

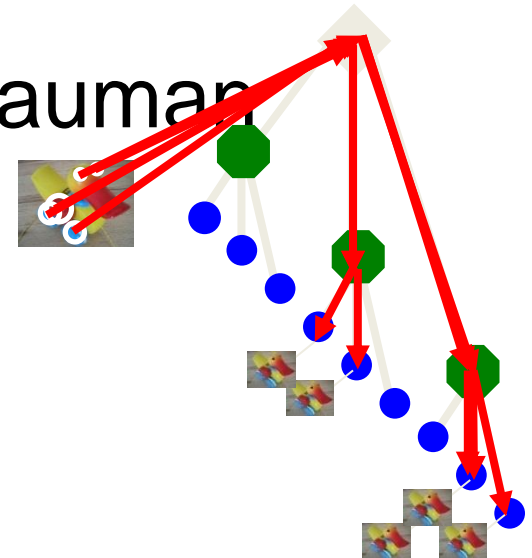
# T14 - Indexing local features

Computer Vision, FCUP, 2013

Miguel Coimbra

Slides by Prof. Kristen Grauman

Index	
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142
511 Traffic Information; 83	Ca d'Zan; 147
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152
AAA (and CAA); 83	Name; 150
AAA National Office; 88	Canaveral Natnl Seashore; 173
Abbreviations,	Cannon Creek Airpark; 130
Colored 25 mile Maps; cover	Canopy Road; 106,169
Exit Services; 196	Cape Canaveral; 174
Travelogue; 85	Castillo San Marcos; 169
Africa; 177	Cave Diving; 131
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93
Air Conditioning, First; 112	Charlotte County; 149
	Charlotte Harbor; 150
	Chautauqua; 116
	Chipley; 114



# Matching local features



# Matching local features



Image 1



Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest  $k$ , or within a thresholded distance)

