Visual Recognition and Machine Learning Summer School

Paris 2011

Instance-level recognition – part 2

Josef Sivic

http://www.di.ens.fr/~josef

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548

Laboratoire d'Informatique, Ecole Normale Supérieure, Paris

With slides from: O. Chum, K. Grauman, I. Laptev, S. Lazebnik, B. Leibe, D. Lowe, J. Philbin, J. Ponce, D. Nister, C. Schmid, N. Snavely, A. Zisserman

Outline

1. Local invariant features (45 mins, C. Schmid)

2. **Matching and recognition with local features (45 mins, J. Sivic)**

3. Efficient visual search (45 mins, J. Sivic)

4. Very large scale visual indexing – recent work (45 mins, C. Schmid)

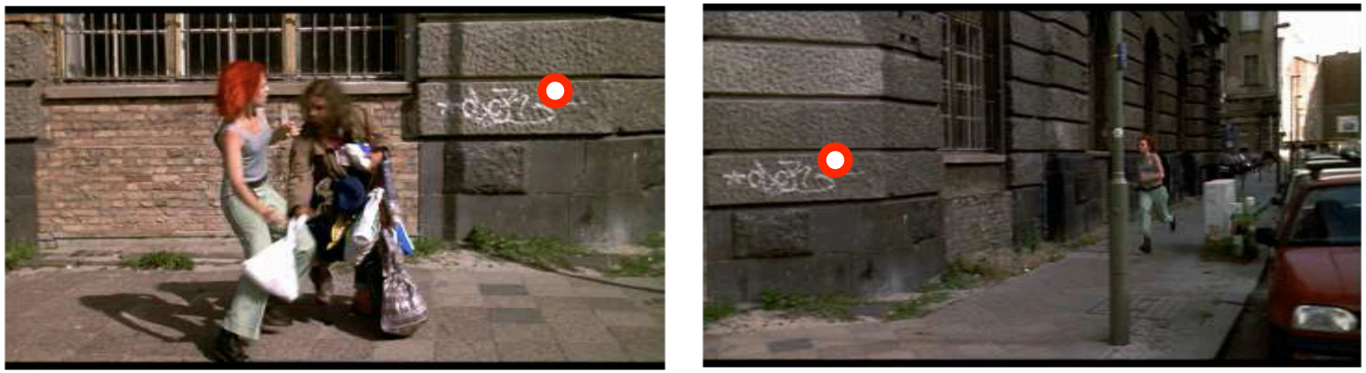
Practical session – Panorama stitching (60 mins)

Download:

http://www.di.ens.fr/willow/events/cvml2011/mosaic.zip

Image matching and recognition with local features

The goal: establish correspondence between two or more images



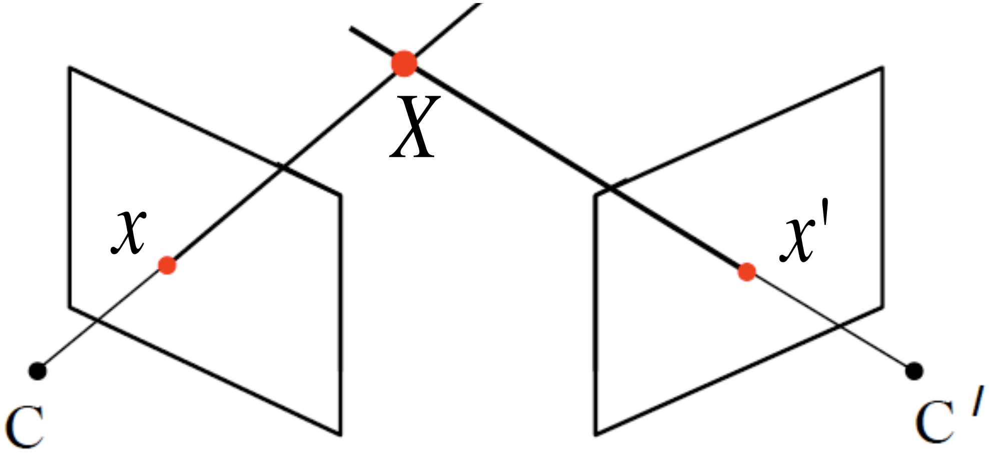


Image points x and x' are in correspondence if they are projections of the same 3D scene point X.

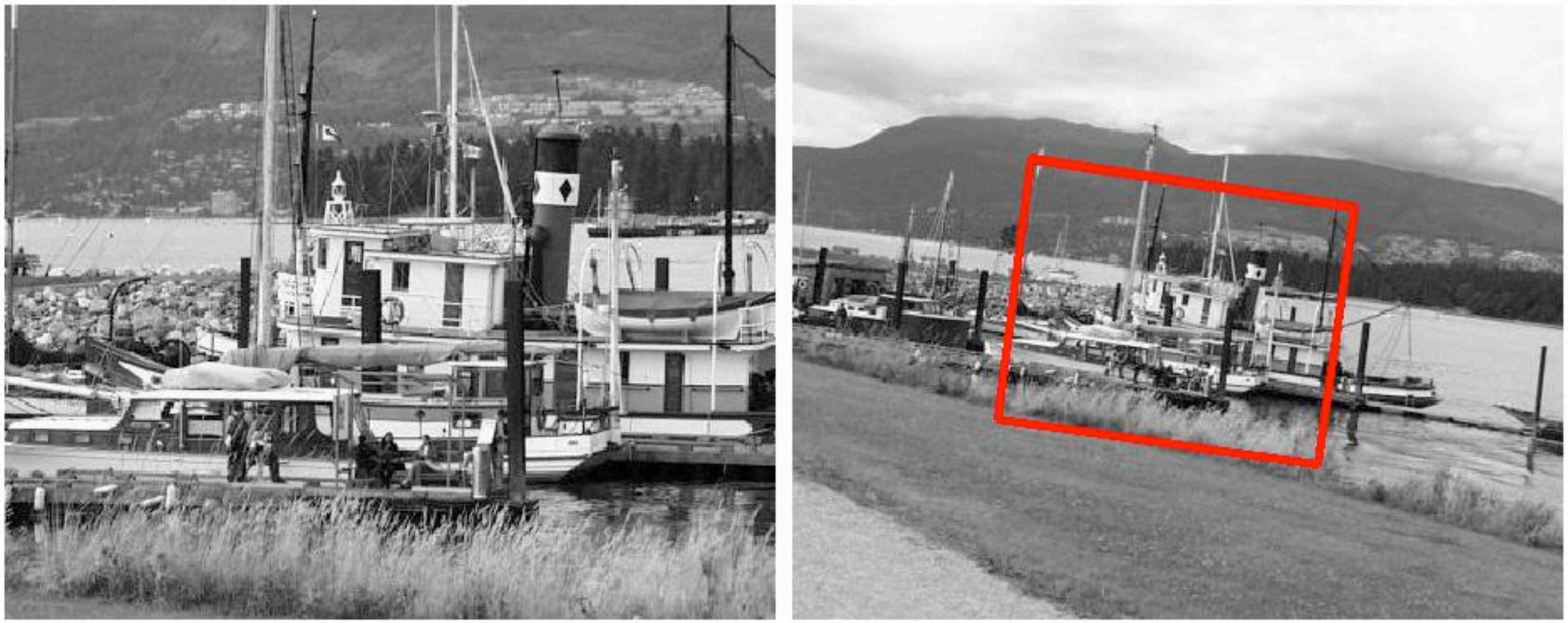
Images courtesy A. Zisserman

Example I: Wide baseline matching

Establish correspondence between two (or more) images.

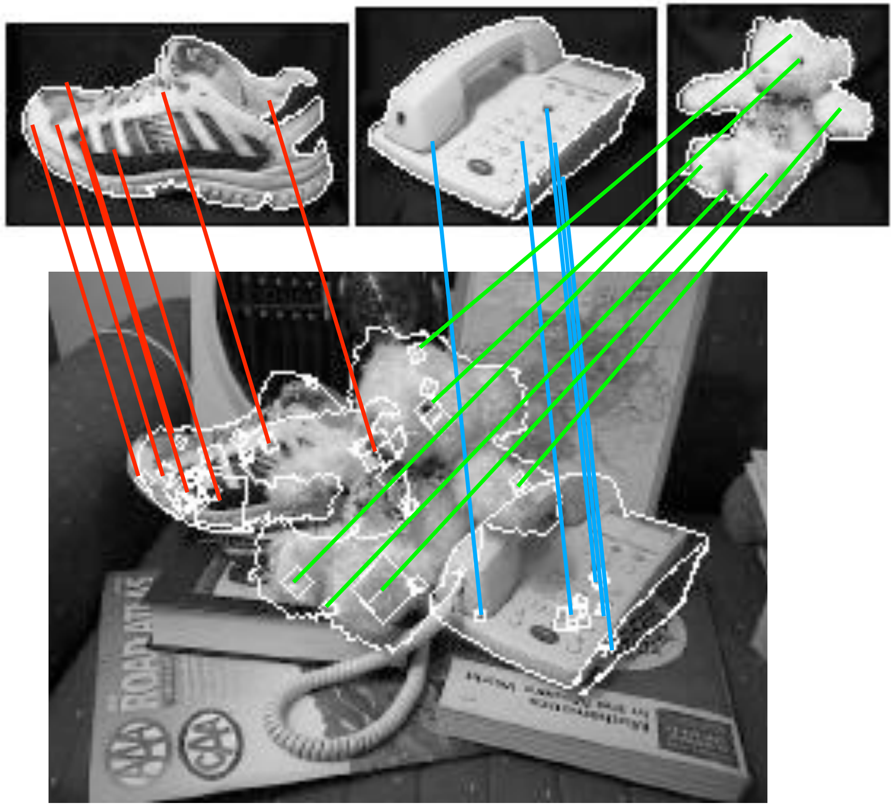
Useful in visual geometry: Camera calibration, 3D reconstruction, Structure and motion estimation, …

Scale/affine – invariant regions: SIFT, Harris-Laplace, etc.



Example II: Object recognition

Establish correspondence between the target image and (multiple) images in the model database.



Model database

Target image

[D. Lowe, 1999]

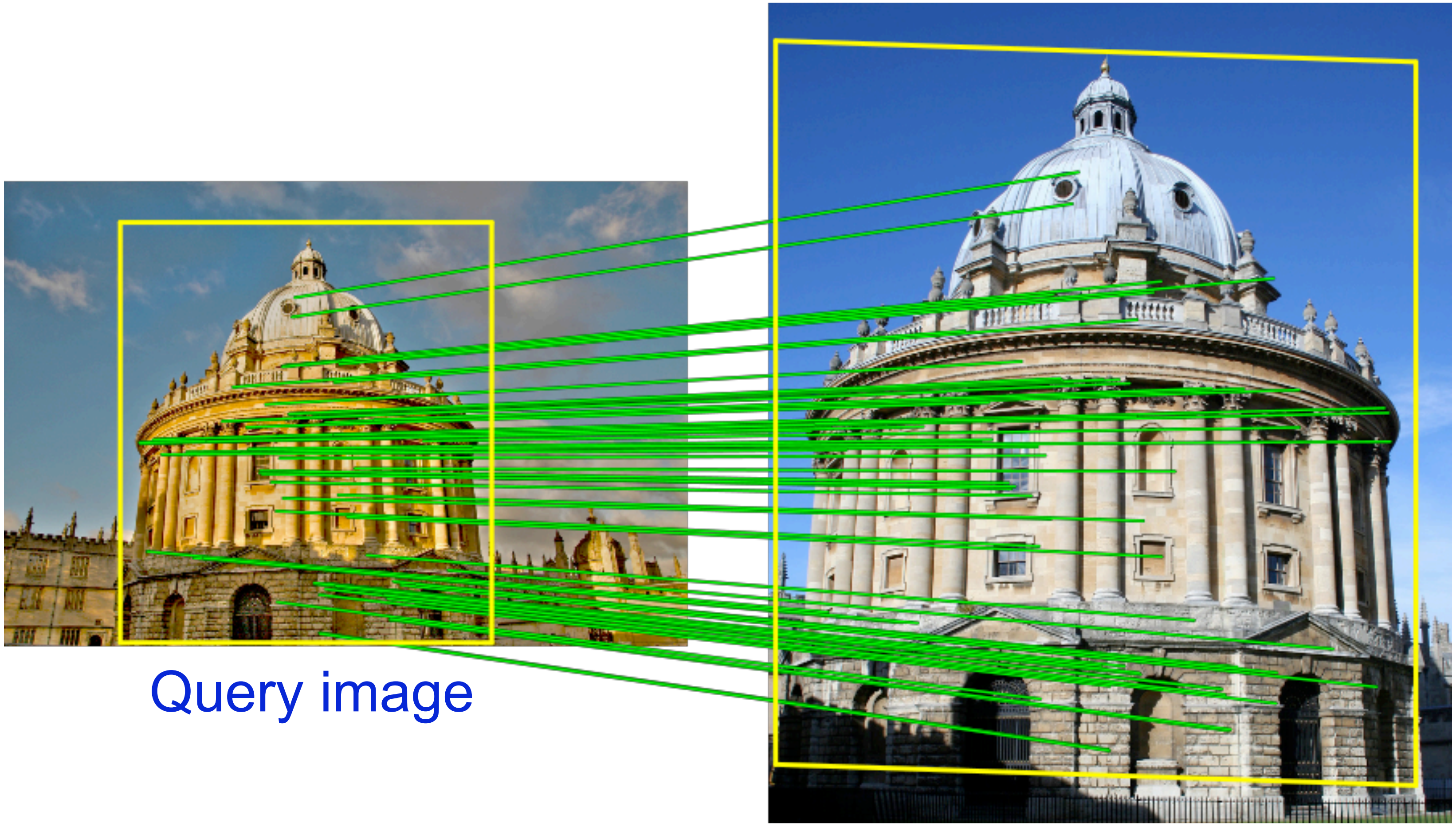
Example III: Visual search

Given a query image, find images depicting the same place / object in a large unordered image collection.

Find these landmarks

...in these images and 1M more

Establish correspondence between the query image and all images from the database depicting the same object / scene.

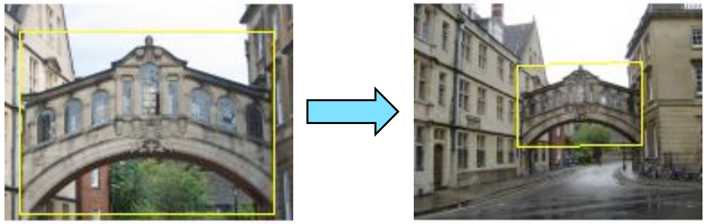


Query image

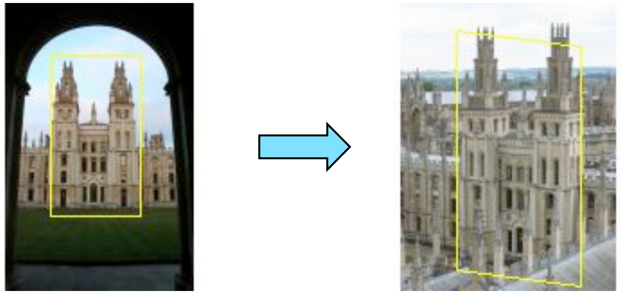
Database image(s)

Why is it difficult?

