Computer Vision

7. Image Transformation, Matching and Panorama

I-Chen Lin

College of Computer Science
National Yang Ming Chiao Tung University

Outline

- Image transformation
- How to represent the environment with a few images?
  - Key point matching.
  - Automatic image stitching.

Several figures are from the following reference lists:

• Image-based Modeling and Rendering, SIGGRAPH'99 course notes.

• L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", Proc. SIGGRAPH'95, pp. 39-46.

• S.E. Chen, "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation", Proc. SIGGRAPH'95, pp. 29-38.

• M. Brown and D.G. Lowe, "Recognising panoramas," Proc., ICCV 2003, pp. 1218-25.

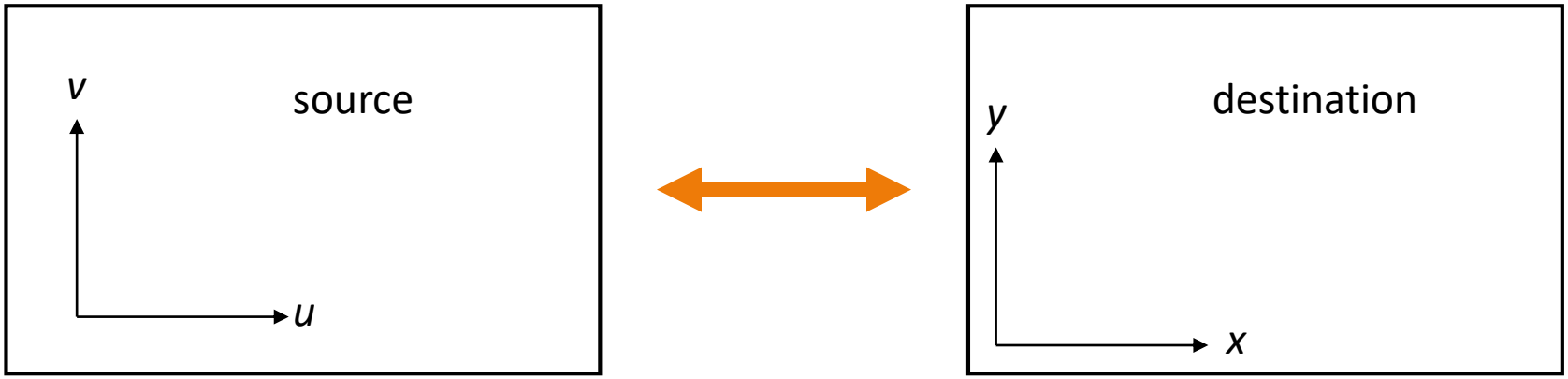
• M. Brown and D.G.Lowe, "Automatic Panoramic Image Stitching using Invariant Features," Intl. J. Computer Vision, vol.74, no.1, pp.59-73.

• R. Szeliski, "Image Alignment and Stitching: A Tutorial," Foundations and Trends in Computer Graphics and Vision, vol.2, no.1., pp.1-104.

Image warping and transformation

- Rearranging pixels of a picture.
  - It's useful for both image processing and for computer graphics (namely, for texture mapping).
- Finding corresponding points in the source and destination images.
- This function is called the "mapping" or "transformation".

Image warping (cont.)



Forward mapping : $(x, y) = f(u, v)$

Inverse mapping : $(u, v) = f'(x, y)$

Mapping types

- How to create the mapping systematically?
- Simple mappings
  - affine mapping
  - projective mapping
  - bilinear mapping
  - ……

Affine mappings

- $u = ax+by+c$
- $v = dx+ey+f$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- A combination of 2-D scale, rotation, and translation transformations.
- Allows a square to be distorted into any parallelogram.
- 6 degrees of freedom.
- Inverse is of same form (is also affine).

Projective mappings

- $u = (ax+by+c)/(gx+hy+i)$
- $v = (dx+ey+f)/(gx+hy+i)$

$$\begin{bmatrix} uq \\ vq \\ q \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$u = uq/q \;,\; v = vq/q$$

- Linear numerator and denominator.
- If g=h=0 then you get affine as a special case.

- Allows a square to be distorted into any quadrilateral.
- 8 degrees of freedom.
- Inverse is of same form (is also projective)

Projective mappings

- Every point correspondence can have two equations:
  - $x(XA_{31} + YA_{32} + A_{33}) = (XA_{11} + YA_{12} + A_{13})$
  - $y(XA_{31} + YA_{32} + A_{33}) = (XA_{21} + YA_{22} + A_{23})$

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

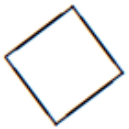
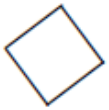
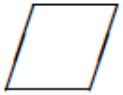
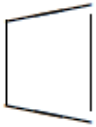
- After rearrangement, a pair of quadrilateral:

$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -x_3Y_3 & -x_3 \\ 0 & 0 & 0 & X_3 & Y_3 & 1 & -y_3X_3 & -y_3Y_3 & -y_3 \\ X_4 & Y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -x_4Y_4 & -x_4 \\ 0 & 0 & 0 & X_4 & Y_4 & 1 & -y_4X_4 & -y_4Y_4 & -y_4 \end{bmatrix} * \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{31} \\ A_{32} \\ A_{33} \end{bmatrix} = 0$$

min E = $|Ux|^2$, subject to $|x|=1$

The x that minimize E is eigenvector e_1 of U^TU

Properties of image transformation

Name	Matrix	Number of d.o.f.	Preserves	Icon
Translation	$[\, \boldsymbol{I} \,|\, \boldsymbol{t} \,]_{2\times 3}$	2	Orientation $+\cdots$	
Rigid (Euclidean)	$[\, \boldsymbol{R} \,|\, \boldsymbol{t} \,]_{2\times 3}$	3	Lengths $+\cdots$	
Similarity	$[\, s\boldsymbol{R} \,|\, \boldsymbol{t} \,]_{2\times 3}$	4	Angles $+\cdots$	
Affine	$[\, \boldsymbol{A} \,]_{2\times 3}$	6	Parallelism $+\cdots$	
Projective	$[\, \tilde{\boldsymbol{H}} \,]_{3\times 3}$	8	Straight lines	

Performing an image warp

- Source scanning:

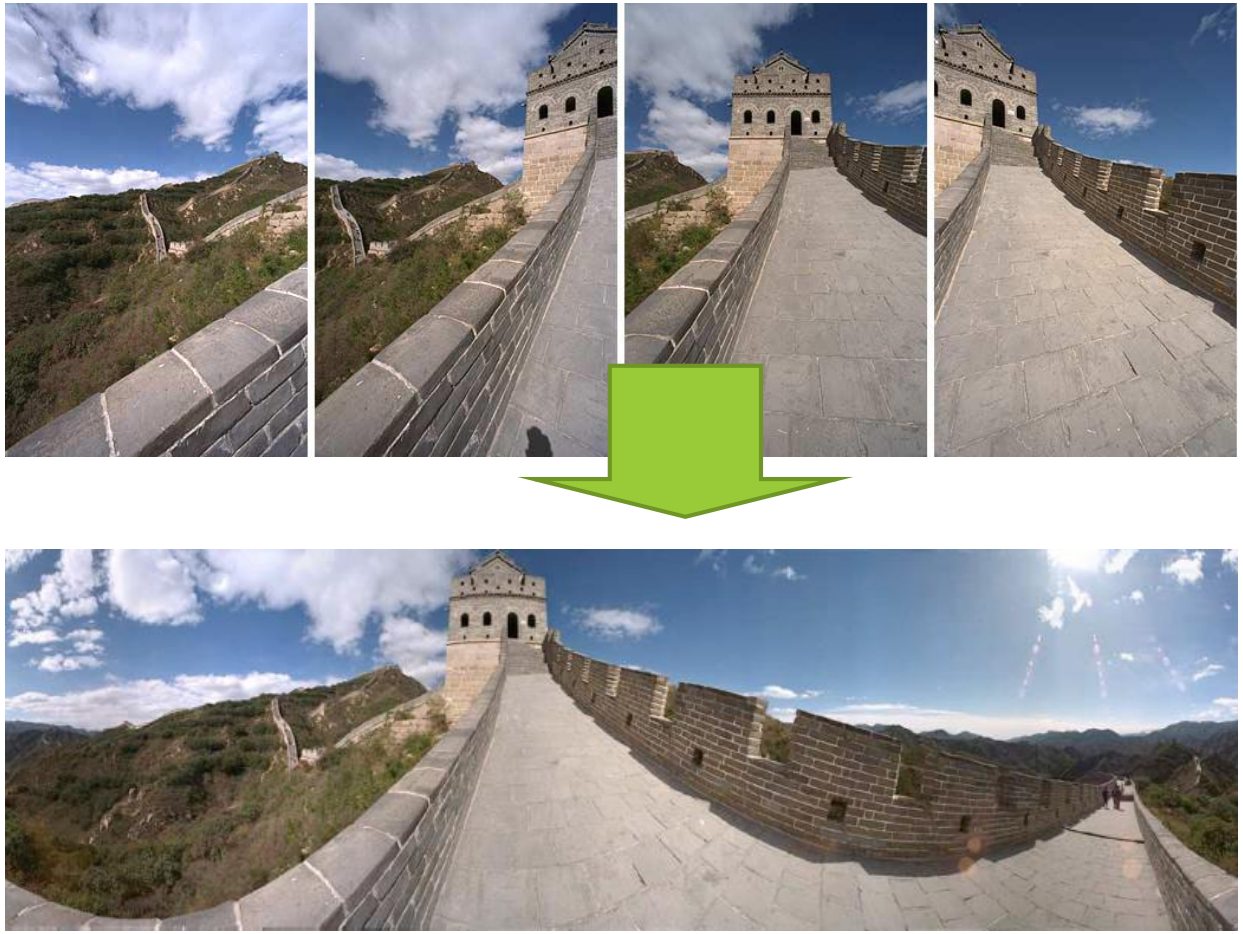
```
for v = vmin to vmax
    for u = umin to umax
        x = x(u,v)
        y = y(u,v)
        copy pixel at source[u,v] to dest[x,y]
```

