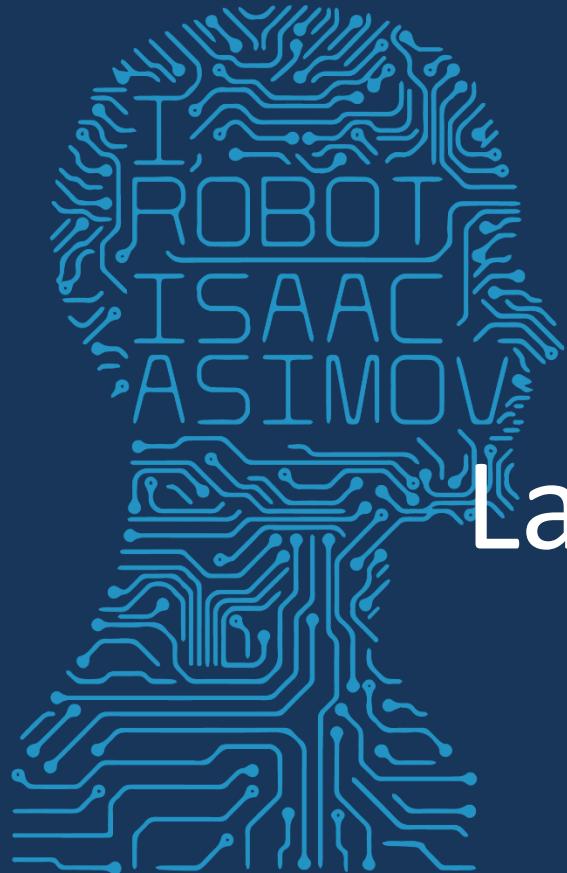


Advanced Computer Vision



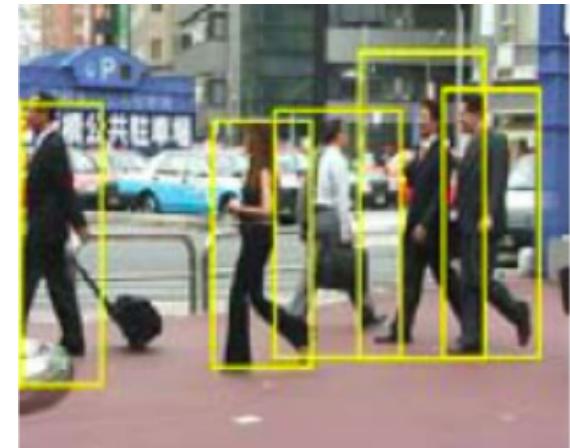
FUTURE VISION

Large-scale Instance
Recognition

Category vs. instance recognition

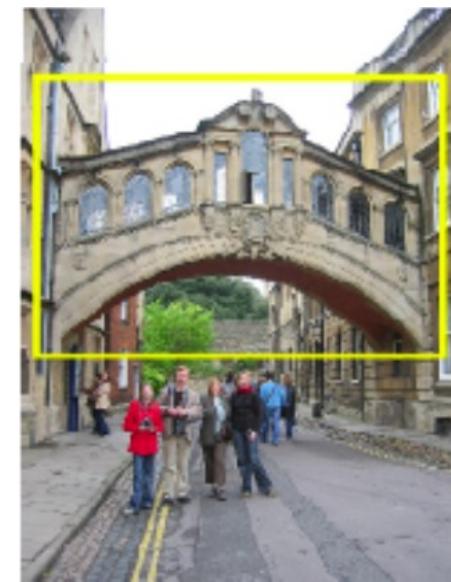
Category:

- Find all the people
- Find all the buildings
- Often within a single image
- Often ‘sliding window’

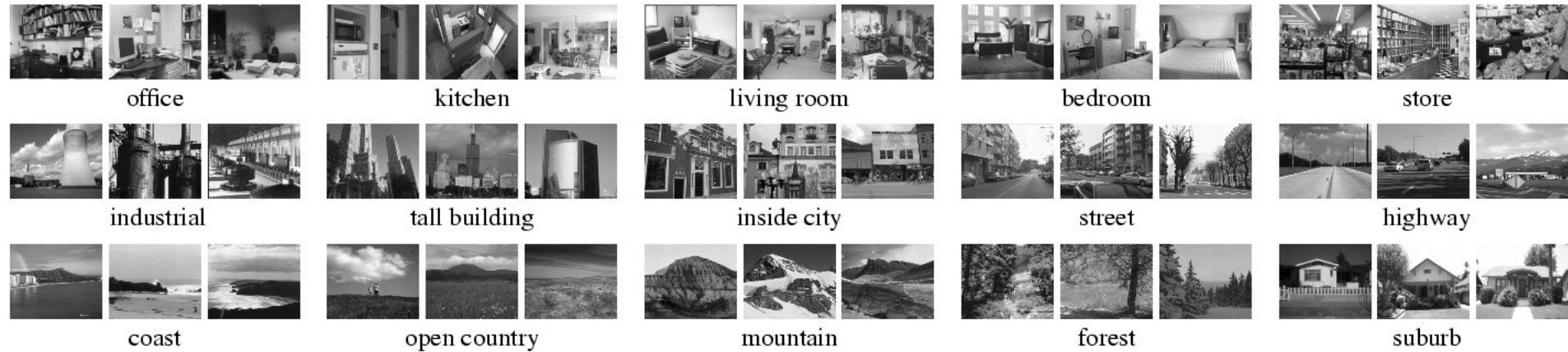


Instance:

- Is this face James?
- Find this specific famous building
- Often within a database of images

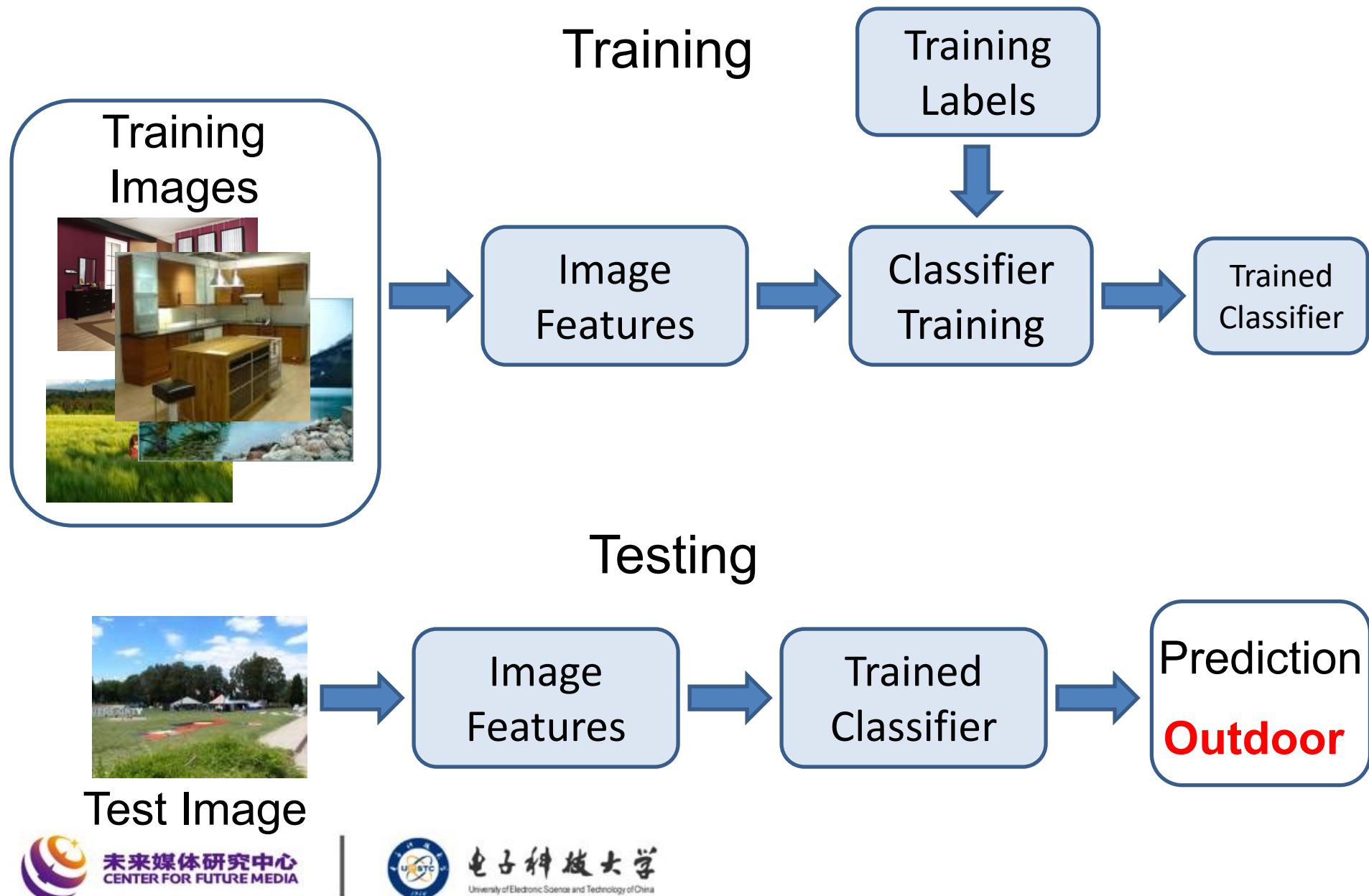


Scene recognition dataset



Instance or category?

Recognition



Recognition Issues

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

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

How to score the retrieval results?

How might we add more spatial verification?

Recognition Issues

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

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

How to score the retrieval results?

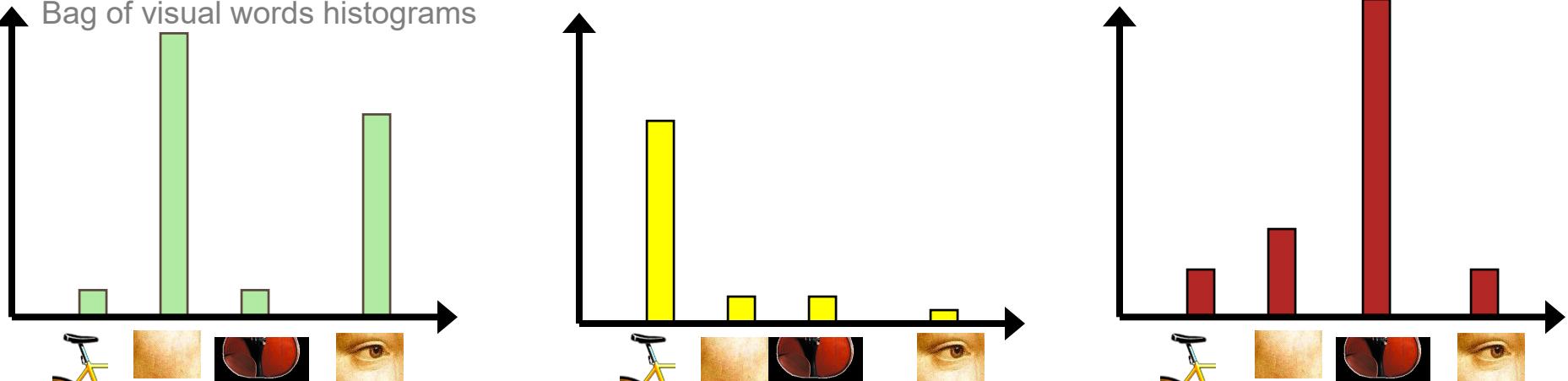
How might we add more spatial verification?



Visual words



Bag of visual words histograms

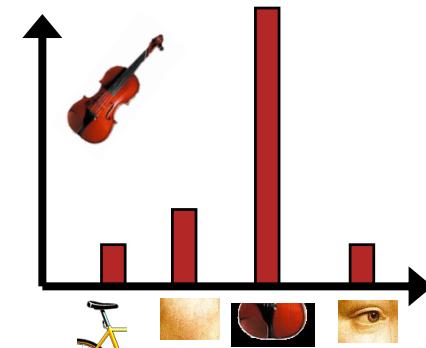
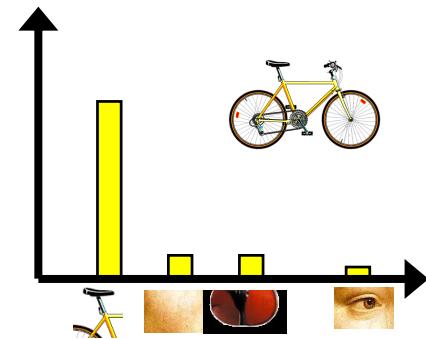
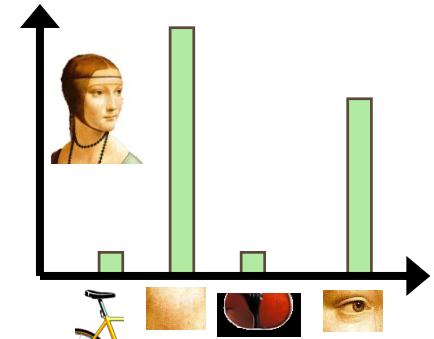


Bags of visual words

Bag of visual words histograms

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

Visual words

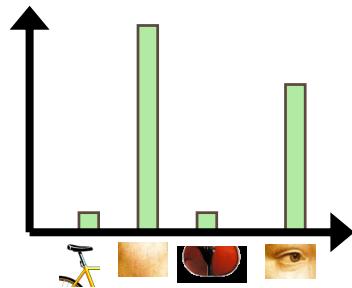


Comparing bags of words

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

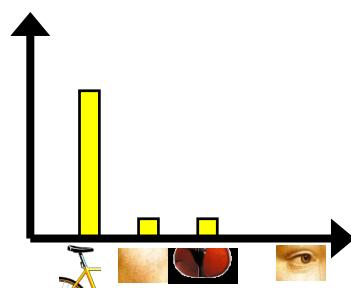
Database image

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



Query

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



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

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

for vocabulary of V words

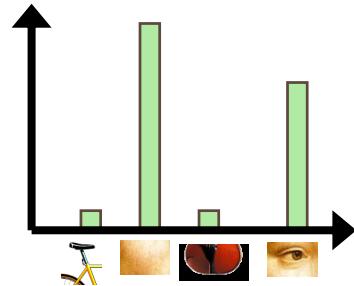
Comparing bags of words

Why might we use cosine similarity here?

What ‘intuitive’ effect does this provide?

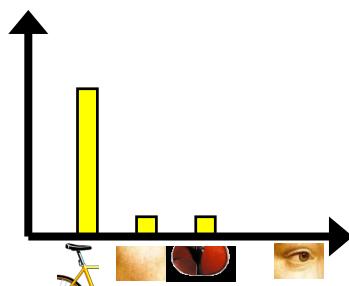
Database image

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



Query

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



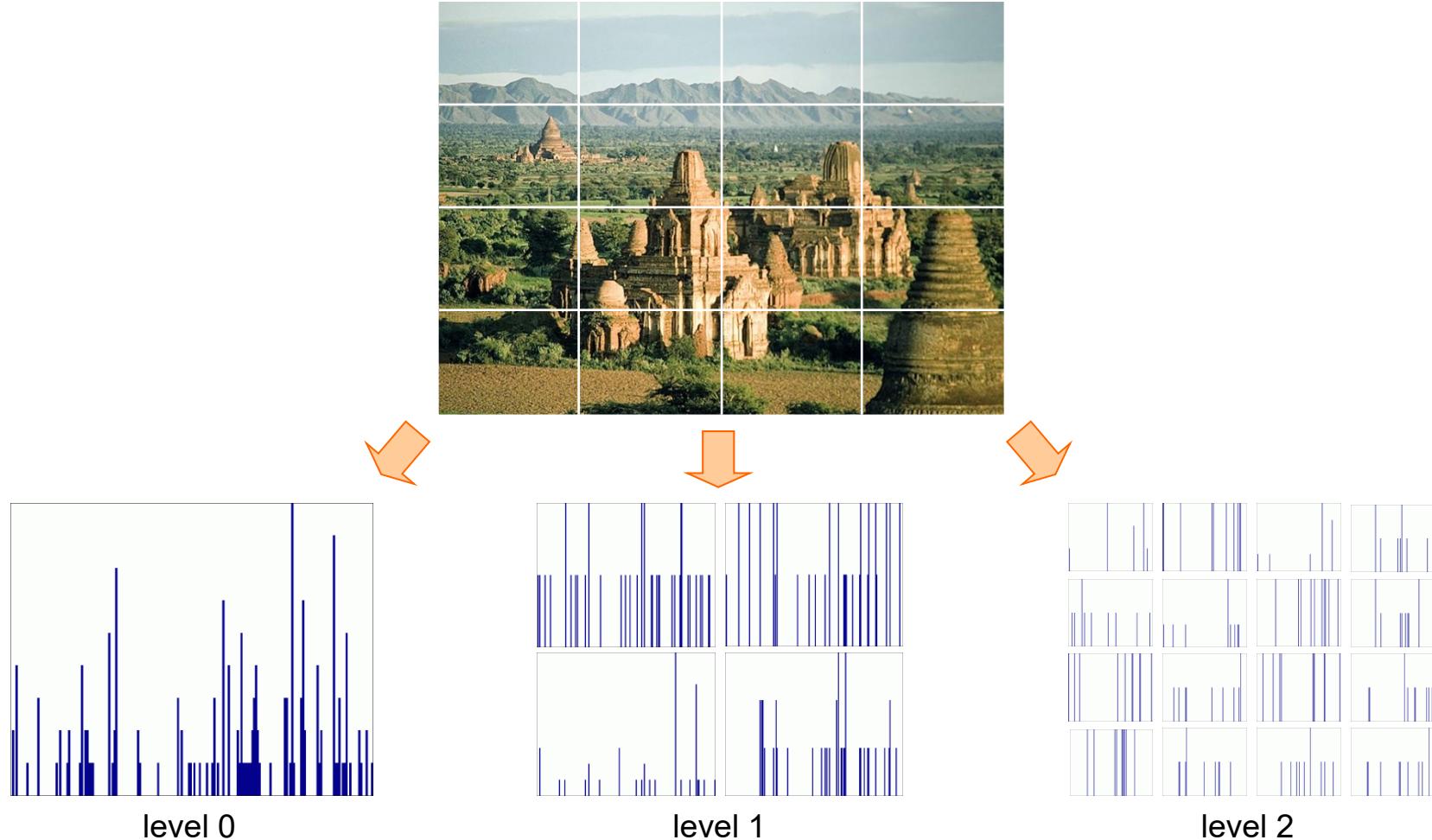
$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

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

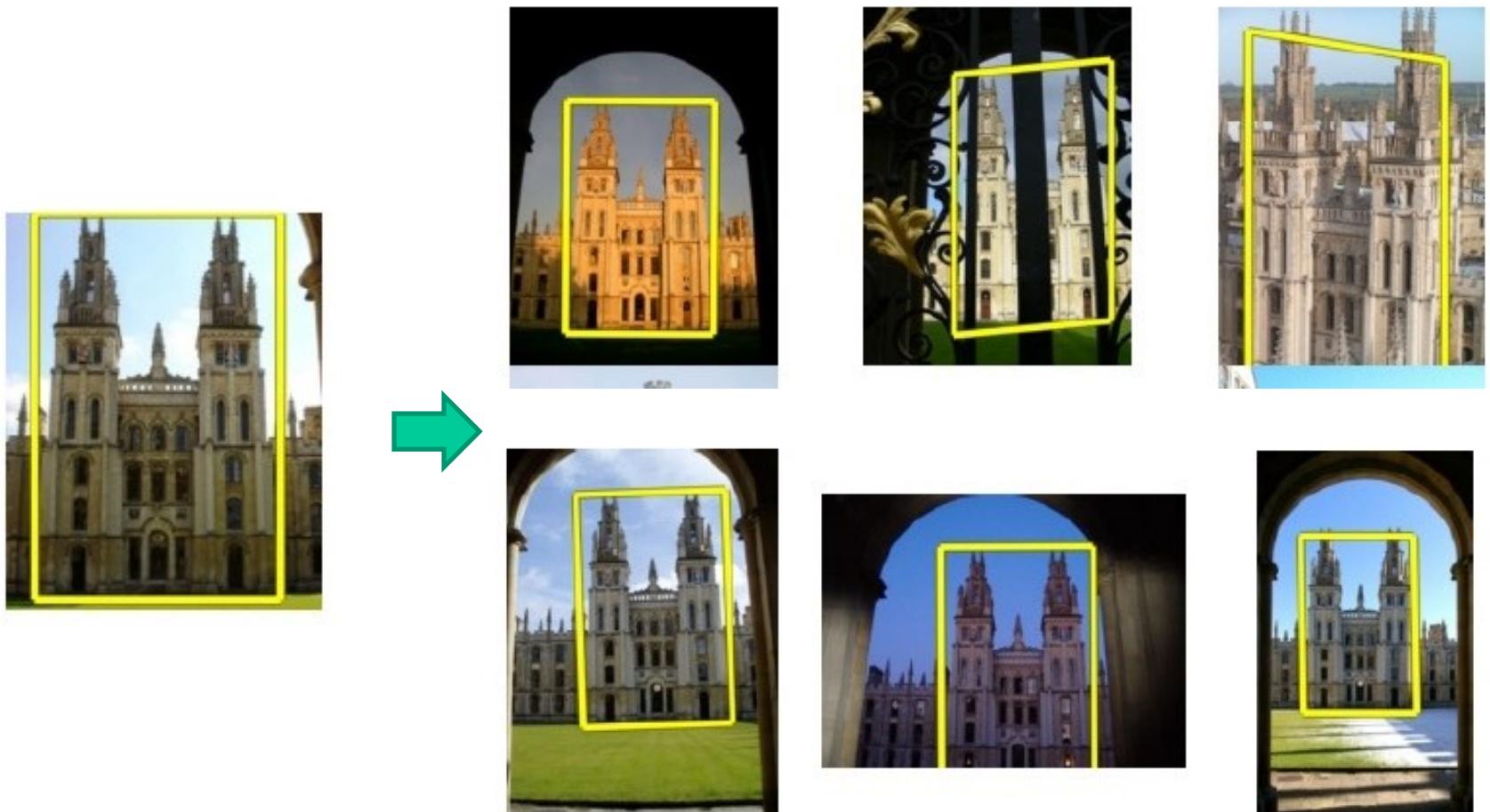
for vocabulary of V words

Spatial pyramid representation

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

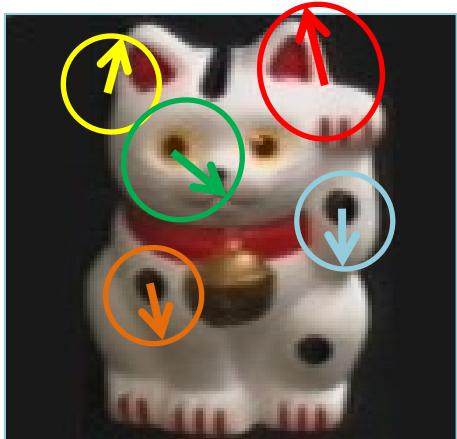


How can we quickly find images in a large database that match a given image region?