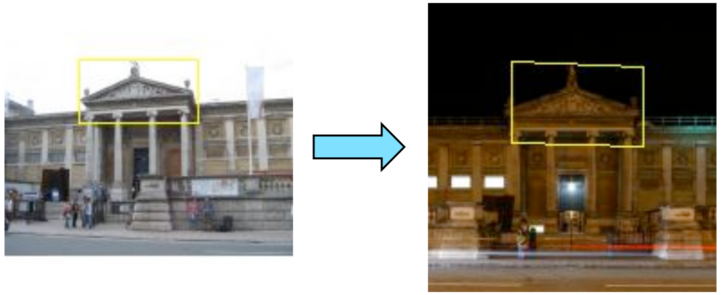
Want to establish correspondence despite possibly large changes in scale, viewpoint, lighting and partial occlusion



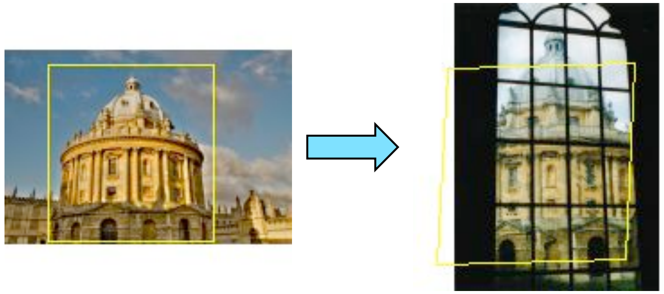
Scale



Viewpoint



Lighting



Occlusion

… and the image collection can be very large (e.g. 1M images)

Approach

Pre-processing (so far):

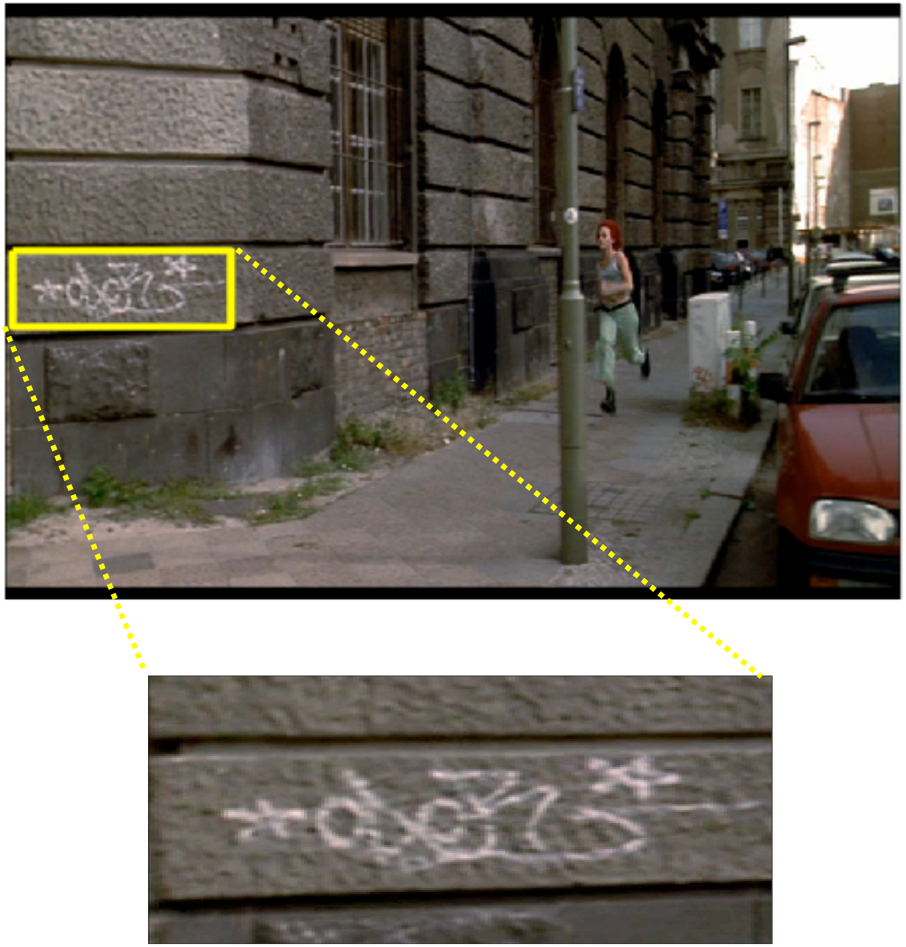
- Detect local features.
- Extract descriptor for each feature.

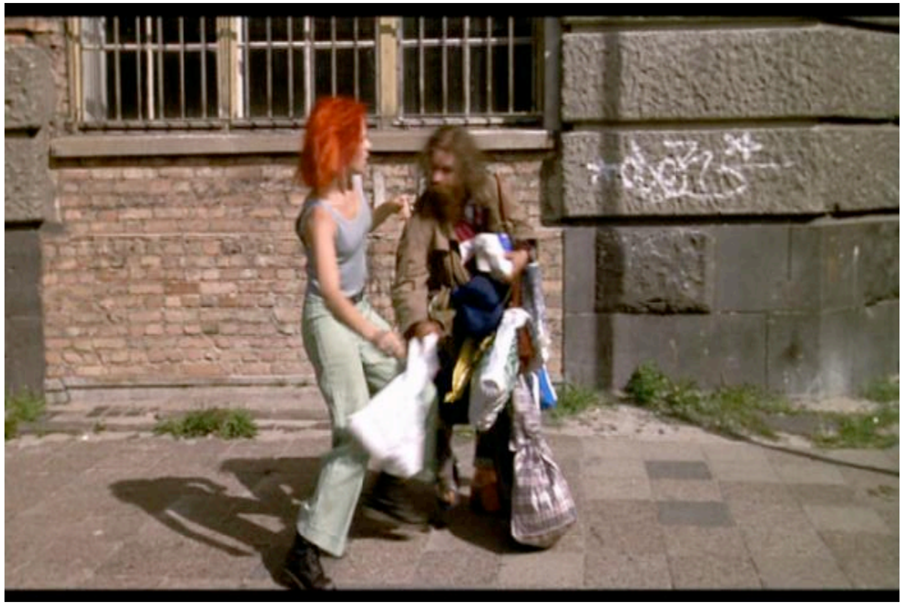
Matching:

1. Establish tentative (putative) correspondences based on local appearance of individual features (their descriptors).

2. Verify matches based on semi-local / global geometric relations.

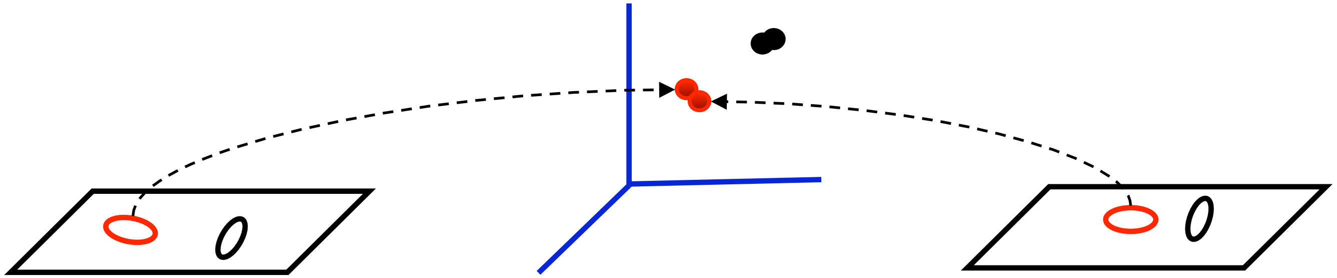
Example I: Two images -“Where is the Graffiti?”





Step 1. Establish tentative correspondence

Establish tentative correspondences between object model image and target image by nearest neighbour matching on SIFT vectors



Model (query) image

$\mathbf{x}_j \in \mathcal{R}^{128}$

128D descriptor space

Target image

$\mathbf{x}_i \in \mathcal{R}^{128}$

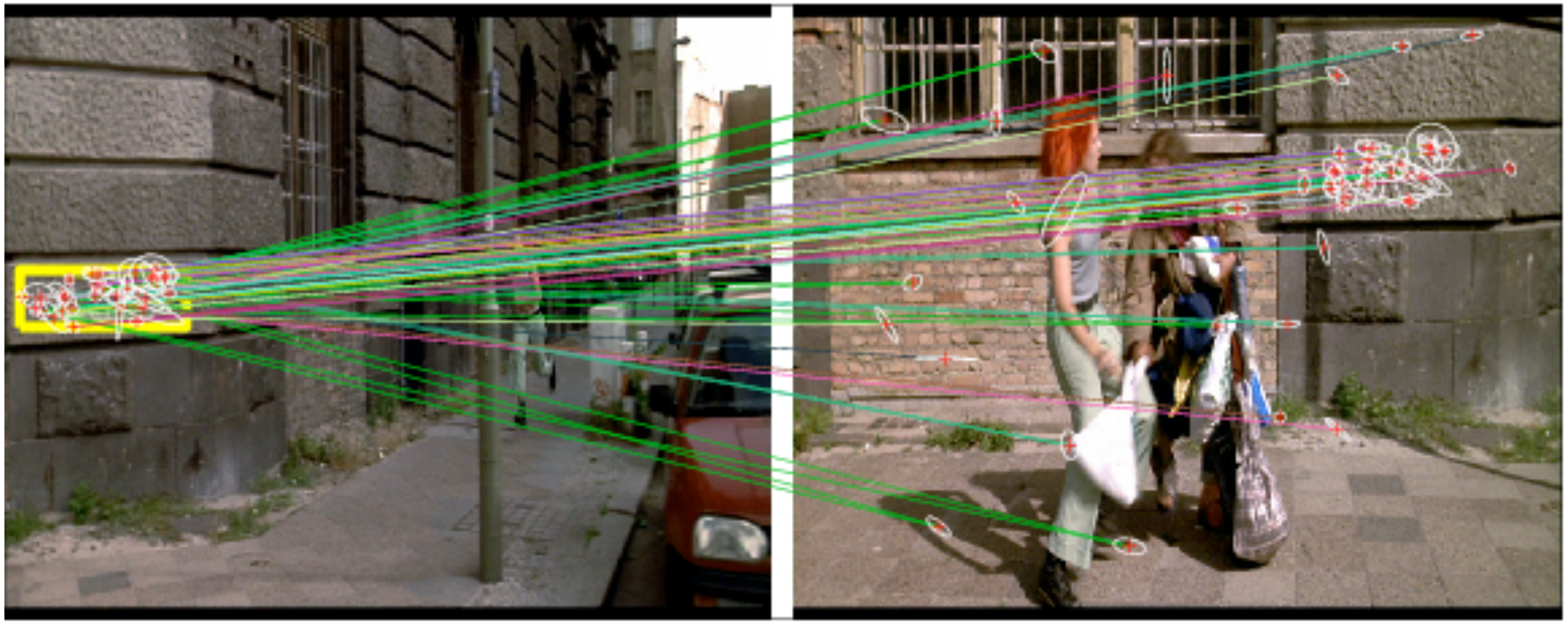
Need to solve some variant of the "nearest neighbor problem" for all feature vectors, $\mathbf{x}_j \in \mathcal{R}^{128}$, in the query image:

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

where, $\mathbf{x}_i \in \mathcal{R}^{128}$, are features in the target image.

Can take a long time if many target images are considered (see parts 3+4).

Problem with matching on local descriptors alone



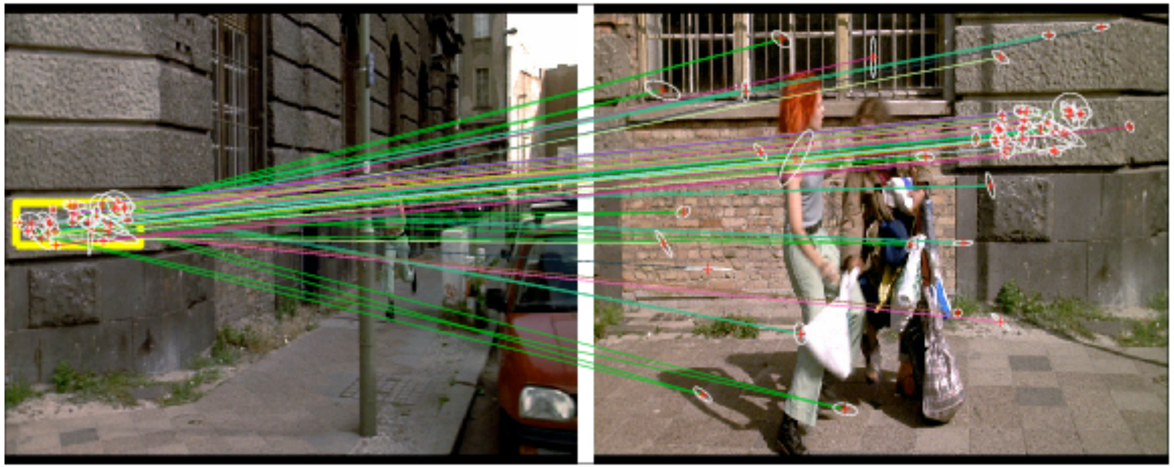
- too much individual invariance
- each region can affine deform independently (by different amounts)
- Locally appearance can be ambiguous

Solution: use semi-local and global spatial relations to verify matches.

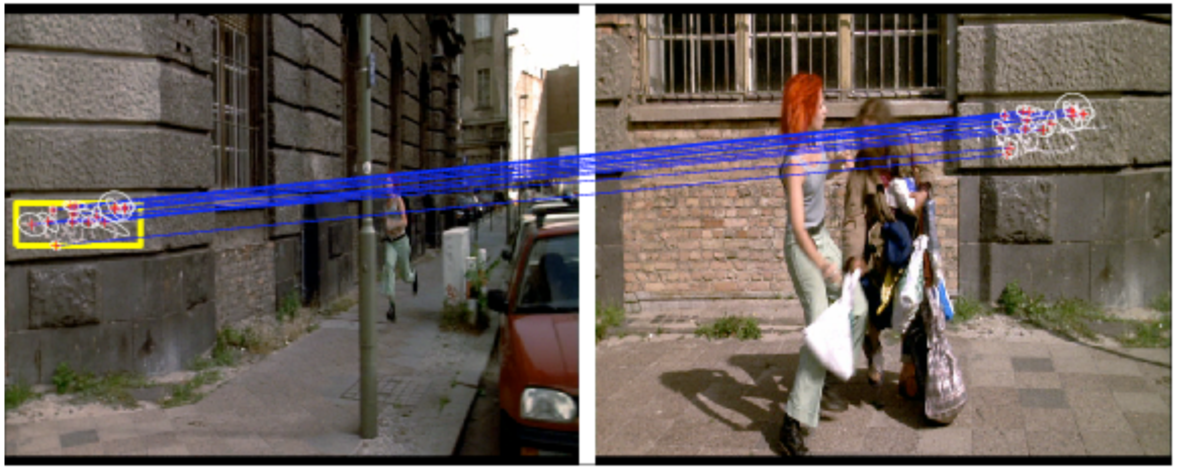
Example I: Two images -“Where is the Graffiti?”

Initial matches

Nearest-neighbor search based on appearance descriptors alone.



After spatial verification



Step 2: Spatial verification (the rest of part 2)

1. Semi-local constraints

   Constraints on spatially close-by matches

2. Global geometric relations

   Require a consistent global relationship between all matches

Semi-local constraints: Example I. – neighbourhood consensus

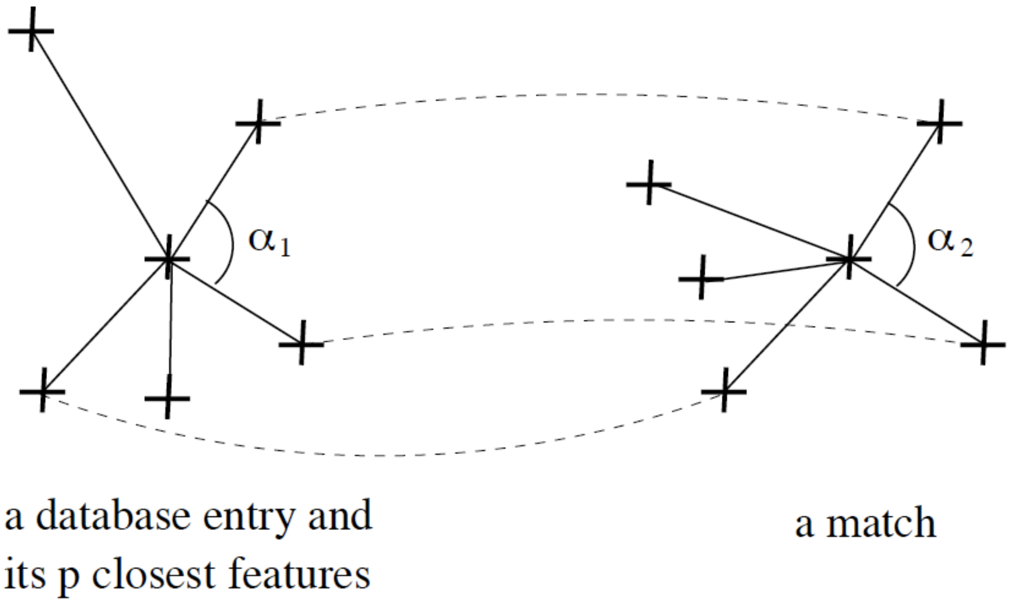
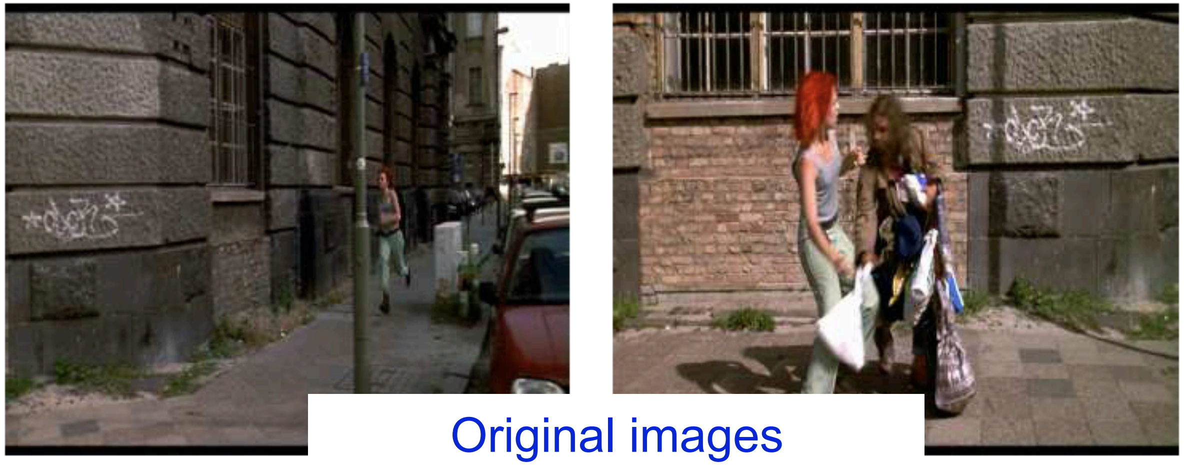


