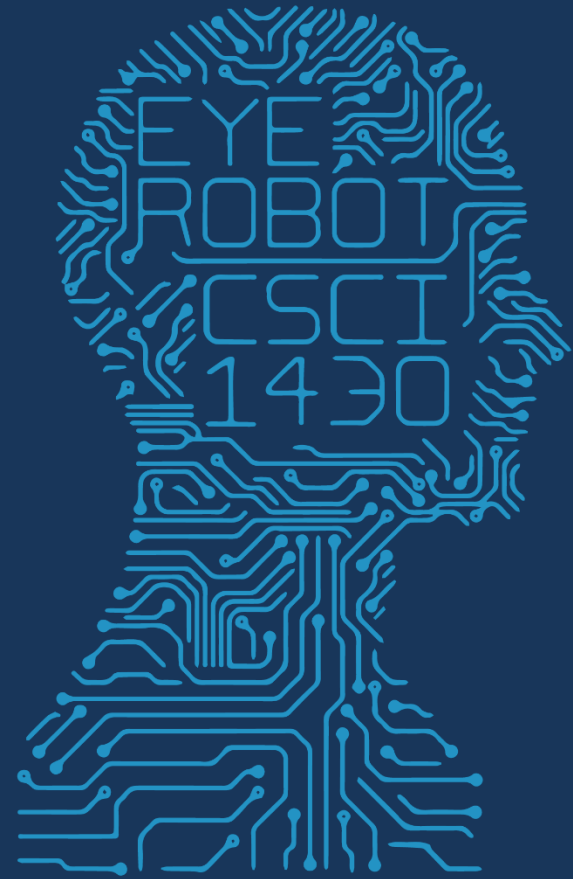




1950

FUTURE VISION



2017 MWF 1PM 368

COMPUTER VISION









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

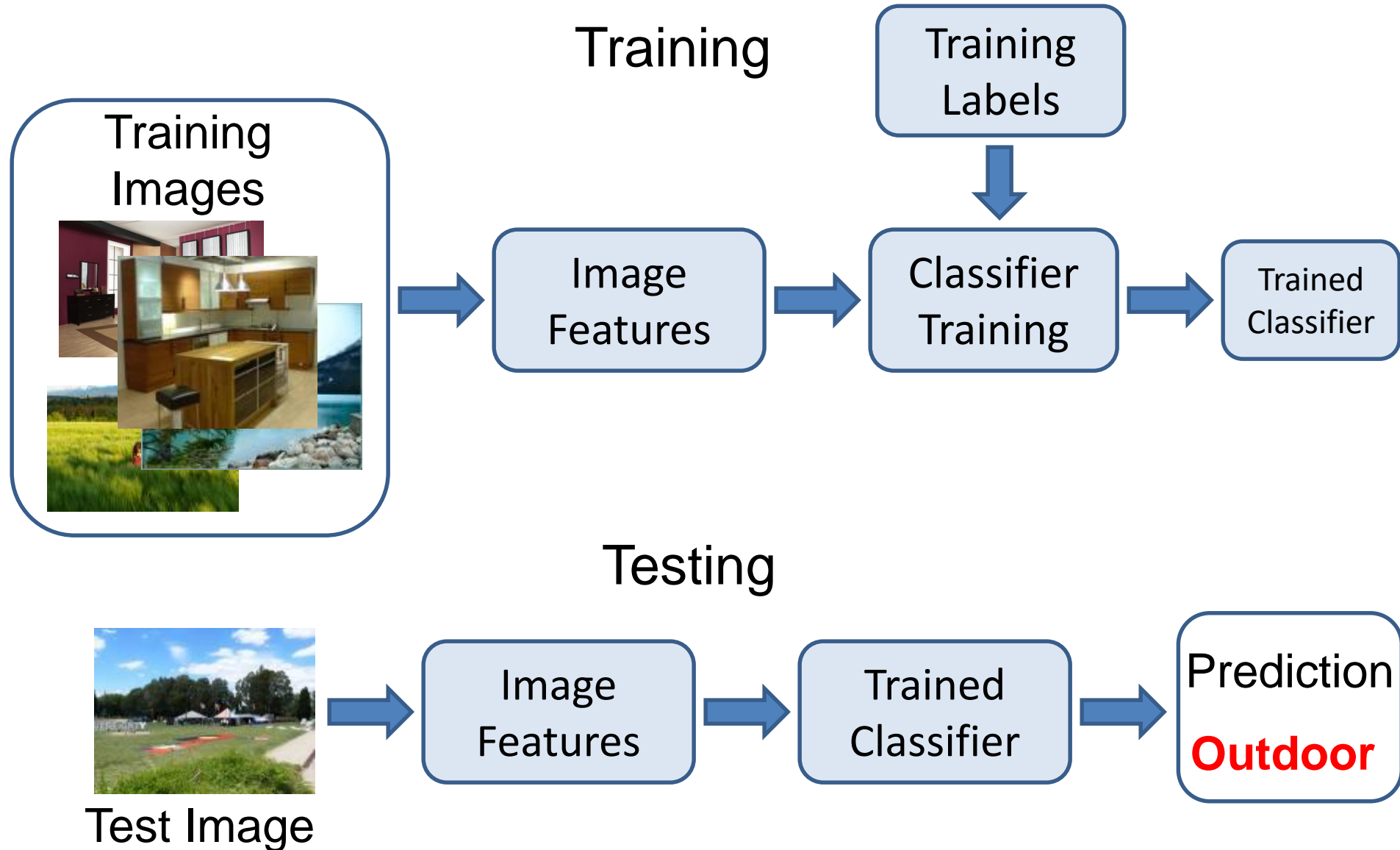


Complete heterochromia in human eyes: one brown and one green/hazel

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?

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?

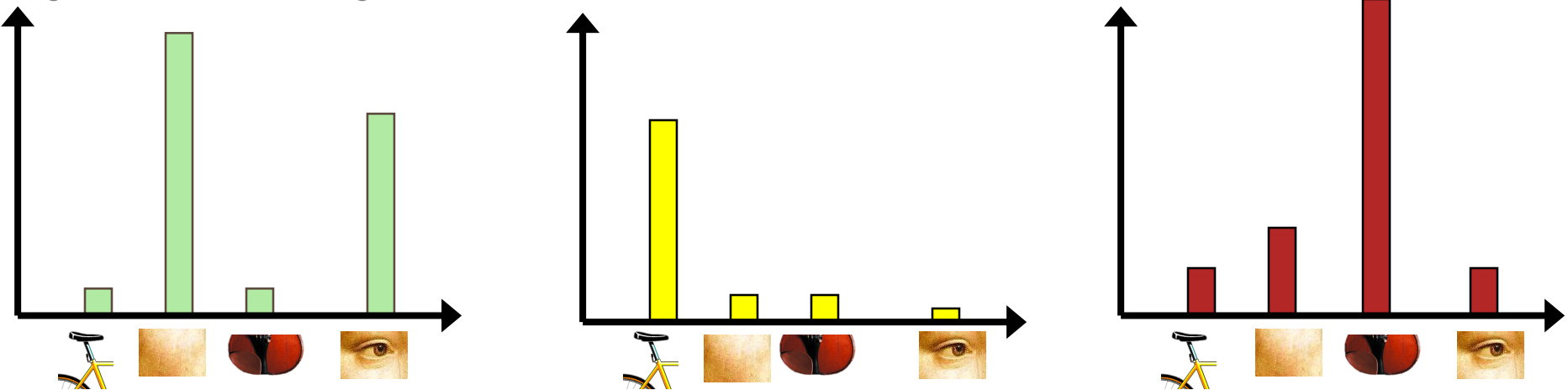
How to score the retrieval results?



Visual words

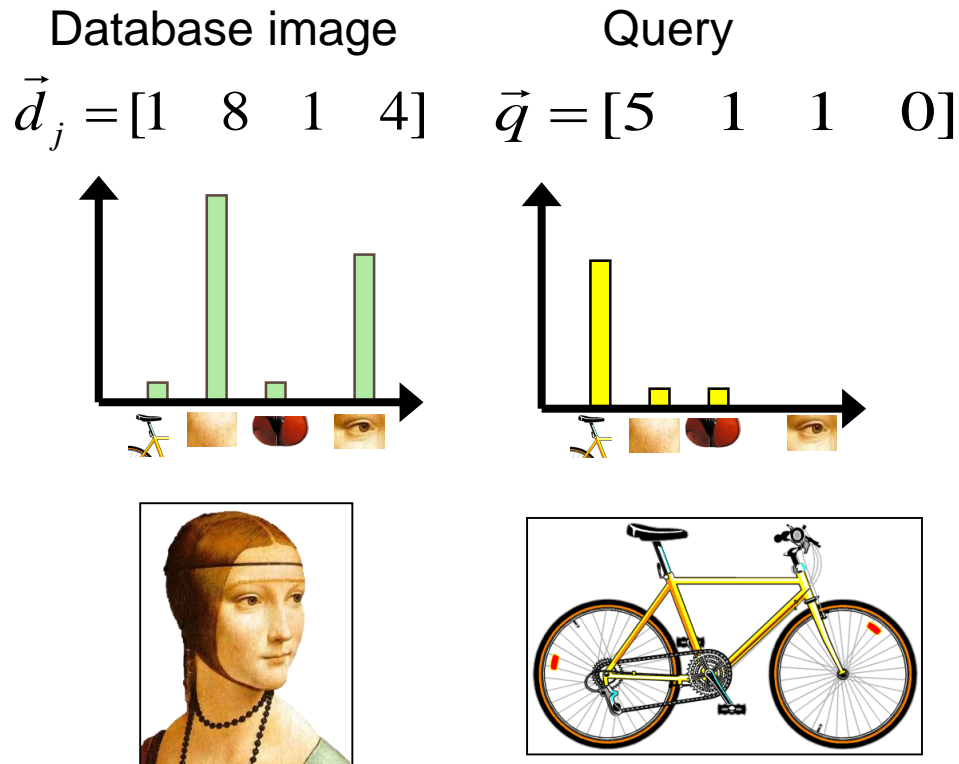


Bag of visual words histograms



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.



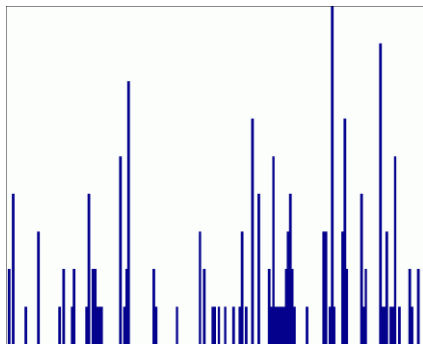
$$\text{sim}(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

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

for vocabulary of V words

Spatial pyramid representation

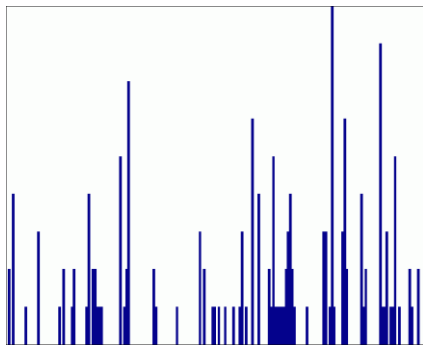
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



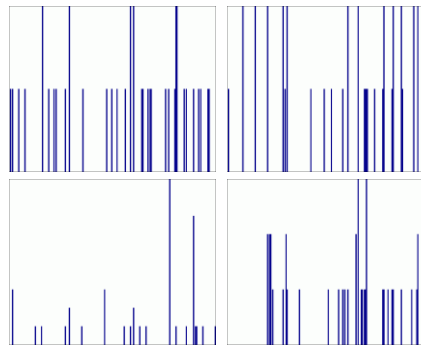
level 0

Spatial pyramid representation

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



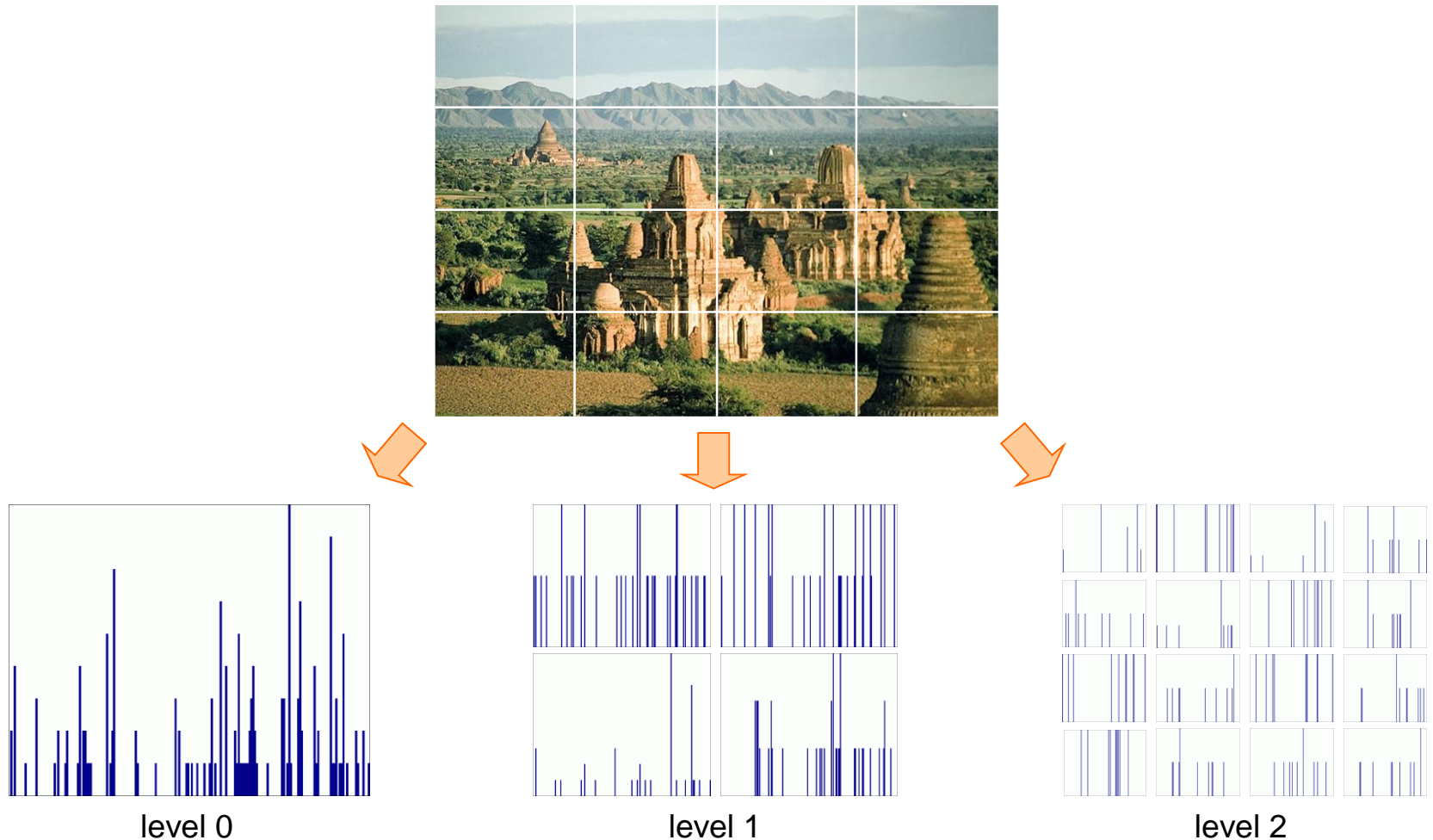
level 0



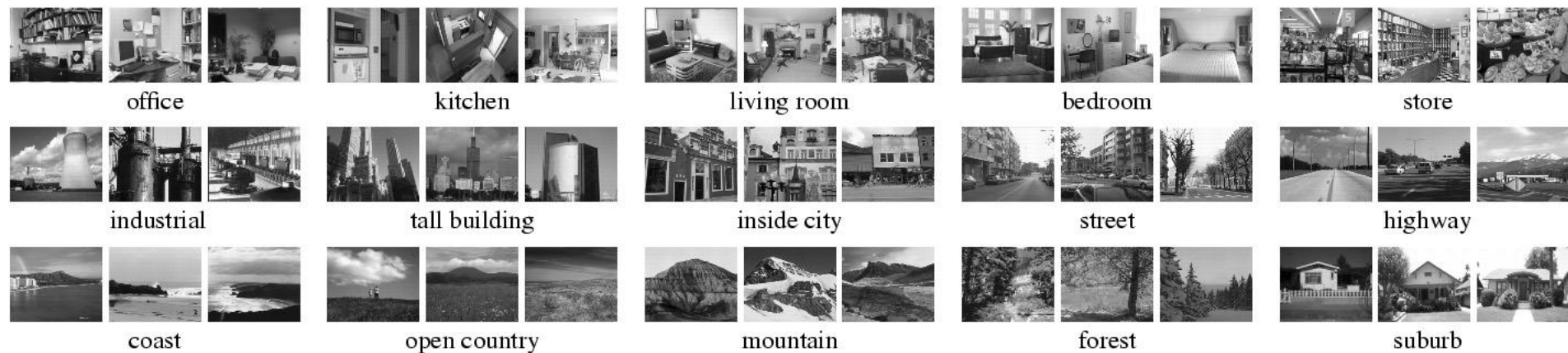
level 1

Spatial pyramid representation

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



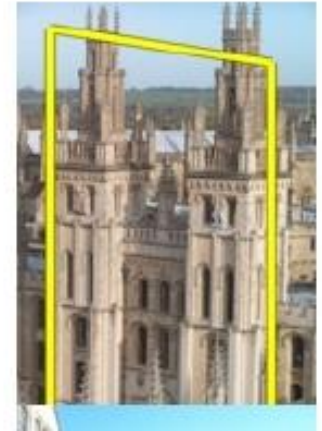
Scene category dataset



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

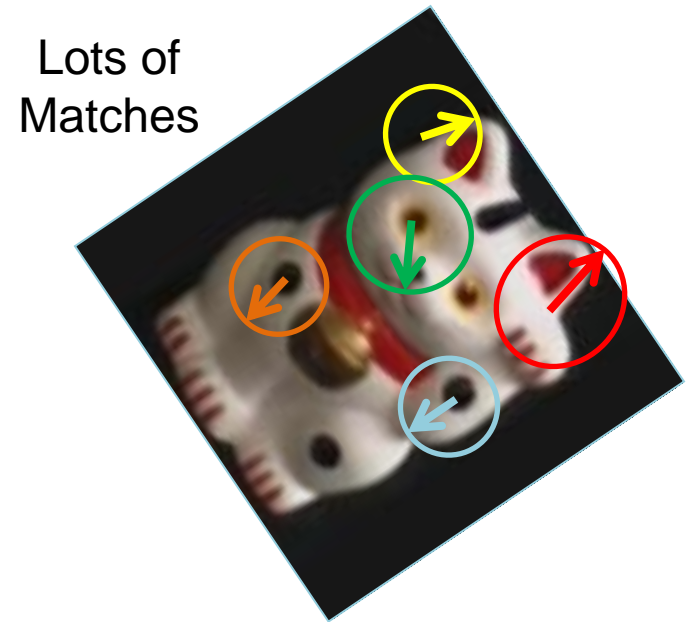
	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 \pm 0.5		72.2 \pm 0.6	
1 (2×2)	53.6 \pm 0.3	56.2 \pm 0.6	77.9 \pm 0.6	79.0 \pm 0.5
2 (4×4)	61.7 \pm 0.6	64.7 \pm 0.7	79.4 \pm 0.3	81.1 \pm 0.3
3 (8×8)	63.3 \pm 0.8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