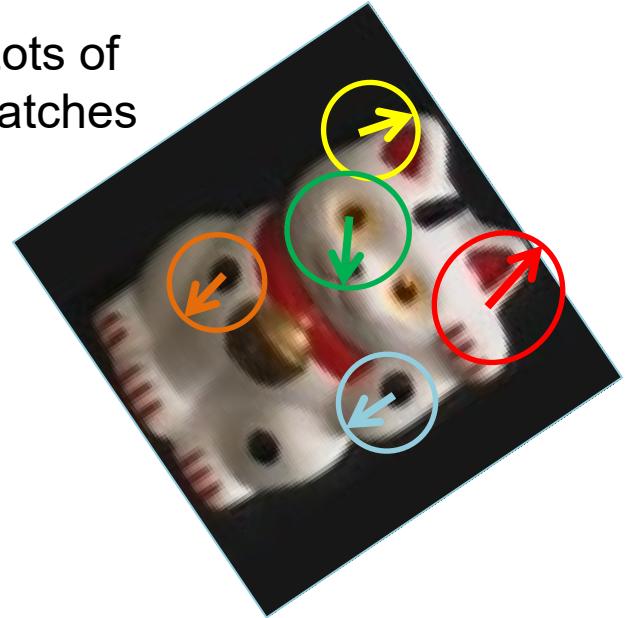


Simple idea

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



Lots of
Matches



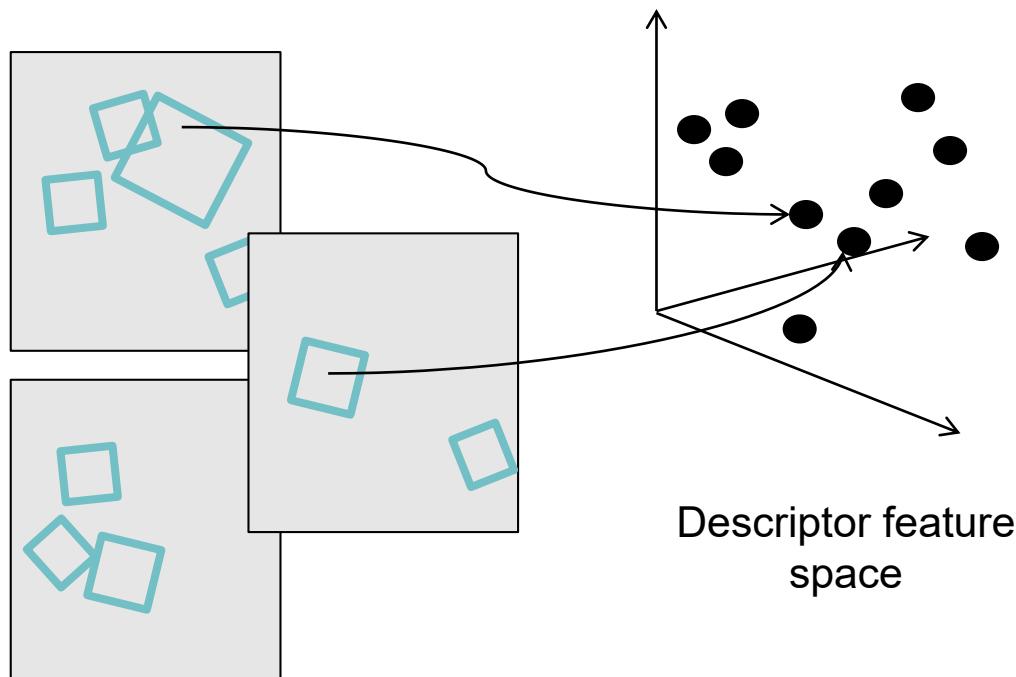
Few or No
Matches



But this will be really, really slow!

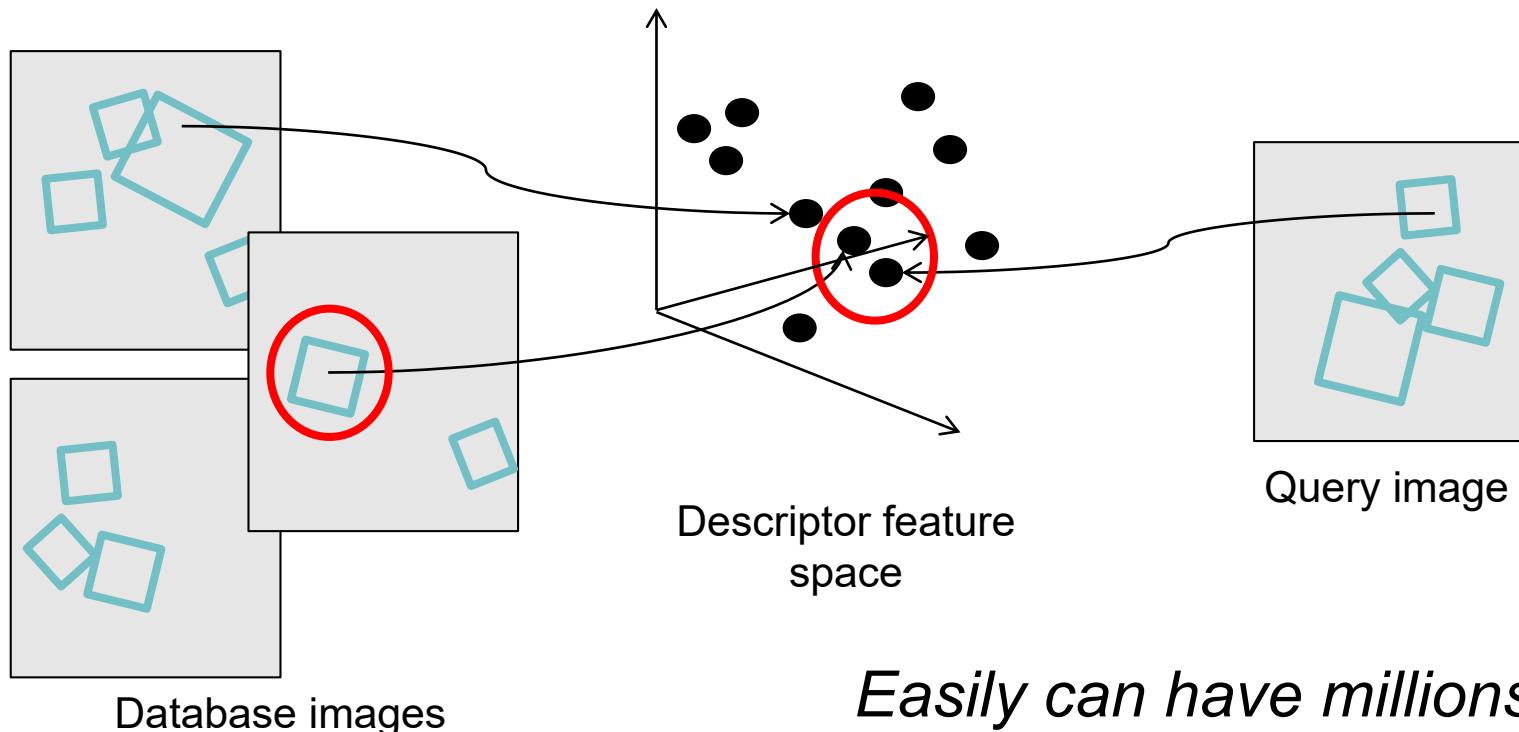
Indexing local features

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



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!



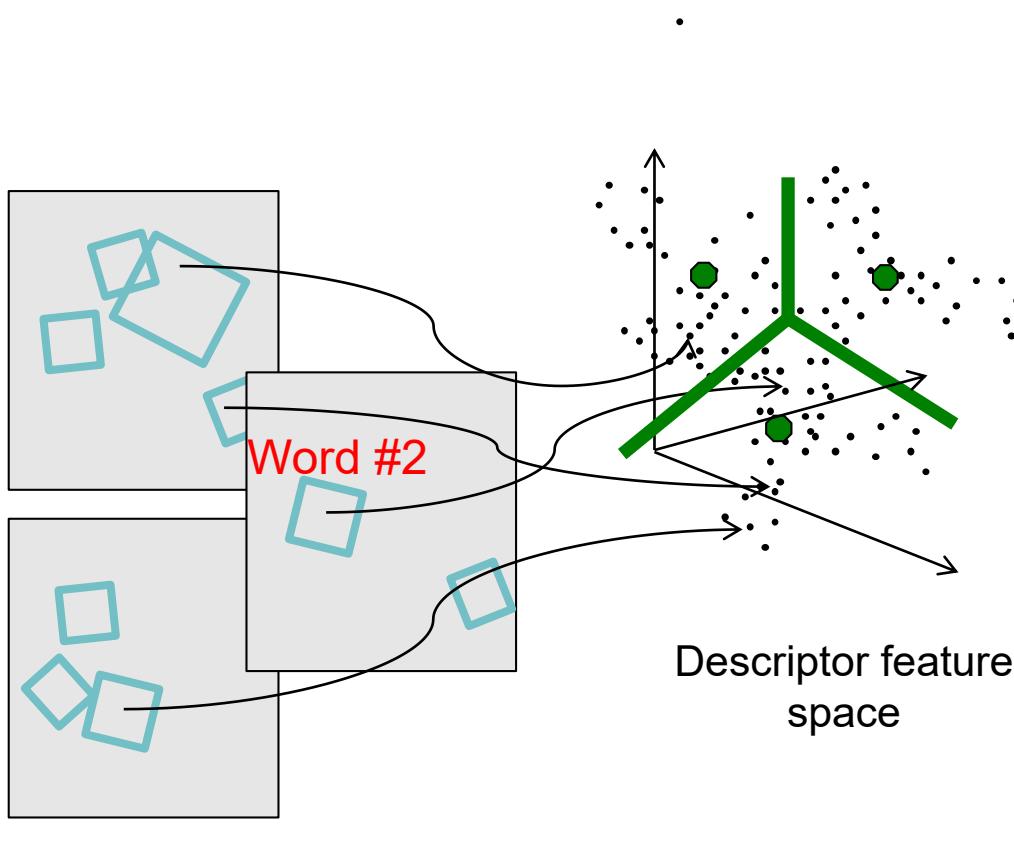
未来媒体研究中心
CENTER FOR FUTURE MEDIA



电子科技大学
University of Electronic Science and Technology of China

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

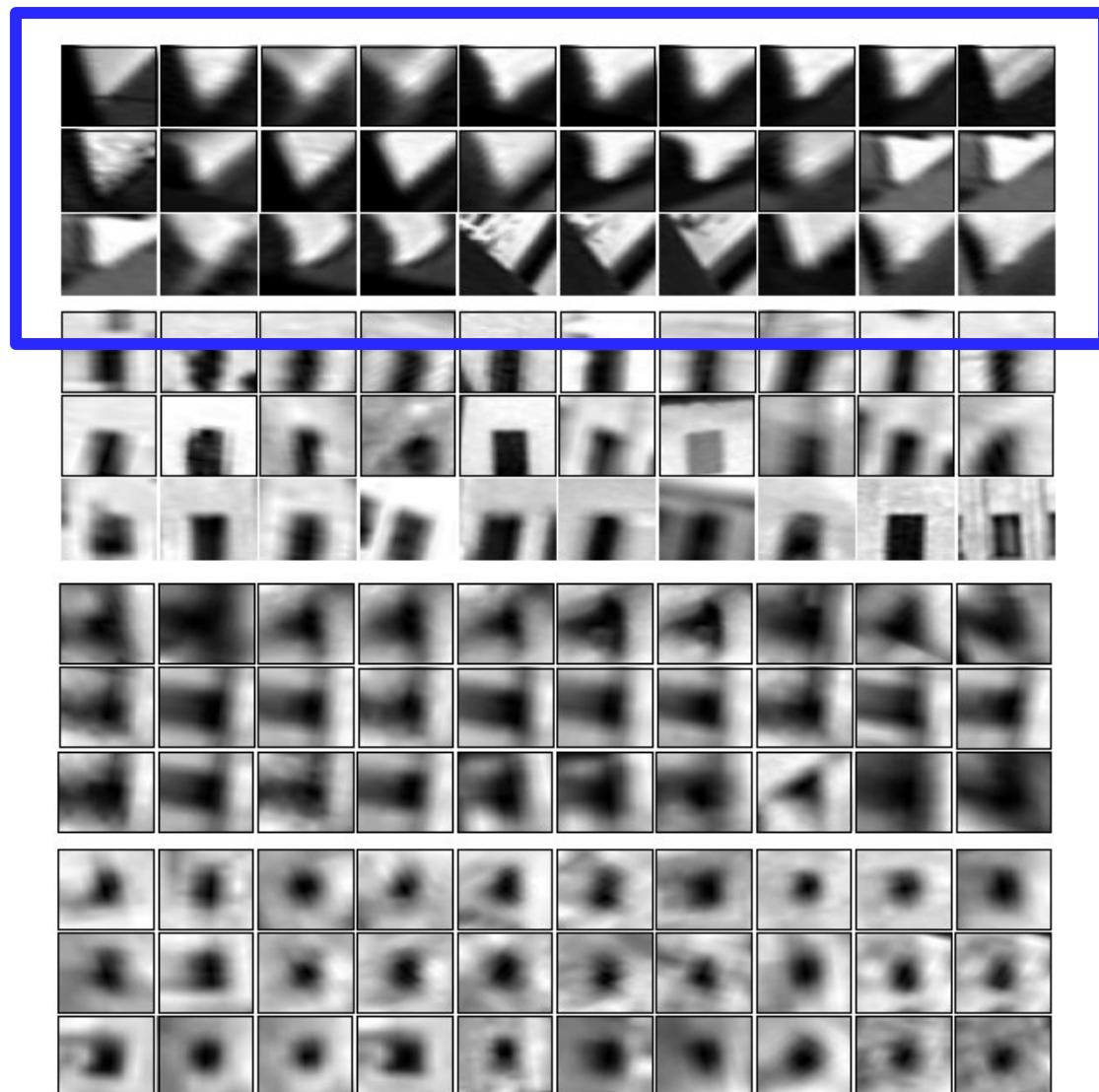
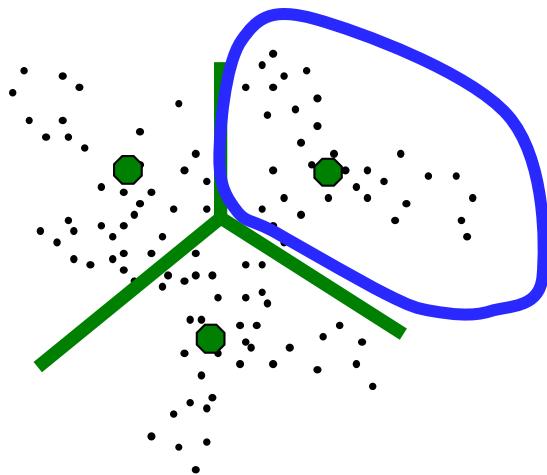
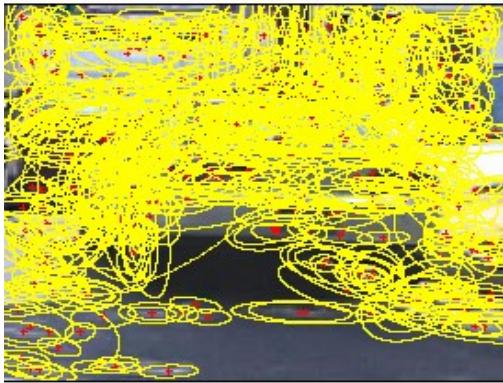
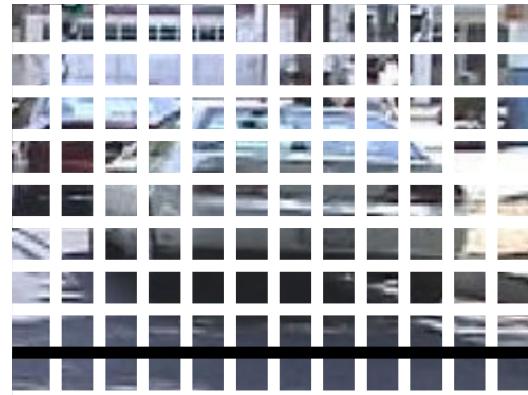


Figure from Sivic & Zisserman, ICCV 2003

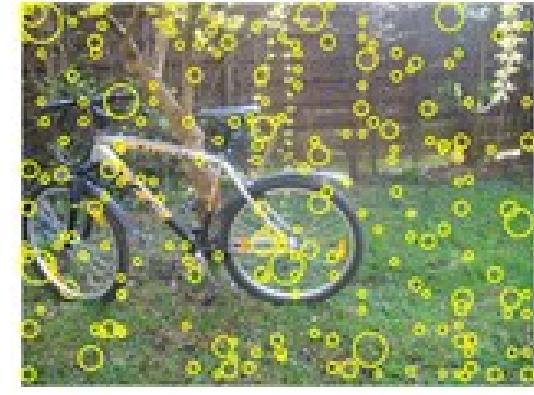
Sampling strategies



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]



未来媒体研究中心
CENTER FOR FUTURE MEDIA



电子科技大学
University of Electronic Science and Technology of China

Image credits: F-F. Li, E. Nowak, J. Sivic
K. Grauman, B. Leibe

Fast lookup: inverted 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
Alefia 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 Mallmen; 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
 A4-C Skyhawk; 117
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; 106,160
Cape Canaveral; 174
Castillo San Marcos; 169
Cave Diving; 131
Cayo Costa, Name; 150
Celebration; 93
Charlotte County; 149
Charlotte Harbor; 150
Chautauqua; 116
Chipley; 114
 Name; 115
Choctawatchee, Name; 115
Circus Museum, Ringling; 147
Citrus; 88,97,130,136,140,180
CityPlace, W Palm Beach; 180
City Maps,
 Ft Lauderdale Expwys; 194-195
 Jacksonville; 163
 Kissimmee 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
Daniel Boone, Florida Walk; 117
Daytona Beach; 172-173
De Land; 87
De Soto, Hernando,
 Name; 150
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
Ticket System; 190
Toll Plazas; 190
Ford, Henry; 152
Fort Barrancas; 122

- 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.



Build Inverted Index from Database

Database images

The figure shows three images from a database. The first image is of the Golden Gate Bridge with words W₇, W₂₃, and W₇ circled. The second image is of the Golden Gate Bridge with words W₆₂, W₇, and W₉₁ circled. The third image is of the Sydney Opera House with words W₁, W₇₆, and W₈ circled. Ellipses below the images indicate there are more images in the database.

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

Image #1 Image #2 Image #3

⋮ ⋮ ⋮

Query Inverted Index



New query image

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

Candidate matches



Image #1



Image #2

Query Inverted 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/sort word counts

Inverted 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.

Recognition Issues

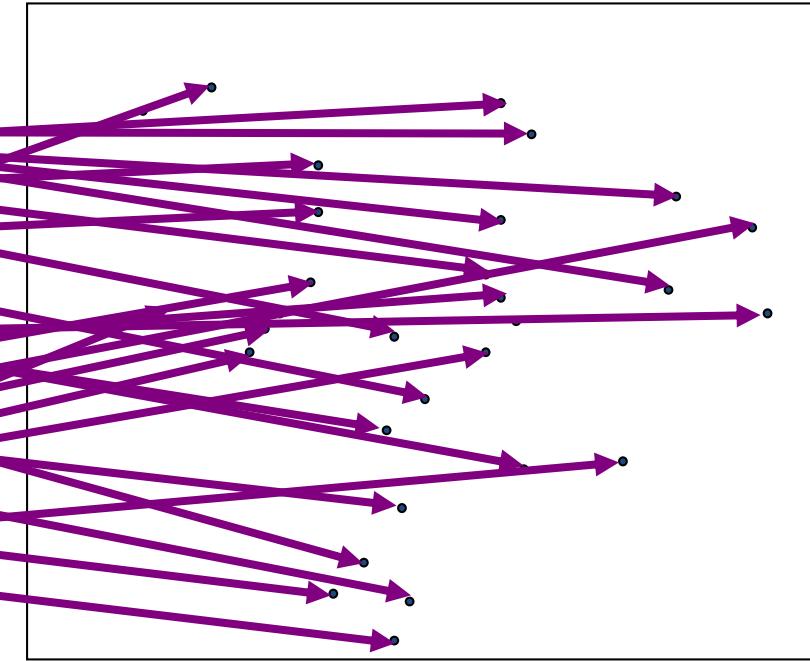
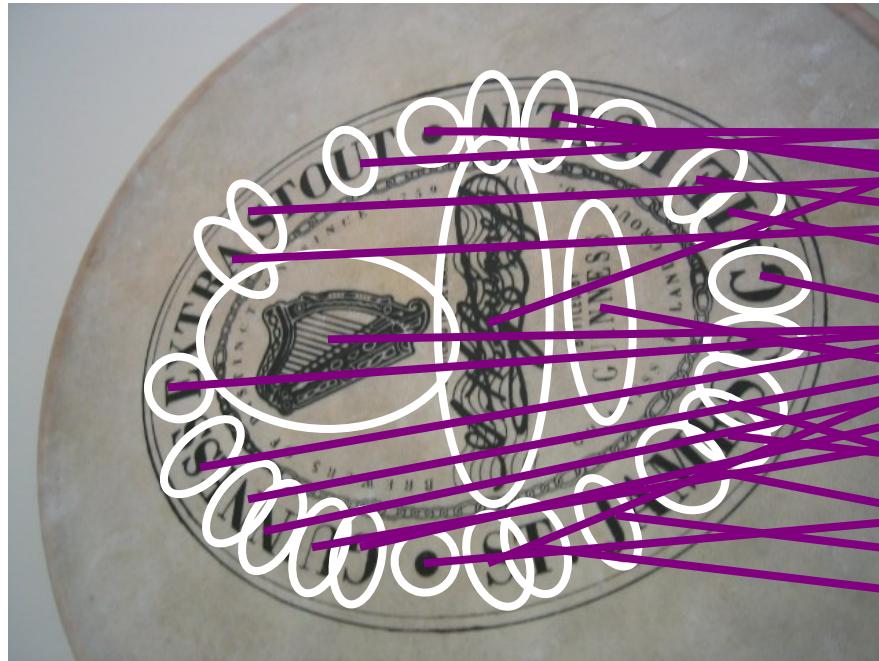
How to summarize the content of an entire image?
And gauge overall similarity?

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

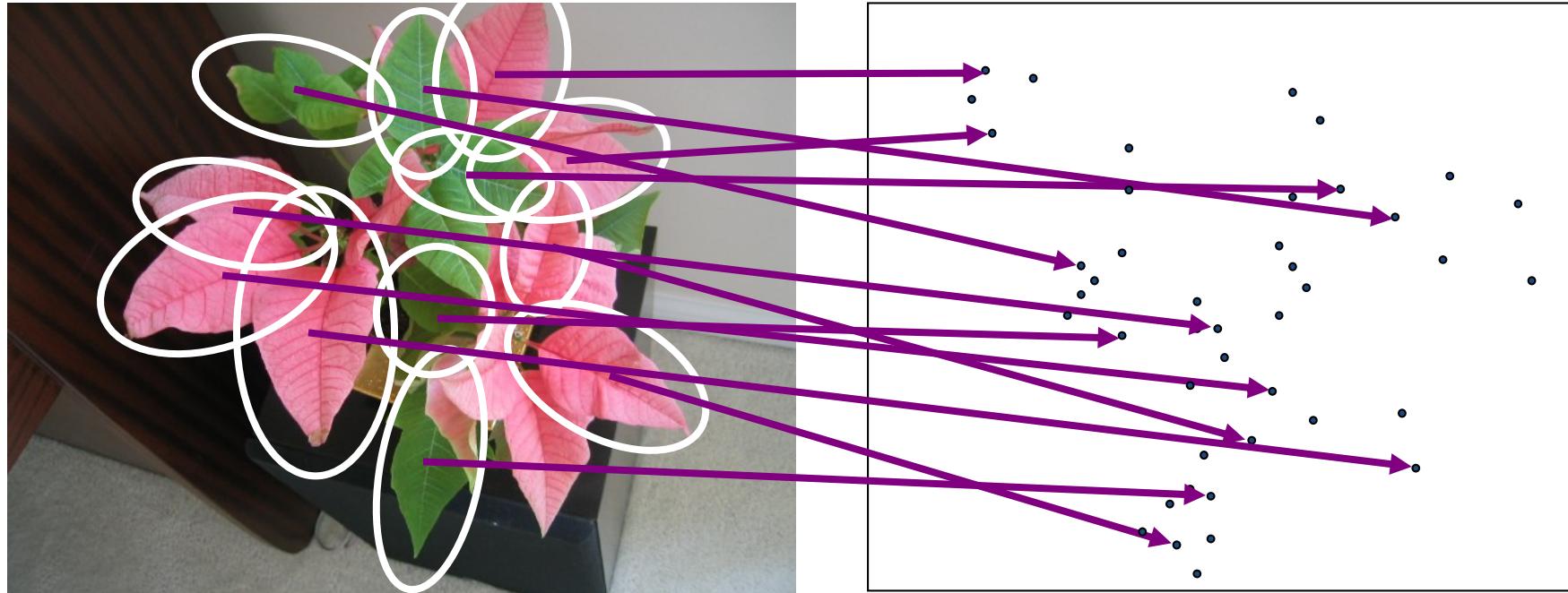
How to score the retrieval results?