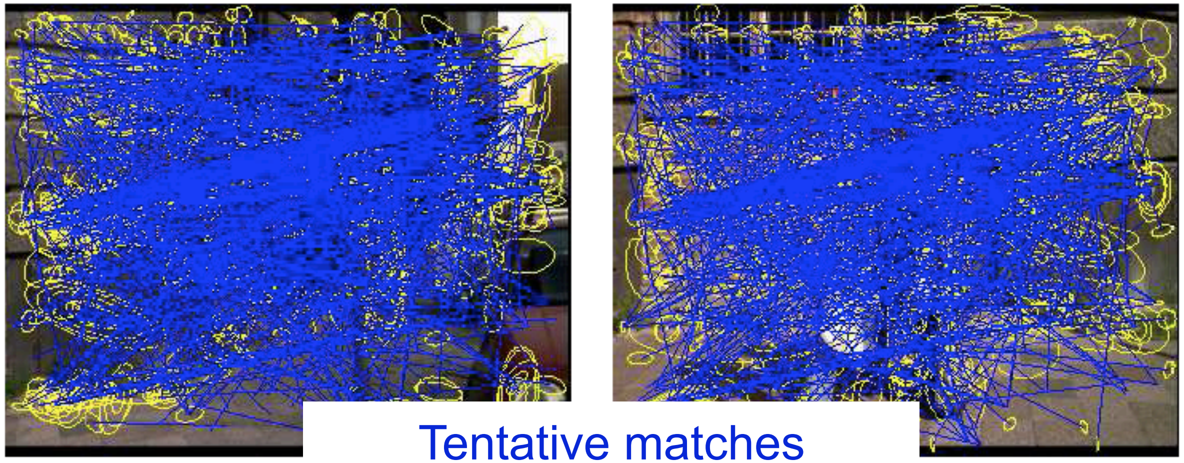
Fig. 4. Semi-local constraints : neighbours of the point have to match and angles have to correspond. Note that not all neighbours have to be matched correctly.

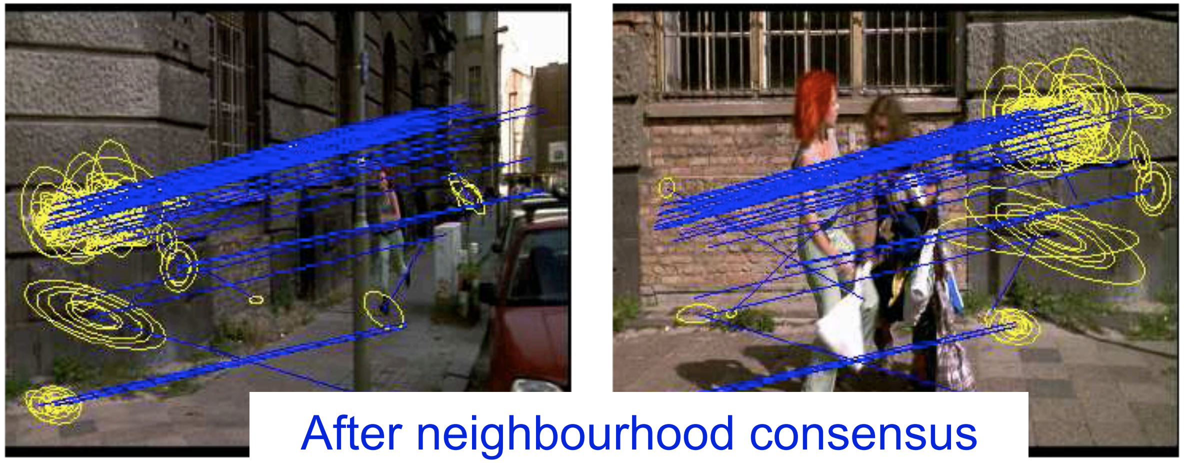
[Schmid&Mohr, PAMI 1997]

Semi-local constraints:

Example I. – neighbourhood consensus


Original images


Tentative matches


After neighbourhood consensus

[Schaffalitzky & Zisserman, CIVR 2004]

Semi-local constraints: Example II.

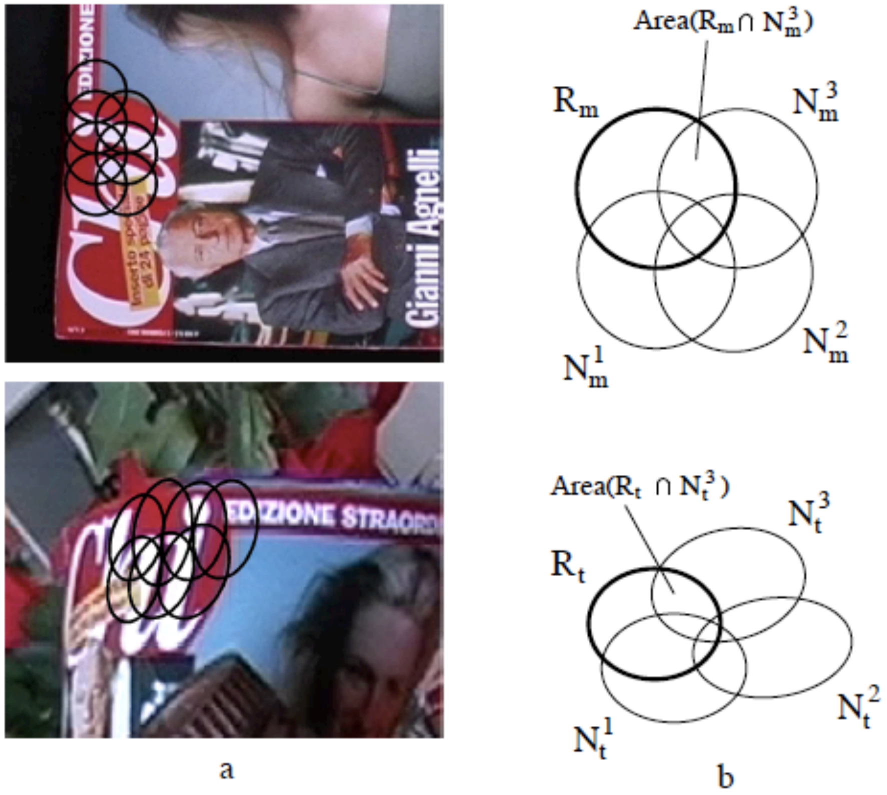
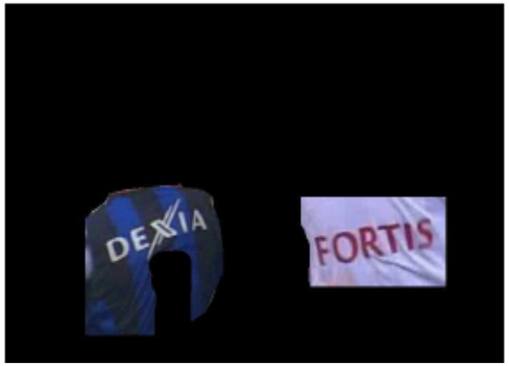
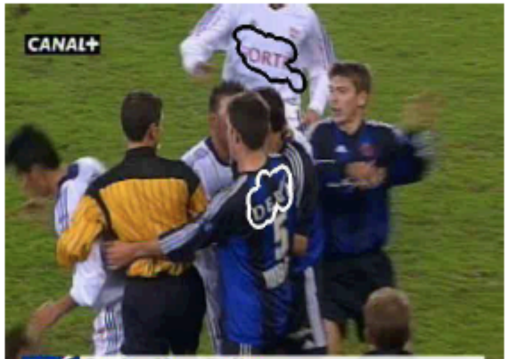


Figure 5: *Surface contiguity filter. a) the pattern of intersection between neighboring correct region matches is preserved by transformations between the model and the test images, because the surface is contiguous and smooth. b) the filter evaluates this property by testing the conservation of the area ratios.*

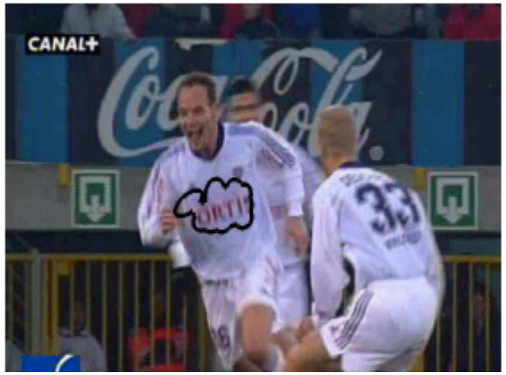
[Ferrari et al., IJCV 2005]



Model image



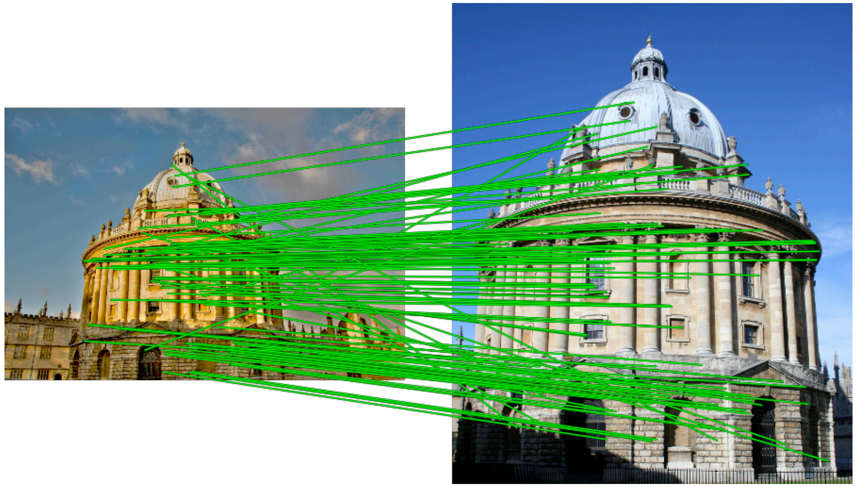
Matched image



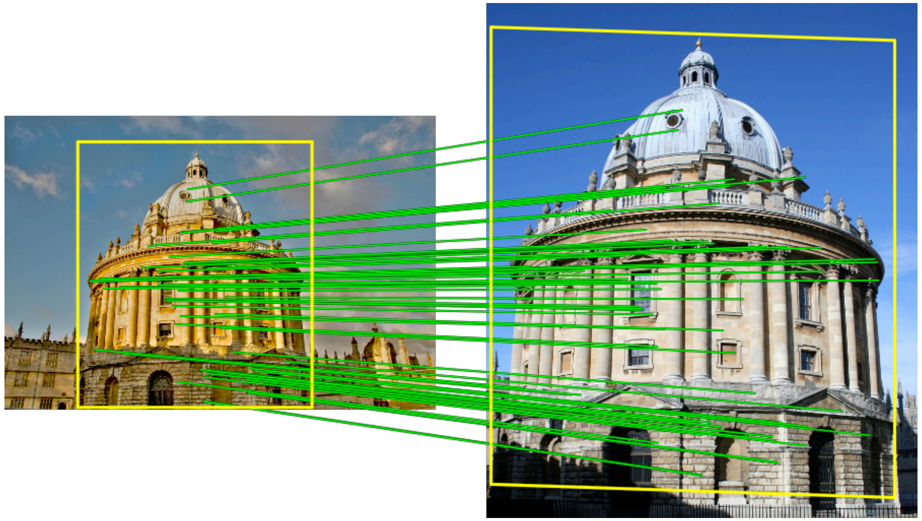
Matched image

Geometric verification with global constraints

- All matches must be consistent with a global geometric relation / transformation.
- Need to simultaneously (i) estimate the geometric relation / transformation and (ii) the set of consistent matches



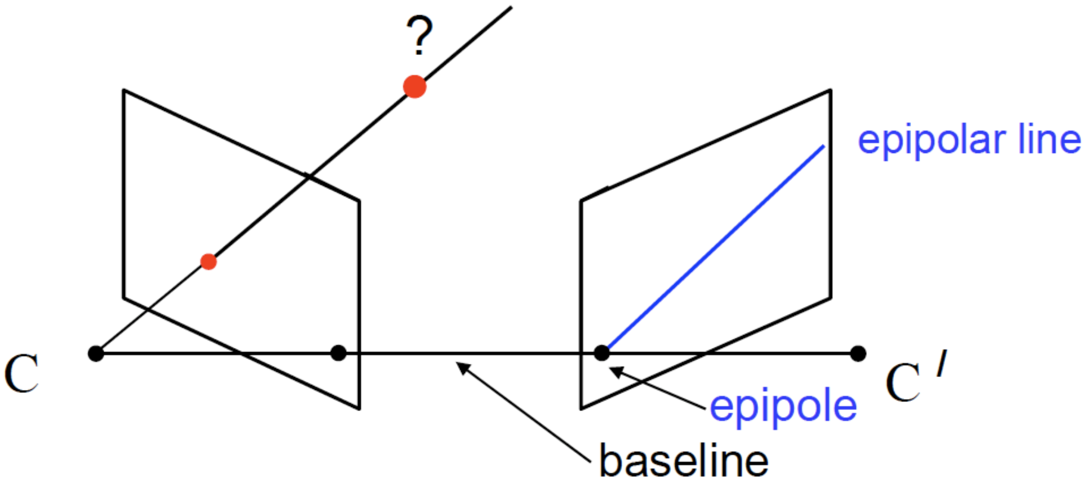
Tentative matches

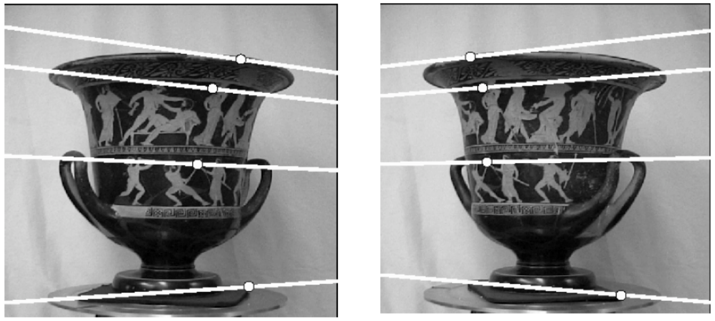


Matches consistent with an affine transformation

Epipolar geometry (not considered here)

In general, two views of a 3D scene are related by the epipolar constraint.





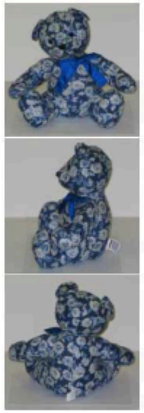
- A point in one view "generates" an epipolar line in the other view
- The corresponding point lies on this line.

