SAIR

Spatial AI & Robotics Lab

CSE 473/573-A

W10: RETRIEVAL & DETECTION

Chen Wang

Spatial AI & Robotics Lab

Department of Computer Science and Engineering

University at Buffalo The State University of New York

SAIR

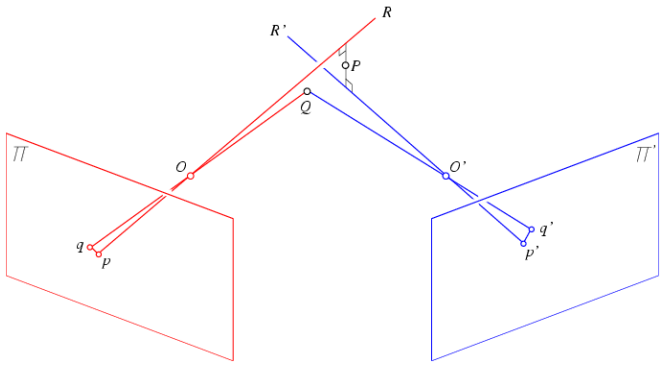
Spatial AI & Robotics Lab

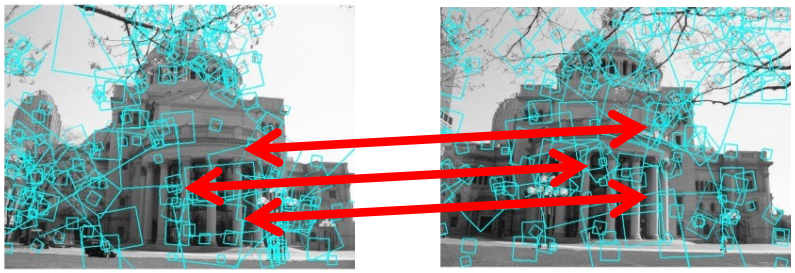
COMPUTER VISION

Instance Retrieval

Multi-view matching

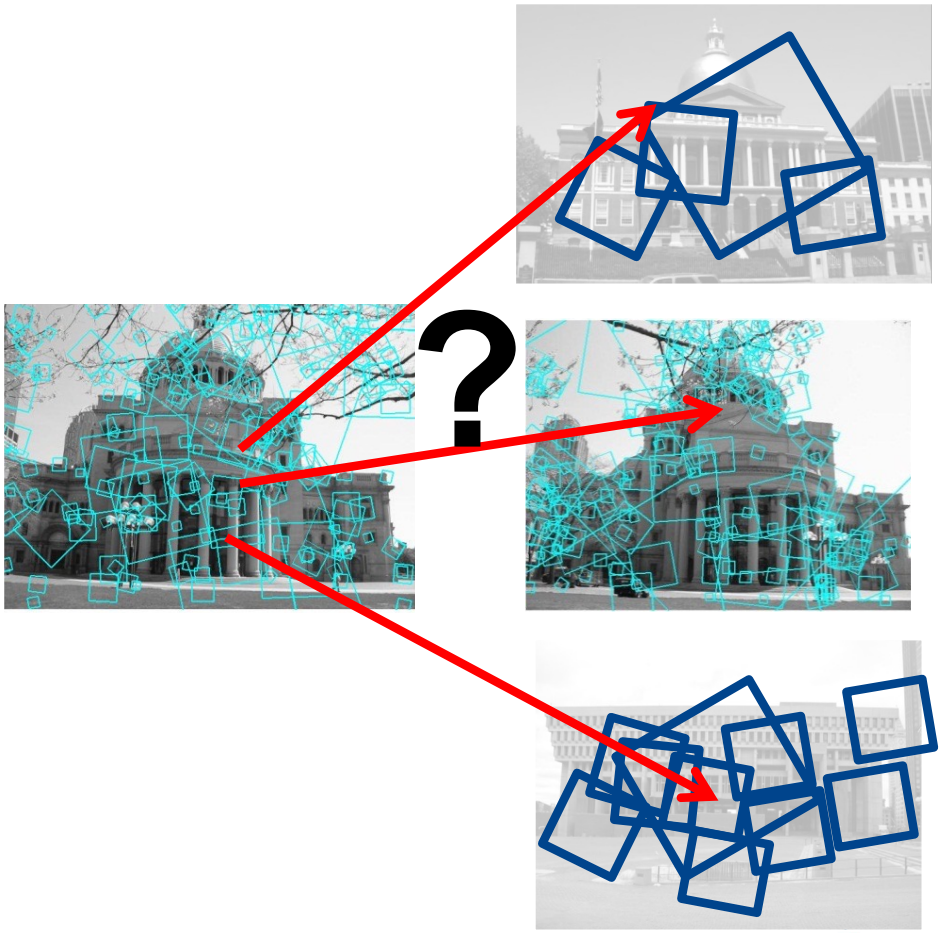
Matching two given views for depth





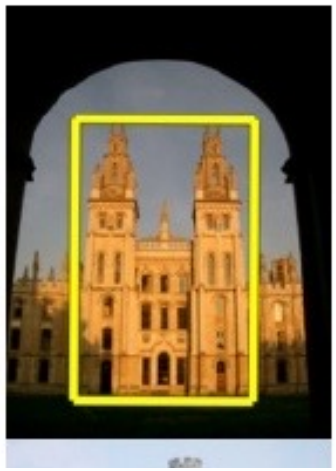
vs

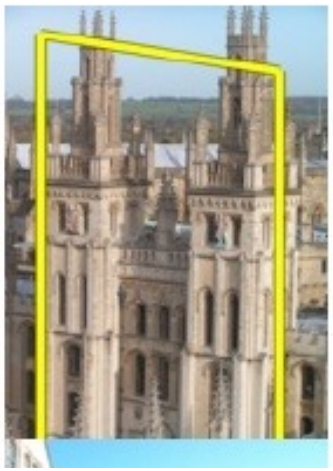
Search for a matching view for recognition

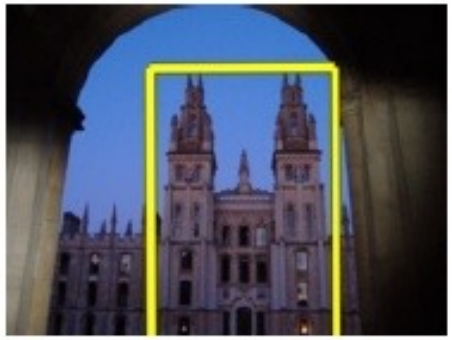


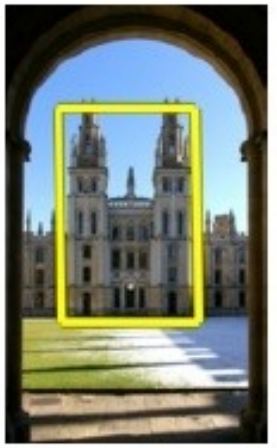
Efficient Retrieval

How to quickly find images in a large database that match a given image region?









Local Retrieval

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

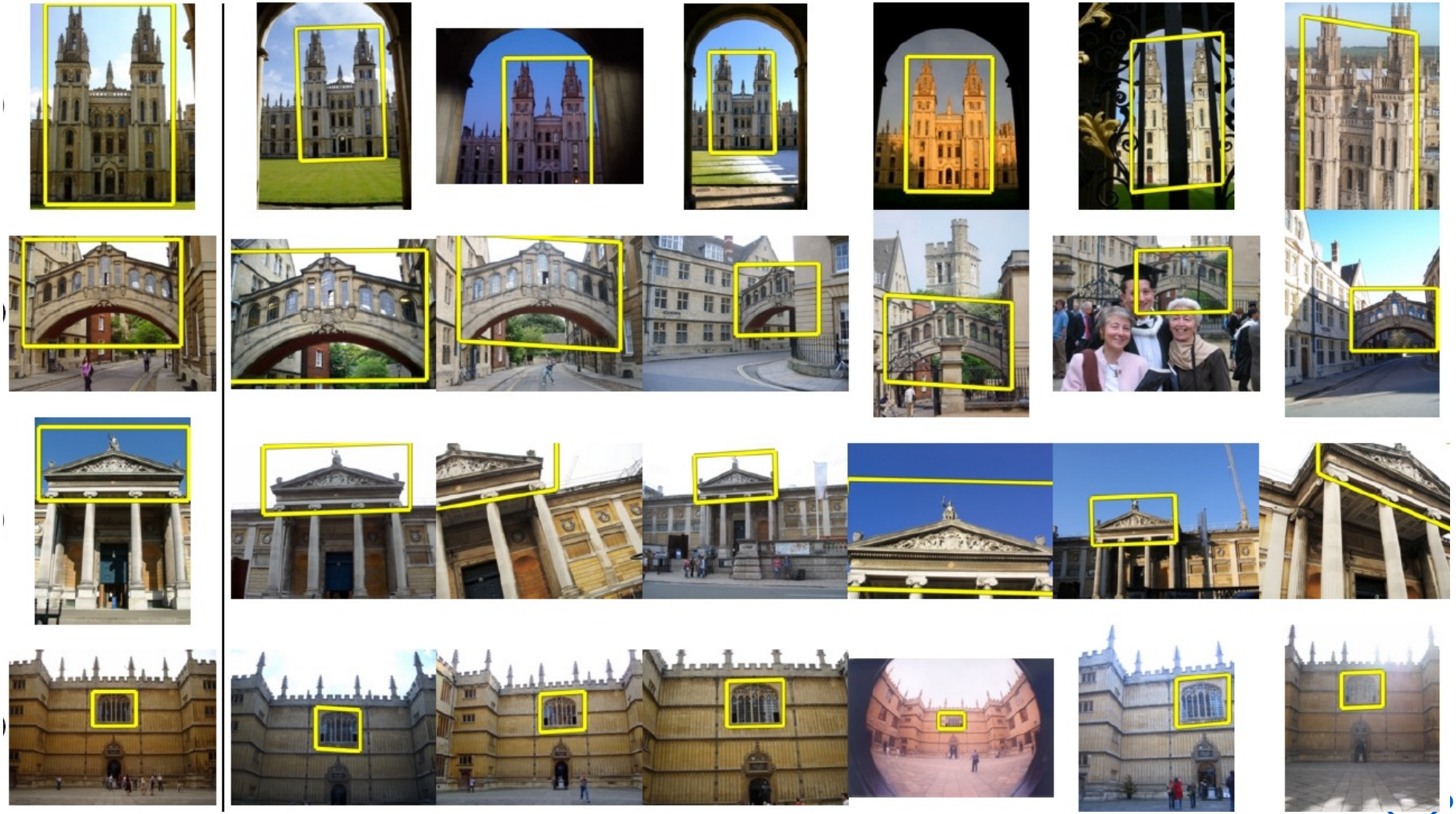
Query region

Retrieved frames

Application: Image Retrieval

Query

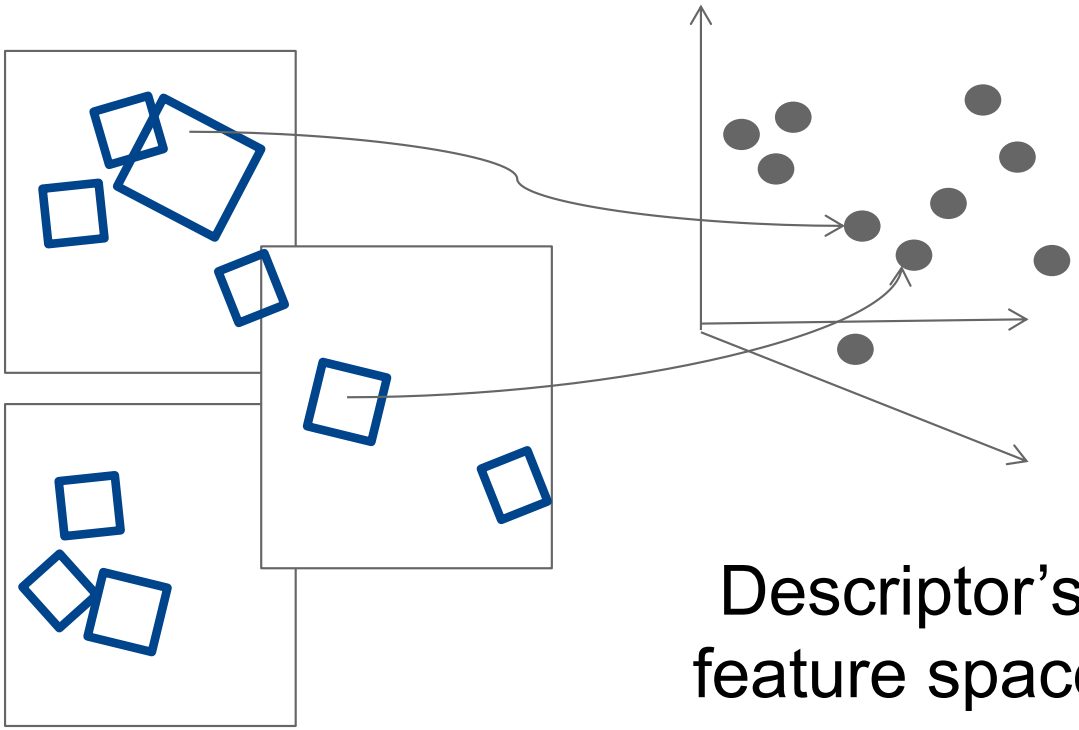
Results from 5k Flickr images (demo available for 100k set)



[Philbin CVPR'07]

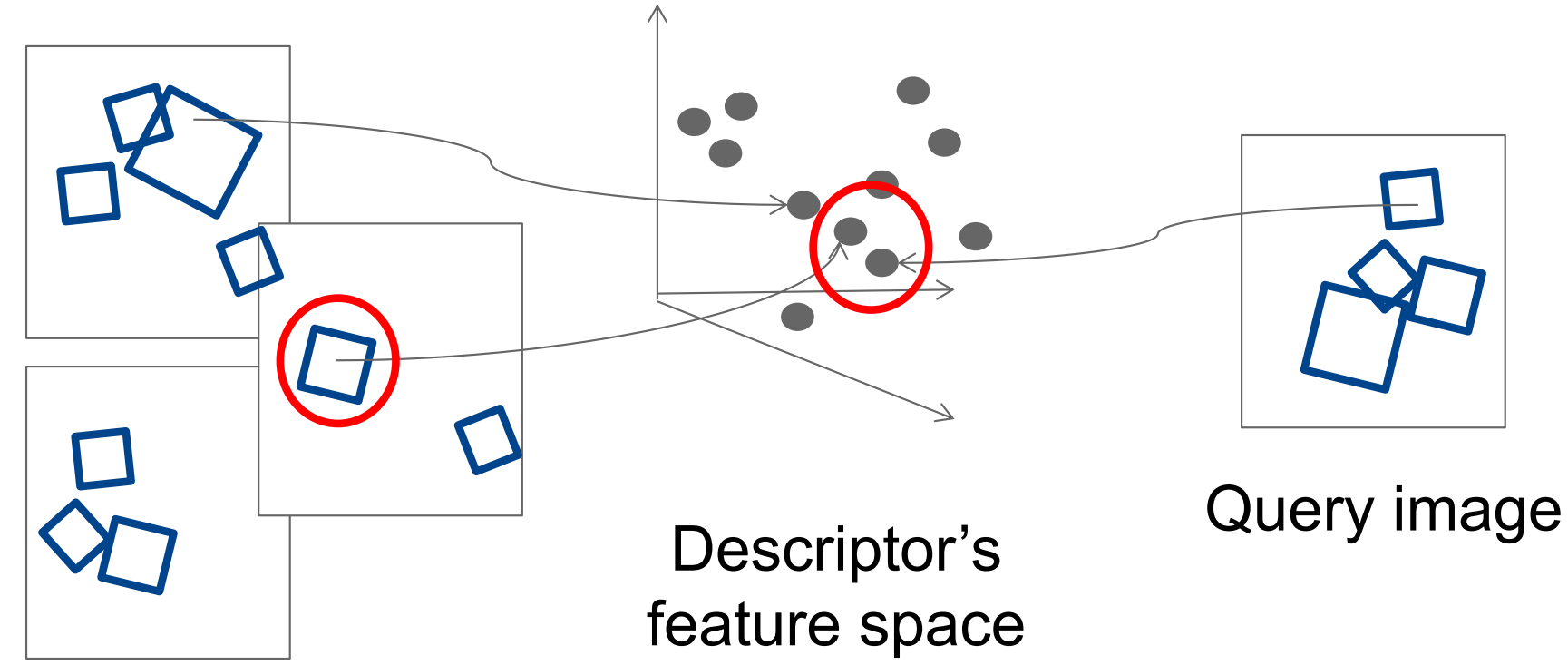
Indexing local features

- Each patch / region has a **descriptor**, which **is a point** in some high-dimensional feature space, e.g., SIFT.



Indexing local features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



Easily have millions of features to search!

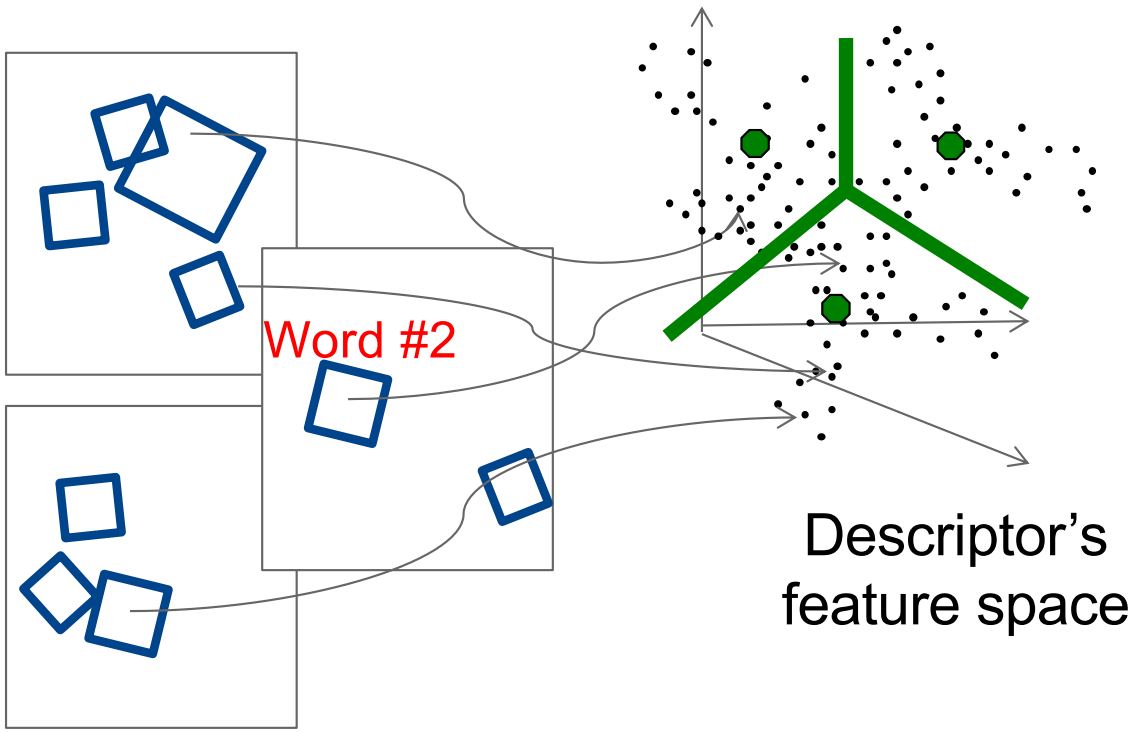
Indexing local features: inverted file index

- 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 map features to "**visual words**".

Index

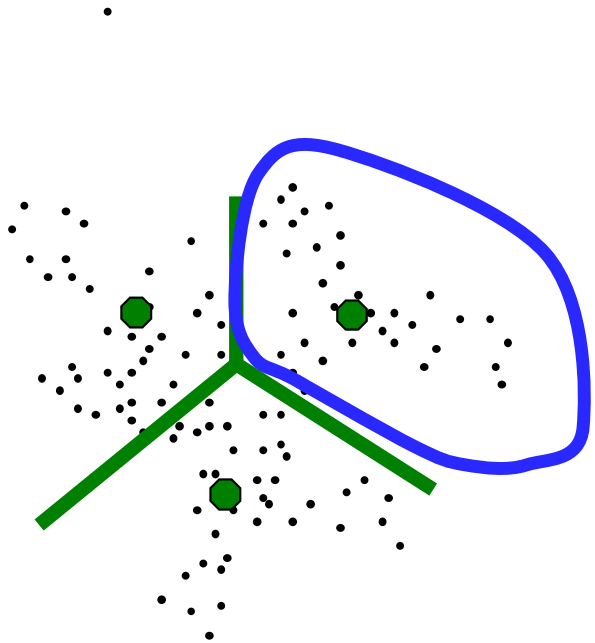
Visual Words

- Map descriptors to “words” by quantizing feature space
  - Quantize via clustering
    - Cluster centers are the prototype “words”
  - Determine which word to assign to each new image region by finding the closest cluster center.



Visual words

- Example: each group of patches belongs to the same visual word



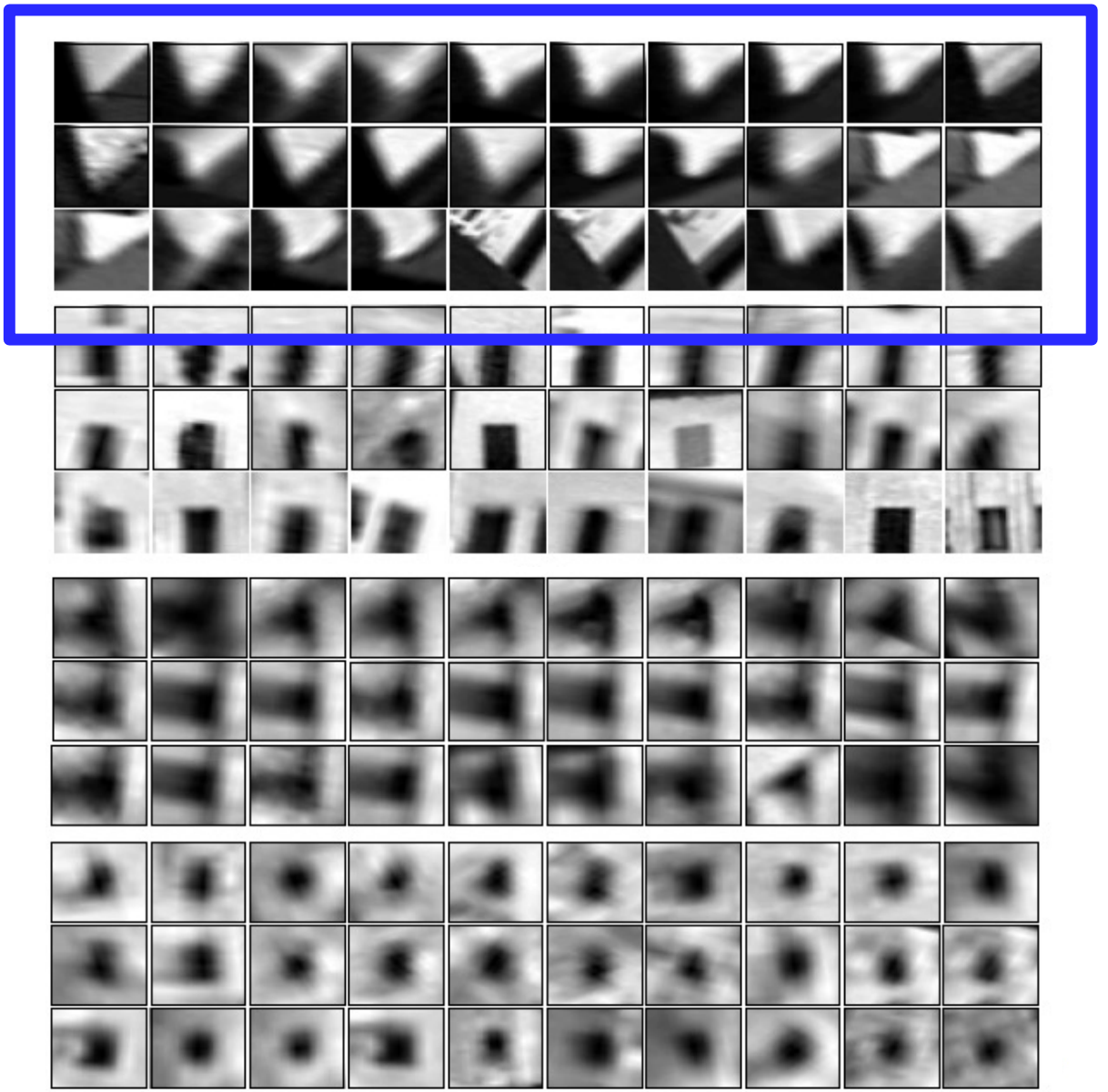


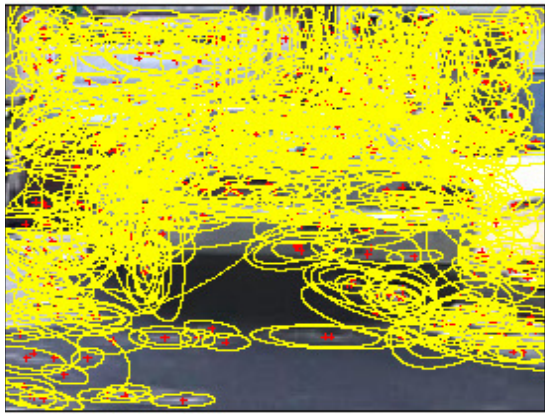
Figure from Sivic & Zisserman, ICCV 2003

Visual vocabulary formation

Things need to consider:

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features, vocabulary size, number of words (universal vocabulary?)

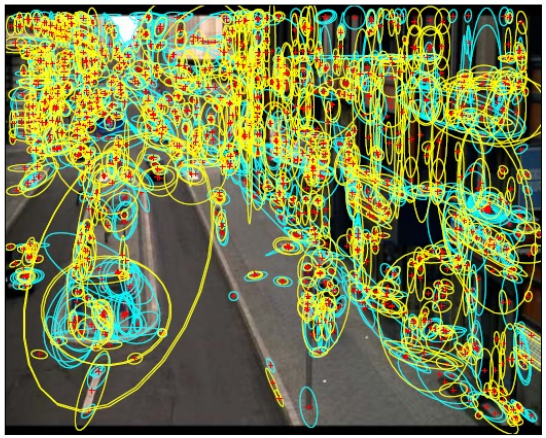
Sampling strategies



Sparse, at interest points

Dense, uniformly

Randomly



Multiple interest operators

- To find **specific**, textured **objects**, **sparse sampling** from interest points often more reliable.
- **Multiple** complementary interest **operators** offer more image coverage.
- For object **categorization**, **dense sampling** offers better coverage.

[See Nowak, Jurie & Triggs, ECCV 2006]

Image credits: F-F. Li, E. Nowak, J. Sivic

Inverted file index

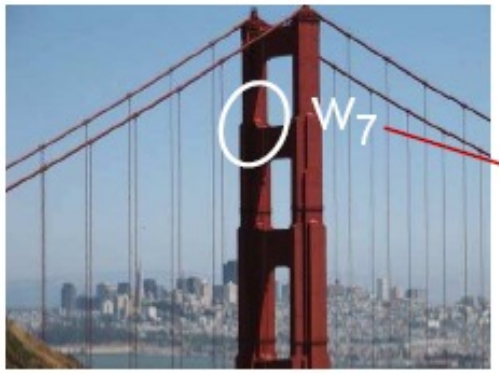
- Database images are loaded into the index mapping words to image numbers

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

…

Inverted file index

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



New query image

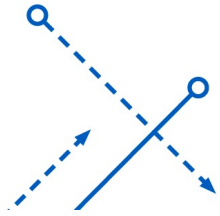
Word #	Image #
1	3
2	
…	
7	1, 2
…	
8	3
9	
10	
…	
91	2
⋮	⋮

Image #1

Image #2

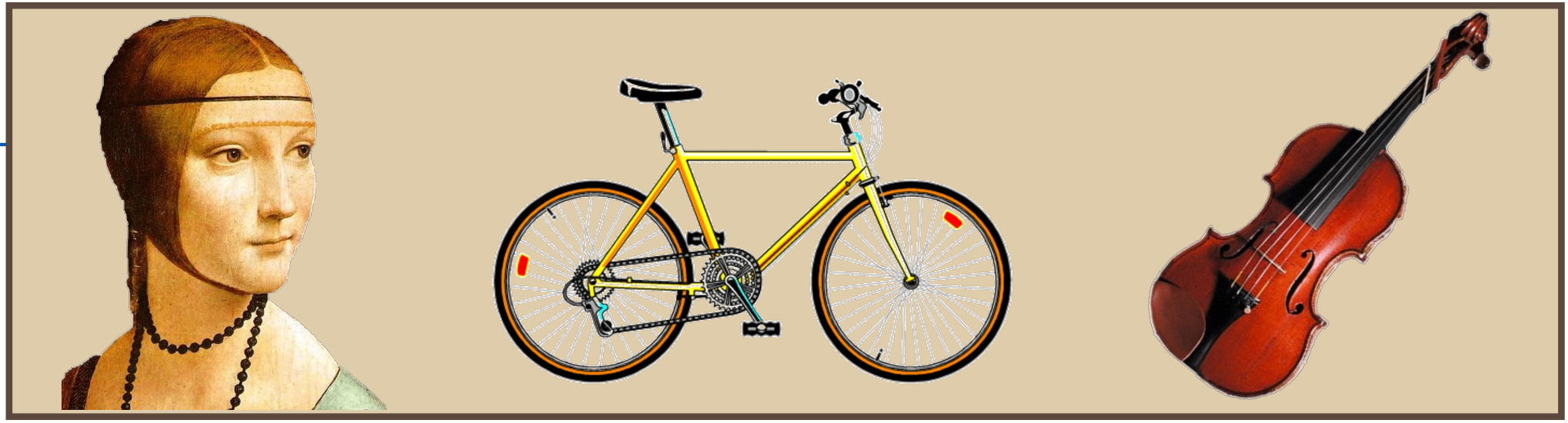
Inverted file index

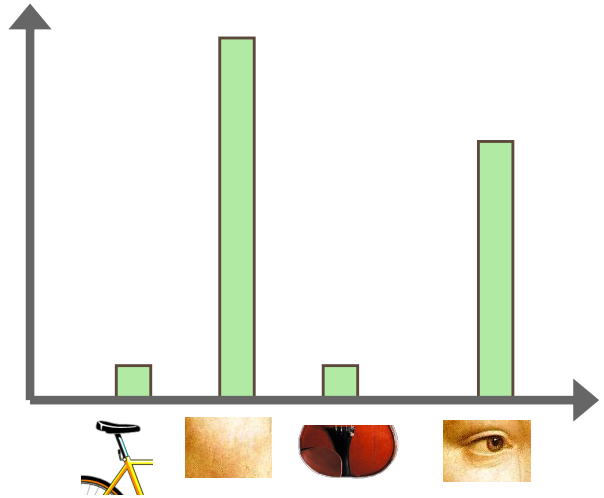
- Key requirement for inverted file index to be efficient:
  - Sparsity
  - If most pages/images contain most words, then it's no better off than exhaustive search.
    - Exhaustive search would mean comparing the word distribution of a query versus every page.