- Destination scanning:

```
for y = ymin to ymax
    for x = xmin to xmax
        u = u(x,y)
        v = v(x,y)
        copy pixel at source[u,v] to dest[x,y]
```

Is there any problem?

Acquiring cylindrical projections



S.E. Chen, “QuickTime VR”, Proc. SIGGRAPH’95

Acquiring cylindrical projections

- Using a regular camera with a panning tripod.
- Camera model and rotations can be inferred from planar reference images. (without camera calibration)
- Combine planar images and then compute the projection onto a cylinder.

Panoramic images

Perspective projection on view plane

Automatic Panoramic Image Stitching using Invariant Features

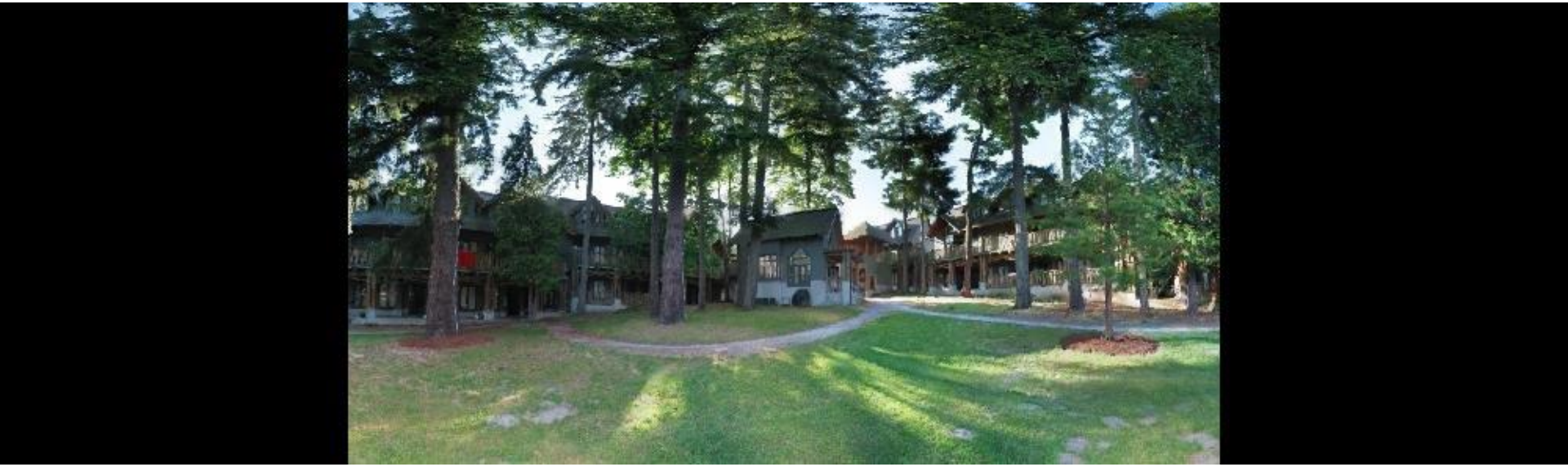
Proposed by M. Brown and D.G. Lowe,
In *Proc., ICCV 2003* and *IJCV 2007*

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 $\not\Rightarrow$ matching images

Why “Recognising Panoramas”?

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

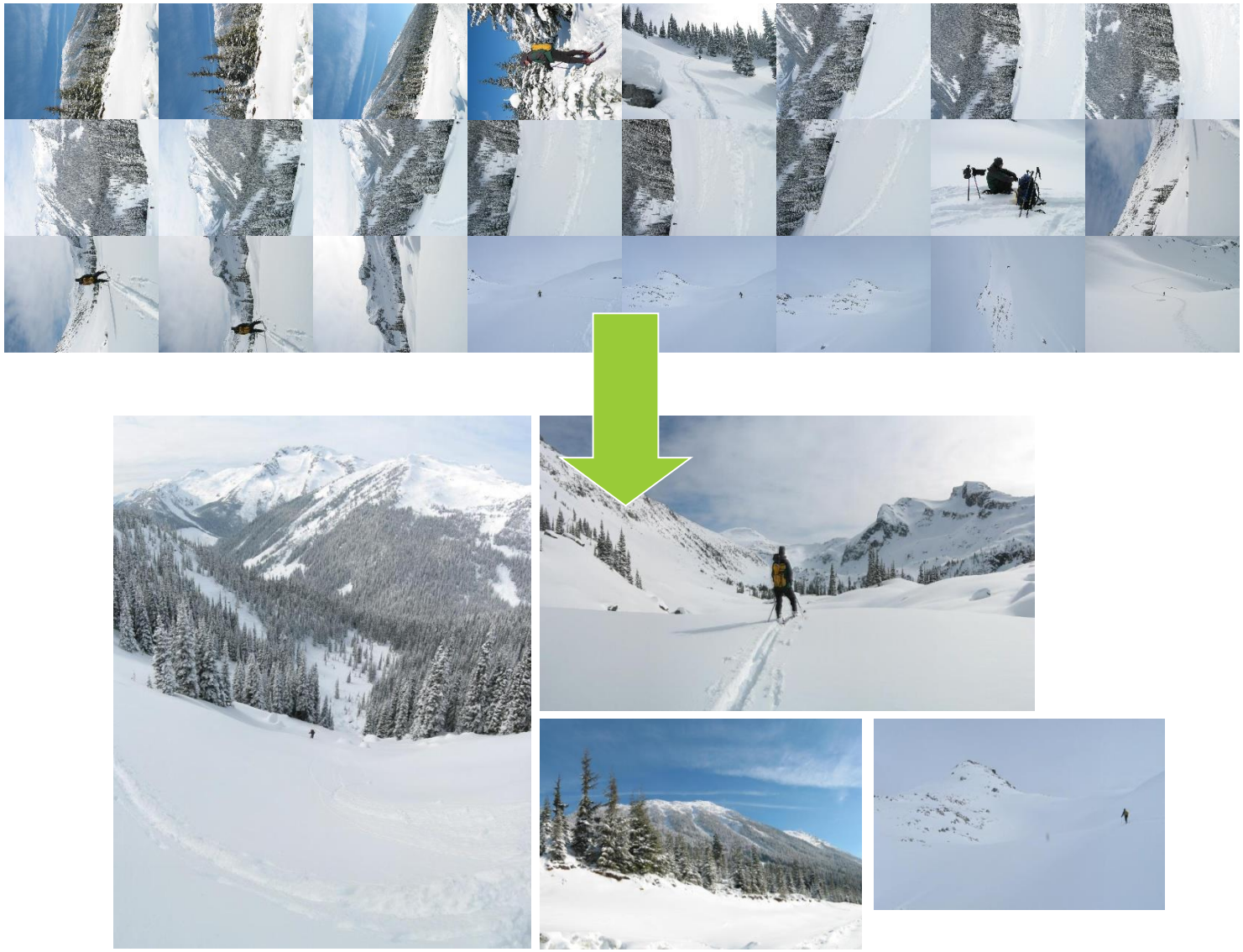
- 2D Rotations (θ, ϕ)
  - Ordering $\not\Rightarrow$ matching images