# Matching local features



Image 1

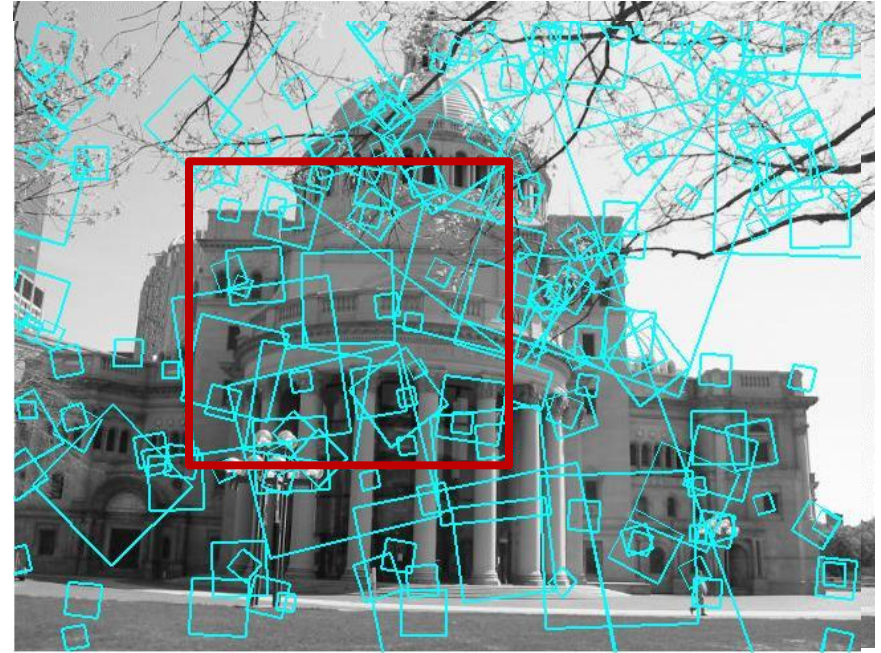
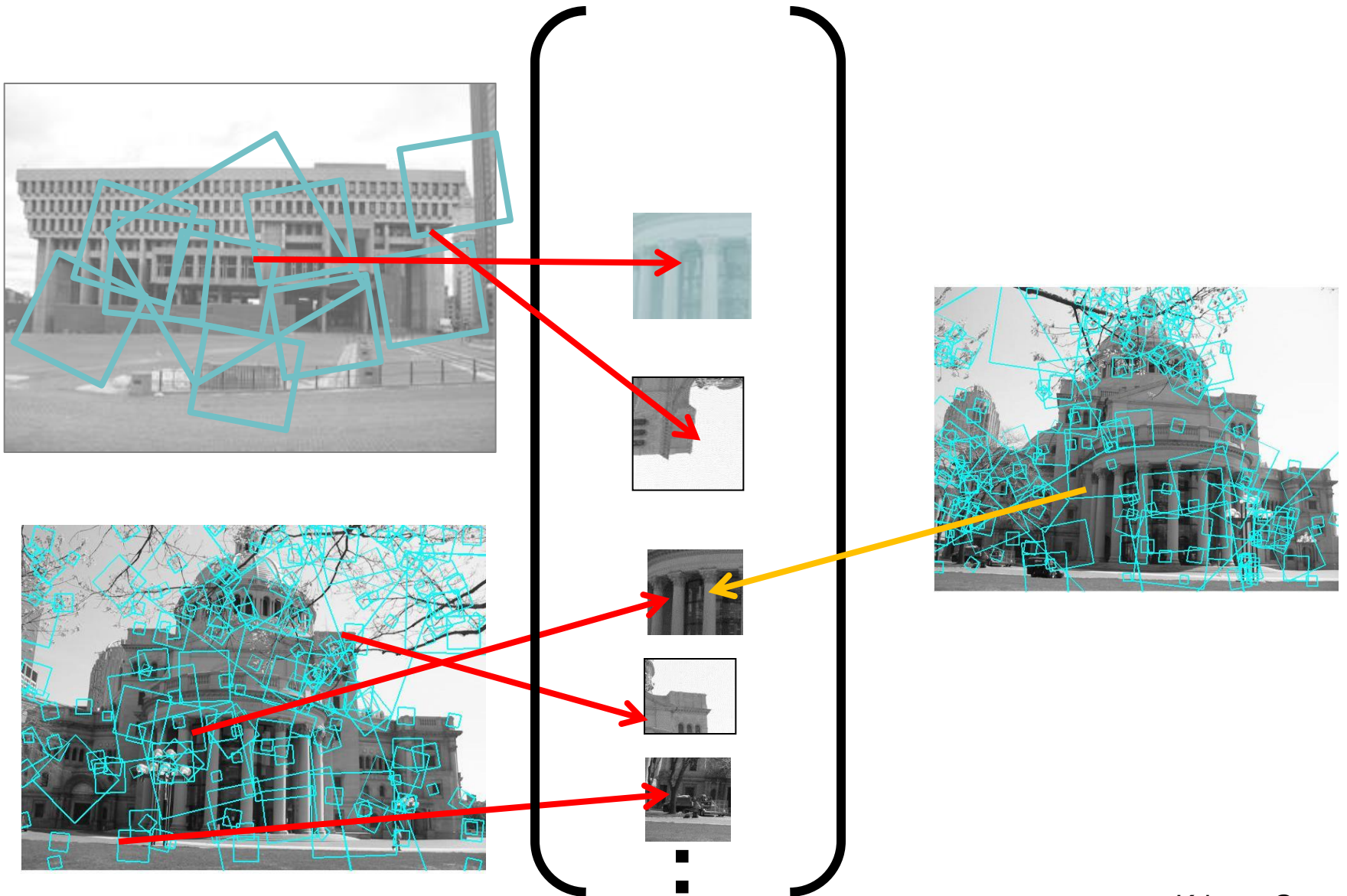


Image 2

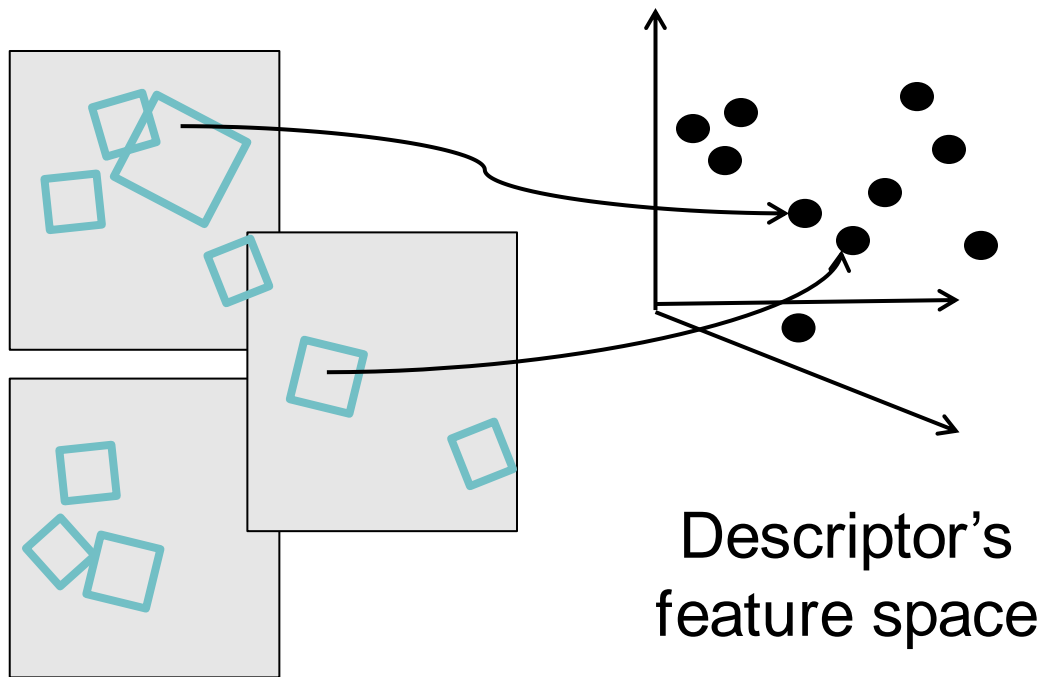
In stereo case, may constrain by proximity if we make assumptions on max disparities.

