

Computer Vision

CS-E4850, 5 study credits

Lecturer: Juho Kannala

Lecture 6: Large-scale object instance recognition/retrieval

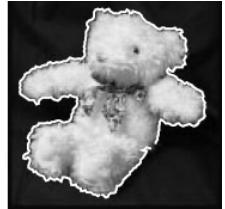
- Given a large image database of object instances, we would like to quickly recognize the objects present in a query image
- Or, given a query image of an object instance, we would like to retrieve all images of the same object from the database

Acknowledgement: many slides from James Hays, Kristen Grauman, Svetlana Lazebnik, Ondrej Chum, David Nister and others (detailed credits on individual slides)

Reading

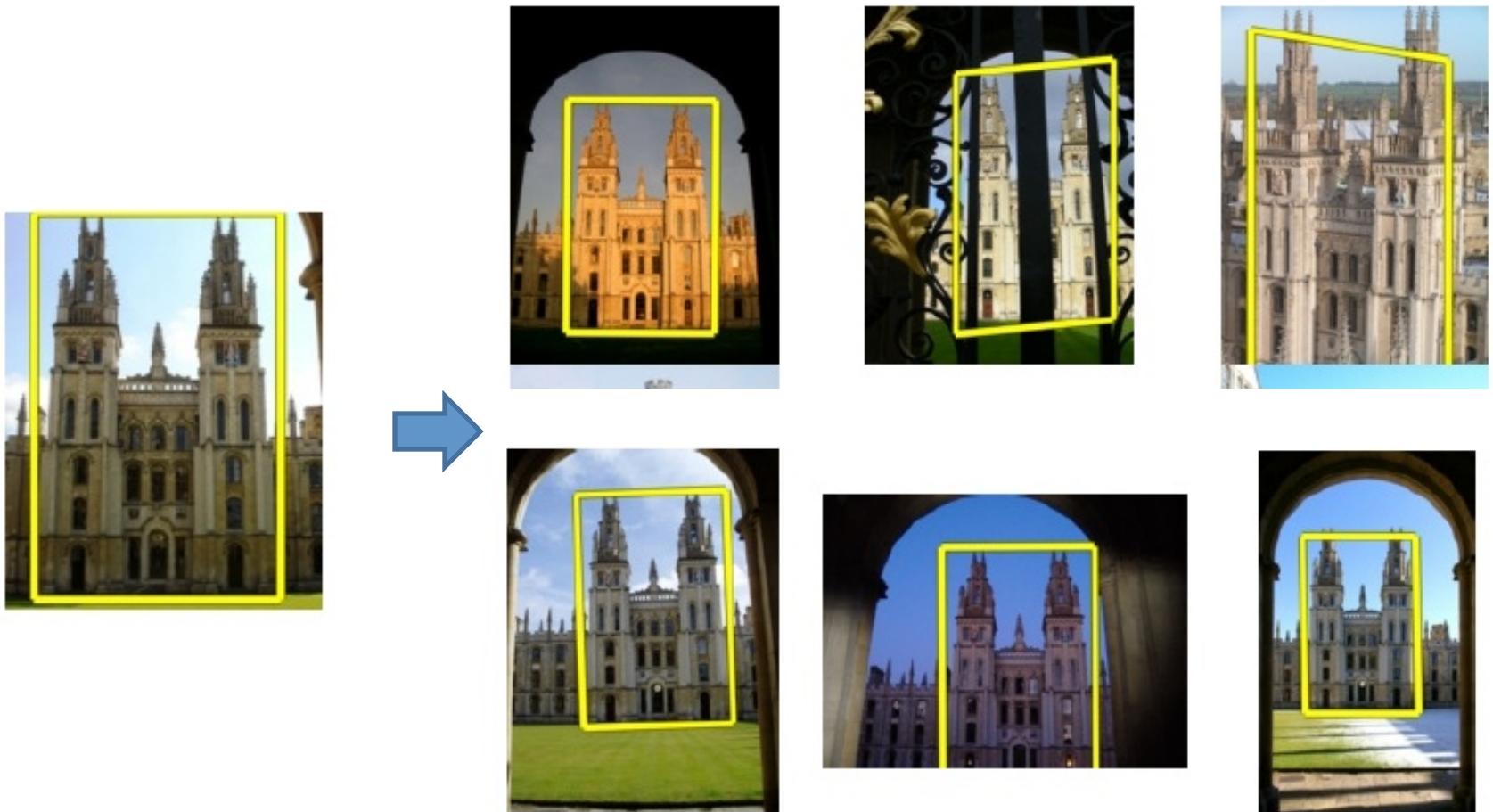
- Szeliski's book, Section 14.3 (pages 602 – 610)
- Sivic & Zisserman: Video Google, 2003
 - <http://www.robots.ox.ac.uk/~vgg/research/vgoogle/>
- Nister & Stewenius: Scalable recognition with a vocabulary tree, 2006
 - <http://vis.uky.edu/~stewe/ukbench/>
- Philbin et al.: Object retrieval with large vocabularies, 2007
 - <http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html>
- Software:
 - <http://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/index.html>

Local features for object instance recognition



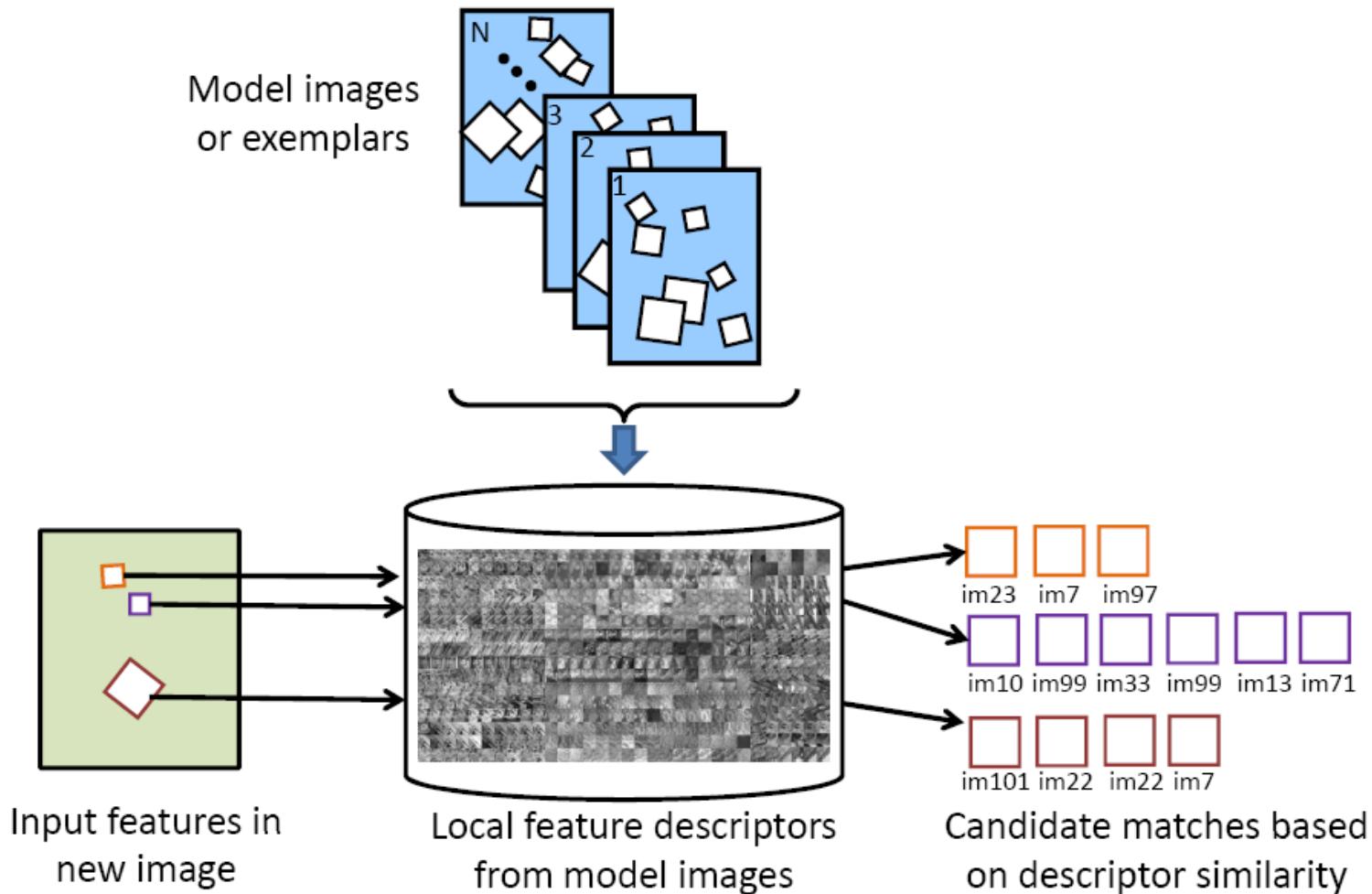
D. Lowe (1999, 2004)

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



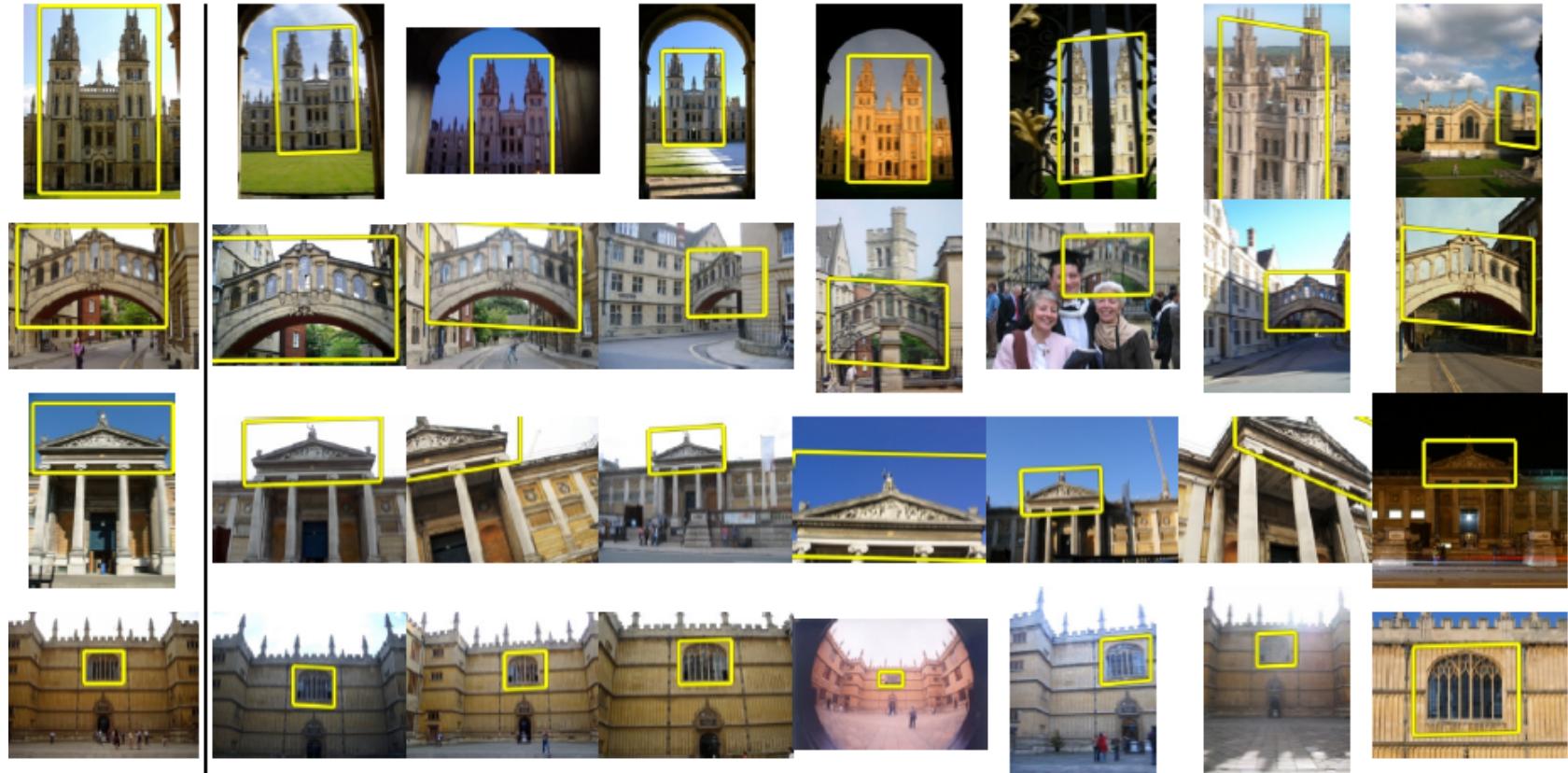
Large-scale image search

Combining local features, indexing, and spatial constraints



Large-scale image search

Combining local features, indexing, and spatial constraints



Large-scale image search

Combining local features, indexing, and spatial constraints

Google Goggles in Action

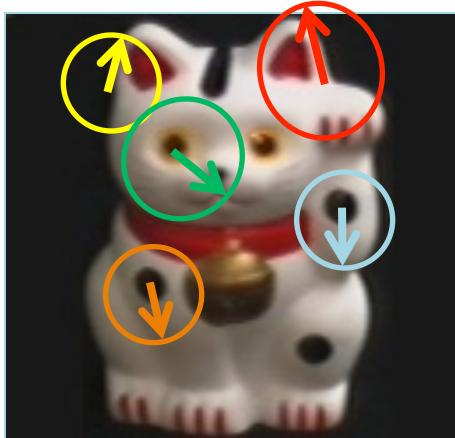
Click the icons below to see the different ways Google Goggles can be used.



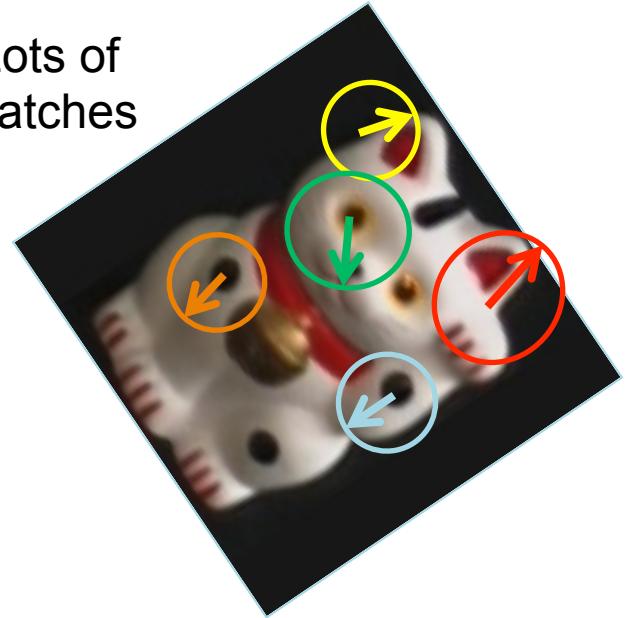
Available on phones that run Android 1.6+ (i.e. Donut or Eclair)

Simple idea

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



Lots of
Matches



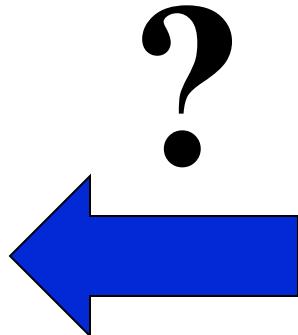
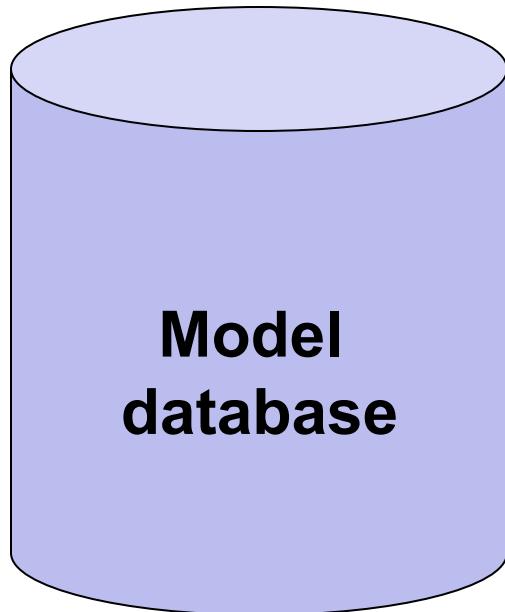
Few or No
Matches



But this will be really, really slow!