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

Overview

- Feature Matching
  - SIFT Features
  - Nearest Neighbor Matching
- Image Matching
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Invariant features

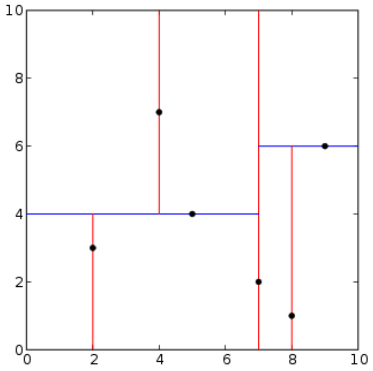
- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

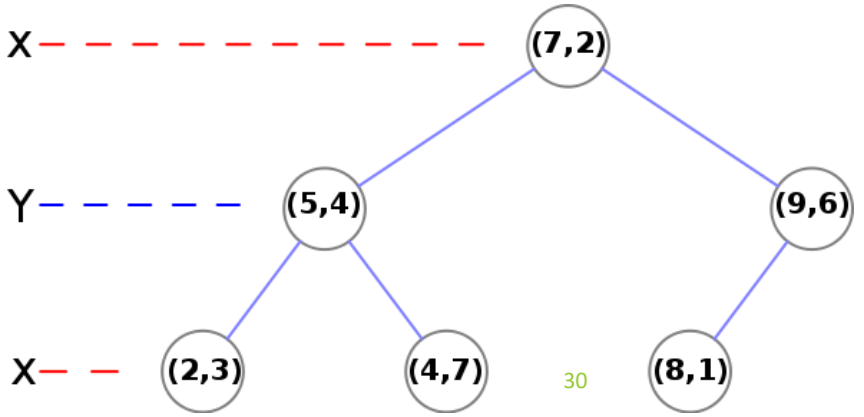
SIFT features

- 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

Nearest neighbor matching

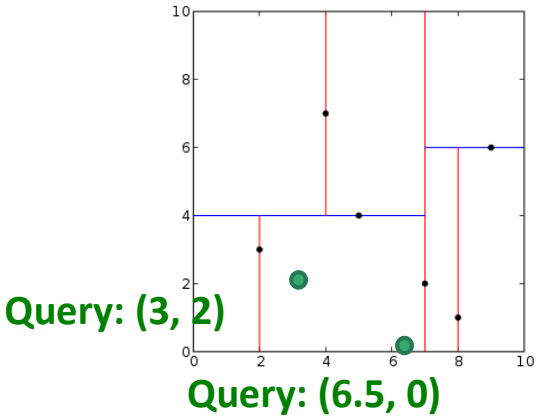
- Find k-NN for each feature
  - k ≈ number of overlapping images (in this work, k = 4)
- Use k-d tree or other searching tech. for speed-up
  - k-d tree recursively bi-partitions data at mean in the dimension of maximum variance.
  - Tree building complexity is less than O(knlogn).

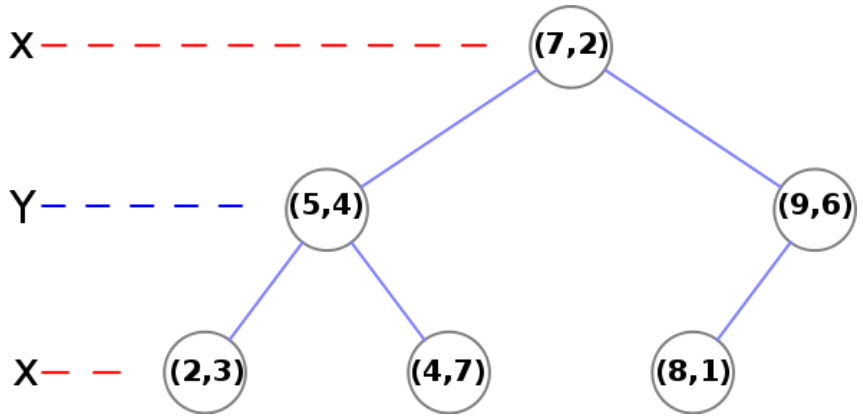




Figures from: http://en.wikipedia.org/wiki/Kd-tree

k-d tree





- In a resursive style
- Top-down search to a leaf node.
- Backward check whether there are “possible” NN through the distance to the hyperplane of the parents’ nodes.

- For a large K(dimension), approximate nearest neighbors are popularly used.

Overview

- Feature Matching
- Image Matching
  - RANSAC for Homography
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Matching with features

- For each keypoint, we have multiple candidates.

- How to correctly recognize the corresponding one?

RANSAC for point matching

- If there're multiple candidates, "*RANdom SAmple Consensus*" is an approach to solve the ambiguity.
- The basic idea is to sample a set of point correspondences.
- Use the sample to estimate a motion model (mapping between 2 views)
  - E.g. translation, fundamental matrix, etc.
- See whether the remaining points can provide support for this solution.
- Copes with a large proportion of outliers.

RANSAC example

- E.g. fitting a straight line

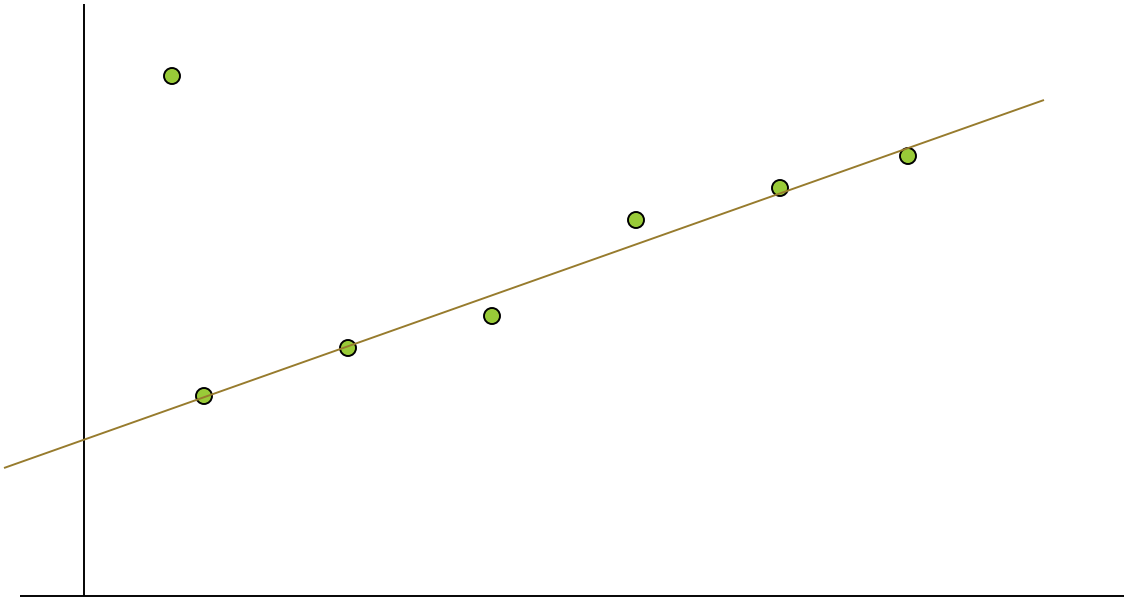
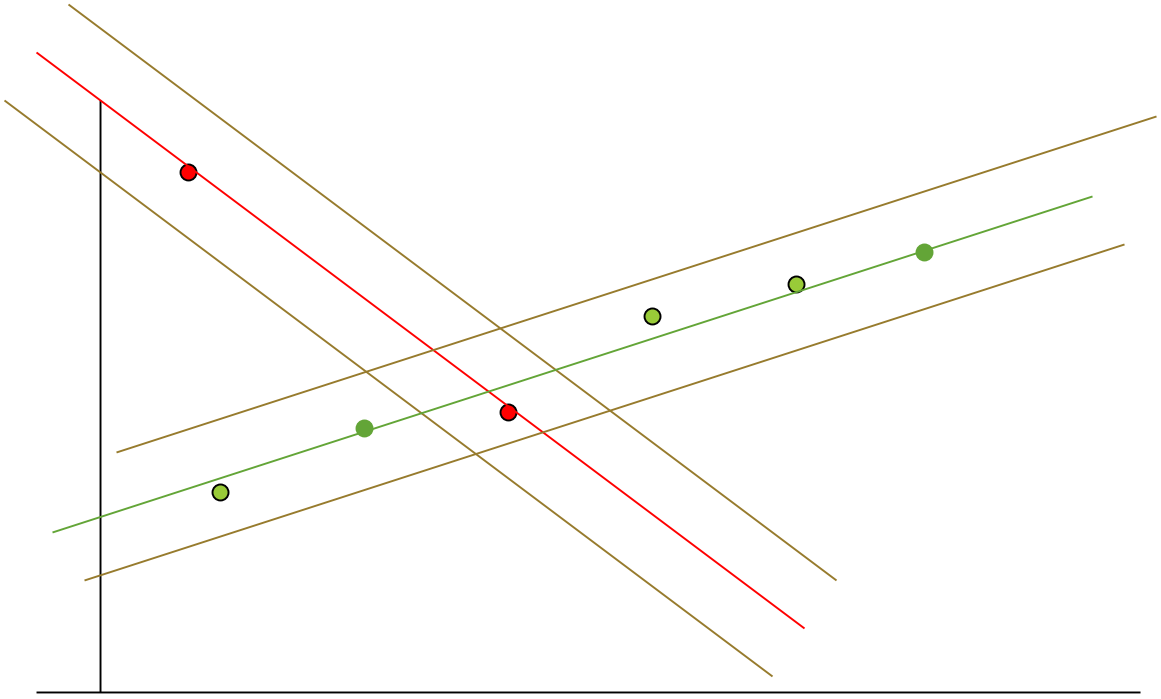


