Specific object recognition

Vittorio Ferrari

Slides by V. Ferrari, L. Van Gool and others

Three main schools

Model-based

- relate image features to model features
- simplify the relation through invariance
- this can keep the modeling effort low
- complexity of objects limited
- can deal with cluttered scenes

Image-based (appearance based)

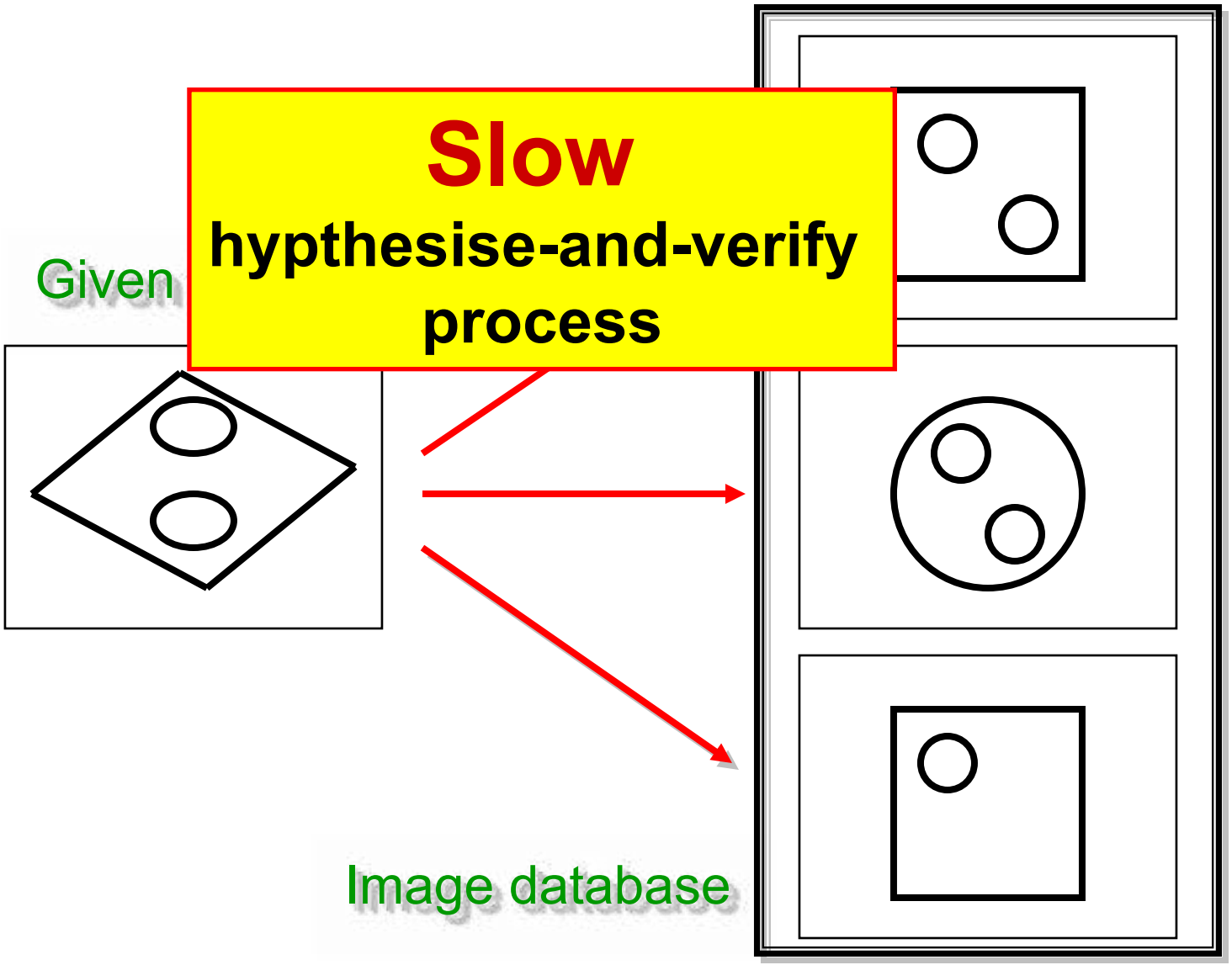
- `model' = all possible views
- acquiring representative data set cumbersome
- can deal with complex objects
- difficulties with variable backgrounds

Hybrid approaches (most of this lecture)

Model-based

Once upon a time...

comparing image features with features in database, trying to figure out type + pose



More recent Example: recognition of planar shapes

- under affine distortions
- using invariant signatures of the outlines

NB invariant-based hashing as a first step
can quickly eliminate the majority of candidates;
the signatures are then used for a final selection and validation

Invariants under affine transformations (one example)

ratios of areas

8 data items – 6 parameters affine transf. params
$\rightarrow$ 2 independent, absolute invariants

affine *invariant coordinates* (x_A, y_A) :

$$x_A = \frac{\left|\bar{x} - \bar{x}_2 \quad \bar{x} - \bar{x}_3\right|}{\left|\bar{x}_1 - \bar{x}_3 \quad \bar{x}_2 - \bar{x}_3\right|} \qquad y_A = \frac{\left|\bar{x} - \bar{x}_3 \quad \bar{x} - \bar{x}_1\right|}{\left|\bar{x}_1 - \bar{x}_3 \quad \bar{x}_2 - \bar{x}_3\right|}$$

|.|: matrix determinant

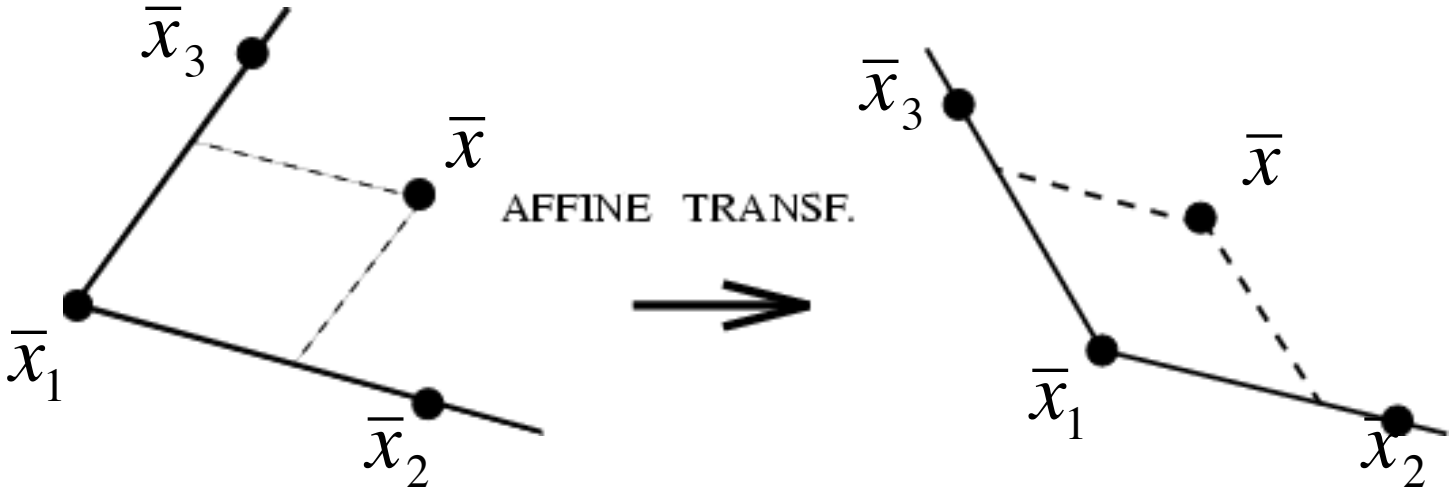


Image-based

Appearance manifold approach

Training (Nayar et al. '96)

for every object :
- sample the set of viewing conditions (mainly viewpoints)
- use these images as feature vectors (after bounding-boxing, rescaling, brightness normalization)
- apply a PCA over all the images (*directly* on the images)
- keep the dominant PCs (10-20 enough already)
- sequence of views for 1 object represent a manifold in space of projections (fit datapoints with splines, then densely resample if desired)

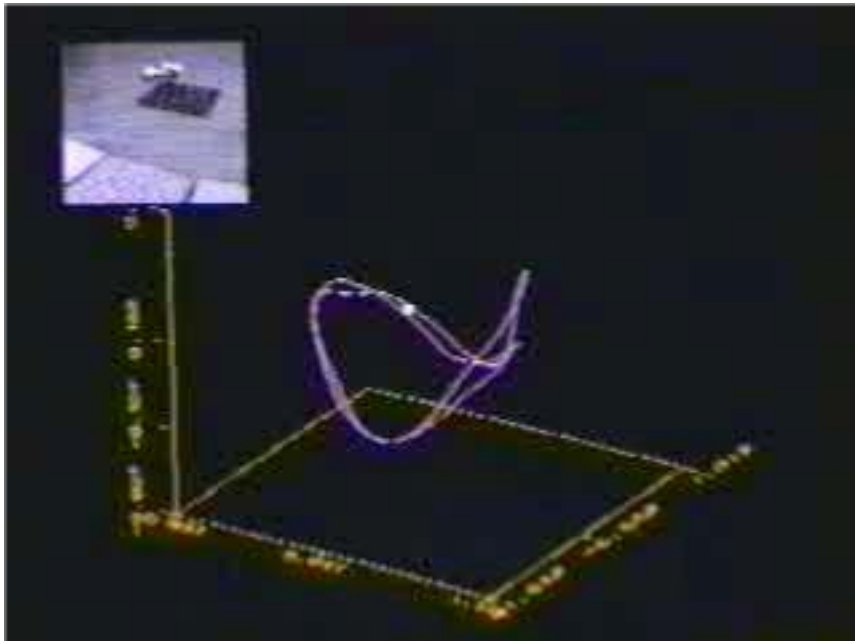
Recognition

what is the nearest manifold to a test image?

Object-pose manifold

Appearance changes projected on PCs (1D pose changes about vertical axis → manifolds are 1D curves, not surfaces)

Sufficient characterization for recognition and pose estimation



Real-time system

(Nayar et al. ‘96)

Real-time system

(Nayar et al. ‘96)

+ generic, any kind of object (3D shapes, …)

+ training = just take images (no sophisticated models)

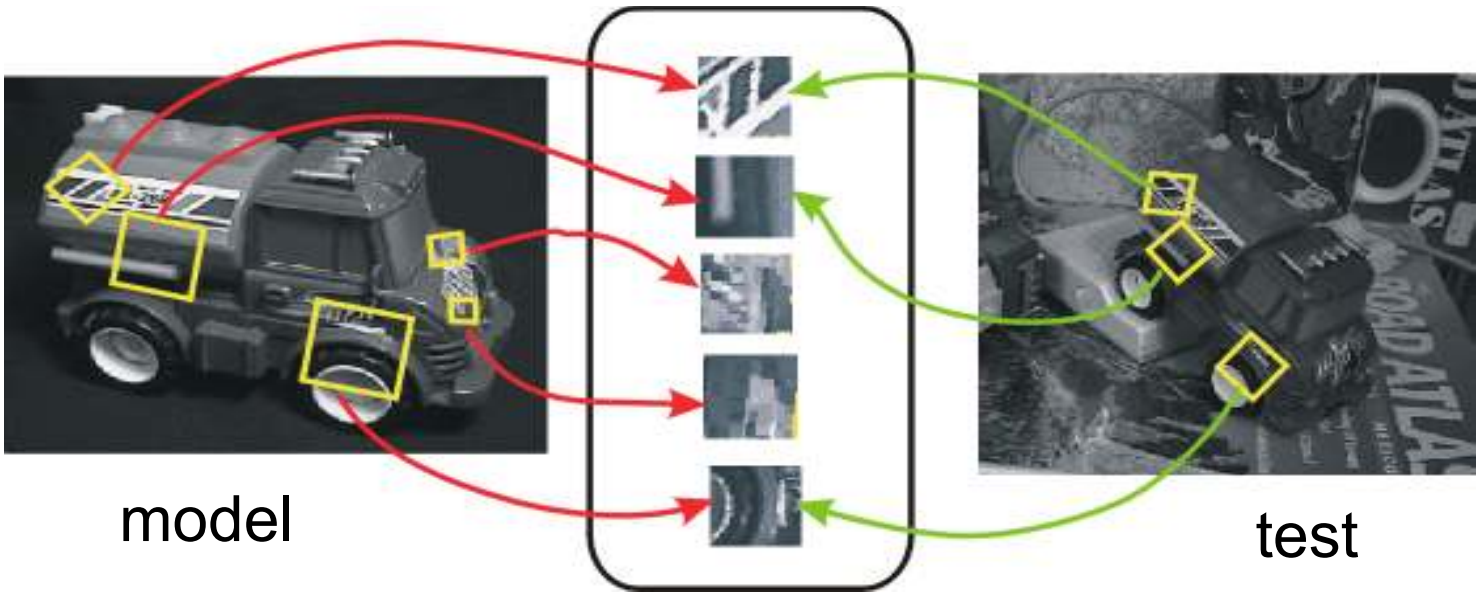
- painful to build large database of images for each and every object (sometimes just not possible)

- need for a clean background (cannot handle clutter)

Hybrid techniques

Basic algorithm

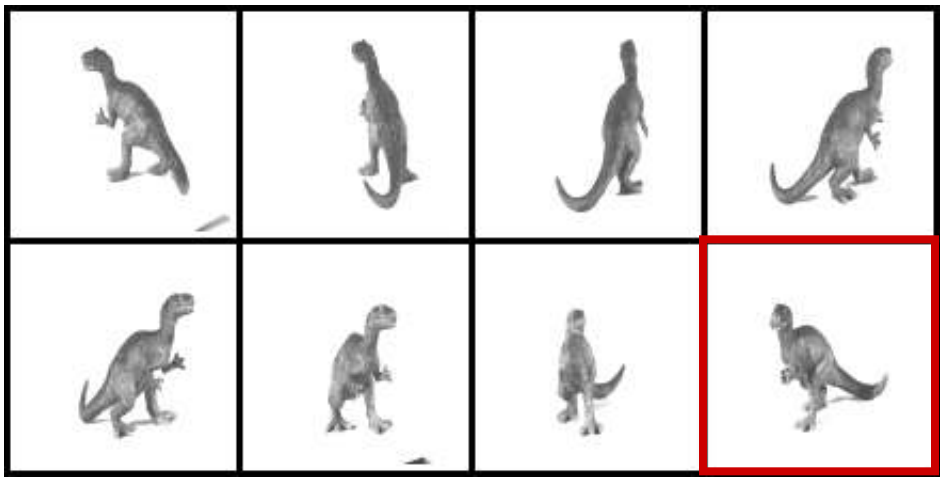
- Detect and describe features in model image(s)
  (done: scale invariant features; next: affine invariant ones)
- Detect and describe features for test image
- Match features (including geometric verification, e.g. RANSAC)
- Count matches
  → many? → object recognized and localized
  → few? → object not present in test image



Images: D. Lowe

Example

Training examples

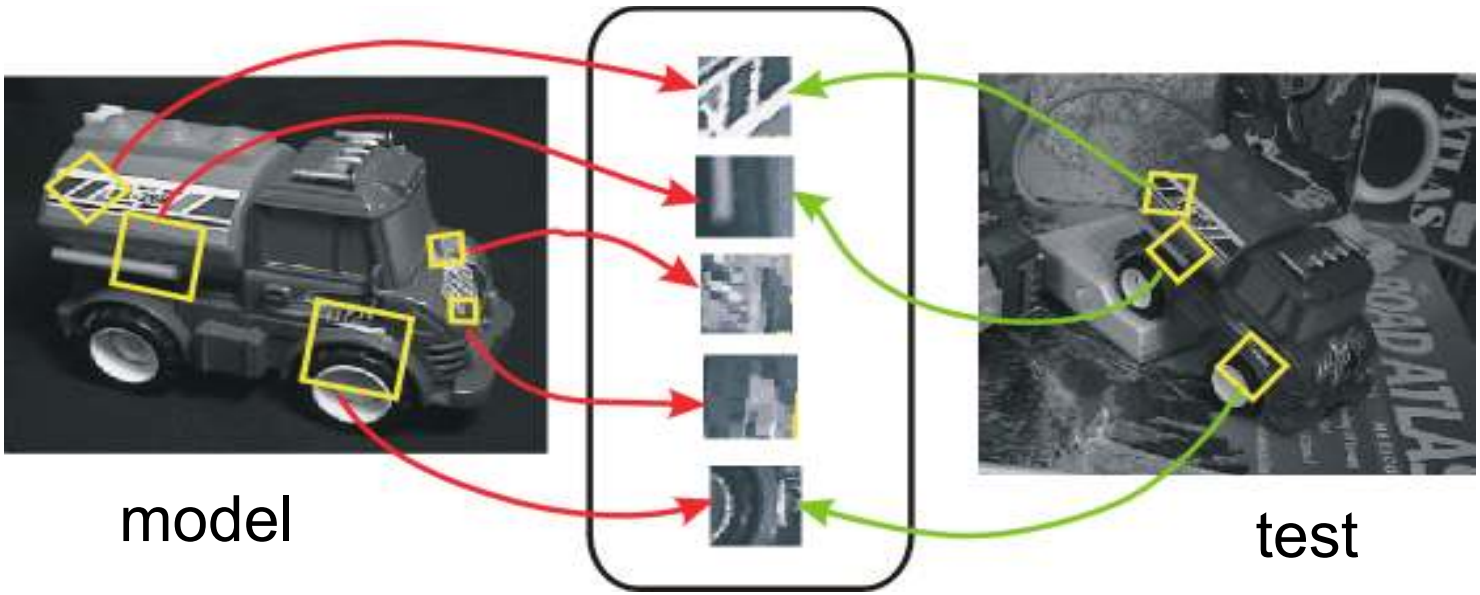


Test image

+ deal with cluttered background
+ need less training images
~ problems with uniform objects

Basic algorithm

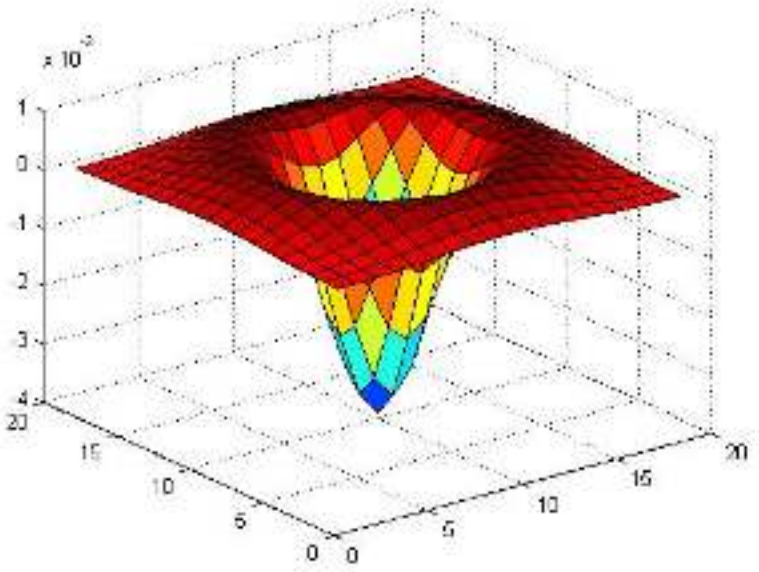
- **Detect** and describe features in model image(s)
  (done: **scale invariant features**; next: affine invariant ones)
- Detect and describe features for test image
- Match features (including geometric verification, e.g. RANSAC)
- Count matches
  → many? → object recognized and localized
  → few? → object not present in test image

