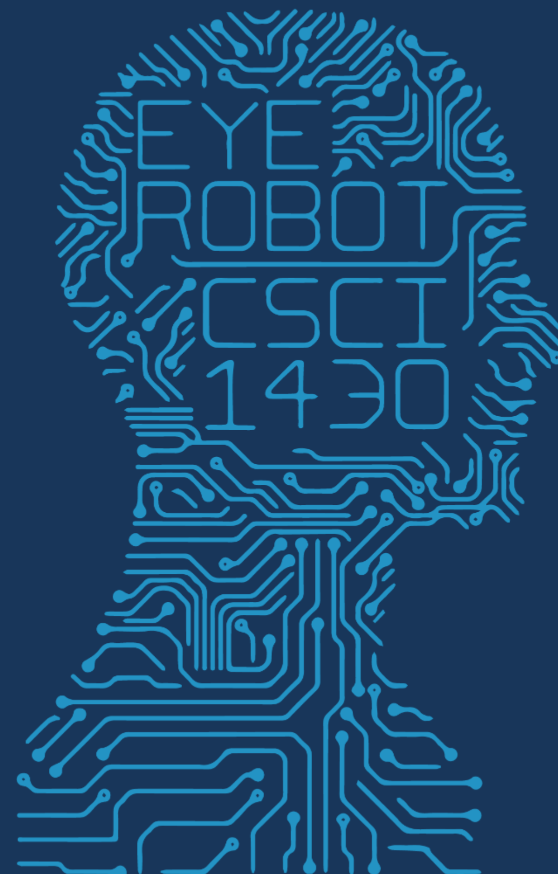
1950

Future Vision



2017 MWF 1pm 368

Computer Vision

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

Indy

Heterochromia iridum

From Wikipedia, the free encyclopedia

Not to be confused with Heterochromatin or Dichromatic (disambiguation).

In anatomy, **heterochromia** (ancient Greek: ἕτερος, *héteros*, different + χρῶμα, *chróma*, color[1]) is a difference in coloration, usually of the iris but also of hair or skin. Heterochromia is a result of the relative excess or lack of melanin (a pigment). It may be inherited, or caused by genetic mosaicism, chimerism, disease, or injury.[2]

Heterochromia of the eye (***heterochromia iridis*** or ***heterochromia iridum***) is of three kinds. In *complete heterochromia*, one iris is a different color from the other. In *sectoral heterochromia*, part of one iris is a different color from its remainder and finally in "central heterochromia" there are spikes of different colours radiating from the pupil.

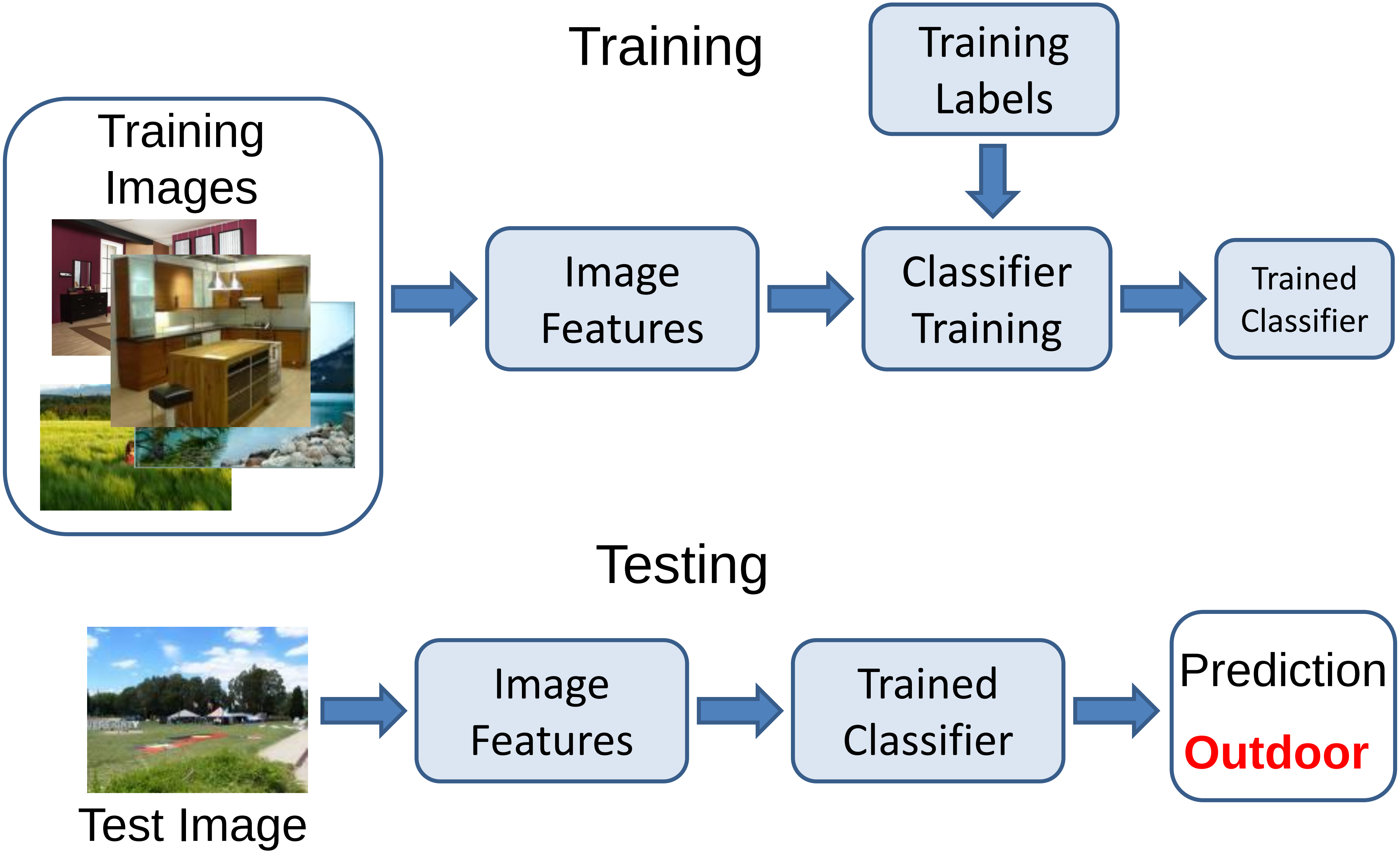
Heterochromia

Complete heterochromia in human eyes: one brown and one green/hazel

Classification and external resources

Specialty	ophthalmology
ICD-10	Q13.2, H20.8, L67.1
ICD-9-CM	364.53
OMIM	142500
DiseasesDB	31289

Instance Recognition



Instance recognition: 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?

Kristen Grauman

Instance recognition: 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?

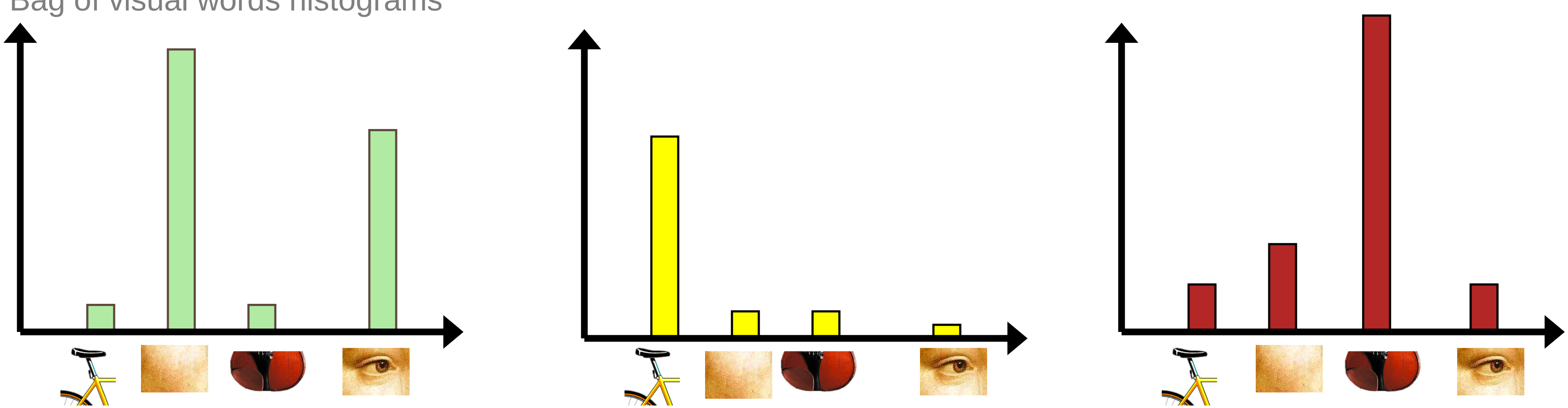
Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?

How to score the retrieval results?

Kristen Grauman

Visual words

Bag of visual words histograms

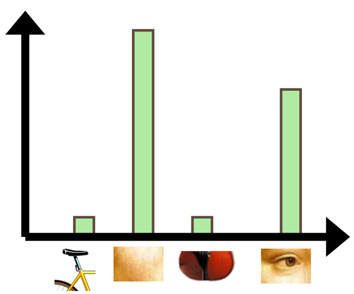


Comparing bags of words

Compute normalized scalar (dot) product between their (possibly weighted) occurrence counts, then rank and pick smallest. *Nearest neighbor* search for similar images.

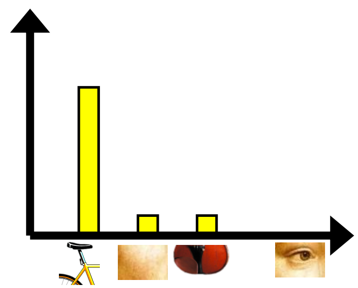
Database image

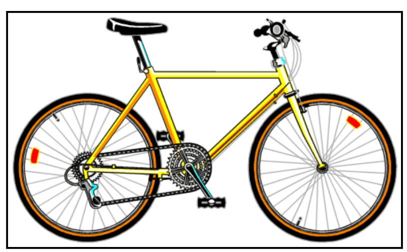
$$\vec{d}_j = [1 \quad 8 \quad 1 \quad 4]$$



Query

$$\vec{q} = [5 \quad 1 \quad 1 \quad 0]$$





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

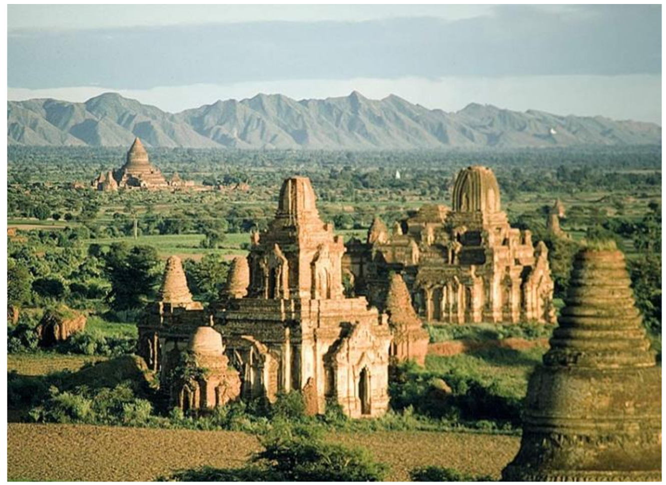
$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

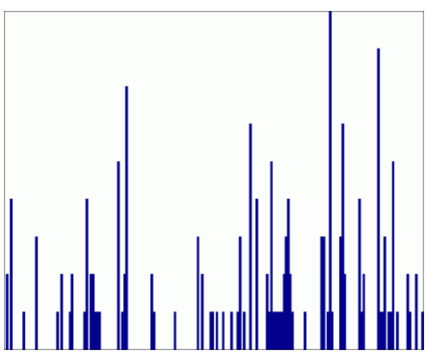
for vocabulary of *V* words

Kristen Grauman

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



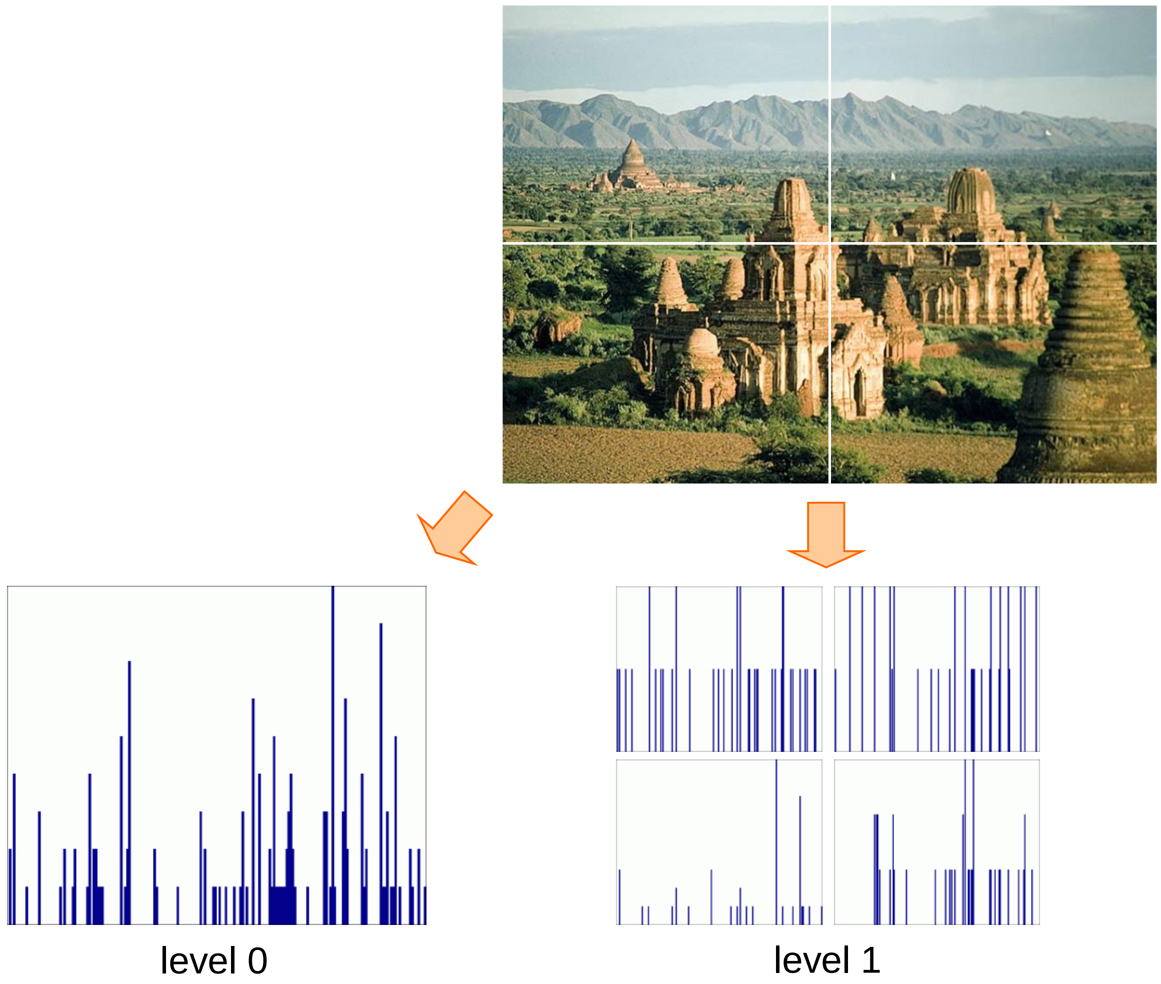


level 0

Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

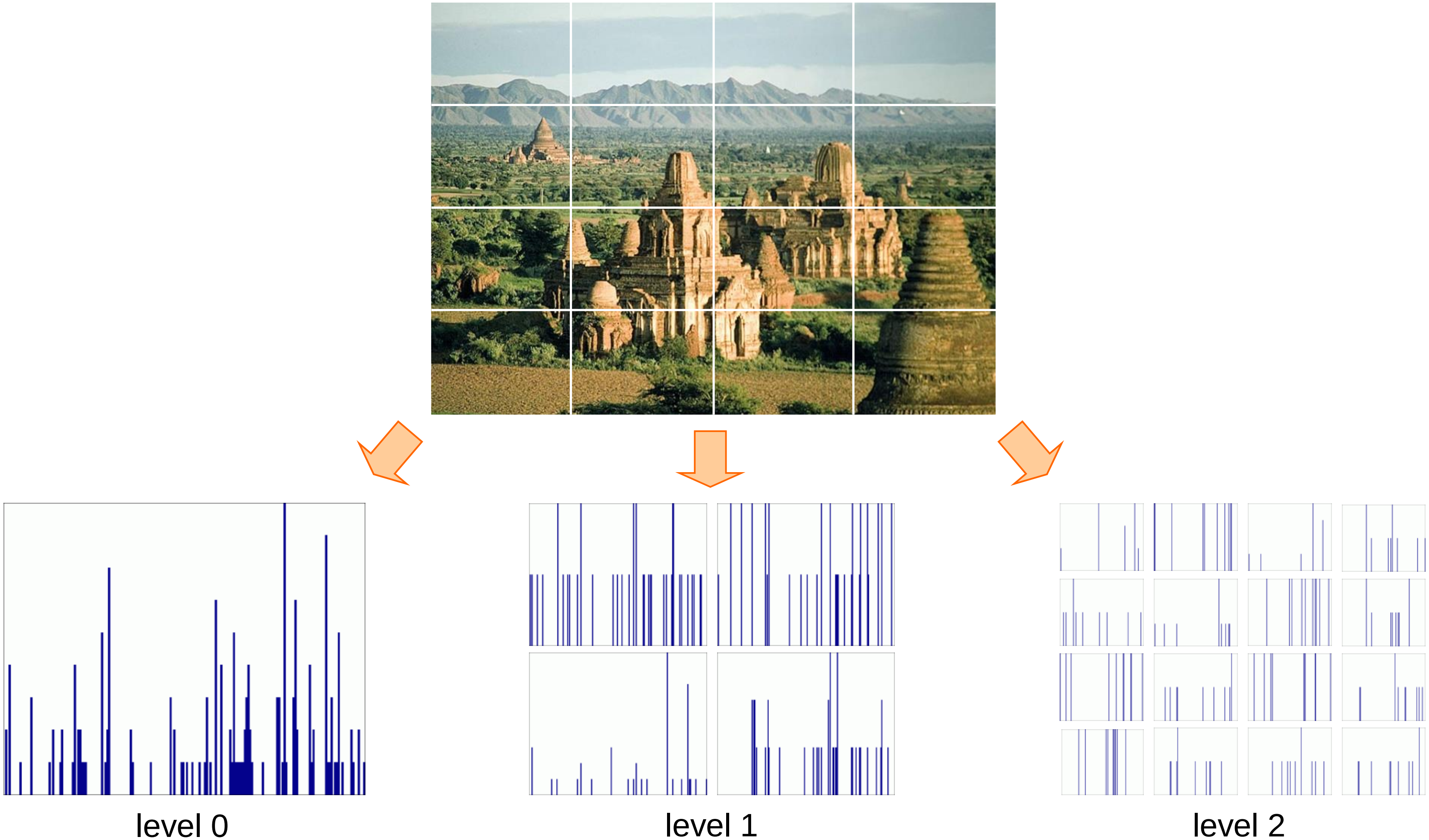
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

