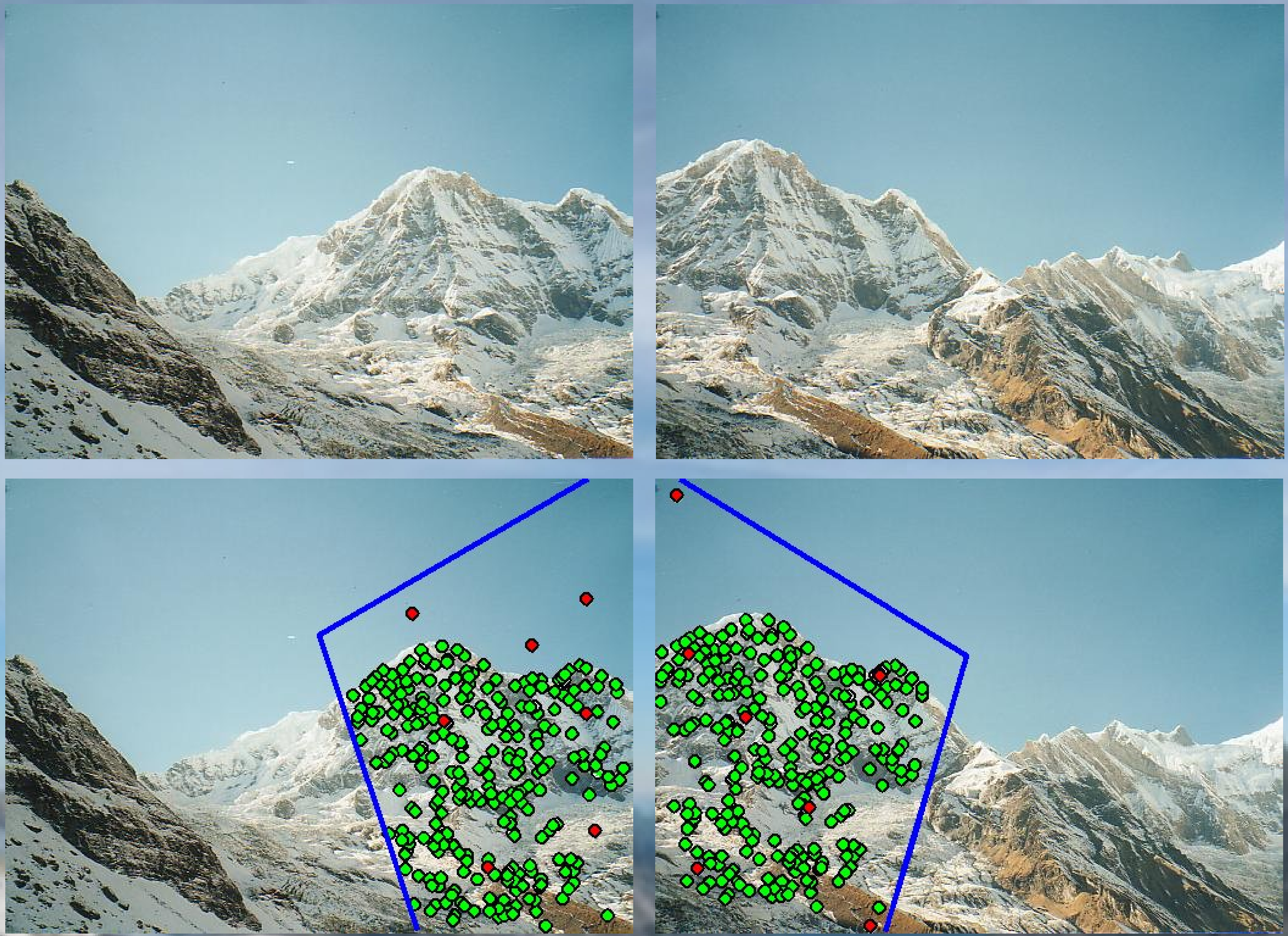
RANSAC: 1D Line Fitting



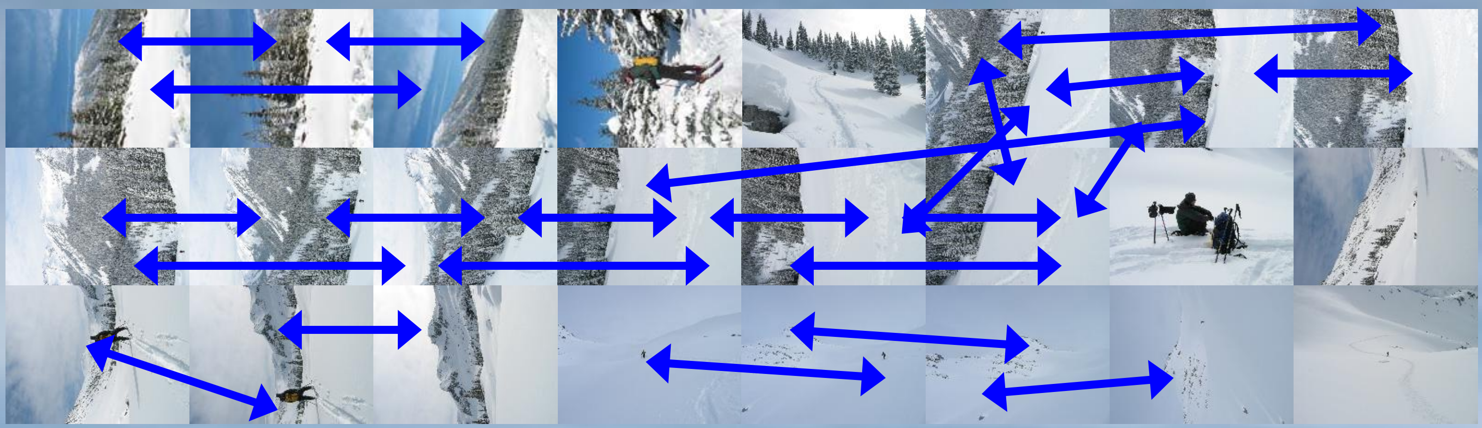
RANSAC: 1D Line Fitting



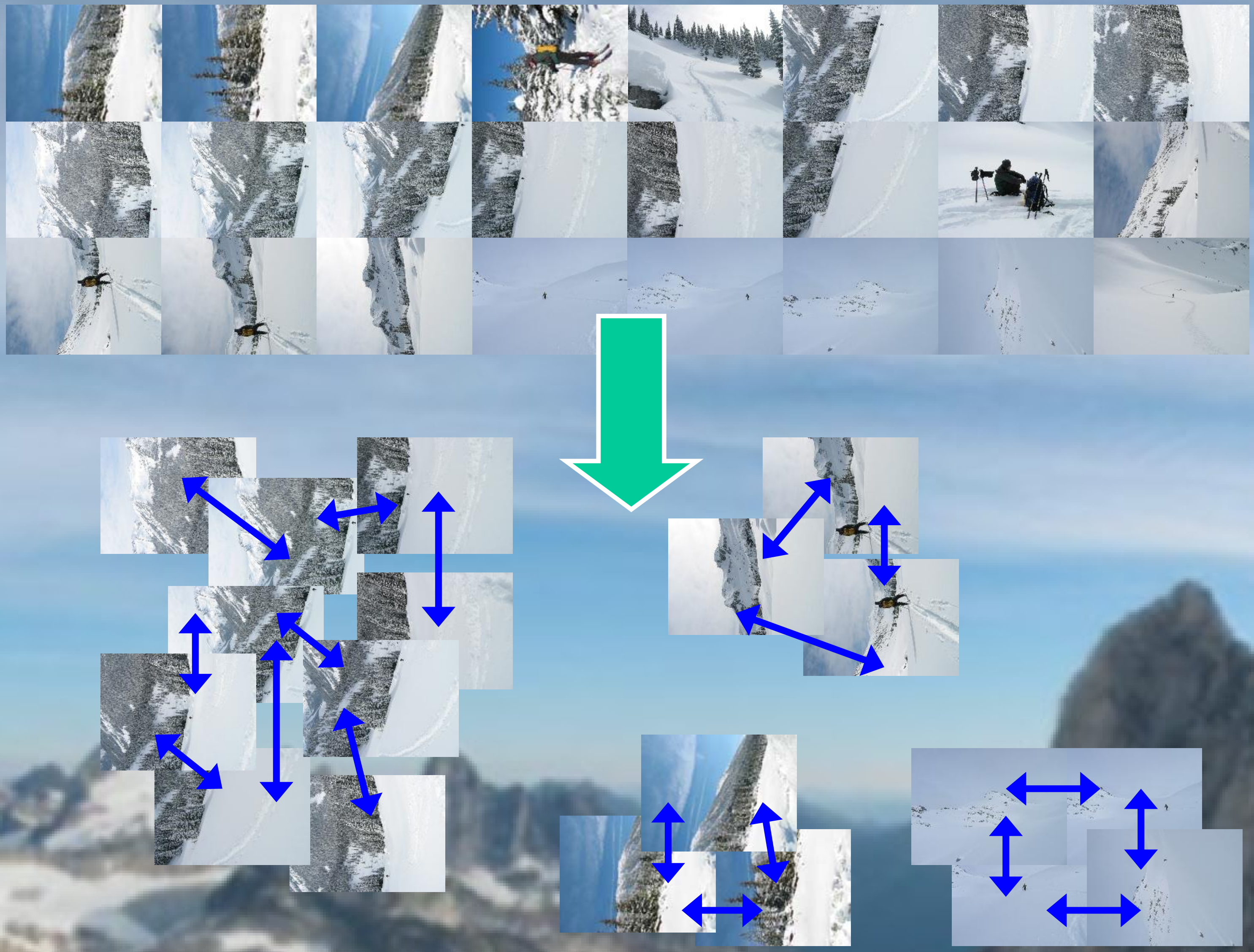
Match Verification



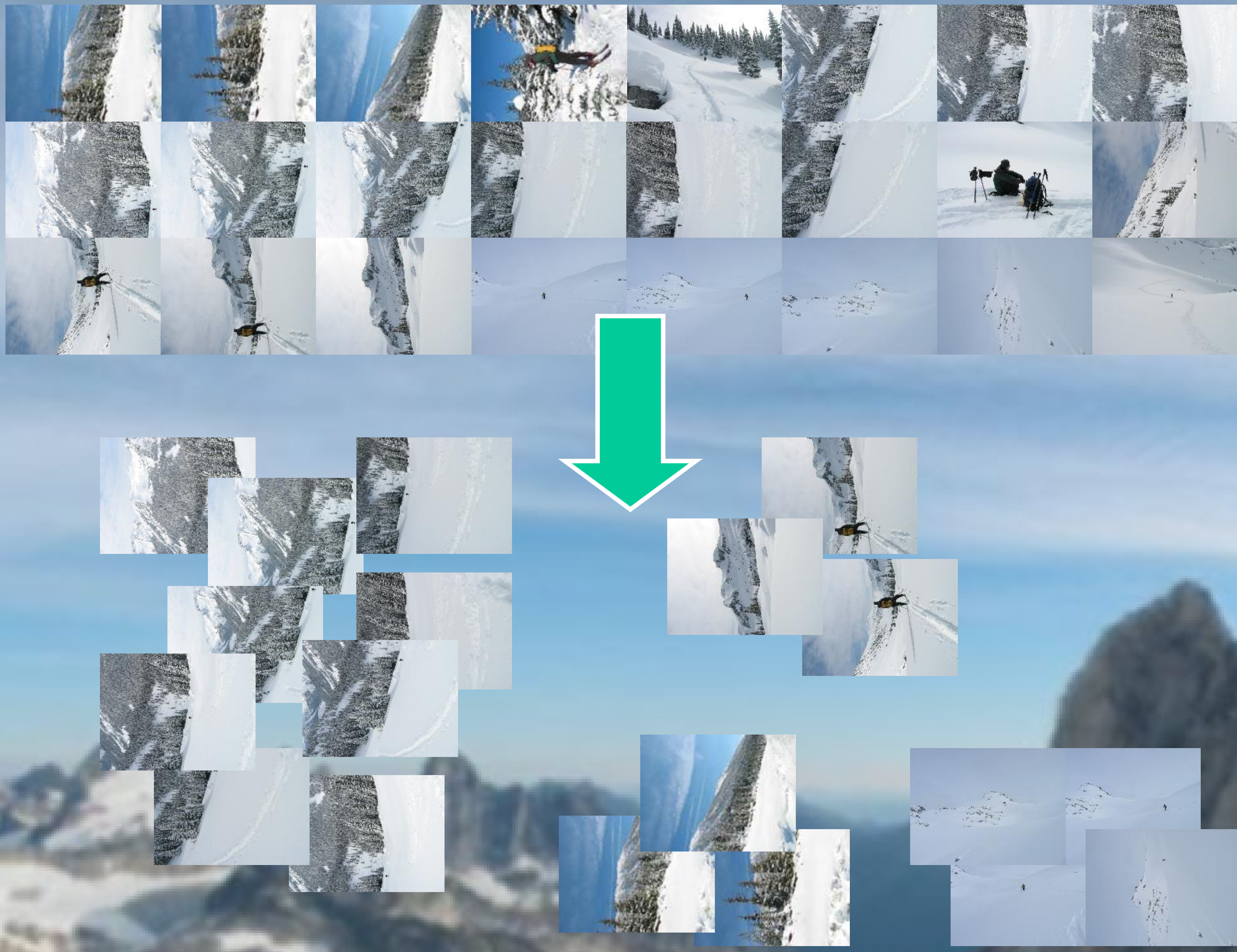
Finding the panoramas



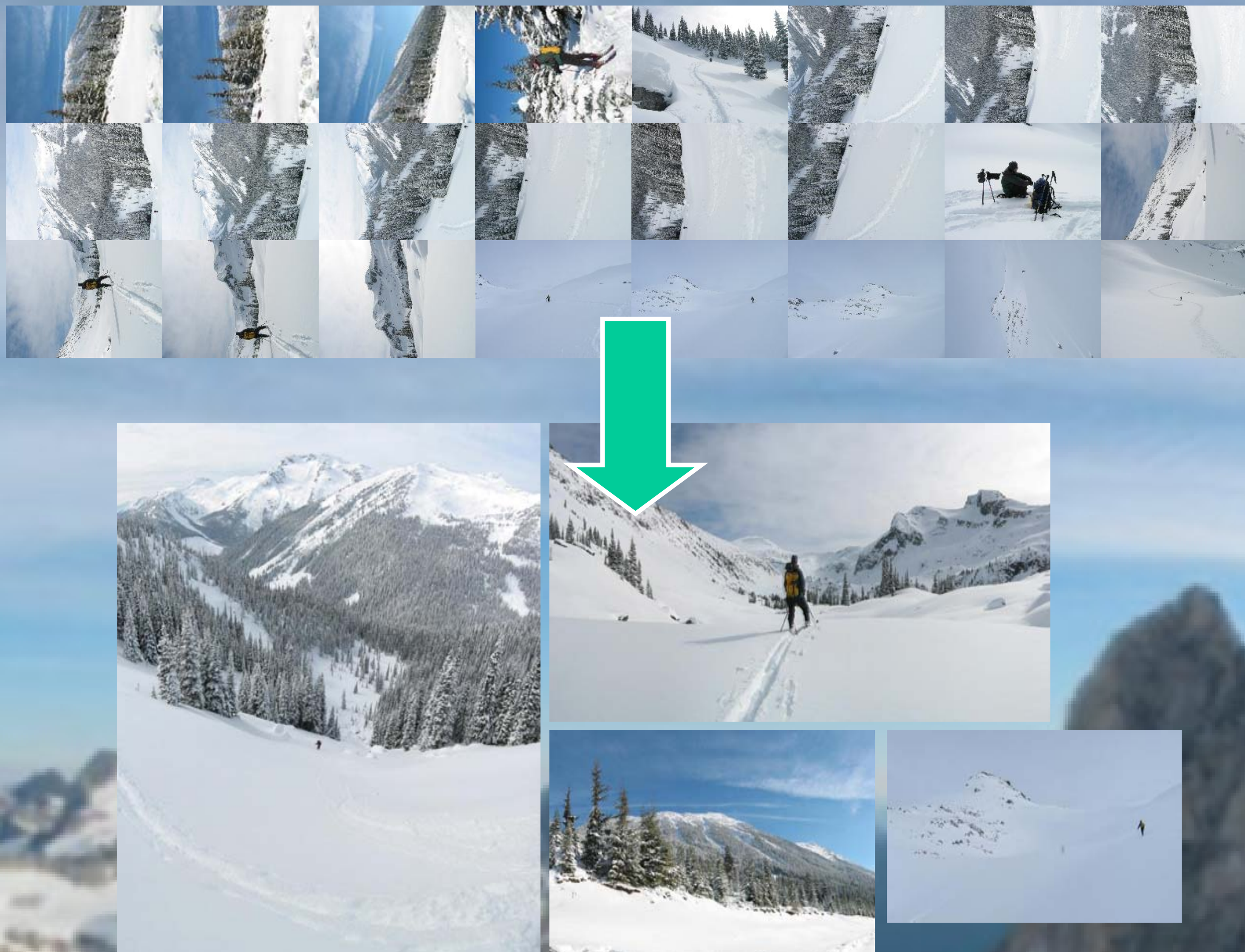
Finding the panoramas



Finding the panoramas



Finding the panoramas



Automatic Stitching

- Feature Matching
- Image Matching
- Image Alignment
  - Bundle Adjustment
  - Automatic Straightening
- Rendering
- Results
- Conclusions

Motion Model Revisited

- Recall our image motion model

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

- Parameterise each camera by rotation and focal length

$$\mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_\times}, \ \ [\boldsymbol{\theta}_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}$$

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Bundle Adjustment

- Sum of squared projection errors

$$e = \sum_{i=1}^{n} \sum_{j \epsilon I(i)} \sum_{k \epsilon F(i,j)} f(\mathbf{r}_{ij}^{k})$$

  - n = #images
  - I(i) = set of image matches to image i
  - F(i, j) = set of feature matches between images i,j
  - r_{ij}^{k} = residual of k^{th} feature match between images i,j