How might we add more spatial verification?

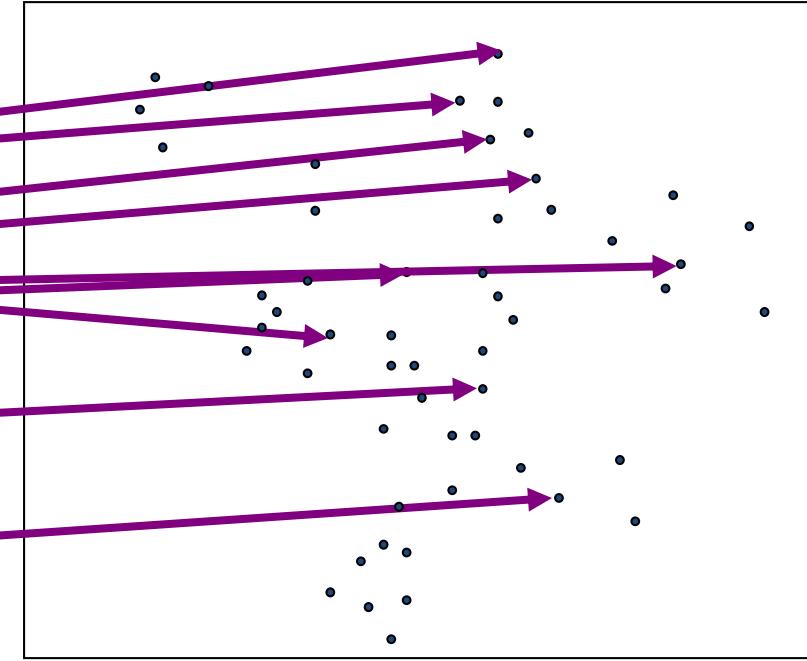
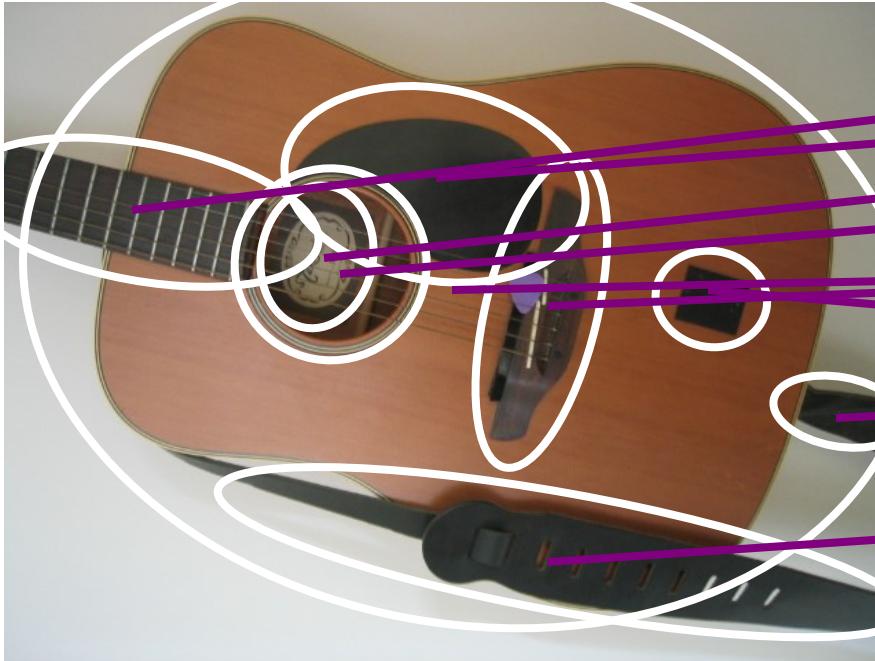
Training the vocabulary tree



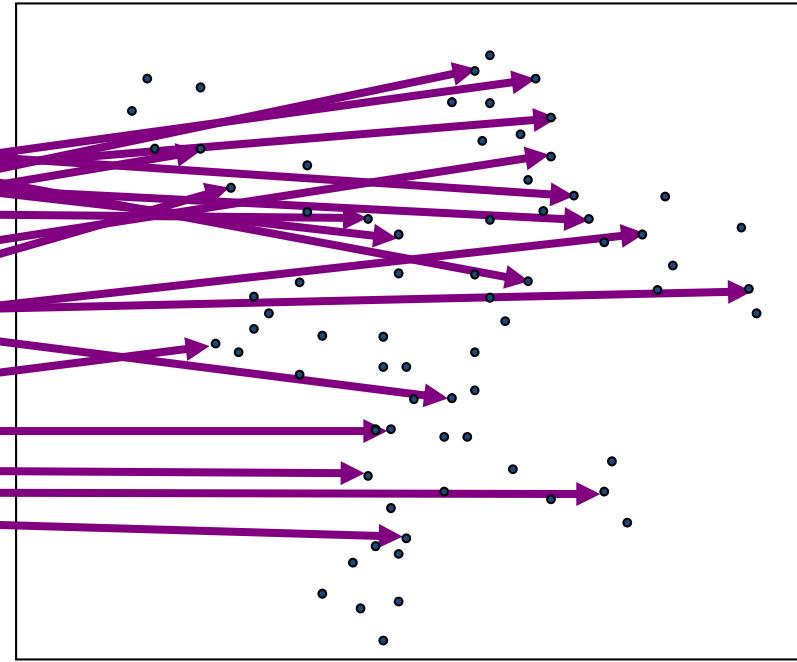
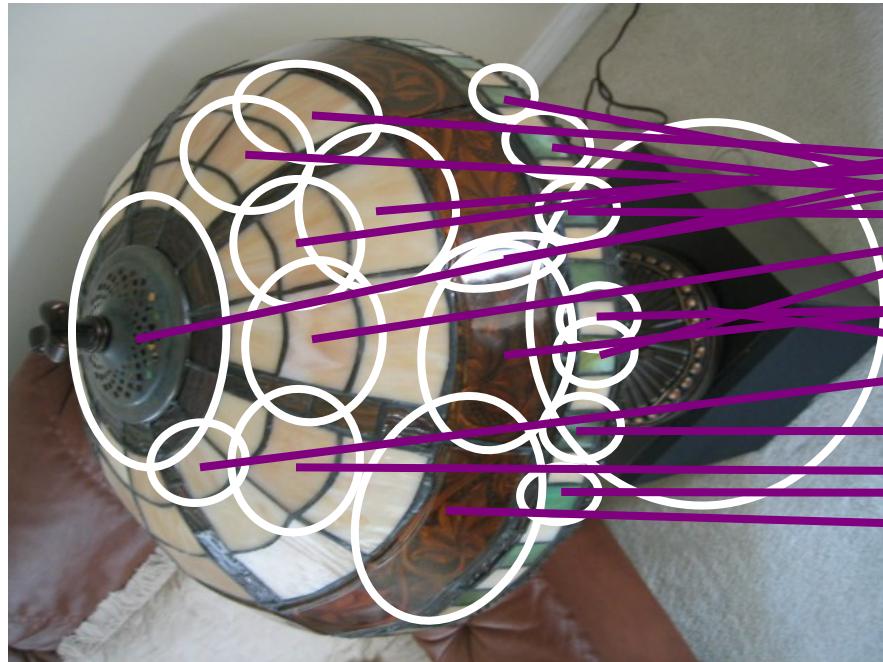
Training the vocabulary tree



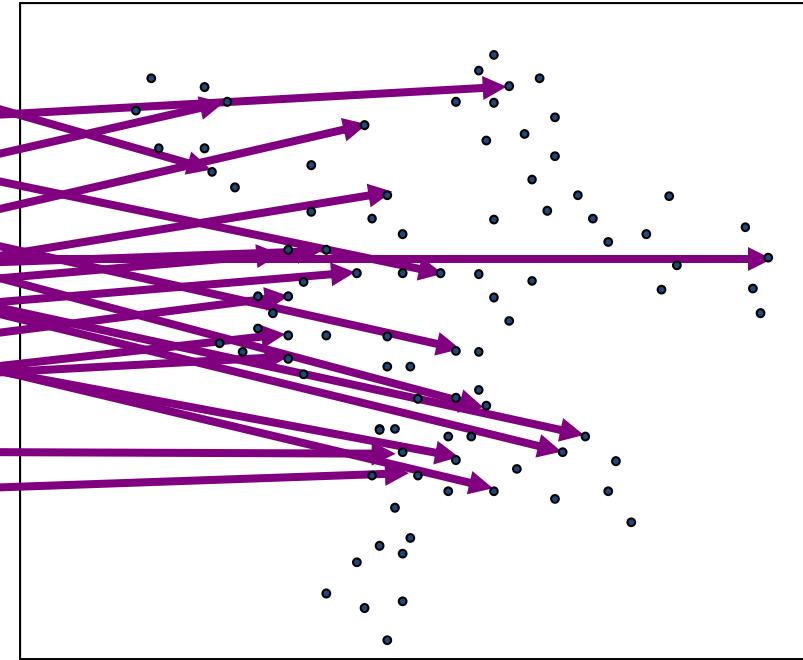
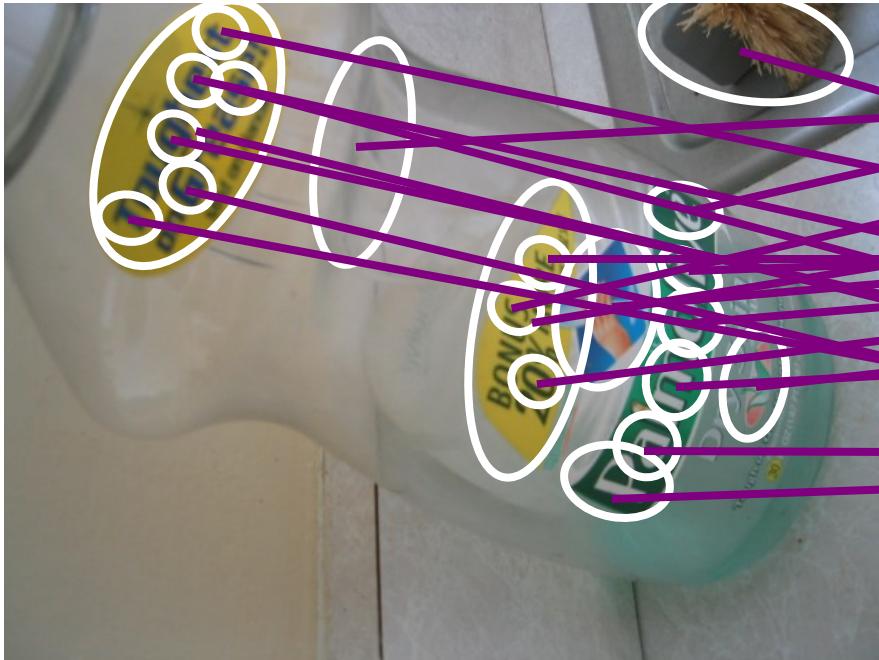
Training the vocabulary tree



