Visual Recognition and Machine Learning Summer School

Paris 2013

Instance-level recognition – part 2

Josef Sivic

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

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548

Departement 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 (C. Schmid)

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

3. Efficient visual search (J. Sivic)

4. Very large scale visual indexing (C. Schmid)

Practical session – Instance-level recognition and search

[Try your wifi network access.]

Image matching and recognition with local features

The goal: establish correspondence between two or more images

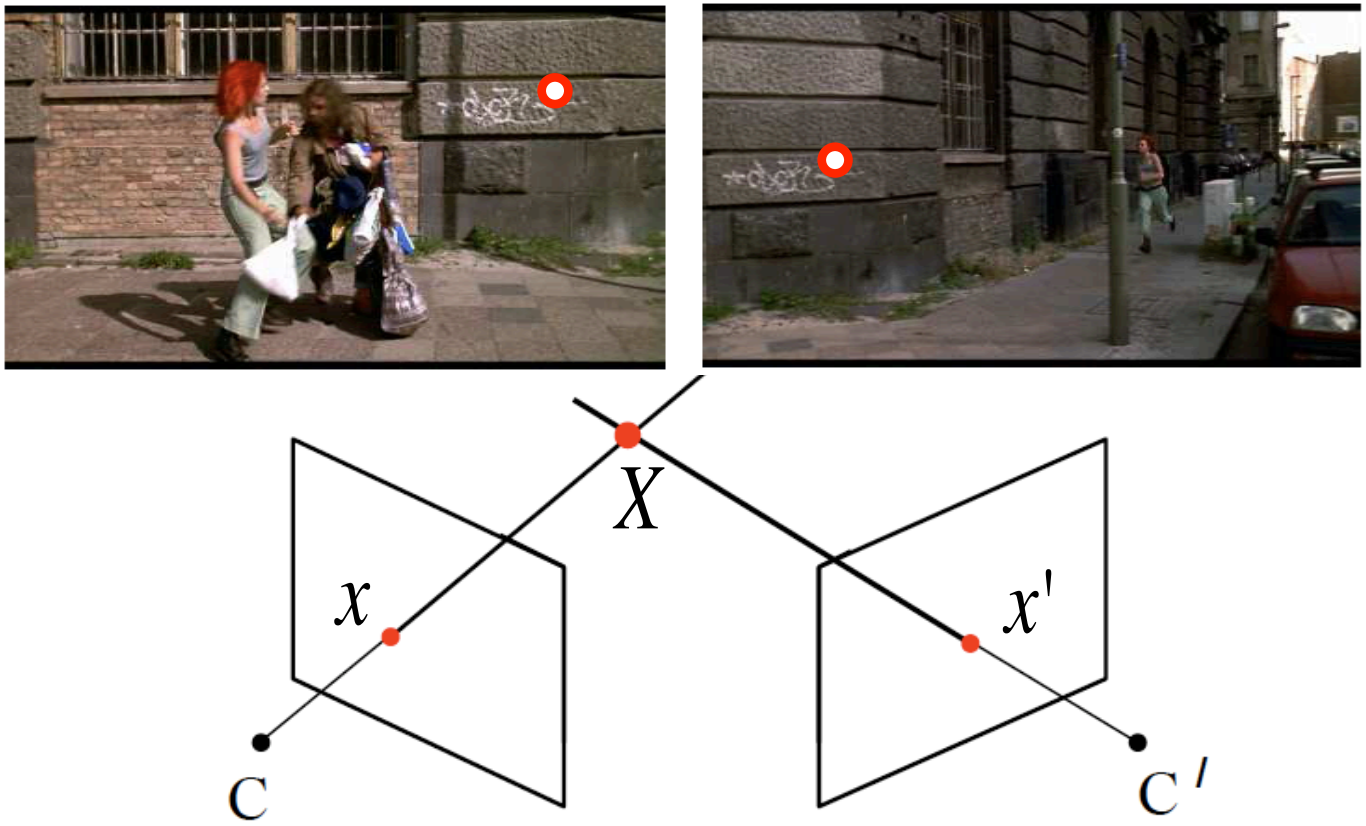


Image points x and x' are in correspondence if they are projections of the same 3D scene point X.

Images courtesy A. Zisserman

Example I: Wide baseline matching and 3D reconstruction

Establish correspondence between two (or more) images.

[Schaffalitzky and Zisserman ECCV 2002]

Example I: Wide baseline matching and 3D reconstruction

Establish correspondence between two (or more) images.

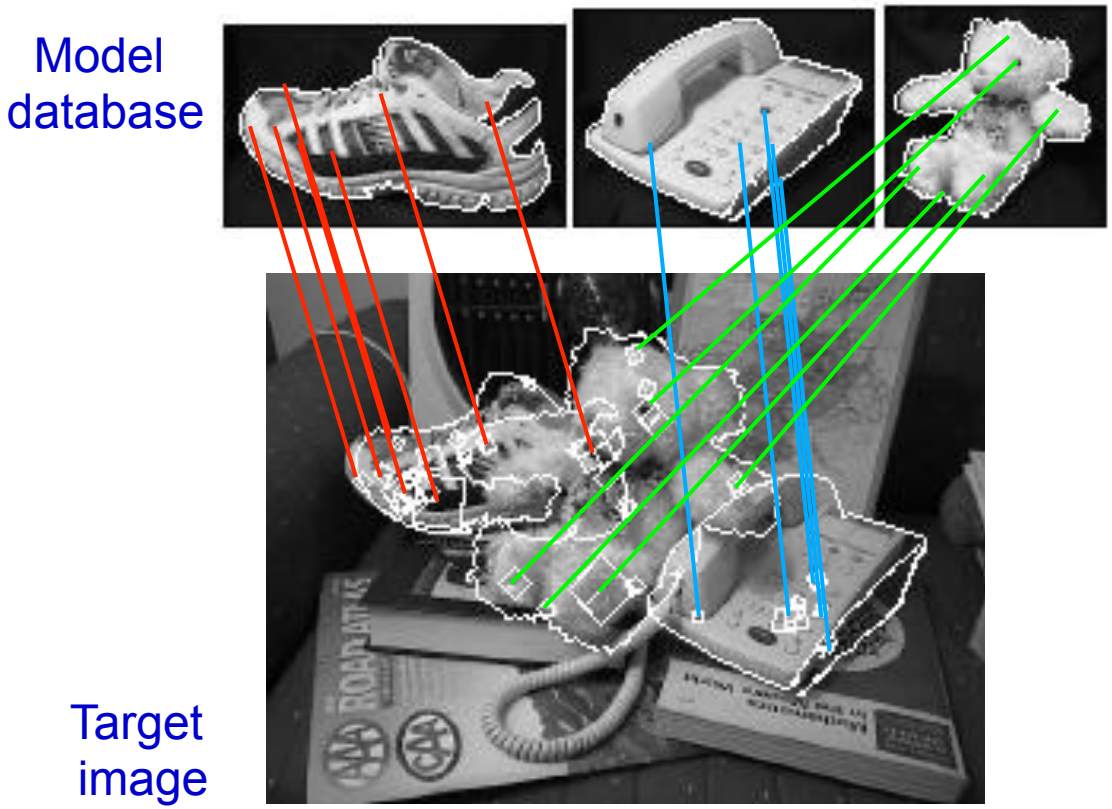
[Schaffalitzky and Zisserman ECCV 2002]

[Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV'09] – Building Rome in a Day

57,845 downloaded images, 11,868 registered images. This video: 4,619 images.

Example II: Object recognition

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



[D. Lowe, 1999]

Sony Aibo (Evolution Robotics)

SIFT usage

- Recognize docking station
- Communicate with visual cards

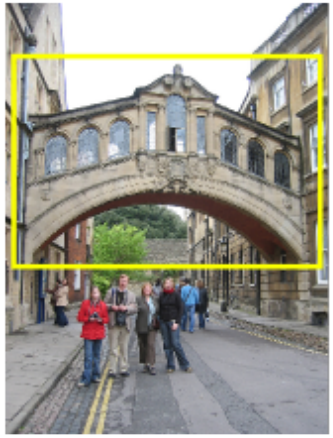
Other uses

- Place recognition
- Loop closure in SLAM

Slide credit: David Lowe

Example III: Visual search

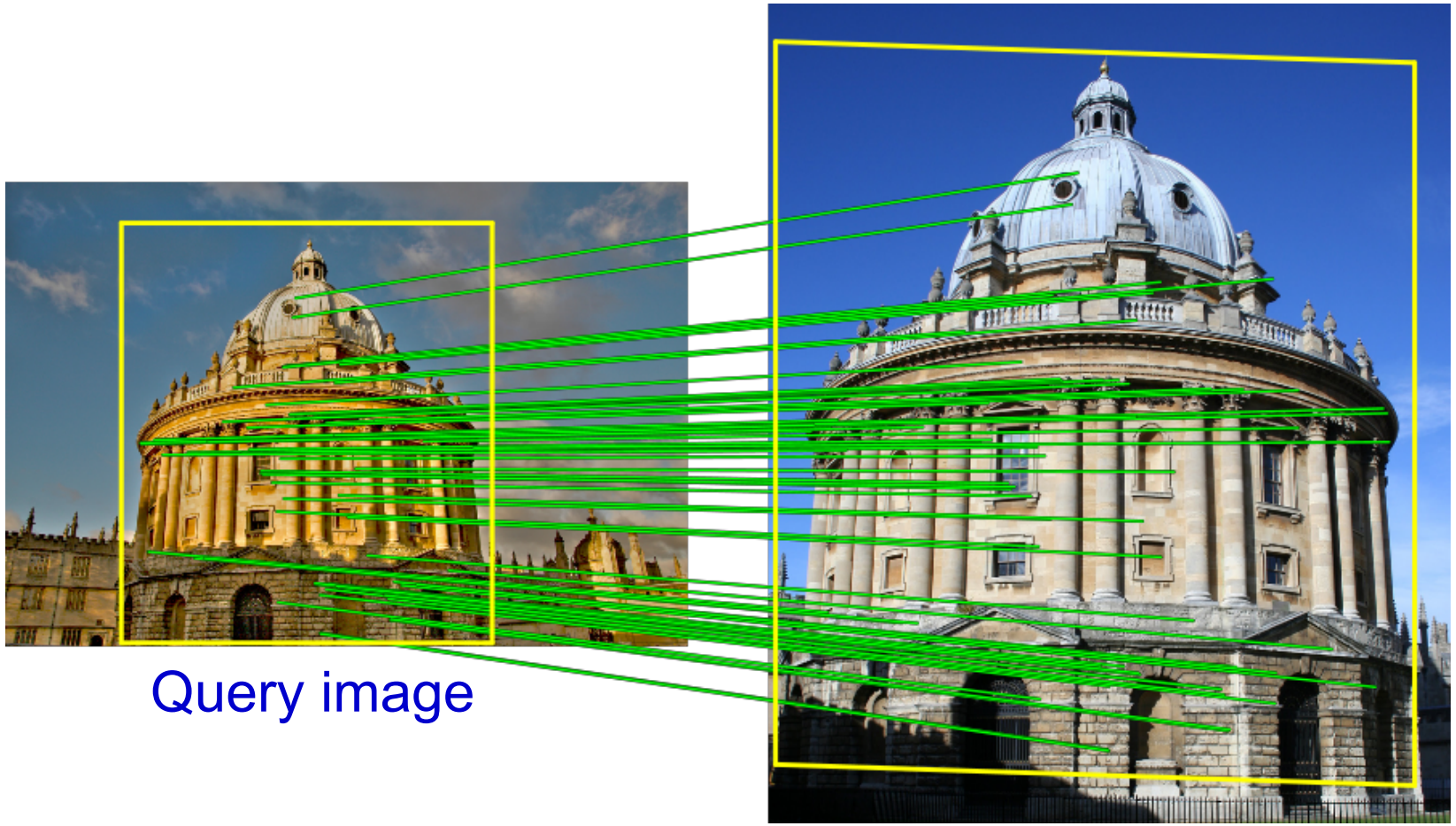
Given a query image, find images depicting the same place / object in a large unordered image collection.



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.

