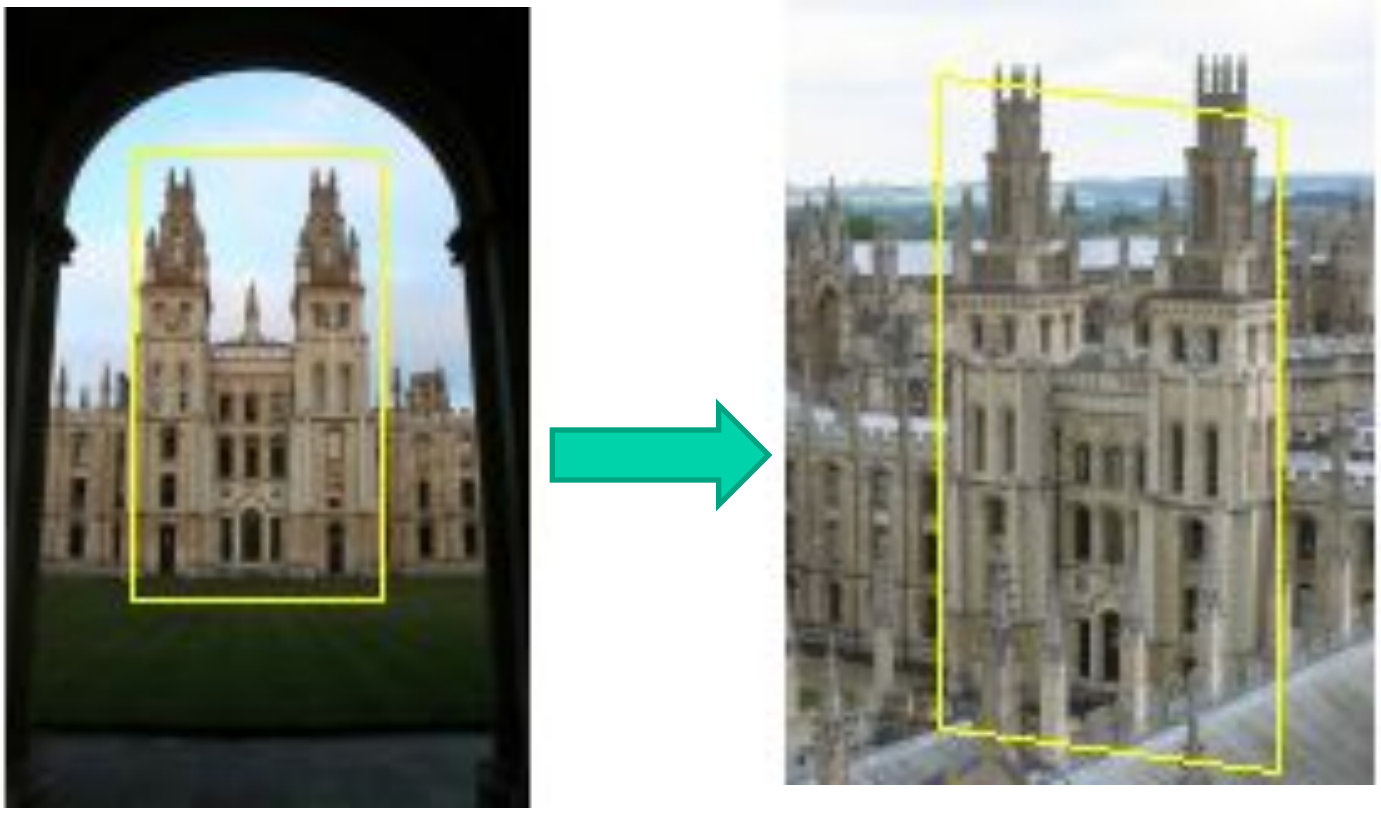
Specific Object Recognition



Object recognition example:



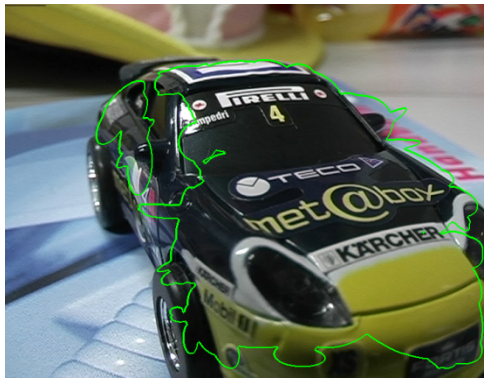
Computer Vision

Specific objects vs. category-level objects







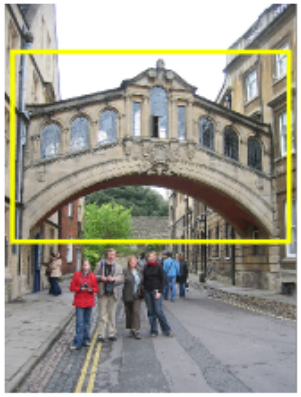




A *specific object* = an instance of an object class
e.g. “my car” instead of “a car”

Example app

search photos on the web for particular places







Find these landmarks ...in these images and 1M more

Slide credit: J. Sivic

Two 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)

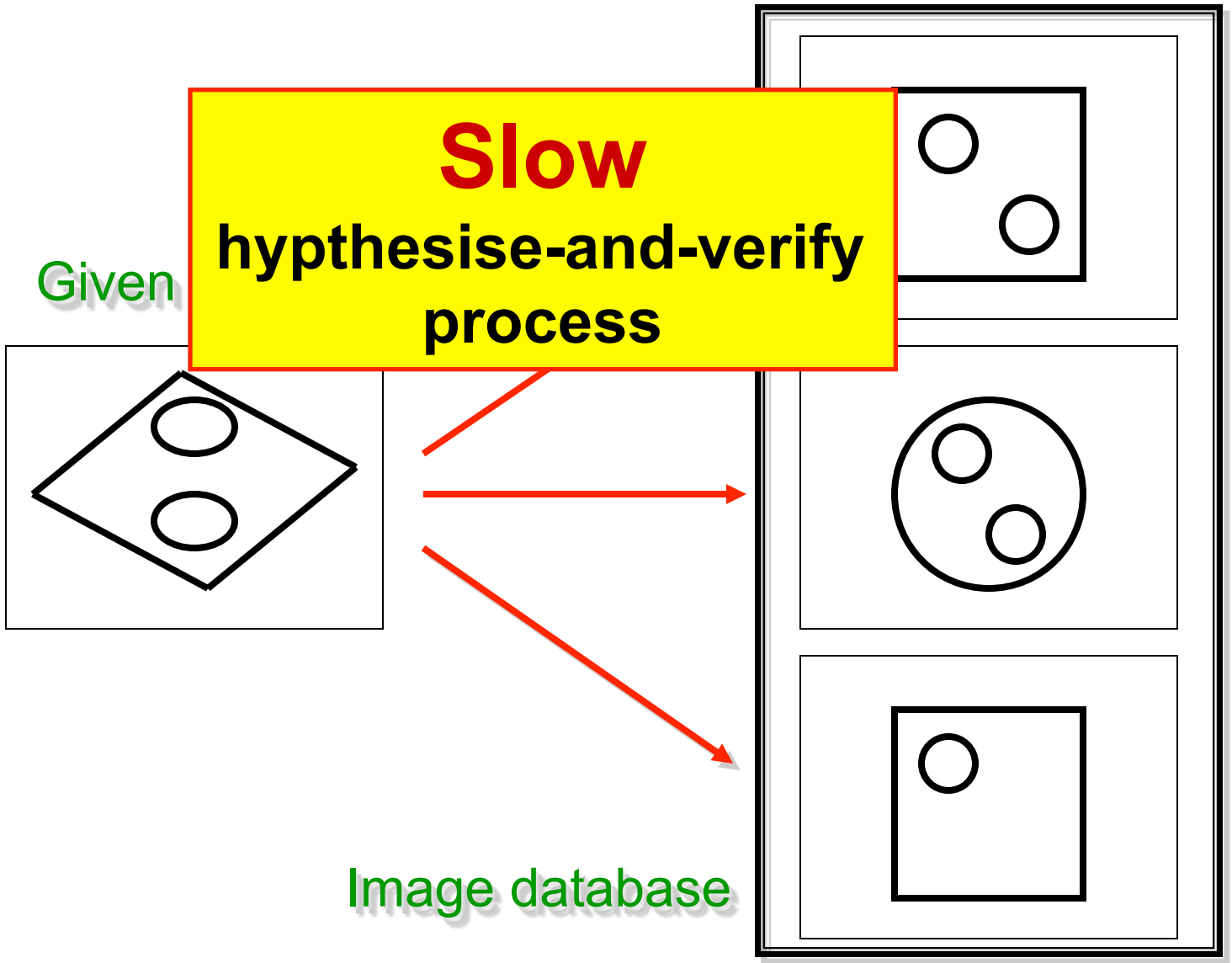
- `model' = all possible views
- acquiring representative data set cumbersome
- can deal with complex objects
- difficulties with variable backgrounds

Hybrid approaches

Model-based

Once upon a time...

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



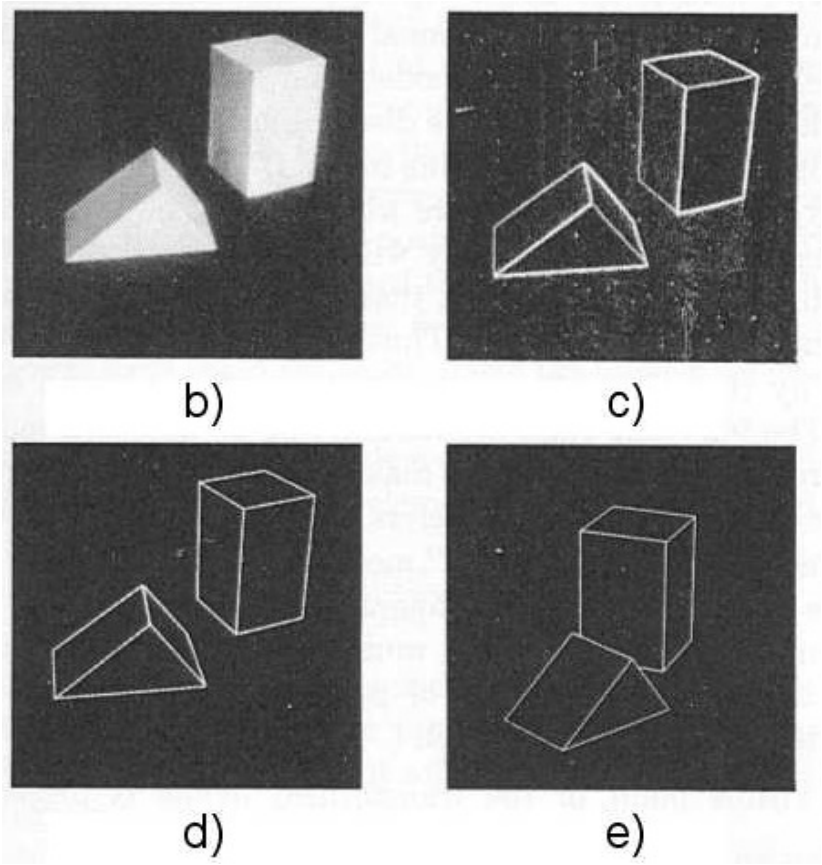
Model-based approaches





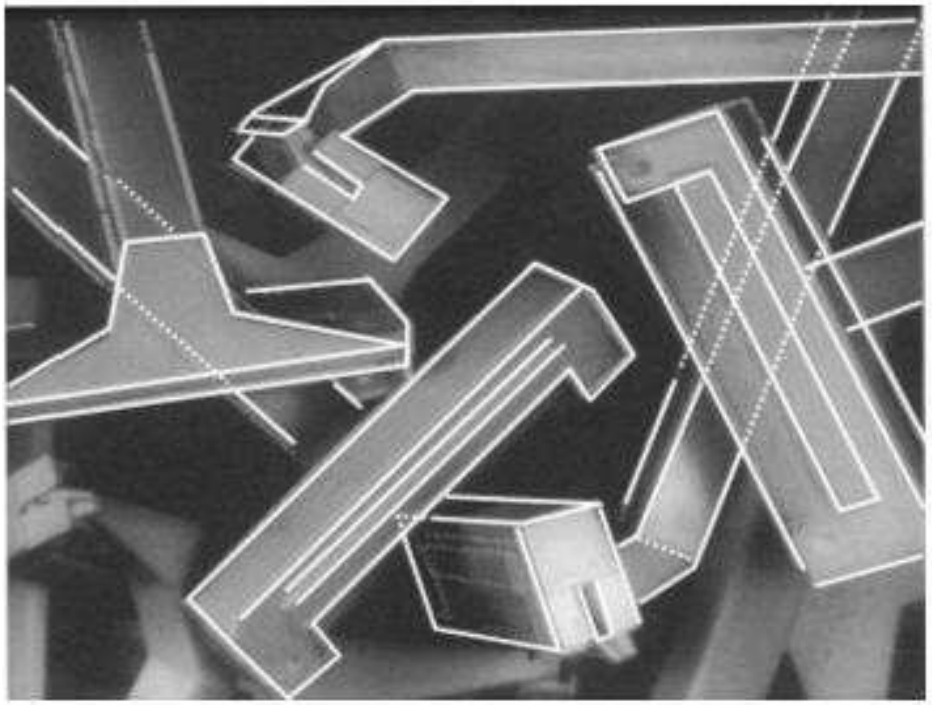
Computer Vision

Early attempts…1965



Blocks world model, Roberts et al., 1965

Early attempts…1985



Dealing with occlusions, Lowe, 1985

More recent Example: Invariant-based recognition of planar shapes

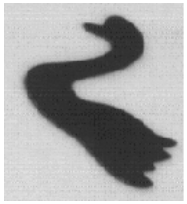
The crucial advantage of invariants is that they decouple object type and pose issues

Invariant-based 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





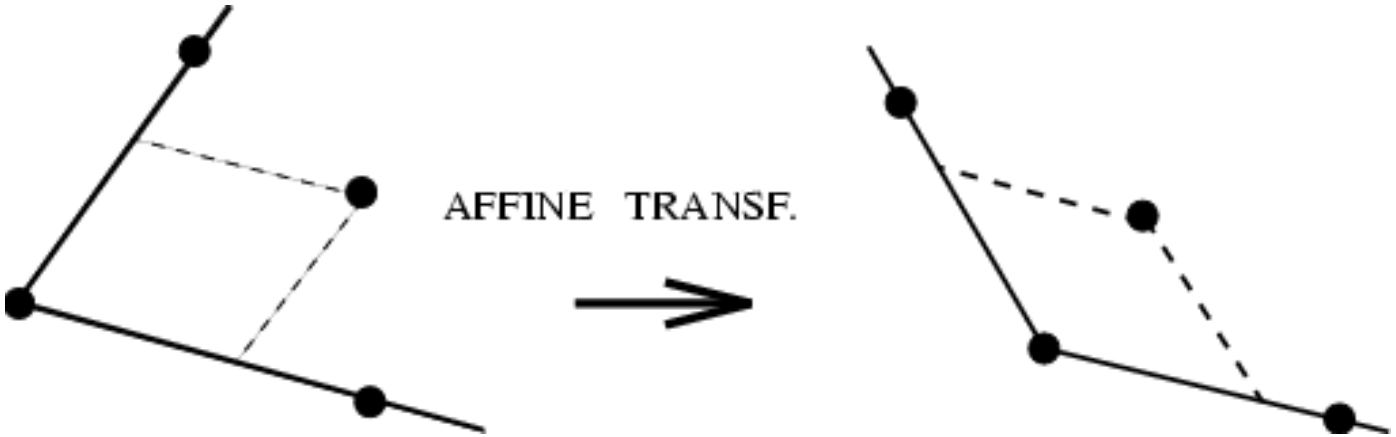
Invariants under affine transf. : ex 1

ratios of areas

8 data items – 6 parameters affine transf.
→ 2 independent, absolute invariants

affine *invariant coordinates* (x_A, y_A) :

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



Computer Vision

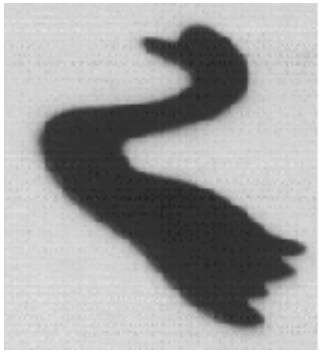
(Rel.) inv. under affine transf. : ex 2

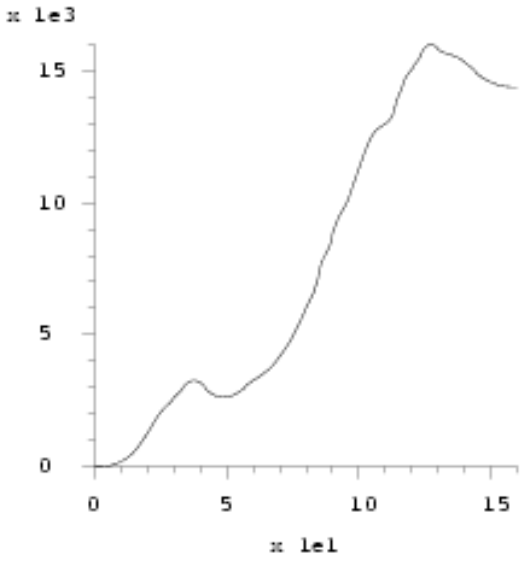
$$\int_{\text{start pt}} \left| \bar{x} - \bar{x}_1 \quad \bar{x}^{(1)} \right| dt$$

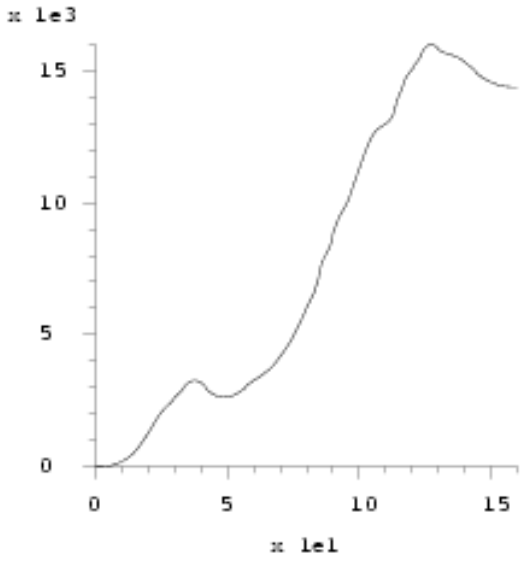
As a function of

$$\int_{\text{start pt}} abs\left(\left| \bar{x}^{(1)} \quad \bar{x}^{(2)} \right|^{\frac{1}{3}} \right) dt$$

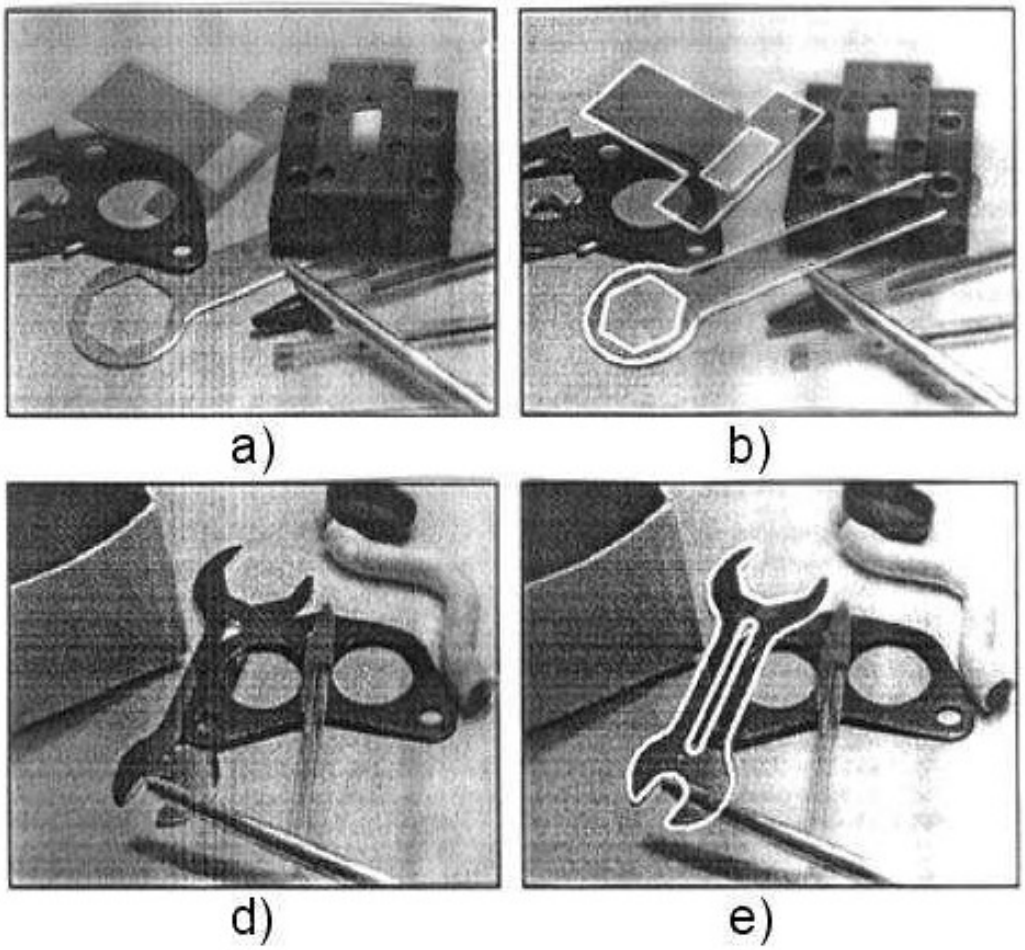








Early attempts…1992



Projective invariance, Rothwell, 1992

Image-based

Appearance based methods

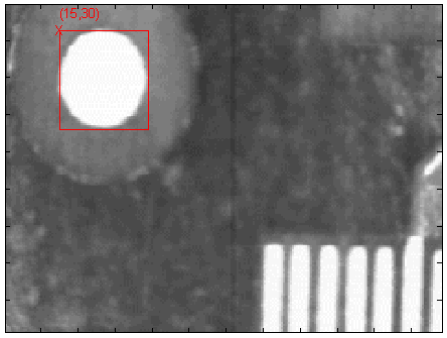
The model of an object is simply its image(s).

