Instance-level recognition

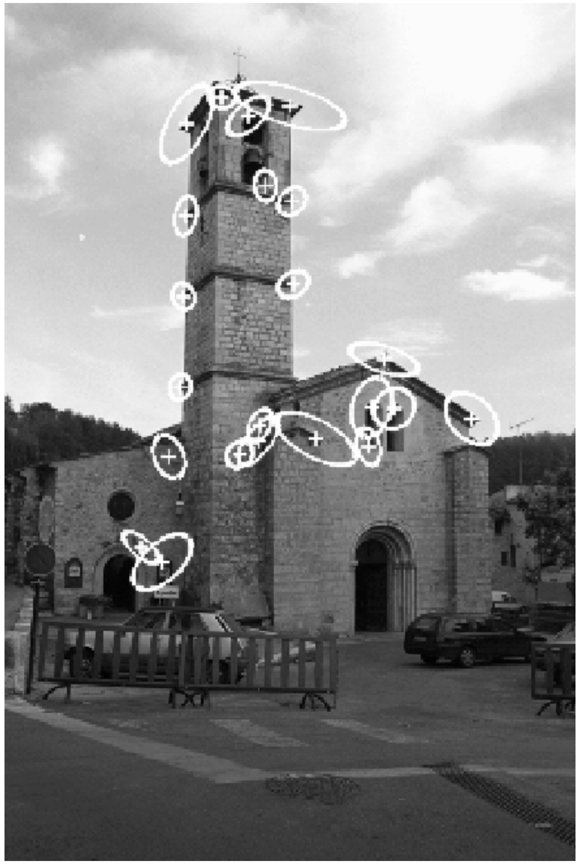
1) Local invariant features

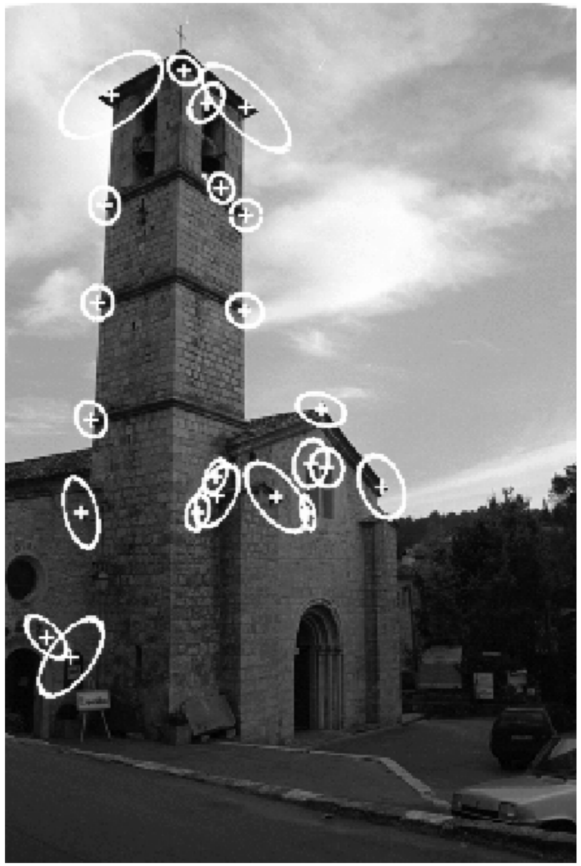
2) Matching and recognition with local features

3) Efficient visual search

4) Very large scale indexing

Matching of descriptors





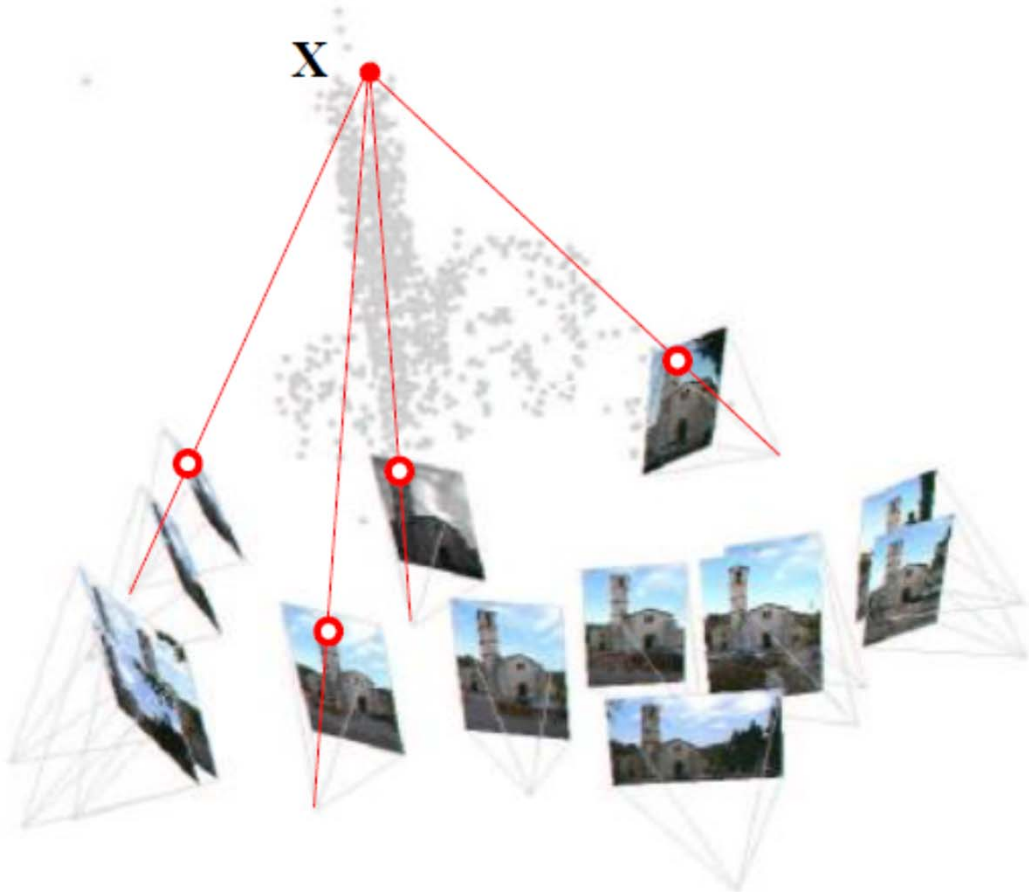
Matching and 3D reconstruction

- Establish correspondence between two (or more) images

[Schaffalitzky and Zisserman ECCV 2002]

Matching and 3D reconstruction

- Establish correspondence between two (or more) images



[Schaffalitzky and Zisserman ECCV 2002]

