Object Recognition with Invariant Features

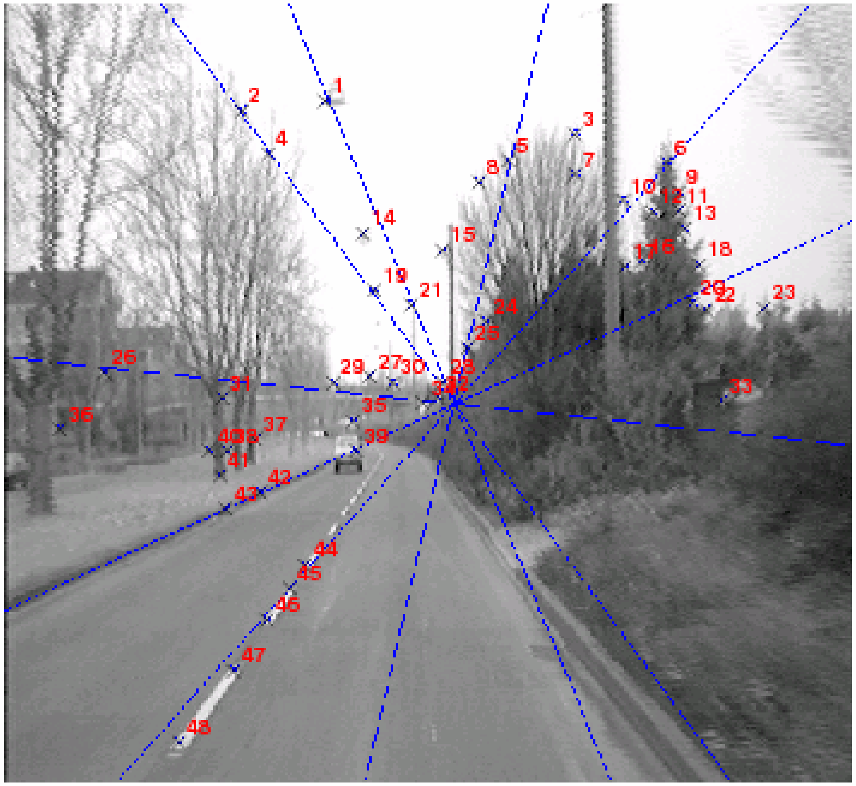
- **Definition:** Identify objects or scenes and determine their pose and model parameters

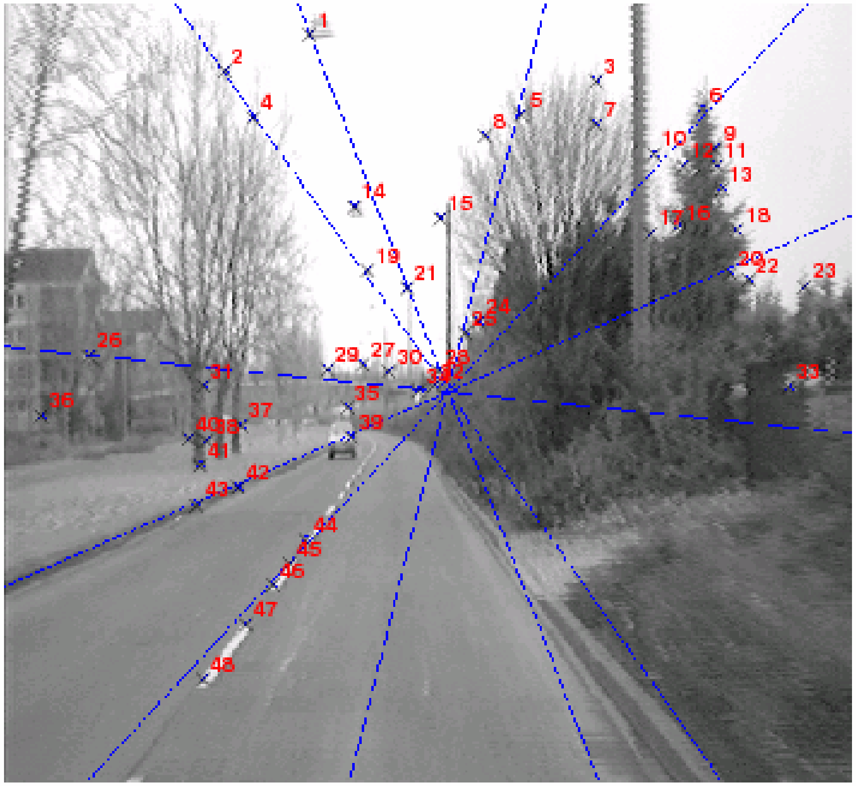
- **Applications**
  - Industrial automation and inspection
  - Mobile robots, toys, user interfaces
  - Location recognition
  - Digital camera panoramas
  - 3D scene modeling, augmented reality

Slides credit: David Lowe

Zhang, Deriche, Faugeras, Luong (95)

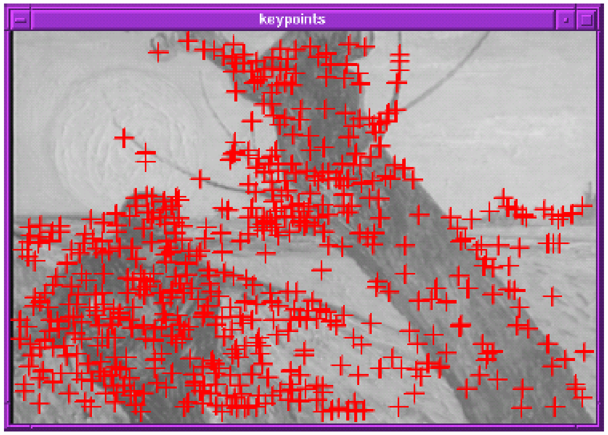
- Apply Harris corner detector
- Match points by correlating only at corner points
- Derive epipolar alignment using robust least-squares

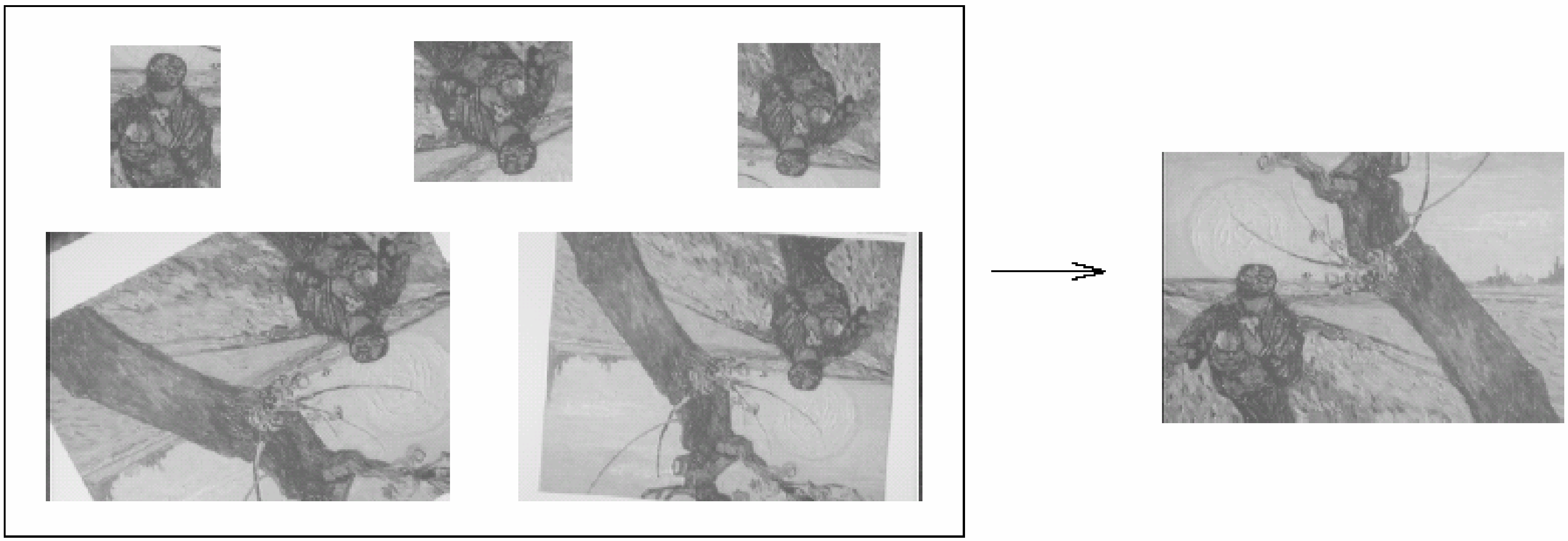




Cordelia Schmid & Roger Mohr (97)

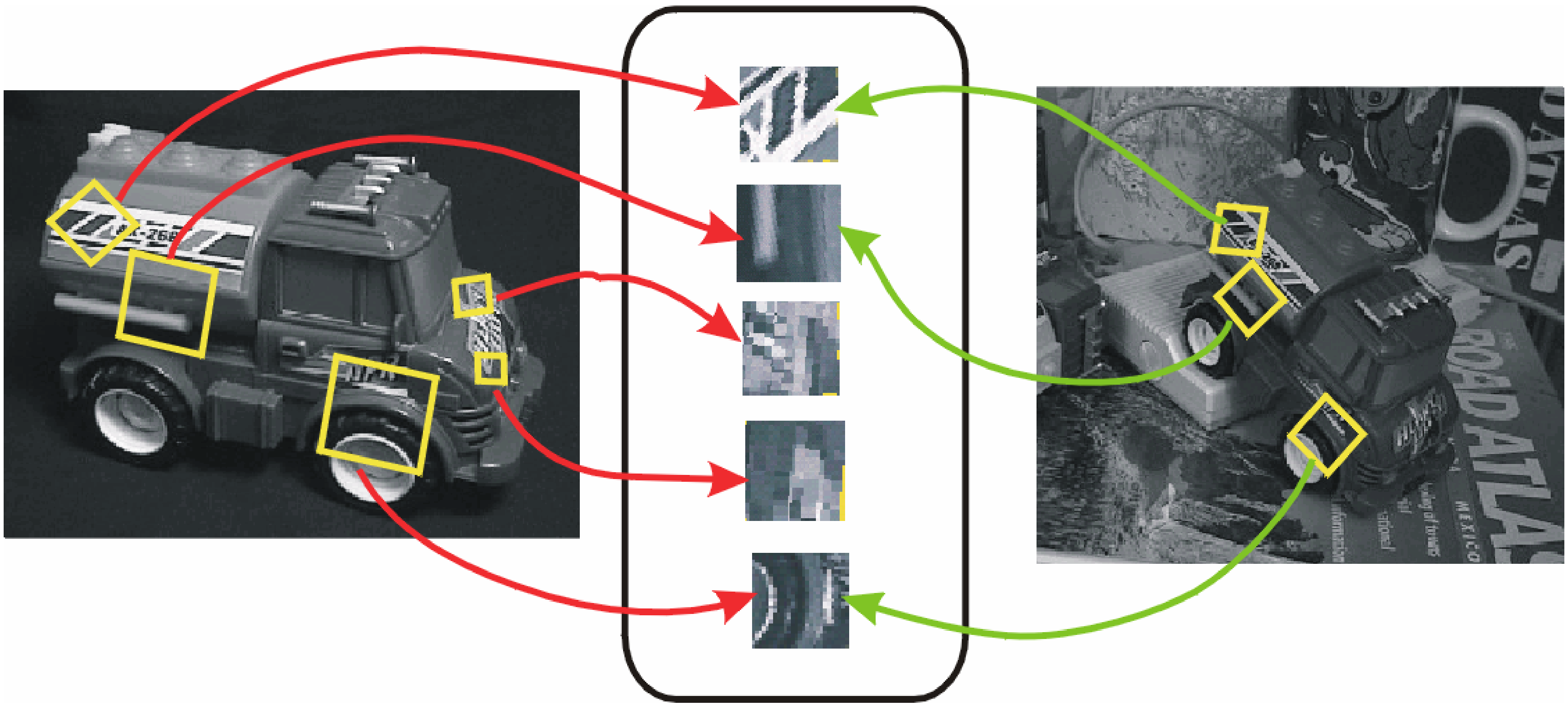
- Apply Harris corner detector
- Use rotational invariants at corner points
  - However, not scale invariant. Sensitive to viewpoint and illumination change.





Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



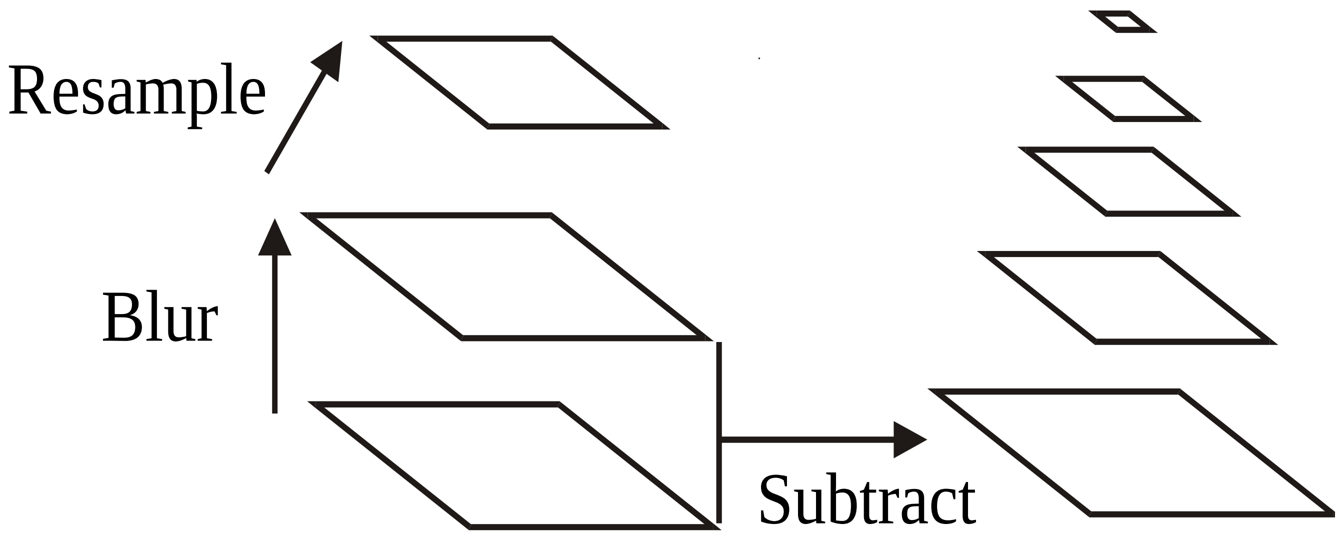
SIFT Features

Advantages of invariant local features

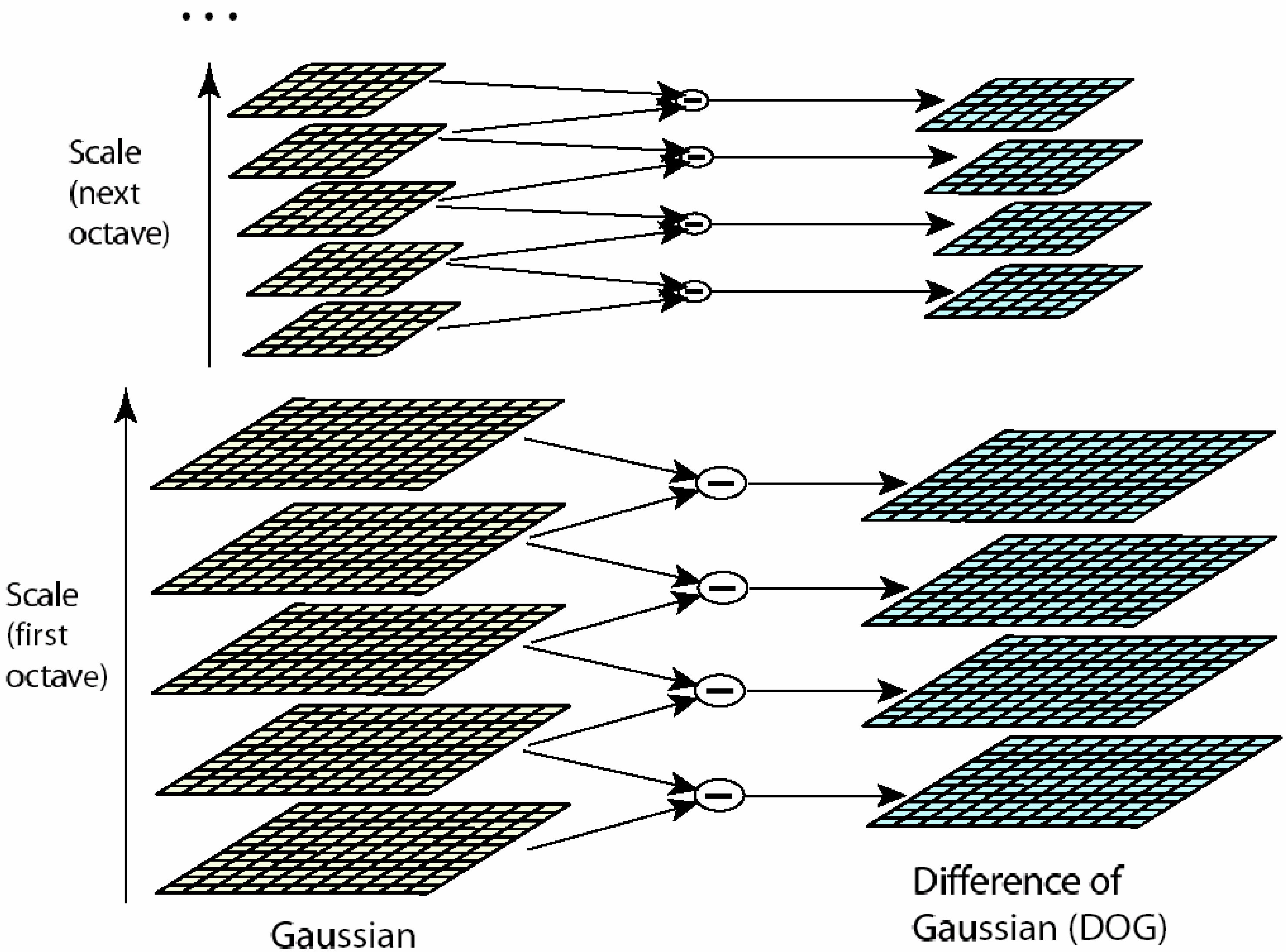
- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