Figure from Prof. J. Rehg, Computer Vision, Georgia Inst. of Tech.

Main idea

- Select 2 points at random
- Fit a line
- “Support” = number of inliers
- Line with most inliers wins

Why will this work ?



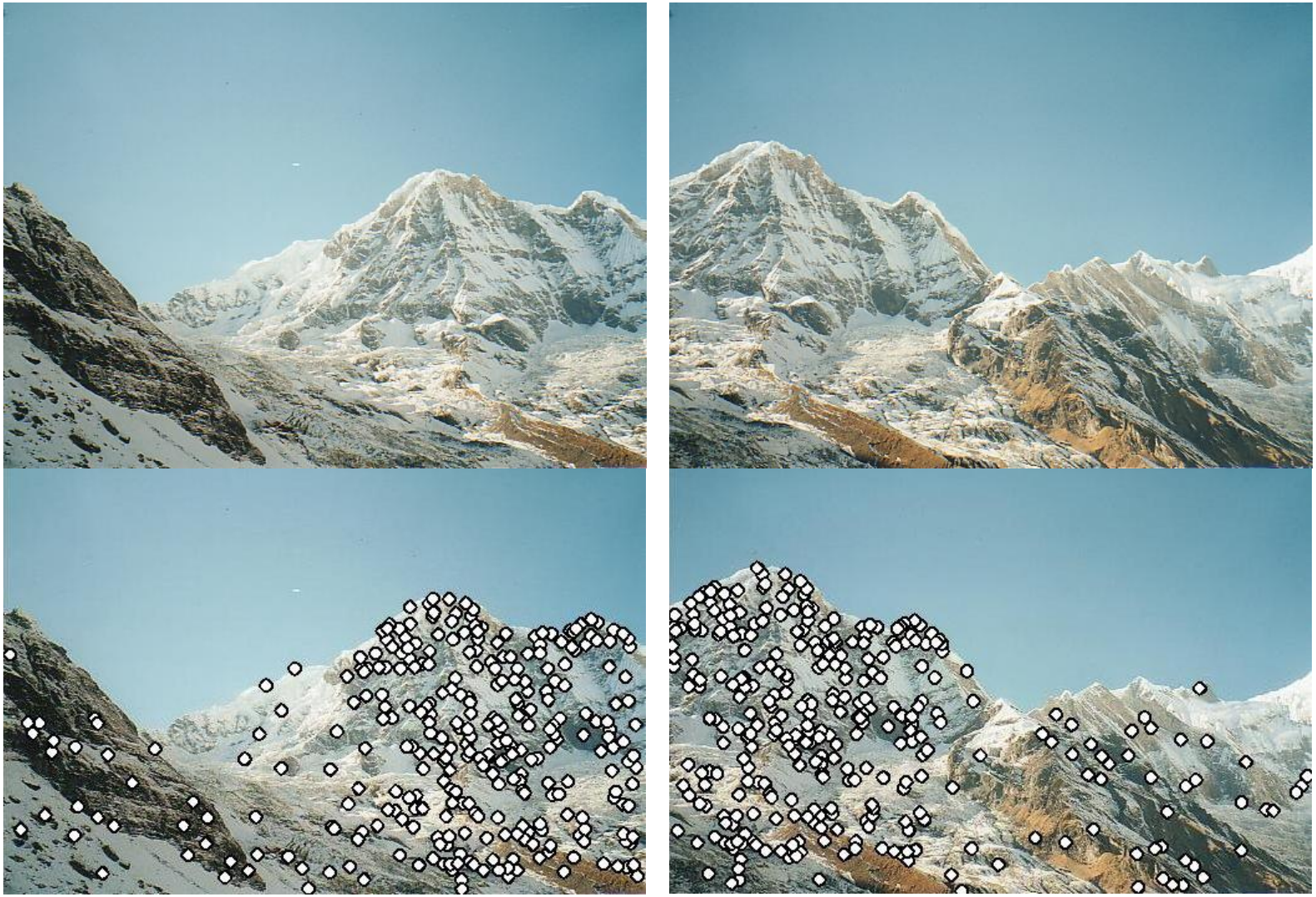
The best line has the most support
More support -> Better fit

Figure from Prof. J. Rehg, Computer Vision, Georgia Inst. of Tech.

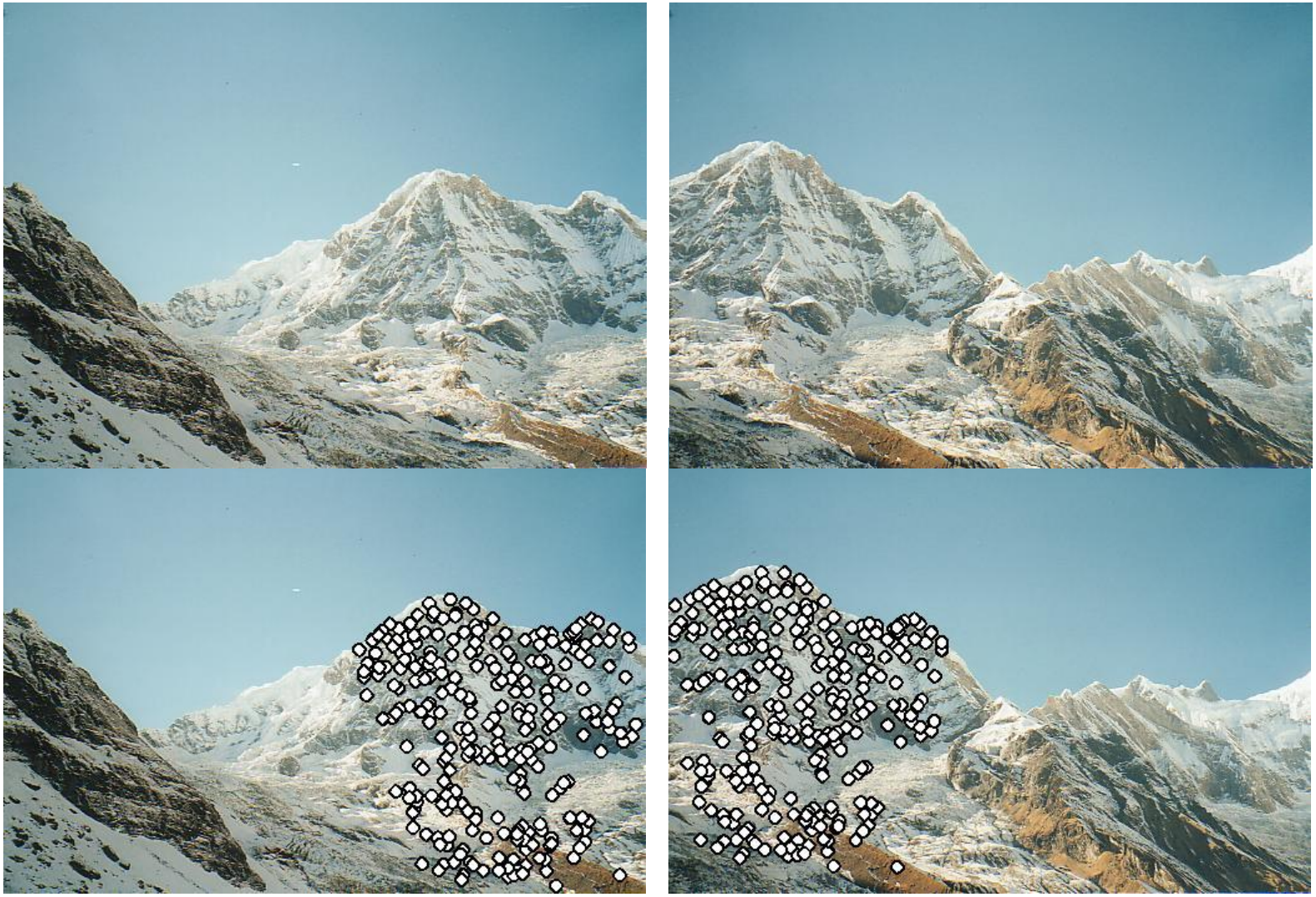
RANSAC

- Objective:
  - Robust fit of a model to data S
- Algorithm
  - Randomly select s points
  - Instantiate a model
  - Get consensus set S_i
  - If $|S_i|>T$, terminate and return model
  - Repeat for N trials, return model with max $|S_i|$