Slide credit: A. Zisserman

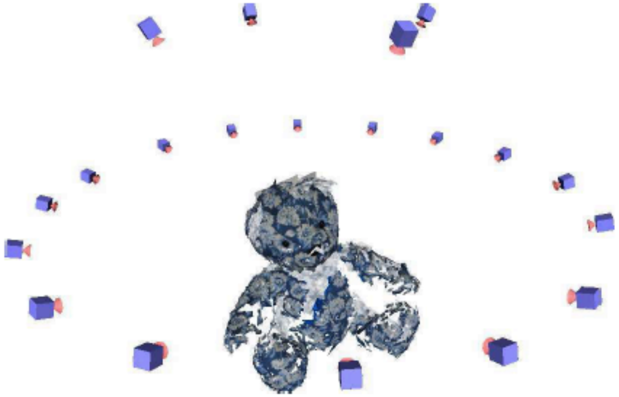
3D constraint: example (not considered here)

- Matches must be consistent with a 3D model

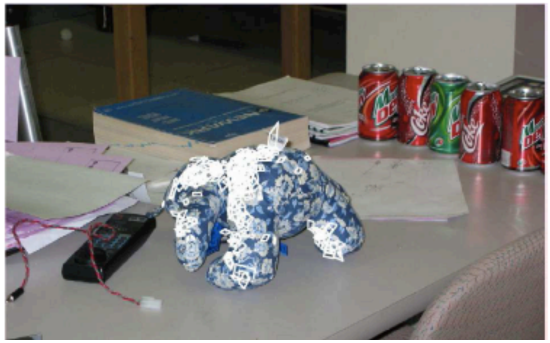
3 (out of 20) images used to build the 3D model



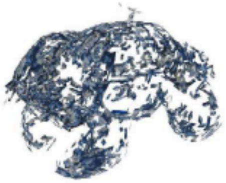
(a)



Recovered 3D model



Object recognized in a previously unseen pose



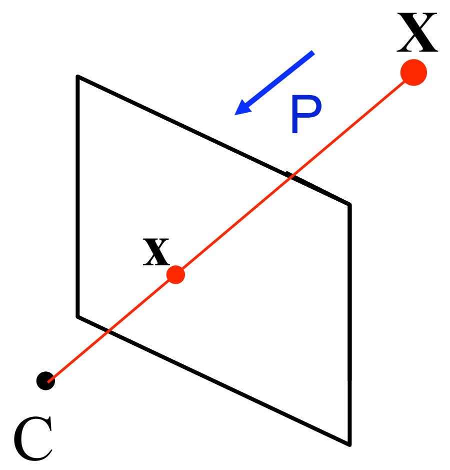
Recovered pose

(d)

[Lazebnik, Rothganger, Schmid, Ponce, CVPR'03]

3D constraint: example (not considered here)

With a given 3D model (set of known 3D points X's) and a set of measured 2D image points x, the goal is to find find camera matrix P and a set of geometrically consistent correspondences x↔X.



$$\mathbf{x} = P\mathbf{X}$$

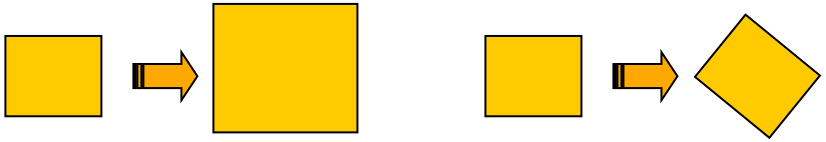
P : 3 × 4 matrix

$\mathbf{X}$: 4-vector

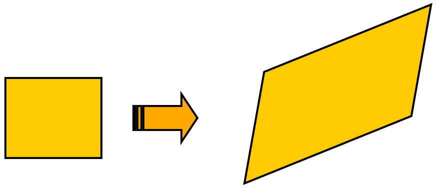
$\mathbf{x}$: 3-vector

2D transformation models

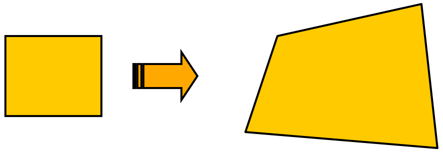
Similarity
(translation,
scale, rotation)



Affine

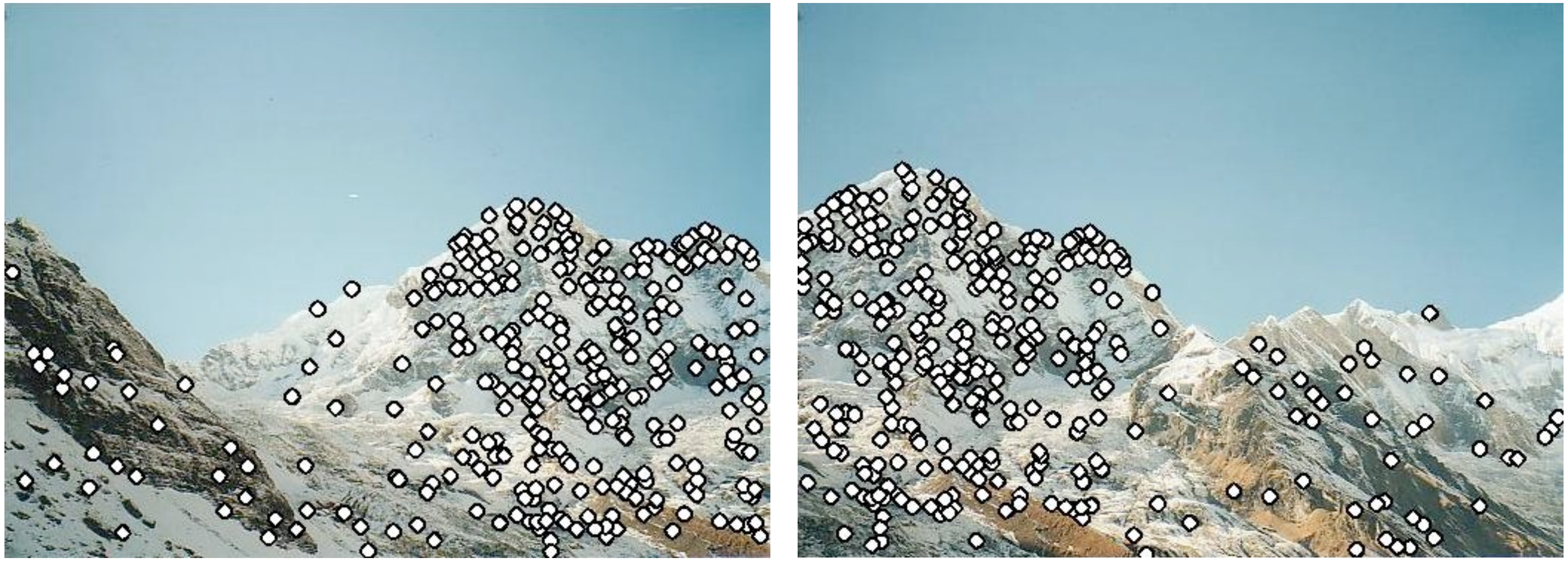


Projective
(homography)



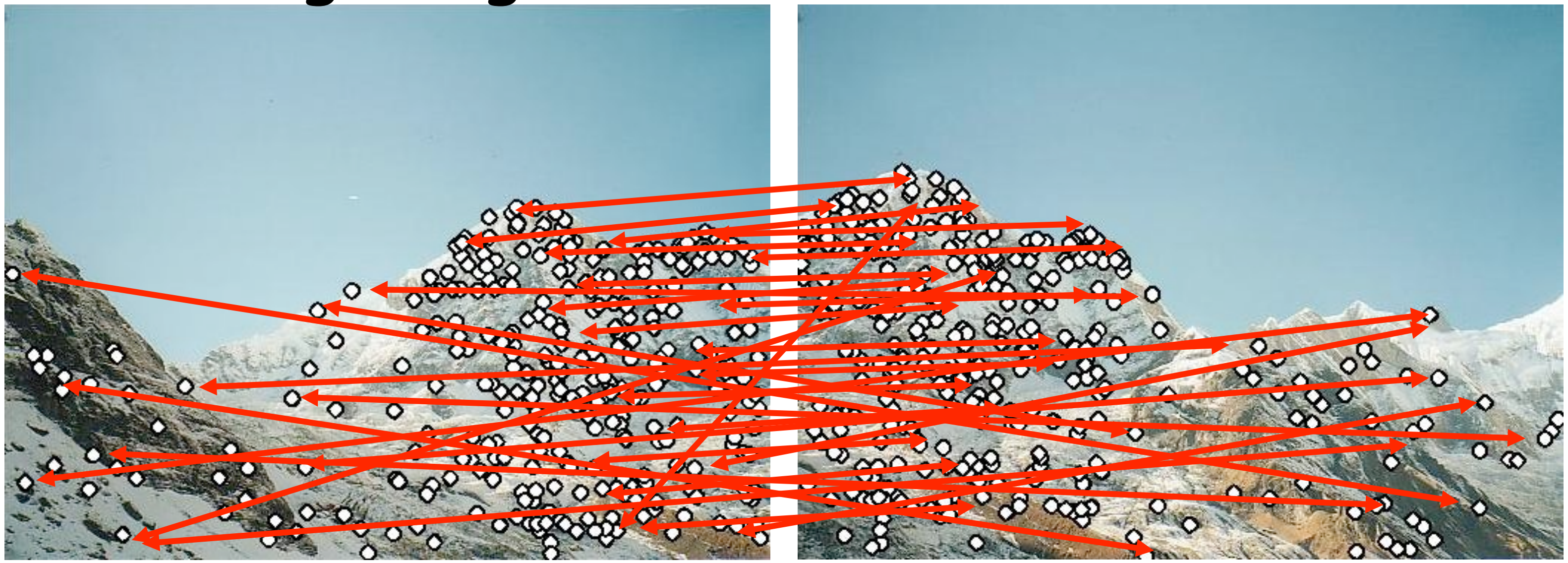
Estimating 2D geometric relations - outline

Estimating 2D geometric relations - outline



Extract features

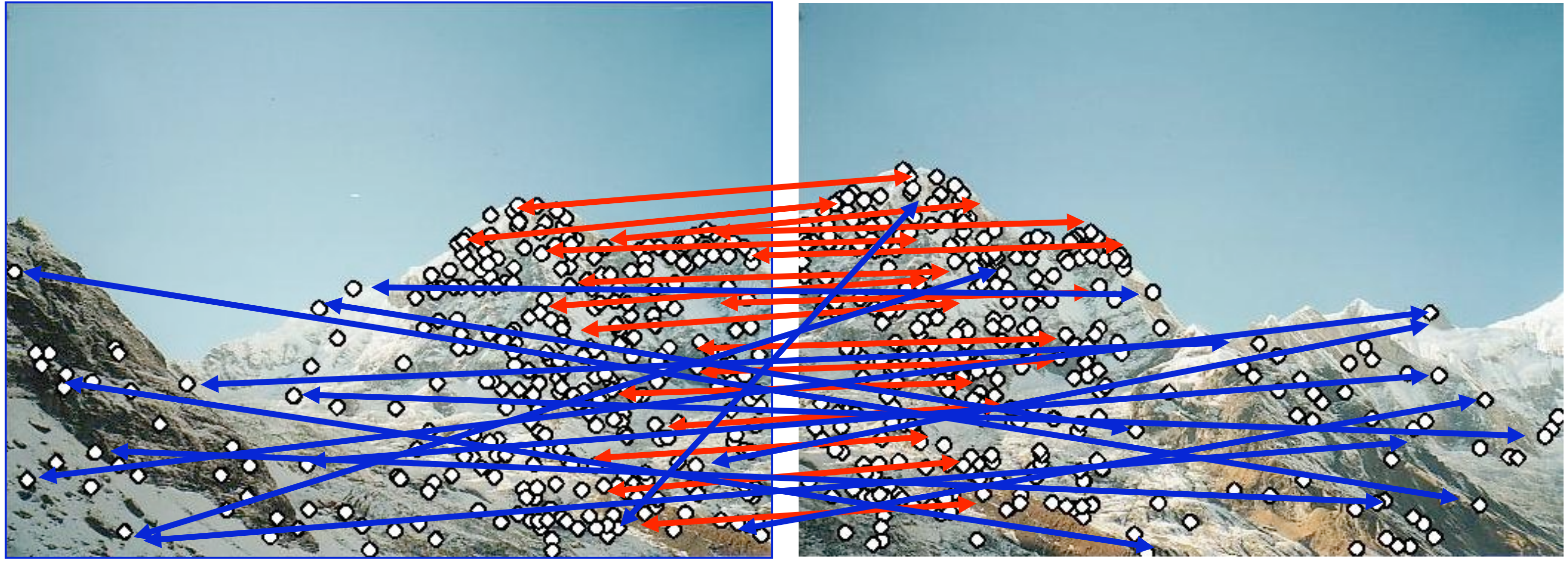
Estimating 2D geometric relations - outline



Extract features

Compute *putative matches*

Estimating 2D geometric relations - outline

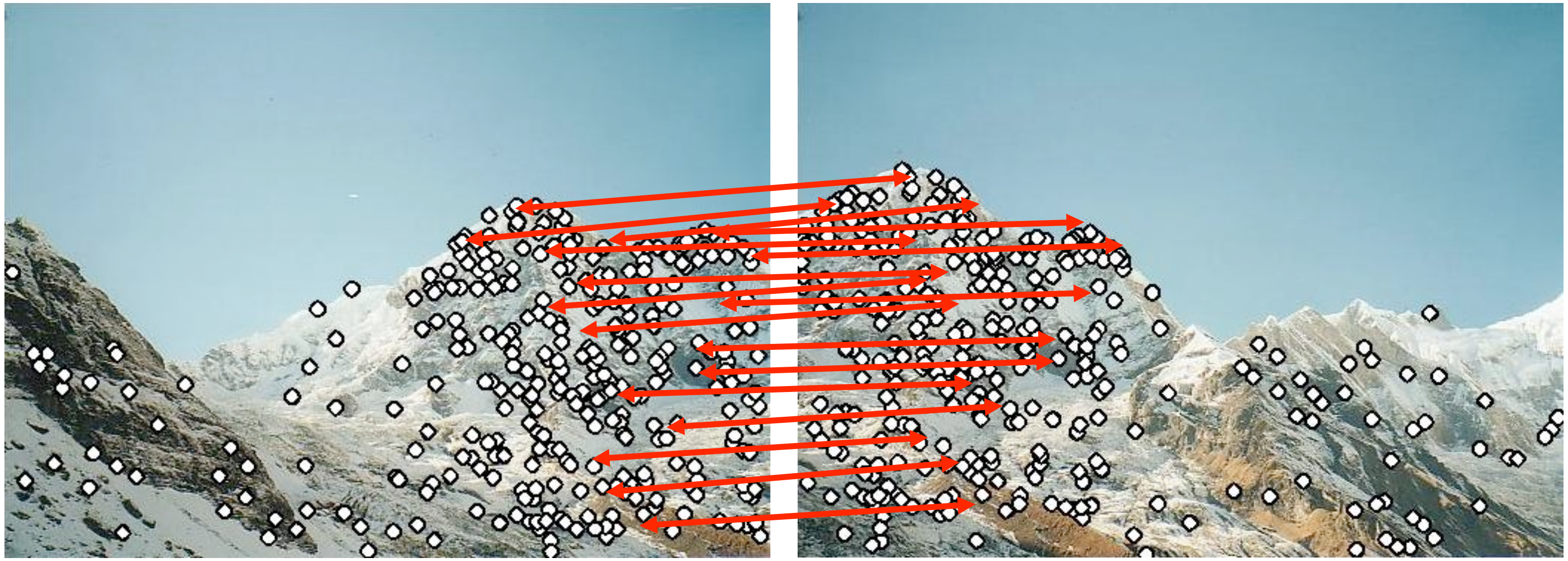


Extract features

Compute *putative matches*

Identify *outliers*

Estimating 2D geometric relations - outline



Extract features

Compute *putative matches*

Identify *outliers*

Estimate transformation from *inliers* (valid matches)

Estimating 2D geometric relations - outline

Extract features

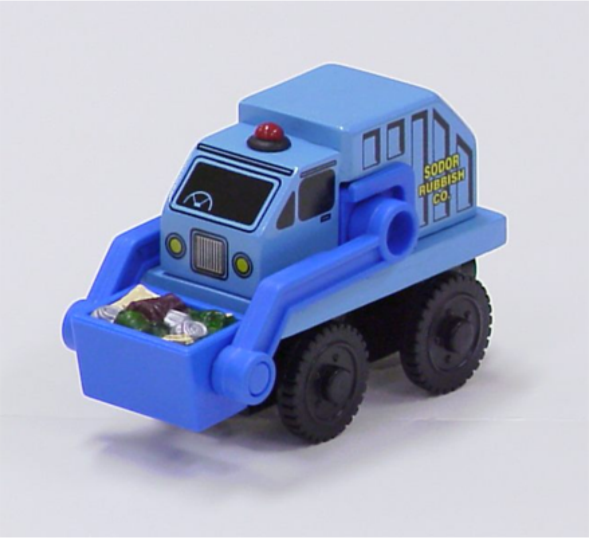
Compute *putative matches*

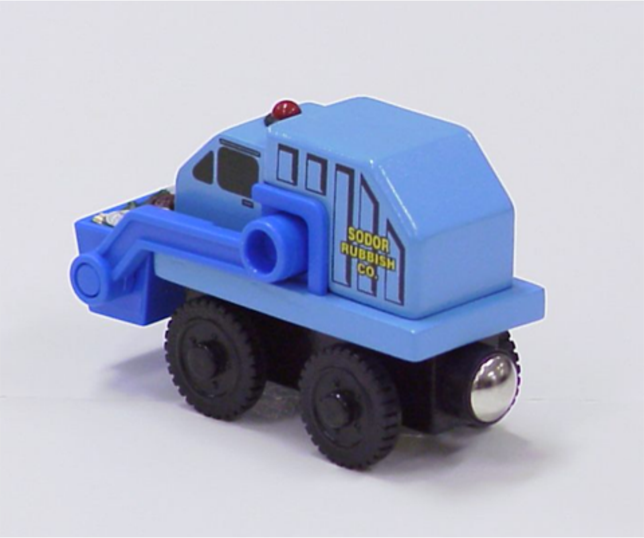
Identify *outliers*

Estimate transformation from *inliers* (valid matches)