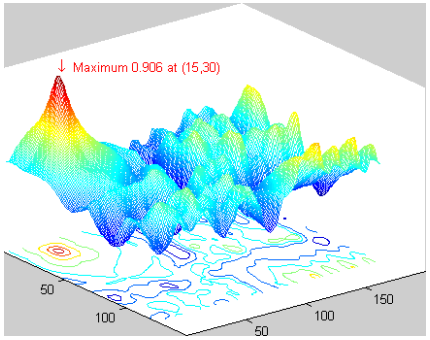
A simple example: Template matching

Shift the template over the image and compare (e.g. NCC or SSD)



Template





The power of Principal Component Anal.

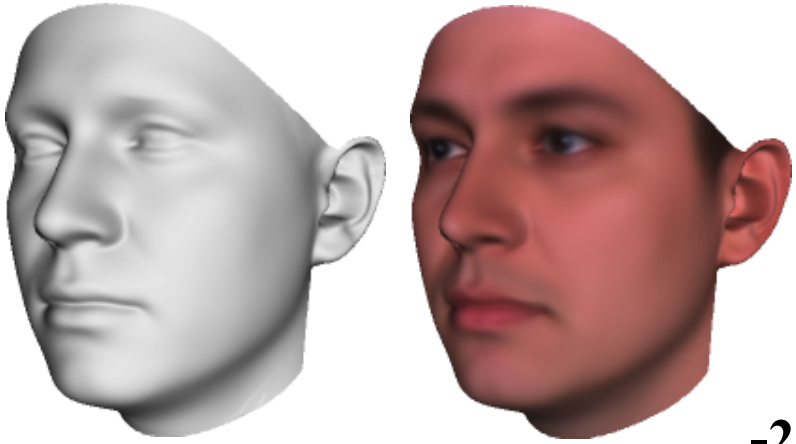
You remember PCA?

… and the Karhunen-Loeve example?

PCA represents data in a lower-dimensional space keeping most of the variance

It was seen to be powerful for similar patterns like faces, that exhibit a lot of redundancy

Eigenfaces for compact face representation



-2*s* +2*s* -2*s* +2*s*





$$\mathbf{S} = \overline{\mathbf{S}} + \sum_{j=1}^{m} \alpha_j \mathbf{S}_j$$

$$\mathbf{T} = \overline{\mathbf{T}} + \sum_{j=1}^{m} \beta_j \mathbf{T}_j$$

Eigenfaces for compact face representation



Modes

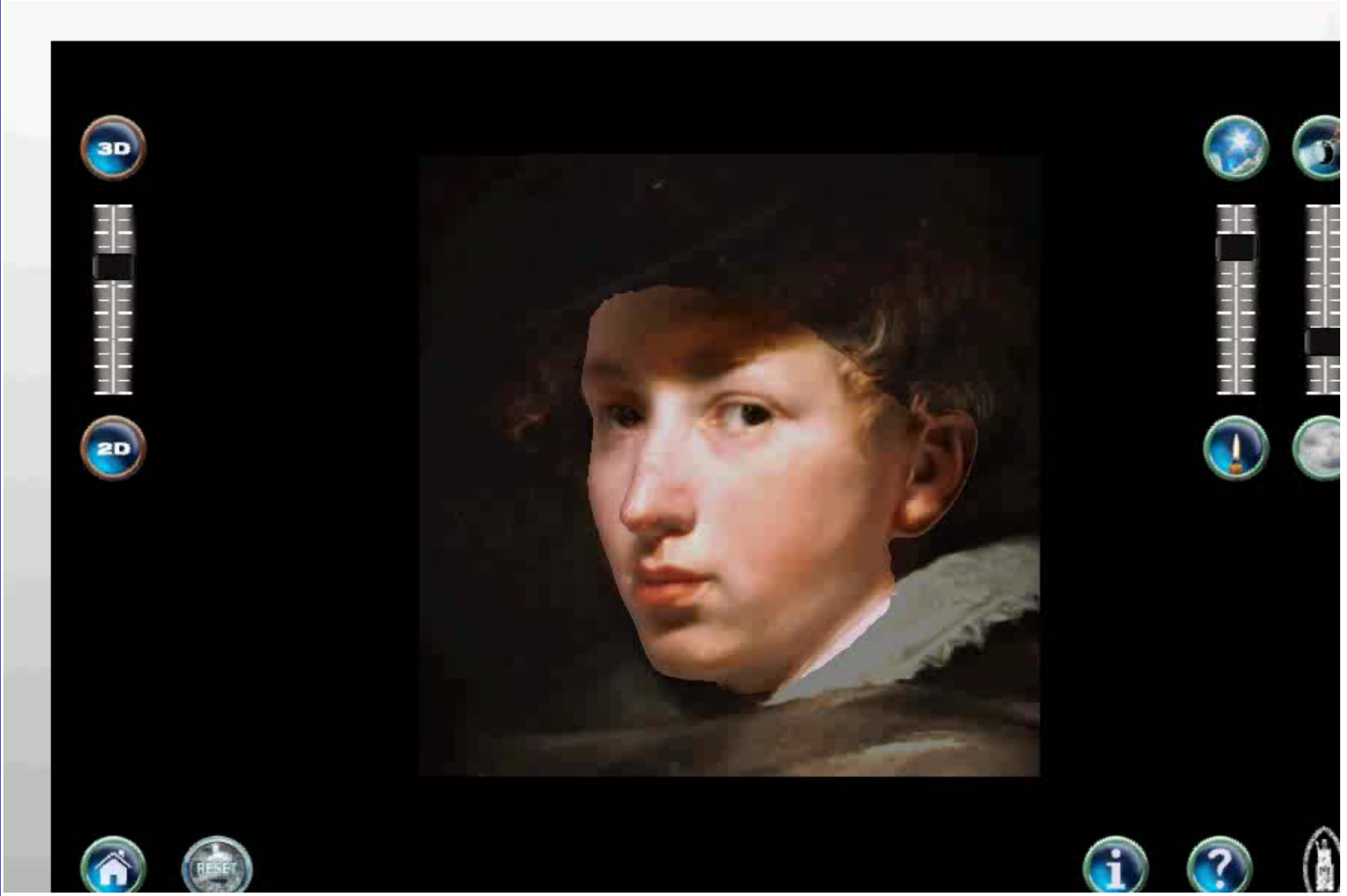
Morphs

Eigenfaces for compact face representation



(self?-) portrait of the young Antoon Van Dijck

Eigenfaces for compact face representation



Computer Vision

Eigenfaces for compact face representation

3D PCA-based reconstruction



Appearance manifold approach

(Nayar et al. '96)

Training

for every object :

- sample the set of viewing conditions (mainly vps)
- 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

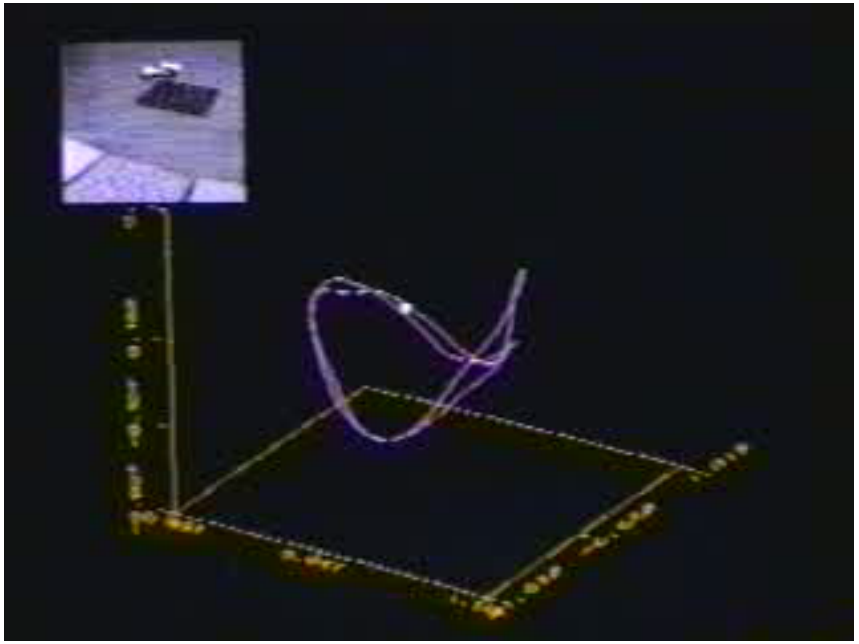
what is the nearest manifold to a test image?



Object-pose manifold

Appearance changes projected on PCs (1D pose changes)

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 (can’t handle clutter)

Comparison between model-based and appearance-based techniques

Pure model-based

Compact model
Can deal with clutter
Slow analysis-by-synthesis
Models difficult to produce
For limited object classes

Pure appearance-based

Large models
Cannot deal with clutter
Efficient
Models easy to produce
For wide classes of objects

Hybrid techniques

Euclidean invariant feature

(Schmid and Mohr '97)

Training

- look for corners (with the Harris detector)
- take circular regions around these points, of different radii (cope a bit with scale changes)
- calculate from the regions invariants under plane rotation
- do this from different viewpoints, where the invariance cuts down on the number of views needed (e.g. no in-plane rotations necessary)

Testing

- compare these invariants with those found for images in the database
- look for consistent placement of candidate matches / epipolar geometry
- decide which object based on the number of matches

Euclidean invariant features

Example (rotation) invariant:

$$G_x G_x + G_y G_y$$

Where G_x and G_y represent horizontal and vertical derivatives of intensity weighted by a Gaussian profile

2nd example invariant:

$$G_{xx} + G_{yy}$$

Where G_{xx} and G_{yy} represent 2nd order Gaussian derivatives

(Compute at different scales, i.e. take into account explicitly)

Example

Training examples



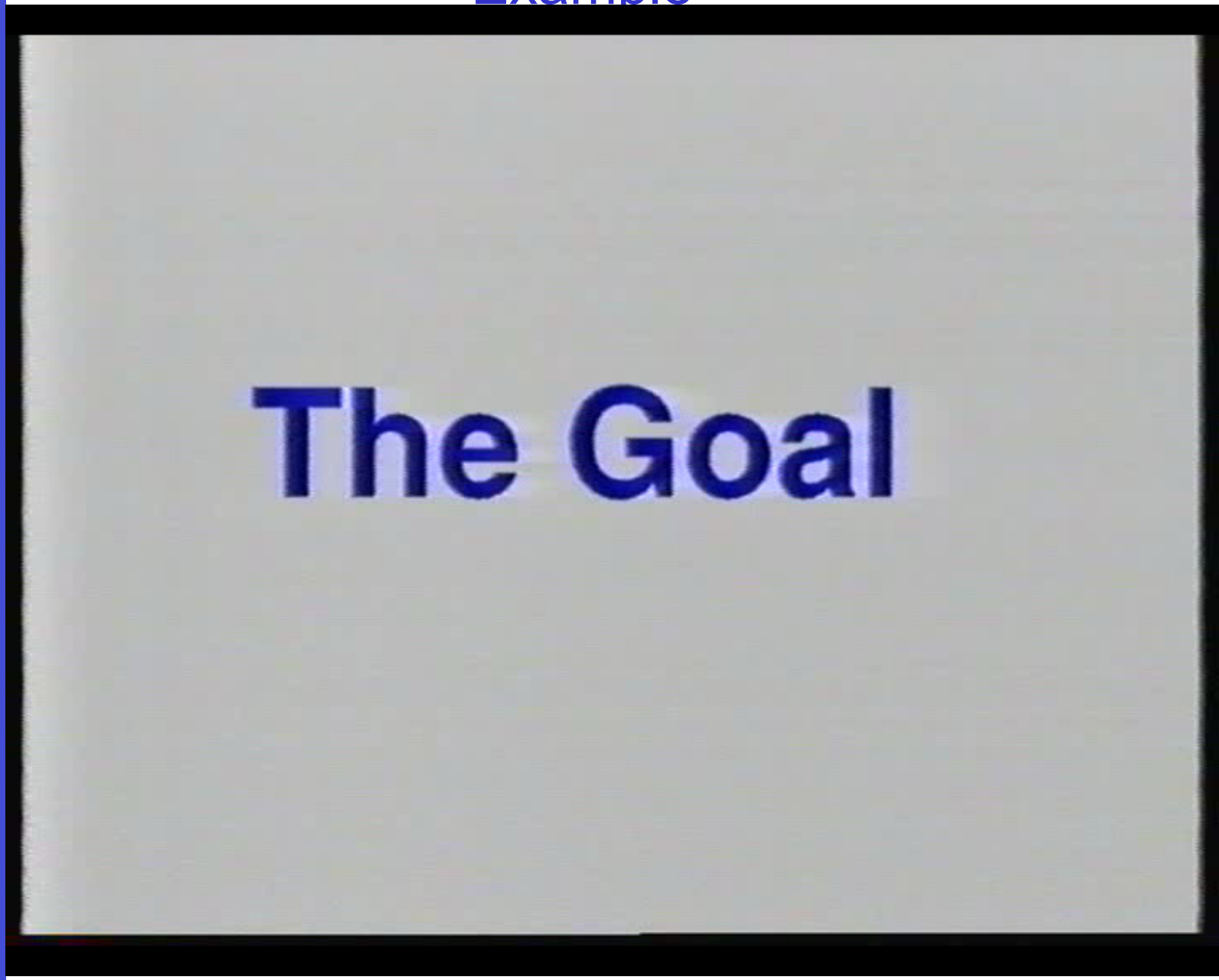
Test image



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

Computer Vision

Example



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)

Recognition using local affine and photometric invariant features

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

Viewpoint

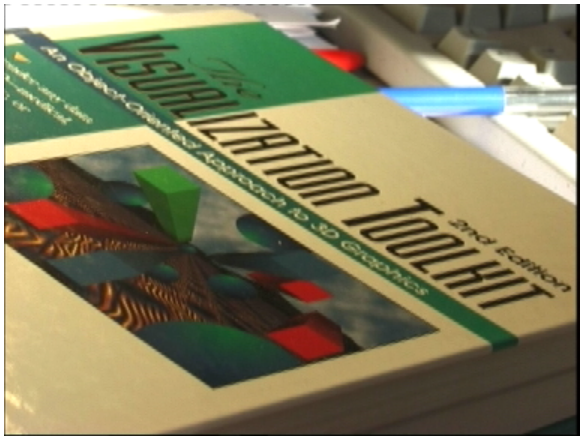




Recognition using local affine and photometric invariant features

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

- Viewpoint
- Illumination

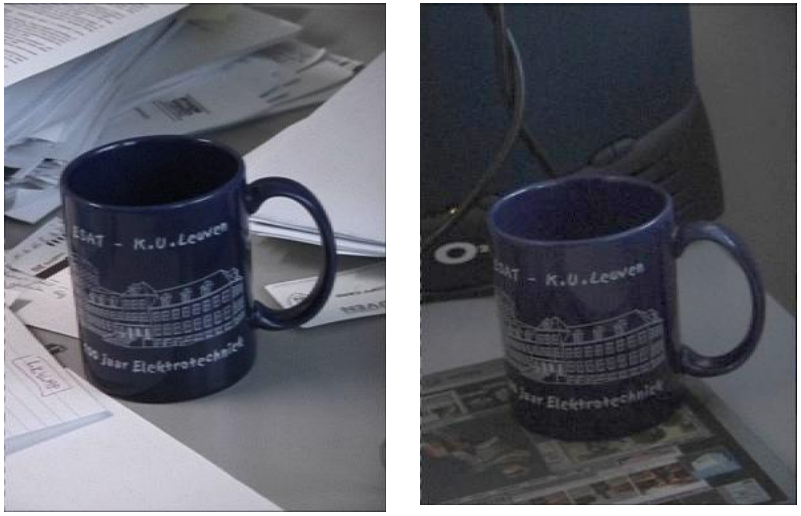




Recognition using local affine and photometric invariant features

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

- Viewpoint
- Illumination
- Background



Recognition using local affine and photometric invariant features

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

- Viewpoint
- Illumination
- Background
- and Occlusions

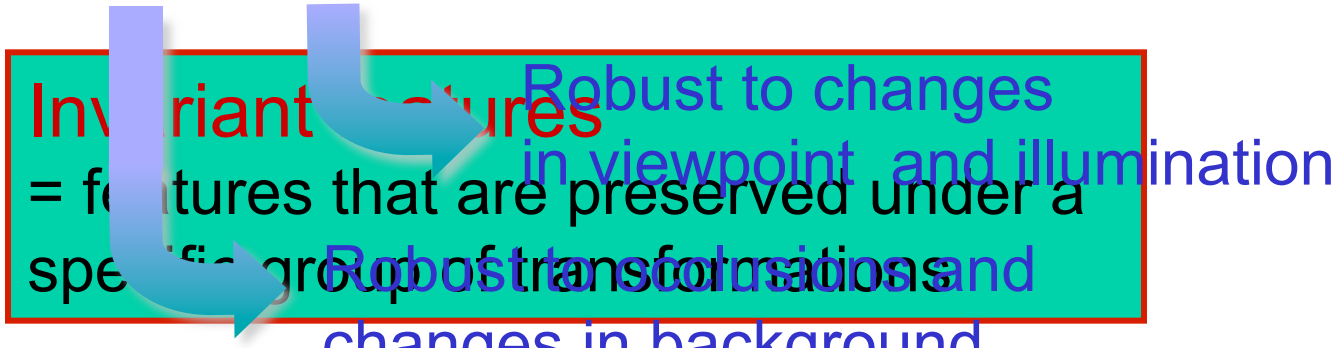


Recognition using local affine and photometric invariant features

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

- Viewpoint
- Illumination
- Background
- and Occlusions

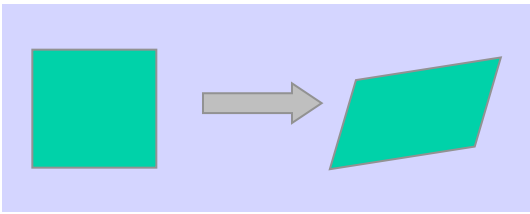
⇒ Use local invariant features



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

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

Local features: desired properties

Repeatability

The same feature can be found in several images despite geometric and photometric transformations