# Indexing local features



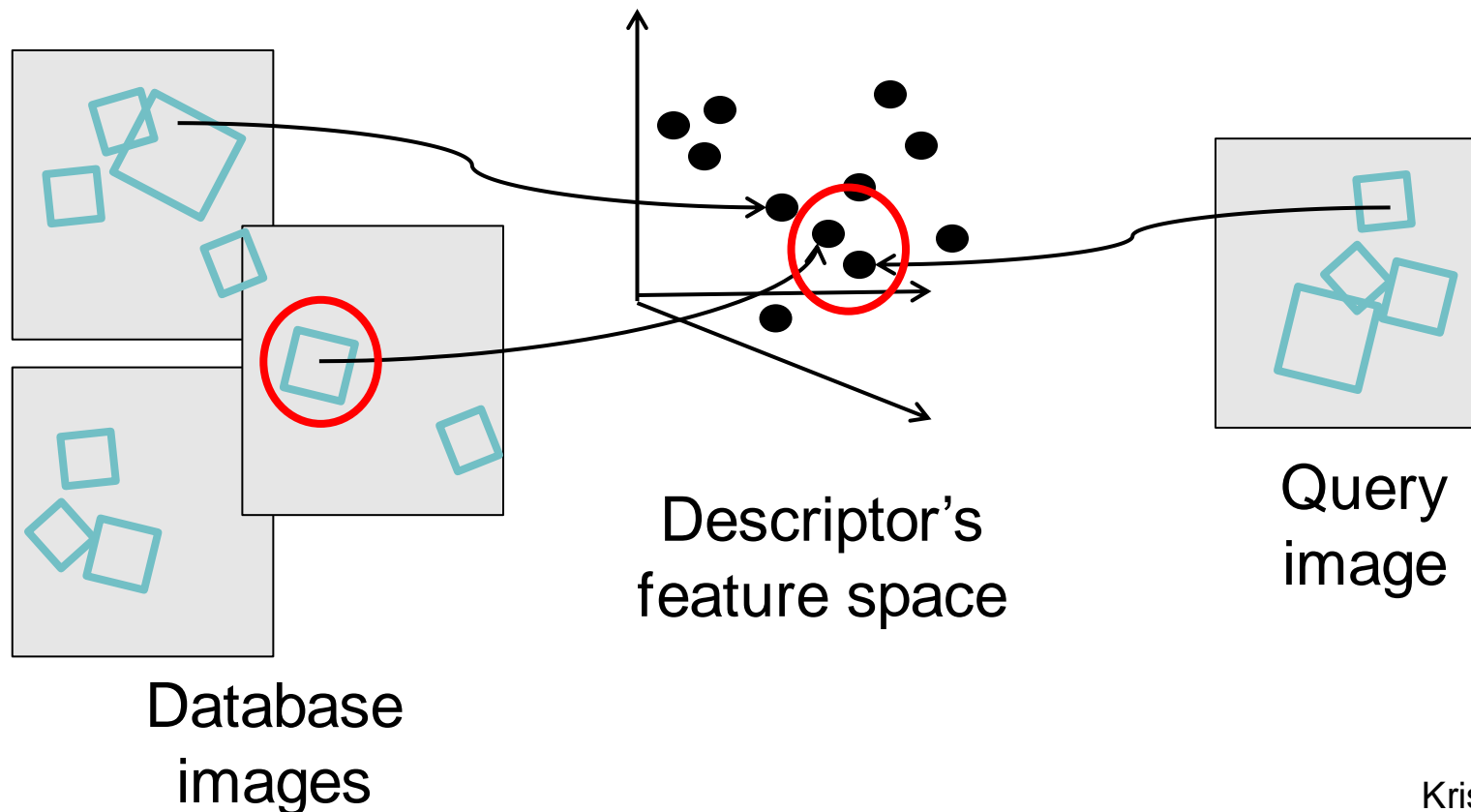
# Indexing local features

- 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.