Scalability: Alignment to large databases

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



Large-scale visual search

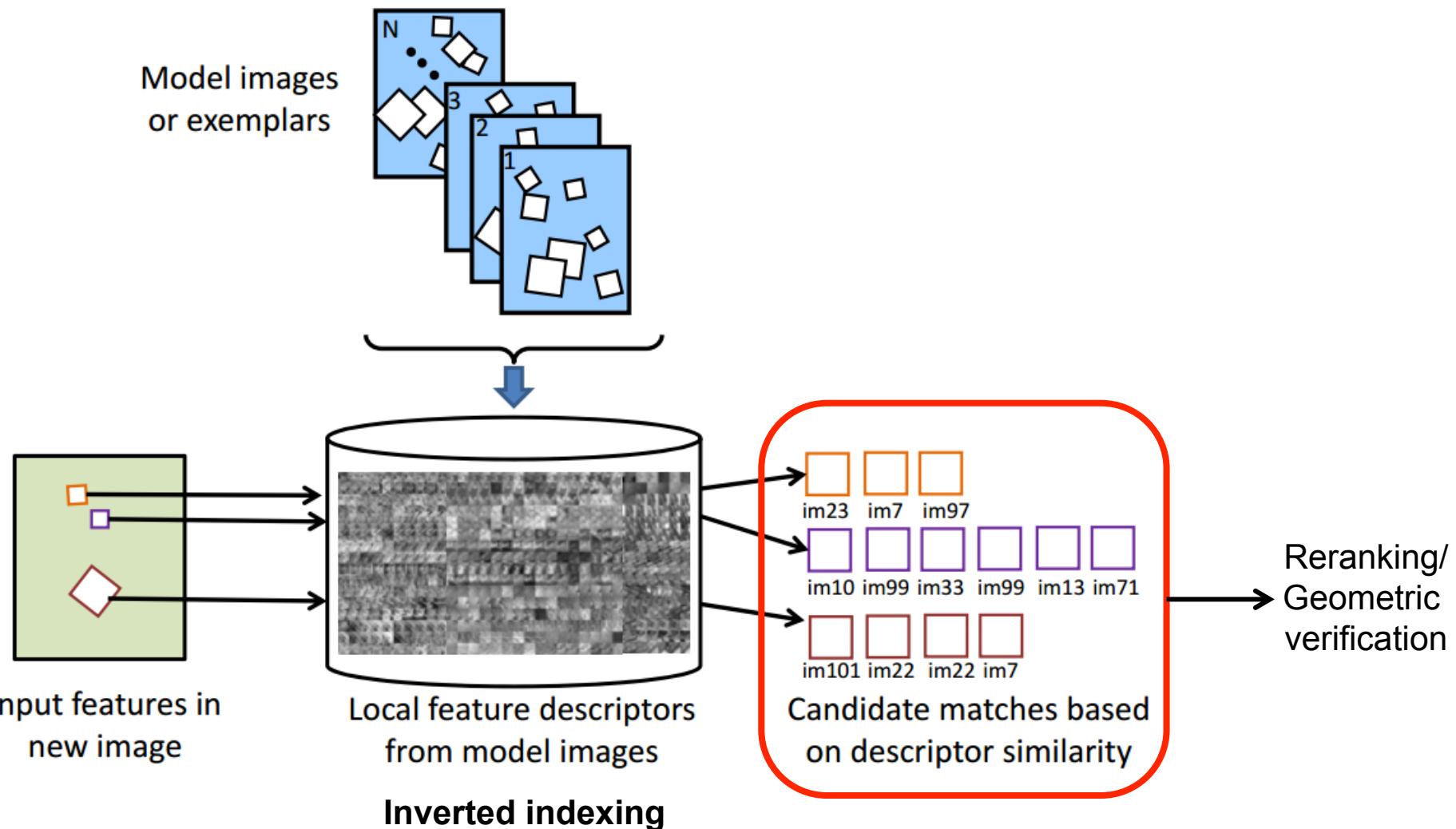
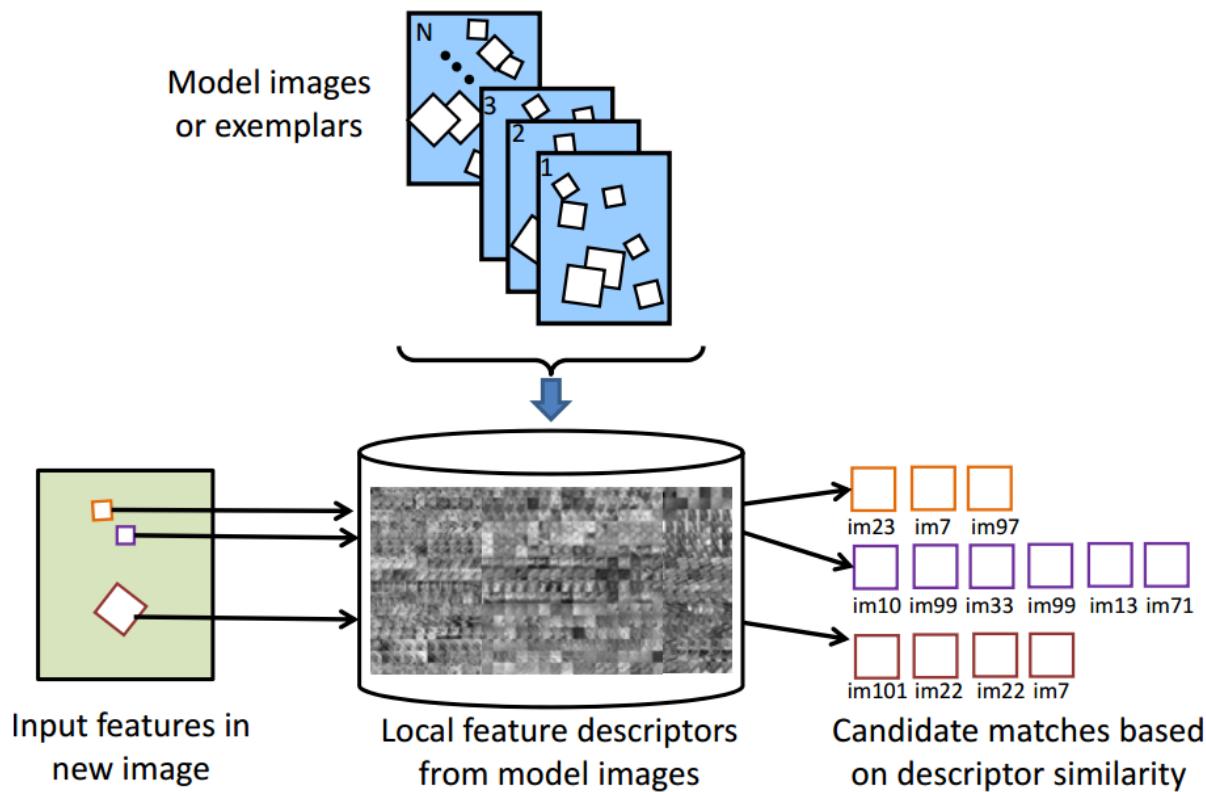


Figure from: Kristen Grauman and Bastian Leibe, [Visual Object Recognition](#), Synthesis Lectures on Artificial Intelligence and Machine Learning, April 2011, Vol. 5, No. 2, Pages 1-181

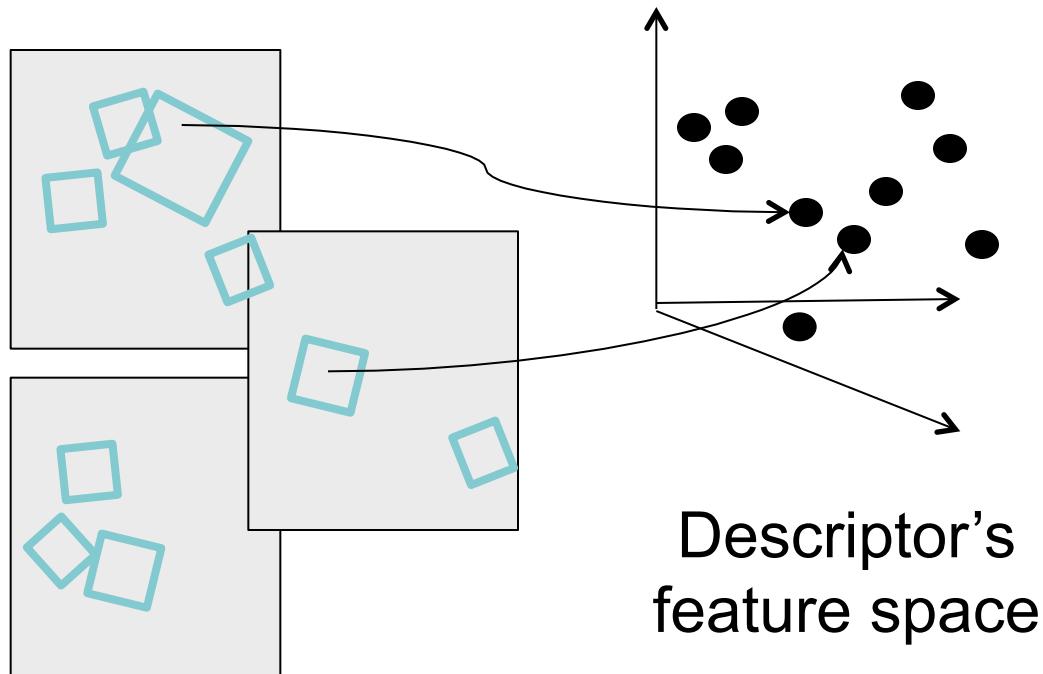
How to do the indexing?



- Idea: find a set of *visual codewords* to which descriptors can be *quantized*

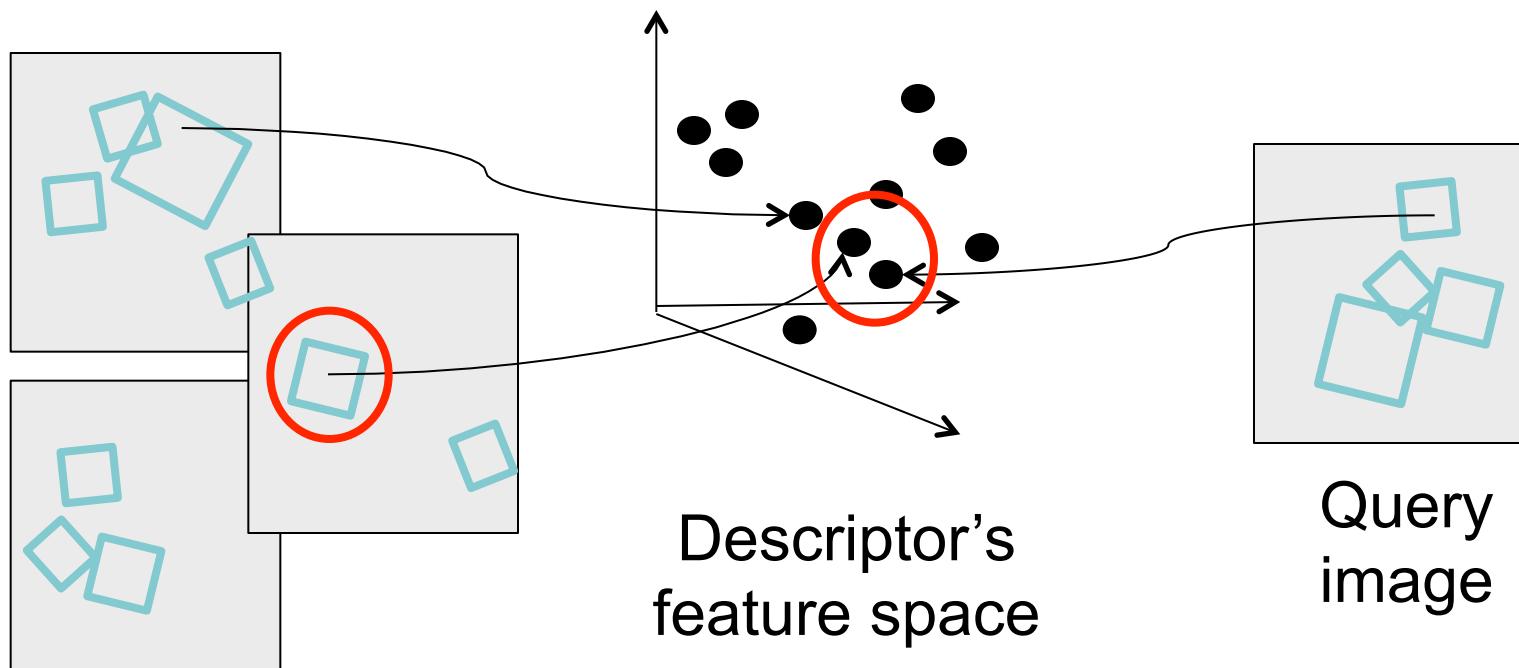
Indexing local features

- 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.



Database
images

Descriptor's
feature space

*Easily can have millions of
features to search!*

Indexing local features: inverted file index

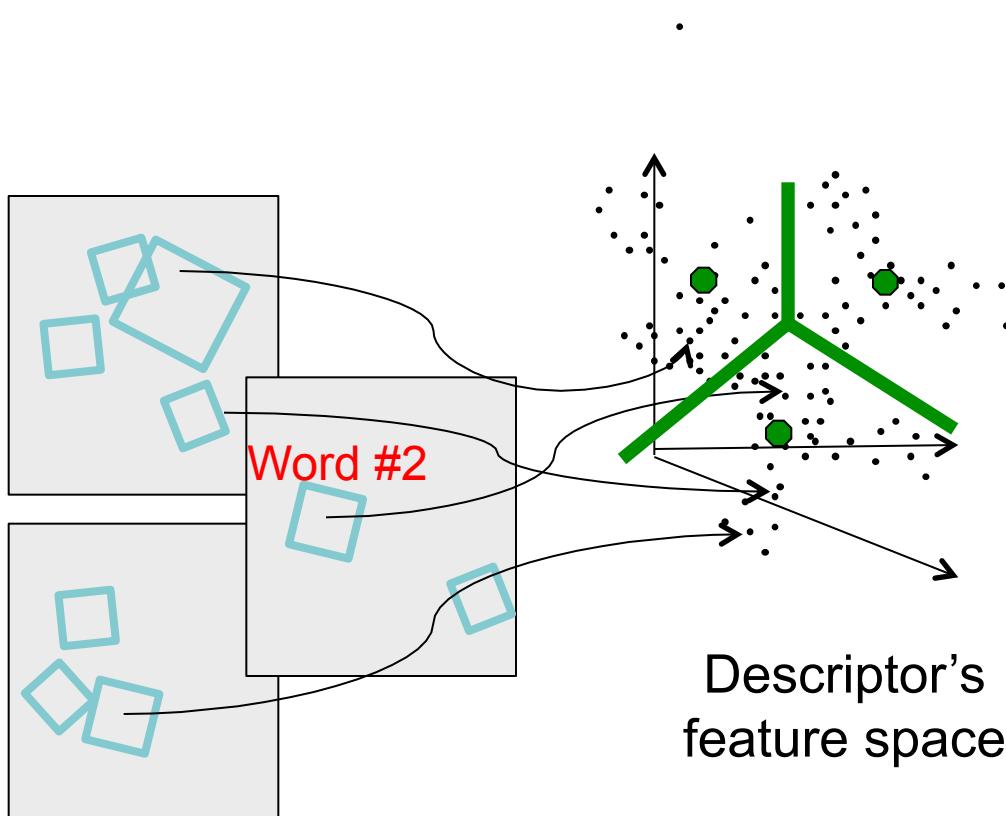
Index

"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142	Eau Gallie; 175
511 Traffic Information; 83	Ca d'Zan; 147	Edison, Thomas; 152
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Reale; 176
AAA National Office; 88	Canaveral Natnl Seashore; 173	Ellenton; 144-145
Abbreviations,	Cannon Creek Airpark; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106,160	Emergency Callboxes; 83
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142,148,157,159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge (I-10); 119
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150	County; 120
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93	Esterero; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglade,90,95,139-140,154-160
Alabama; 124	Charlotte Harbor; 150	Draining of; 156,181
Alachua; 132	Chautauqua; 116	Wildlife MA; 160
County; 131	Chipley; 114	Wonder Gardens; 154
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawatchee, Name; 115	Fantasy of Flight; 95
Alfred B Macay Gardens; 106	Circus Museum, Ringling; 147	Fayer Dykes SP; 171
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180	Fires, Forest; 166
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180	Fires, Prescribed; 148
Alligator Hole (definition); 157	City Maps,	Fisherman's Village; 151
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195	Flagler County; 171
Alligators; 100,135,138,147,156	Jacksonville; 163	Flagler, Henry; 97,165,167,171
Anastasia Island; 170	Kissimmee Expwys; 192-193	Florida Aquarium; 186
Anhica; 108-109,146	Miami Expressways; 194-195	Florida,
Apalachicola River; 112	Orlando Expressways; 192-193	12,000 years ago; 187
Appleton Mus of Art; 136	Pensacola; 26	Cavern SP; 114
Aquifer; 102	Tallahassee; 191	Map of all Expressways; 2-3
Arabian Nights; 94	Tampa-St. Petersburg; 63	Mus of Natural History; 134
Art Museum, Ringling; 147	St. Augustine; 191	National Cemetery ; 141
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141	Part of Africa; 177
Aucilla River Project; 106	Clearwater Marine Aquarium; 187	Platform; 187
Babcock-Web WMA; 151	Collier County; 154	Sheriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 130
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Barefoot Mallmen; 182	Columbia County; 101,128	Supreme Court; 107
Barge Canal; 137	Coquina Building Material; 165	Florida's Turnpike (FTP); 178,189
Bee Line Expy; 80	Corkscrew Swamp, Name; 154	25 mile Strip Maps; 66
Belz Outlet Mall; 89	Cowboys; 95	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big "I"; 165	Cracker, Florida; 88,95,132	Exit Services; 189
Big Cypress; 155,158	Crostown Expy; 11,35,98,143	HEFT; 76,161,190
Big Foot Monster; 105	Cuban Bread; 184	History; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Names; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161	Service Plazas; 190
Blue Angels	Daniel Beach Hurricane; 184	Spur SR91; 76
	Daniel Boone, Florida Walk; 117	Ticket System; 190
	Daytona Beach; 172-173	Toll Plazas; 190
	De Land; 87	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.
- To use this idea, we'll need to map our features to "visual words".

Visual words

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



- Quantize via clustering, let cluster centers be the prototype “words”
- Determine which word to assign to each new image region by finding the closest cluster center.