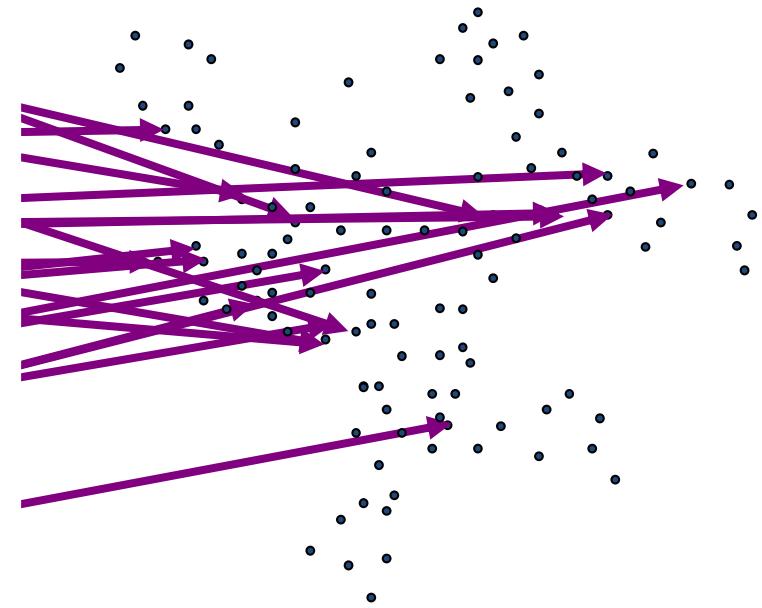
Training the vocabulary tree

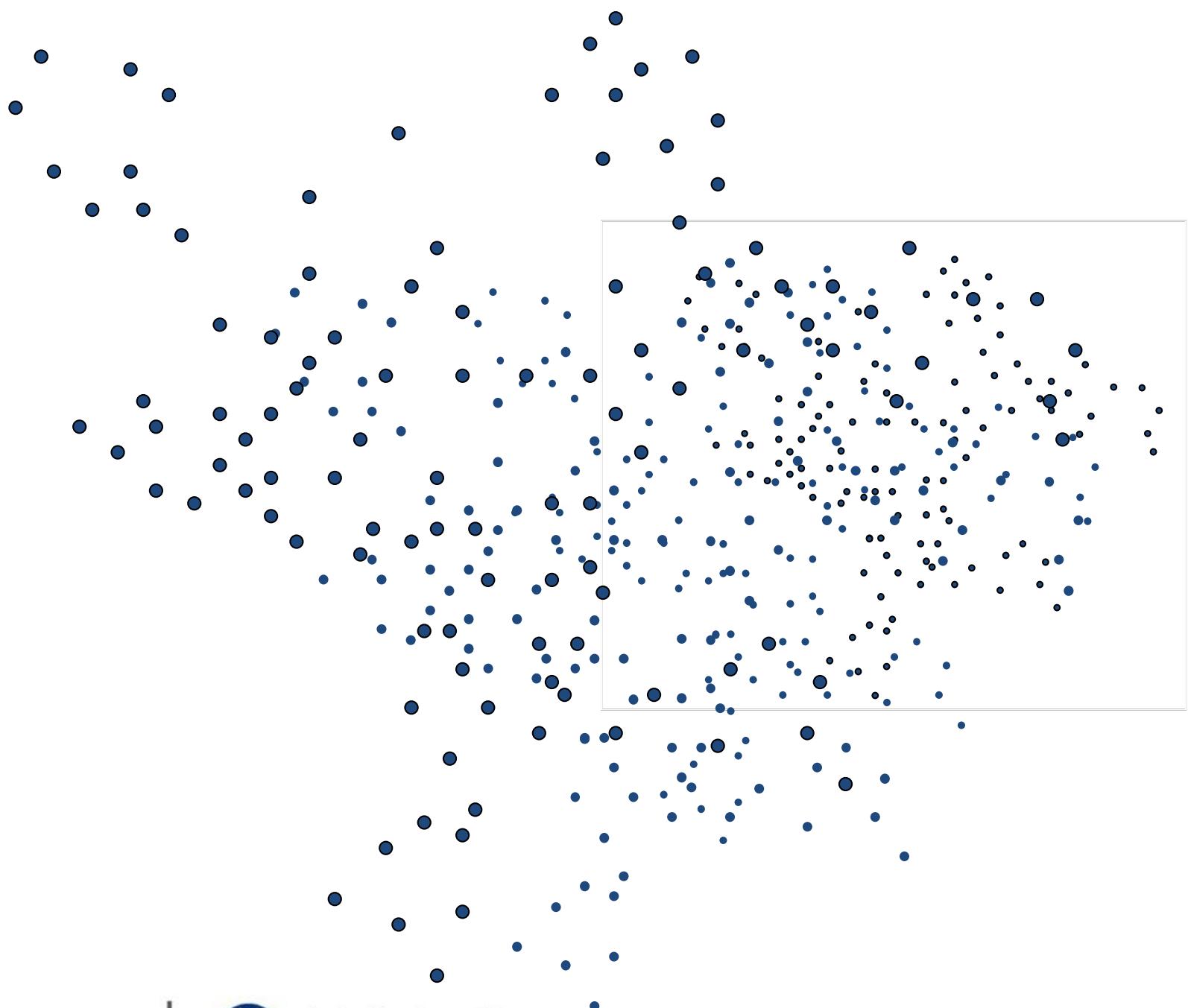


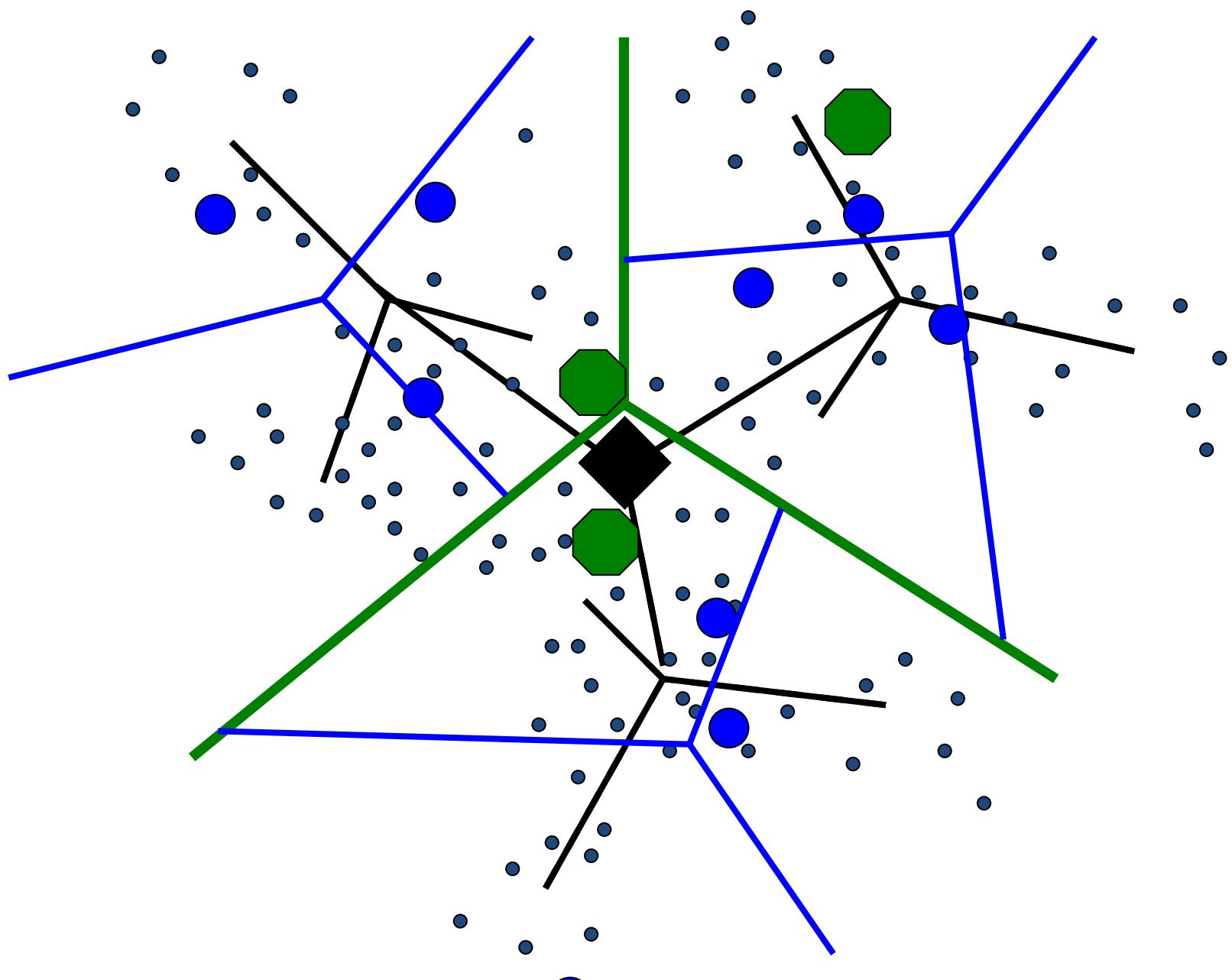
Training the vocabulary tree



Training the vocabulary tree



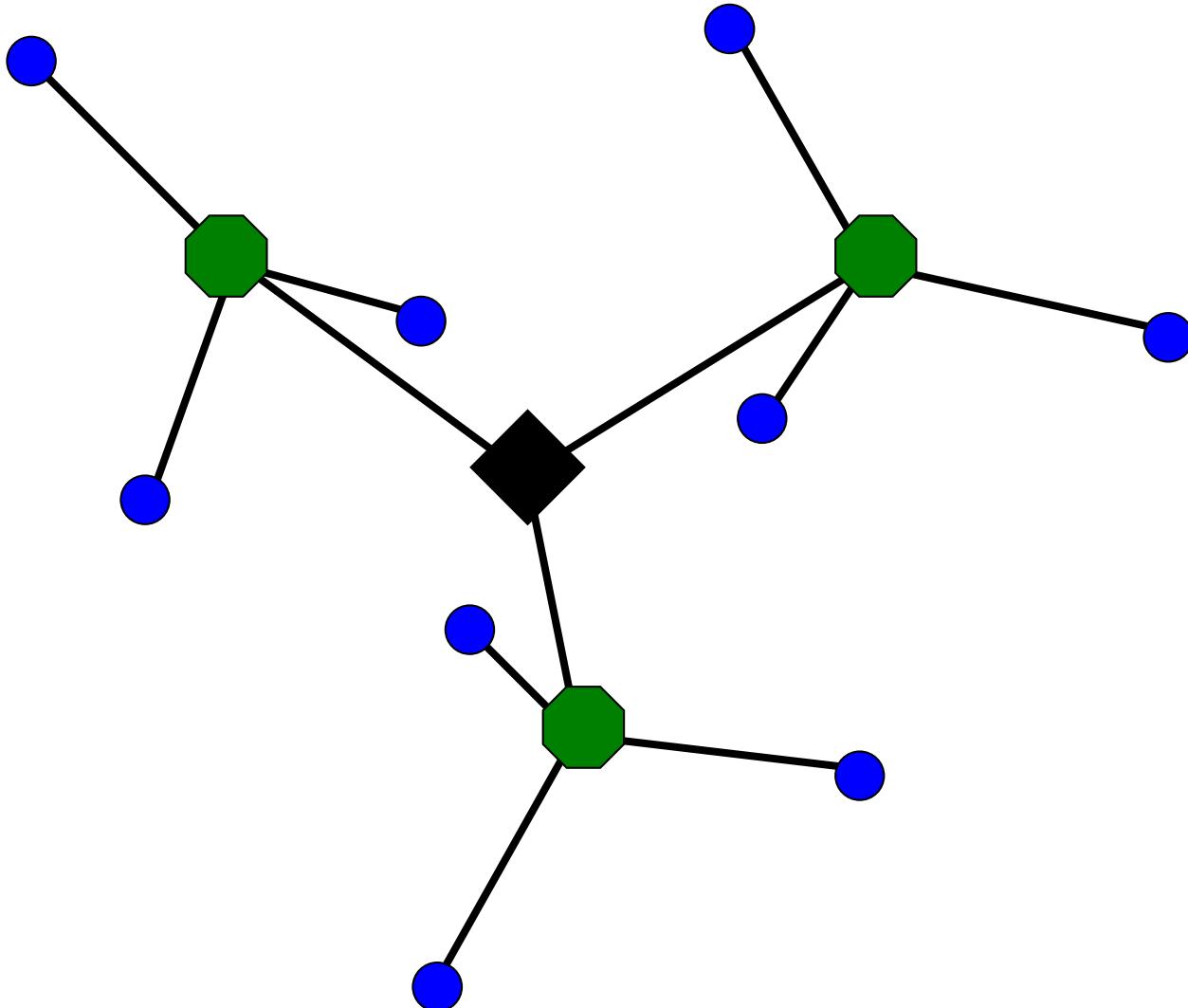


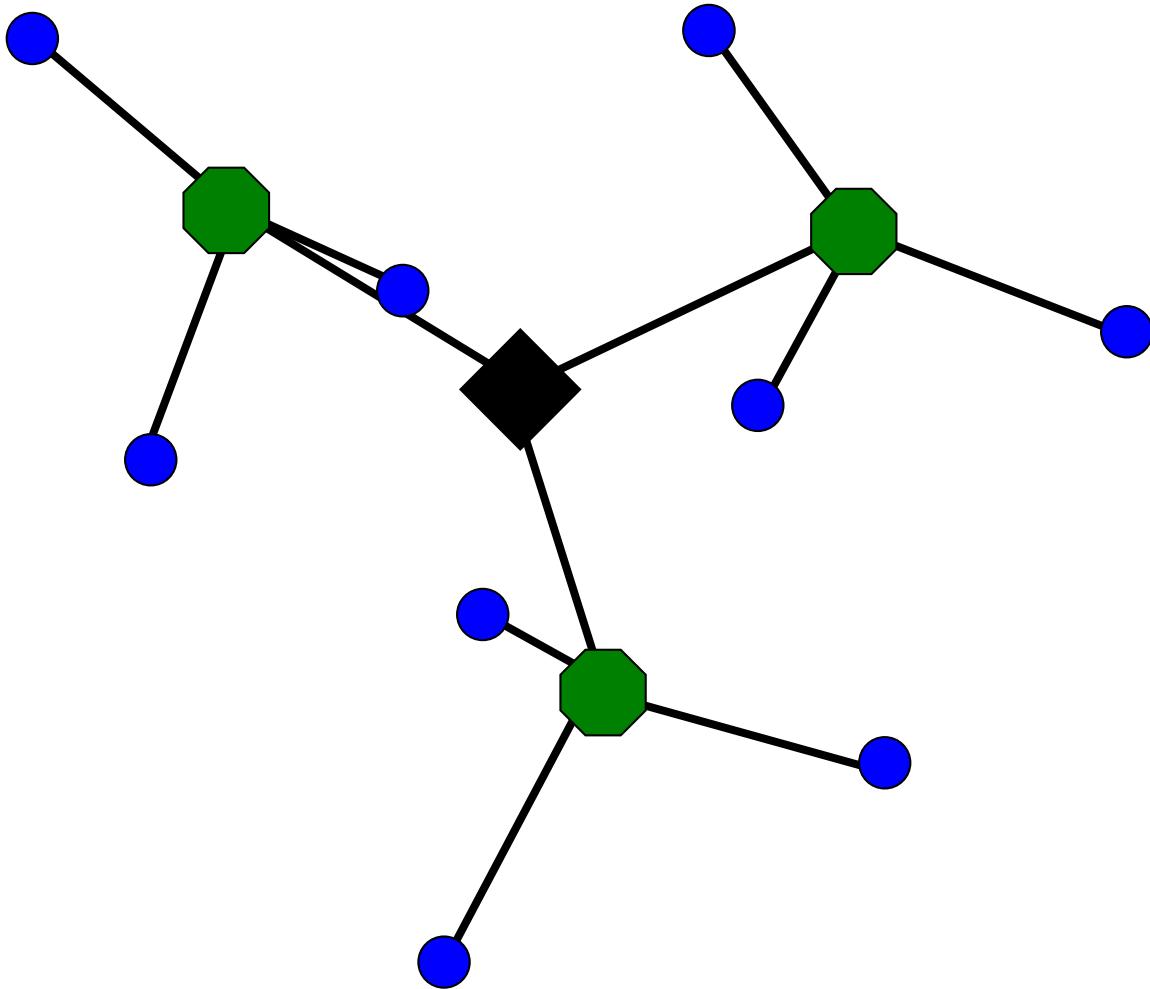


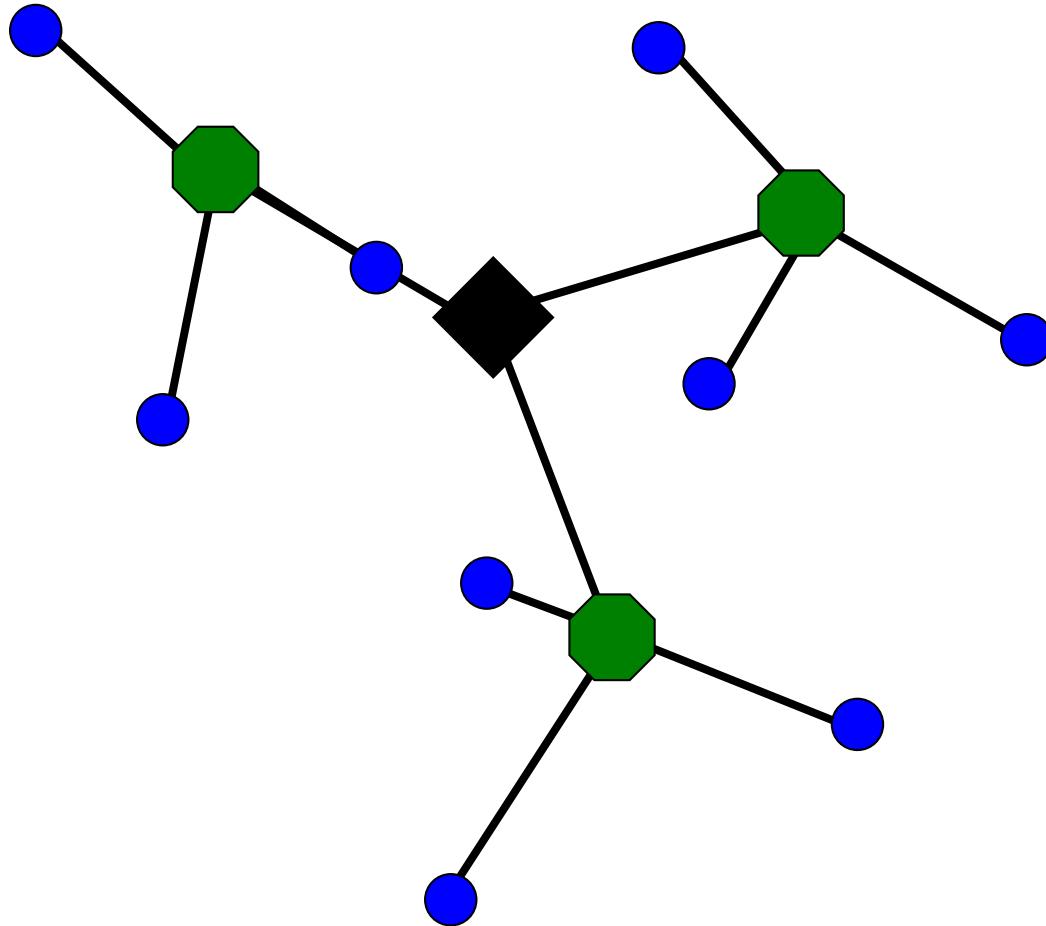
未来媒体研究中心
CENTER FOR FUTURE MEDIA

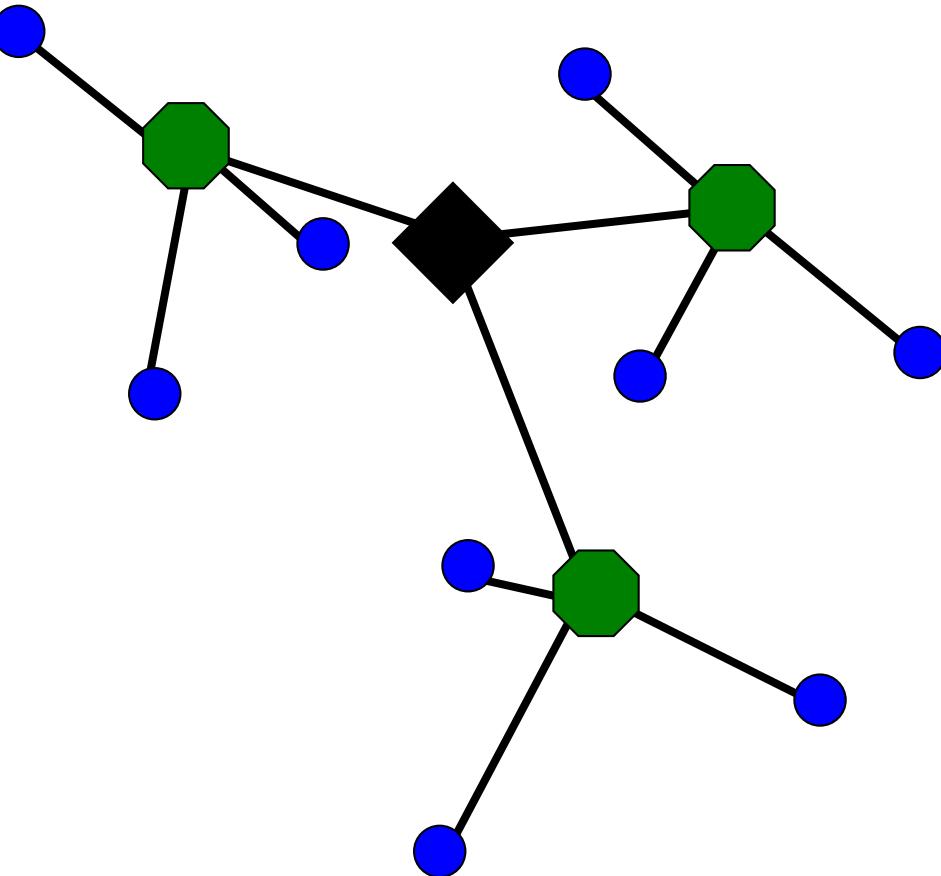


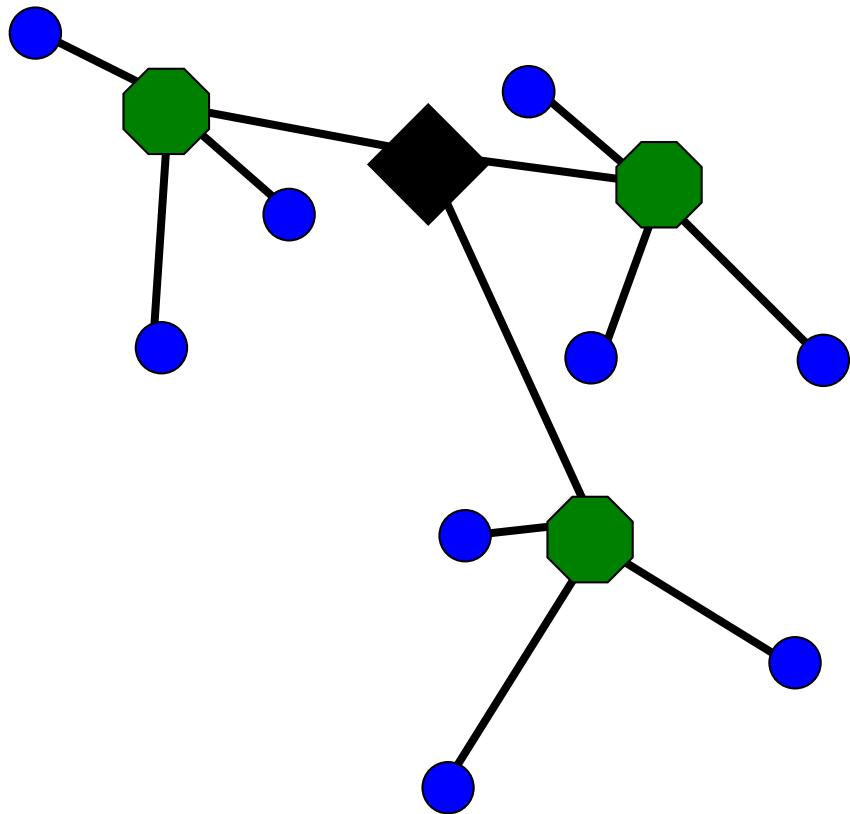
电子科技大学
University of Electronic Science and Technology of China