Building Rome in a Day

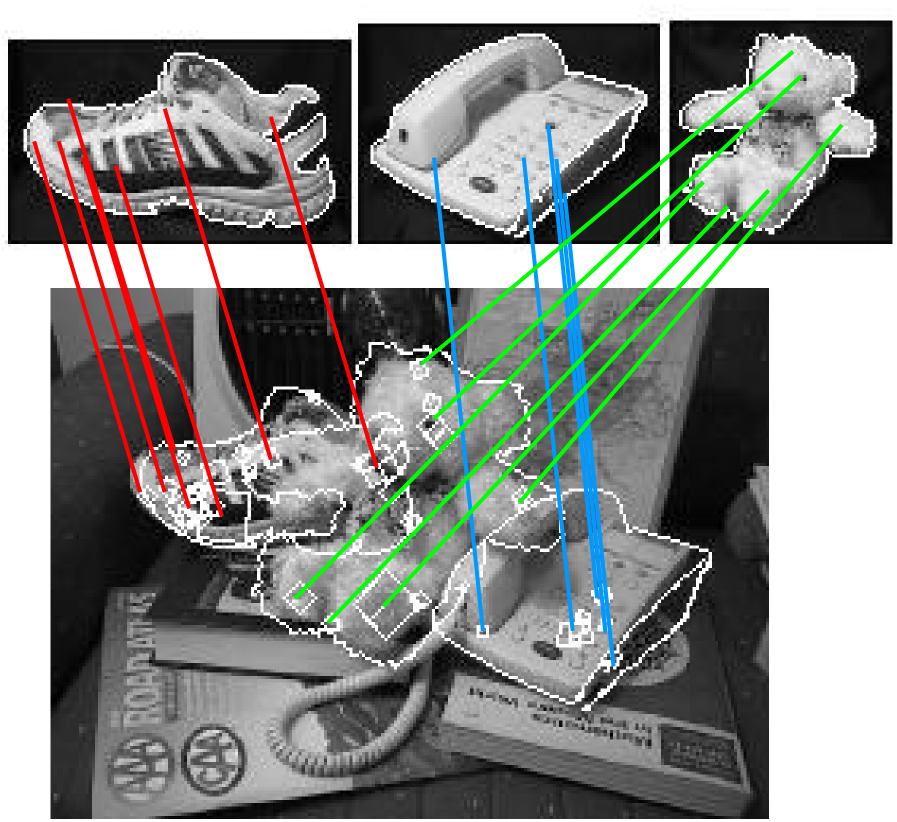
57,845 downloaded images, 11,868 registered images

[Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV'09]

Object recognition

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

Model database

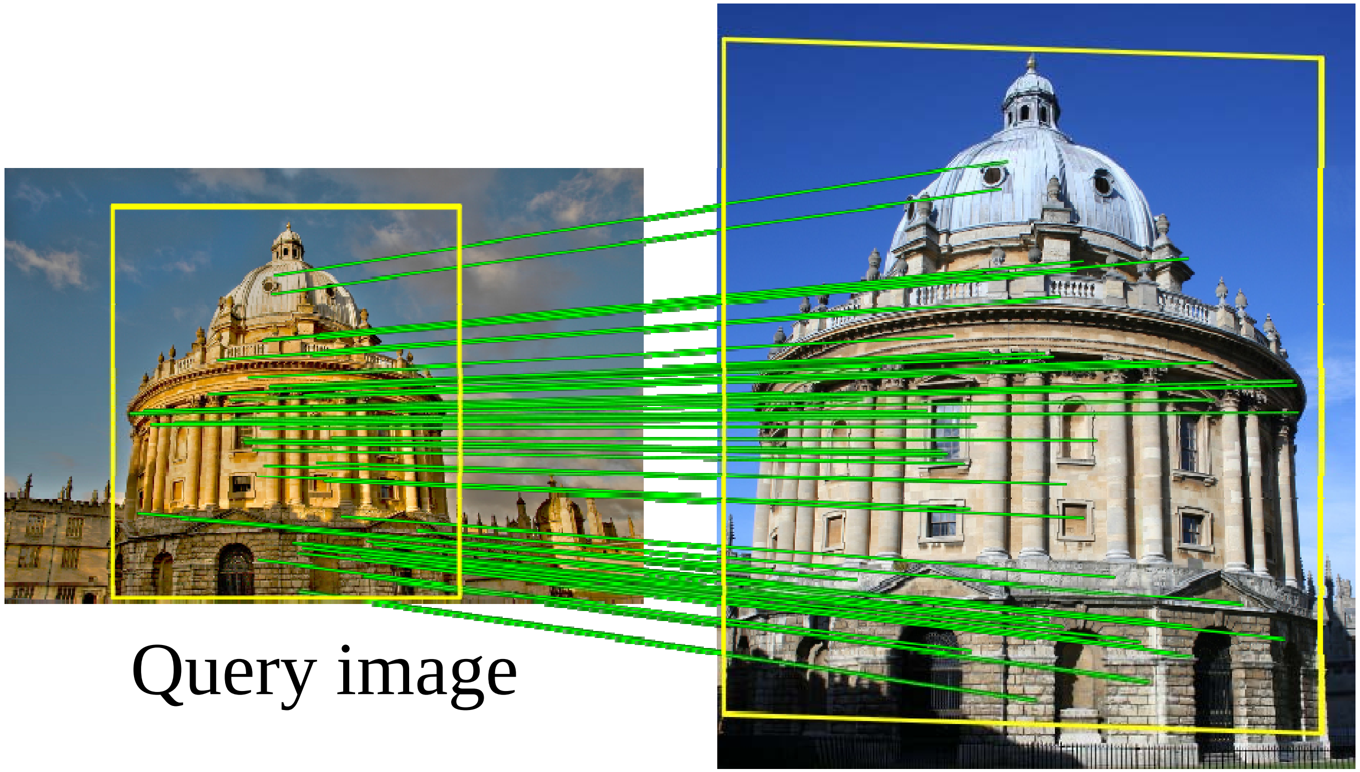


Target image

[D. Lowe, 1999]

Visual search

- Establish correspondence between the query image and all images from the database depicting the same object or scene