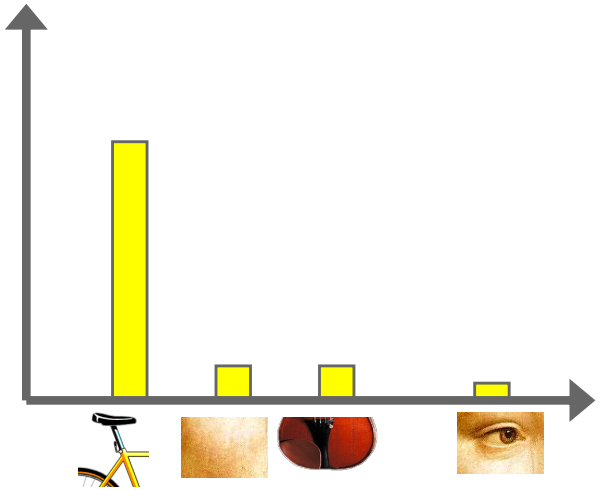


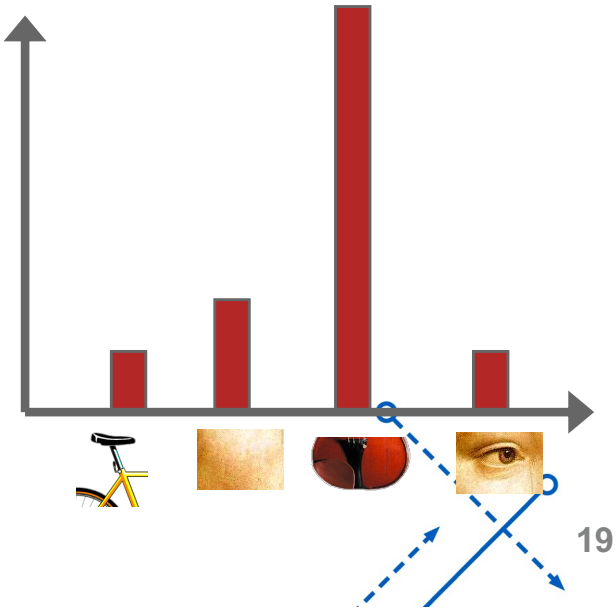
Instance recognition: remaining issues

- How to summarize the content of an entire image? And estimate overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?



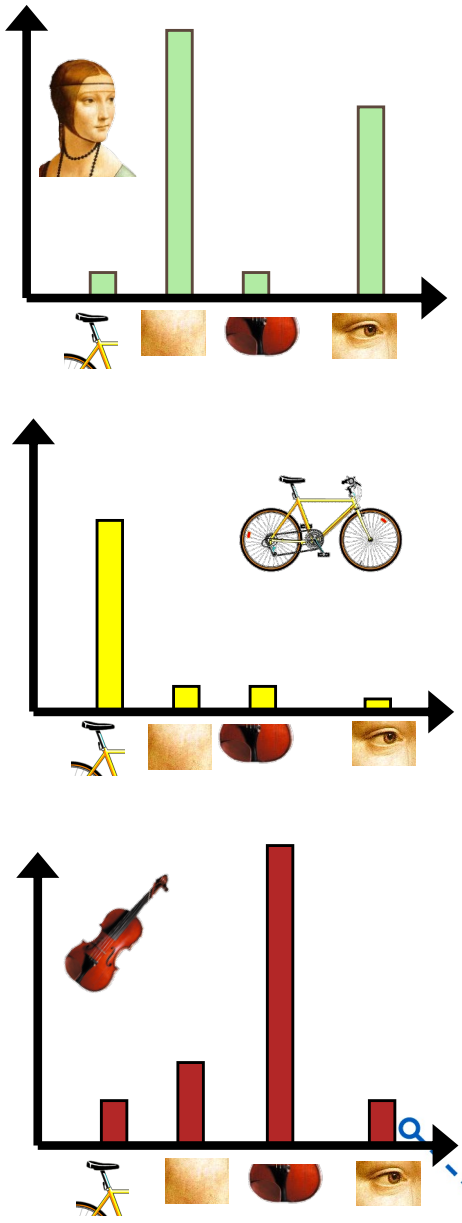






Bags of visual words

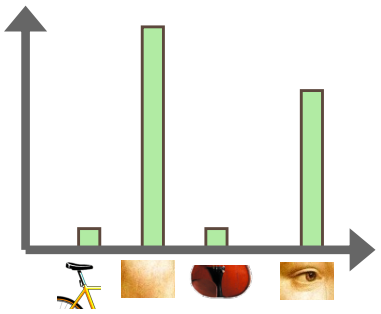
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



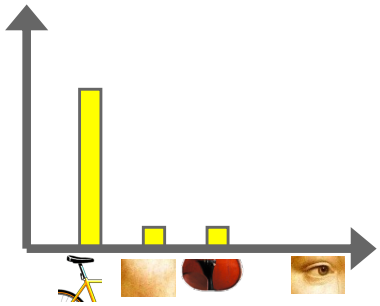
Comparing bags of words

- Rank frames by normalized inner product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.

[1 8 1 4]

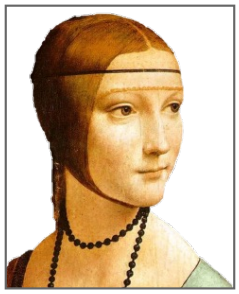


[5 1 1 0]

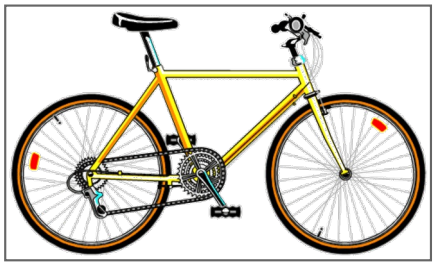


$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

for vocabulary of *V* words



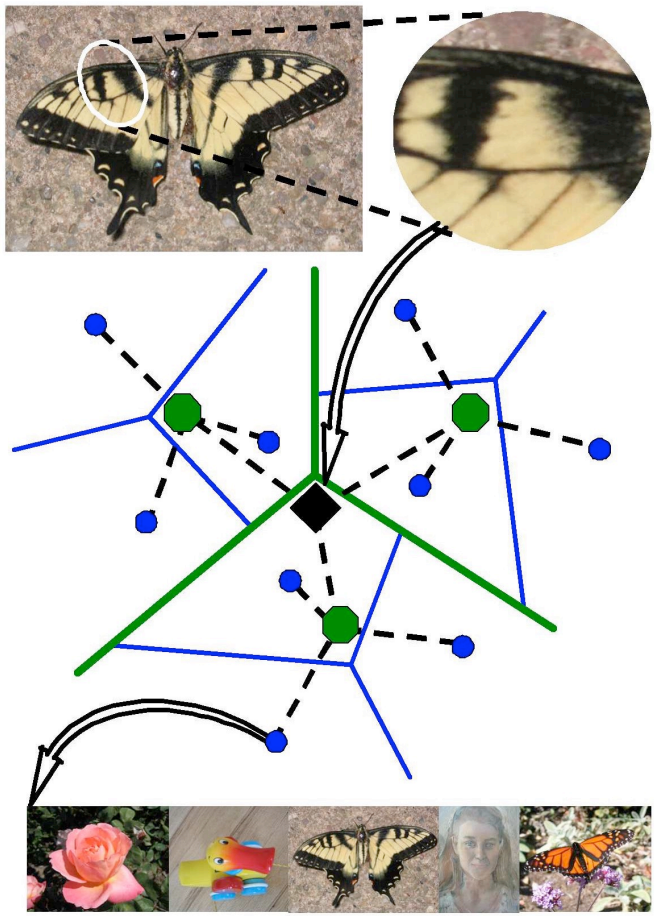
$\vec{d}_j$



$\vec{q}$

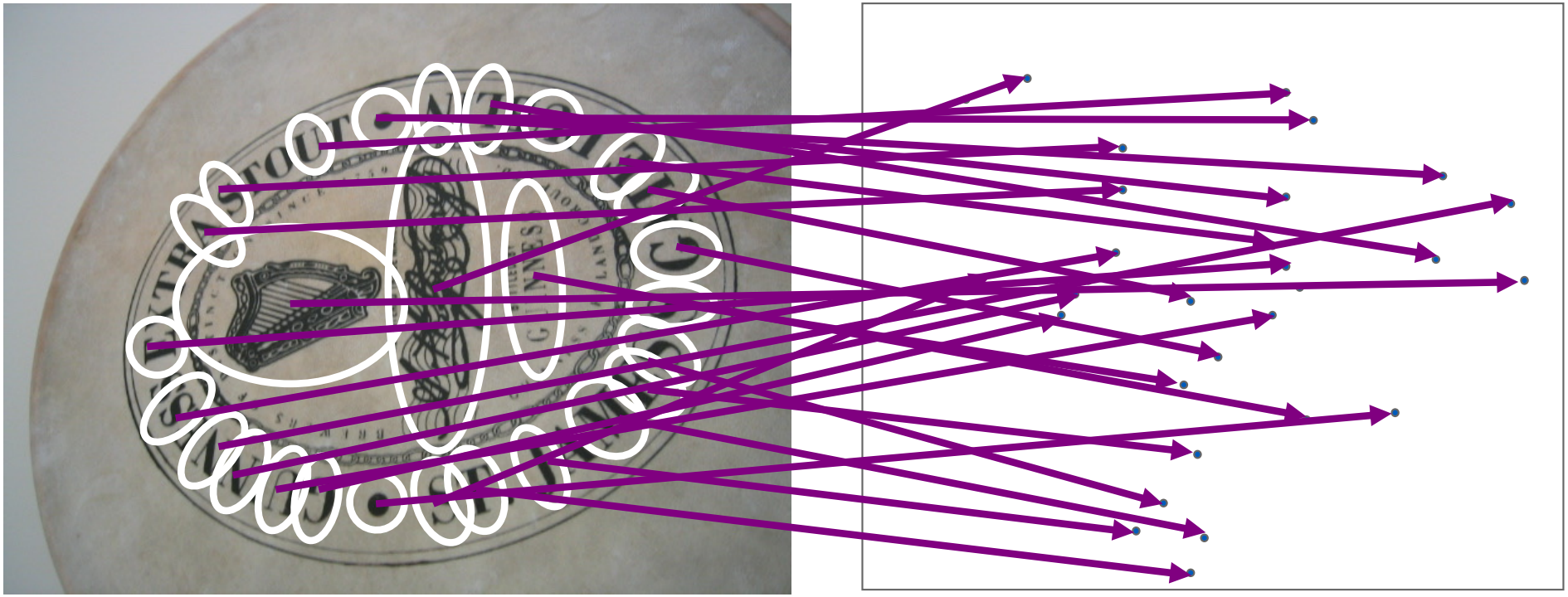
Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - **Vocabulary trees**



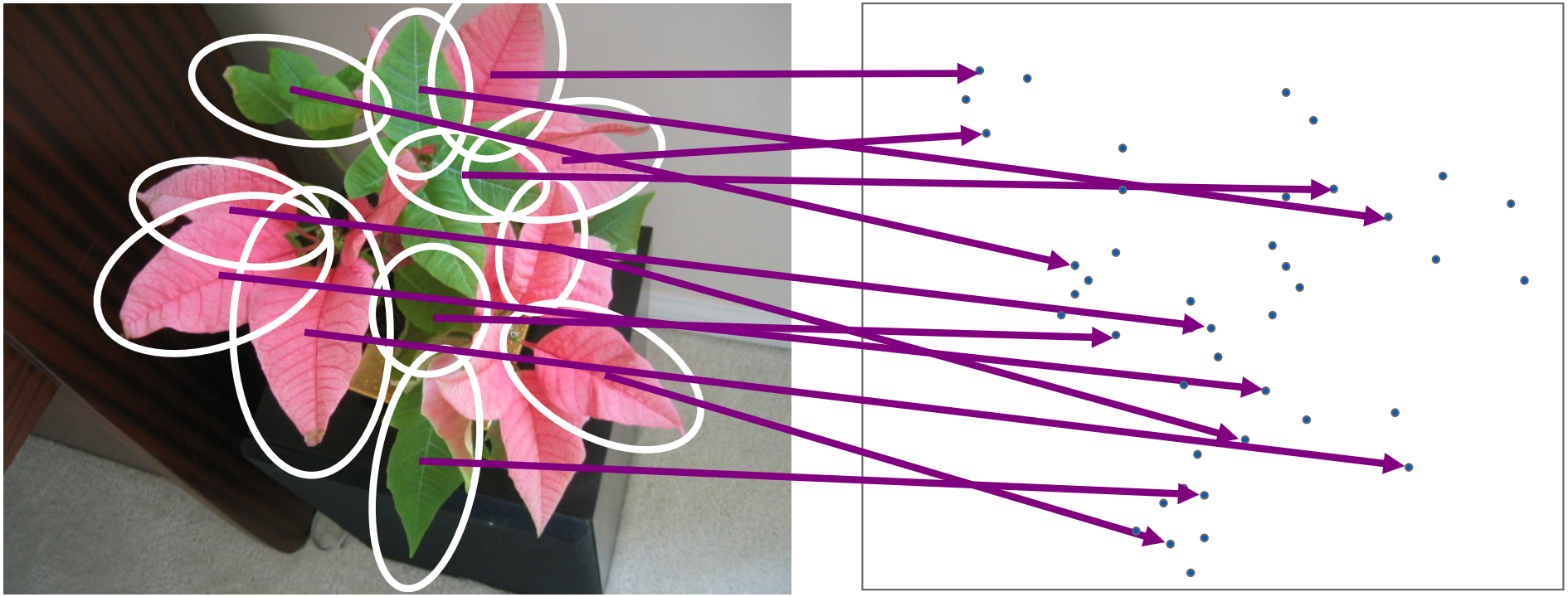
(Nister & Stewenius, 2006)

Recognition with K-tree



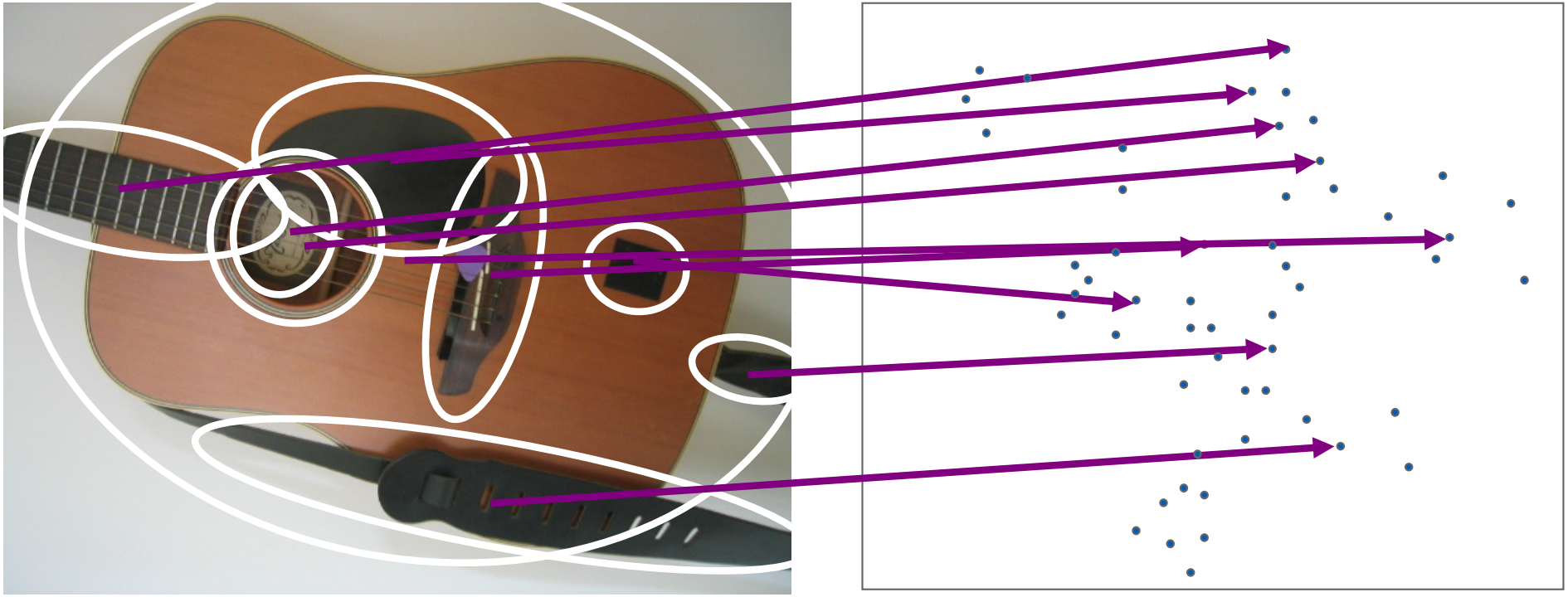
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



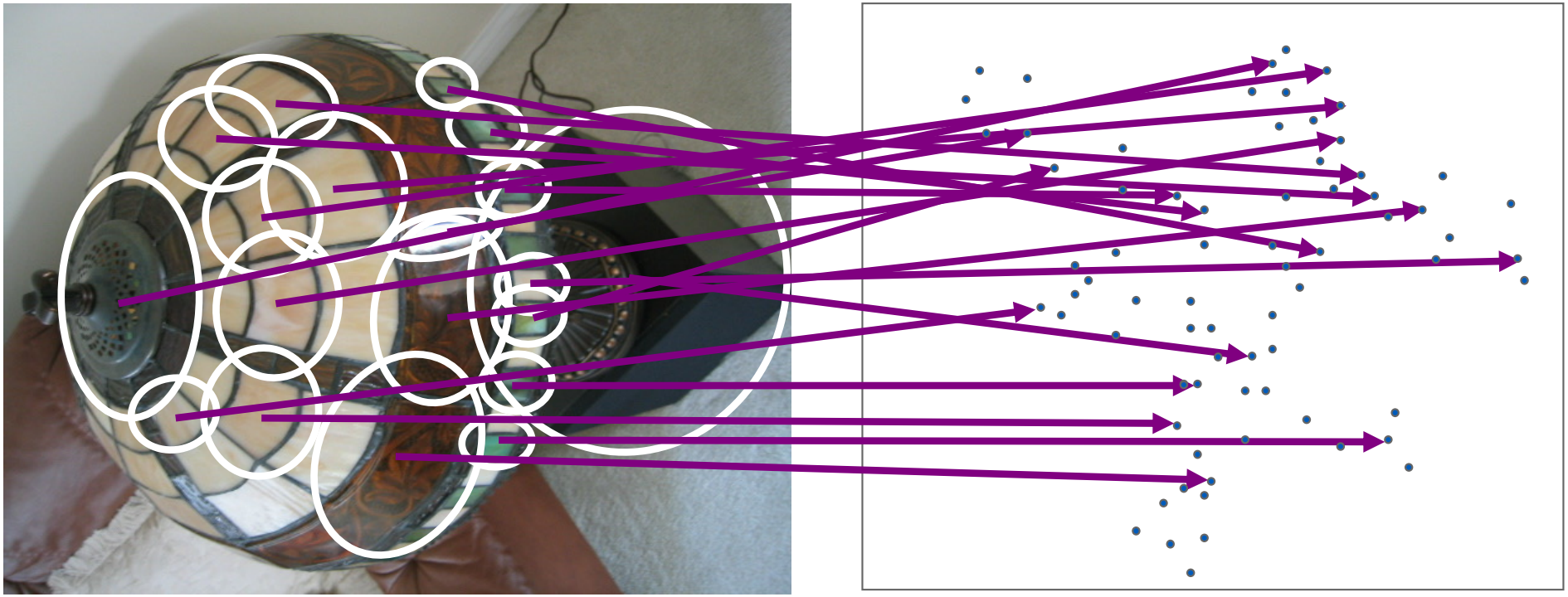
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



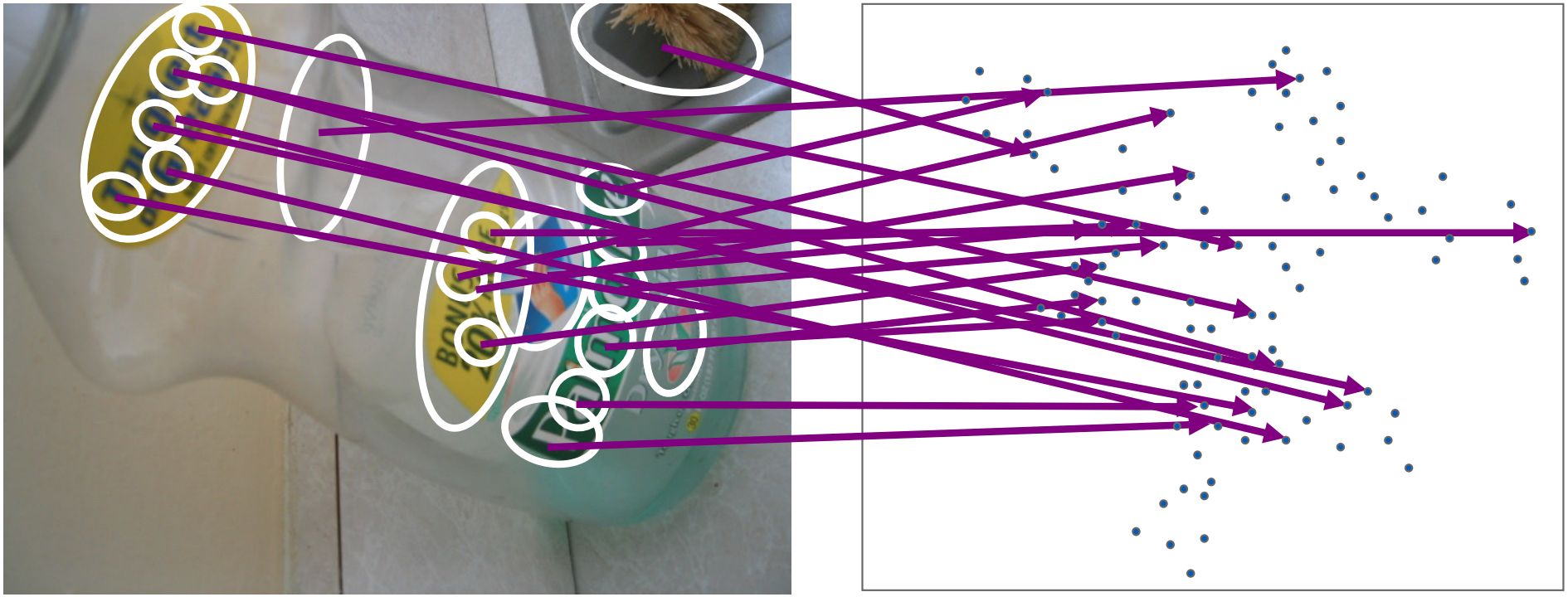
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



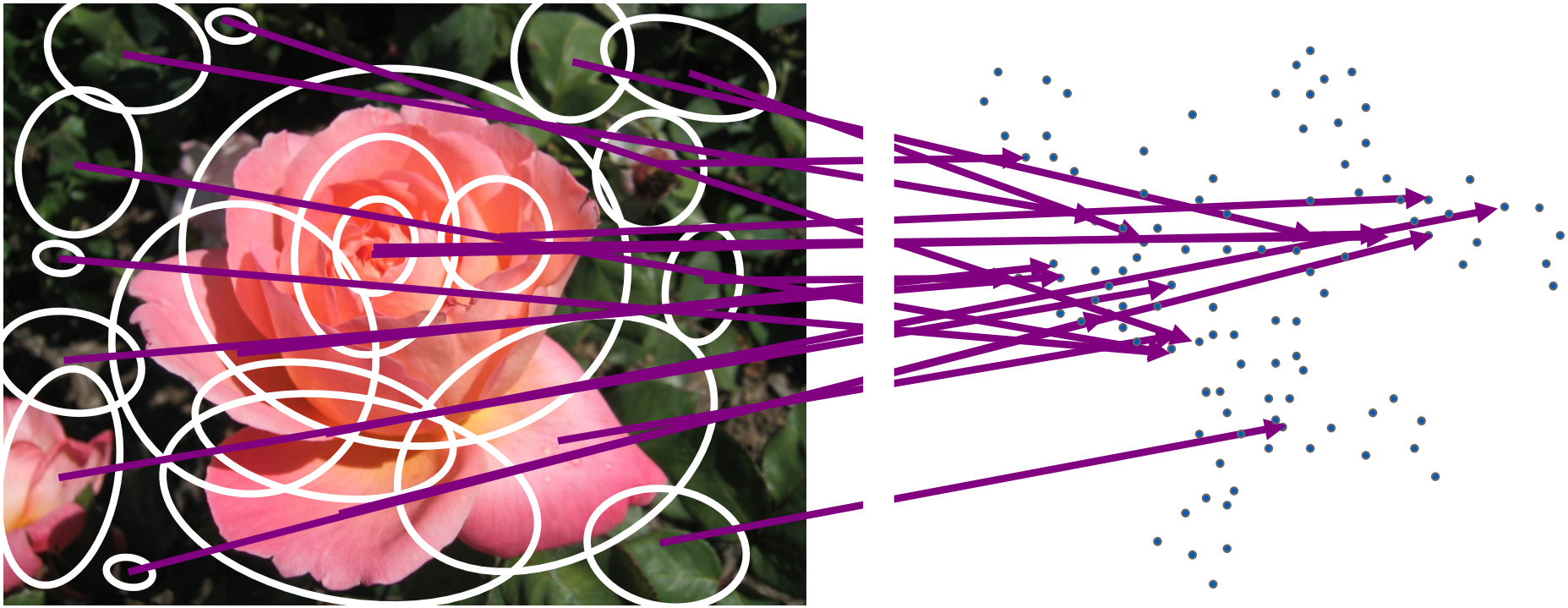
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



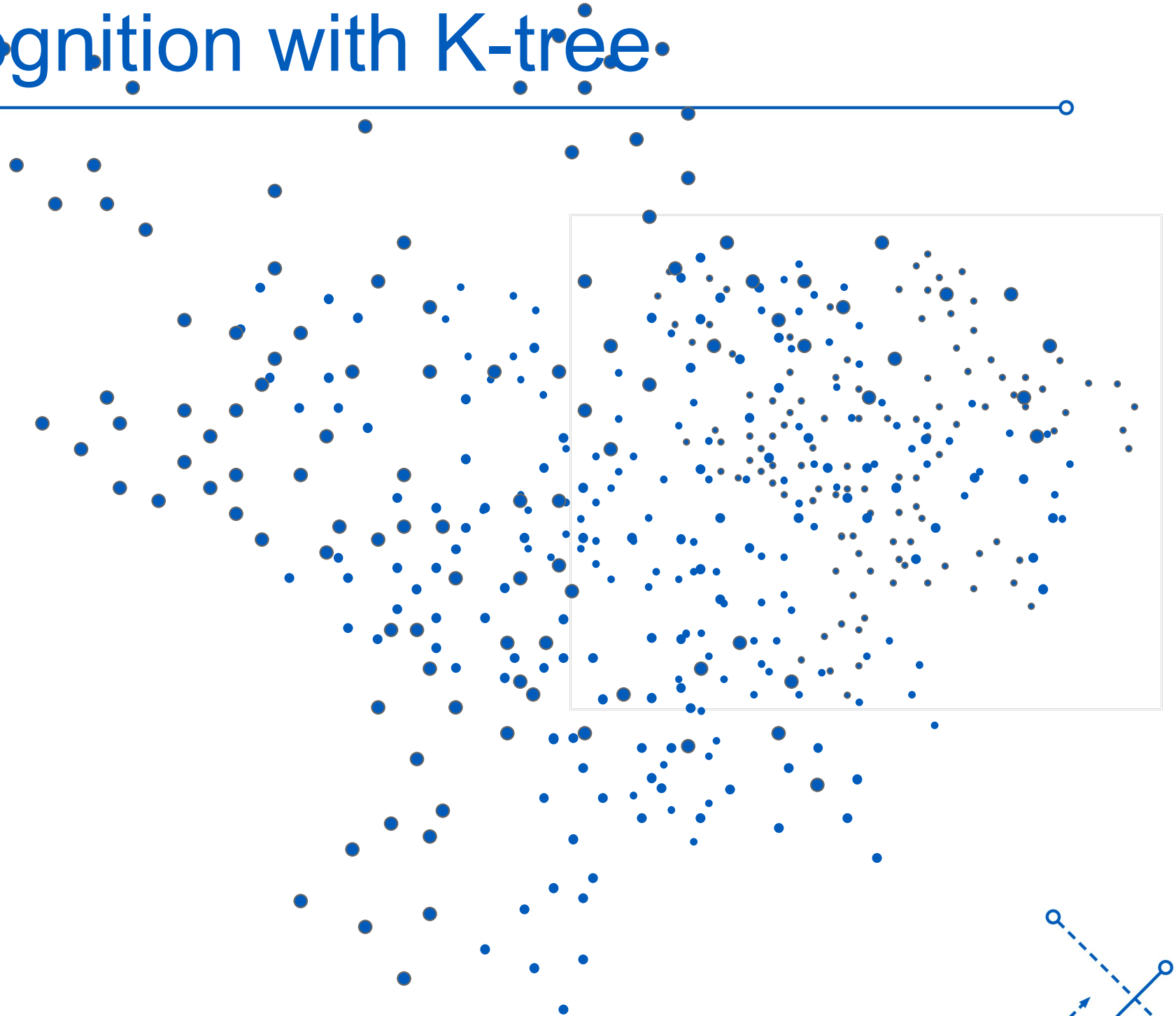
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