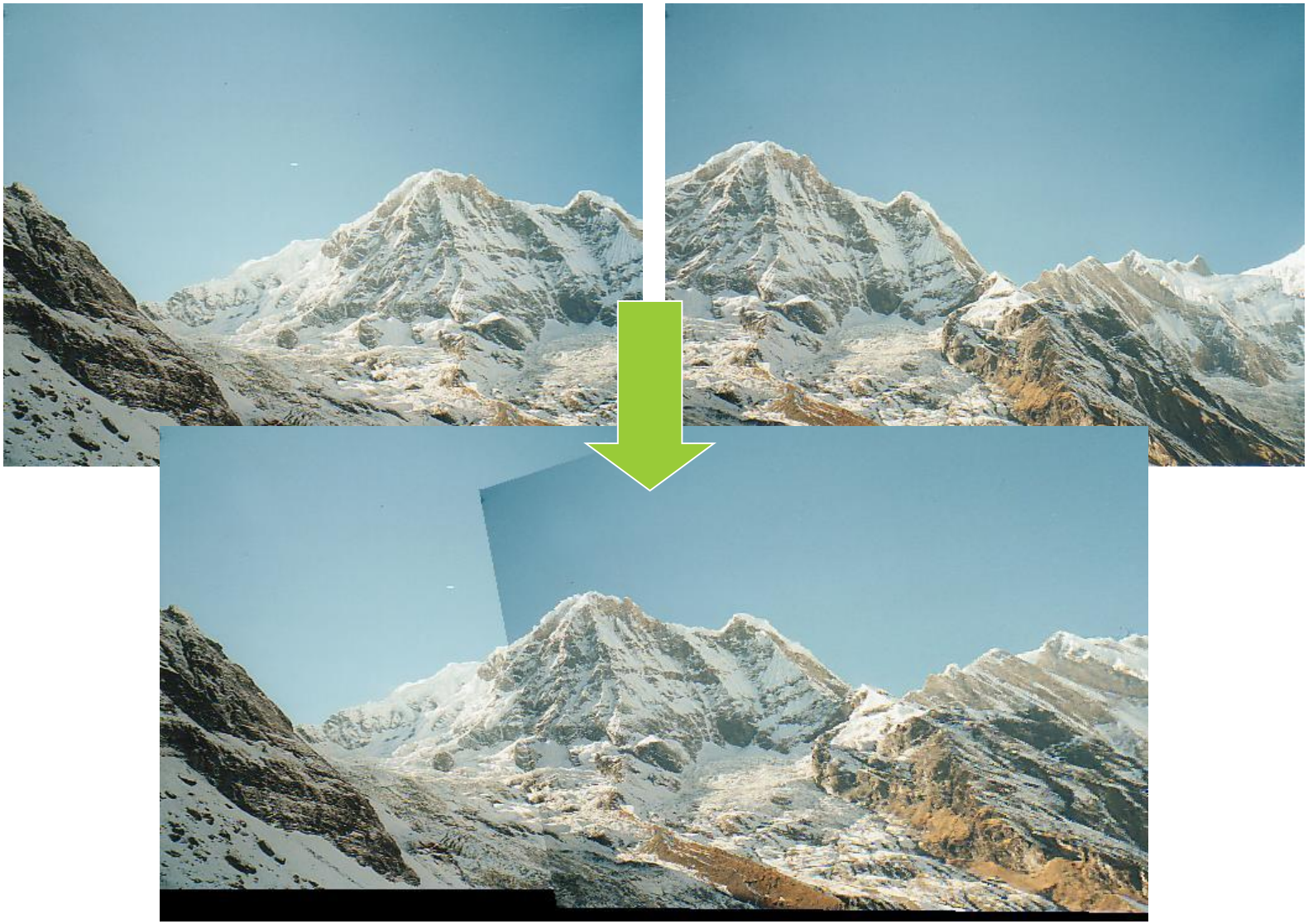
RANSAC for homography



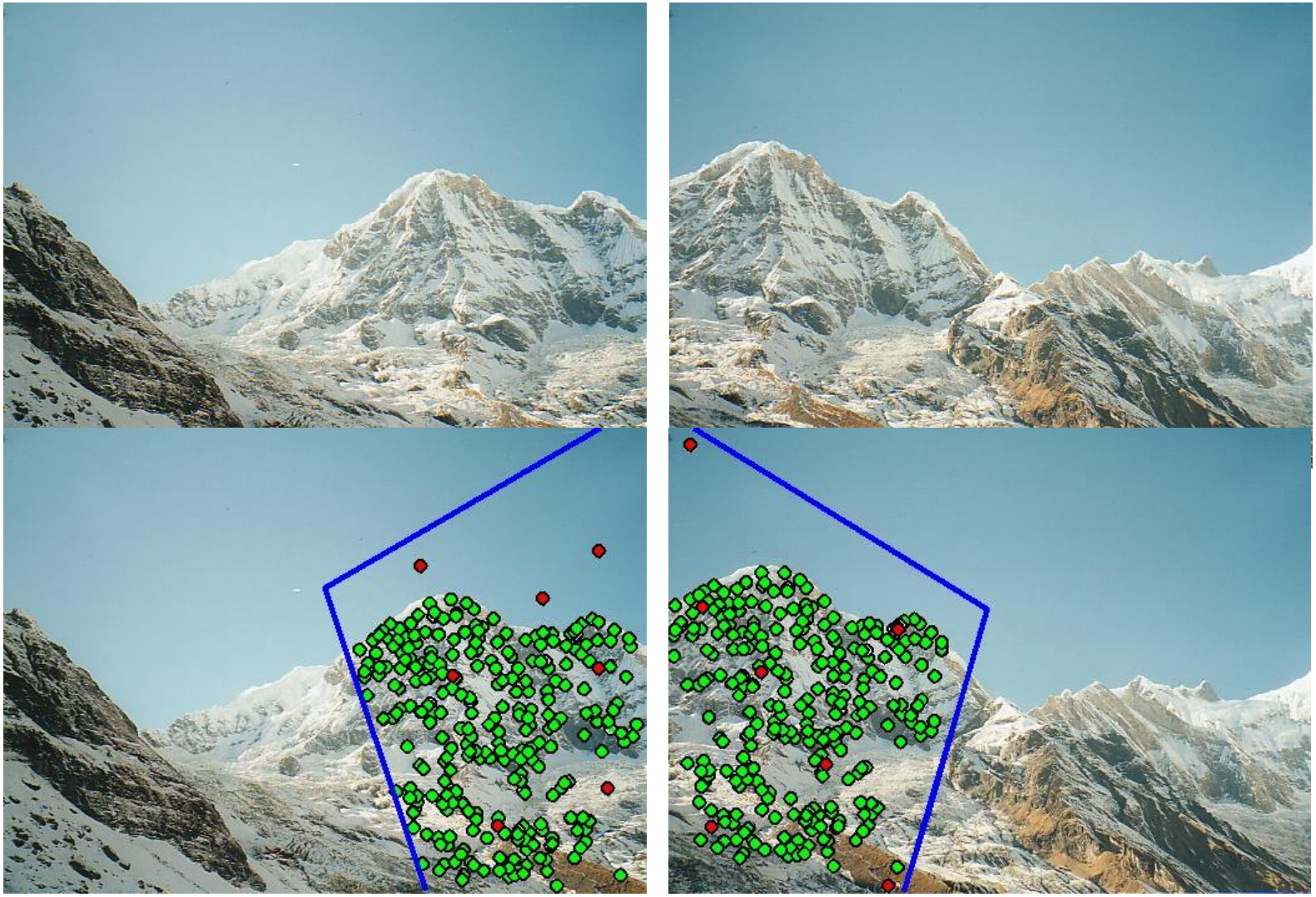
RANSAC for homography



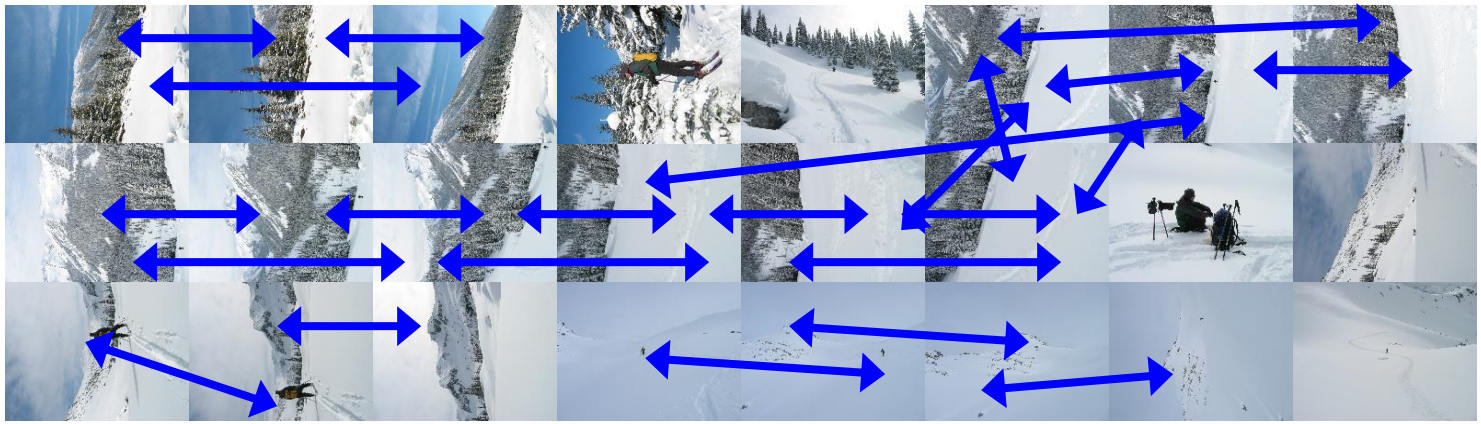