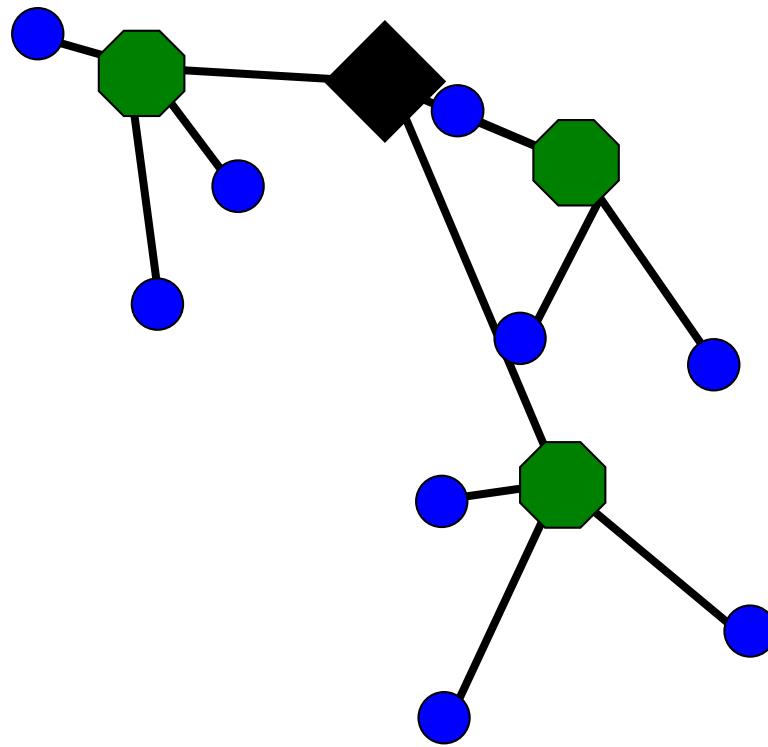


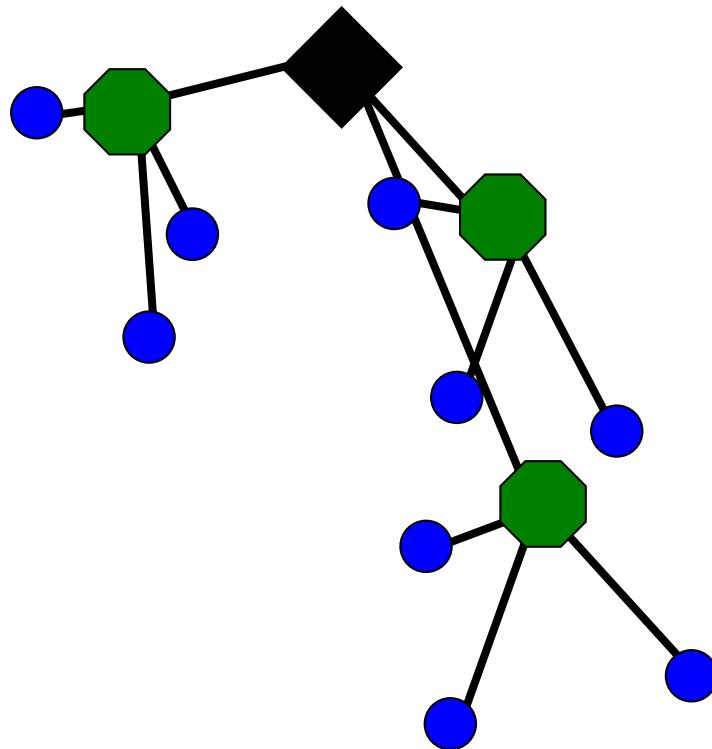


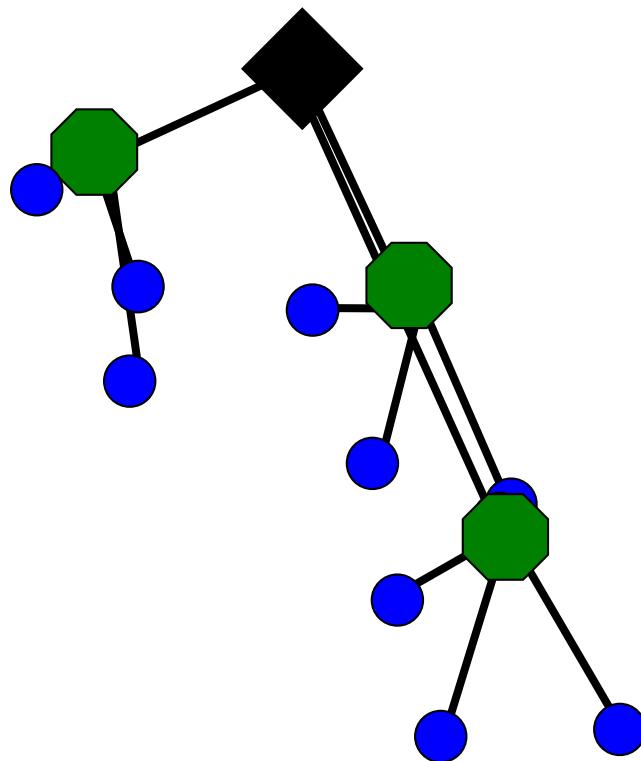


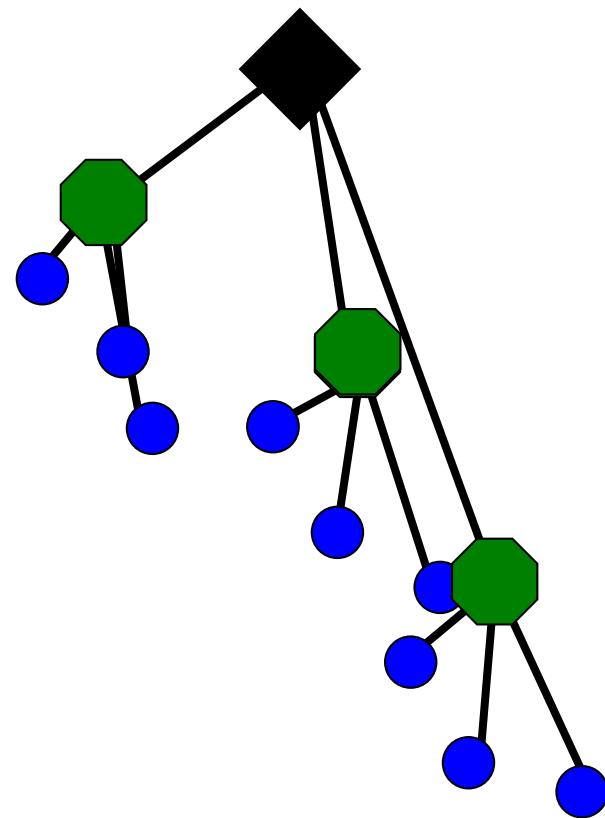




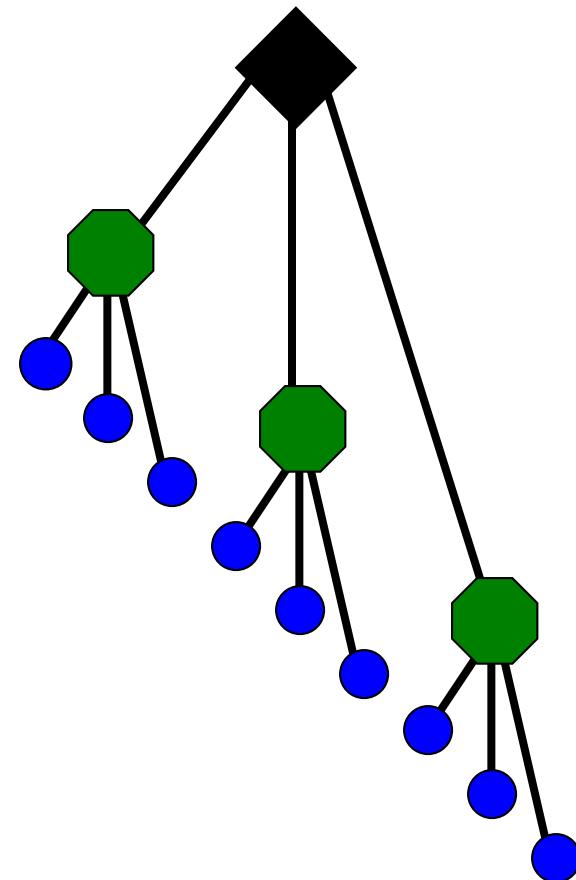




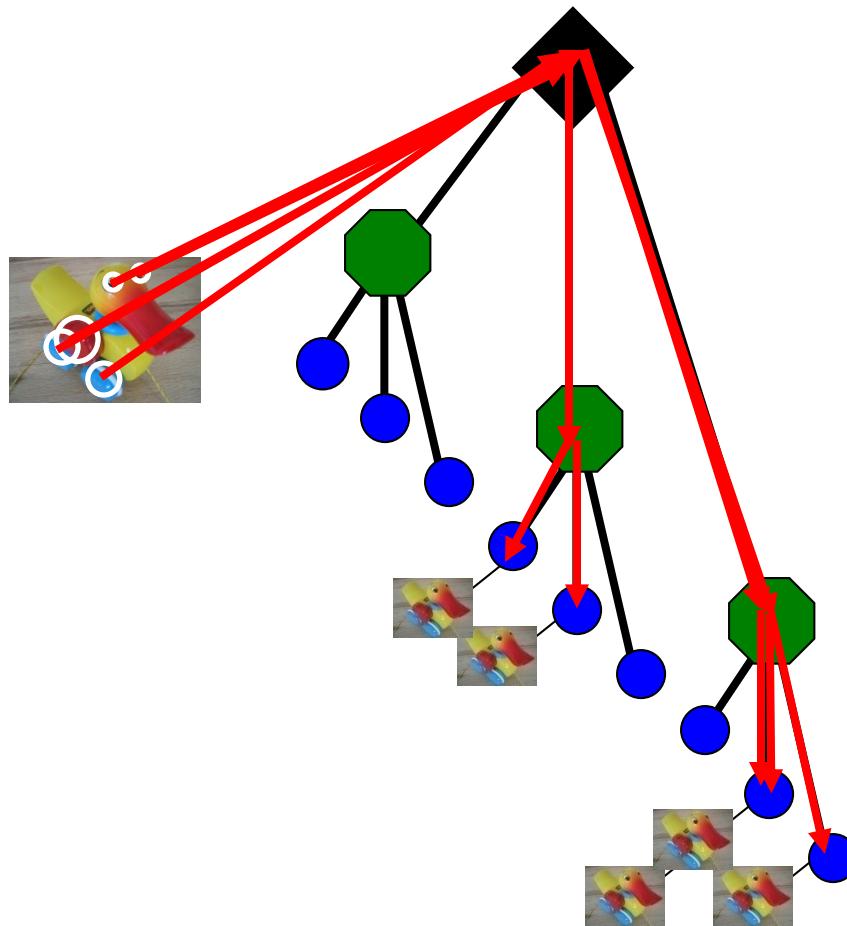


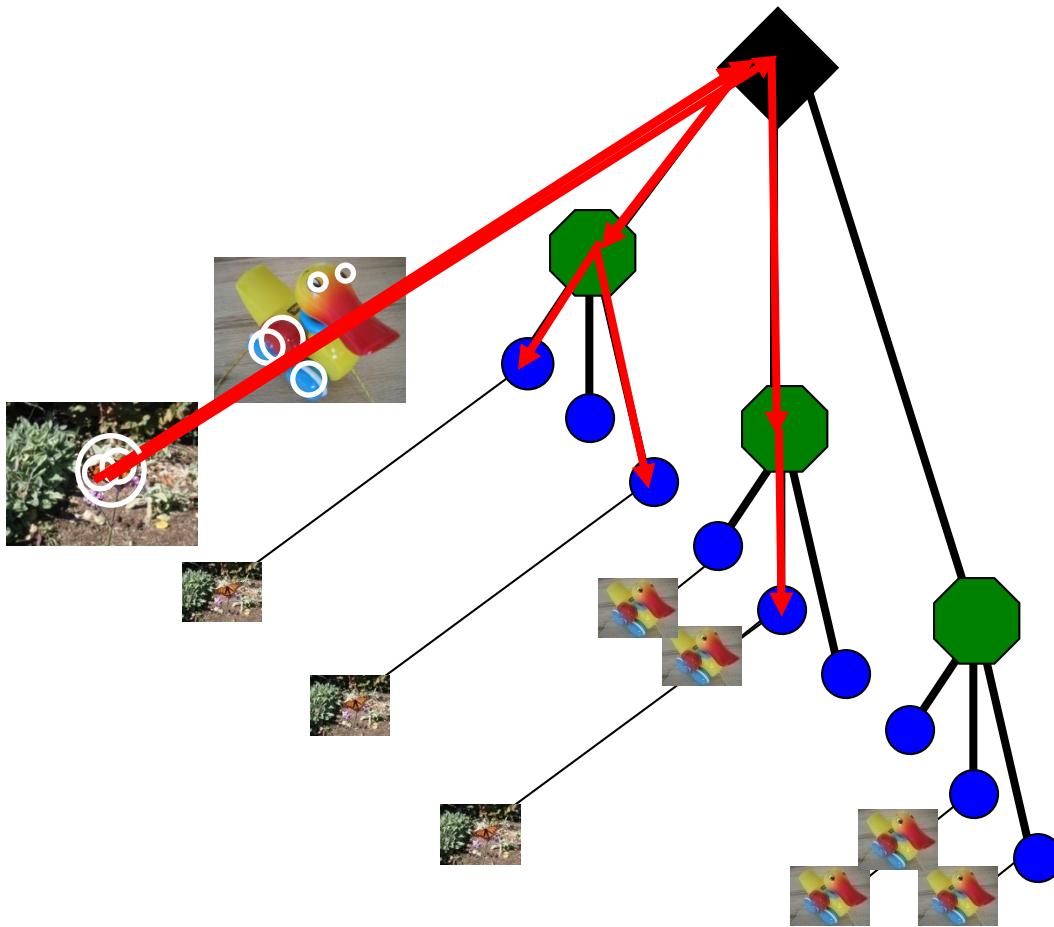


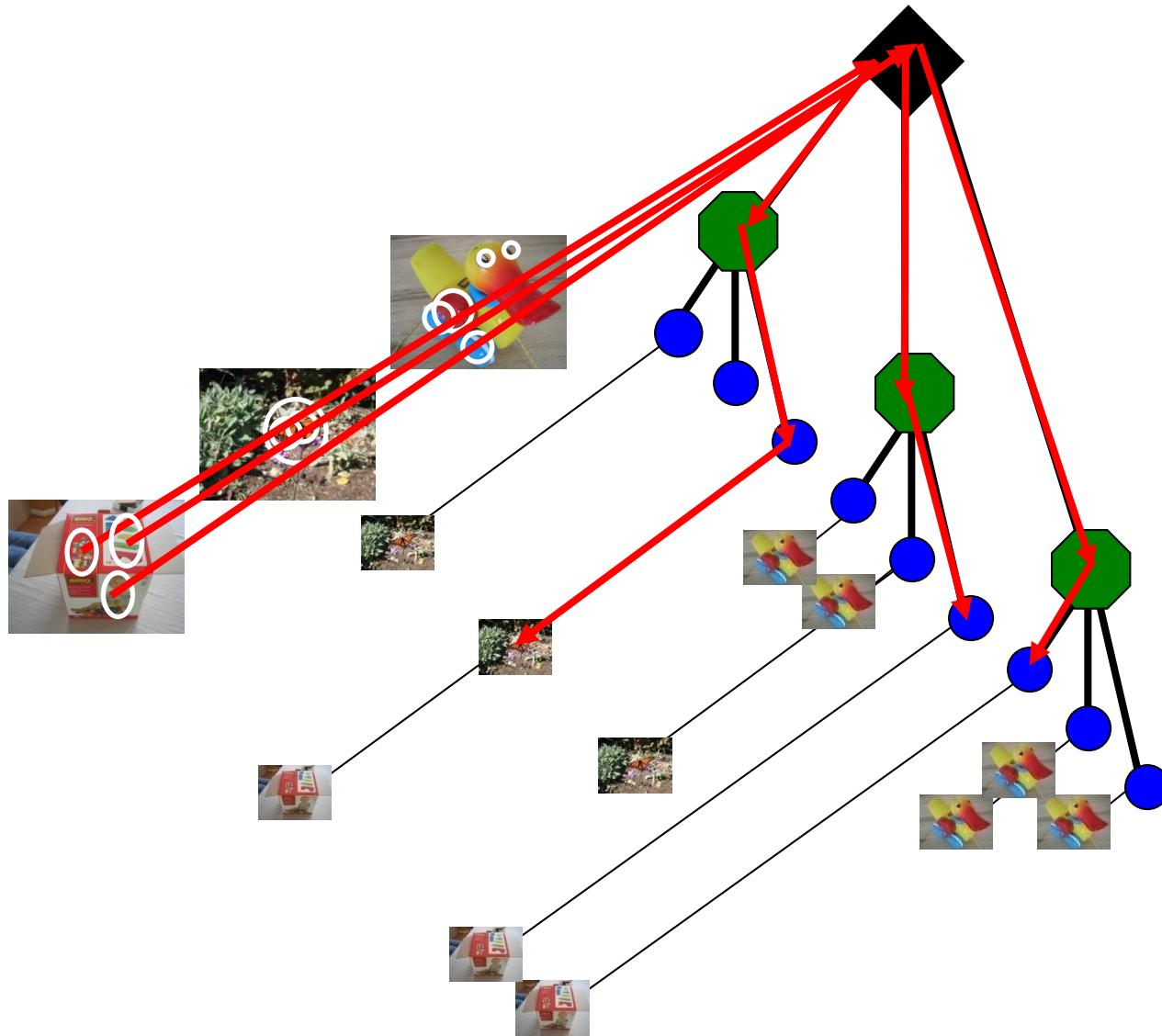
Vocabulary tree built recursively



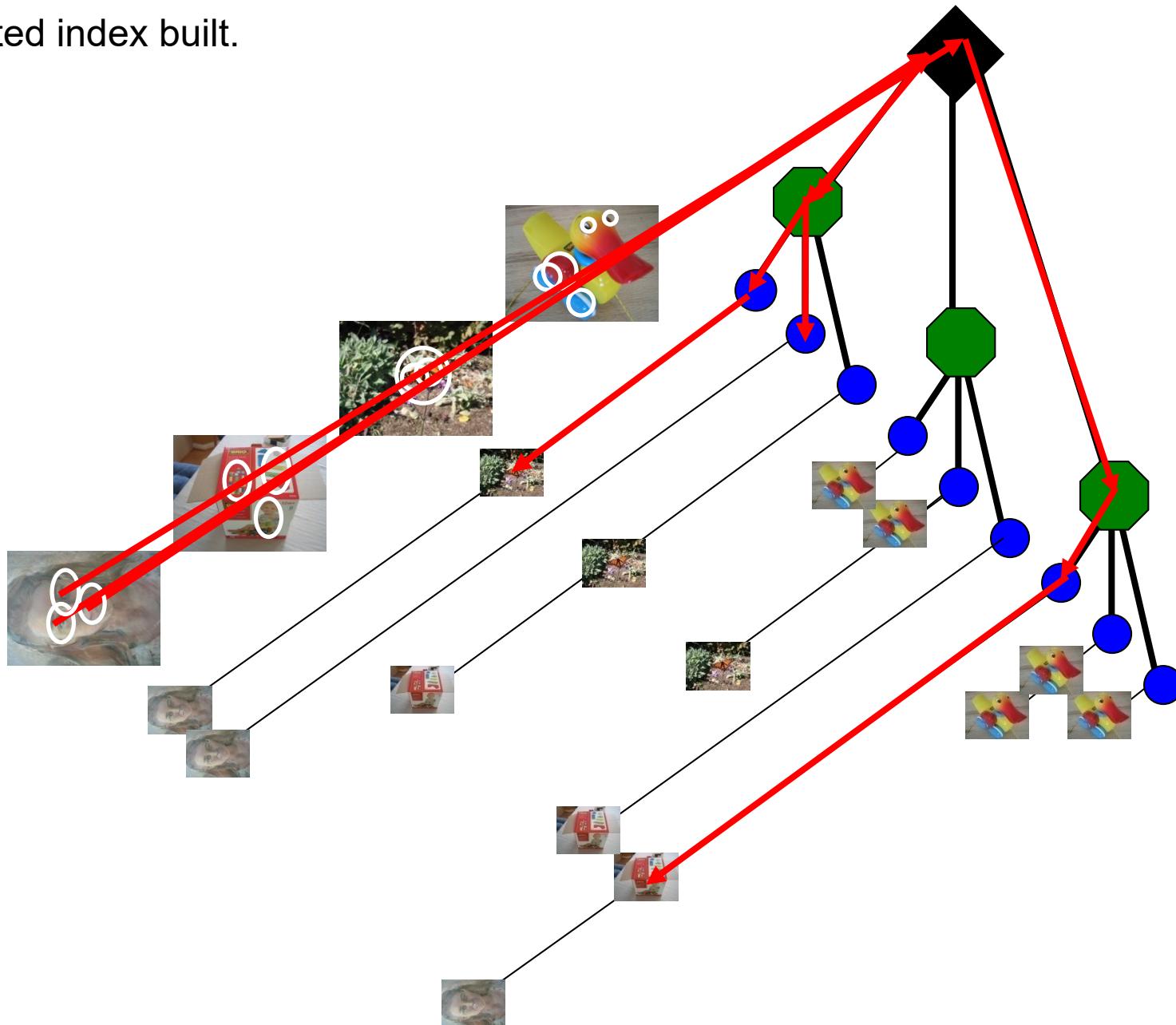
Each leaf has inverted index

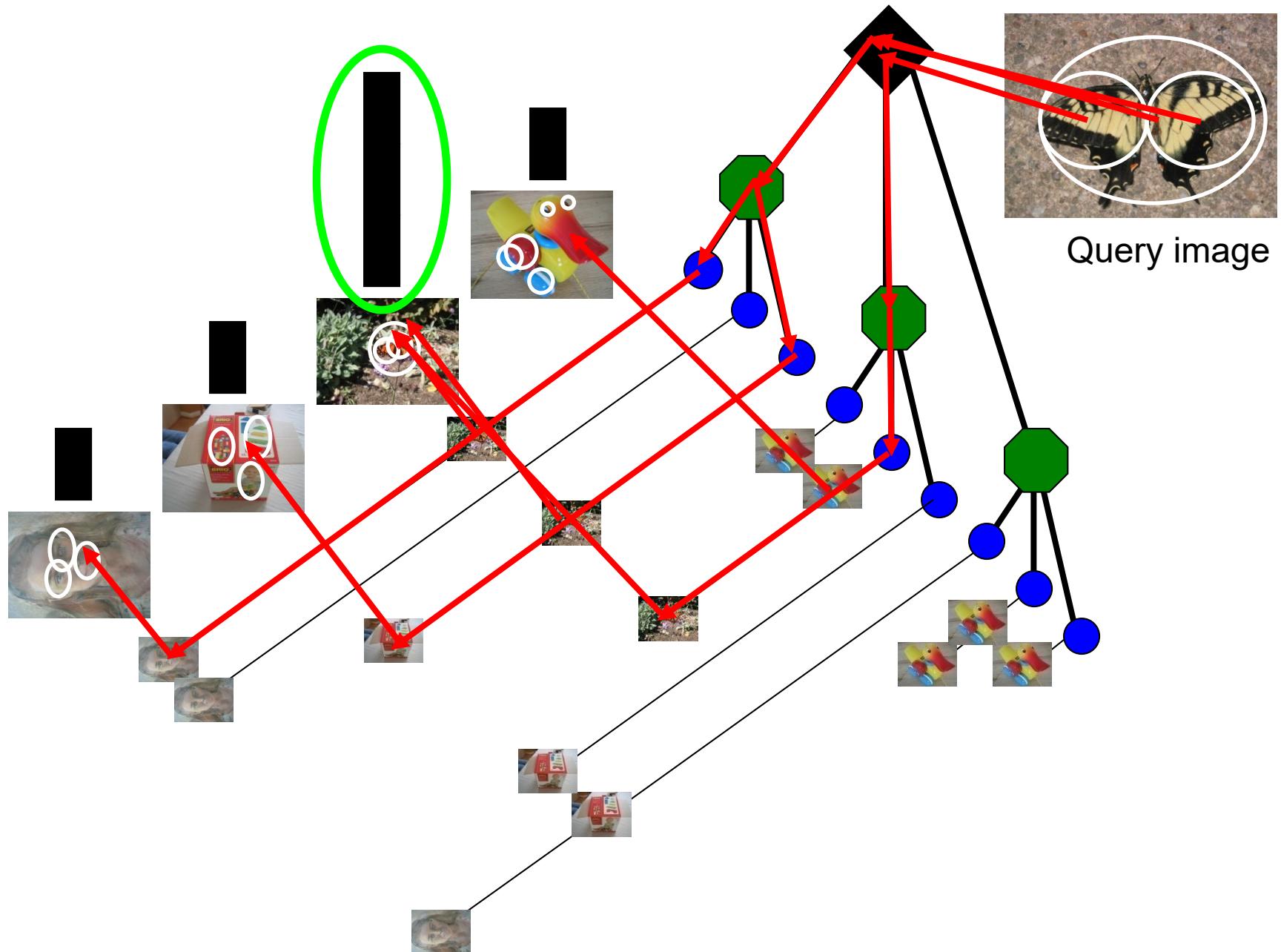






Inverted index built.



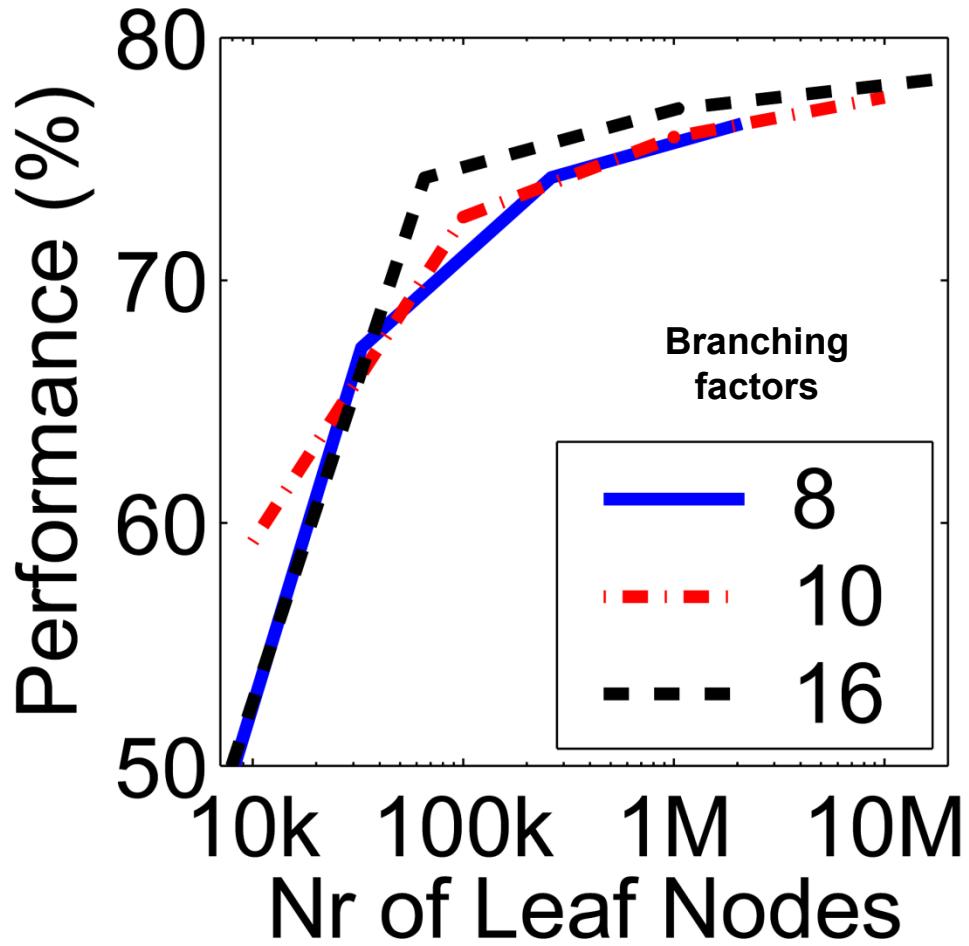


未来媒体研究中心
CENTER FOR FUTURE MEDIA



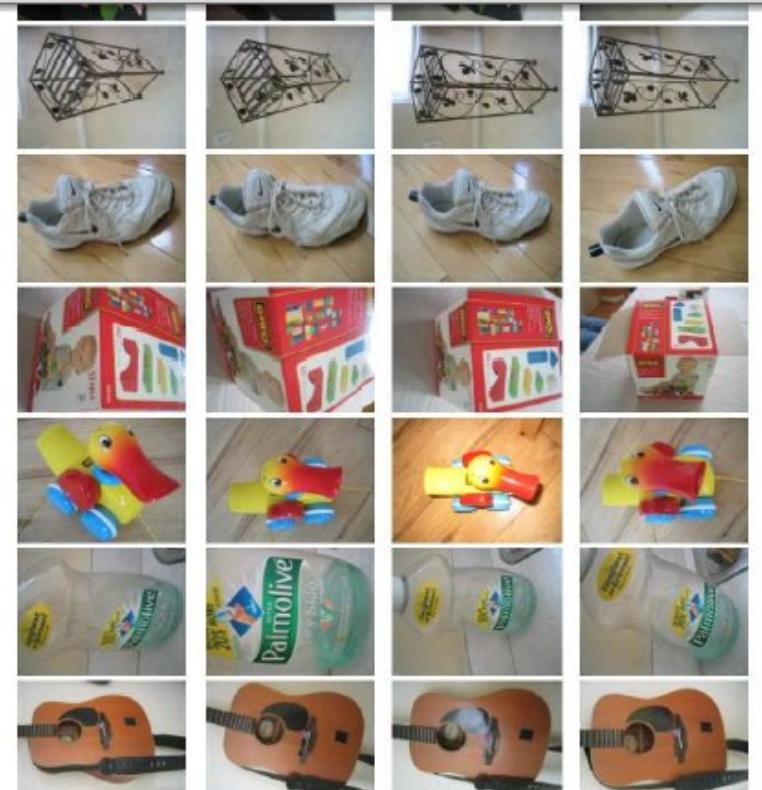
电子科技大学
University of Electronic Science and Technology of China

Vocabulary size



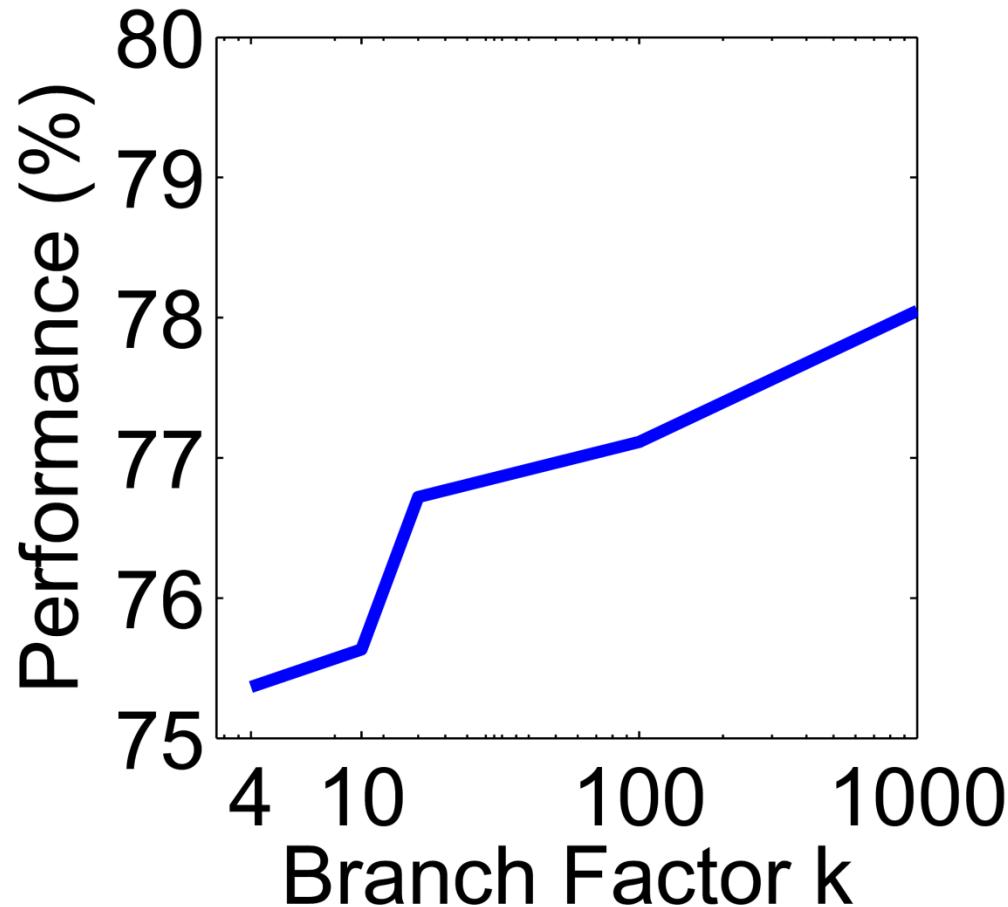
Influence on performance, sparsity

Recognition with 6347 images



Nister & Stewenius, CVPR 2006

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

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?

How to score the retrieval results?

How might we add more spatial verification?