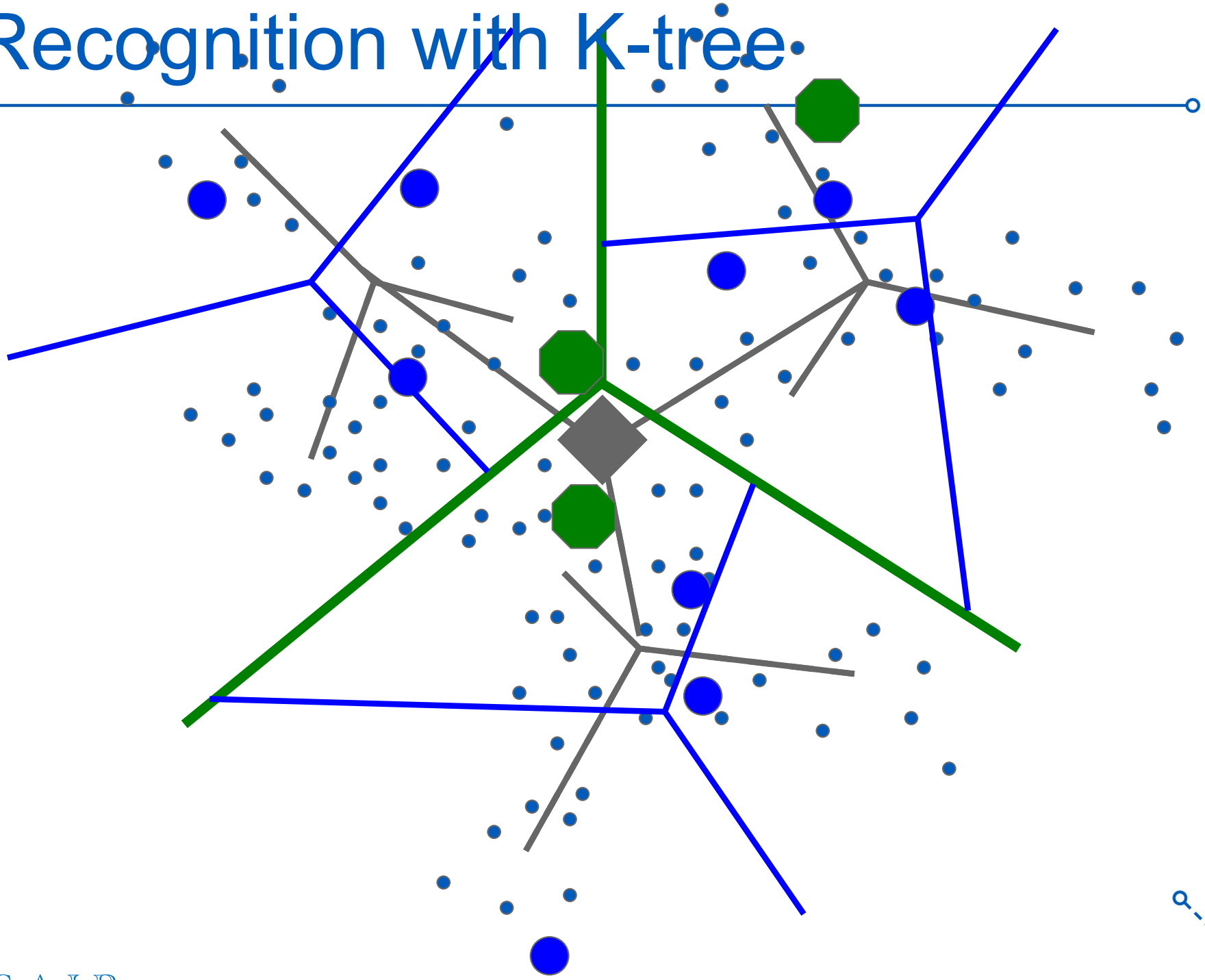
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



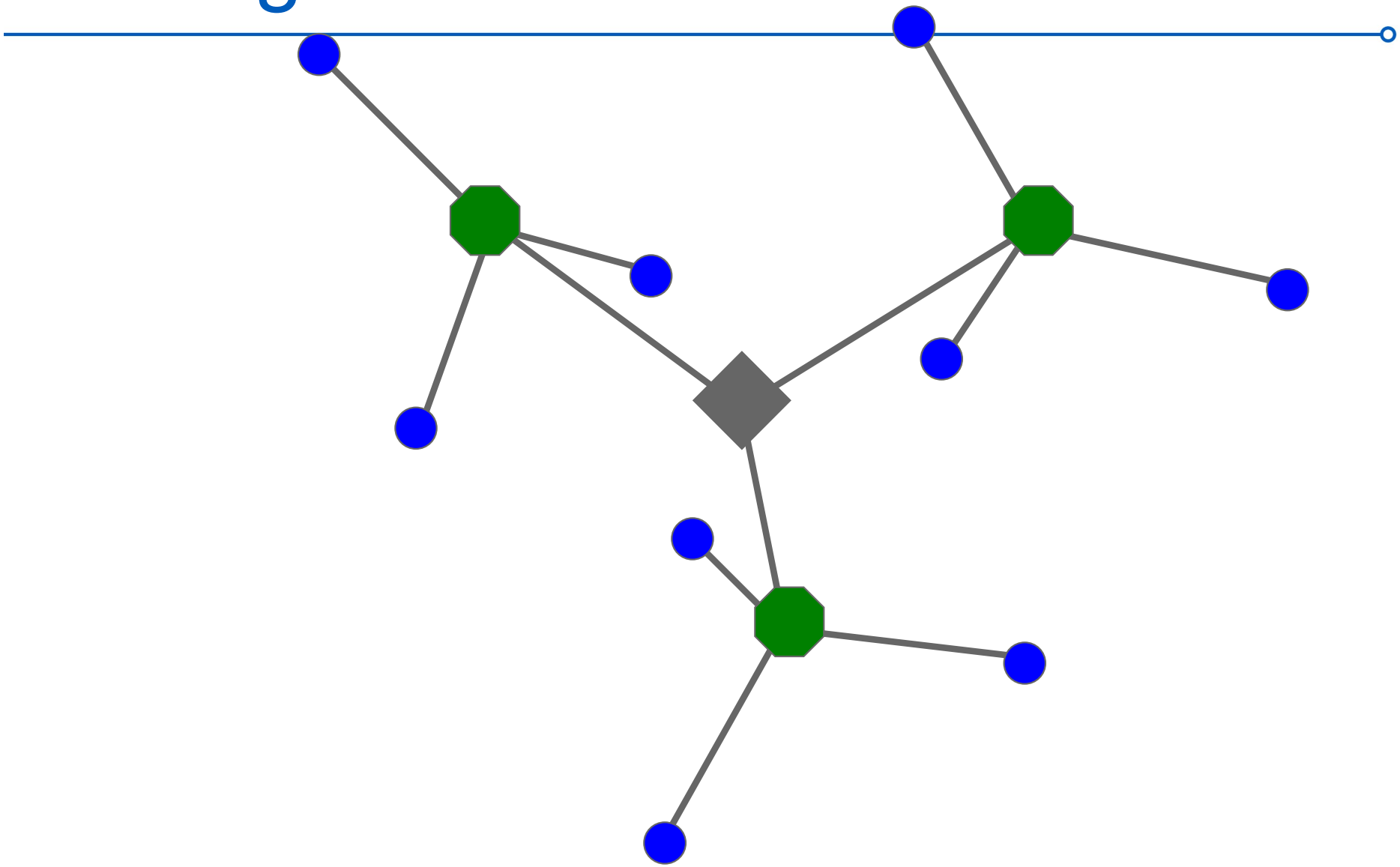
Following slides by David Nister (CVPR 2006)

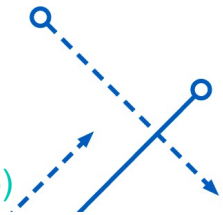
Recognition with K-tree



Following slides by David Nister (CVPR 2006)

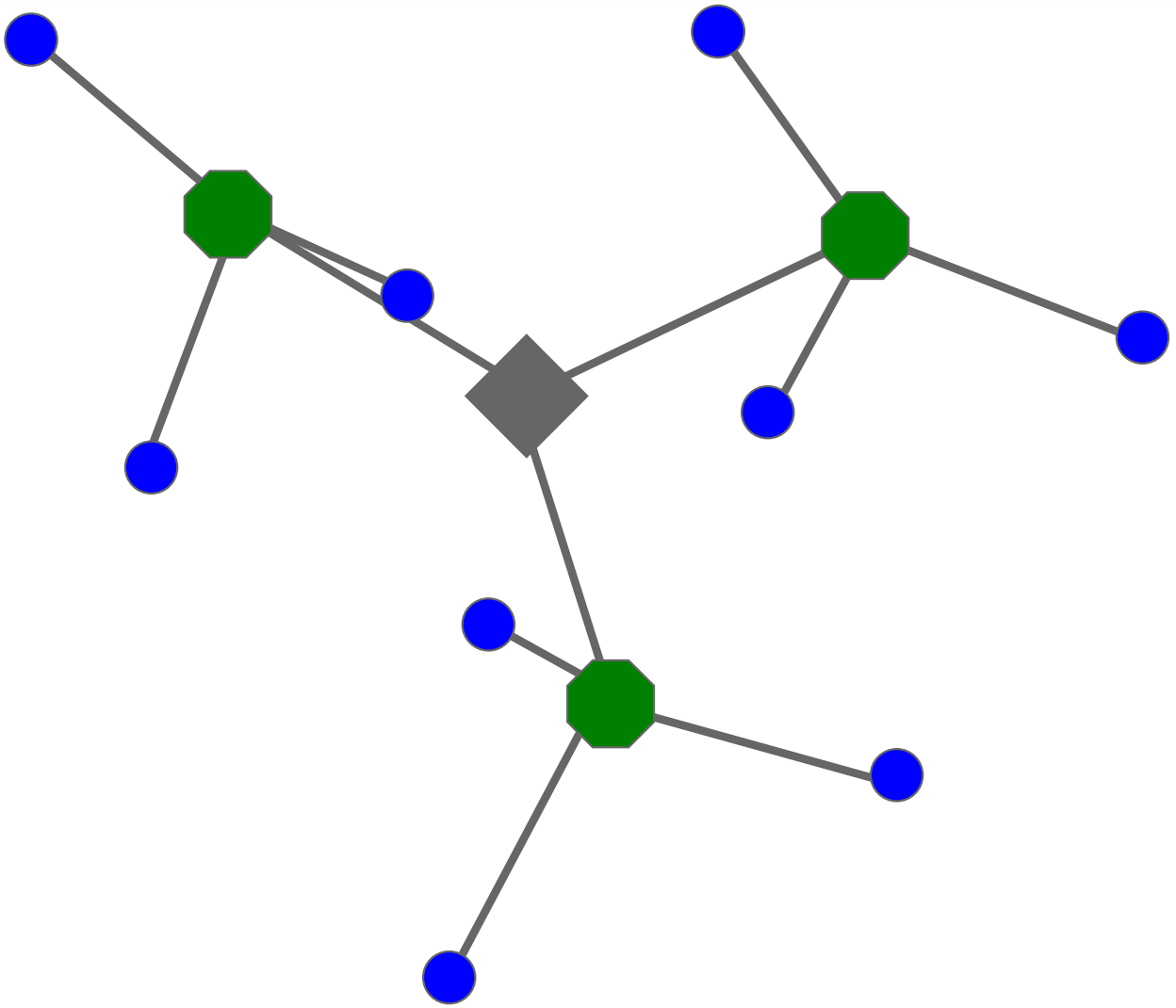
Recognition with K-tree



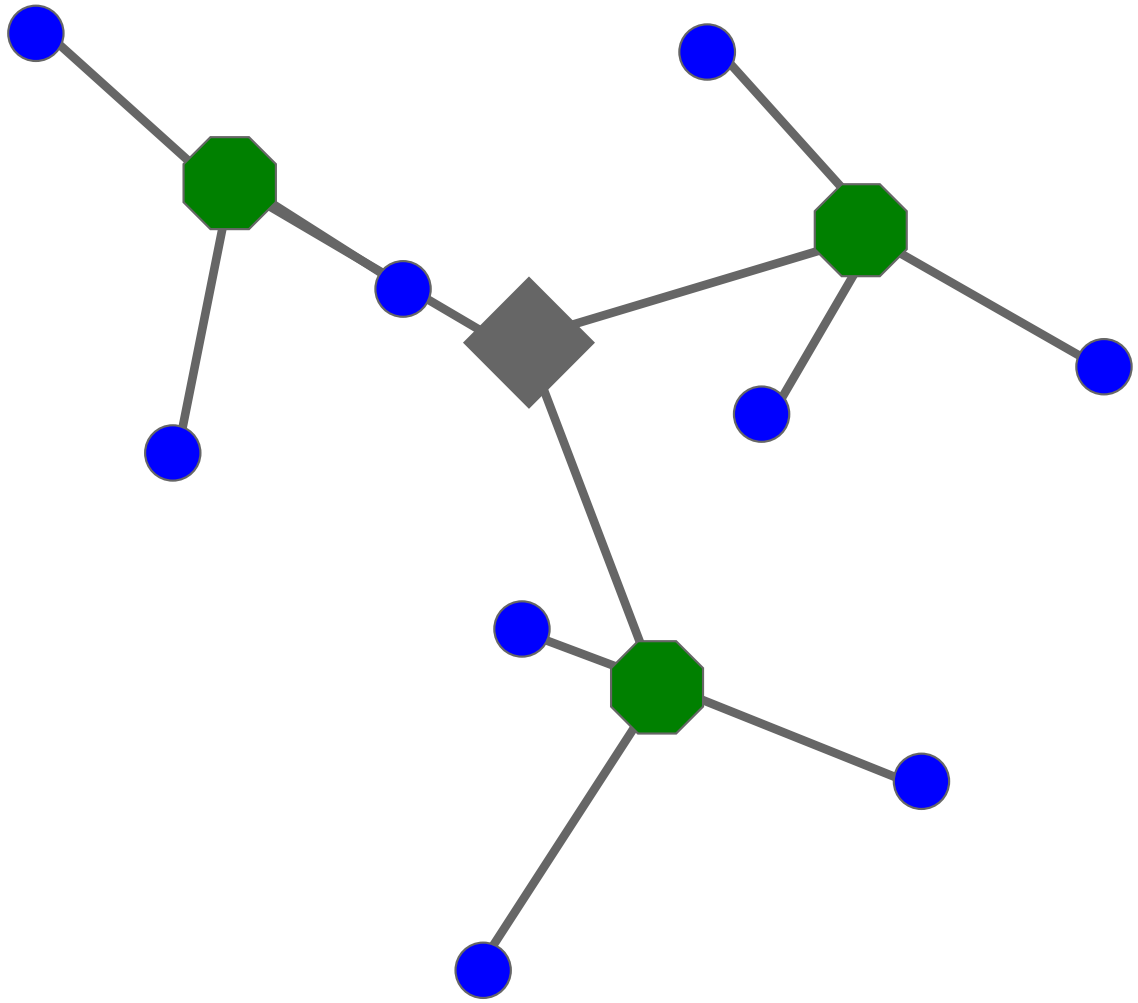


Following slides by David Nister (CVPR 2006)

Recognition with K-tree

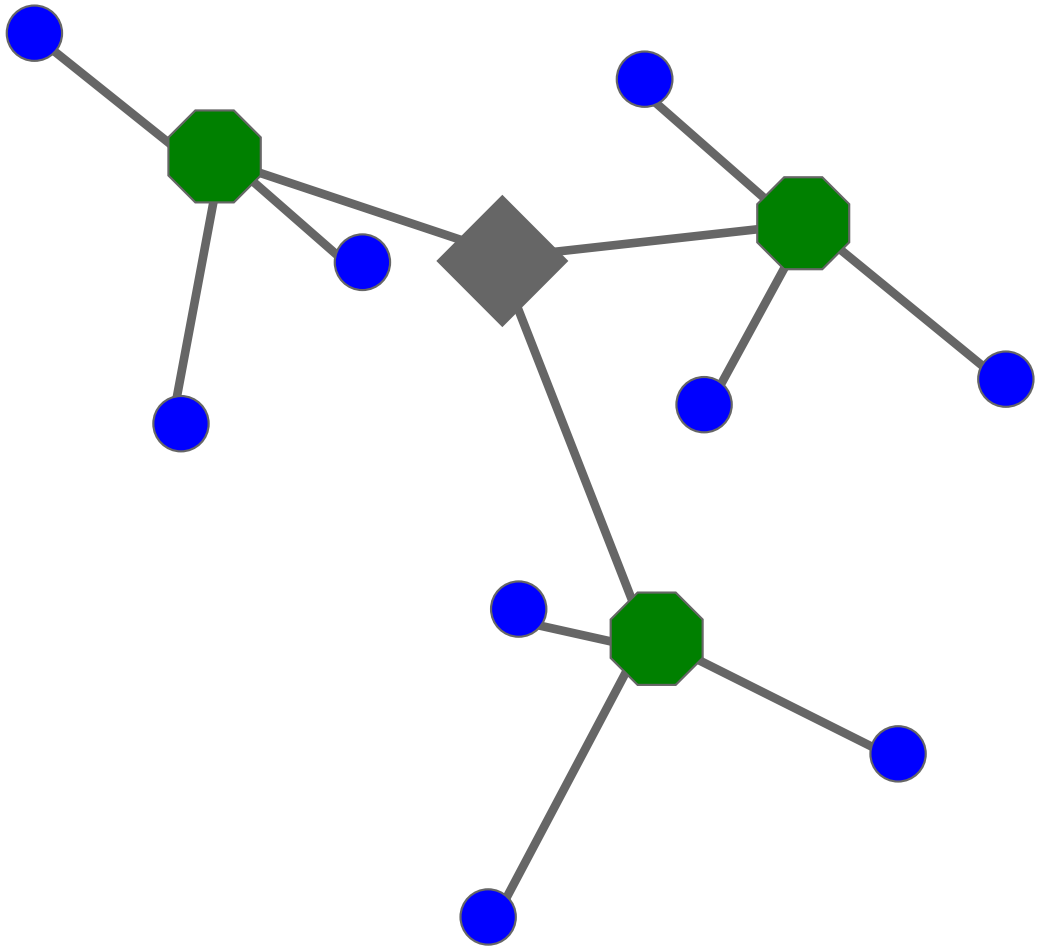


Recognition with K-tree



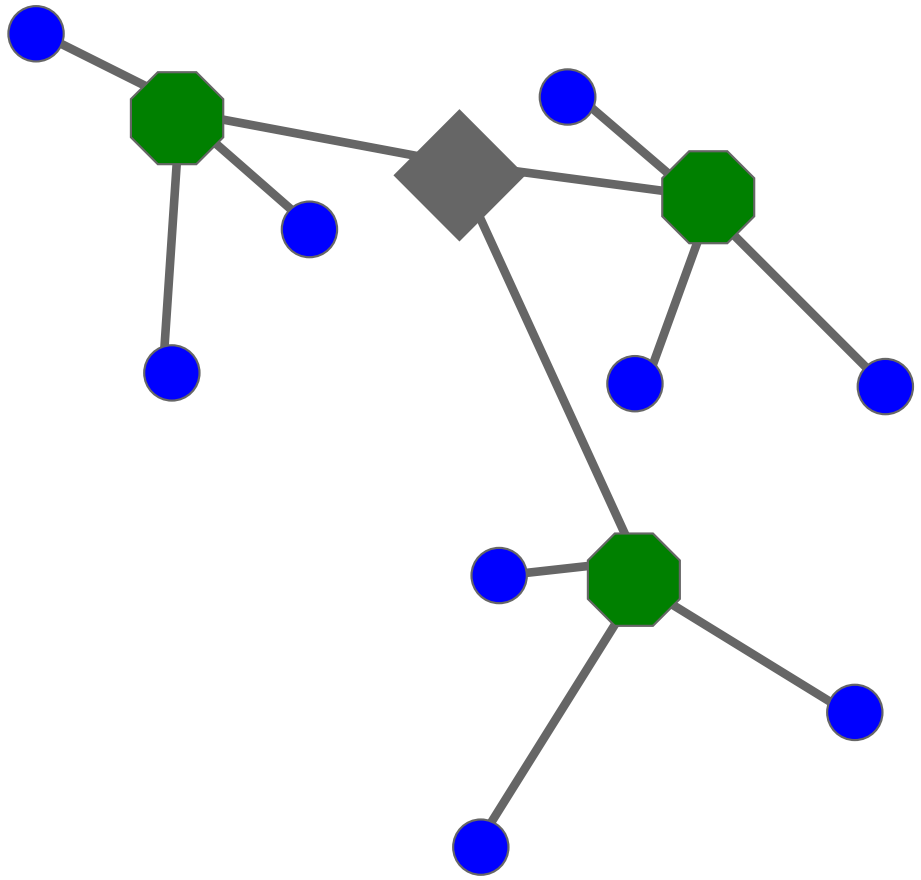
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



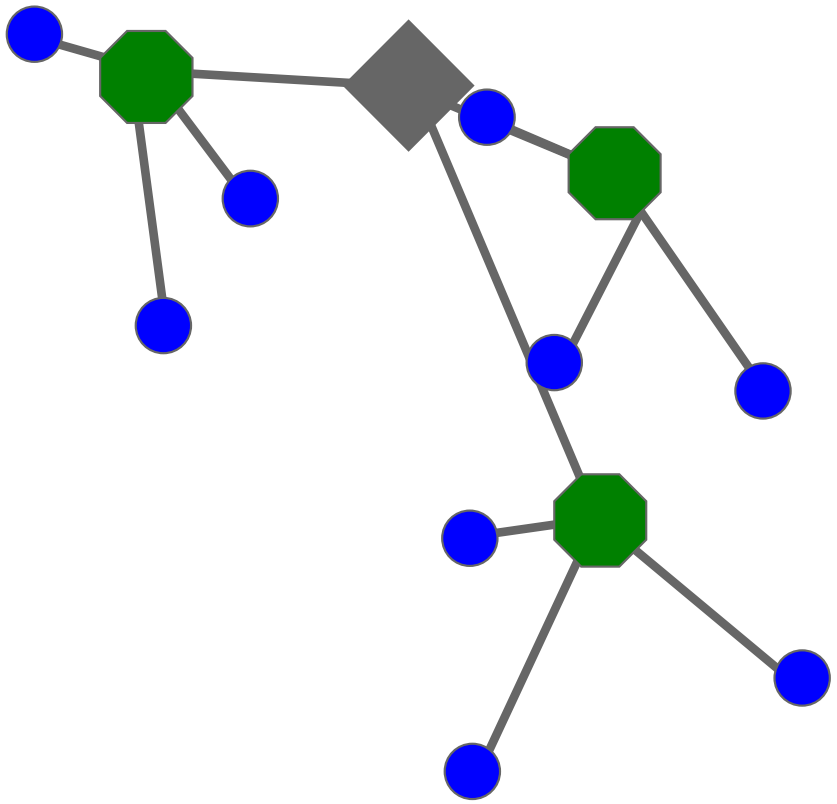
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



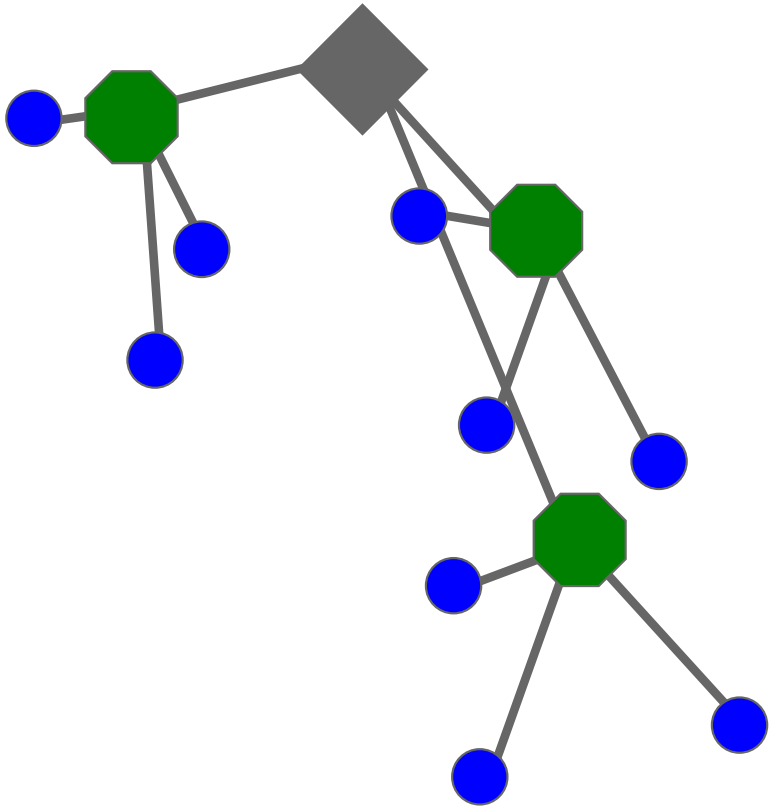
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



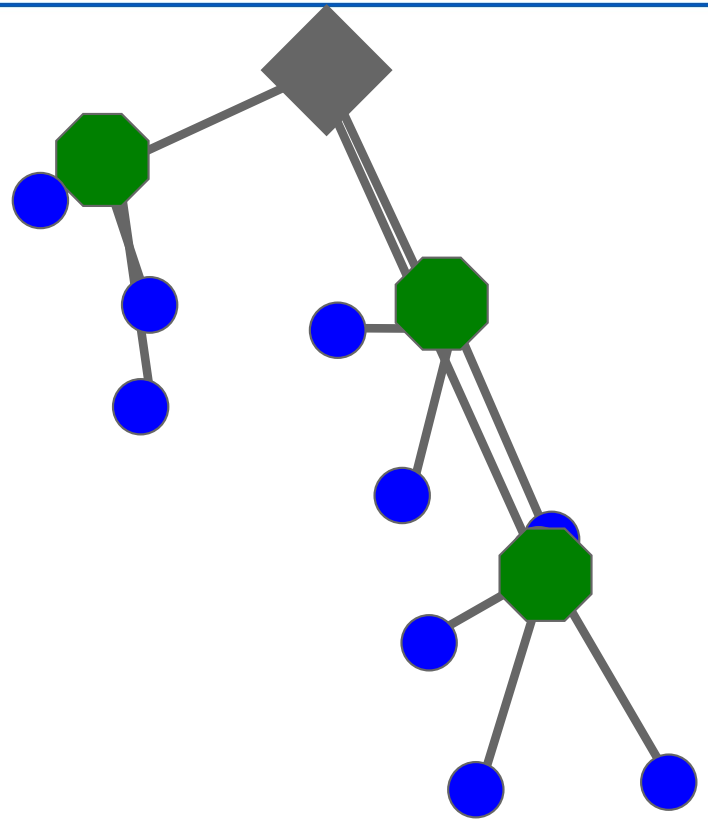
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



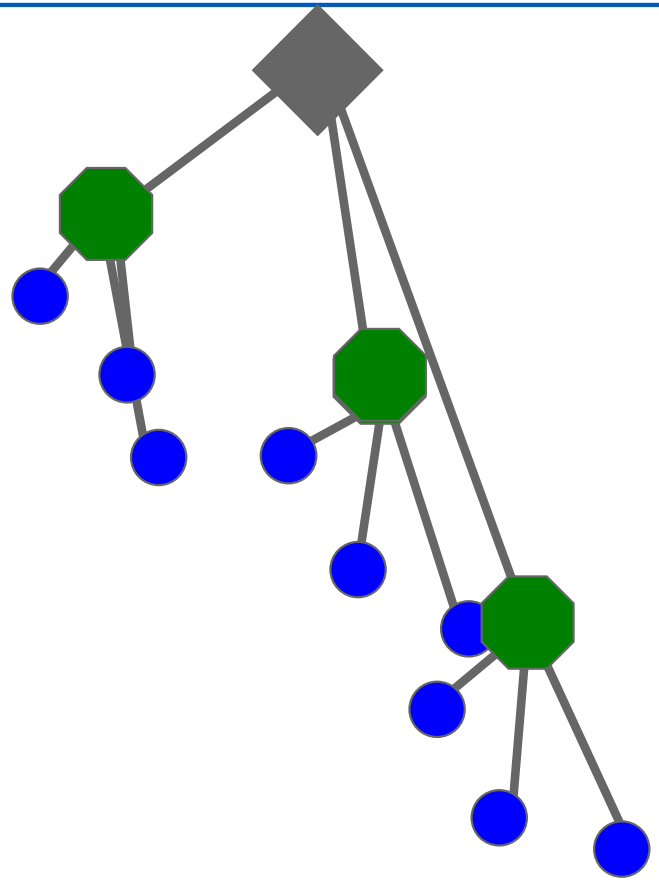
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