RANSAC for homography



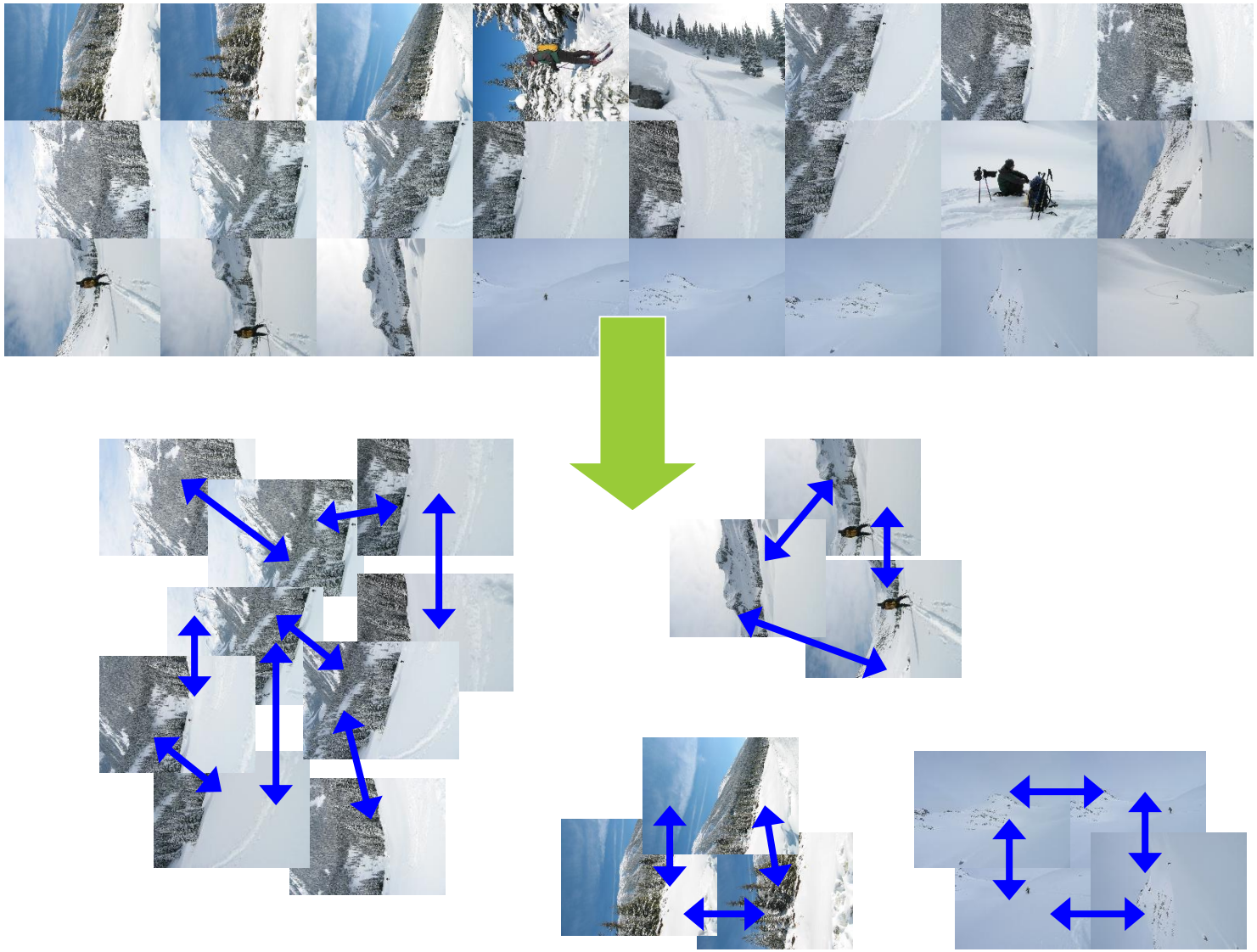
Probabilistic model for verification



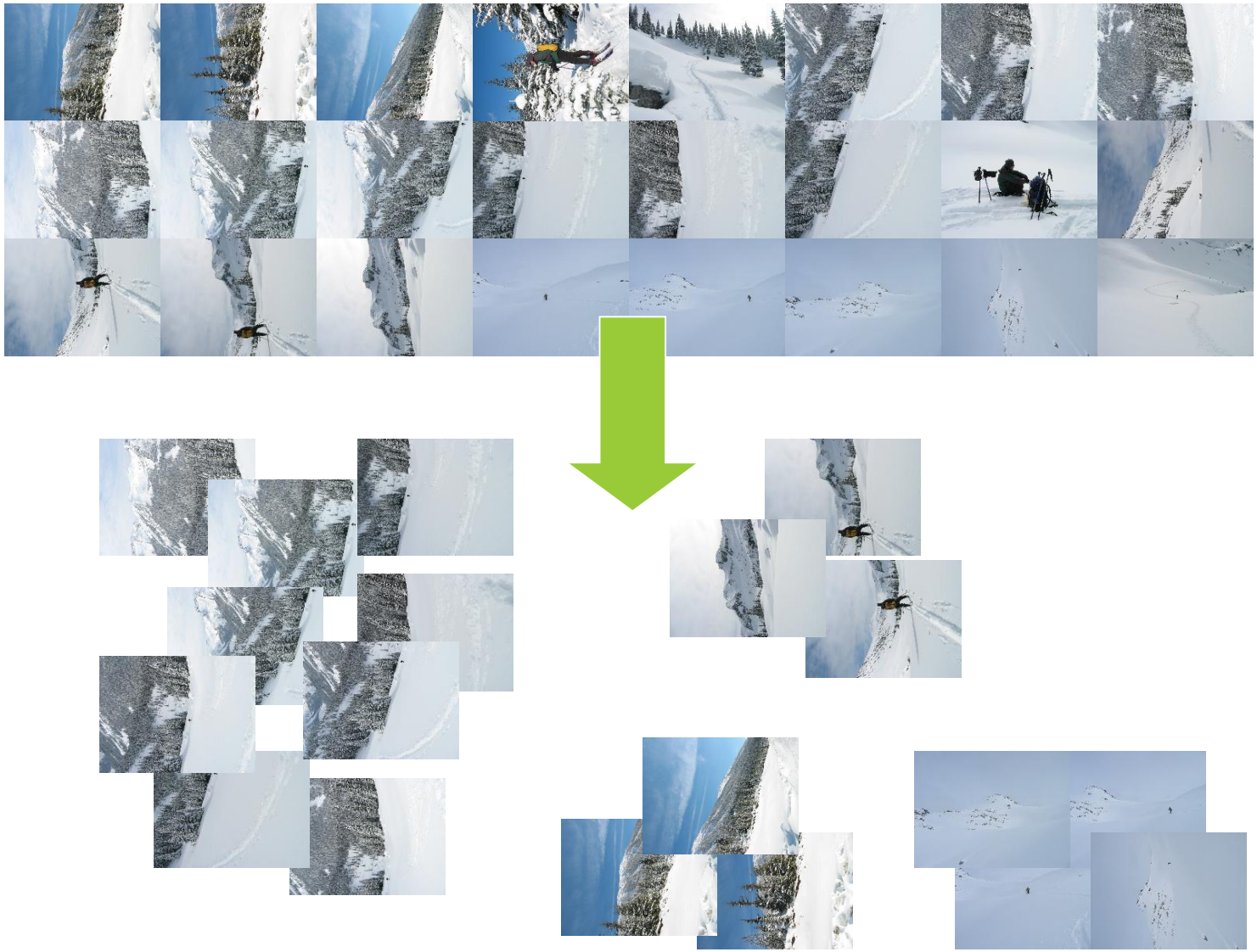
Finding the panoramas (match 6 images at most)