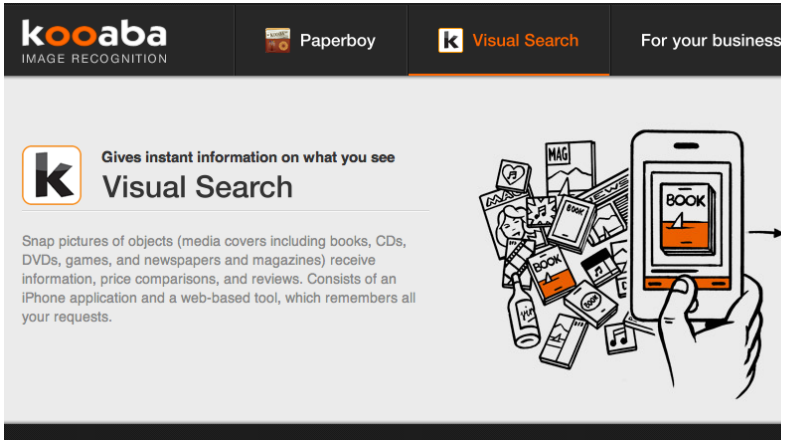


Query image

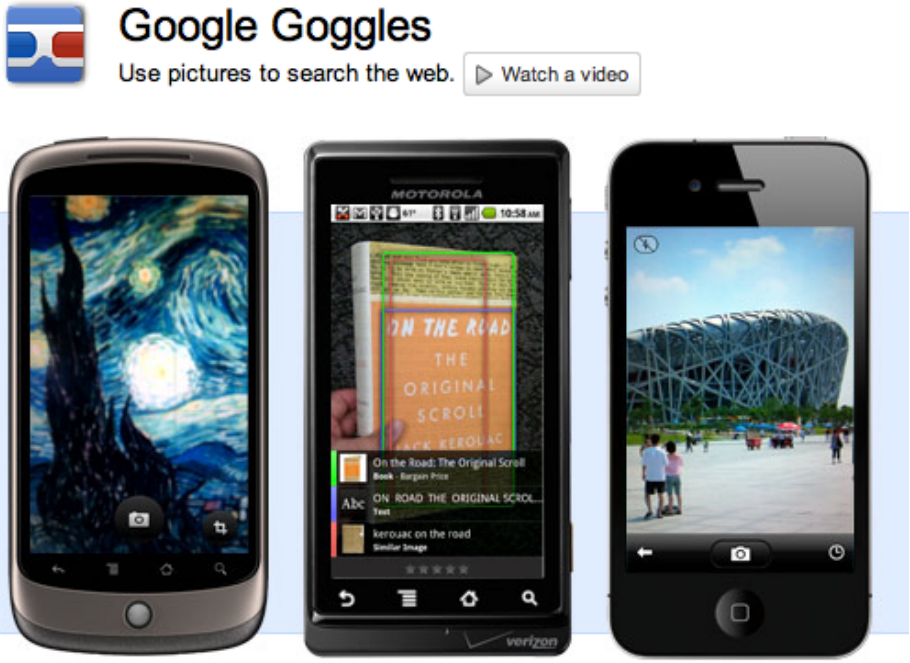
Database image(s)

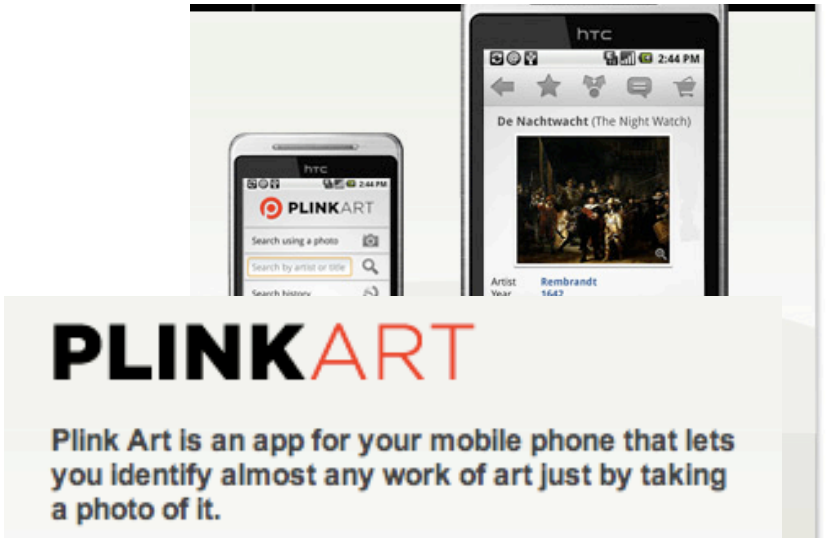
Mobile visual search

Bing visual scan

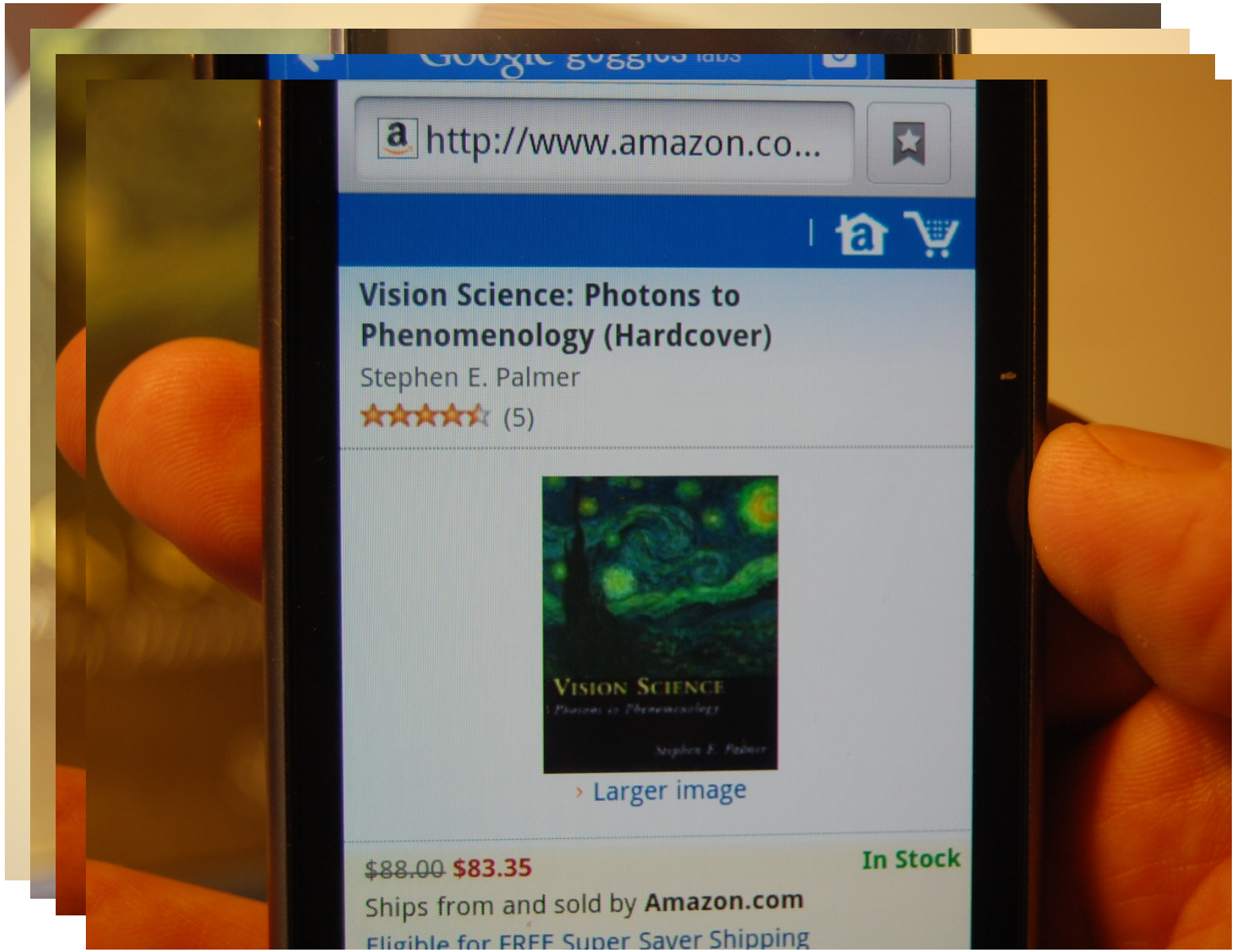


and others… Snaptell.com, Millpix.com





Example

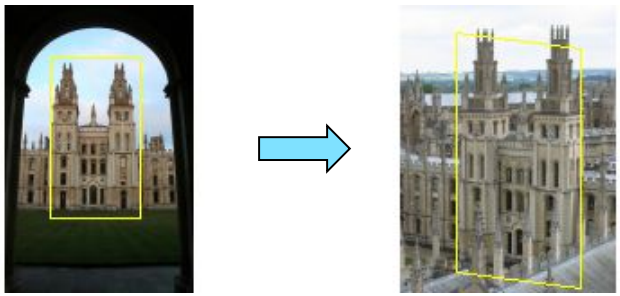


Slide credit: I. Laptev

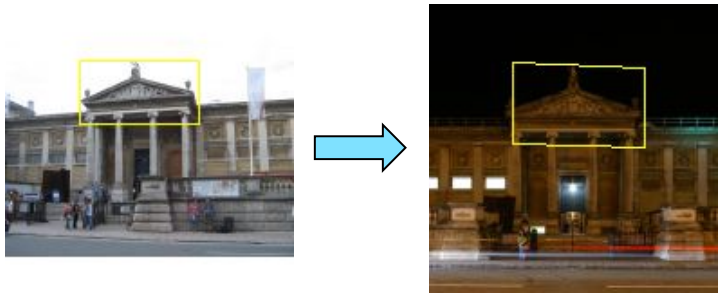
Why is it difficult?

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

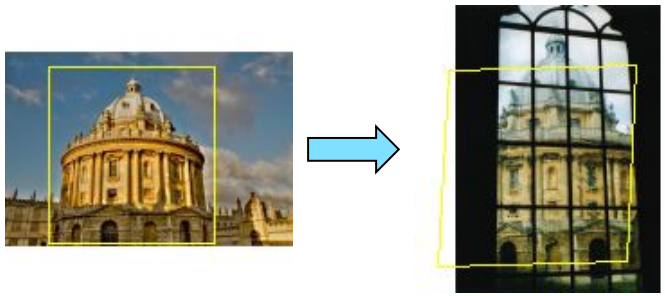
Scale



Viewpoint



Lighting



Occlusion

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

Approach

Pre-processing (so far):

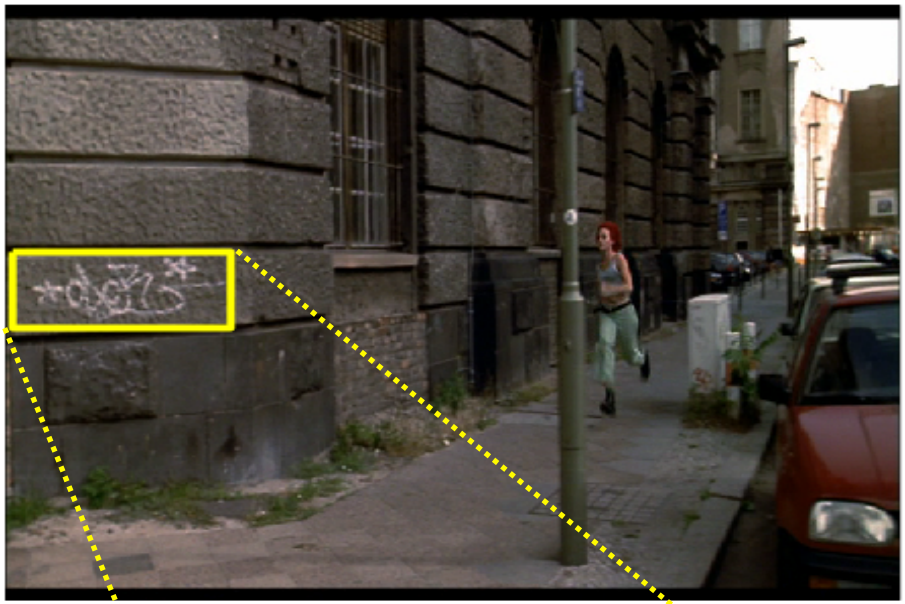
- Detect local features.
- Extract descriptor for each feature.

Matching:

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

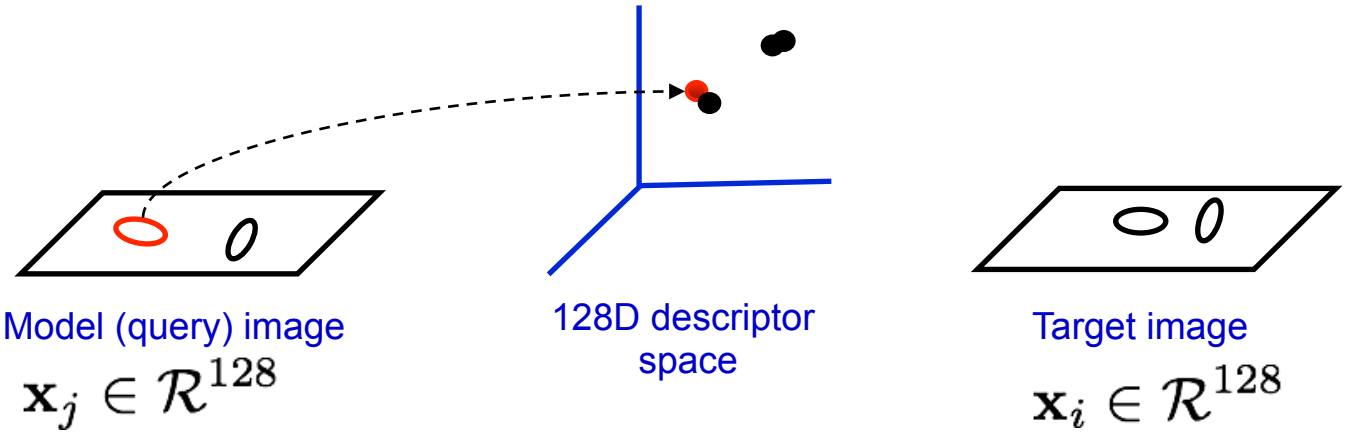
2. Verify matches based on semi-local / global geometric relations.