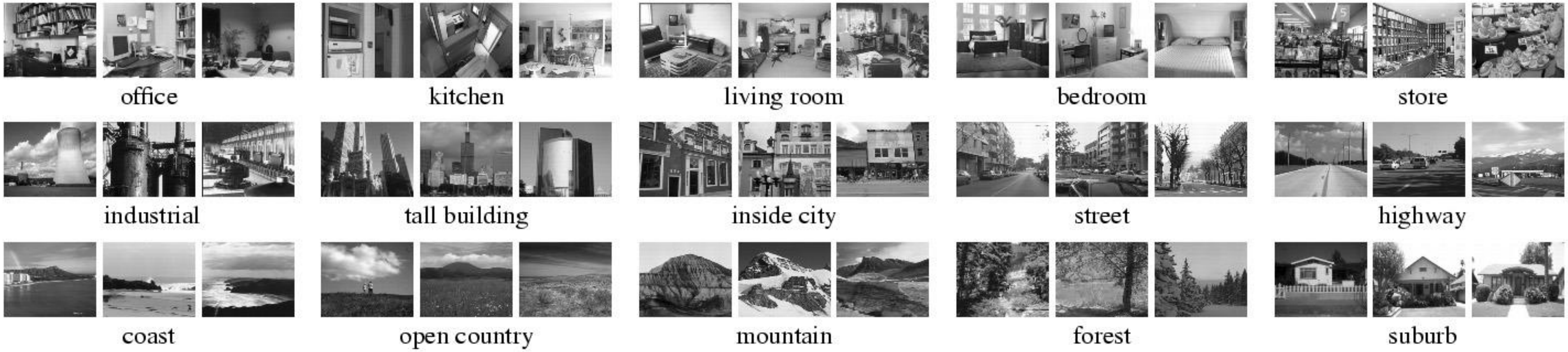
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

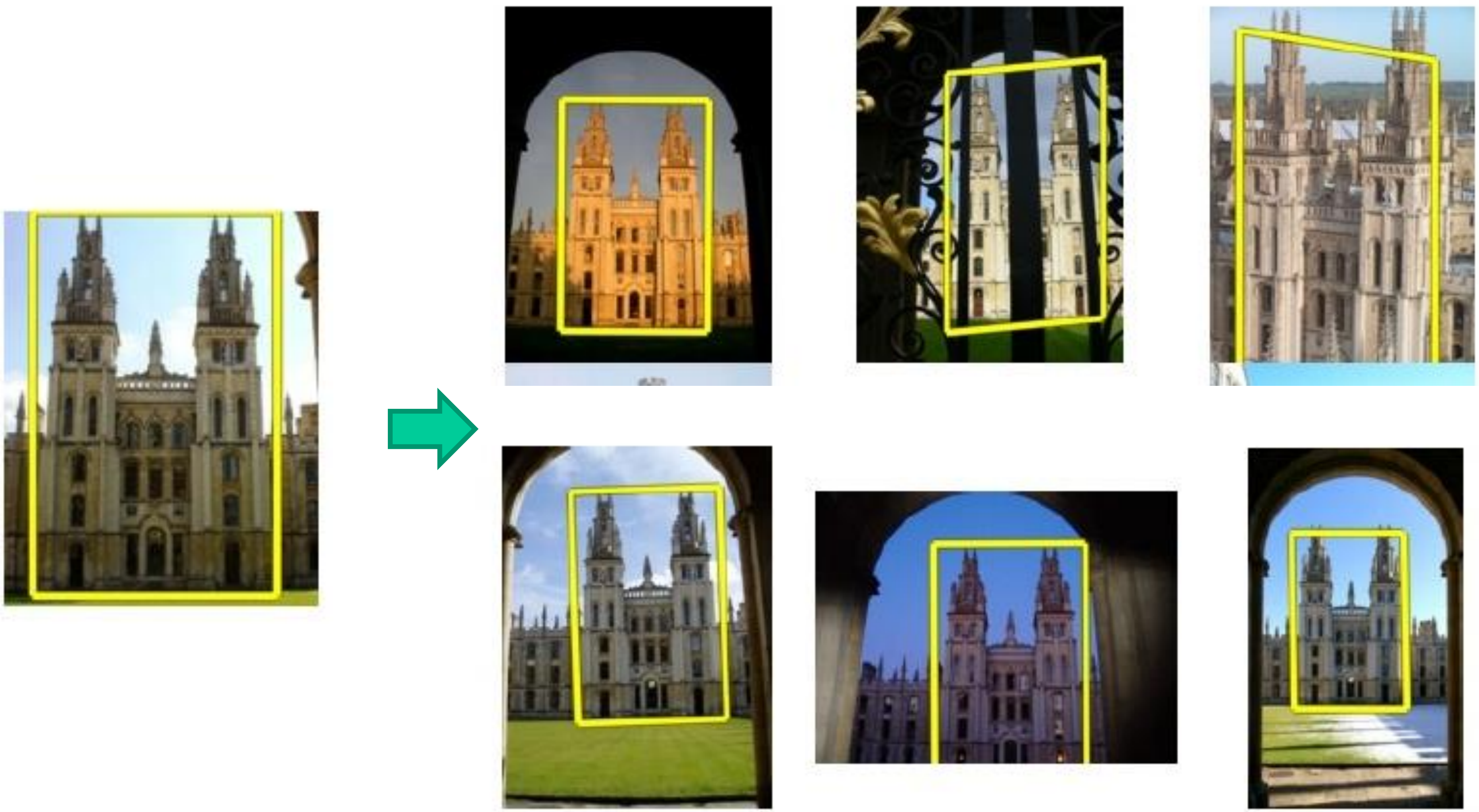
Scene category dataset



Multi-class classification results
(100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 ±0.5		72.2 ±0.6	
1 (2×2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4×4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8×8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

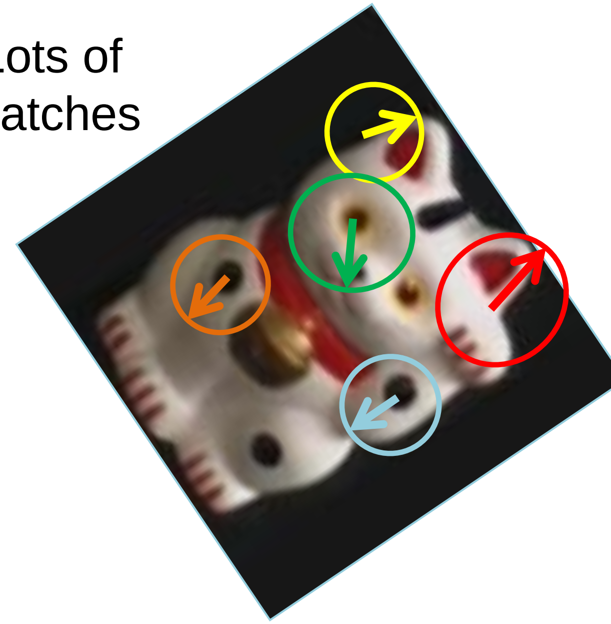
How can we quickly find images in a large database that match a given image region?



Simple idea

See how many keypoints are close to keypoints in each other image

Lots of Matches

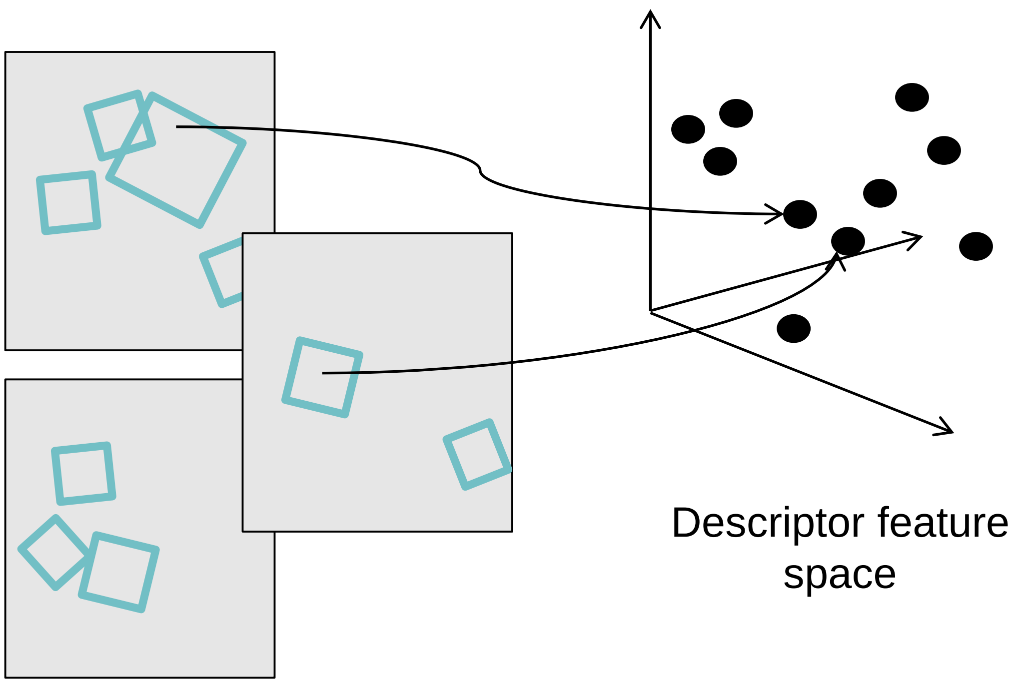


Few or No Matches

But this will be really, really slow!

Indexing local features

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

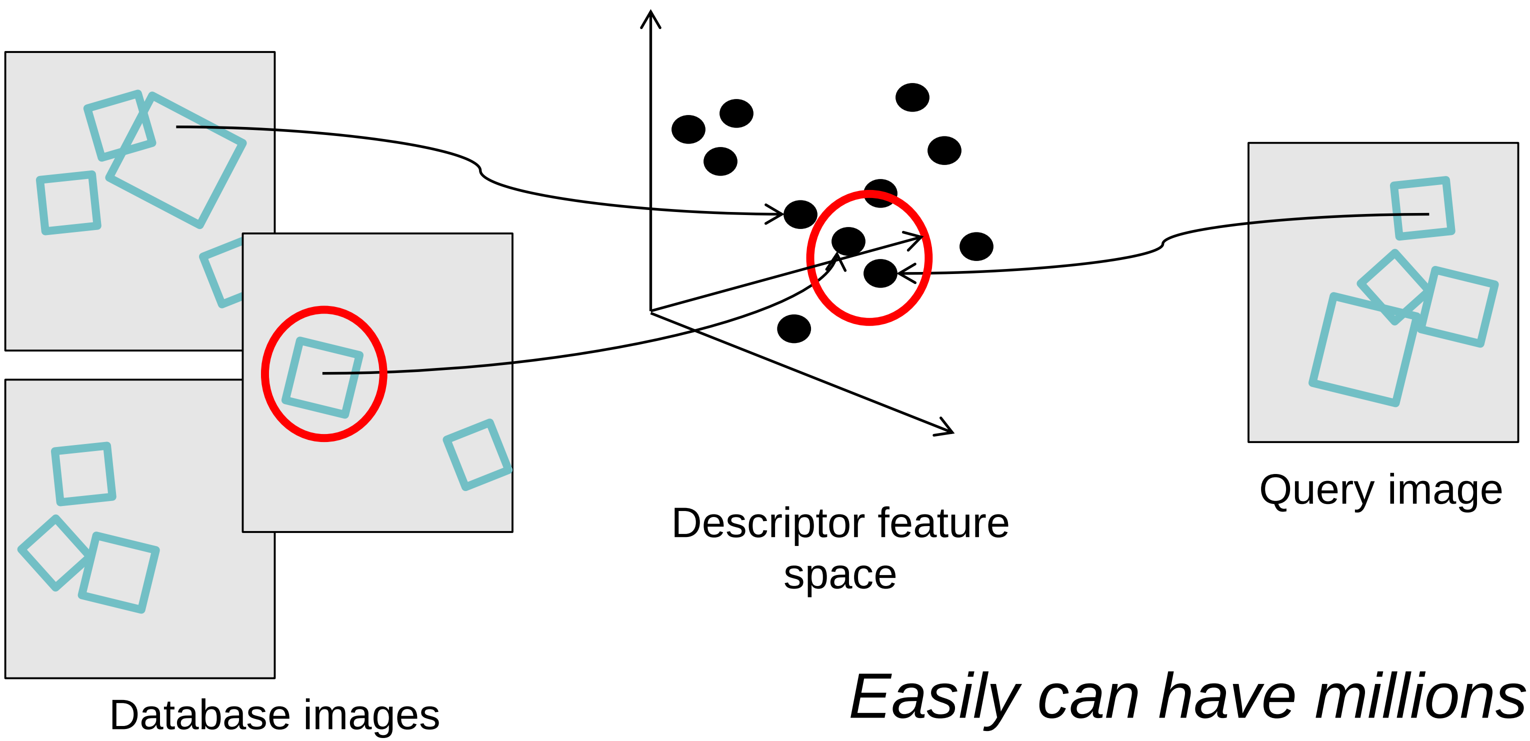


Descriptor feature space

Kristen Grauman

Indexing local features

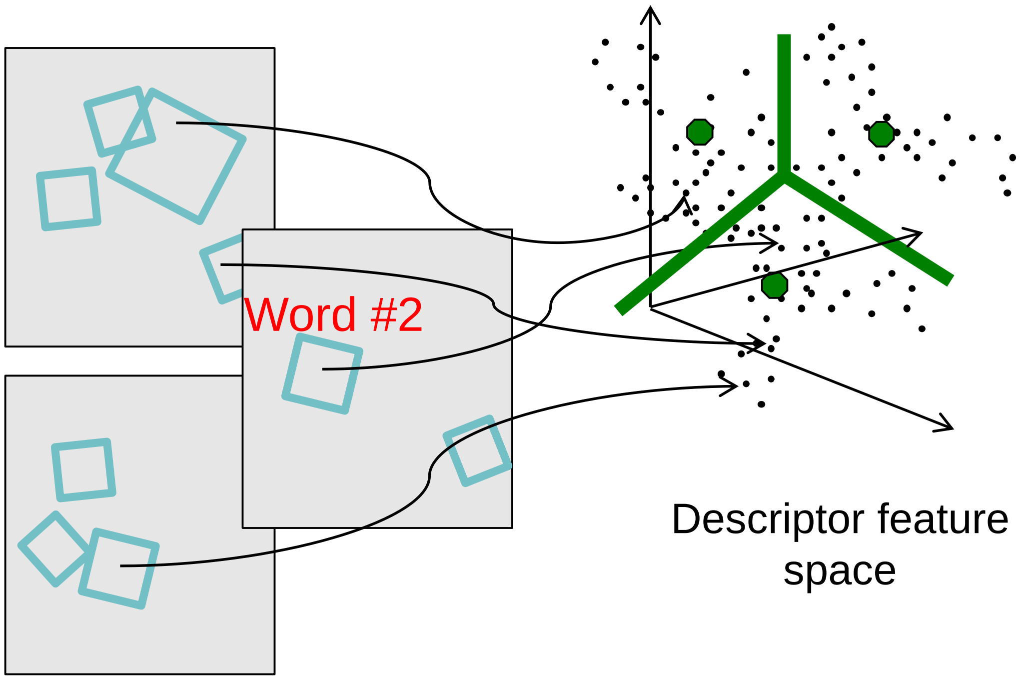
- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



Easily can have millions of features to search!

Visual words

Map high-dimensional descriptors to tokens/words by quantizing the feature space.

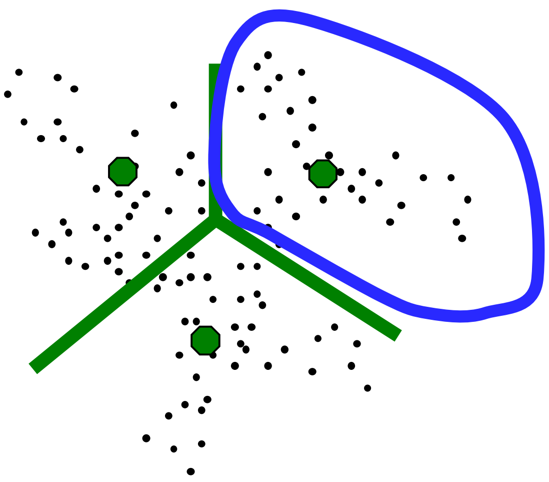


- Quantize via clustering; cluster centers are the visual “words”
- Assign word to each image region by finding the closest cluster center.