Example: estimating 2D affine transformation

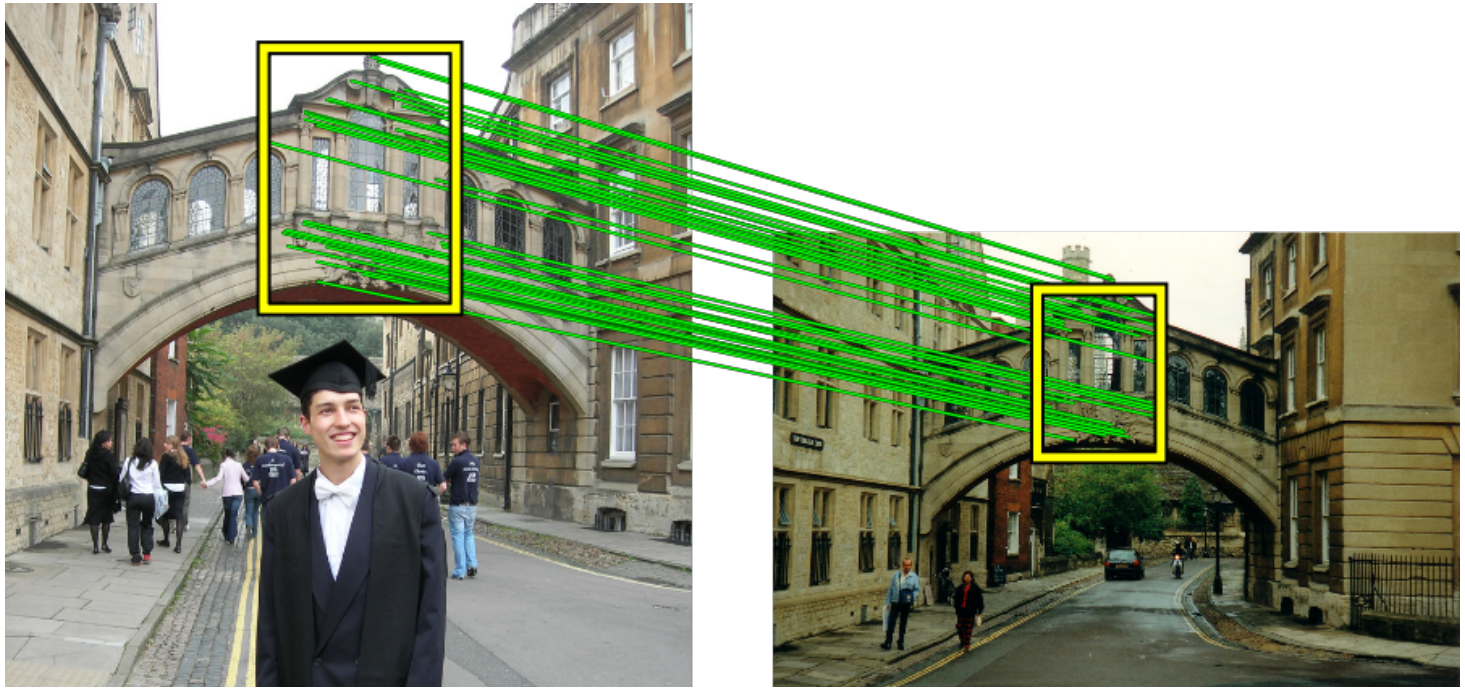
- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models





Example: estimating 2D affine transformation

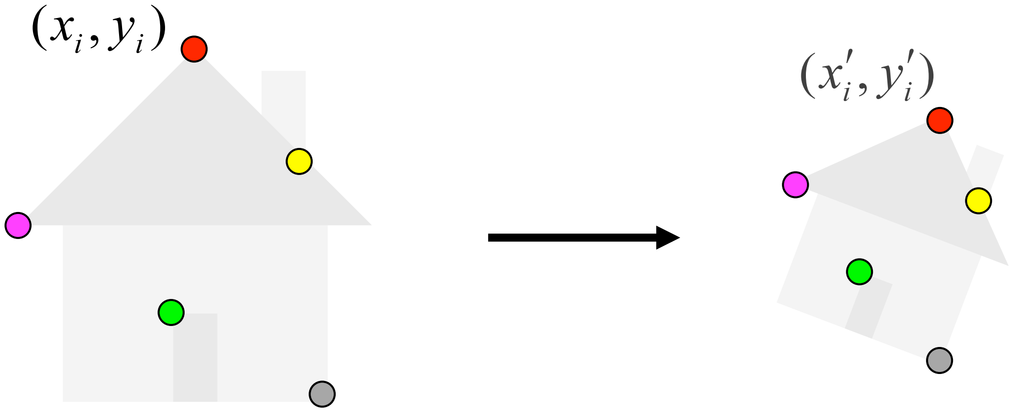
- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models



Matches consistent with an affine transformation

Fitting an affine transformation

Assume we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

Fitting an affine transformation

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x'_i \\ y'_i \\ \cdots \end{bmatrix}$$

Linear system with six unknowns

Each match gives us two linearly independent equations: need at least three to solve for the transformation parameters

Dealing with outliers

The set of putative matches may contain a high percentage (e.g. 90%) of outliers

How do we fit a geometric transformation to a small subset of all possible matches?

Possible strategies:

- RANSAC
- Hough transform

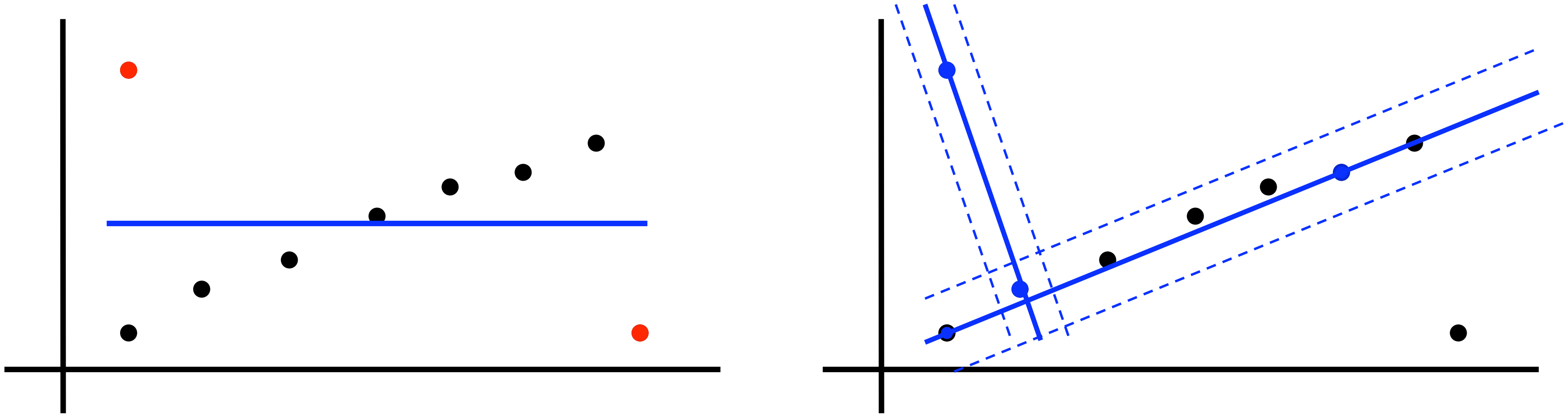
Strategy 1: RANSAC

RANSAC loop (Fischler & Bolles, 1981):

- Randomly select a *seed group* of matches
- Compute transformation from seed group
- Find *inliers* to this transformation
- If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
- Keep the transformation with the largest number of inliers

Example: Robust line estimation - RANSAC

Fit a line to 2D data containing outliers



There are two problems

1. a line fit which minimizes perpendicular distance
2. a classification into inliers (valid points) and outliers

Solution: use robust statistical estimation algorithm RANSAC

(RANdom Sample Consensus) [Fishler & Bolles, 1981]

Slide credit: A. Zisserman

RANSAC robust line estimation

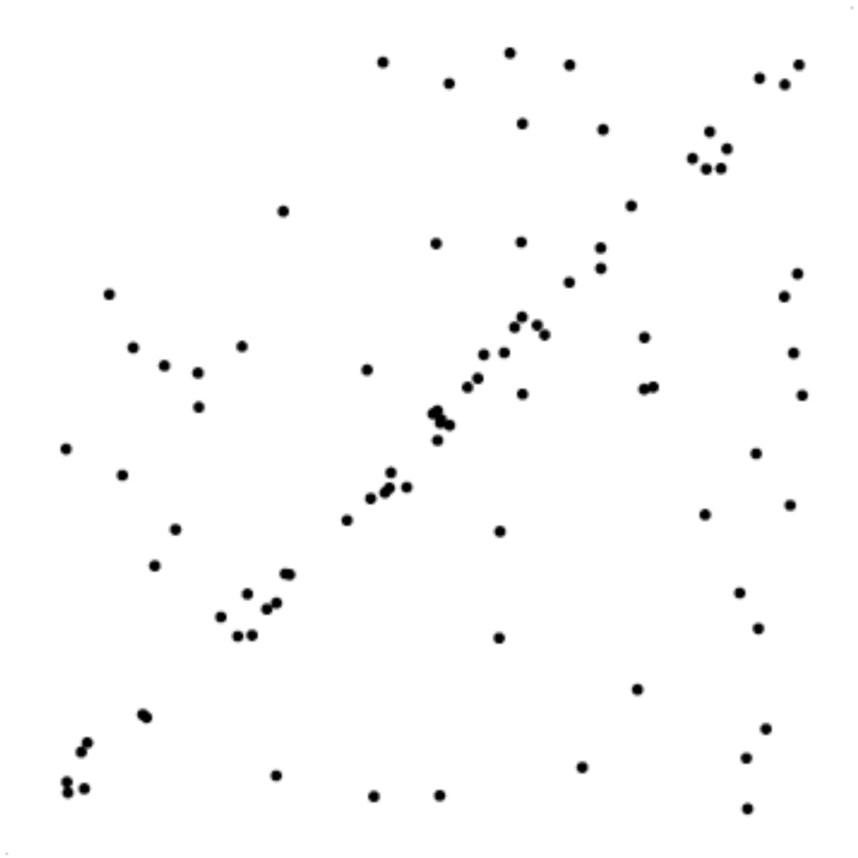
Repeat

1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

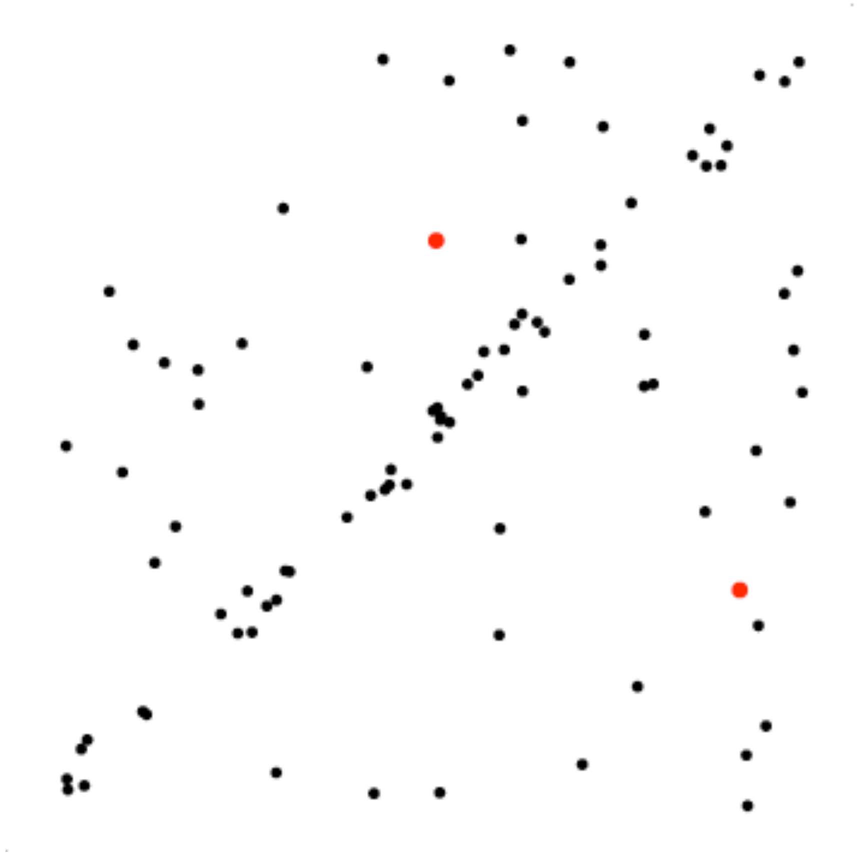
Choose the line with the largest number of inliers

- Compute least squares fit of line to inliers (regression)