K-means clustering

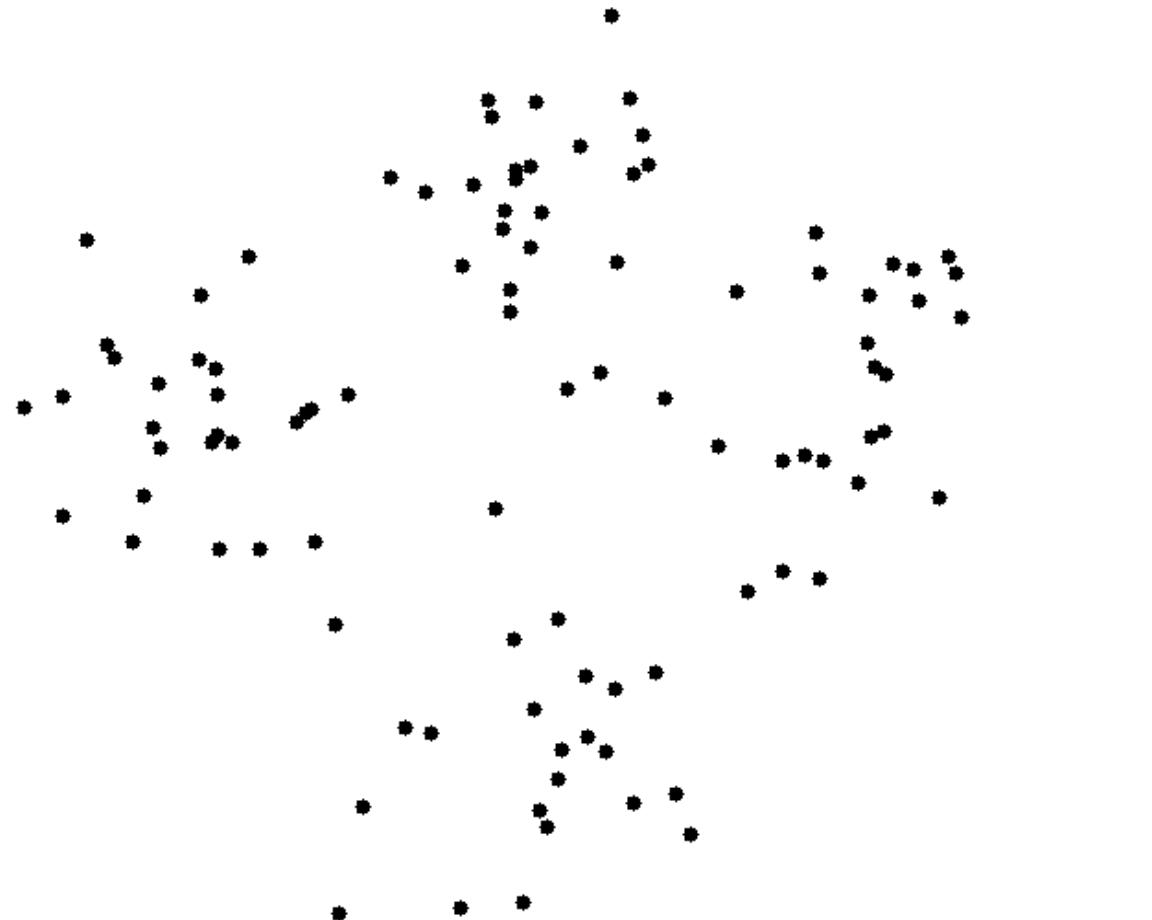
- Want to minimize sum of squared Euclidean distances between points \mathbf{x}_i and their nearest cluster centers \mathbf{m}_k

$$D(X, M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$

Algorithm:

- Randomly initialize K cluster centers
- Iterate until convergence:
 - Assign each data point to the nearest center
 - Recompute each cluster center as the mean of all points assigned to it

K-means demo



Source: <http://shabal.in/visuals/kmeans/1.html>

Another demo: <http://www.kovan.ceng.metu.edu.tr/~maya/kmeans/>

Visual words

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

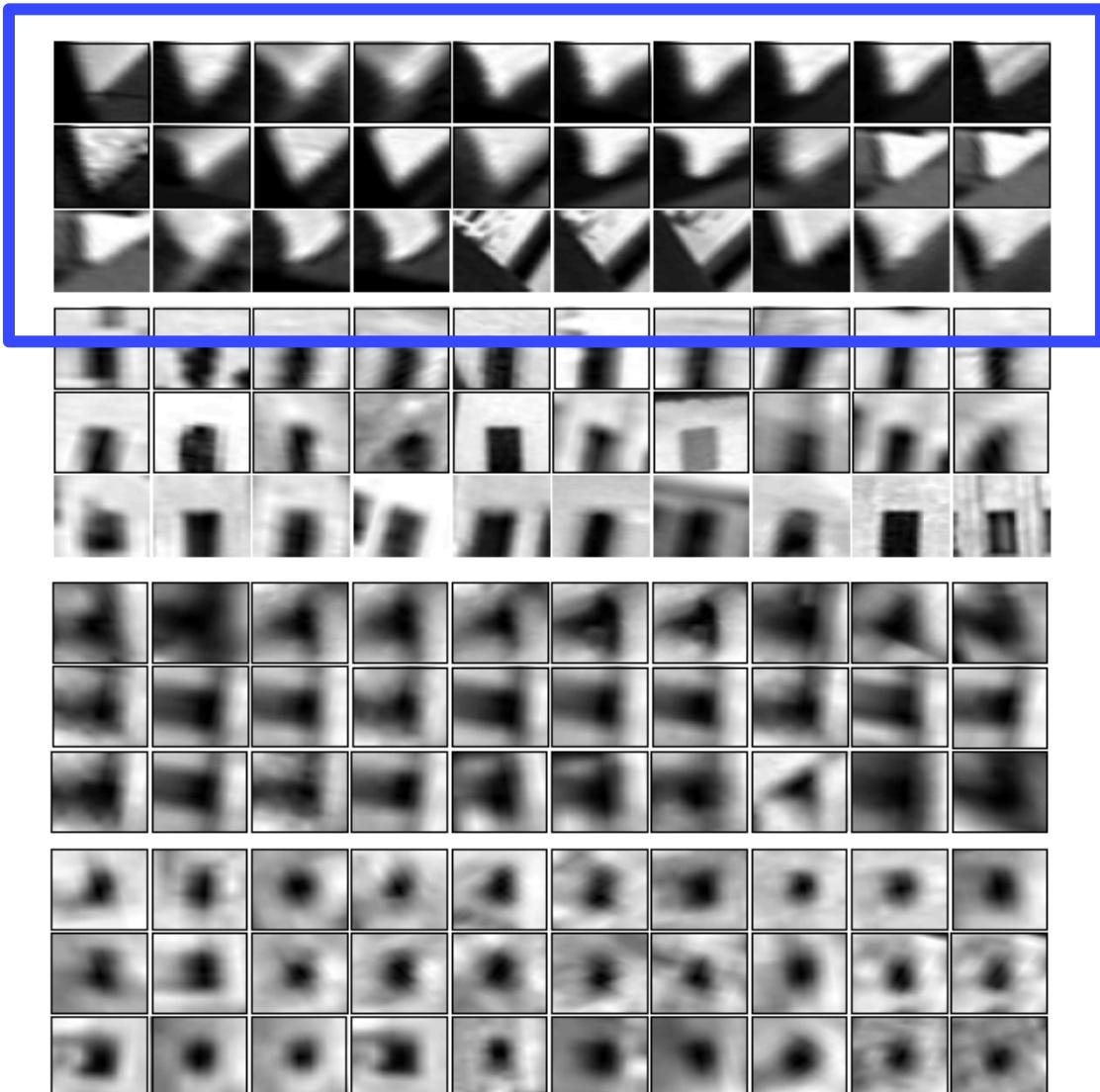
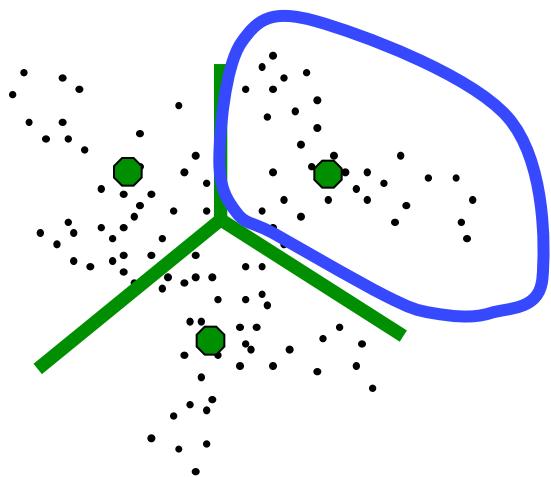
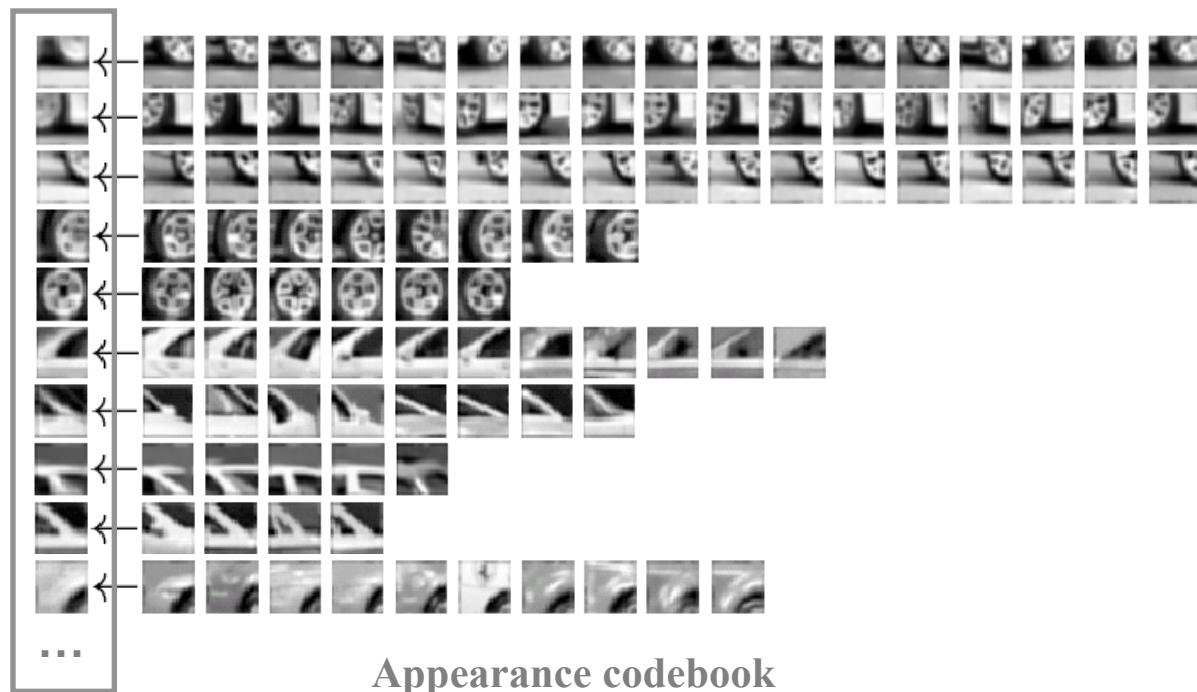
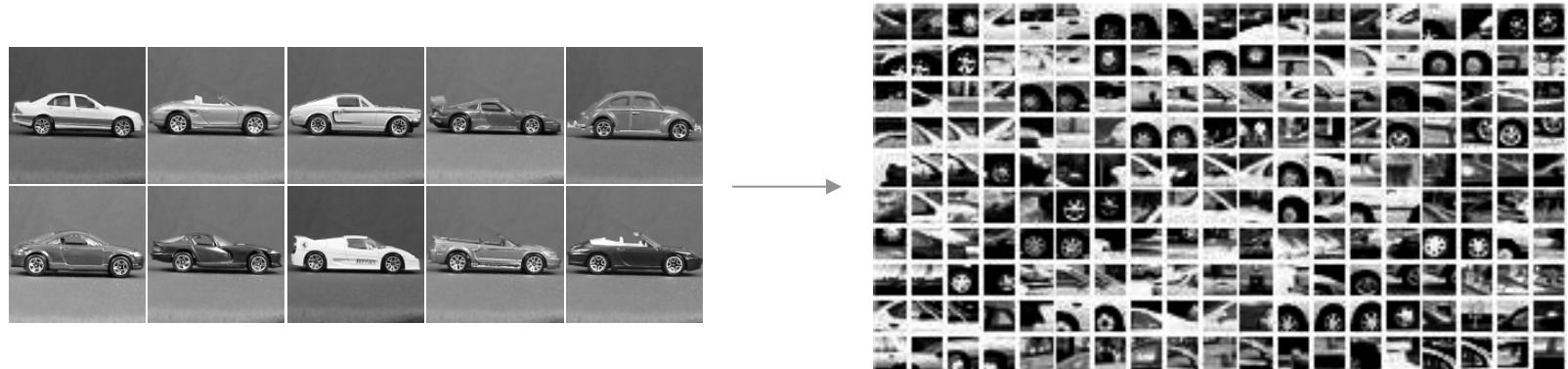
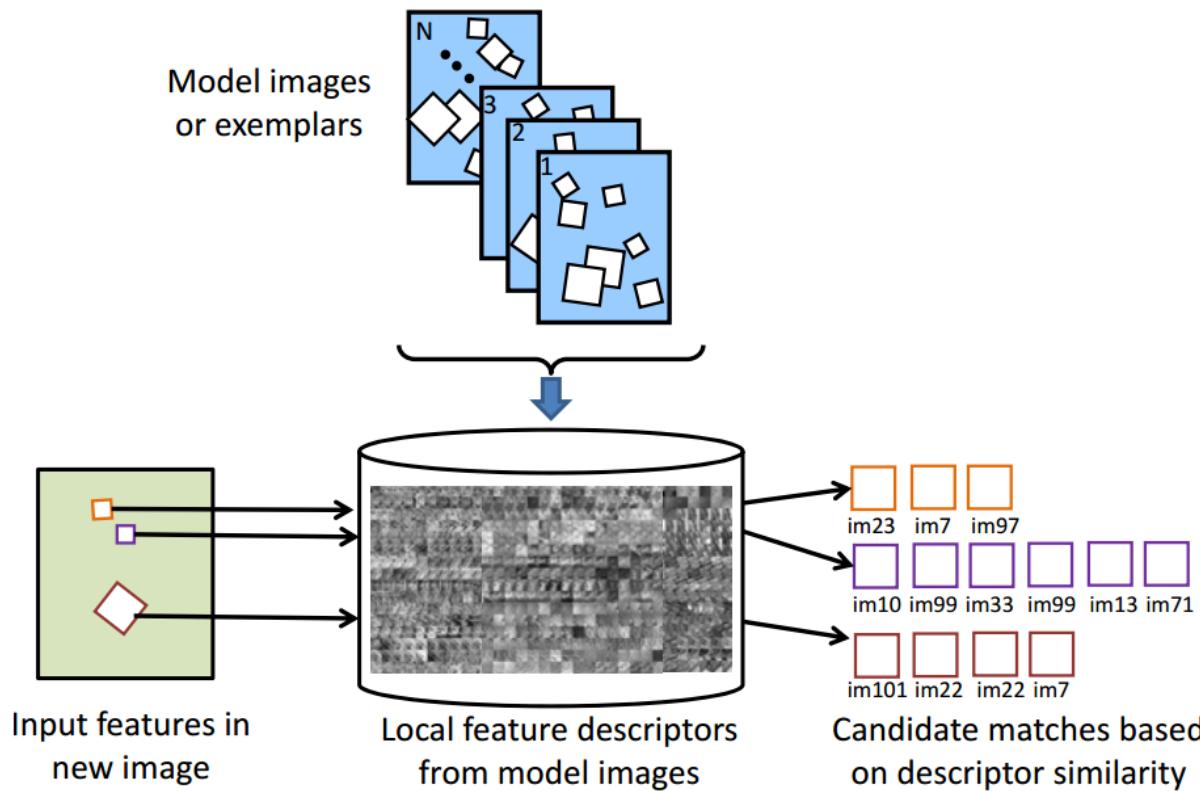


Figure from Sivic & Zisserman, ICCV 2003 Kristen Grauman

Example of a visual codebook



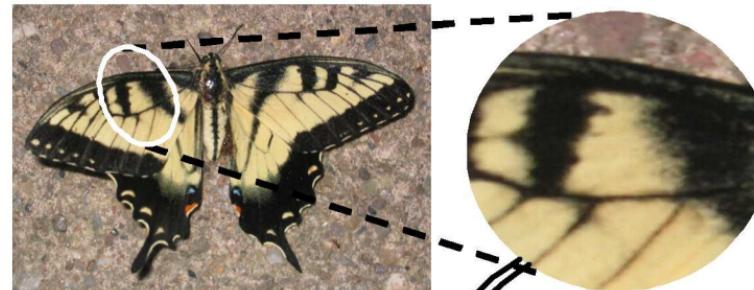
How to do the indexing?



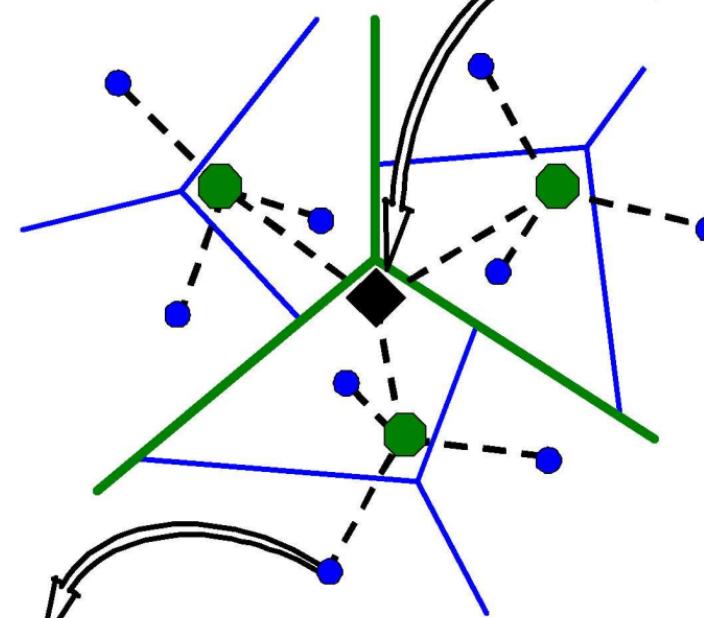
- Cluster descriptors in the database to form codebook
- At query time, quantize descriptors in query image to nearest codevectors
- Problem solved?