Build Scale-Space Pyramid

- All scales must be examined to identify scale-invariant features
- An efficient function is to compute the Difference of Gaussian (DOG) pyramid (Burt & Adelson, 1983)

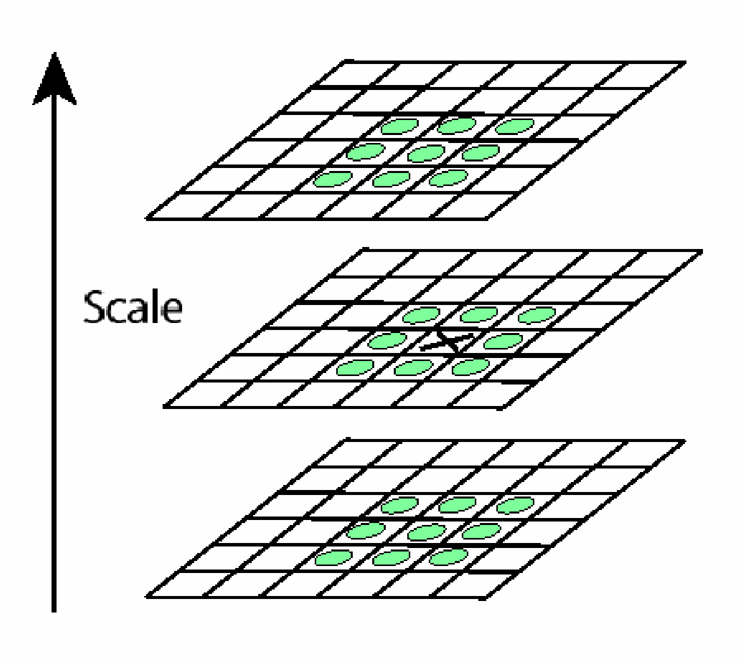


Scale space processed one octave at a time



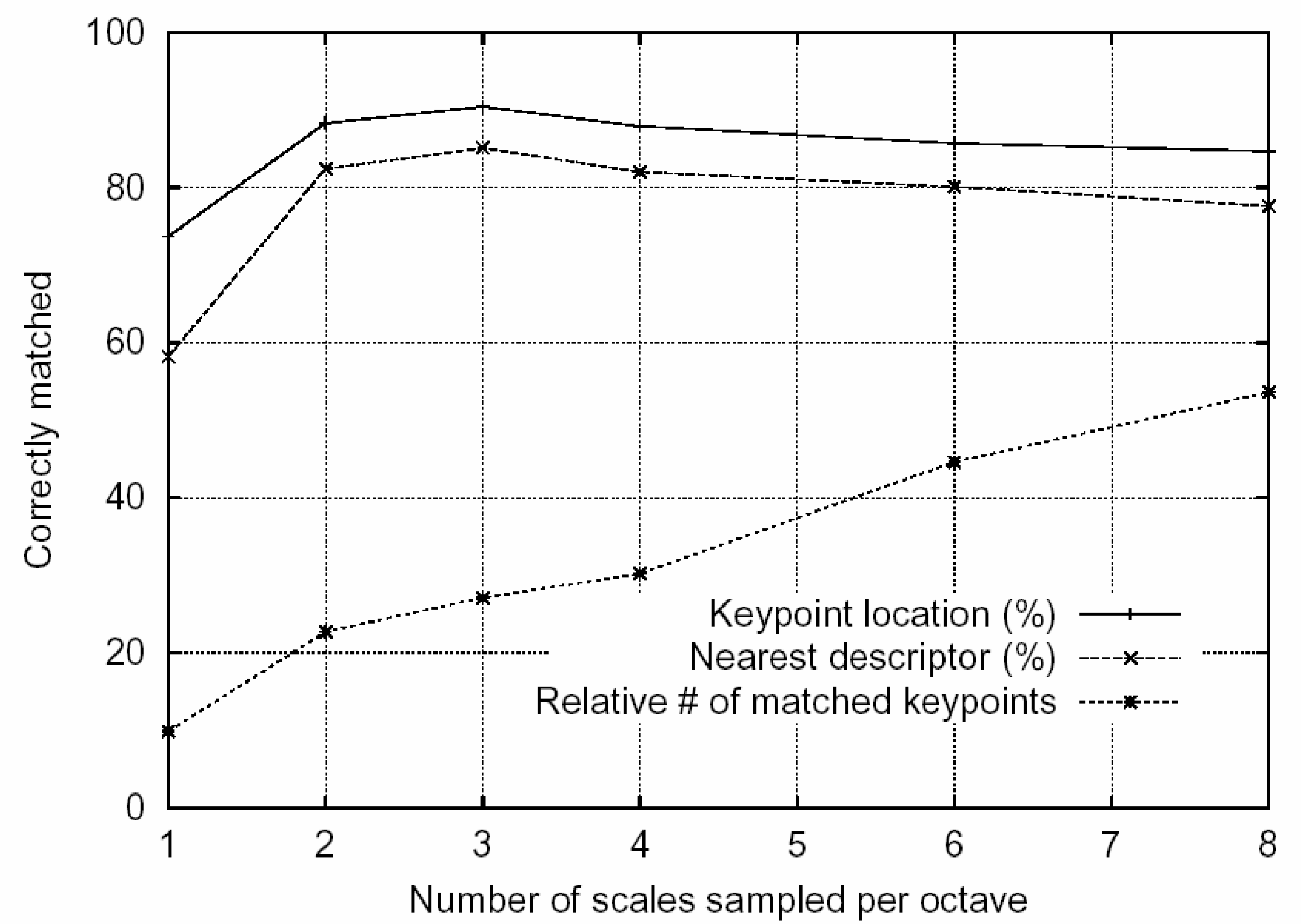
Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space



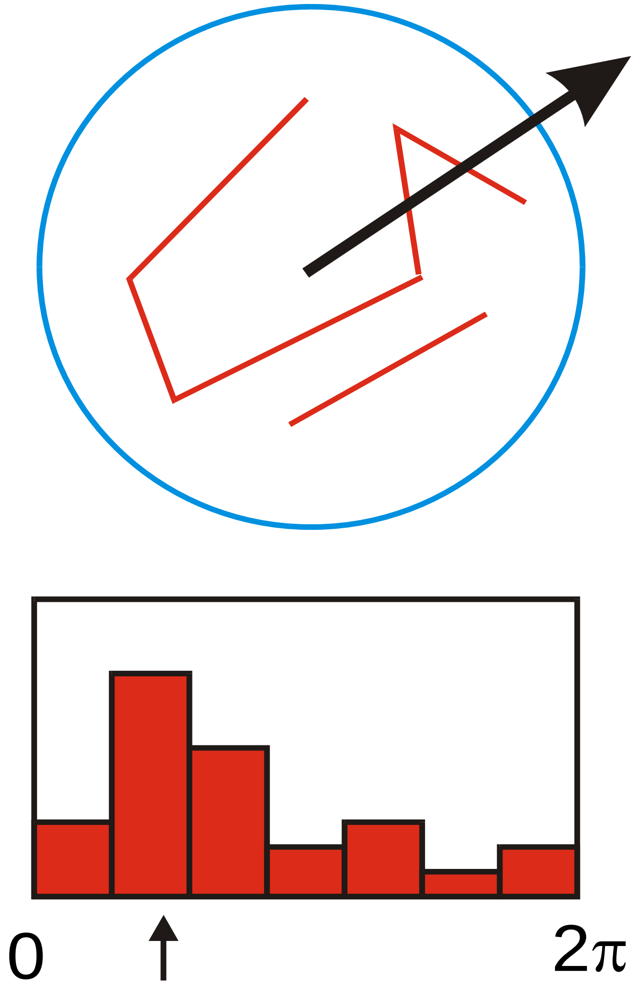
Sampling frequency for scale

More points are found as sampling frequency increases, but accuracy of matching decreases after 3 scales/octave



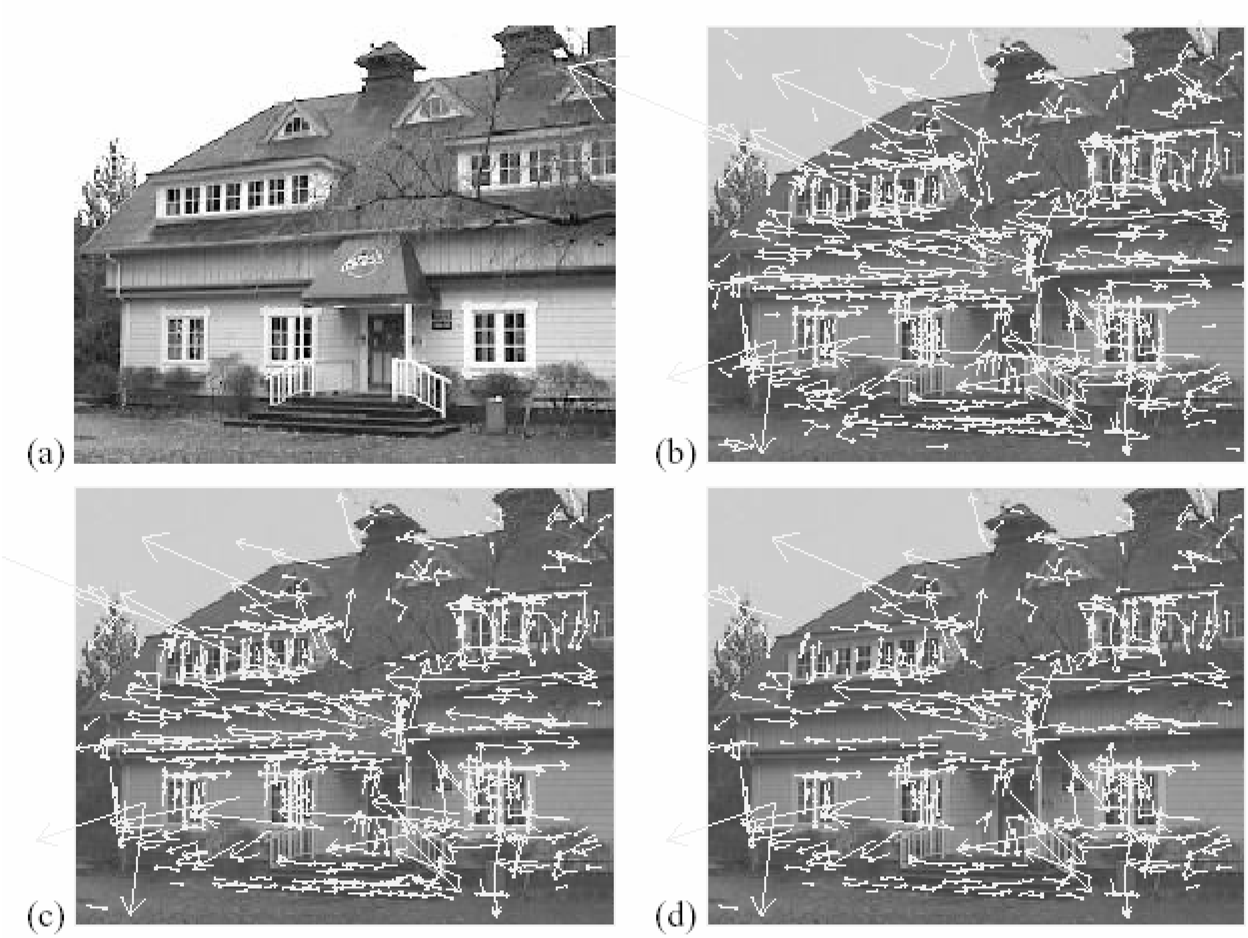
Select canonical orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)



Example of keypoint detection

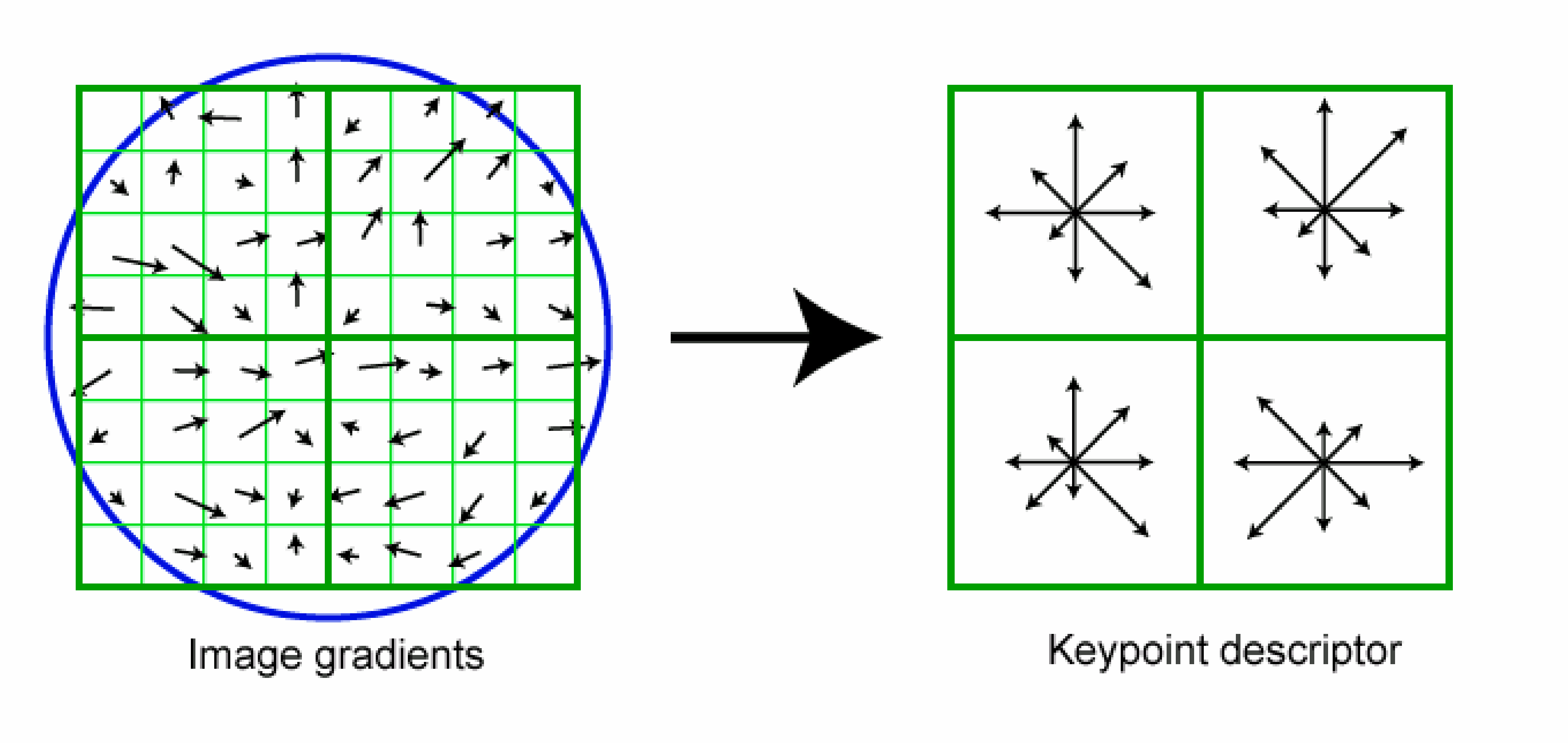
Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