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



Simple idea

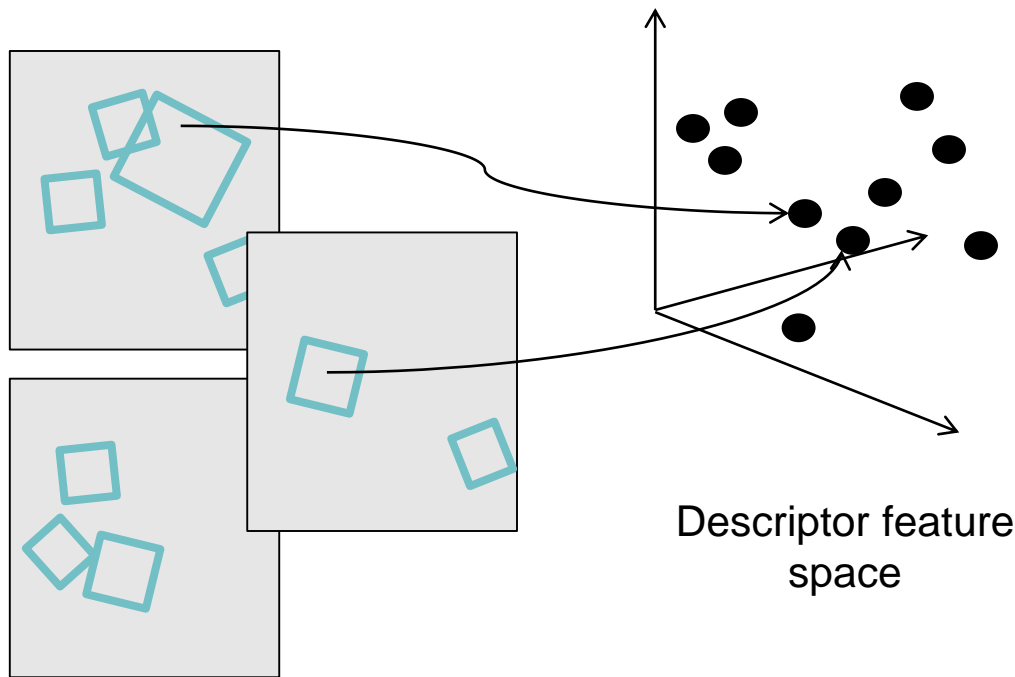
See how many keypoints are close to keypoints in each other image



But this will be really, really slow!

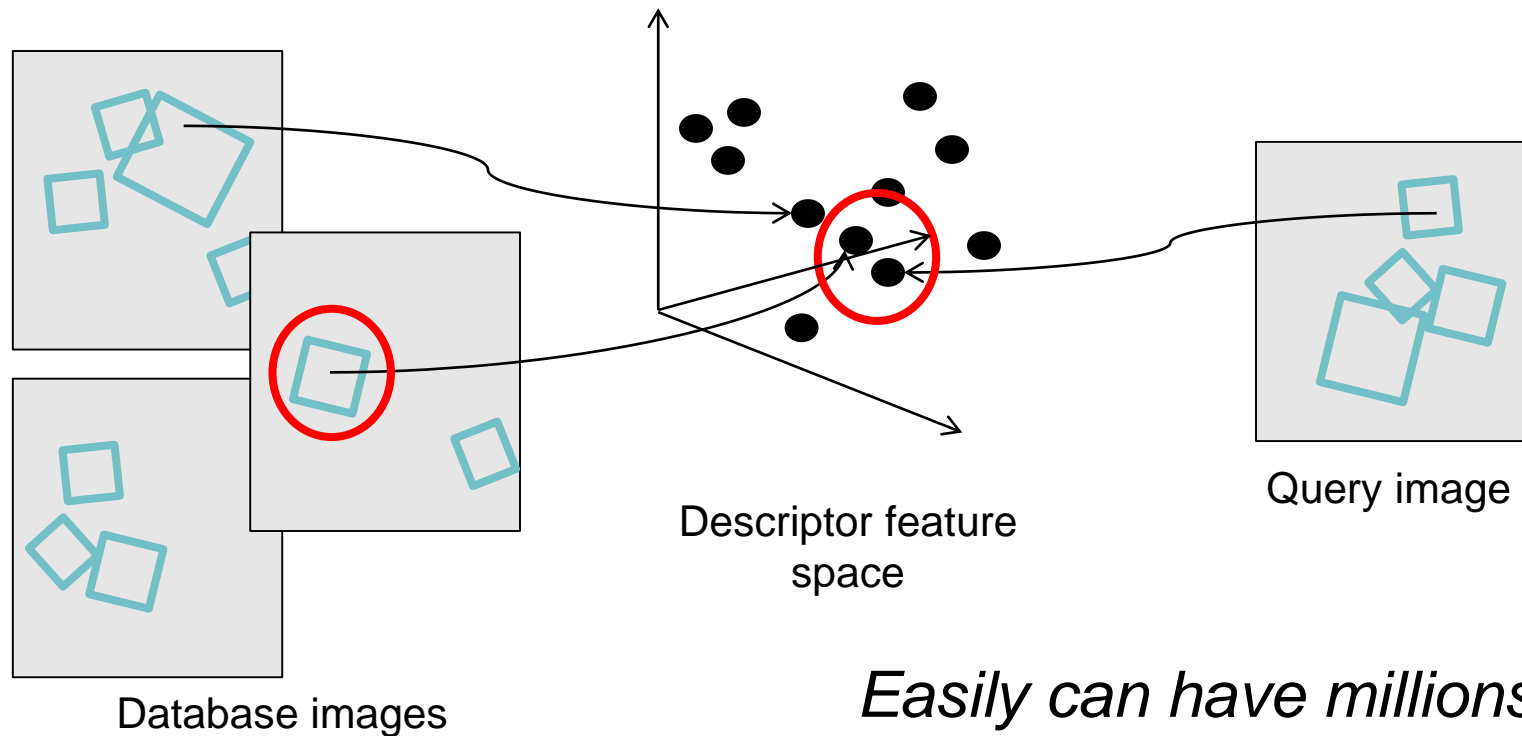
Indexing local features

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



Indexing local features

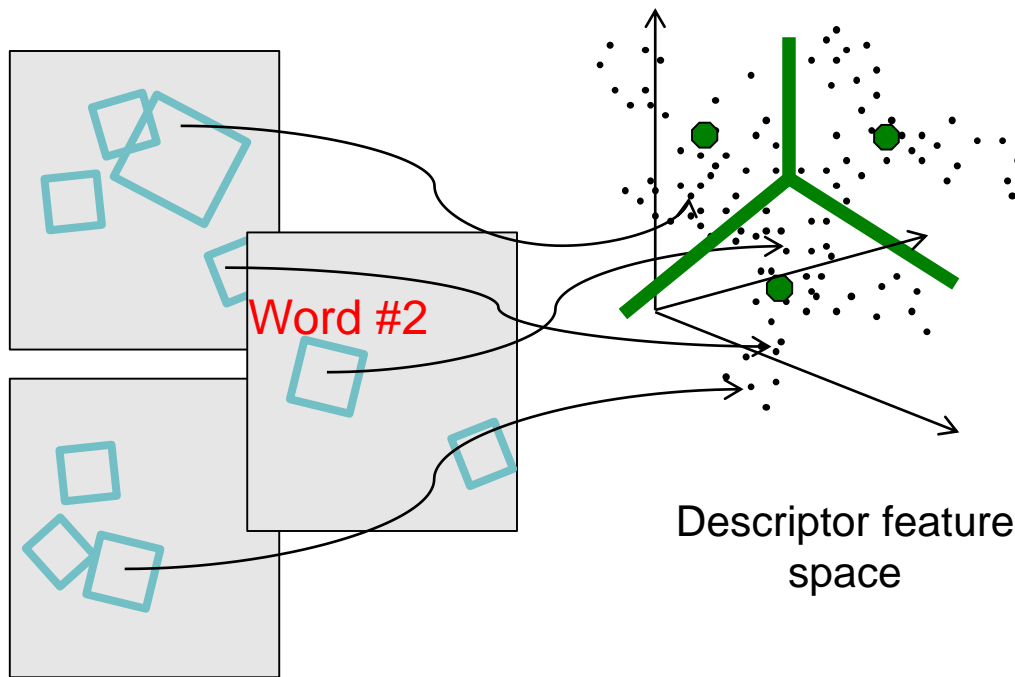
- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



Easily can have millions of features to search!

Visual words

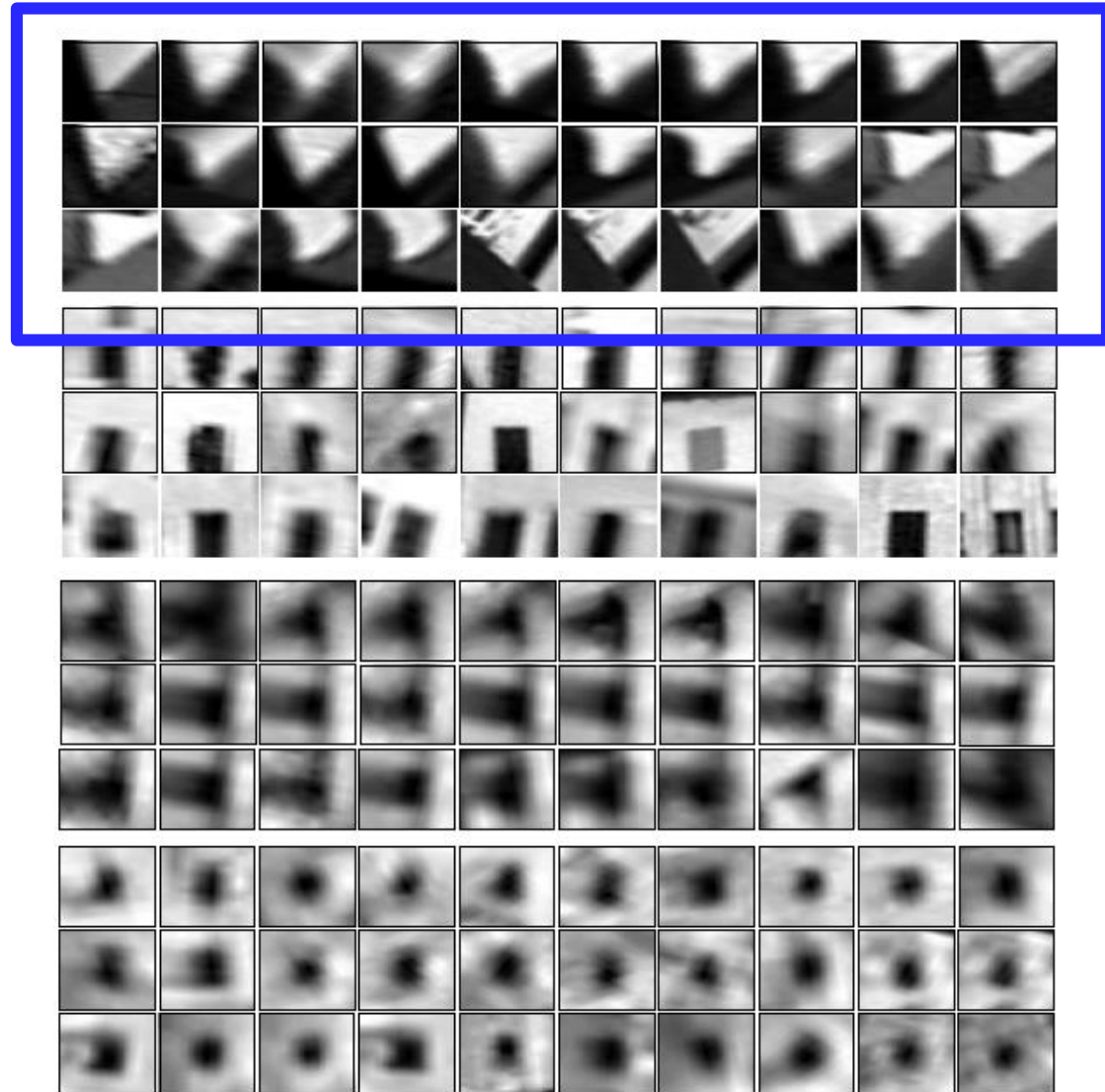
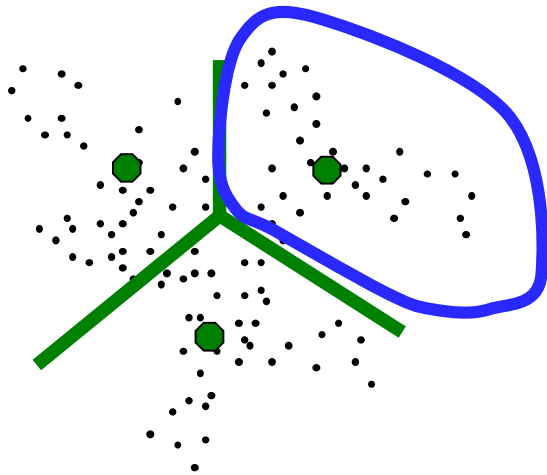
Map high-dimensional descriptors to tokens/words by quantizing the feature space.



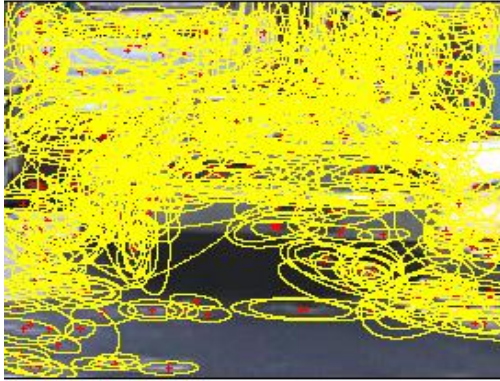
- Quantize via clustering; cluster centers are the visual “words”
- Assign word to each image region by finding the closest cluster center.

Visual words

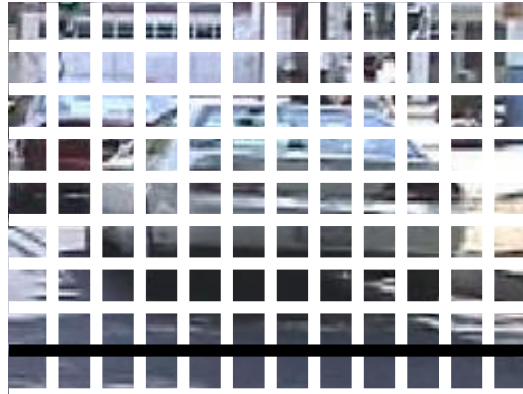
- Example: each group of patches belongs to the same visual word



Sampling strategies



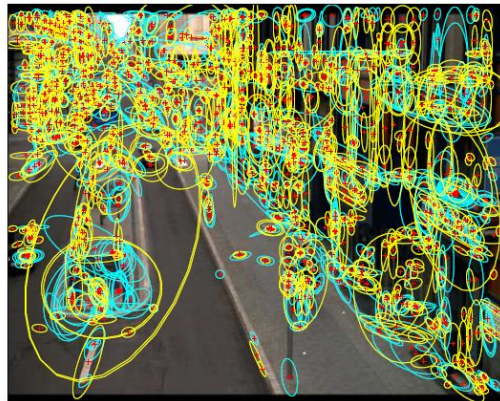
Sparse, at interest points



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]