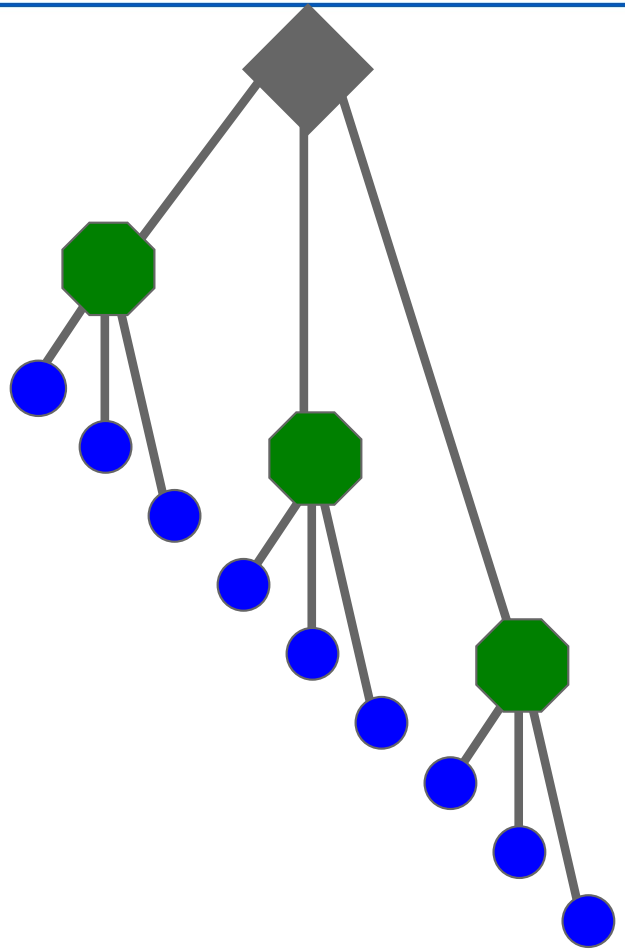
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



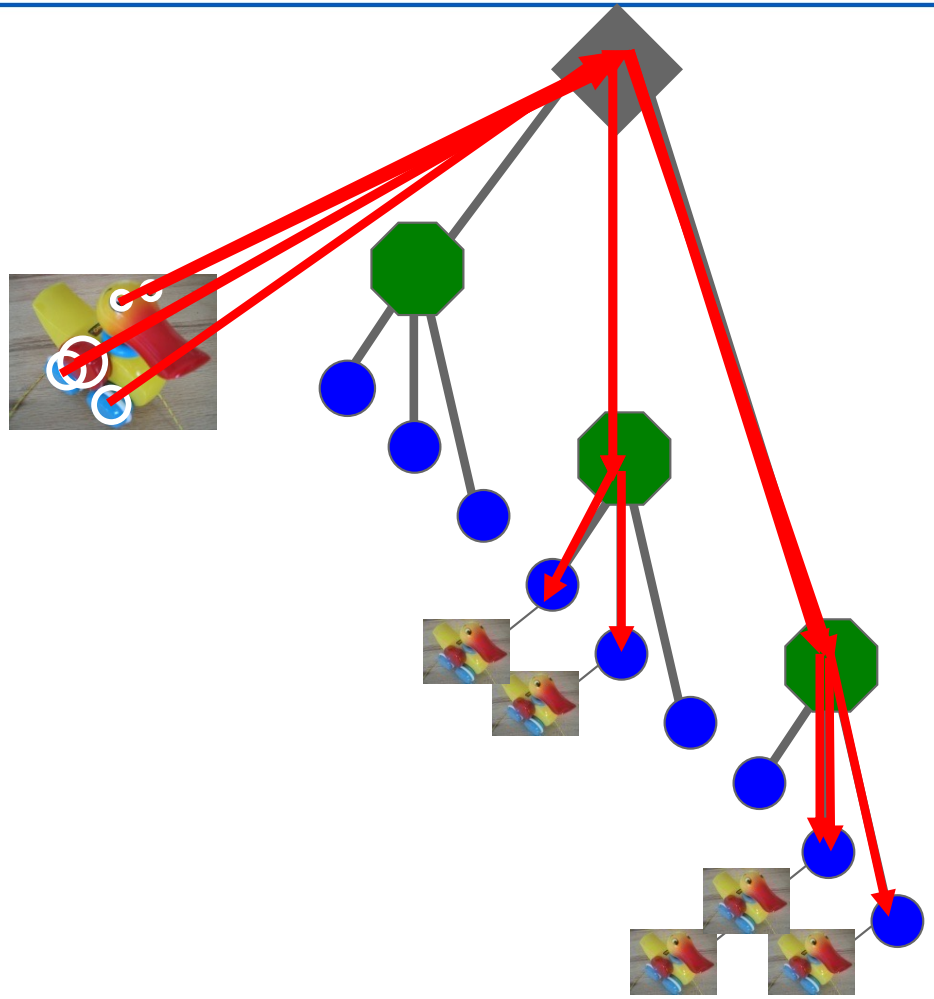
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



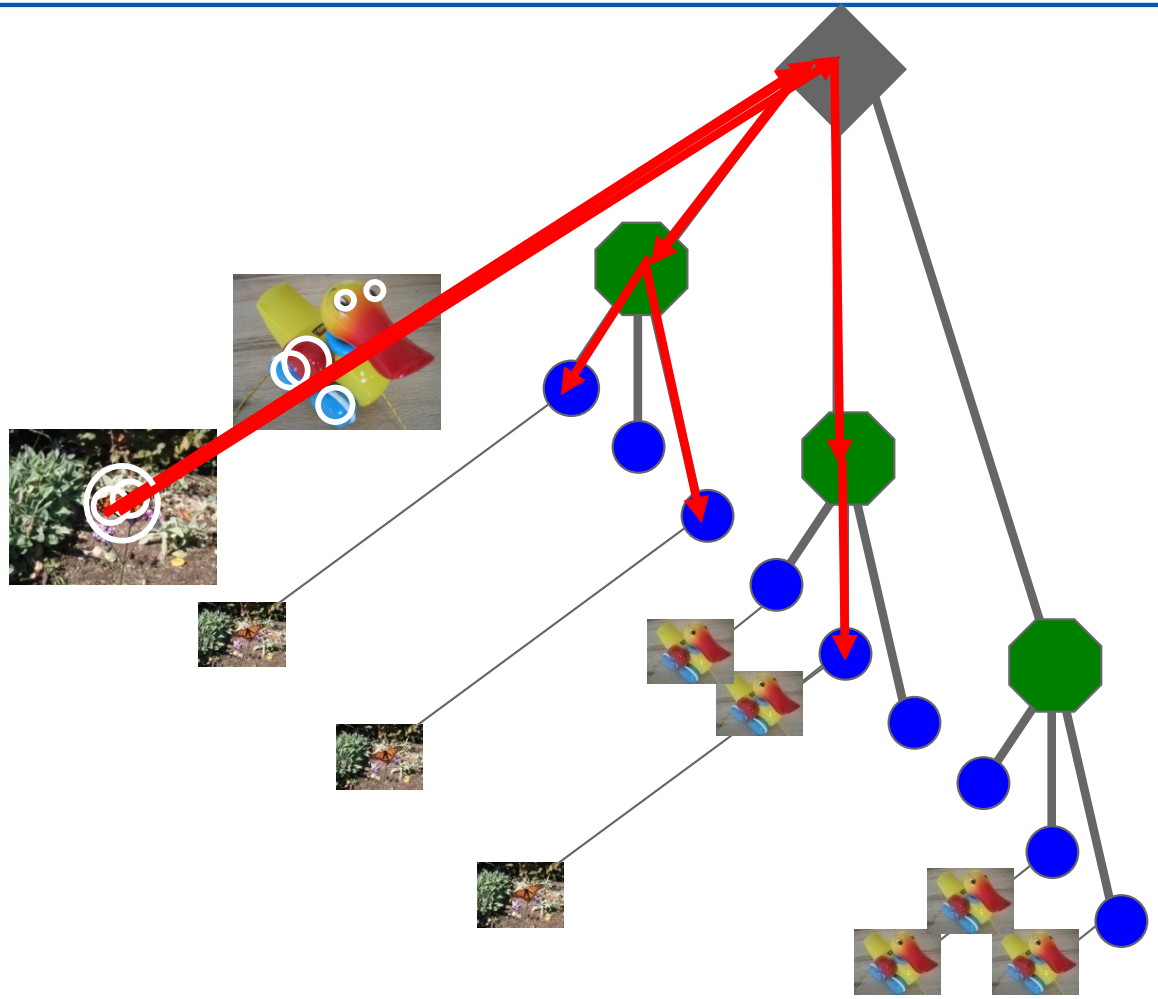
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



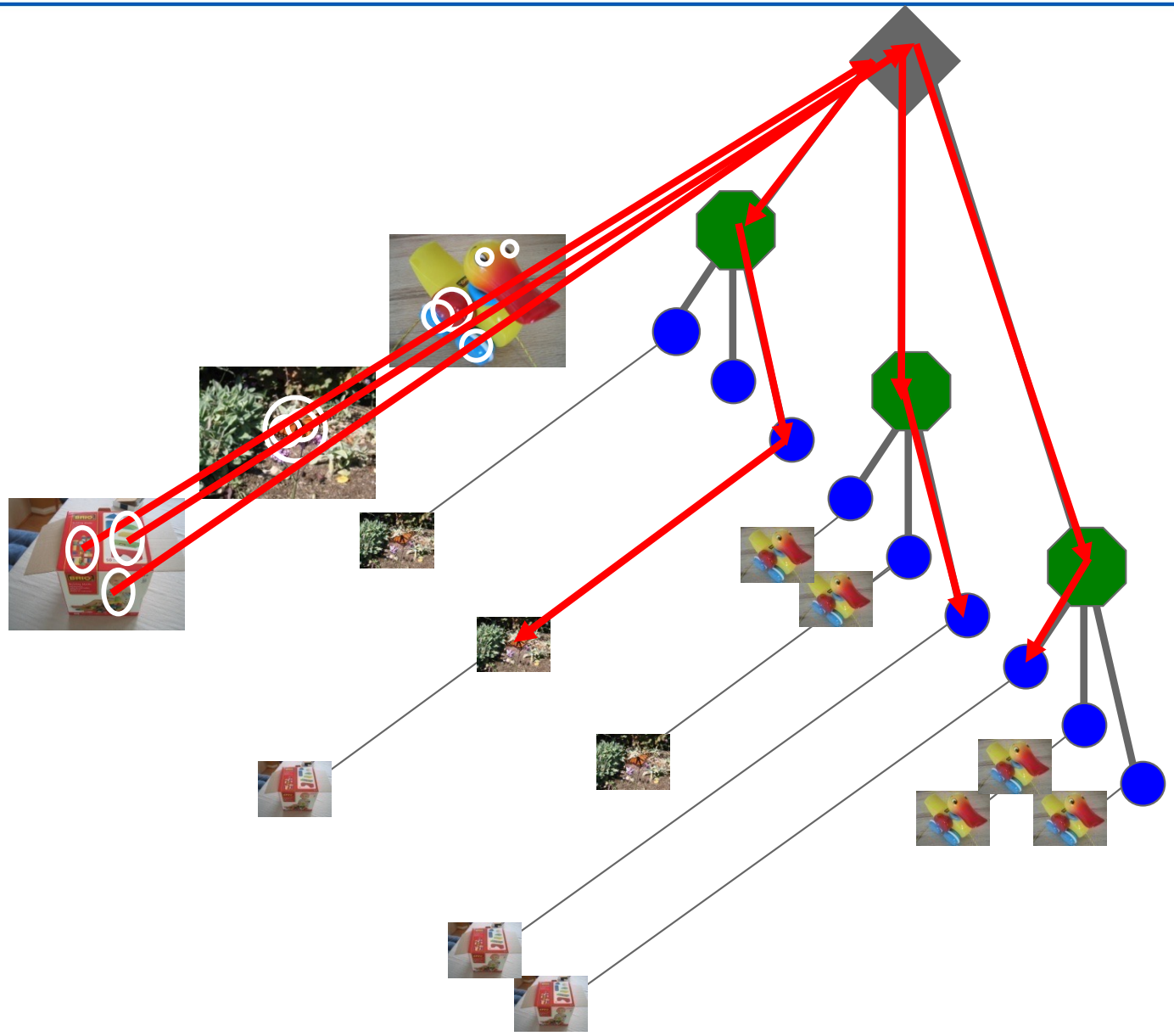
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



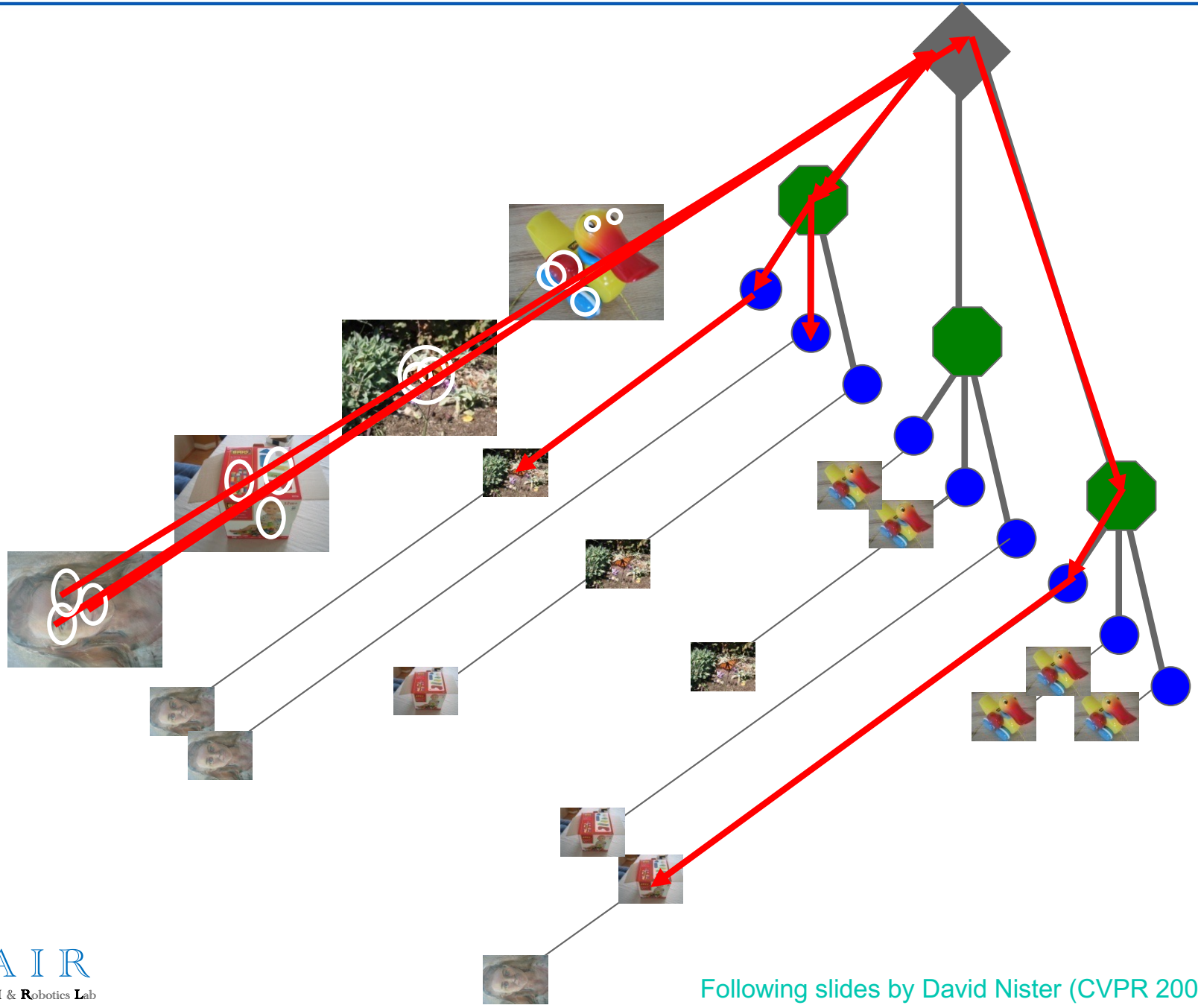
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



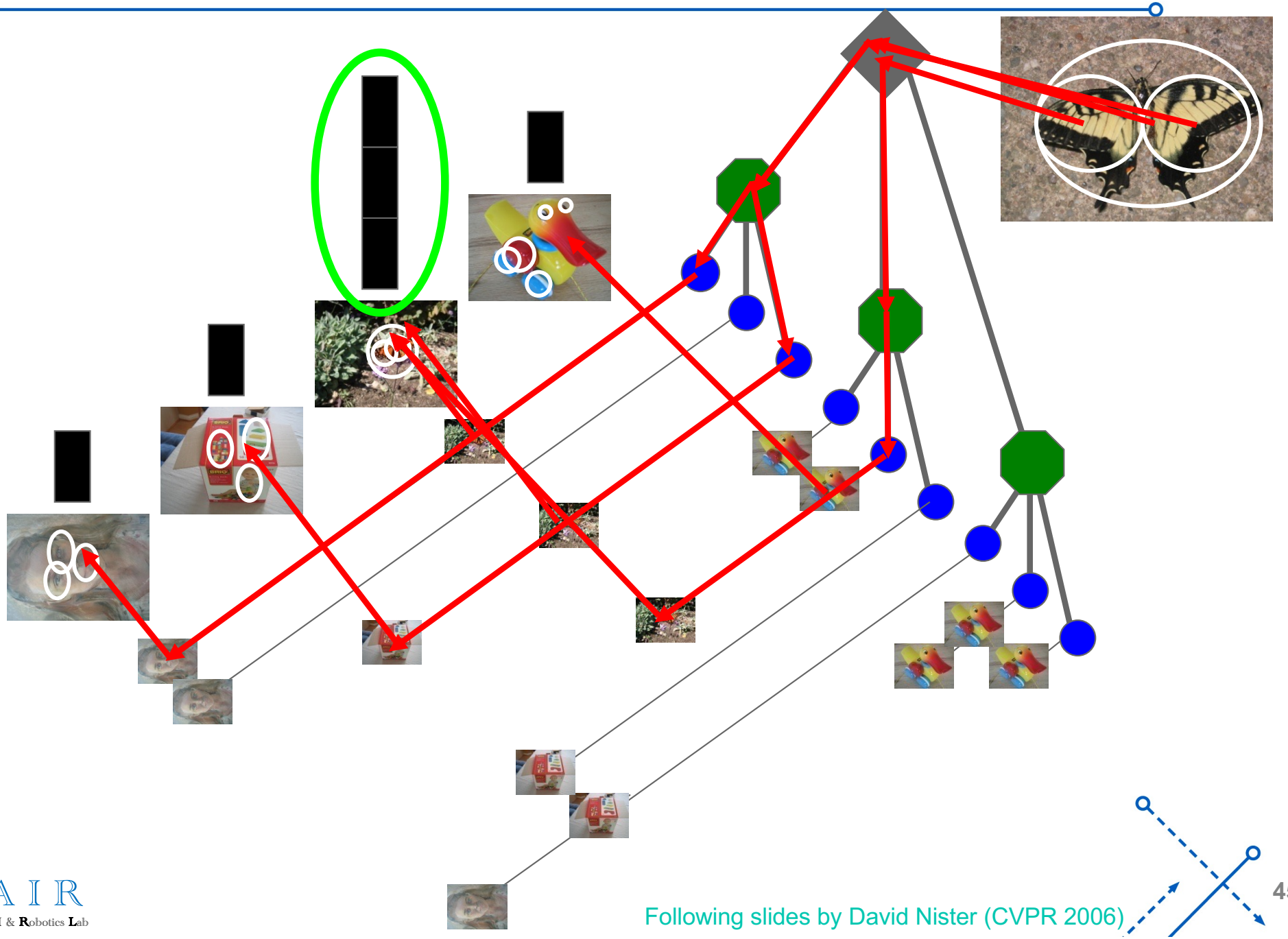
Following slides by David Nister (CVPR 2006)

Recognition with K-tree



Following slides by David Nister (CVPR 2006)

Recognition with K-tree



Following slides by David Nister (CVPR 2006)

Vocabulary Tree: Performance

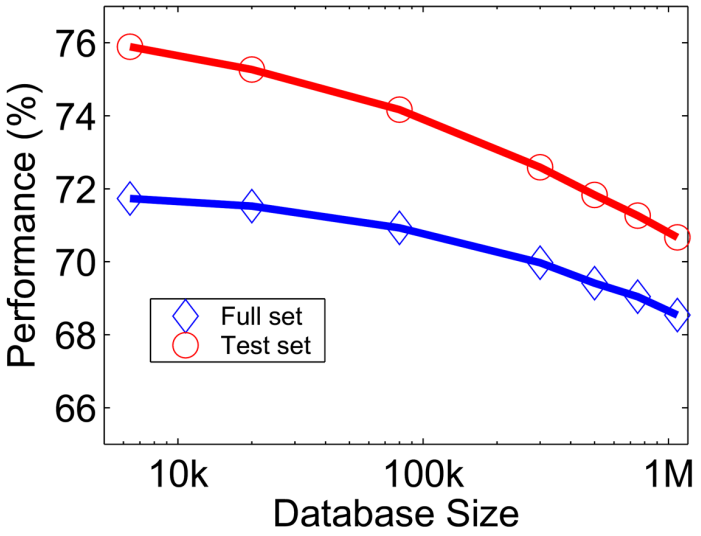
Evaluated on large databases

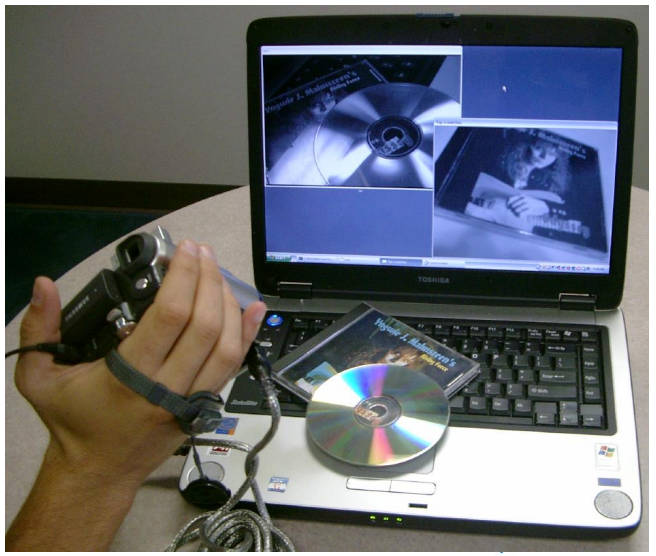
- Indexing with up to 1M images

Online recognition for database of 50,000 CD covers

- Retrieval in ~1s

Find experimentally that large vocabularies can be beneficial for recognition





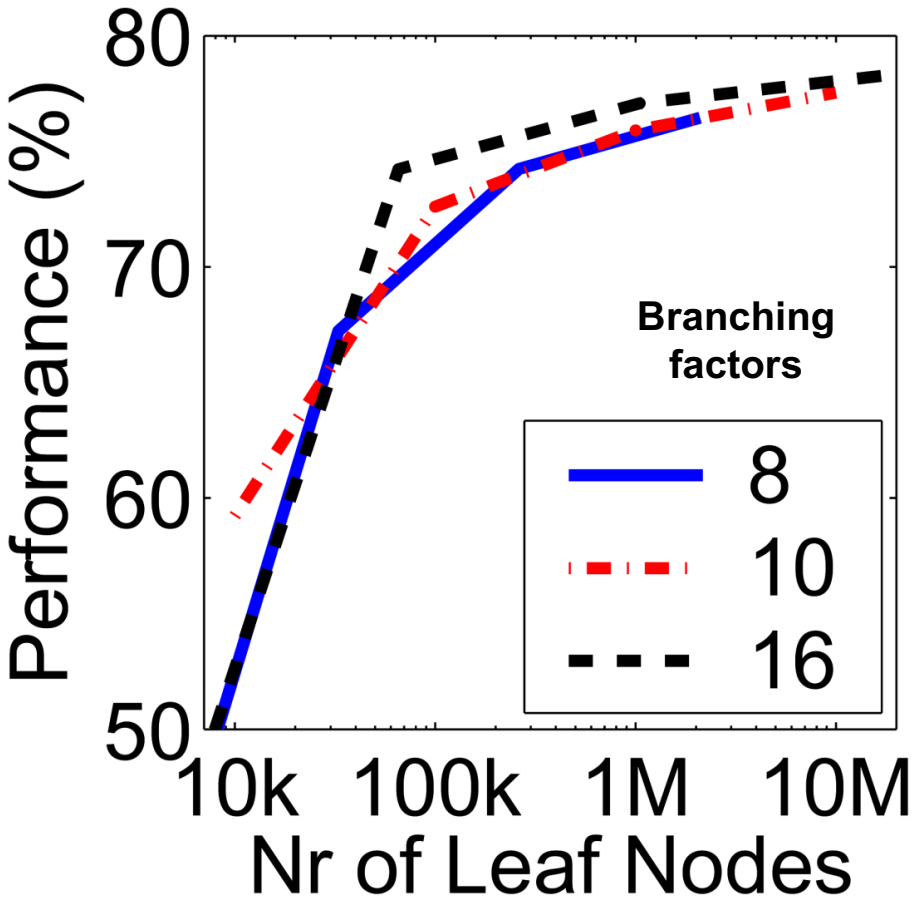
[Nister & Stewenius, CVPR'06]

Slide credit: J. Sivic

Instance recognition: remaining issues

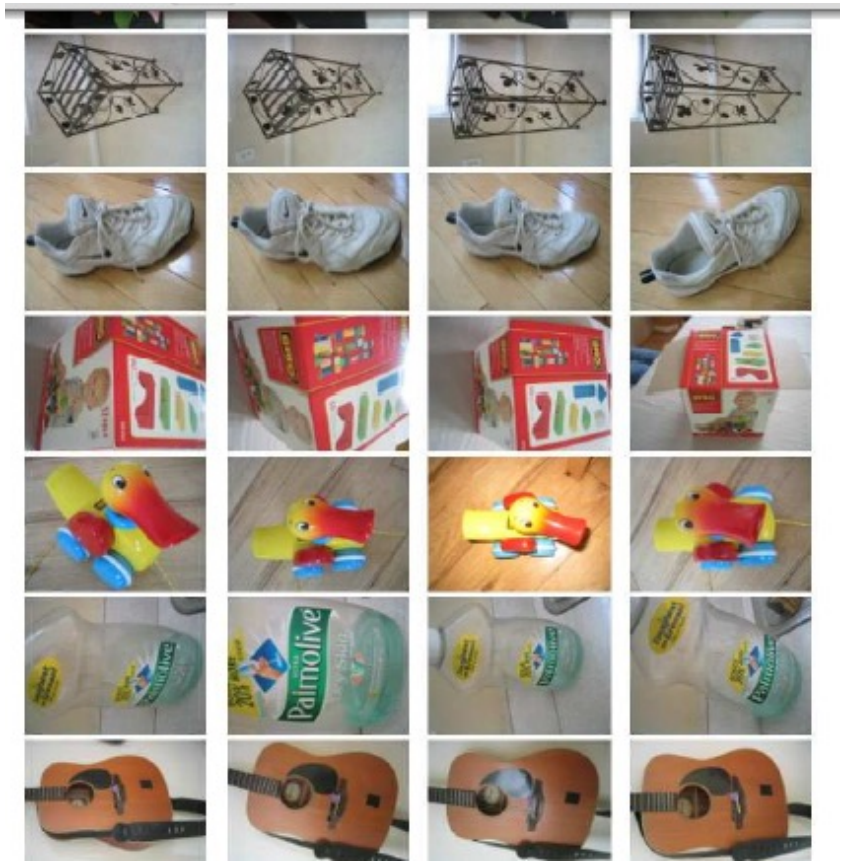
- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size



Influence on performance, sparsity

Results for recognition task with 6347 images



Nister & Stewenius, CVPR 2006

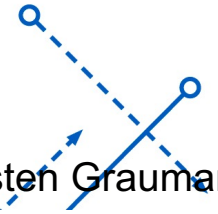
Visual words/bags of words

Pro

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides fixed dimensional vector representation for sets
+ good results in practice

Cons

- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

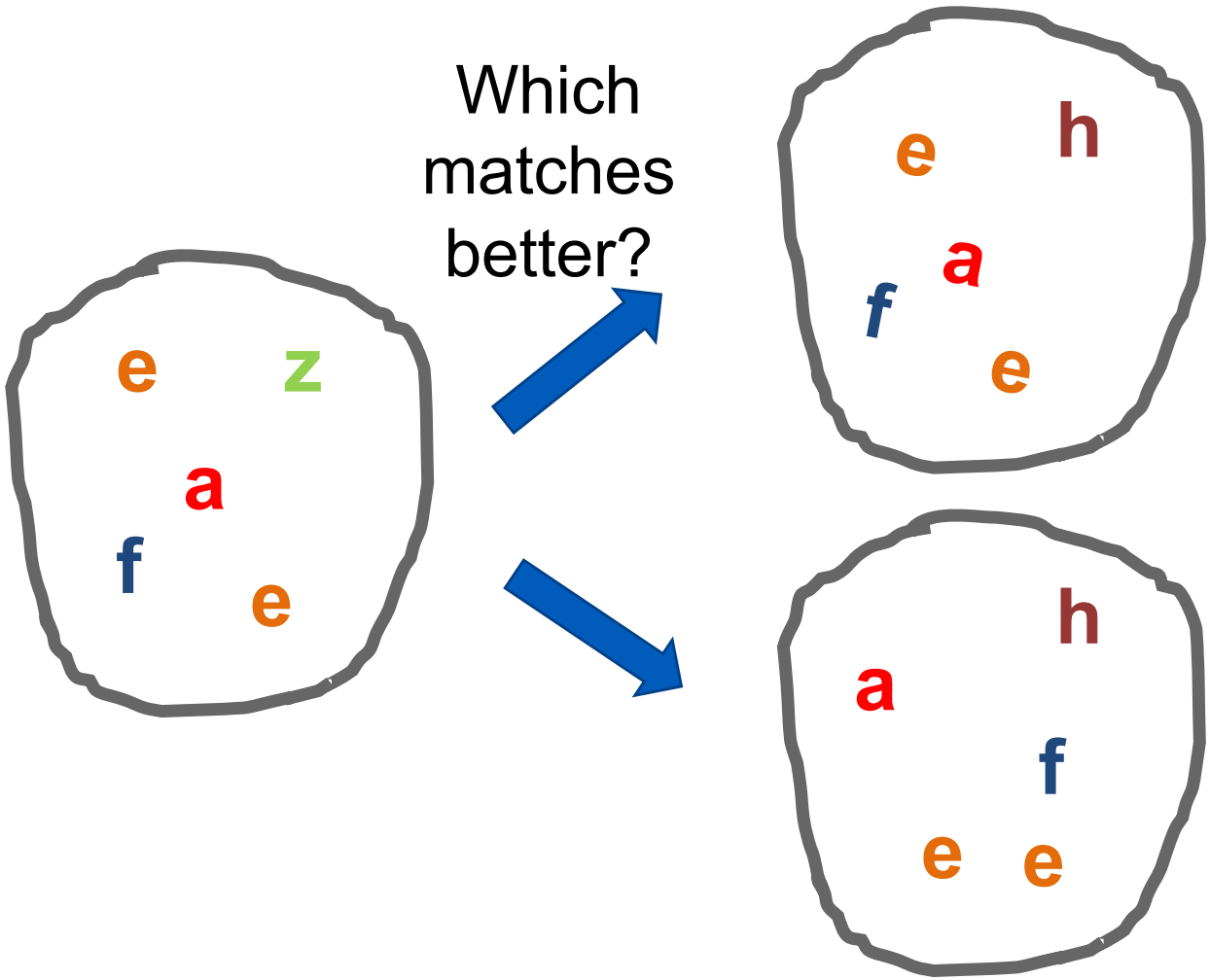


Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

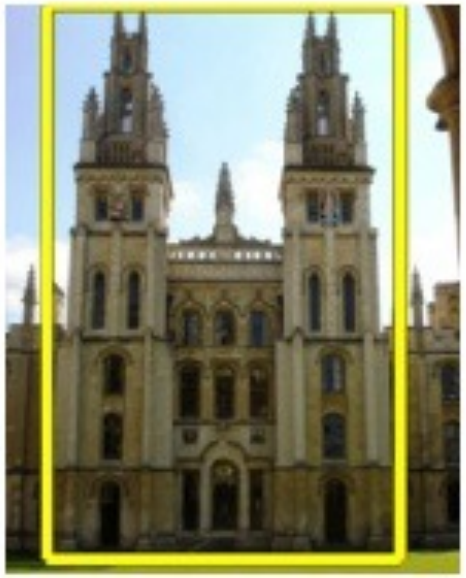
Can we be more accurate?