Kristen Grauman

Visual words

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



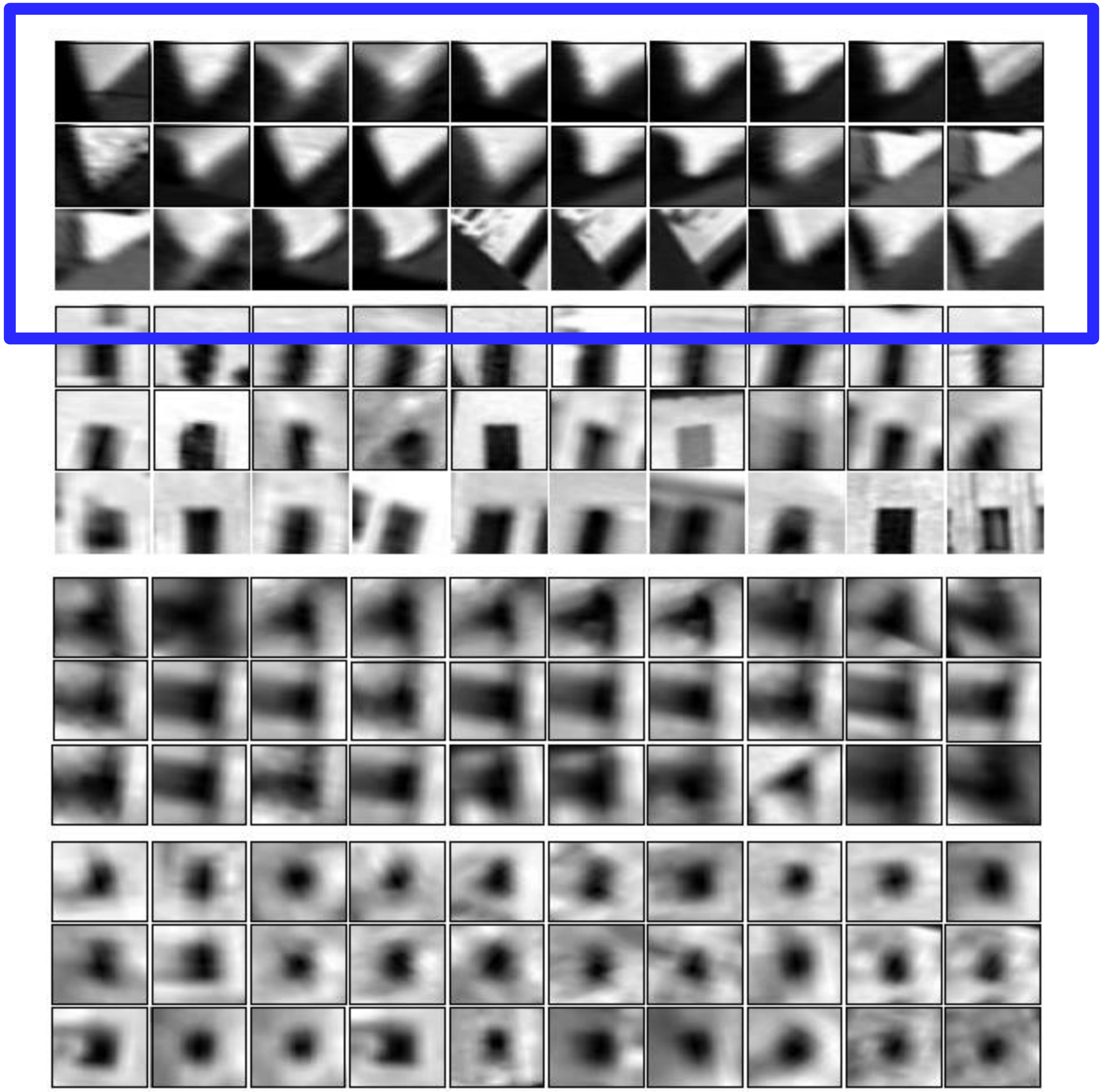
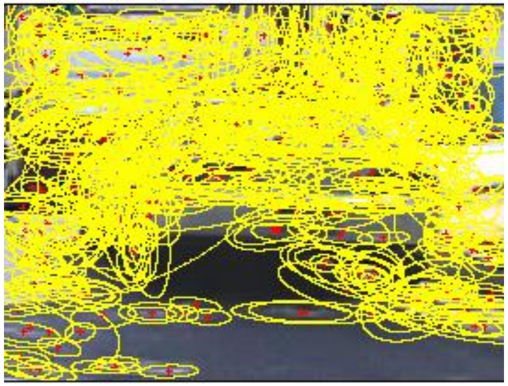


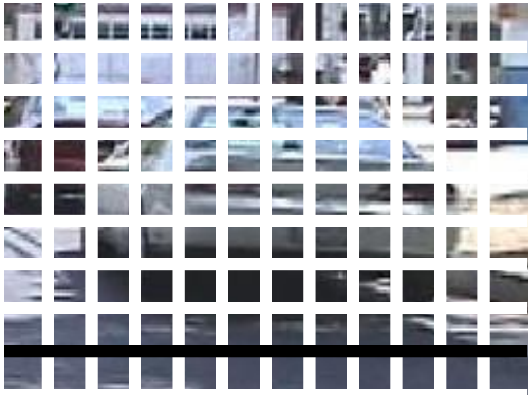
Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman

Sampling strategies

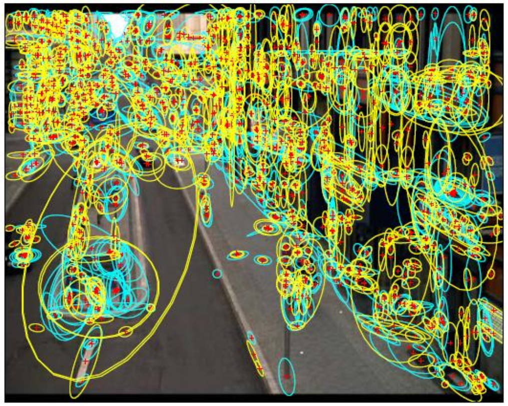


Sparse, at interest points



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

Fast lookup: 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.

Kristen Grauman

Inverted file index

Database images

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

Kristen Grauman

Inverted file index

New query image

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

…

Candidate matches

Image #1

Image #2

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

Kristen Grauman

Inverted file index

New query image

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

...

Candidate matches

Image #1

Image #2

1. Extract words in query
2. Inverted file index to find relevant frames
3. Compare word counts

Kristen Grauman

Inverted file index

Key requirement: *sparsity*.

If most images contain most words, then we’re not better off than exhaustive search.

- Exhaustive search would mean comparing the visual word distribution of a query versus every page.

Instance recognition: remaining issues

How to summarize the content of an entire image? And gauge overall similarity?

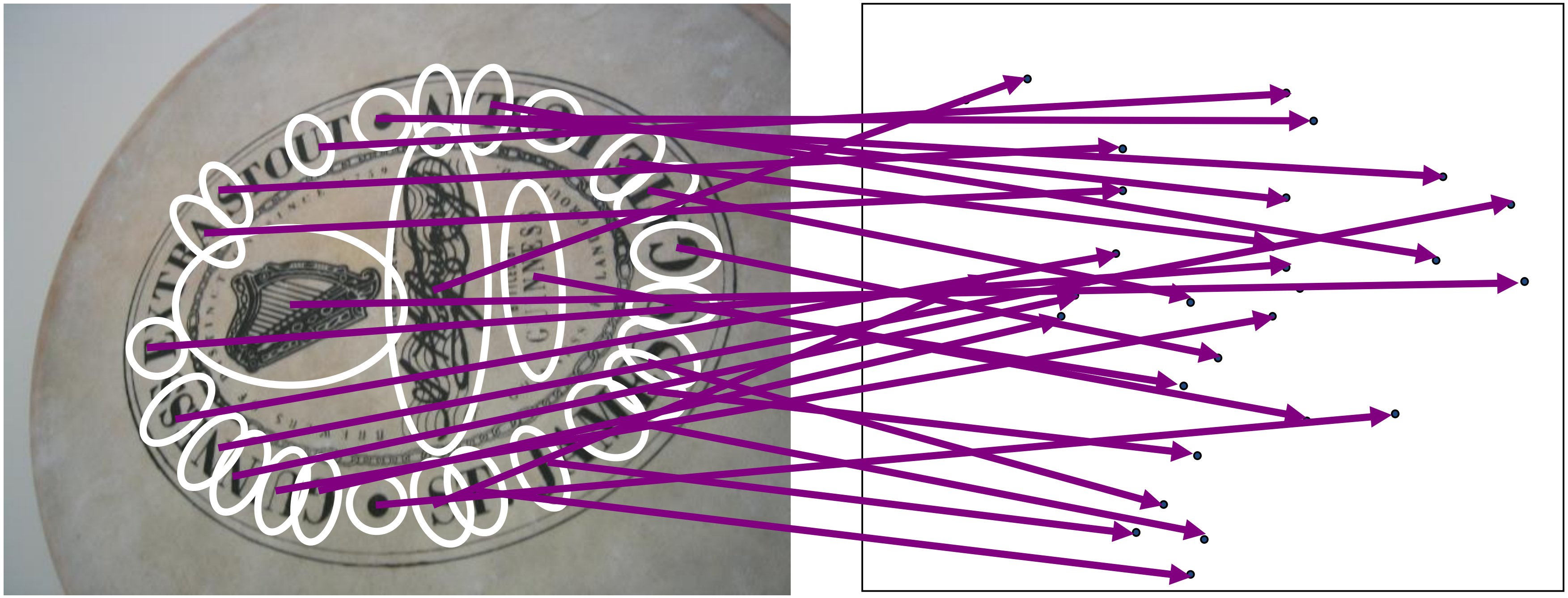
How large should the vocabulary be? How to perform quantization efficiently?

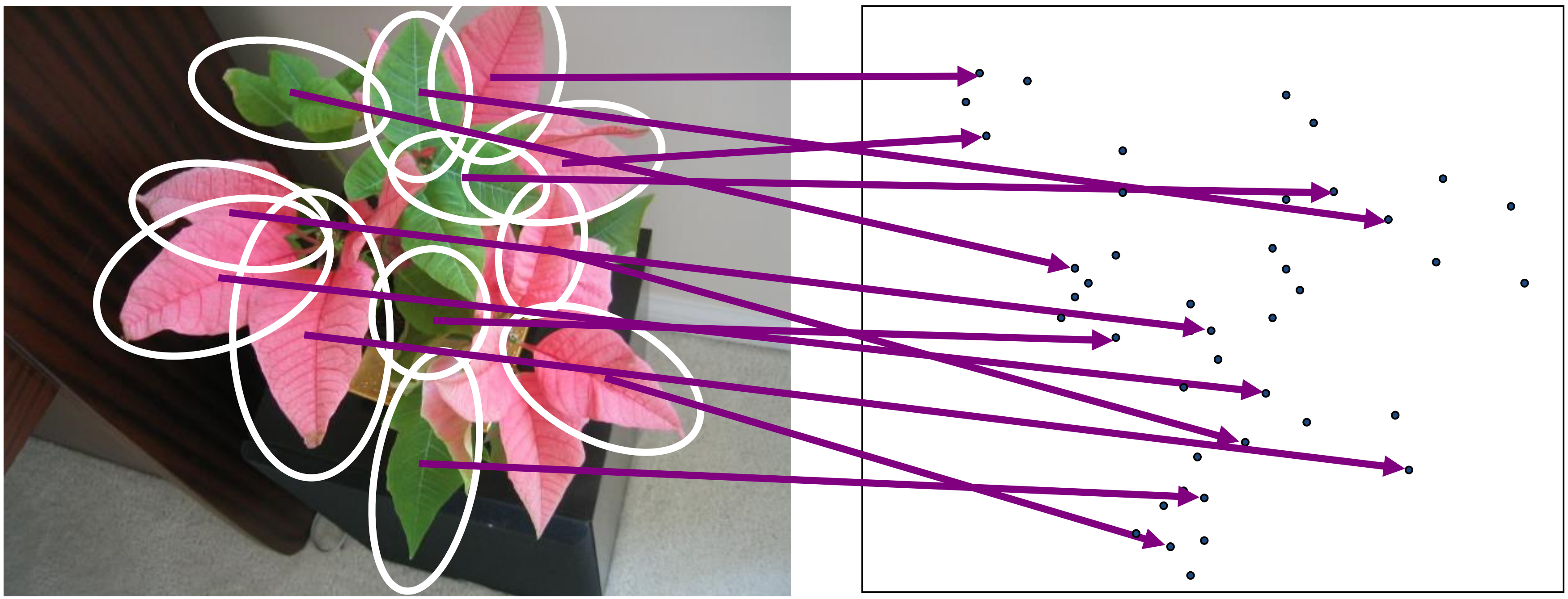
Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?

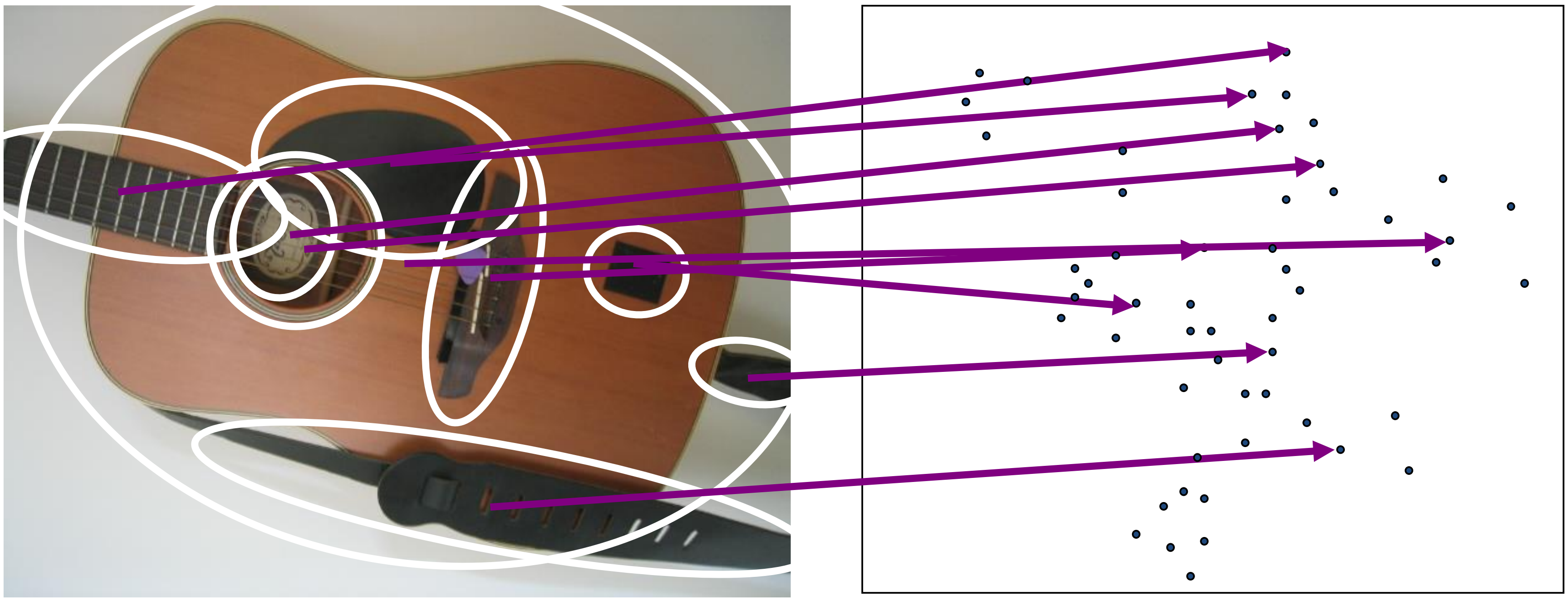
How to score the retrieval results?

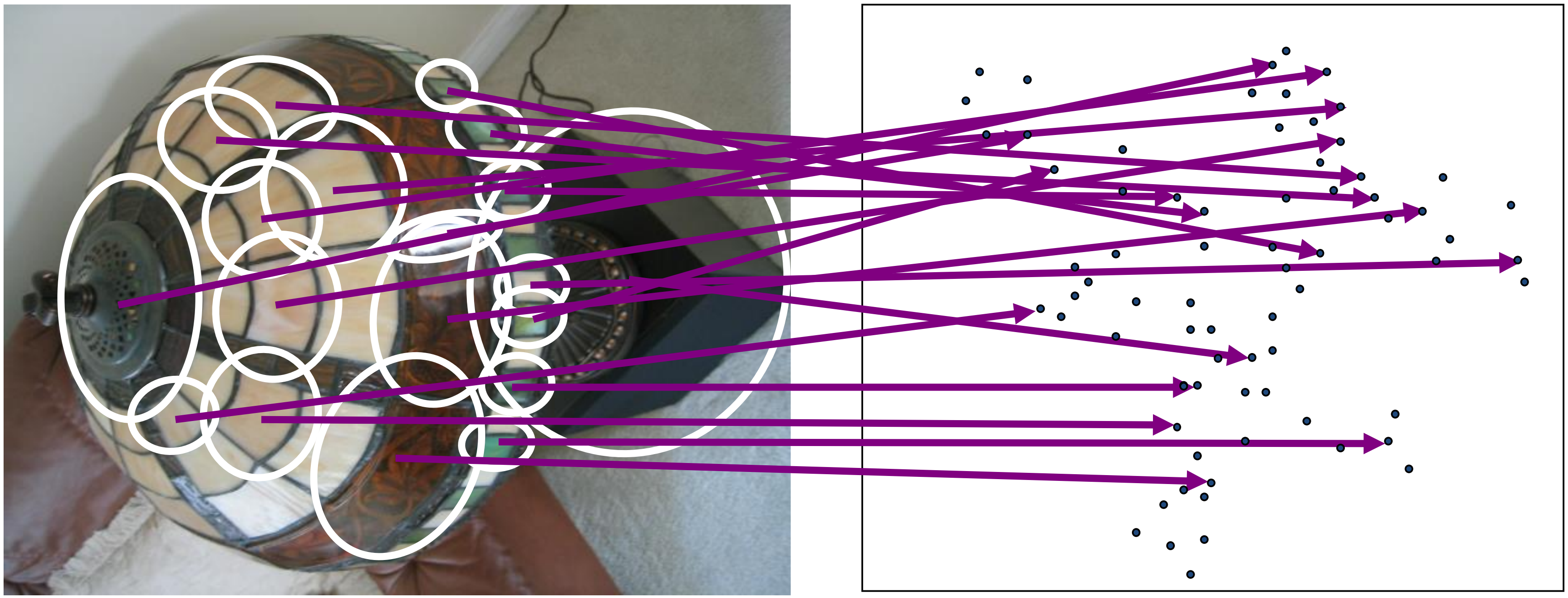
Following slides by David Nister (CVPR 2006)