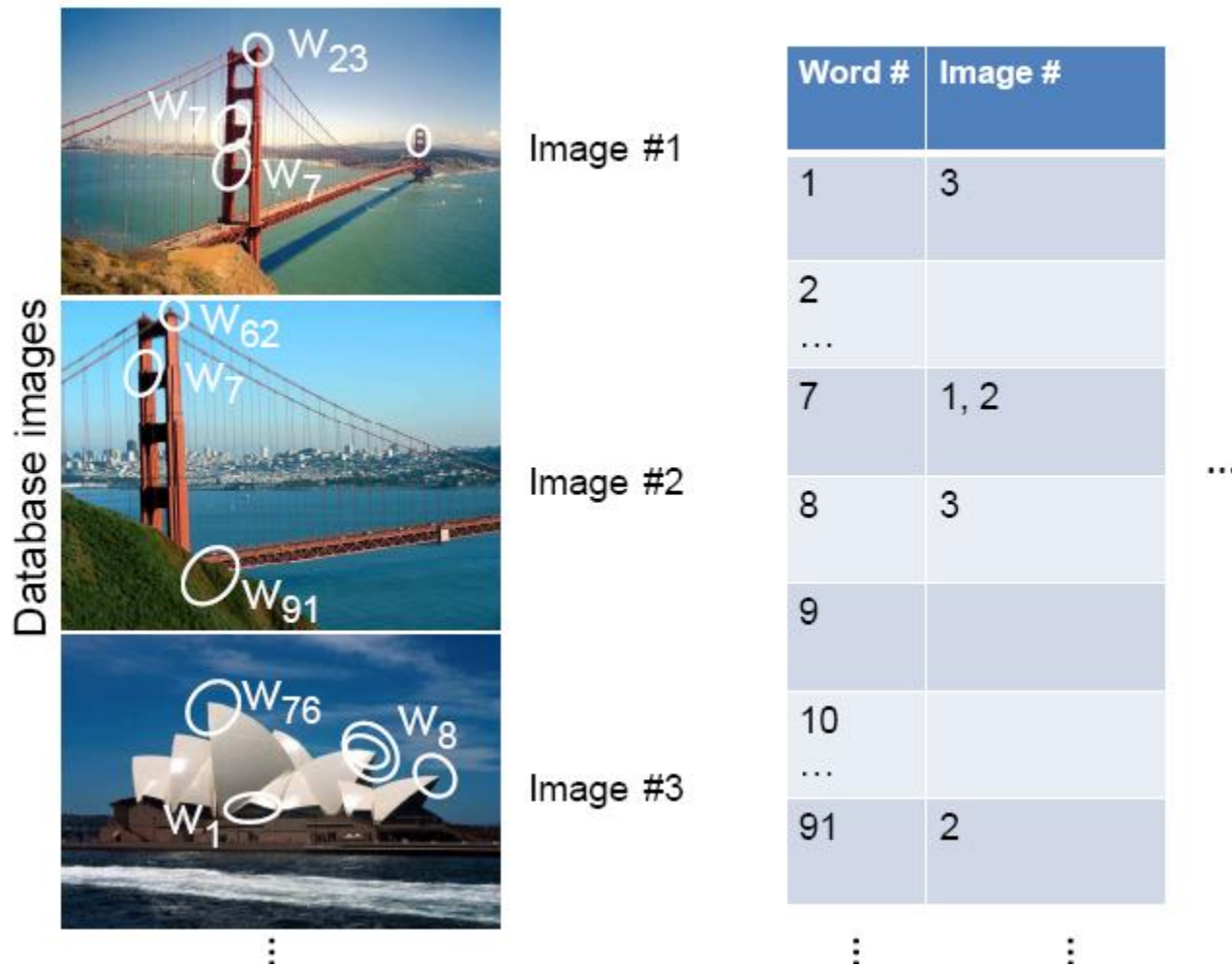
Fast lookup: inverted file index

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
 Belz Outlet Mall; 89
 Bernard Castro; 136
 Big "I"; 165
 Big Cypress; 155,158
 Big Foot Monster; 105
 Billie Swamp Safari; 160
 Blackwater River SP; 117
 Blue Angels
 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
 Chieflay; 114
 Name; 115
 Choctawhatchee, 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. Augustine; 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
 Crosstown Expy; 11,35,98,143
 Cuban Bread; 184
 Dade Battlefield; 140
 Dade, Maj. Francis; 139-140,161
 Dania Beach Hurricane; 184
 Daniel Boone, Florida Walk; 117
 Daytona Beach; 172-173
 De Land; 87
 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; 168
 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
 HEFT; 76,161,190
 History; 189
 Names; 189
 Service Plazas; 190
 Spur SR91; 76
 Ticket System; 190
 Toll Plazas; 190
 Ford, Henry; 152

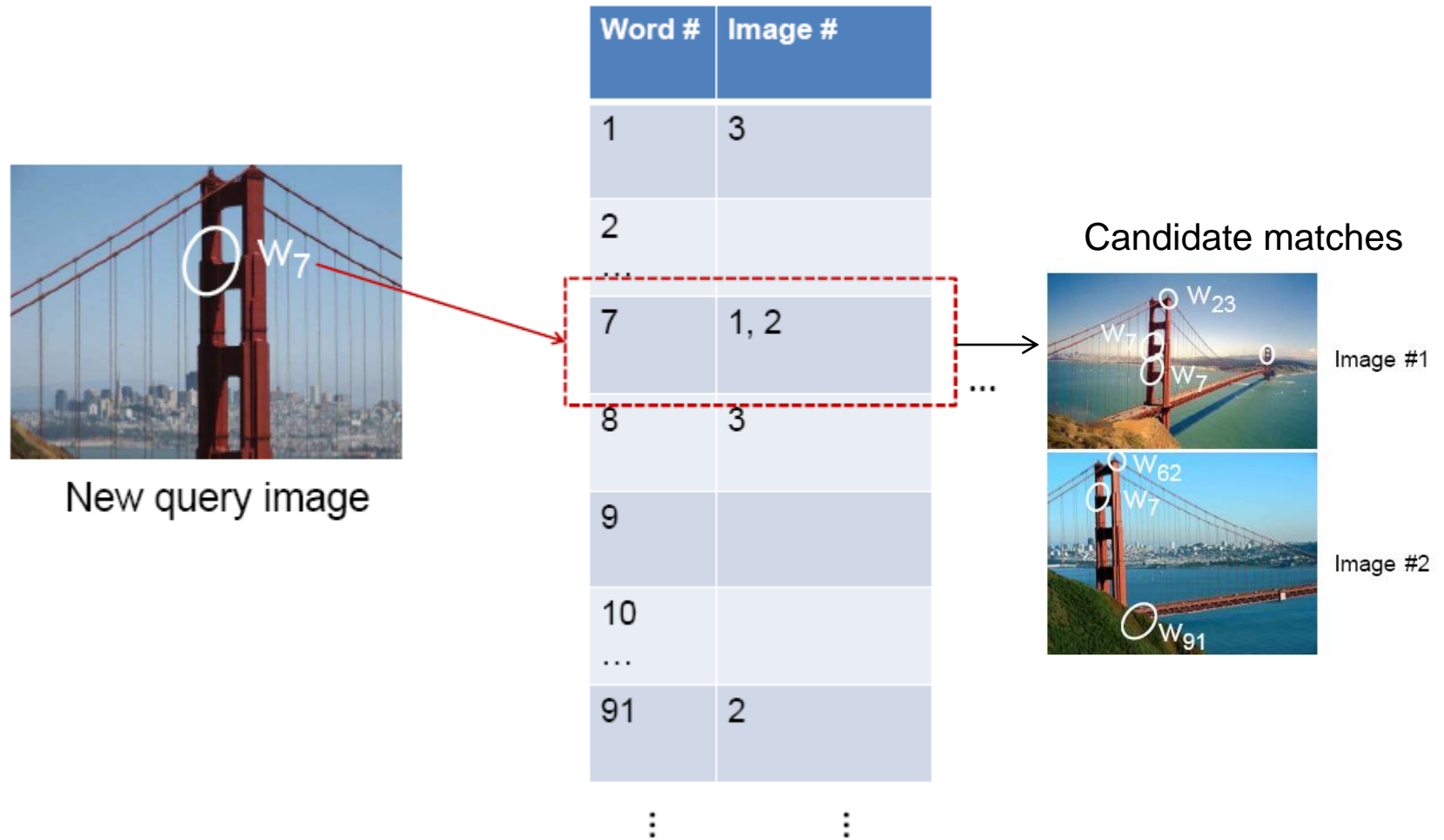
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.

Inverted file index



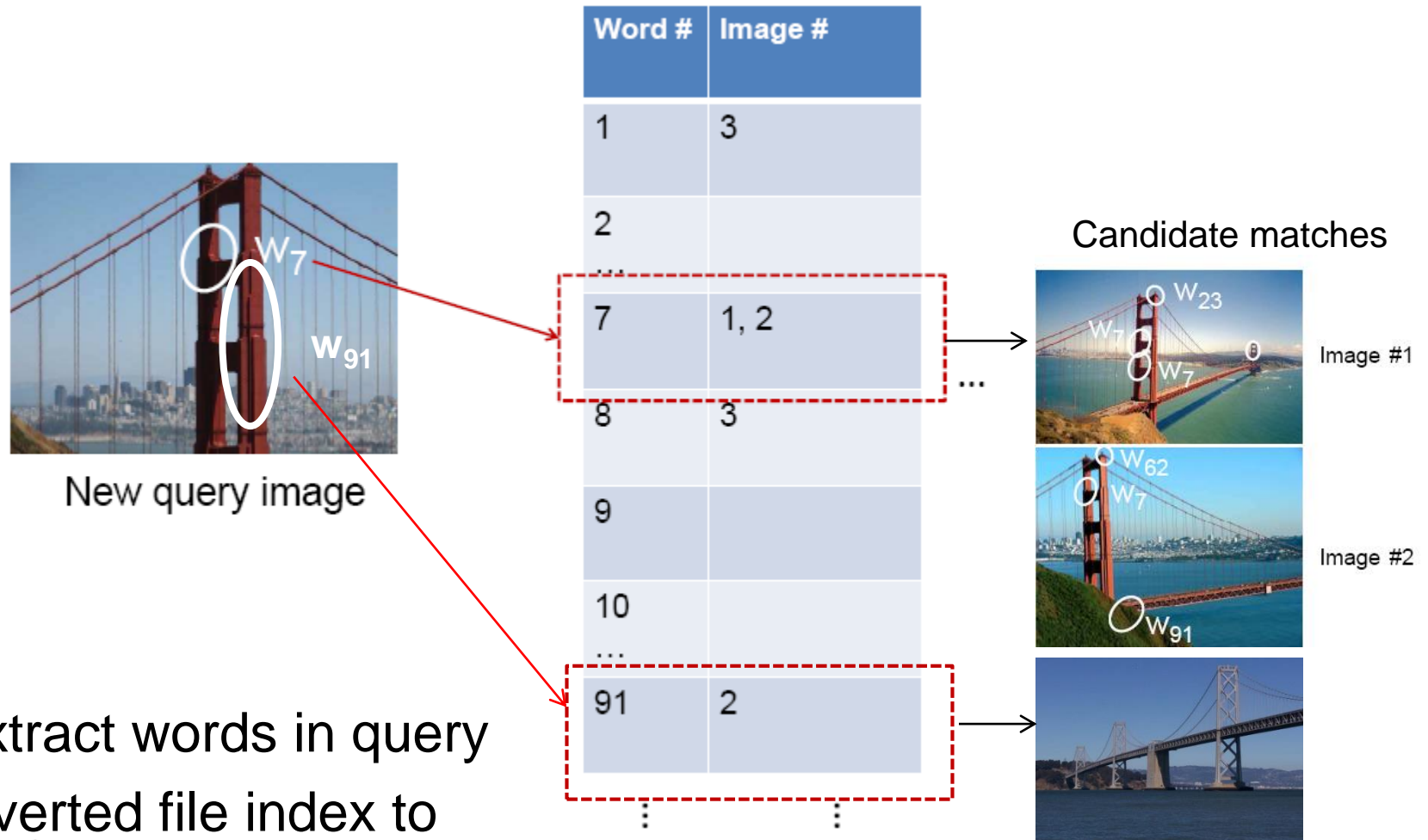
- Database images are loaded into the index mapping words to image numbers

Inverted file index



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

Inverted file index



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

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.

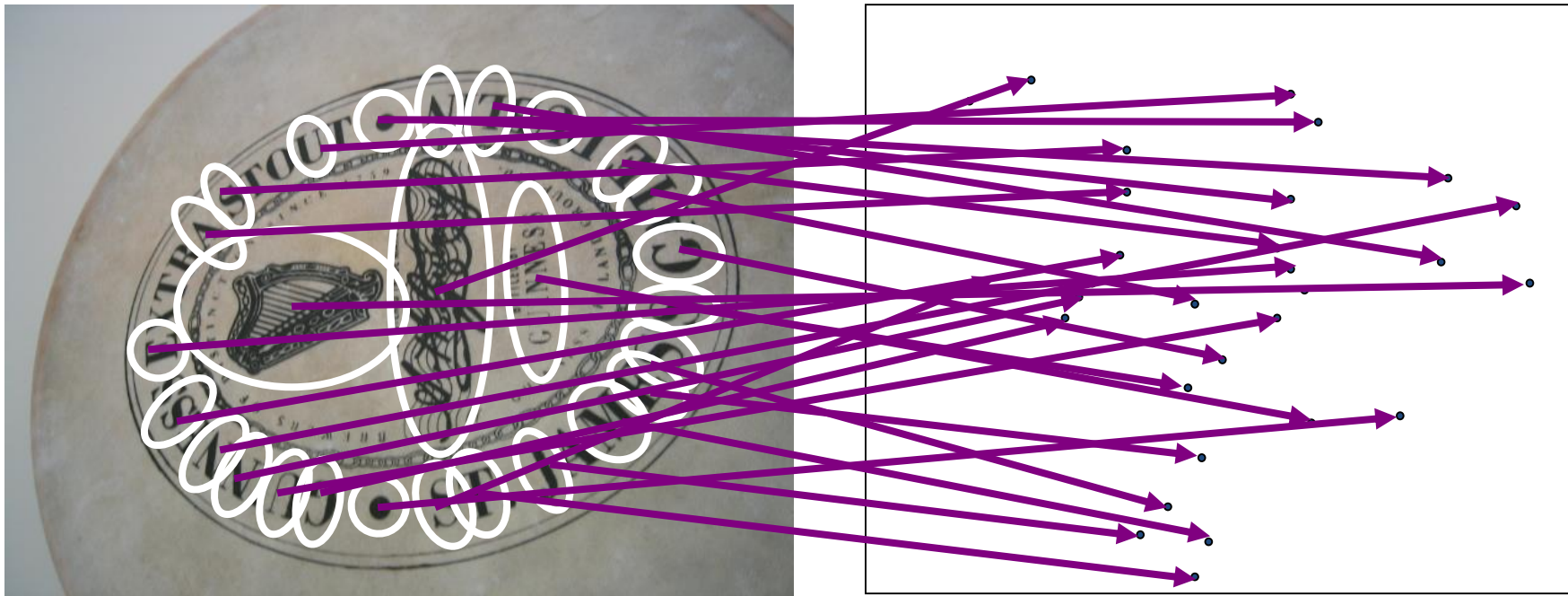
Instance recognition: remaining issues

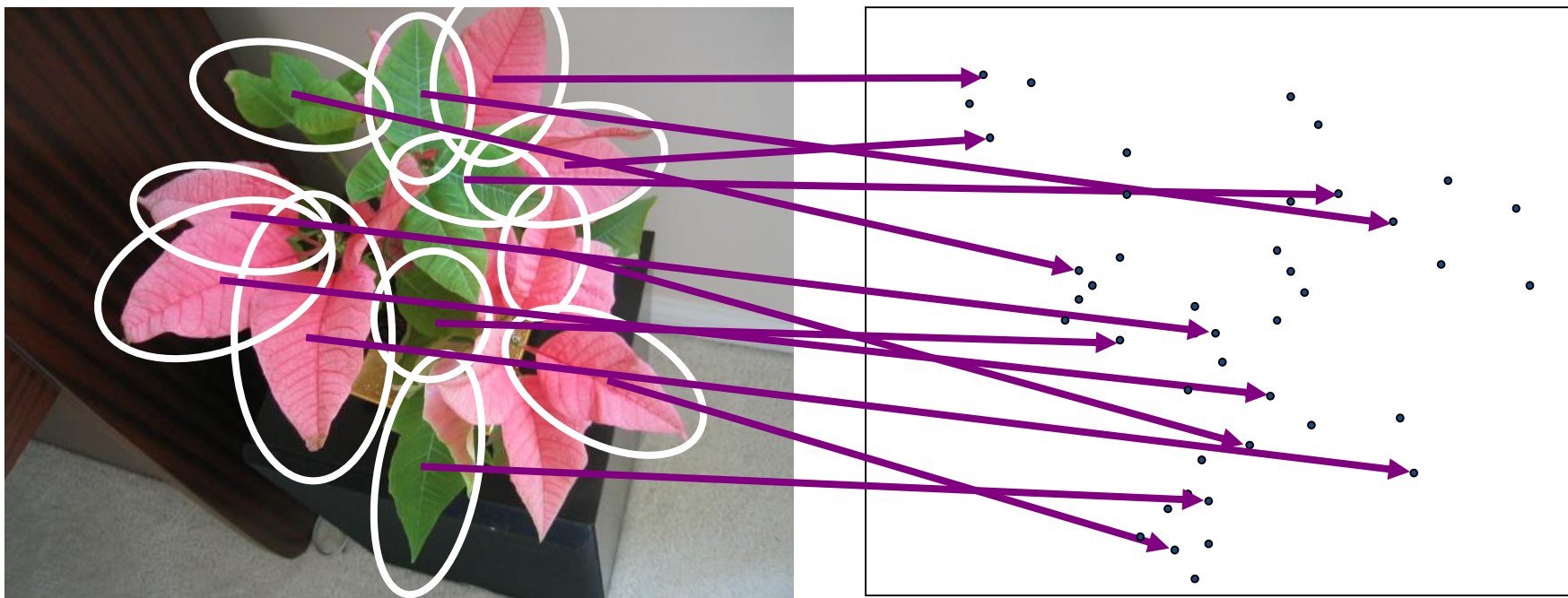
How to summarize the content of an entire image?
And gauge overall similarity?