So far, we treat each image as containing a “bag of words”, with no spatial information



Can we be more accurate?

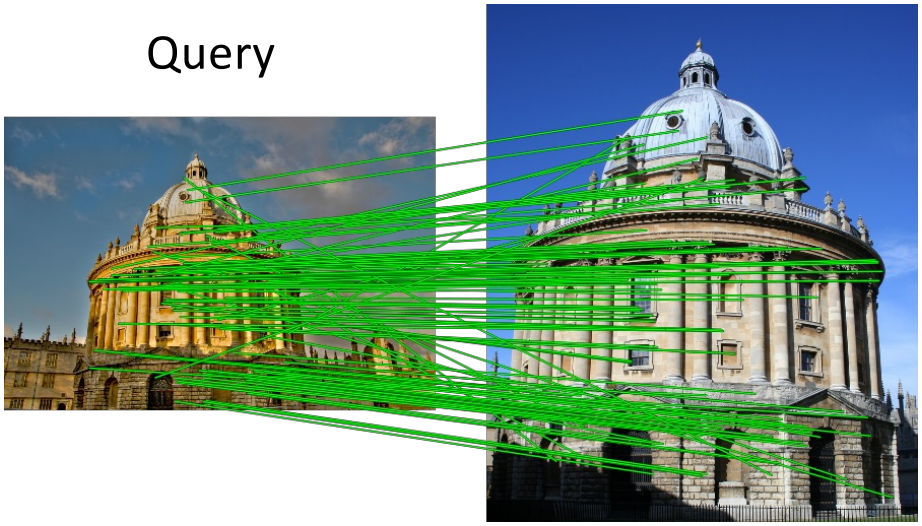
So far, we treat each image as containing a “bag of words”, with no spatial information



Real objects have consistent geometry

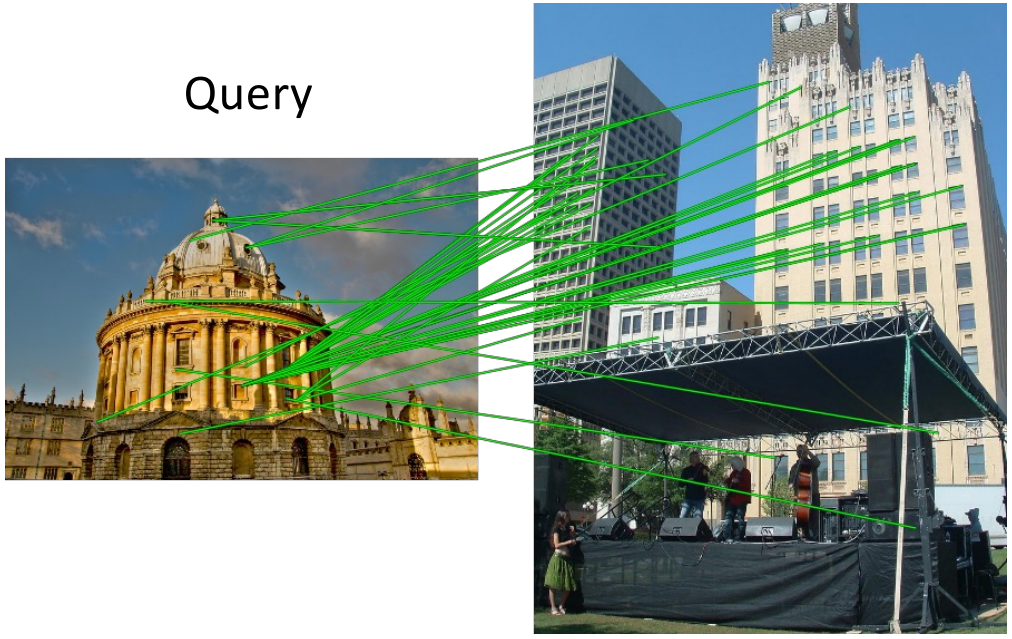
Spatial Verification

Query



DB image with high BoW similarity

Query



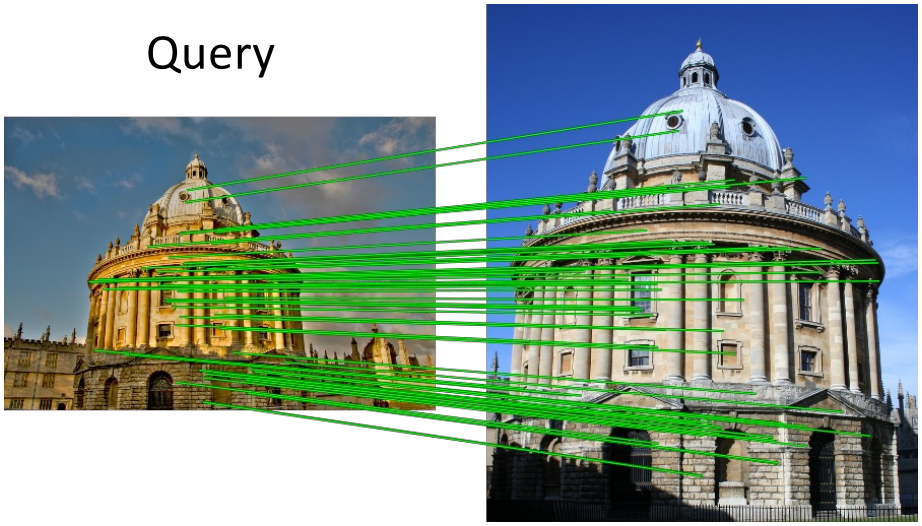
DB image with high BoW similarity

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum

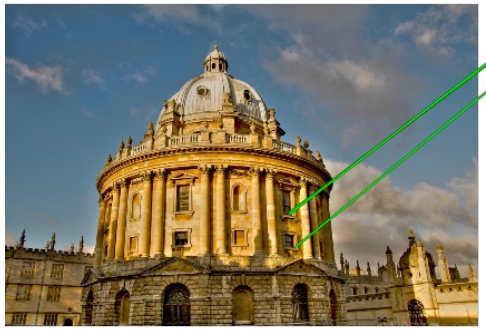
Spatial Verification

Query



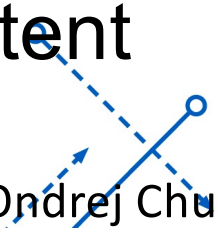
DB image with high BoW similarity

Query



DB image with high BoW similarity

Only some of the matches are mutually consistent



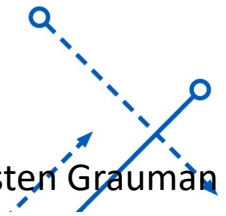
Slide credit: Ondrej Chum

Spatial Verification: three basic strategies

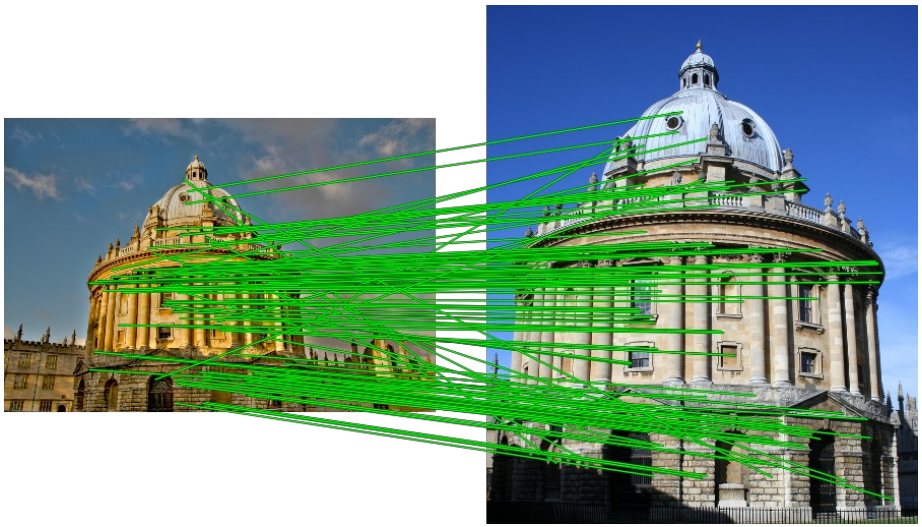
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., “success” if find a transformation with > N inlier correspondences

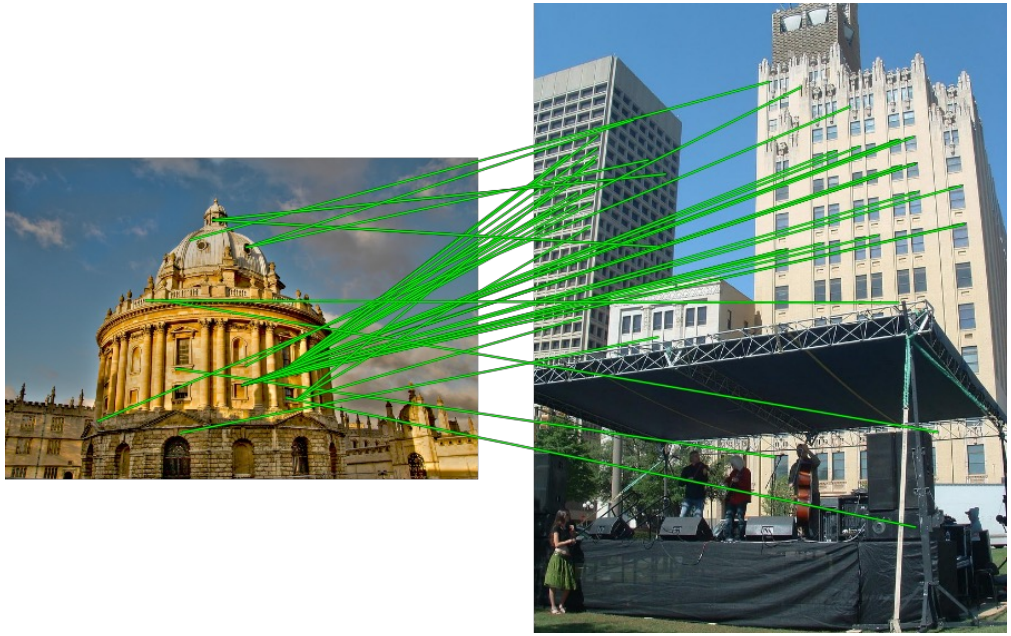
- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

- Triplet Verification

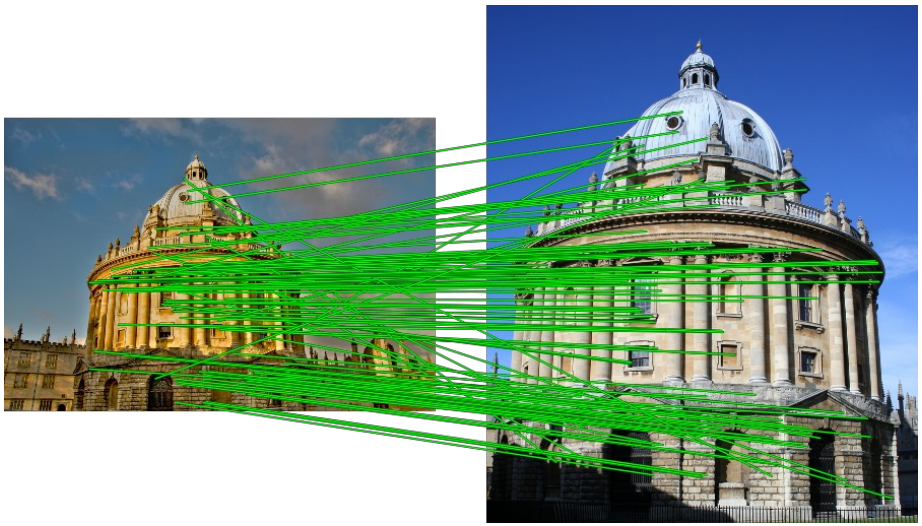


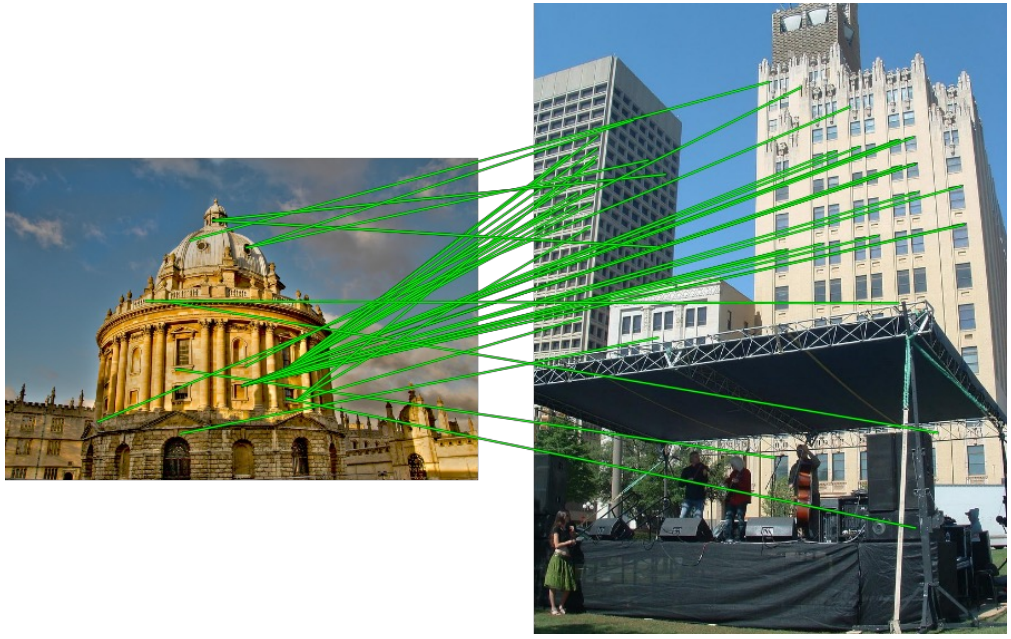
RANSAC verification

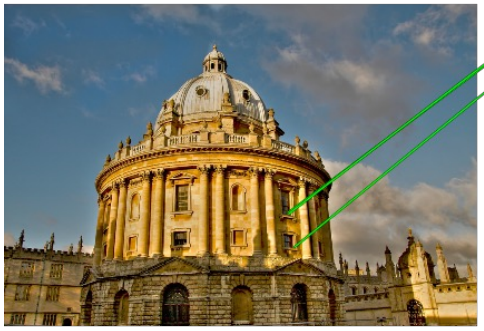




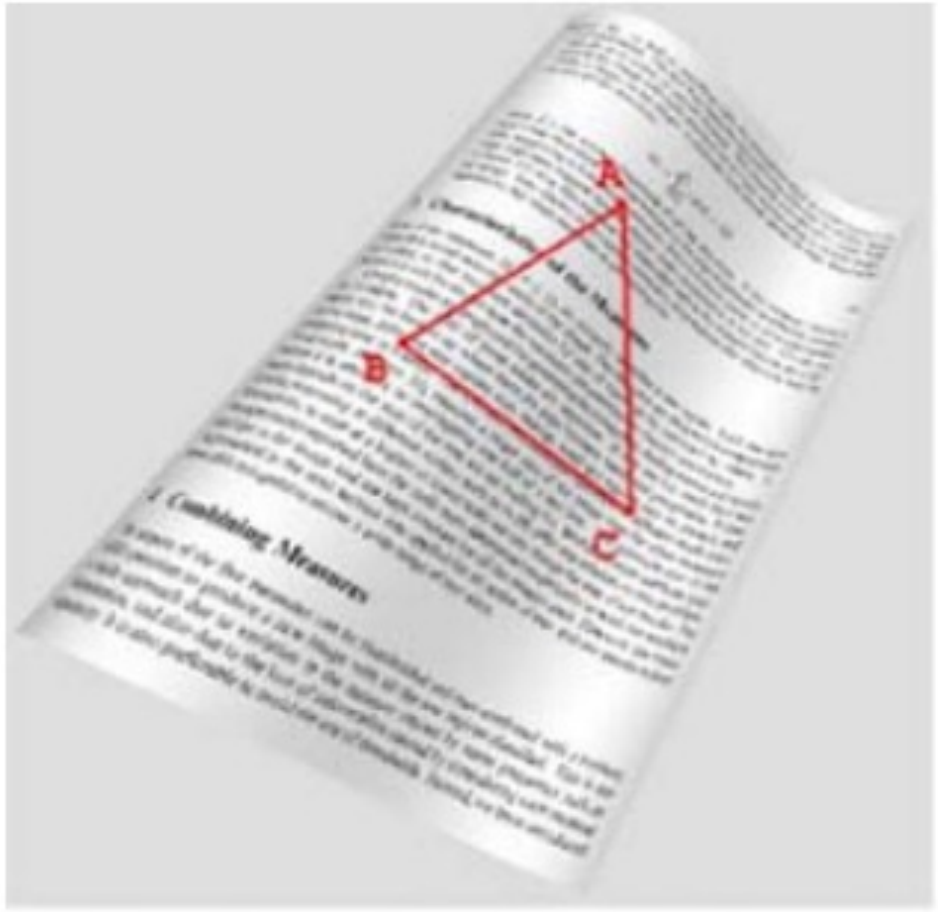
RANSAC

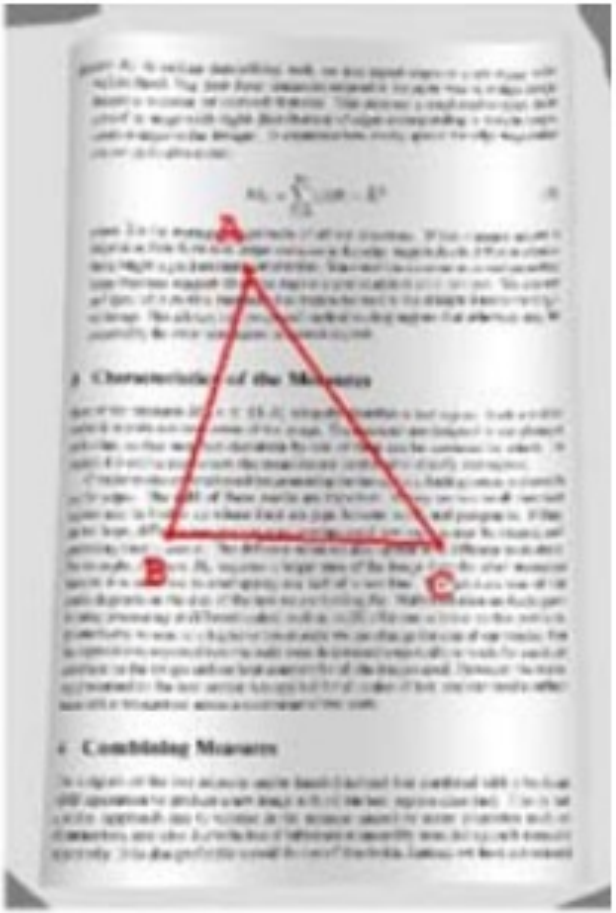






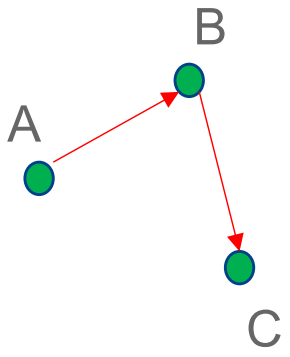
Triplet Verification

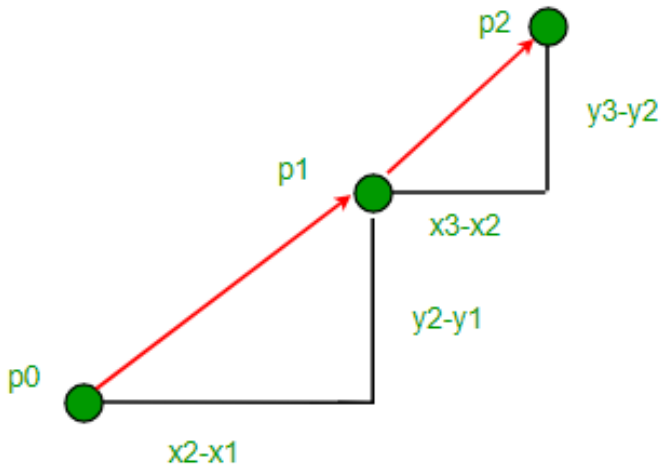




Using Slope to Determine Orientation

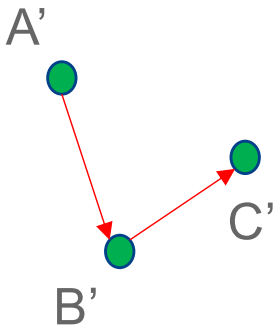
Consider the slopes….





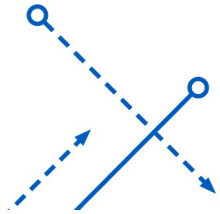
Slope(BC) – Slope(AB)?

< 0



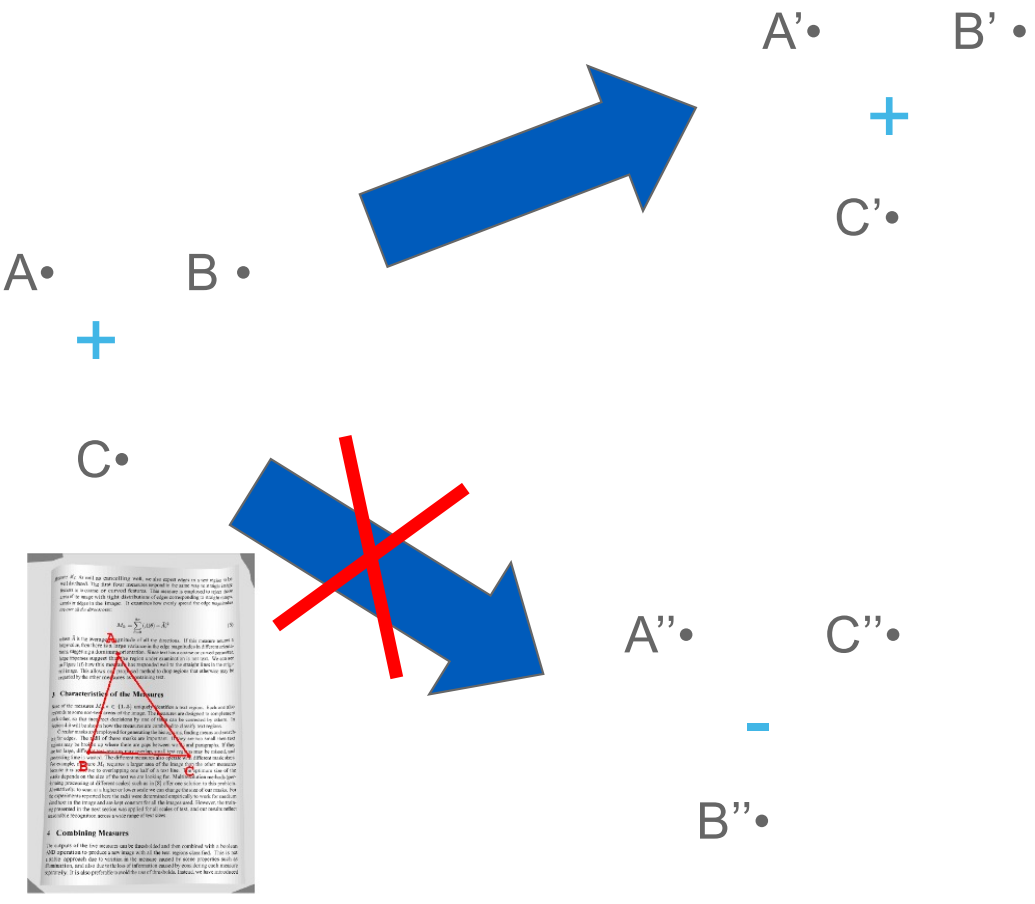
Slope(BC) – Slope(AB)?