Precision and Recall

True positive (tp) – correct attribution

True negative (tn) – correct rejection

False positive (fp) – incorrect attribution

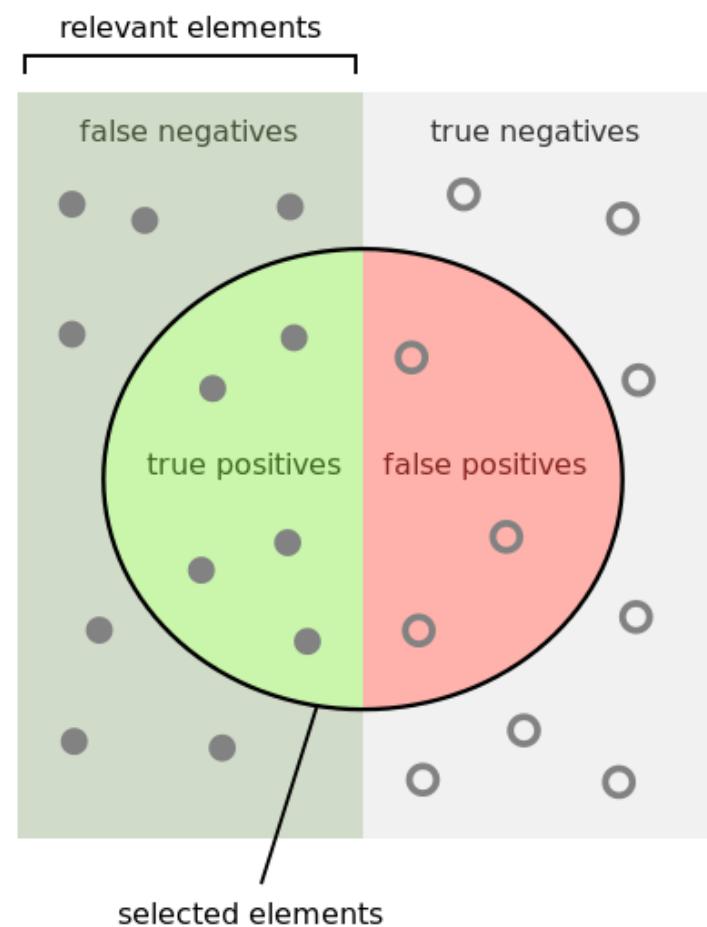
False negative (fn) – incorrect rejection

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

Precision = #relevant / #returned

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

Recall = #relevant / #total relevant



How many selected items are relevant?

$$\text{Precision} = \frac{\text{green}}{\text{red+green}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{green}}{\text{green+red}}$$

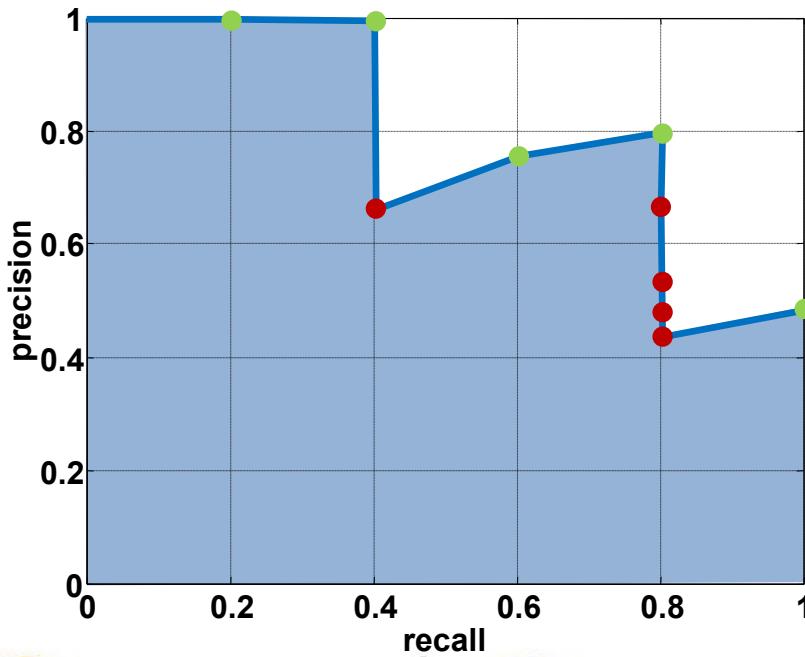
Scoring retrieval quality



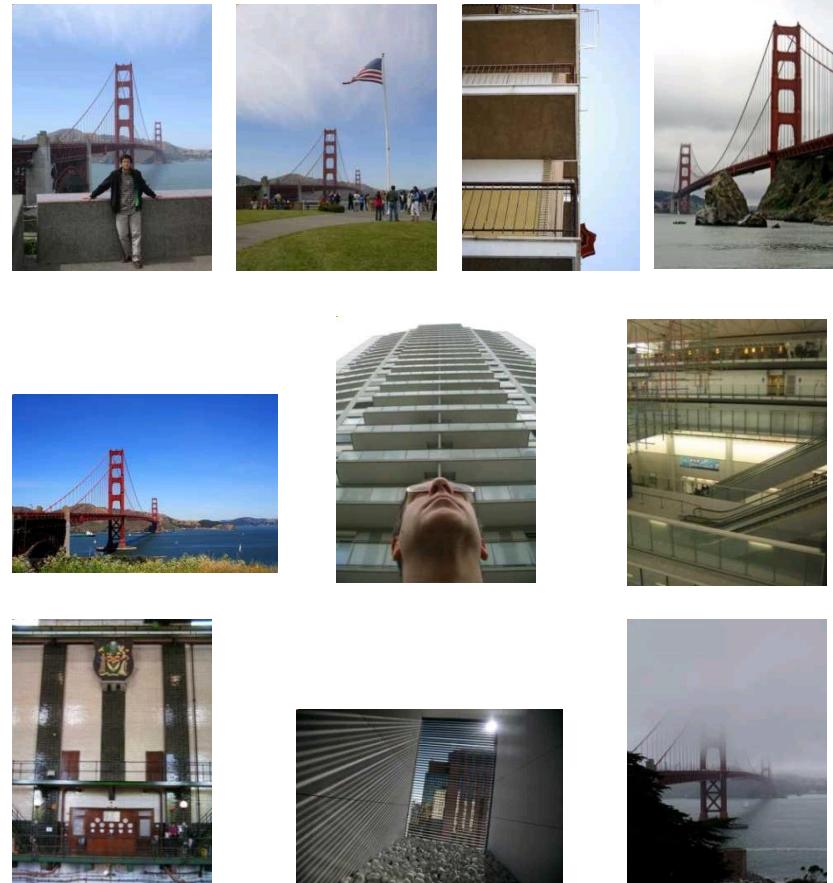
Query

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
Anhakca; 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
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. Augsutine; 191
Civil War; 100,108,127,138,141
Clearwater Marine Aquarium; 187
Collier County; 154
Collier, Barron; 152
Colonial Spanish Quarters; 168
Columbus County; 101,128
Coquina Building Material; 165
Corkscrew Swamp, Name; 154
Cowboys; 95
Crab Trap II; 144
Cracker, Florida; 88,95,132
Crossstown 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 pred
compa
\$660b
annoy
China'
deliber
agrees
yuan i
govern
also n
demar
countr
yuan a
permitt
the US
freely.
it will t
allowin

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



tf-idf weighting

- Term frequency – inverse document frequency
- Describe image 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

Query expansion

Query: ***golf green***

Results:

- How can the grass on the ***greens*** at a ***golf*** course be so perfect?
- For example, a skilled ***golfer*** expects to reach the ***green*** on a par-four hole in ...
- Manufactures and sells synthetic ***golf*** putting ***greens*** and mats.

Irrelevant result can cause a ‘topic drift’:

- Volkswagen ***Golf***, 1999, ***Green***, 2000cc, petrol, manual, , hatchback, 94000miles, 2.0 GTi, 2 Registered Keepers, HPI Checked, Air-Conditioning, Front and Rear Parking Sensors, ABS, Alarm, Alloy

Query expansion

Results



Query image

Spatial verification



New query

New results



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

Recognition Issues

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

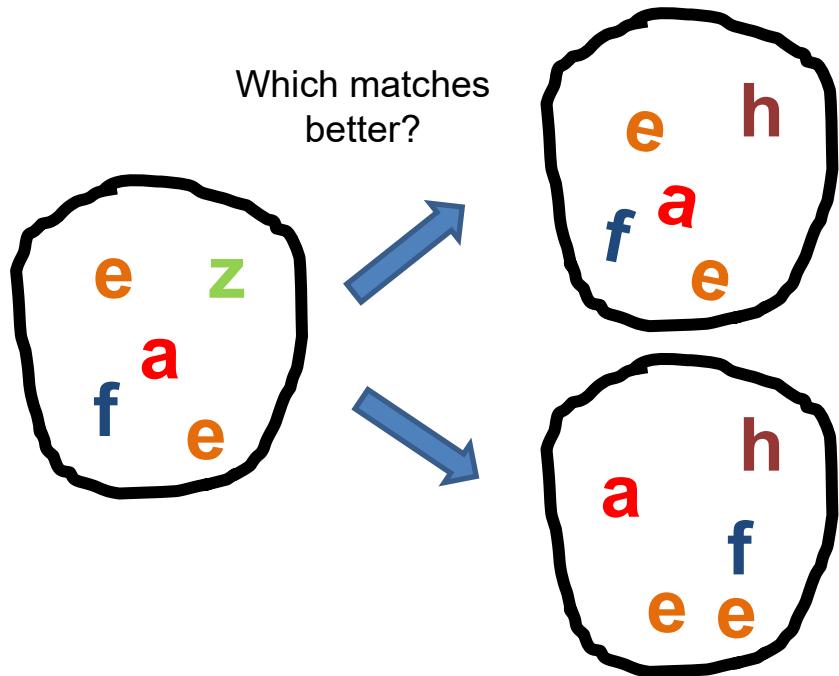
How large should the vocabulary be? How to
perform quantization efficiently?

How to score the retrieval results?

How might we add more spatial verification?

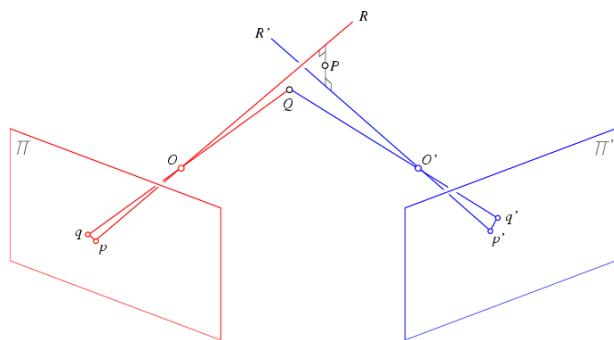
Can we be more accurate?

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

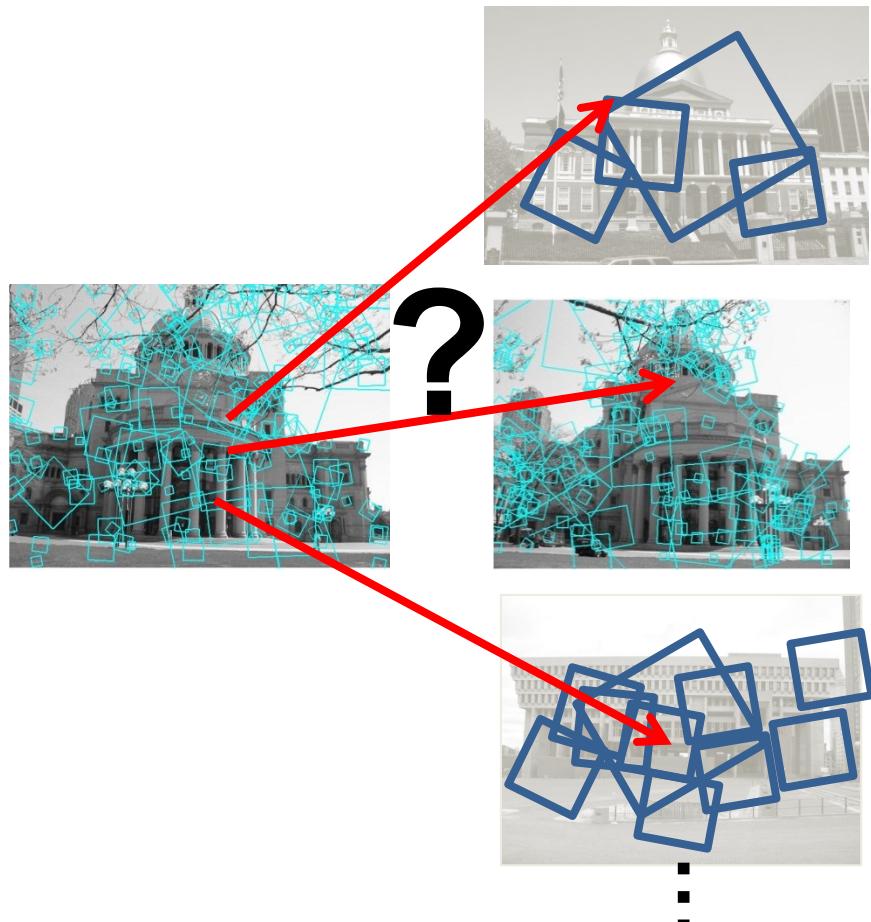


Real objects have
consistent geometry

Multi-view matching



VS



Matching two given
views for depth

Search for a matching
view for recognition

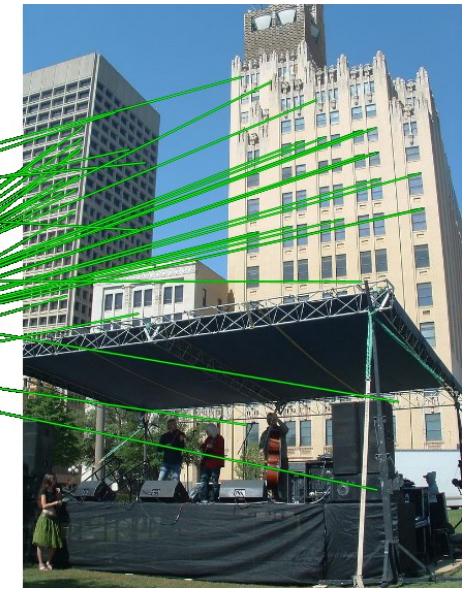
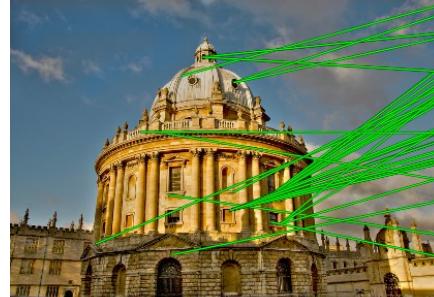
Spatial Verification

Query



DB image with high BoW
similarity

Query

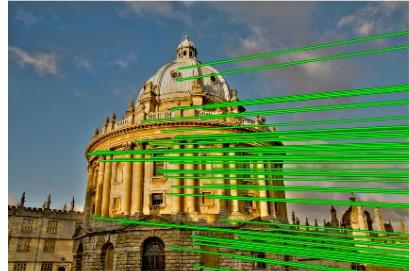


DB image with high BoW
similarity

Both image pairs have many visual words in common.

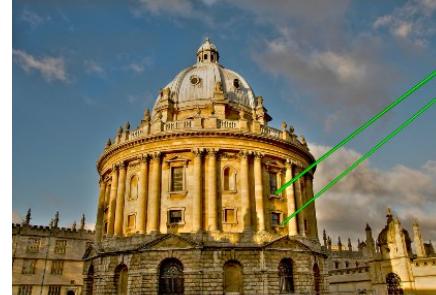
Spatial Verification

Query



DB image with high BoW
similarity

Query



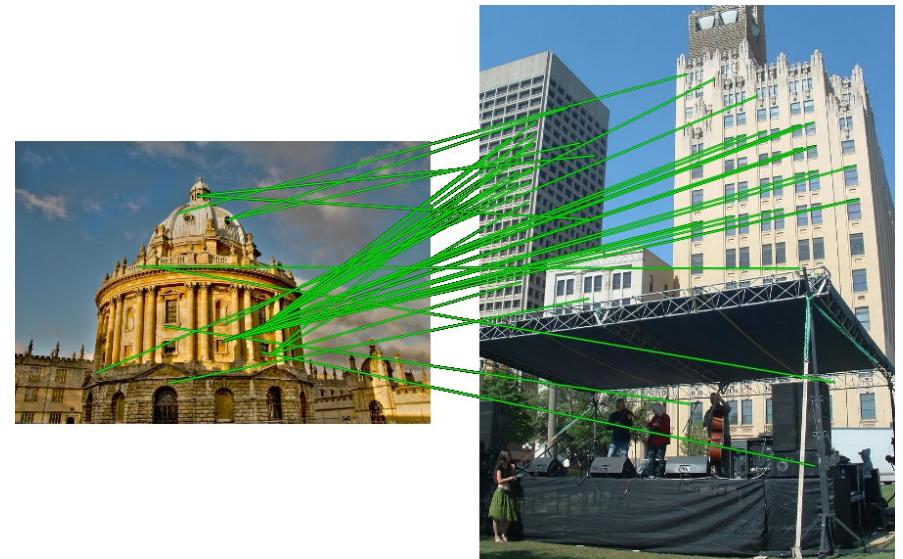
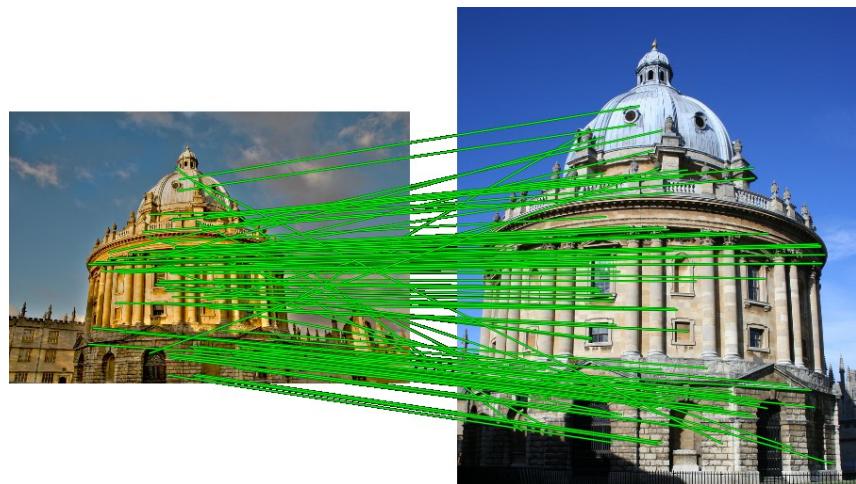
DB image with high BoW
similarity

Only some of the matches are mutually consistent with real-world geometry imaged by a camera.

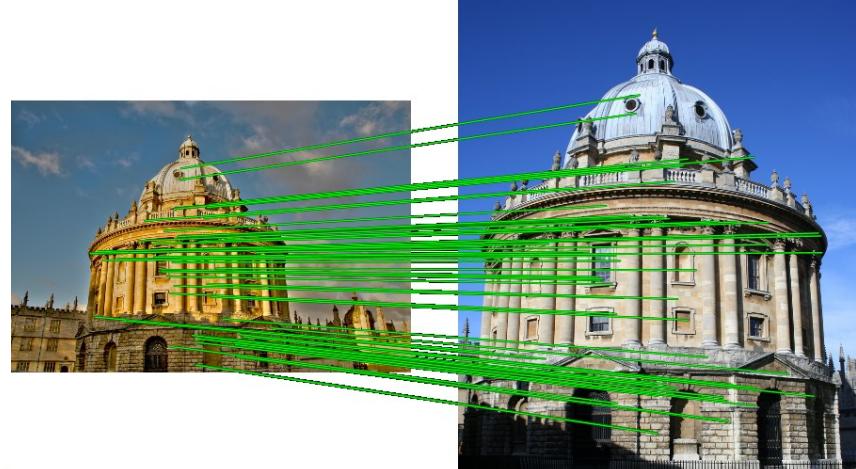
Spatial Verification: two basic strategies

- 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

397 Well-sampled Categories



...at least 100 unique images each.



未来媒体研究中心
CENTER FOR FUTURE MEDIA



电子科技大学
University of Electronic Science and Technology of China

Evaluating Human Scene Classification



?

Accuracy

98%

90%

68%

bathroom(100%)



beauty salon(100%)



bedroom(100%)



bullring(100%)



playground(100%)



podium outdoor(100%)



greenhouse outdoor(100%)



tennis court outdoor(100%)



wind farm(100%)



veterinarians office(100%)



riding arena(100%)



Scene category

Most confusing categories

Inn (0%)



Bayou (0%)



Basilica (0%)



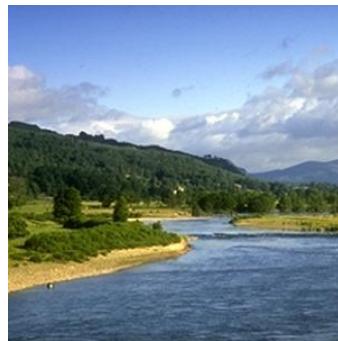
Restaurant patio (44%)



Chalet (19%)



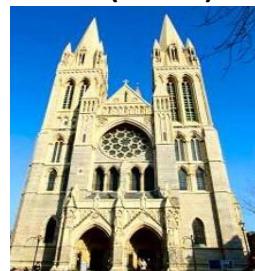
River (67%)



Coast (8%)



Cathedral(29%)



Courthouse (21%)



未来媒体研究中心
CENTER FOR FUTURE MEDIA



电子科技大学
University of Electronic Science and Technology of China

Conclusion: humans can do it

- The SUN database is reasonably consistent and categories can be told apart by humans.
- With many very specific categories, humans get it right 2/3rds of the time *from experience and from exploring the label space.*

So, how do humans classify scenes?

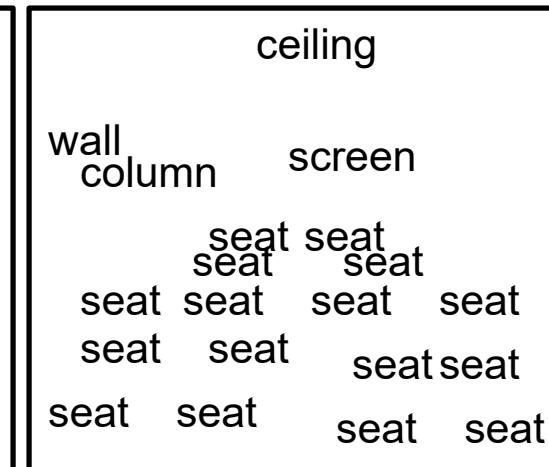
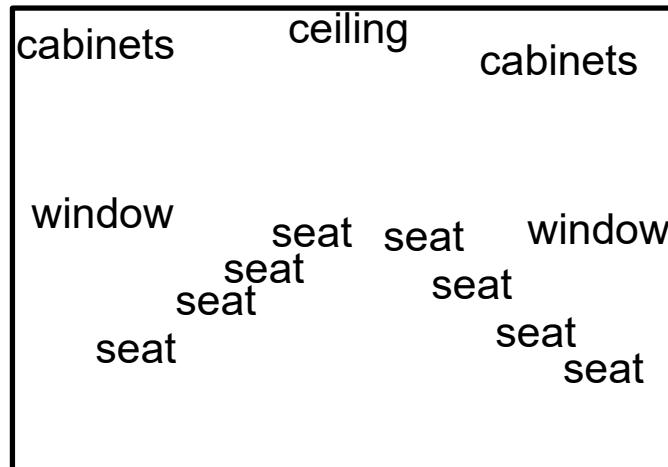
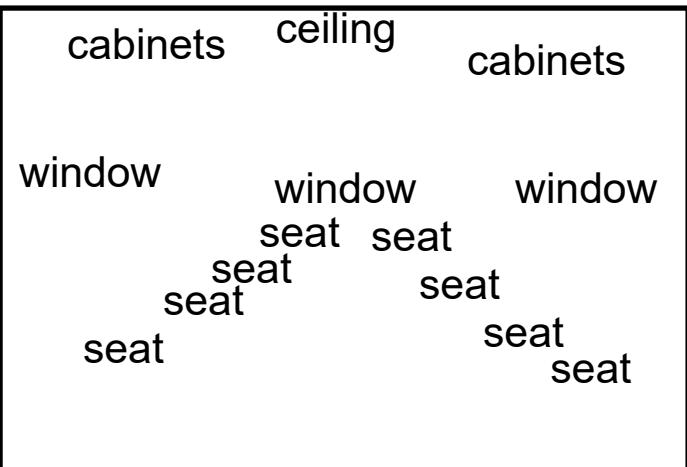
How do we classify scenes?



Ceiling Light		Ceiling Lamp		
Door	Door	Painting	mirror	
Wall	Door	wall		wall
		Fireplace	armchair	Lamp
Floor		armchair		phone
		Coffee table		alarm
				Side-table
				carpet

Different objects, different spatial layout

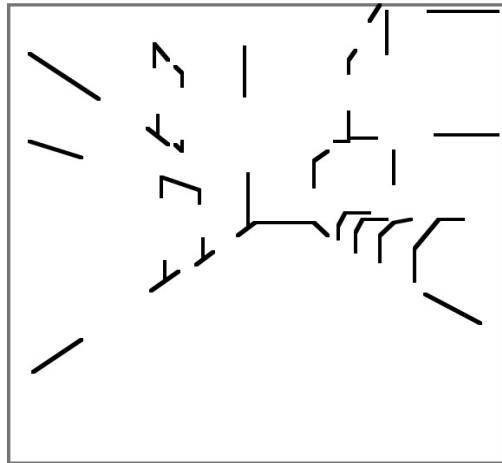
Which are the important elements?