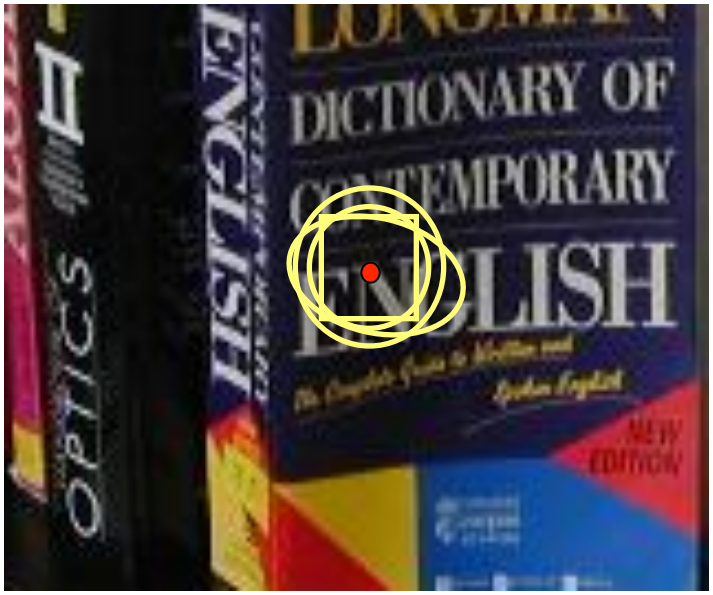
Distinctiveness

Each feature has a distinctive description

As we saw we can go further with invariance than similarities (as in ex), hoping to increase repeatability, but it risks to reduce distinctiveness

Local invariant features



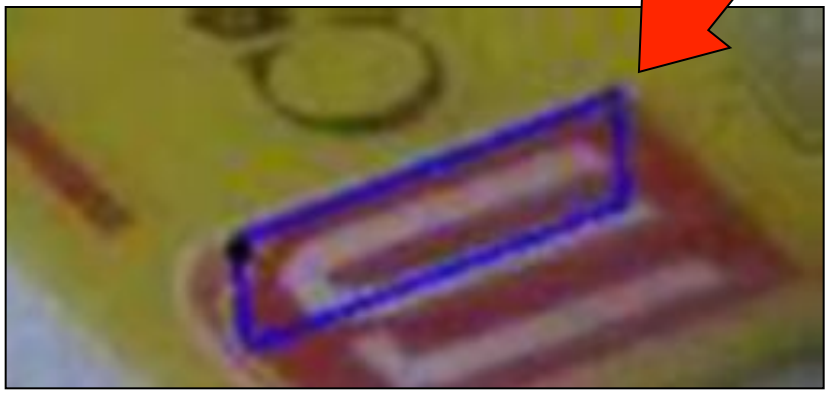


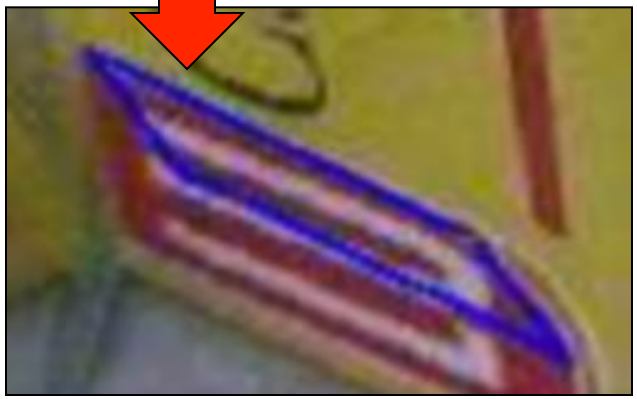
… e.g. by going for invariance under affinities rather than similarity

Increasing the level of invariance: `Invariant Neighbourhoods'

And regions extracted based on local info only







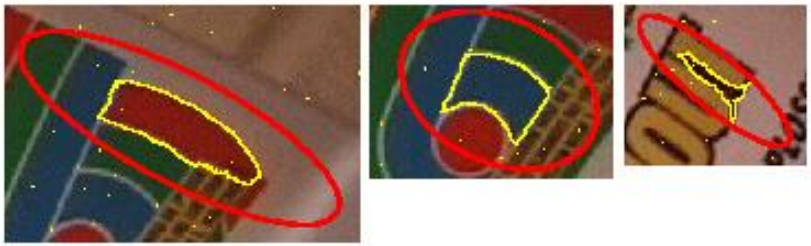
Computer Vision

Intensity based method







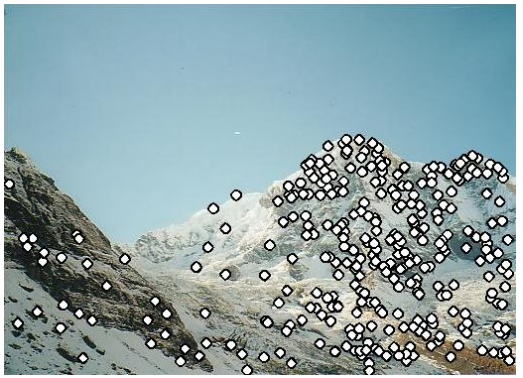


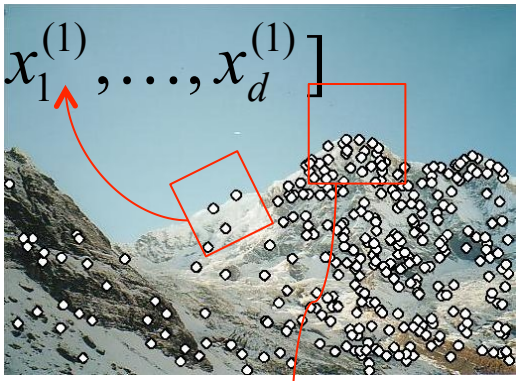
Local features: main components

1) **Detection:** Identify interest points

2) **Description:** Extract feature vector descriptors around them

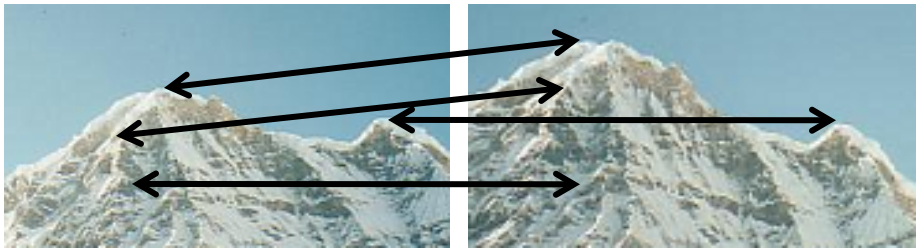
3) **Matching:** Determine correspondence between descriptors in two views





$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$



Robustifying

Hybrid techniques

Supporting the matching step

1) Too slow if naively done

2) Will often fail when only based on descriptor matching

Supporting the matching step

1) Too slow if naively done

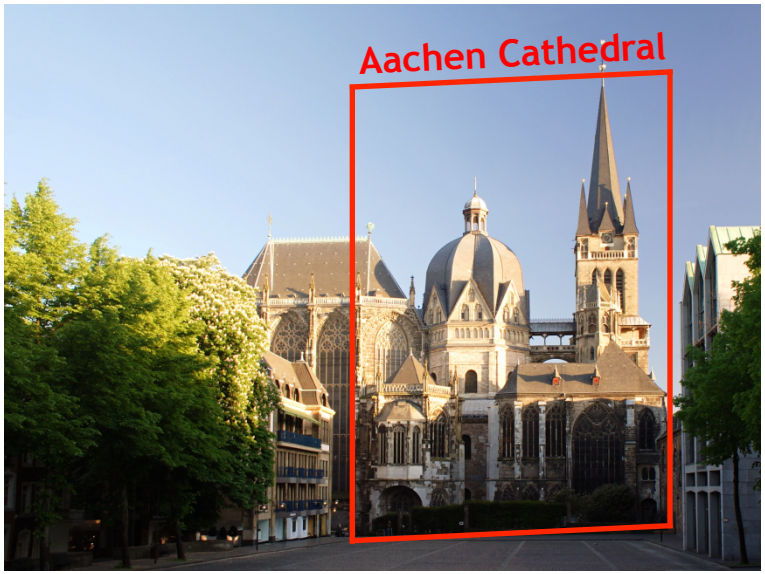
2) Will often fail when only based on descriptor matching

Supporting the matching step

1) Hierarchical vocabulary tree for speed-up

Example Applications



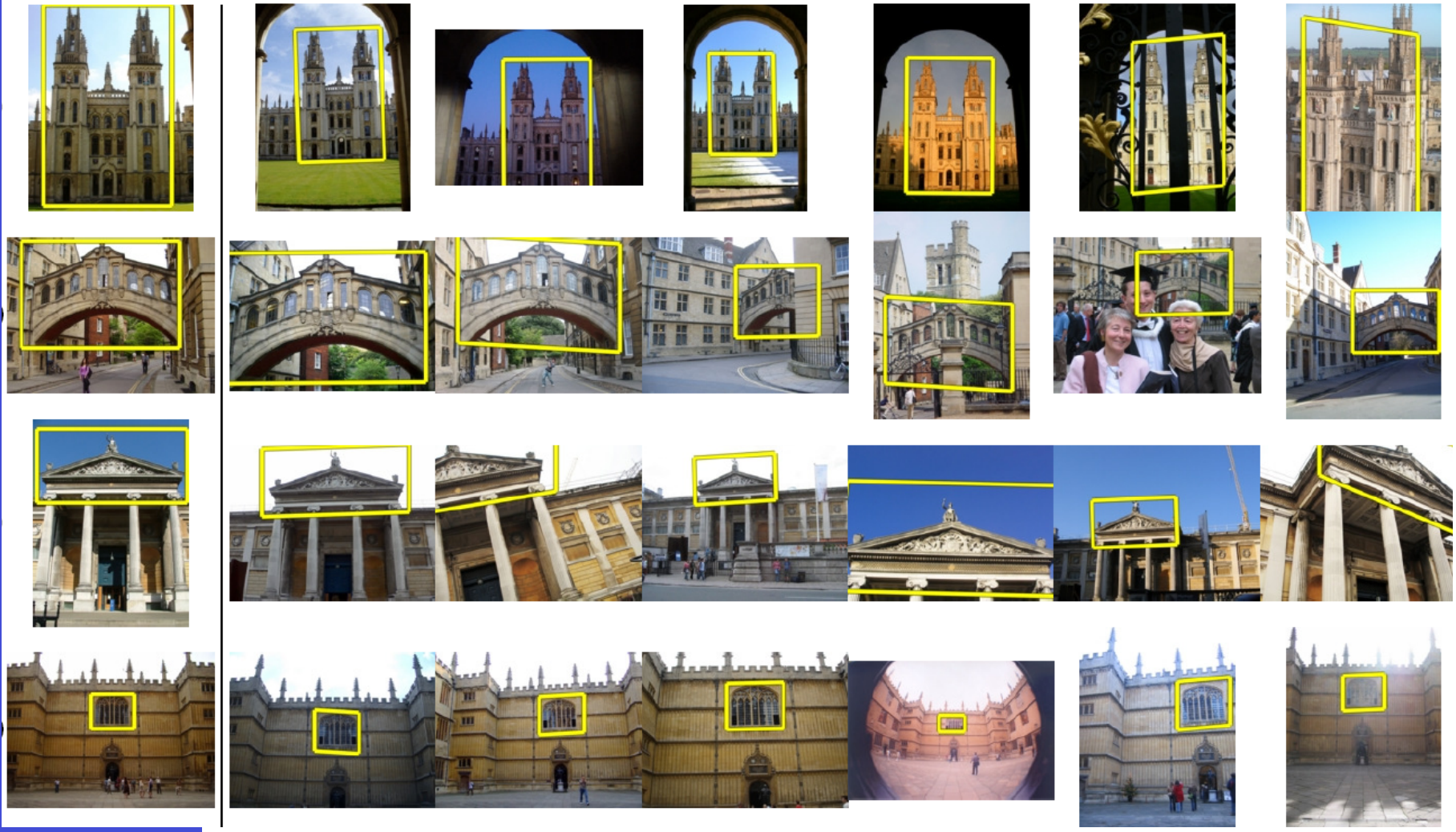


Mobile tourist guide

- Self-localization
- Object / landmark recognition
- Augmented reality
- …
- e.g. kooaba

[Quack, Leibe, Van Gool, CIVR'08]

Application: Large-Scale Retrieval



Query Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR'07]

Indexing local features

With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

Quantize/cluster the descriptors into `visual words'

Low-dimensional descriptors : can use efficient data structures for nearest neighbor search

High-dimensional descriptors: approximate nearest neighbor search methods more practical

Inverted file indexing schemes

Computer Vision

Indexing local features: inverted file index



Index

"Along I-75," From Detroit to Florida; *inside back cover*
"Drive I-95," From Boston to Florida; *inside back cover*
1929 Spanish Trail Roadway; 101-102,104
511 Traffic Information; 83
A1A (Barrier Isl) - I-95 Access; 86
AAA (and CAA); 83
AAA National Office; 88
Abbreviations,
Colored 25 mile Maps; cover
Exit Services; 196
Travelogue; 85
Africa; 177
Agricultural Inspection Stns; 126
Ah-Tah-Thi-Ki Museum; 160
Air Conditioning, First; 112
Alabama; 124
Alachua; 132
County; 131
Alafia River; 143
Alapaha, Name; 126
Alfred B Maclay Gardens; 106
Alligator Alley; 154-155
Alligator Farm, St Augustine; 169
Alligator Hole (definition); 157
Alligator, Buddy; 155
Alligators; 100,135,138,147,156
Anastasia Island; 170
Anhaica; 108-109,146
Apalachicola River; 112
Appleton Mus of Art; 136
Aquifer; 102
Arabian Nights; 94
Art Museum, Ringling; 147
Aruba Beach Cafe; 183
Aucilla River Project; 106
Babcock-Web WMA; 151
Bahia Mar Marina; 184
Baker County; 99
Barefoot Mailmen; 182
Barge Canal; 137
Bee Line Expy; 80
Butterfly Center, McGuire; 134
CAA (see AAA)
CCC, The; 111,113,115,135,142
Ca d'Zan; 147
Caloosahatchee River; 152
Name; 150
Canaveral Natnl Seashore; 173
Cannon Creek Airpark; 130
Canopy Road; 106,169
Cape Canaveral; 174
Castillo San Marcos; 169
Cave Diving; 131
Cayo Costa, Name; 150
Celebration; 93
Charlotte County; 149
Charlotte Harbor; 150
Chautauqua; 116
Chipley; 114
Name; 115
Choctawatchee, Name; 115
Circus Museum, Ringling; 147
Citrus; 88,97,130,136,140,180
CityPlace, W Palm Beach; 180
City Maps,
Ft Lauderdale Expwys; 194-195
Jacksonville; 163
Kissimmee Expwys; 192-193
Miami Expressways; 194-195
Orlando Expressways; 192-193
Pensacola; 26
Tallahassee; 191
Tampa-St. Petersburg; 63
St. Augsutine; 191
Civil War; 100,108,127,138,141
Clearwater Marine Aquarium; 187
Collier County; 154
Collier, Barron; 152
Colonial Spanish Quarters; 168
Columbia County; 101,128
Coquina Building Material; 165
Corkscrew Swamp, Name; 154
Cowboys; 95
Crab Trap II; 144
Cracker, Florida; 88,95,132
Driving Lanes; 85
Duval County; 163
Eau Gallie; 175
Edison, Thomas; 152
Eglin AFB; 116-118
Eight Reale; 176
Ellenton; 144-145
Emanuel Point Wreck; 120
Emergency Callboxes; 83
Epiphytes; 142,148,157,159
Escambia Bay; 119
Bridge (I-10); 119
County; 120
Estero; 153
Everglade,90,95,139-140,154-160
Draining of; 156,181
Wildlife MA; 160
Wonder Gardens; 154
Falling Waters SP; 115
Fantasy of Flight; 95
Fayer Dykes SP; 171
Fires, Forest; 166
Fires, Prescribed ; 148
Fisherman's Village; 151
Flagler County; 171
Flagler, Henry; 97,165,167,171
Florida Aquarium; 186
Florida,
12,000 years ago; 187
Cavern SP; 114
Map of all Expressways; 2-3
Mus of Natural History; 134
National Cemetery ; 141
Part of Africa; 177
Platform; 187
Sheriff's Boys Camp; 126
Sports Hall of Fame; 130
Sun 'n Fun Museum; 97
Supreme Court; 107
Florida's Turnpike (FTP), 178,189
25 mile Strip Maps; 66
Administration; 189
Coin System; 190
Exit Services; 189

For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…

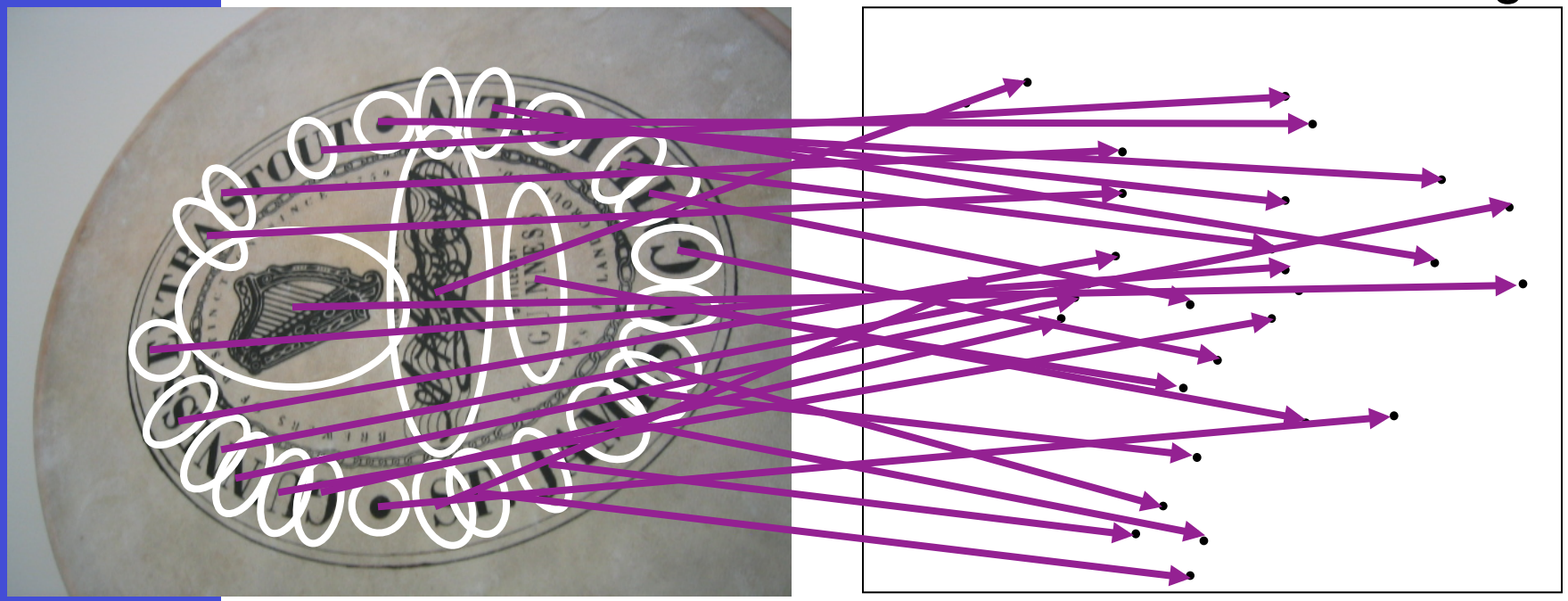
We want to find all *images* in which a *feature* occurs.

To use this idea, we'll need to map our features to "visual words".