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



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:

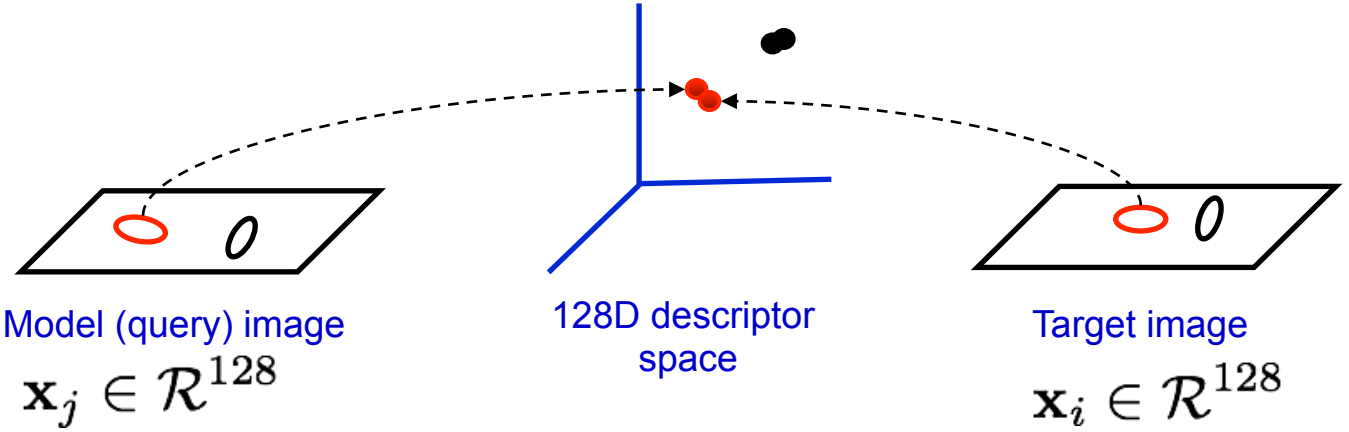
$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||,$$

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 later).

Step 1. Establish tentative correspondence

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



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:

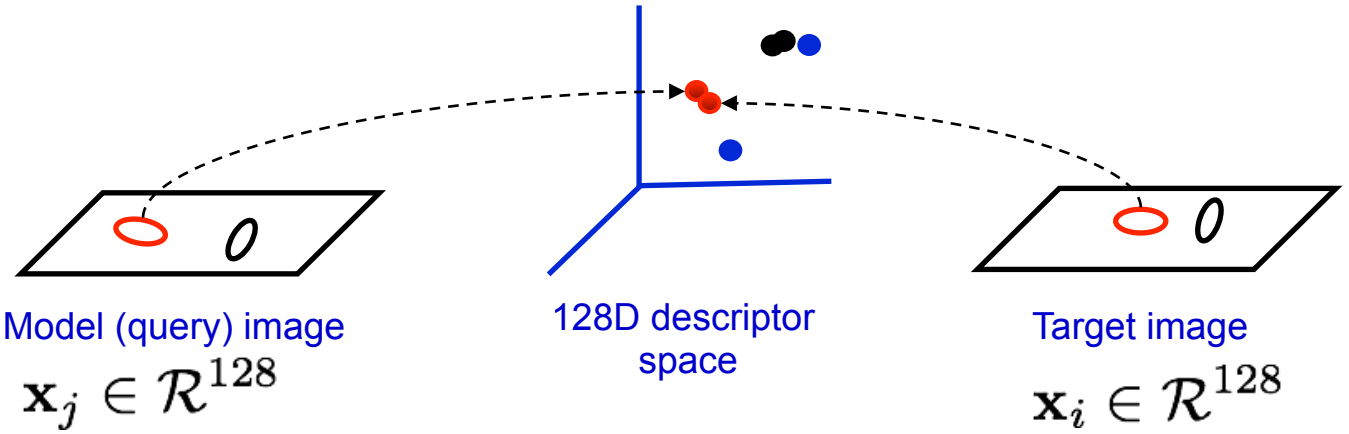
$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||,$$

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 later).

Step 1. Establish tentative correspondence

Examine the distance to the 2nd nearest neighbour [Lowe, IJCV 2004]



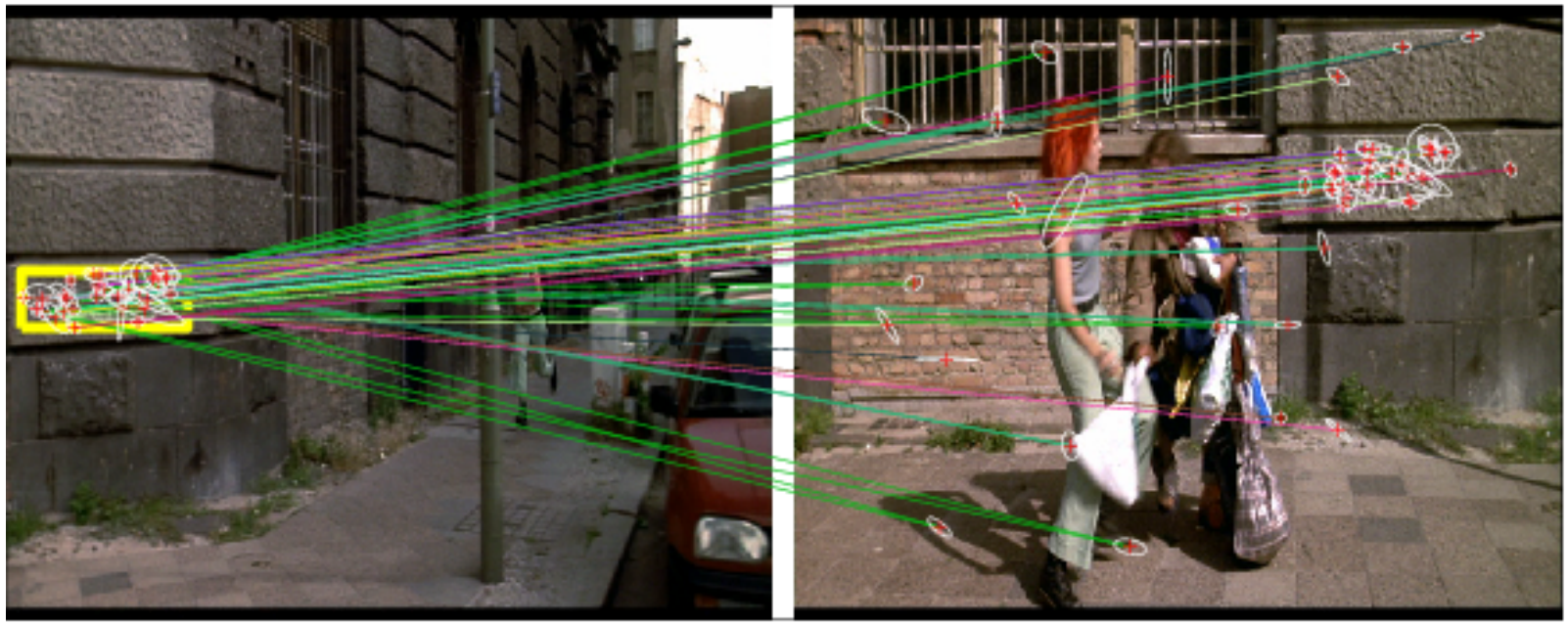
If the 2nd nearest neighbour is much further than the 1st nearest neighbour
Match is more “unique” or discriminative.

Measure this by the ratio: $r = d_{1NN} / d_{2NN}$

r is between 0 and 1
r is small the match is more unique.

See the practical later today for an example.

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

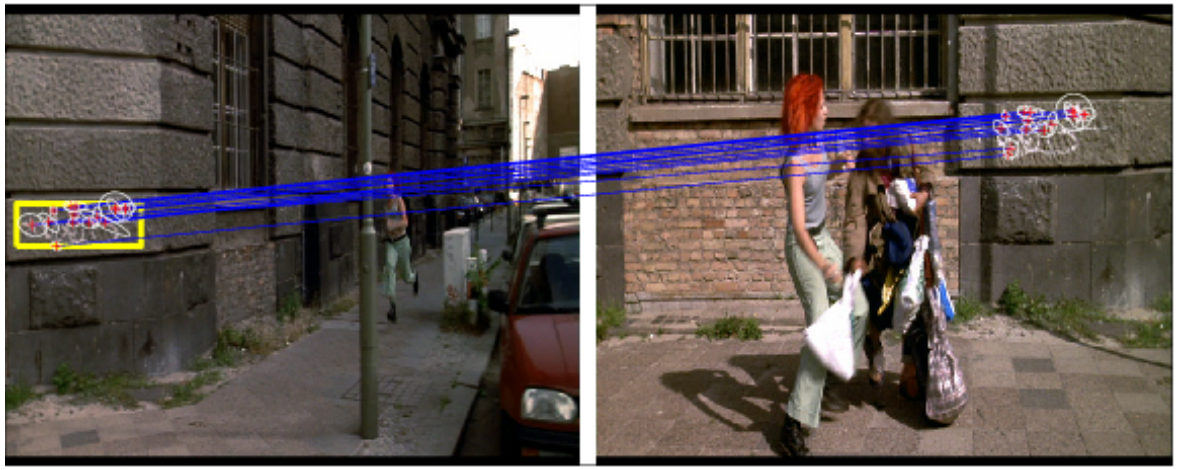
Solution: use semi-local and global spatial relations to verify matches.

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

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

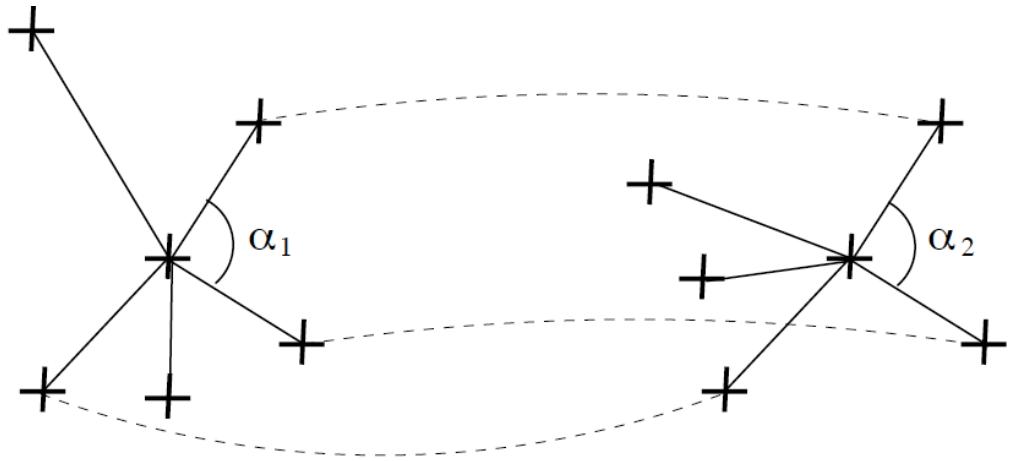


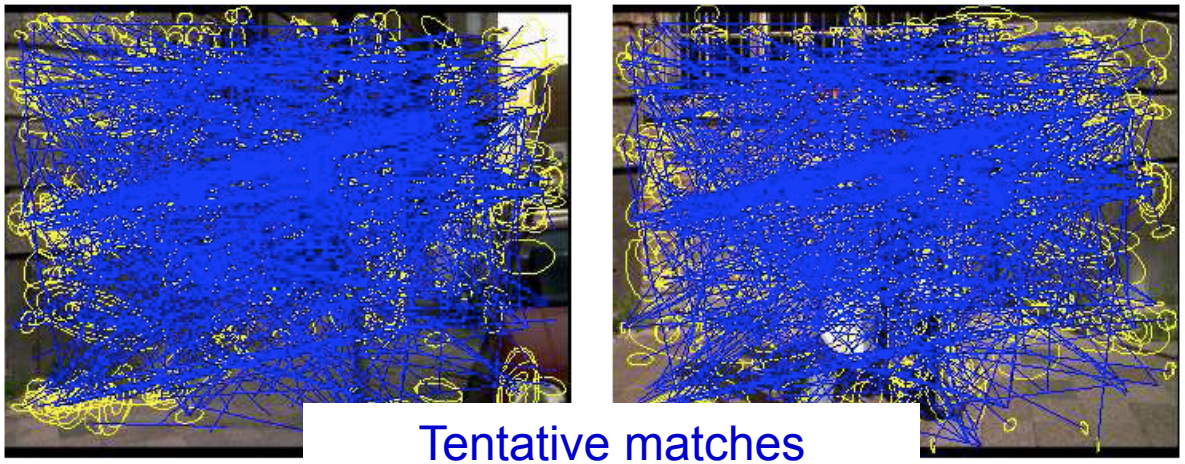
Fig. 4. Semi-local constraints : neighbours of the point have to match and angles have to correspond. Note that not all neighbours have to be matched correctly.