- Huber (robust) error function

$$f(\mathbf{x}) = \begin{cases} |\mathbf{x}|^2, & \text{if } |\mathbf{x}| < \sigma \\ 2\sigma|\mathbf{x}| - \sigma^2, & \text{if } |\mathbf{x}| \geq \sigma \end{cases}$$

Bundle Adjustment

- Adjust rotation, focal length of each image to minimise error in matched features

Bundle Adjustment

- Adjust rotation, focal length of each image to minimise error in matched features

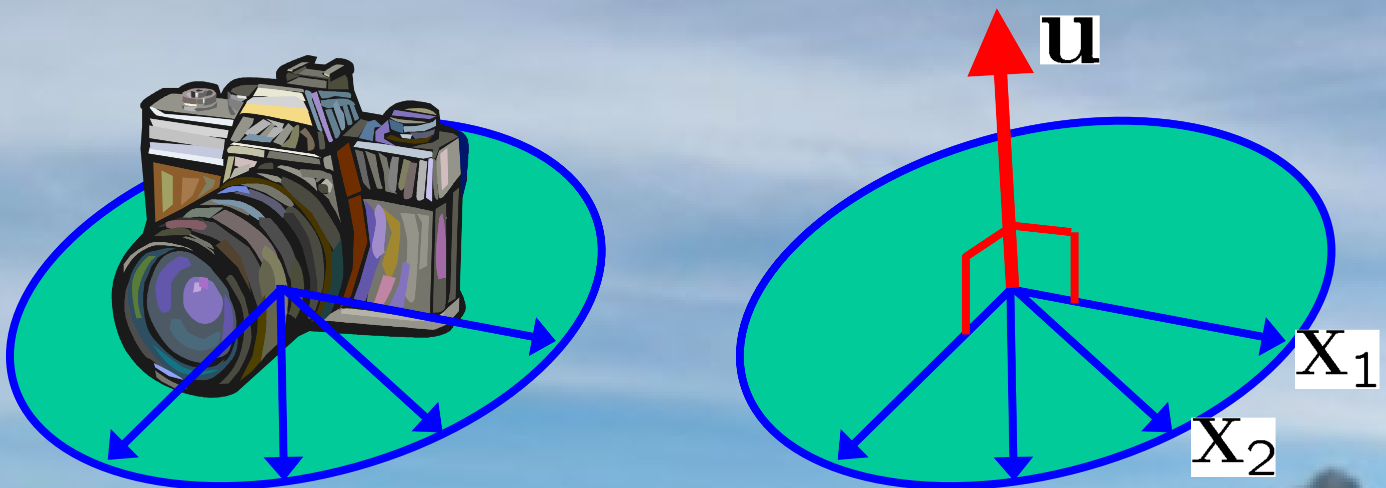
Automatic Stitching

- Feature Matching
- Image Matching
- Image Alignment
  - Bundle Adjustment
  - Automatic Straightening
- Rendering
- Results
- Conclusions