Efficient indexing technique: Vocabulary trees

Test image



Vocabulary tree
with inverted
index

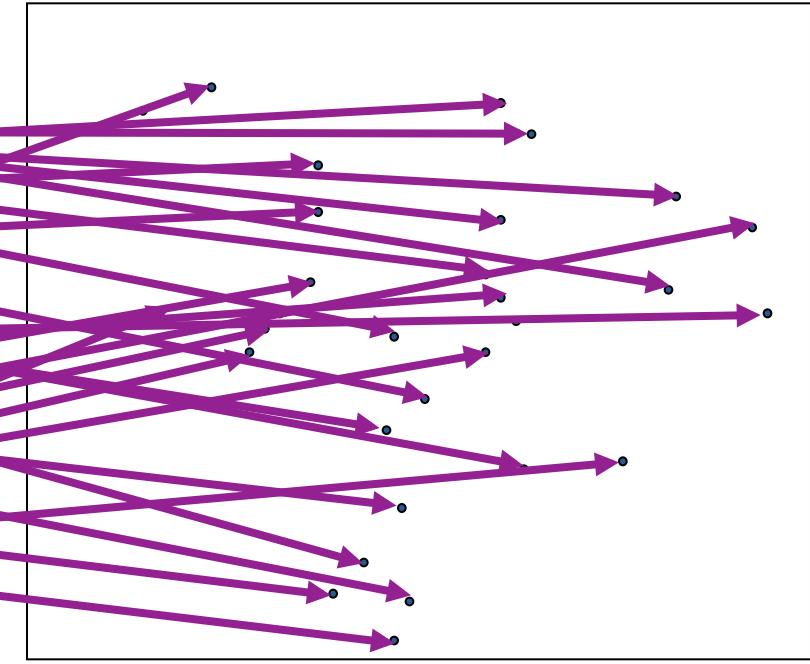
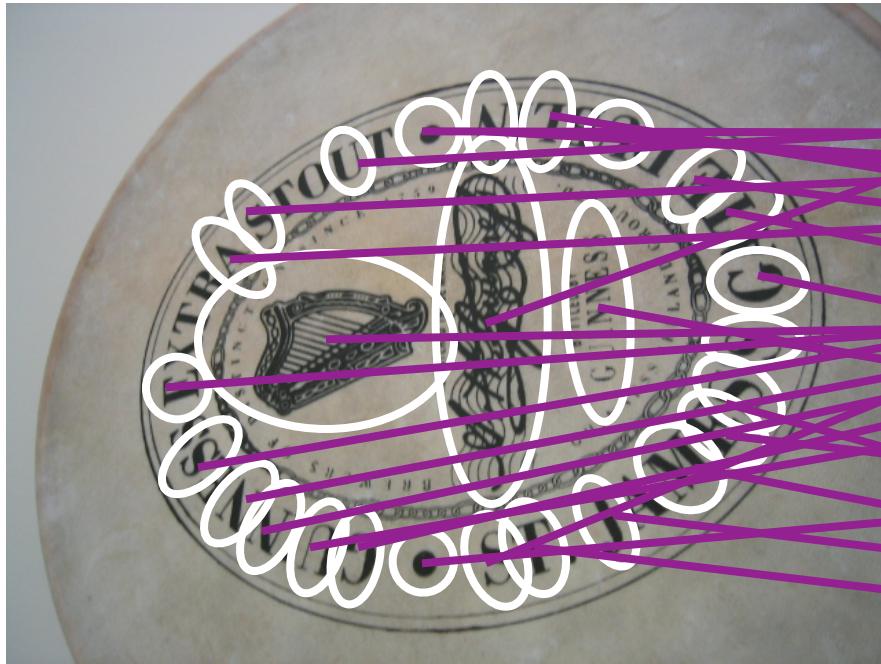


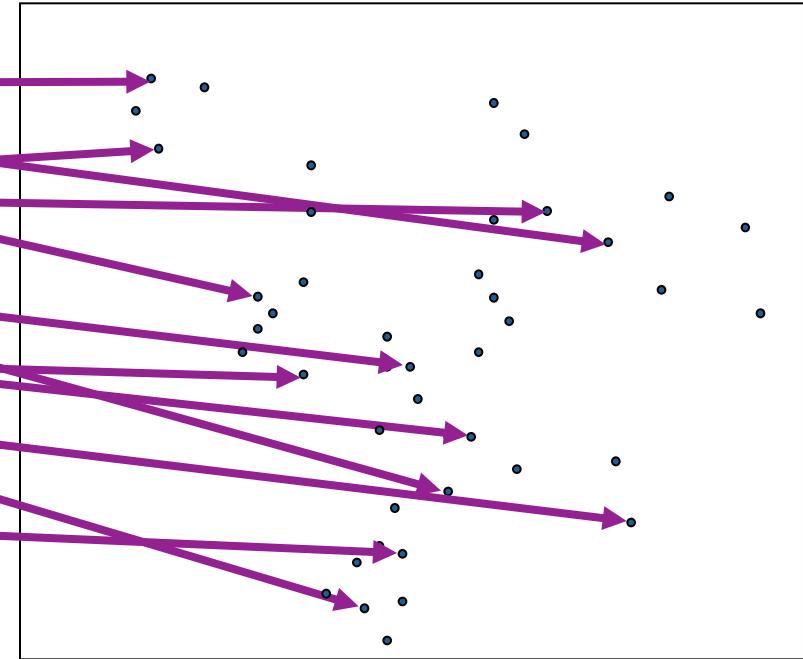
Database

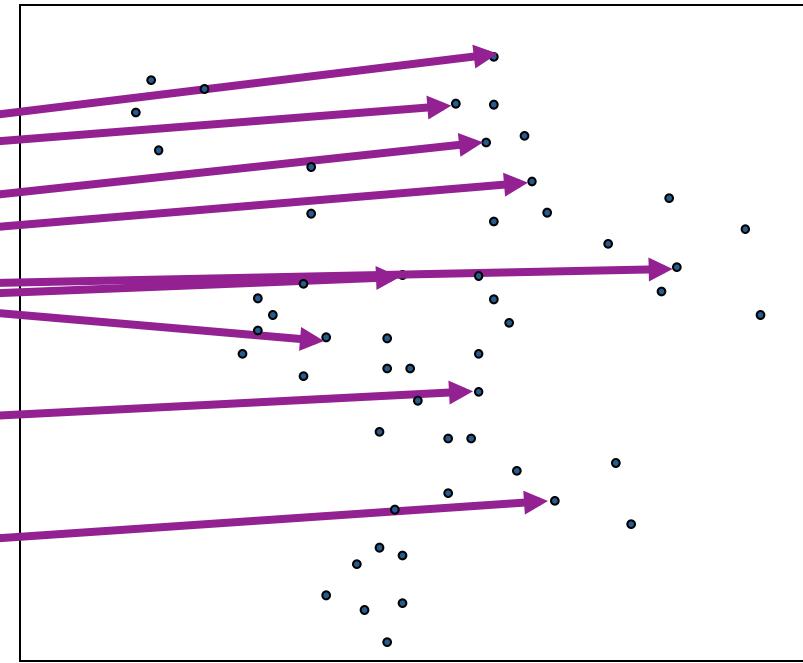
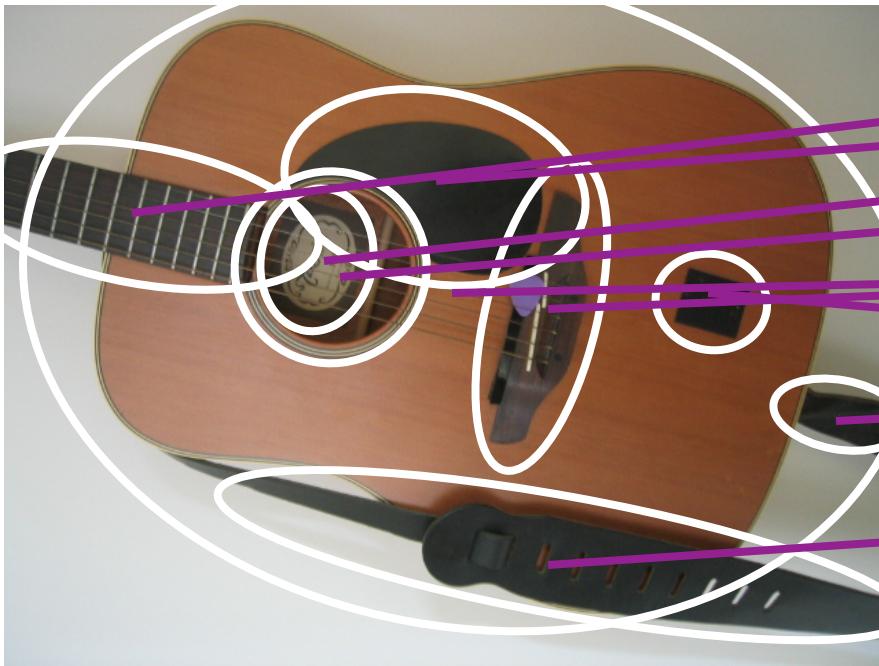


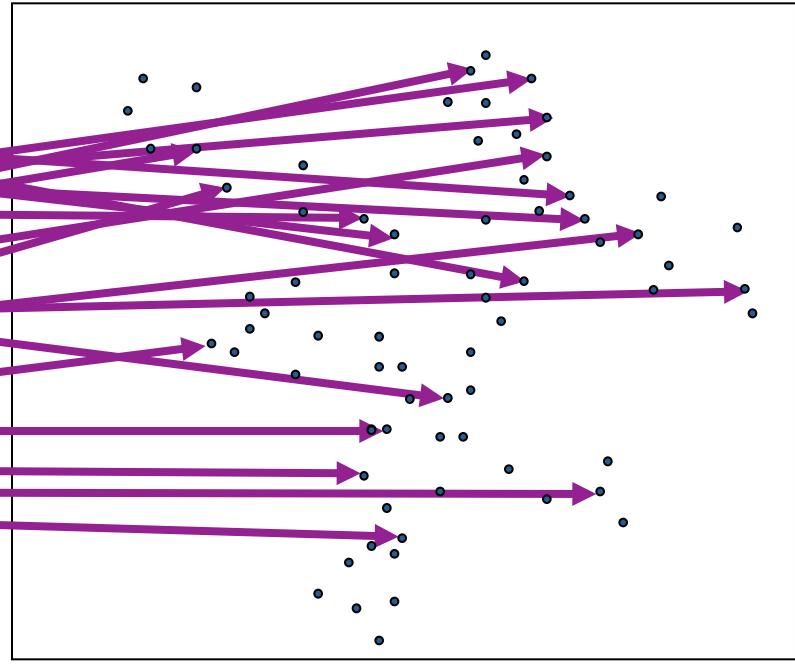
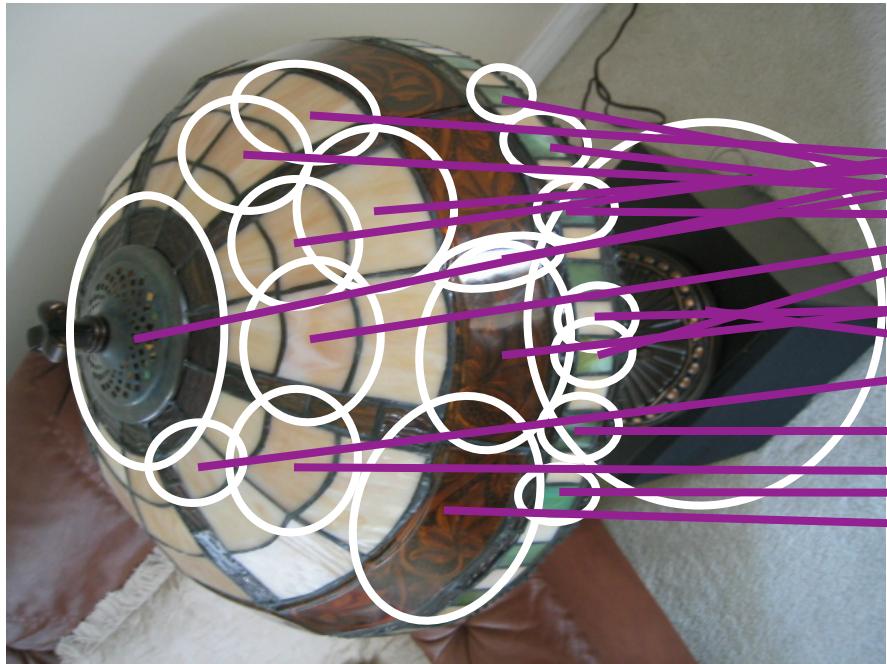
Recognition with K-tree

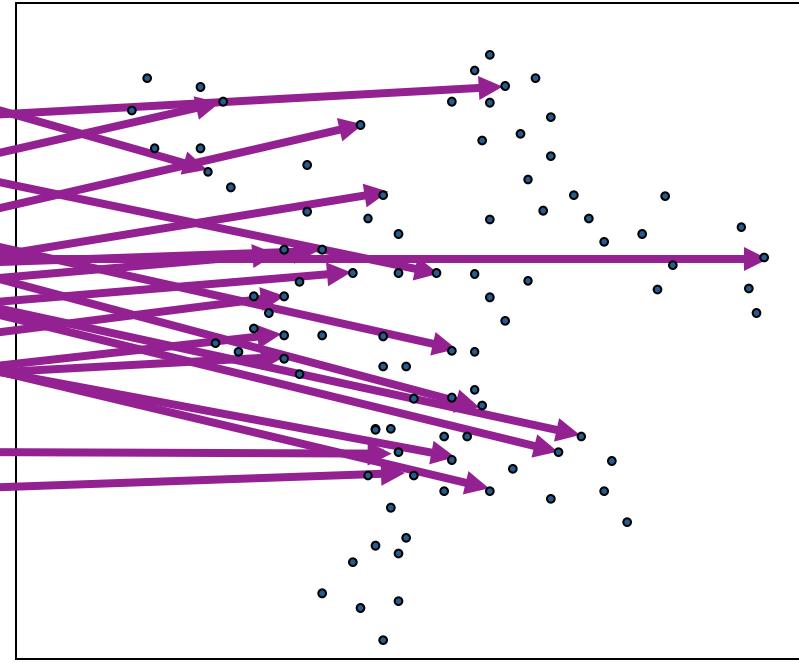
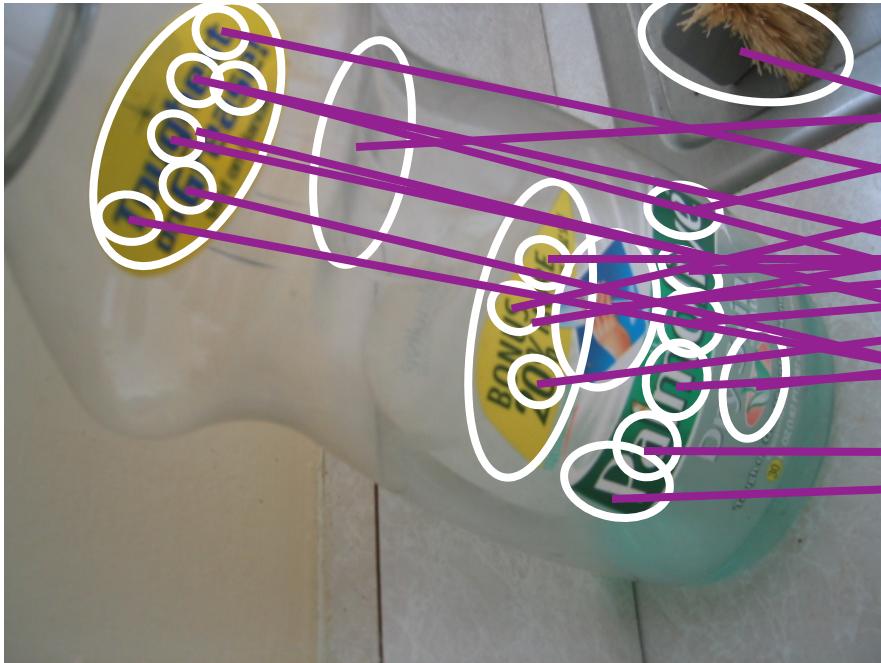
Following slides by David Nister (CVPR 2006)