[Schmid&Mohr, PAMI 1997]

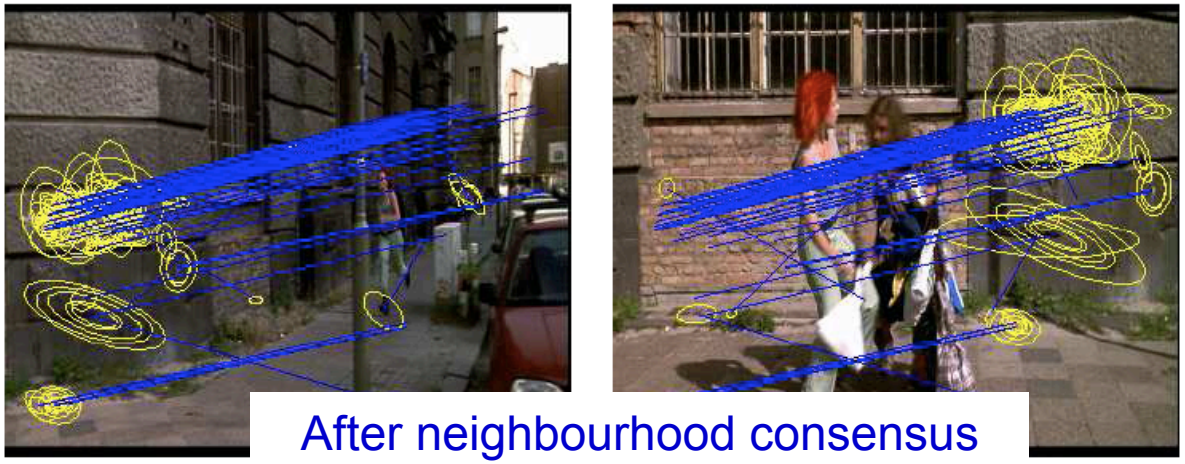
Semi-local constraints:

Example I. – neighbourhood consensus

Original images



Tentative matches



After neighbourhood consensus

[Schaffalitzky & Zisserman, CIVR 2004]

Geometric verification with global constraints

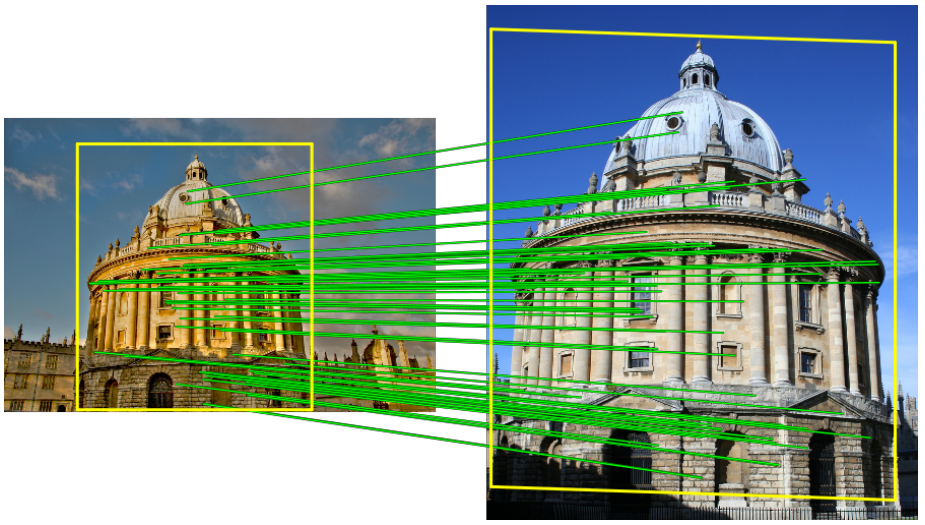
- All matches must be consistent with a global geometric relation / transformation.

- Need to simultaneously:

(i) estimate the geometric transformation and

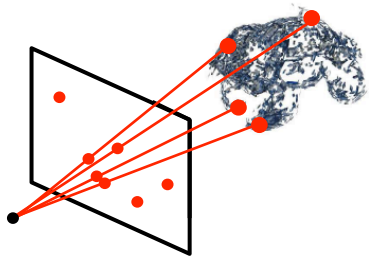
(ii) estimate the set of consistent matches

Tentative matches



Matches consistent with an affine transformation

Examples of global constraints

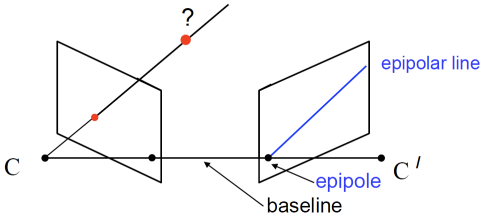


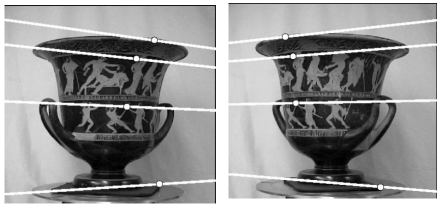
1 view and known 3D model.

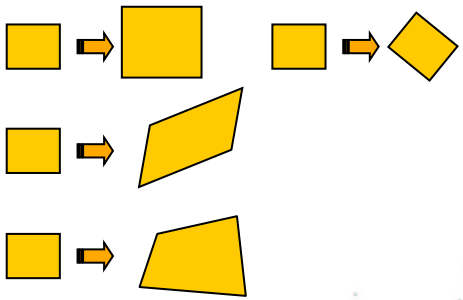
- Consistency with a (known) 3D model.

2 views

- Epipolar constraint
- **2D transformations**
  - Similarity transformation
  - Affine transformation
  - Projective transformation







N-views

Are images consistent with a 3D model?

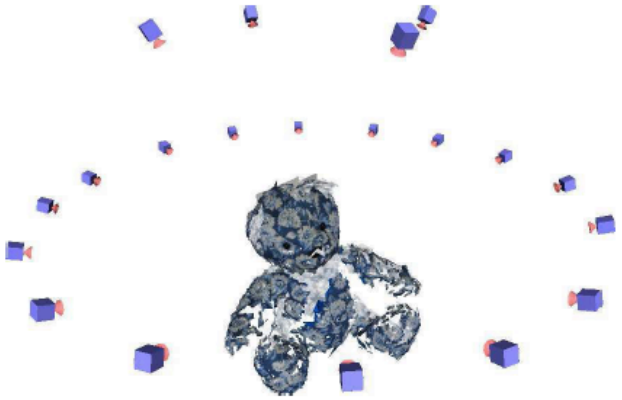
3D constraint: example

- Matches must be consistent with a 3D model

Offline: Build a 3D model

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

(a)



Recovered 3D model

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

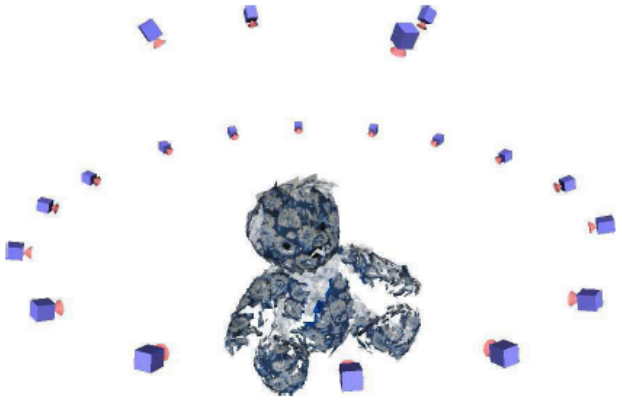
3D constraint: example

- Matches must be consistent with a 3D model

Offline: Build a 3D model

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

(a)



Recovered 3D model

At test time:

Object recognized in a previously unseen pose

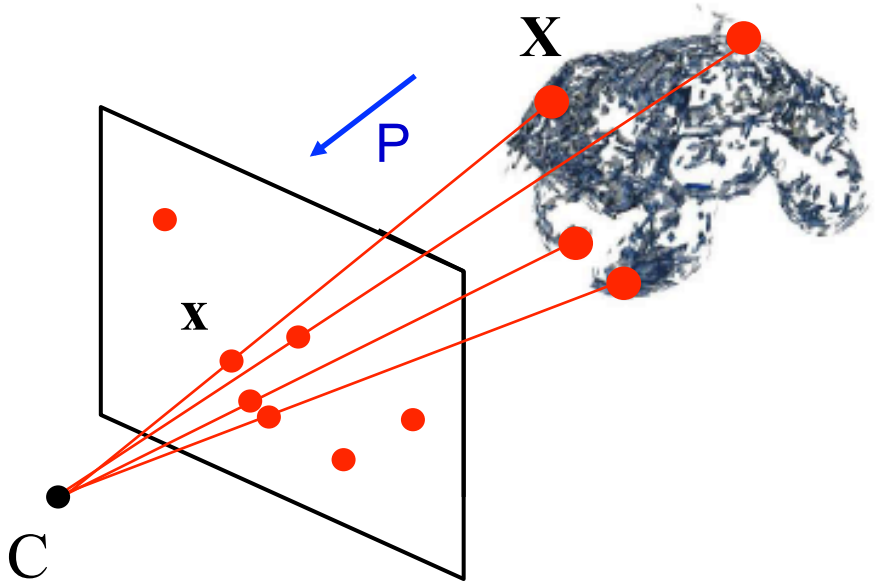
Recovered pose

(d)

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

3D constraint: example

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 camera matrix P and a set of geometrically consistent correspondences $x \leftrightarrow X$.



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

P : 3 x 4 matrix

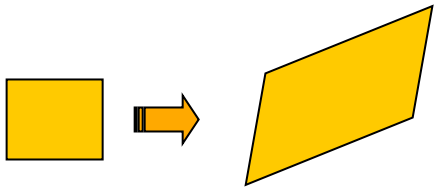
$\mathbf{X}$: 4-vector

$\mathbf{x}$: 3-vector

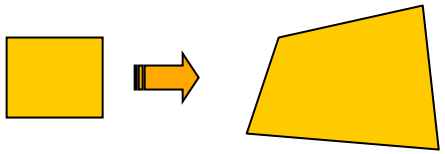
2D transformation models

Similarity
(translation,
scale, rotation)

Affine

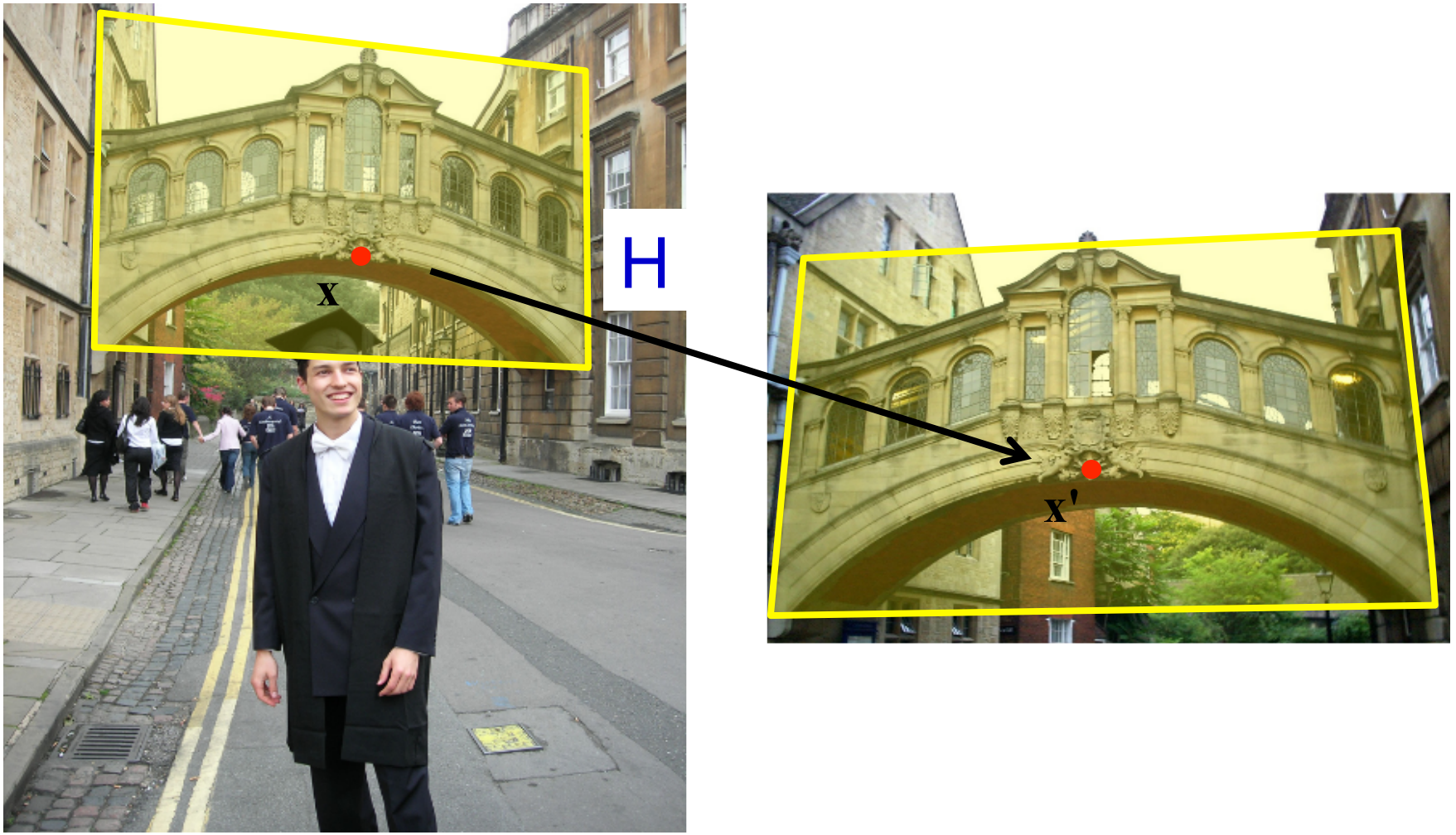


Projective
(homography)