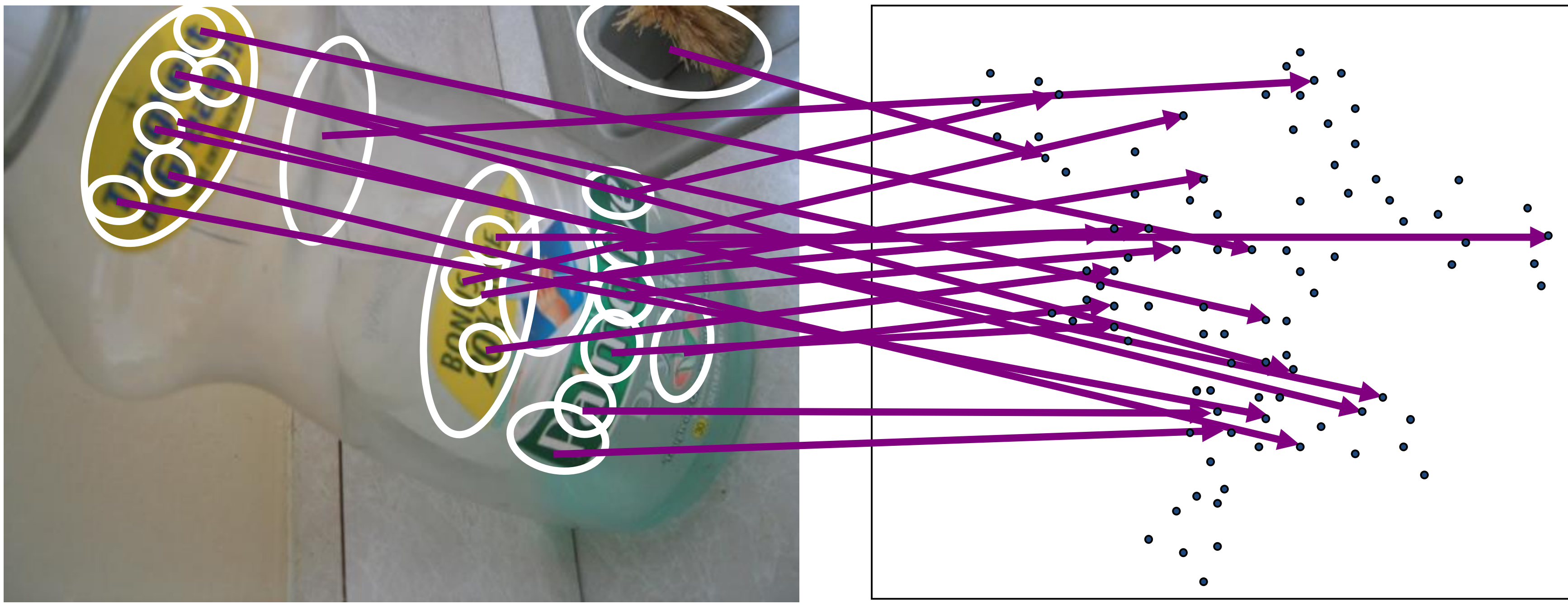
Kristen Grauman

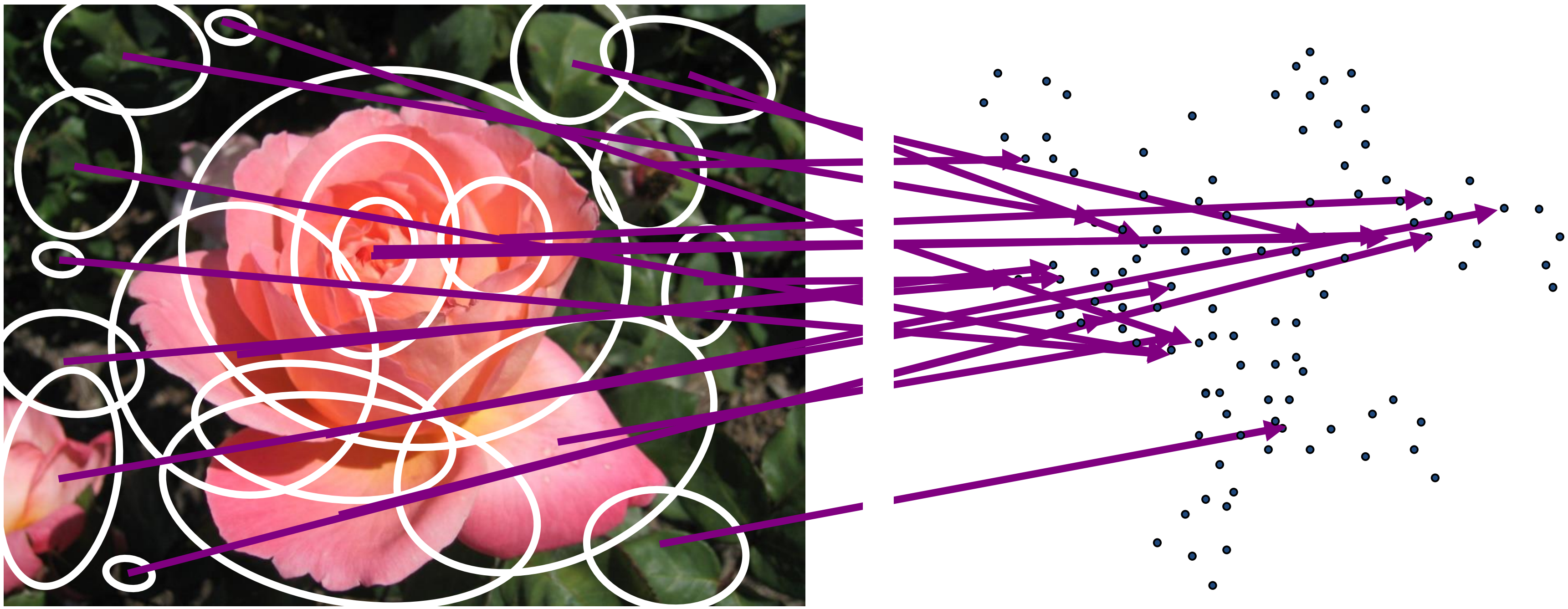


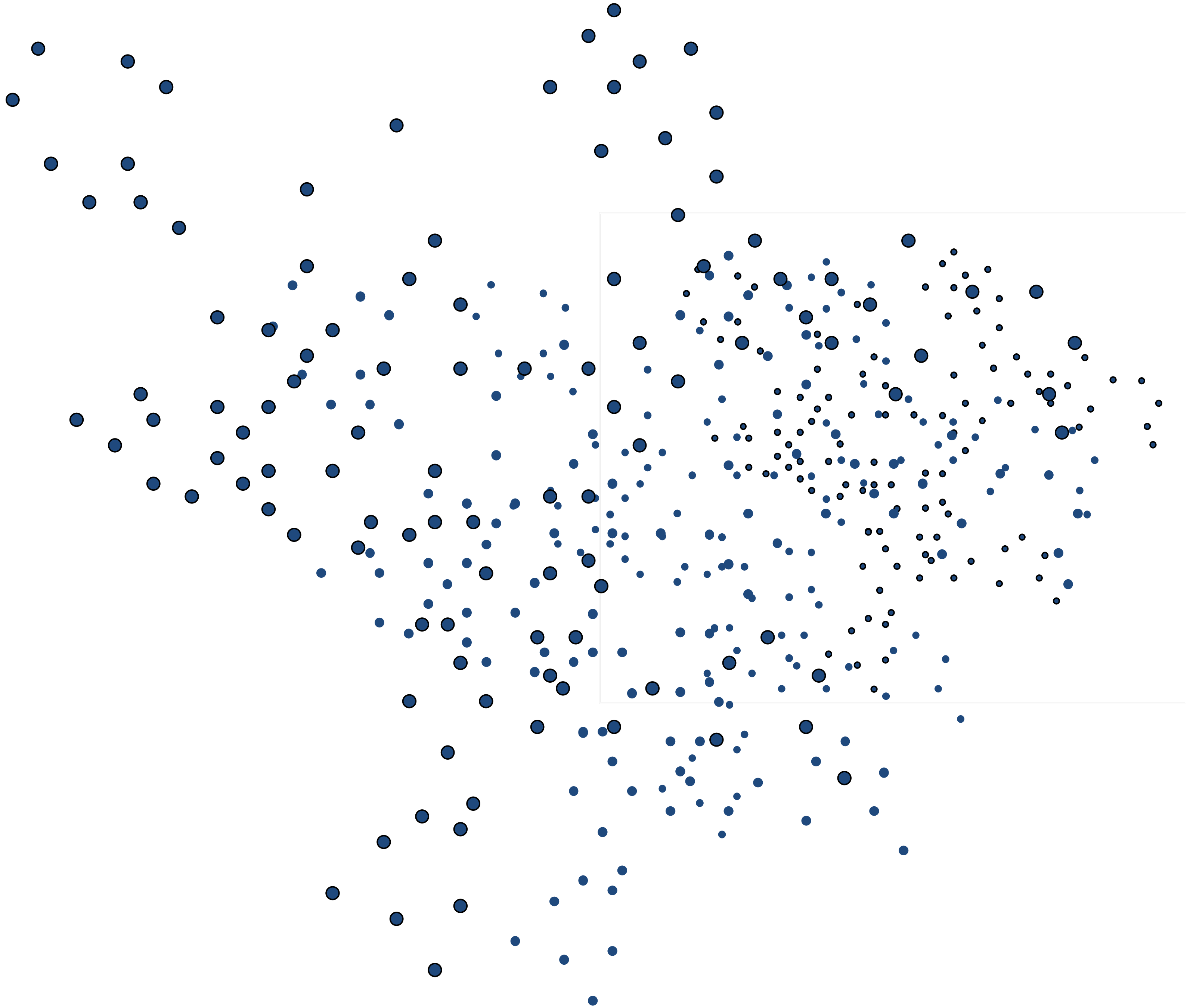


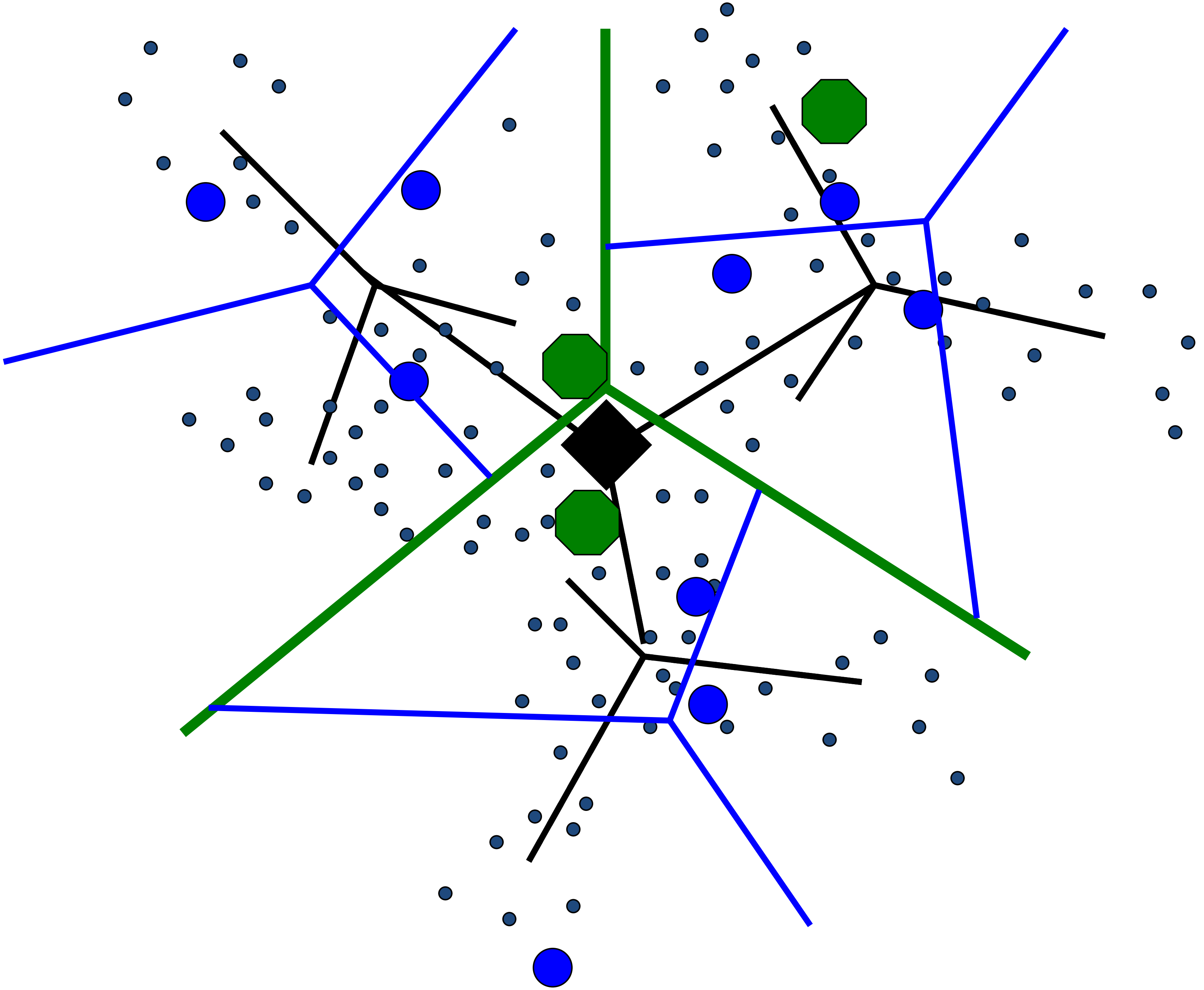


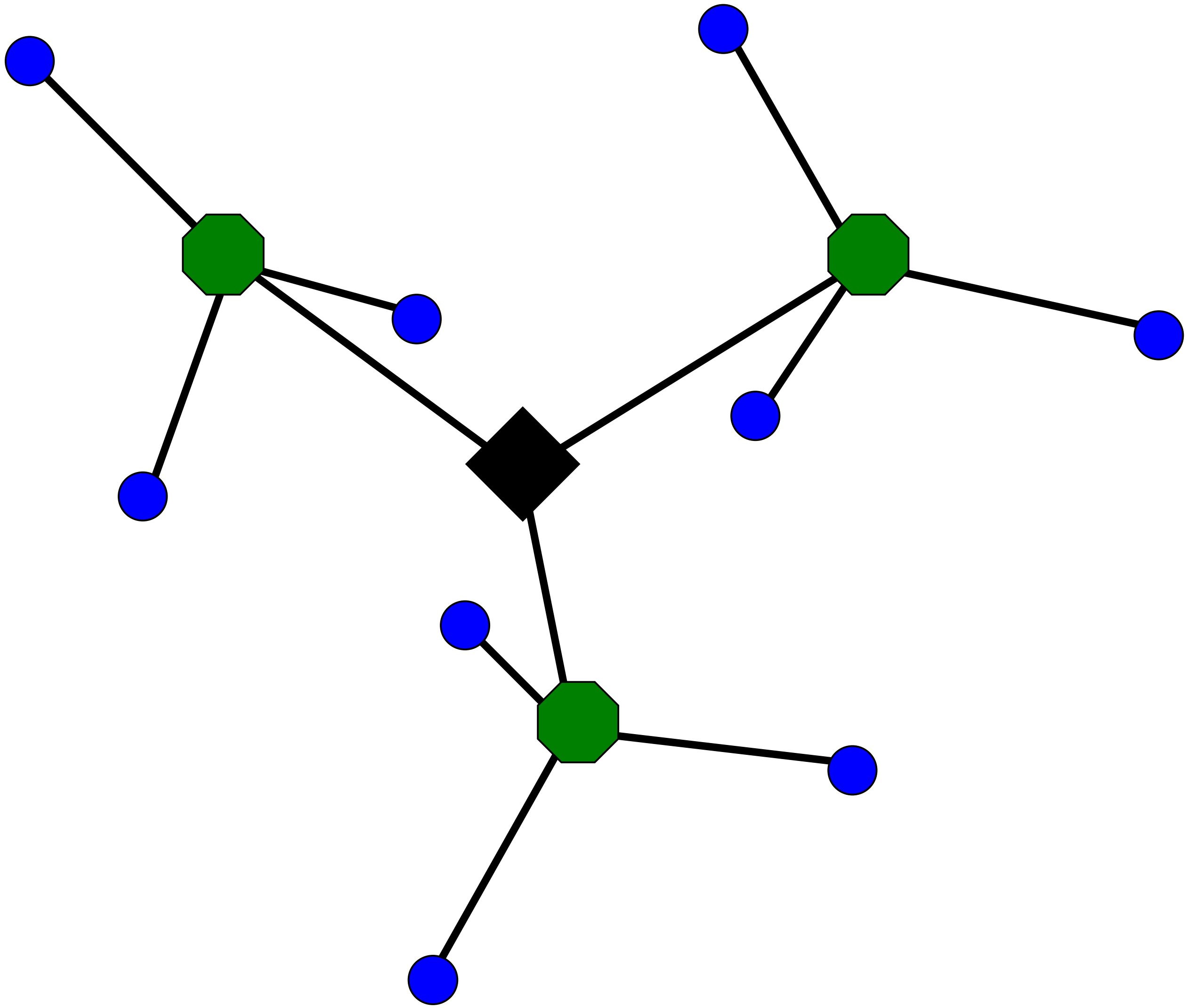


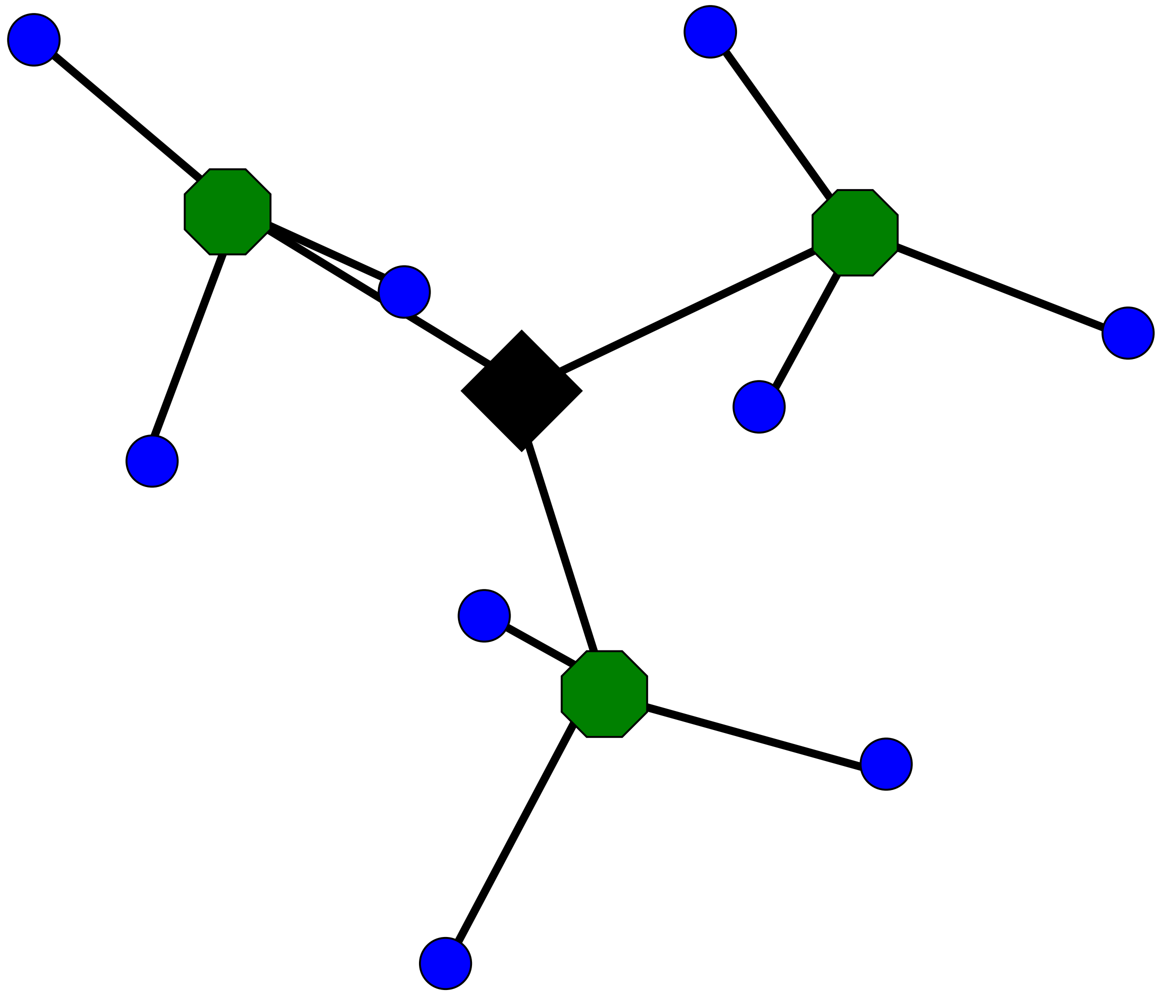


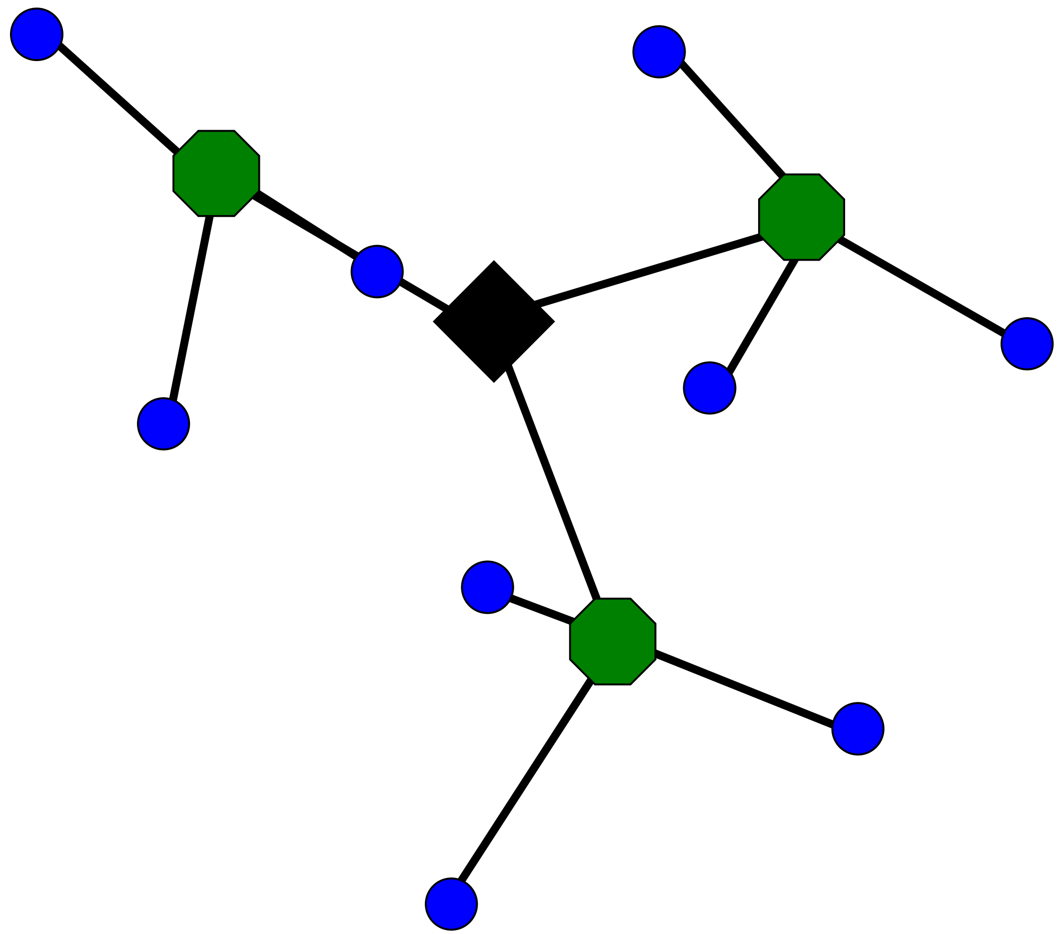


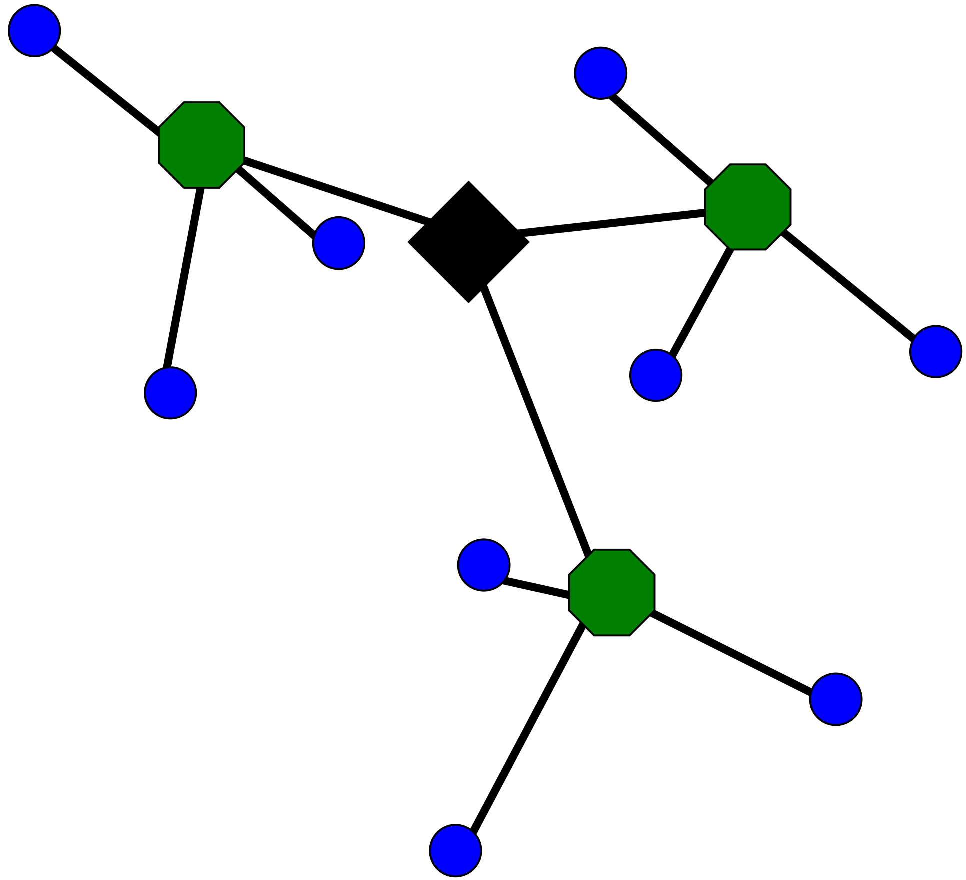


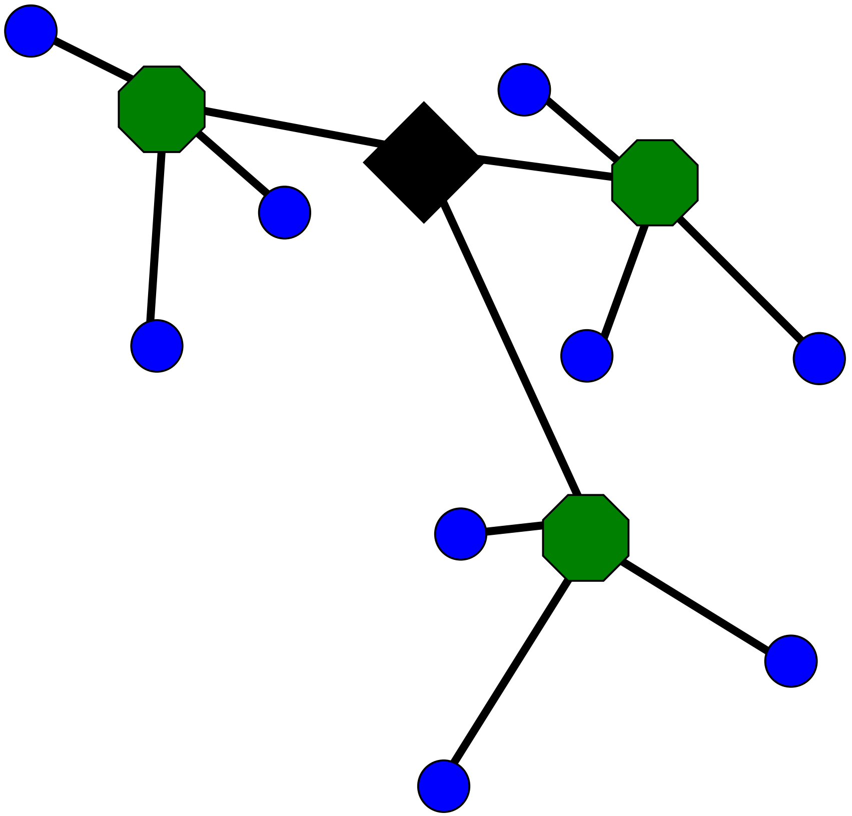