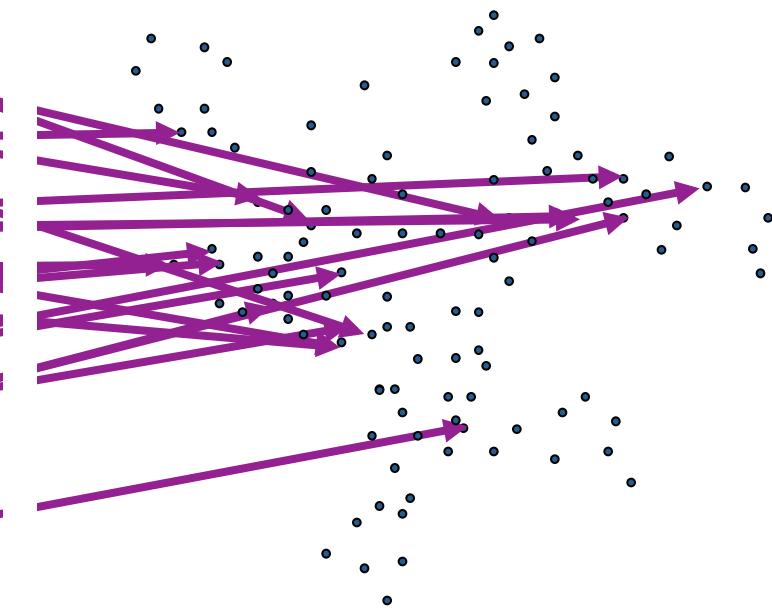


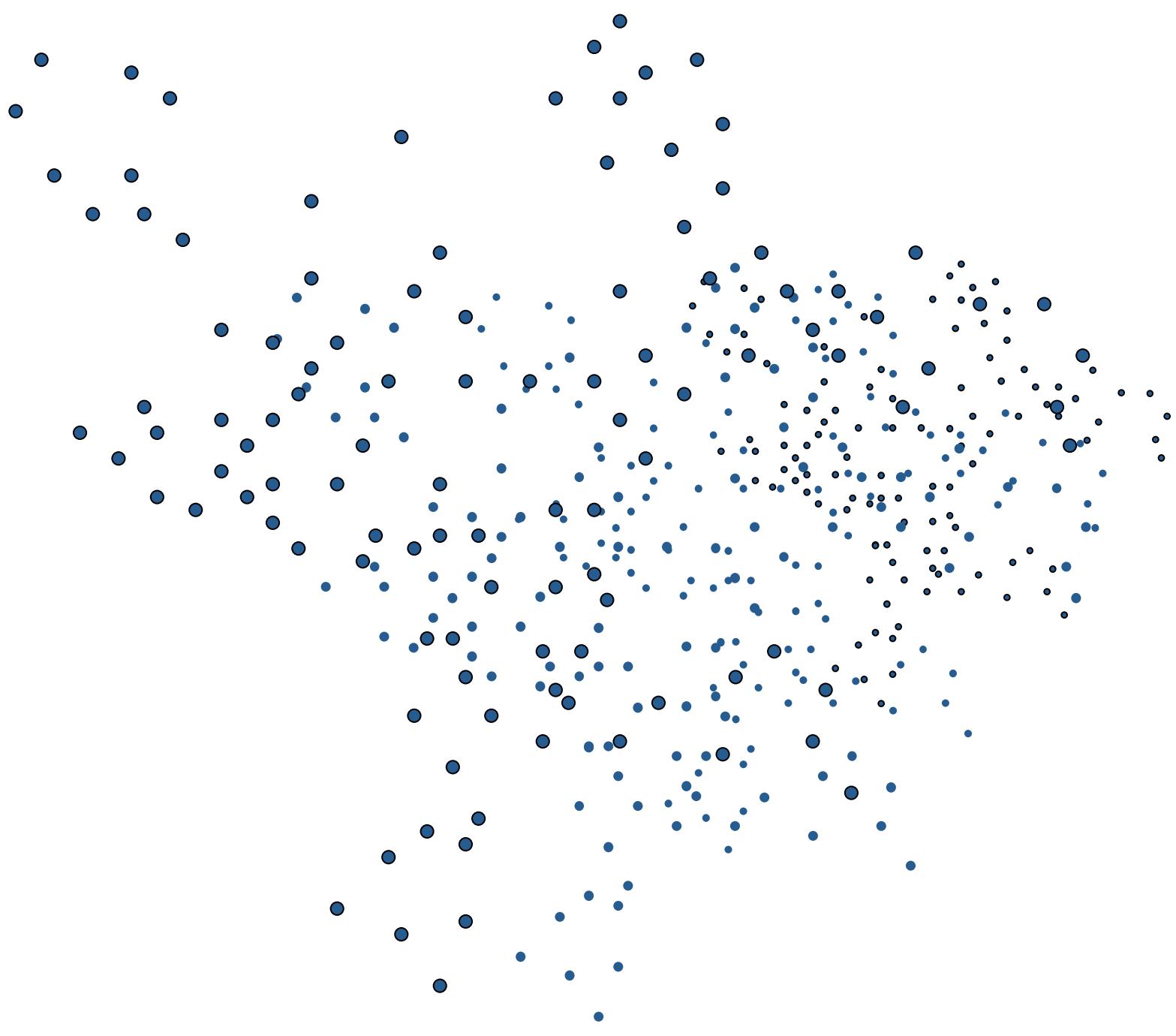


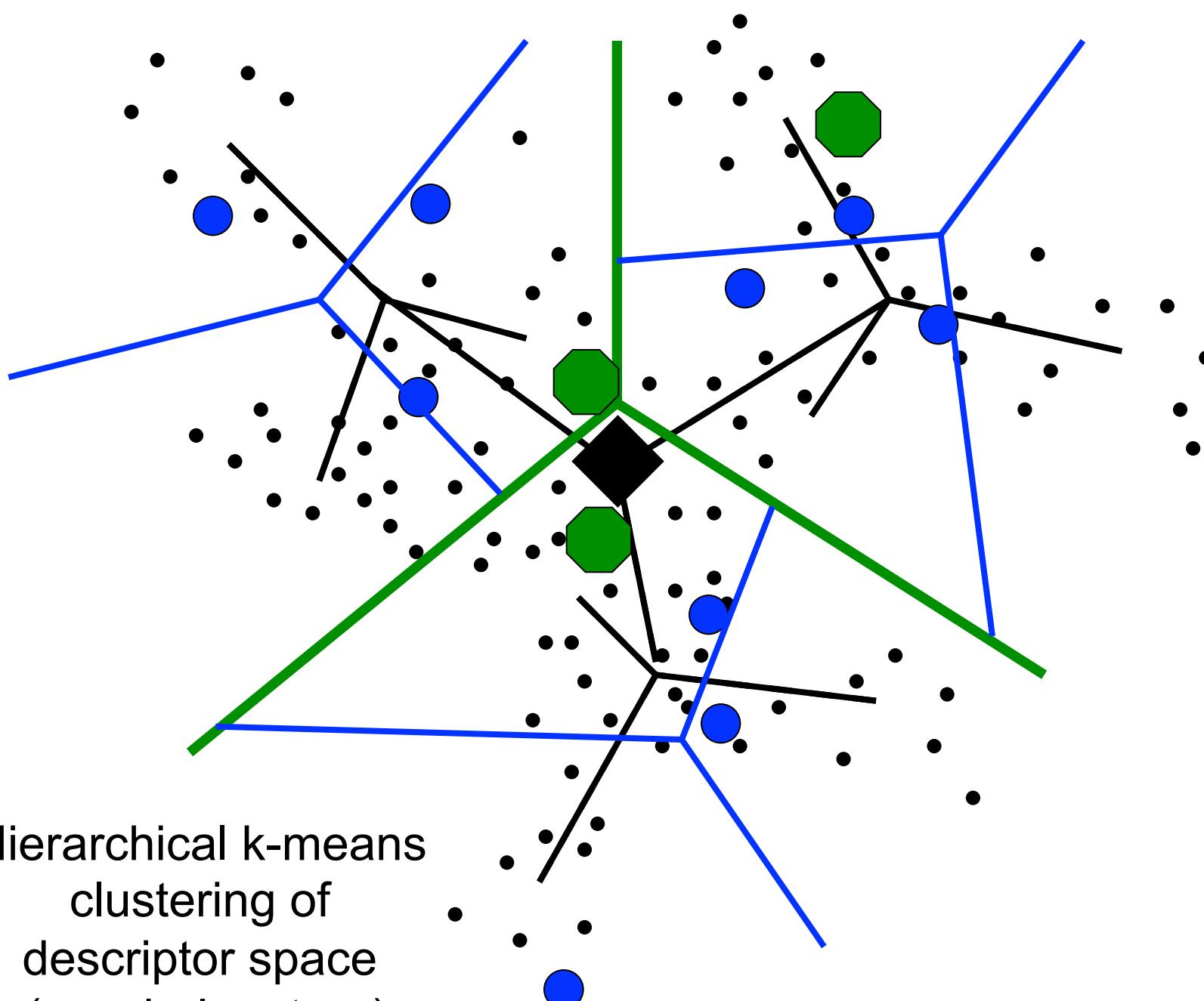






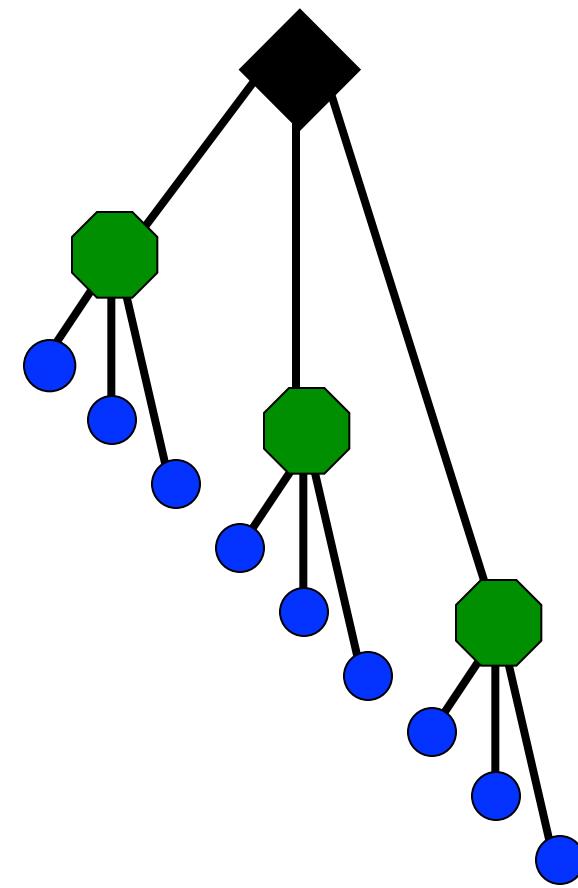






Hierarchical k-means
clustering of
descriptor space
(vocabulary tree)