# Indexing local features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?



# Indexing local features: inverted file index

Index	
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142
511 Traffic Information; 83	Ca d'Zan; 147
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152
AAA (and CAA); 83	Name; 150
AAA National Office; 88	Canaveral Natnl Seashore; 173
Abbreviations,	Cannon Creek Airpark; 130
Colored 25 mile Maps; cover	Canopy Road; 106,169
Exit Services; 196	Cape Canaveral; 174
Travelogue; 85	Castillo San Marcos; 169
Africa; 177	Cave Diving; 131
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93
Air Conditioning, First; 112	Charlotte County; 149
Alabama; 124	Charlotte Harbor; 150
Alachua; 132	Chautauqua; 116
County; 131	ChIPLEY; 114
Alafia River; 143	Name; 115
Alapaha, Name; 126	Choctawatchee, Name; 115
Alfred B Maclay Gardens; 106	Circus Museum, Ringling; 147
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180
Alligator Hole (definition); 157	City Maps,
Alligator, Buddy; 155	Fl Lauderdale Expwys; 194-195
Alligators; 100,135,138,147,156	Jacksonville; 163
Anastasia Island; 170	Kissimmee Expwys; 192-193
Anhaica; 109-109,146	Miami Expressways; 194-195
Apalachicola River; 112	Orlando Expressways; 192-193
Appleton Mus of Art; 136	Pensacola; 26
Aquifer; 102	Tallahassee; 191
Arabian Nights; 94	Tampa-St. Petersburg; 63
Art Museum, Ringling; 147	St. Augustine; 191
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141
Aucilla River Project; 106	Clearwater Marine Aquarium; 187
Babcock-Web WMA; 151	Collier County; 154
Bahia Mar Marina; 184	Collier, Barron; 152
Baker County; 99	Colonial Spanish Quarters; 168
Barefoot Mailmen; 182	Columbia County; 101,128
Barge Canal; 137	Coquina Building Material; 165
Bee Line Expy; 80	Corkscrew Swamp, Name; 154
Belz Outlet Mall; 89	Cowboys; 95
Bernard Castro; 136	Crab Trap II; 144
Big "I"; 165	Cracker, Florida; 88,95,132
Big Cypress; 155,158	Crosstown Expy; 11,35,98,143
Big Foot Monster; 105	Cuban Bread; 184
Billie Swamp Safari; 160	Dade Battlefield; 140
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161
Blue Angels	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; 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

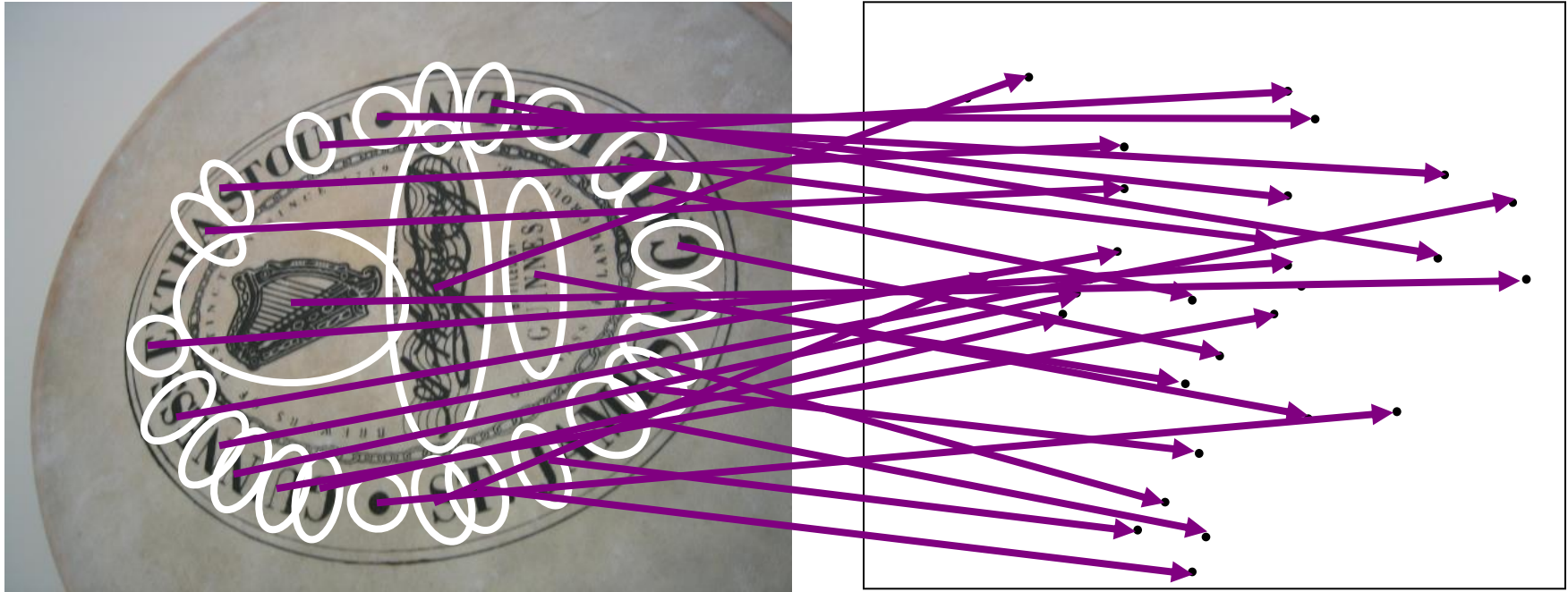
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