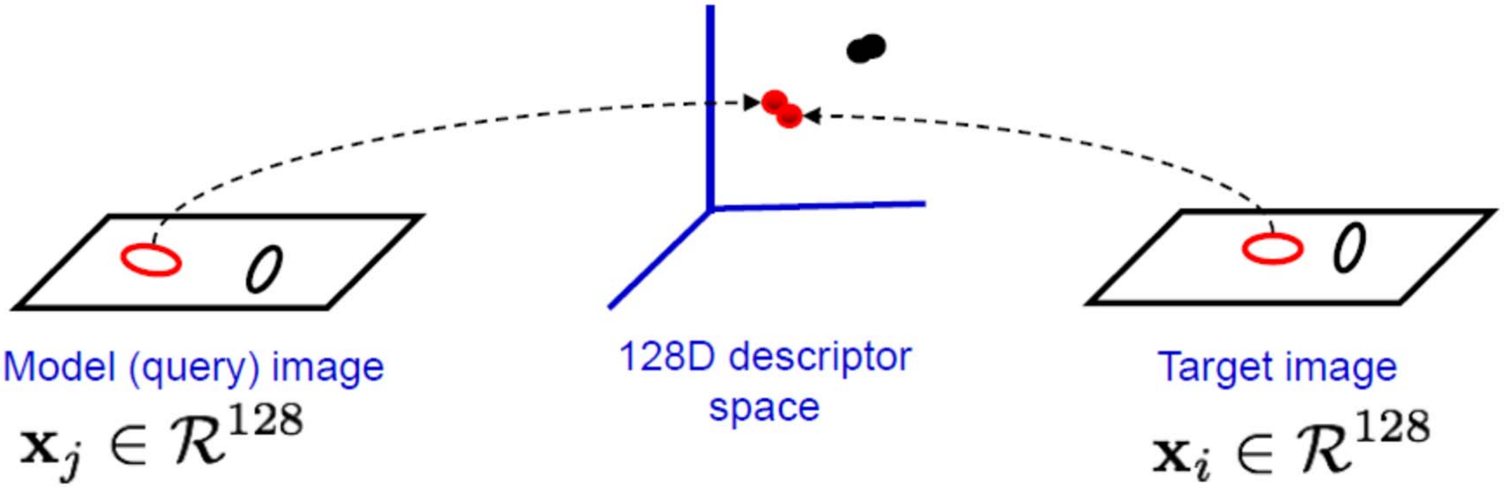


Query image

Database image(s)

Matching of descriptors

- Find the nearest neighbor in the second image for each descriptor, for example SIFT



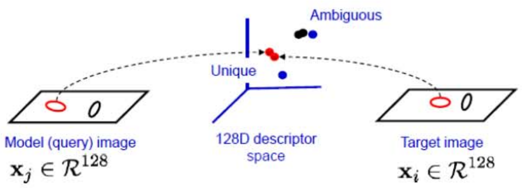
Need to solve some variant of the "nearest neighbor problem" for all feature vectors, $\mathbf{x}_j \in \mathcal{R}^{128}$, in the query image:

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

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

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1) [Lowe, '04]



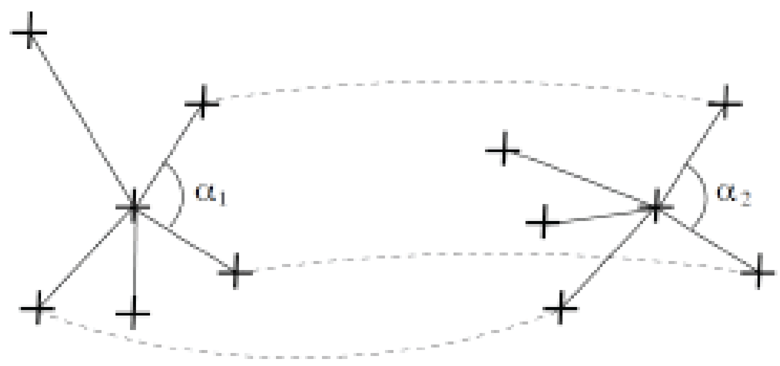
If the 2nd nearest neighbour is much further than the 1st nearest neighbour, the match is more "unique" or discriminative.

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

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

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)



Neighbors of the point have to match and angles have to correspond.
Note that in practice not all neighbors have to be matched correctly.

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)
  - Backwards matching (matches are NN in both directions)

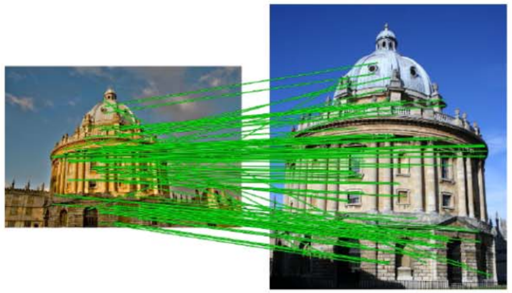
Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)
  - Backwards matching (matches are NN in both directions)

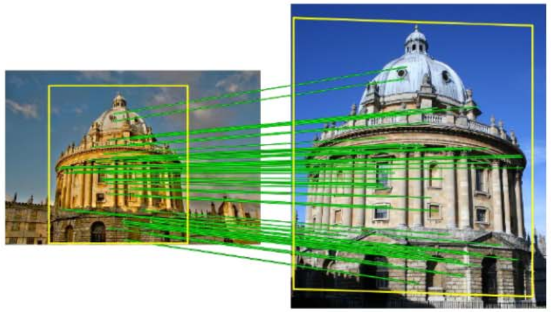
- Geometric verification with global constraint
  - All matches must be consistent with a global geometric transformation
  - However, there are many incorrect matches
  - Need to estimate simultaneously the geometric transformation and the set of consistent matches

Geometric verification with global constraint

- Example of a geometric verification

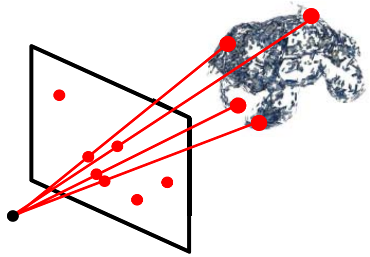


Tentative matches



Matches consistent with an affine transformation

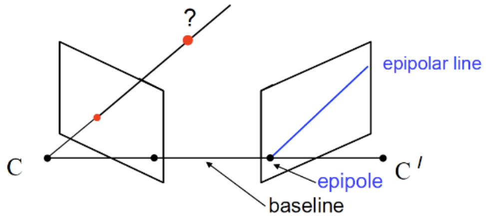
Examples of global constraints

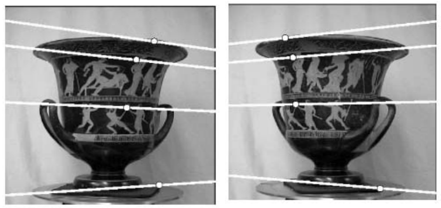


1 view and known 3D model.

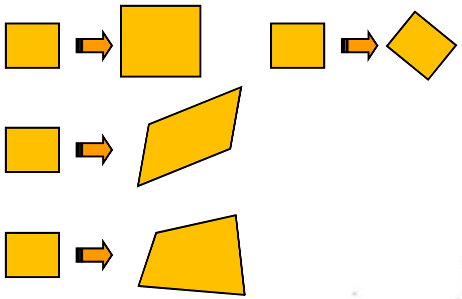
- Consistency with a (known) 3D model.

2 views





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



N-views

Are images consistent with a 3D model?