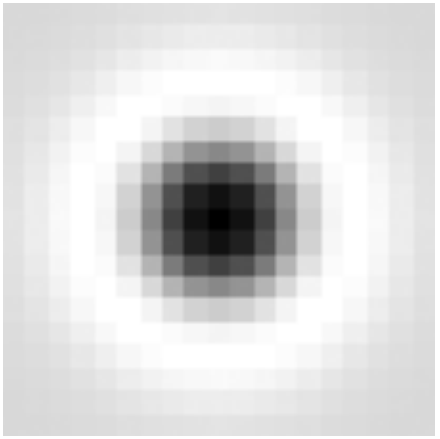


Images: D. Lowe

Blob detection in 2D

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

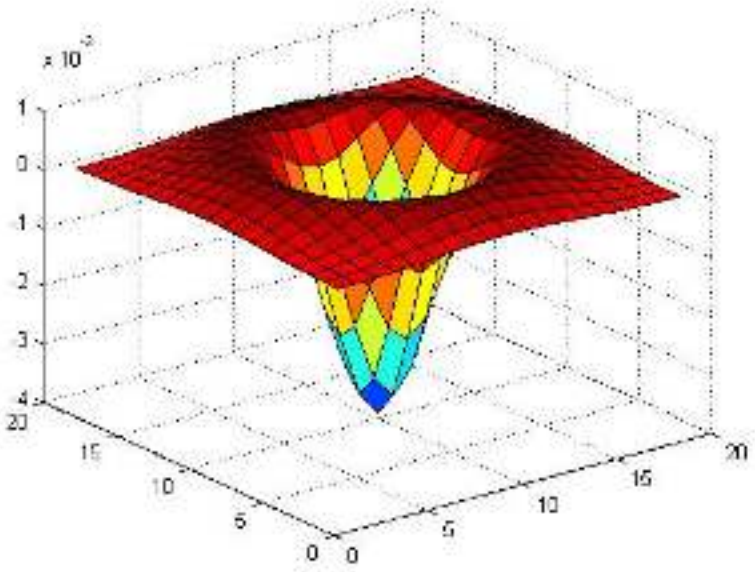


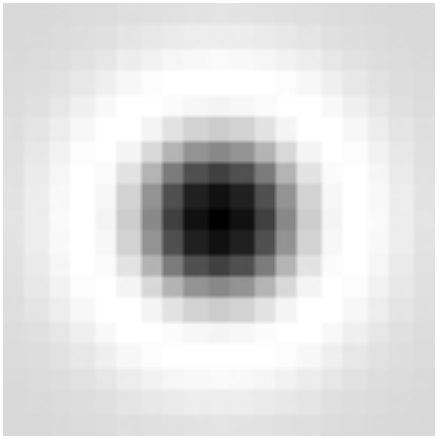


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection in 2D

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

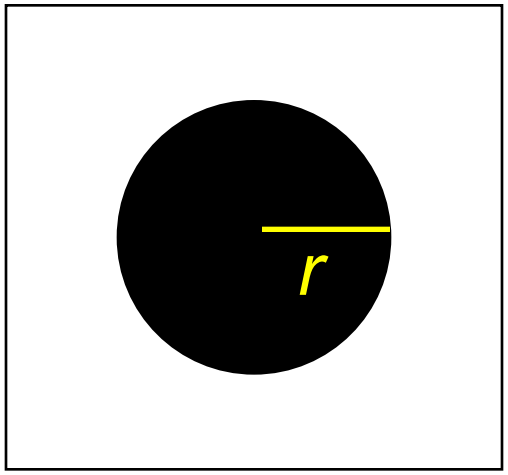




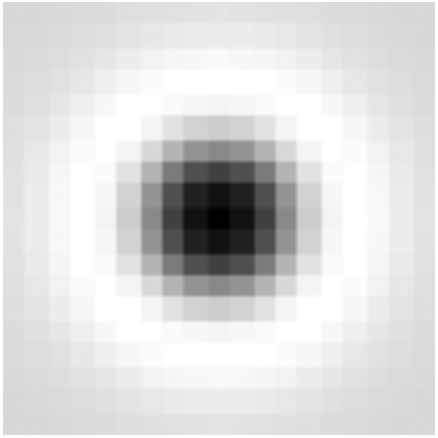
Scale-normalized: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

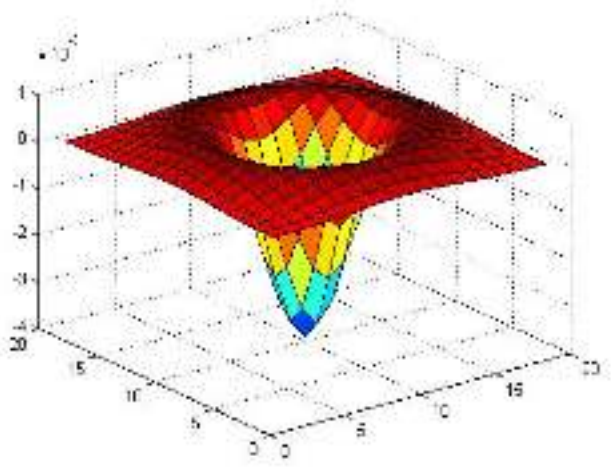
Scale selection

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?



image





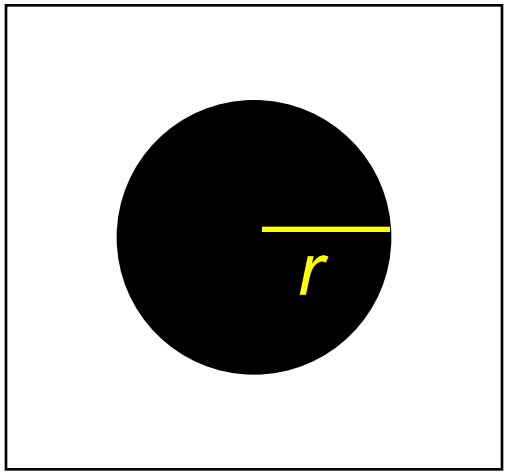
Laplacian

Scale selection

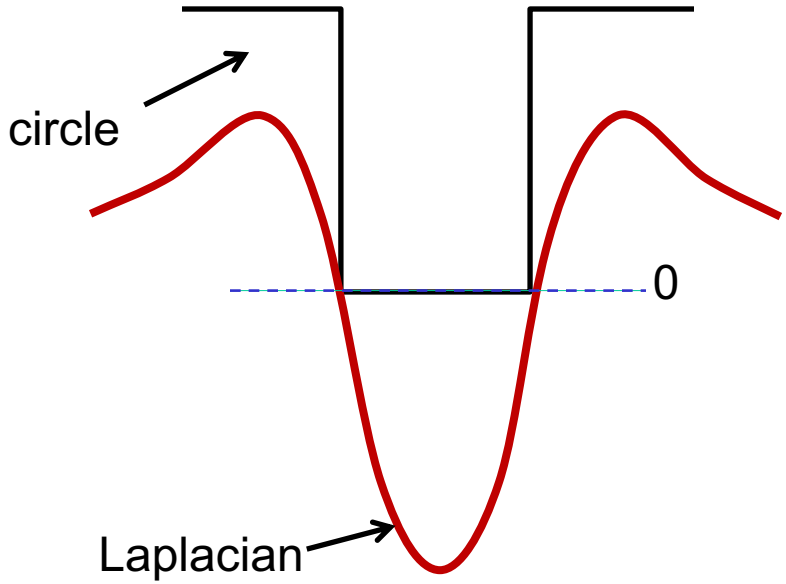
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$

- Therefore, the maximum response occurs at $\sigma = r / \sqrt{2}$.

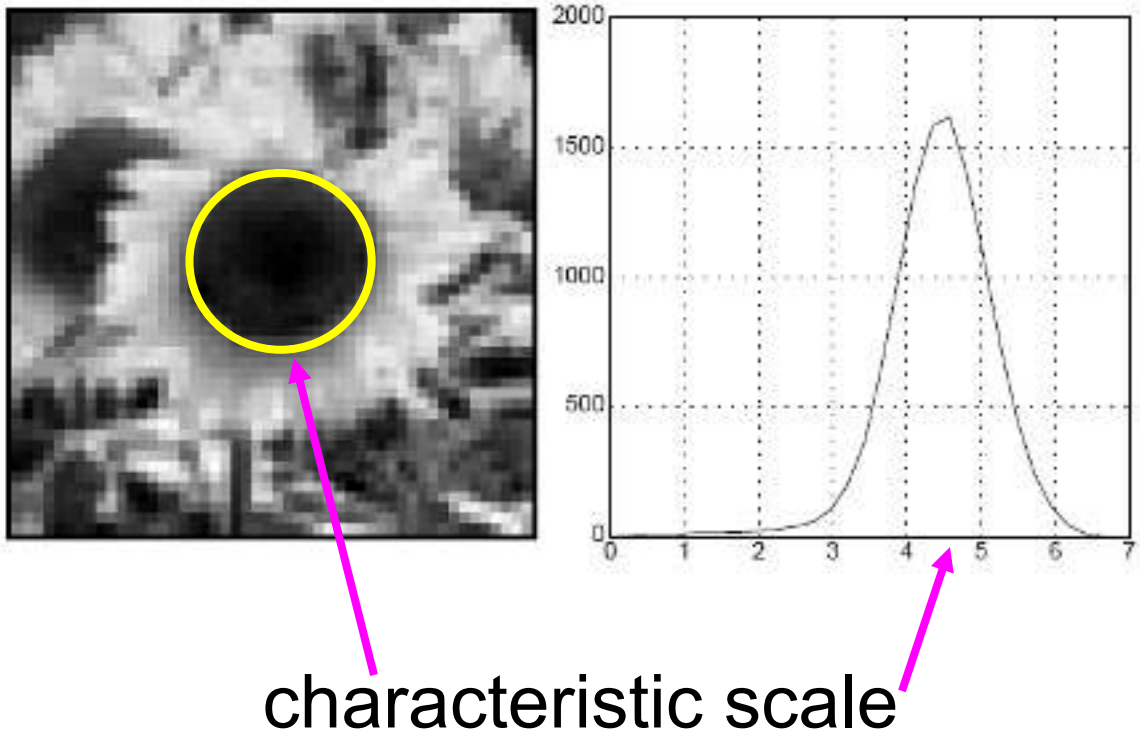


image



Characteristic scale

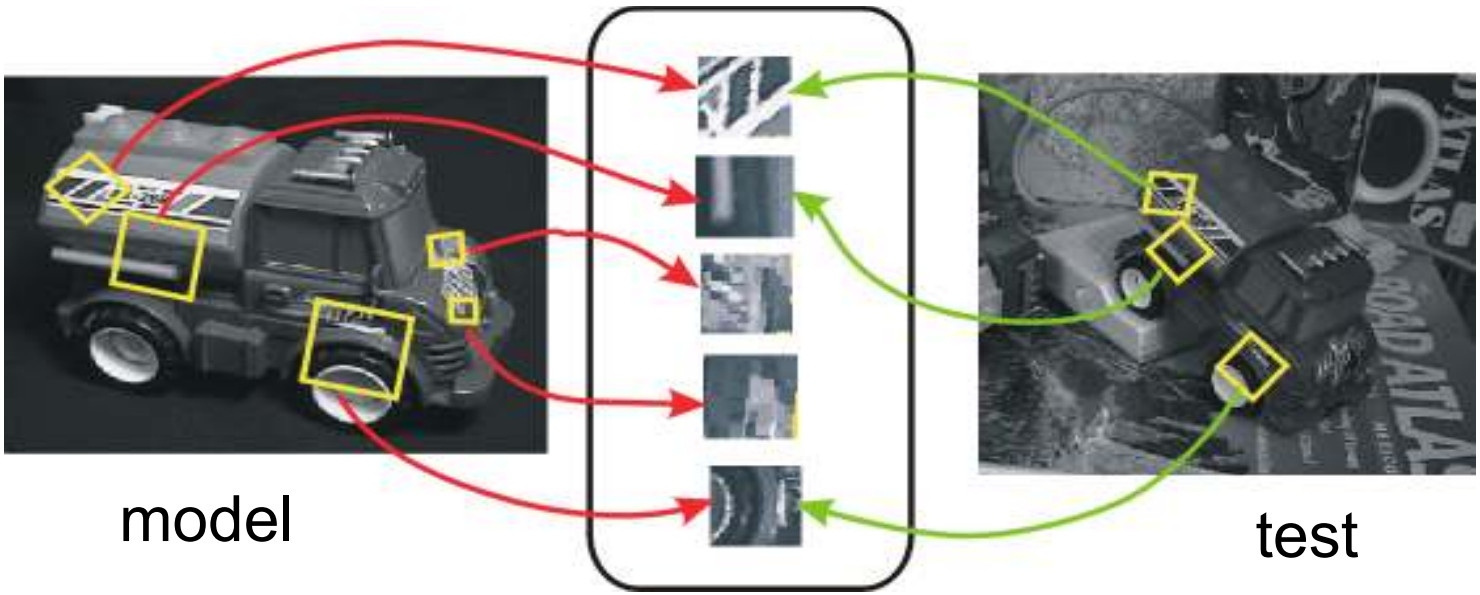
- We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center



T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

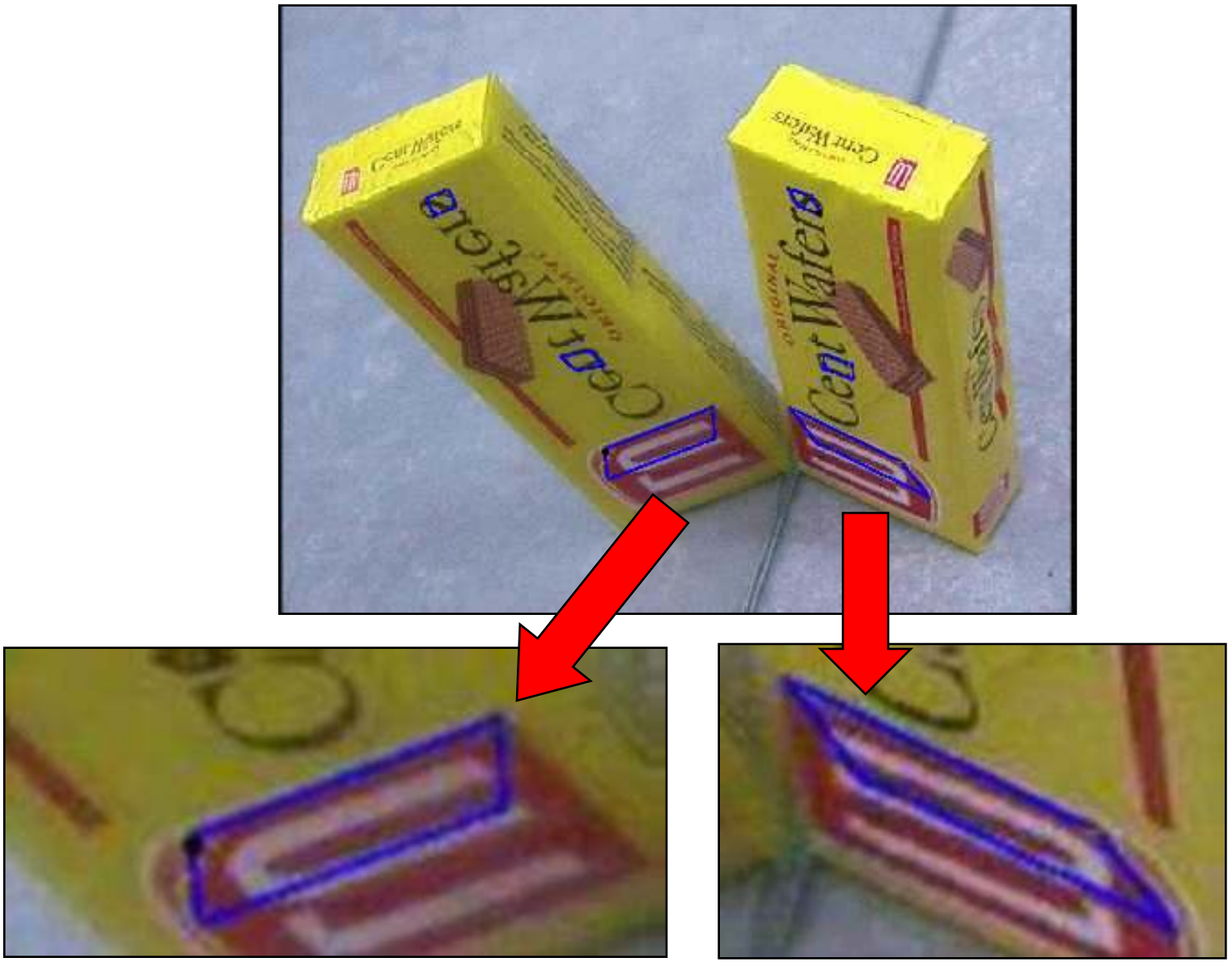
Basic algorithm

- **Detect** and describe features in model image(s)
  (done: scale invariant features; next: **affine invariant ones**)
- Detect and describe features for test image
- Match features (including geometric verification, e.g. RANSAC)
- Count matches
  → many? → object recognized and localized
  → few? → object not present in test image



Images: D. Lowe

Increasing the level of invariance to Affine

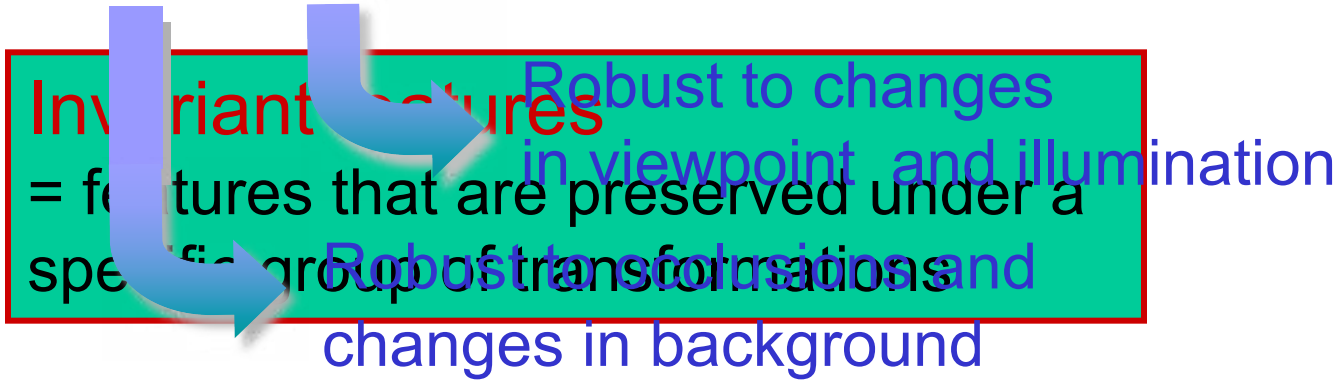


Recognition using local affine and photometric invariant features

Hybrid approach that aims to deal with **large variations** in

- Viewpoint
- Illumination
- Background
- and Occlusions

$\Rightarrow$ Use local invariant features



Recognition using local affine and photometric invariant features

Aims to deal with **large variations** in

Viewpoint

Recognition using local affine and photometric invariant features

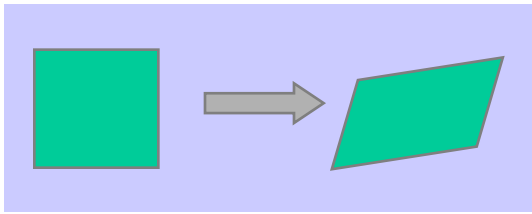
Aims to deal with **large variations** in

- Viewpoint
- Illumination

Transformations for planar objects

Affine geometric deformations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$



Linear photometric changes

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} s_R & 0 & 0 \\ 0 & s_G & 0 \\ 0 & 0 & s_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} o_R \\ o_G \\ o_B \end{bmatrix}$$

Works for *locally* roughly planar objects and roughly orthographic cameras (or far away camera) → quite general in practice

Local invariant features

‘Affine invariant region’

Local invariant features

regions we look for cover the same physical part of the scene

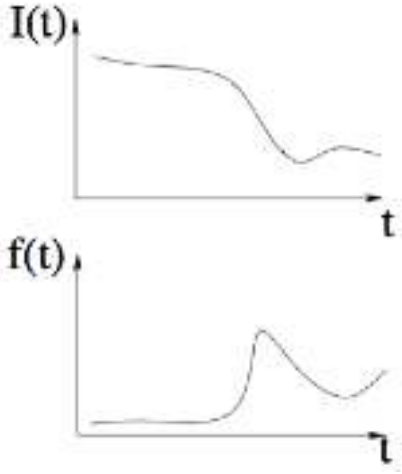
Local invariant features

Intensity-based region extraction of

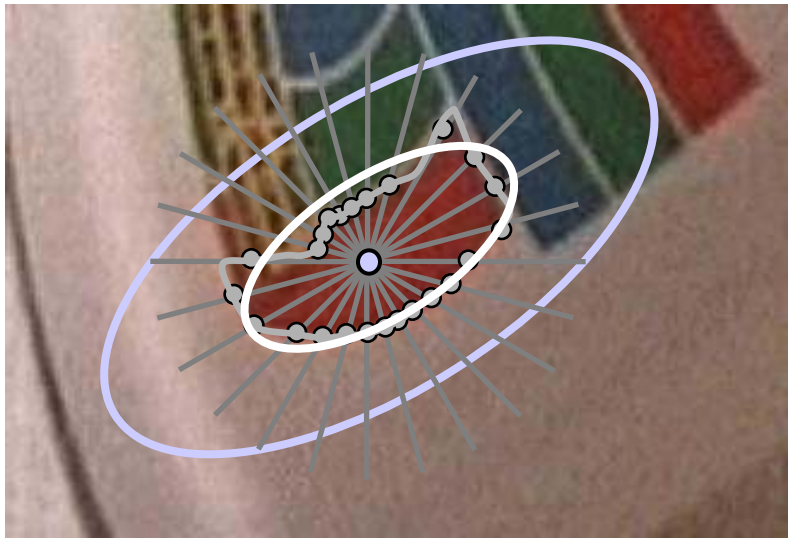
Tuytelaars, Van Gool. “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions”. BMVC 2000.