Automatic Straightening

Automatic Straightening

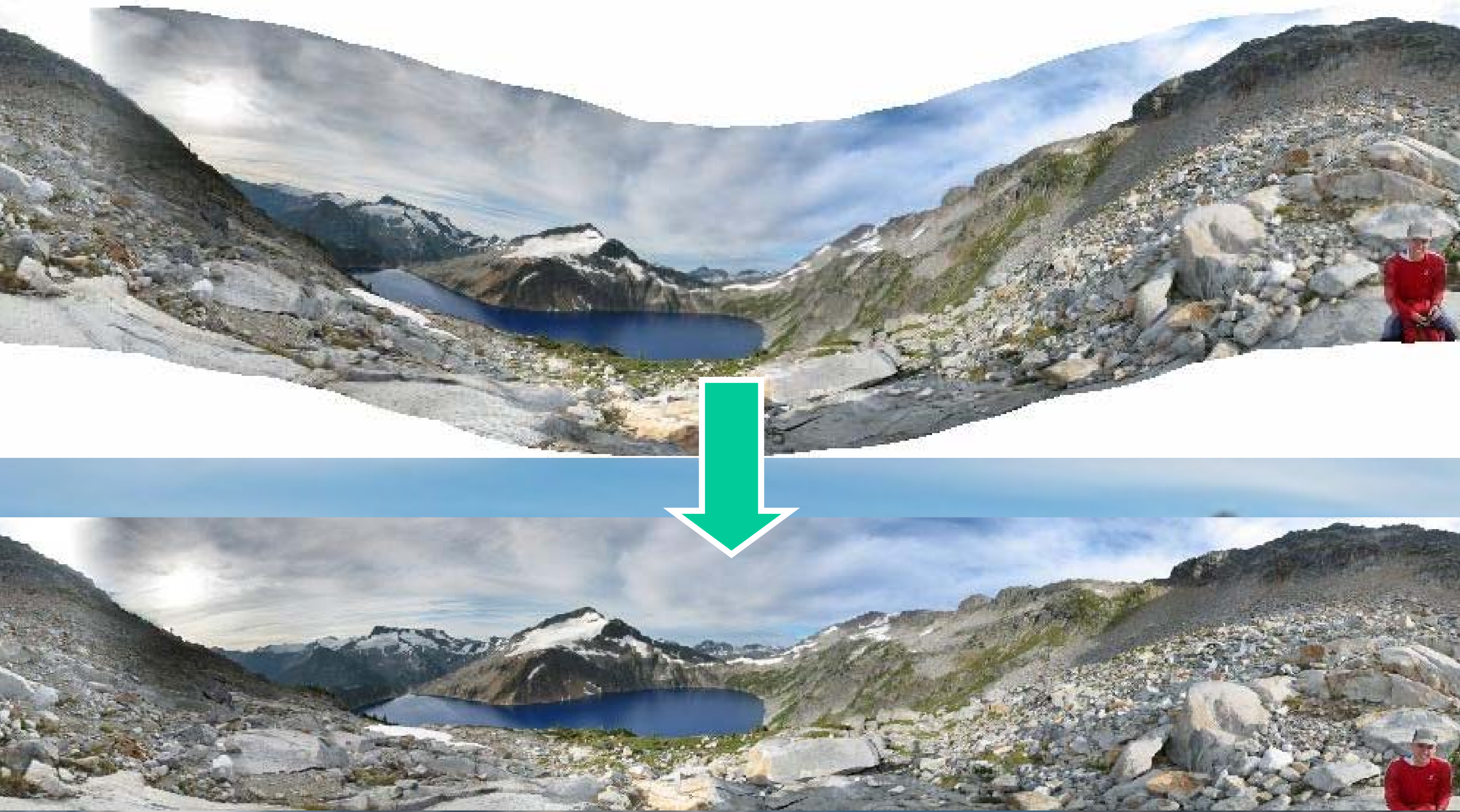
- Heuristic: user does not *twist* camera relative to horizon



- Up-vector perpendicular to plane of camera x vectors

$$\left(\sum_i \mathbf{X}_i \mathbf{X}_i^T\right)\mathbf{u} = \mathbf{0}$$

Automatic Straightening



Automatic Stitching

- Feature Matching
- Image Matching
- Image Alignment
- Rendering
- Results
- Conclusions

Gain Compensation

- No gain compensation

Gain Compensation

- Gain compensation

  - Single gain parameter g_i for each image

$$e = \sum_i \sum_j \sum_{\mathbf{u}_i \epsilon \mathcal{R}(i,j)} (g_i I_i(\mathbf{u}_i) - g_j I_j(\mathbf{u}_j))^2$$

(Better solution = HDR [Debevec 1997])