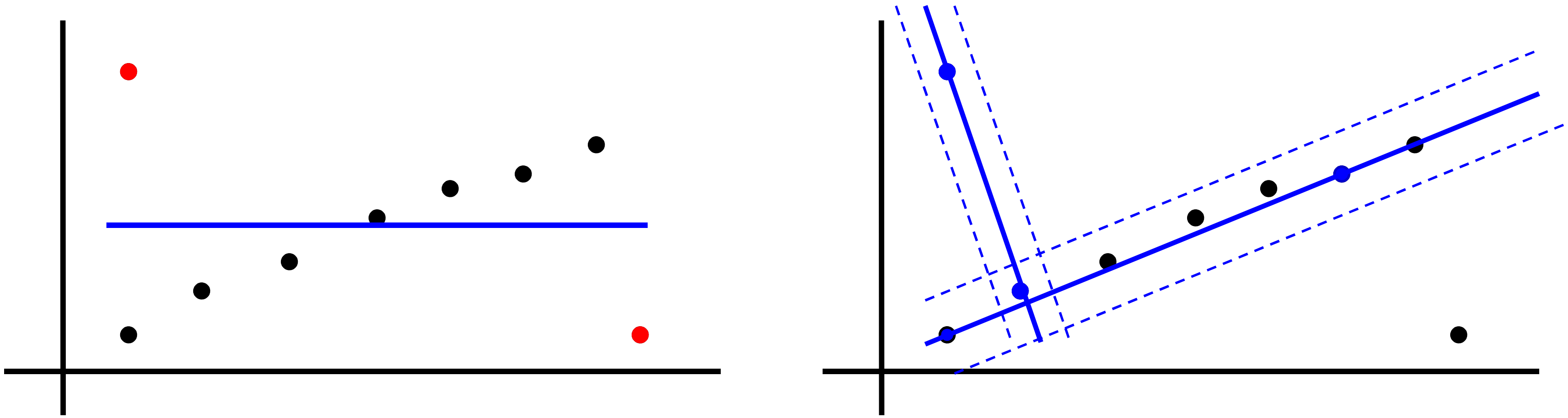
Matching of descriptors

- Geometric verification with global constraint
  - All matches must be consistent with a global geometric transformation
  - However, there are many incorrect matches
  - Need to estimate simultaneously the geometric transformation and the set of consistent matches

- Robust estimation of global constraints
  - RANSAC (RANdom Sampling Consensus) [Fishler&Bolles'81]
  - Hough transform [Lowe'04]

RANSAC: Example of robust line estimation

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

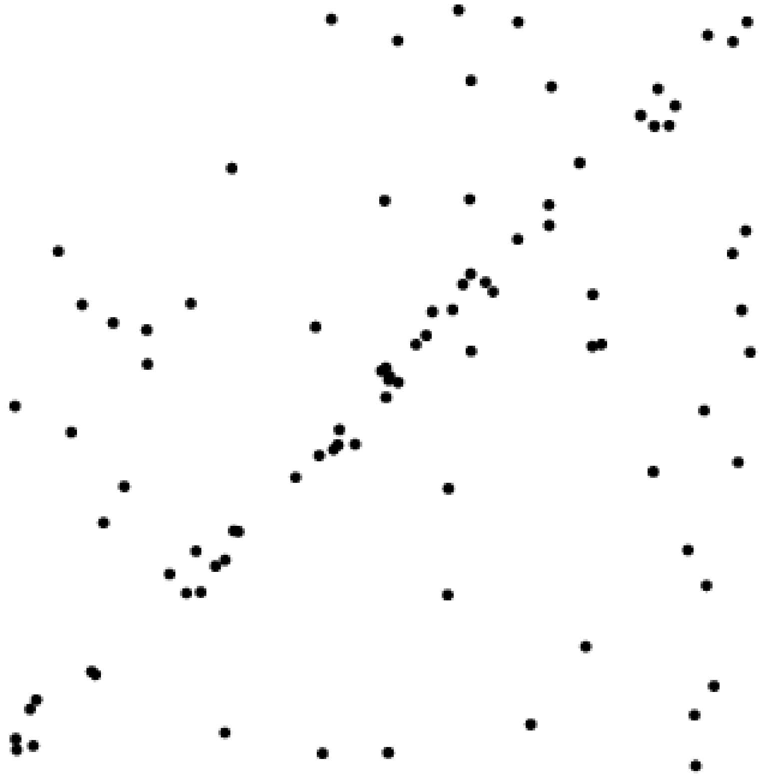
Repeat

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)

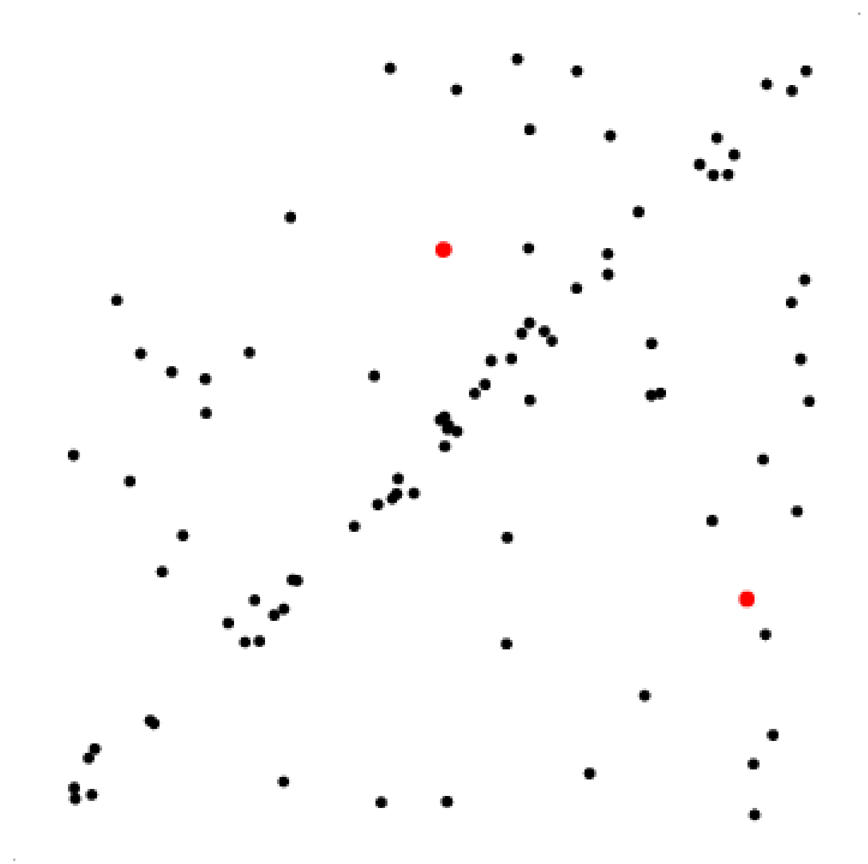
Choose the line with the largest number of inliers