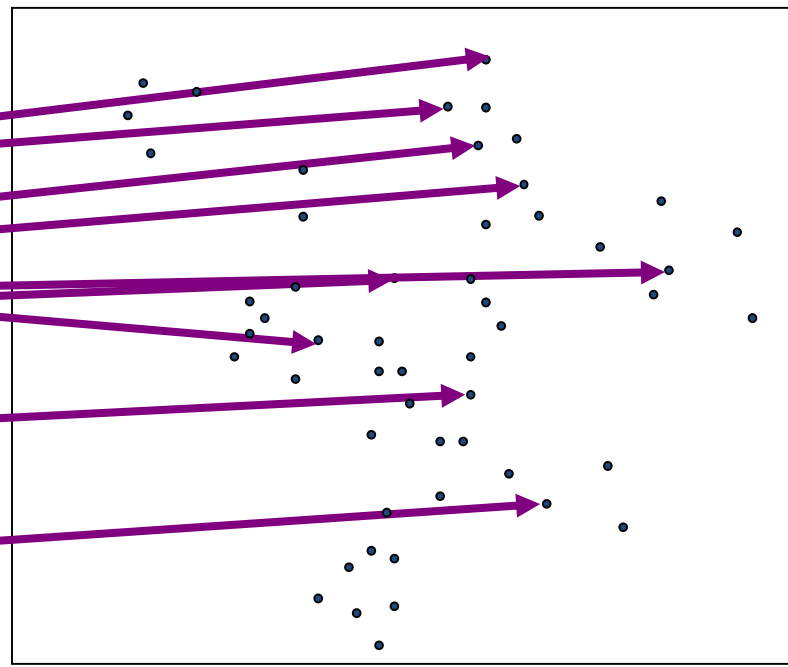
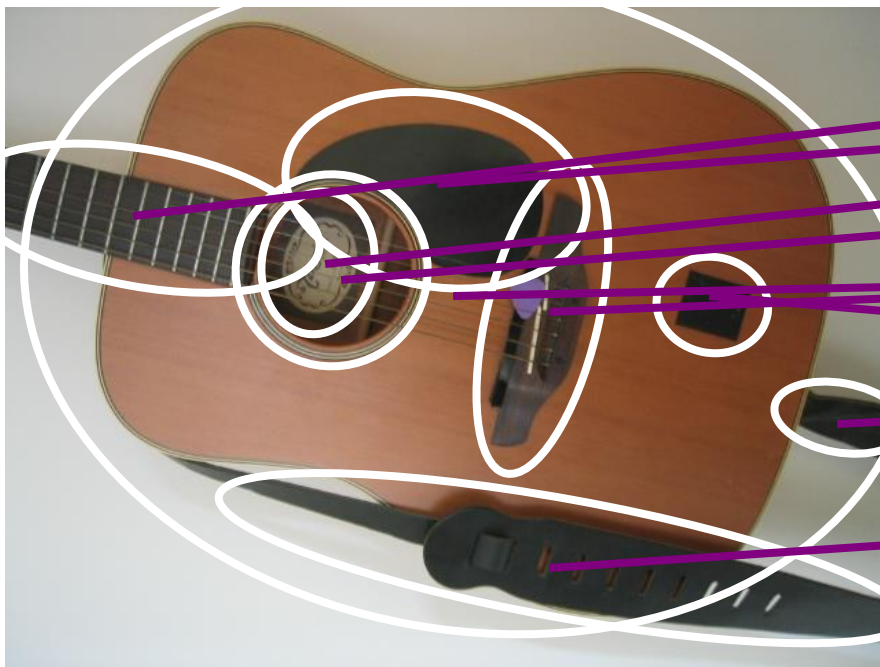
How large should the vocabulary be? How to
perform quantization efficiently?

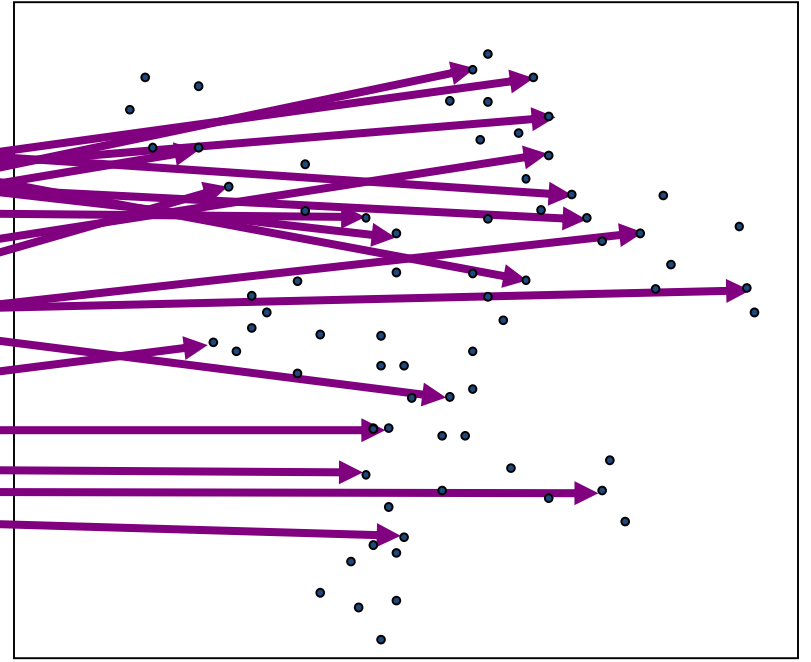
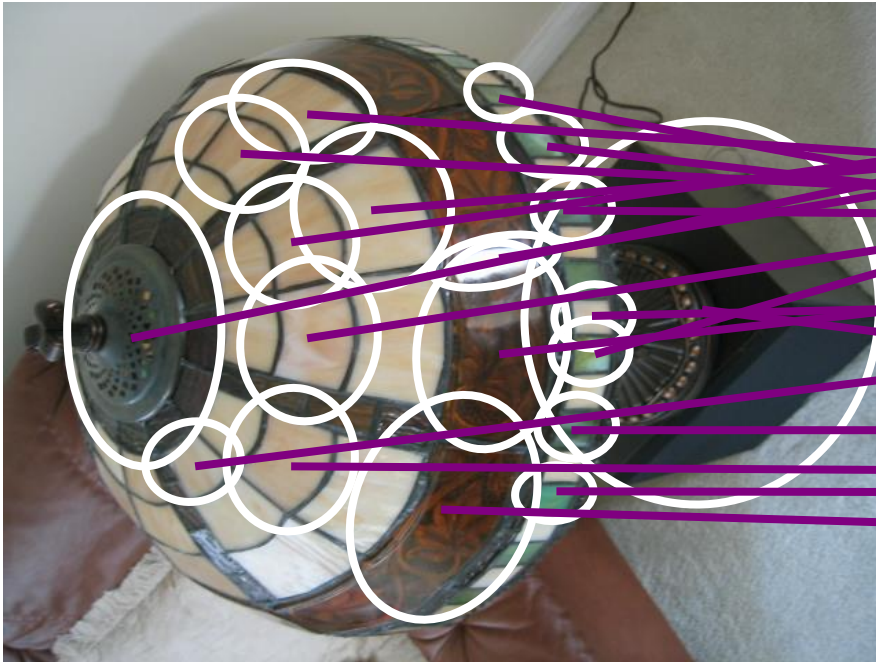
Is having the same set of visual words enough to
identify the object/scene? How to verify spatial
agreement?

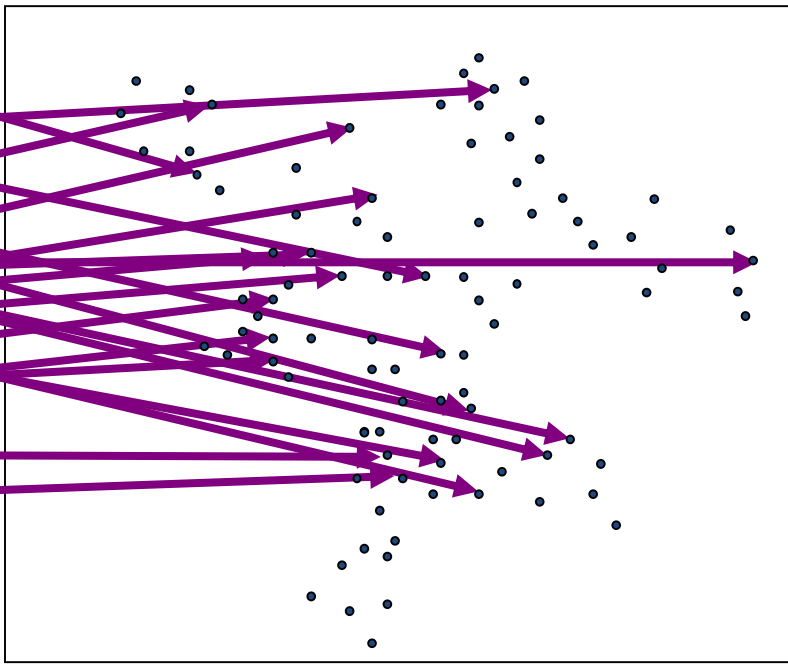
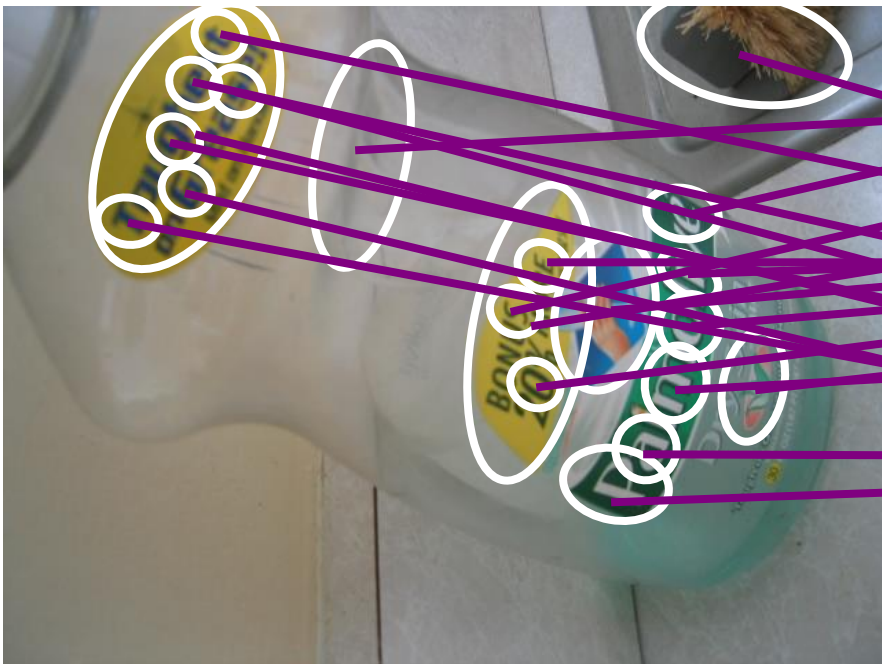
How to score the retrieval results?