Multi-band Blending

- No blending

Multi-band Blending

- Linear blending

  - Each pixel is a weighted sum

$$I^{linear} = \frac{\sum_i I^i \, W^i}{\sum_i W^i}$$

Multi-band Blending

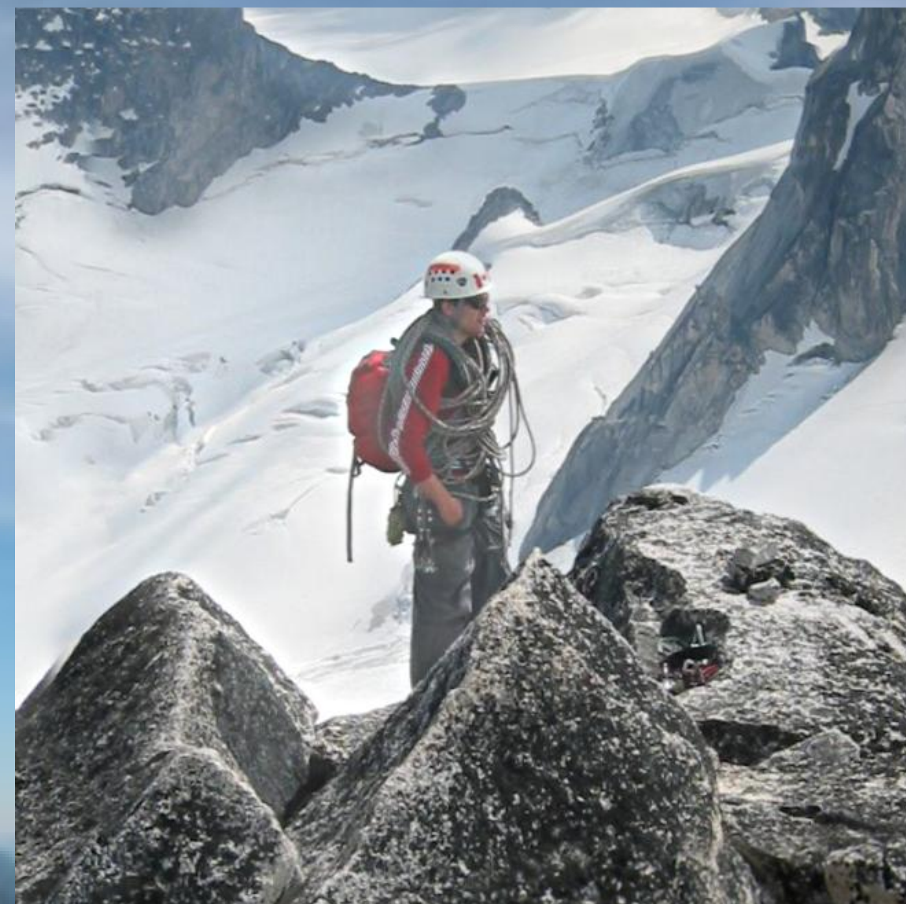
- Multi-band blending

  - Each pixel is a weighted sum (for each band)

$$I_{k\sigma}^{multi} = \frac{\sum_i I_{k\sigma}^{i}\ W_{k\sigma}^{i}}{\sum_i W_{k\sigma}^{i}}$$

Multi-band Blending

- Linear blending
- Multi-band blending



[Burt Adelson 1983]

2-band Blending

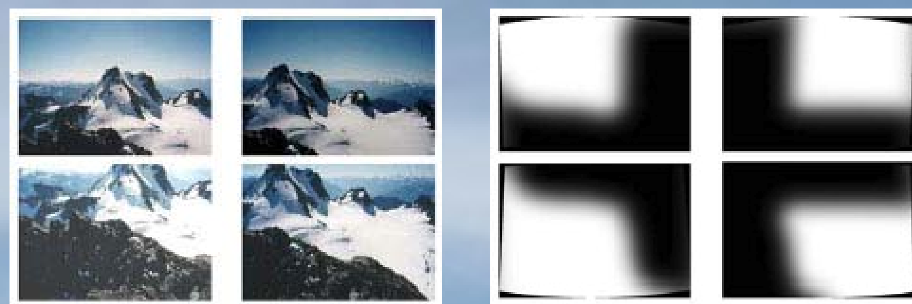
2-band Blending

Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Seam Selection

- (simple) Choose image with max “weight”:



- (better) ...also minimise error on seams

[Agarwala et al SIGGRAPH 04]

Automatic Stitching

- Feature Matching
- Image Matching
- Image Alignment
- Rendering
- Results
- Conclusions


Demo

Evaluation

- 200+ test sequences...

Ground Truth

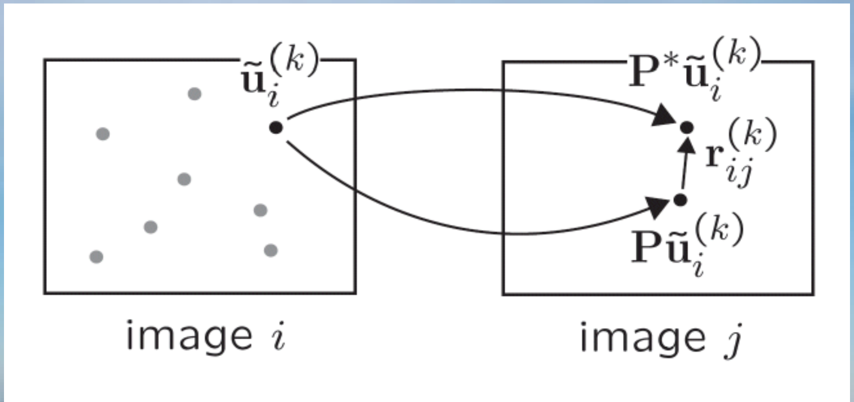
- Real: stitch “by hand”
- Synthetic: sample virtual camera views