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

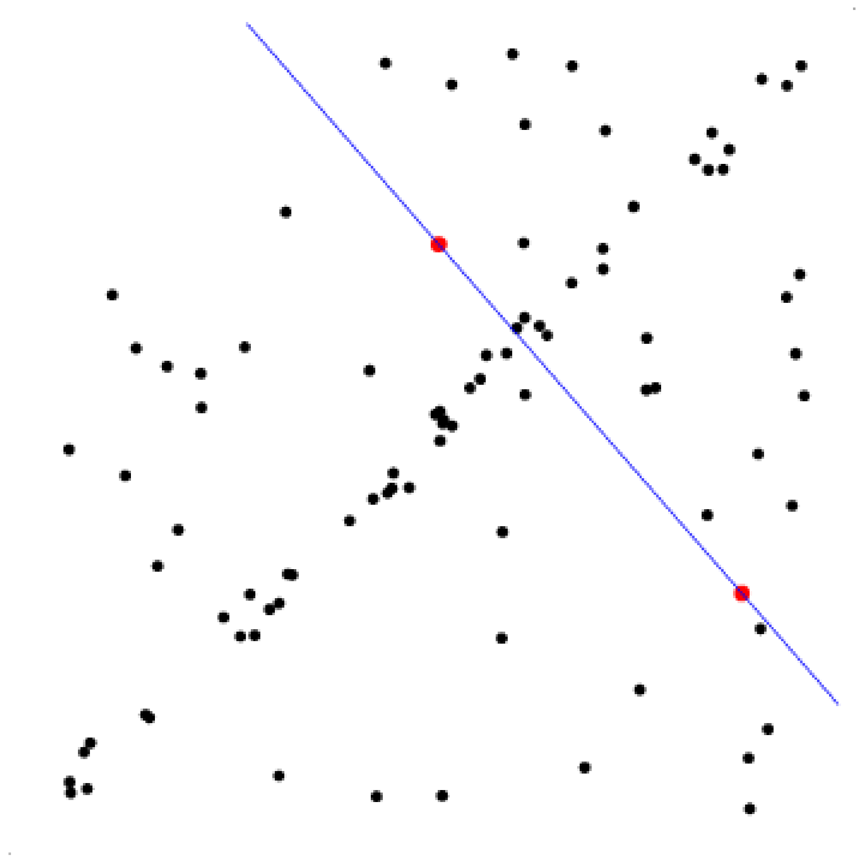
Slide credit: A. Zisserman



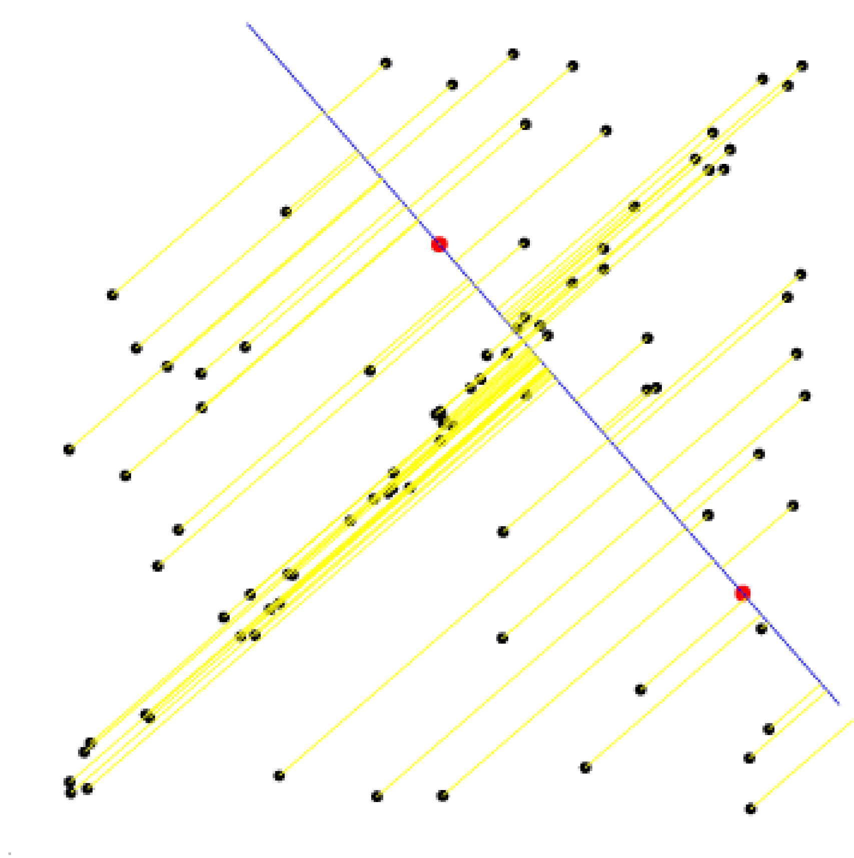
Slide credit: O. Chum



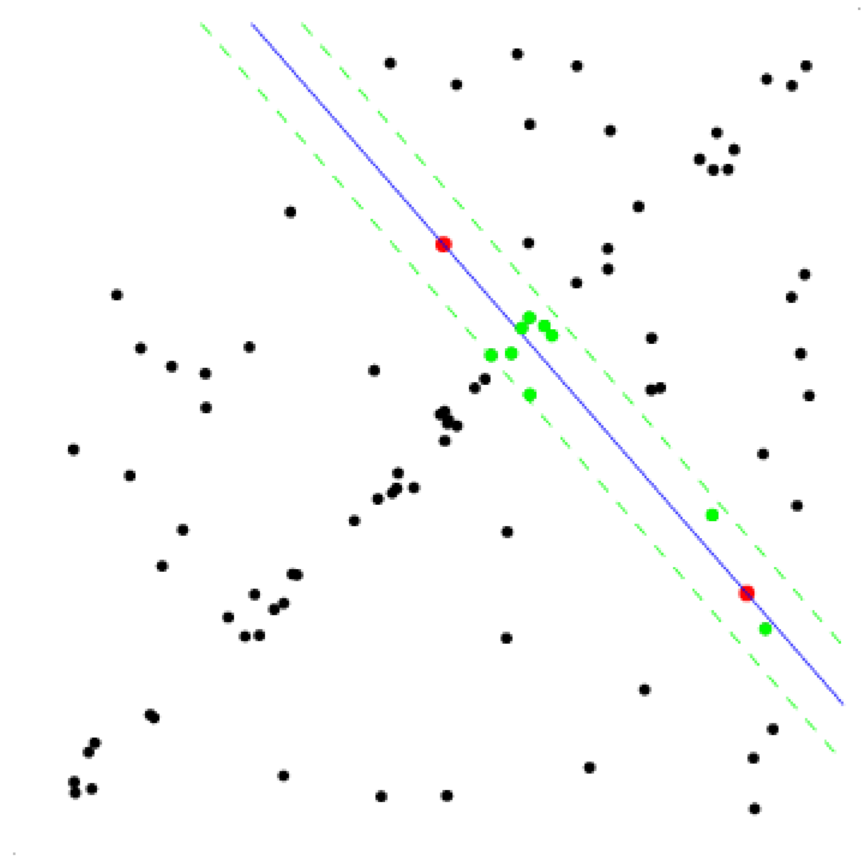
Slide credit: O. Chum



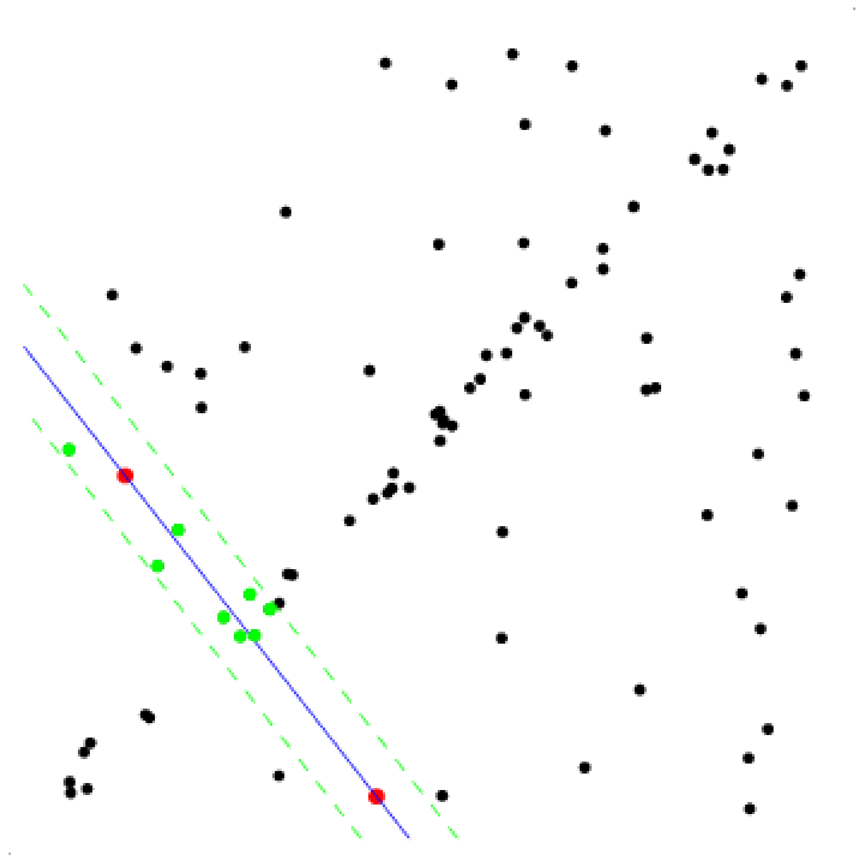
Slide credit: O. Chum



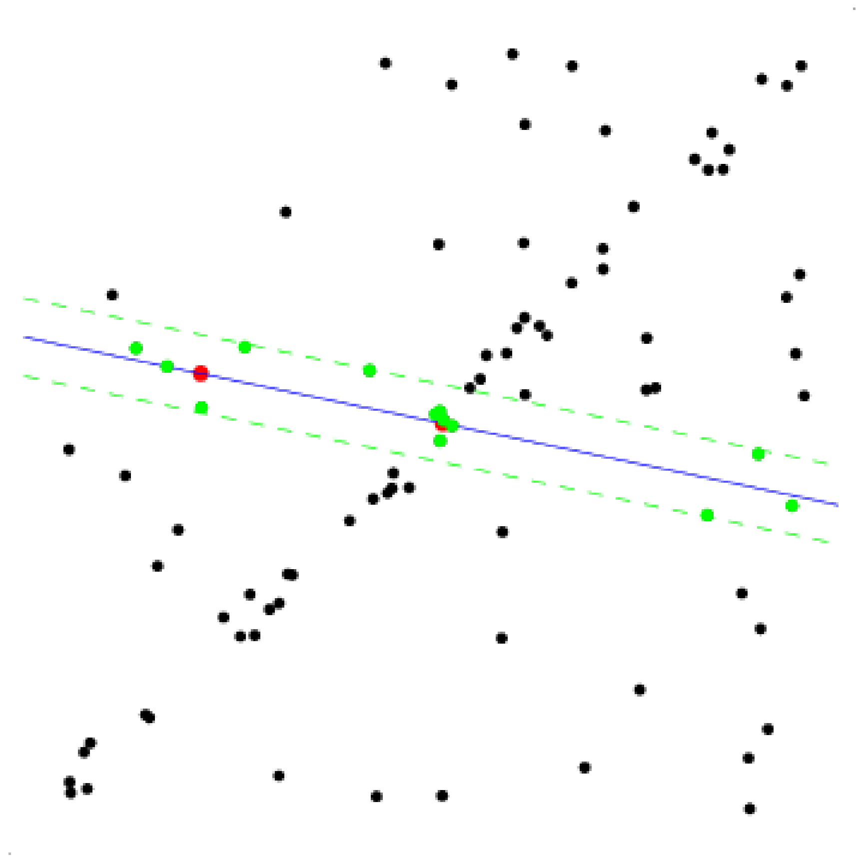
Slide credit: O. Chum



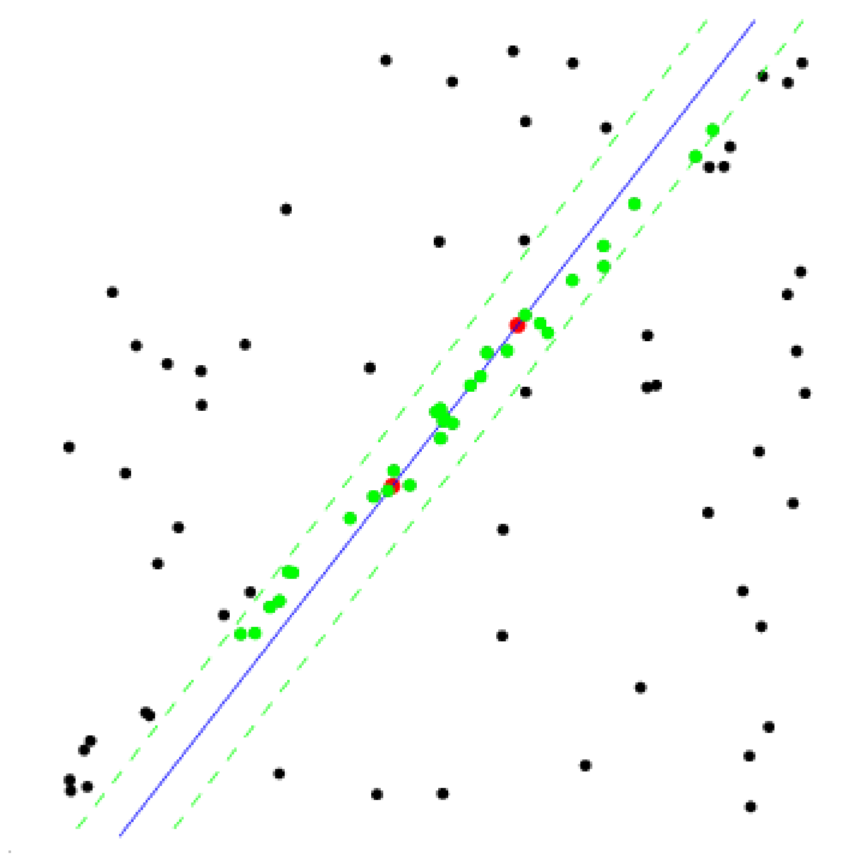
Slide credit: O. Chum