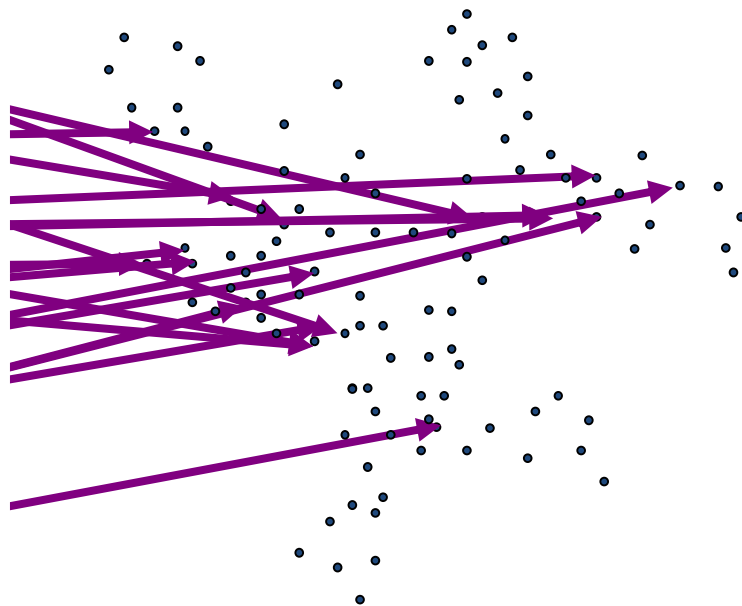


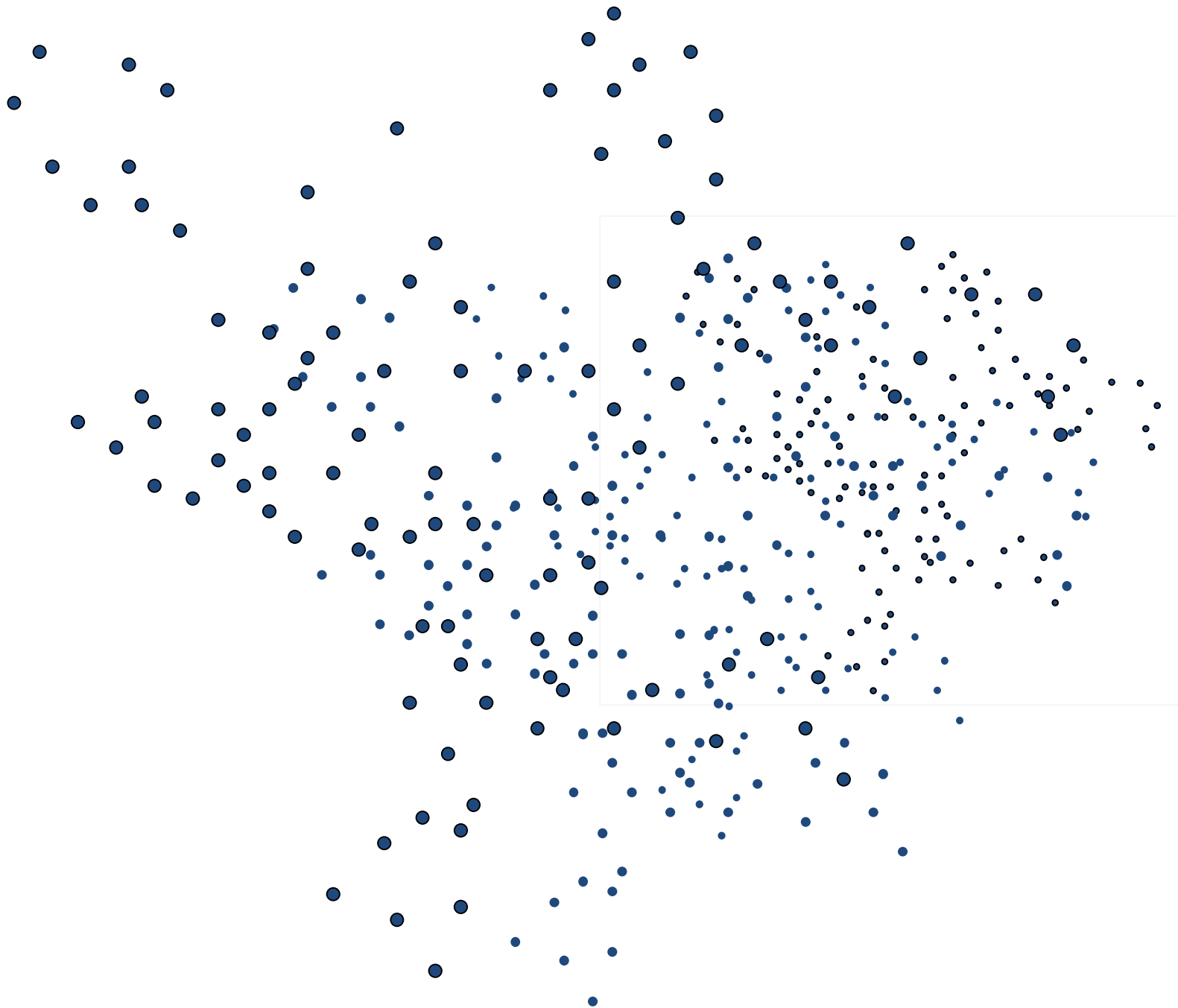


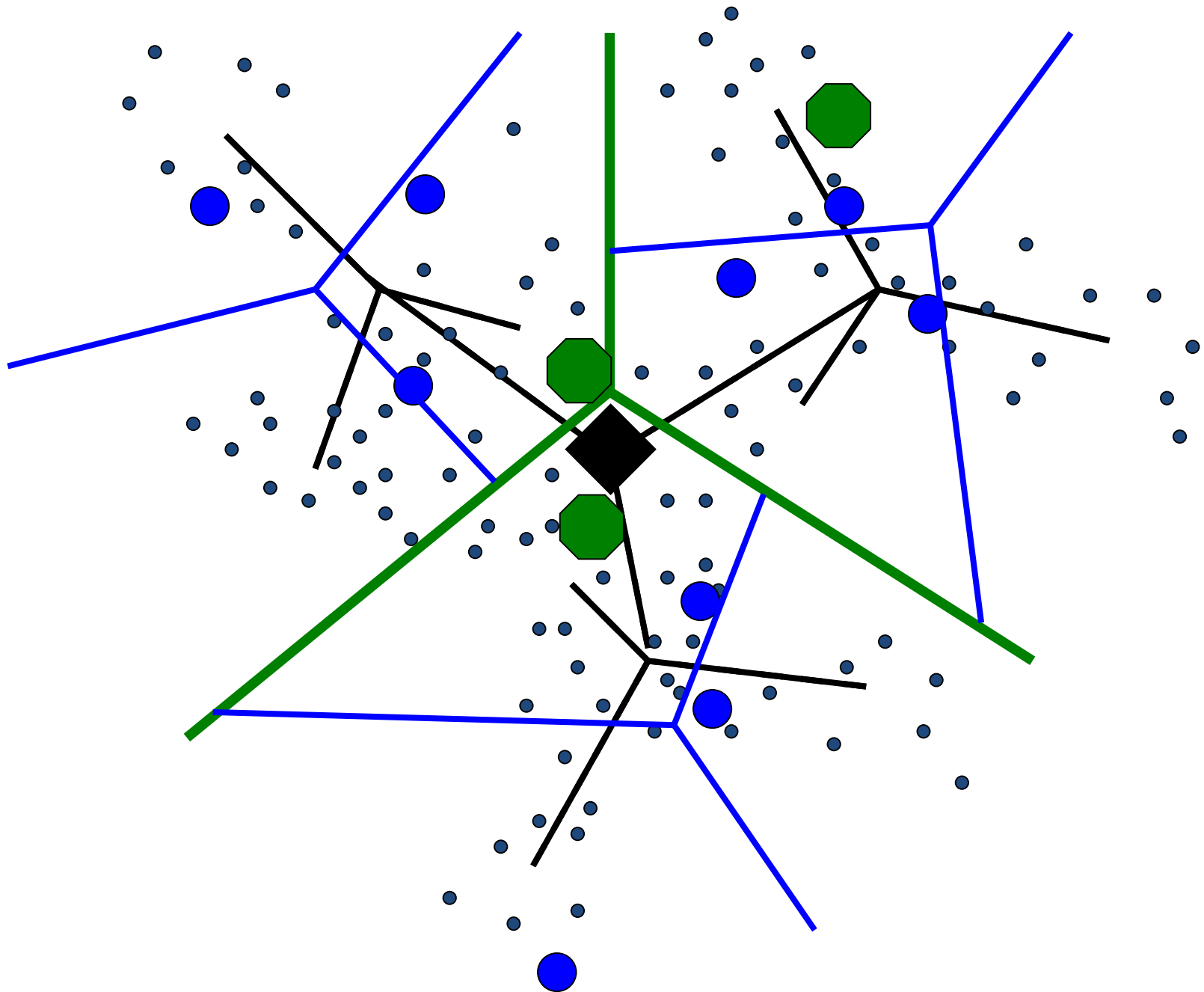


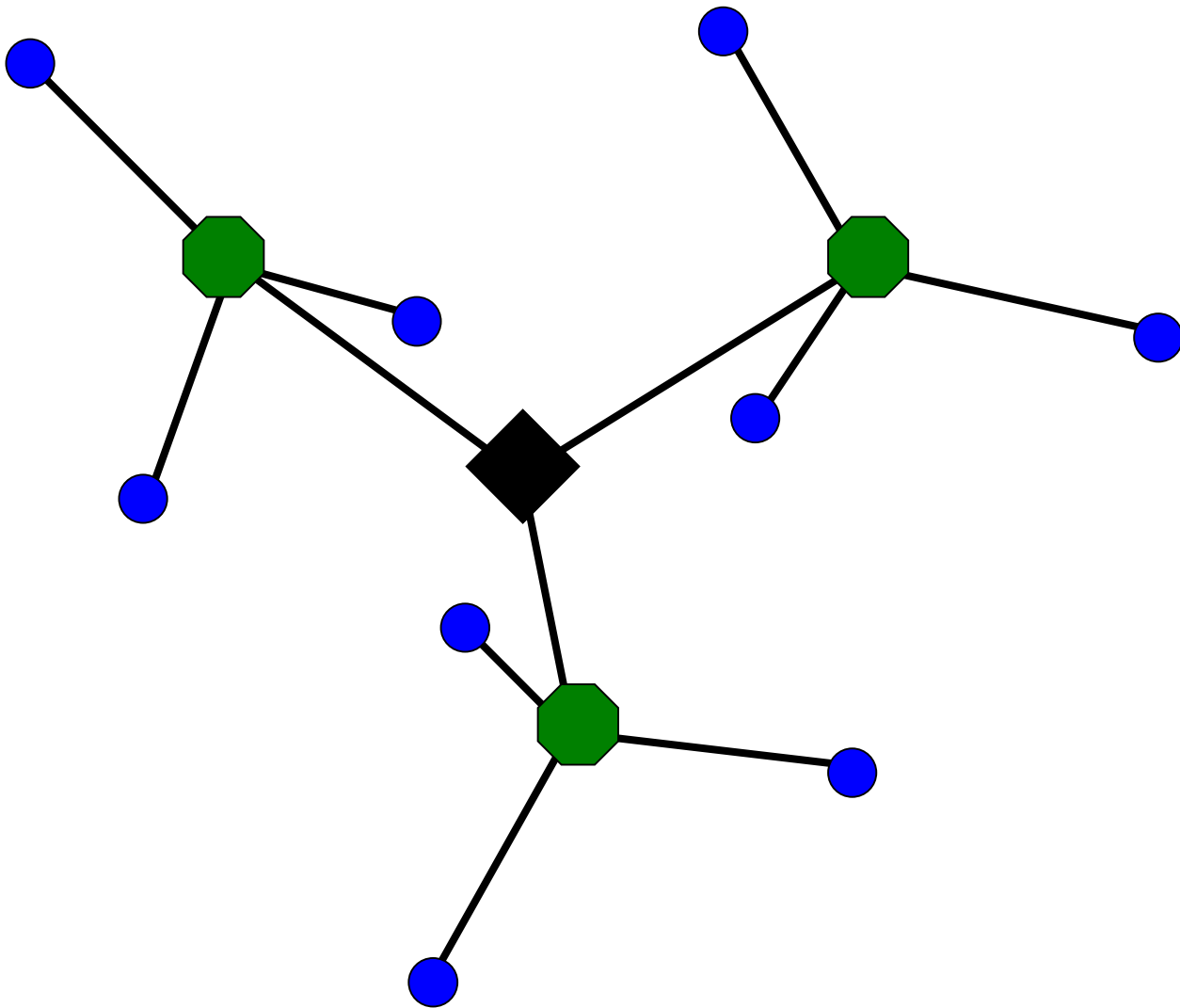


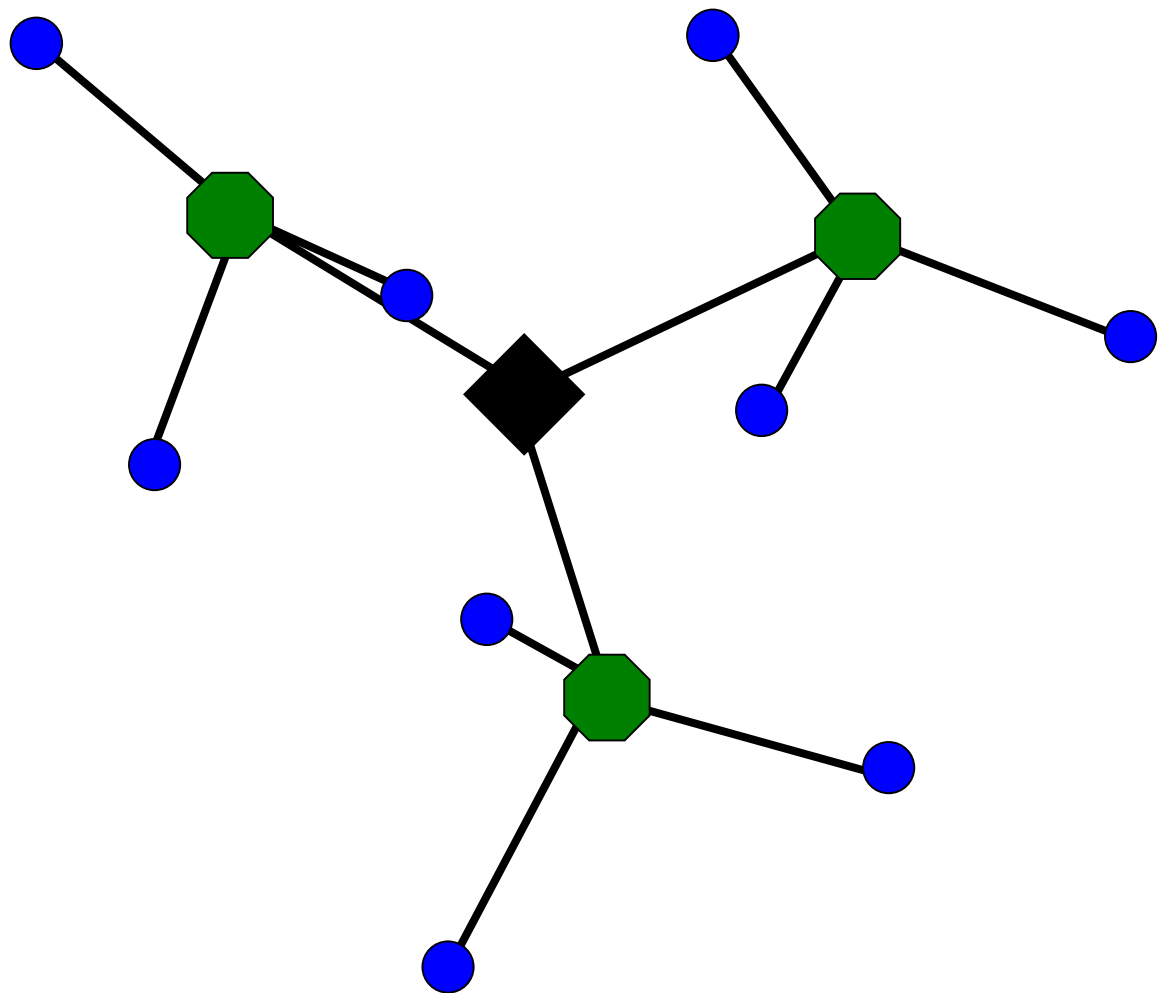


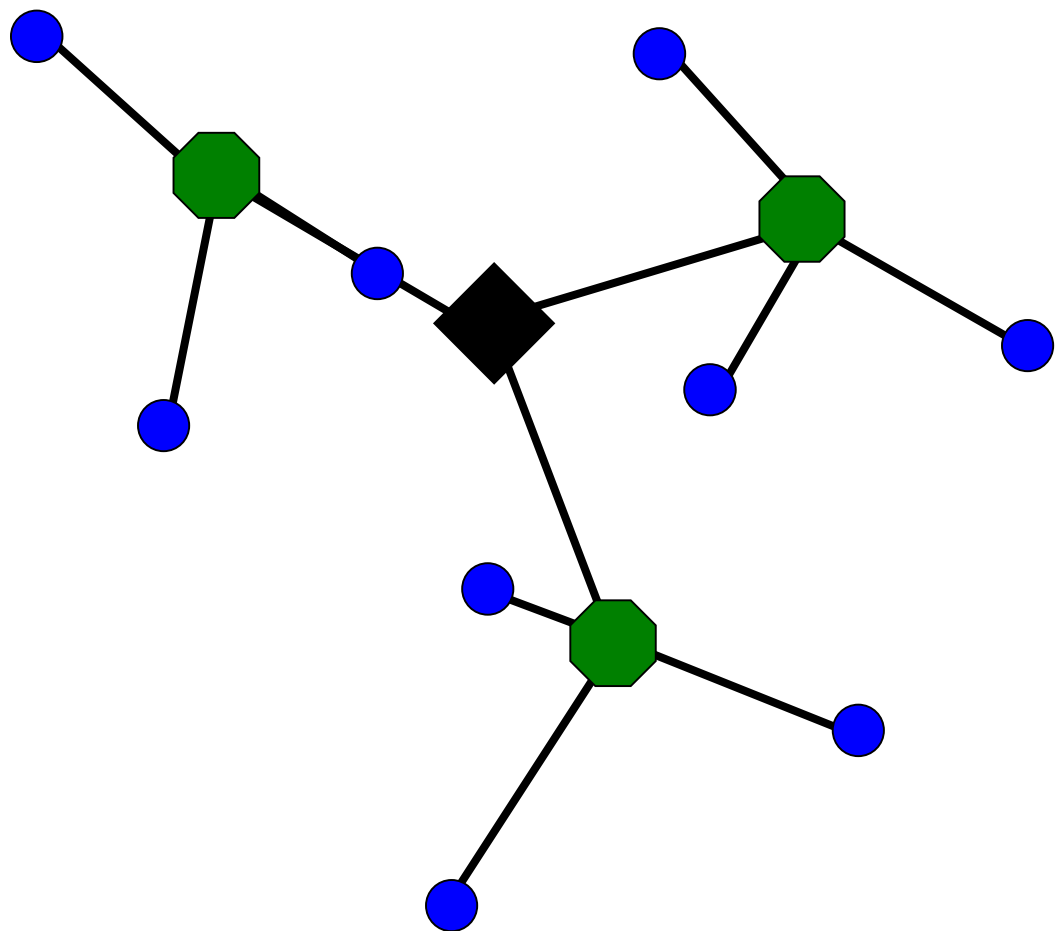


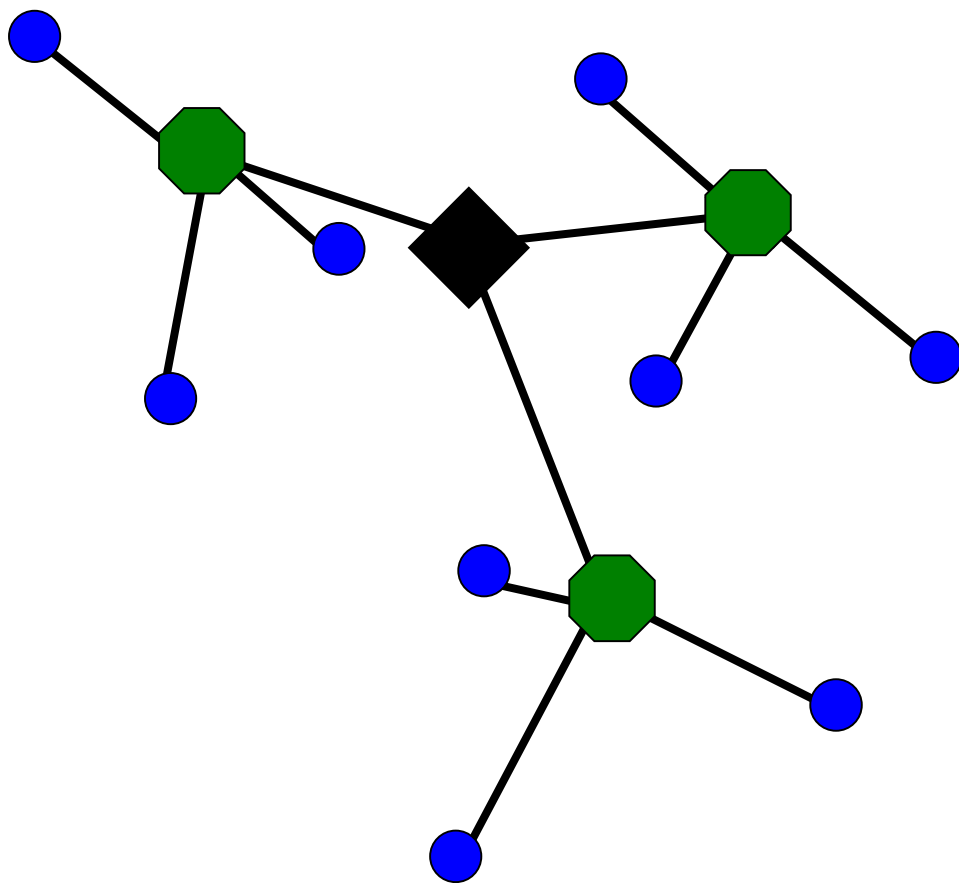


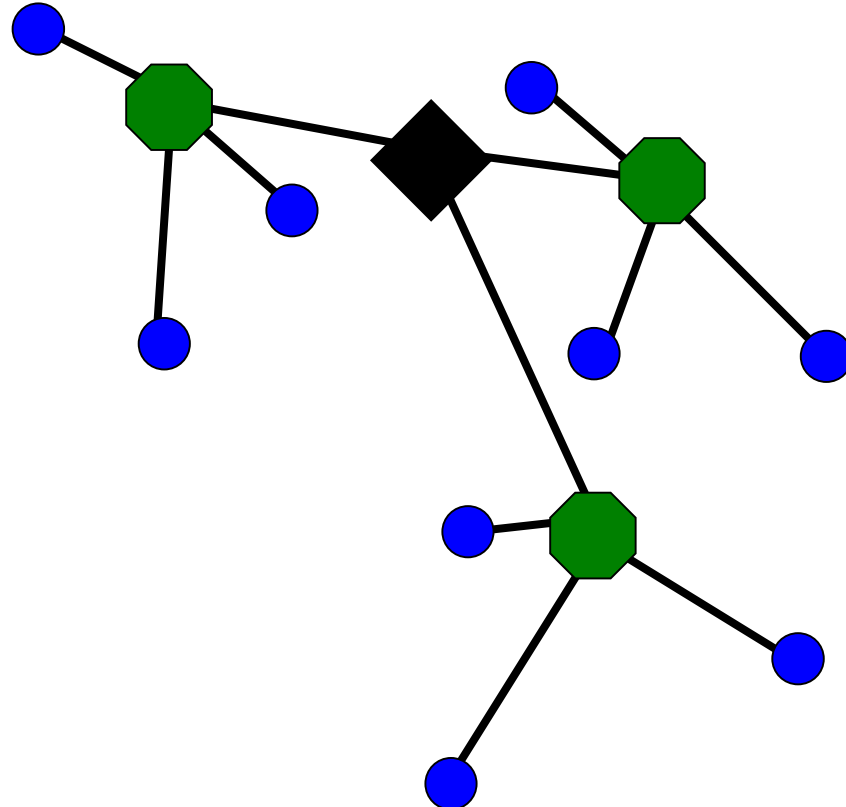