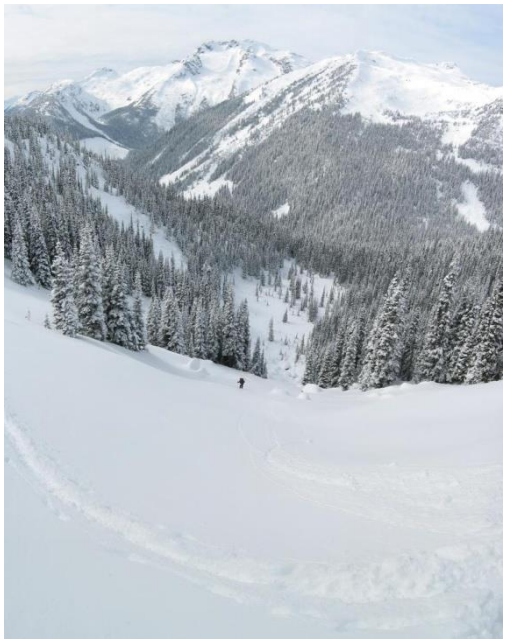
Finding the panoramas



Finding the panoramas



Finding the panoramas



Overview

- Feature Matching
- Image Matching
- Bundle Adjustment
  - Error function
- Multi-band Blending
- Results
- Conclusions

Error function

f: Require knowledge about camera and R, T

Let's skip the detail of this part !

- Sum of squared projection errors

$$e = \sum_{i=1}^{n} \sum_{j \epsilon \mathcal{I}(i)} \sum_{k \epsilon \mathcal{F}(i,j)} f(\mathbf{r}_{ij}^{k})^2$$

  - 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

- Robust error function

$$f(\mathbf{x}) = \begin{cases} |\mathbf{x}|, & \text{if } |\mathbf{x}| < x_{max} \\ x_{max}, & \text{if } |\mathbf{x}| \geq x_{max} \end{cases}$$

Bundle adjustment

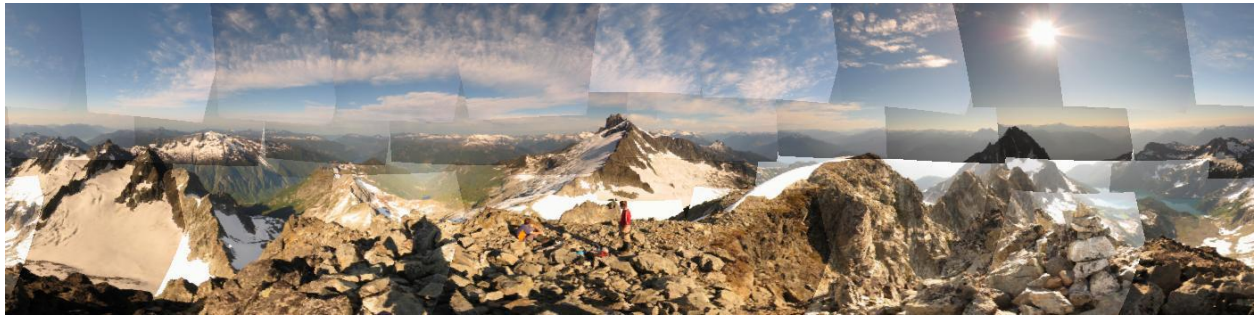
- New images initialised with rotation, focal length of best matching image.

Bundle adjustment

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

Gain compensation

N_{ij} : the number of pixels in image i that overlap in image j



- Find the optimize gains of g_i according to means of overlapping regions between image pair i and j

$$e = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} N_{ij}\left((g_i\bar{I}_{ij} - g_j\bar{I}_{ji})^2/\sigma_N^2 + (1-g_i)^2/\sigma_g^2\right)$$

Overview

- Feature Matching
- Image Matching
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Multi-band blending

- Burt & Adelson 1983
  - Blend frequency bands over range $\propto \lambda$

Linear blending

- View-dependent weight:

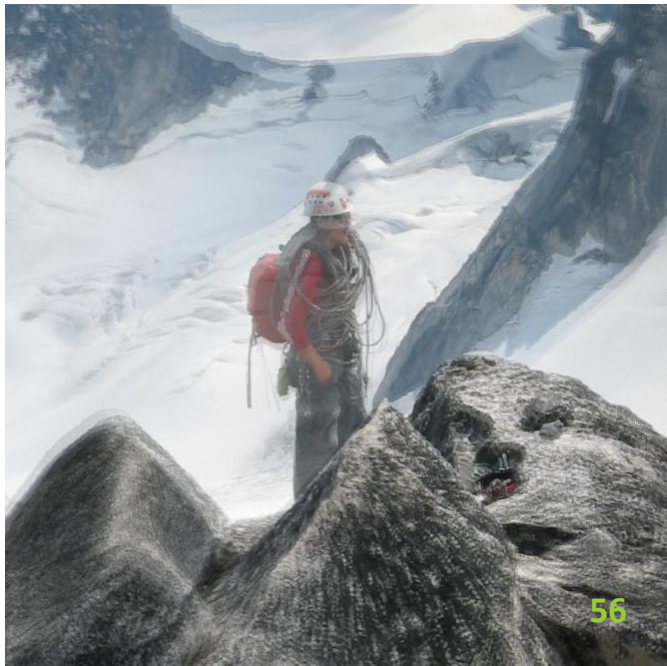
$$W(x,y) = w(x)w(y)$$

where *w(x)* varies linearly from 1 at the centre of the image to 0 at the edge

- Linear blending by the weights:

$$I^{linear}(\theta,\phi) = \frac{\sum_{i=1}^{n} I^i(\theta,\phi)W^i(\theta,\phi)}{\sum_{i=1}^{n} W^i(\theta,\phi)}$$

  - Cause blurring of high frequency detail if there are small registration errors



Multi-band blending

- Mainly use the most frontal pixels.

$$W^i_{max}(\theta,\phi) = \begin{cases} 1 & \text{if } W^i(\theta,\phi) = \arg\max_j W^j(\theta,\phi) \\ 0 & \text{otherwise} \end{cases}$$

- Propagate the weights to neighbors by Gaussian filter.

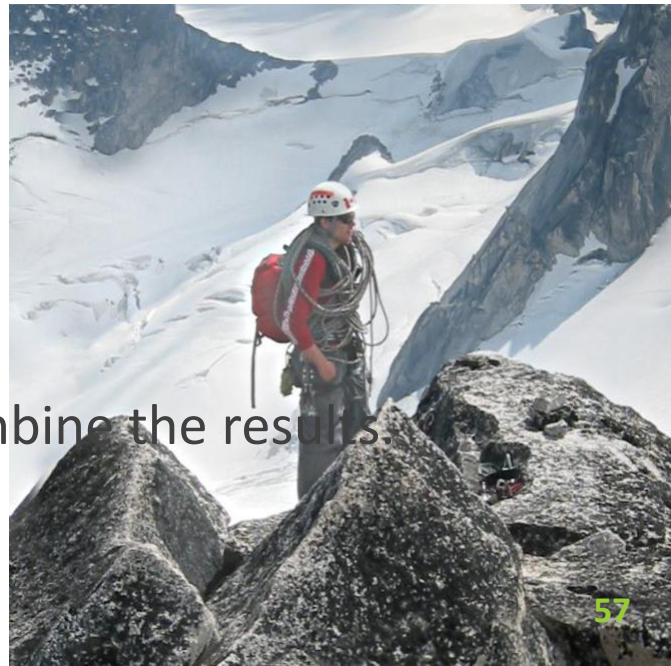
$$W^i_\sigma(\theta,\phi) = W^i_{max}(\theta,\phi) * g_\sigma(\theta,\phi)$$

- Divide the image into multiple bands.