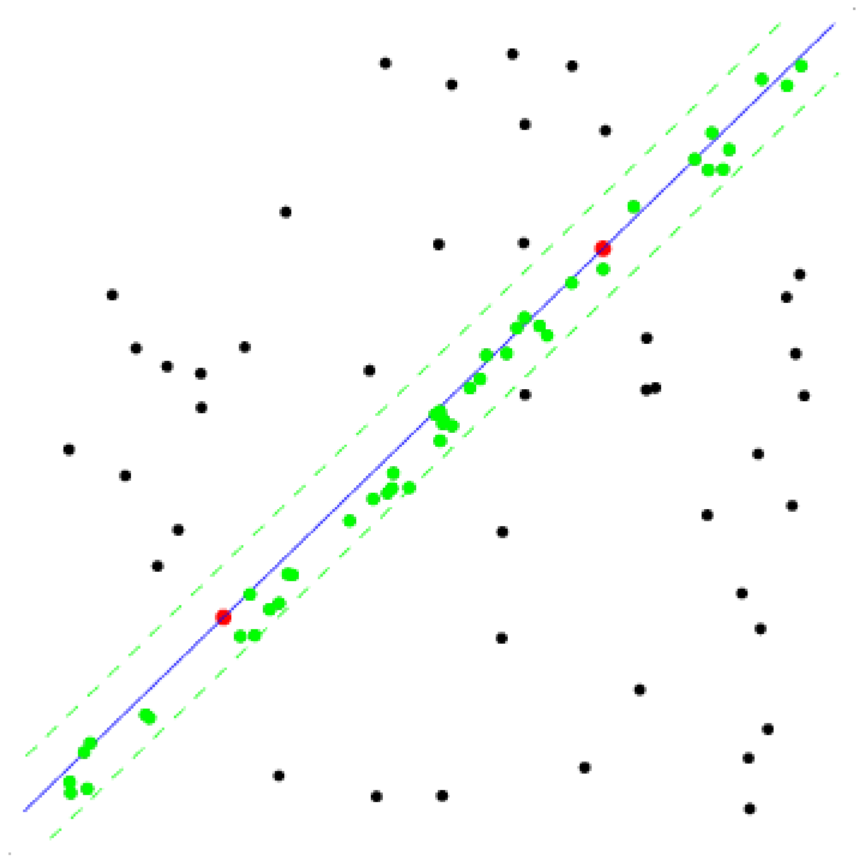
Slide credit: O. Chum



Slide credit: O. Chum



Slide credit: O. Chum

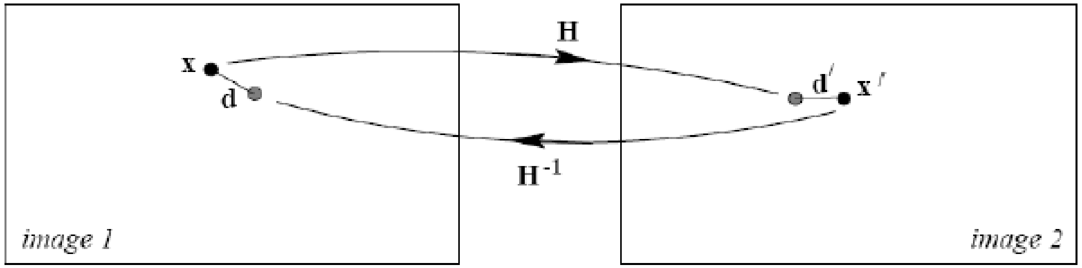


Slide credit: O. Chum

Algorithm RANSAC

- Robust estimation with RANSAC of a homography
  - Repeat
    - Select 4 point matches
    - Compute 3x3 homography
    - Measure support (number of inliers within threshold, i.e. $d^2_{transfer} < t$)

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