Intensity based method

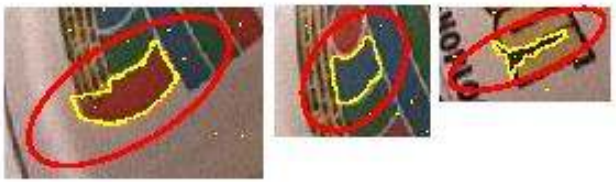
1. Search intensity extrema
2. Observe intensity profile along rays
3. Search maximum of invariant function f(t) along each ray
4. Connect local maxima
5. Fit ellipse (preserves affine covariance)
6. Double ellipse size (better descriptors)

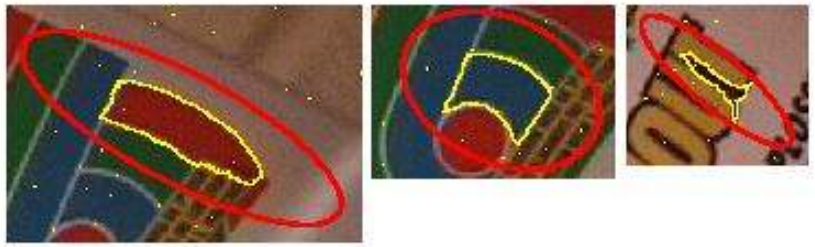


$$f(t) = \frac{abs(I_0 - I)}{\max\left(\frac{\int abs(I_0 - I)dt}{t}, d\right)}$$



Intensity based method





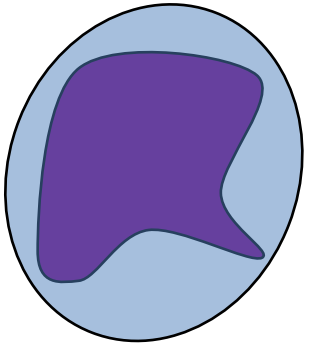
How do we fit the ellipse?

- Geometric Moments:

$$m_{pq} = \int_{\sim 2} x^p y^q f(x,y) dxdy$$

Fact: moments m_{pq} uniquely determine the function f

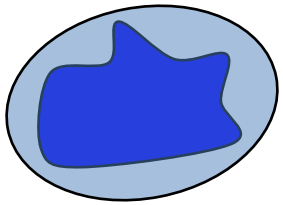
Taking f to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse



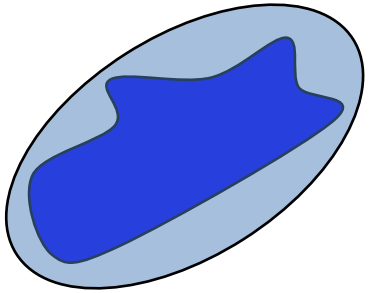
This ellipse will have the same moments of orders up to 2 as the original region

Affine Covariant Detection

- Second order moments → Covariance matrix of region points defines an ellipse:



$$q = Ap$$



$$p^T \Sigma_1^{-1} p = 1$$

$$q^T \Sigma_2^{-1} q = 1$$

$$\Sigma_1 = \left\langle pp^T \right\rangle_{\text{region 1}}$$

$$\Sigma_2 = \left\langle qq^T \right\rangle_{\text{region 2}}$$

($p = [x, y]^T$ is relative to the center of mass)

$$\Sigma_2 = A \Sigma_1 A^T$$

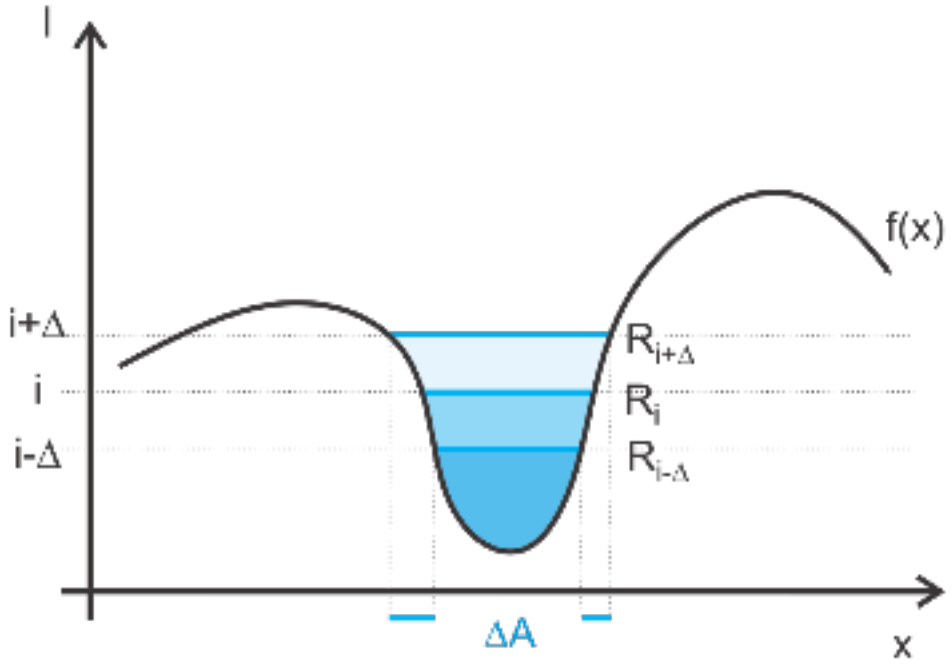
Ellipses computed for corresponding regions also correspond!

The Maximally Stable Extremal Regions

- Region - a contiguous subset of image – connected component
- Extremal Region R_i – all pixels are of equal or lower/higher intensity than i
- Maximally Stable Extremal Region – a region R_i for which the relative change of area

$$q(i) = |R_{i+\Delta} \setminus R_{i-\Delta}| / |R_i|$$

is local minimum for a range of intensities i.



The Maximally Stable Extremal Regions

- Consecutive image thresholding by all thresholds
- Maintain list of Connected Components
- Regions = Connected Components
  with stable area over multiple thresholds selected
- Final step: fit an ellipse to the region

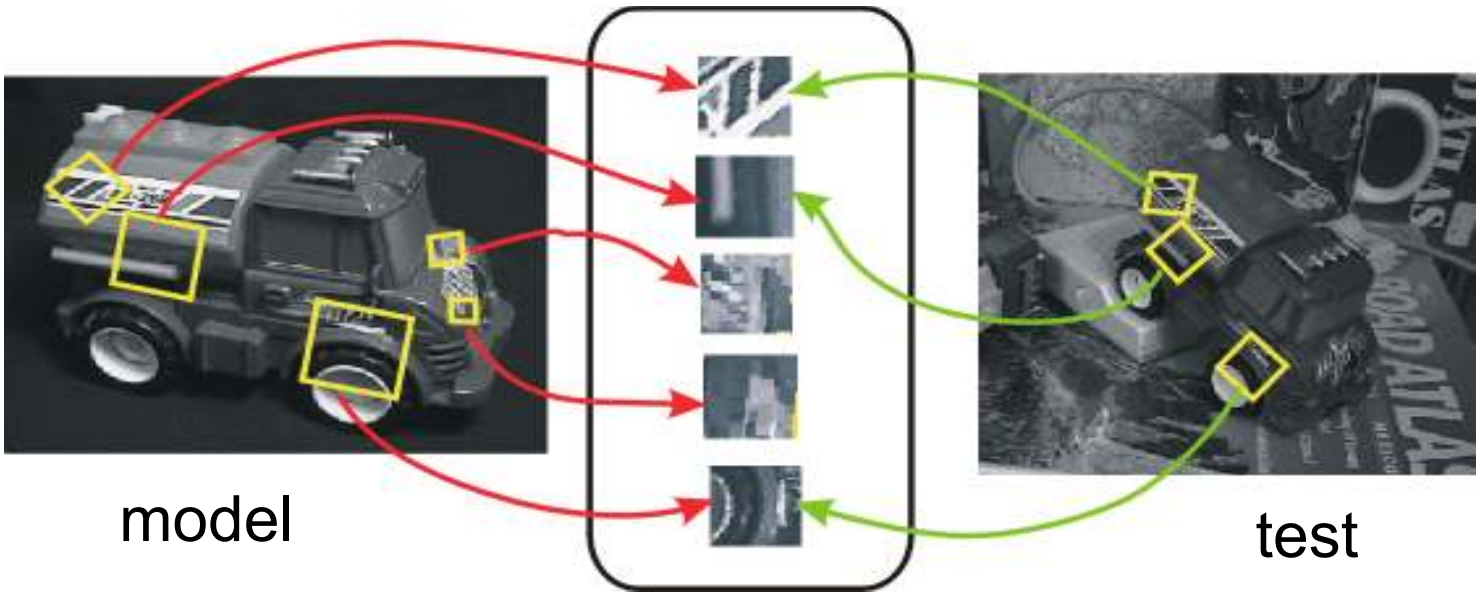
The Maximally Stable Extremal Regions

- Covariant with continuous deformations of images
- Invariant to affine transformation of pixel intensities
- Enumerated in O(n log log n), real-time computation

Matas, Chum, Urban, Pajdla: “Robust wide baseline stereo from maximally stable extremal regions”. BMVC 2002

Basic algorithm

- Detect and **describe features** in model image(s)
  (done: scale invariant features; next: affine invariant ones)
- Detect and **describe features** for test image
- Match features (including geometric verification, e.g. RANSAC)
- Count matches
  → many? → object recognized and localized
  → few? → object not present in test image



Images: D. Lowe

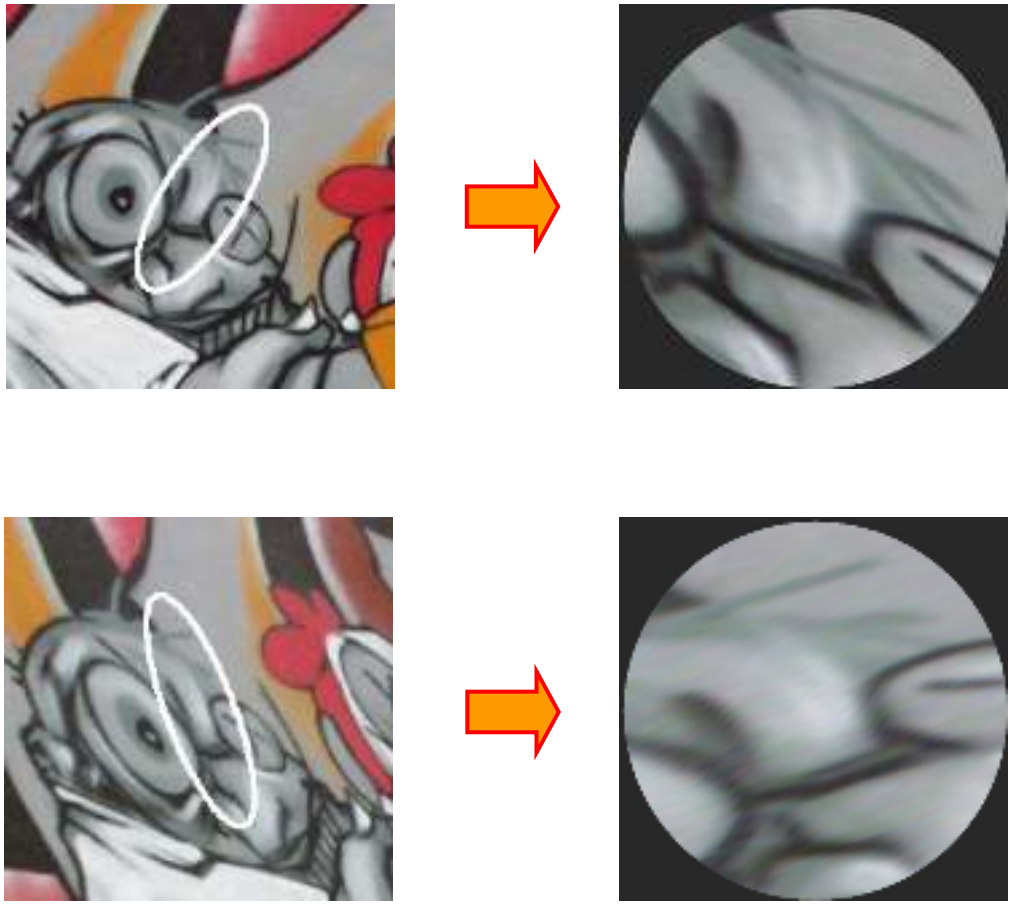
From covariant detection to invariant description

- Geometrically transformed versions of the same neighborhood will give rise to regions that are related by the same transformation
- What to do if we want to compare the appearance of these image regions?
  - *Normalization*: transform these regions into same-size circles

Images: K. Mikolajczyk

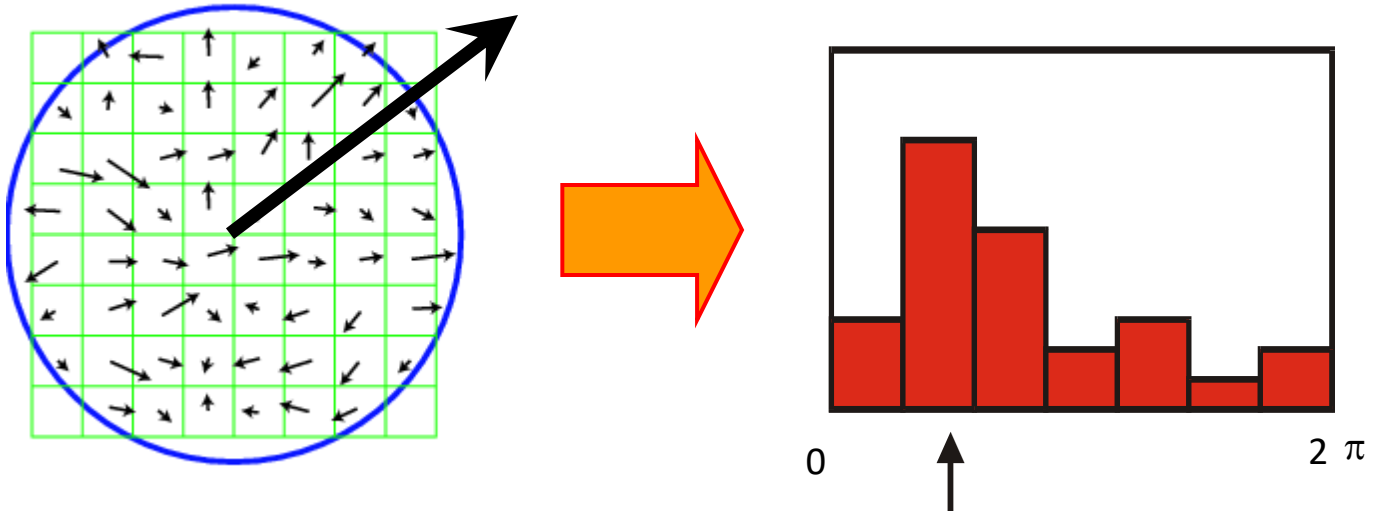
Affine normalization

- Problem: There is no unique transformation from an ellipse to a unit circle
  - We can rotate or flip a unit circle, and it still stay a unit circle



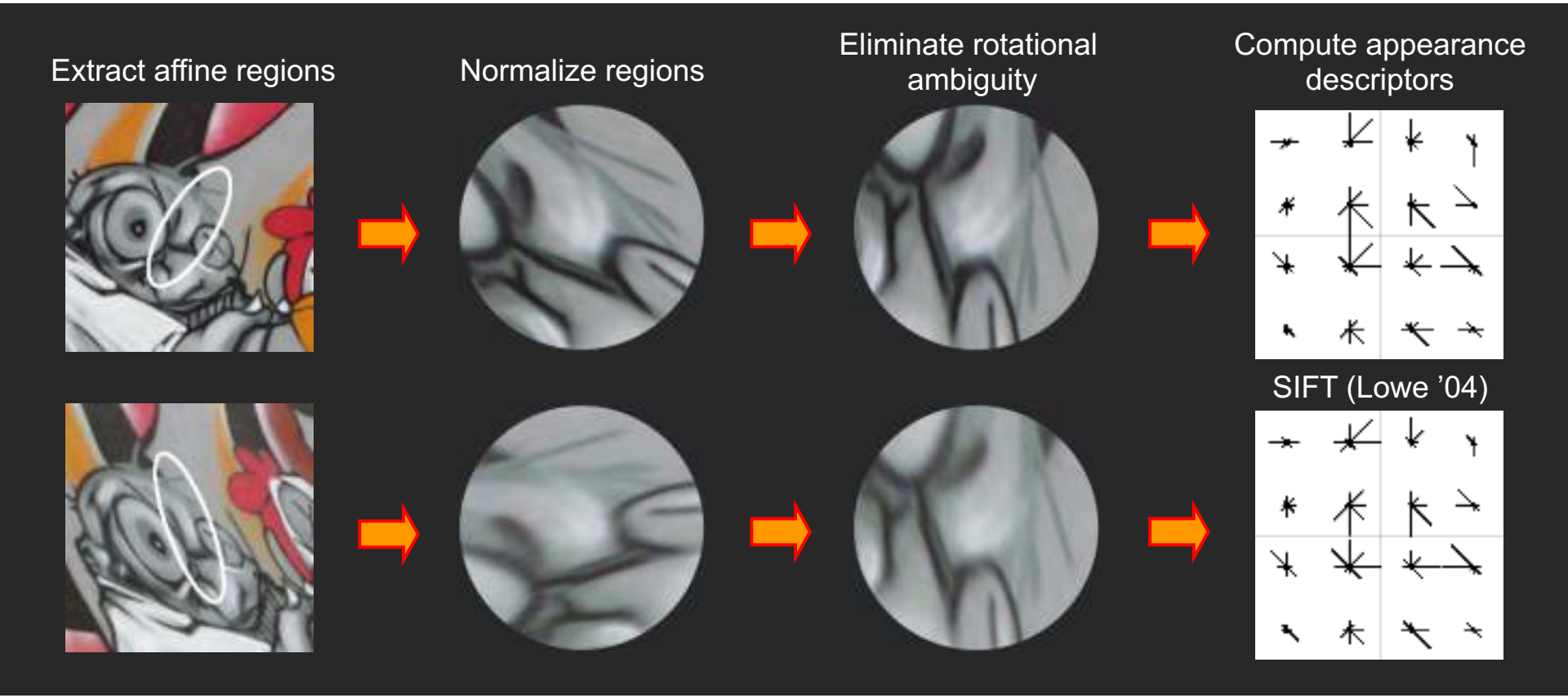
Eliminating rotation ambiguity

- To assign a unique orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram



slide credit: Lazebnik

From covariant regions to invariant features



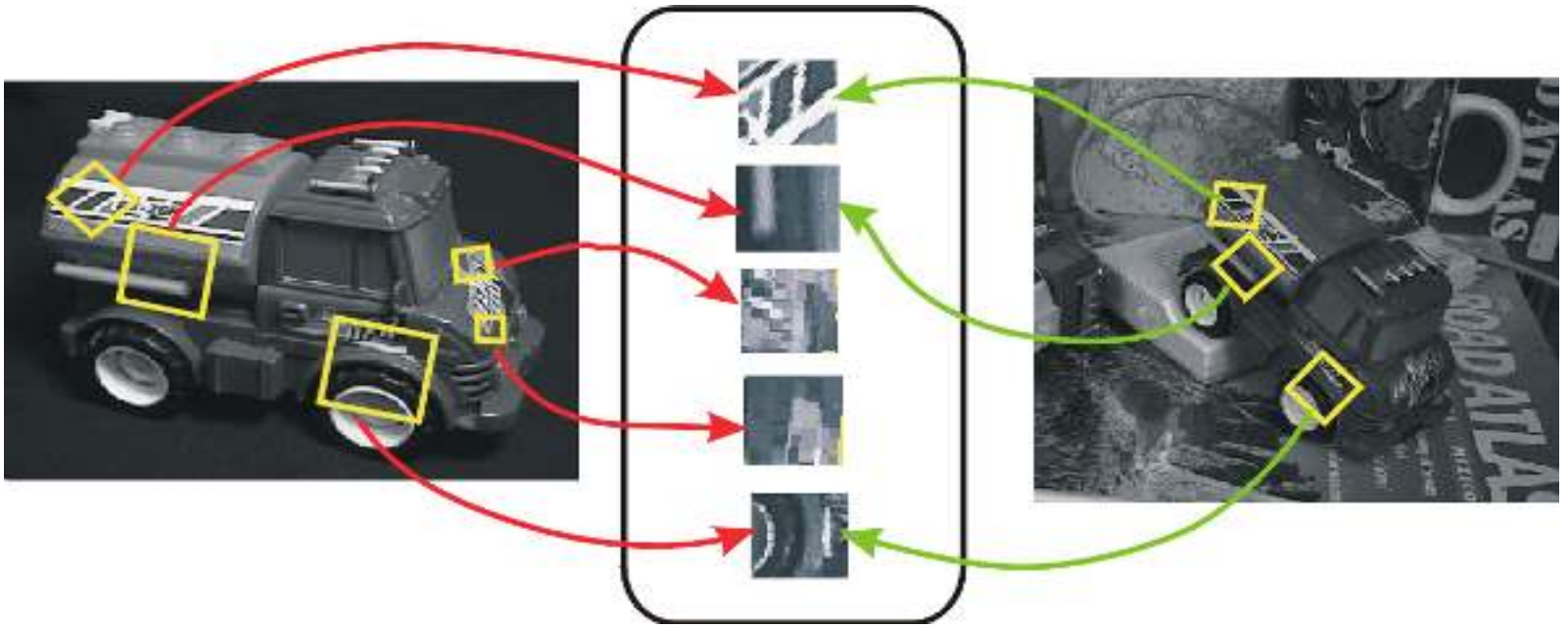
Invariance vs. covariance

Invariance:

- features(transform(image)) = features(image)

Covariance:

- features(transform(image)) = transform(features(image))