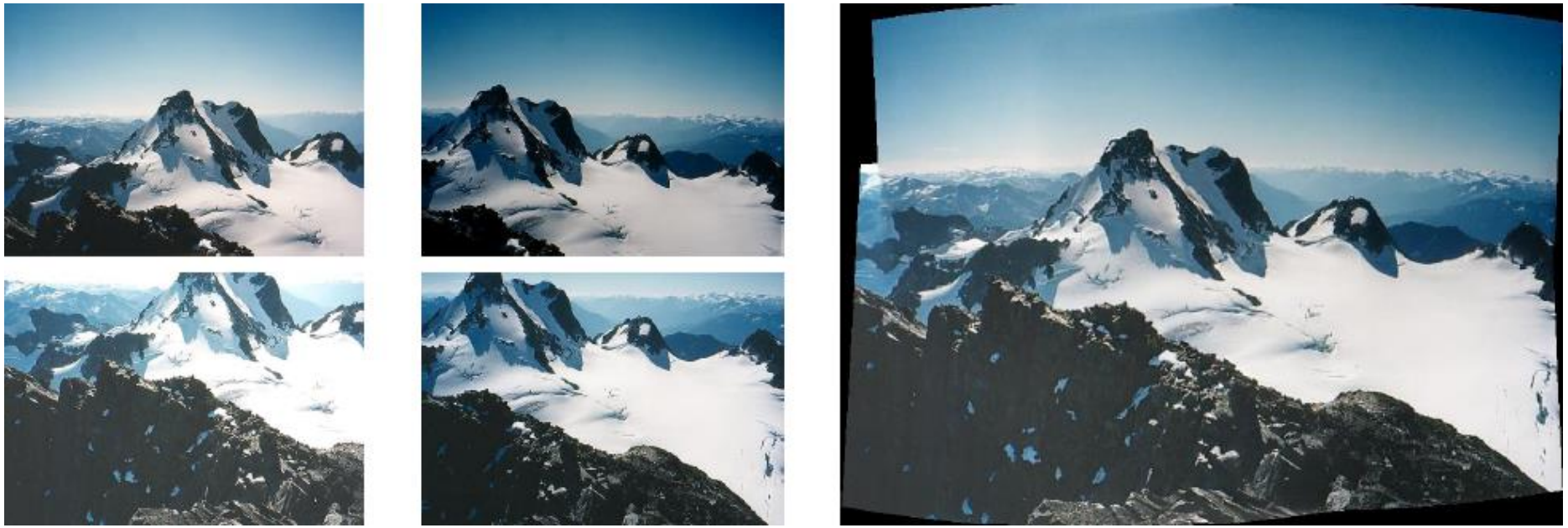
$$\begin{aligned} B^i_{(k+1)\sigma} &= I^i_{k\sigma} - I^i_{(k+1)\sigma} \\ I^i_{(k+1)\sigma} &= I^i_{k\sigma} * g_{\sigma'} \\ W^i_{(k+1)\sigma} &= W^i_{k\sigma} * g_{\sigma'} \end{aligned}$$

- Apply linear blending on each band, and combine the results.

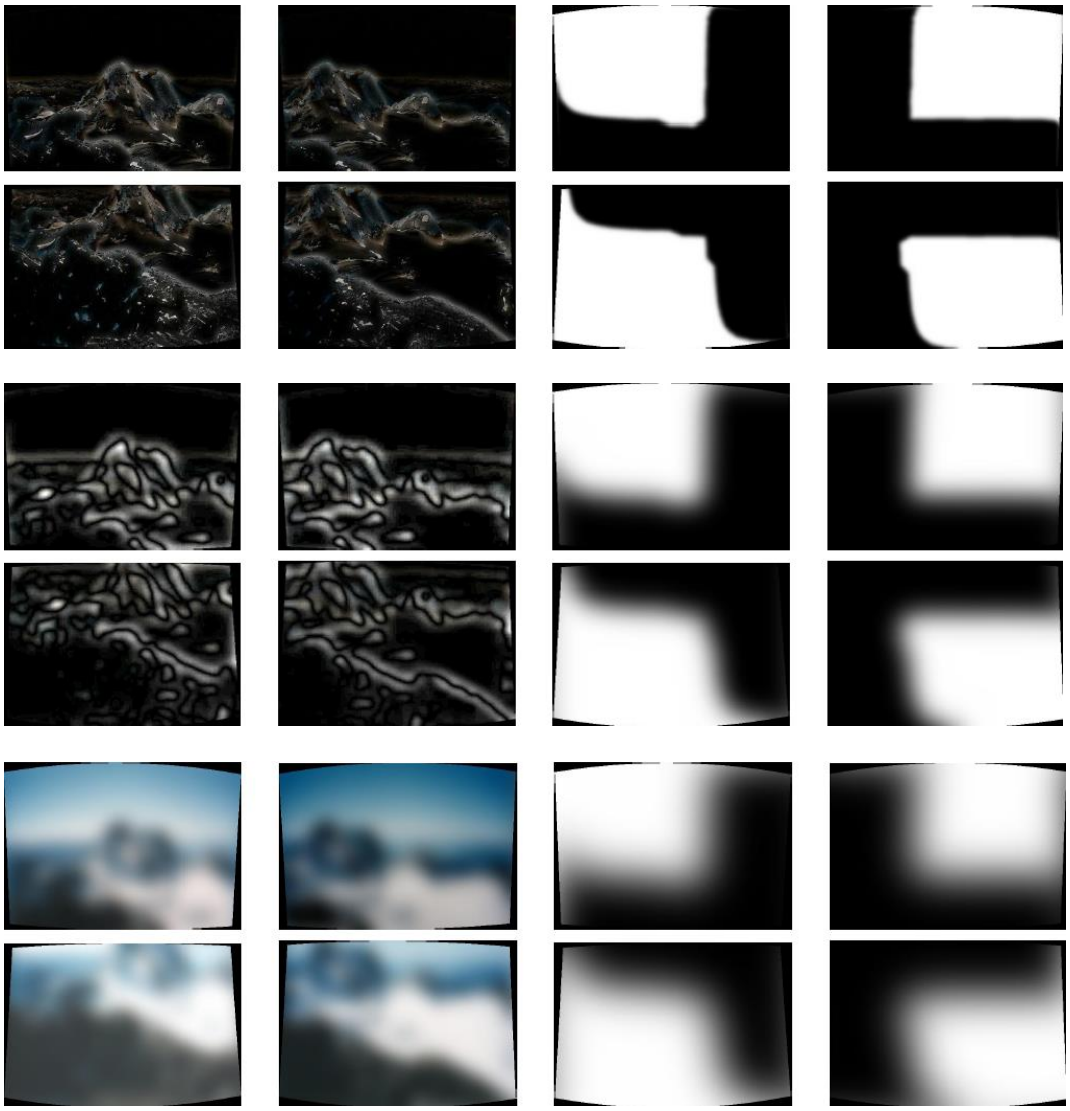
$$I^{multi}_{k\sigma}(\theta,\phi) = \frac{\sum_{i=1}^n B^i_{k\sigma}(\theta,\phi) W^i_{k\sigma}(\theta,\phi)}{\sum_{i=1}^n W^i_{k\sigma}(\theta,\phi)}$$



Multi-band blending



Multi-band blending



Band 1 scale 0 to σ

Band 2 scale σ to 2σ

Band 3 lower than 2σ

Results

Conclusions

- Fully automatic panoramas
  - A recognition problem…
- Invariant feature based method
  - SIFT features, RANSAC, Bundle Adjustment, Multi-band Blending
  - O(nlogn)
- Future work of this article
  - Advanced camera modelling
    - radial distortion, camera motion, scene motion, vignetting, exposure, high dynamic range, flash …
  - Full 3D case – recognising 3D objects/scenes in unordered datasets