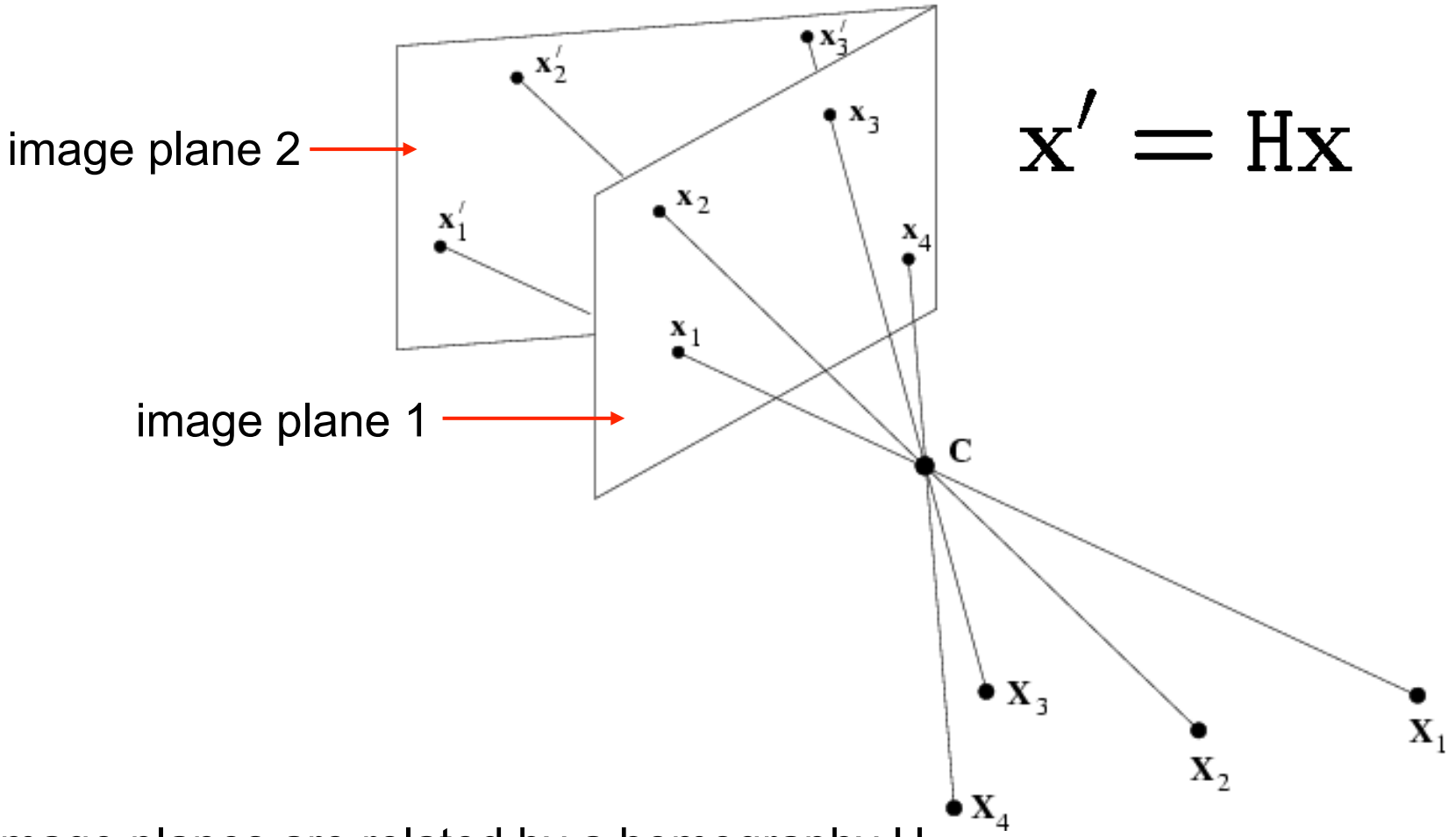


Planes in the scene induce *homographies*

Points on the plane transform as $x' = H x$, where x and x' are image points (in homogeneous coordinates), and H is a 3x3 matrix.

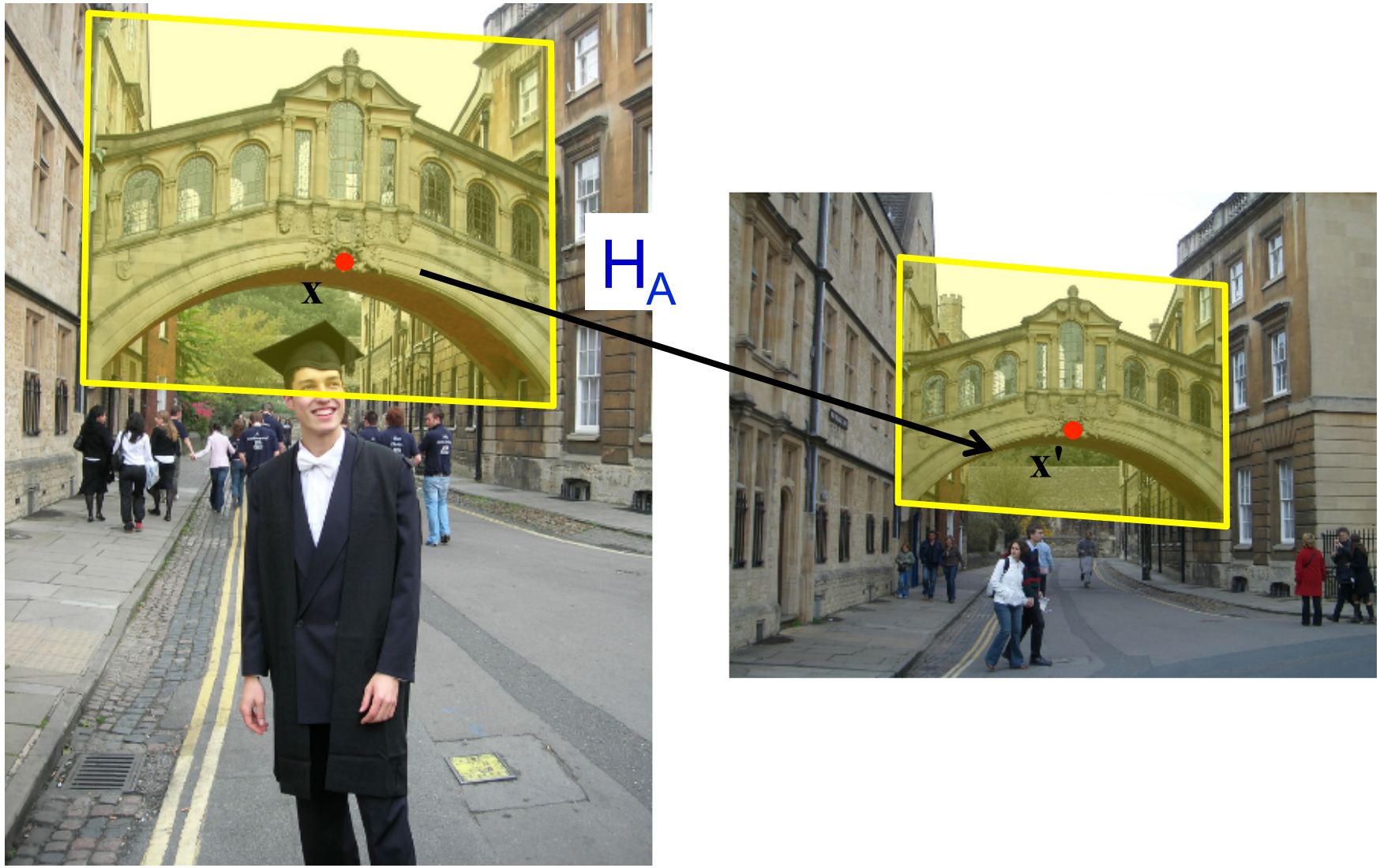


Case II: Cameras rotating about their centre



- 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

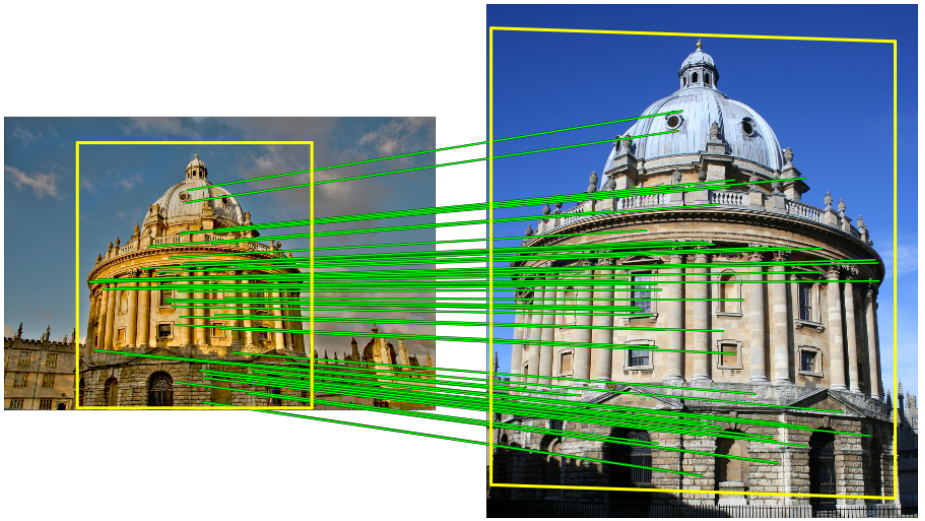
Homography is often approximated well by 2D affine geometric transformation



Homography is often approximated well by 2D affine geometric transformation – Example II.