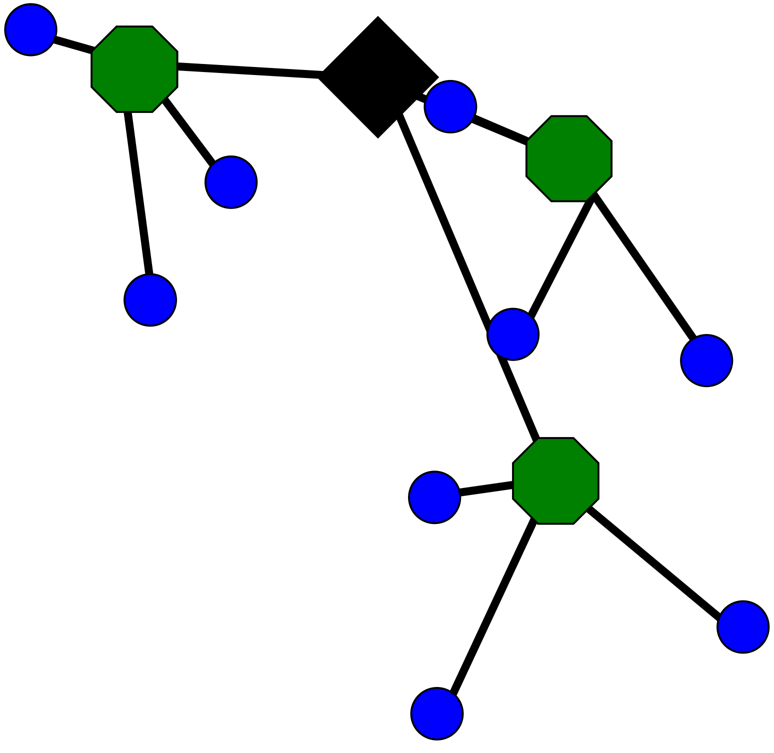


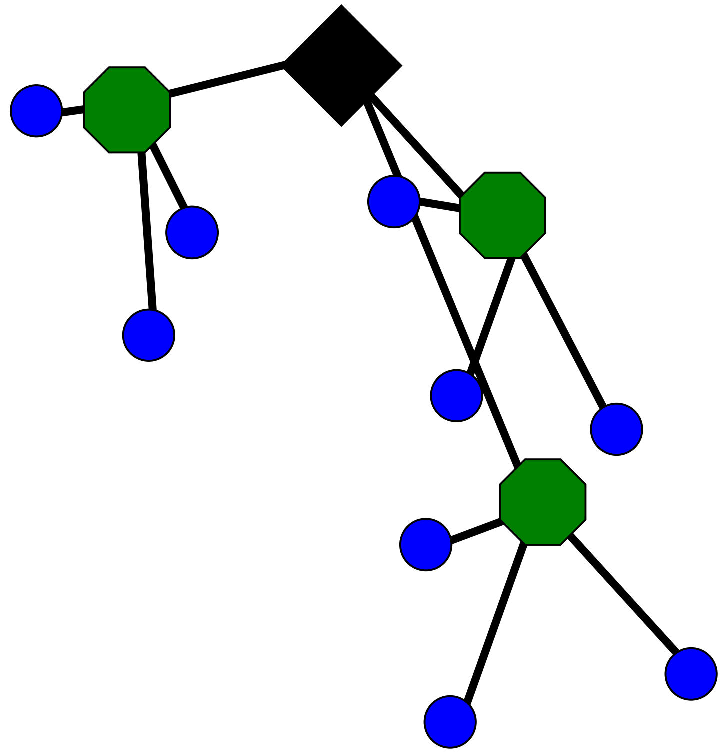


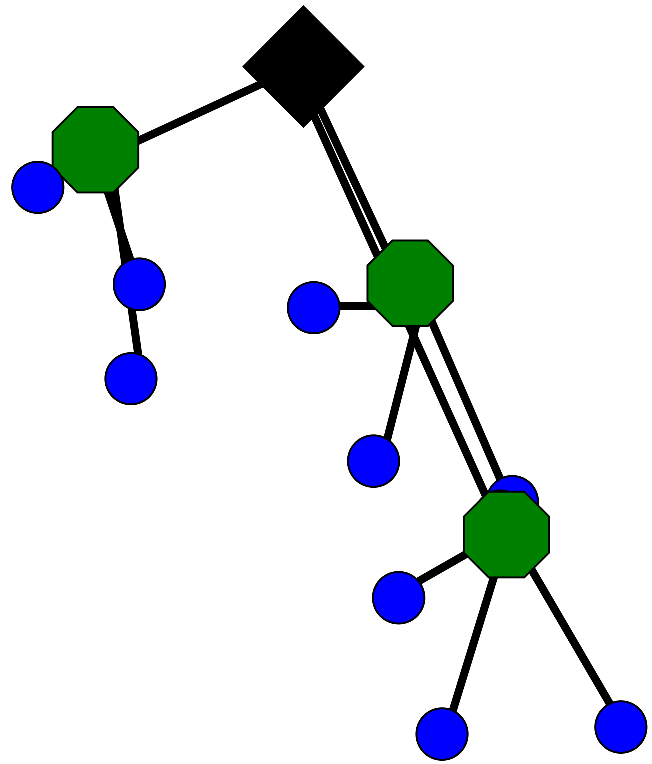


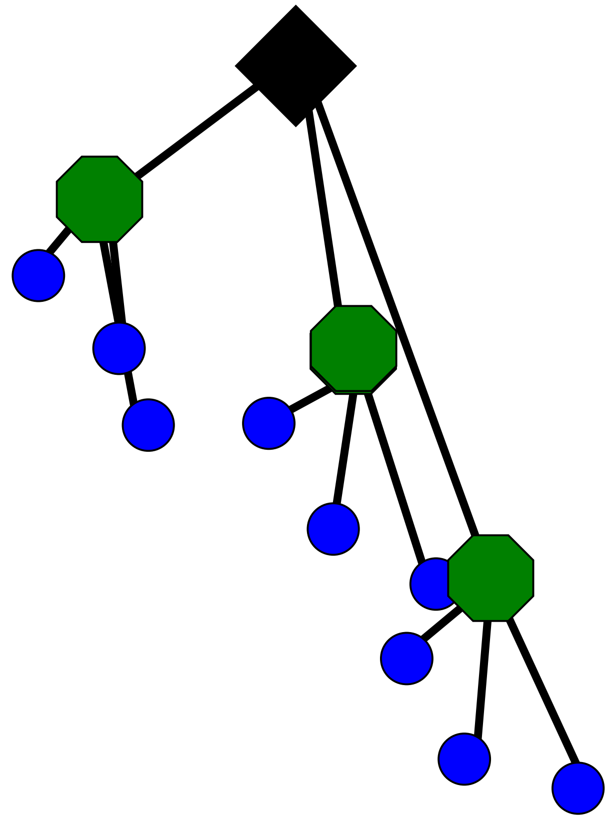


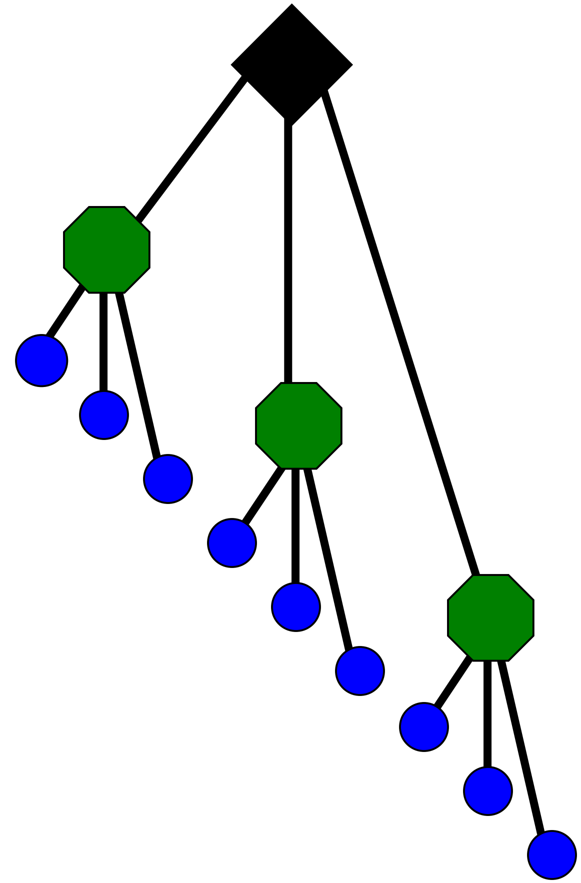


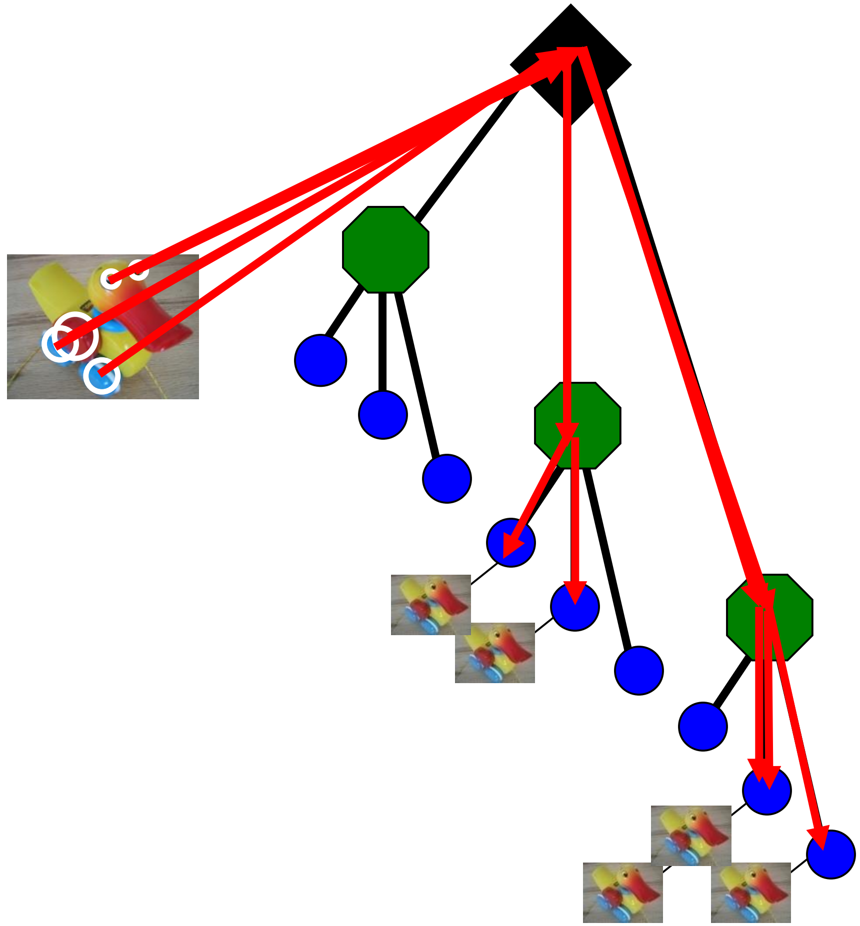


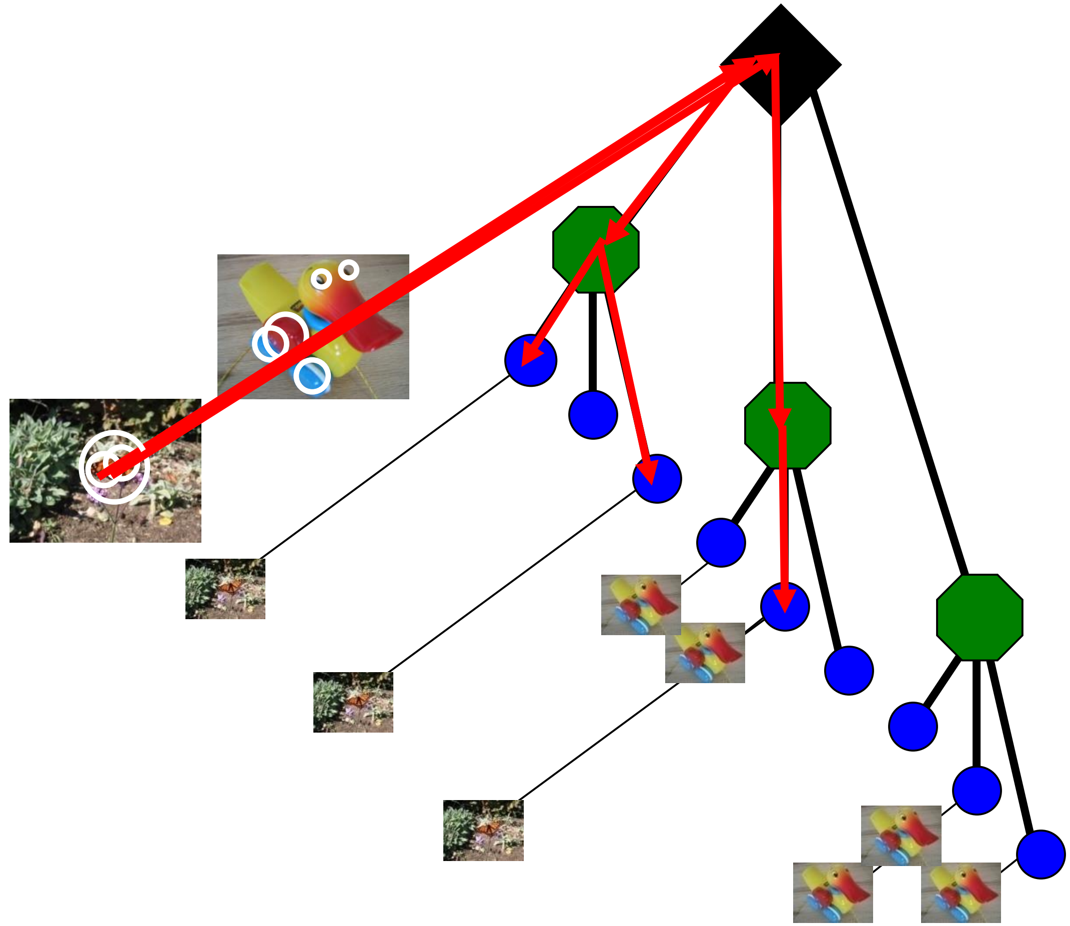


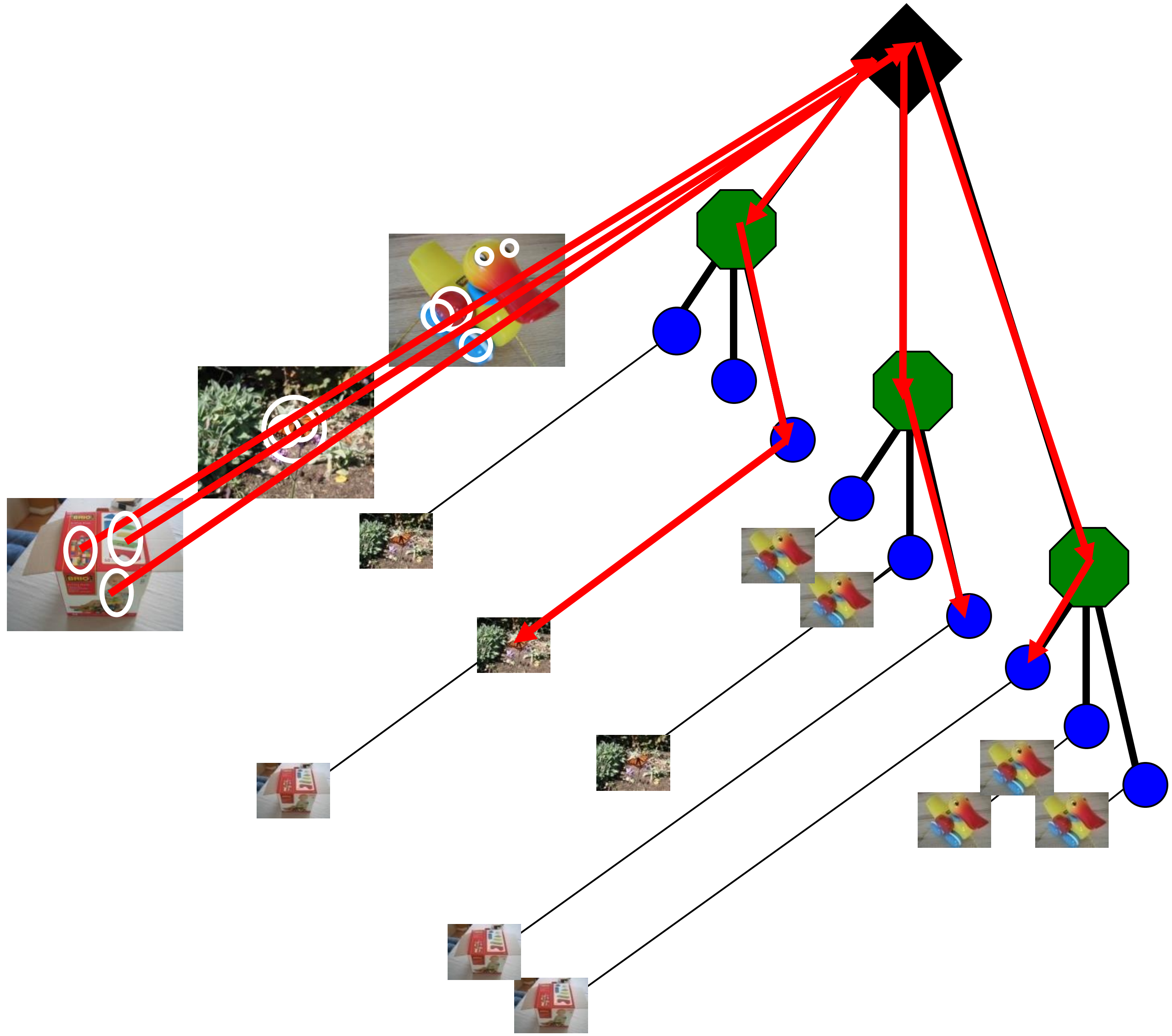


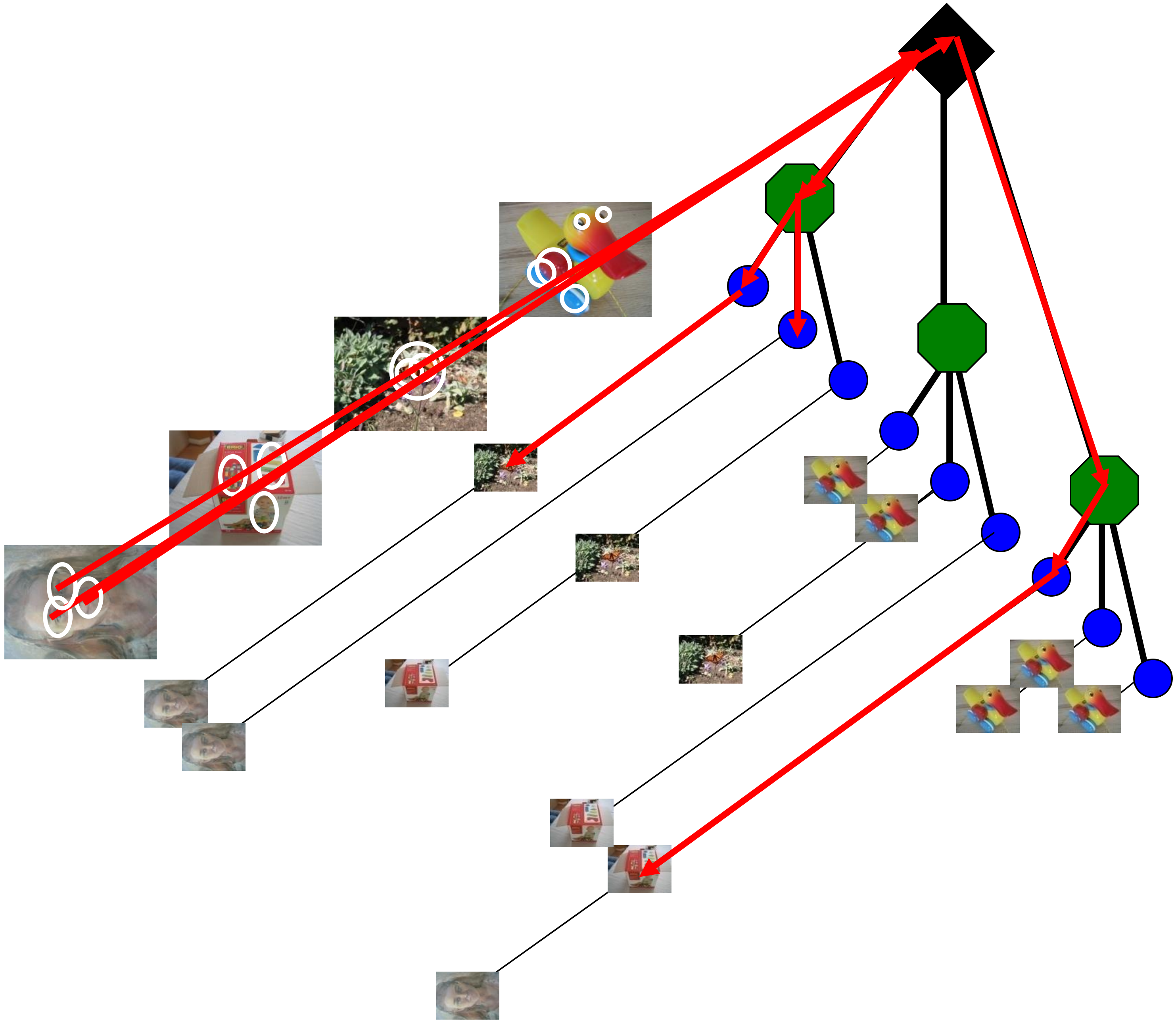


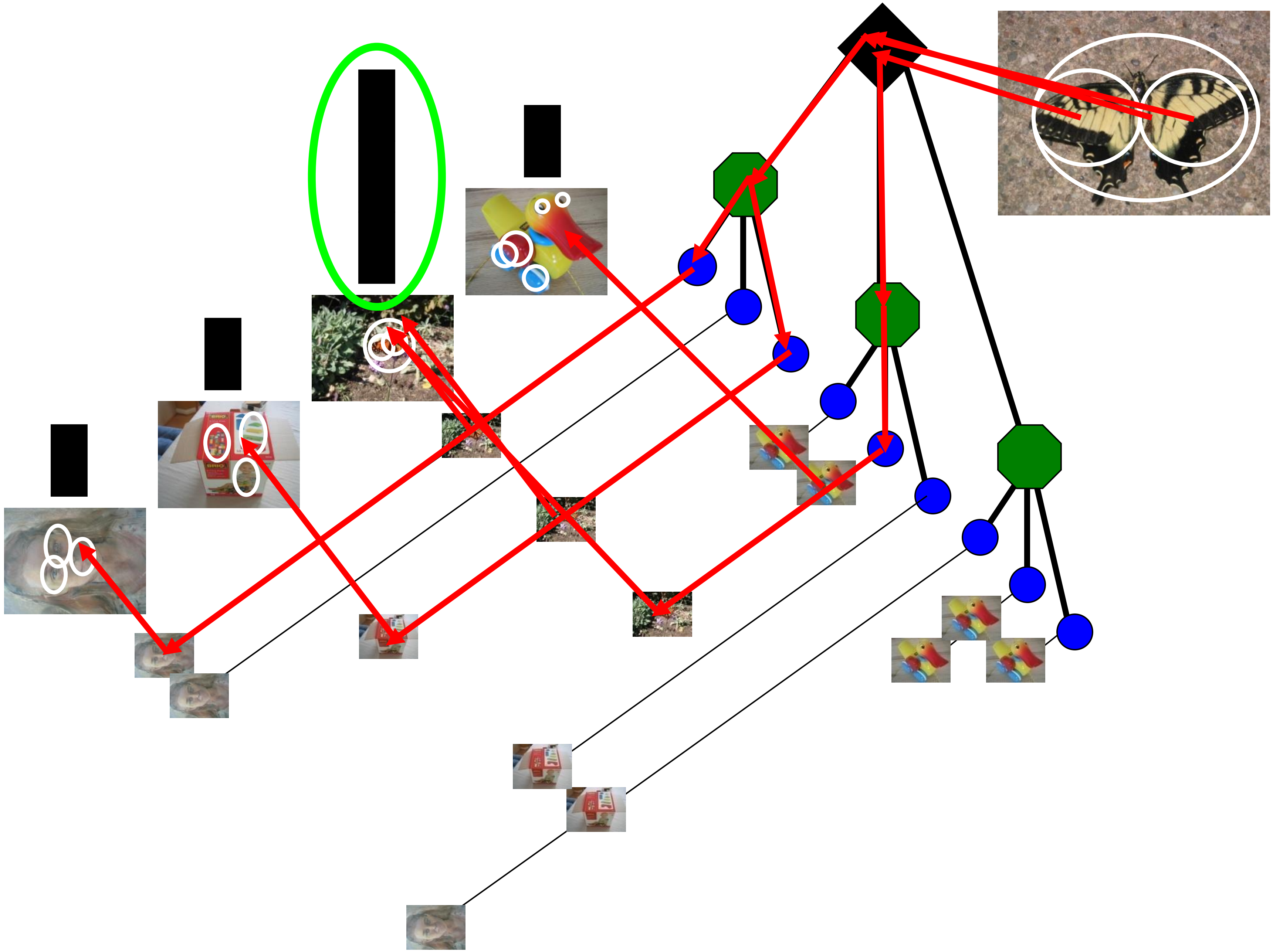