Kristen Grauman

Visual words: main idea

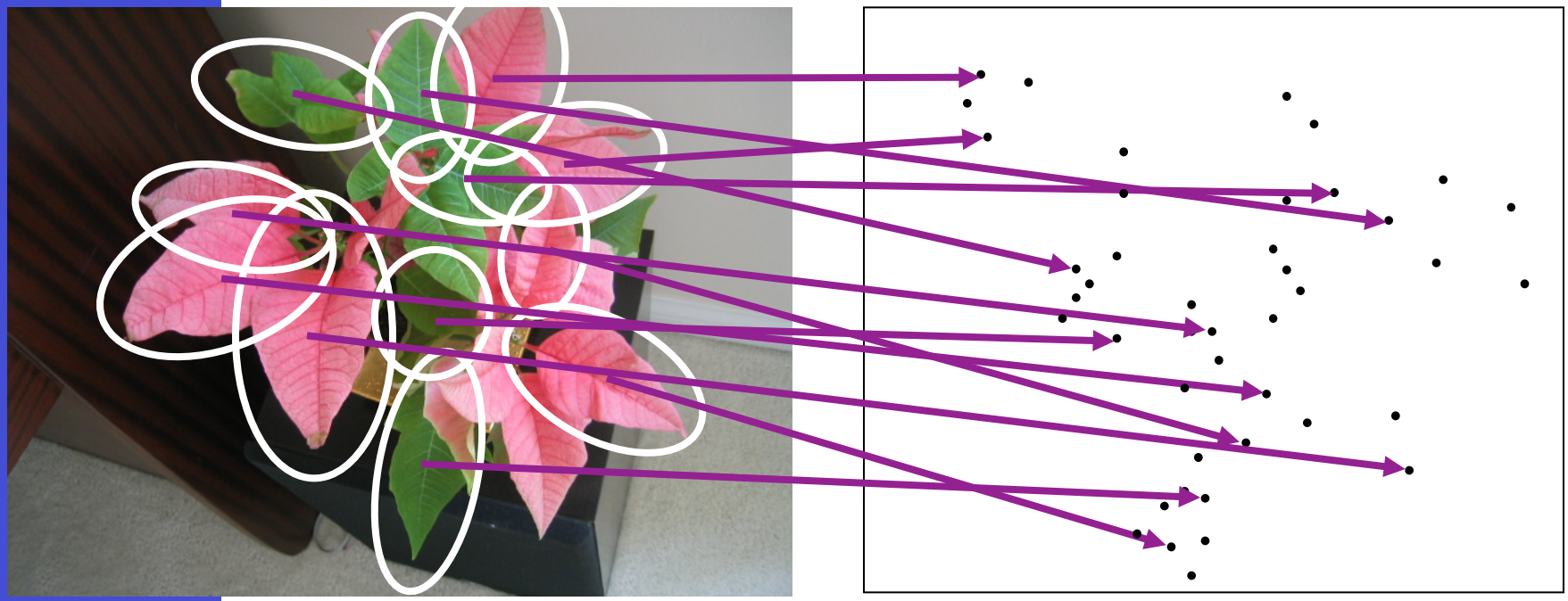
- Extract some local features from a number of images



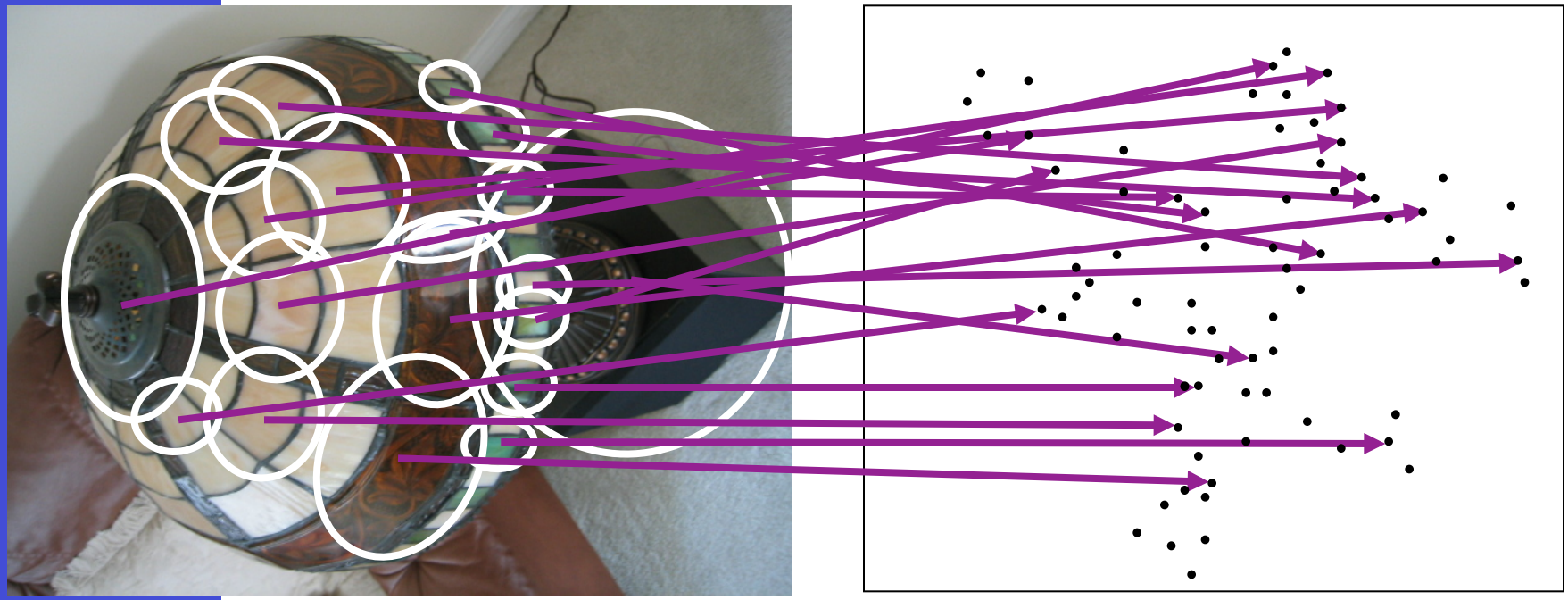
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

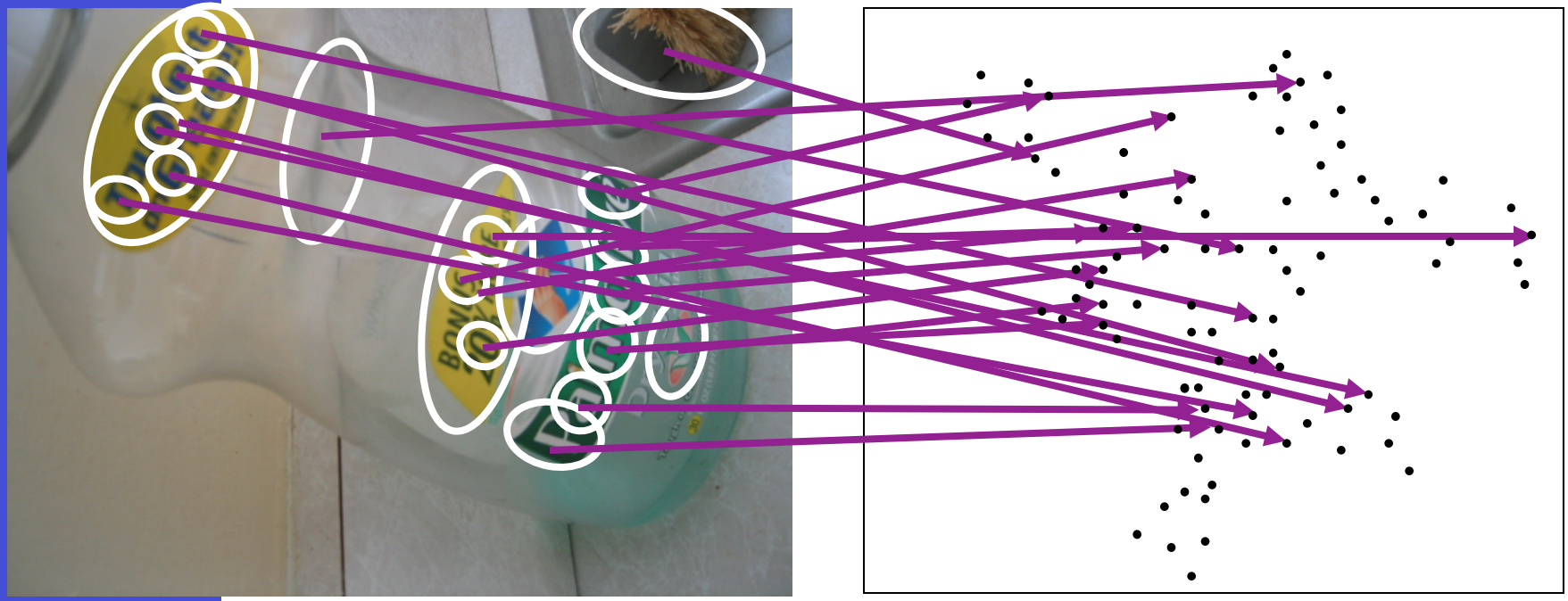
Visual words: main idea



Visual words: main idea



Visual words: main idea

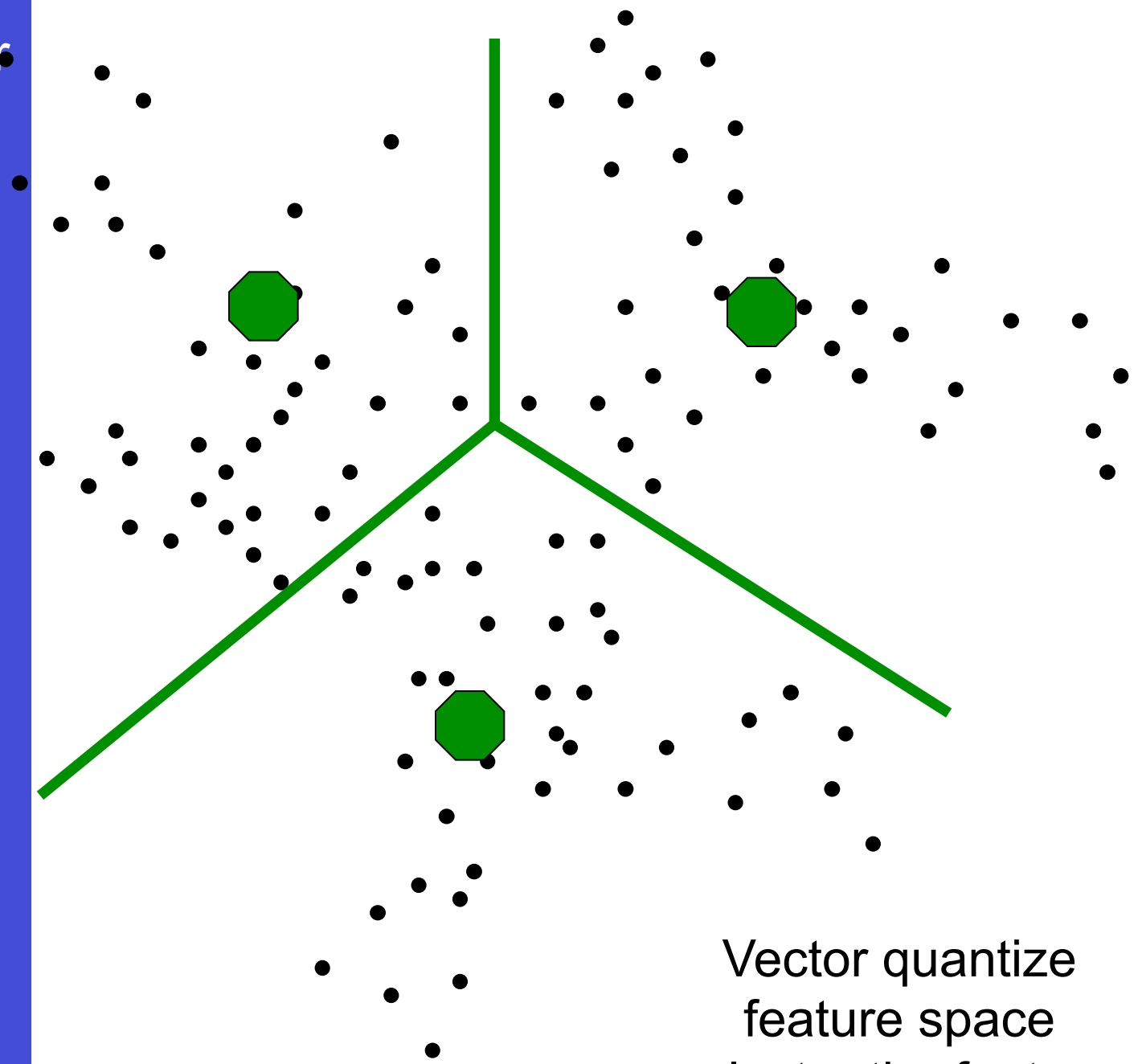


Computer Vision



Each point is a local descriptor, e.g. SIFT vector.

Computer Vision

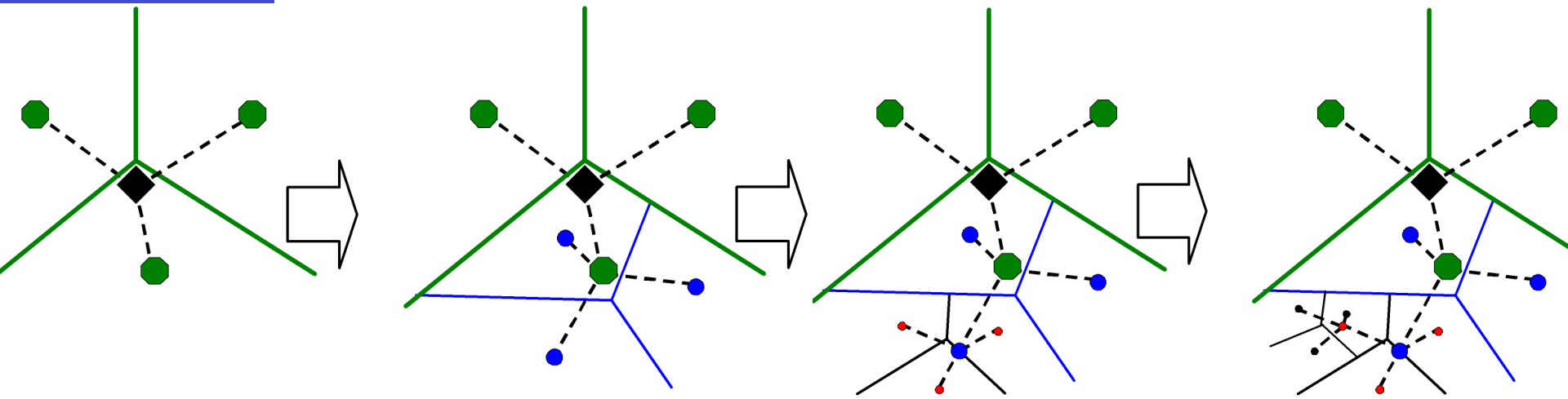


Vector quantize
feature space
= cluster the features

K-means clustering

1. randomly initialize K cluster centers
2. Assign each feature to nearest cluster center
3. Recompute cluster center (mean)
4. Iterate from 2, until convergence

Hierarchical K-means clustering

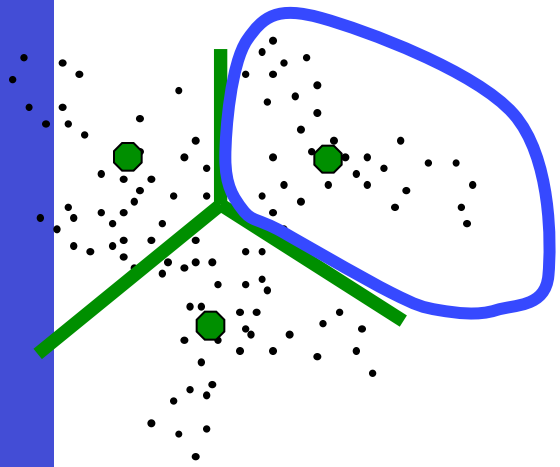


Allows to use larger vocabularies and thereby yields better results

In the example k=3, but typically it is chosen higher, e.g. k=10 and 6 layers could be used for search in about 1M images