Two images with similar camera viewpoint

Tentative matches



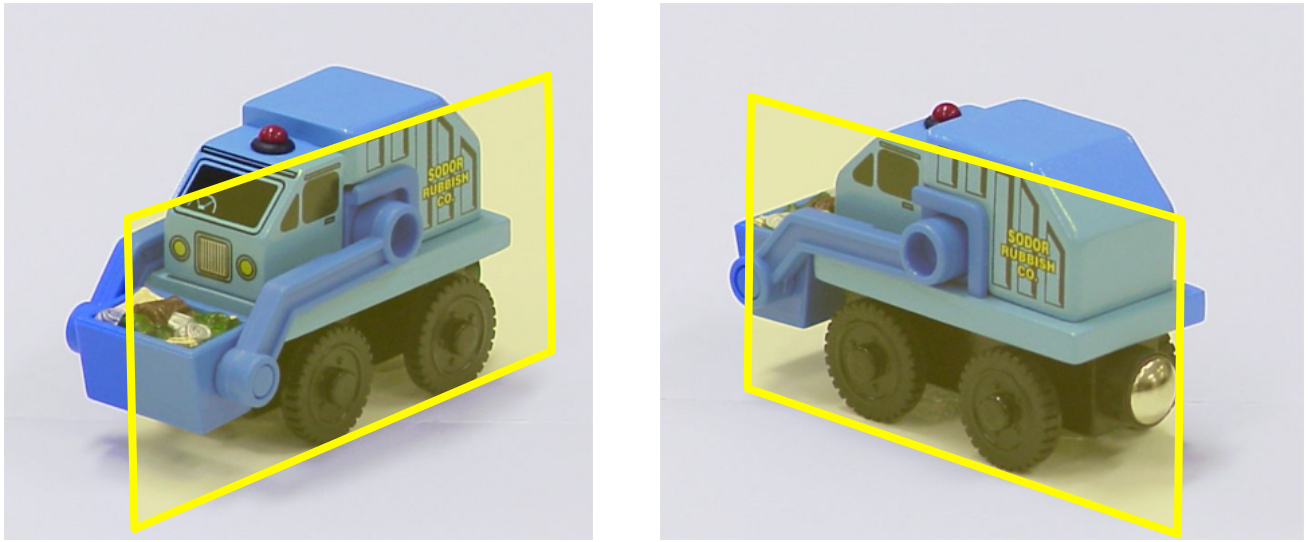
Matches consistent with an affine transformation

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

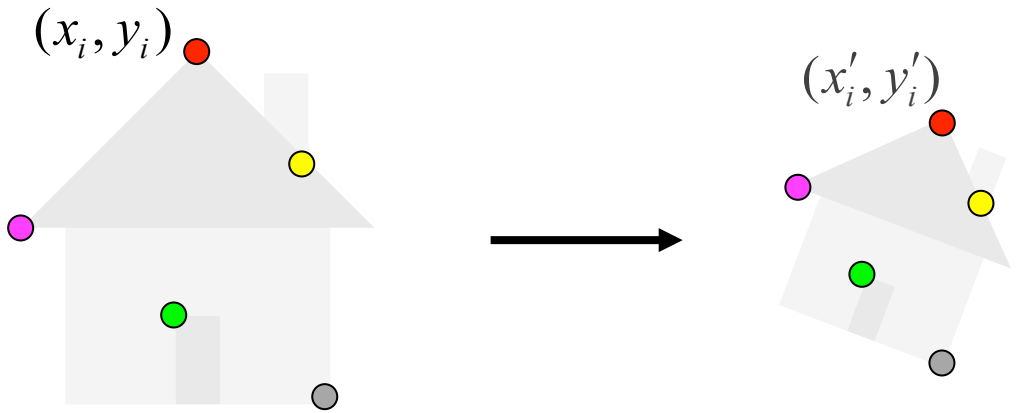
Example: estimating 2D affine transformation

- 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



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

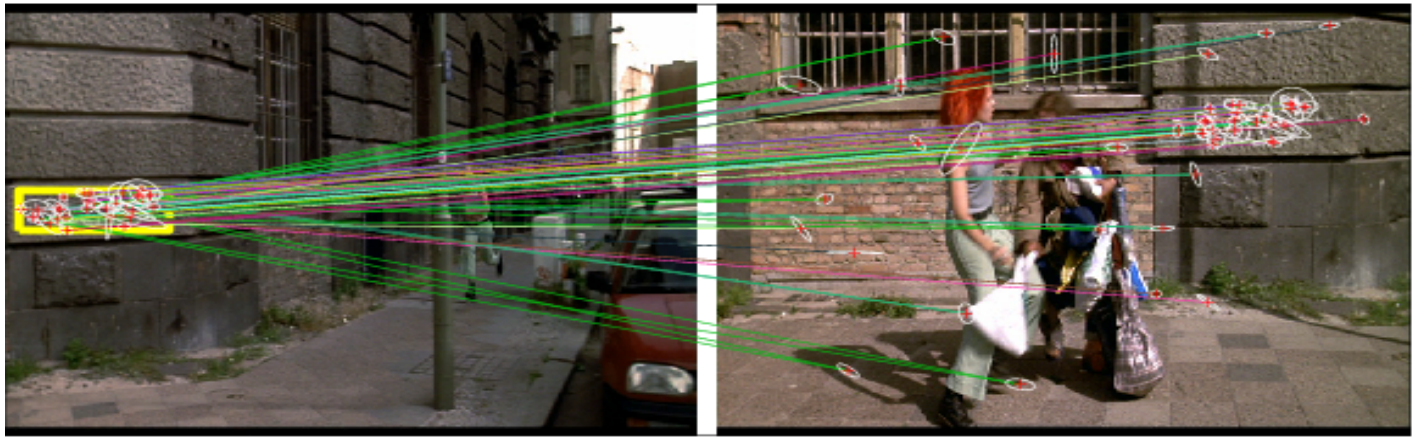
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

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



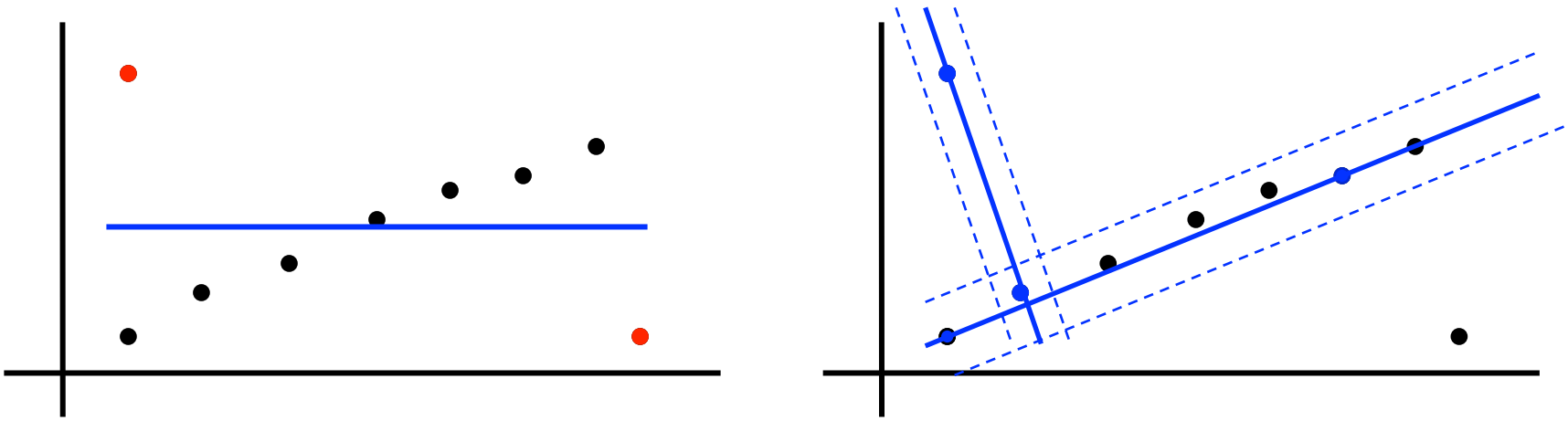
How do we fit a geometric transformation to a small subset of all possible matches?

Possible strategies:

- **RANSAC**
- Hough transform

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

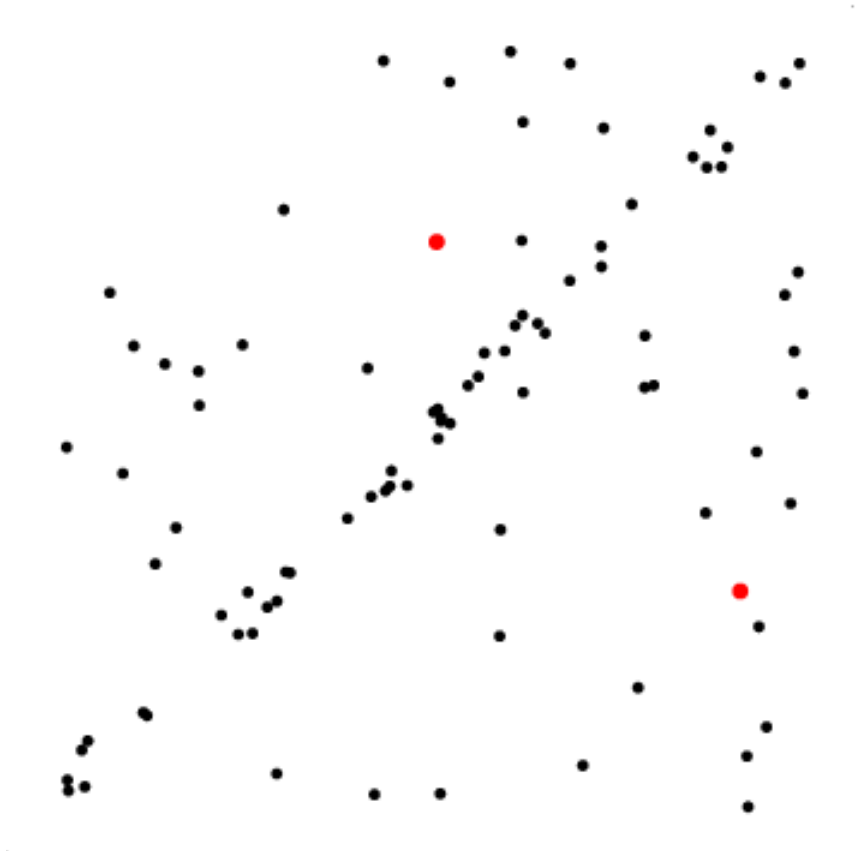
1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

Choose the line with the largest number of inliers

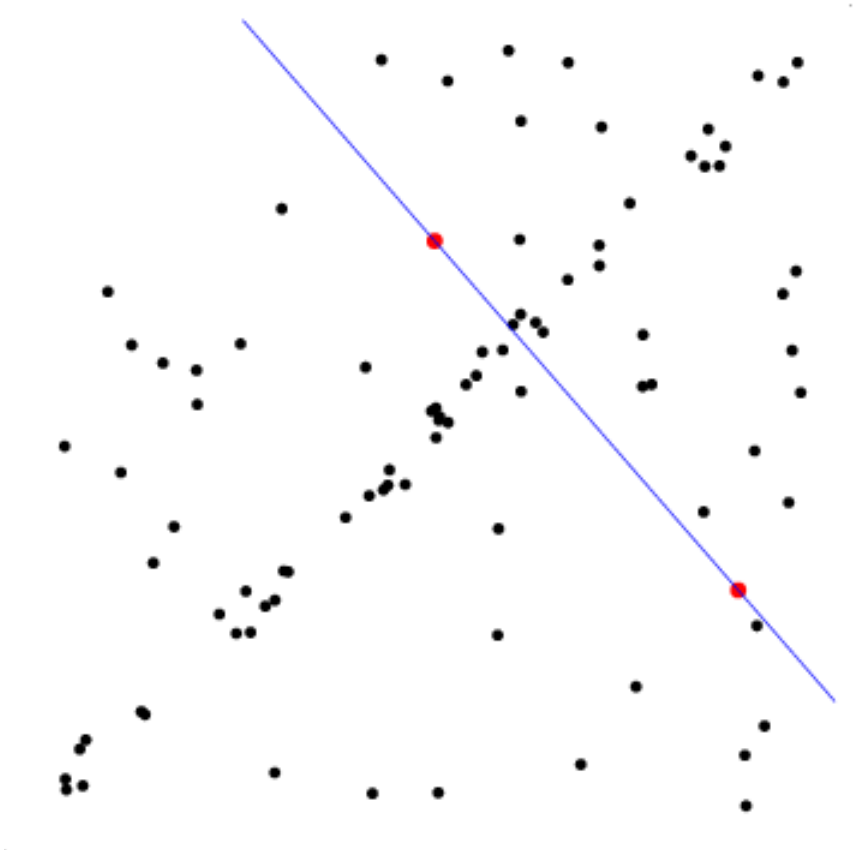
- Compute least squares fit of line to inliers (regression)