> 0

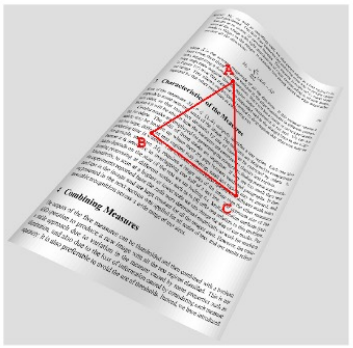


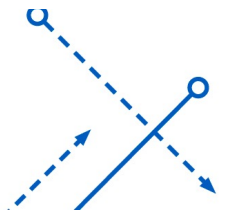
Triplet Verification

- Use the spatial relationship between “visual words”.
- Orientation of triplet is invariant under rotation, translation, scaling, warping and even crinkling



$$\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}$$





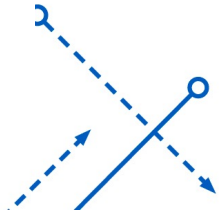
Triplet Verification: Scoring

- When viewed from another angle

$$Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \times Sign(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}) = 1$$

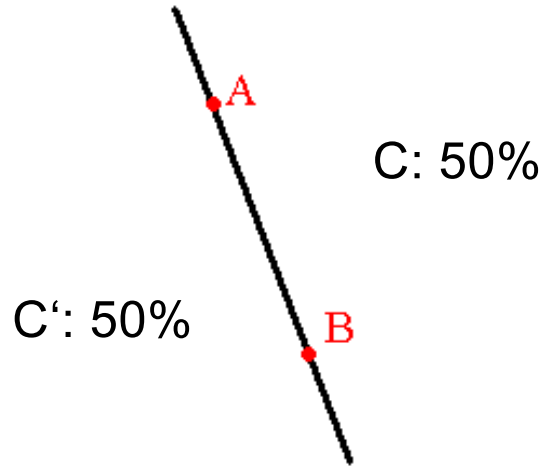
- Score is simply

$$\sum_{A,B,C \in S} (Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \times Sign(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}))$$



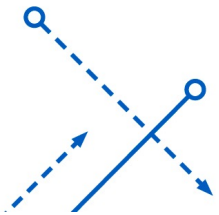
Triplet Verification: Failure rate

- Assume one triplet:
  - P(A,B,C clockwise) = 0.5
  - P(A,B,C counterclockwise) = 0.5
- Triplets are independent.
- Then possibility that M triplets out of N triplets accidentally satisfies orientation verification is



$$Q(N,M) = (\frac{1}{2})^N \binom{N}{M}$$

N	10	30	50	60
M	5	20	40	50
Q(N,M)	0.25	0.027	0.0001	6.5×10^{-8}

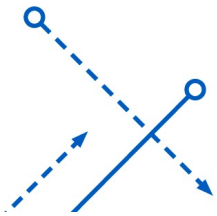


Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Precision and Recall

- Precision – How precise are the answers you gave?
  - (# Relevant)/(# Total Returned)
- Recall – How many did you find of the ones that could be found?
  - (# Relevant)/(# Total Relevant)
- F Measure – Harmonic Mean of Precision and Recall
  - (2*Precision*Recall)/(Precison+Recall)



Precision and Recall Curve (PR curve)

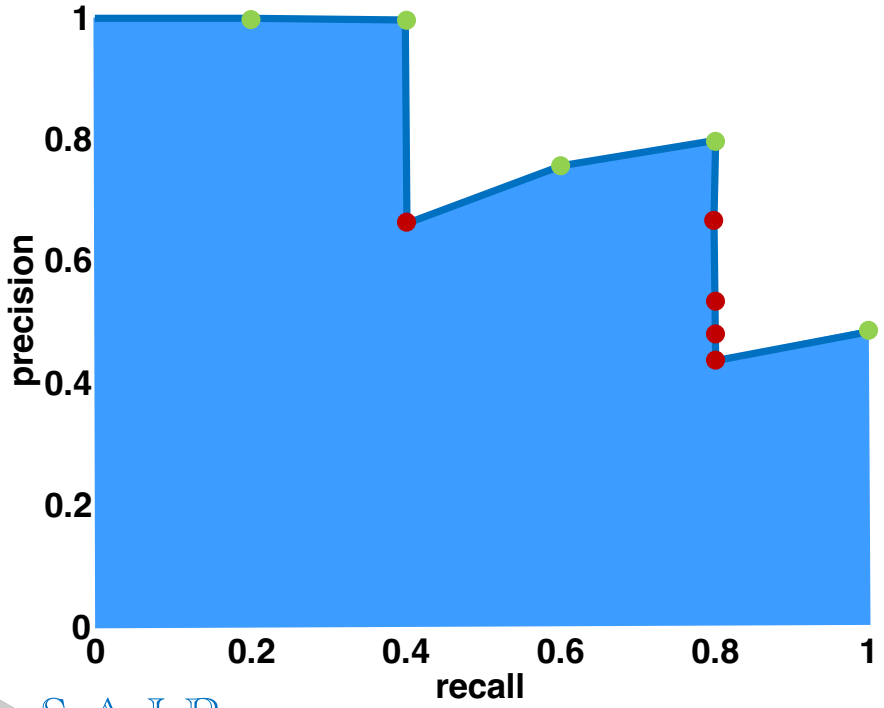
Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant

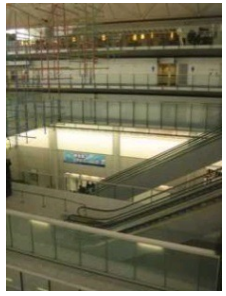
Query

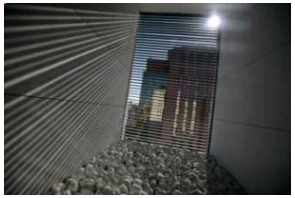
Results (ordered):









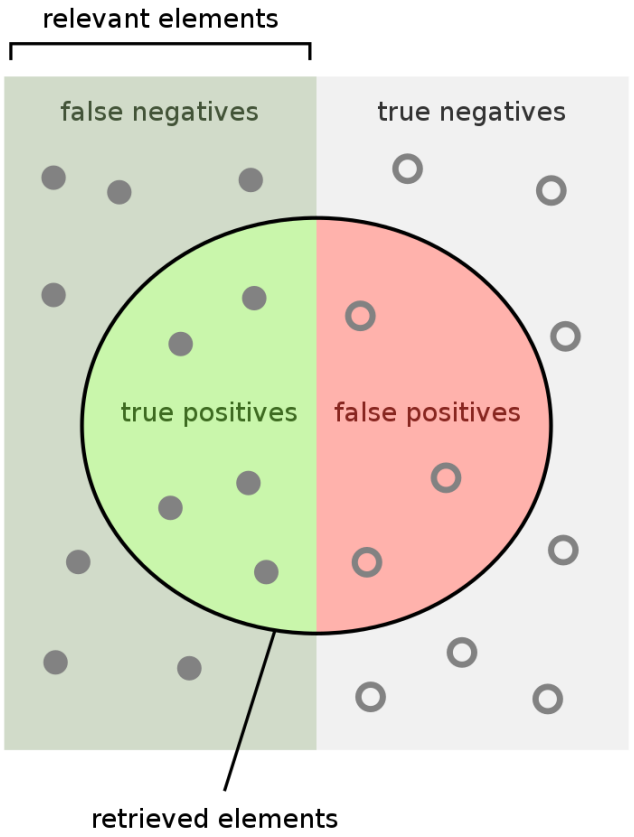

Slide credit: Ondrej Chum

Precision, Recall, F1 (Recap)

$$\text{Precision} = \frac{\text{True Positive}}{\text{Predicted Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Actual Positive}}$$

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



How many retrieved items are relevant?

Precision =

How many relevant items are retrieved?

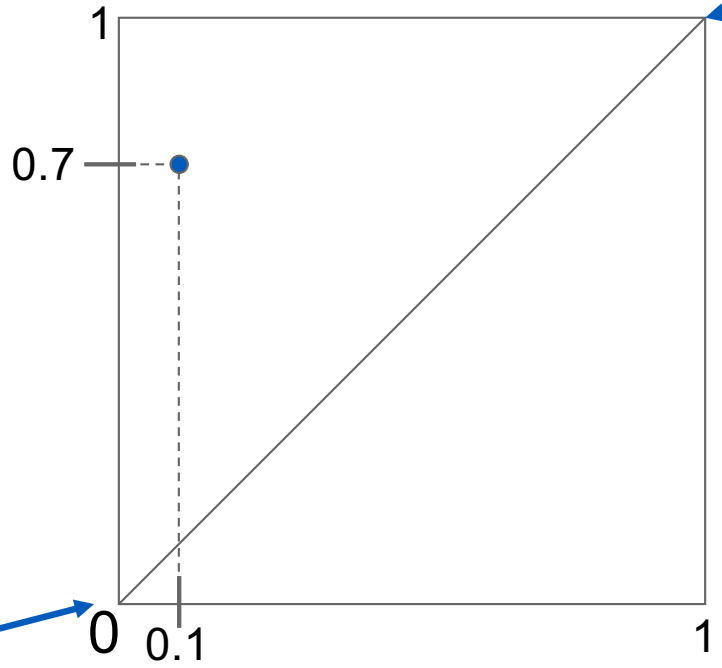
Recall =

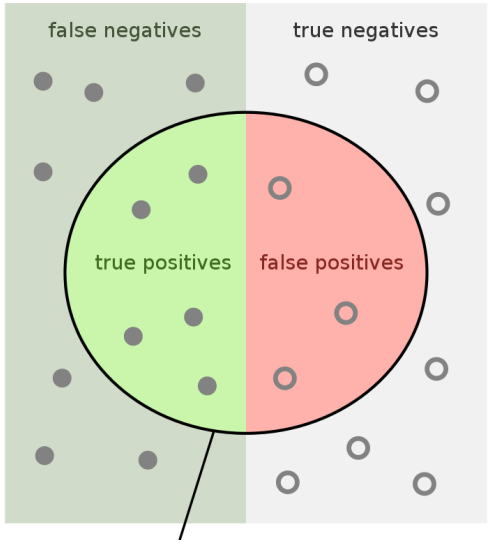
ROC curve (Recap)

- How can we measure the performance of a feature matcher?

Lowest threshold
all pairs as match

Recall = 





True Positive Rate

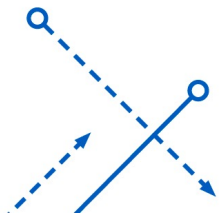
$$\frac{\#\,TP}{\#\,TP + FN}$$

Highest threshold
no match

False Positive Rate

$$\frac{\#\,FP}{\#\,FP + TN}$$

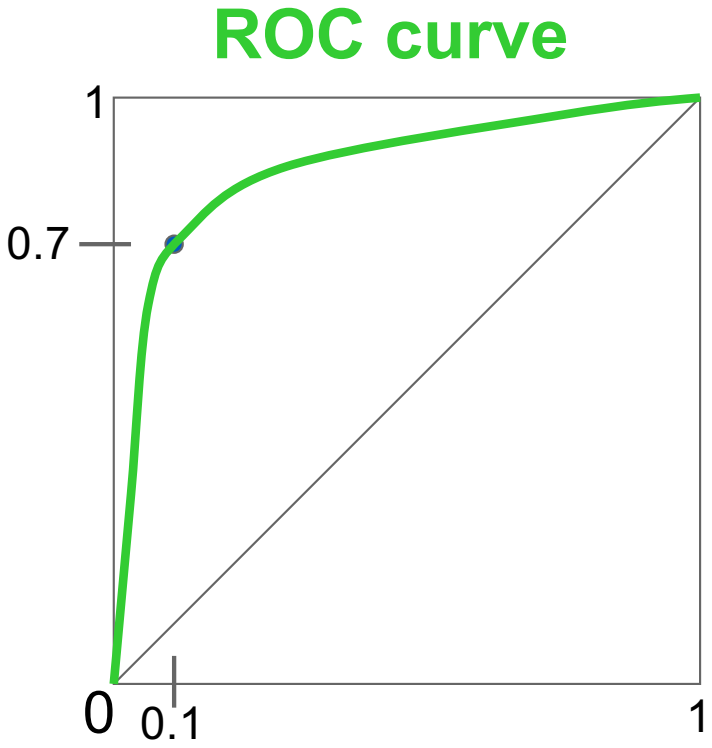
Fallout = 



ROC curve (Recap)

ROC Curves (Receiver Operator Characteristic)

- Generated by counting # correct/incorrect matches, for different thresholds.
- Want to maximize area under the curve (AUC)
- Useful for comparing different feature matching methods.



true positive rate

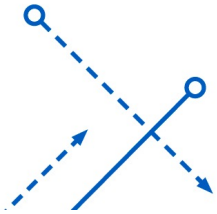
$$\frac{\#\ TP}{\#\ TP + FN}$$

false positive rate

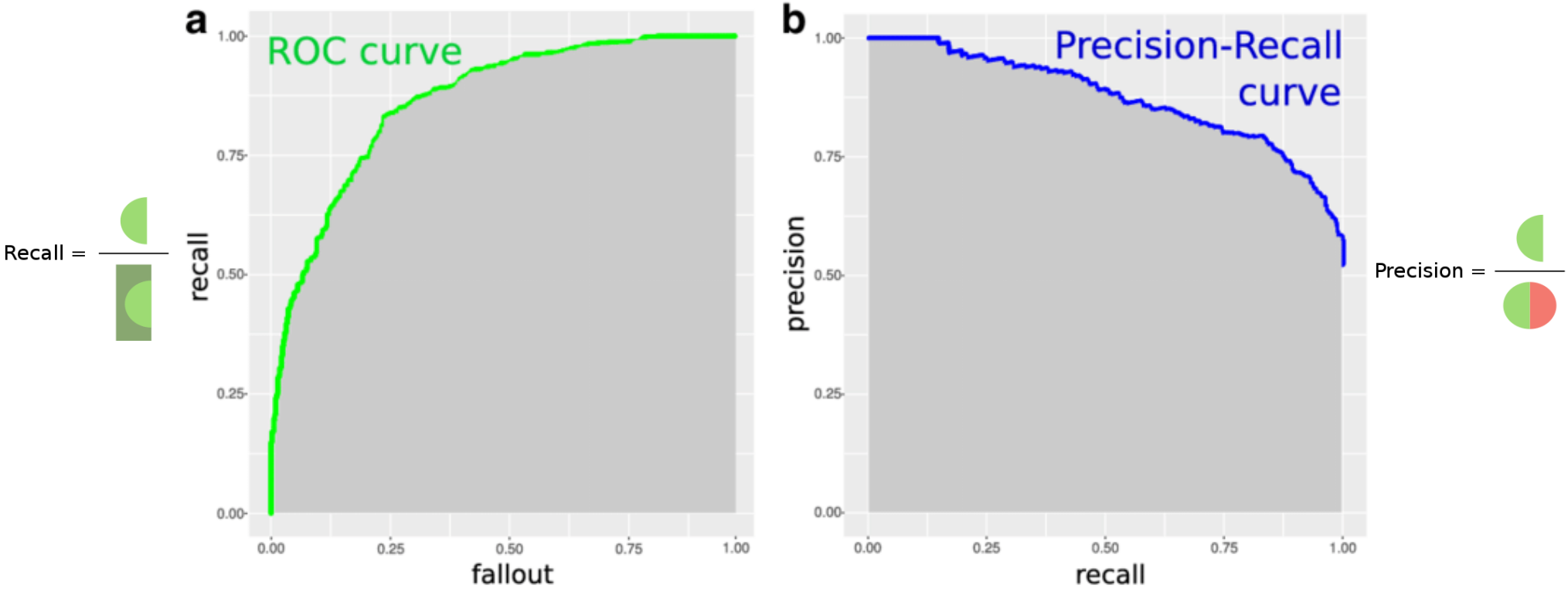
$$\frac{\#\ FP}{\#\ FP + TN}$$

ROC curve VS. PR curve

- ROC Curves (Lecture 4)
  - summarize the trade-off between the **true positive rate** and **false positive rate** for a predictive model using different probability thresholds.
- PR curves
  - summarize the trade-off between the **true positive rate** and the **positive predictive value** for a predictive model using different probability thresholds.



ROC curve VS. PR curve



Recall =

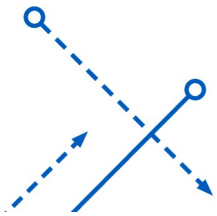
Precision =

Fallout =

Recall =

ROC curve VS. PR curve

- ROC: appropriate for balanced datasets/classes.
- PR curves: appropriate for imbalanced datasets.
- If the proportion of positive to negative instances changes in a test set, the ROC curves will not change, so do not depend on class distributions.
- ROC “weights” equally true positives and true negatives (not expected sometimes, e.g., place recognition).



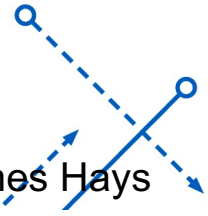
More explanation

Summary

- **Matching local invariant features**
  - Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
  - Summarize image by distribution of words
  - Index individual words
- **Inverted index**: pre-compute index to enable faster search at query time
- **Recognition of instances via alignment:** matching local features followed by spatial verification
  - Robust fitting : RANSAC, GHT

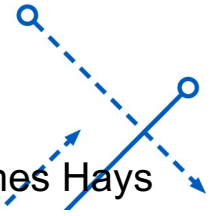
Lessons from a Decade Later

- For *Category* recognition
  - Bag of Feature models remained the state of the art until Deep Learning.
  - Spatial layout either isn't that important or its too difficult to encode.
  - Quantization error is, in fact, the bigger problem. Advanced feature encoding methods address this.
  - Bag of feature models are nearly obsolete. At best they seem to be inspiring tweaks to deep models, e.g., NetVLAD (deep learning model for place recognition).



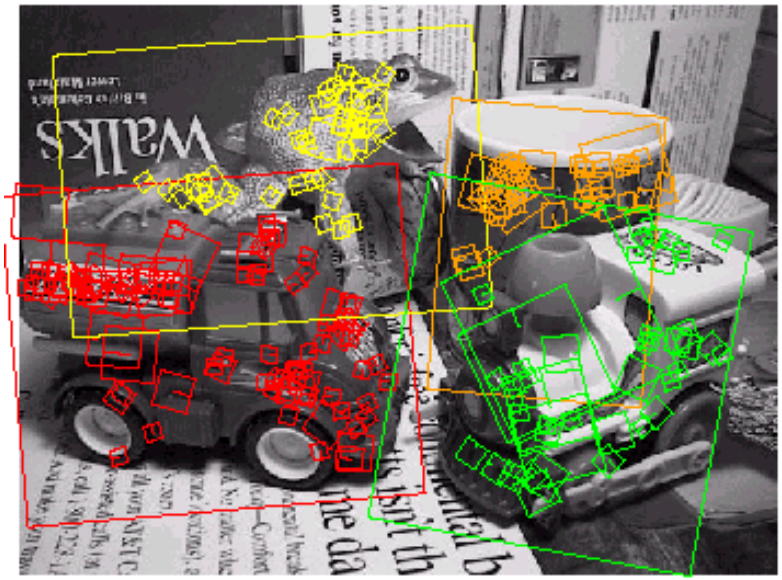
Lessons from a Decade Later

- For *instance* retrieval (this lecture)
  - Deep learning has taking over except in the field of SLAM.
  - Learn better local features (replace SIFT), e.g., MatchNet
  - or learn better image embeddings (replace the histograms of visual features), e.g., Vo and Hays 2016.
  - or learn to do spatial verification, e.g., DeTone, Malisiewicz, and Rabinovich 2016.
  - or learn a monolithic deep network to recognition all locations, e.g., Google's PlaNet 2016.

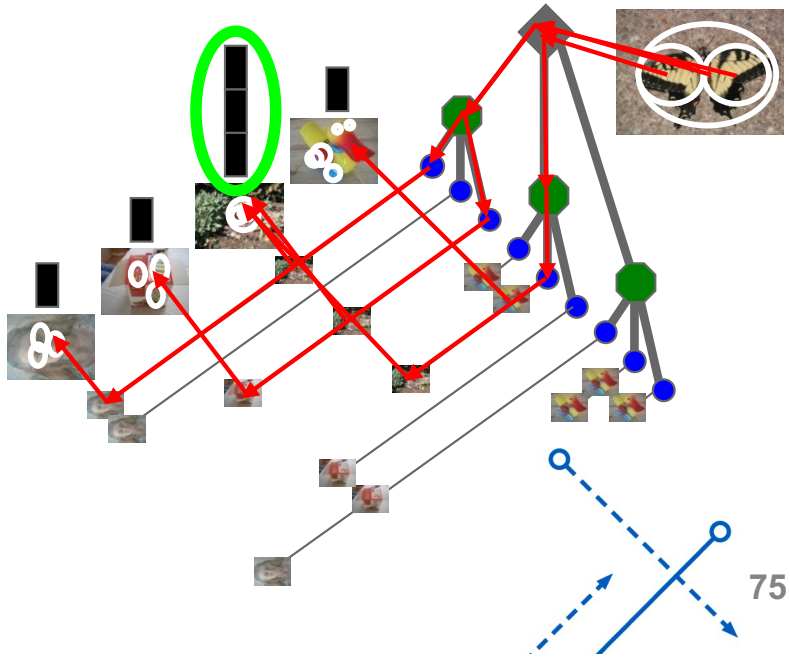


Things to remember

- Object instance recognition
  - Find keypoints, compute descriptors
  - Match descriptors
  - Vote for / fit affine parameters
  - Return object if # inliers > T



- Keys to efficiency
  - Visual words
    - Used for many applications
  - Inverse document file
    - Used for web-scale search



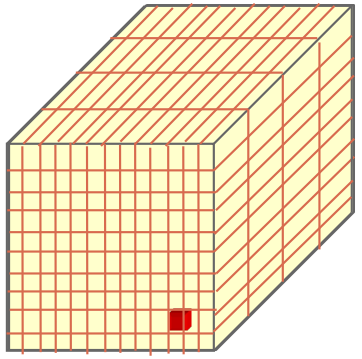
SAIR

Spatial AI & Robotics Lab

COMPUTER VISION

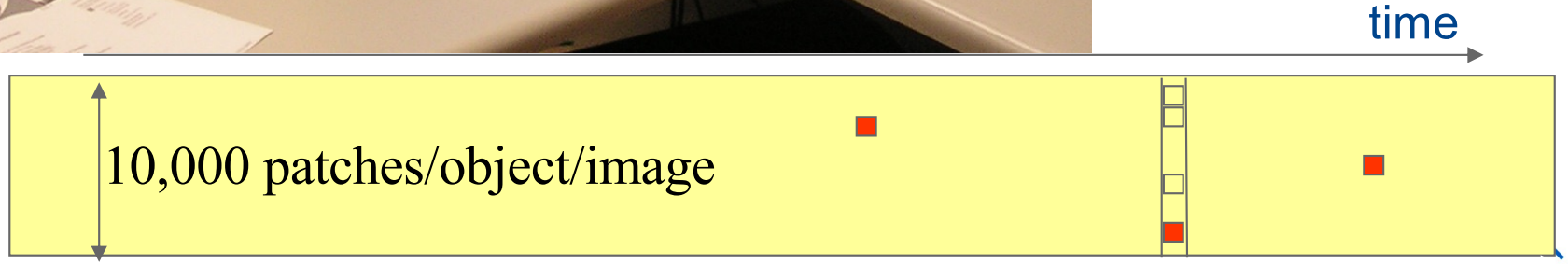
Object Detection

Why is detection hard?



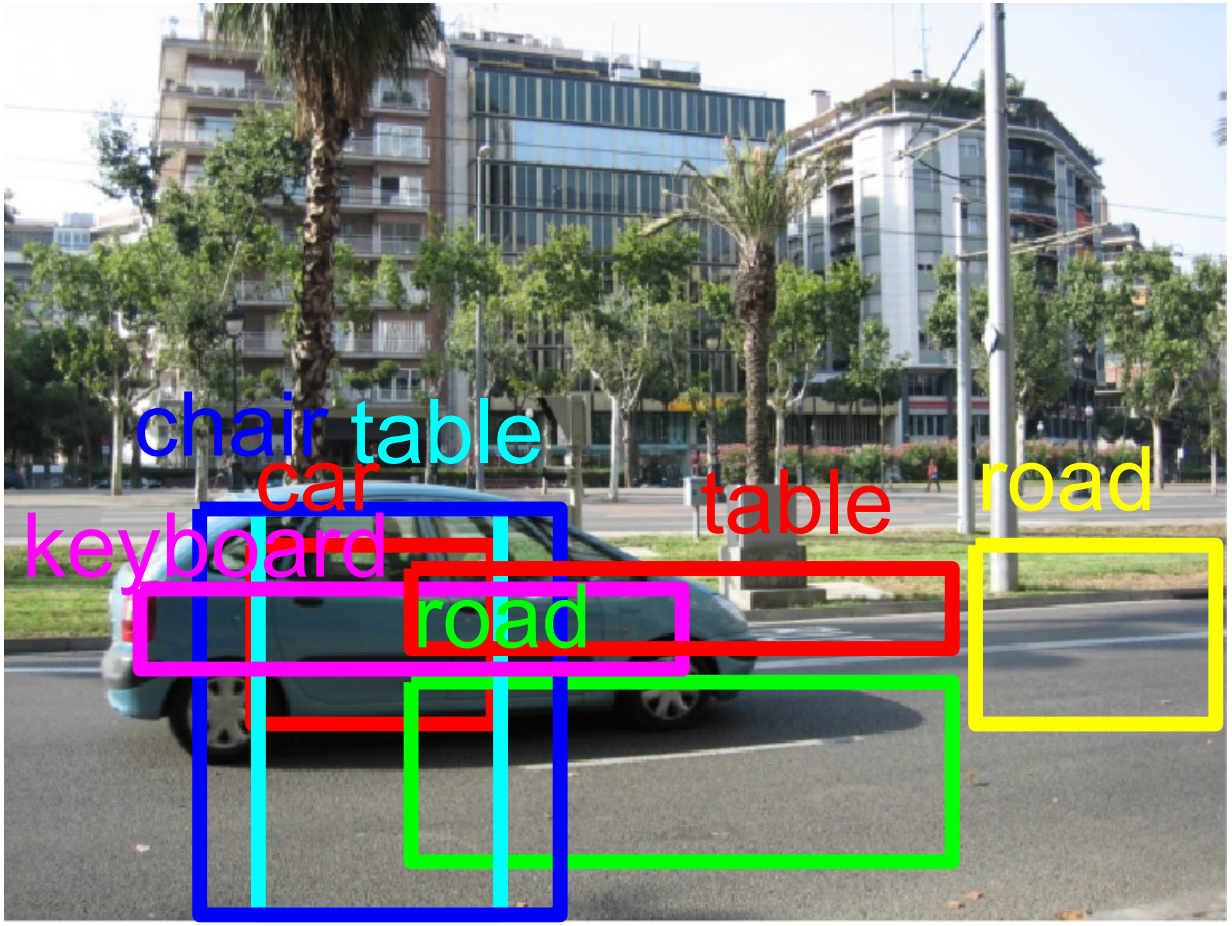
We want to do this for ~ 1000 objects

1,000,000 images/day



Why is detection hard?

If we have 1000 categories (detectors), and each detector produces 1 false every 10 images, we will have 100 false alarms per image… pretty much garbage…

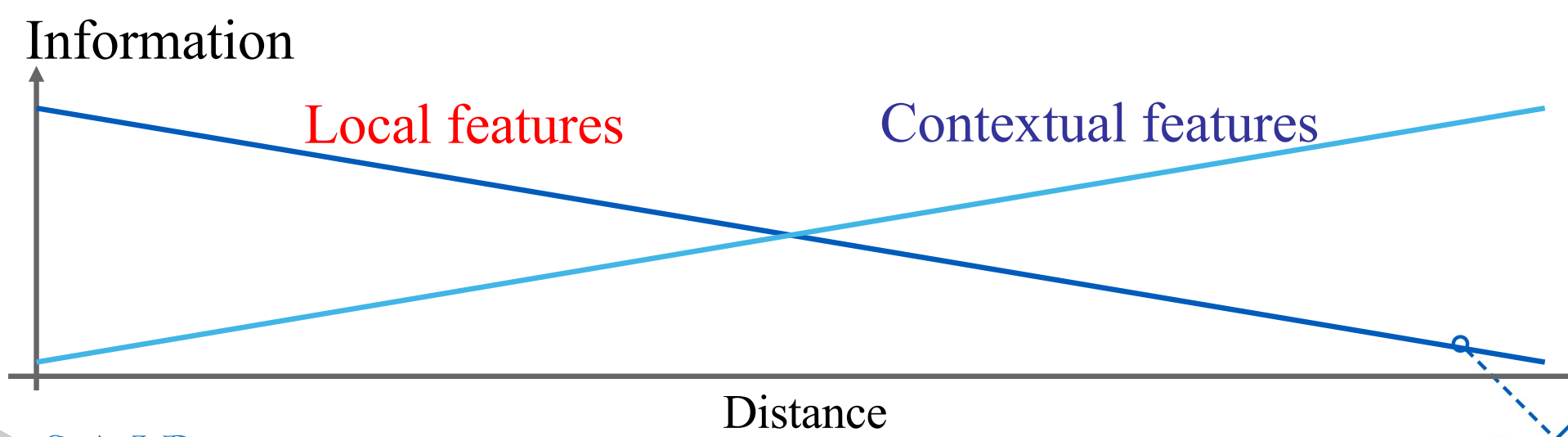


Slide credit: A. Torralba

Local information is helpful

Slide credit: A. Torralba

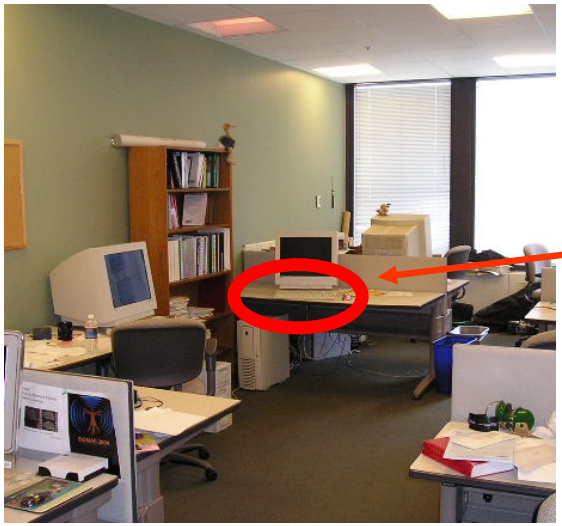
Is local information enough?



Slide credit: A. Torralba

Is local information even enough?

We know there is a keyboard present in this scene even if we cannot see it clearly.

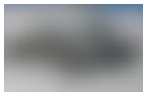


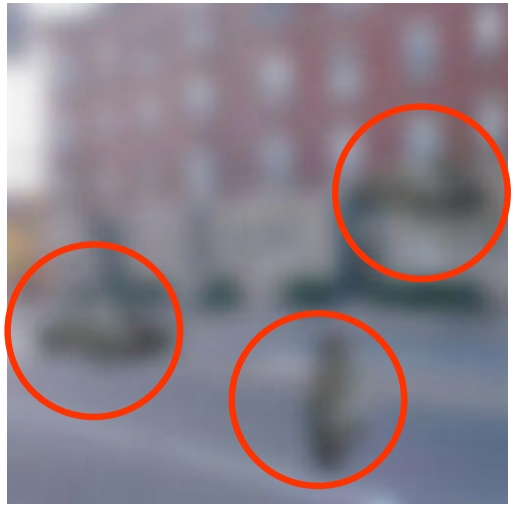
We know there is no keyboard present in this scene

… even if there is one indeed.