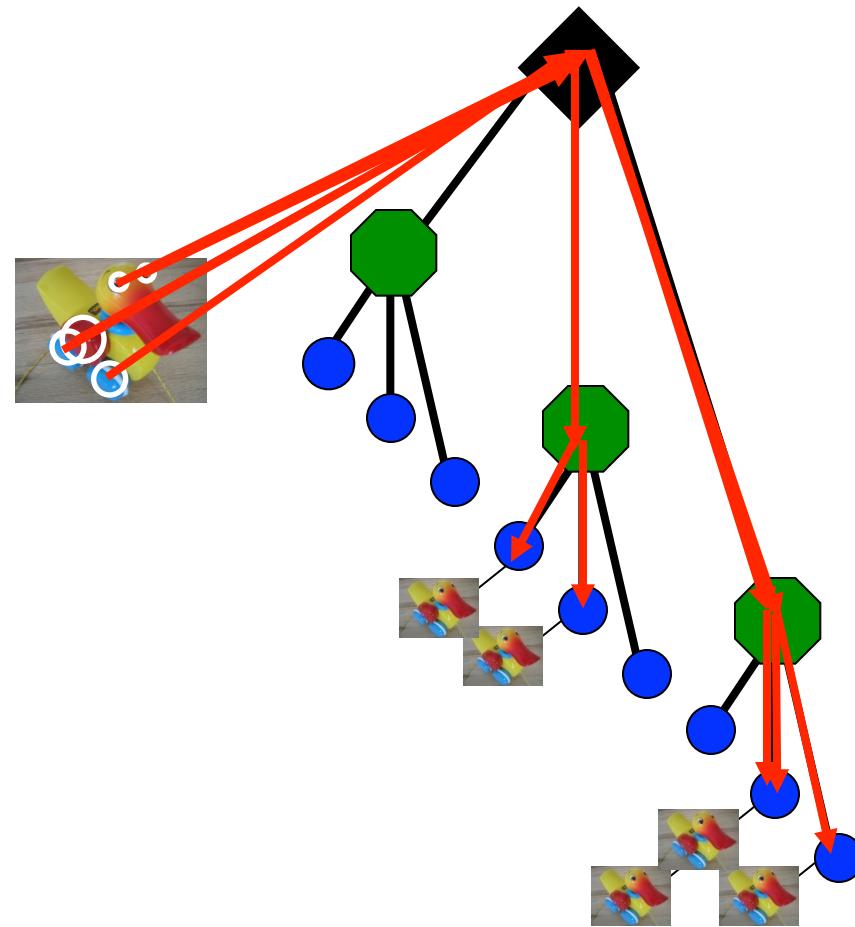
Slide credit: D. Nister



Vocabulary tree/inverted index

Slide credit: D. Nister

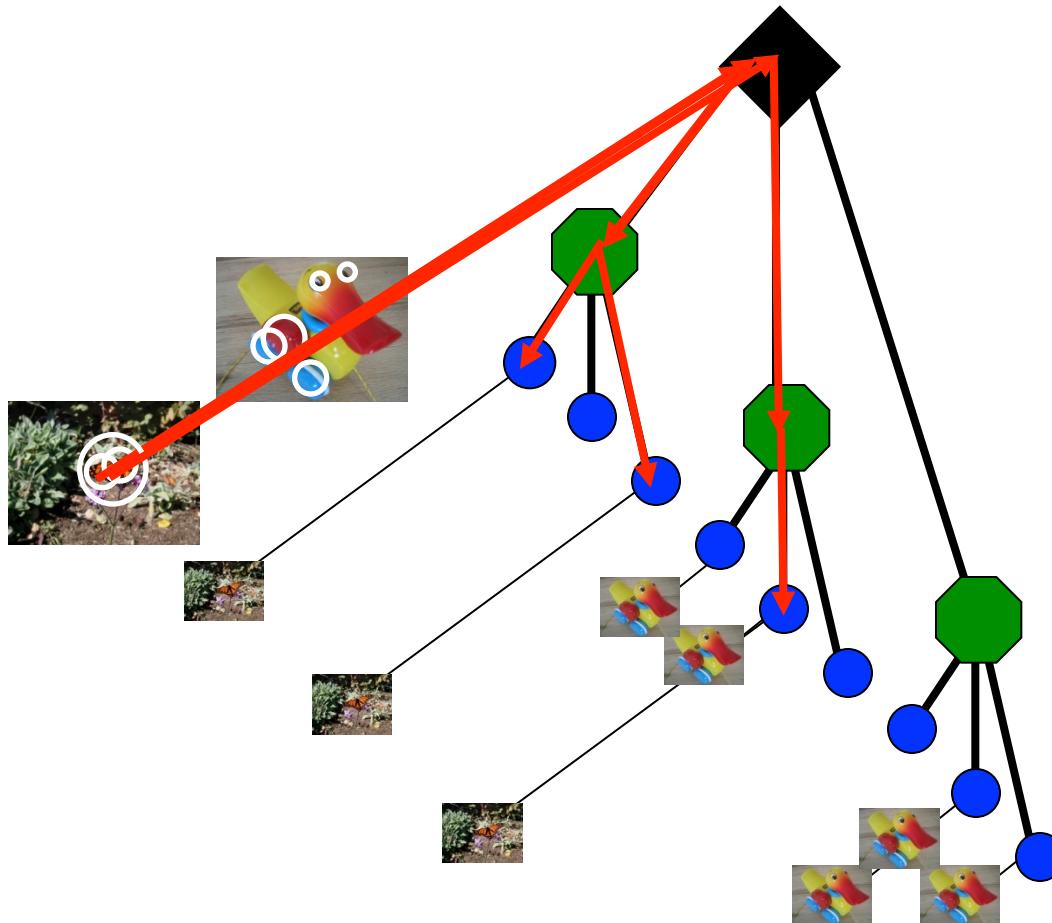
Model images



Populating the vocabulary tree/inverted index

Slide credit: D. Nister

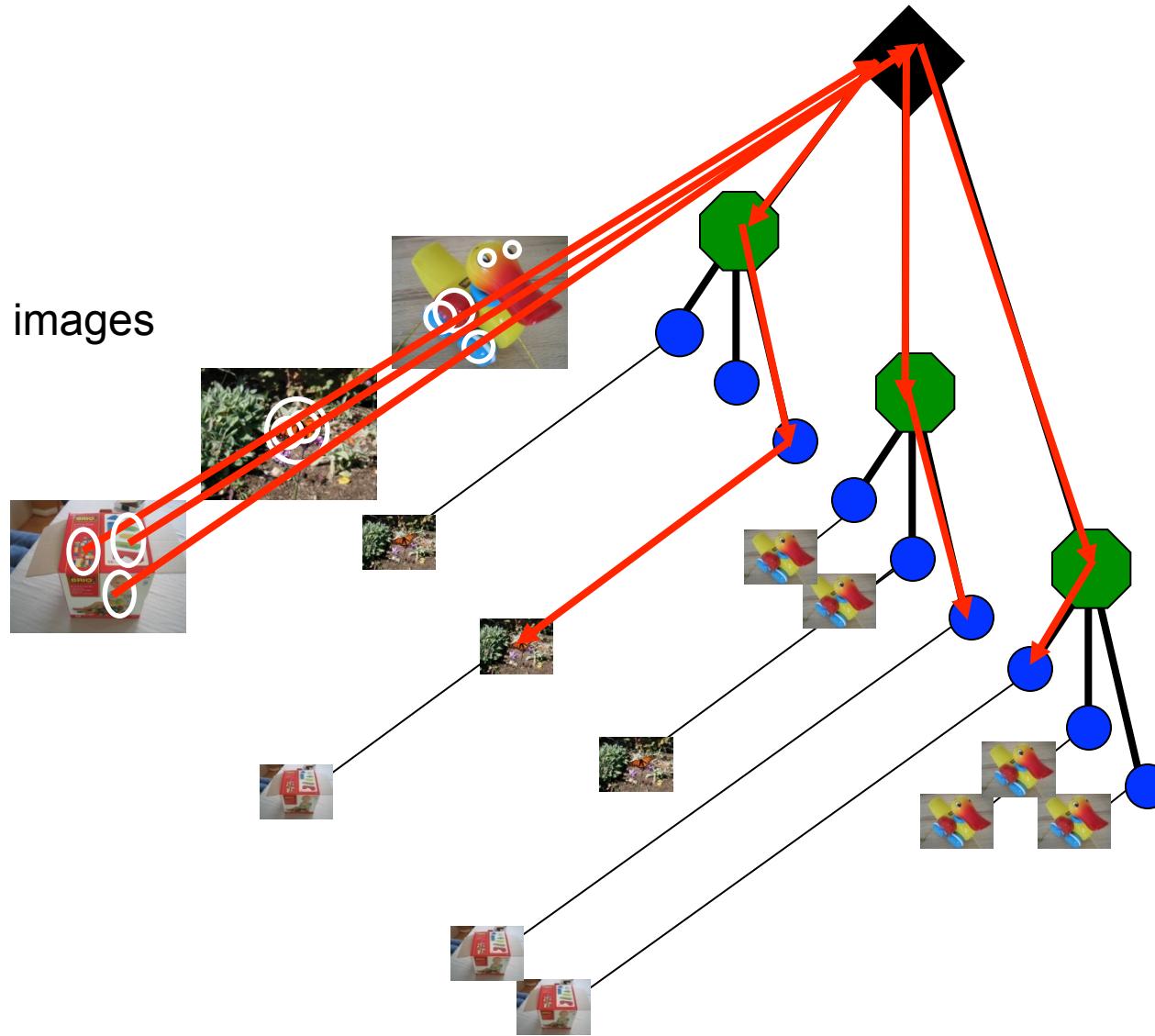
Model images



Populating the vocabulary tree/inverted index

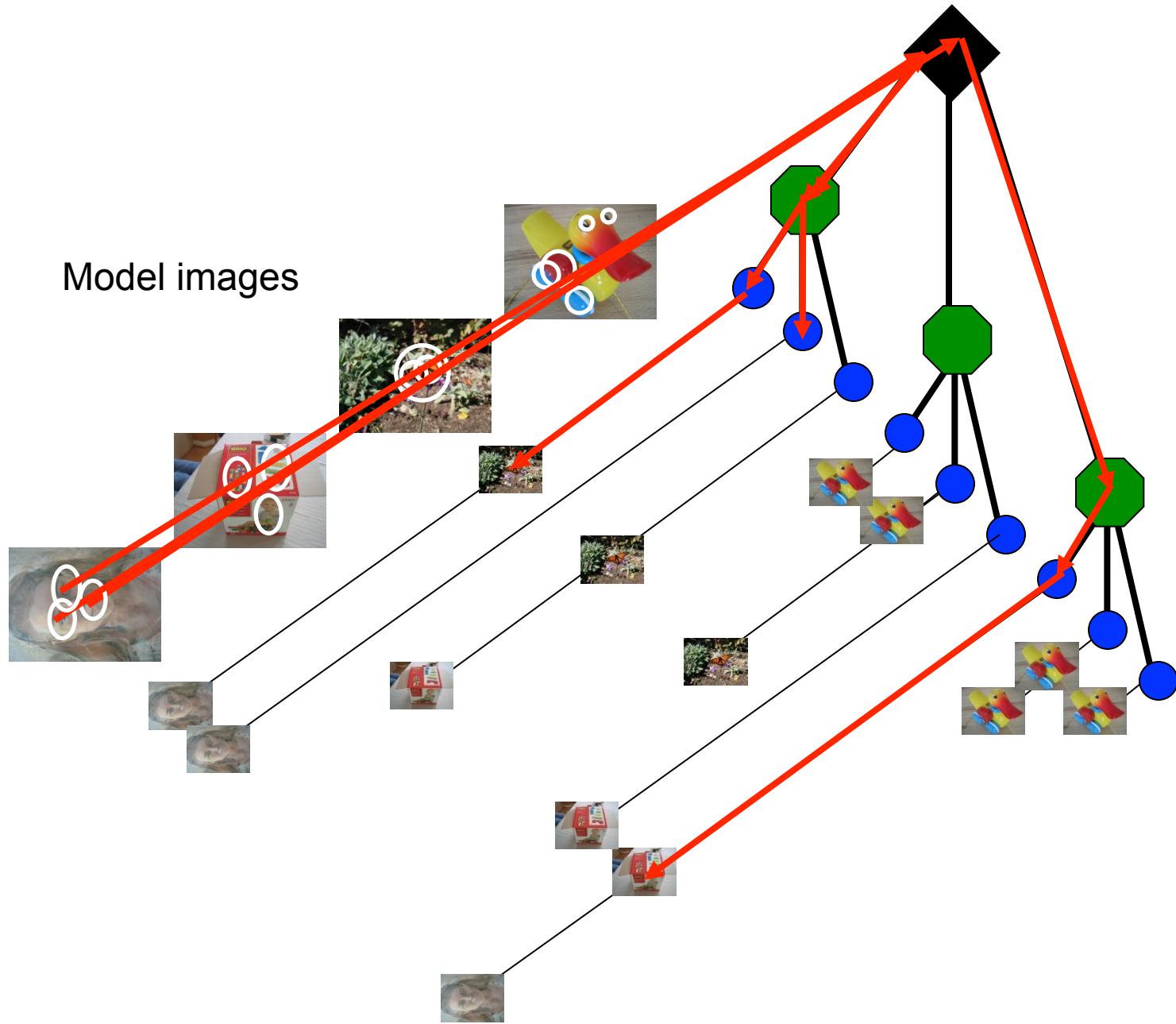
Slide credit: D. Nister

Model images



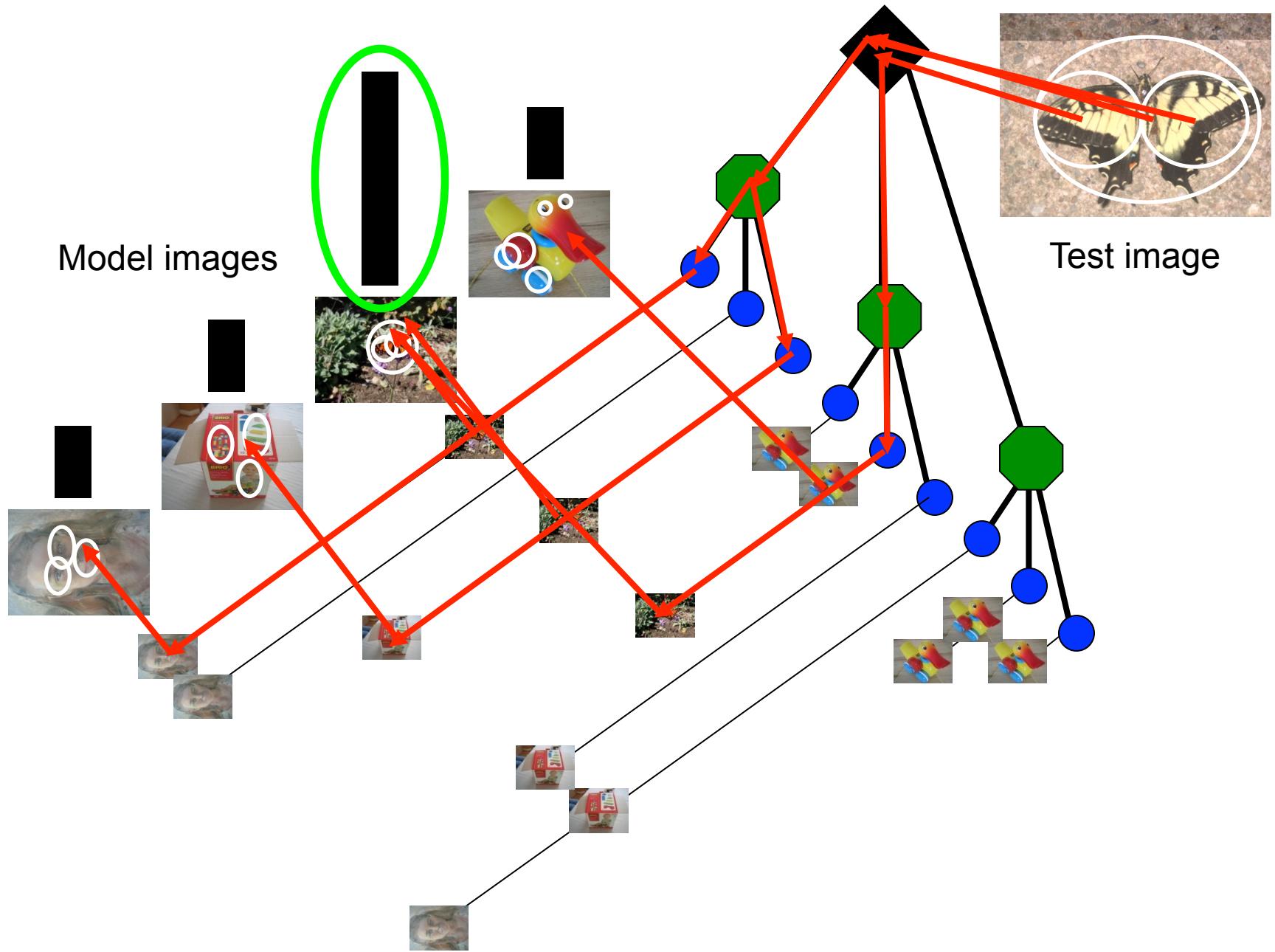
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

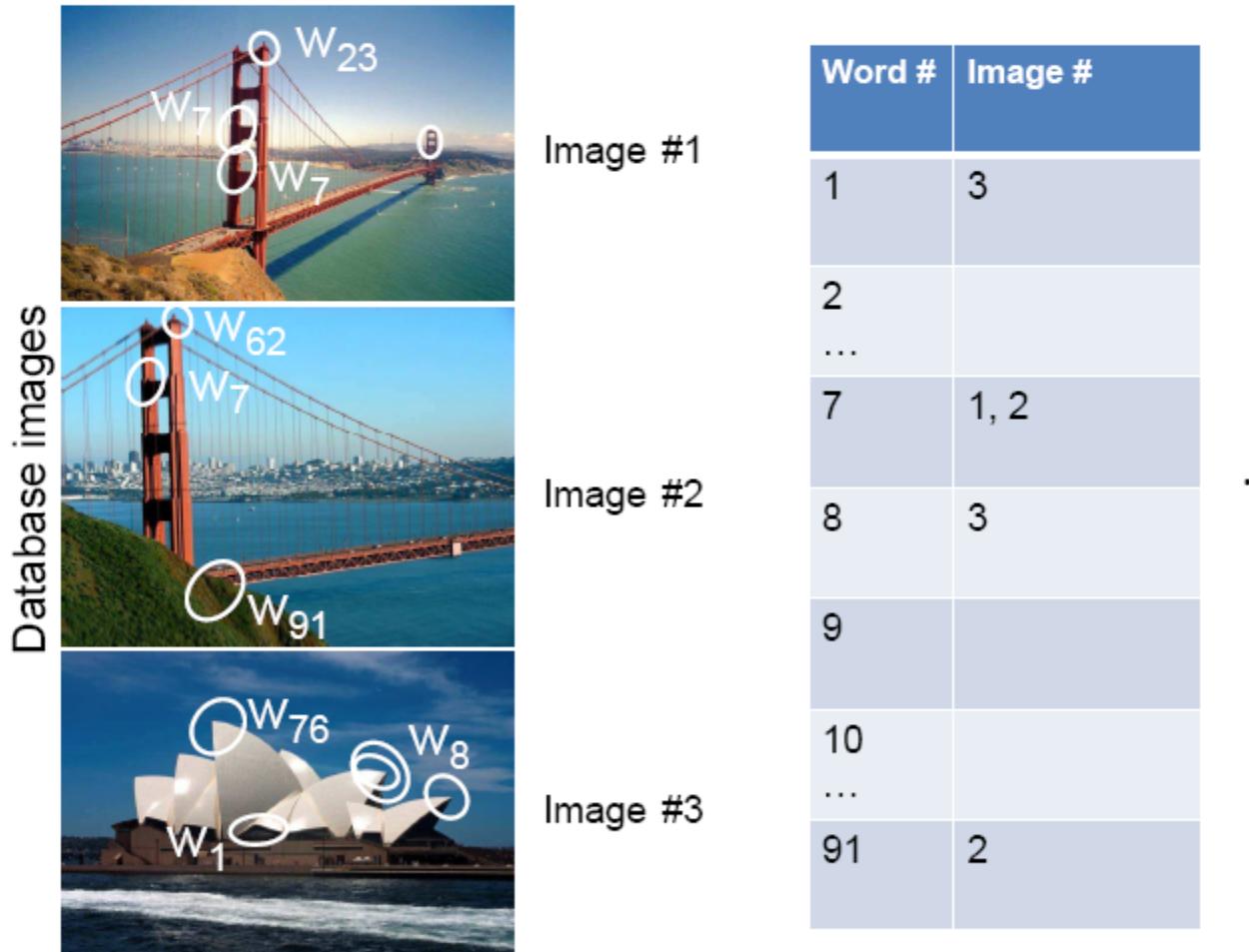
Slide credit: D. Nister



Looking up a test image

Slide credit: D. Nister

Inverted file index



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

Inverted file index



New query image

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



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

Inverted file index

- Key requirement for inverted file index to be efficient: sparsity
- If most pages/images contain most words then you're not 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 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?

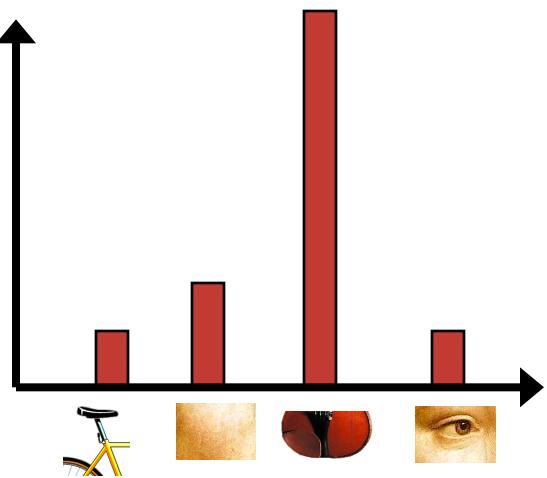
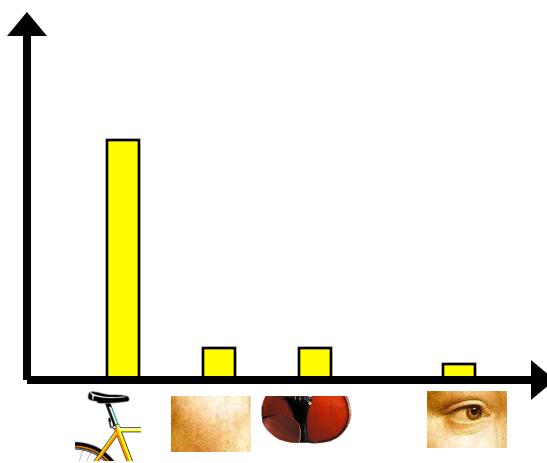
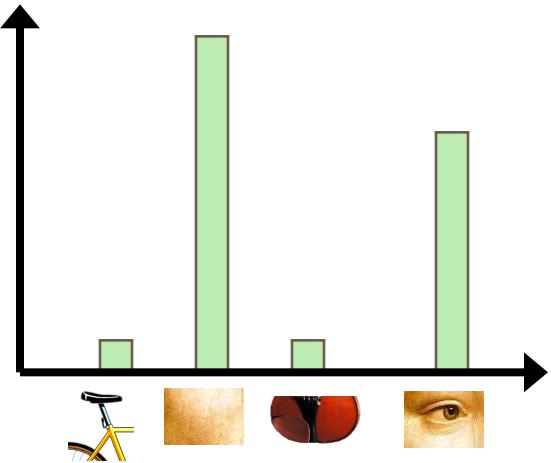
Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach us through our eyes. For a long time it was believed that the retinal image was processed directly in the visual centers in the brain. In 1960, Hubel and Wiesel discovered that the visual system is more complex than previously thought. Following the work of H. G. Külz, who had mapped to the various columns of the primary visual cortex, Hubel and Wiesel have been able to demonstrate that the message about the image falling on the retina undergoes a top-down analysis in a system of nerve cells stored in columns. In this system each column has its specific function and is responsible for a specific detail in the pattern of the retinal image.



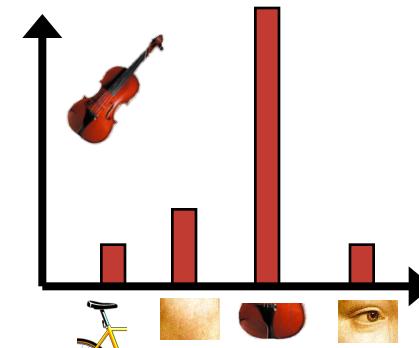
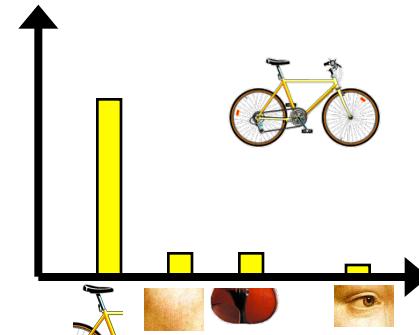
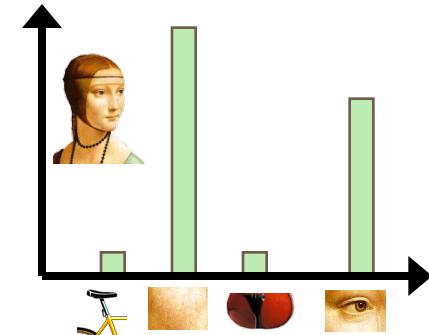
China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$660bn. This will annoy the US, which China's leaders believe deliberately agreed to let the yuan rise. The government also needs to increase domestic demand so that the country can buy more from the country. China has been allowed to let the yuan against the dollar rise slowly and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.