Similar objects, and similar spatial layout
Different lighting, different materials, different “stuff”

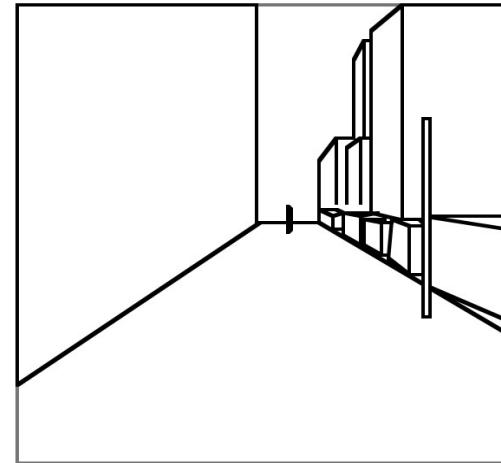
Scene emergent features

“Recognition via features that are not those of individual objects but “emerge” as objects are brought into relation to each other to form a scene.” – Biederman 81



Suggestive edges and junctions

Biederman, 1981



Simple geometric forms

Biederman, 1981



Blobs

Bruner and Potter, 1969



Textures

Oliva and Torralba,
2001

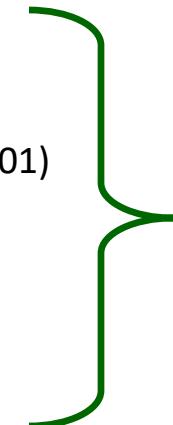


未来媒体研究中心
CENTER FOR FUTURE MEDIA



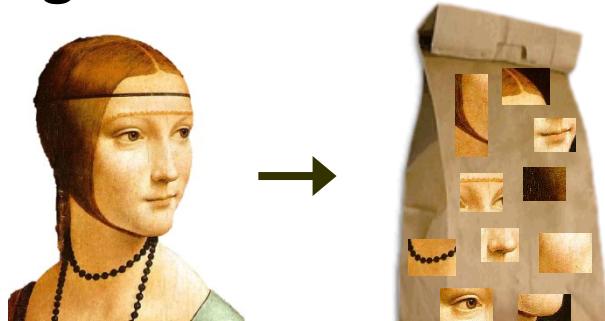
电子科技大学
University of Electronic Science and Technology of China

Global Image Descriptors

- Tiny images (Torralba et al, 2008)
 - Color histograms
 - Self-similarity (Shechtman and Irani, 2007)
 - Geometric class layout (Hoiem et al, 2005)
 - Geometry-specific histograms (Lalonde et al, 2007)
 - Dense and Sparse SIFT histograms
 - Berkeley texton histograms (Martin et al, 2001)
 - HoG 2x2 spatial pyramids
 - Gist scene descriptor (Oliva and Torralba, 2008)
- 
- Texture
Features

Global Texture Descriptors

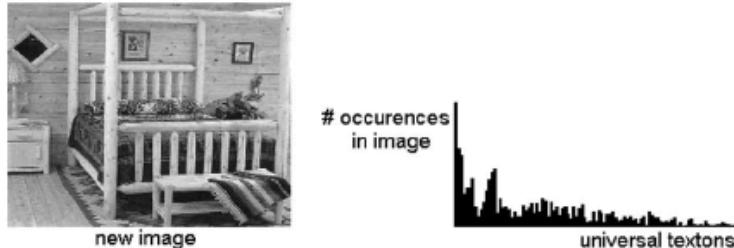
Bag of words



Sivic et. al., ICCV 2005

Fei-Fei and Perona, CVPR 2005

Non-localized textons



Walker, Malik. Vision Research 2004

...

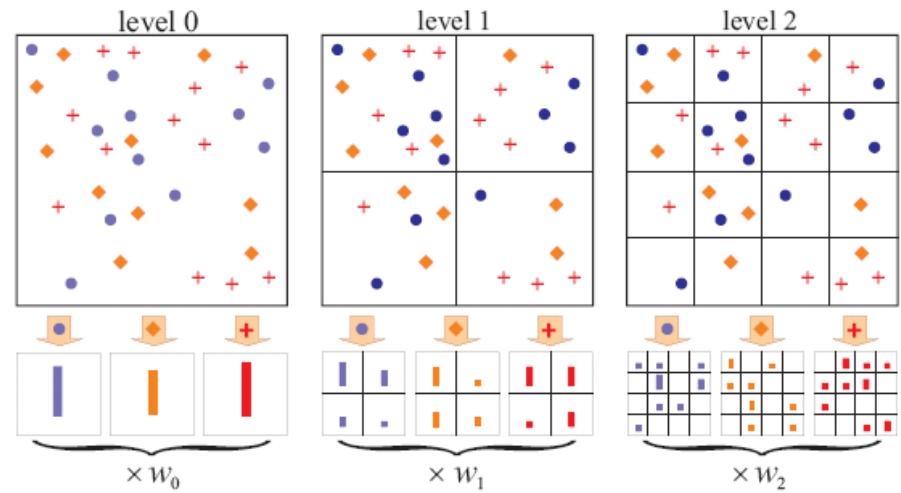
R. Datta, D. Joshi, J. Li, and J. Z. Wang, **Image Retrieval: Ideas, Influences, and Trends of the New Age**,
ACM Computing Surveys. vol. 40. no. 2. pp. 5:1-60. 2008.

Spatially organized textures



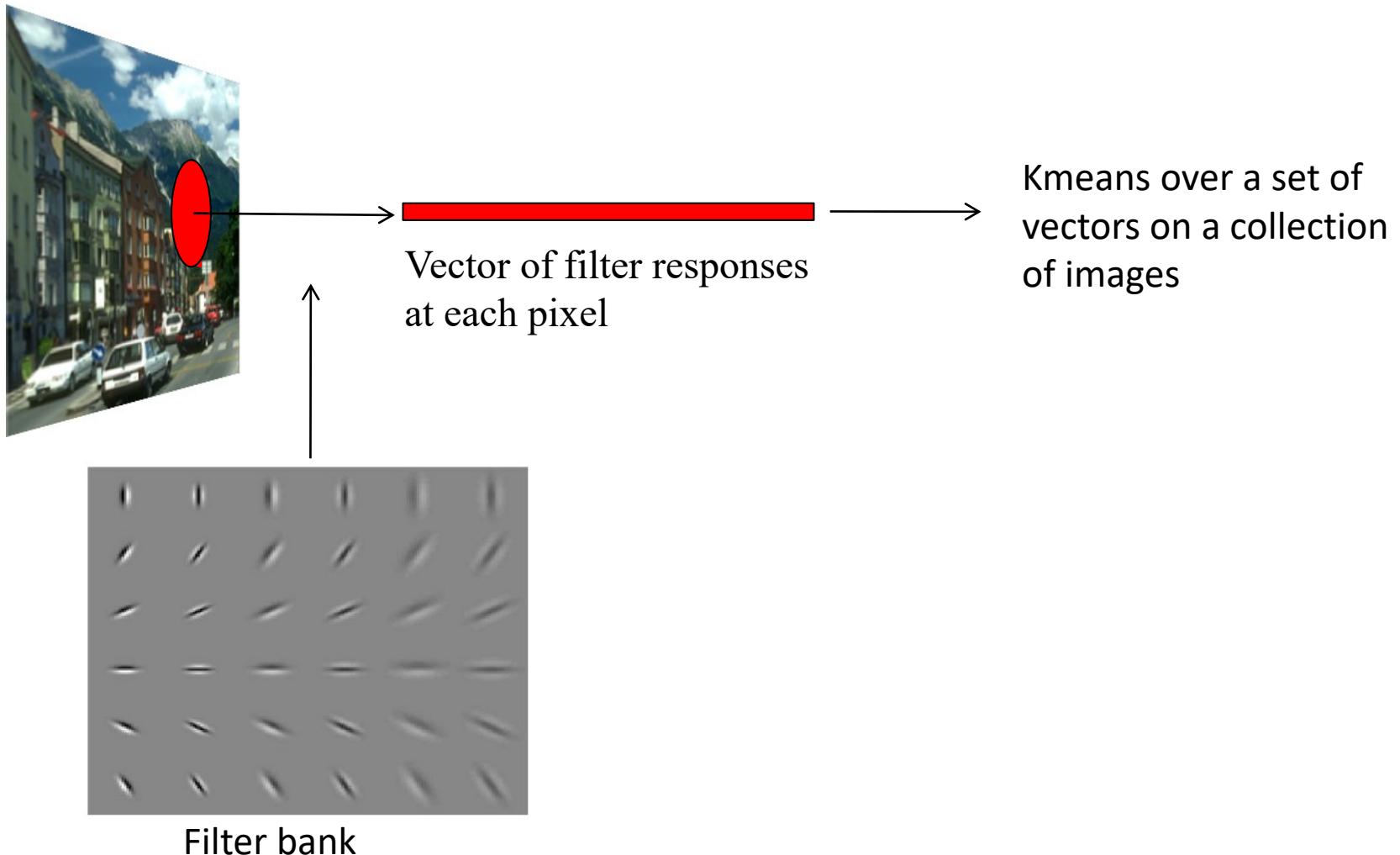
M. Gorkani, R. Picard, ICPR 1994

A. Oliva, A. Torralba, IJCV 2001



S. Lazebnik, et al, CVPR 2006

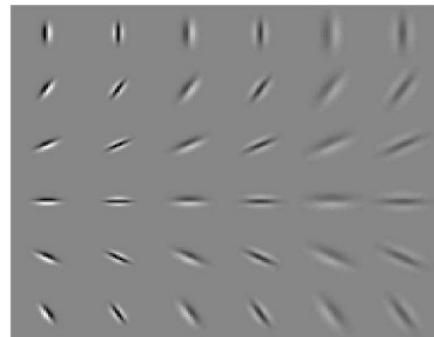
Textons



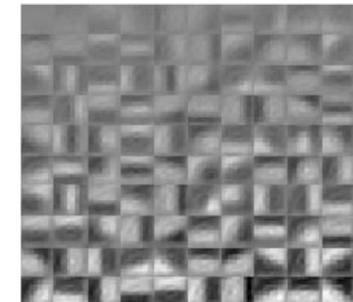
Textons



Filter bank



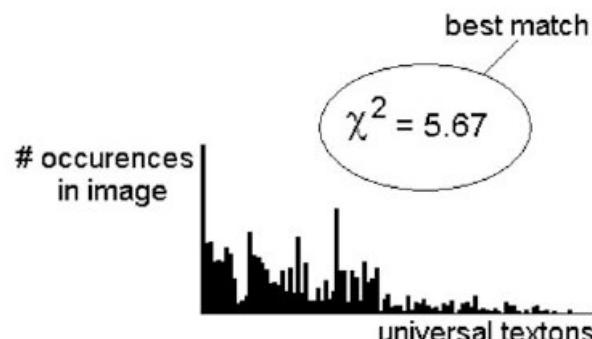
K-means (100 clusters)



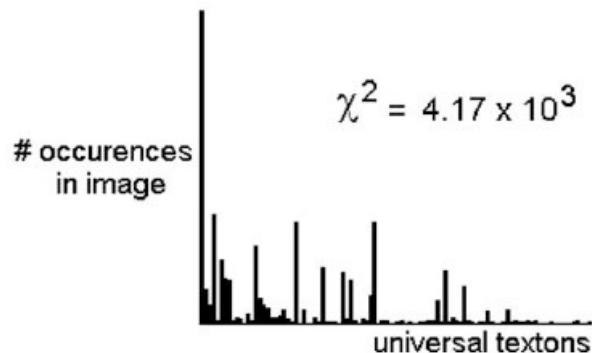
Malik, Belongie, Shi, Leung, 1999



label = bedroom



label = beach



Walker, Malik, 2004



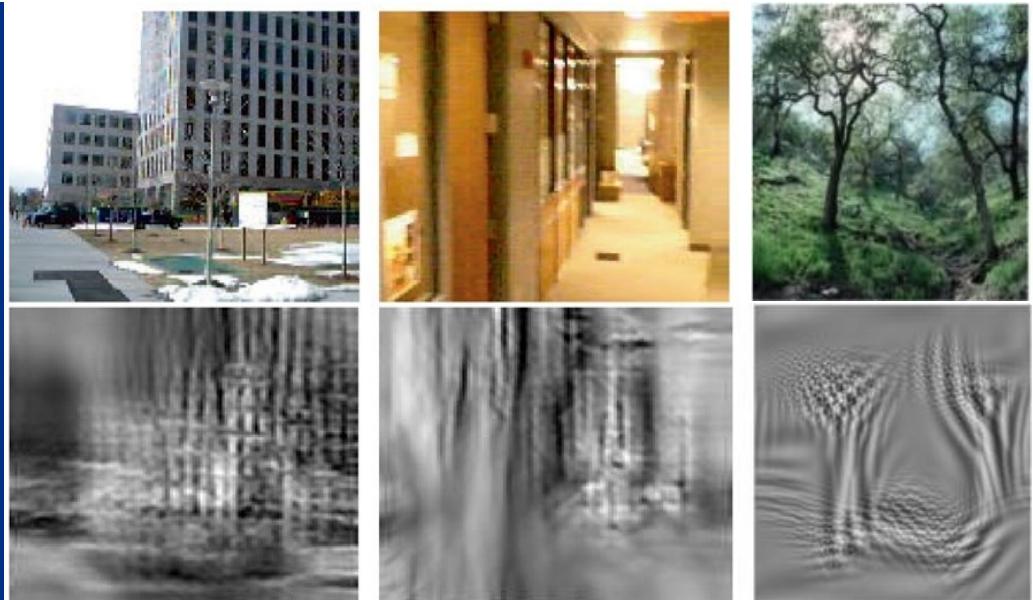
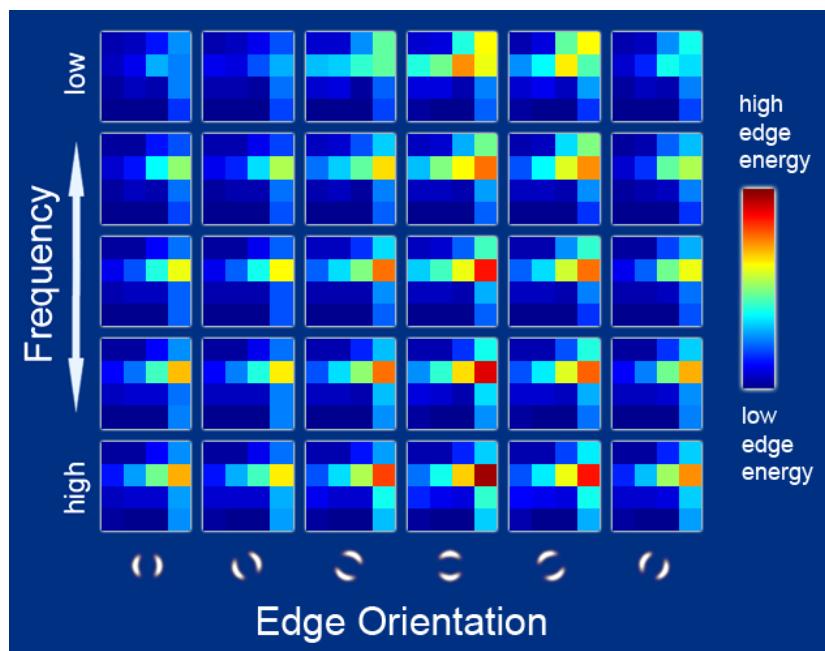
未来媒体研究中心
CENTER FOR FUTURE MEDIA



电子科技大学
University of Electronic Science and Technology of China

Global scene descriptors: GIST

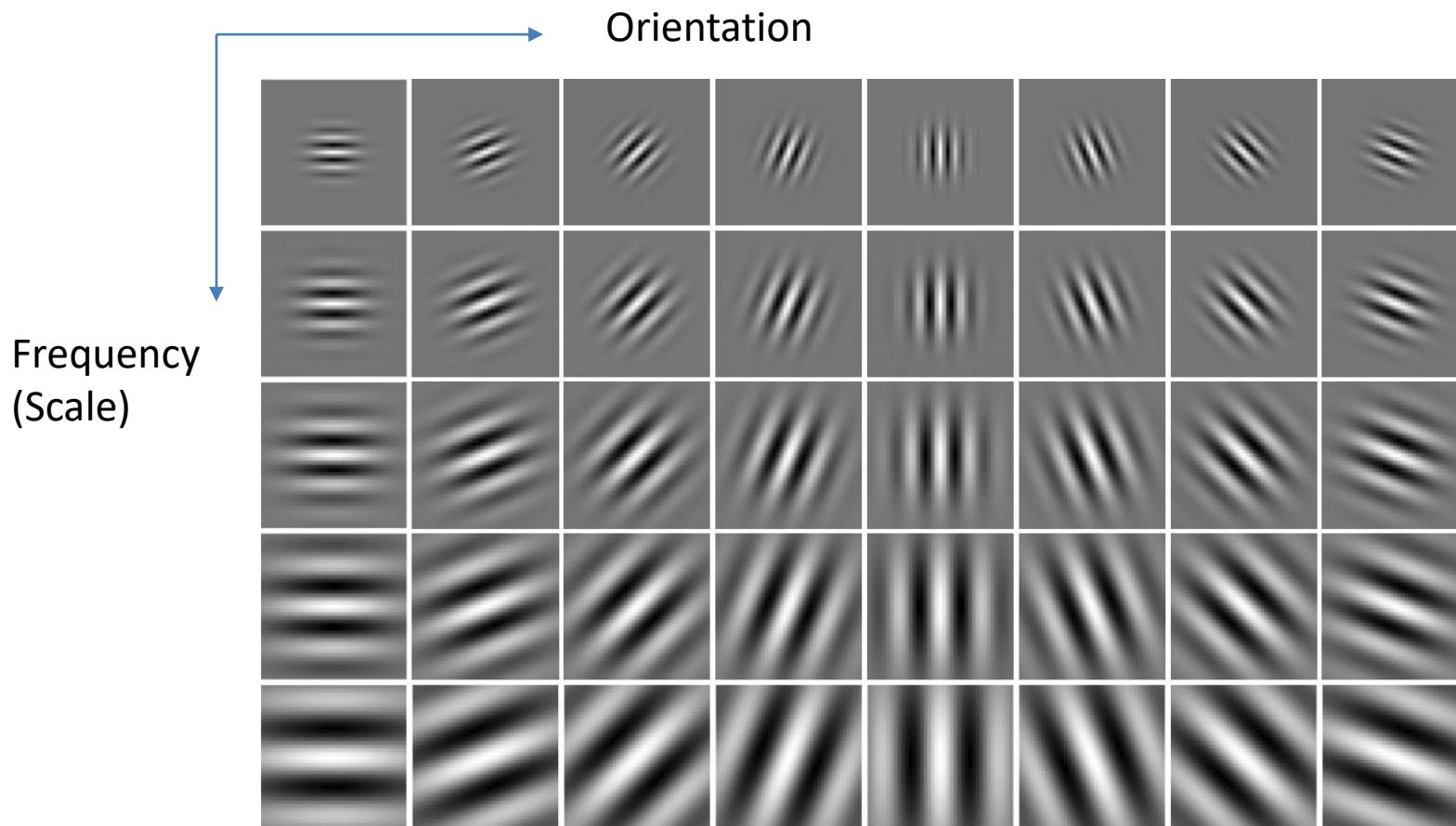
- The “gist” of a scene: Oliva & Torralba (2001)



<http://people.csail.mit.edu/torralba/code/spatialevelope>

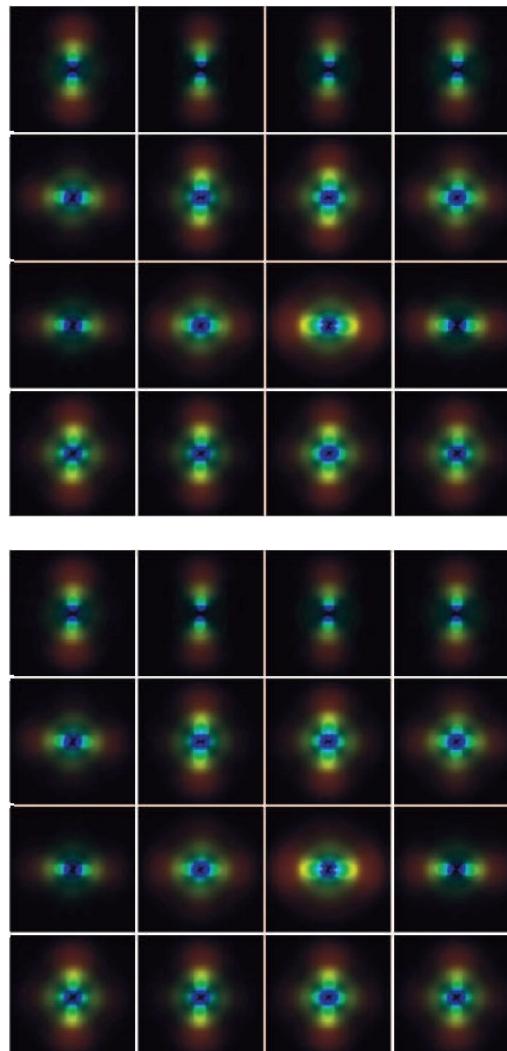
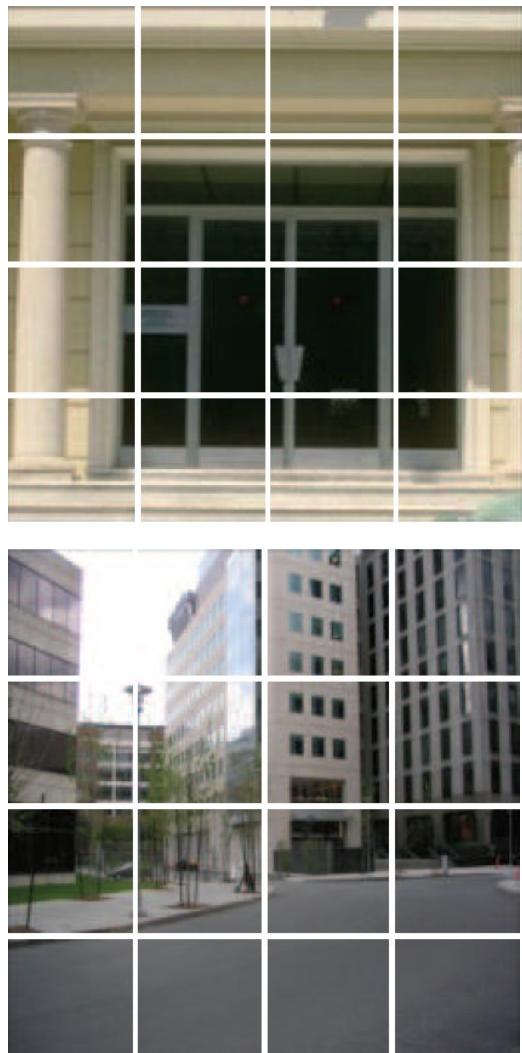
Gabor filter

- Sinusoid modulated by a Gaussian kernel



Gist descriptor

Oliva and Torralba, 2001



Apply oriented Gabor filters over different scales.