Covariant detection => invariant description

Images: D. Lowe

Feature descriptors

- Simplest descriptor: vector of raw intensity values
- How to compare two such vectors?
  - Sum of squared differences (SSD)

$$\mathrm{SSD}(u,v) = \sum_i \left(u_i - v_i\right)^2$$

    - Not invariant to intensity change

  - Normalized correlation

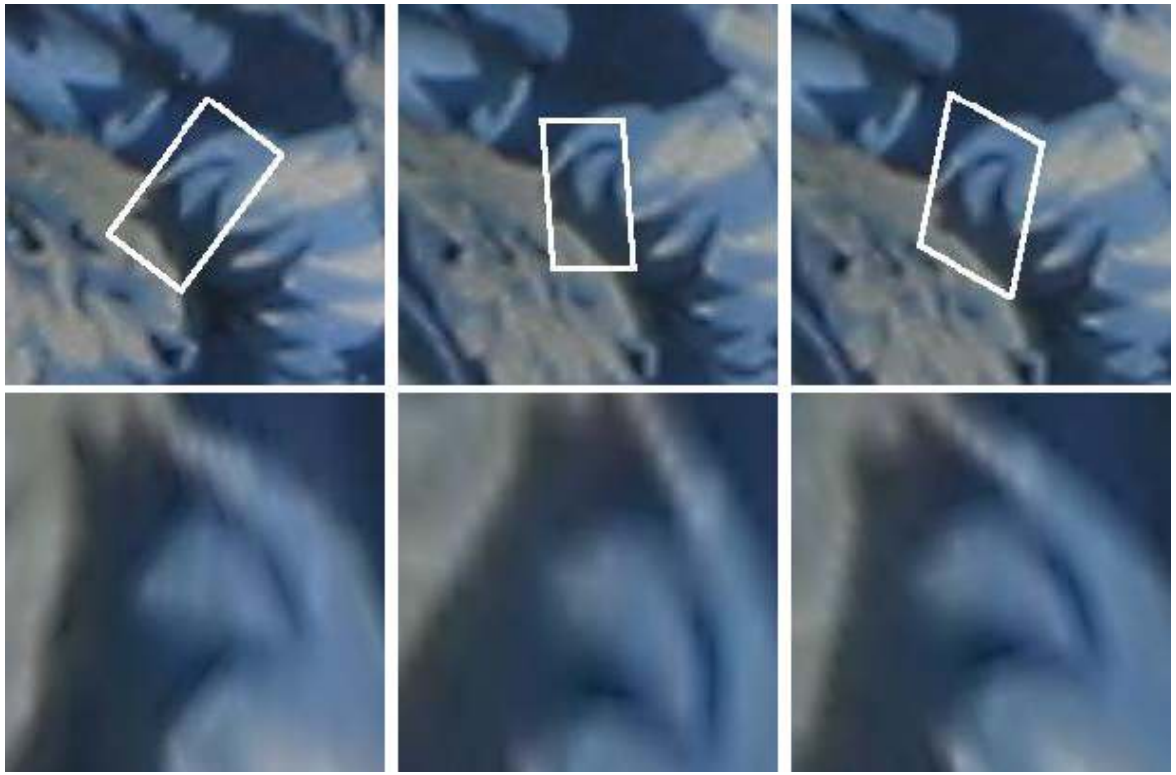
$$\rho(u,v) = \frac{\sum_i (u_i - \bar{u})(v_i - \bar{v})}{\sqrt{\left(\sum_j (u_j - \bar{u})^2\right)\left(\sum_j (v_j - \bar{v})^2\right)}}$$

    - Invariant to affine intensity change

slide credit: Lazebnik

Disadvantage of intensity vectors as descriptors

- Small misalignments can affect the matching score a lot



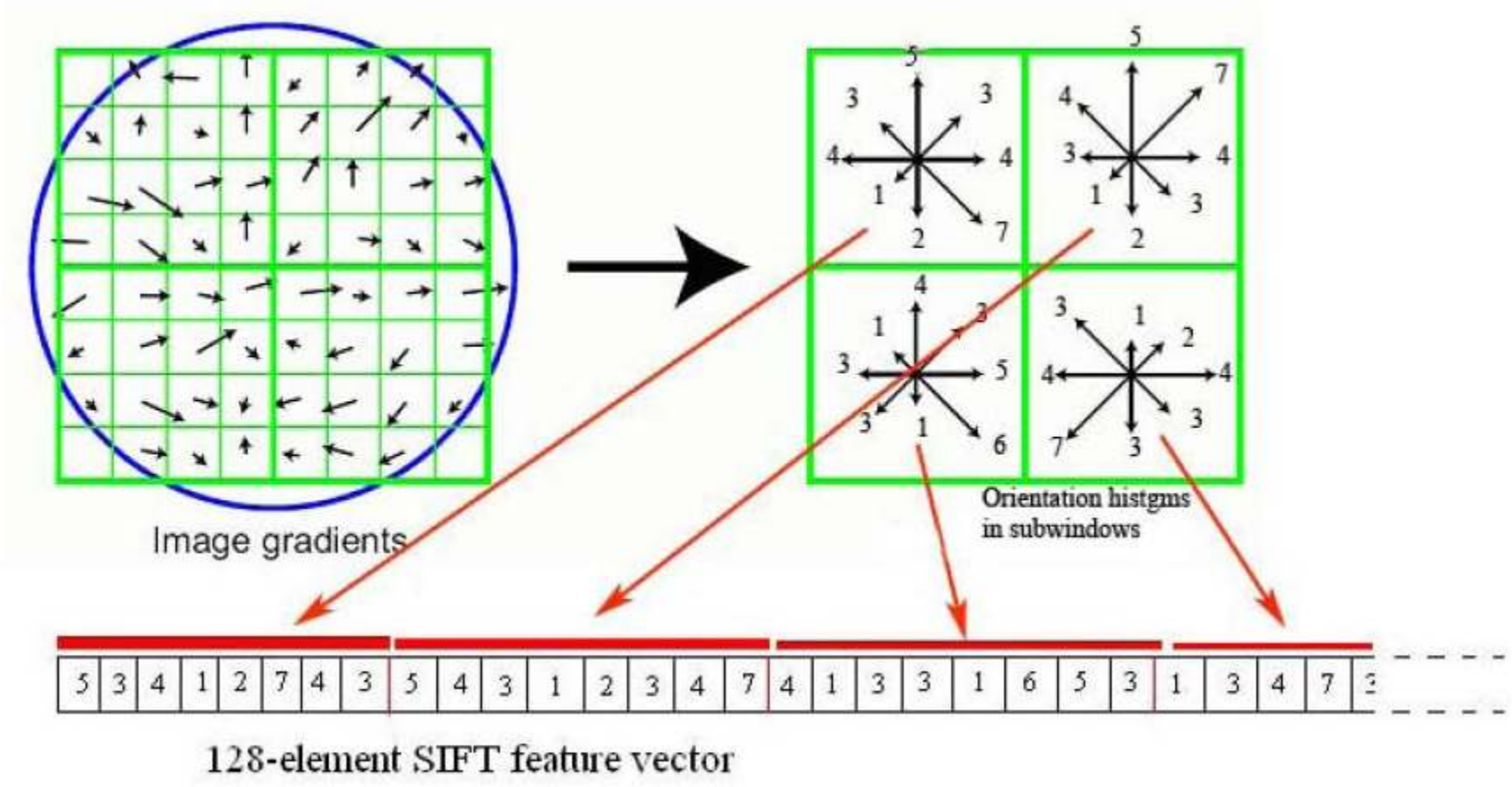
Images: F. Rothganger

SIFT Descriptor

A 4x4 histogram lattice of orientation histograms

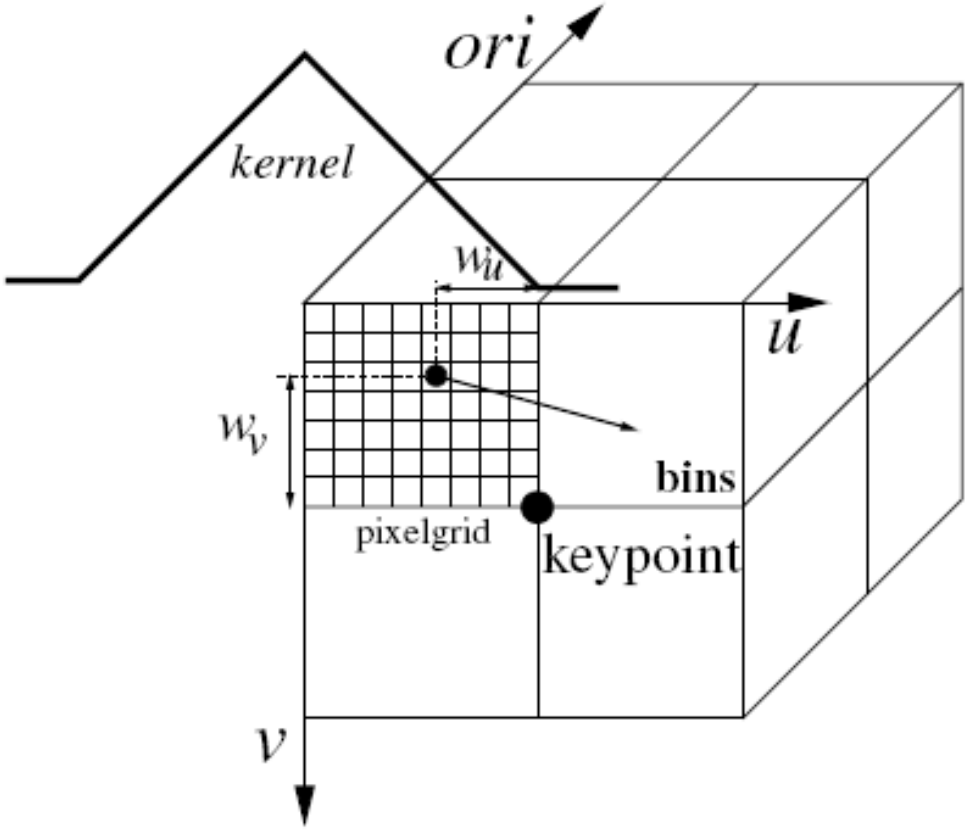
Orientations quantized (with interpolation) into 8 bins

Each bin contains a weighted sum of the norms of the image gradients around its center, with complex normalization



SIFT Descriptor

It can be viewed as a 3–D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction (normally discretised into 8 bins)



SIFT advantages

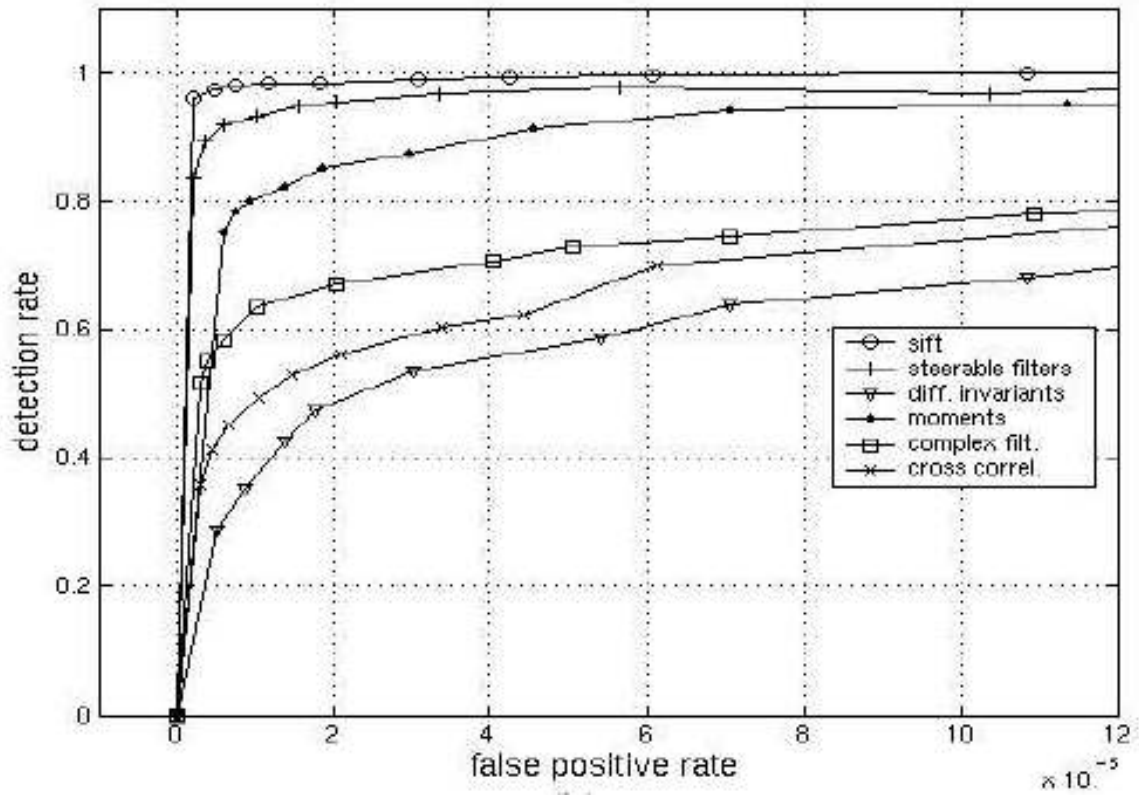
- Based on gradient orientations, which are robust to illumination changes
- Spatial binning gives tolerance to small shifts in location and scale, affine change.
- Explicit orientation normalization
- Photometric normalization by making all vectors unit norm
- Orientation histogram gives robustness to small local deformations

SIFT performance

Empirically found to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5

Rotation = 45^0

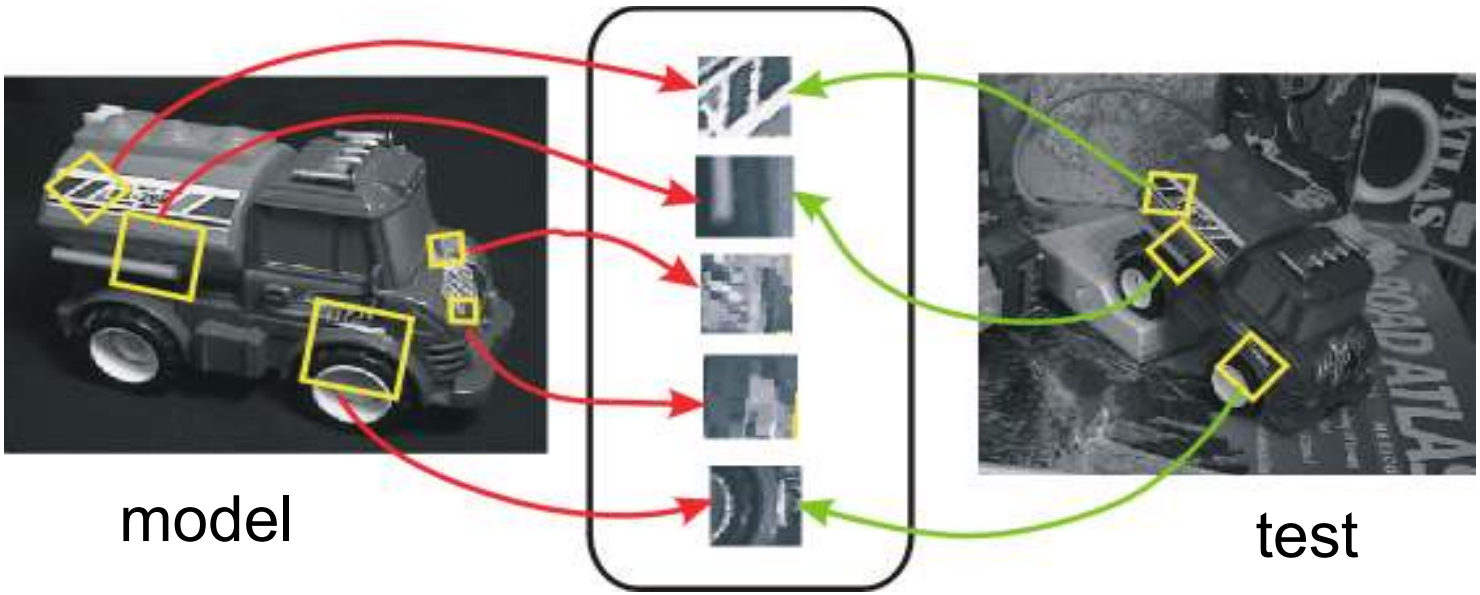


D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. IJCV 2004

K.Mikolajczyk, C.Schmid. “A Performance Evaluation of Local Descriptors”. CVPR 2003

Basic algorithm

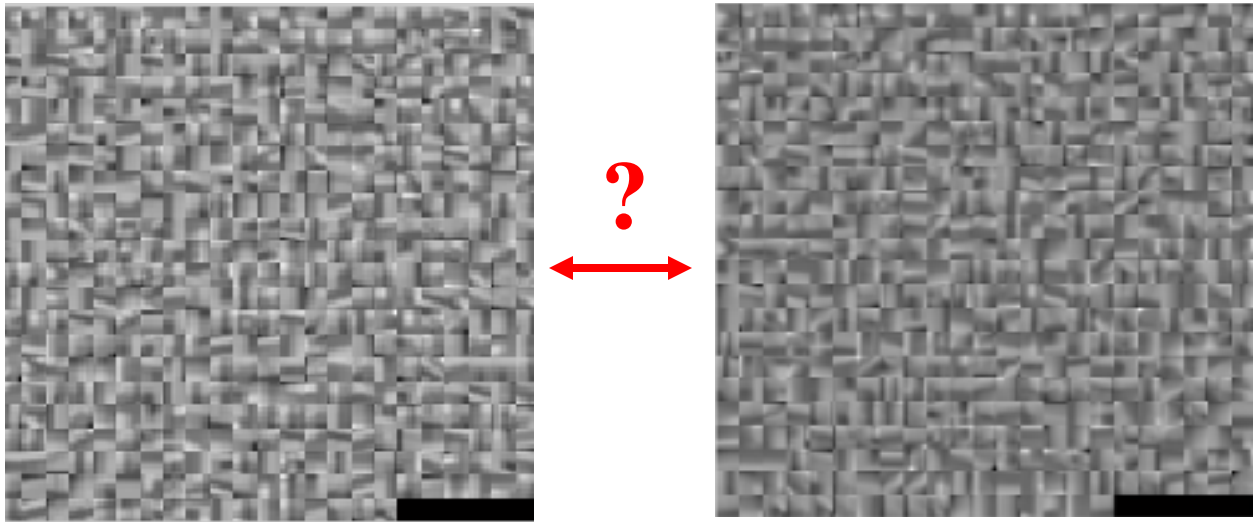
- Detect and describe features in model image(s)
  (done: scale invariant features; next: affine invariant ones)
- Detect and describe features for test image
- **Match features** (including geometric verification, e.g. RANSAC)
- Count matches
  → many? → object recognized and localized
  → few? → object not present in test image



Images: D. Lowe

Feature matching

- Generating *putative matches*: for each patch in one image, find a short list of patches in the other image that could match it based solely on appearance

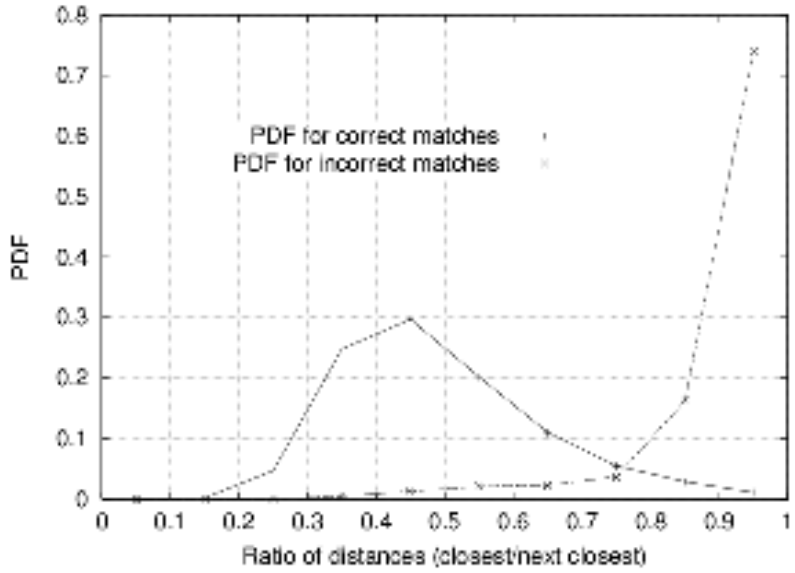


- Simple algorithm:
  1. Compare all pairs of descriptors
  2. Keep all pairs with distance below a threshold

Slide edited from Lazebnik

Feature space mismatch rejection

- How can we tell which putative matches are more reliable?
- Heuristic: compare distance of **nearest** neighbor to that of **second** nearest neighbor
  - Ratio of closest distance to second-closest distance will be *high* for features that are *not* distinctive