Visual words

Ex: each group of patches belongs to same visual word



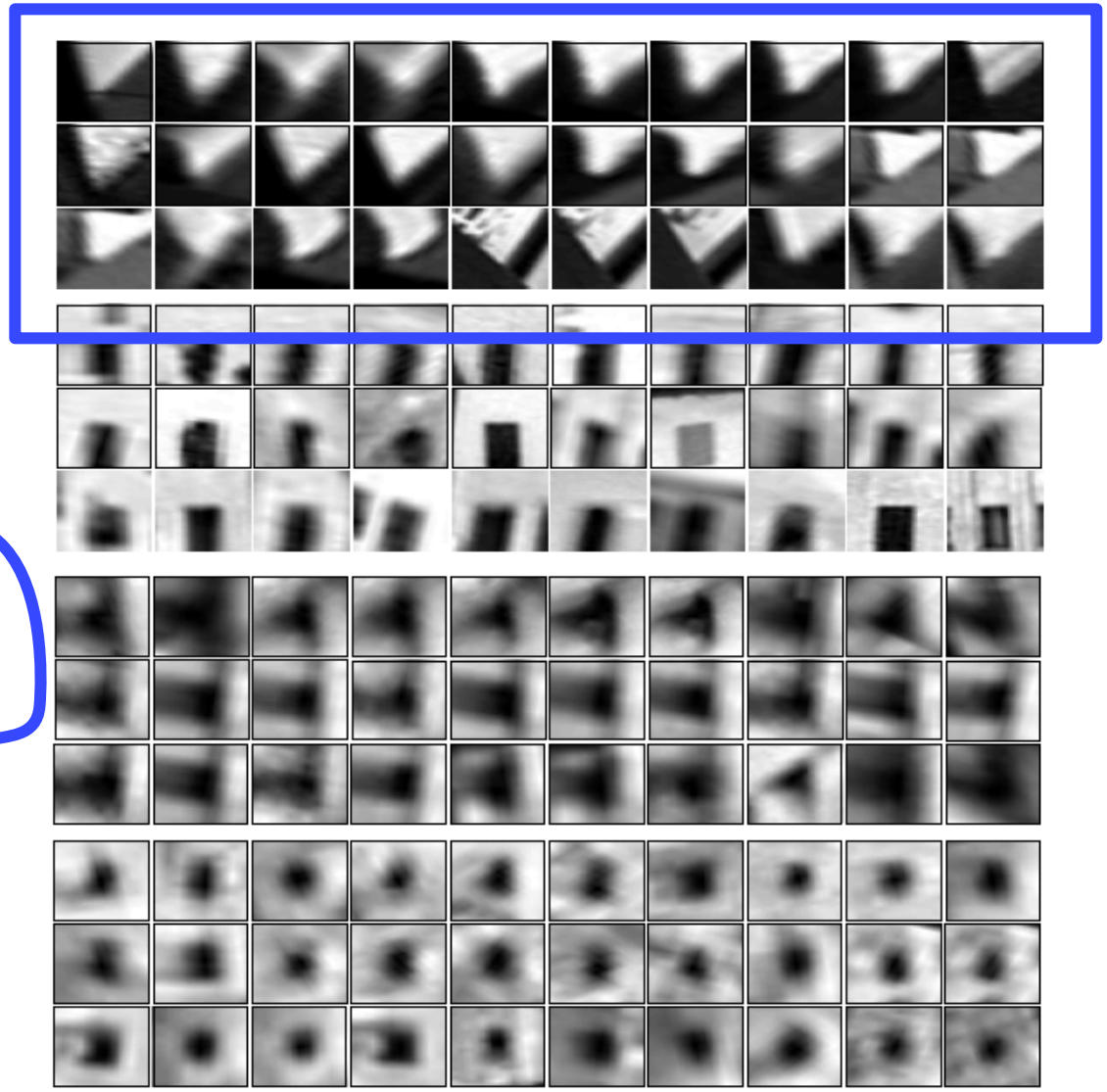


Figure from Sivic & Zisserman, ICCV 2003

Inverted file index



Image #1

Image #2

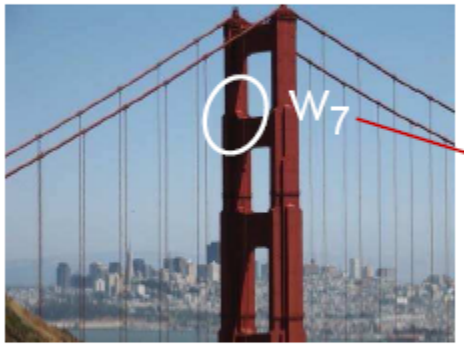
Image #3

Word #	Image #
1	3
2 …	
7	1, 2
8	3
9	
10 …	
91	2

…

Database images are loaded into the index mapping words to image numbers

Inverted file index



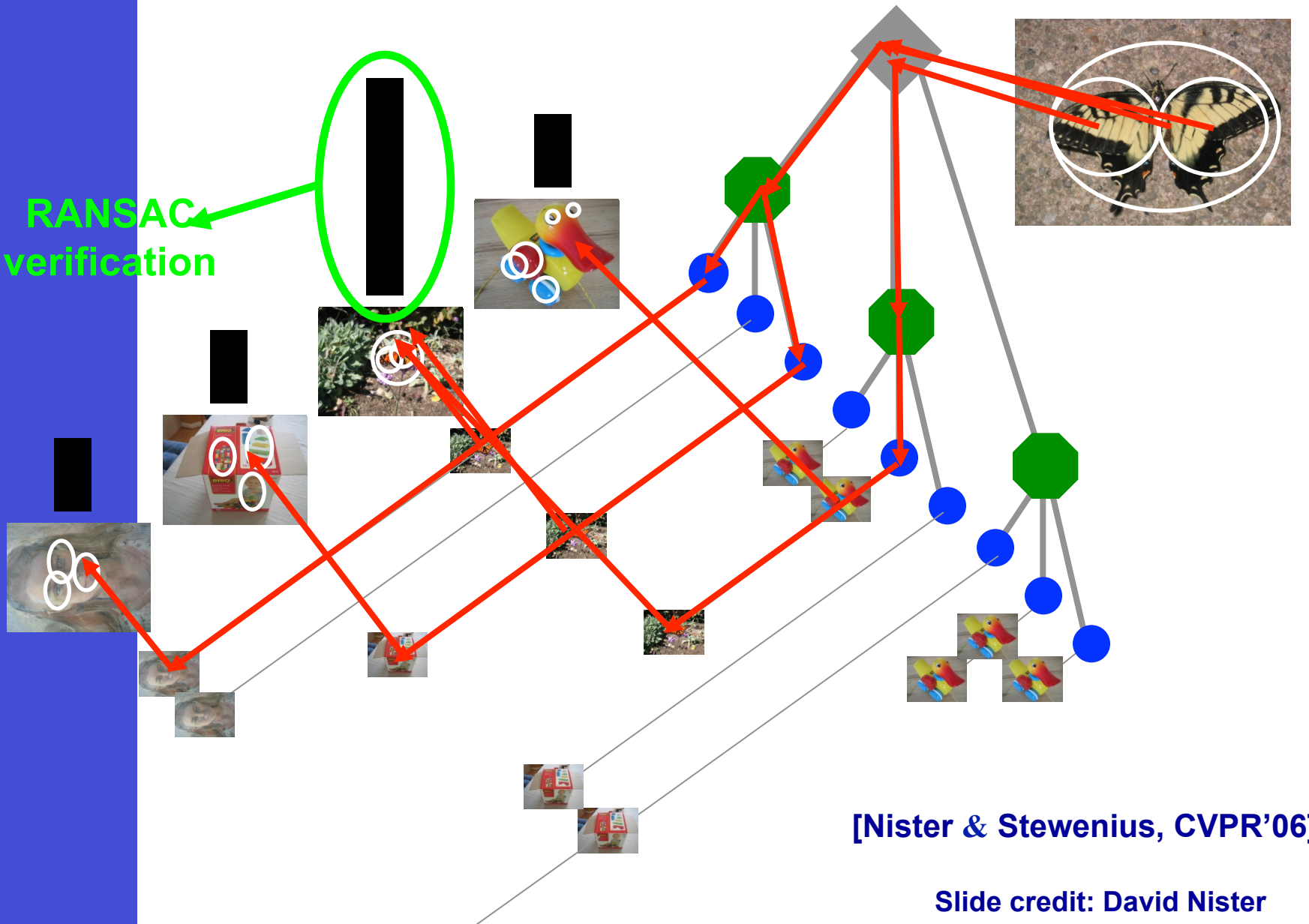
New query image

Word #	Image #
1	3
2	
…	
7	1, 2
8	3
9	
10	
…	
91	2

…

New query image is mapped to indices of database images that share a word.

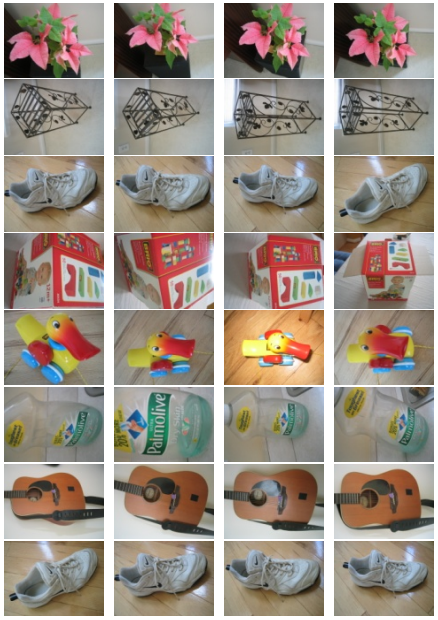
Retrieval with a Vocabulary Tree



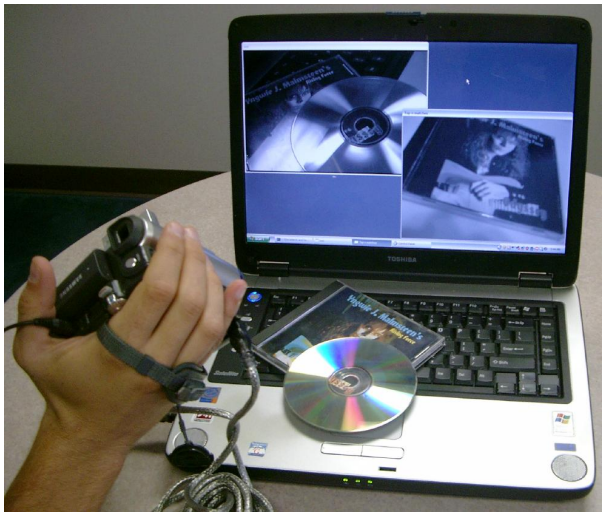
[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Performance



- Evaluated on large databases
  - Indexing with up to 1M images
- Online recognition for database of 50,000 CD covers
  - Retrieval in ~1s
- Best with very large visual vocabularies



NOTE: object class recognition typically done with smaller vocabularies

Supporting the matching step

1) Too slow if naively done

2) Will often fail when only based on descriptor matching

Supporting the matching step

Matching based on descriptors only is rather fragile.

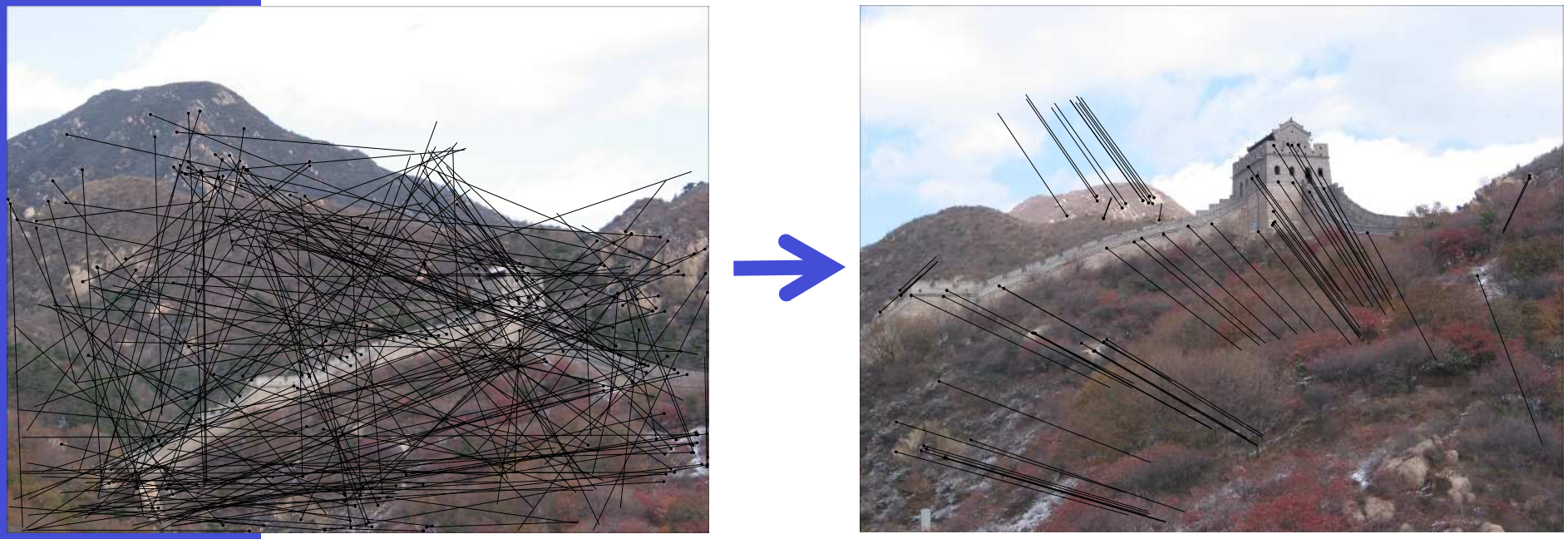
Typically, one needs to add a test on the configuration of the matches.

Epipolar geometry and projective matching are often used tests, using RANSAC to withstand unavoidable mismatches.

… but make strong assumptions about the scene; we discuss expansion-extraction as a robust but computationally costly alternative

Supporting the matching step

Ex. cleaning matches based on RANSAC-Ep.Geom.



[Chum, Werner, Matas]

Search for corresponding regions

1. Extract affine invariant regions
2. Describe region with feature vector of moment invariants
3. Search for corresponding regions based on Mahalanobis distance
4. Check cross-correlation (after normalization)
5. Check geometric consistency of correspondences

Semi-local constraints

= check consistency of correspondences

- Epipolar constraint (RANSAC) based on 7 points
- Projectivity constraint (RANSAC) based on 4 points

- Alternative: two affine regions rigidly moving:
- Geometric constraints
- Photometric constraints

} based on a combination of only 2 regions

$$\det\begin{bmatrix} a_{23}-b_{23} & b_{13}-a_{13} & a_{13}b_{23}-b_{13}a_{23} \\ a_{22}-b_{22} & b_{12}-a_{12} & a_{12}b_{23}-b_{13}a_{22}+a_{13}b_{22}-b_{12}a_{23} \\ a_{21}-b_{21} & b_{11}-a_{11} & a_{11}b_{23}-b_{13}a_{21}+a_{13}b_{21}-b_{11}a_{23} \end{bmatrix} \leq \delta$$

Supporting the matching step

Thus, matching based on descriptors only is rather fragile, and, typically, one needs to add a test on the configuration of the matches.

Epipolar geometry and projective matching are often used tests, using RANSAC to withstand unavoidable mismatches.

… but make strong assumptions about the scene; we discuss expansion-extraction as a robust but computationally costly alternative

Supporting the matching step

… and RANSAC does typically not withstand more than 30% of outliers; our initial matches may be too noisy to satisfy this condition

Here we describe a system, which in fact combines recognition with segmentation, and uses TOPOLOGICAL FILTERING instead of RANSAC

What comes first ?

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



?



?

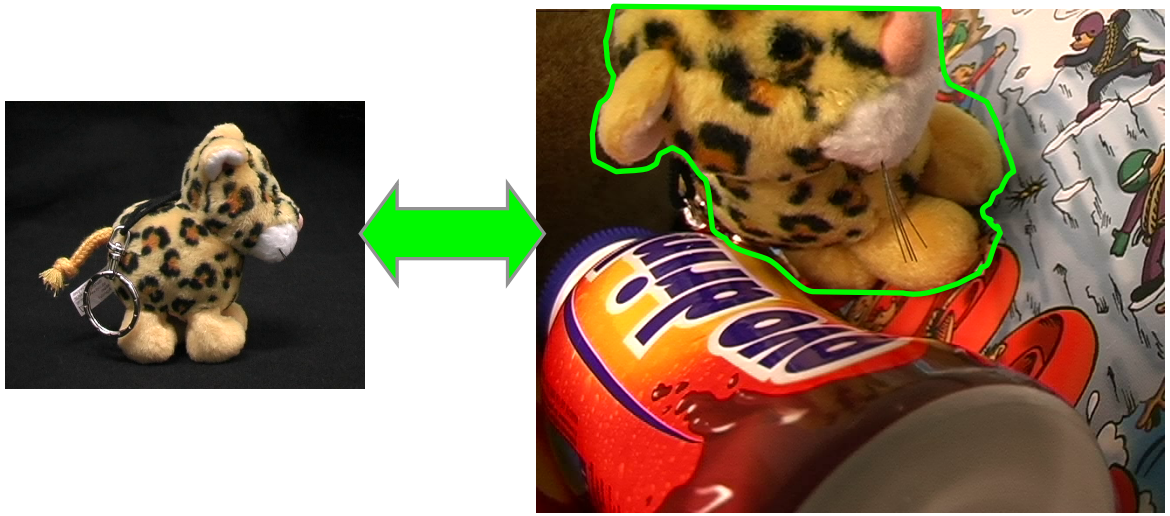


?



What comes first ?

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

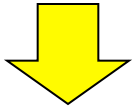


Recap: affine invariant regions





automatically adapt shape to viewpoint



allow to find correspondences between two *different* views

Computer Vision

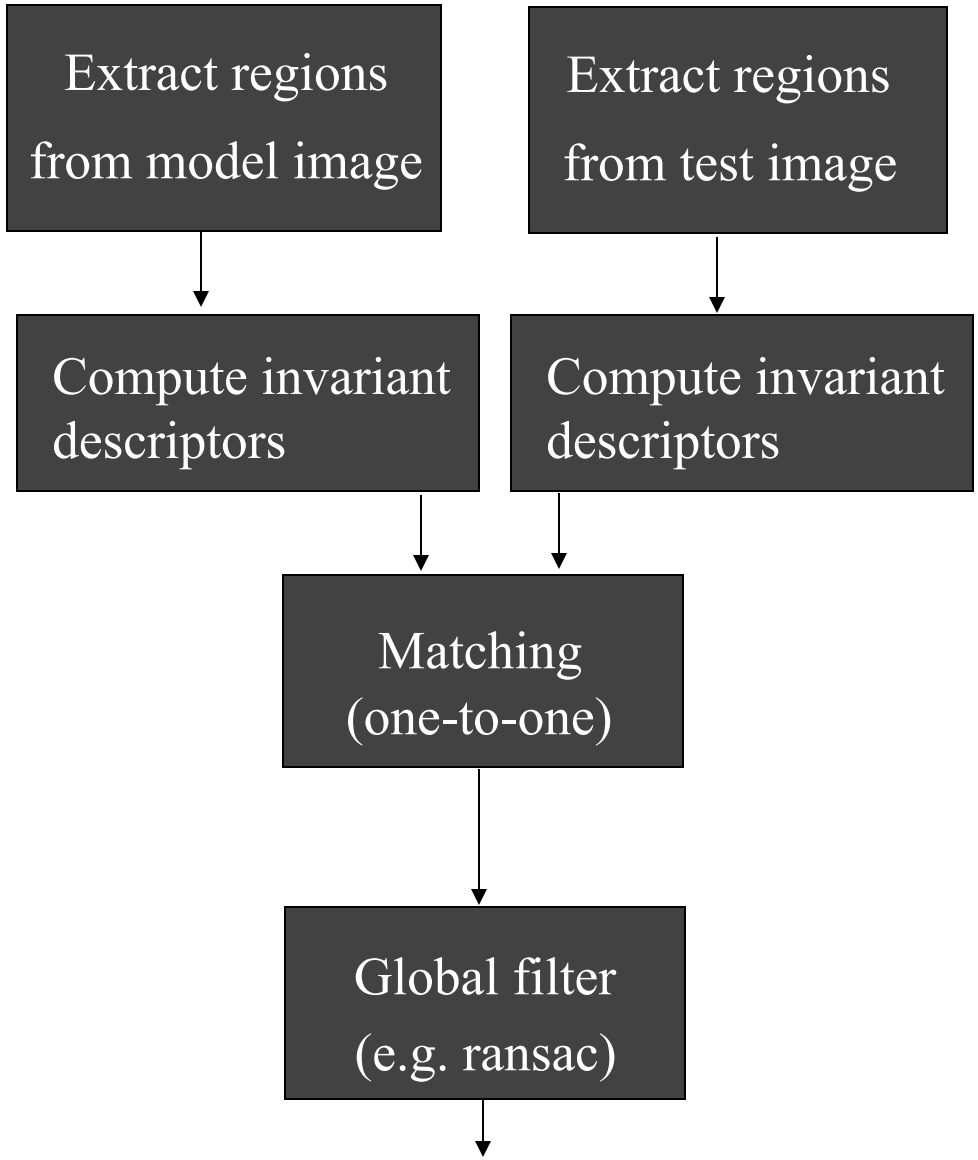
Usual approach with *affine invariant regions*



model image



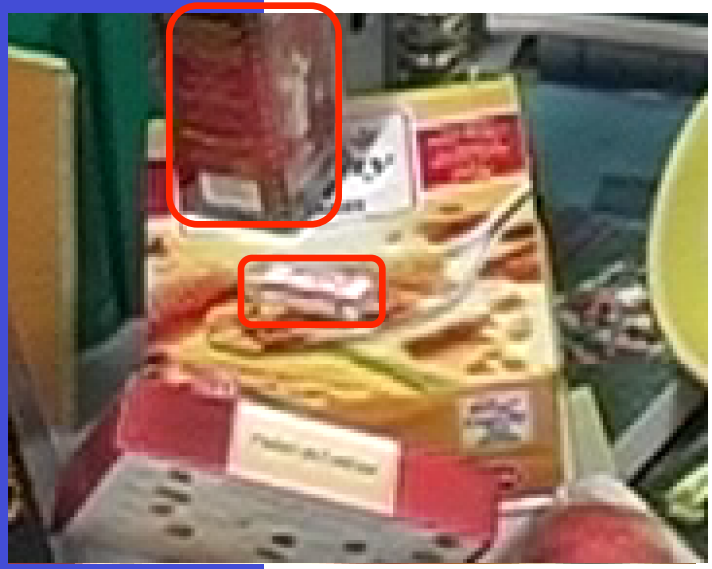
test image



Computer Vision

Why difficult ?