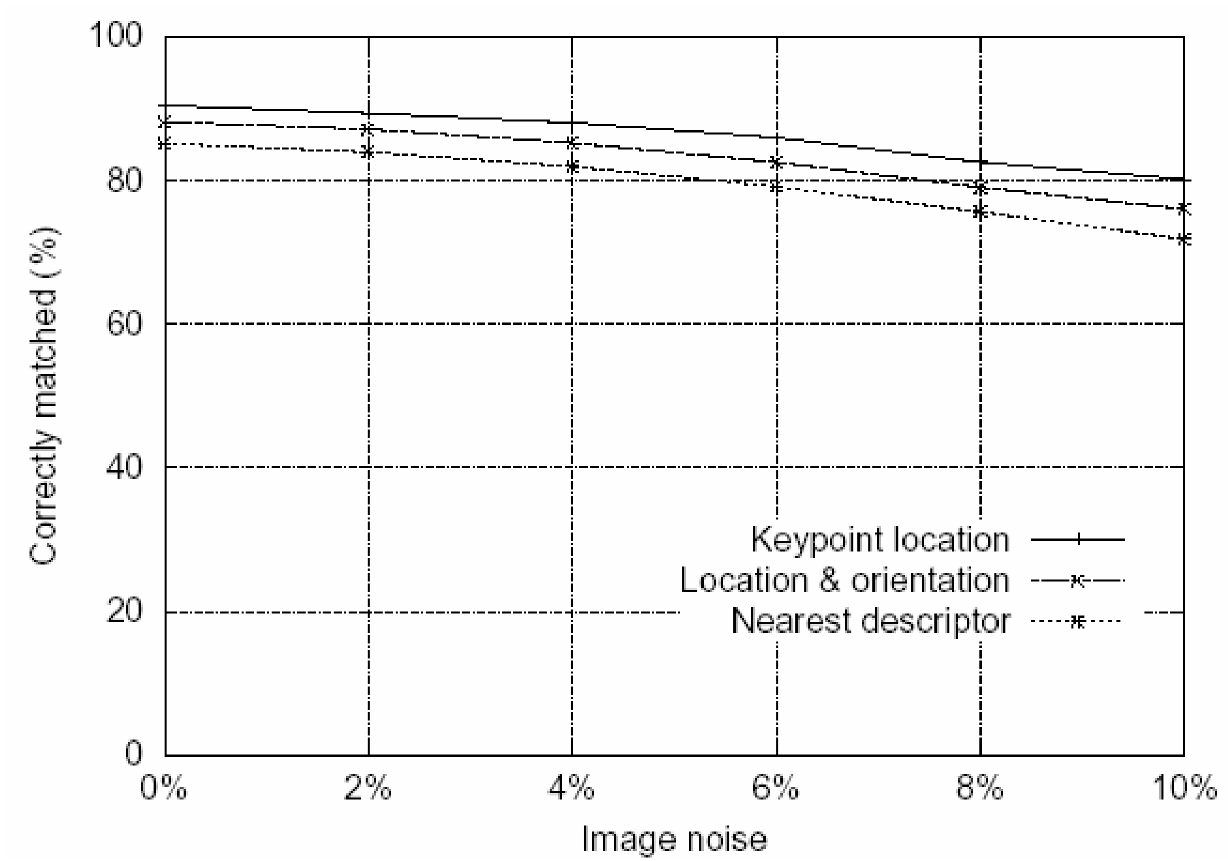
SIFT vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



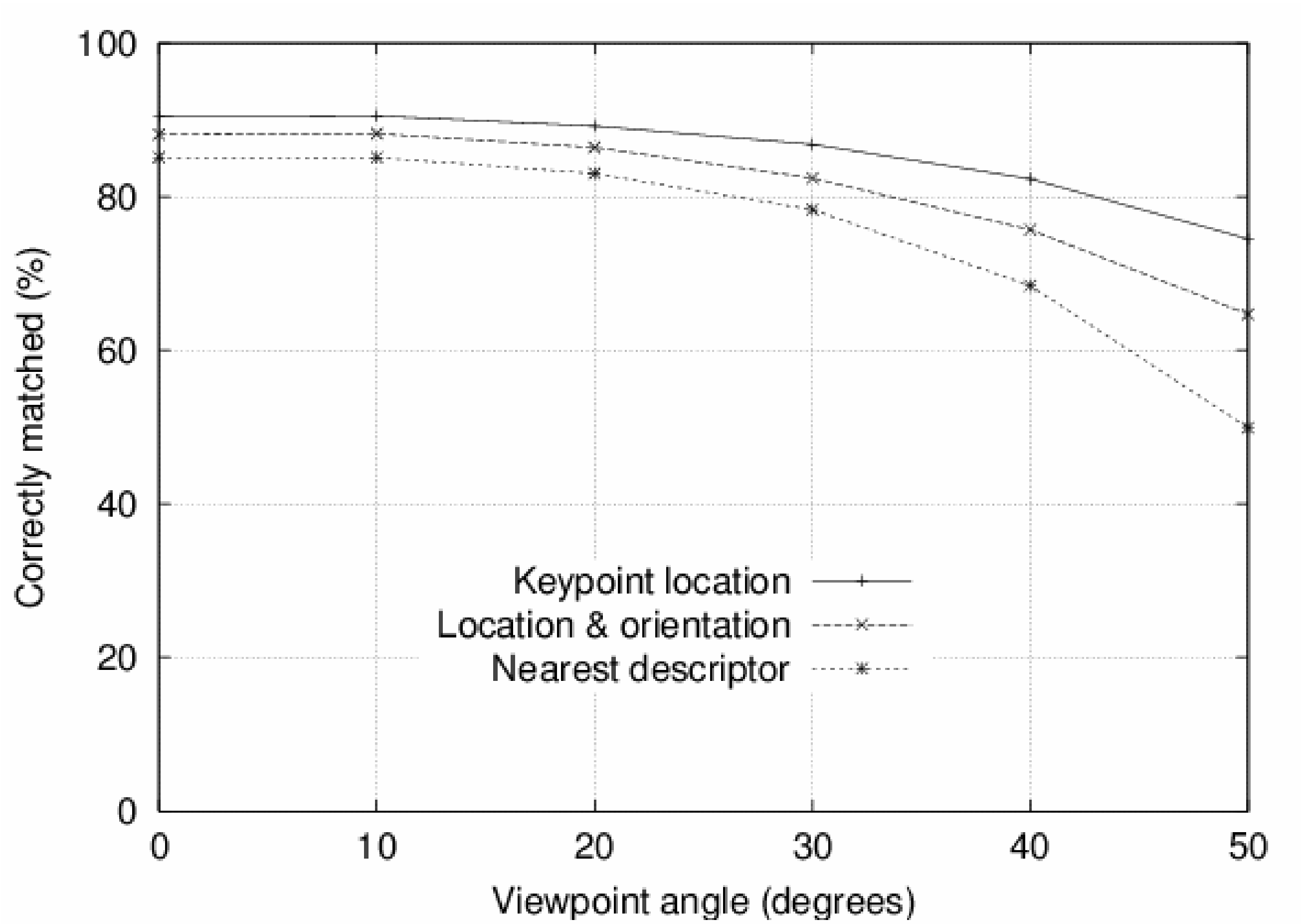
Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



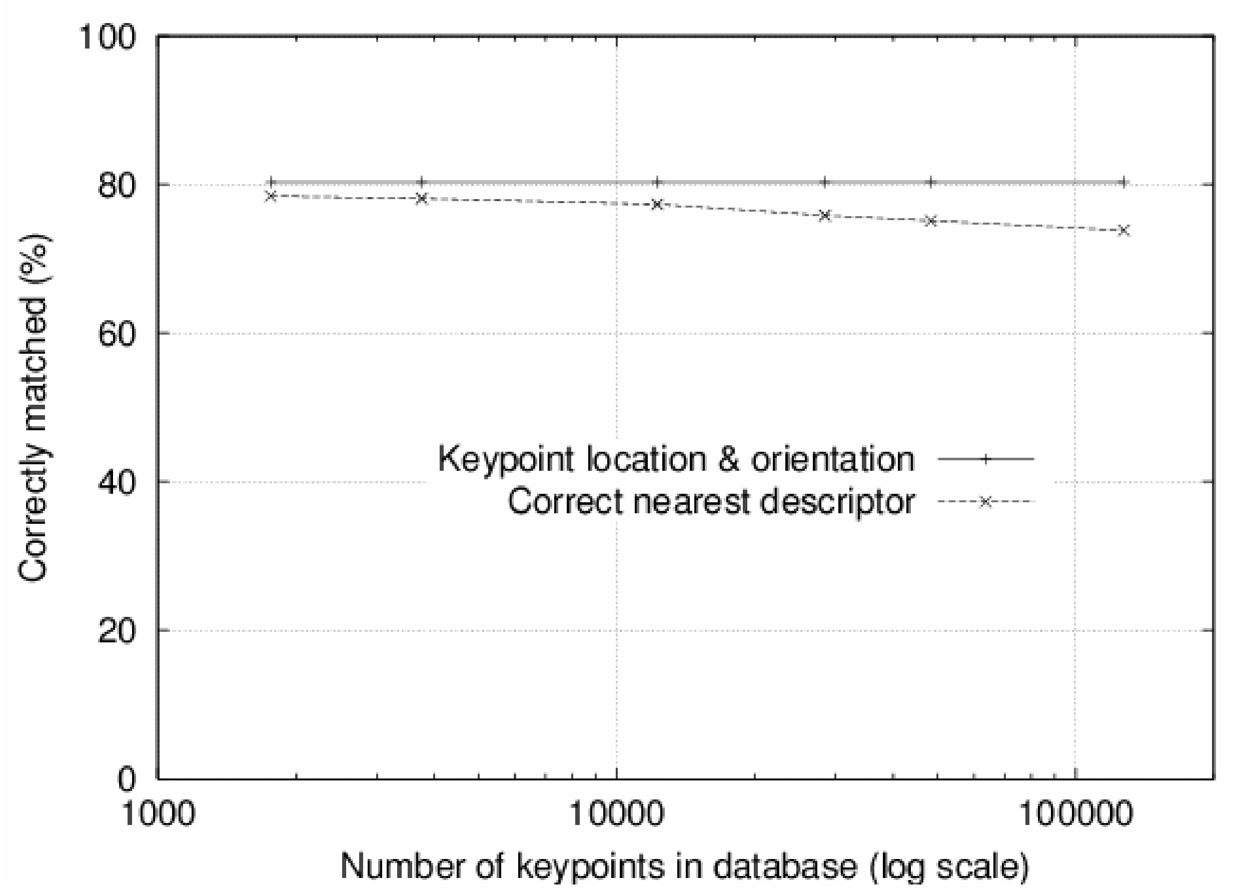
Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



Distinctiveness of features

- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match



Nearest-neighbor matching to feature database

- Hypotheses are generated by **approximate nearest neighbor** matching of each feature to vectors in the database
  - We use best-bin-first (Beis & Lowe, 97) modification to k-d tree algorithm
  - Use heap data structure to identify bins in order by their distance from query point
- **Result:** Can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time

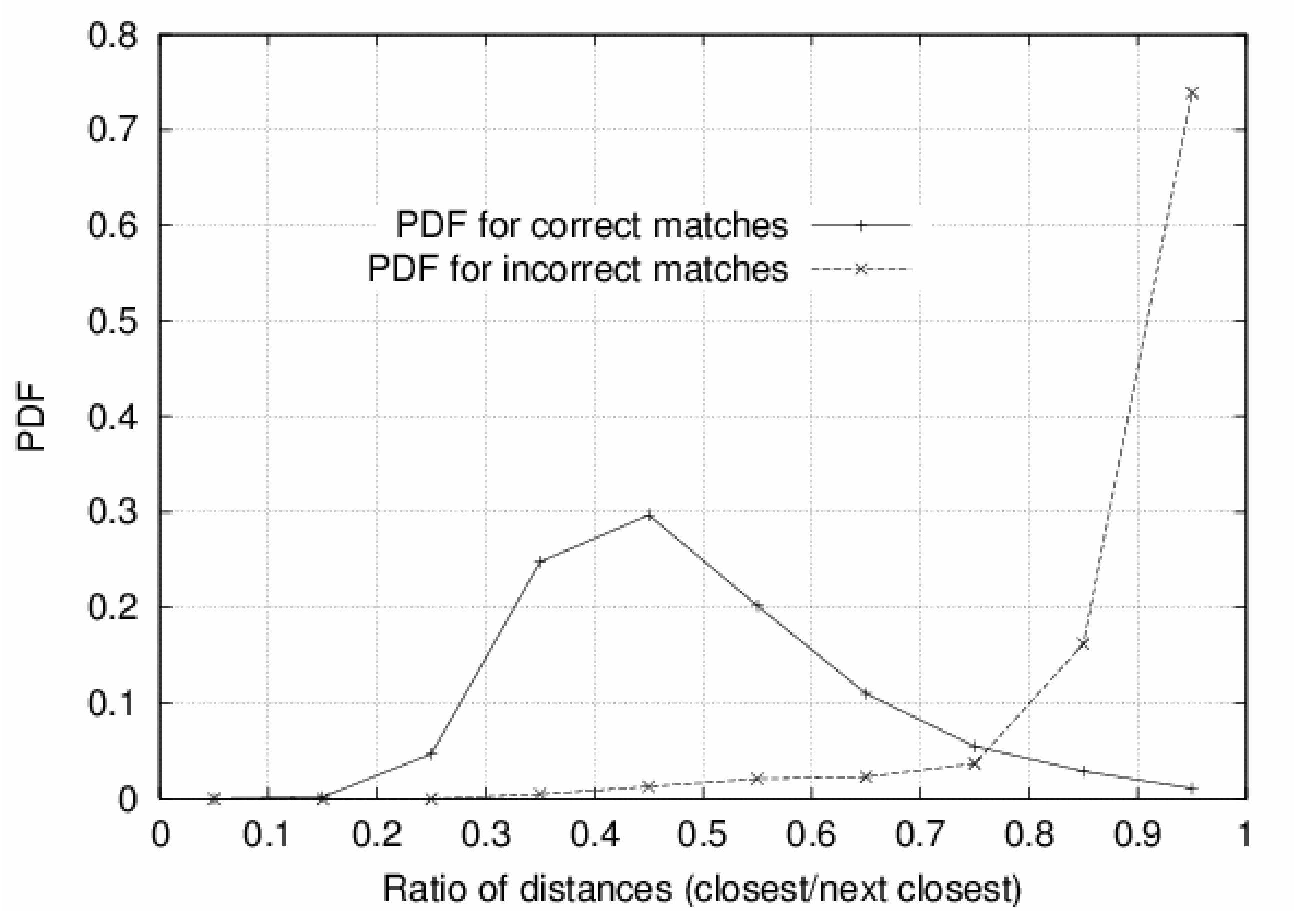
Detecting 0.1% inliers among 99.9% outliers

- We need to recognize clusters of just 3 consistent features among 3000 feature match hypotheses
- LMS or RANSAC would be hopeless!