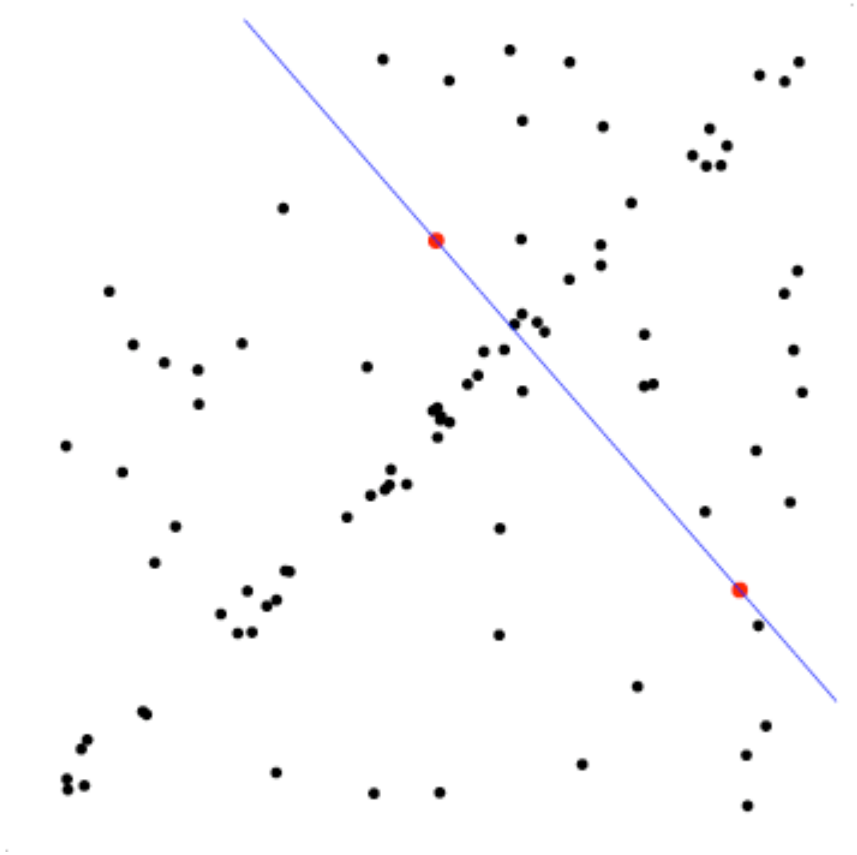
Slide credit: A. Zisserman



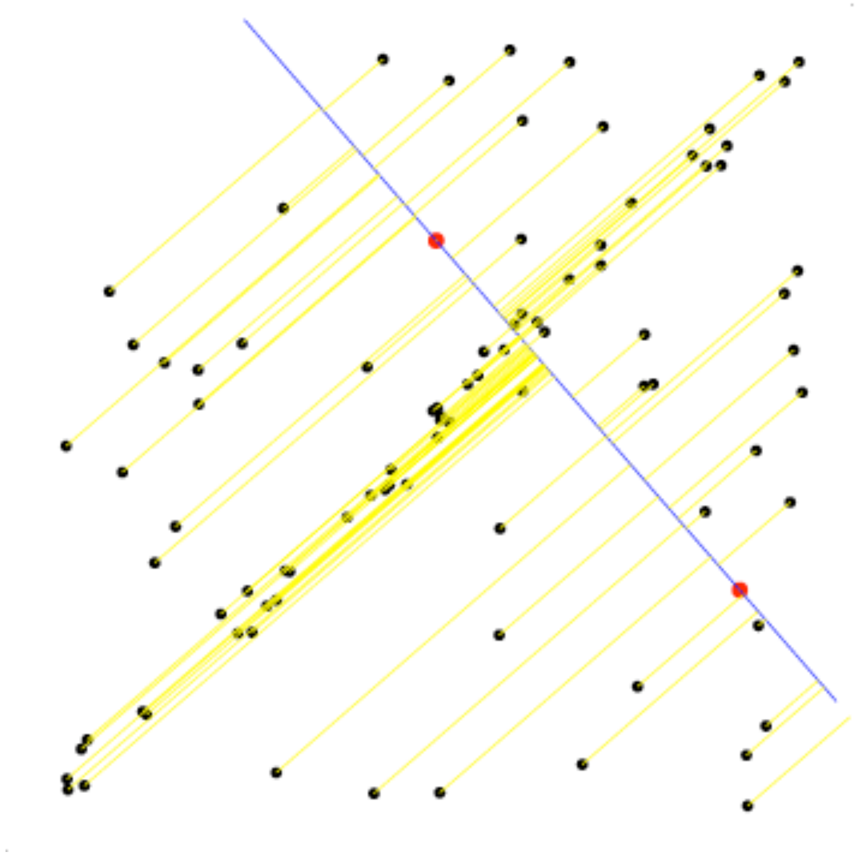
Slide credit: O. Chum



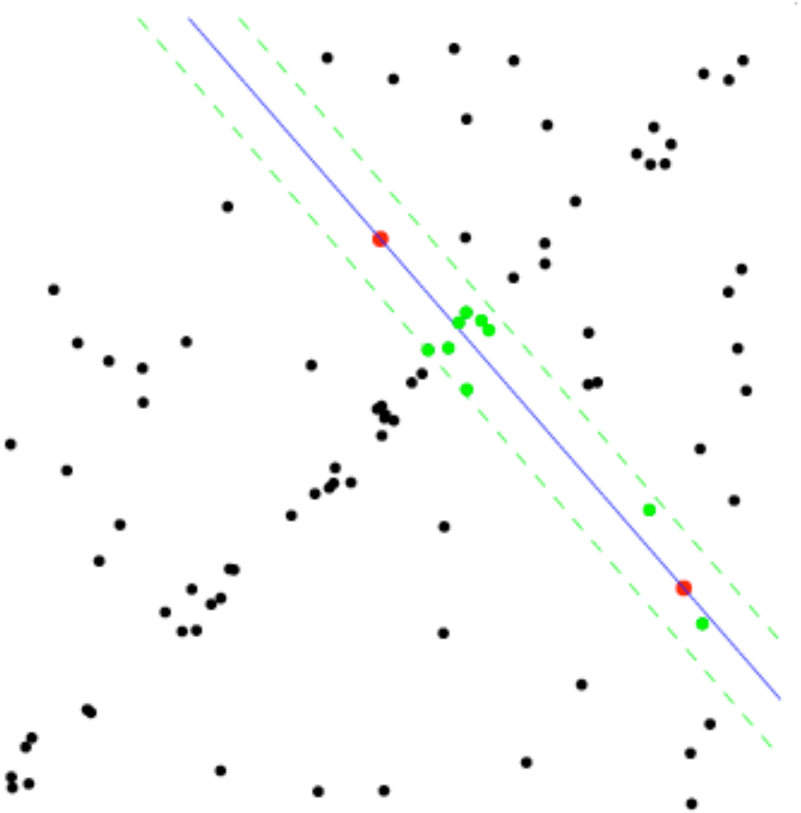
Slide credit: O. Chum



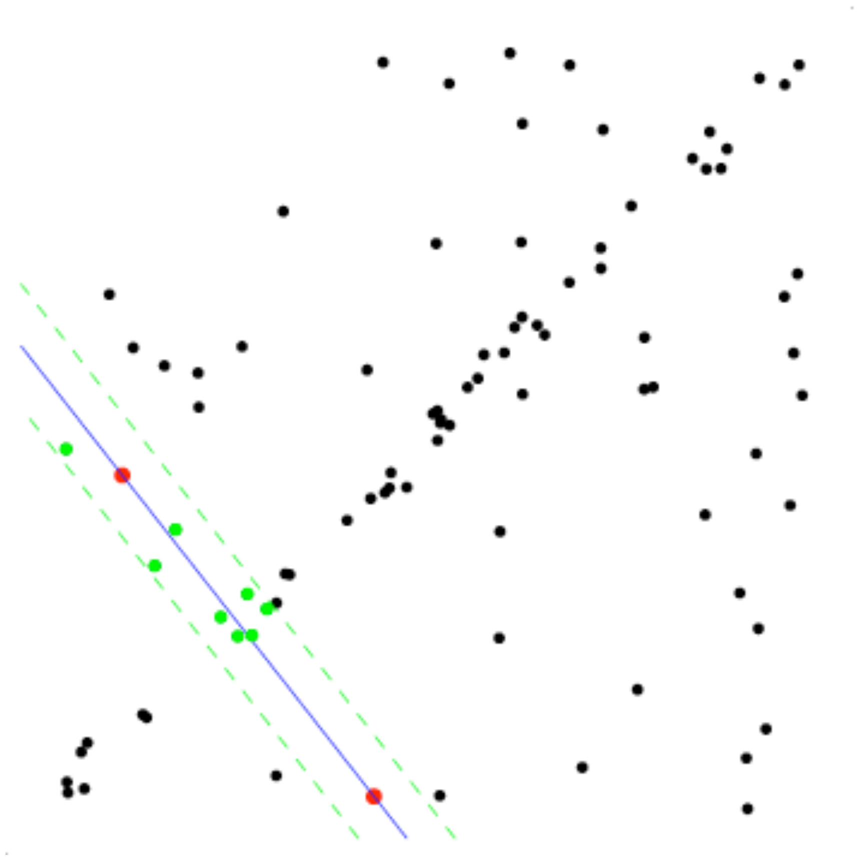
Slide credit: O. Chum



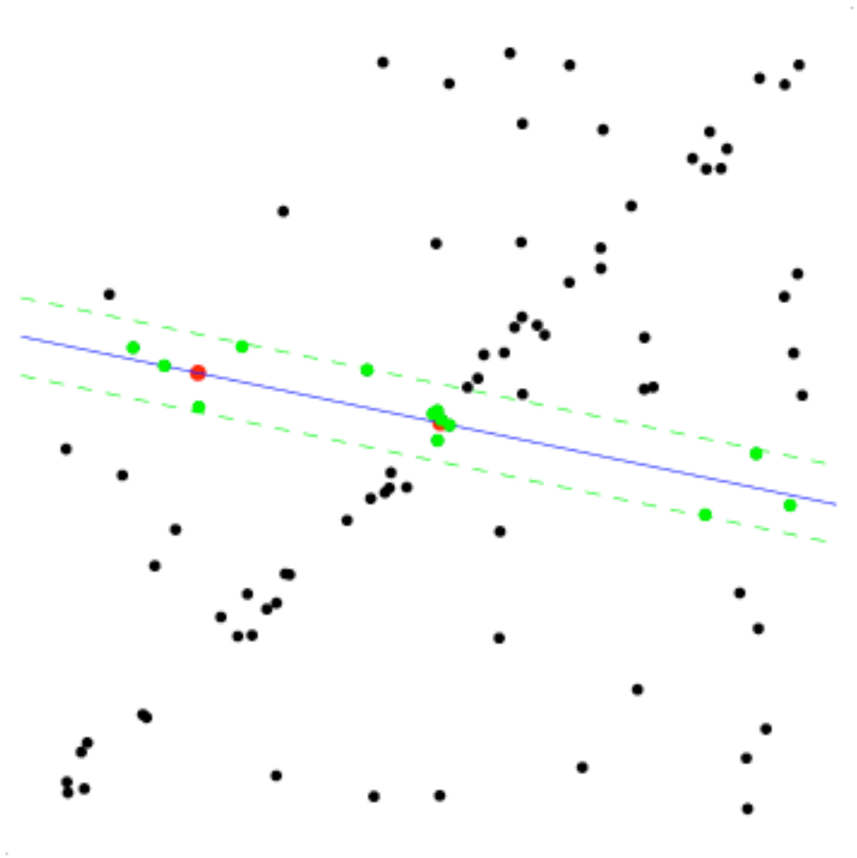
Slide credit: O. Chum



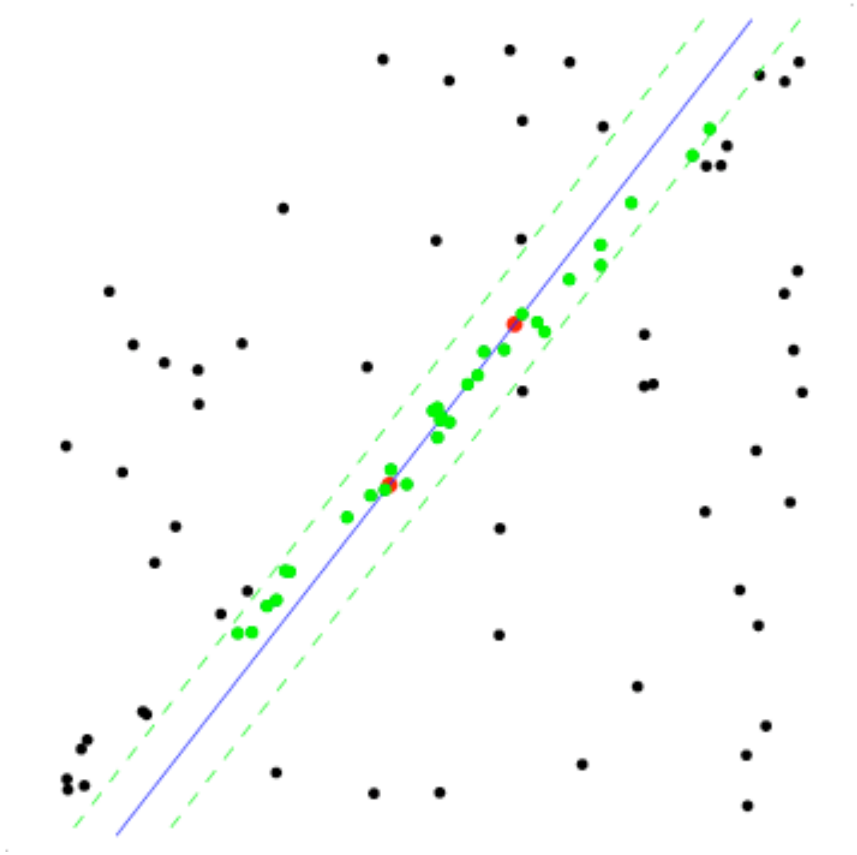
Slide credit: O. Chum



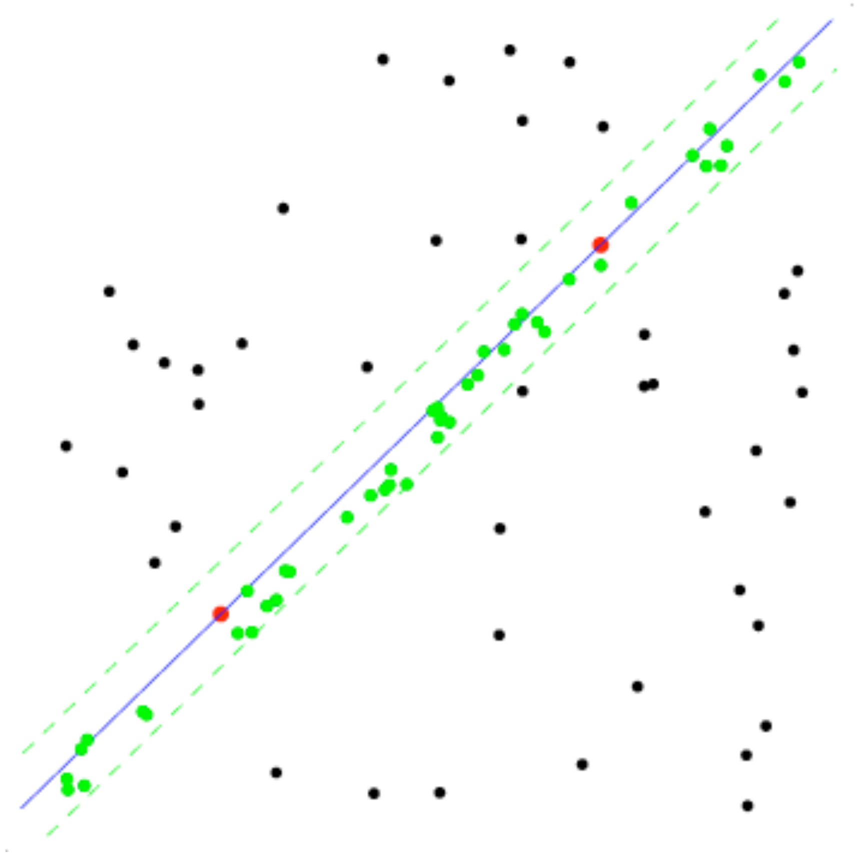
Slide credit: O. Chum



Slide credit: O. Chum



Slide credit: O. Chum



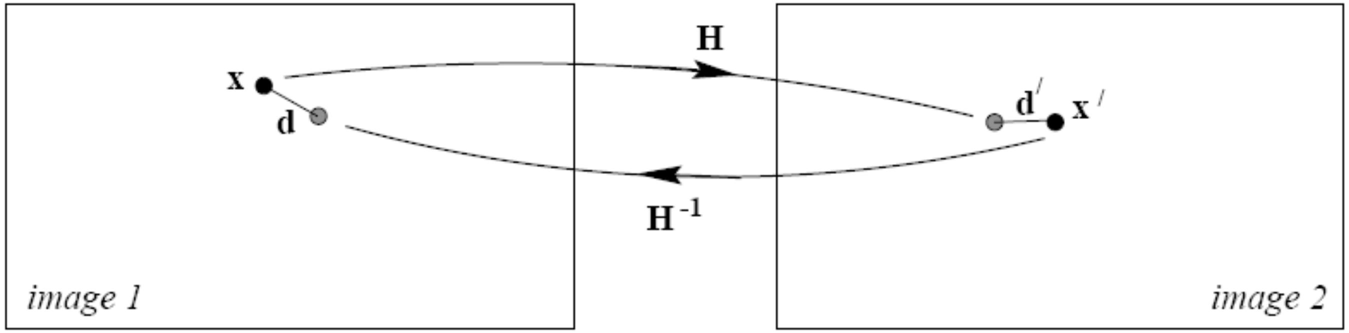
Slide credit: O. Chum

Algorithm summary – RANSAC robust estimation of 2D affine transformation

Repeat

1. Select 3 point to point correspondences
2. Compute H (2x2 matrix) + t (2x1) vector for translation
3. Measure support (number of inliers within threshold distance, i.e. $d^2_{\text{transfer}} < t$)

$$d^2_{\text{transfer}} = d(\mathbf{x}, \mathtt{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathtt{H}\mathbf{x})^2$$



Choose the (H,t) with the largest number of inliers

(Re-estimate (H,t) from all inliers)

How many samples?

Number of samples *N*

- Choose *N* so that, with probability *p*, at least one random sample is free from outliers
- e.g.:
  - > *p*=0.99
  - > outlier ratio: *e*

$$\left(1-\underbrace{\overbrace{(1-e)}^{\text{Probability a randomly picked point is an inlier}}{}^{s}}_{\text{Probability of all points in a sample (of size s) are inliers}}\right)^{N}=1-p$$

Source: M. Pollefeys

How many samples?