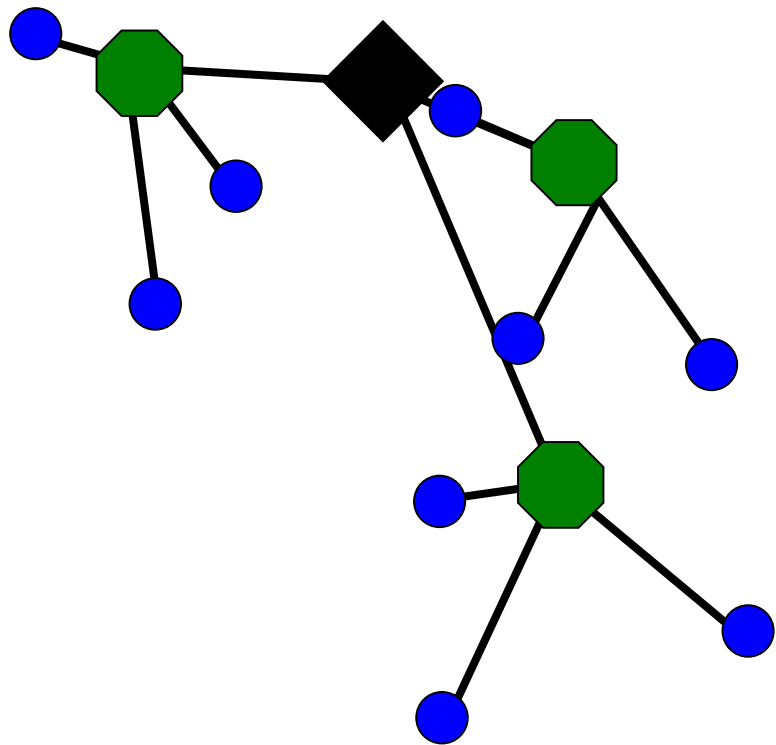


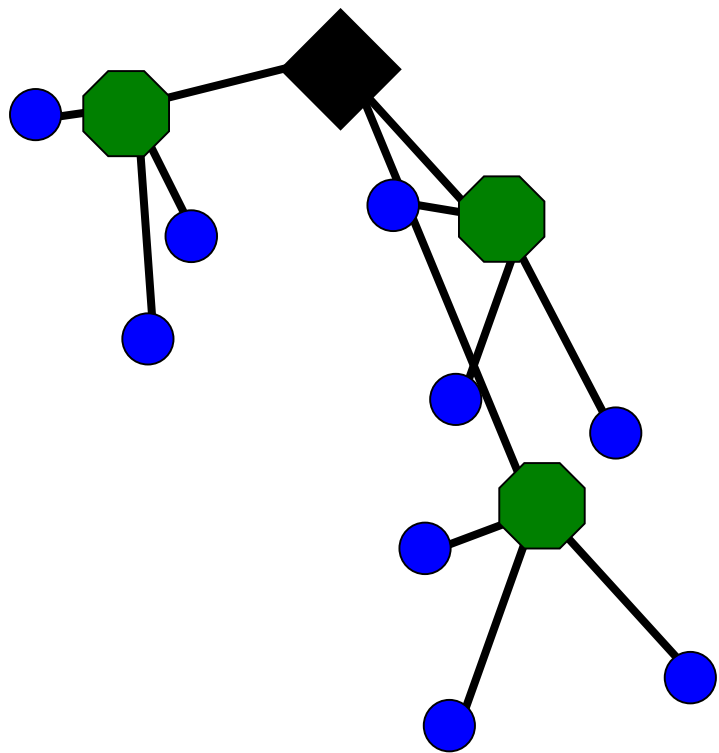


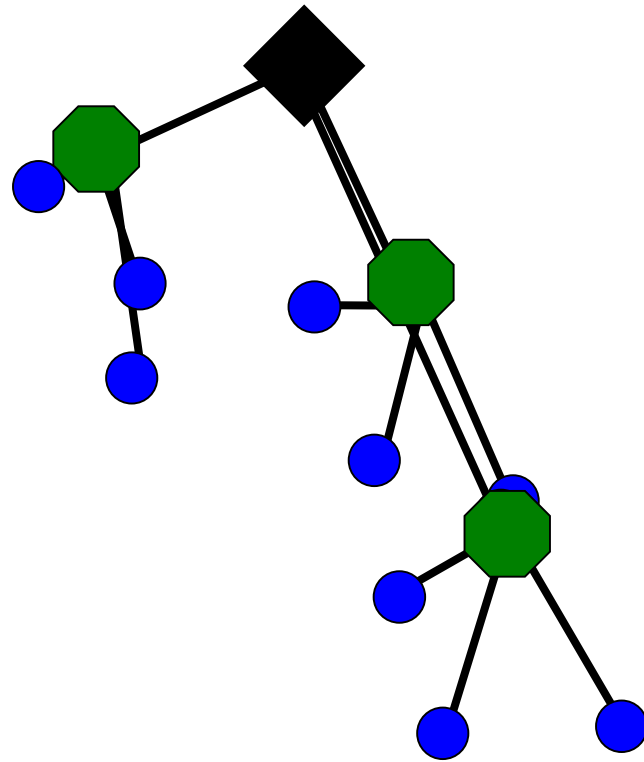


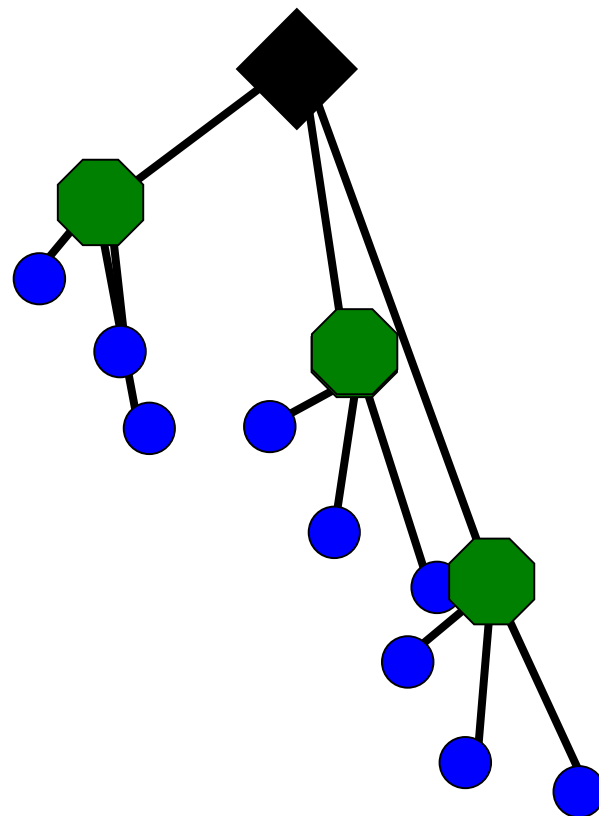


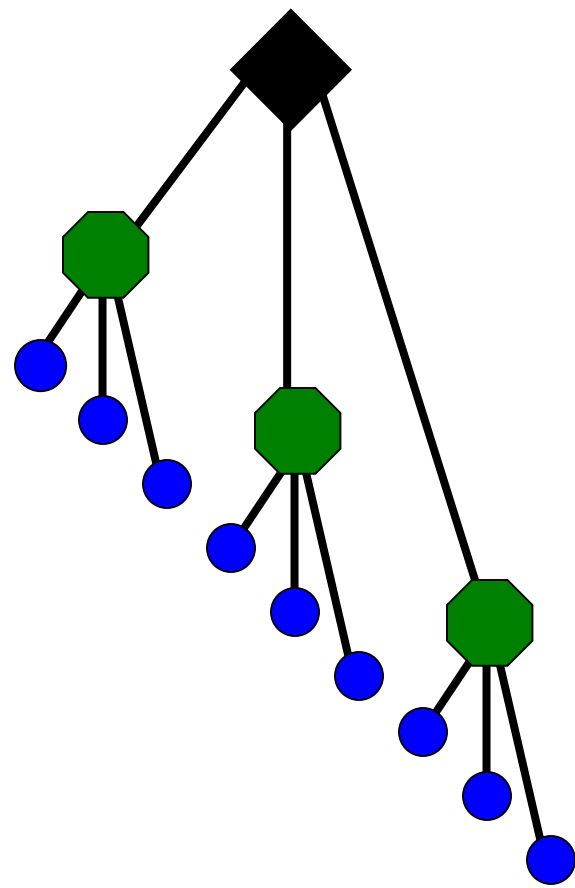


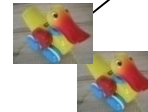
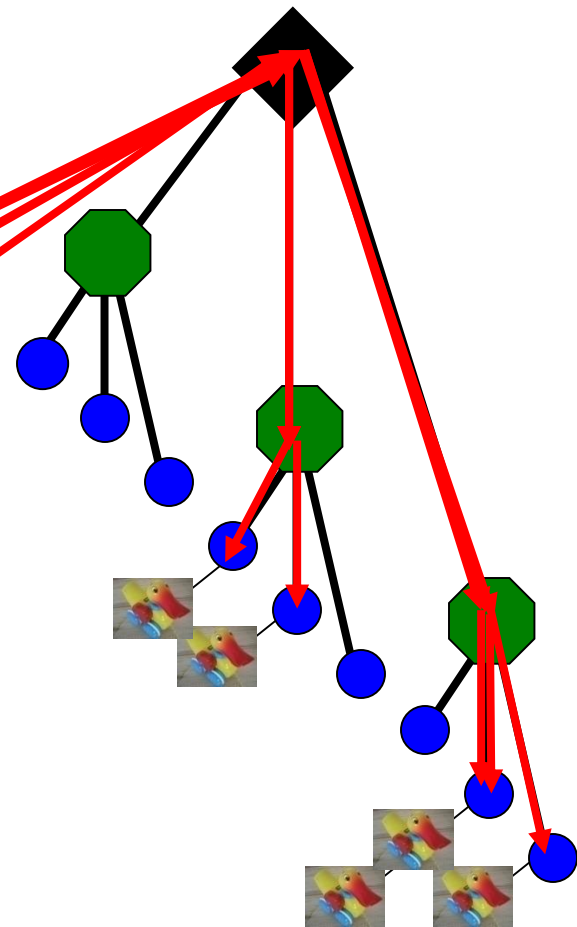
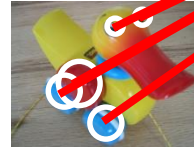