- **Generalized Hough transform**
  - Vote for each potential match according to model ID and pose
  - Insert into multiple bins to allow for error in similarity approximation
  - Check collisions

Probability of correct match

- Compare distance of **nearest** neighbor to **second** nearest neighbor (from different object)
- Threshold of 0.8 provides excellent separation



Model verification

1. Examine all clusters with at least 3 features
2. Perform least-squares affine fit to model.
3. Discard outliers and perform top-down check for additional features.
4. Evaluate probability that match is correct
   - Use Bayesian model, with probability that features would arise by chance if object was *not* present (Lowe, CVPR 01)

Solution for affine parameters

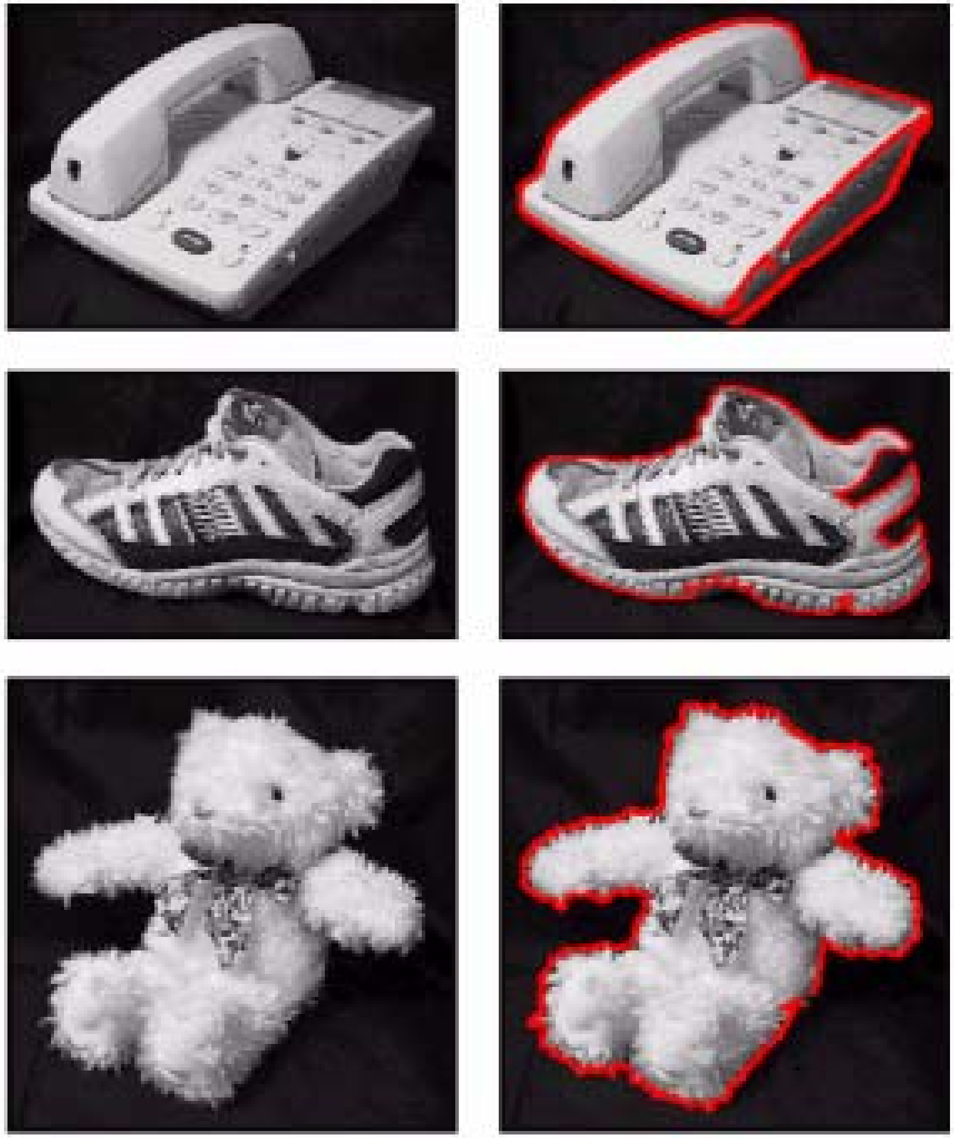
- Affine transform of [x,y] to [u,v]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- Rewrite to solve for transform parameters:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & & & \\ & & \dots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

3D Object Recognition

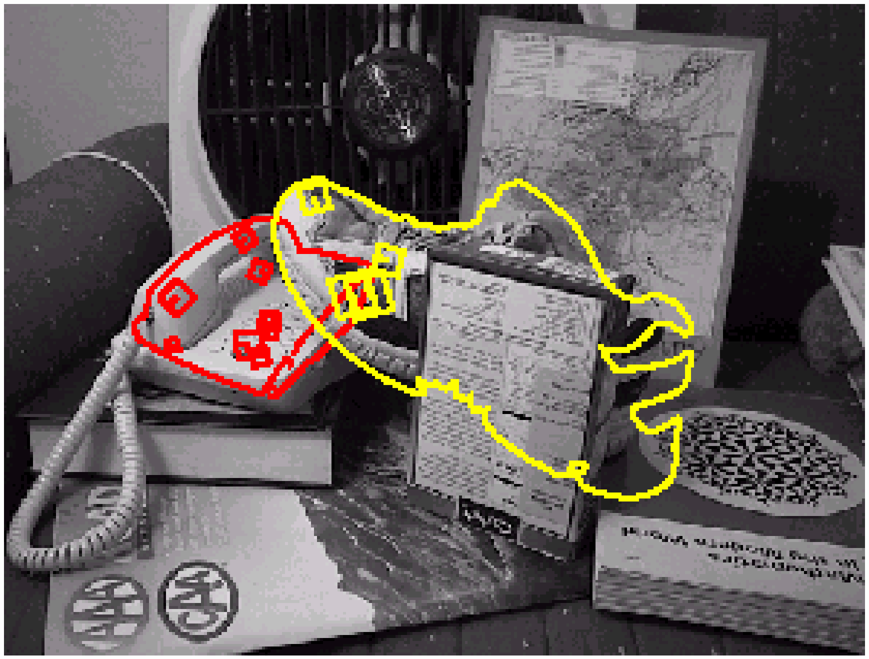


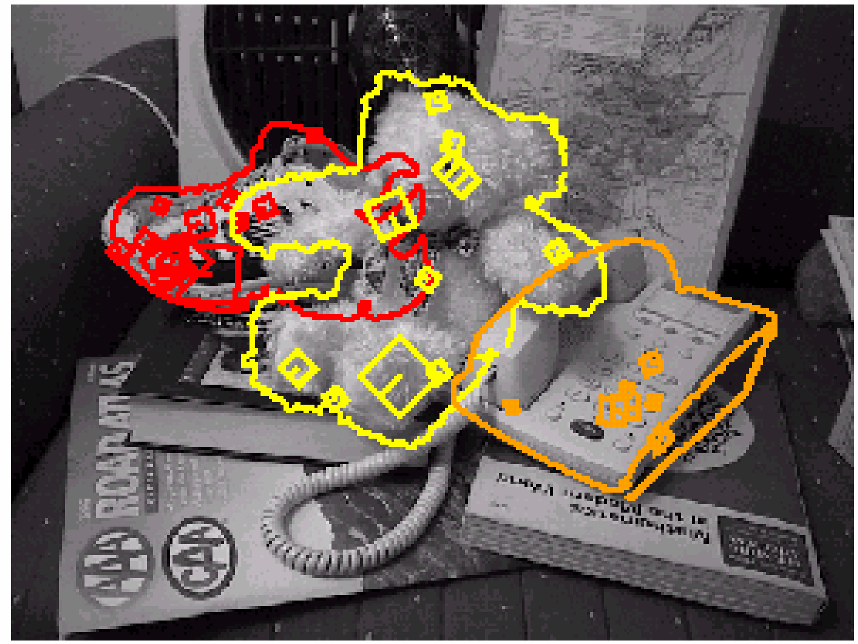
- Extract outlines with background subtraction

3D Object Recognition

- Only 3 keys are needed for recognition, so extra keys provide robustness
- Affine model is no longer as accurate