Stitched panorama

Synthetic camera views

Error function

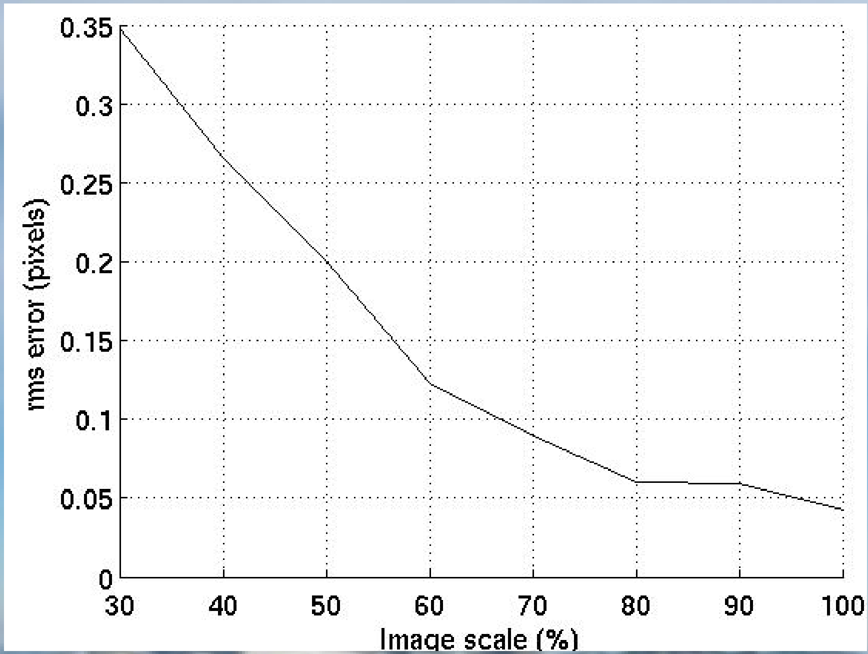
- Compare test stitch with ground truth
  - Ground truth $\mathbf{P}^* = \{\mathbf{P}_1^*, \mathbf{P}_2^*, ...\mathbf{P}_n^*\}$
  - Test stitch $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, ...\mathbf{P}_n\}$
- Evaluation function
  - sum of pairwise projection errors wrt ground truth

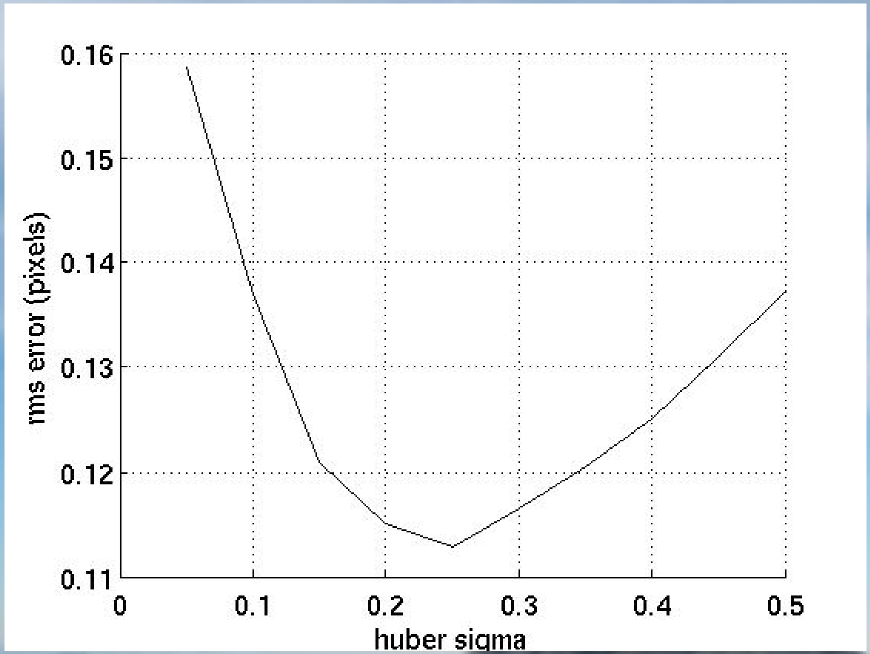


$$e(\mathbf{P}^*, \mathbf{P}) = \sum_i \sum_j \sum_{\mathbf{u}_i^{(k)} \epsilon \mathcal{O}(i,j)} |\mathbf{r}_{ij}^{(k)}|^2$$

Results

- Testing performance (image scale)
- Tuning parameters (Huber sigma)





Conclusions

- Image Stitching using Local Features
  - c.f. "direct methods": fast, robust,
  - 2D stitching is a recognition problem
- Multi-Image Matching Framework
  - Local features, RANSAC, bundle adjustment, blending
- Future Work
  - camera model += radial distortion, camera translation, scene motion, vignetting, HDR, flash …
  - Full 3D case
    - e.g. Photo Tourism [Snavely et al SIGGRAPH 2006]

http://research.microsoft.com/~brown