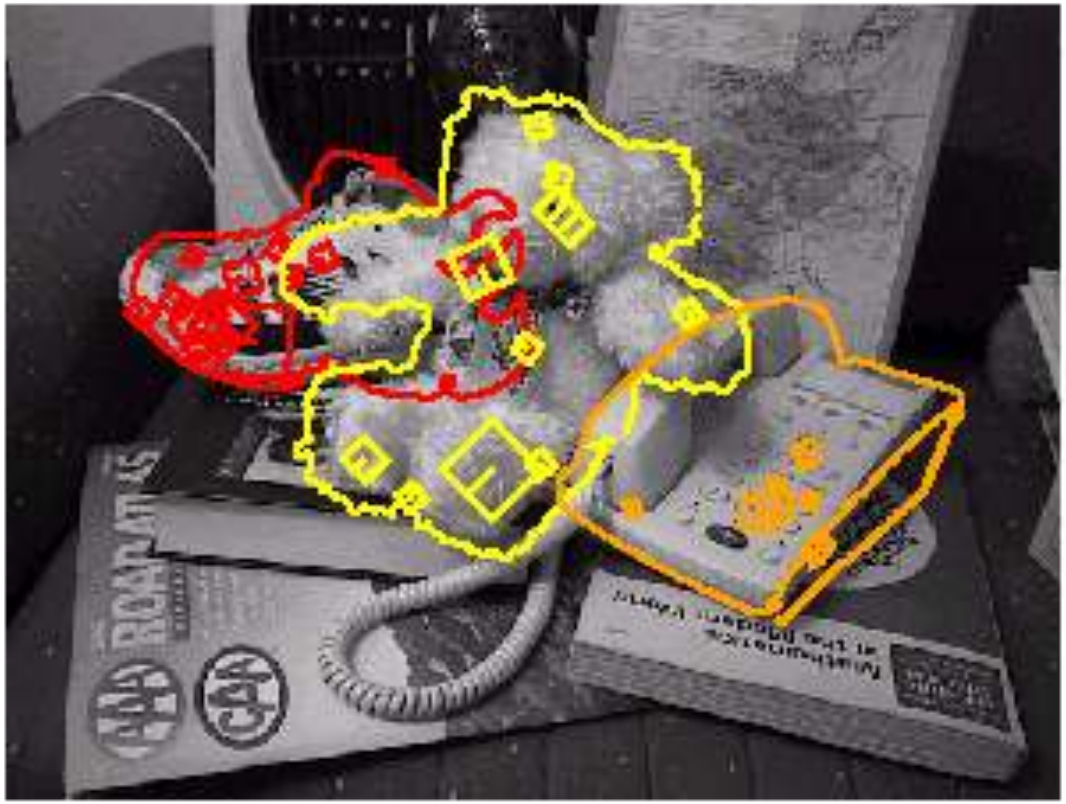


Threshold of 0.8 provides good separation

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

Slide: Lazebnik

Reading



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

Comparison between model-based and appearance-based techniques

Pure model-based	Pure appearance-based
Compact model	Large models
Can deal with clutter	Cannot deal with clutter
Slow analysis-by-synthesis	Efficient
Models difficult to produce	Models easy to produce
For limited object classes	For wide classes of objects

Hybrid tecniques

+ Rather compact model
+ Can deal with clutter and partial occlusion
+ Efficient
+ Models easy to produce (take images, and fewer than in pure appearance based)
+ For rather wide class of objects (almost as wide as in pure appearance based)

Extensions

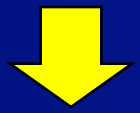
1. Challenging imaging conditions, where feature matching produces too many mismatches
   → correspondence expansion techniques
   (Ferrari et al ECCV 2004)

2. Scaling to large-scale databases containing thousands of model objects
   → indexing techniques
   (Nister CVPR 2006)

Extension 1:
Dealing with highly challenging imaging conditions

Recap: affine invariant regions

automatically adapt shape to viewpoint



allow to find correspondences between two *different* views

How to deal with challenging cases ?

model image

?

?

test image

How to deal with challenging cases ?

model image

?

?

test image

Usual approach with *affine invariant regions*

model image

test image

Extract regions from model image

Extract regions from test image

Compute invariant descriptors

Compute invariant descriptors

Matching (one-to-one)

failure: 0/5

[Tuytelaars, BMVC00]

Global filter (e.g. ransac)

(decision)

Why difficult ?

Large scale change

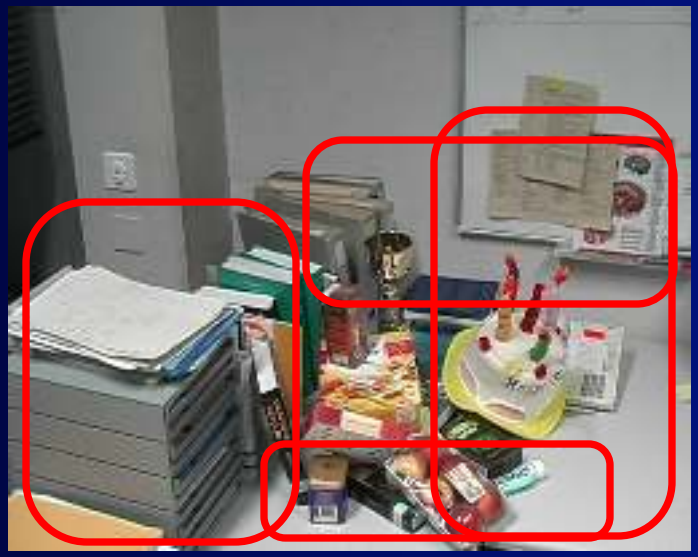
Modest resolution (720x576)

Loss of information

Occlusion

Few *repeated* object regions, less accurate shape

Why difficult ?



Large scale change

Modest resolution (720x576)

Loss of information

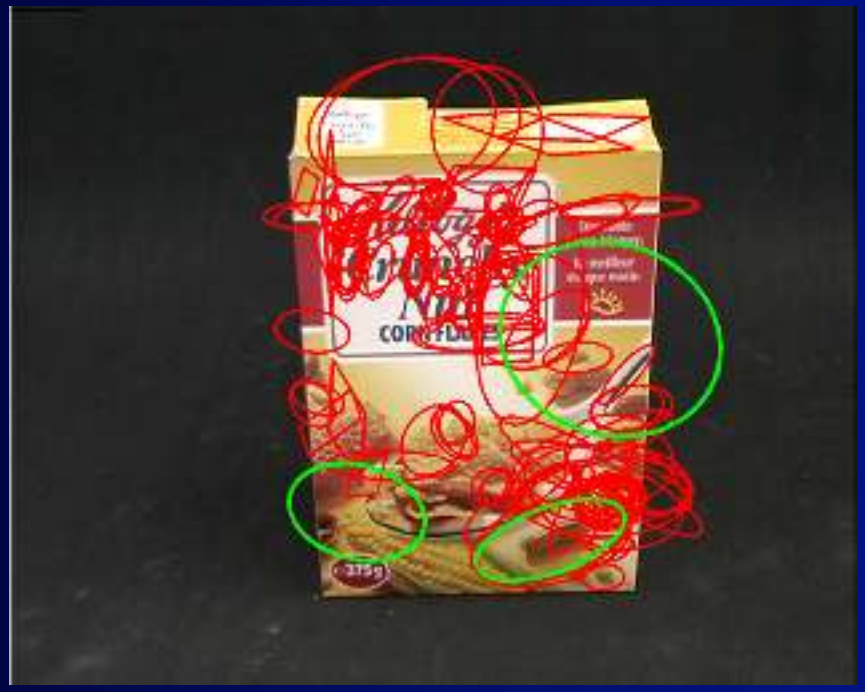
Occlusion

Few *repeated* object regions, less accurate shape

Extensive clutter

Few correct matches (and many mismatches)

New approach: image exploration



1. Soft-matching
   (one-to-many:
   max number of correct matches)

   ⟶ 3/217

New approach: image exploration

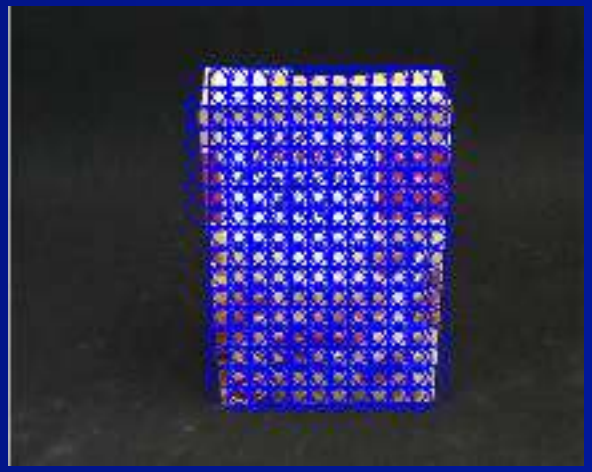
correct (zoom 2x)

wrong

1. Soft-matching
   (one-to-many:
   max number of correct matches)

   ⟶ 3/217

2. Model coverage



New approach: image exploration

correct (zoom 2x)

wrong

1. Soft-matching
   (one-to-many:
   max number of correct matches)

   ⟶ 3/217

2. Model coverage

3. Expansion
   (*construct* new matches)

4. Contraction
   (remove matches)

⟶ **202/220**

New approach: image exploration

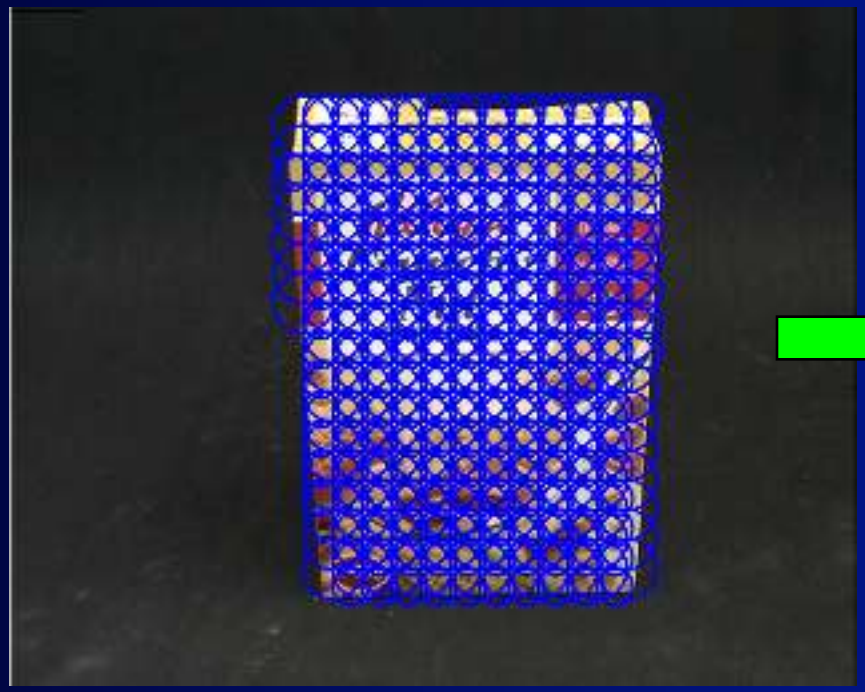
Expansion and Contraction phases help each other !

Effect: gradually ***explore*** image around initial matches

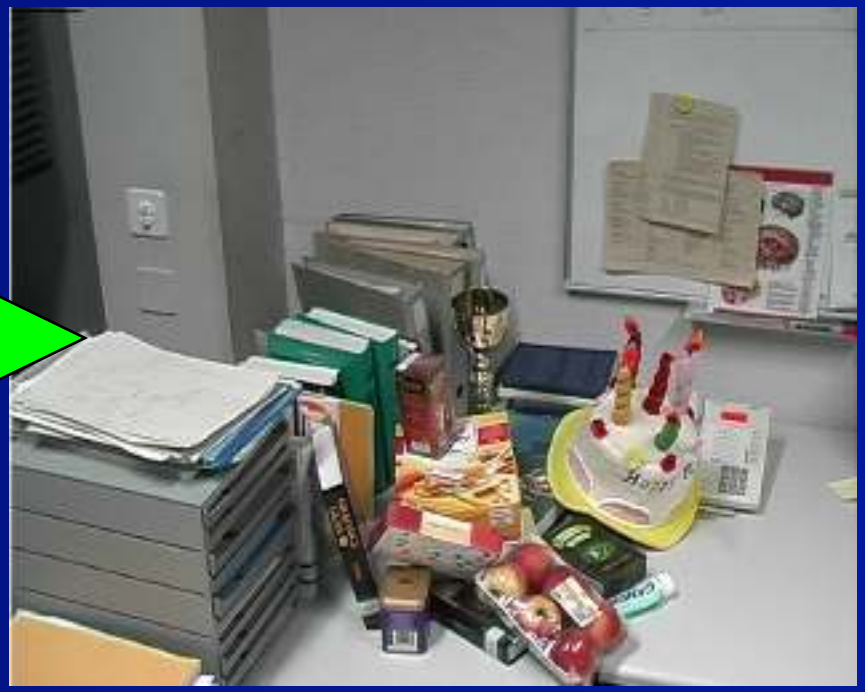
\+ *Power*: very few initial correct matches suffice

\+ *Functionality*: cover with matches ⟶ approximate segmentation

Expansion



?



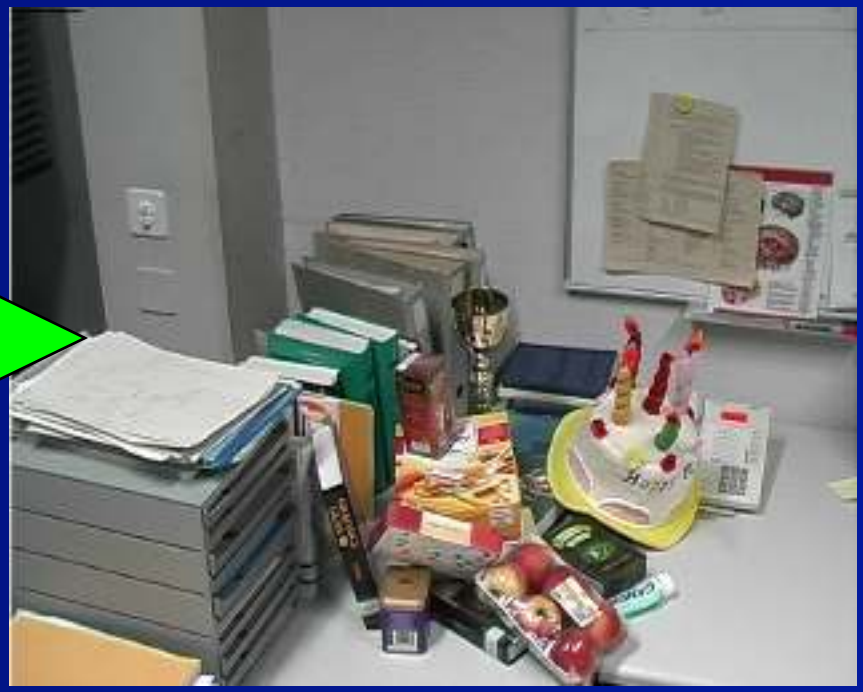
Coverage regions

Goal: construct corresponding regions for the coverage ones

Expansion

unmatched region

?



Expansion

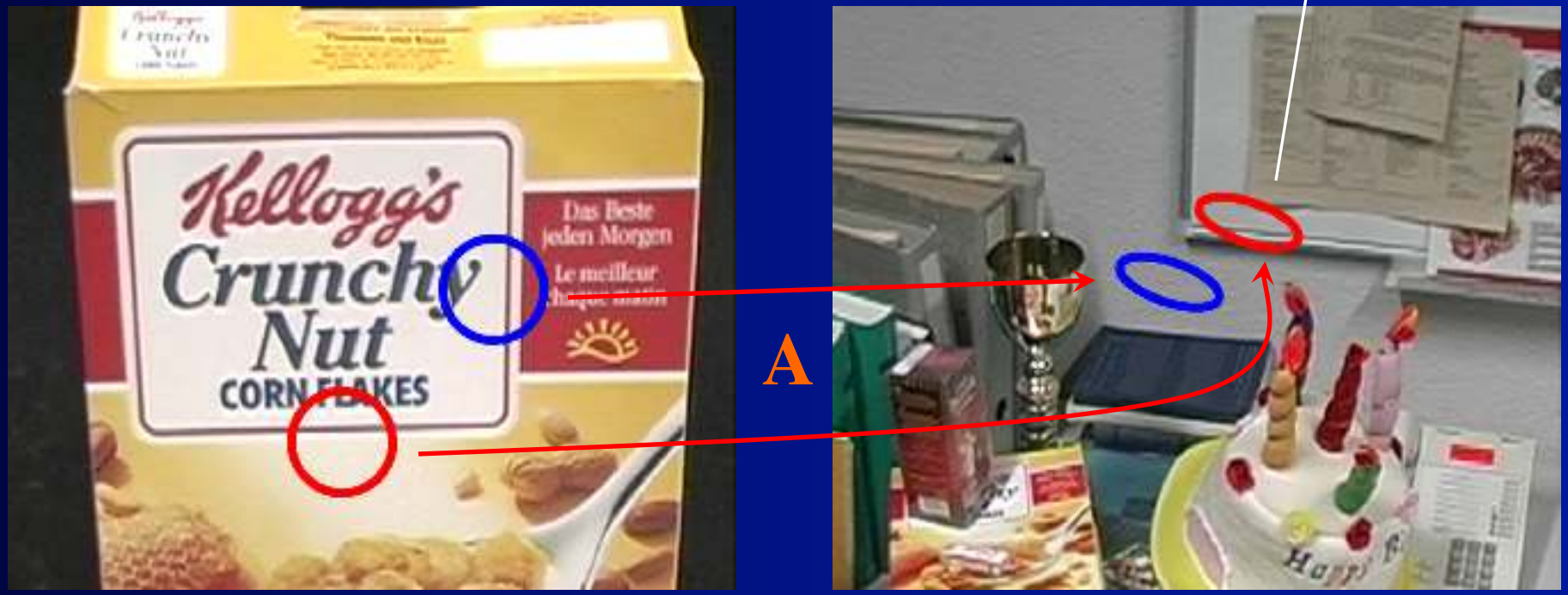
unmatched region

support matches

Idea: project via affine transformation of a support

Expansion

propagation attempt
(low similarity)



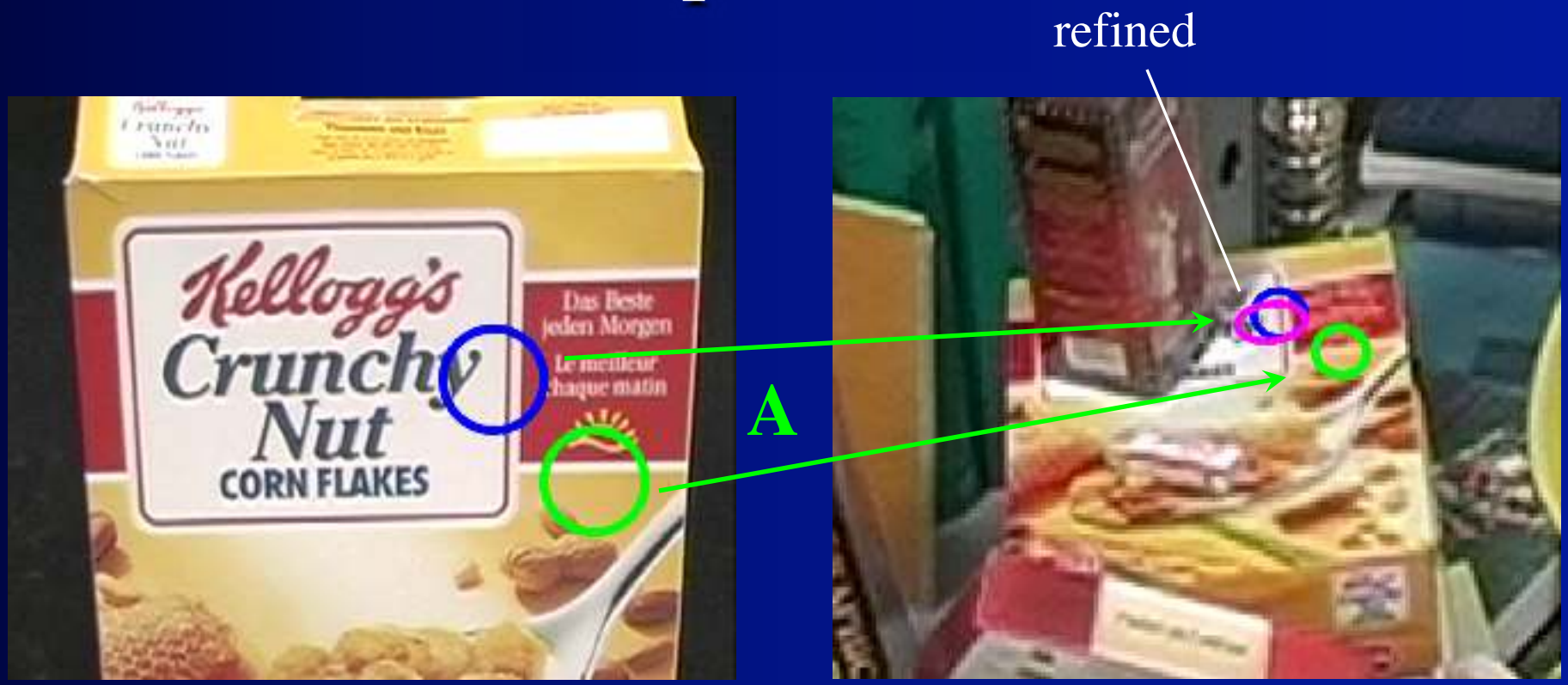
the mismatched support yields a poor propagation attempt

Expansion

the correct support yields a good propagation attempt

⟶ **Winner**

Expansion



the best attempt is refined [Ferrari, CVPR 03]

⟶ adapts to perspective effects, curved surfaces and deformations

Expansion

The higher the percentage of correct matches, the better it works !