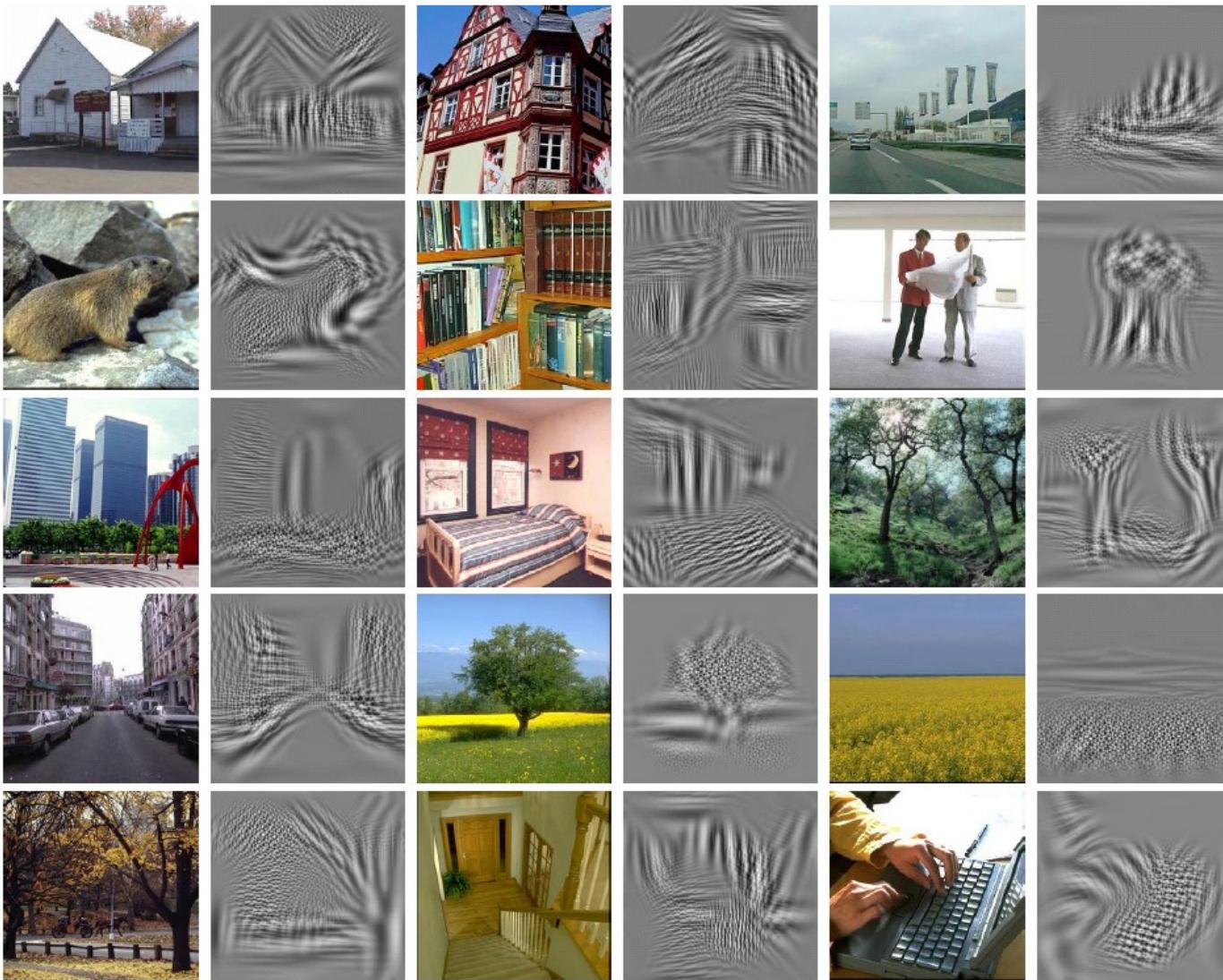
Average filter energy per bin.

Similar to SIFT (Lowe 1999) applied to the entire image.

8 orientations
4 scales
x 16 bins
512 dimensions

M. Gorkani, R. Picard, ICPR 1994; Walker, Malik. Vision Research 2004; Vogel et al. 2004;
Fei-Fei and Perona, CVPR 2005; S. Lazebnik, et al, CVPR 2006; ...

Example visual gists



Global features (I) ~ global features (I')



未来媒体研究中心
CENTER FOR FUTURE MEDIA



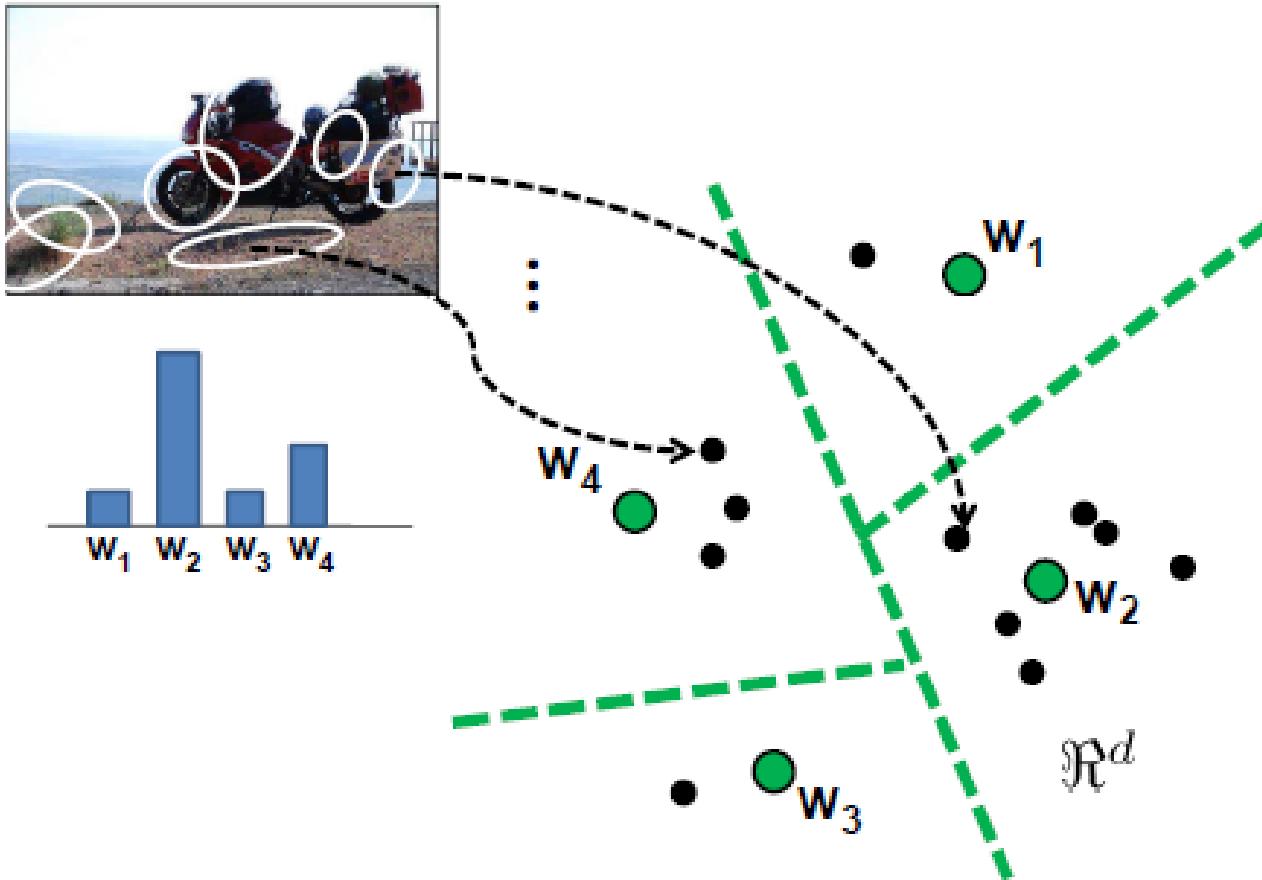
电子科技大学
University of Electronic Science and Technology of China

Oliva & Torralba (2001)

Better Bags of Visual Features

- More advanced quantization / encoding methods that are near the state-of-the-art in image classification and image retrieval.
 - Mixtures of Gaussians
 - Soft assignment (a.k.a. Kernel Codebook)
 - VLAD – Vectors of Locally-Aggregated Descriptors
- Deep learning has taken attention away from these methods...

Standard Kmeans Bag of Words

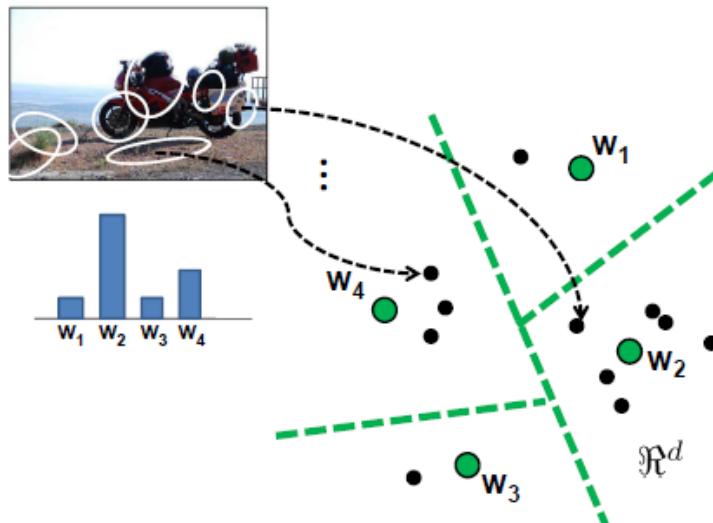


http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

Motivation

- *Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region

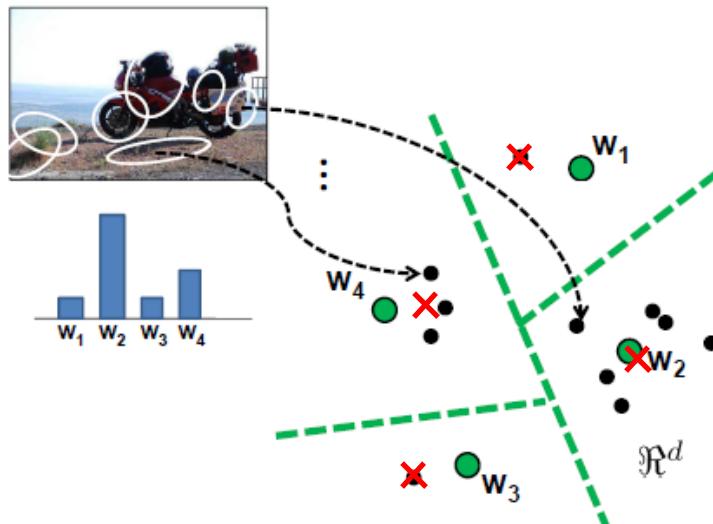
- Why not instead use **statistics**?



Motivation

- *Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region

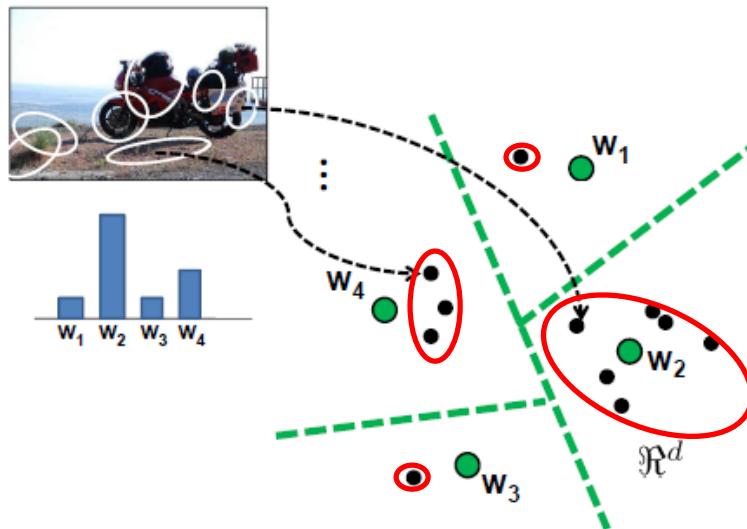
- Why not instead use **statistics**? For instance:
 - mean c



Motivation

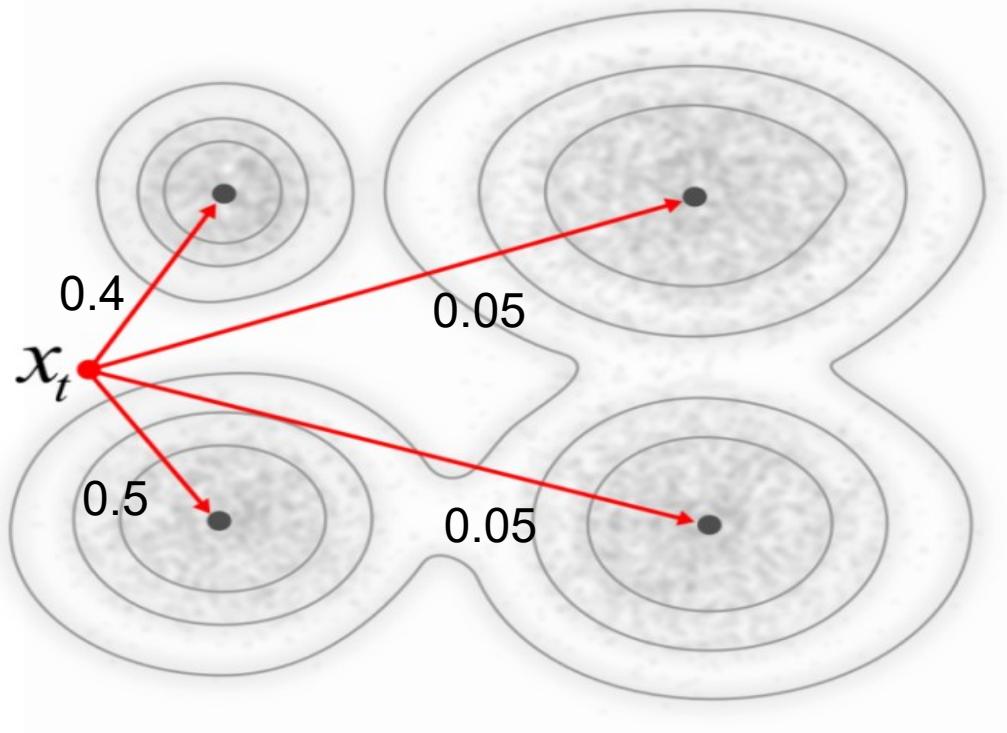
- *Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region

- Why not include **statistics**? For instance:
 - mean c
 - (co)var



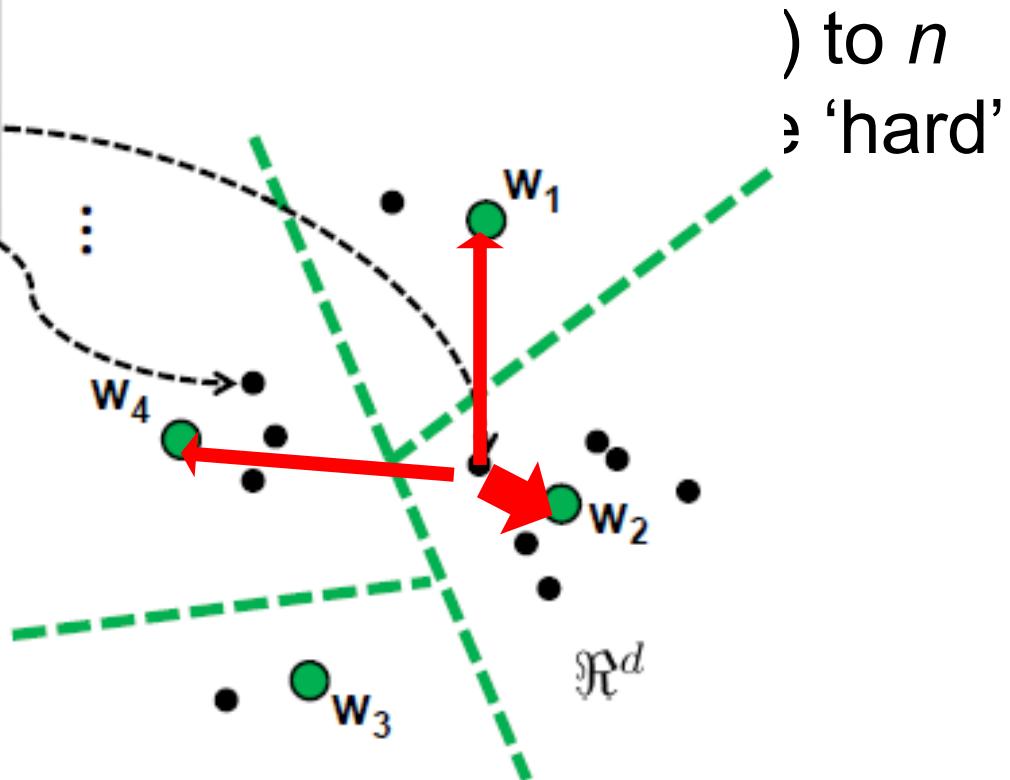
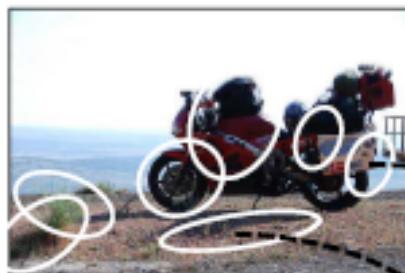
Mixture of Gaussians (GMM)

- GMM can be thought of as “soft” kmeans.
- Each component has a mean and a standard deviation along each direction (or full cov)
- Can easily model distributions



Simple case: Soft Assignment

- “Kernel codebook encoding” by Chatfield et al. 2011.
- Cast mos vote \rightarrow \mathbb{R}^d) to n ‘hard’



Simple case: Soft Assignment

- “Kernel codebook encoding” by Chatfield et al. 2011.
- Cast a set of proportional votes (weights) to n most similar clusters, rather than a single ‘hard’ vote.
- This is fast and easy to implement using an inverted file index / less



New query image

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

kes

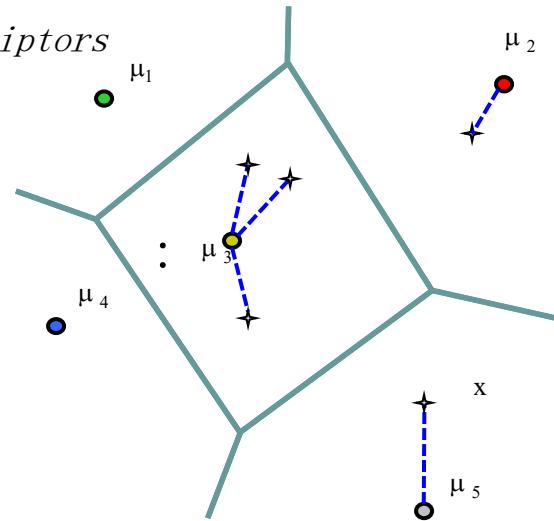
VLAD - Vectors of Locally-Aggregated Descriptors

Given a codebook $\{\mu_i, i = 1 \dots N\}$,
e.g. learned with K-means, and a set of
local descriptors $X = \{x_t, t = 1 \dots T\}$

- ① assignNN(x_t) = $\arg \min_{\mu_i} \|x_t - \mu_i\|$

①

assign descriptors



VLAD - Vectors of Locally-Aggregated Descriptors

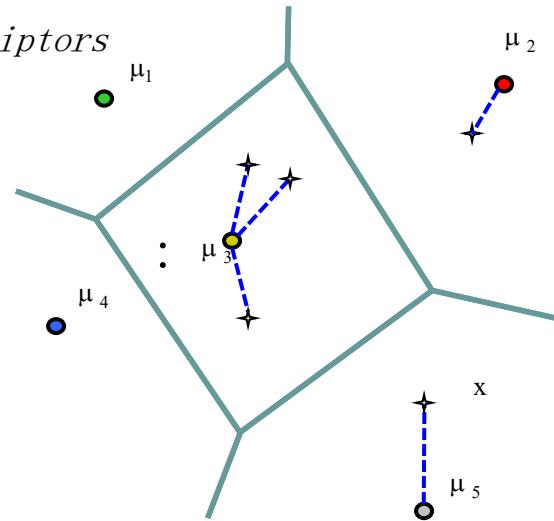
Given a codebook $\{\mu_i, i = 1 \dots N\}$,
e.g. learned with K-means, and a set of
local descriptors $X = \{x_t, t = 1 \dots T\}$

- ① assignNN(x_t) = $\arg \min_{\mu_i} \|x_t - \mu_i\|$

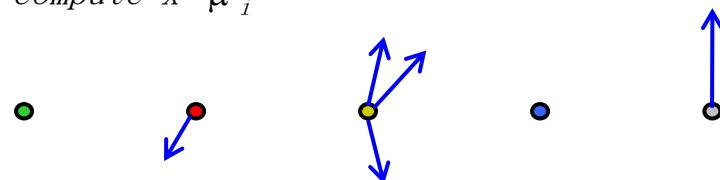
- ②③ compute: $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$

- ④

① assign descriptors



② compute $x - \mu_i$



VLAD - Vectors of Locally-Aggregated Descriptors

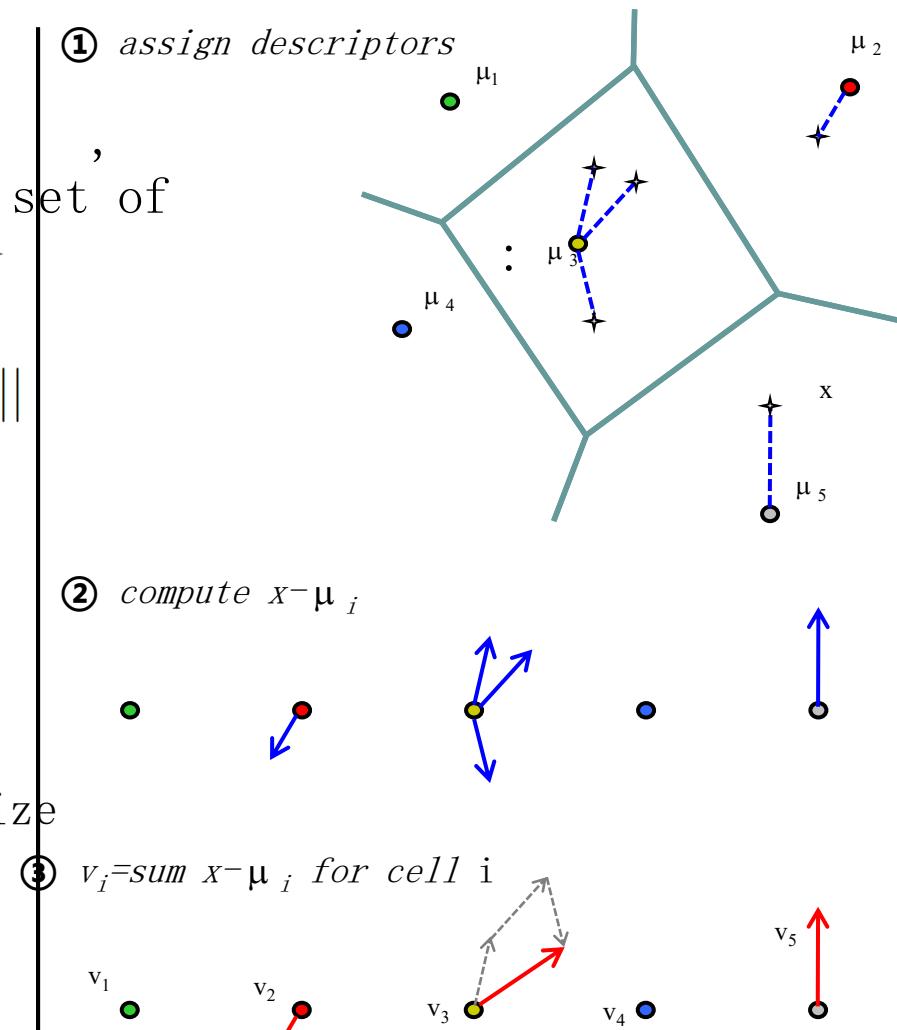
Given a codebook $\{\mu_i, i = 1 \dots N\}$,
e.g. learned with K-means, and a set of
local descriptors $X = \{x_t, t = 1 \dots T\}$

- ① assignNN(x_t) = $\arg \min_{\mu_i} \|x_t - \mu_i\|$

- ②③ compute: $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$

- concatenate v_i , ℓ_2^+

normalize



A first example: the VLAD

A graphical representation $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$



Jégou, Douze, Schmid and Pérez, “Aggregating local descriptors into a compact image representation” CVPR’10.

Summary

- **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
- **Evaluation:**
- **Additional spatial verification alignment:**
 - Robust fitting : RANSAC, Generalized Hough Transform
 - We will do this in detail later on in the course

Lessons from a decade later

For *Category* recognition (project 3)

- 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.