Bags of visual words

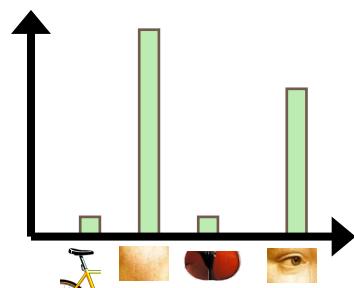
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



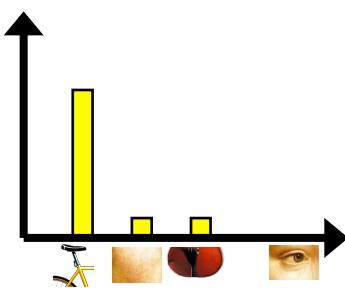
Comparing bags of words

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

[1 8 1 4]



[5 1 1 0]



\vec{d}_j

\vec{q}

$$\begin{aligned} sim(d_j, q) &= \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|} \\ &= \frac{\sum_{i=1}^V d_j(i) * q(i)}{\sqrt{\sum_{i=1}^V d_j(i)^2} * \sqrt{\sum_{i=1}^V q(i)^2}} \end{aligned}$$

for vocabulary of V words

Inverted file index and bags of words similarity



New query image

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

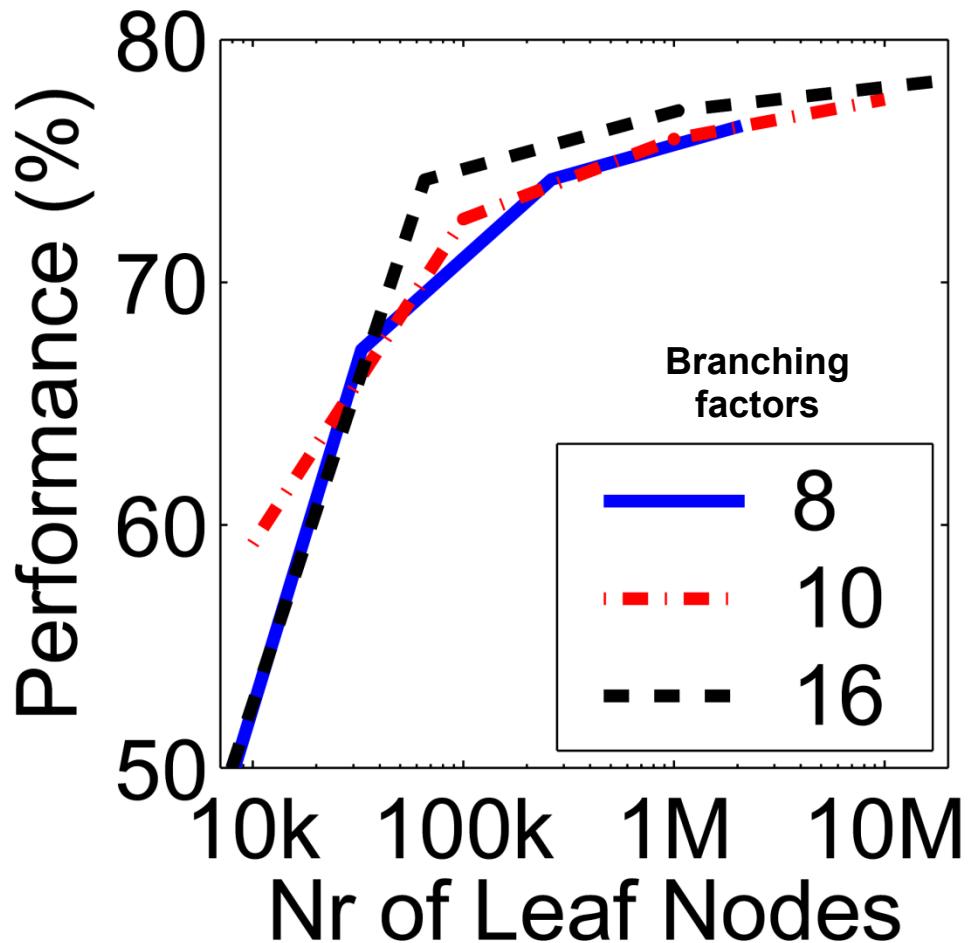


1. Extract words in query
2. Inverted file index to find relevant frames
3. Compare word counts

Instance recognition: remaining issues

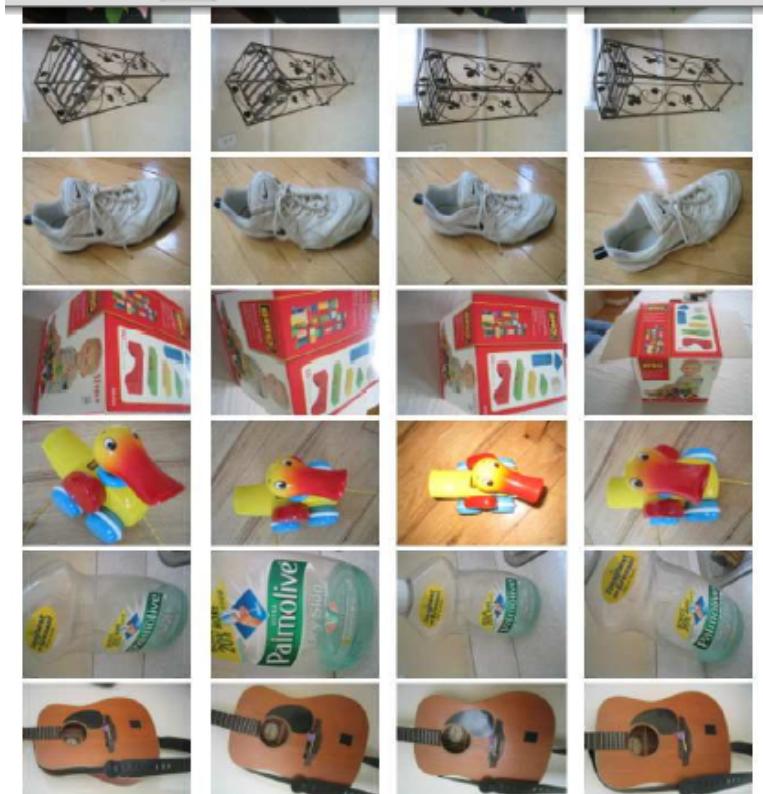
- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size



Influence on performance, sparsity

Results for recognition task
with 6347 images



Nister & Stewenius, CVPR 2006
Kristen Grauman

Vocabulary trees: complexity

Number of words given tree parameters: branching factor and number of levels

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

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.

110,000,000
Images in
5.8 Seconds

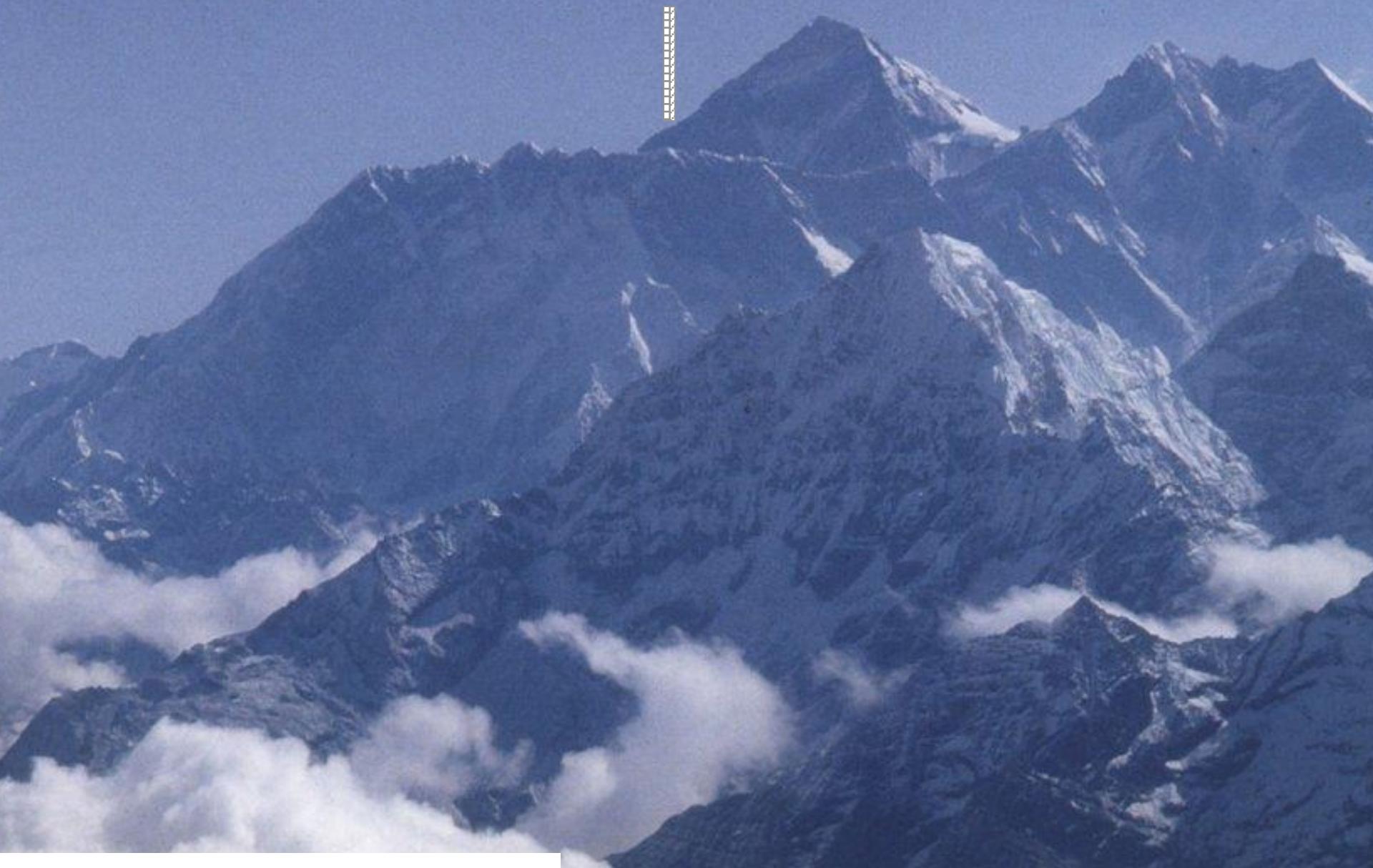


Slide Credit: Nister



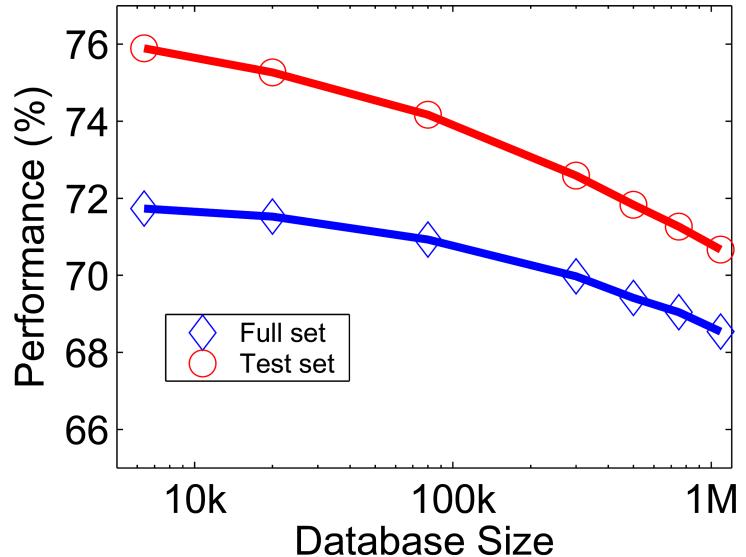
Slide Credit: Nister





Slide Credit: Nister

Performance



ImageSearch at the VizCentre

New query: Browse... Send File

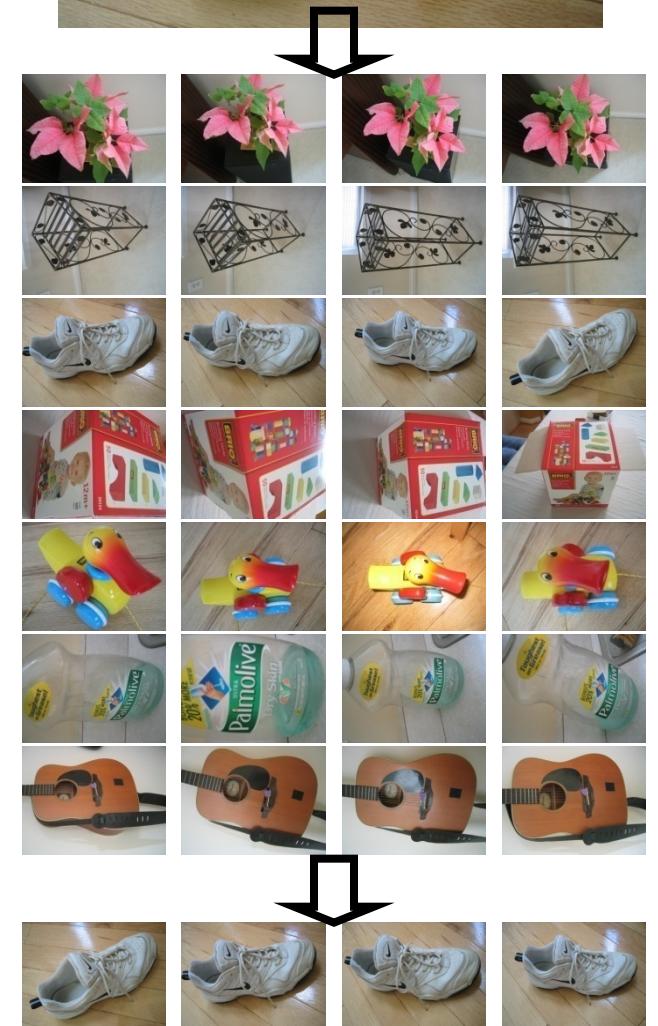
File is 500x320



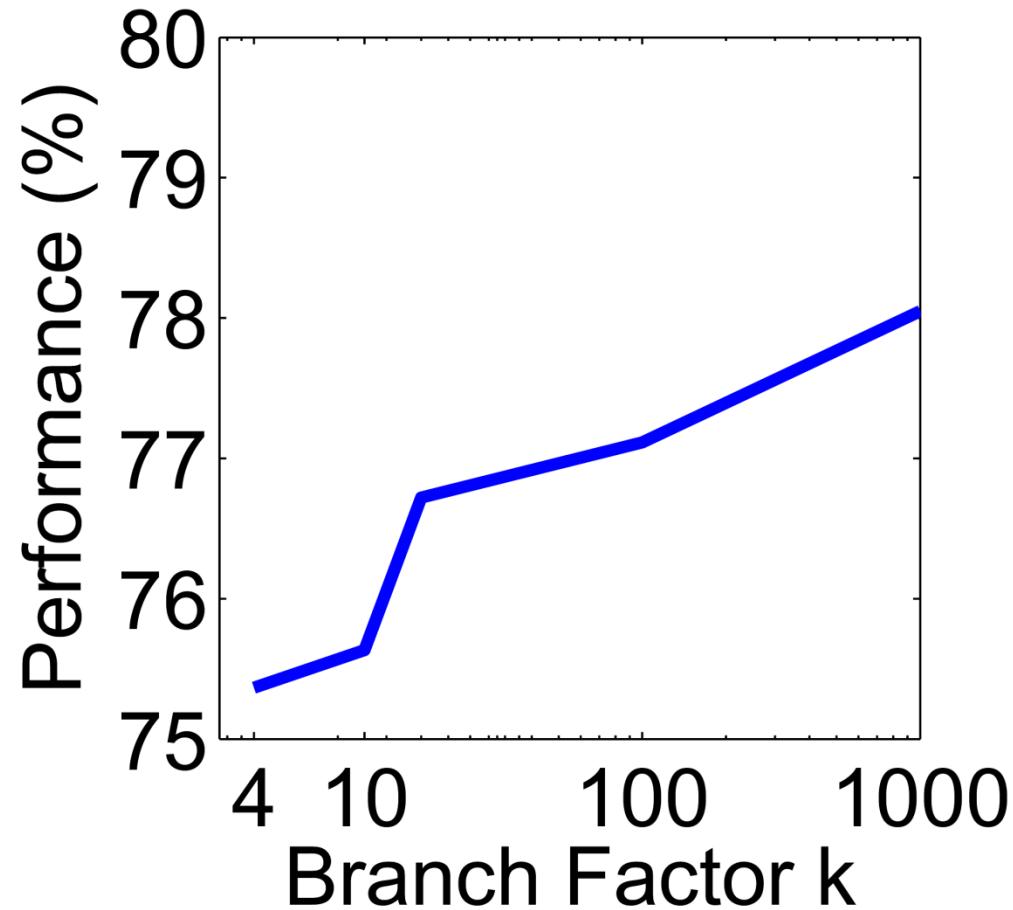
Top n results of your query.



bourne/im1000043322.pgm bourne/im1000043323.pgm bourne/im1000043326.pgm bourne/im1000043327.pgm



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

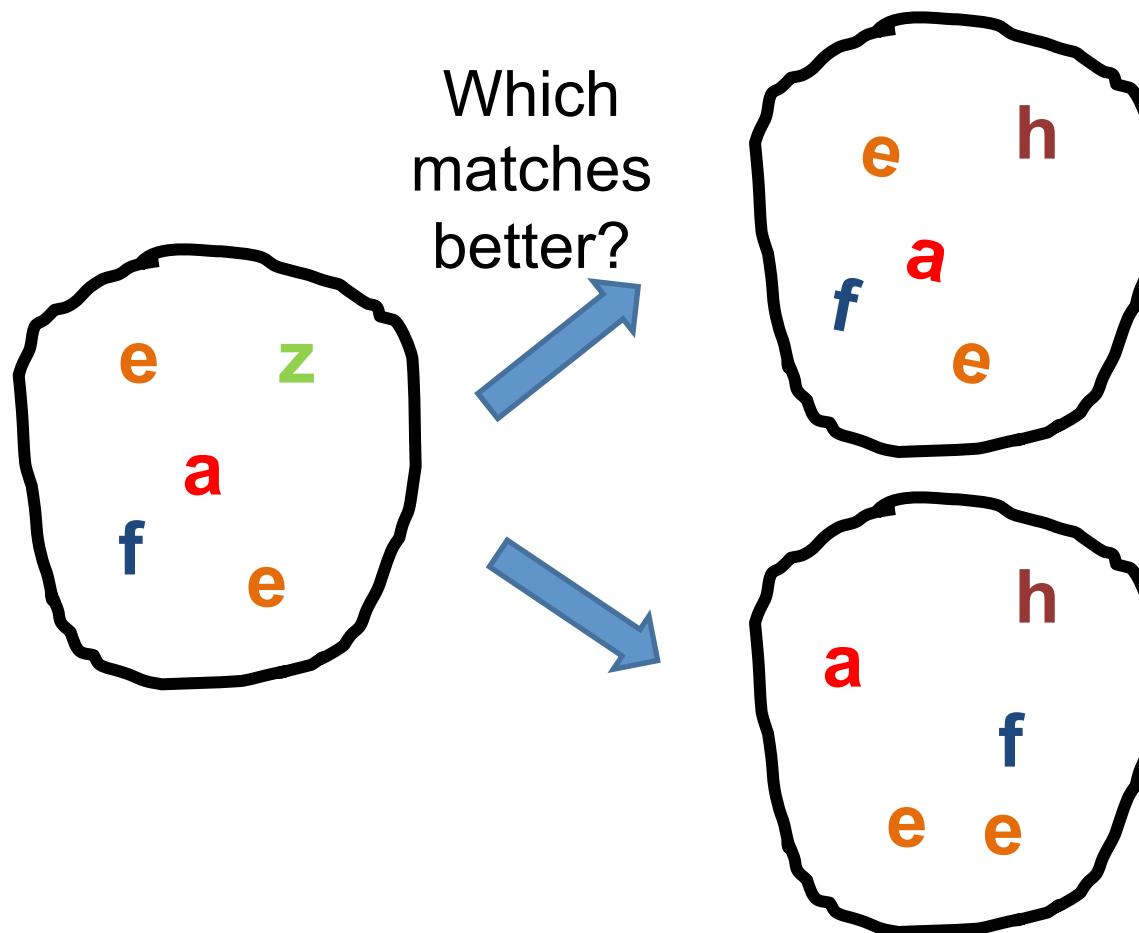
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

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?