Slide credit: A. Torralba

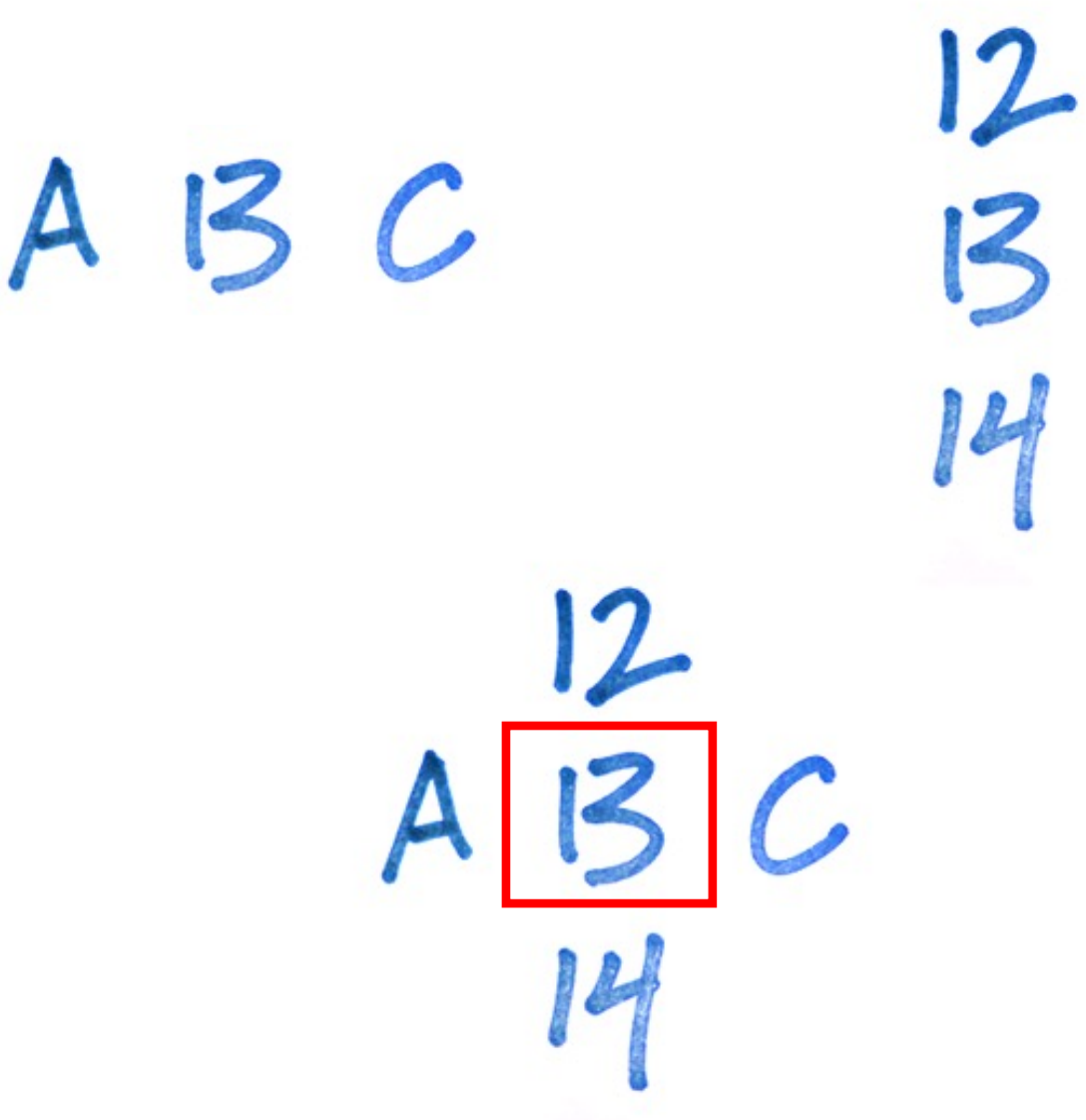
The multiple personalities of a blob





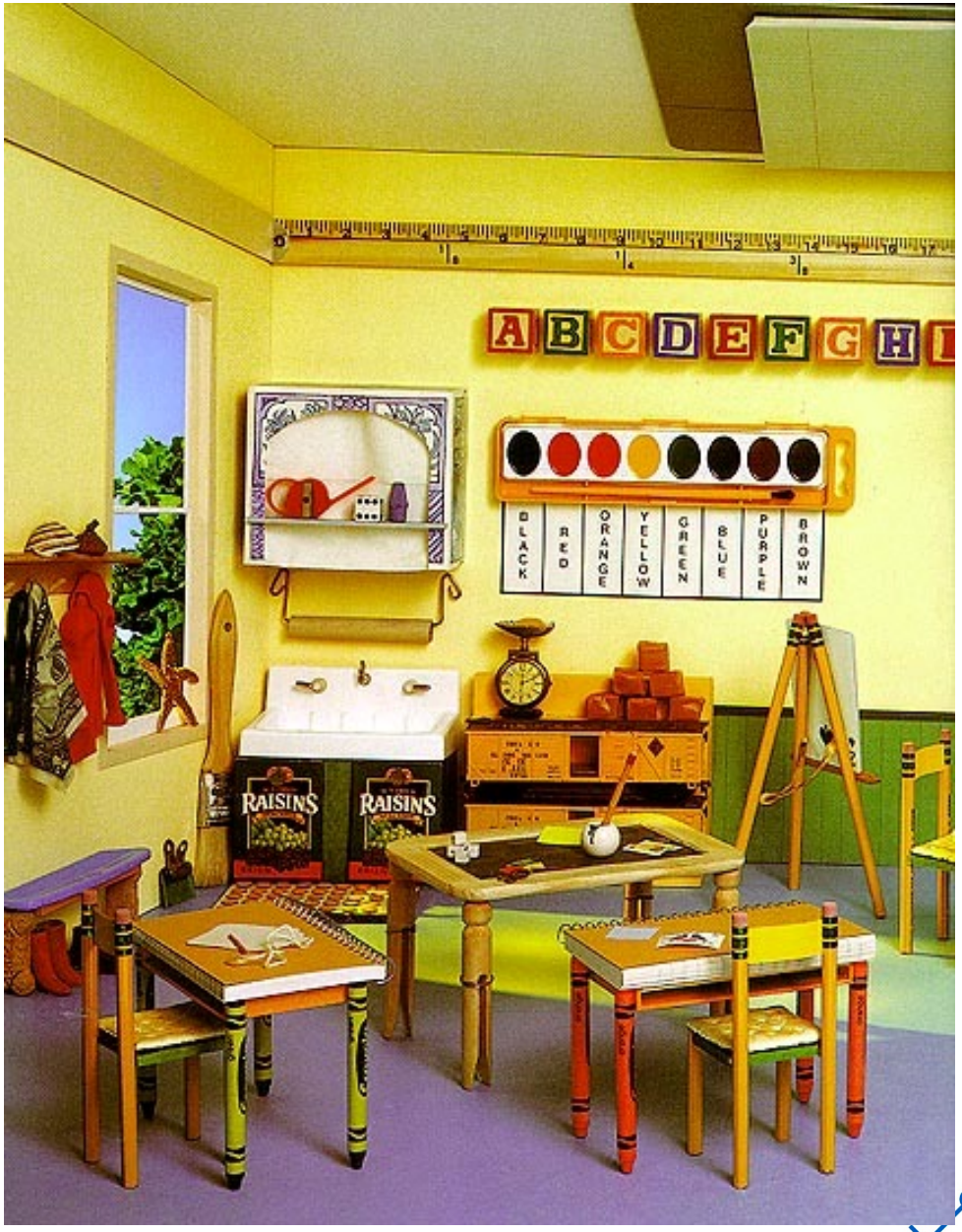
Slide credit: A. Torralba

The multiple personalities of a blob



Slide credit: A. Torralba

Look-Alikes by Joan Steiner



Slide credit: A. Torralba

The context challenge

- What are the hidden objects?

Slide credit: A. Torralba

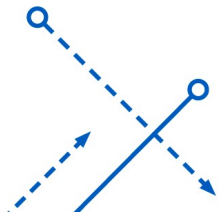
The context challenge

- What are the hidden objects?

Slide credit: A. Torralba

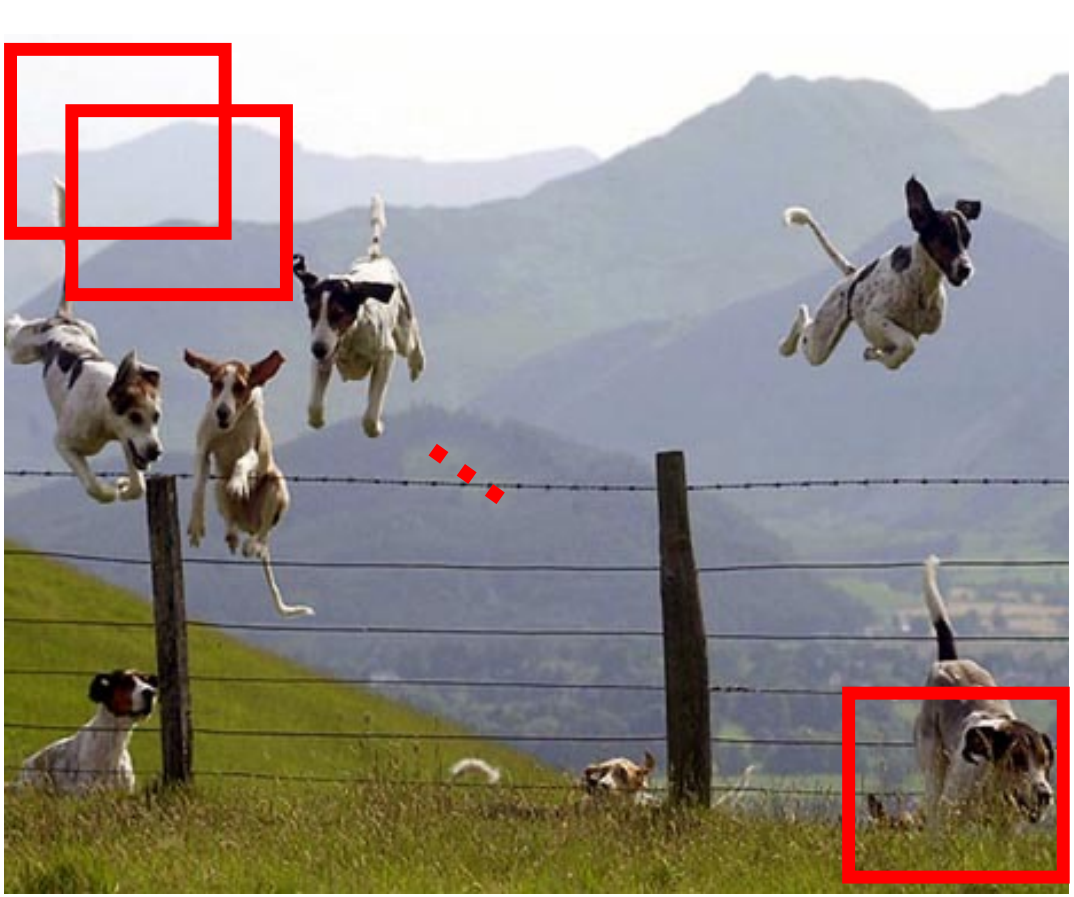
Object detection vs Scene Recognition

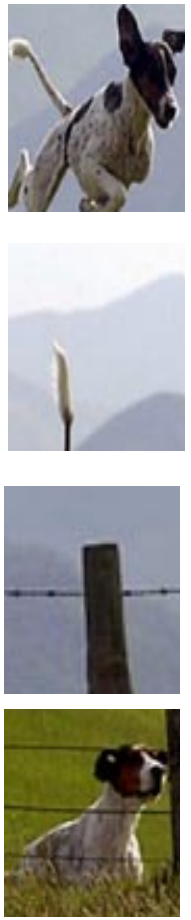
- Objects (even if deformable and articulated) probably have more **consistent shapes** than scenes.
- Scenes can be defined by distribution of “stuff” – materials and surfaces with **arbitrary shape**.
- Objects are **“things” that own their boundaries**
- Bag of words (BoW) were less popular for object detection because they **throw away shape** info.

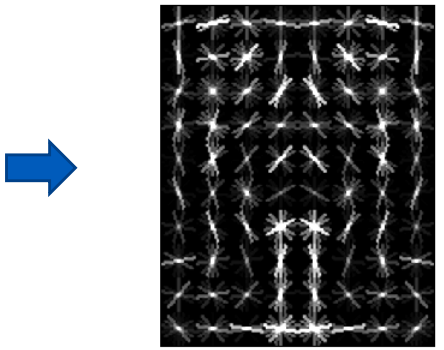


Object Category Detection

- Focus on object search: “Where is it?”
- Build templates that quickly differentiate object patch from background patch







Object or
Non-Object?

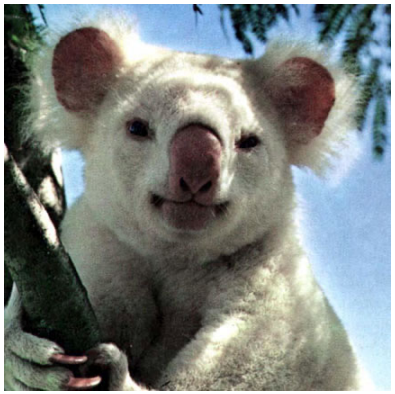
Challenges in modeling the object class

Illumination

Object pose

Clutter

Occlusions



Intra-class appearance

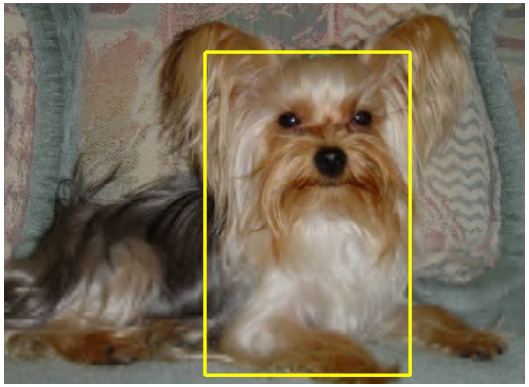
Viewpoint

Slide from K. Grauman, B. Leibe

Challenges in modeling the non-object class

True Detections

Bad Localization

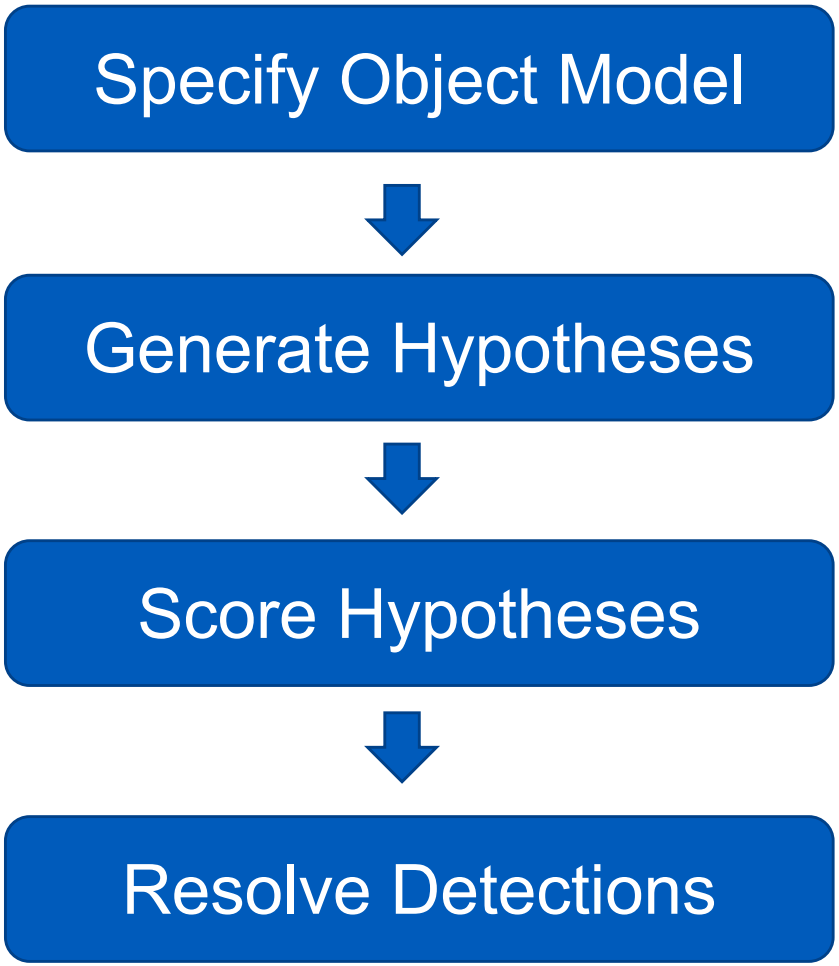


Confused with Similar Object

Misc. Background

Confused with Dissimilar Objects

General Process of Object Recognition

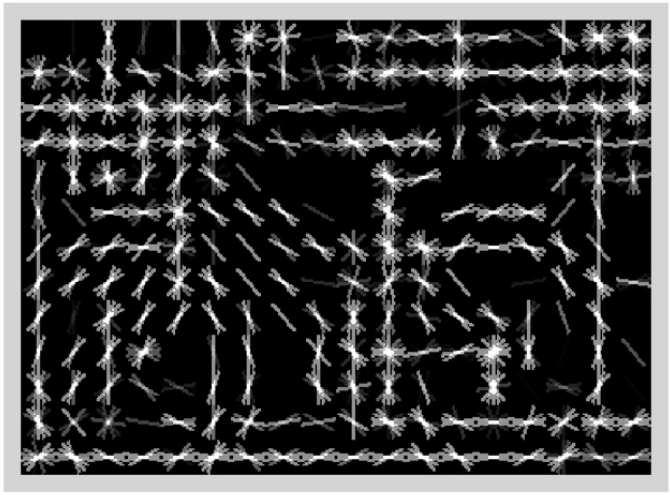


What are the object parameters?

Specifying an object model

1. Statistical Template in Bounding Box
   - Object is somewhere (x,y,w,h) in image
   - Features defined wrt bounding box coordinates

Image

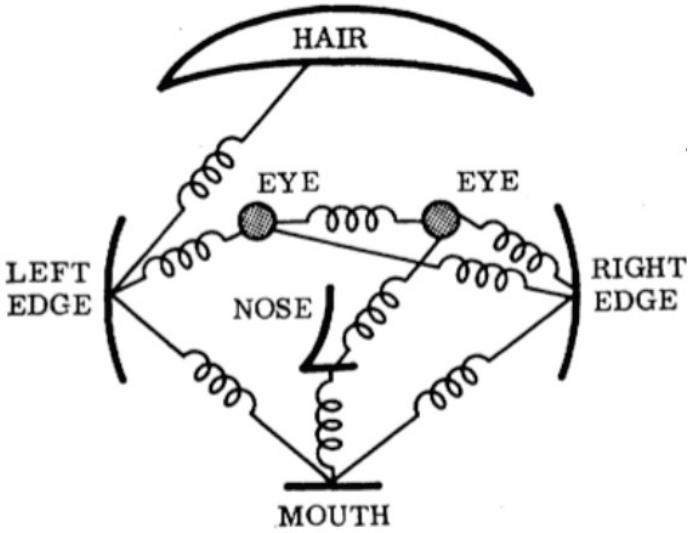


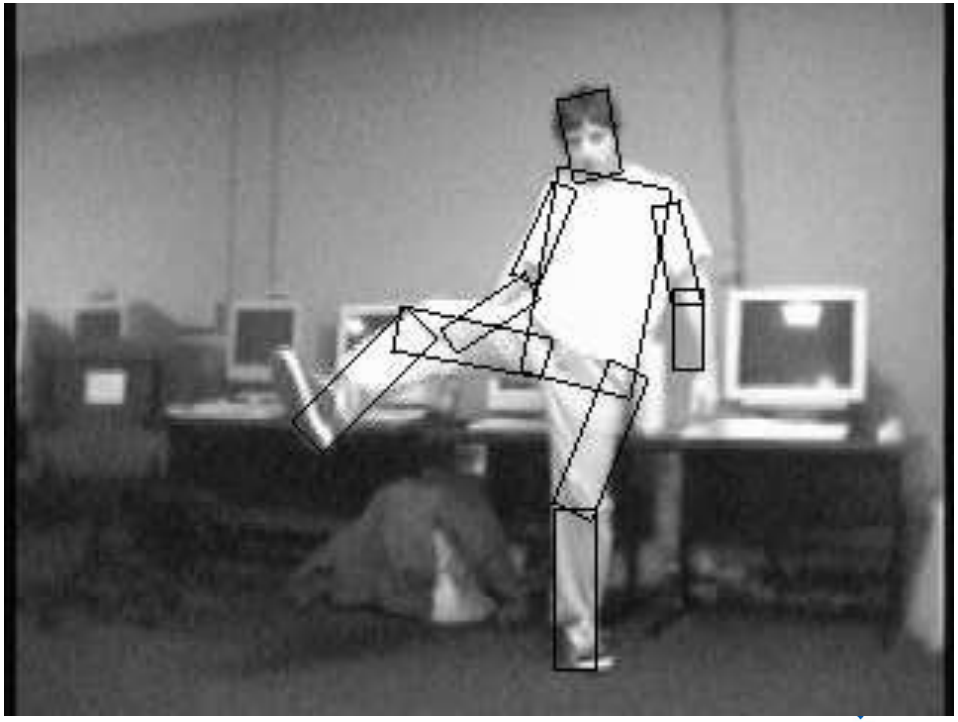
Template Visualization
(HoG features)

Images from Felzenszwalb

Specifying an object model

2. Articulated parts model
   - Object is configuration of parts
   - Each part is detectable



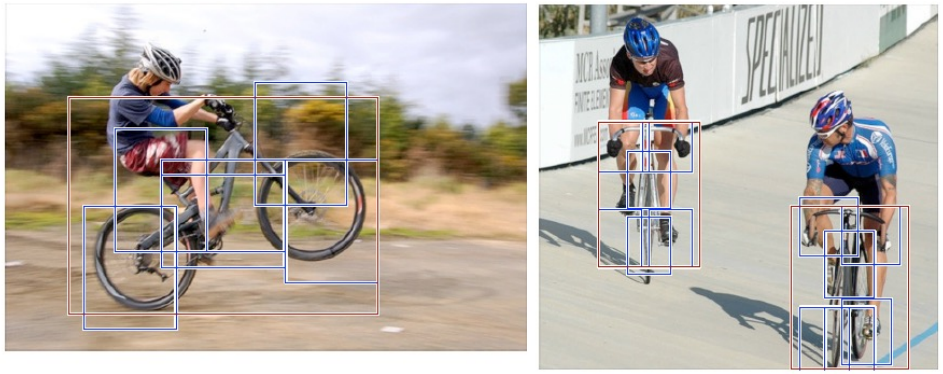


Images from Felzenszwalb

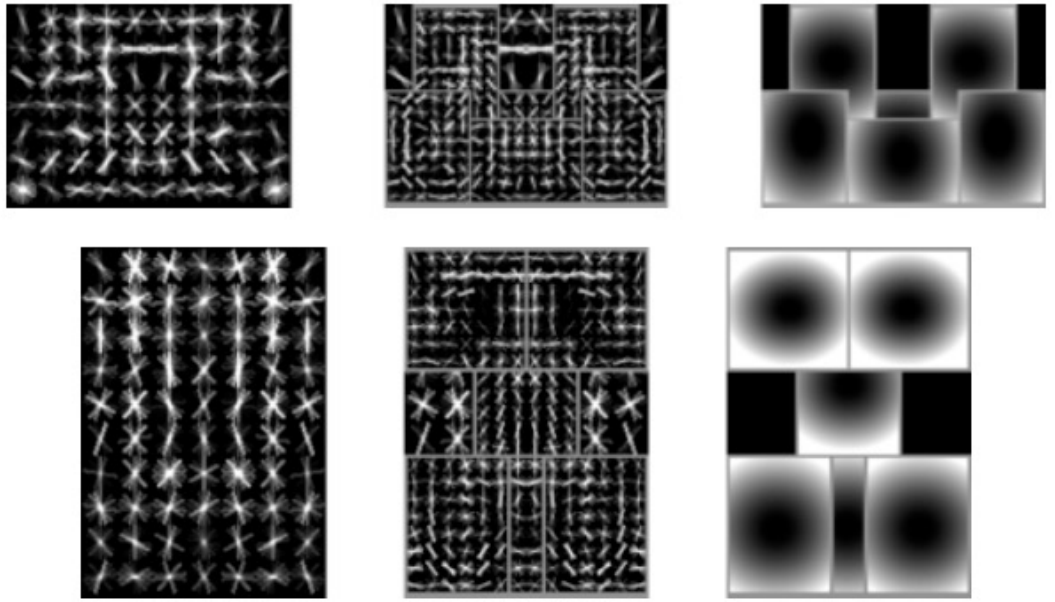
Specifying an object model

3. Hybrid template/parts model

Detections



Template Visualization
HoG features



root filters
coarse resolution

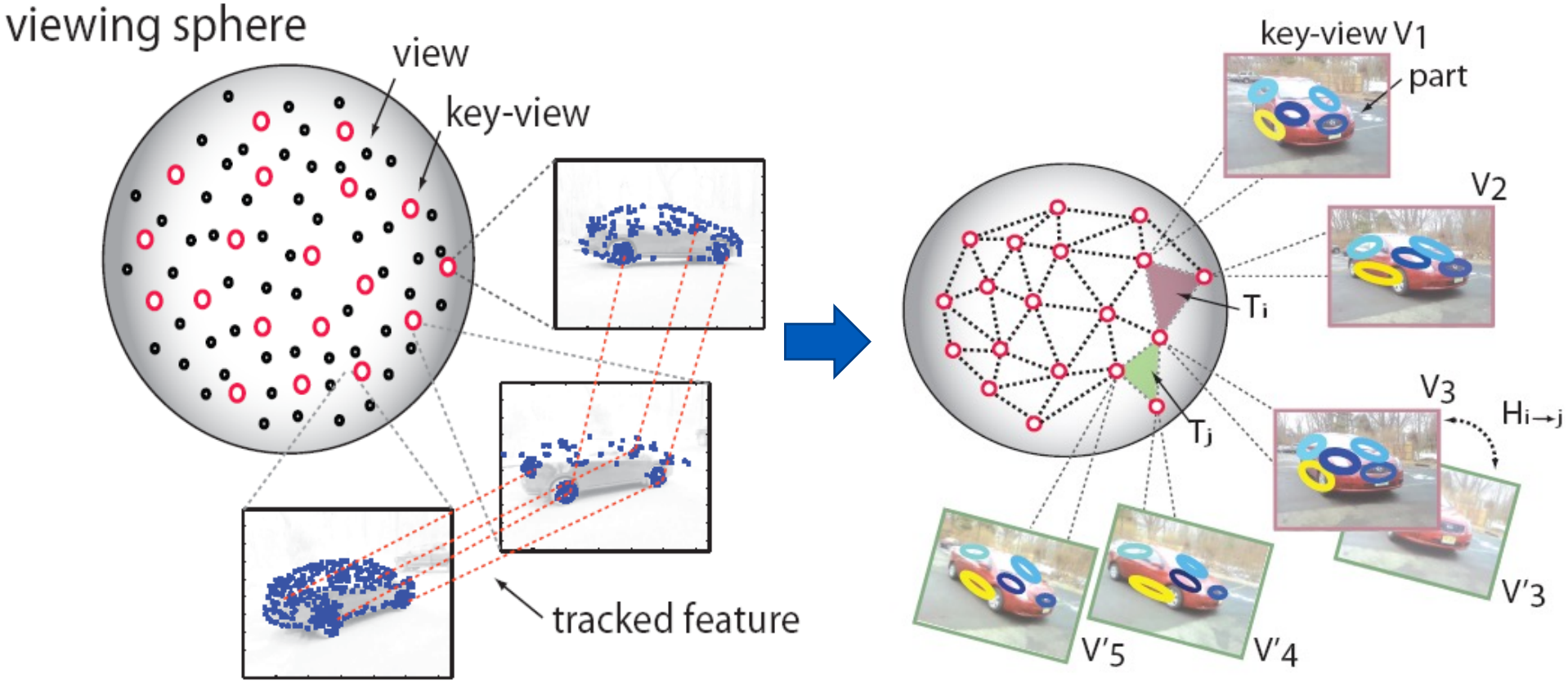
part filters
finer resolution

deformation
models

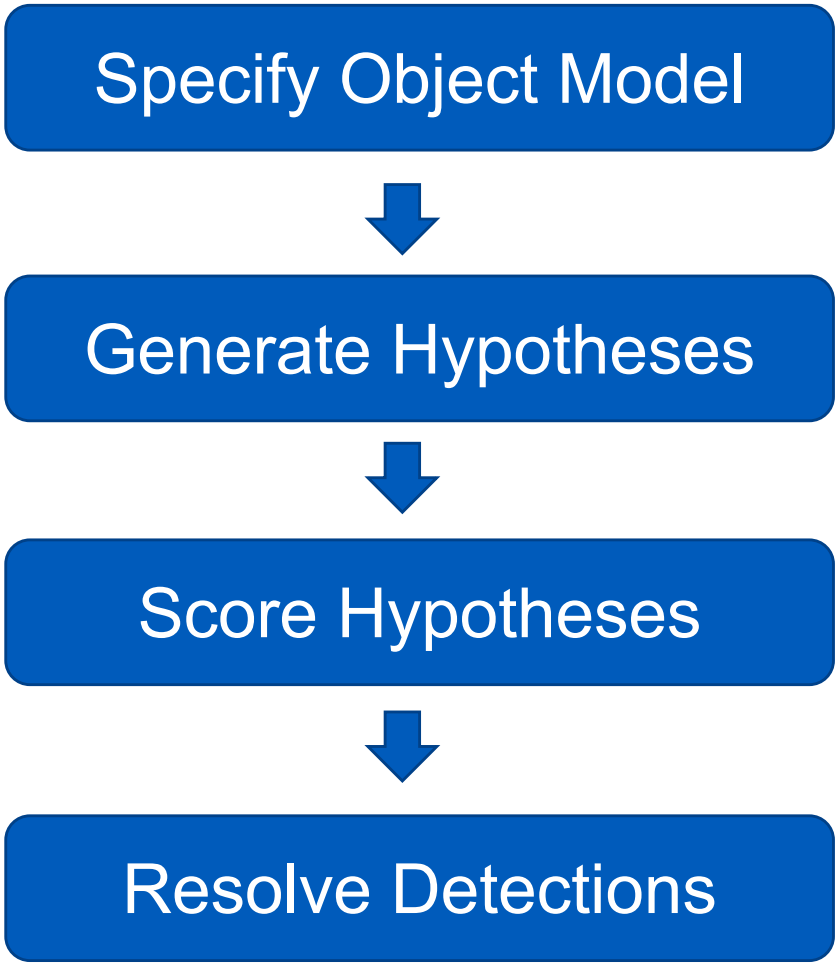
Felzenszwalb et al. 2008

Specifying an object model

4. 3D-ish model
   - Object is collection of 3D planar patches under affine transformation