Slide credit: A. Zisserman

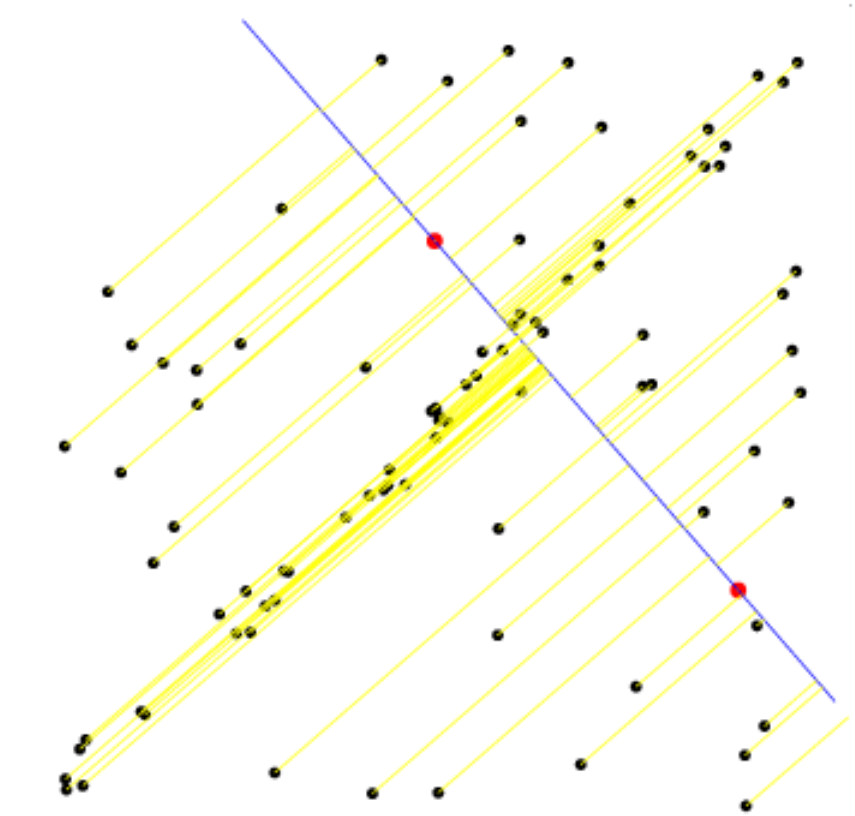
Slide credit: O. Chum



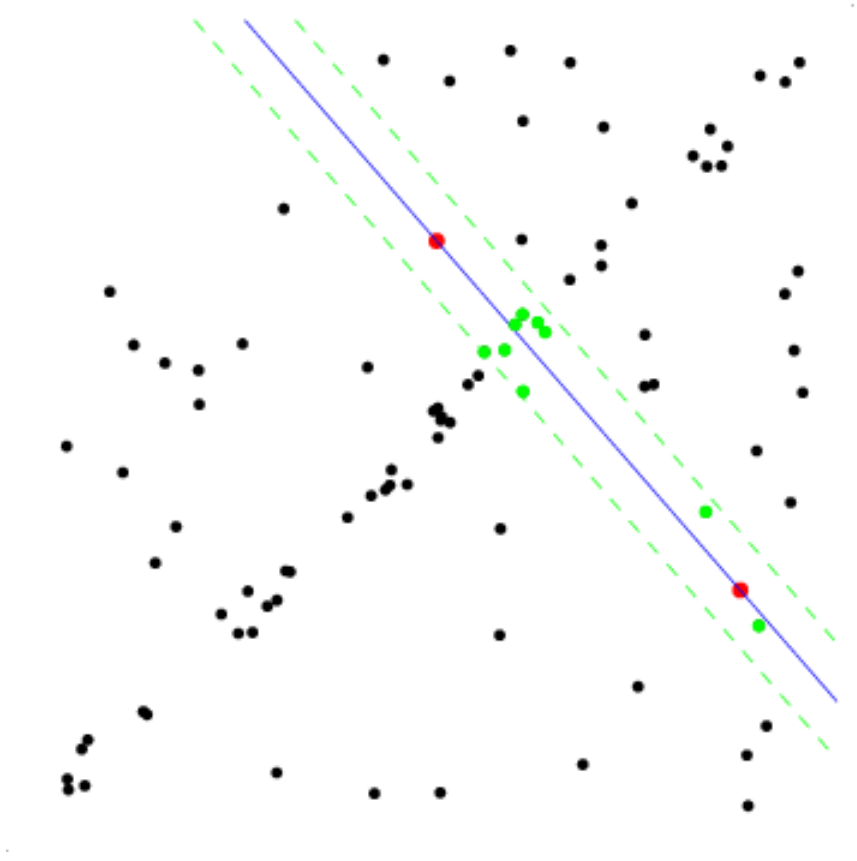
Slide credit: O. Chum



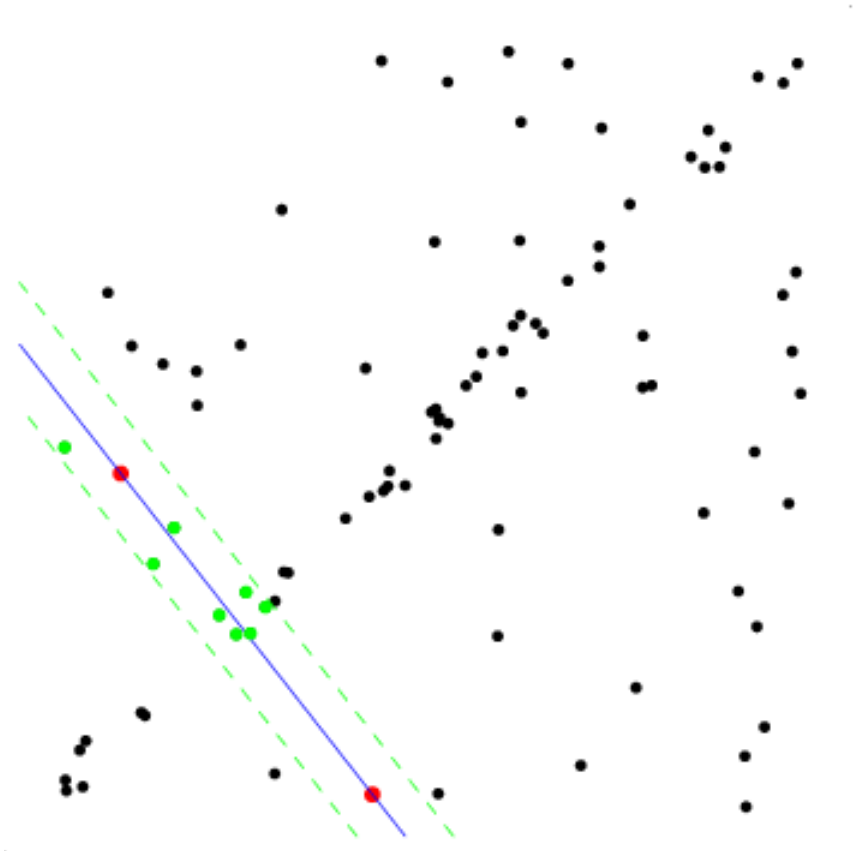
Slide credit: O. Chum



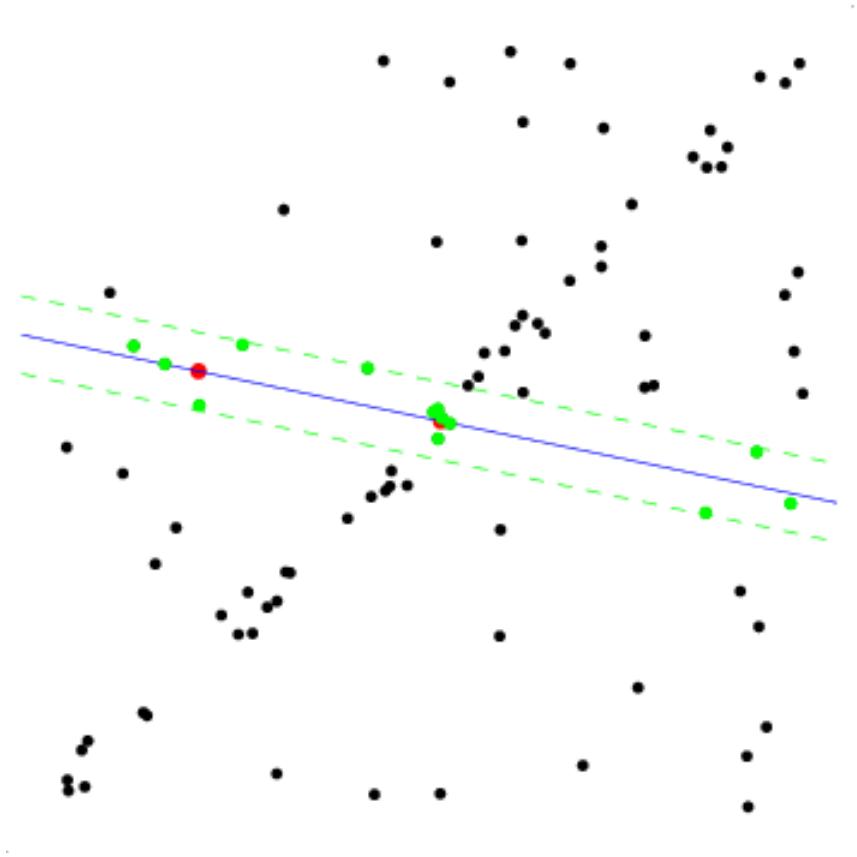
Slide credit: O. Chum



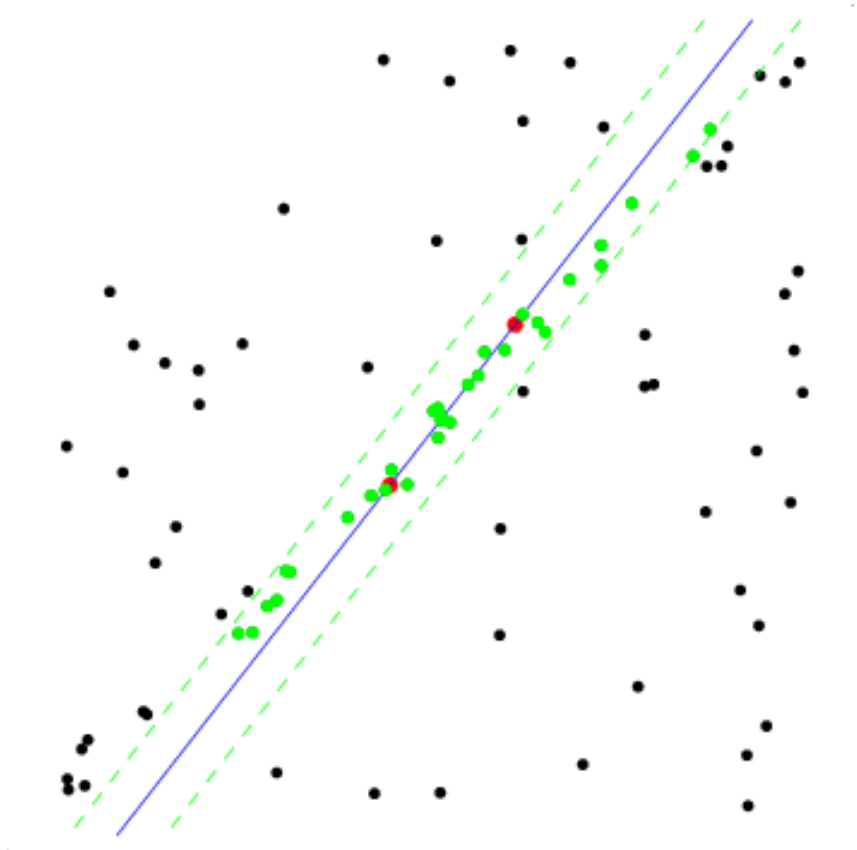
Slide credit: O. Chum



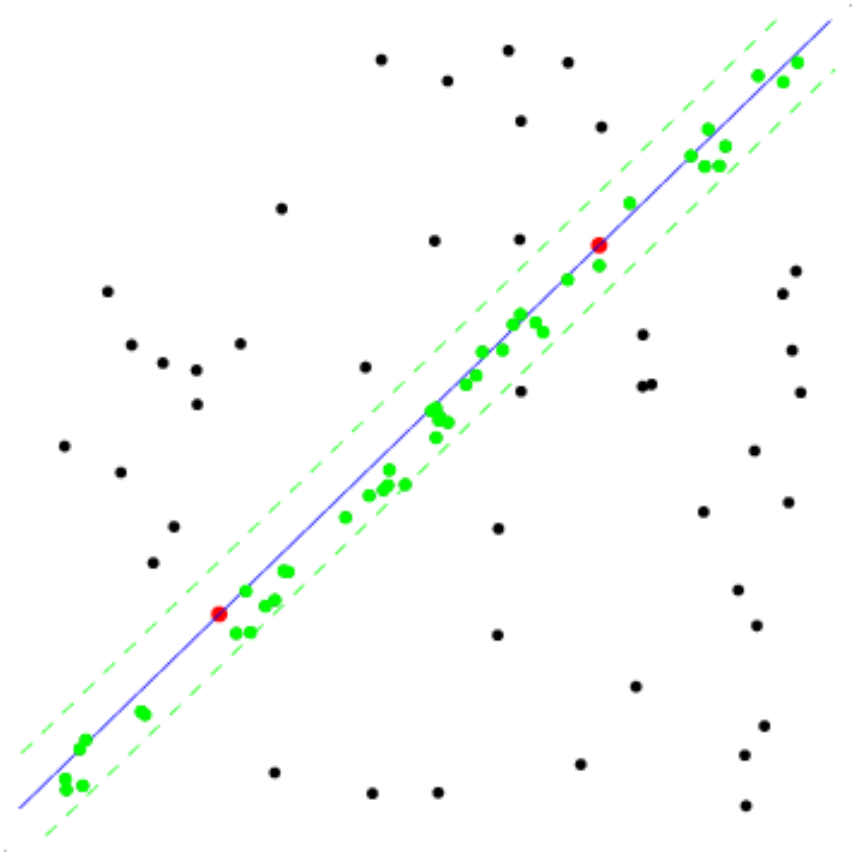
Slide credit: O. Chum



Slide credit: O. Chum



Slide credit: O. Chum



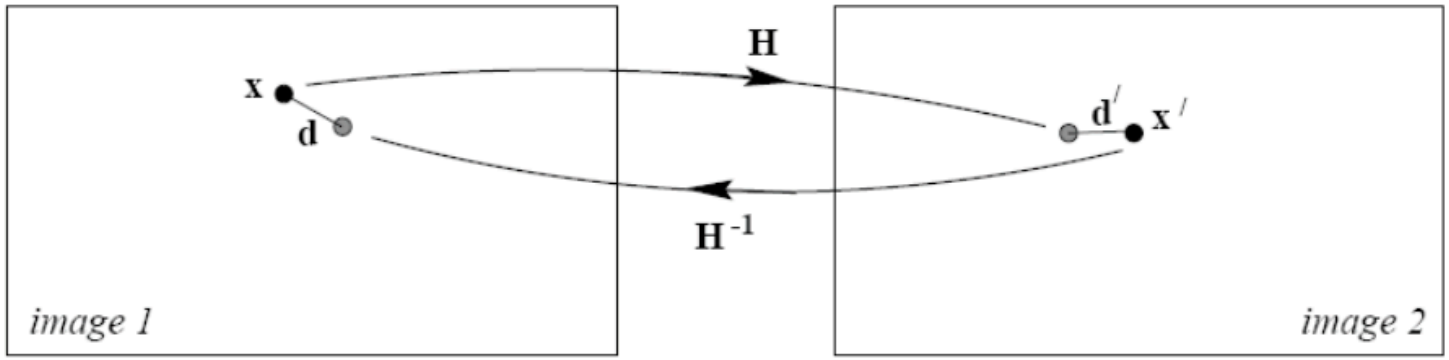
Slide credit: O. Chum

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. $d^2_{\text{transfer}} < t$)