Large scale change

Modest resolution (720x576)

Loss of information

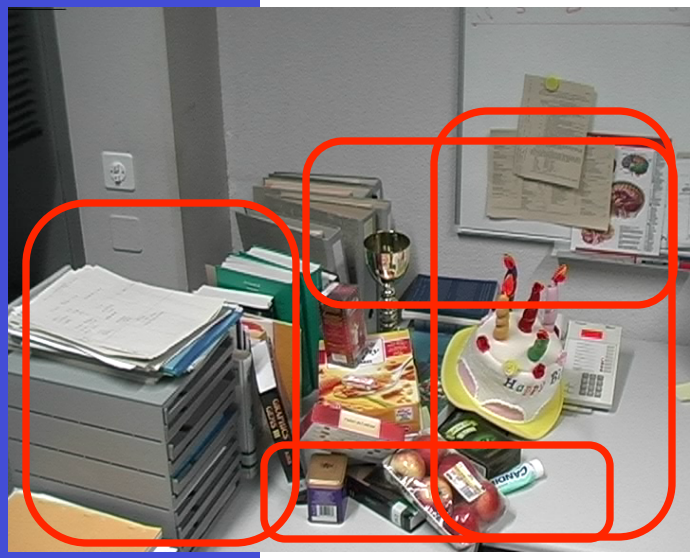
Occlusion

Few *repeated* object regions, less accurate shape

Computer Vision

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

Computer Vision

New approach: image exploration

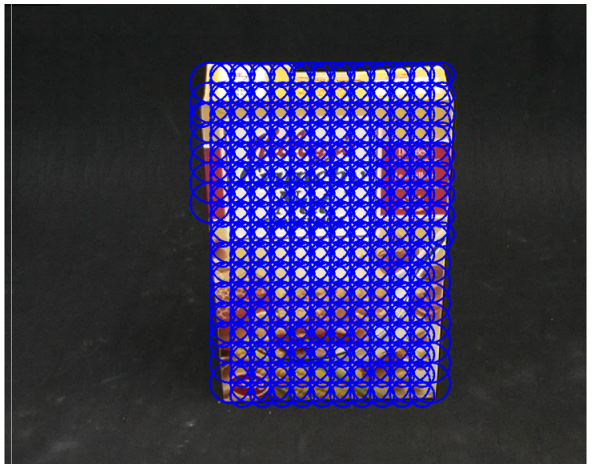




1. Soft-matching
   (one-to-many:
   max number of correct matches)

   ⟶ 3/217

2. Model coverage



Computer Vision

New approach: image exploration





1. Soft-matching
   (one-to-many:
   max nmb ok matches)

   → 3/217

2. Model coverage

3. Expansion
   (*construct* new matches)

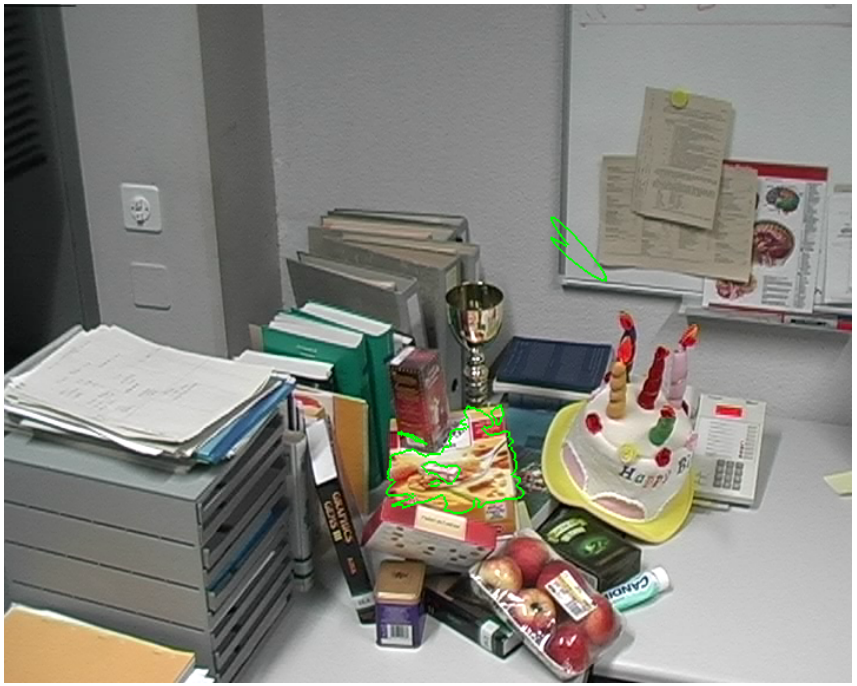
4. Contraction
   (remove matches)

→ **202/220**

Computer Vision

New approach: image exploration





Expansion and Contraction phases help each other !

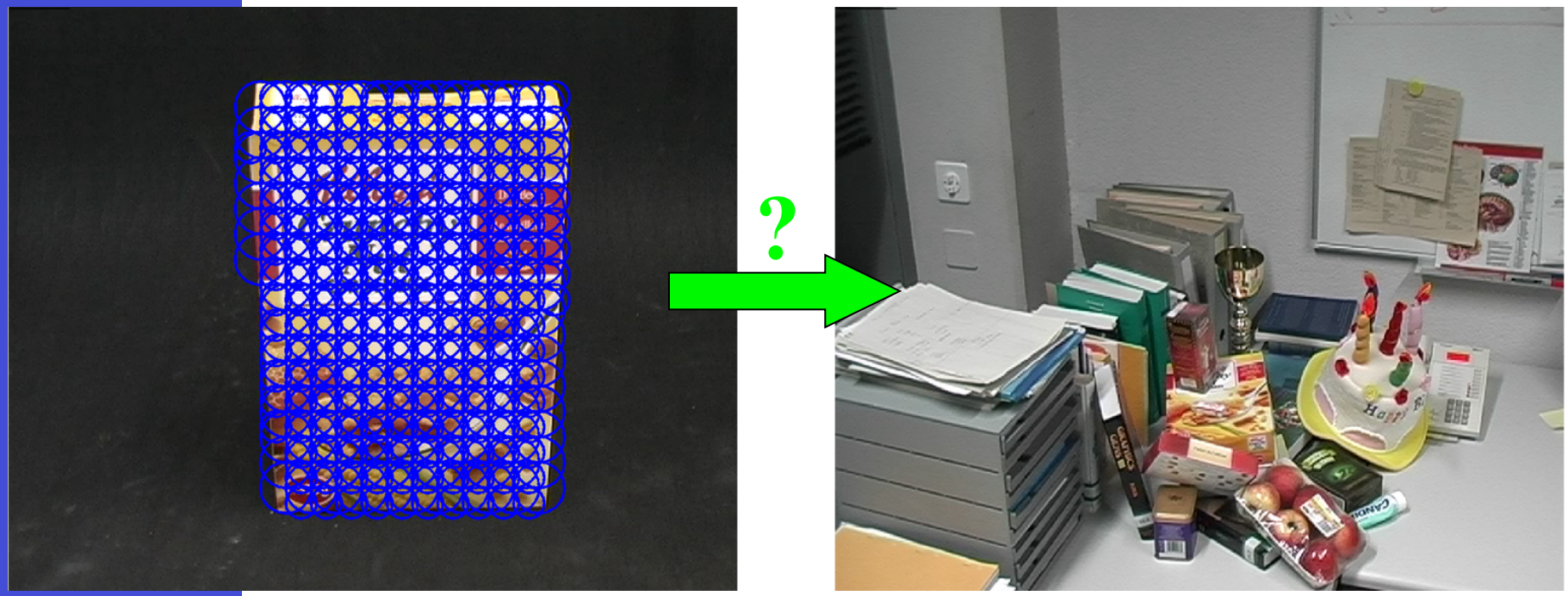
Effect: gradually ***explore*** image around initial matches



+ *Power*: very few initial correct matches suffice

+ *Functionality*: cover with matches ⟶ approximate

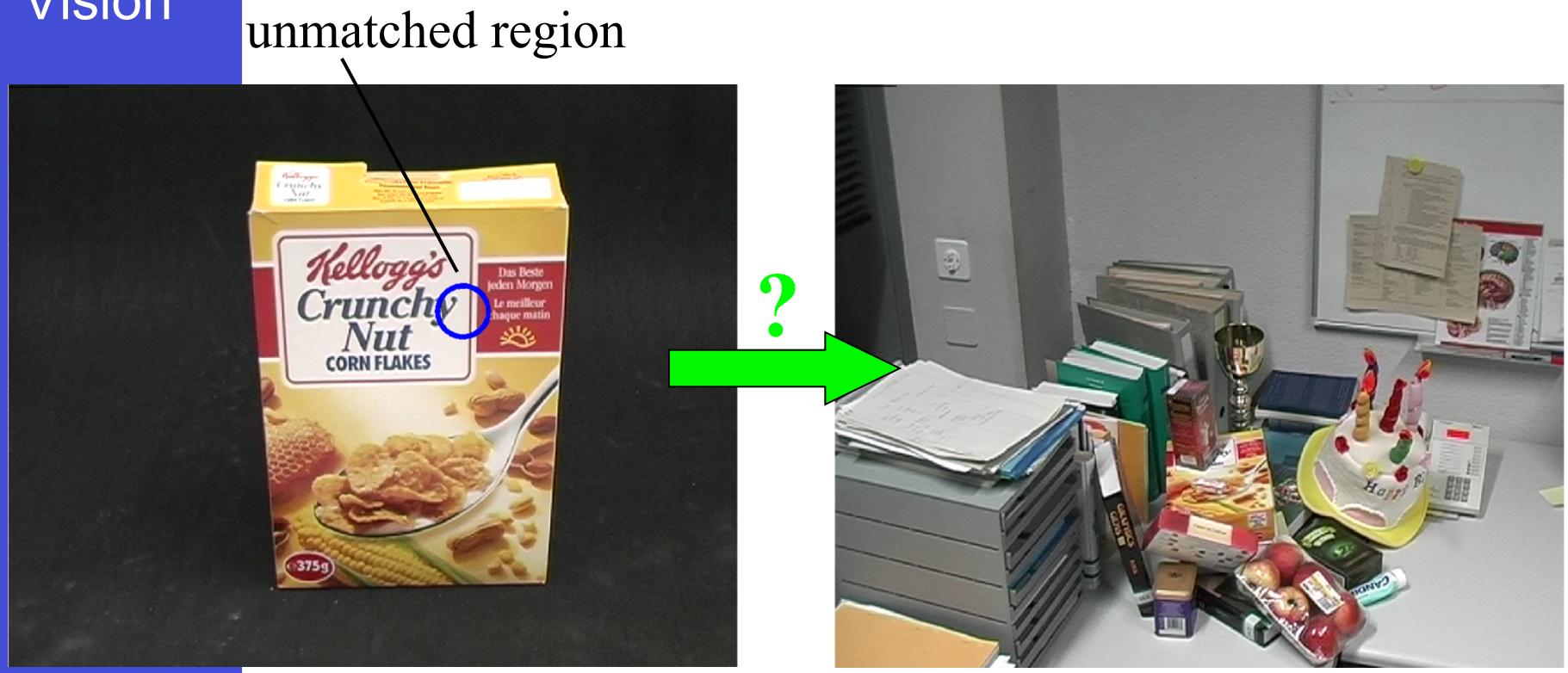
Expansion



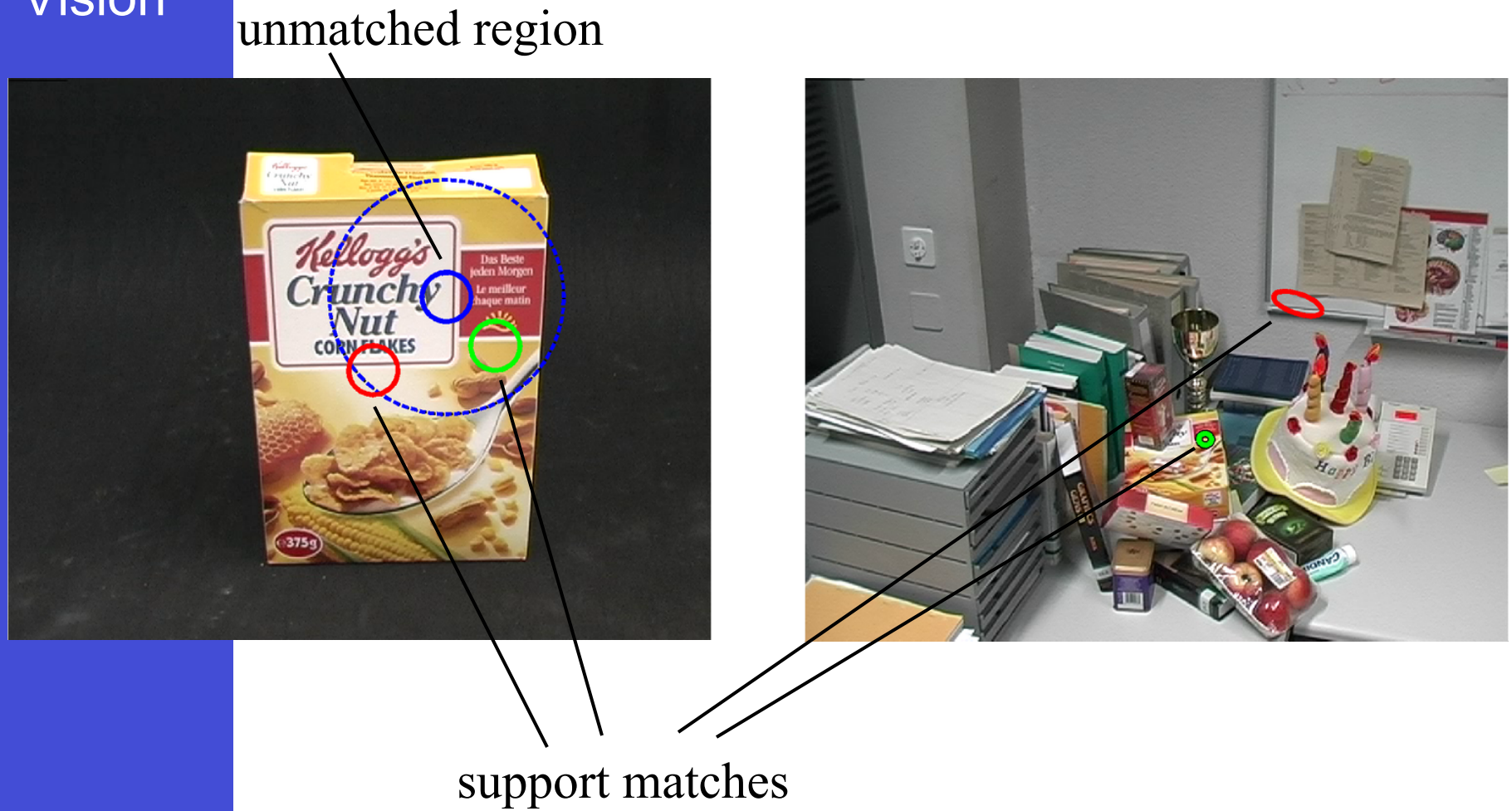
Coverage regions

Goal: construct corresponding regions for the coverage ones

Expansion


unmatched region
Kellogg's
Crunchy Nut
CORN FLAKES
Das Beste jeden Morgen
Le meilleur chaque matin
375g
?

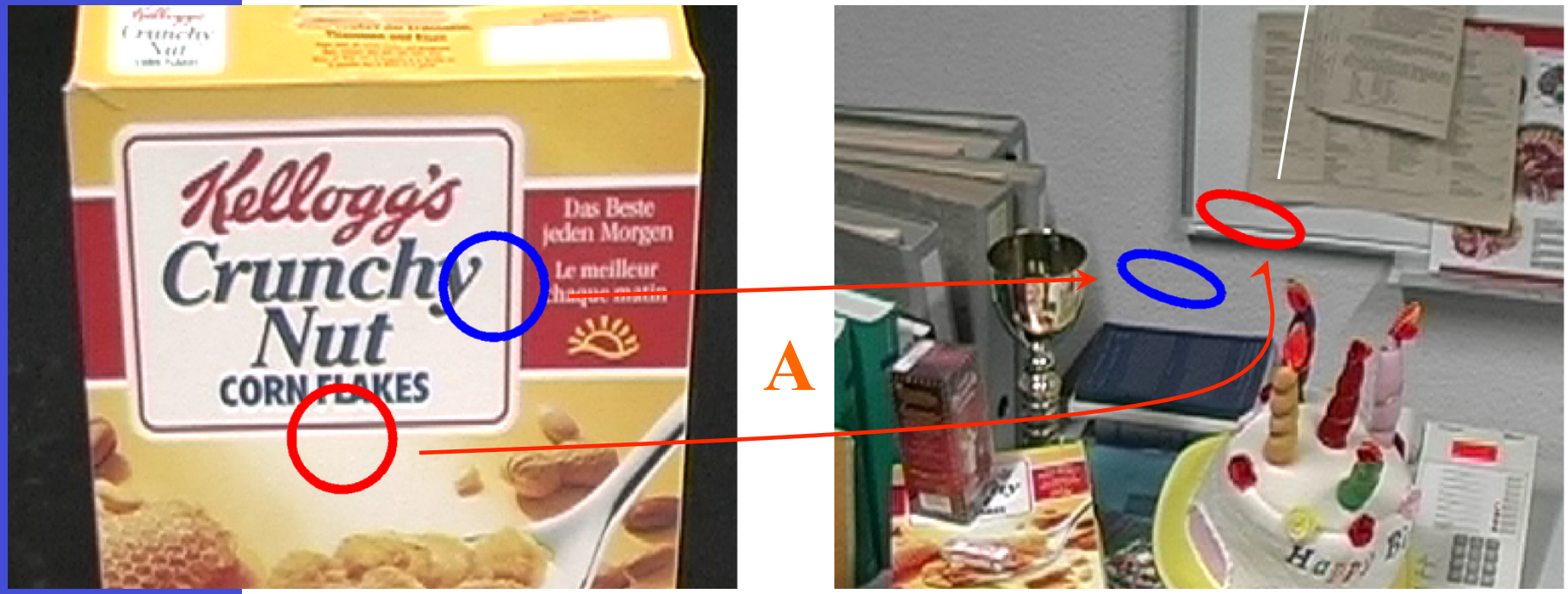
Expansion



Idea: project via affine transformation of a support

Expansion

propagation attempt
(low similarity)