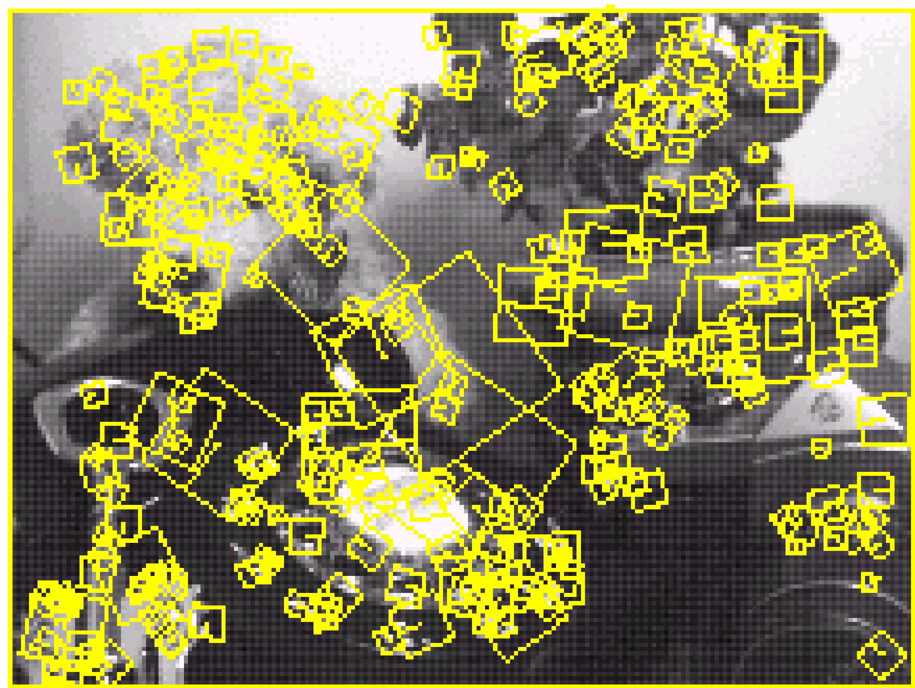
Recognition under occlusion





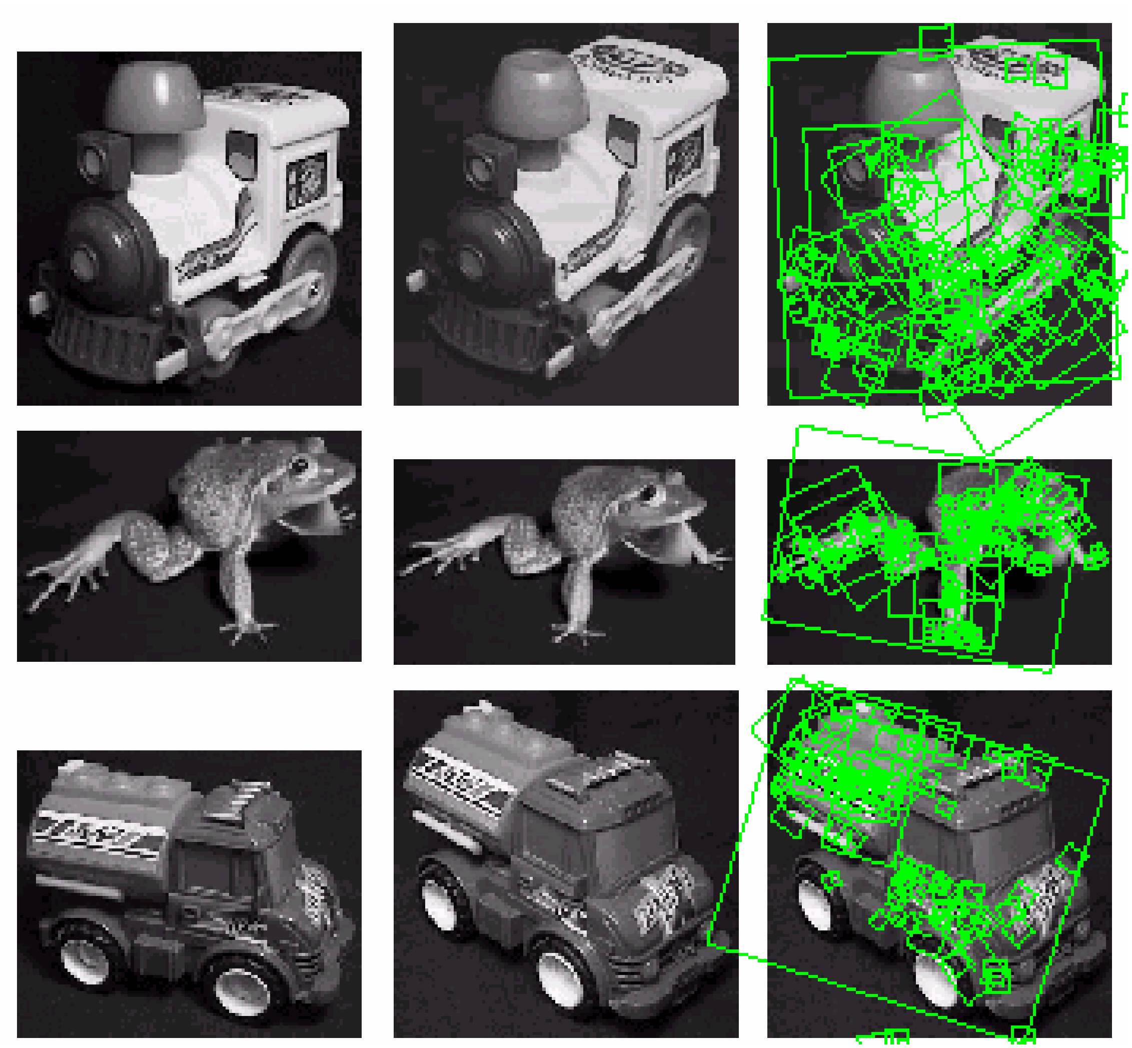
Test of illumination invariance

- Same image under differing illumination

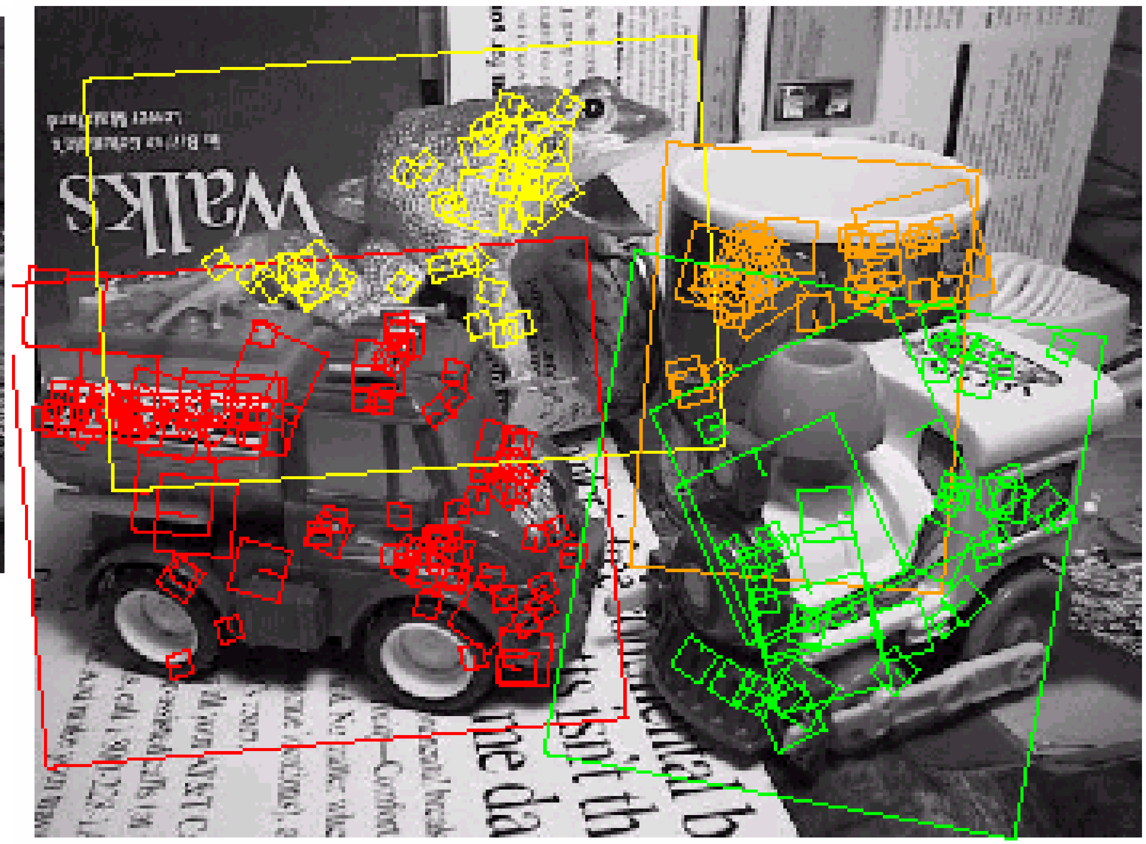


273 keys verified in final match

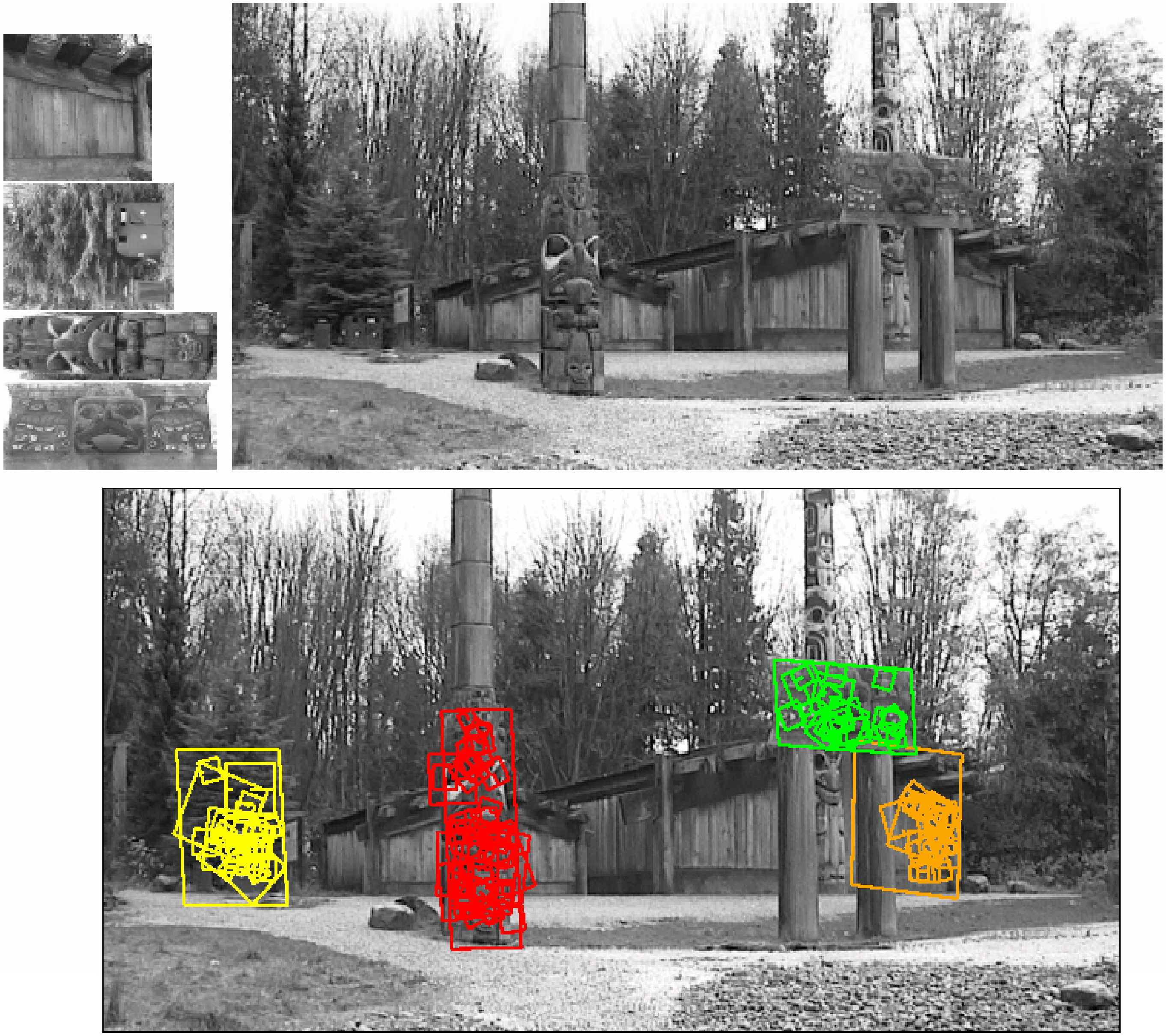
Examples of view interpolation



Recognition using View Interpolation



Location recognition

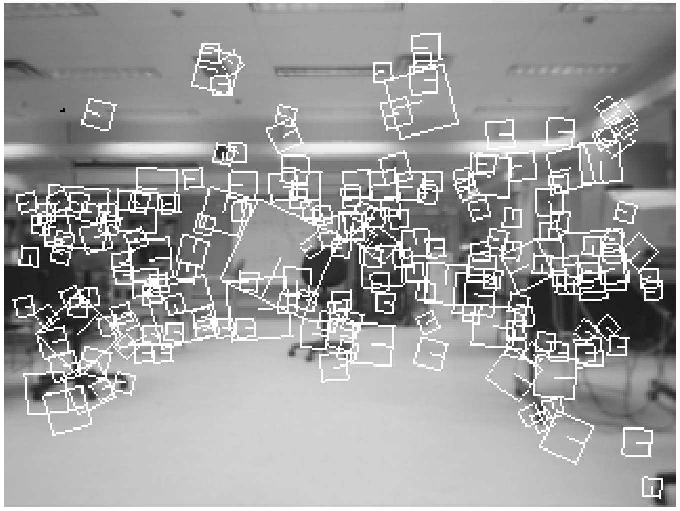


Robot localization results

- Joint work with Stephen Se, Jim Little

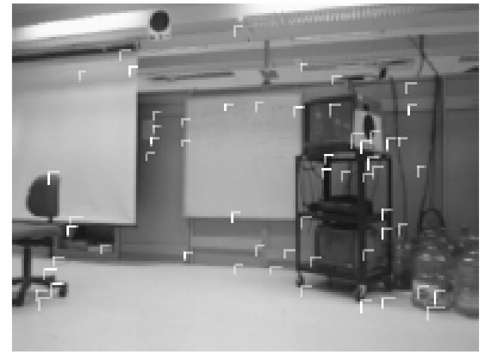
- **Map registration:** The robot can process 4 frames/sec and localize itself within 5 cm
- **Global localization:** Robot can be turned on and recognize its position anywhere within the map
- **Closing-the-loop:** Drift over long map building sequences can be recognized. Adjustment is performed by aligning submaps.

Robot Localization



(a)

(b)



(c)

(d)

(e)

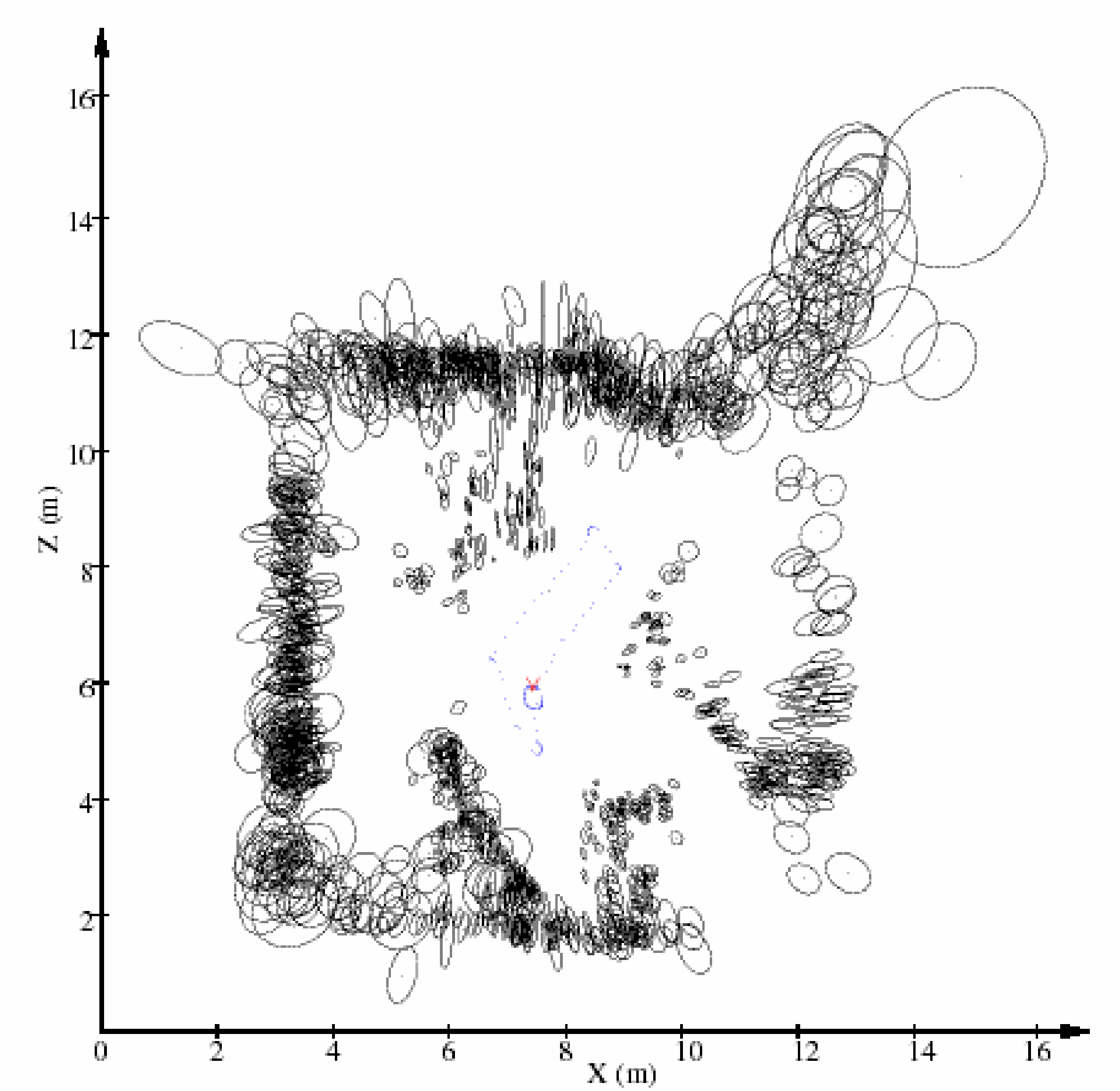
(f)

(g)

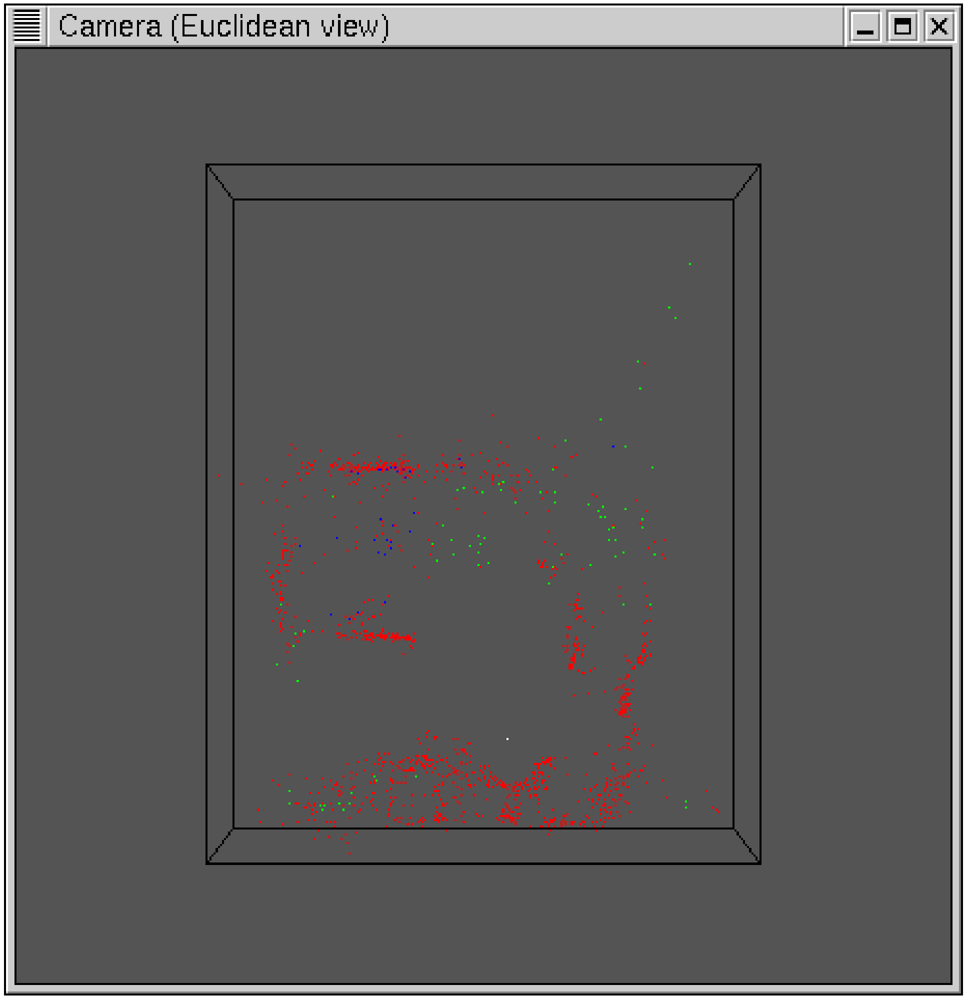
(h)

(i)

Map continuously built over time

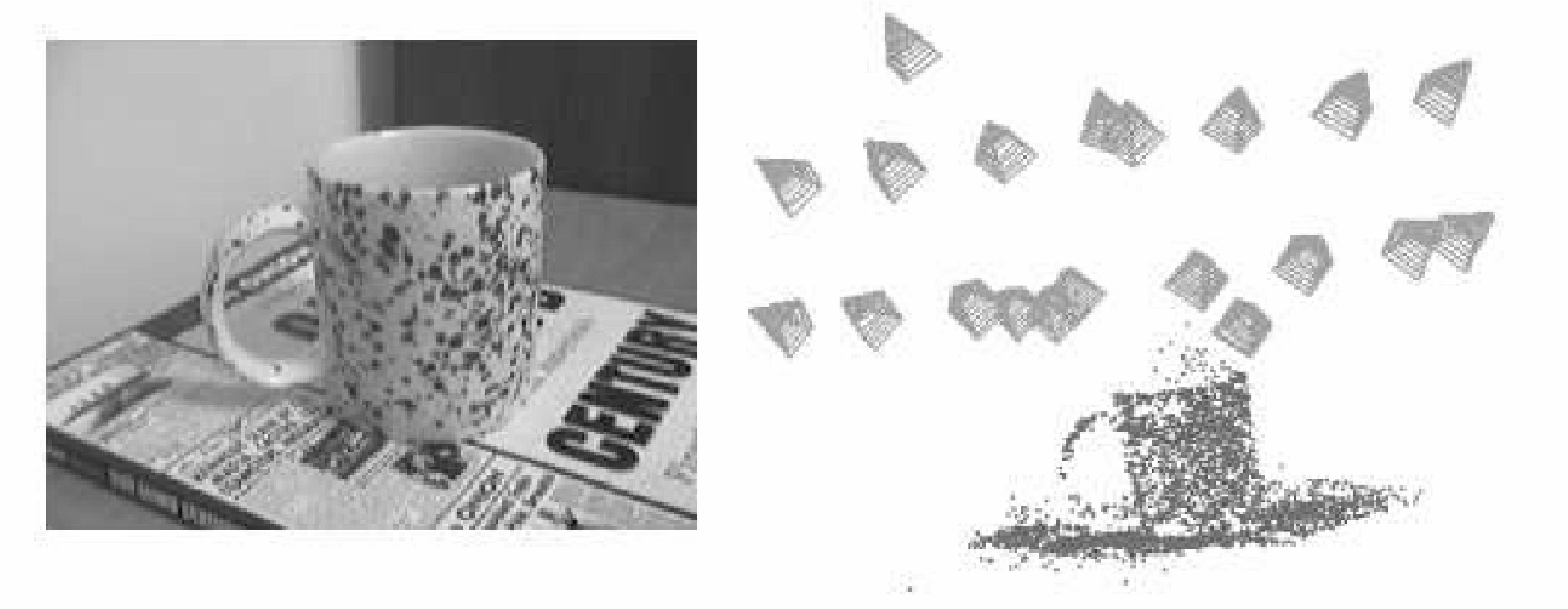


Locations of map features in 3D



Augmented Reality (with Iryna Gordon)

- Solve for 3D structure from multiple images
- Recognize scenes and insert 3D objects



Shows one of 20 images taken with handheld camera

3D Structure and Virtual Object Placement

- Solve for cameras and 3D points:
  - Uses bundle adjustment with Levenberg-Marquardt and robust metric
  - Initialize all cameras at the same location and points at the same depths
  - Solve bas-relief ambiguity by trying both options
- Insert object into scene:

Set location in one image, move along epipolar in other, adjust orientation

Jitter Reduction

- Minimize change in camera location, while keeping solution within expected noise range:

$$\min_{\mathbf{p}_t} \sum_j ||w_{tj}(\Pi(\mathbf{a}_{tj}) - \mathbf{x}_{tj})||^2 + \alpha||W(\mathbf{p}_t - \mathbf{p}_{t-1})||^2$$

p – camera pose

W – diagonal matrix for relative changes in camera parameters

Adjust α to keep residual within noise level of data so that object does not lag large motions