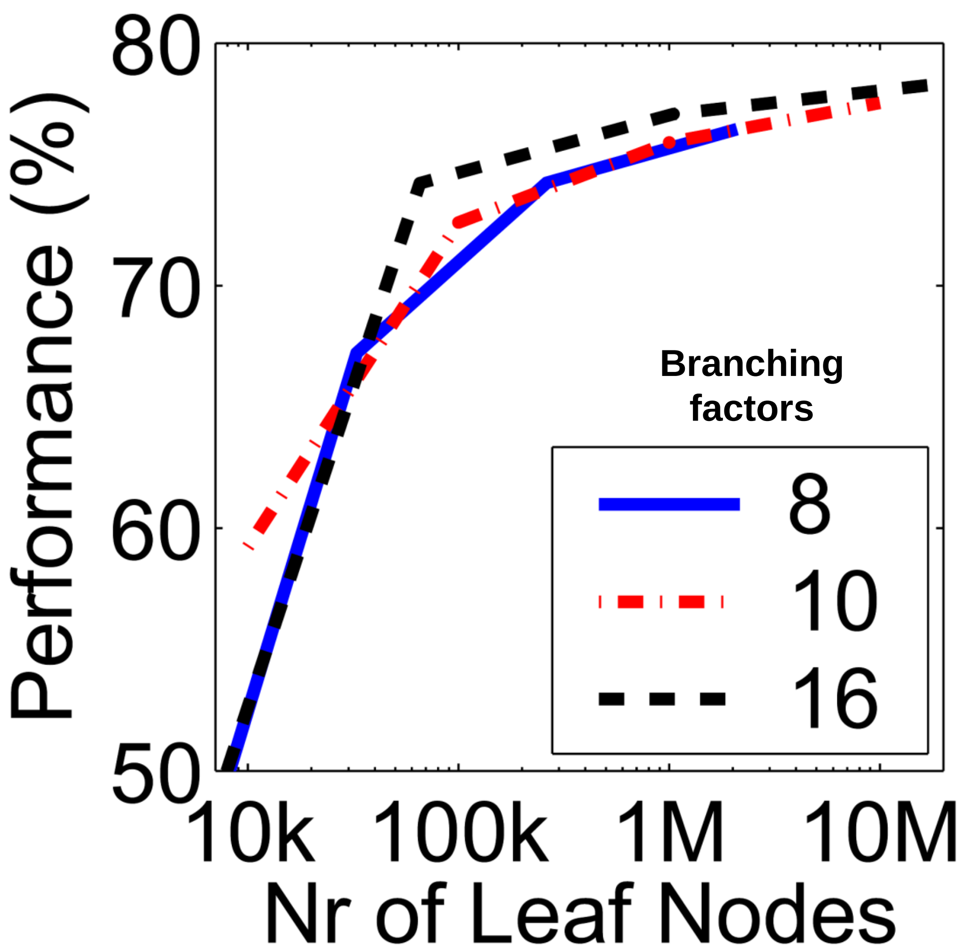




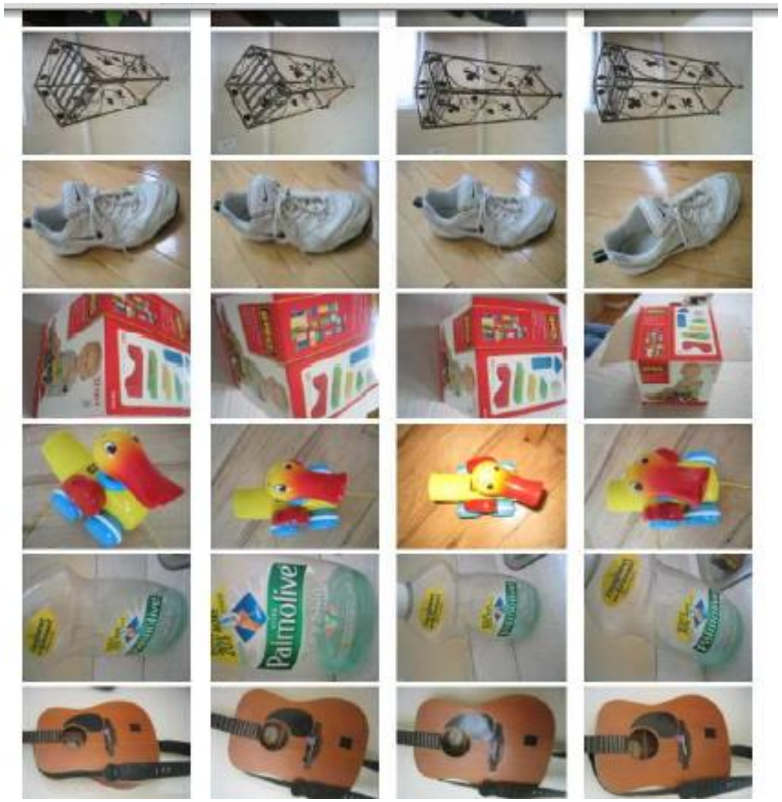




Vocabulary size



Recognition with 6347 images



Nister & Stewenius, CVPR 2006

Influence on performance, sparsity

Kristen Grauman

Vocabulary trees: complexity

Number of words given tree parameters:

$\text{branching\_factor}^{\wedge\text{number\_of\_levels}}$

Word assignment cost vs. flat vocabulary:

$O(k)$ for flat

$O(\log_{\text{branching\_factor}}(k) * \text{branching\_factor})$

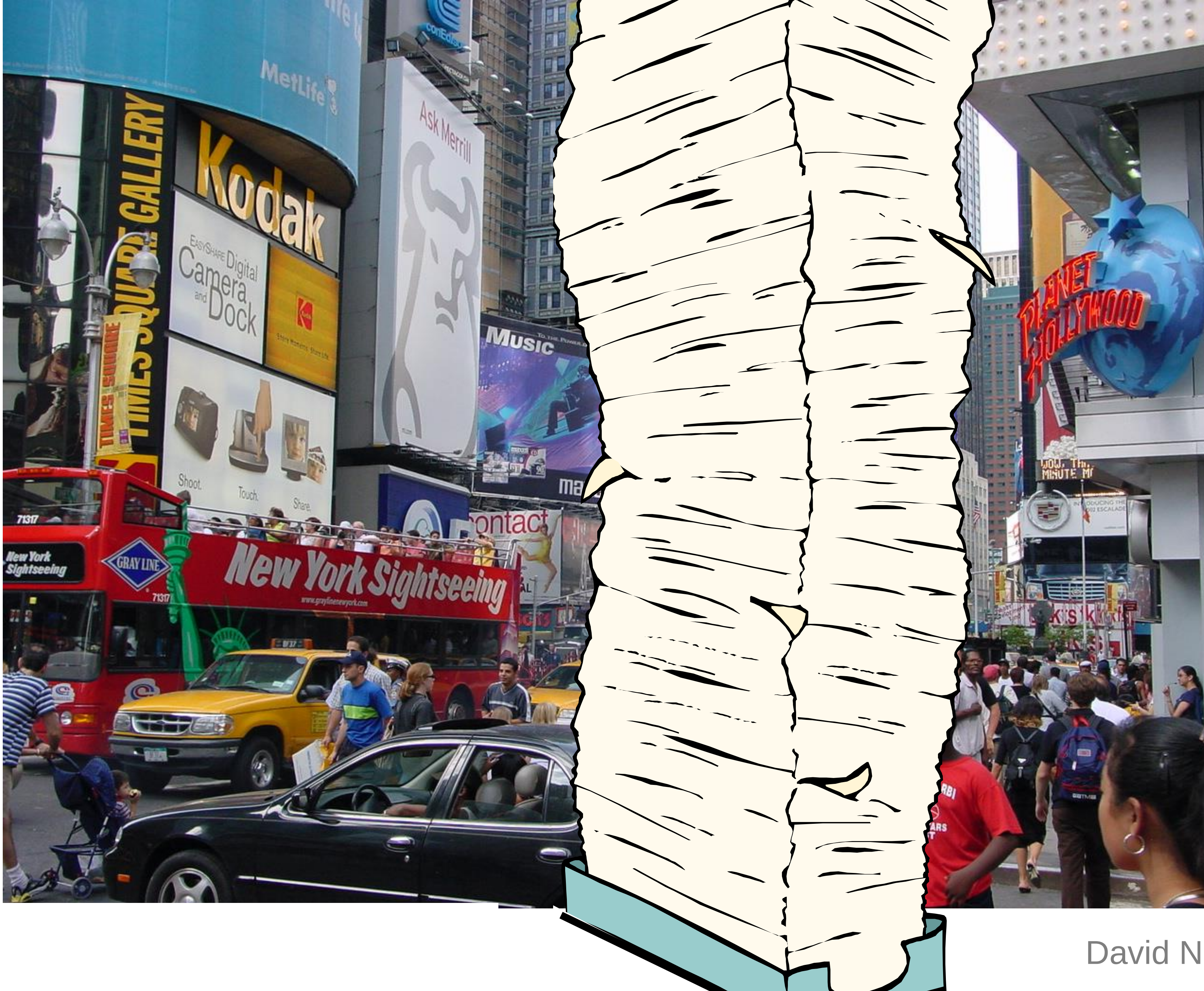
Is this like a kd-tree?

Yes, but with better partitioning and defeatist search.

This hierarchical data structure is lossy – you might not find your true nearest cluster.

(2006) 110,000,000 images in 5.8 Seconds

David Nister

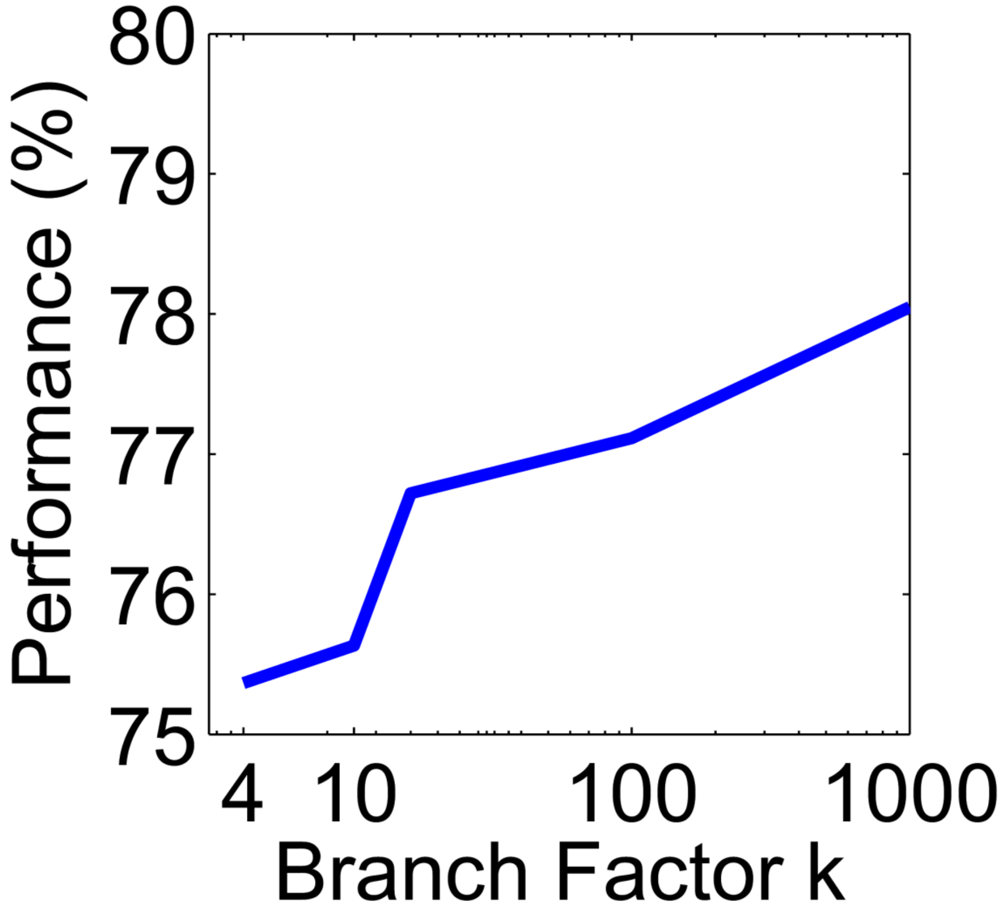


David Nister

David Nister

David Nister

Higher branch factor works better (but slower)



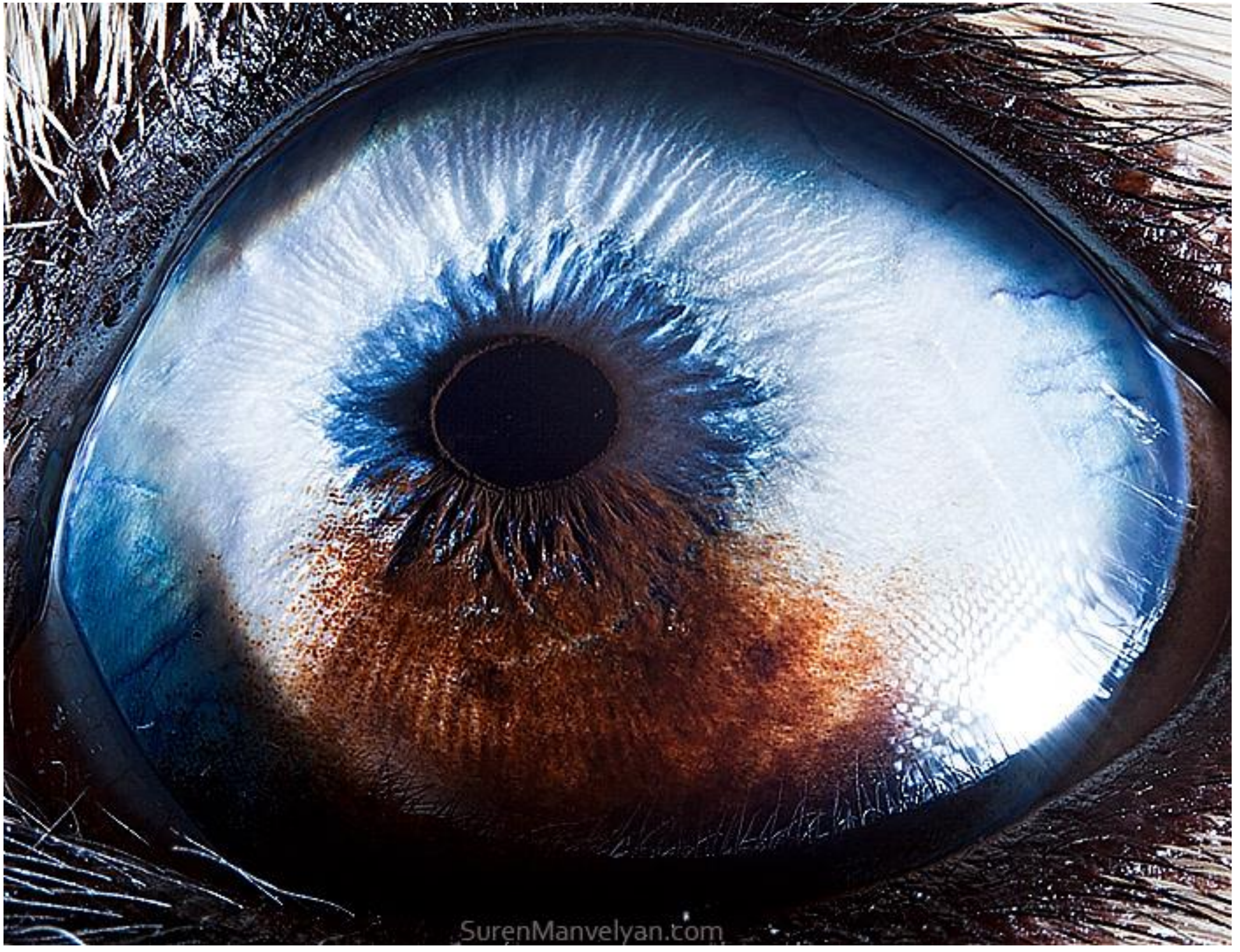
Visual words/bags of words

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

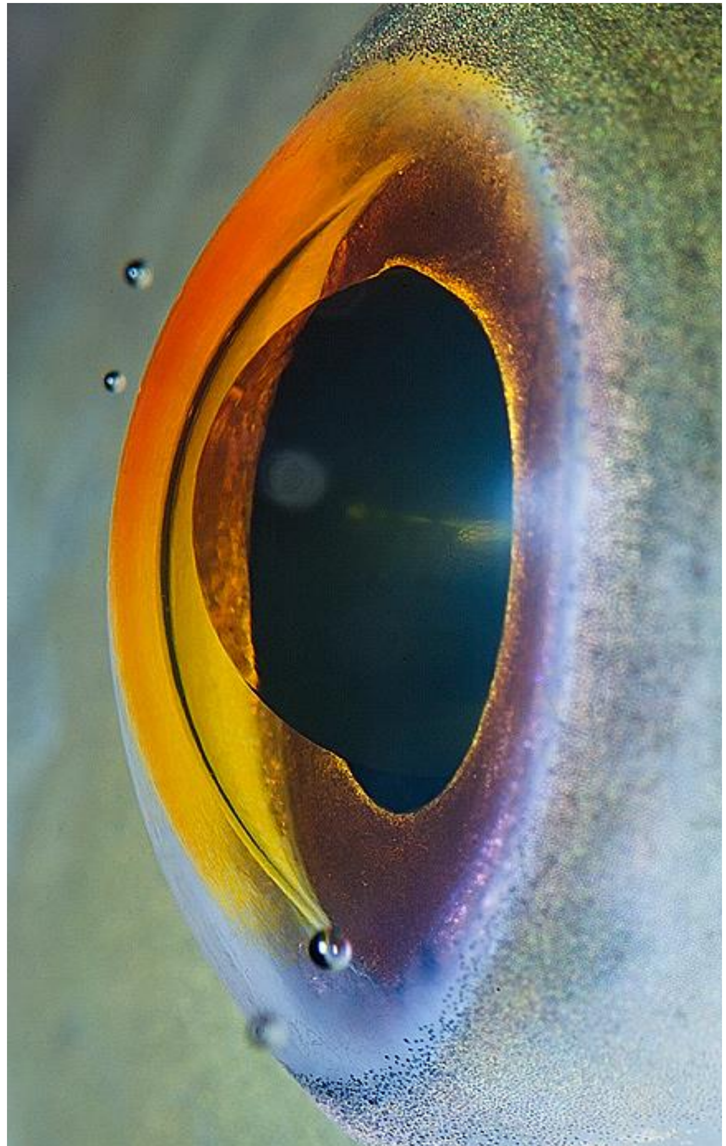
- background and foreground mixed when bag covers whole image -> *is it really instance recognition?*
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

Kristen Grauman

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116



By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116



By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

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?

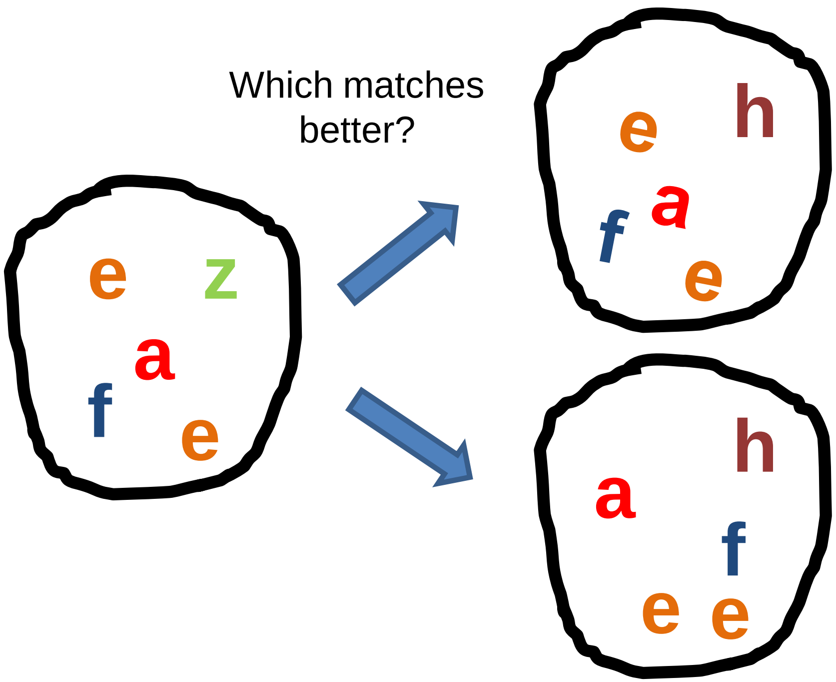
Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?

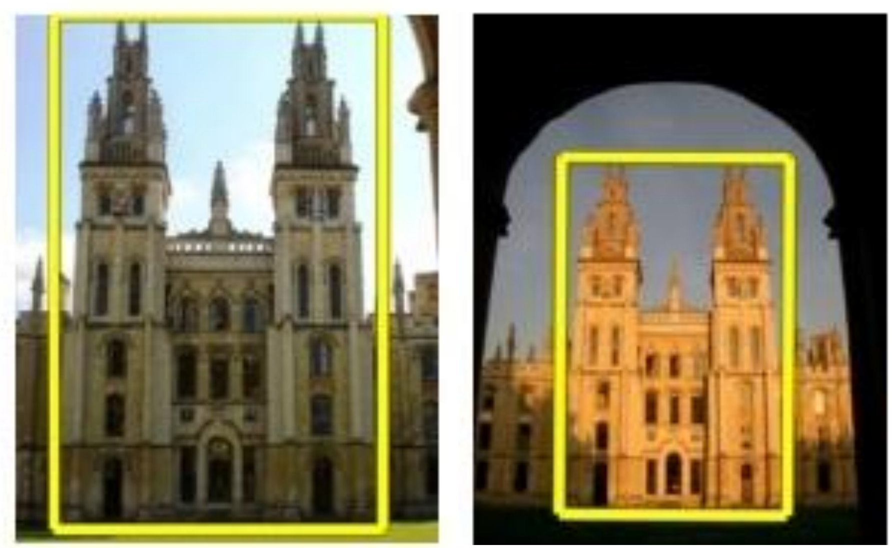
How to score the retrieval results?

Kristen Grauman

Can we be more accurate?