Number of samples *N*

- Choose *N* so that, with probability *p*, at least one random sample is free from outliers
- e.g.:
  - > p=0.99
  - > outlier ratio: e

$$\overbrace{\left(\underbrace{1-\left(1-e\right)^s}_{\text{Probability of at least one point in a sample (of size s) is an outlier}}\right)^N}^{\text{Probability that all N samples (of size s) are corrupted (contain an outlier)}} = 1-p$$

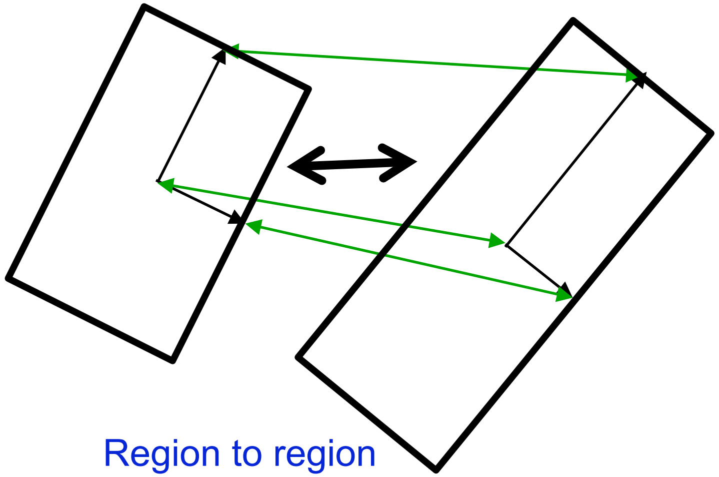
$$N = \log\left(1-p\right)/\log\left(1-\left(1-e\right)^s\right)$$

	proportion of outliers e						
s	5%	10%	20%	30%	40%	50%	90%
1	2	2	3	4	5	6	43
2	2	3	5	7	11	17	458
3	3	4	7	11	19	35	4603
4	3	5	9	17	34	72	4.6e4
5	4	6	12	26	57	146	4.6e5
6	4	7	16	37	97	293	4.6e6
7	4	8	20	54	163	588	4.6e7
8	5	9	26	78	272	1177	4.6e8

Source: M. Pollefeys

How to reduce the number of samples needed?

1. Reduce the proportion of outliers.
2. Reduce the sample size
   - use simpler model (e.g. similarity instead of affine tnf.)
   - use local information (e.g. a region to region correspondence is equivalent to (up to) 3 point to point correspondences).



Region to region correspondence

Number of samples *N*

	proportion of outliers e						
s	5%	10%	20%	30%	40%	50%	90%
1	2	2	3	4	5	6	43
2	2	3	5	7	11	17	458
3	3	4	7	11	19	35	4603
4	3	5	9	17	34	72	4.6e4
5	4	6	12	26	57	146	4.6e5
6	4	7	16	37	97	293	4.6e6
7	4	8	20	54	163	588	4.6e7
8	5	9	26	78	272	1177	4.6e8

RANSAC (references)

M. Fischler and R. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Comm. ACM, 1981

R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., 2004.

Extensions:

B. Tordoff and D. Murray, “Guided Sampling and Consensus for Motion Estimation, ECCV’03

D. Nister, “Preemptive RANSAC for Live Structure and Motion Estimation, ICCV’03

Chum, O.; Matas, J. and Obdrzalek, S.: Enhancing RANSAC by Generalized Model Optimization, ACCV’04

Chum, O.; and Matas, J.: Matching with PROSAC - Progressive Sample Consensus , CVPR 2005

Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching, CVPR’07

Chum, O. and Matas. J.: Optimal Randomized RANSAC, PAMI’08

Strategy 2: Hough Transform

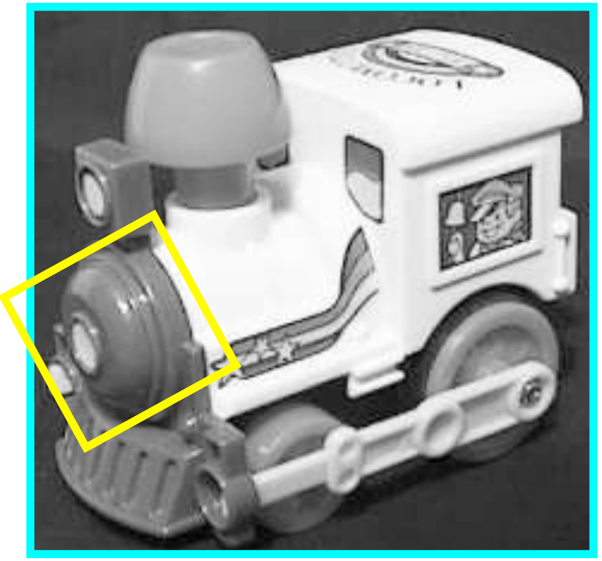
- Origin: Detection of straight lines in cluttered images
- Can be generalized to arbitrary shapes
- Can extract feature groupings from cluttered images in linear time.
- Illustrate on extracting sets of local features consistent with a similarity transformation.

Hough transform for object recognition

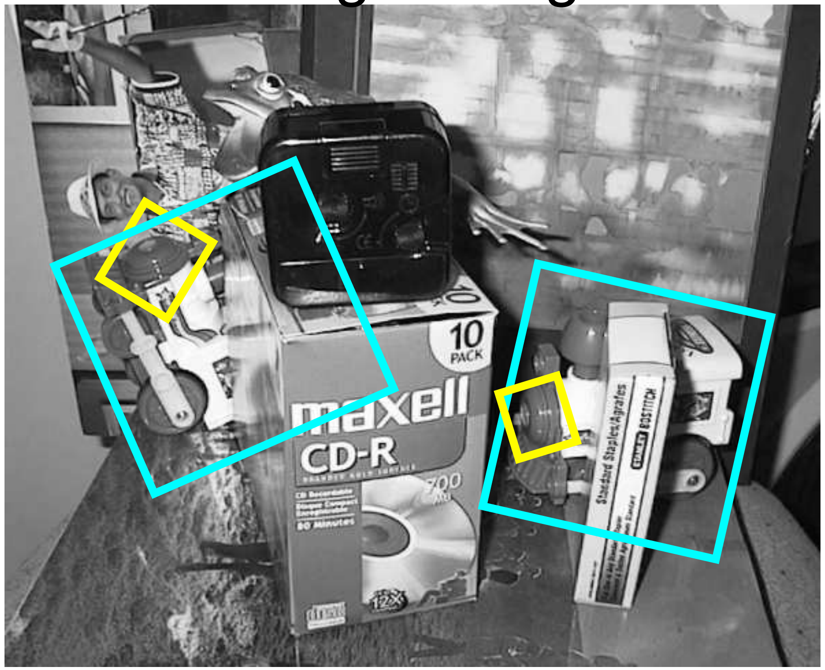
Suppose our features are scale- and rotation-covariant

- Then a single feature match provides an alignment hypothesis (translation, scale, orientation)

model



Target image

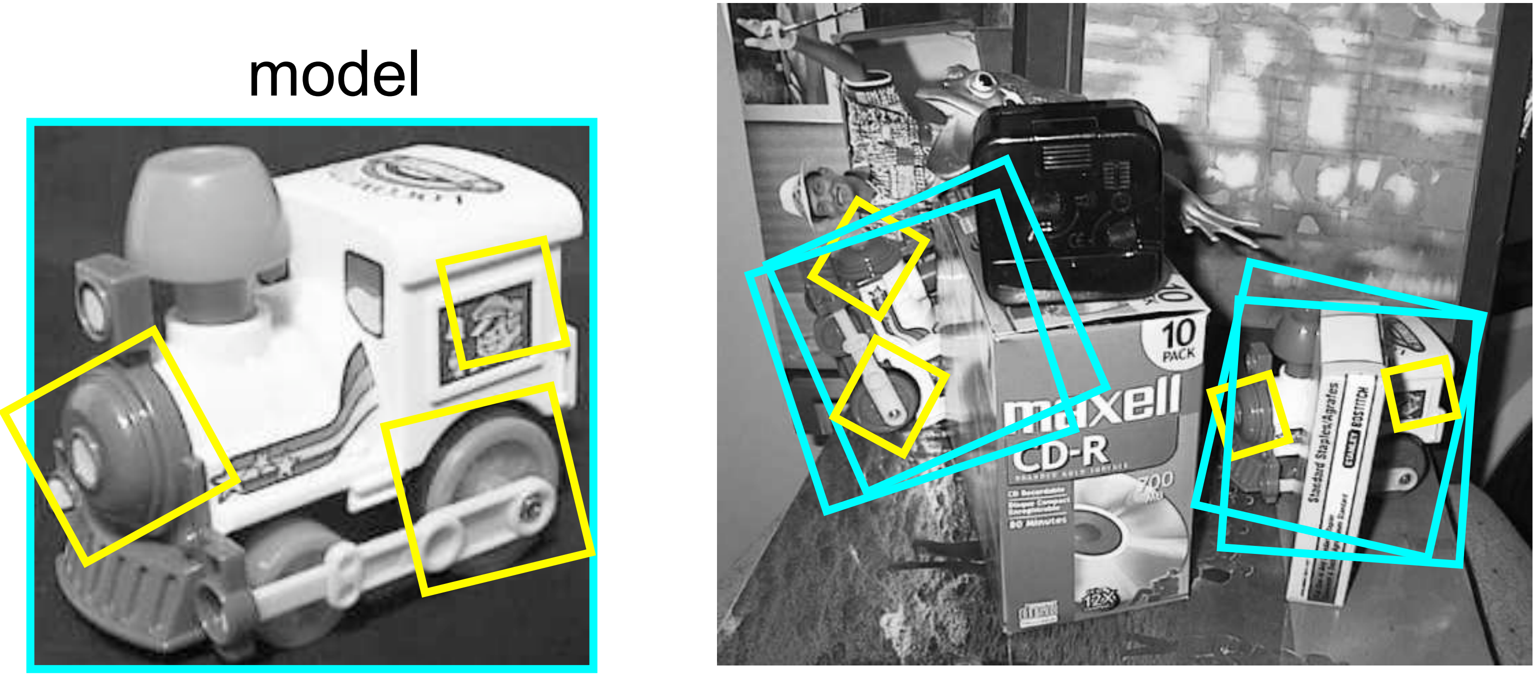


David G. Lowe. **“Distinctive image features from scale-invariant keypoints”,** *IJCV* 60 (2), pp. 91-110, 2004.

Hough transform for object recognition

Suppose our features are scale- and rotation-covariant

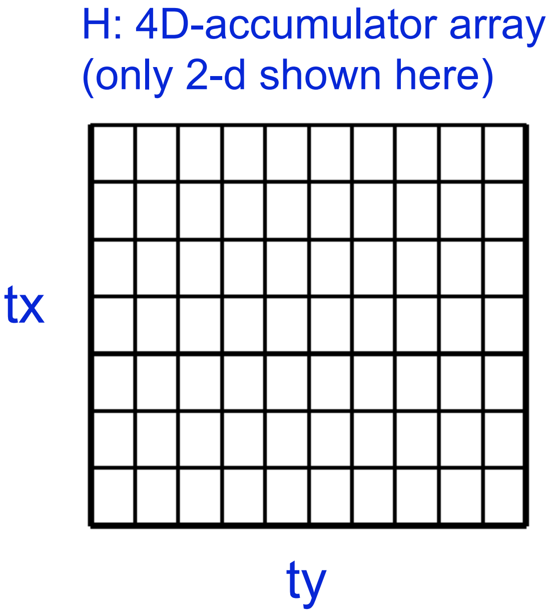
- Then a single feature match provides an alignment hypothesis (translation, scale, orientation)
- Of course, a hypothesis obtained from a single match is unreliable
- Solution: Coarsely quantize the transformation space. Let each match vote for its hypothesis in the quantized space.



David G. Lowe. **"Distinctive image features from scale-invariant keypoints",** *IJCV* 60 (2), pp. 91-110, 2004.

Basic algorithm outline

```
1. Initialize accumulator H
   to all zeros
2. For each tentative match
         compute transformation
              hypothesis: tx, ty, s, θ
         H(tx,ty,s,θ) = H(tx,ty,s,θ) + 1
     end
   end
3. Find all bins (tx,ty,s,θ) where H(tx,ty,s,θ) has at least
   three votes
```



- Correct matches will consistently vote for the same transformation while mismatches will spread votes.
- Cost: Linear scan through the matches (step 2), followed by a linear scan through the accumulator (step 3).

Hough transform details (D. Lowe's system)

Training phase: For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)

Test phase: Let each match between a test and a model feature vote in a 4D Hough space

- Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
- Vote for two closest bins in each dimension

Find all bins with at least three votes and perform geometric verification

- Estimate least squares *affine* transformation
- Use stricter thresholds on transformation residual
- Search for additional features that agree with the alignment

Hough transform in object recognition (references)

P.V.C. Hough, Machine Analysis of Bubble Chamber Pictures, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

D. Lowe, “Distinctive image features from scale-invariant keypoints”, *IJCV* 60 (2), 2004.

H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, ECCV’2008

Extensions (object category detection):

B. Leibe, A. Leonardis, and B. Schiele., Combined Object Categorization and Segmentation with an Implicit Shape Model, in ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague, May 2004.

S. Maji and J. Malik, Object Detection Using a Max-Margin Hough Tranform, CVPR’2009

A. Lehmann, B. Leibe, L. Van Gool. Fast PRISM: Branch and Bound Hough Transform for Object Class Detection, IJCV (to appear), 2010.

O. Barinova, V. Lempitsky, P. Kohli, On the Detection of Multiple Object Instances using Hough Transforms, CVPR, 2010

Comparison

Hough Transform

Advantages

- Can handle high percentage of outliers (>95%)
- Extracts groupings from clutter in linear time

Disadvantages

- Quantization issues
- Only practical for small number of dimensions (up to 4)

Improvements available

- Probabilistic Extensions
- Continuous Voting Space
- Can be generalized to arbitrary shapes and objects

RANSAC

Advantages

- General method suited to large range of problems
- Easy to implement
- "Independent" of number of dimensions

Disadvantages

- Basic version only handles moderate number of outliers (<50%)

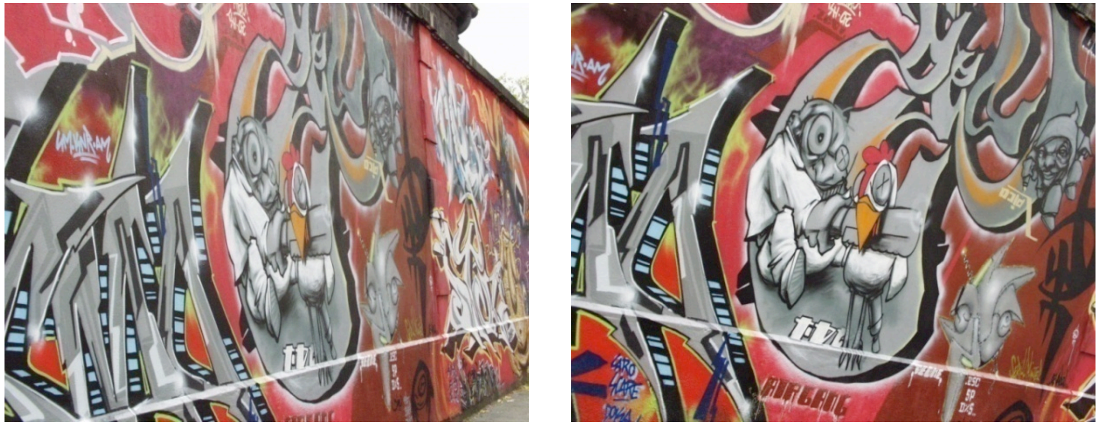
Many variants available, e.g.

- PROSAC: Progressive RANSAC [Chum05]
- Preemptive RANSAC [Nister05]

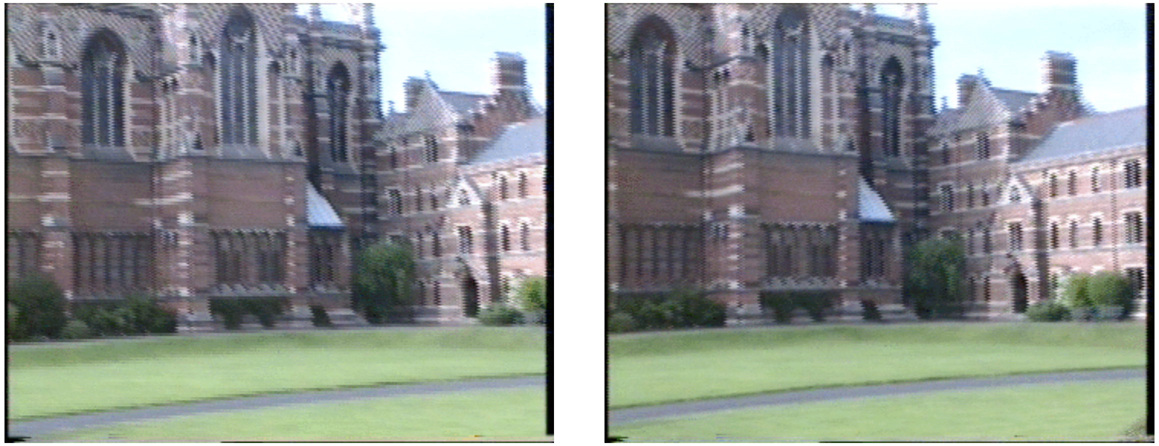
Slide credit: K. Grauman, B. Leibe

Beyond affine transformations

What is the transformation between two views of a planar surface?