Contraction

Sidedness constraint

check it for all triples

⟶ mismatches are involved in more violating triples !

Contraction

Algorithm

1. count violated constraints per match
2. remove match with highest count
3. iterate to 1

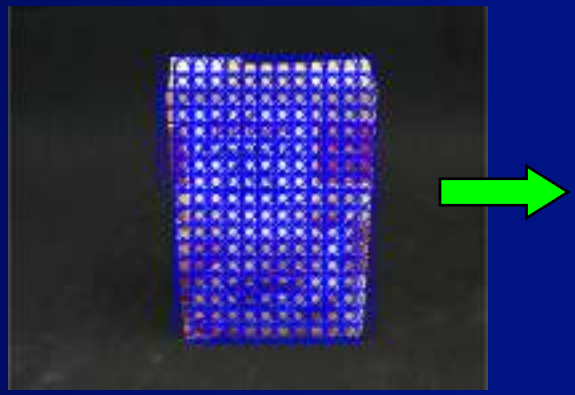
Contraction

+ allows non-rigid deformations

takes better decisions with a higher percentage of correct matches !

Exploring the image

expansion



Exploring the image

expansion

⟶ higher percentage ⟶ helps next contraction

Exploring the image

contraction

⟶ higher percentage ⟶ helps next expansion

Exploring the image

expansion

expansion and contraction *help each other*

⟶ progressive increase of amount / percentage / extent

Exploring the image

contraction

expansion and contraction *help each other*

⟶ progressive increase of amount / percentage / extent

Exploring the image

expansion

expansion and contraction *help each other*

⟶ progressive increase of
- amount
- percentage
- extent

Exploring the image

contraction

expansion and contraction *help each other*

⟶ progressive increase of amount / percentage / extent

Exploring the image

final

expansion and contraction *help each other*

⟶ progressive increase of amount / percentage / extent

Exploring the image

final

+ *Power*: a single correct match per smooth surface suffices

+ *Functionality*: approximate segmentation

Simultaneous Recognition and Segmentation

Recognition

identify correct matches

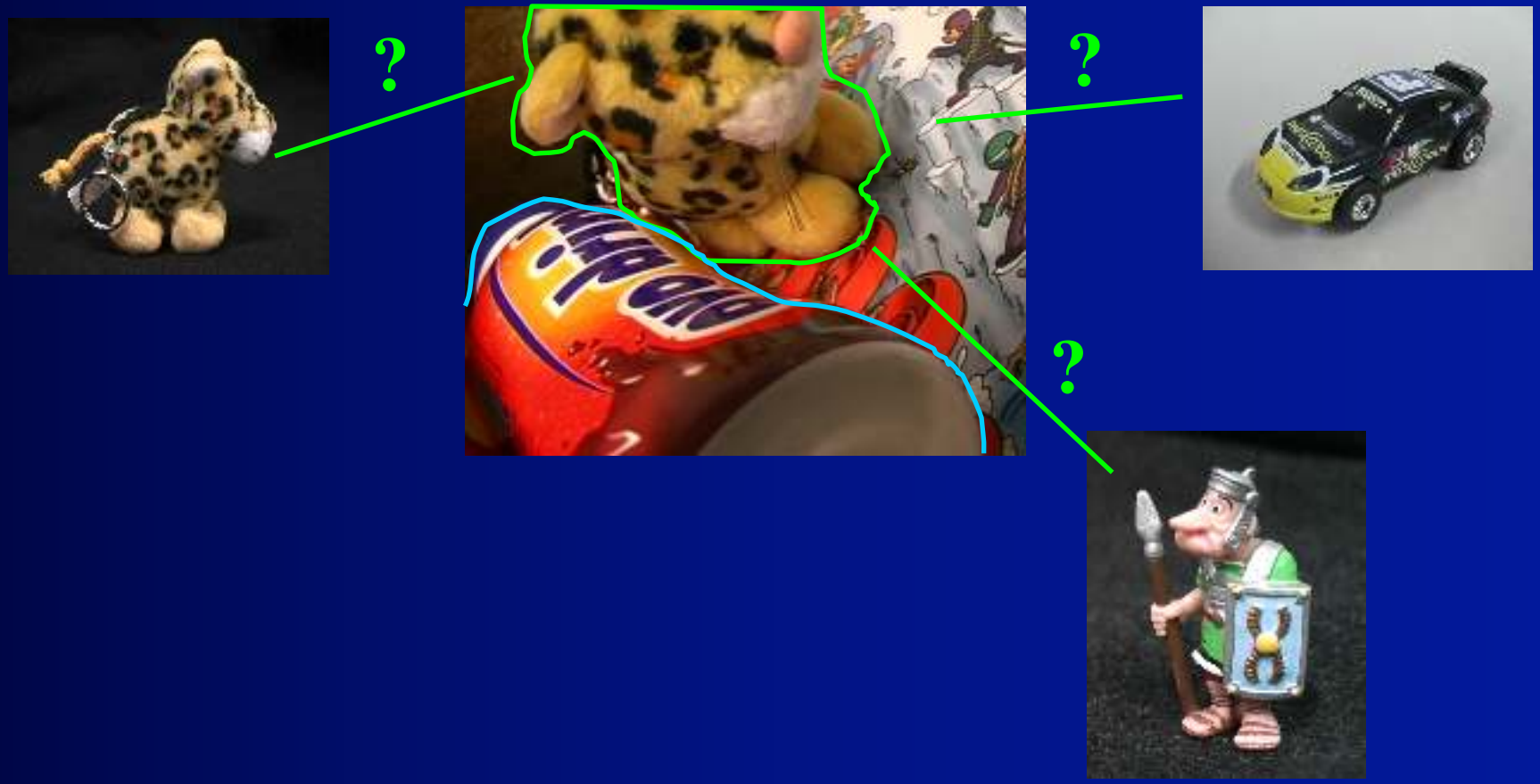
Segmentation

cover object with matches

two aspects of the
same process

What comes first ?

The classic view: segmentation ***before*** recognition



What comes first ?

The modern view: segmentation ***with*** recognition

Results: model objects (planar)

1 model view each

Results: model objects (curved)

6 model views

Results: model objects (curved)

6 model views

Results: model objects (3D)

8 model views

Results: model objects (3D)

8 model views

Results: model objects (3D)

8 model views

Results: model objects (3D)

1 model view

Results

closest model view

occlusion, clutter, scale

Results

Zoom 2

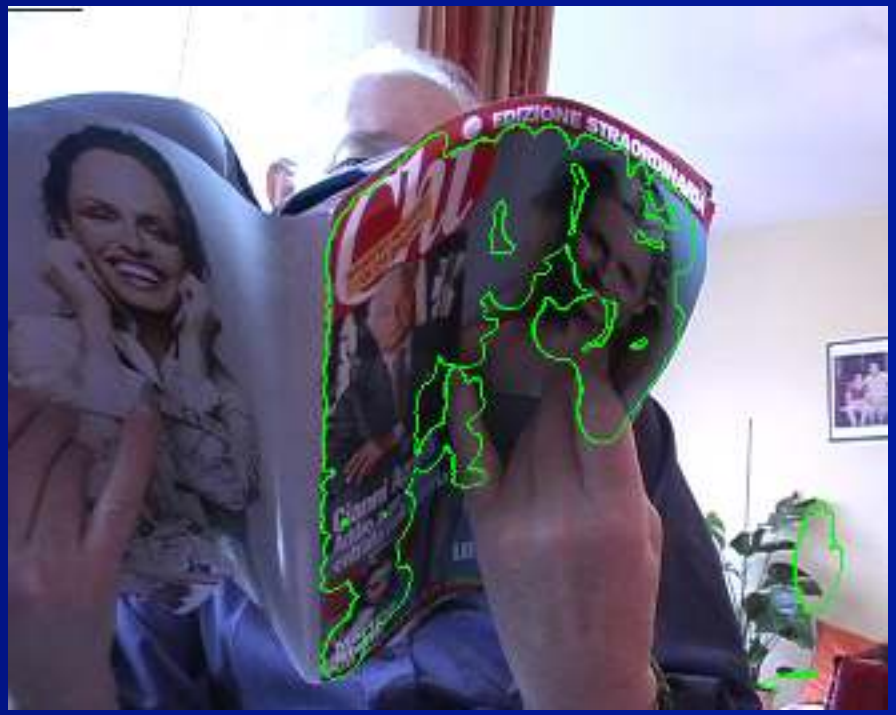
Zoom 2

occlusion, clutter, scale

Results

Folding, extensive clutter, scale, occlusion

Results



Folded, darker, occluded

Results

(whole object)

(total coverage)

Viewpoint change, clutter

Results

closest model view

Viewpoint change, clutter

Results

(whole object)

(total coverage)

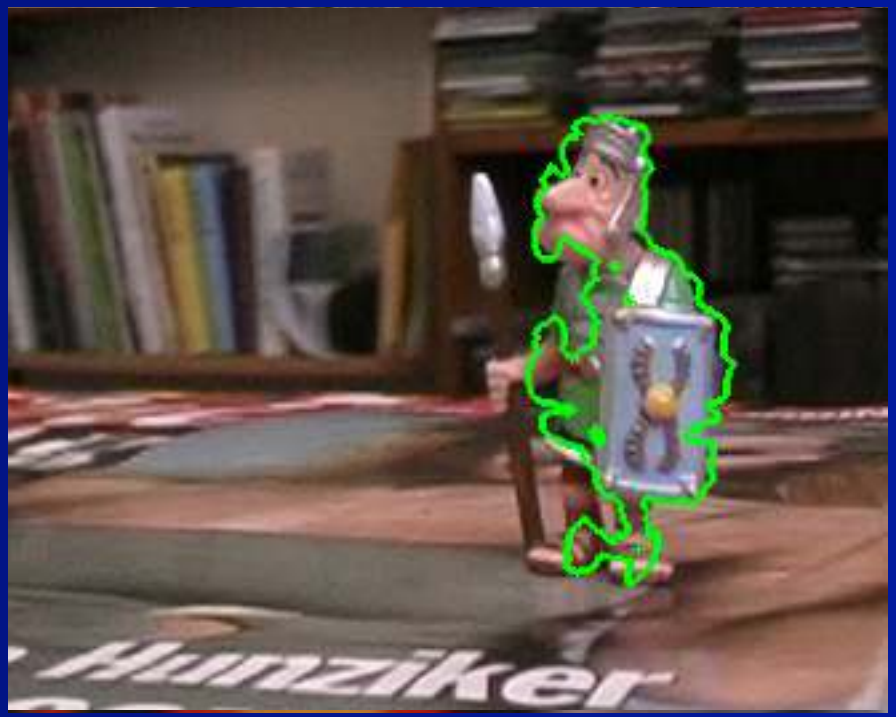
Viewpoint change, clutter

Results

closest model view

scale change, heavy occlusion (89%)

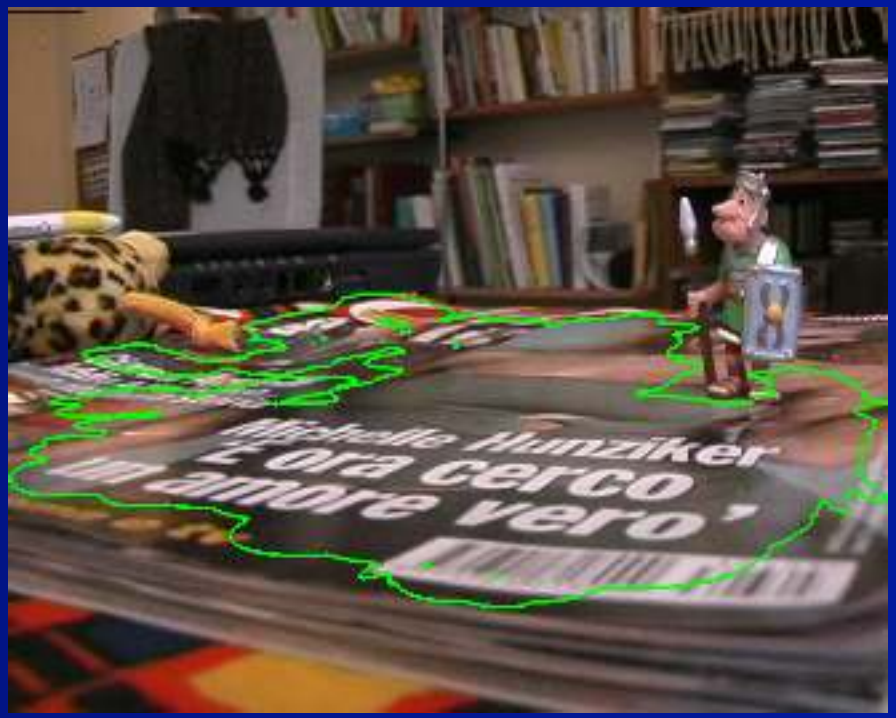
Results



zoom 2x

Large scale change (3x), extensive clutter

Results



very large out-plane-rotation

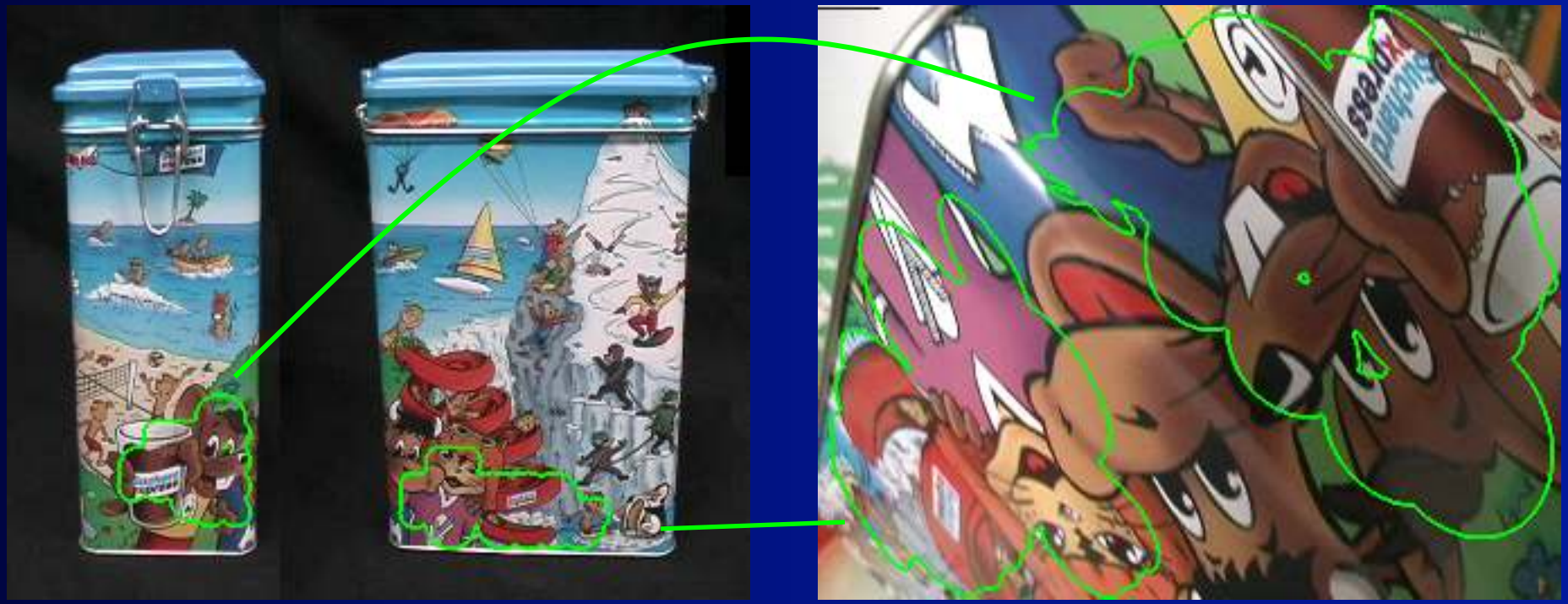
Results

closest model views

some ‘mis-blobs’ (CVPR04)

large viewpoint change, scale, clutter

Results



two matching views

(total coverage)

Small visible portion, scale change 4x

Results

(whole object)

(total coverage)

Extensive clutter, scale, occlusion, blur

Results

(whole object)

(total coverage)

Divided by occlusion, viewpoint change

Results

(whole object)

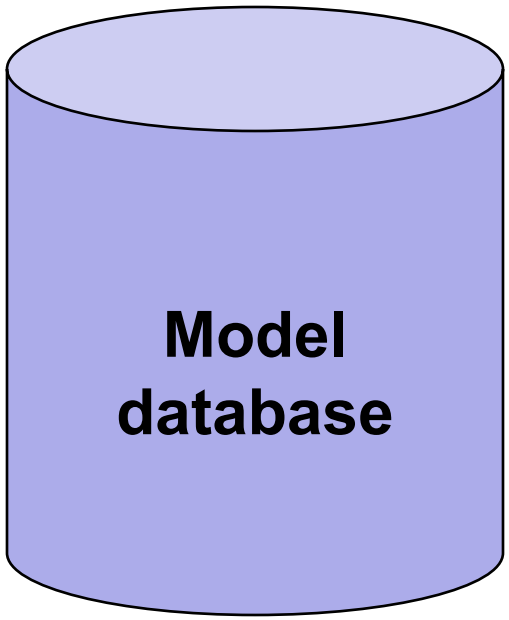
(total coverage)

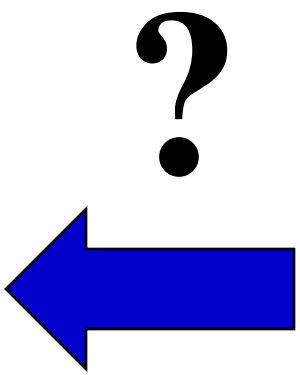
Viewpoint change

Extension 2:
Scaling to large databases

Scalability: matching to large databases

- What if we need to align a test image with thousands or millions of images in a model database?
  - Efficient putative match generation
    - Approximate descriptor similarity search, inverted indices



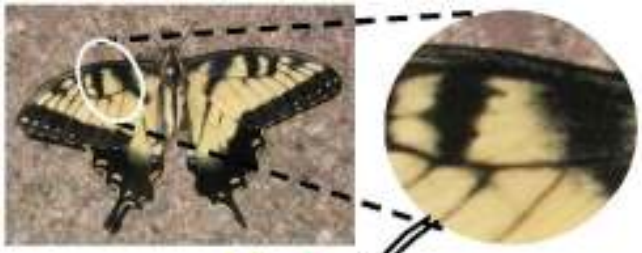


Test image

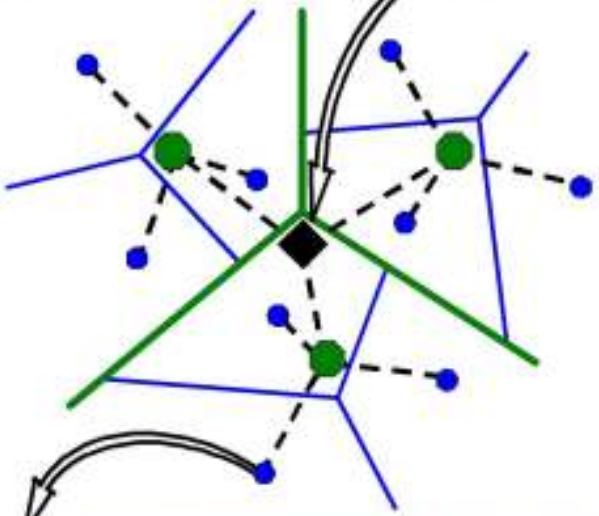
Scalability: matching to large databases

- What if we need to align a test image with thousands or millions of images in a model database?
  - Efficient putative match generation
    - Fast nearest neighbor search, inverted indexes