General Process of Object Recognition



Propose an alignment of the model to the image

Generating hypotheses

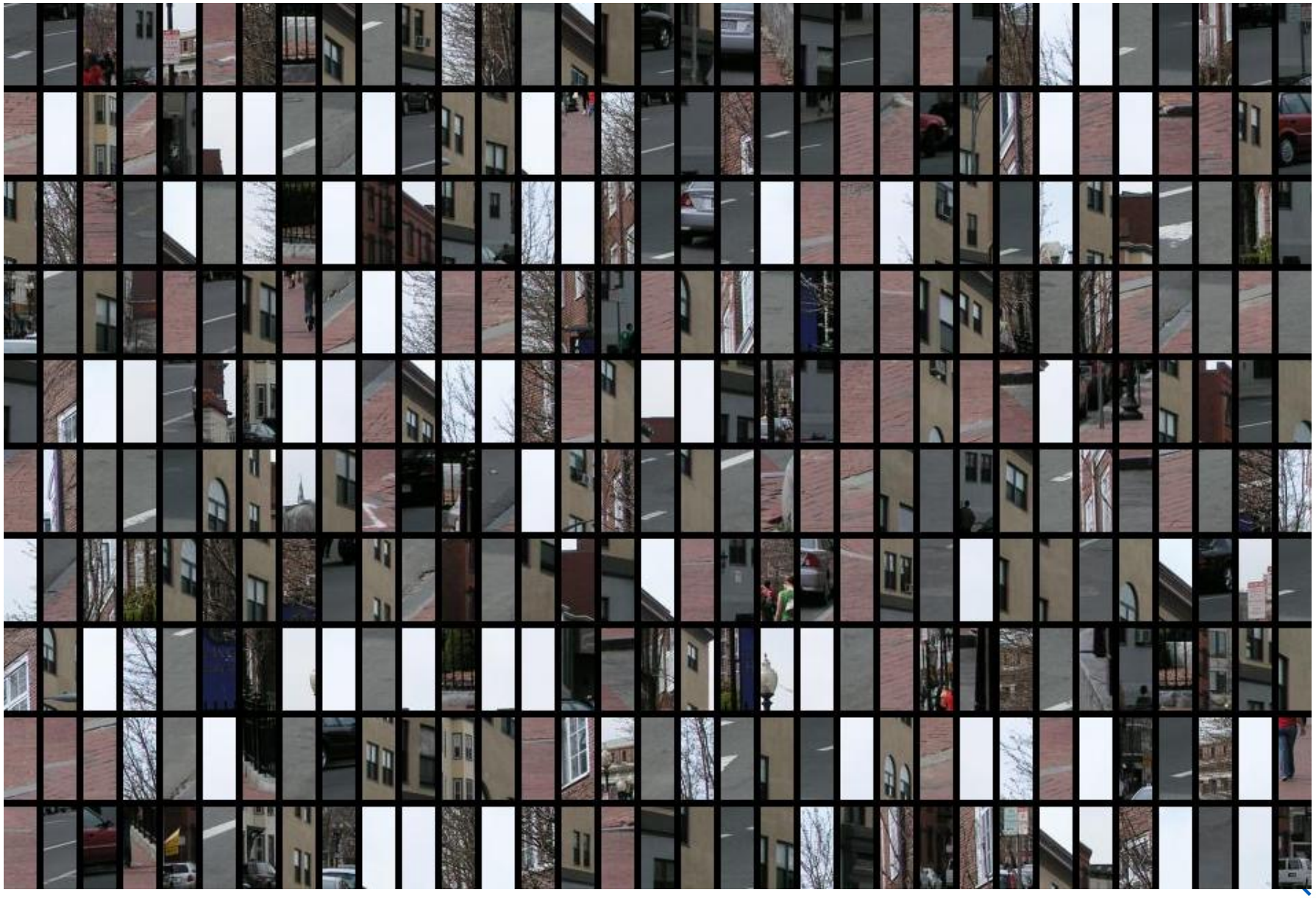
1. Sliding window
   - Test patch at each **location** and **scale**

Generating hypotheses

1. Sliding window
   - Test patch at each **location** and **scale**

Note – Template did not change size

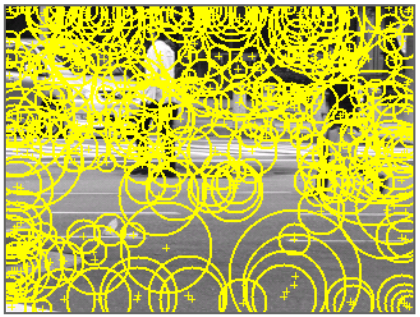
Each window is separately **classified**



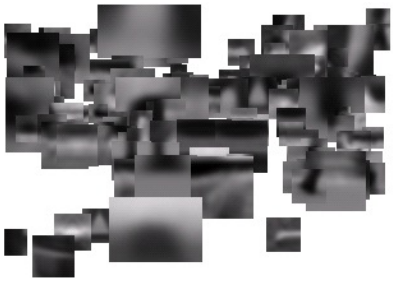
Generating hypotheses

2. Voting from patches/keypoints

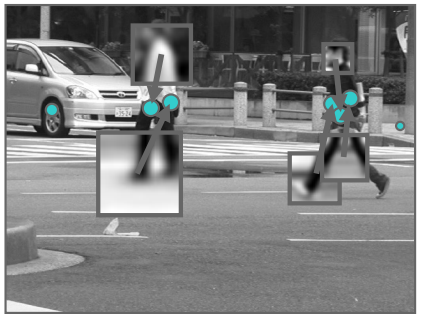
Interest Points



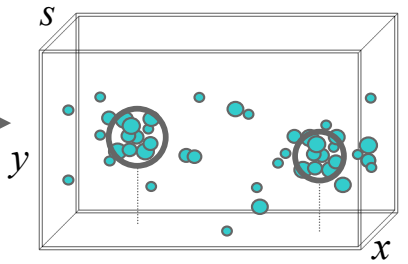
Matched Codebook Entries



Probabilistic Voting



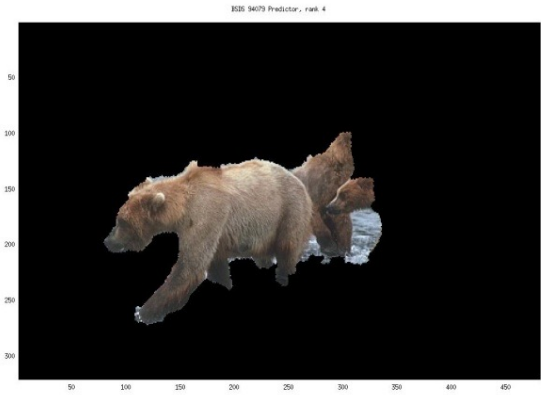
3D Voting Space (continuous)

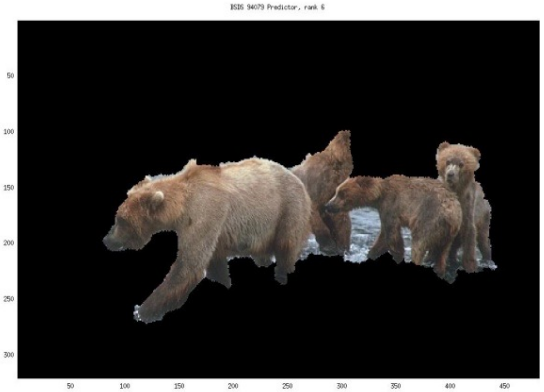


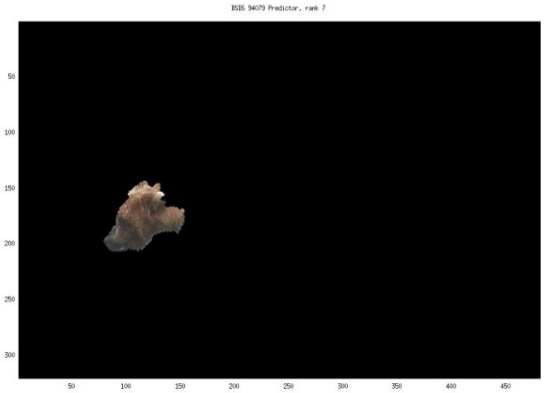
ISM model by Leibe et al.

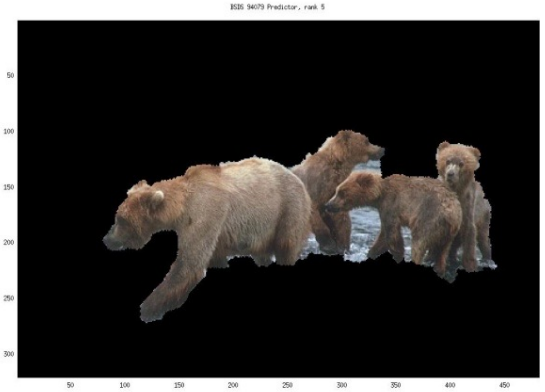
Generating hypotheses

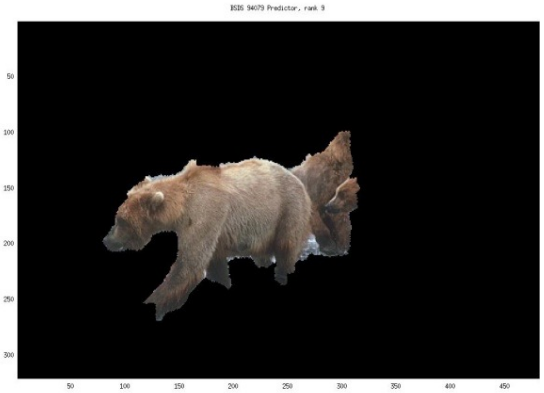
3. Region-based proposal



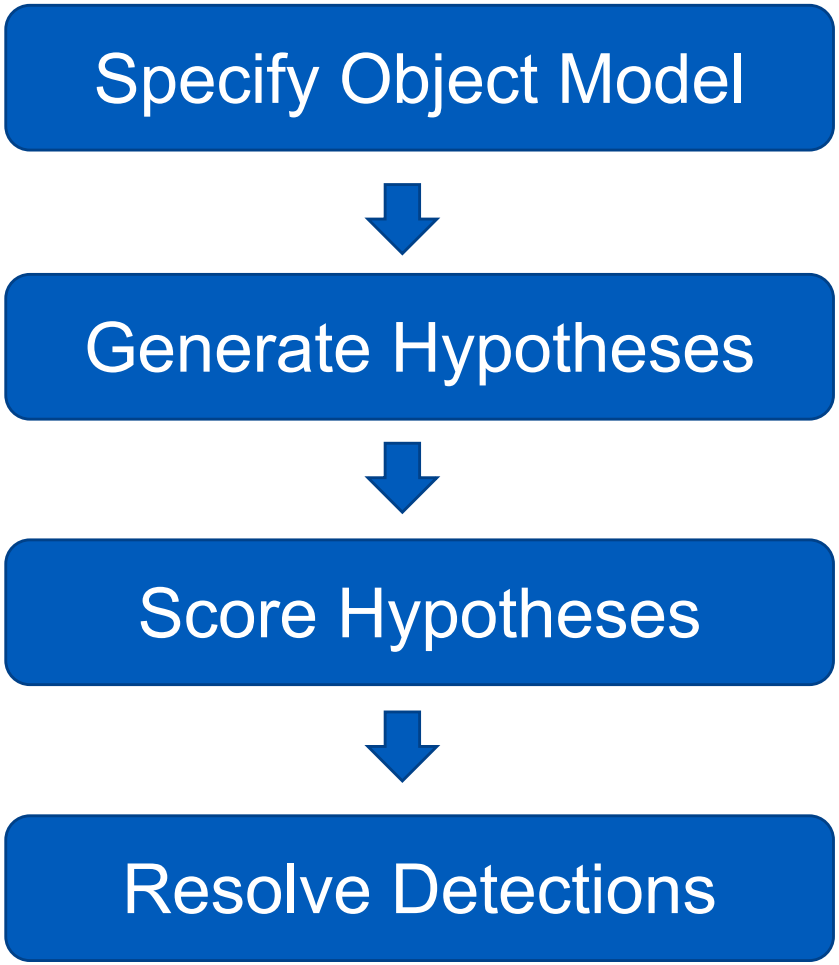






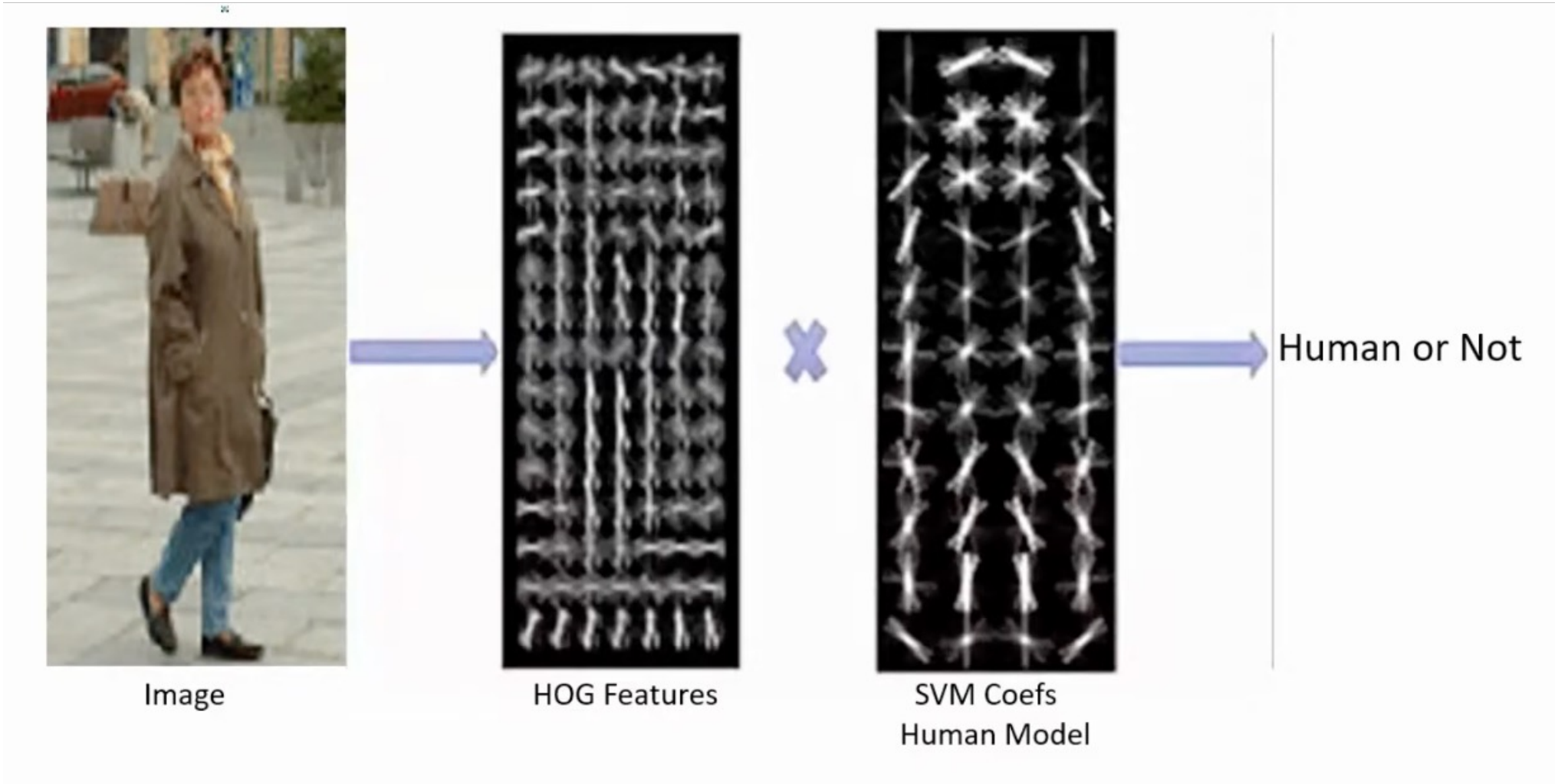


General Process of Object Recognition



Mainly-gradient based features, usually based on summary representation, many classifiers.

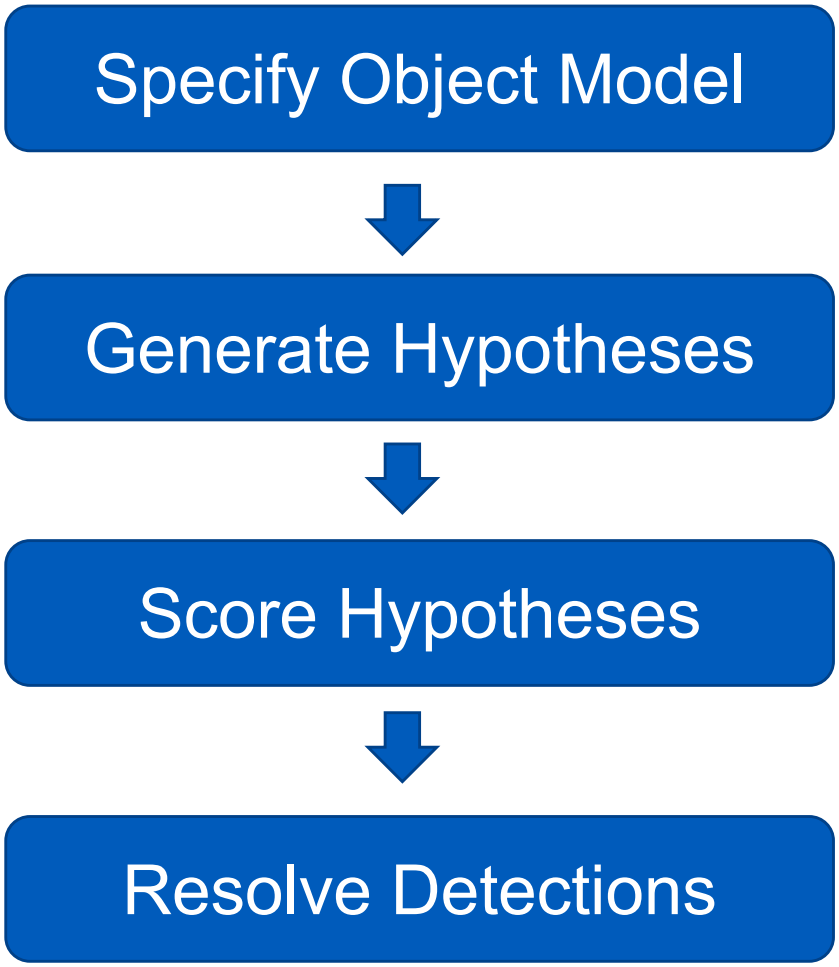
HOG + SVM for Object Detection



Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition, 2005

https://youtu.be/sDByl84n5mY

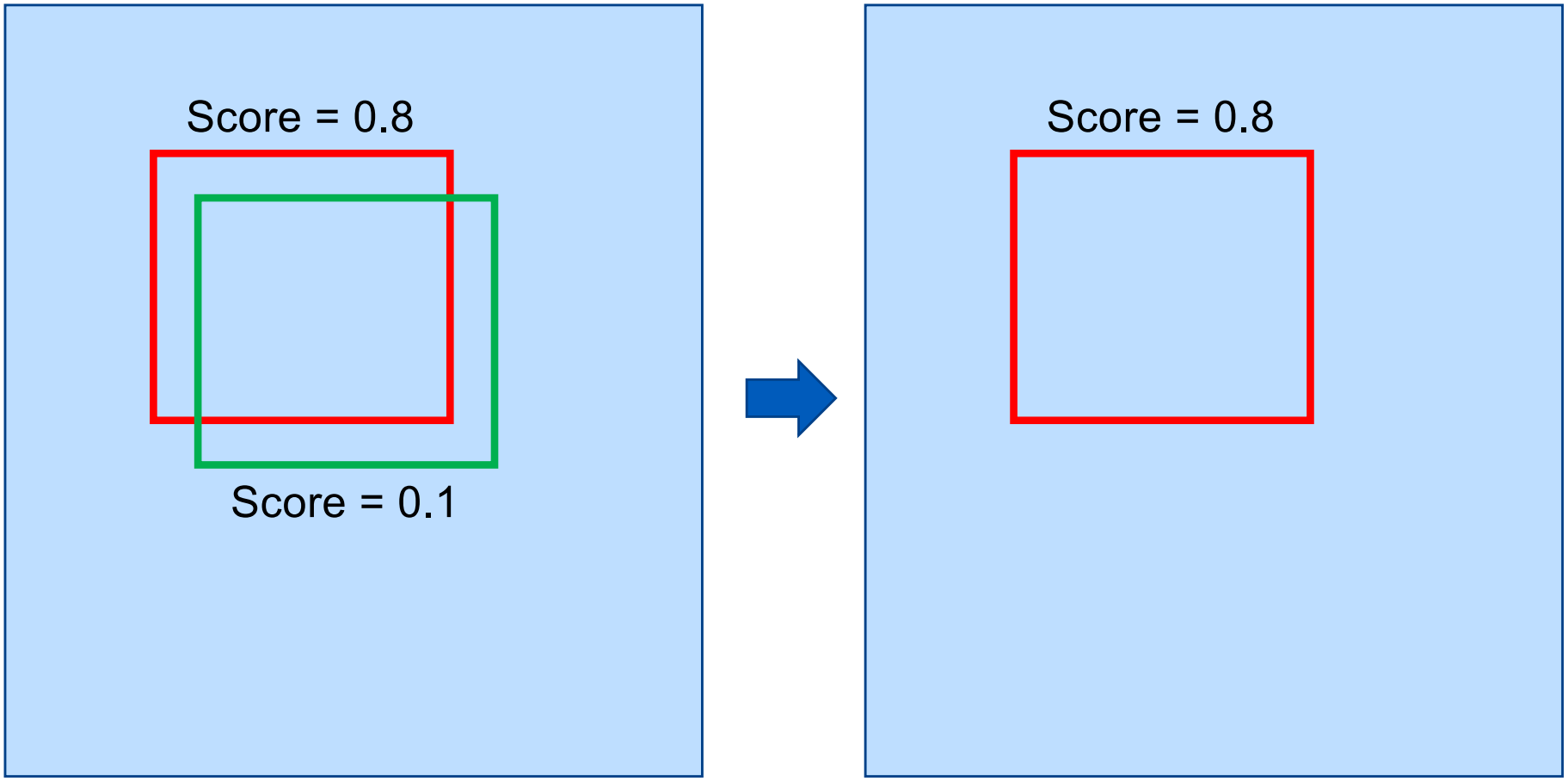
General Process of Object Recognition



Rescore each proposed object based on whole set

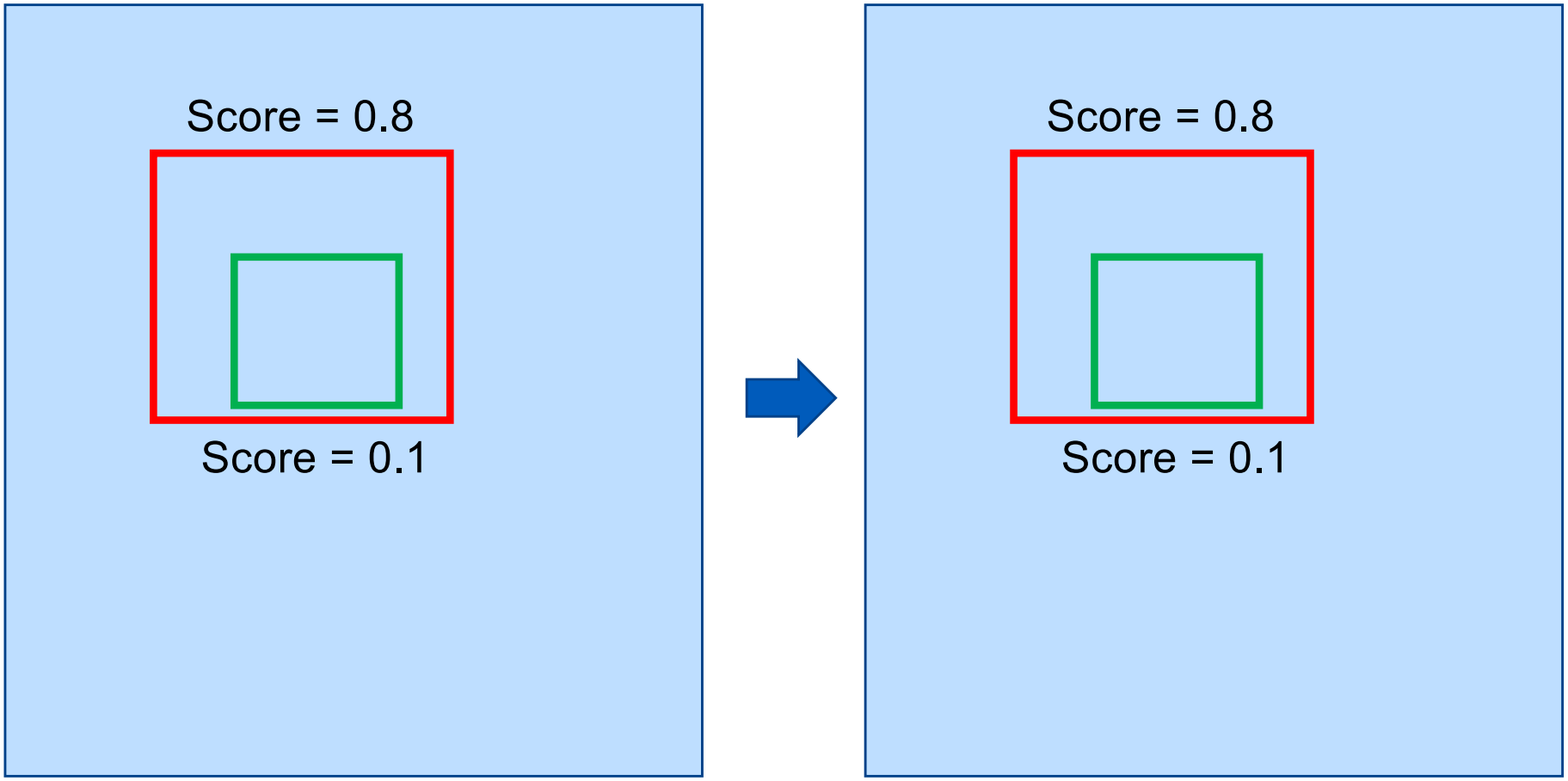
Resolving detection scores

1. Non-max suppression



Resolving detection scores

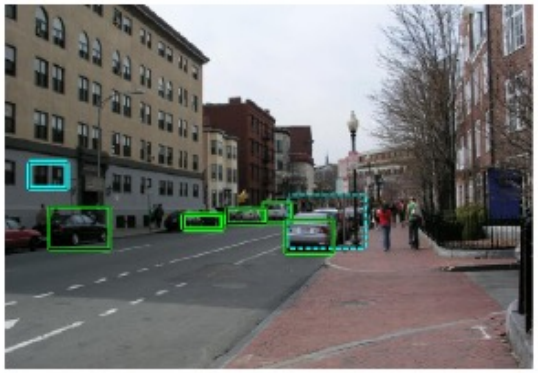
1. Non-max suppression



"Overlap" score is below some threshold

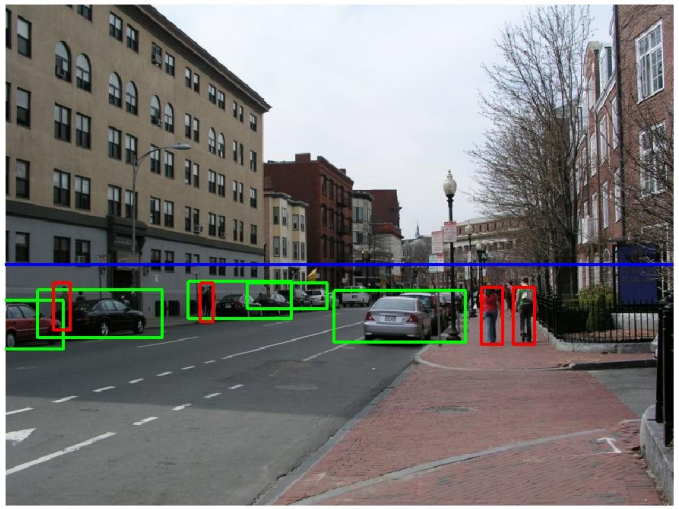
Resolving Detection Scores

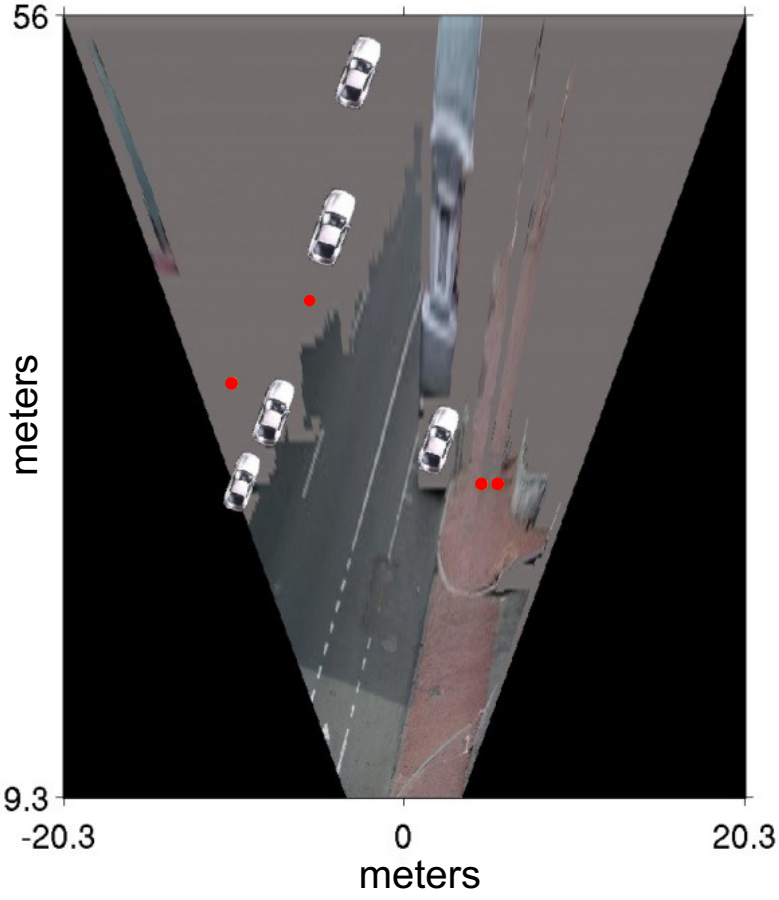
2. Context/reasoning



(g) Car Detections: Local

(h) Ped Detections: Local





Hoiem et al., 2006