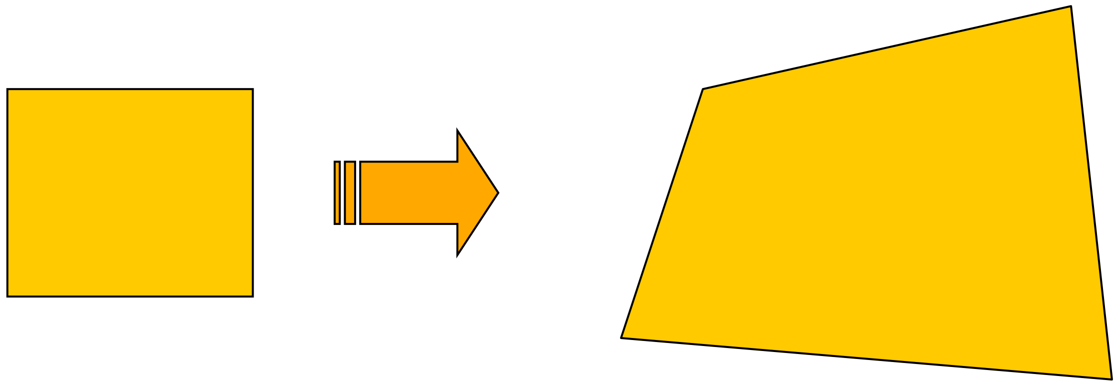
What is the transformation between images from two cameras that share the same center?



Images courtesy of
C. Schmid and
A. Zisserman.

Beyond affine transformations

Homography: plane projective transformation
(transformation taking a quad to another arbitrary quad)



Fitting a homography

Recall: homogenenous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogenenous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogenenous image coordinates

Fitting a homography

Recall: homogenenous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogenenous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogenenous image coordinates

Equation for homography:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Fitting a homography

Equation for homography:

$$\lambda_i \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\lambda_i \mathbf{x}_i' = \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \mathbf{x}_i$$

9 entries, 8 degrees of freedom (scale is arbitrary)

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = 0$$

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} y_i'\,\mathbf{h}_3^T \mathbf{x}_i - \mathbf{h}_2^T \mathbf{x}_i \\ \mathbf{h}_1^T \mathbf{x}_i - x_i'\,\mathbf{h}_3^T \mathbf{x}_i \\ x_i'\,\mathbf{h}_2^T \mathbf{x}_i - y_i'\,\mathbf{h}_1^T \mathbf{x}_i \end{bmatrix}$$

$$\begin{bmatrix} 0^T & -\mathbf{x}_i^T & y_i'\,\mathbf{x}_i^T \\ \mathbf{x}_i^T & 0^T & -x_i'\,\mathbf{x}_i^T \\ -y_i'\,\mathbf{x}_i^T & x_i'\,\mathbf{x}_i^T & 0^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0$$

3 equations, only 2 linearly independent

Direct linear transform

$$\begin{bmatrix} 0^T & \mathbf{x}_1^T & -y_1' \mathbf{x}_1^T \\ \mathbf{x}_1^T & 0^T & -x_1' \mathbf{x}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathbf{x}_n^T & -y_n' \mathbf{x}_n^T \\ \mathbf{x}_n^T & 0^T & -x_n' \mathbf{x}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0 \qquad \mathbf{A}\,\mathbf{h} = 0$$

H has 8 degrees of freedom (9 parameters, but scale is arbitrary)

One match gives us two linearly independent equations

Four matches needed for a minimal solution (null space of 8x9 matrix)

More than four: homogeneous least squares

Application: Panorama stitching

Camera rotating about its centre:

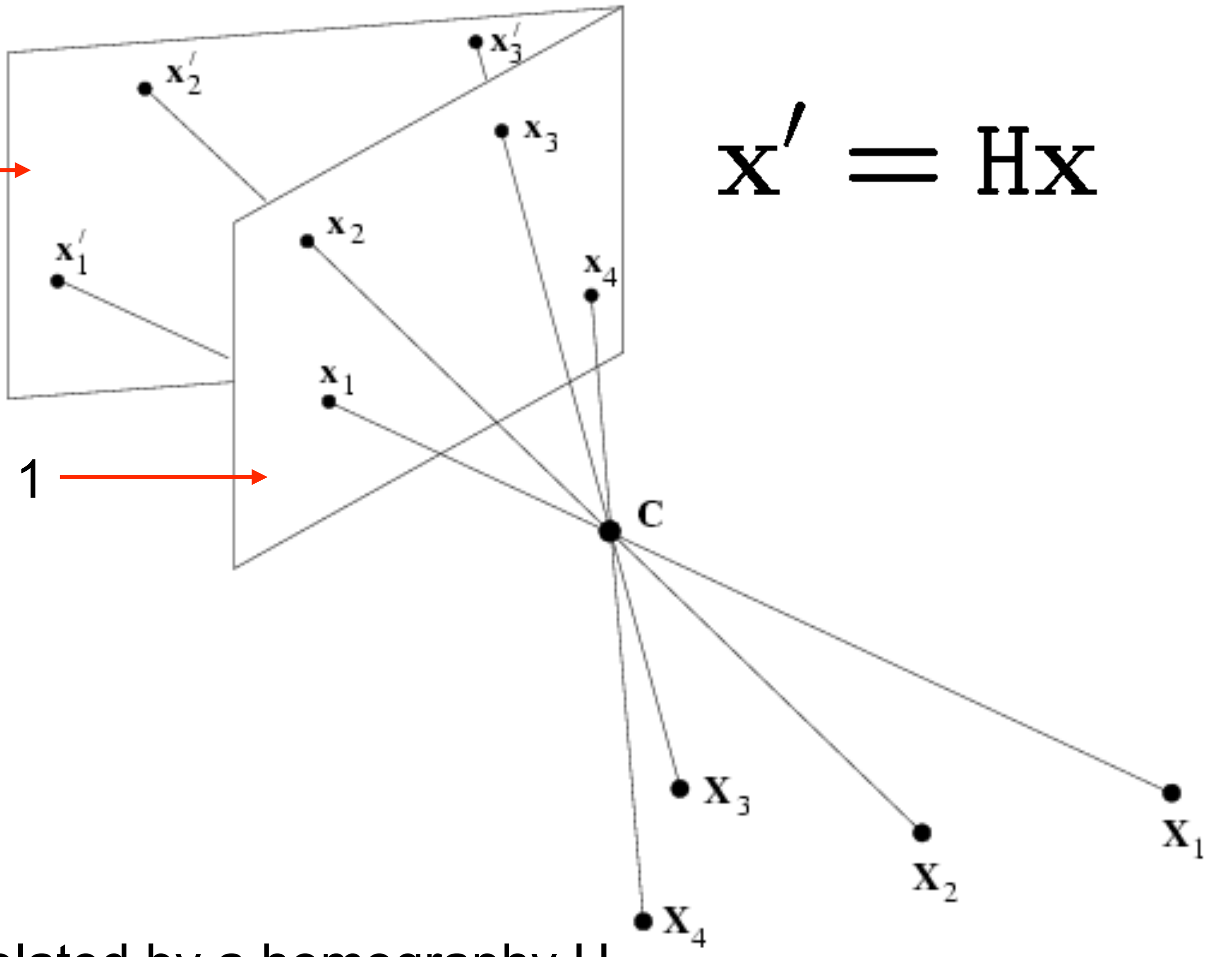


image plane 2

image plane 1

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

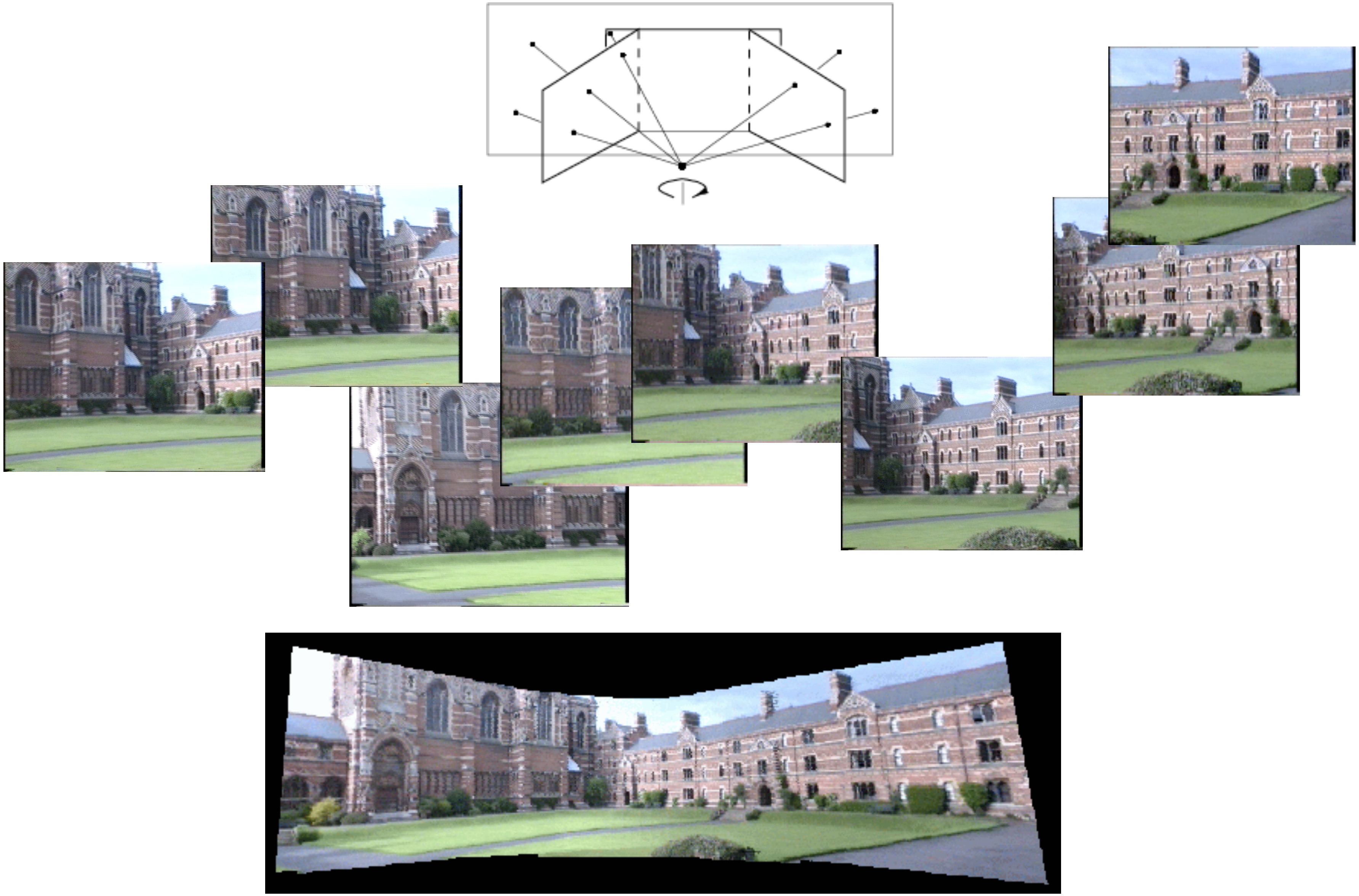
P = K [I | 0]   P' = K' [R | 0]

H = K' R K^(-1)

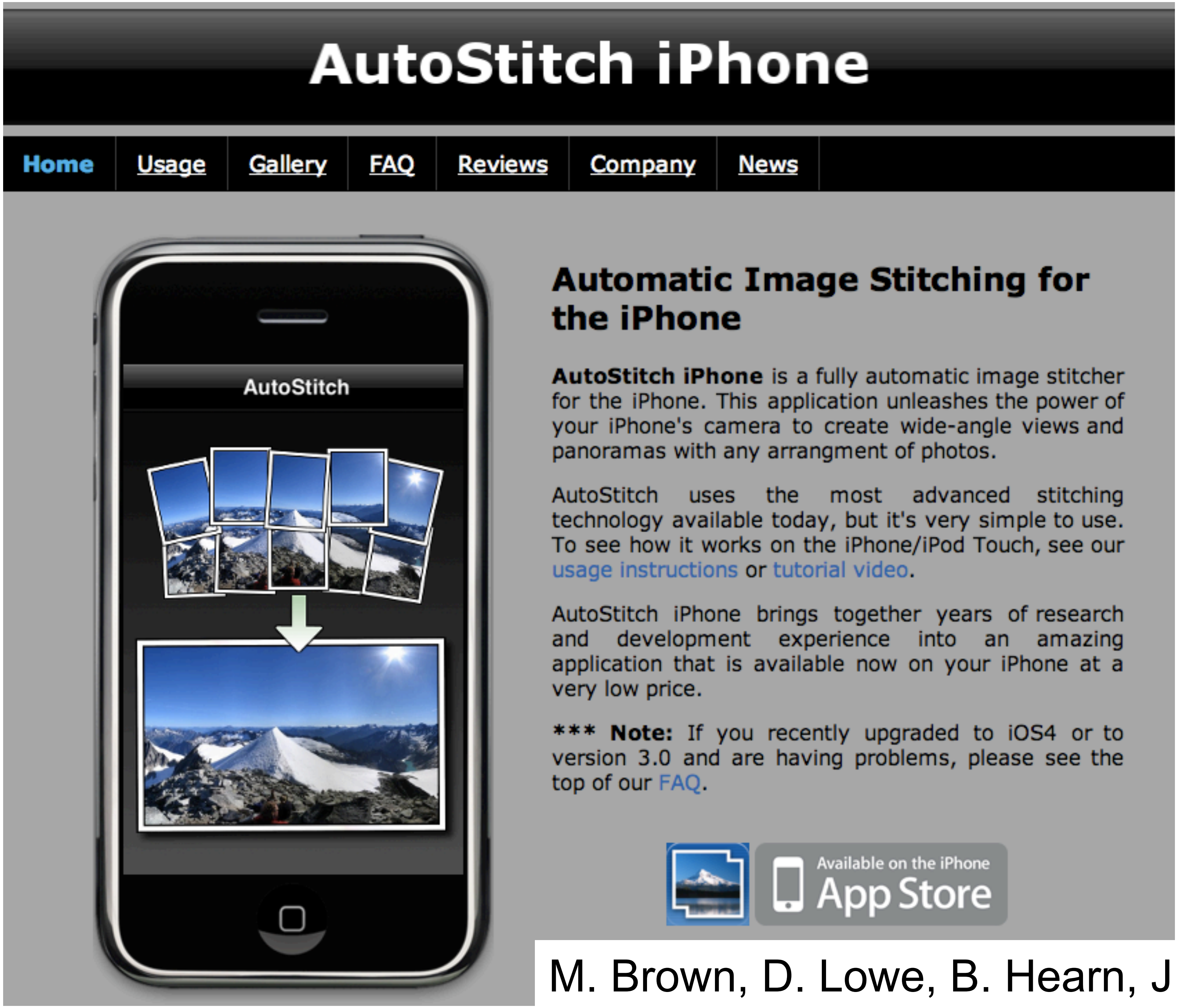
- The two image planes are related by a homography H
- H depends only on the relation between the image planes and camera centre, C, not on the 3D structure

Image: A. Zisserman.

Application: Panorama stitching



Images courtesy of A. Zisserman.



AutoStitch iPhone

Automatic Image Stitching for the iPhone

AutoStitch iPhone is a fully automatic image stitcher for the iPhone. This application unleashes the power of your iPhone's camera to create wide-angle views and panoramas with any arrangment of photos.

AutoStitch uses the most advanced stitching technology available today, but it's very simple to use. To see how it works on the iPhone/iPod Touch, see our usage instructions or tutorial video.

AutoStitch iPhone brings together years of research and development experience into an amazing application that is available now on your iPhone at a very low price.

*** **Note:** If you recently upgraded to iOS4 or to version 3.0 and are having problems, please see the top of our FAQ.

M. Brown, D. Lowe, B. Hearn, J. Beis

Outline

1. Local invariant features (45 mins, C. Schmid)

2. Matching and recognition with local features (45 mins, J. Sivic)

3. **Efficient visual search (45 mins, J. Sivic)**

4. Very large scale visual indexing – recent work (45 mins, C. Schmid)

Practical session – Panorama stitching (60 mins)

Download:

http://www.di.ens.fr/willow/events/cvml2011/mosaic.zip