Test image

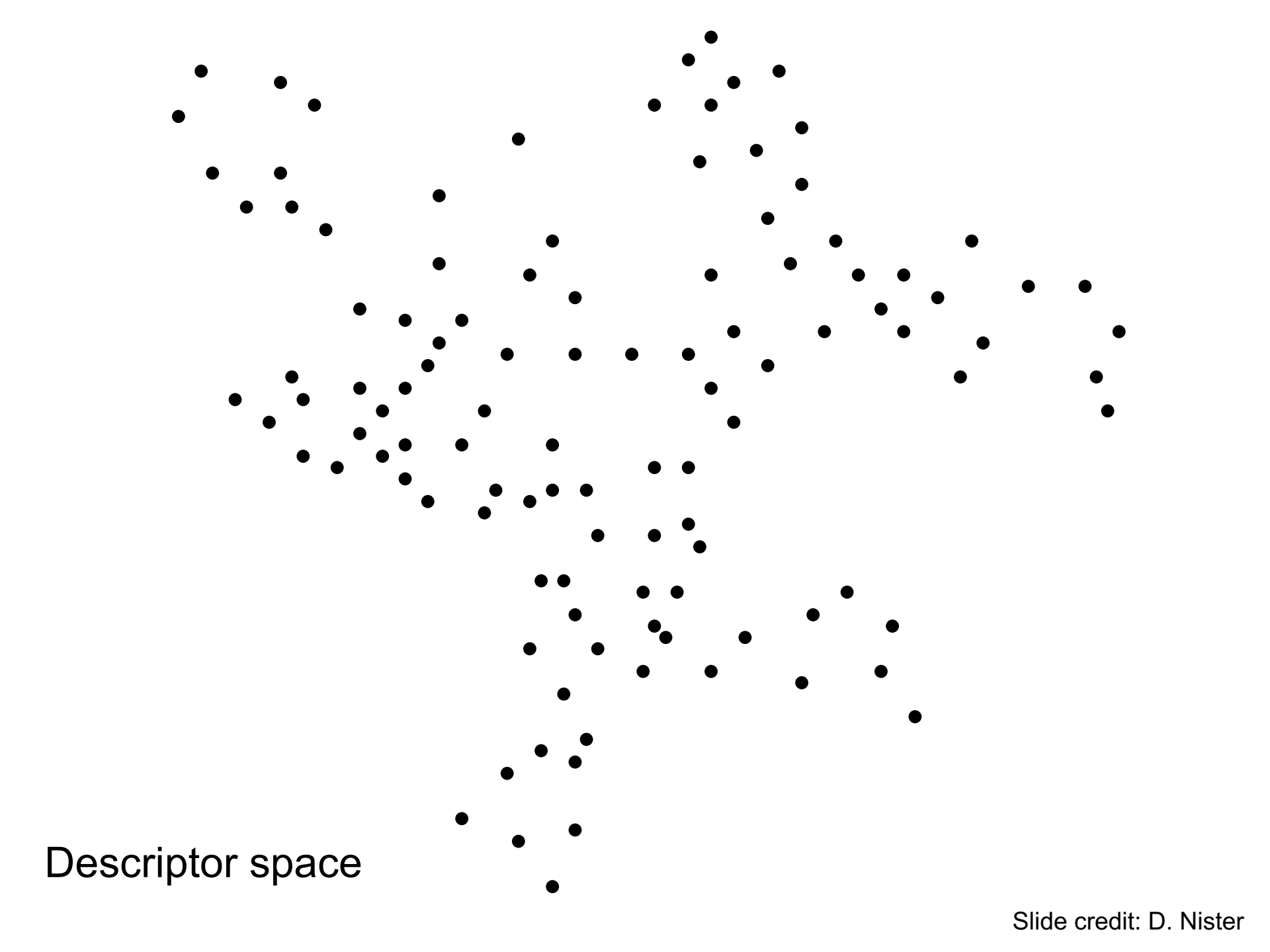


Vocabulary tree with inverted index

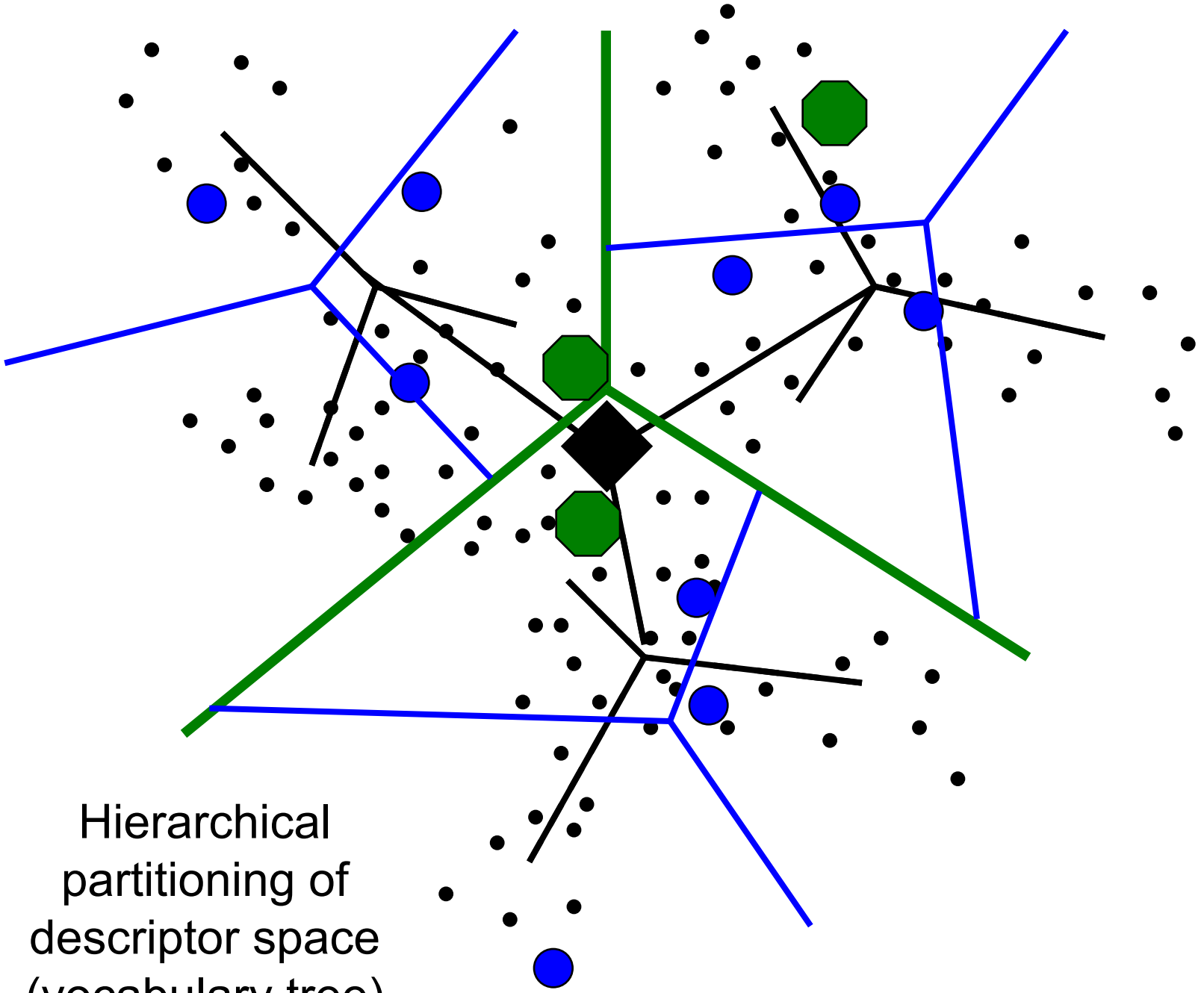


Database

D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

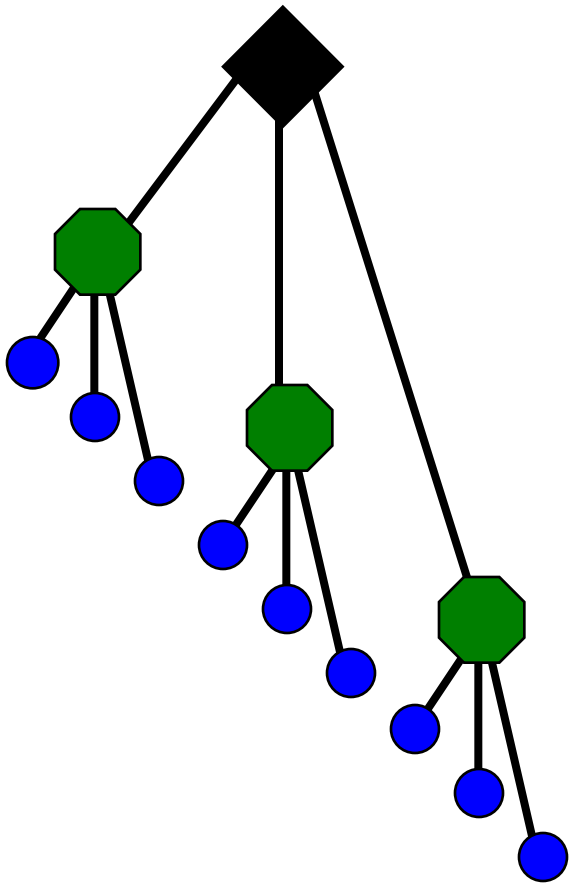


Slide credit: D. Nister



Hierarchical partitioning of descriptor space (vocabulary tree)

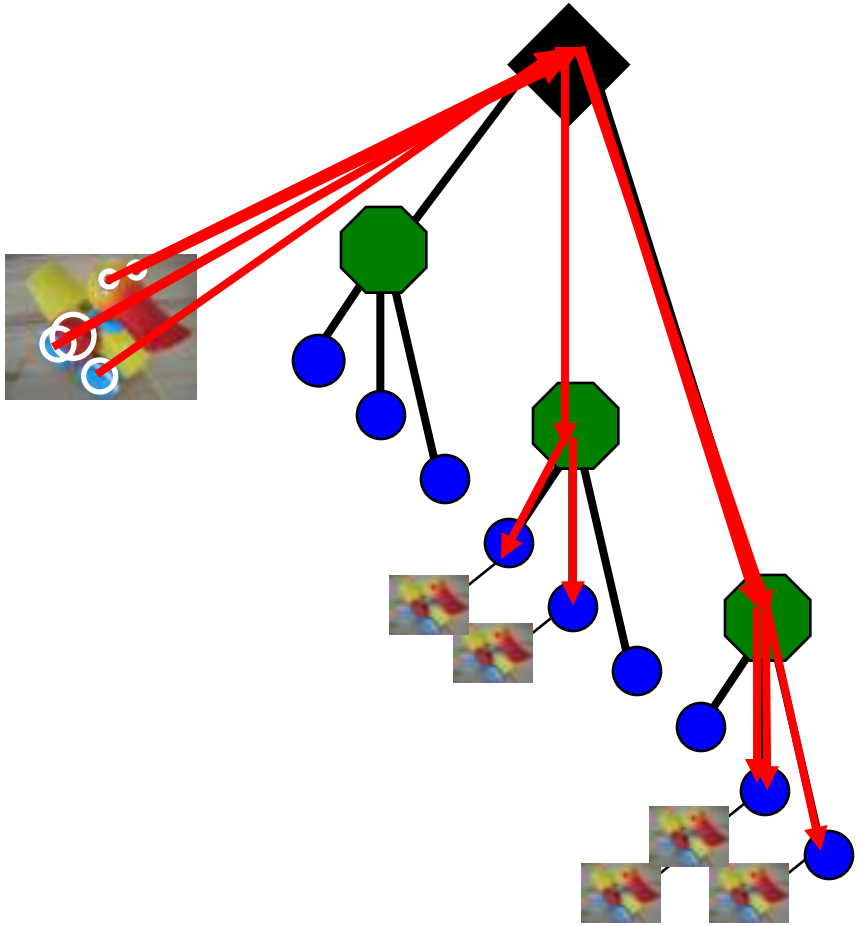
Slide credit: D. Nister



Vocabulary tree/inverted index

Slide credit: D. Nister

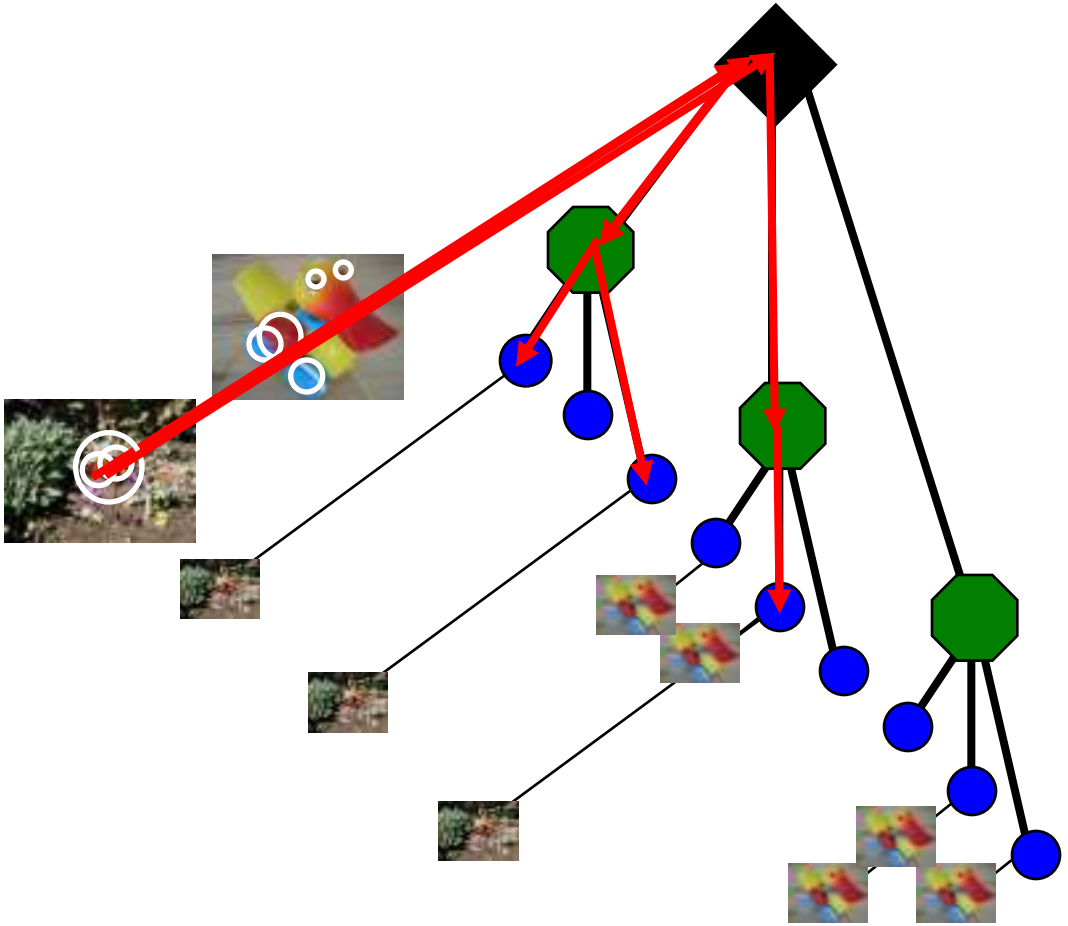
Model images



Populating the vocabulary tree/inverted index

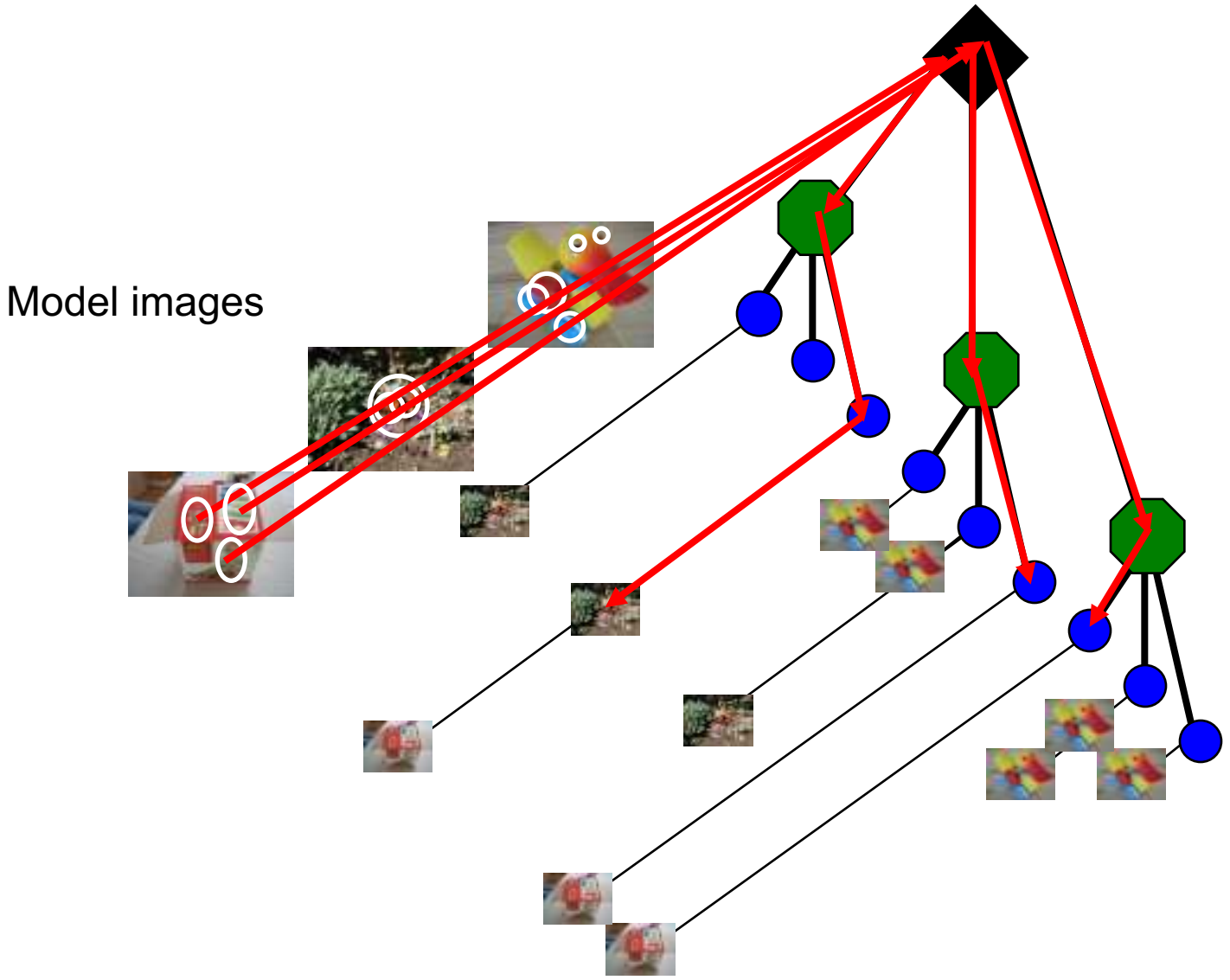
Slide credit: D. Nister

Model images



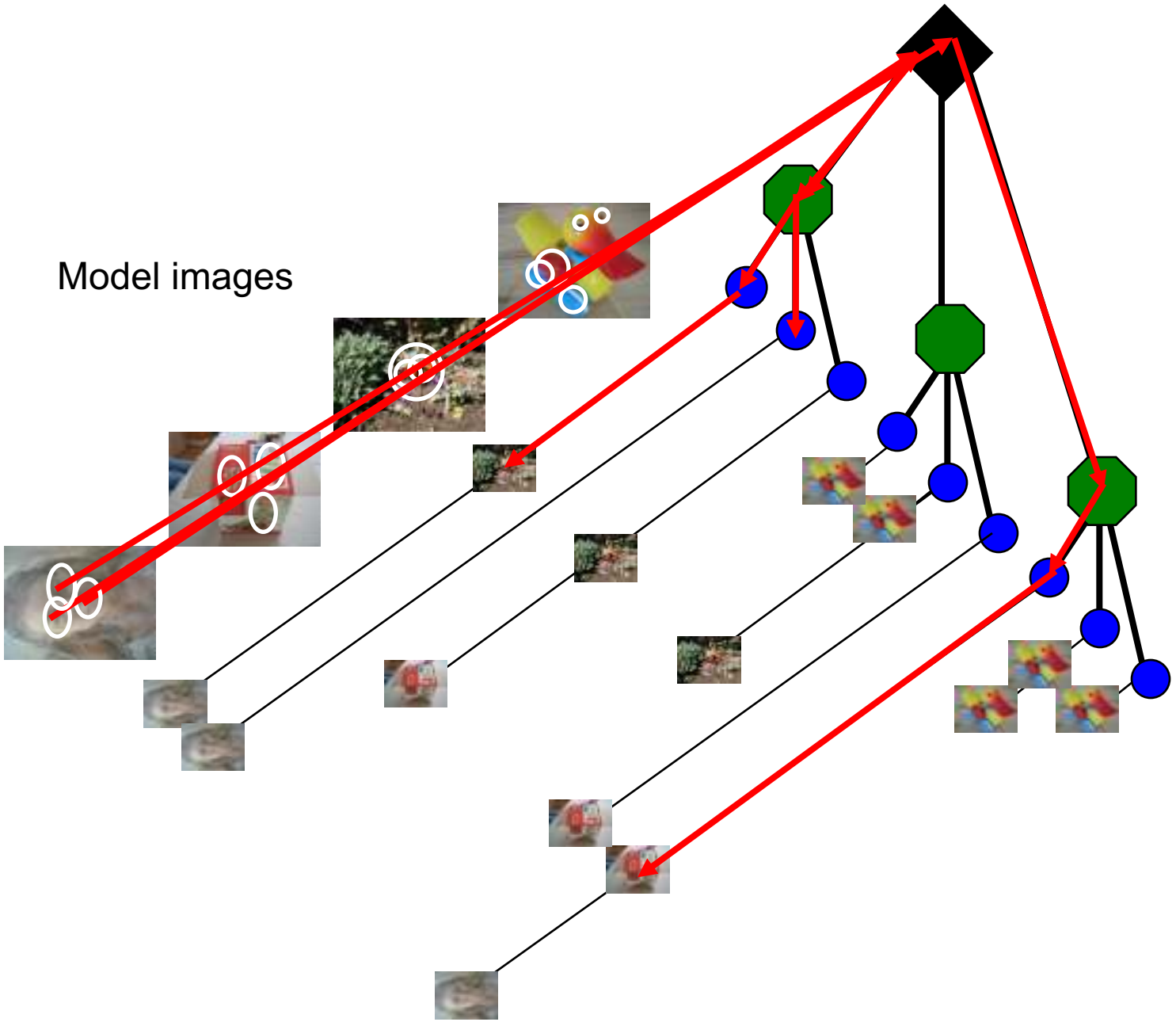
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