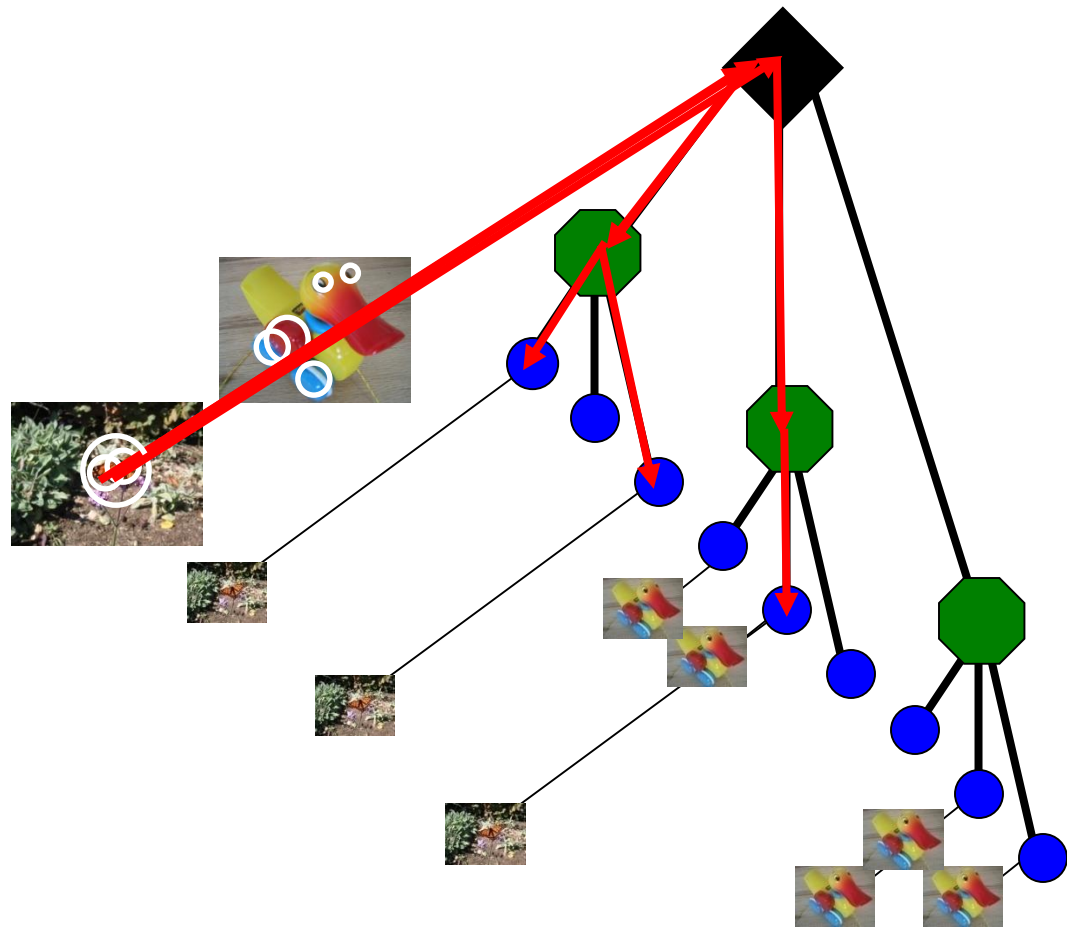


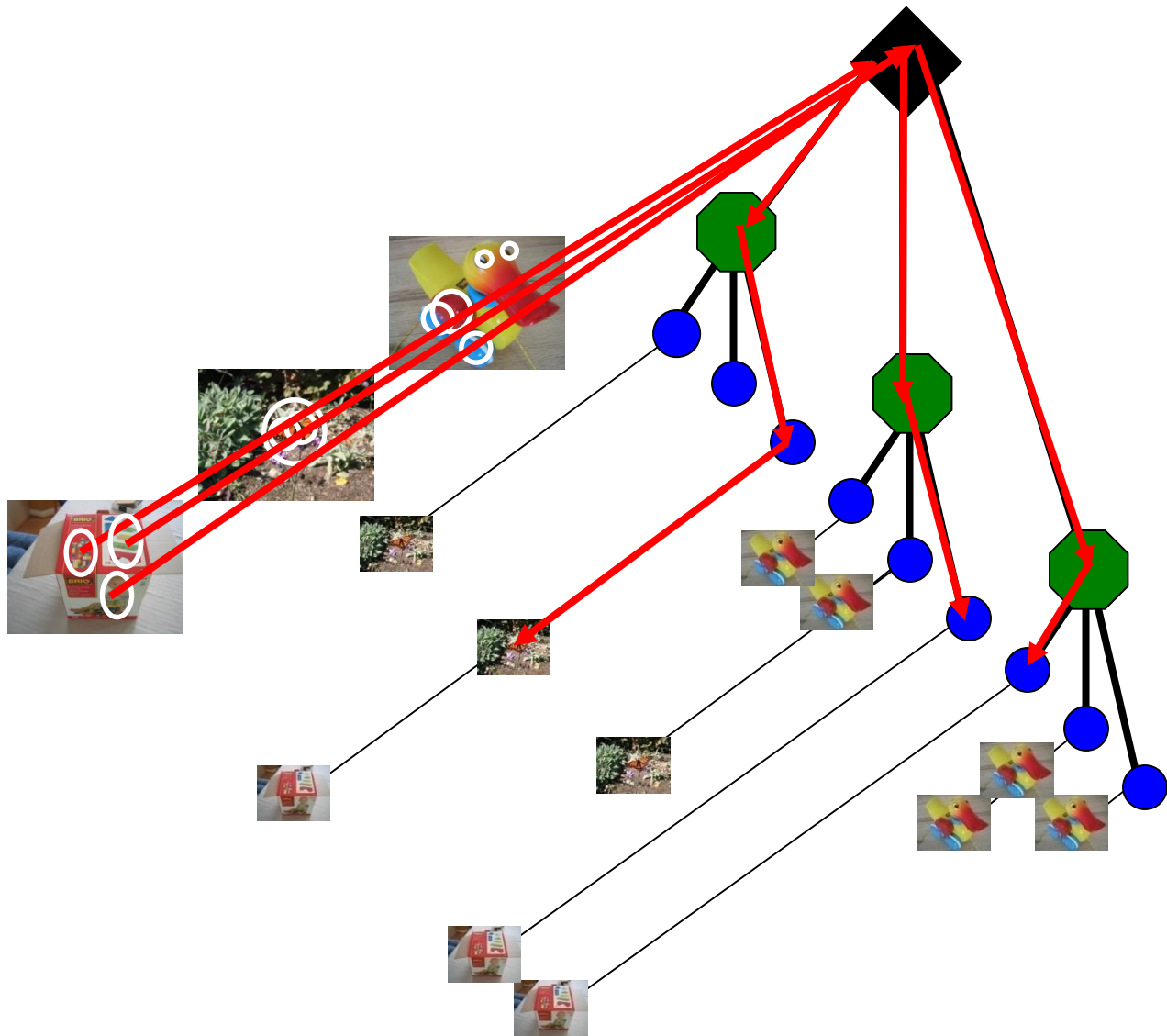


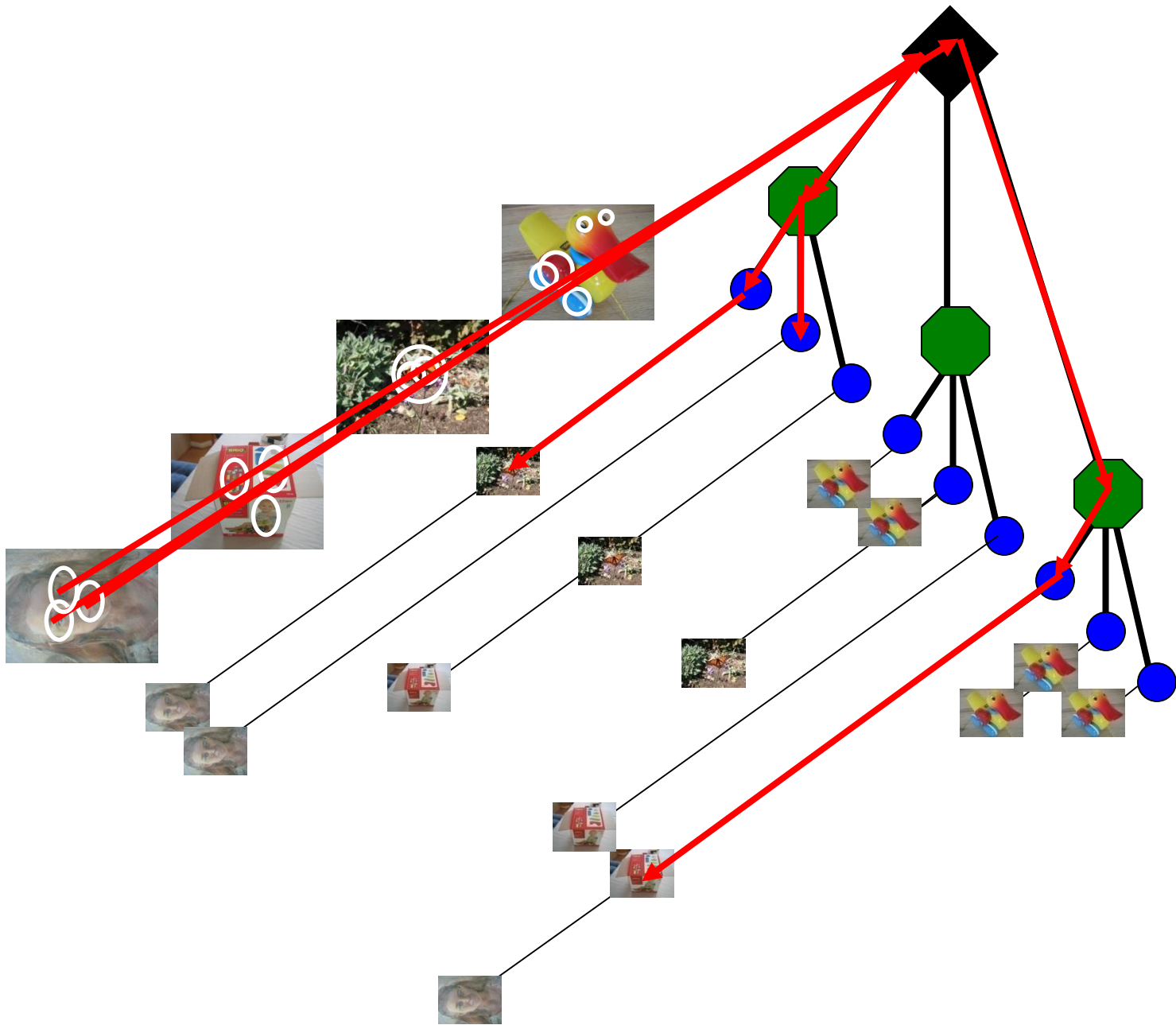


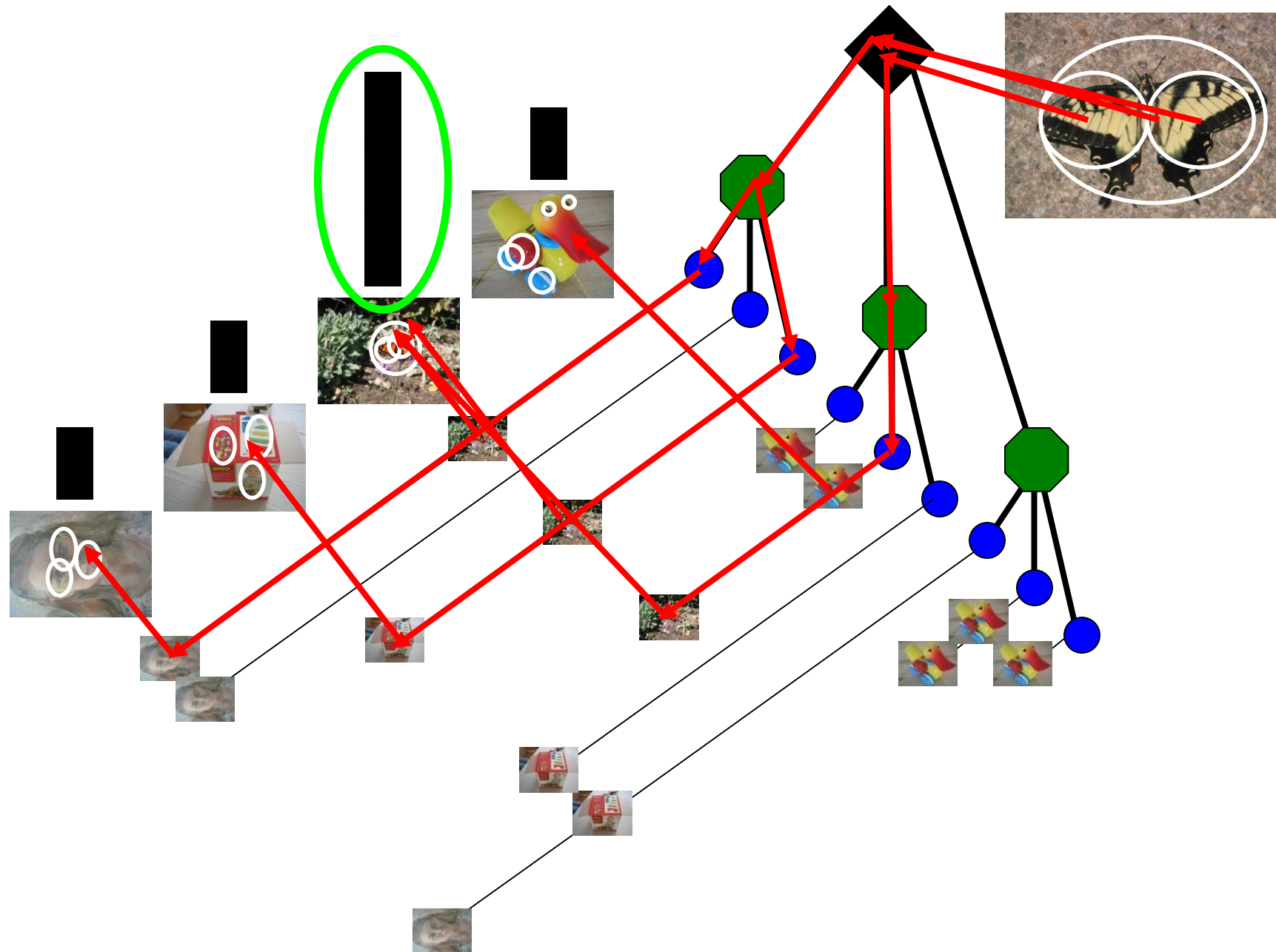




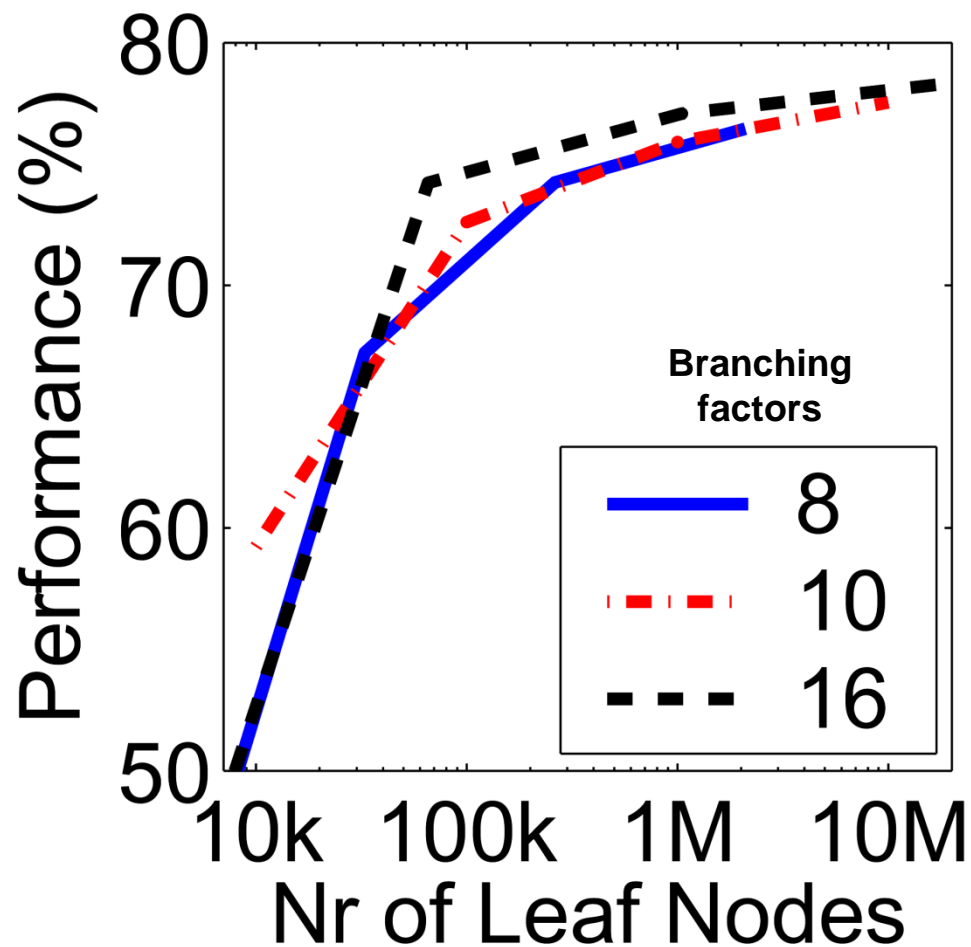






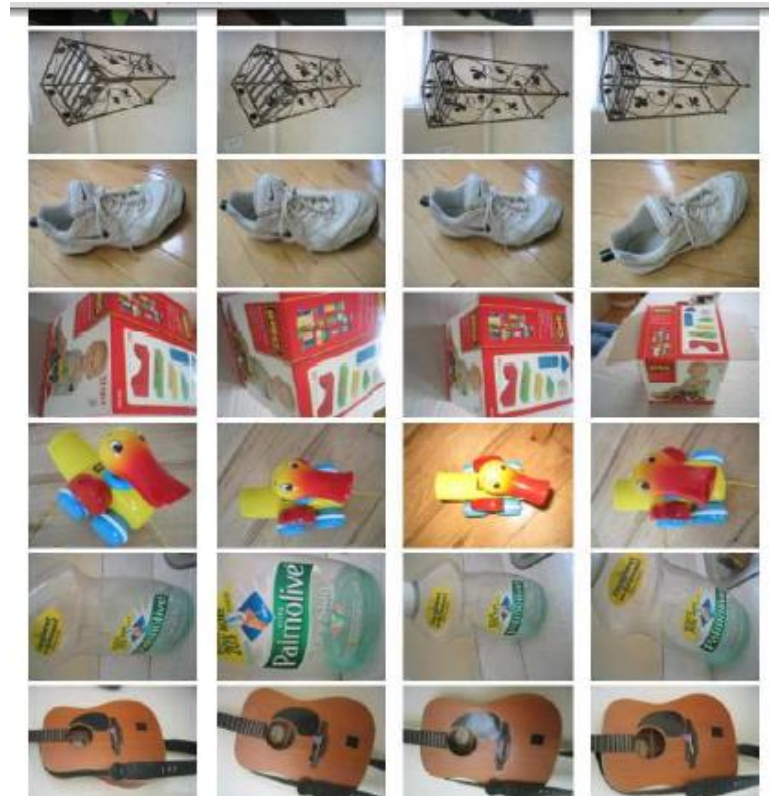


Vocabulary size



Influence on performance, sparsity

Recognition with 6347 images



Nister & Stewenius, CVPR 2006

Vocabulary trees: complexity

Number of words given tree parameters:

$$\text{branching_factor}^{\text{number_of_levels}}$$

Word assignment cost vs. flat vocabulary:

$O(k)$ for flat

$$O(\log_{\text{branching_factor}}(k) * \text{branching_factor})$$

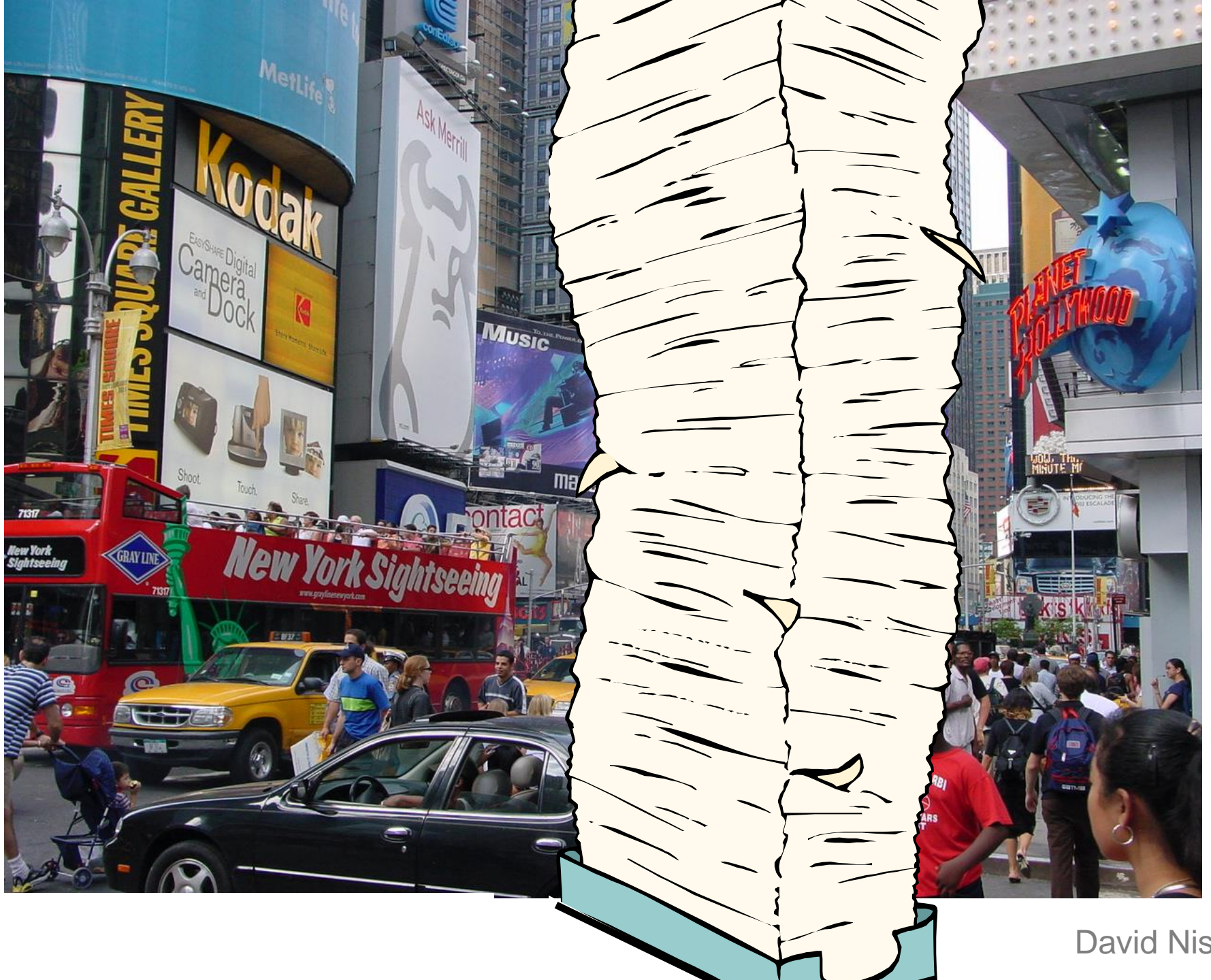
Is this like a kd-tree?