  - Choose (H with the largest number of inliers)
  - Re-estimate H with all inliers

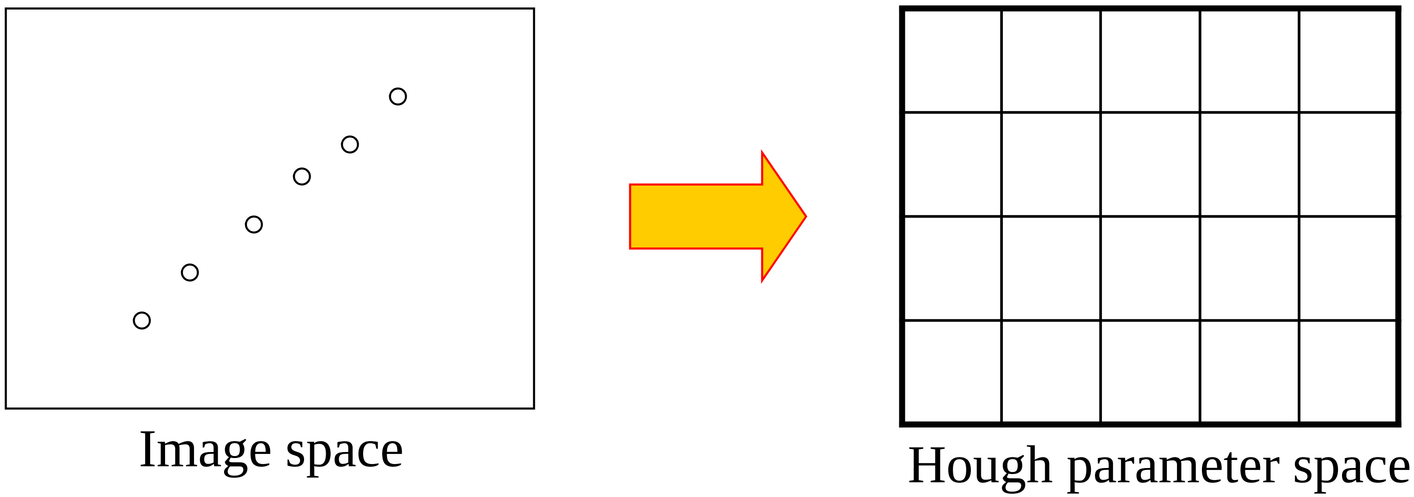
Matching of descriptors

- Geometric verification with global constraint
  - All matches must be consistent with a global geometric transformation
  - However, there are many incorrect matches
  - Need to estimate simultaneously the geometric transformation and the set of consistent matches