Augmentation Examples

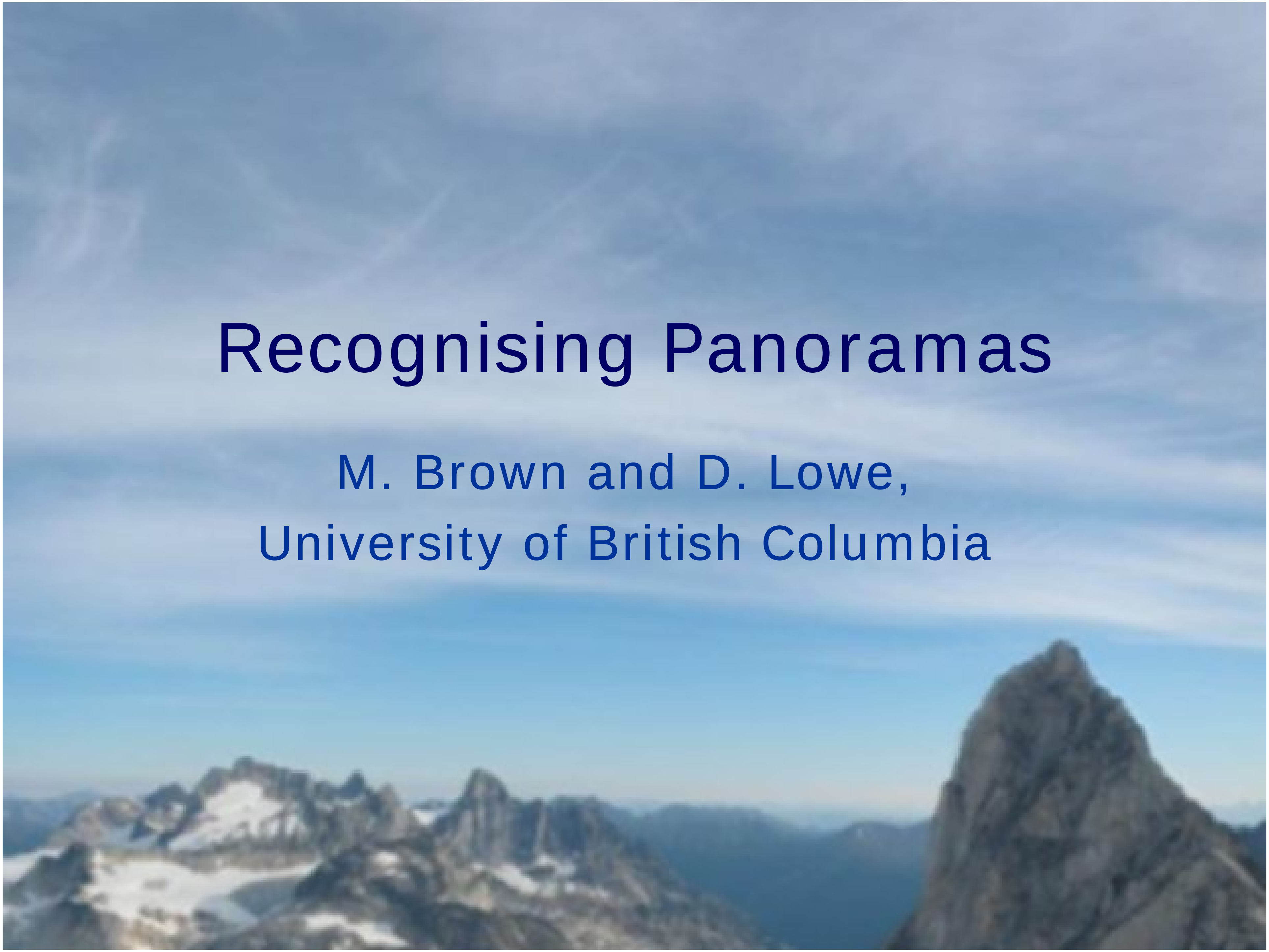
- Example of augmented tracking (executes about 5 frames/sec)

- Play

Sony Aibo (Evolution Robotics)

SIFT usage:

- **Recognize charging station**
- **Communicate with visual cards**


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

Introduction

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°

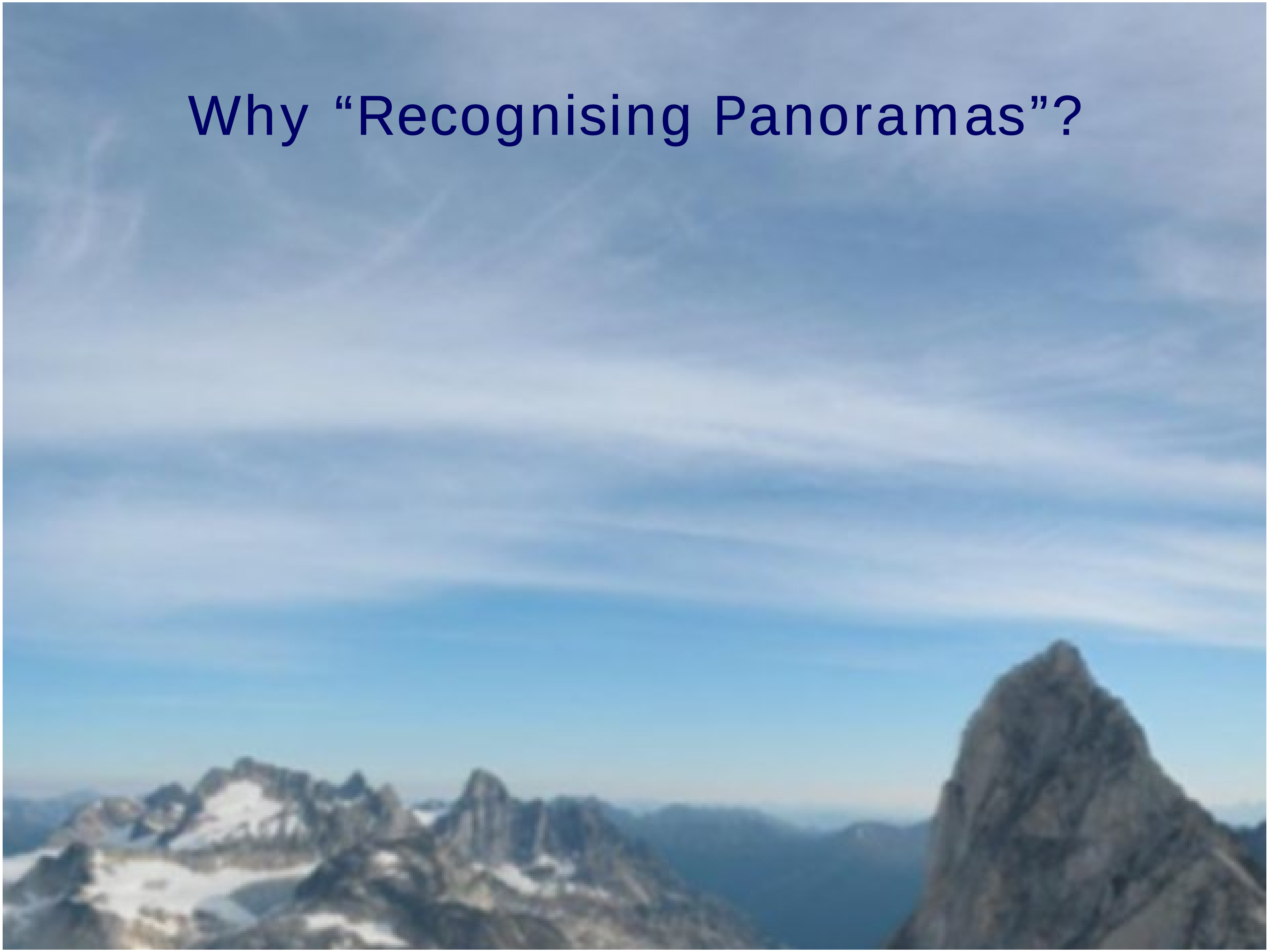
Introduction

- 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 “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

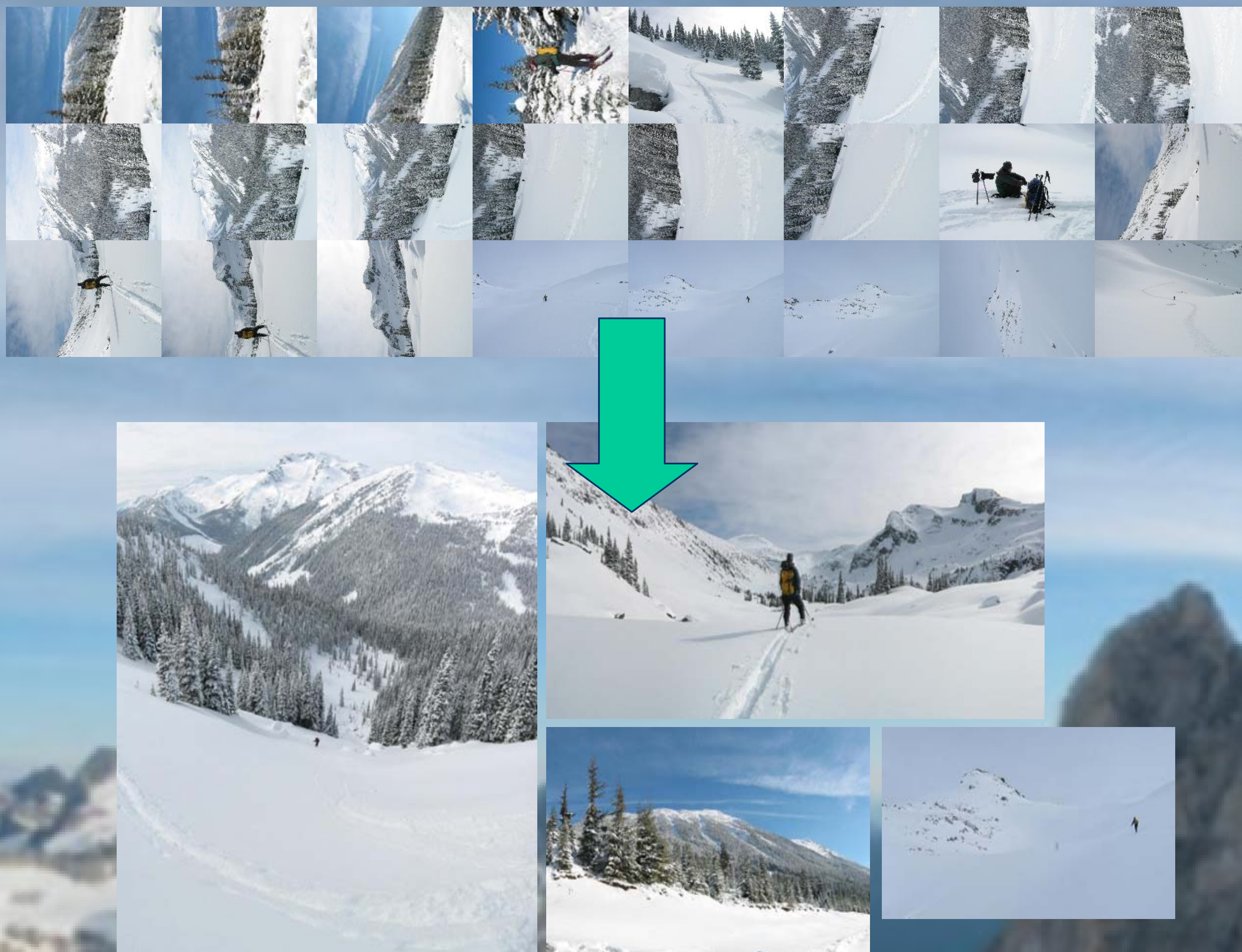
- 2D Rotations (θ, ϕ)
  - Ordering $\not\Rightarrow$ matching images

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
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Overview

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

Overview

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

Overview

- Feature Matching
  - SIFT Features
  - Nearest Neighbour 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

Overview

- Feature Matching
  - SIFT 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)