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

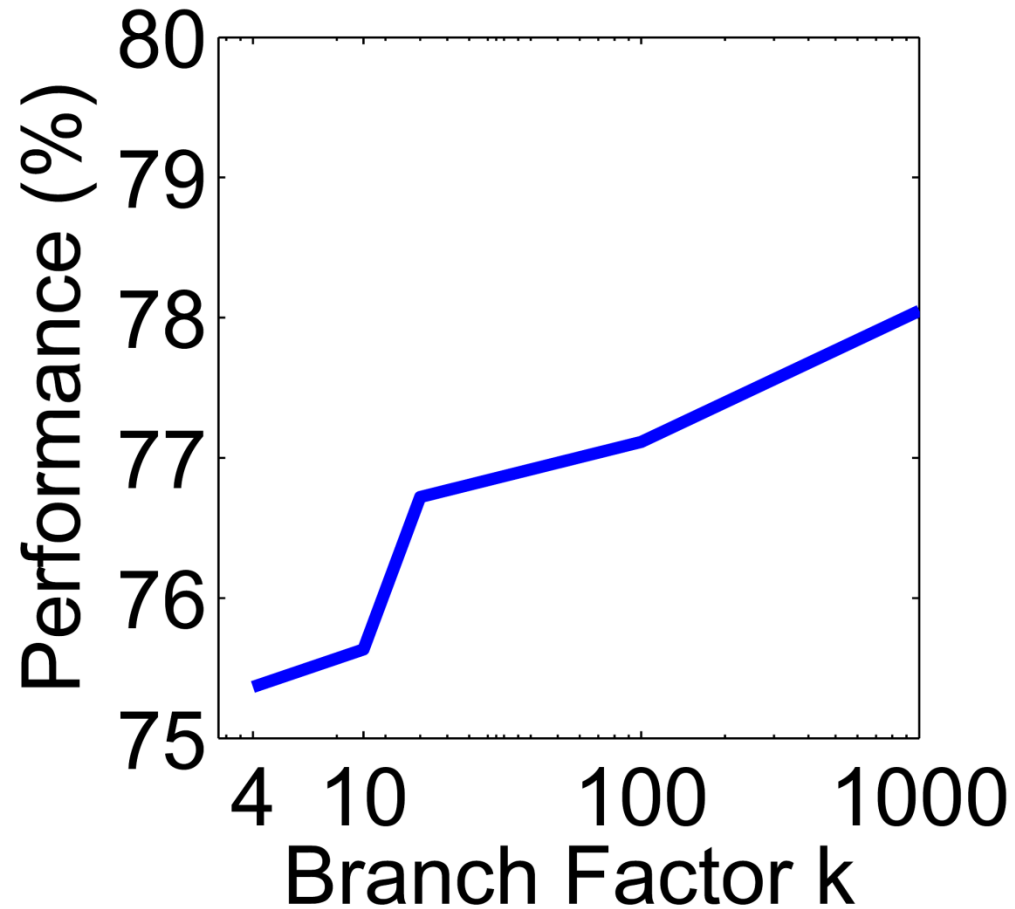








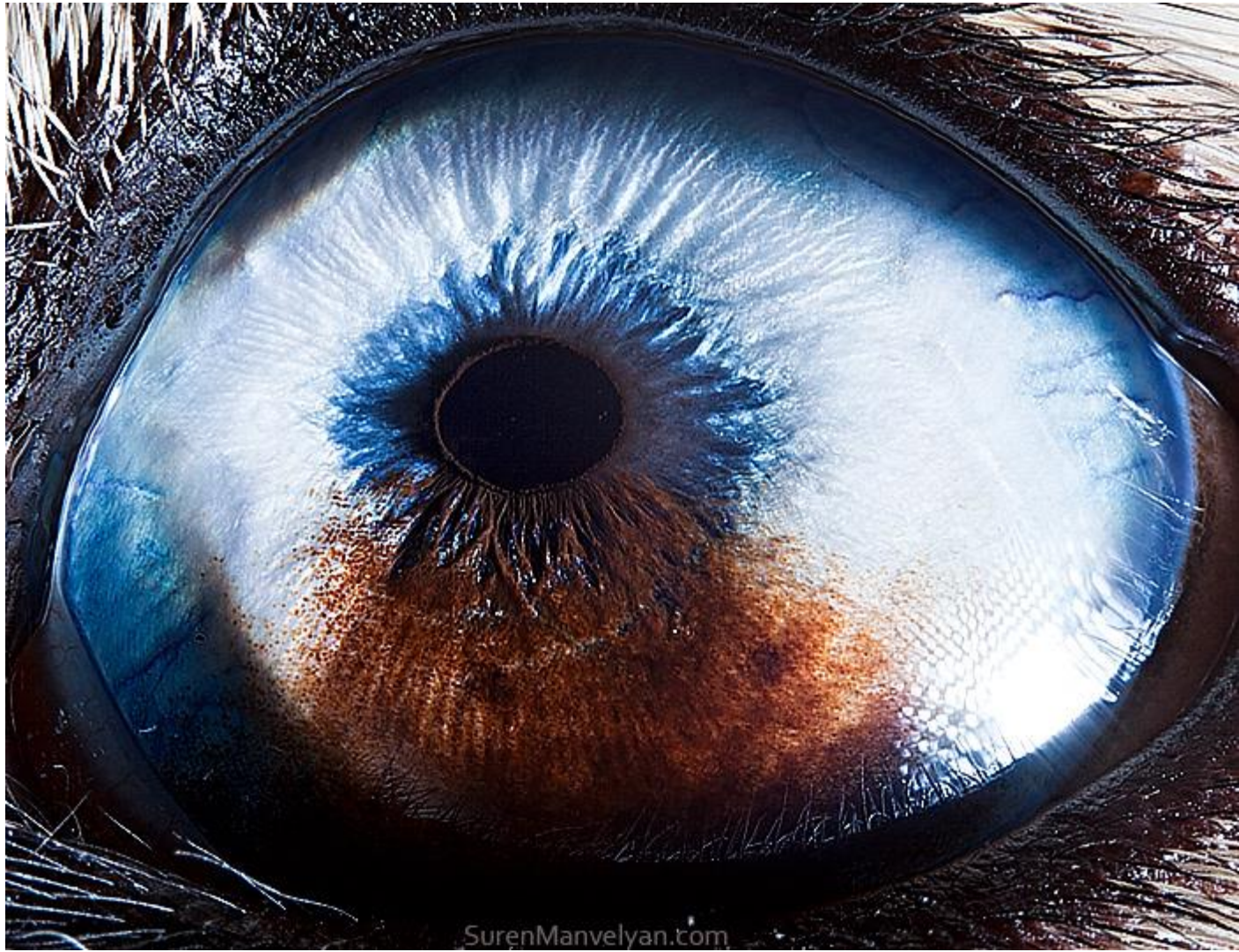
Higher branch factor works better
(but slower)



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







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?

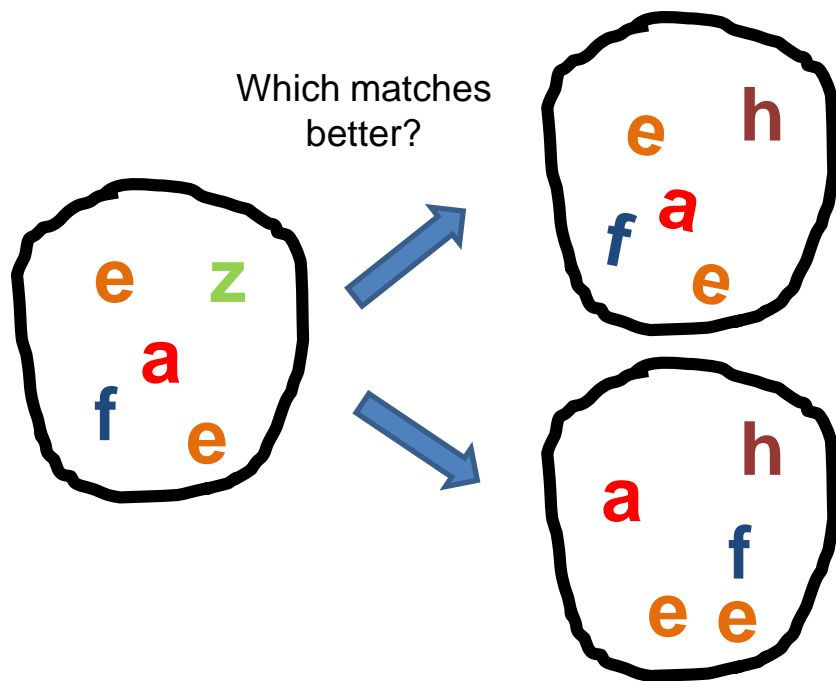
How large should the vocabulary be? How to
perform quantization efficiently?

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?

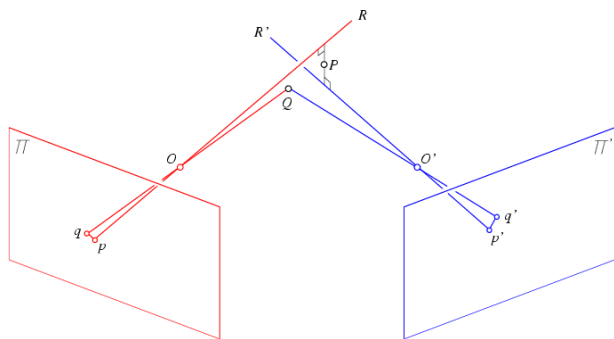
Can we be more accurate?

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

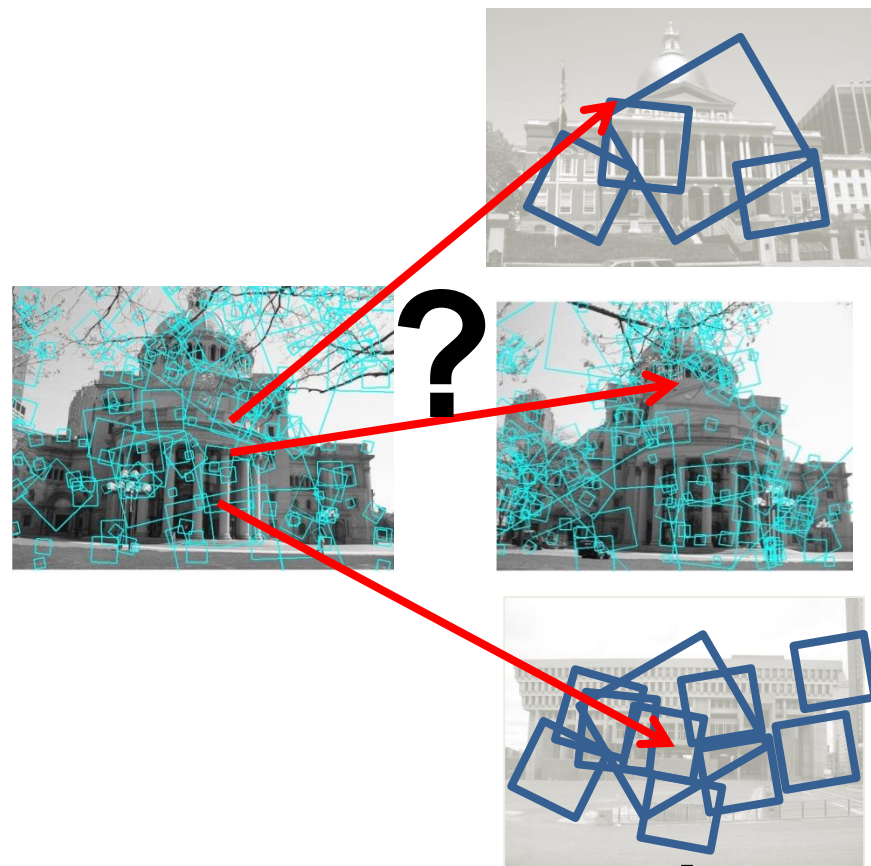


Real objects have consistent geometry

Multi-view matching



vs



Matching two given
views for depth

Search for a matching
view for recognition

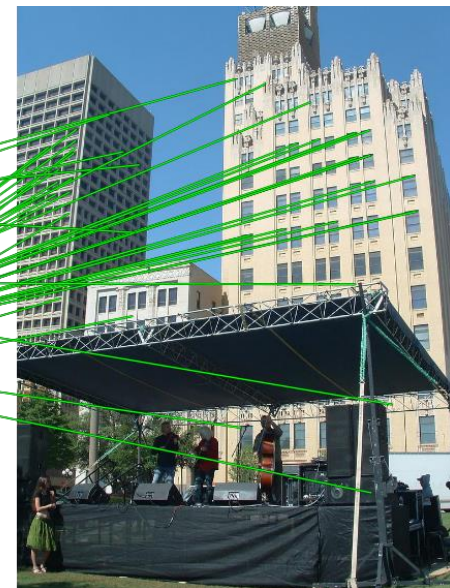
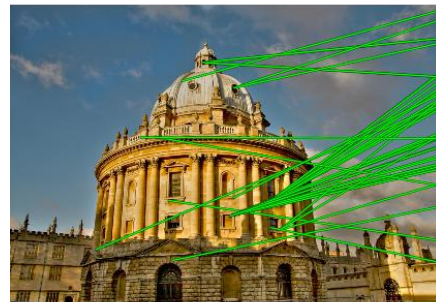
Spatial Verification

Query



DB image with high BoW
similarity

Query



DB image with high BoW
similarity

Both image pairs have many visual words in common.

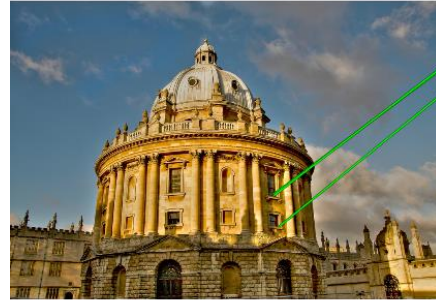
Spatial Verification

Query



DB image with high BoW
similarity

Query



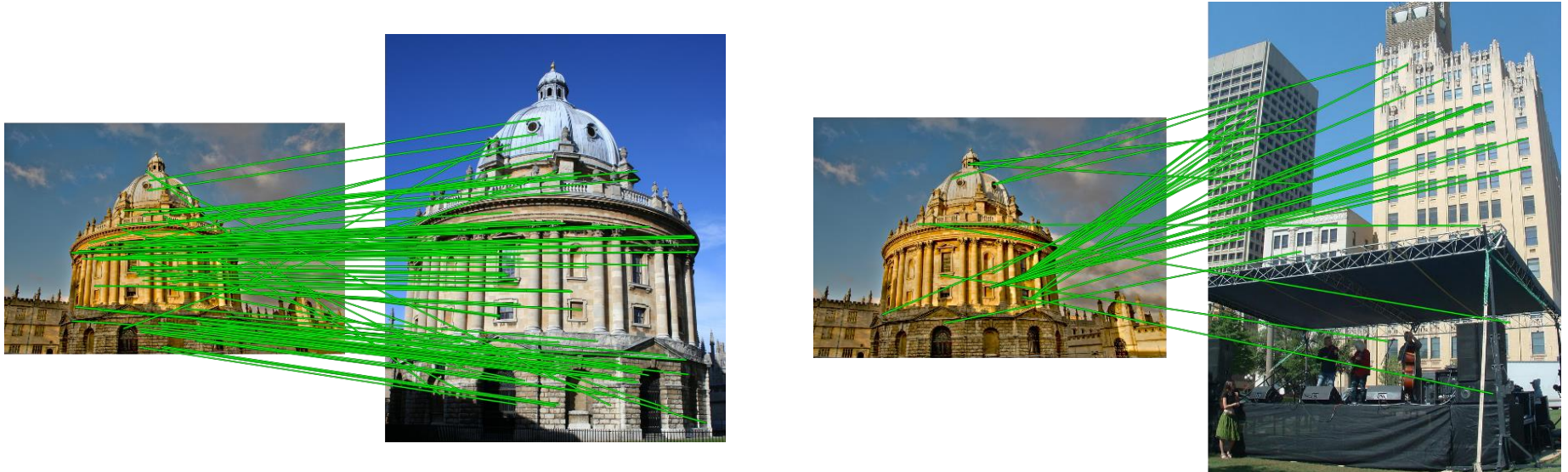
DB image with high BoW
similarity

Only some of the matches are mutually consistent.

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

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

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

True positive (tp) – correct attribution

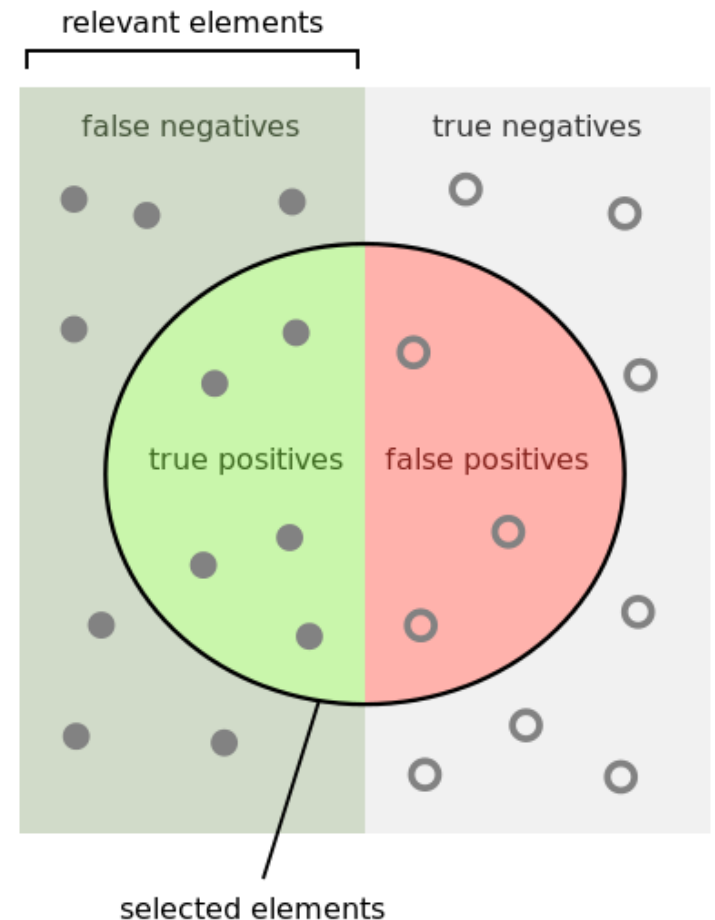
True negative (tn) – correct rejection

False positive (fp) – incorrect attribution

False negative (fn) – incorrect rejection

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

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



How many selected items are relevant?

Precision = $\frac{\text{green semi-circle}}{\text{green semi-circle} + \text{red semi-circle}}$

How many relevant items are selected?

Recall = $\frac{\text{green semi-circle}}{\text{green semi-circle} + \text{green rectangle}}$

Scoring retrieval quality



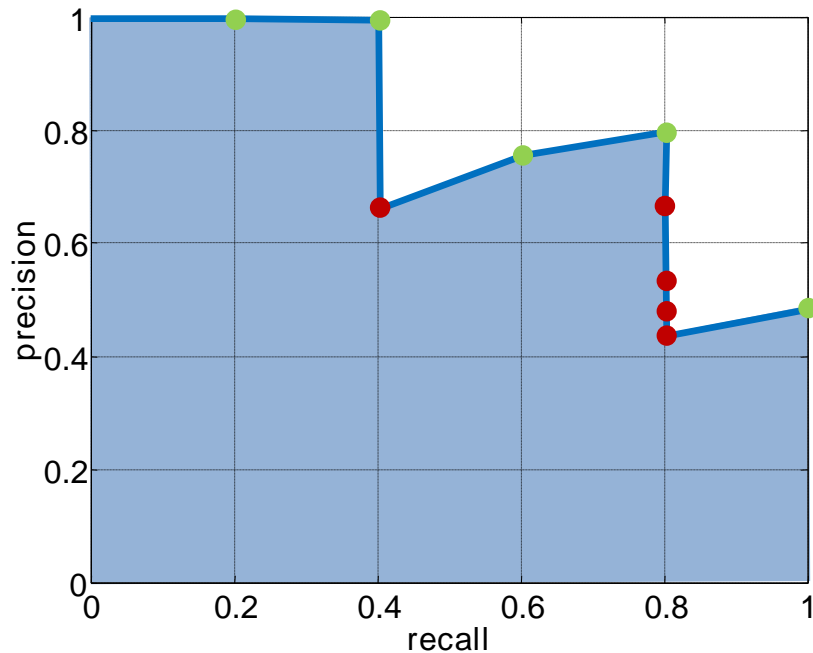
Query

Database size: 10 images

Relevant (total): 5 images

precision = $\# \text{relevant} / \# \text{returned}$

recall = $\# \text{relevant} / \# \text{total relevant}$



Results (ordered):



Query expansion

Results



Query image

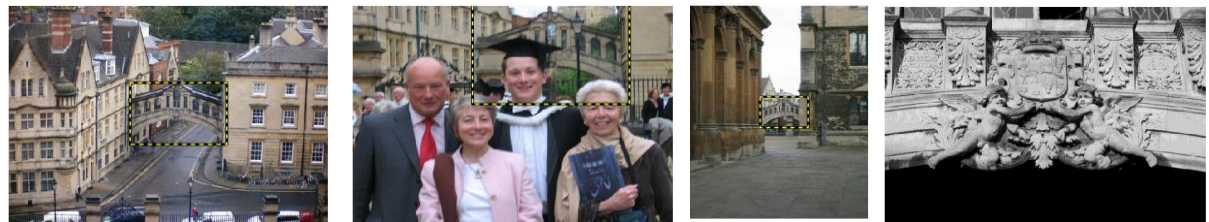


Spatial verification



New query

New results



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

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.