- Robust estimation of global constraint
  - RANSAC (RANdom Sampling Consensus) [Fishler&Bolles'81]
  - **Hough transform [Lowe'04]**

Strategy 2: Hough transform

- General outline:
  - Discretize parameter space into bins
  - For each feature point in the image, put a vote in every bin in the parameter space that could have generated this point
  - Find bins that have the most votes



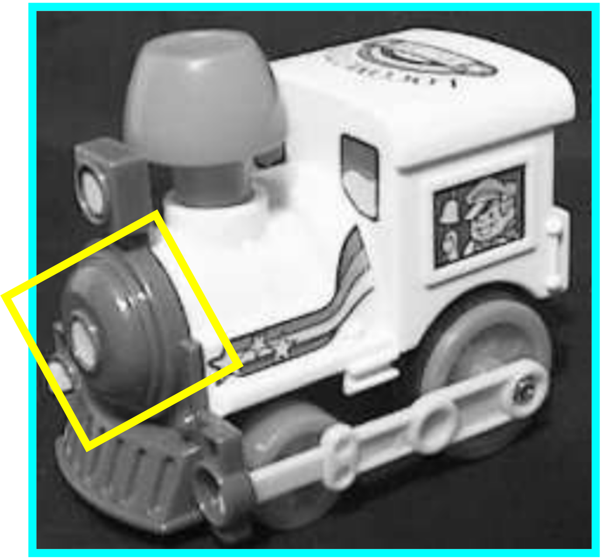
P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

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

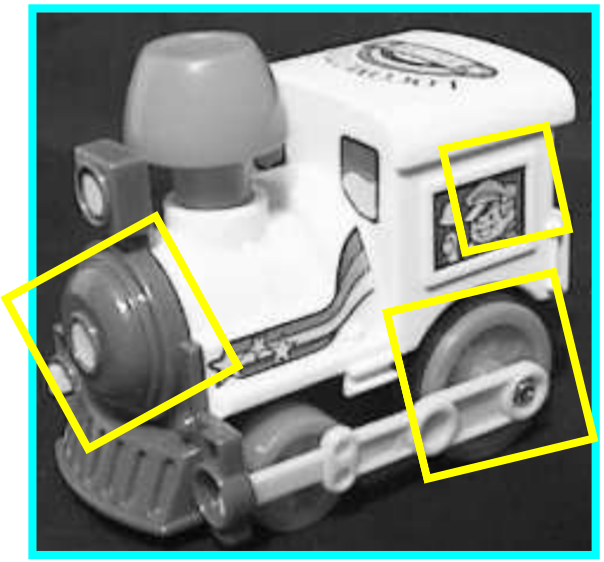
David G. Lowe. **"Distinctive image features from scale-invariant keypoints",** *IJCV* 60 (2), pp. 91-110, 2004.

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.

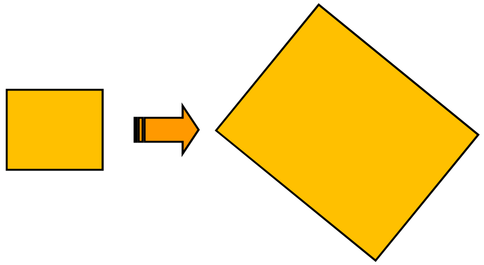
model



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

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?

Compute similarity transformation from a single correspondence:

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

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

$$s = s'_A / s_A$$

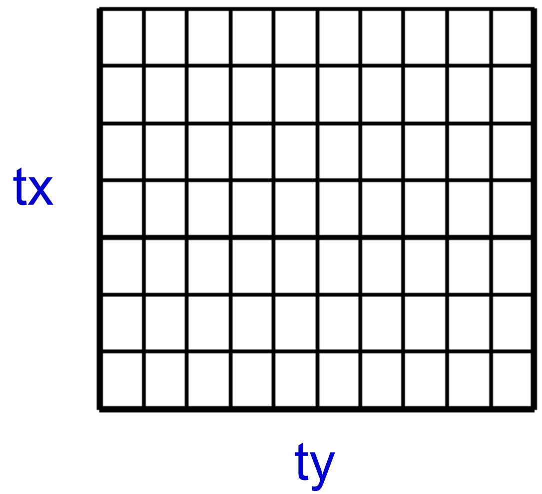
$$t_x = x'_A - sR(\theta)x_A$$

$$t_y = y'_A - sR(\theta)y_A$$

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

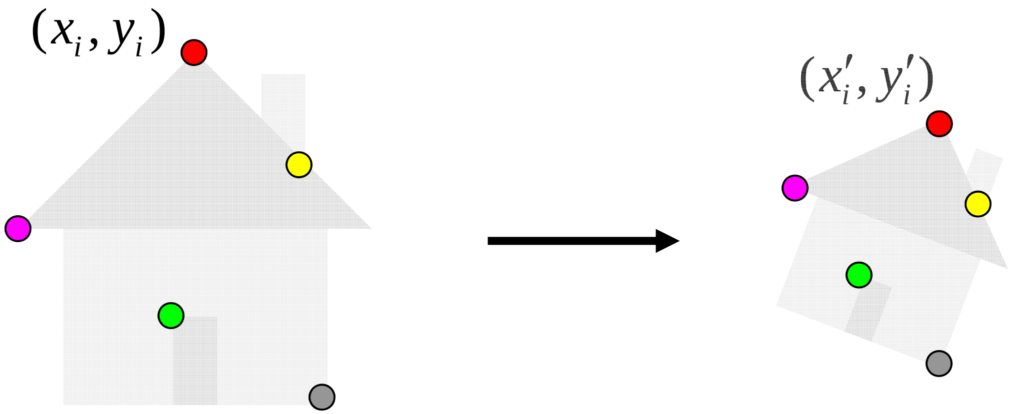
H: 4D-accumulator array
(only 2-d shown here)



- 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).

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

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]

Summary

Finding correspondences in images is useful for

- Image matching, panorama stitching
- Object recognition
- Large scale image search: next part of the lecture

Beyond local point matching

- Semi-local relations
- Global geometric relations:
  - Epipolar constraint
  - 3D constraint (when 3D model is available)
  - 2D tnfs: Similarity / Affine / Homography
- Algorithms:
  - RANSAC
  - Hough transform

$$\boxed{\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0}$$

$$\boxed{\mathrm{x} = \mathrm{PX}}$$

$$\boxed{\mathrm{x}' = \mathrm{Hx}}$$