the mismatched support yields a poor propagation attempt

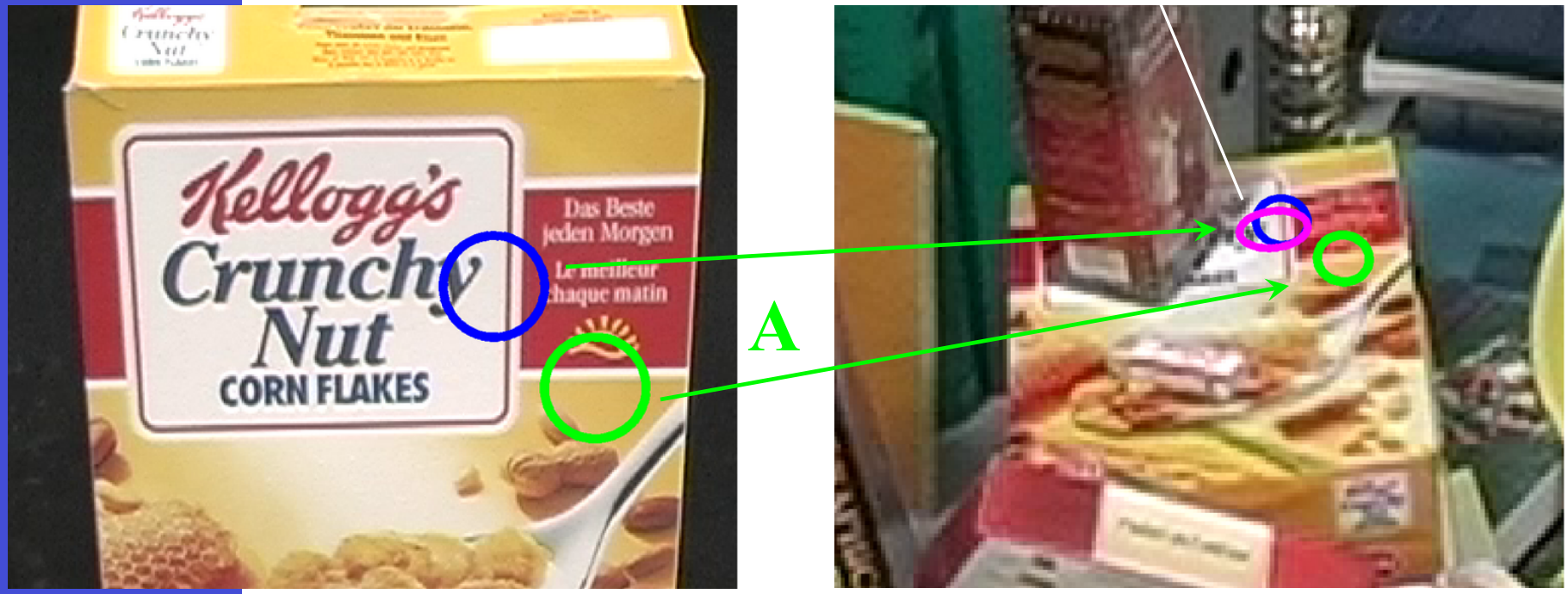
Expansion



the correct support yields good propagation attempt

→ **Winner**

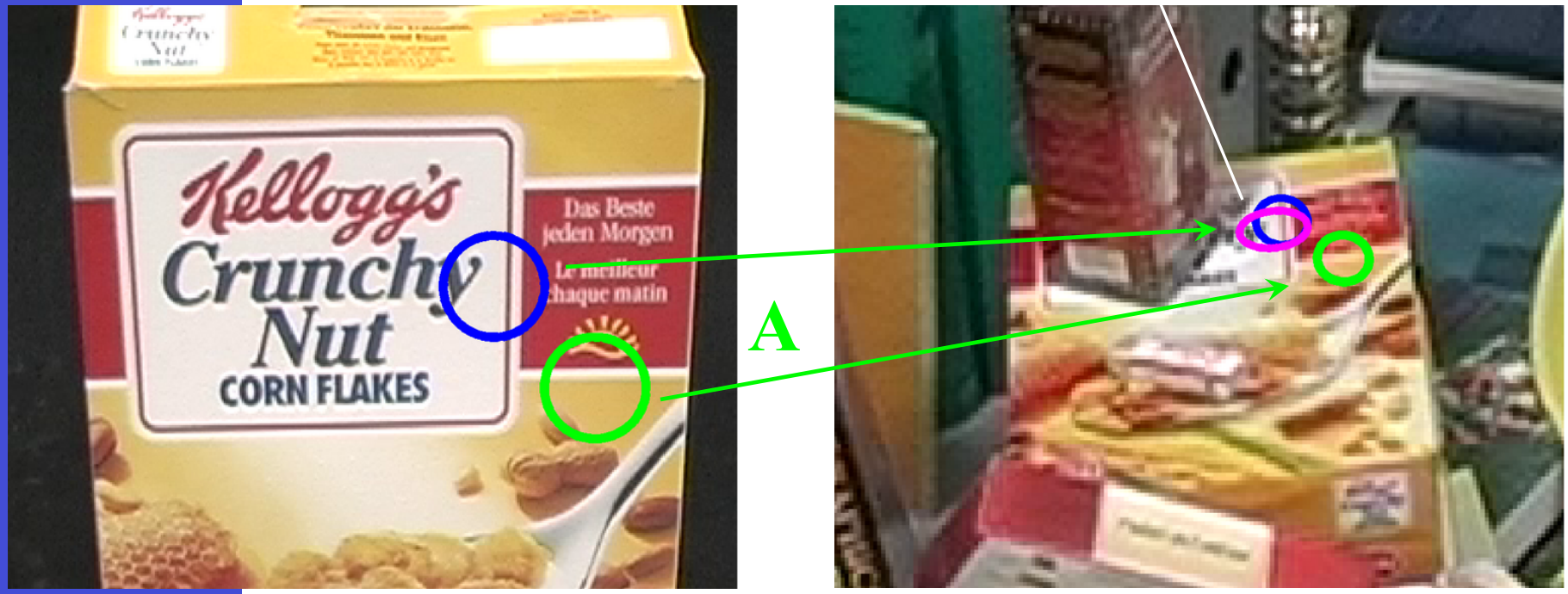
Expansion



the best attempt is refined

→ adapts to perspective effects, curved surfaces & deformations

Expansion

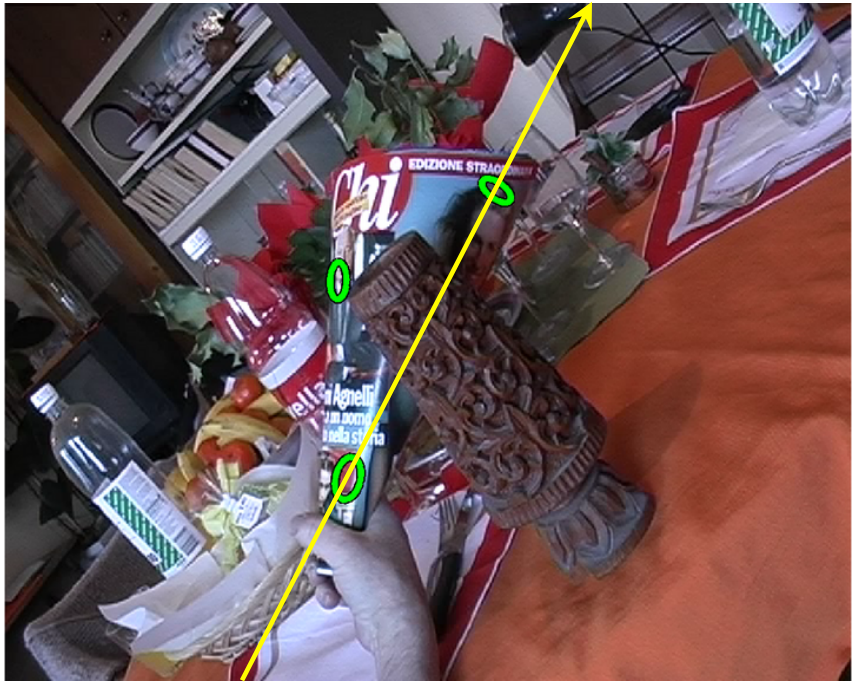


The higher the percentage of correct matches,

the better it works !

Contraction



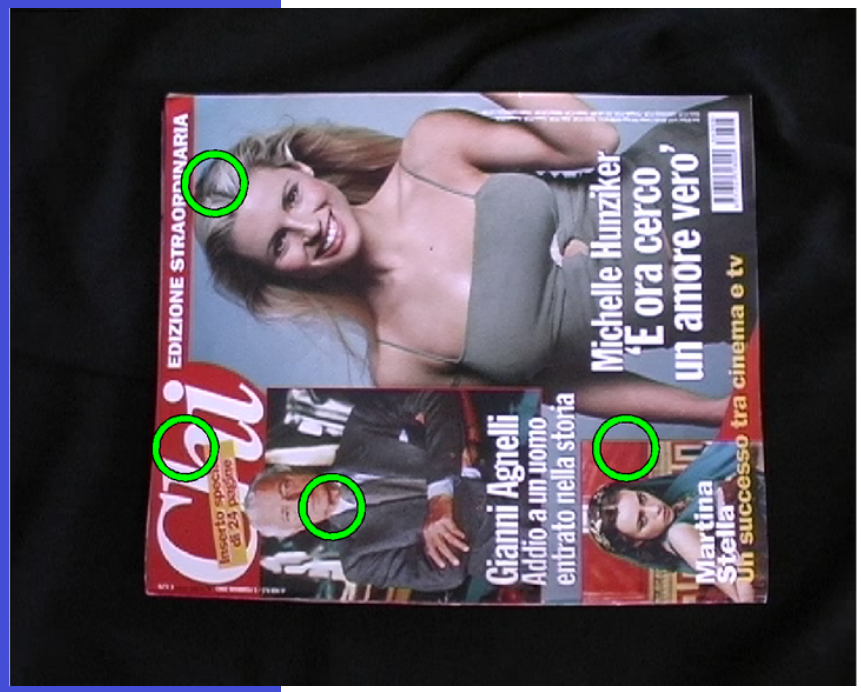


Sidedness constraint

check it for all triples

$\longrightarrow$ mismatches are involved in more violating triples !

Contraction

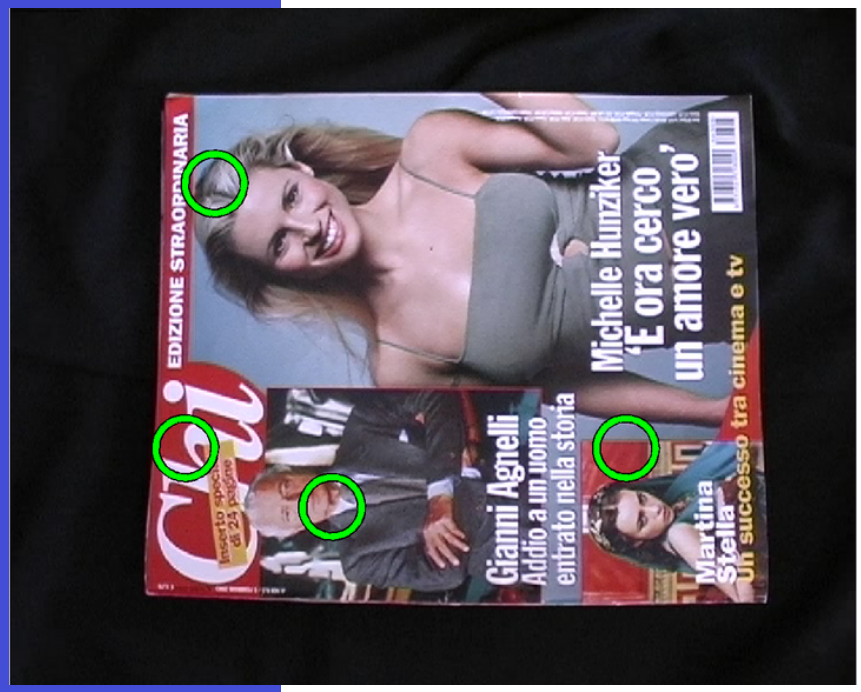




Algorithm

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

Contraction





+ allows for non-rigid deformations

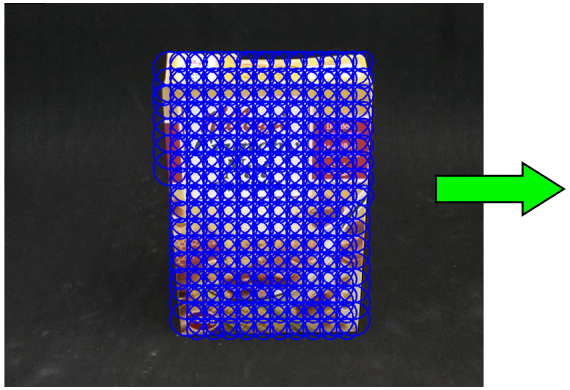
better decisions with higher percentage of correct matches

Exploring the image





expansion



Computer Vision

Exploring the image





expansion

→ higher percentage → helps next contraction

Computer Vision

Exploring the image





contraction

→ higher percentage → helps next expansion

Computer Vision

Exploring the image





expansion

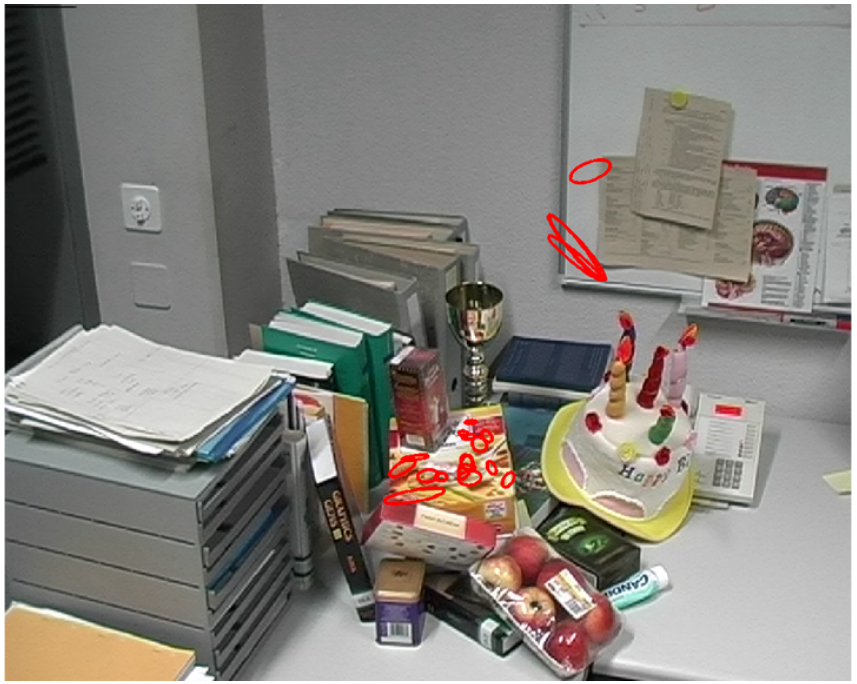
expansion and contraction *help each other*

→ progressive increase of — amount / percentage / extent

Computer Vision

Exploring the image





contraction

expansion and contraction *help each other*

→ progressive increase of — amount / percentage / extent

Computer Vision

Exploring the image





expansion

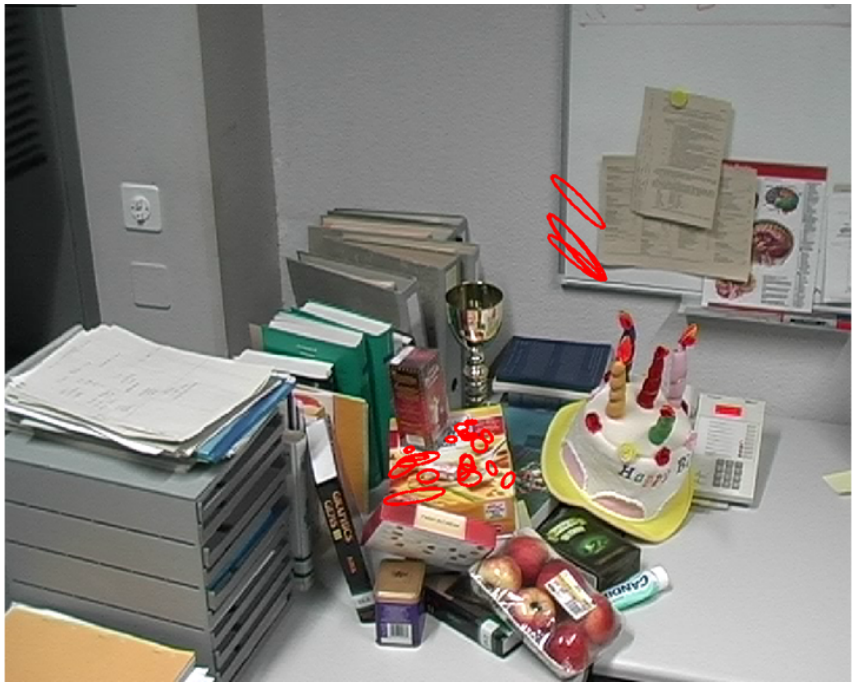
expansion and contraction *help each other*

→ progressive increase of — amount / percentage / extent

Computer Vision

Exploring the image





contraction

expansion and contraction *help each other*

→ progressive increase of — amount / percentage / extent

Computer Vision

Exploring the image





final

expansion and contraction *help each other*

→ progressive increase of — amount / percentage / extent

Exploring the image





+ *Power*: single correct match per smooth surface suffices
+ *Functionality*: approximate segmentation

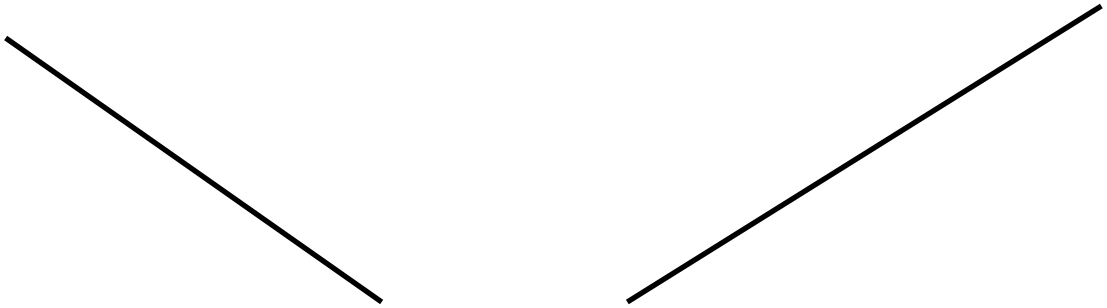
Simultaneous Recognition and Segmentation

Recognition

identify correct matches

Segmentation

cover object with matches



two aspects of the
same process

Computer Vision

Results: model objects (planar)







1 model view each

Results: model objects (curved)



6 model views

Computer Vision

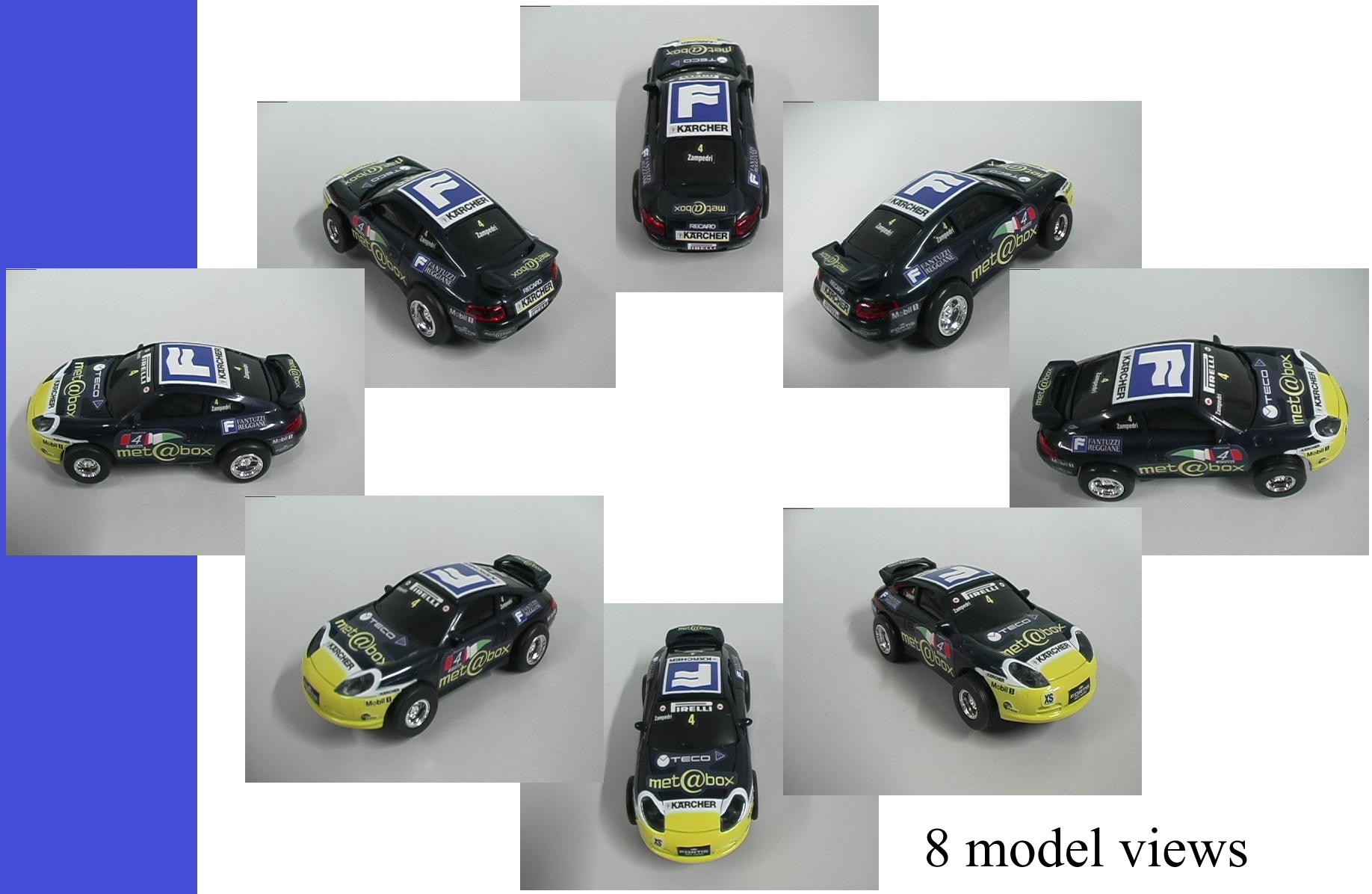
Results: model objects (curved)