Overview

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

Overview

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

Overview

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

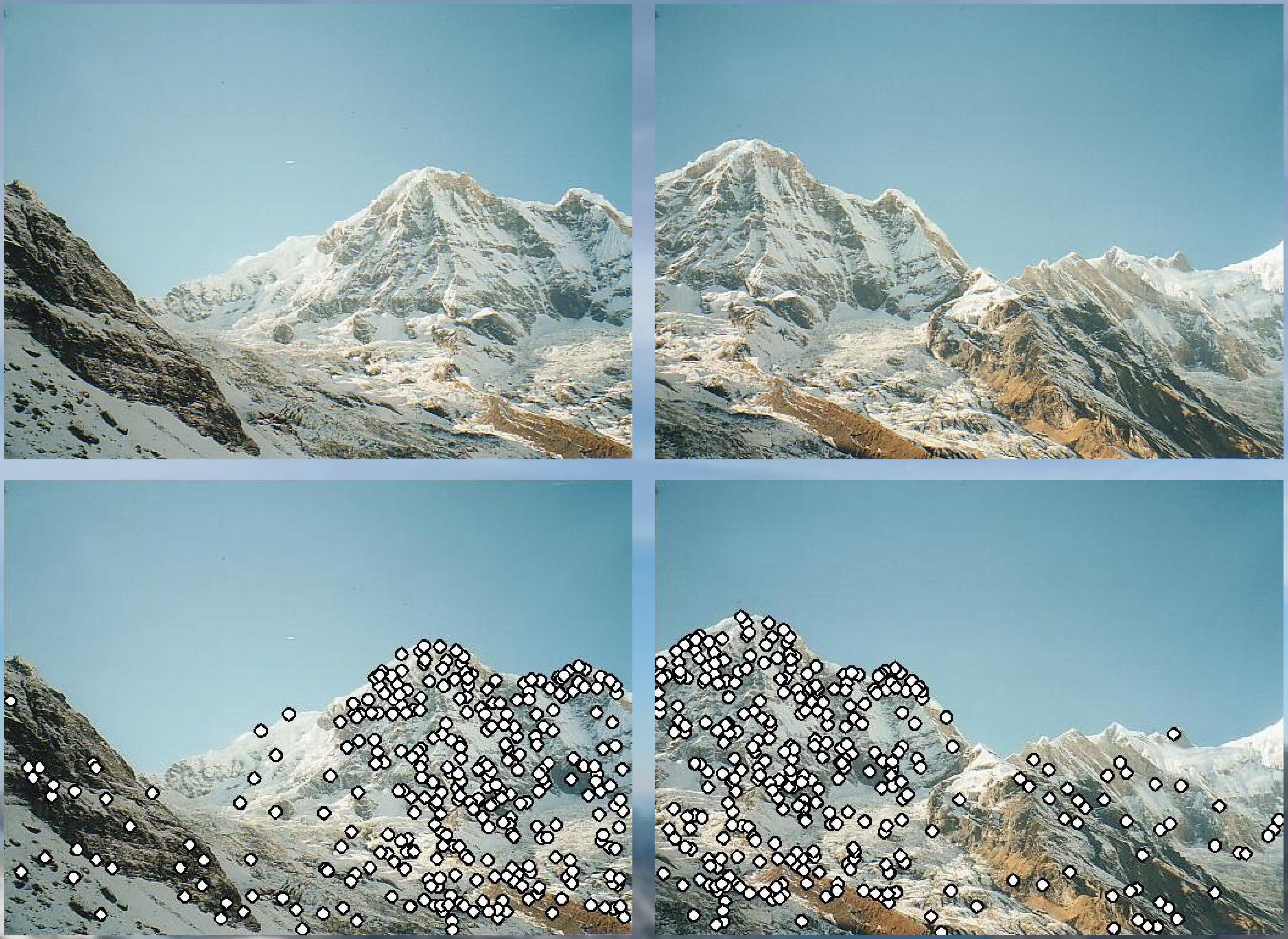
Overview

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

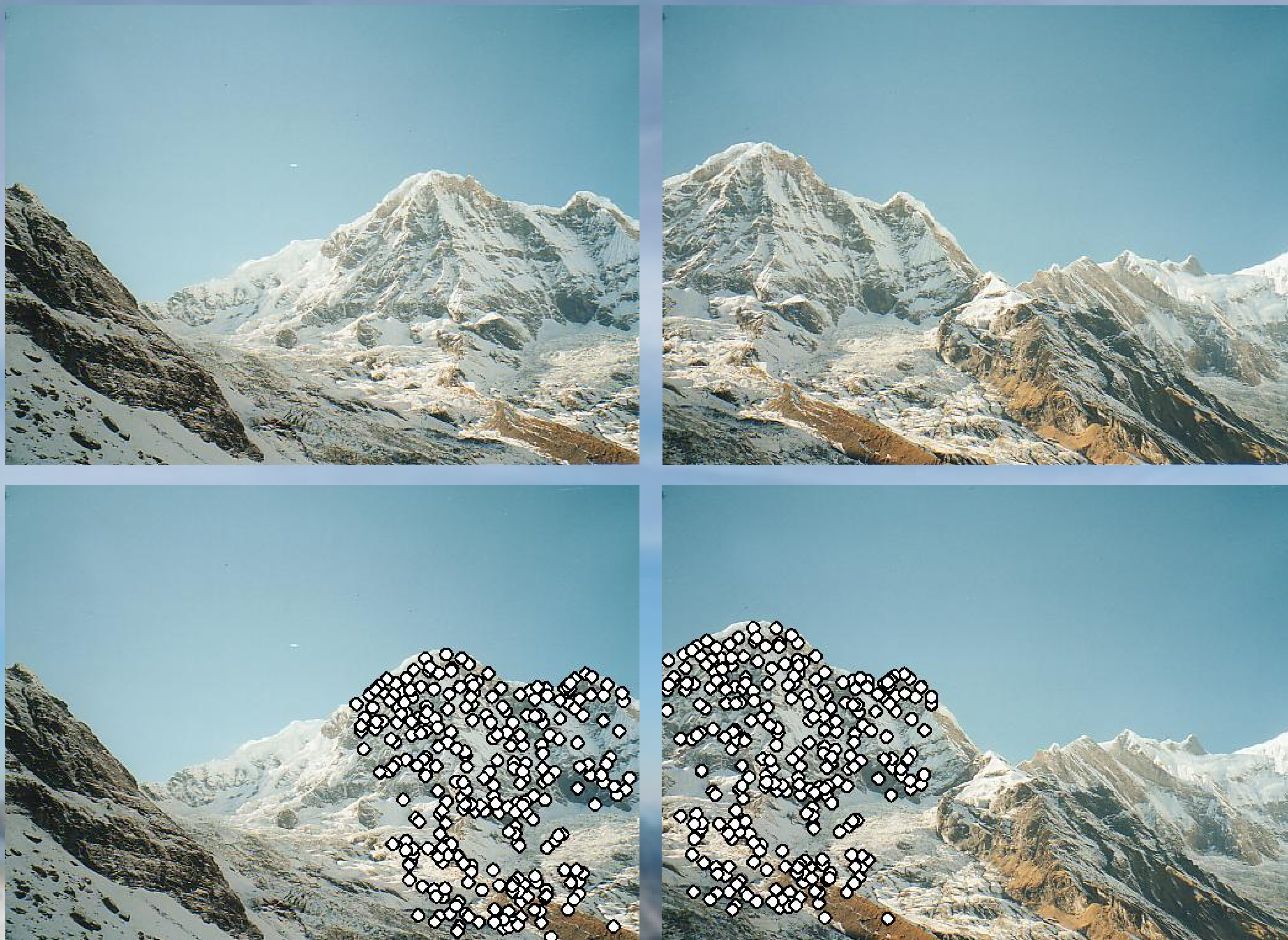
Overview

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

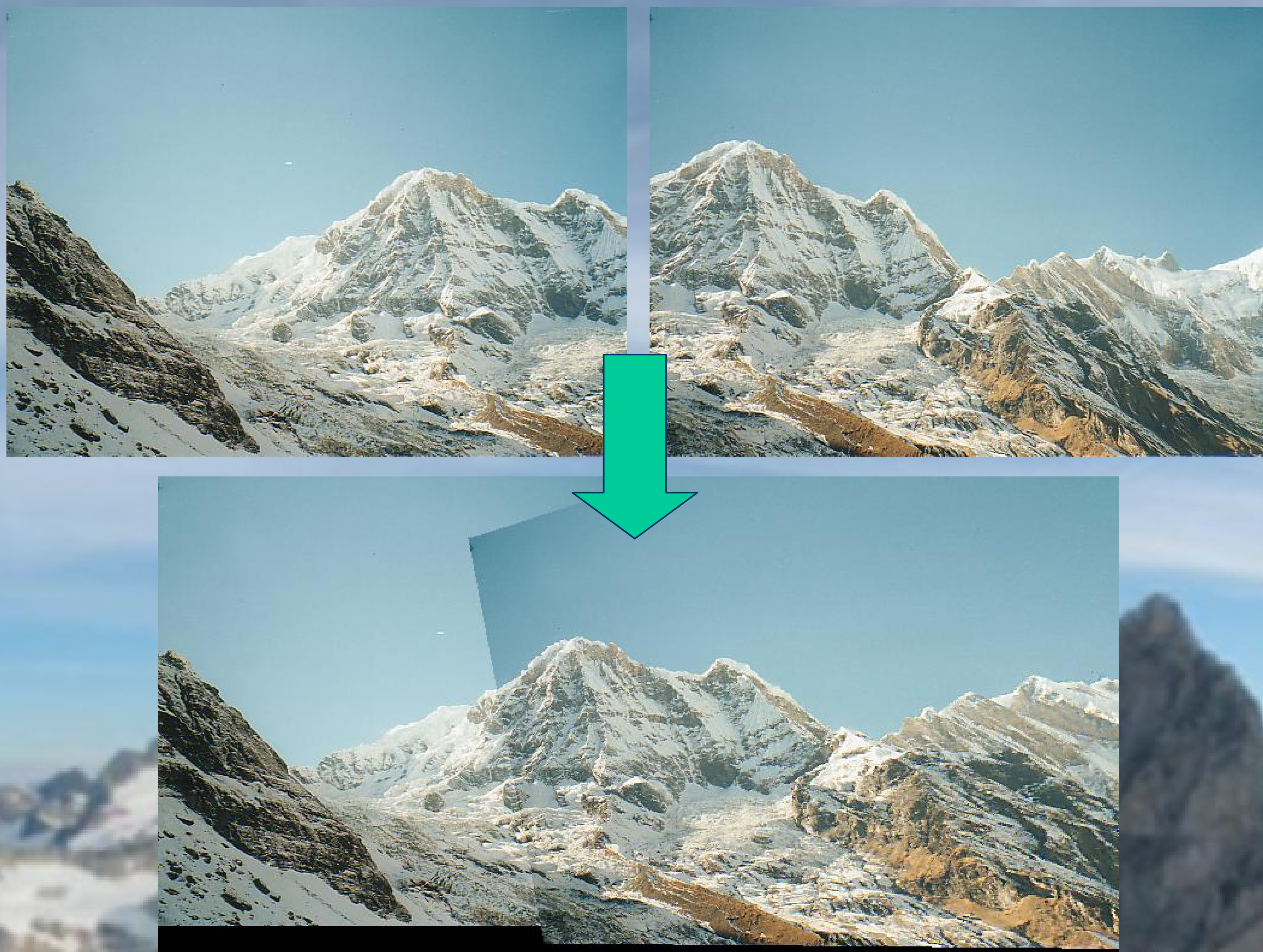
RANSAC for Homography



RANSAC for Homography



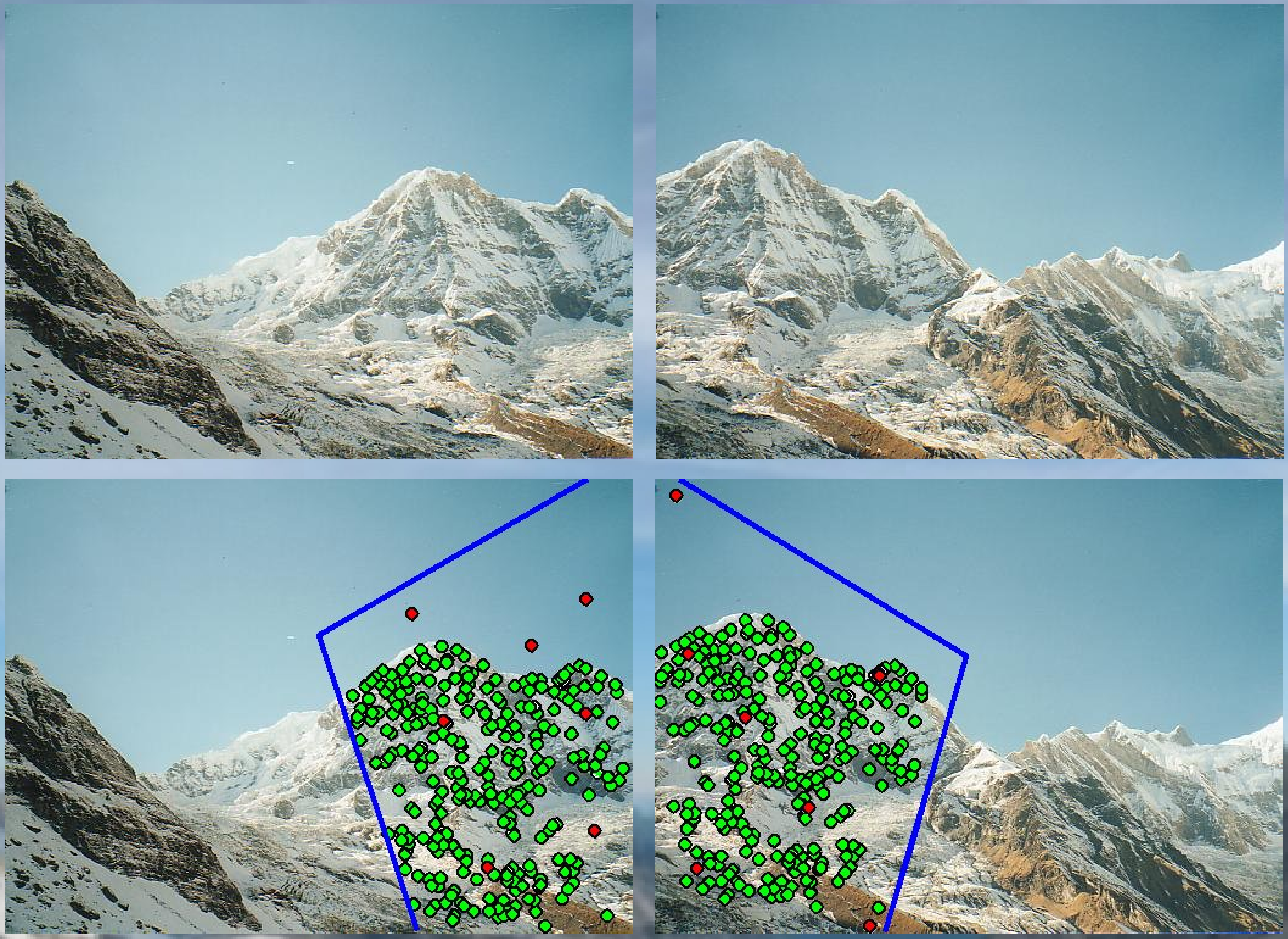
RANSAC for Homography



Overview

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

Probabilistic model for verification



Probabilistic model for verification

- Compare probability that this set of RANSAC inliers/outliers was generated by a correct/false image match

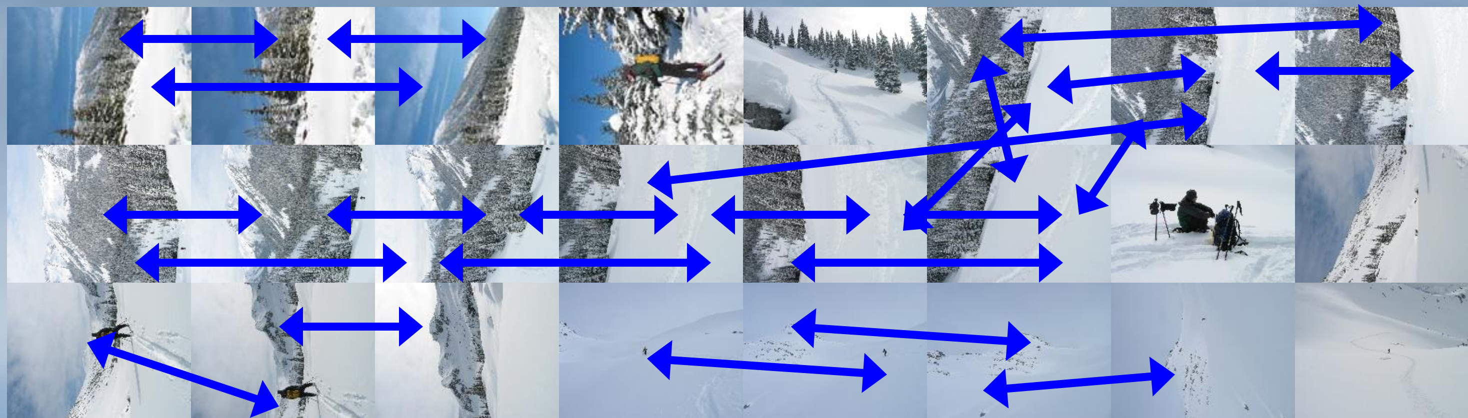
$$\frac{B(n_i; n_f, p_1)}{B(n_i; n_f, p_0)} \underset{reject}{\overset{accept}{\gtrless}} \frac{1}{\frac{1}{p_{min}} - 1}$$

  - n_i = #inliers, n_f = #features
  - p_1 = p(inlier | match), p_0 = p(inlier | ~match)
  - p_{min} = acceptance probability

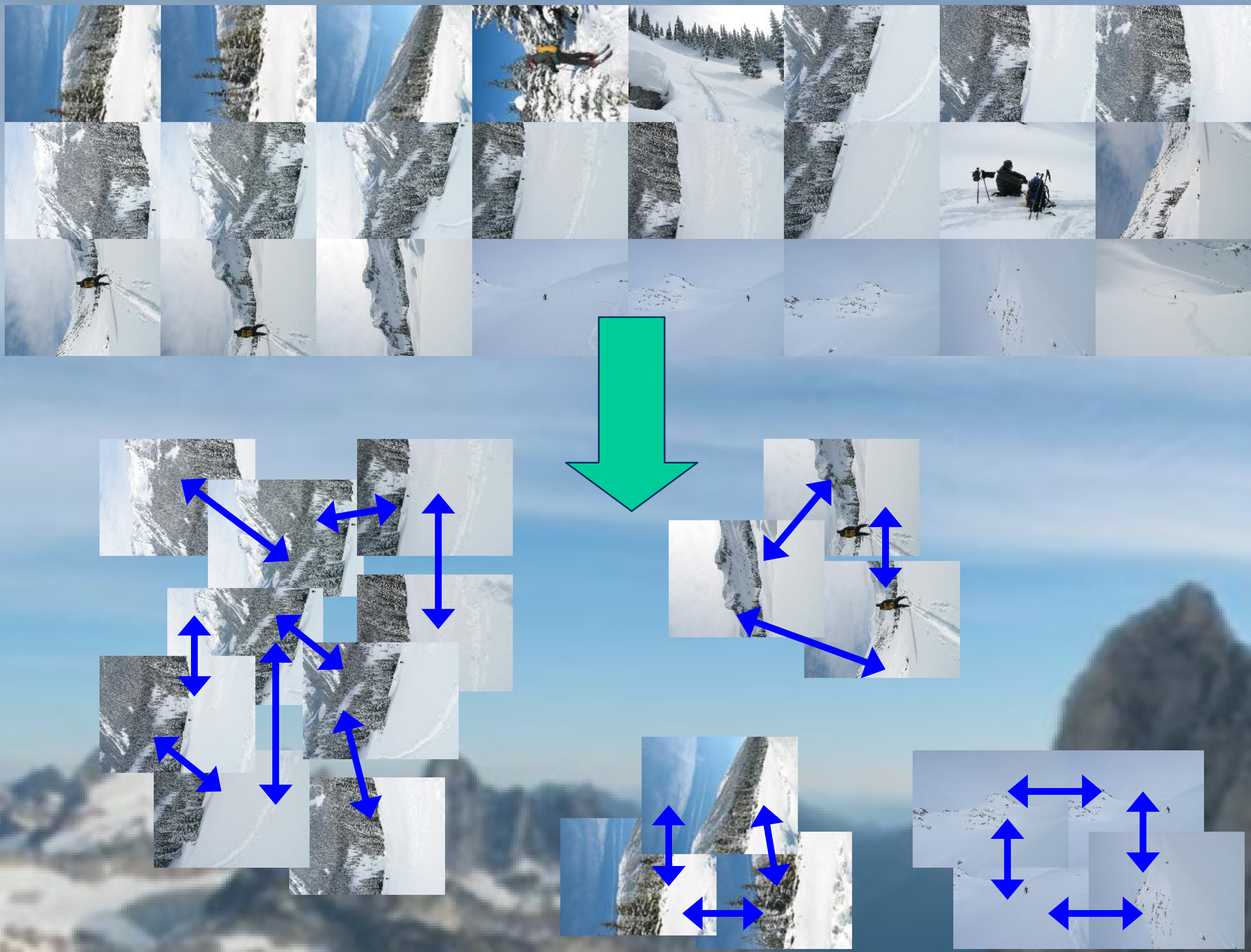
- Choosing values for p_1, p_0 and p_{min}