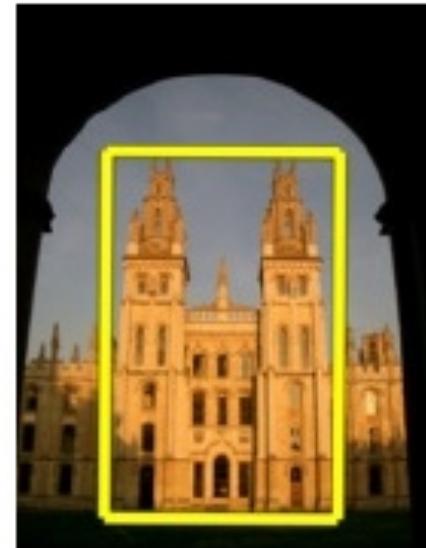
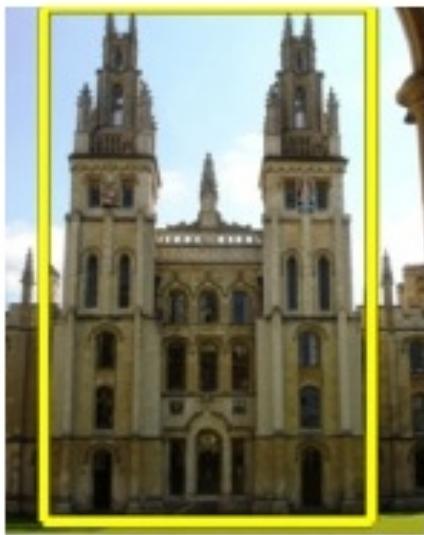
Can we be more accurate?

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

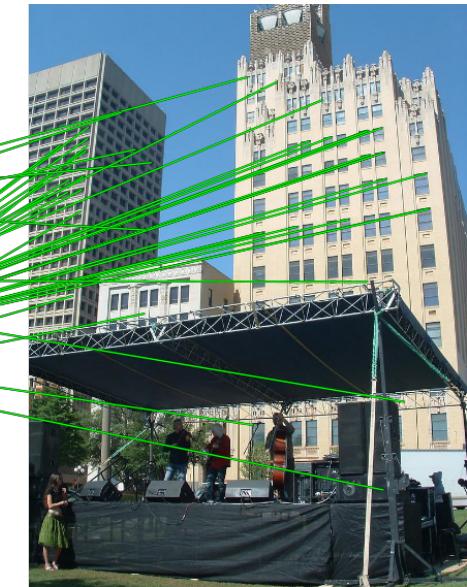
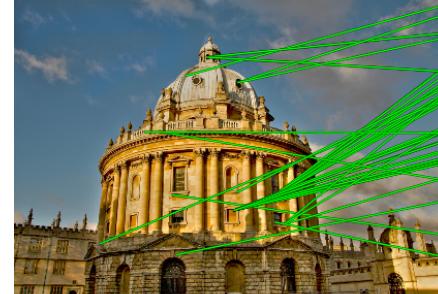
Spatial Verification

Query



DB image with high BoW
similarity

Query

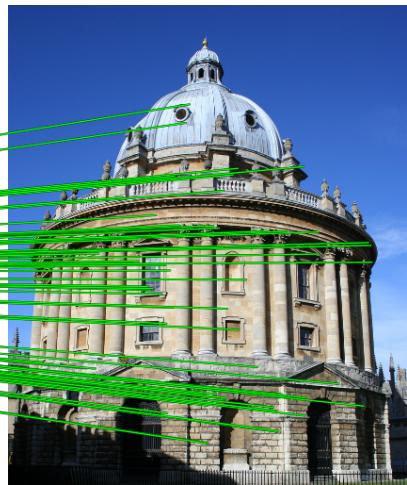
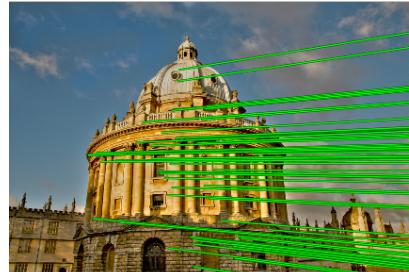


DB image with high BoW
similarity

Both image pairs have many visual words in common.

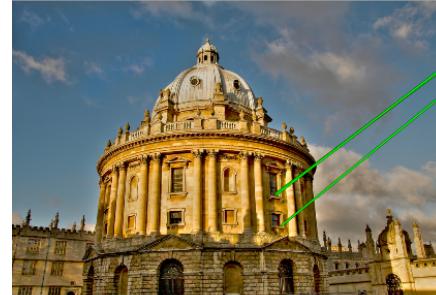
Spatial Verification

Query



DB image with high BoW
similarity

Query



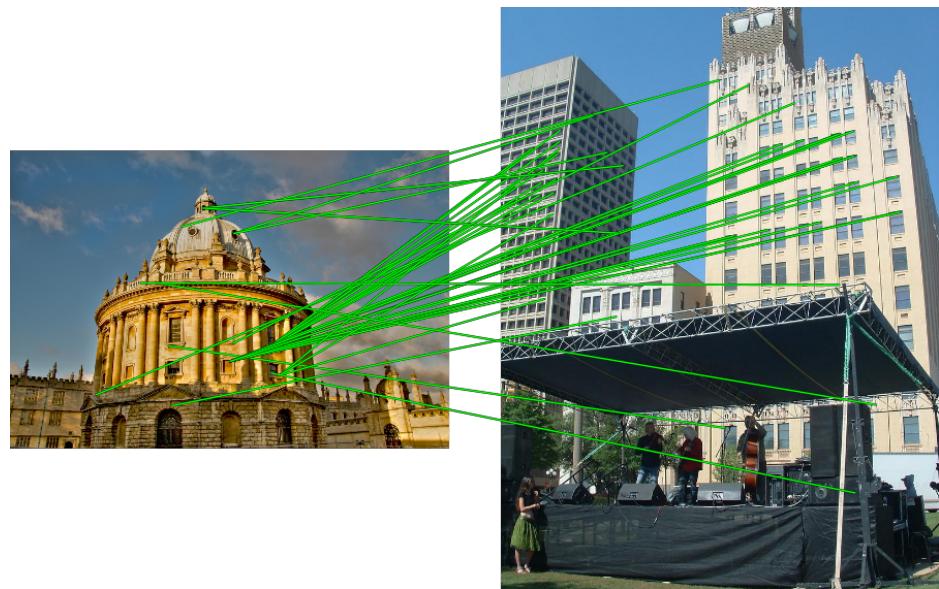
DB image with high BoW
similarity

Only some of the matches are mutually consistent

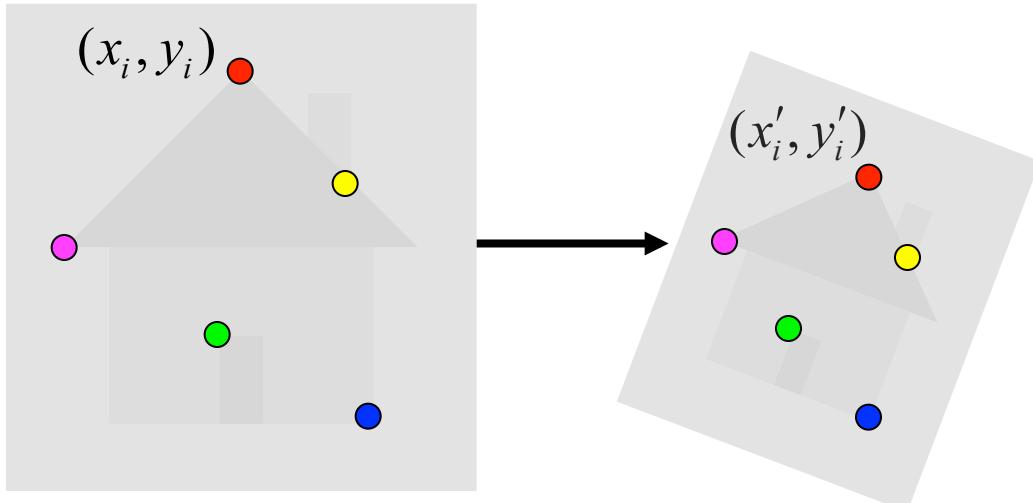
Spatial Verification: two basic strategies

- 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

RANSAC verification



Recall: Fitting an affine transformation

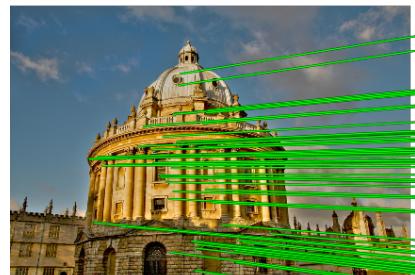
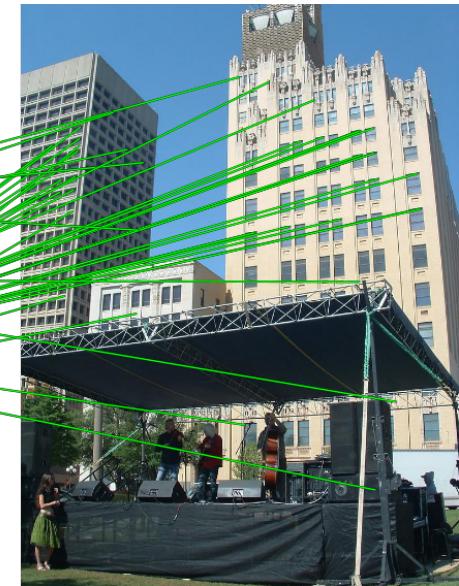
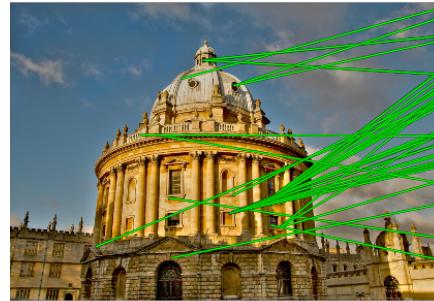


Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \dots & & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ \dots \end{bmatrix}$$

RANSAC verification



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?

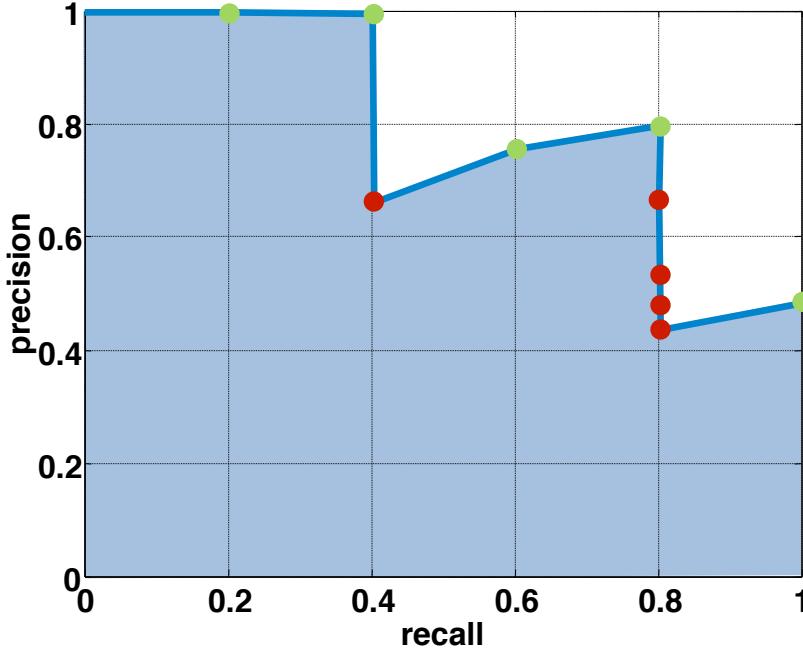
Scoring retrieval quality



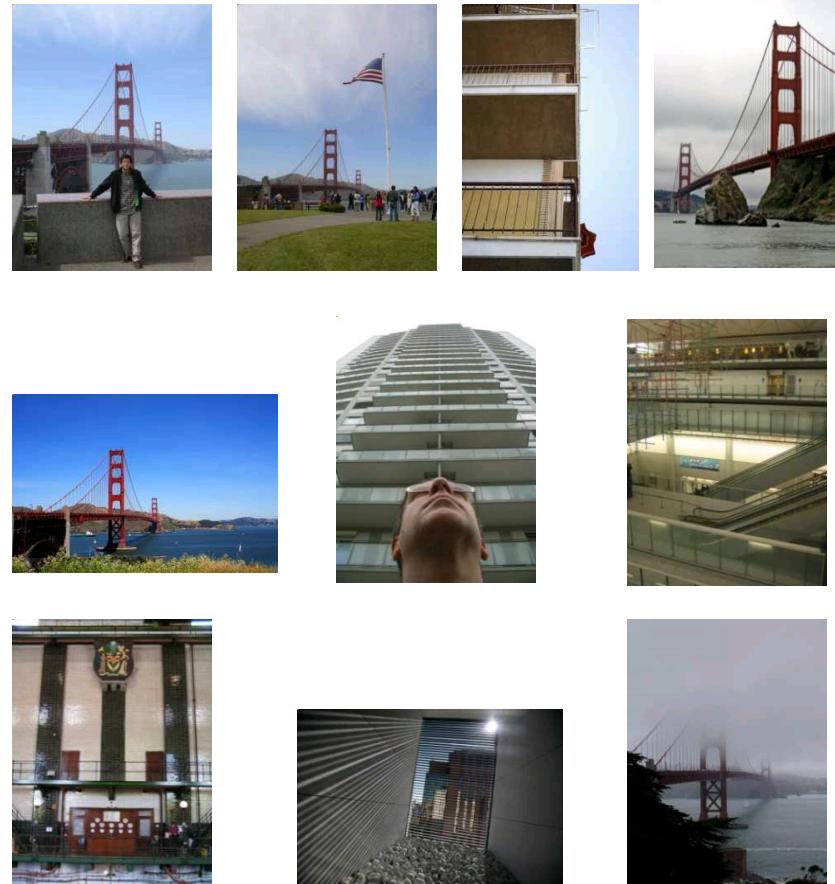
Query

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

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



Results (ordered):



What else can we borrow from text retrieval?

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 MacIay 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
Anhala; 109-109,146
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
Butterfly Center, McGuire; 134
CAA (see AAA)
CCC, The; 111,113,115,135,142
Ca d'Zan; 147
Caloosahatchee River; 152
Name; 150
Canaveral Natnl Seashore; 173
Cannon Creek Airpark; 130
Canopy Road; 106,169
Cape Canaveral; 174
Castillo San Marcos; 169
Cave Diving; 131
Cayo Costa, Name; 150
Celebration; 93
Charlotte County; 149
Charlotte Harbor; 150
Chautauqua; 116
Chipley; 114
Name; 115
Choctawhatchee, Name; 115
Circus Museum, Ringling; 147
Citrus; 88,97,130,136,140,180
CityPlace, W Palm Beach; 180
City Maps,
 Ft Lauderdale Expwy; 194-195
Jacksonville; 163
Kissimmee Expwy; 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
Driving Lanes; 85
Duval County; 163
Eau Gallie; 175
Edison, Thomas; 152
Eglin AFB; 116-118
Eight Reale; 176
Ellenton; 144-145
Emanuel Point Wreck; 120
Emergency Callboxes; 83
Epiphytes; 142,148,157,159
Escambia Bay; 119
Bridge (I-10); 119
County; 120
Estero; 153
Everglade, 90,95,139-140,154-160
Draining of; 156,181
Wildlife MA; 160
Wonder Gardens; 154
Falling Waters SP; 115
Fantasy of Flight; 95
Fayer Dykes SP; 171
Fires, Forest; 166
Fires, Prescribed; 148
Fisherman's Village; 151
Flagler County; 171
Flagler, Henry; 97,165,167,171
Florida Aquarium; 186
Florida,
 12,000 years ago; 187
Cavern SP; 114
Map of all Expressways; 2-3
Mus of Natural History; 134
National Cemetery; 141
Part of Africa; 177
Platform; 187
Sheriff's Boys Camp; 126
Sports Hall of Fame; 130
Sun 'n Fun Museum; 97
Supreme Court; 107
Florida's Turnpike (FTP), 178,189
25 mile Strip Maps; 66
Administration; 189
Coin System; 190
Exit Services; 189
HEFT; 76,161,190
History; 189
Names; 189
Service Plazas; 190
Spur SR91; 76

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% growth in exports to \$750bn, compared with \$660bn. The US has been annoyed that China's exports have been growing so rapidly, deliberately agreed to let the yuan rise against the dollar, and agreed to increase its imports from China. The Chinese government also needs to encourage domestic demand so it can sell more to the rest of the world.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% growth in exports to \$750bn, compared with \$660bn. The US has been annoyed that China's exports have been growing so rapidly, deliberately agreed to let the yuan rise against the dollar, and agreed to increase its imports from China. The Chinese government also needs to encourage domestic demand so it can sell more to the rest of the world. The Chinese government has been trying to control the yuan against the dollar, but the US wants the yuan to be allowed to float freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.

tf-idf weighting

- Term frequency – inverse document frequency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Number of occurrences of word i in document d

Number of words in document d

Total number of documents in database

Number of documents word i occurs in, in whole database

Recognition via alignment

Pros:

- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

Cons:

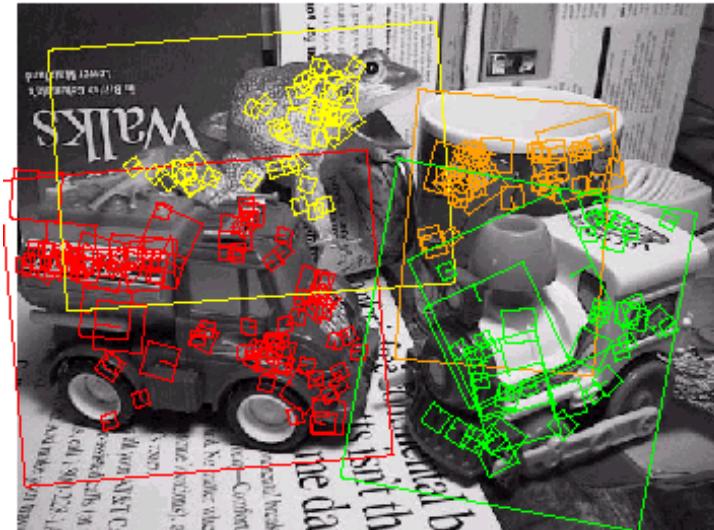
- Spatial verification as post-processing – not seamless, expensive for large-scale problems
- Not suited for category recognition.

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

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