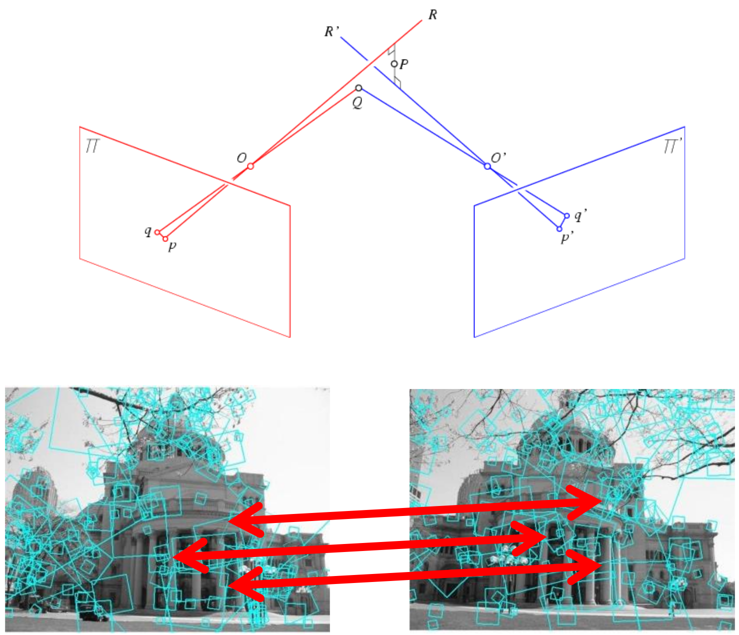
So far, we treat each image as containing a "bag of words", with no spatial information



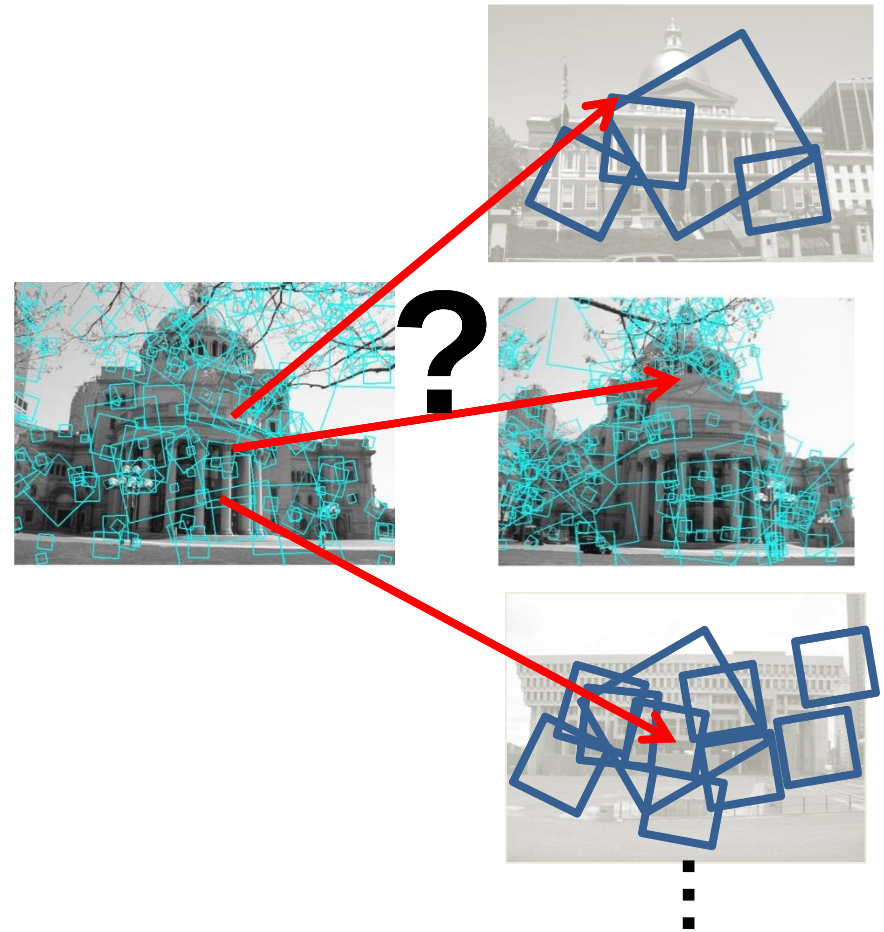


Real objects have consistent geometry

Multi-view matching



vs



Matching two given views for depth

Search for a matching view for recognition

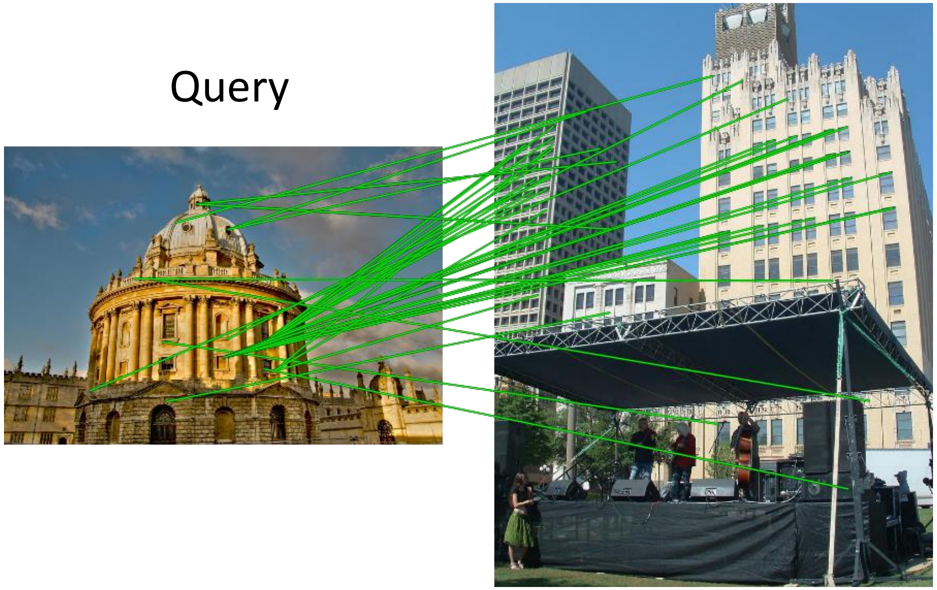
Kristen Grauman

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.

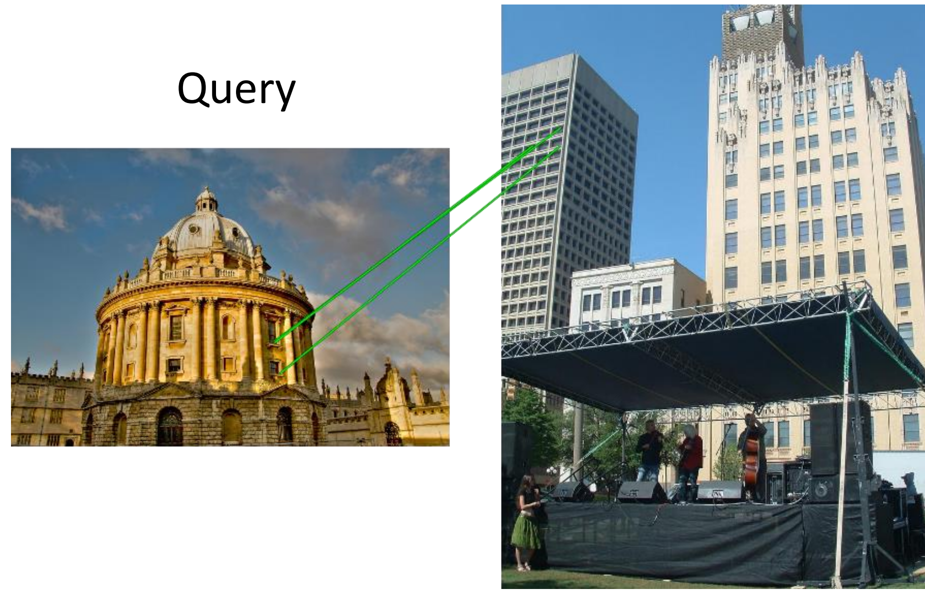
Slide credit: Ondrej Chum

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

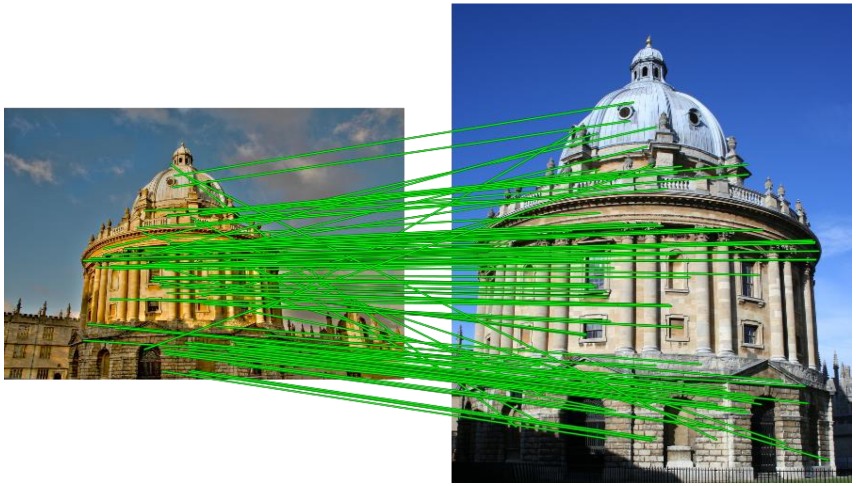
Spatial Verification: two basic strategies

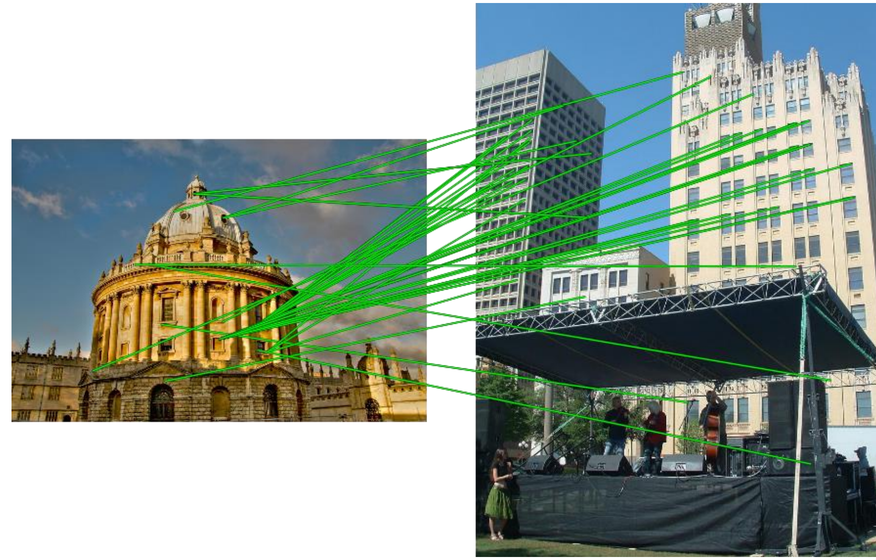
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., “success” if find a transformation with > N inlier correspondences

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

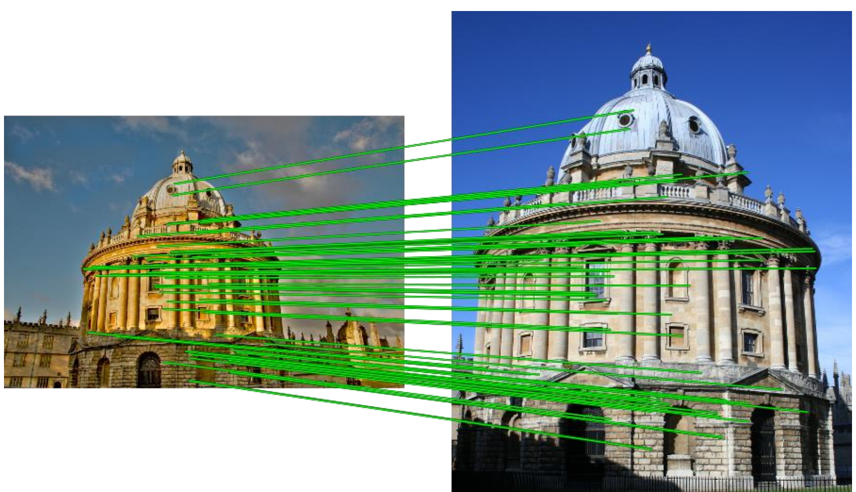
Kristen Grauman

No verification





RANSAC verification



Fails to meet threshold
on # inliers! Good!

Recognition via alignment

Pros:

- Effective for reliable features within clutter
- Great for matching specific instances

Cons:

- Expensive post-process (how long for proj3?!)
- Not suited for category recognition

Kristen Grauman

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?

Kristen Grauman

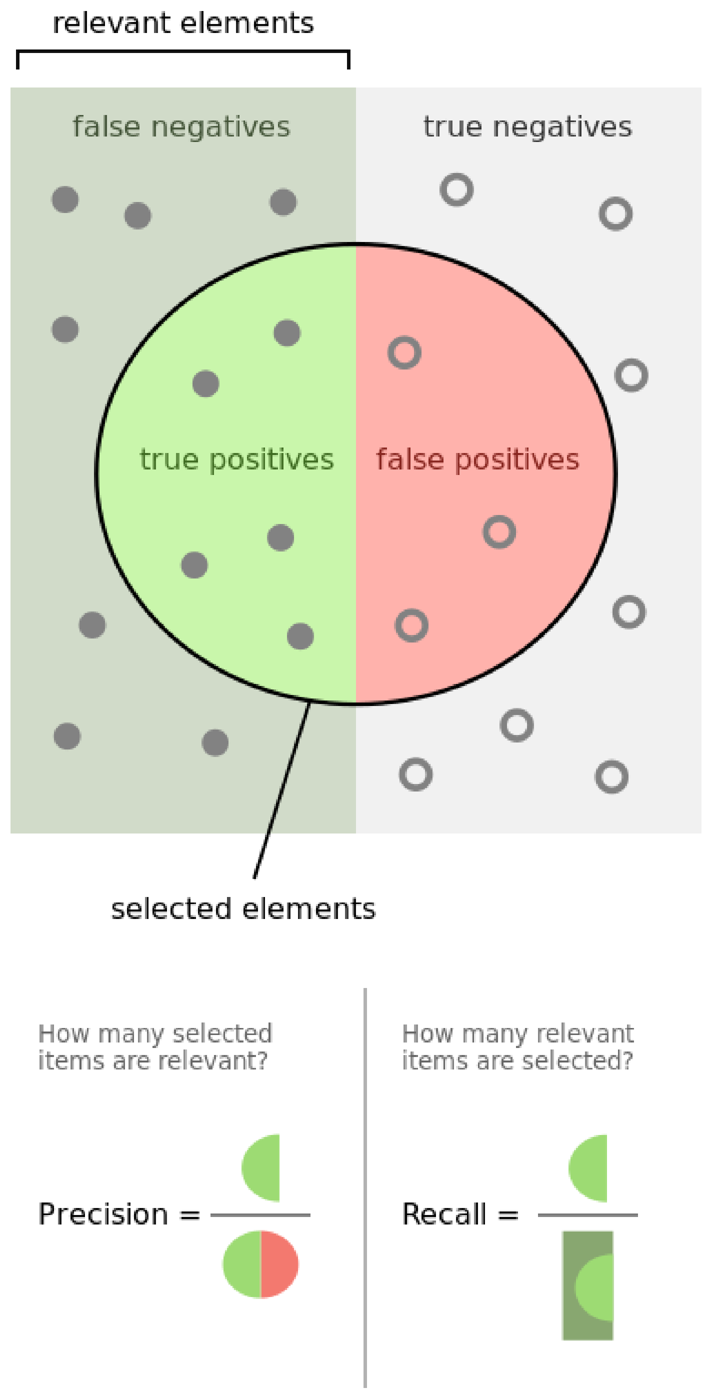
Precision and Recall

True positive (tp) – correct attribution
True negative (tn) – correct rejection

False positive (fp) – incorrect attribution
False negative (fn) – incorrect rejection

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

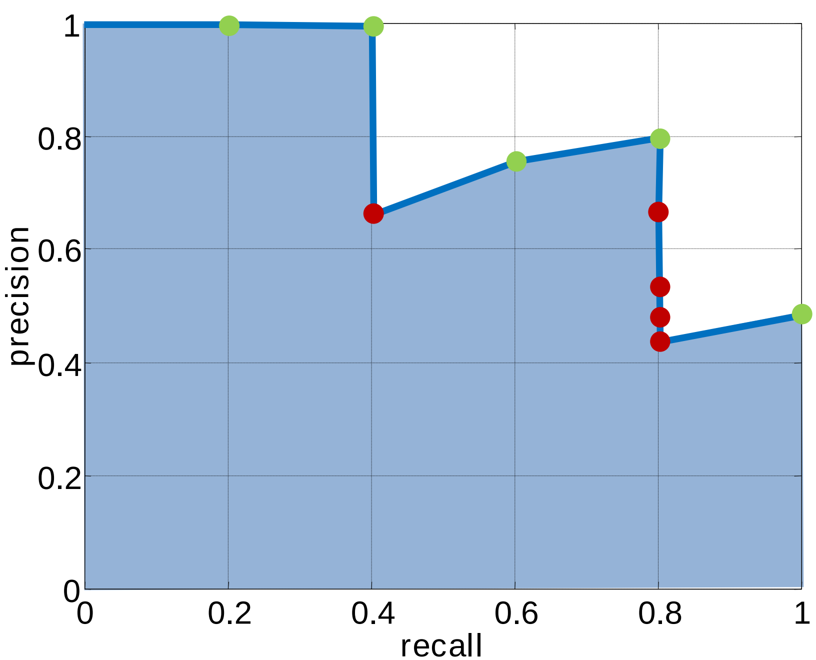


Scoring retrieval quality

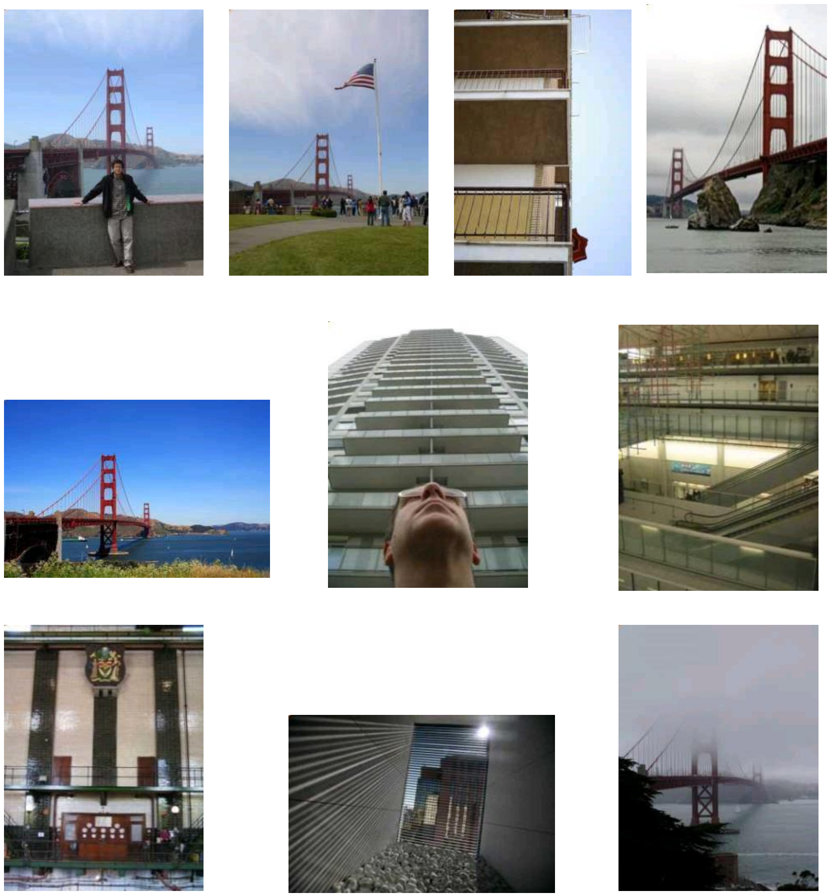
Query

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

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



Results (ordered):



Slide credit: Ondrej Chum

Query expansion

Chum, Philbin, Sivic, Isard, Zisserman: Total Recall..., ICCV 2007 Ondrej Chum

Summary

- **Matching local invariant features**
  - Useful for multi-view geometry and to find objects/scenes.
- **Bag of words**: quantize feature space into discrete visual words
  - Summarize image by distribution of words
- **Inverted index**: visual word index for faster query time
- **Recognition of instances via alignment:** matching local features followed by spatial verification
  - Robust fitting : RANSAC, Generalized Hough Transform

Kristen Grauman

Lessons from a decade later

For *Category* recognition (project 4)

- 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.

Lessons from a decade later

For *instance* retrieval (this lecture):

- deep learning is taking over.
- learn better local features (replace SIFT)
  e.g., MatchNet 2015
- learn better image embeddings (replace visual word histograms)
  e.g., Vo and Hays 2016.
- learn spatial verification
  e.g., DeTone, Malisiewicz, and Rabinovich 2016.
- learn a monolithic deep network to recognition all locations
  e.g., Google's PlaNet 2016.

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116