$$n_i > 5.9 + 0.22 n_f$$

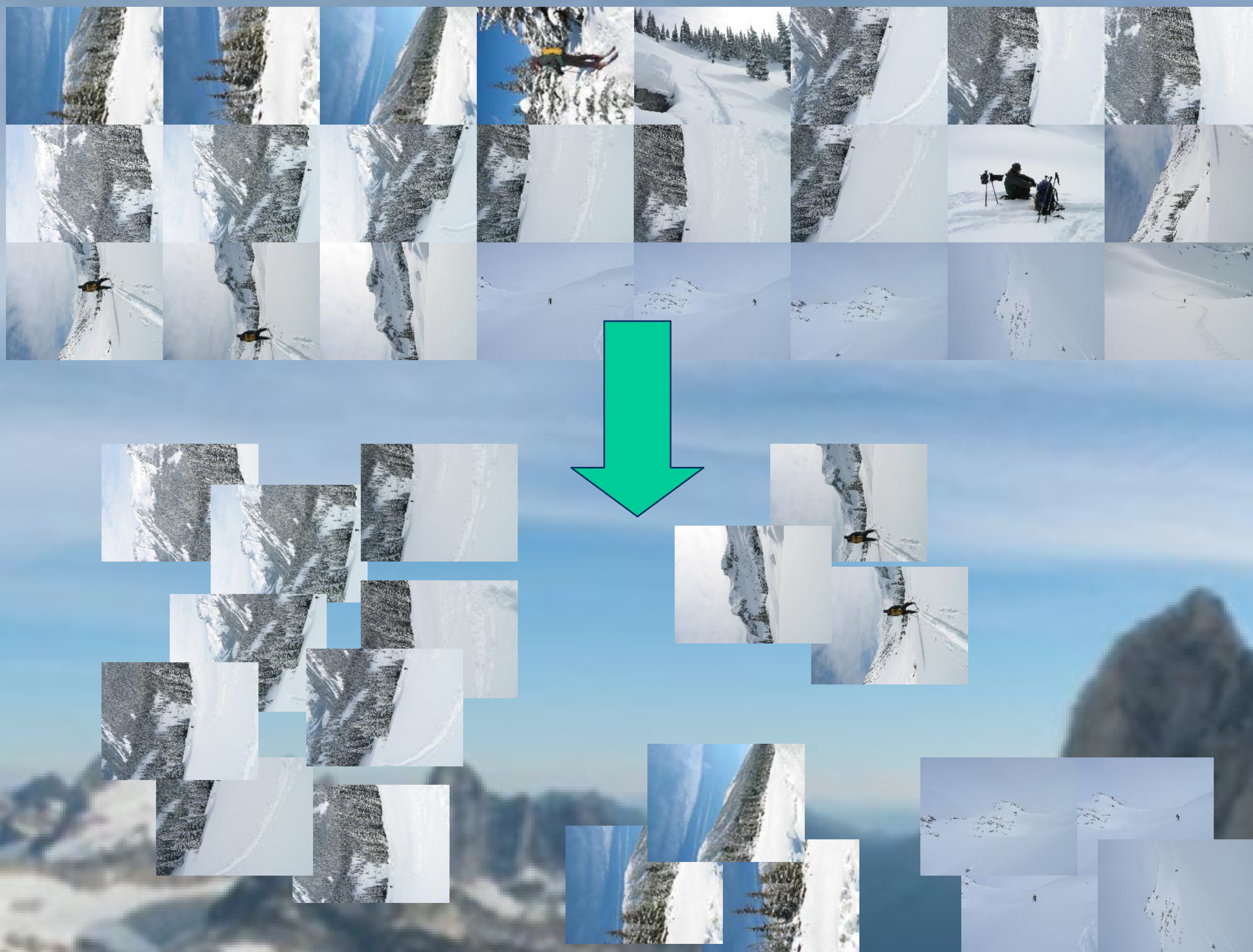
Finding the panoramas



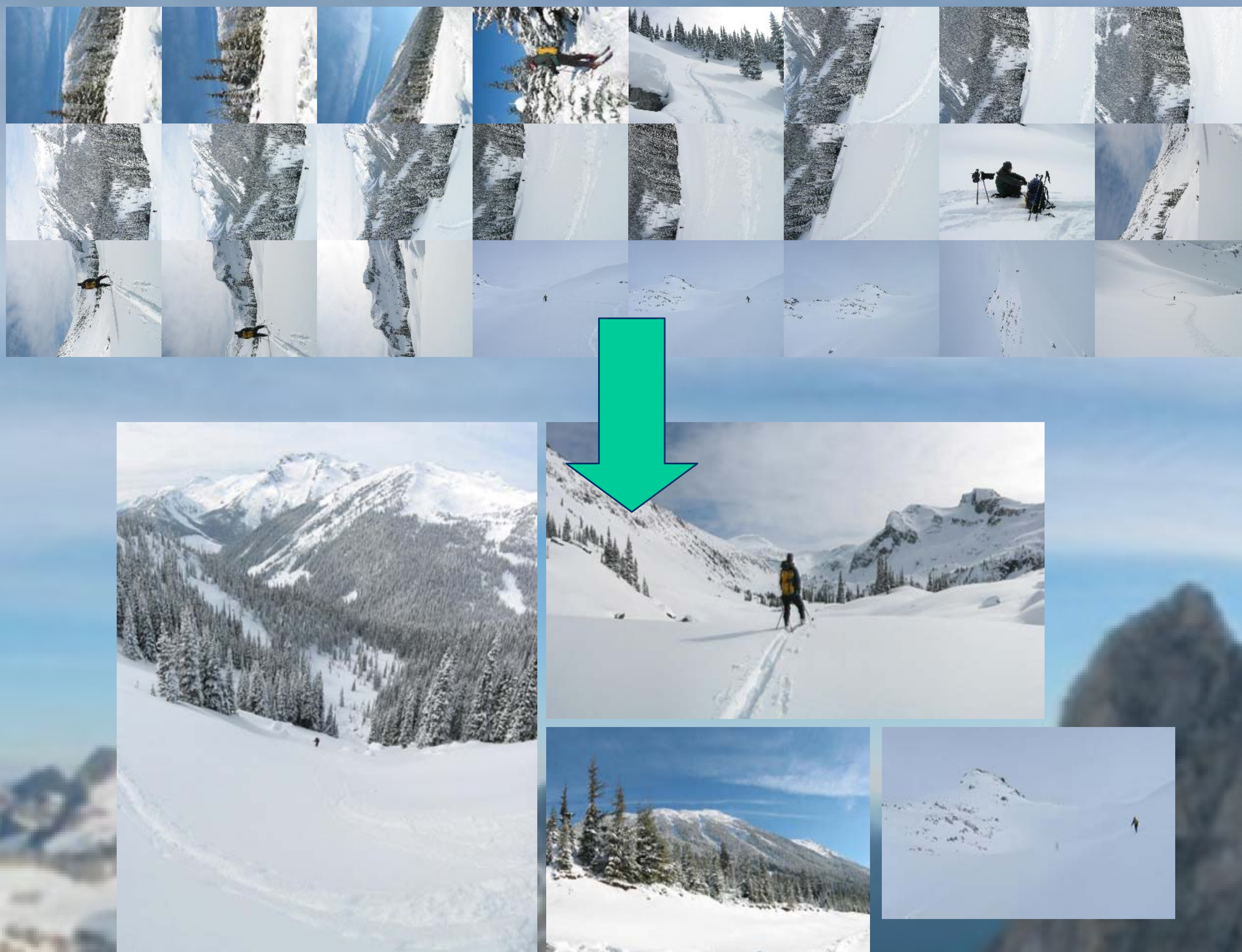
Finding the panoramas



Finding the panoramas



Finding the panoramas



Overview

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

Overview

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

Overview

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

Overview

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

Overview

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

Error function

- 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

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

Homography for Rotation

- Parameterise each camera by rotation and focal length

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

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

- This gives pairwise homographies

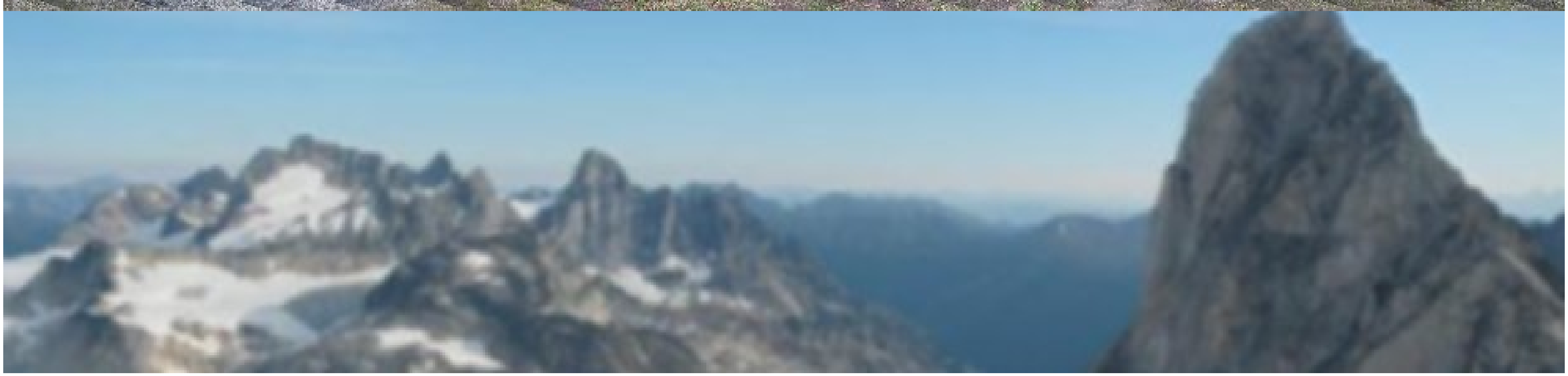
$$\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j, \;\; \mathbf{H}_{ij} = \mathbf{K}_i\mathbf{R}_i\mathbf{R}_j^T\mathbf{K}_j^{-1}$$

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



Overview

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

Overview

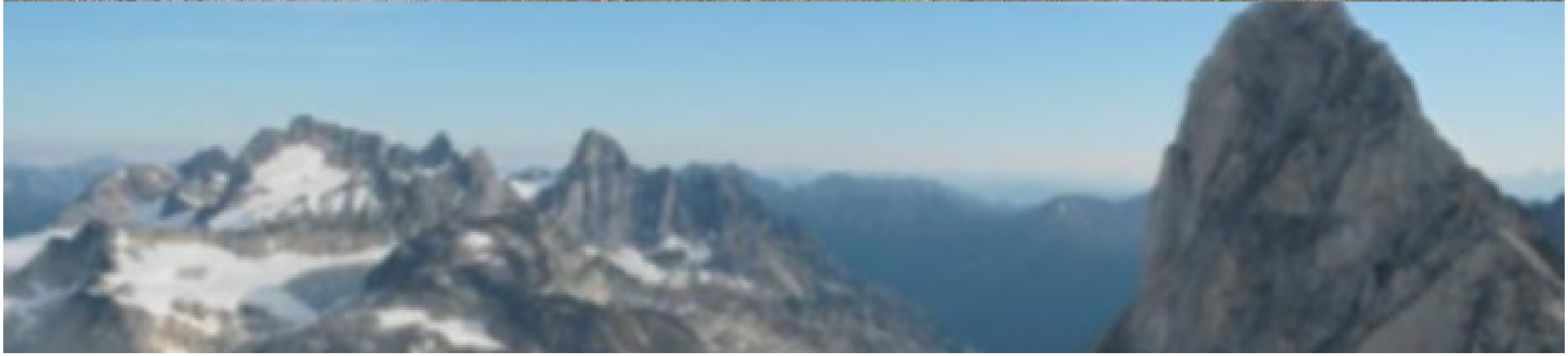
- Feature Matching
- Image Matching
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Overview

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

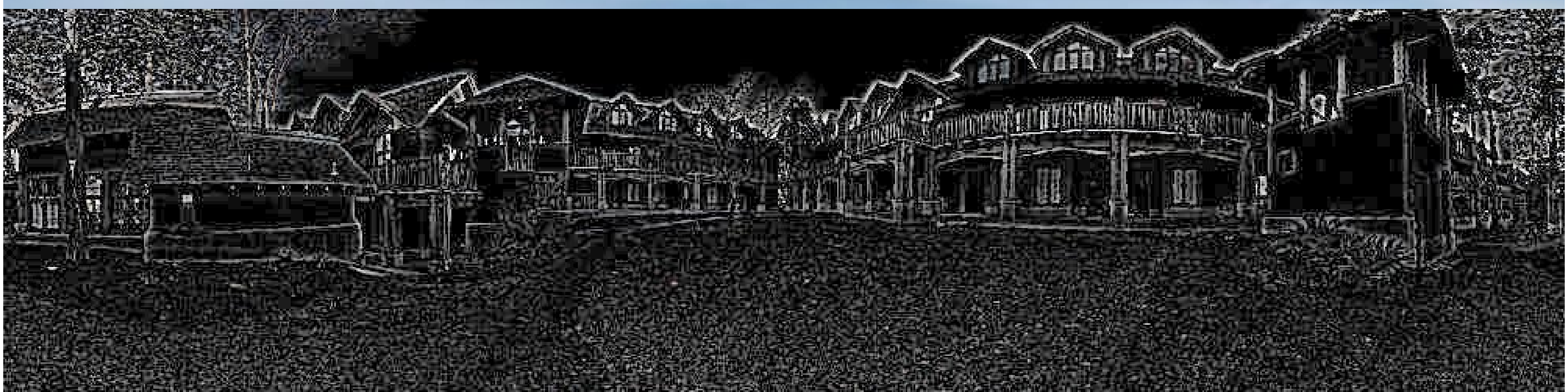
Multi-band Blending

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



2-band Blending

Low frequency (λ > 2 pixels)



High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

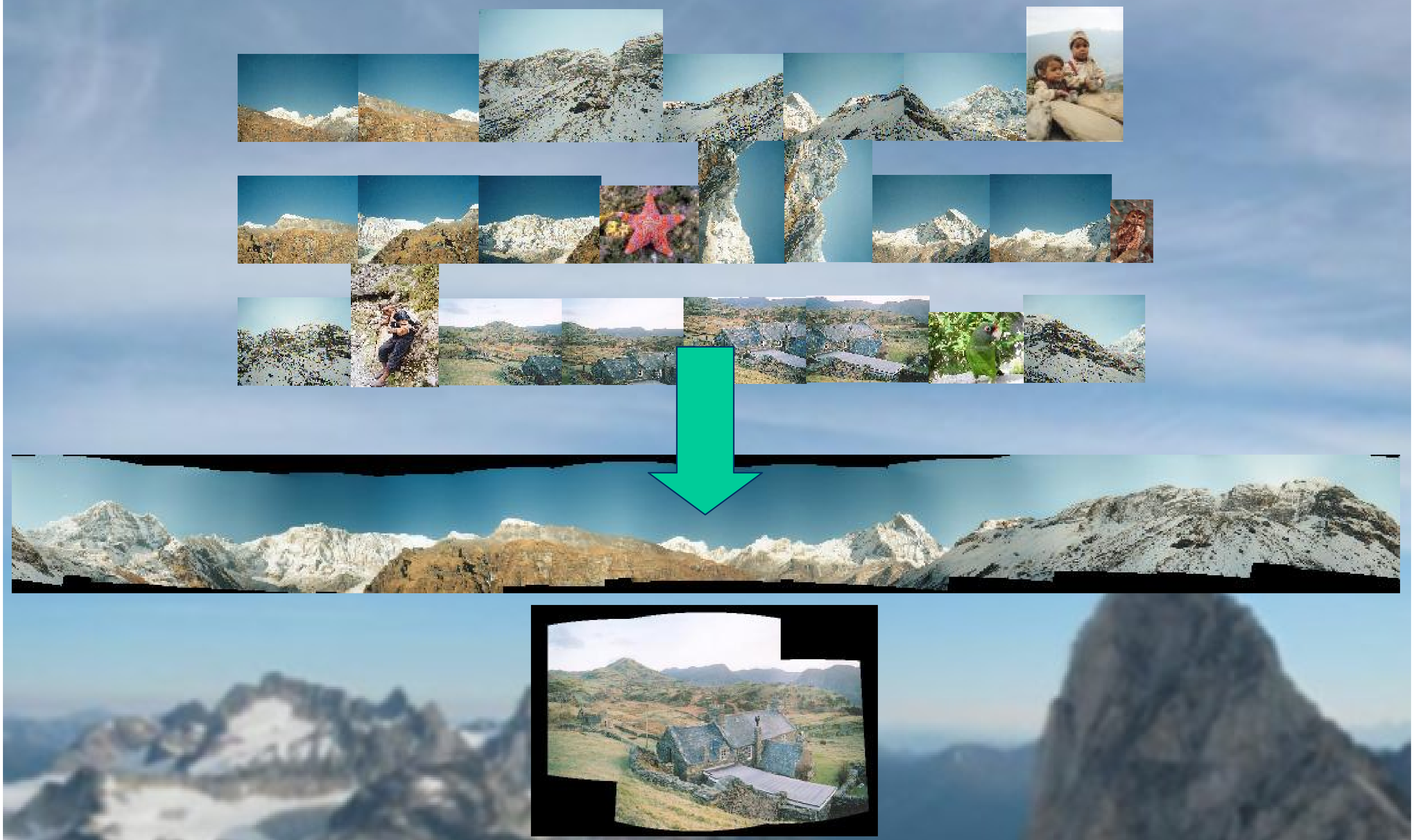
Overview

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

Overview

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

Results



Results

Overview

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

Overview

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

Conclusions

- Fully automatic panoramas
  - A recognition problem…
- Invariant feature based method
  - SIFT features, RANSAC, Bundle Adjustment, Multi-band Blending
  - O(nlogn)
- Future Work
  - 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

http://www.cs.ubc.ca/~mbrown/panorama/panorama.html