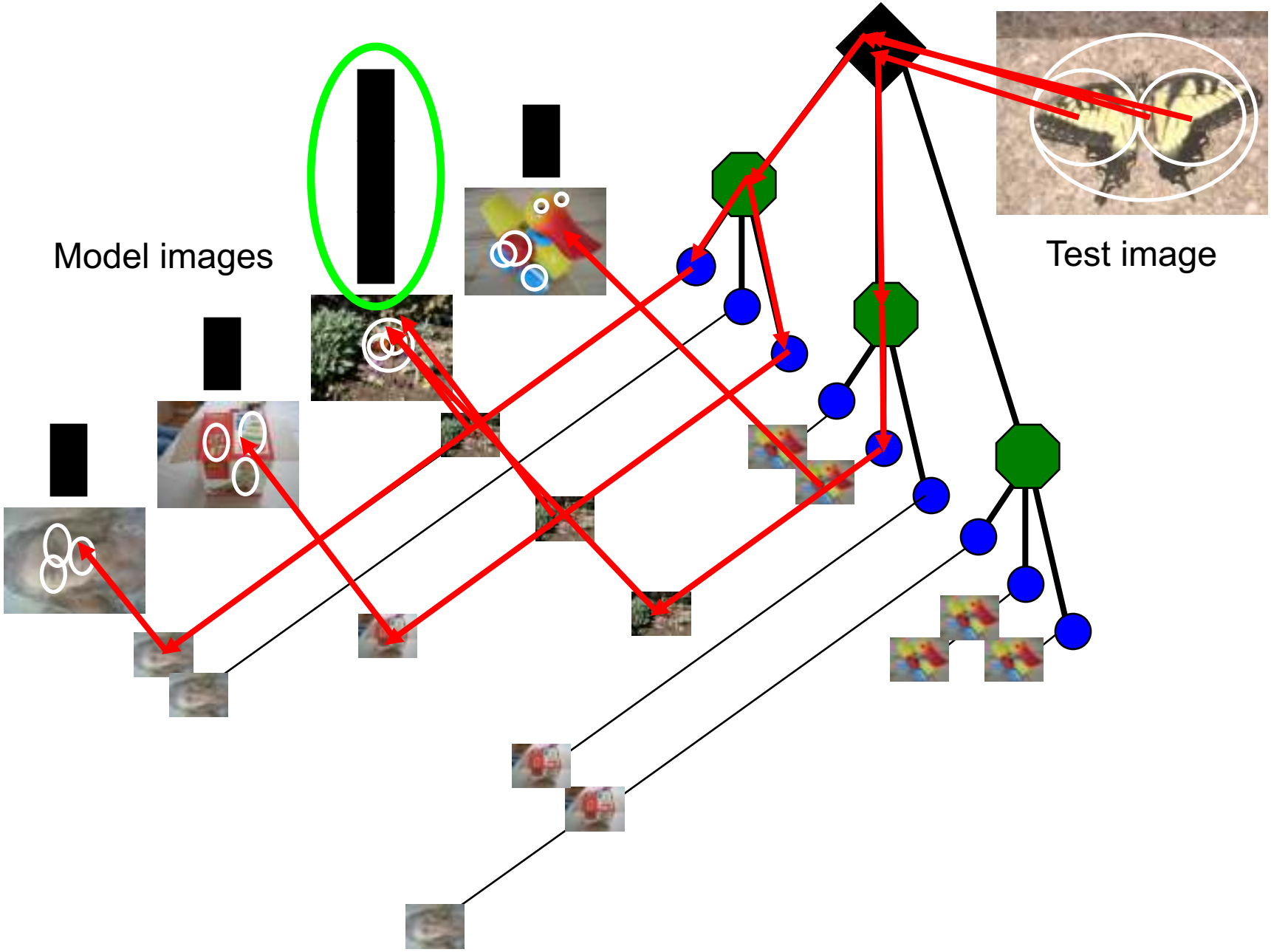
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Looking up a test image

Slide credit: D. Nister

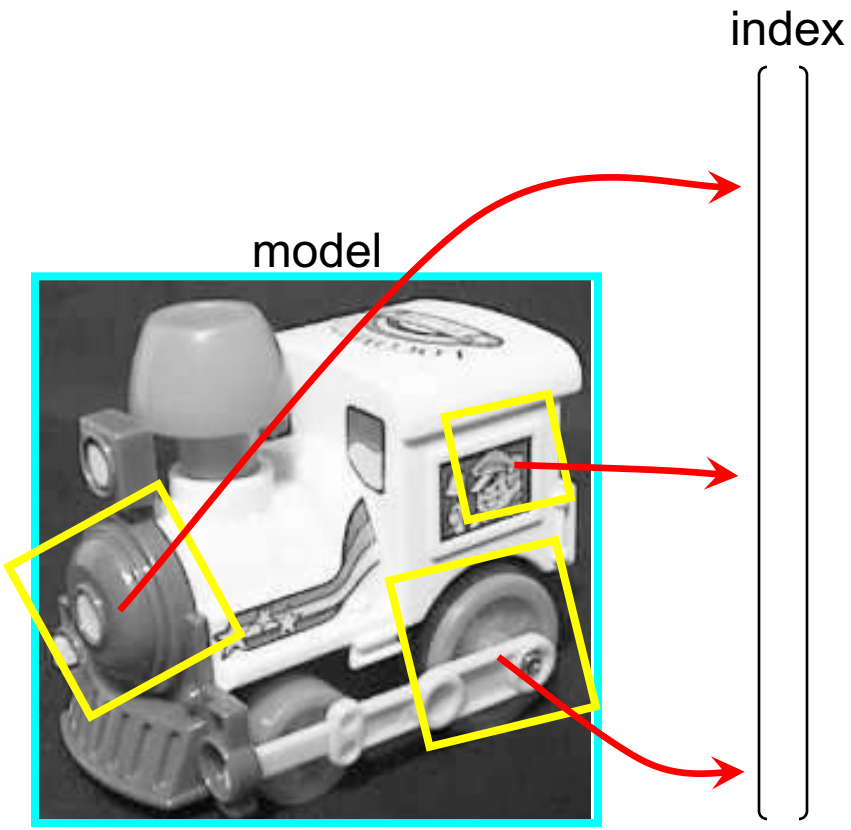
Extremely efficient matching

+ Match a test image to the database in complexity "independent" of the number of images in the database (depends only on the depth of the tree)
+ Matching complexity only O(log(n)) with n the number of leaves in the tree (= quantized 'words')
+ In practice can search instantly in >1M images on a laptop

- Descriptor similarity approximation depends on number of leaves, different densities in different regions of space, …
- Quantization boundary artifacts
- Many matches does not always imply the object is present (no geometric verification yet)

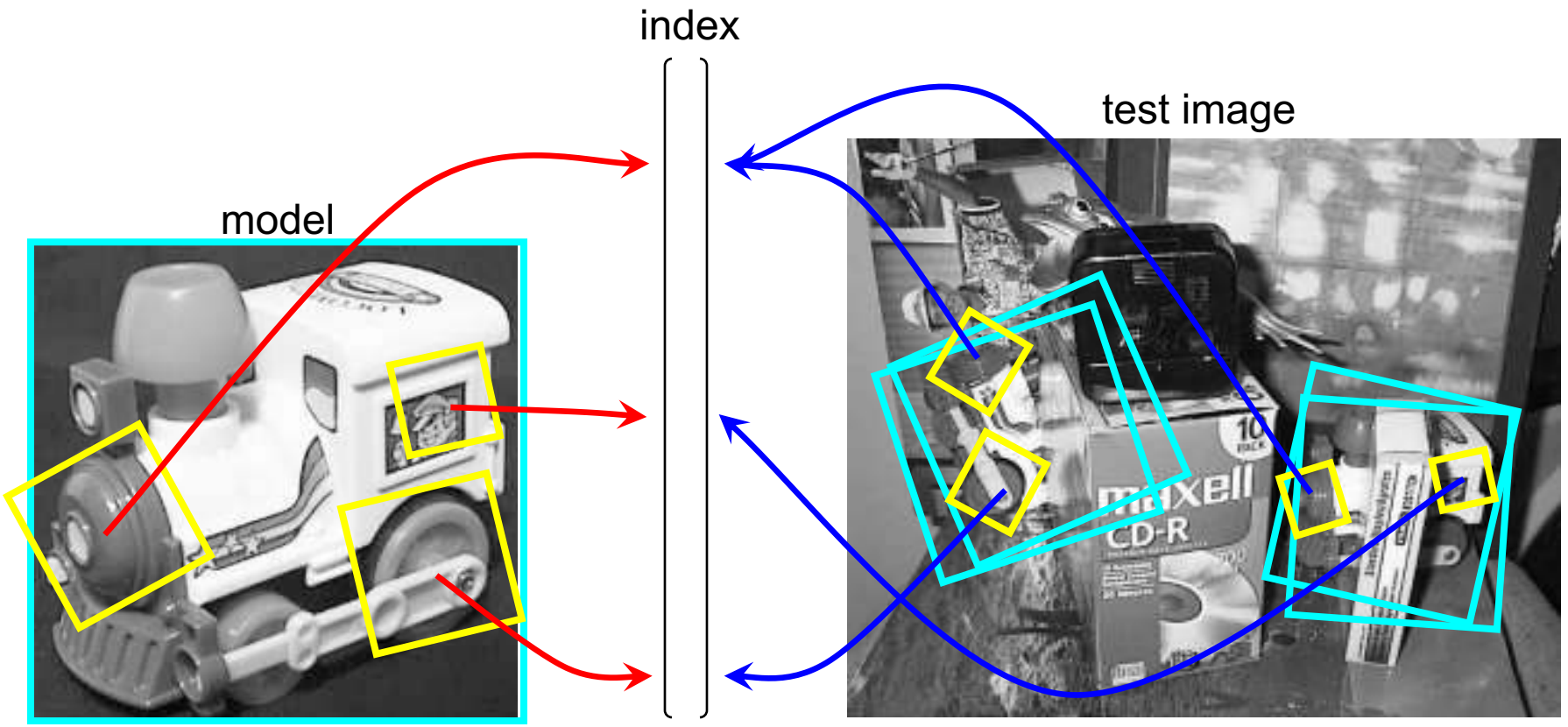
Voting for geometric transformations

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



Voting for geometric transformations

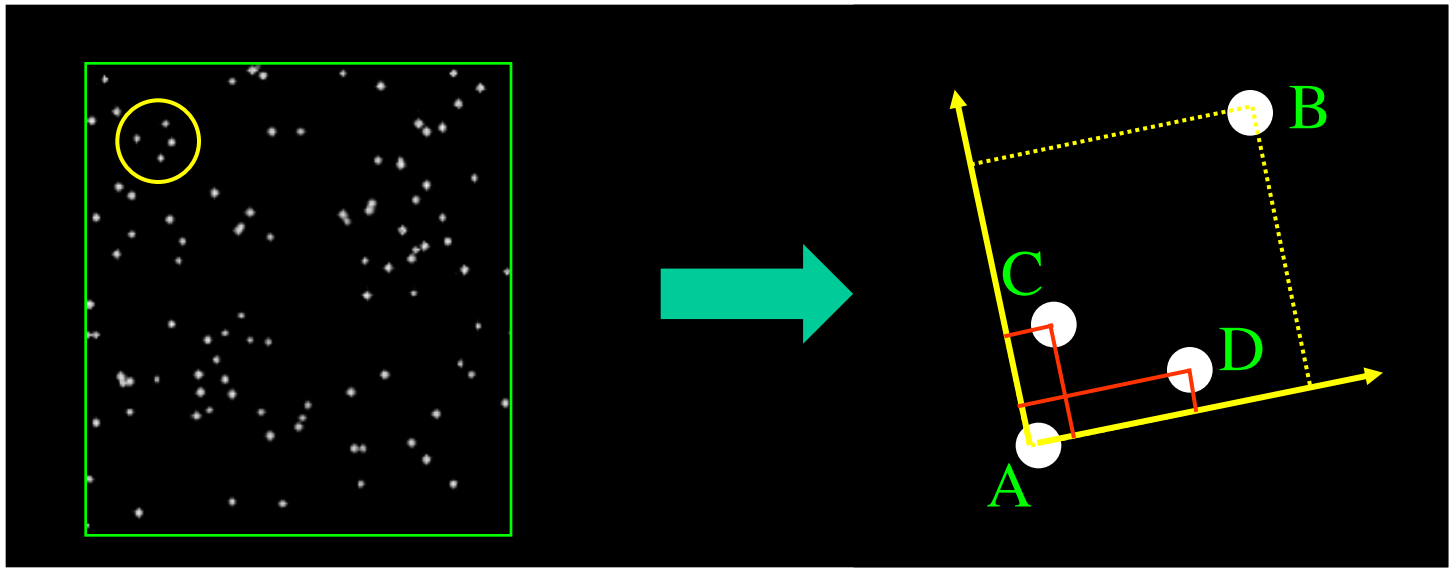
- **Test phase:** Each match between a test and model feature votes in a 4D Hough space (location, scale, orientation) with coarse bins
- Hypotheses receiving some minimal amount of votes can be subjected to more detailed geometric verification



Indexing with geometric invariants

- When we don't have feature descriptors, we can take n-tuples of neighboring features and compute invariant features from their geometric configurations

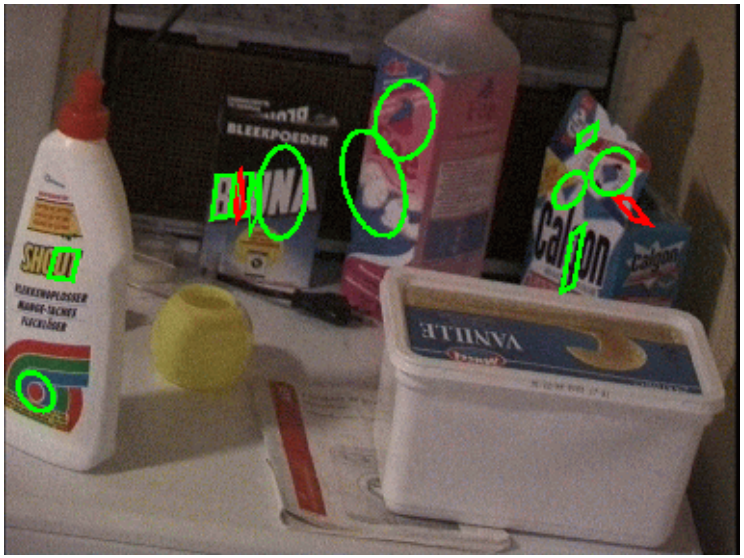
- Application: searching the sky: http://www.astrometry.net/

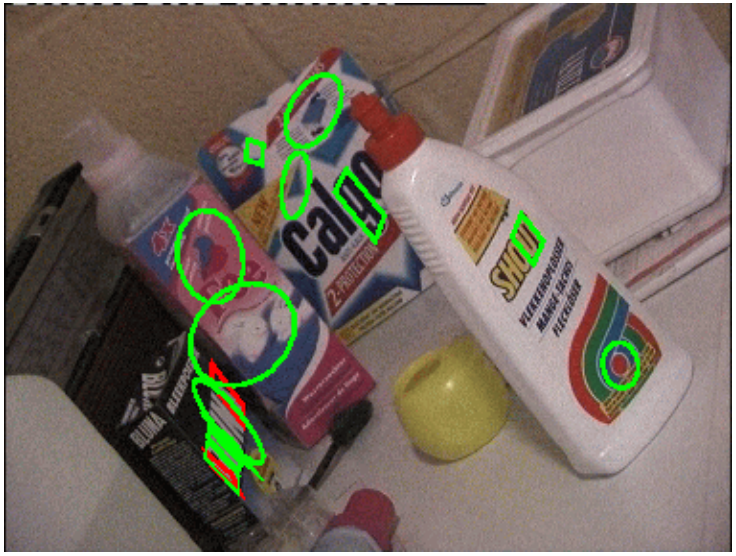


Other applications of invariant regions

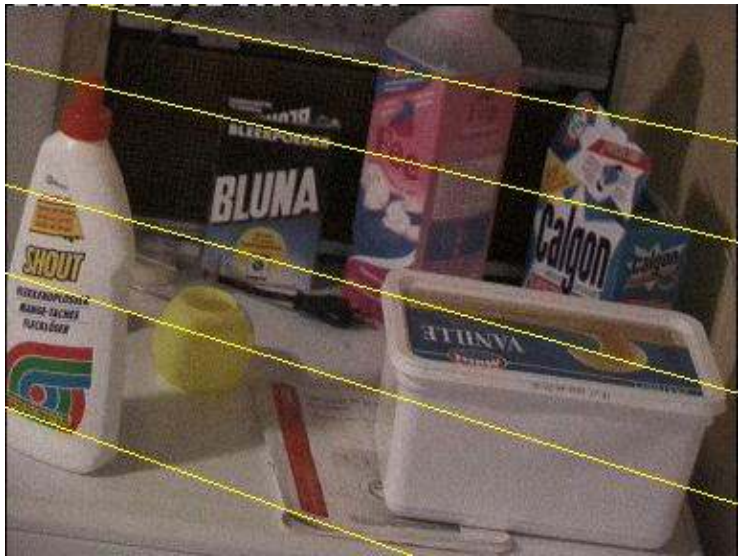
1. **Wide baseline stereo**
2. **Image database retrieval**
3. **Tracking for Augmented Reality**

Wide-baseline stereo





Wide-baseline stereo



Content-based image retrieval from database

= Searching of ‘similar’ images in a database based on image content

- Local features

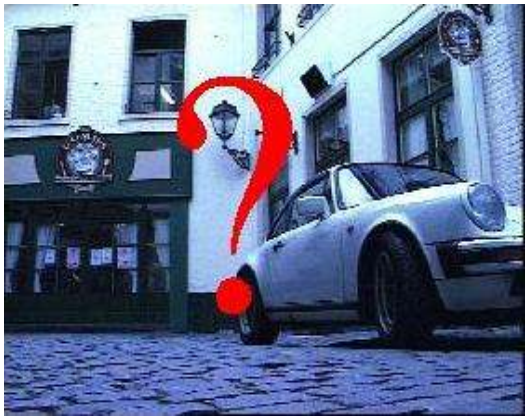
  Similarity = images contain the *same* object or the *same* scene

- Voting principle

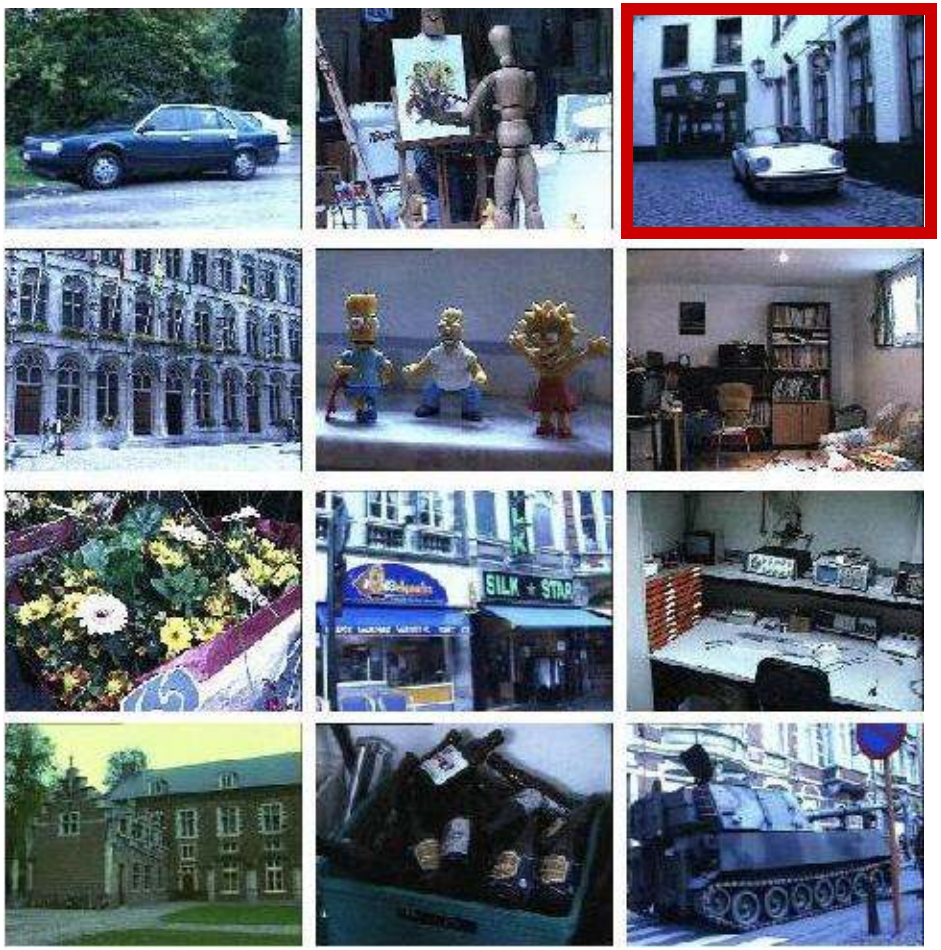
  Based on the number of similar regions

Content-based image retrieval from database

Search image



Database (> 450 images)



Application:
tracker for Augmented Reality