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

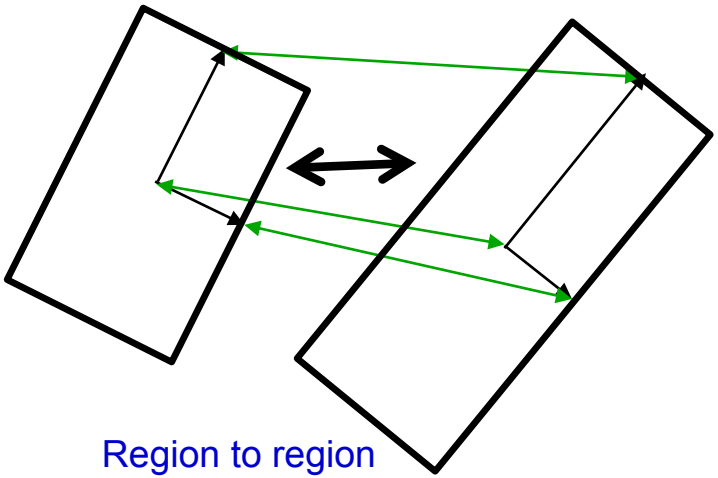


Choose the (H,t) with the largest number of inliers

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

How many samples are needed?

1. Depends on the proportion of outliers.
2. Depends on the sample size “s”
   - 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).



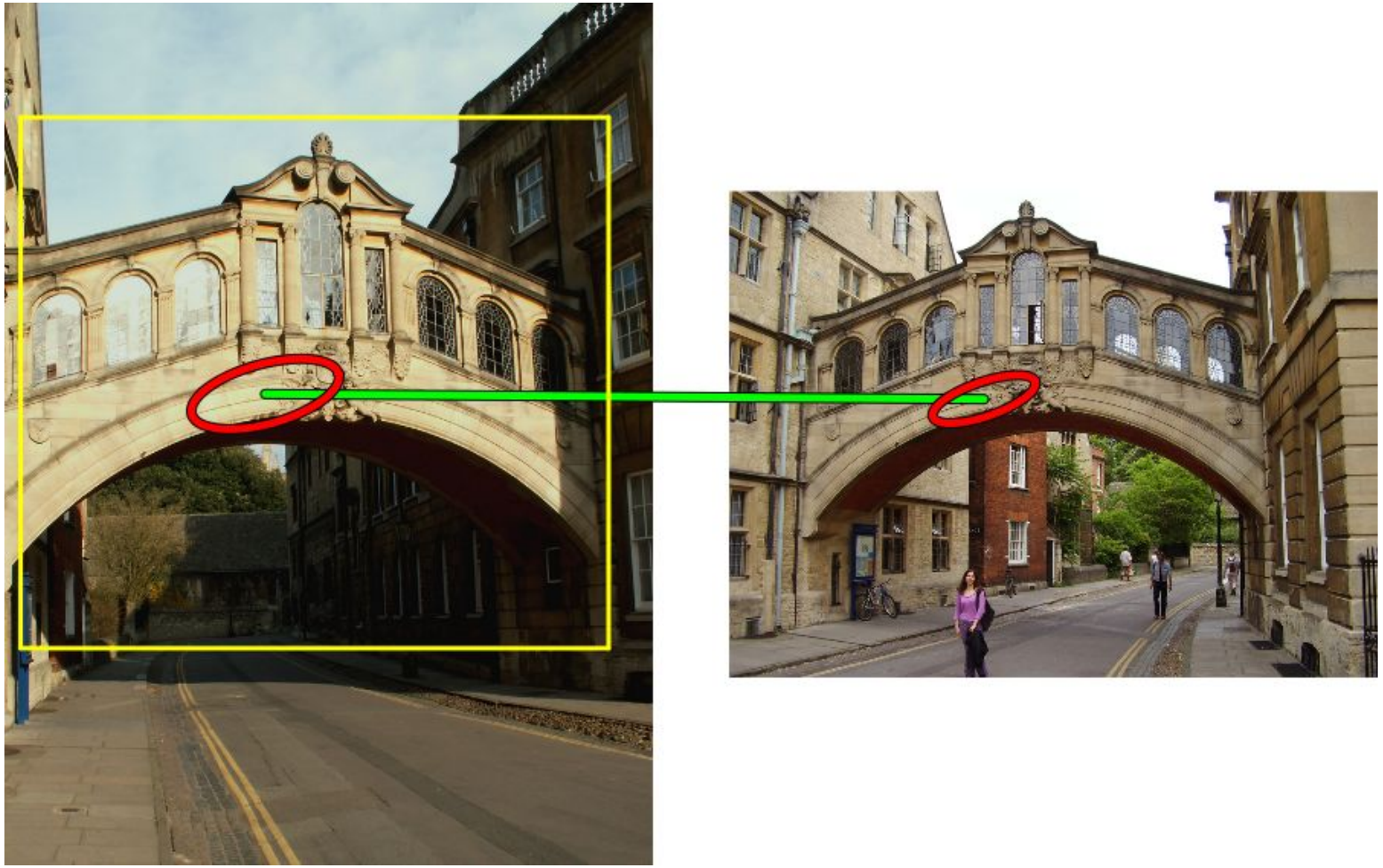
Region to region correspondence

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

Example: restricted affine transform

1. Test each correspondence



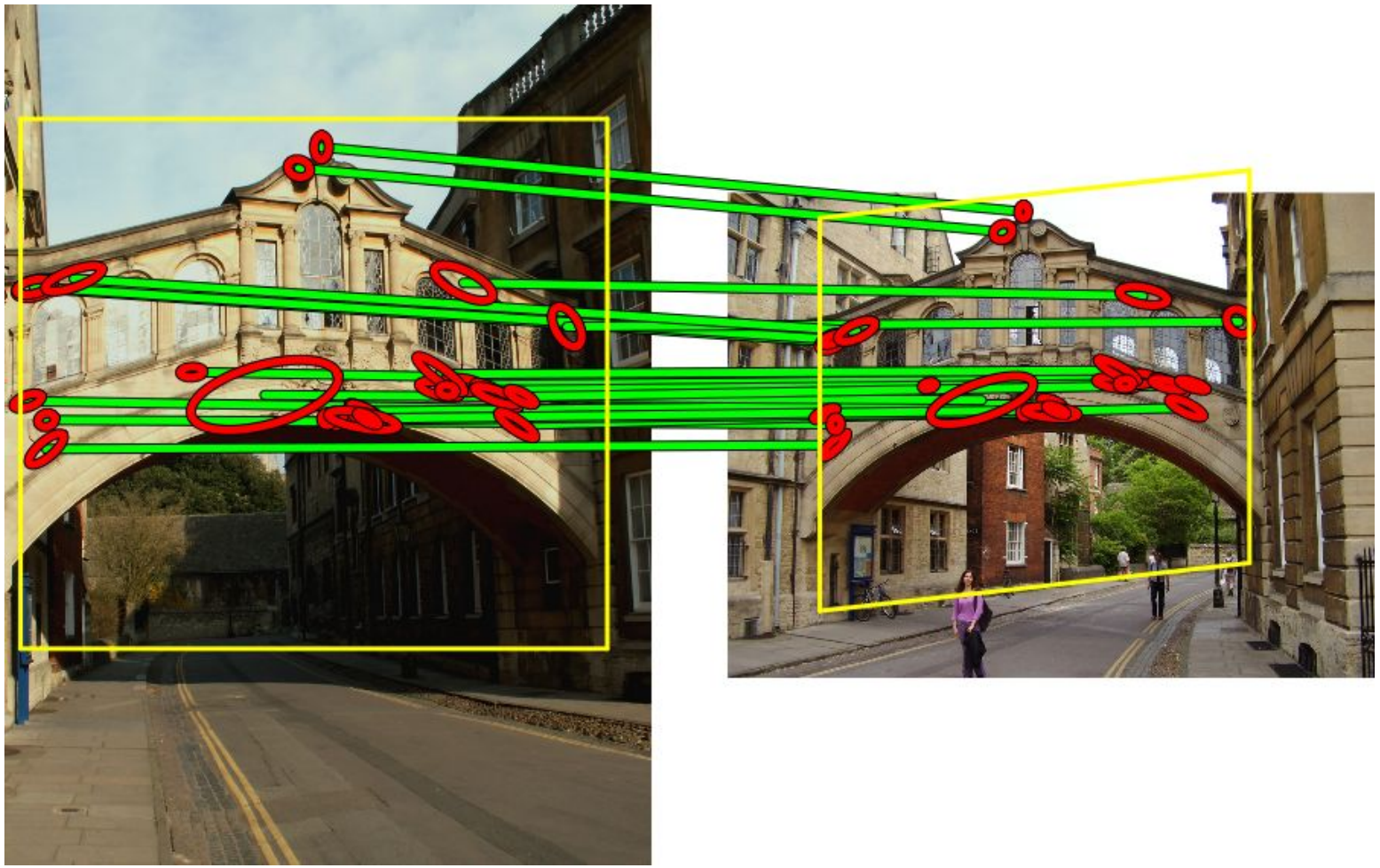
Example: restricted affine transform

2. Compute a (restricted) planar affine transformation (5 dof)

Need just one correspondence

Example: restricted affine transform

3. Score by number of consistent matches

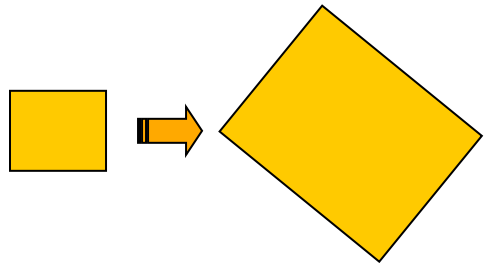


Re-estimate full affine transformation (6 dof)

Example II: (see practical later today)

Similarity transformation is specified by four parameters: scale factor s, rotation θ, and translations t_x and t_y.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = sR(\theta) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



Recall, each SIFT detection has: position (x_i, y_i), scale s_i, and orientation θ_i.

How many correspondences are needed to compute similarity transformation?

Example II: (see practical later today)

Compute similarity transformation from a single correspondence:

$$(x_A, y_A, s_A, \theta_A) \leftrightarrow (x'_A, y'_A, s'_A, \theta'_A)$$

$$\theta = \theta'_A - \theta_A$$

$$t_x = x'_A - x_A$$

$$t_y = y'_A - y_A$$

$$s = s'_A / s_A$$

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

Lebeda, Matas, Chum: Fixing the locally optimized RANSAC, BMVC’12 (code available).

Geometric verification for visual search (references)

Schmid and Mohr, Local gray-value invariants for image retrieval, PAMI 1997

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. CVPR (2007)

Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. CVPR (2009)

Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: CVPR (2009)

Jegou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV 87(3), 316–336 (2010)

Lin, Z., Brandt, J.: A local bag-of-features model for large-scale object retrieval. ECCV 2010)

Zhang, Y., Jia, Z., Chen, T.: Image retrieval with geometry preserving visual phrases. In: CVPR (2011)

Tolias, G., Avrithis, Y.: Speeded-up, relaxed spatial matching. In: ICCV (2011)

Shen, X., Lin, Z., Brandt, J., Avidan, S., Wu, Y.: Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In: CVPR. IEEE (2012)

H. Stewénius, S. Gunderson, J. Pilet. Size matters: exhaustive geometric verification for image retrieval, ECCV 2012.

Outline

1. Local invariant features (C. Schmid)

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

3. **Efficient visual search (J. Sivic)**

4. Very large scale visual indexing (C. Schmid)

Practical session – Instance-level recognition and search