6 model views

Computer Vision

Results: model objects (3D)



8 model views

Computer Vision

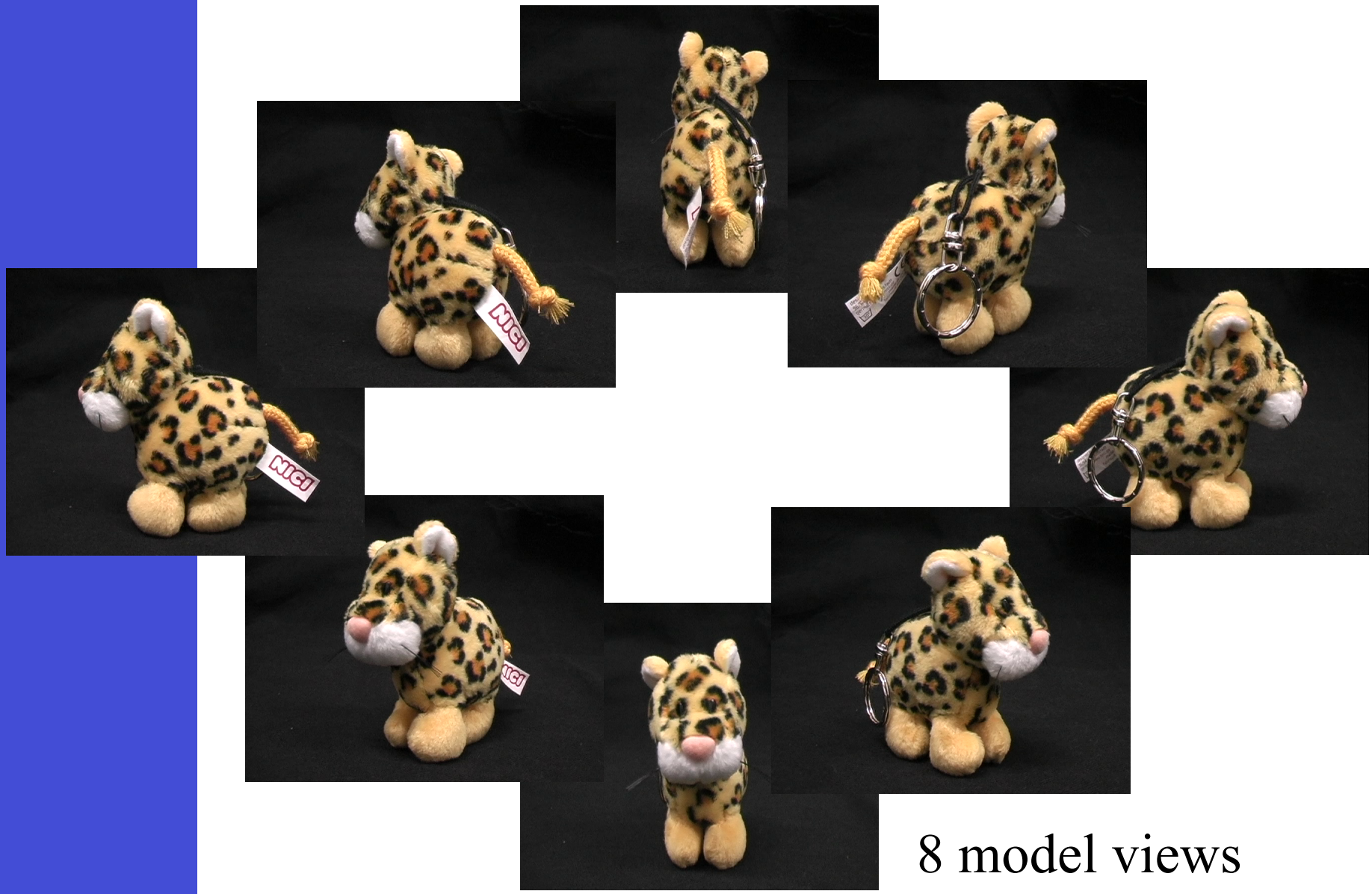
Results: model objects (3D)



8 model views

Computer Vision

Results: model objects (3D)



8 model views

Results: model objects (3D)



1 model view

Results





closest model view

occlusion, clutter, scale

Computer Vision

Results



Zoom 2



Zoom 2

occlusion, clutter, scale

Computer Vision

Results

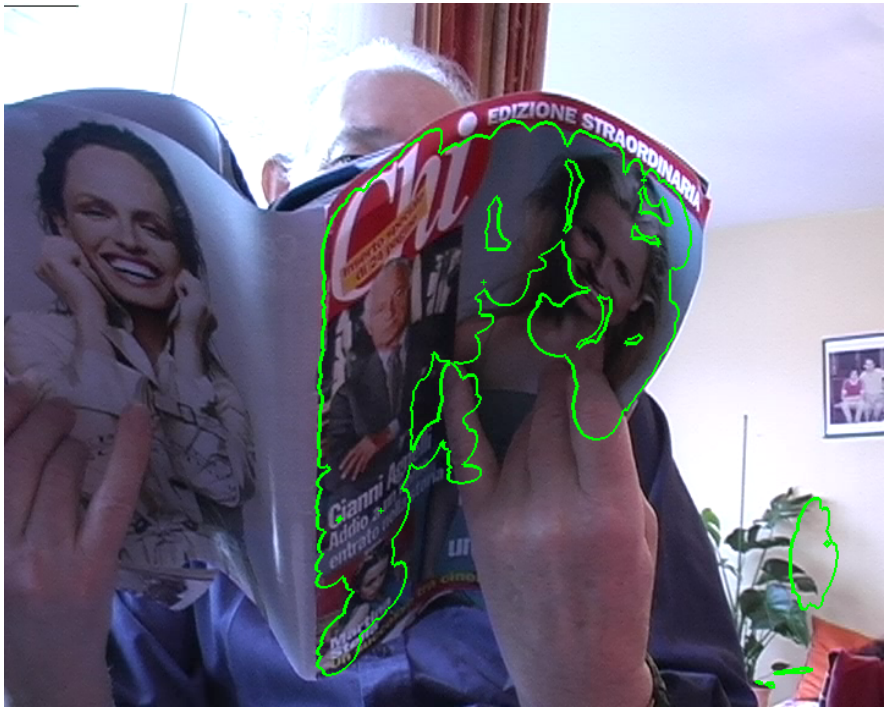




extensive clutter, scale, occlusion

Results





Folded, darker, occluded

Computer Vision

Results



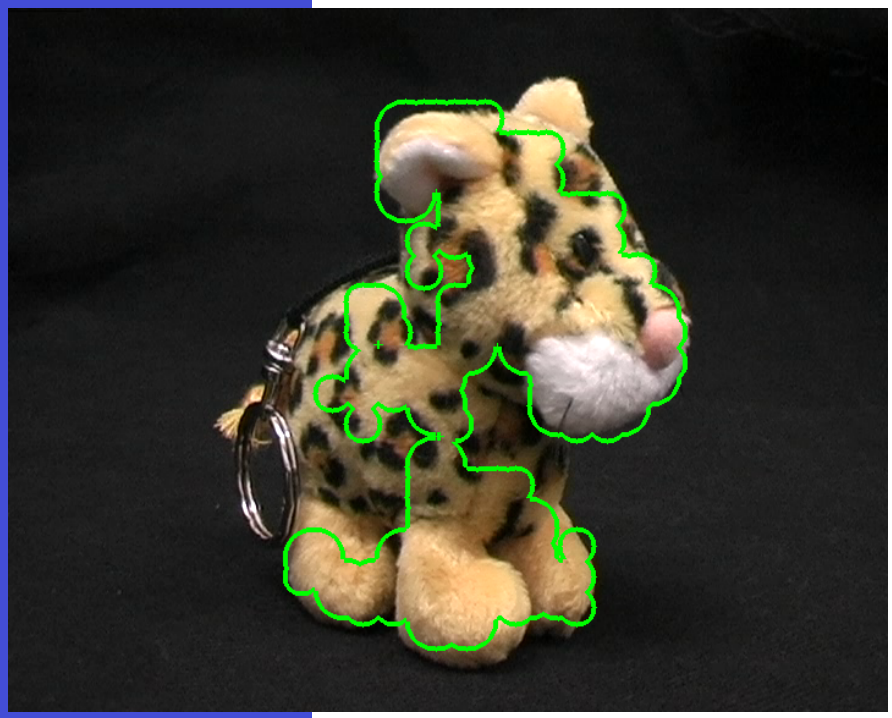
(whole object)



(total coverage)

Viewpoint change, clutter

Results





closest model view

Viewpoint change, clutter

Results



(whole object)



(total coverage)

Viewpoint change, clutter

Computer Vision

Results





closest model view

scale change, heavy occlusion (89%)

Results





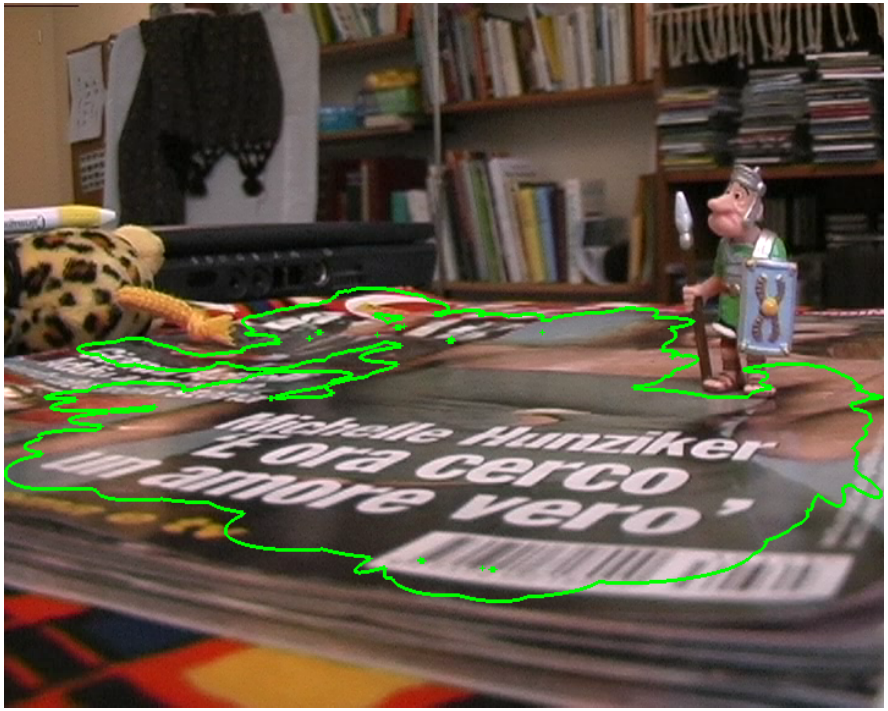
zoom 2x

Large scale change (3x), extensive clutter

Computer Vision

Results





very large out-plane-rotation

Computer Vision

Results



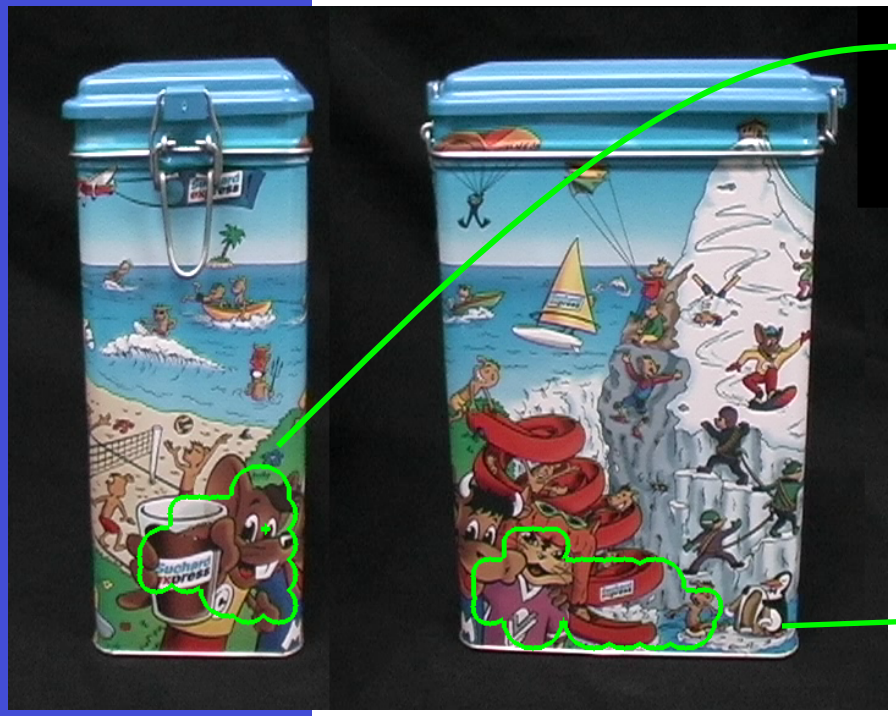
closest model views



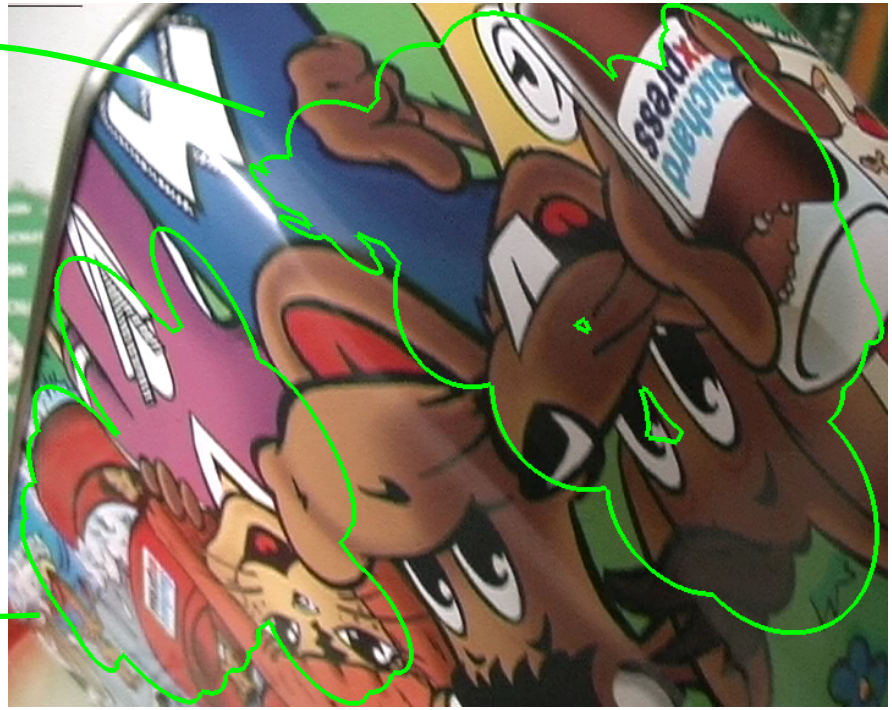
some 'mis-blobs' (CVPR04)

large viewpoint change, scale, clutter

Results



two matching views



(total coverage)

Small visible portion, scale change 4x

Computer Vision

Results



(whole object)



(total coverage)

Extensive clutter, scale, occlusion, blur

Computer Vision

Results



(whole object)



(total coverage)

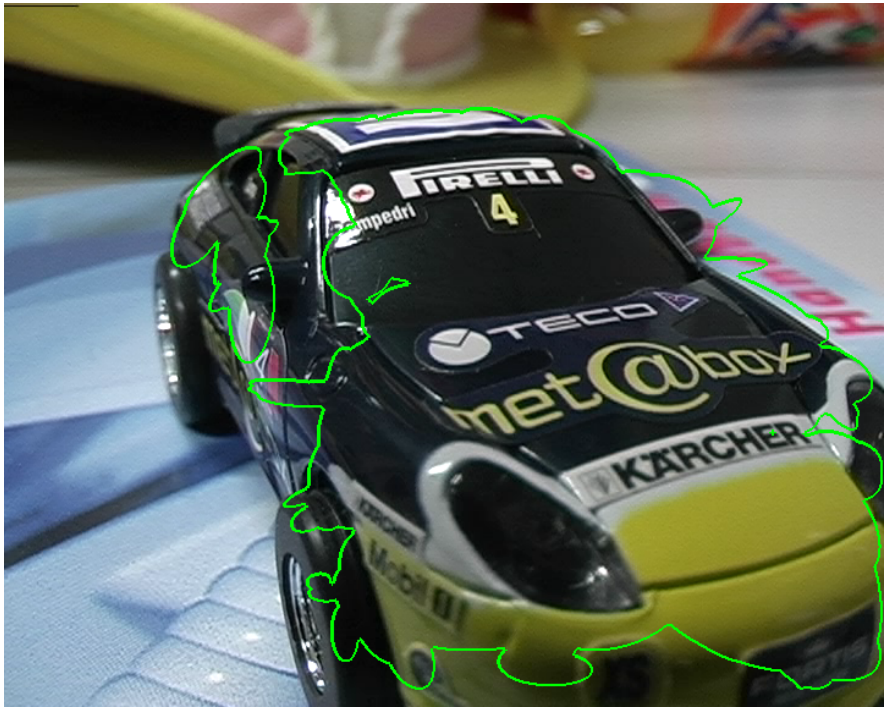
Divided by occlusion, viewpoint change

Computer Vision

Results



(whole object)



(total coverage)

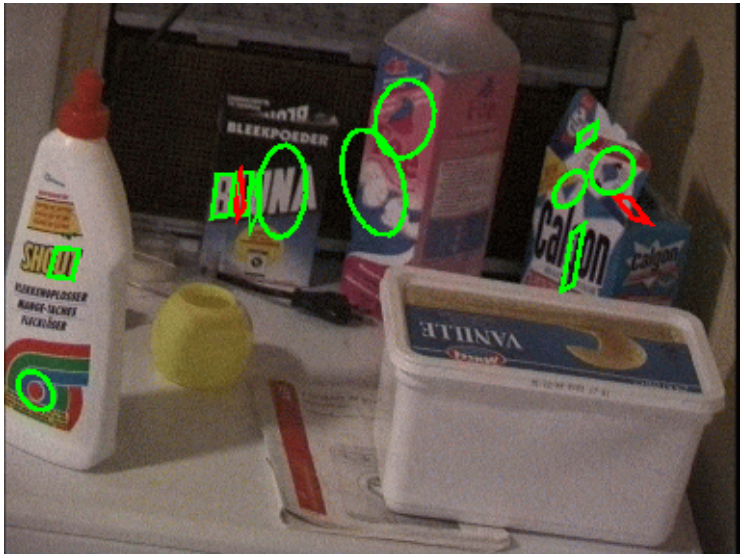
Viewpoint change

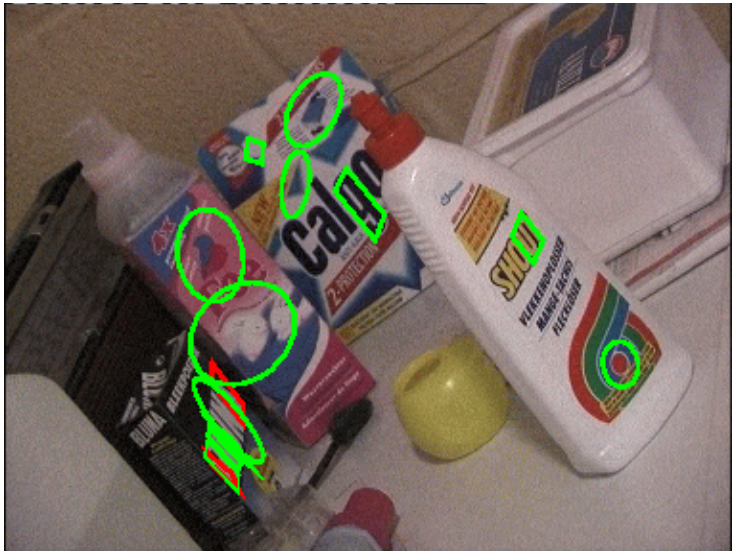
Other applications of invariant neighbourhoods

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

Computer Vision

Wide-baseline stereo





Computer Vision

Wide-baseline stereo