# Text retrieval vs. image search

- What makes the problems similar, different?

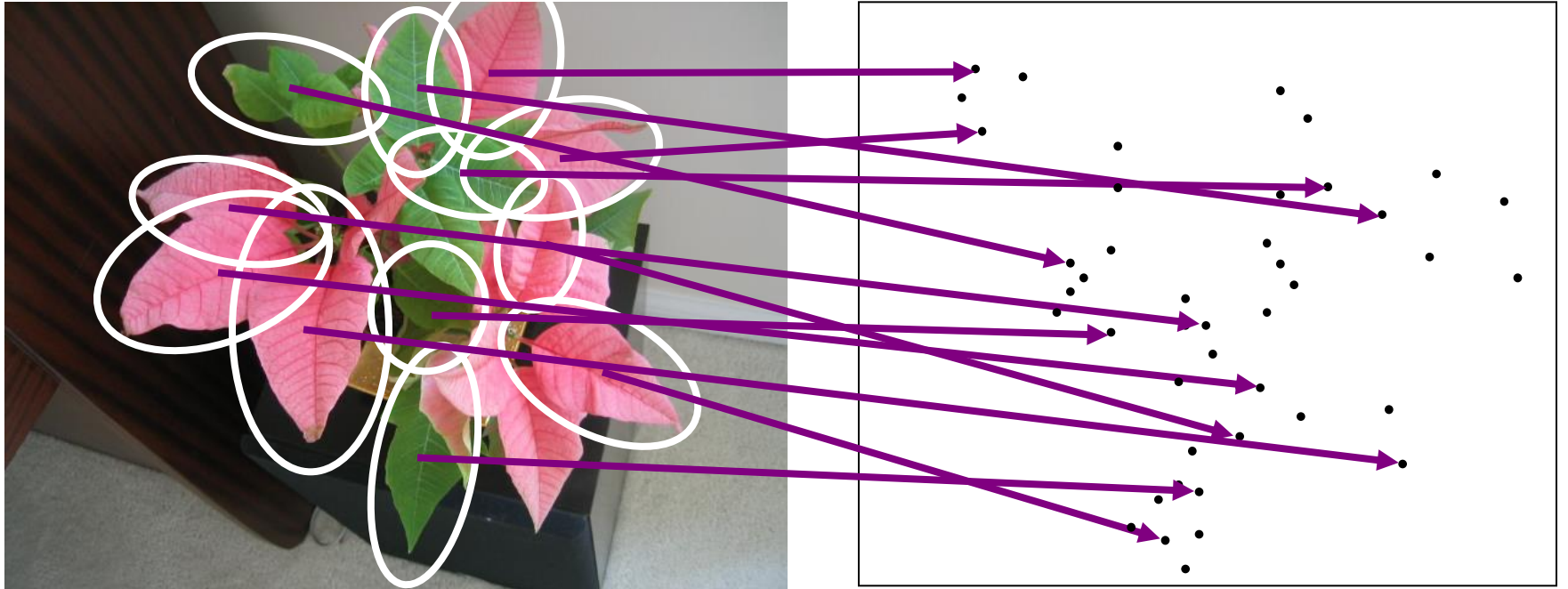
# Visual words: main idea

- Extract some local features from a number of images ...

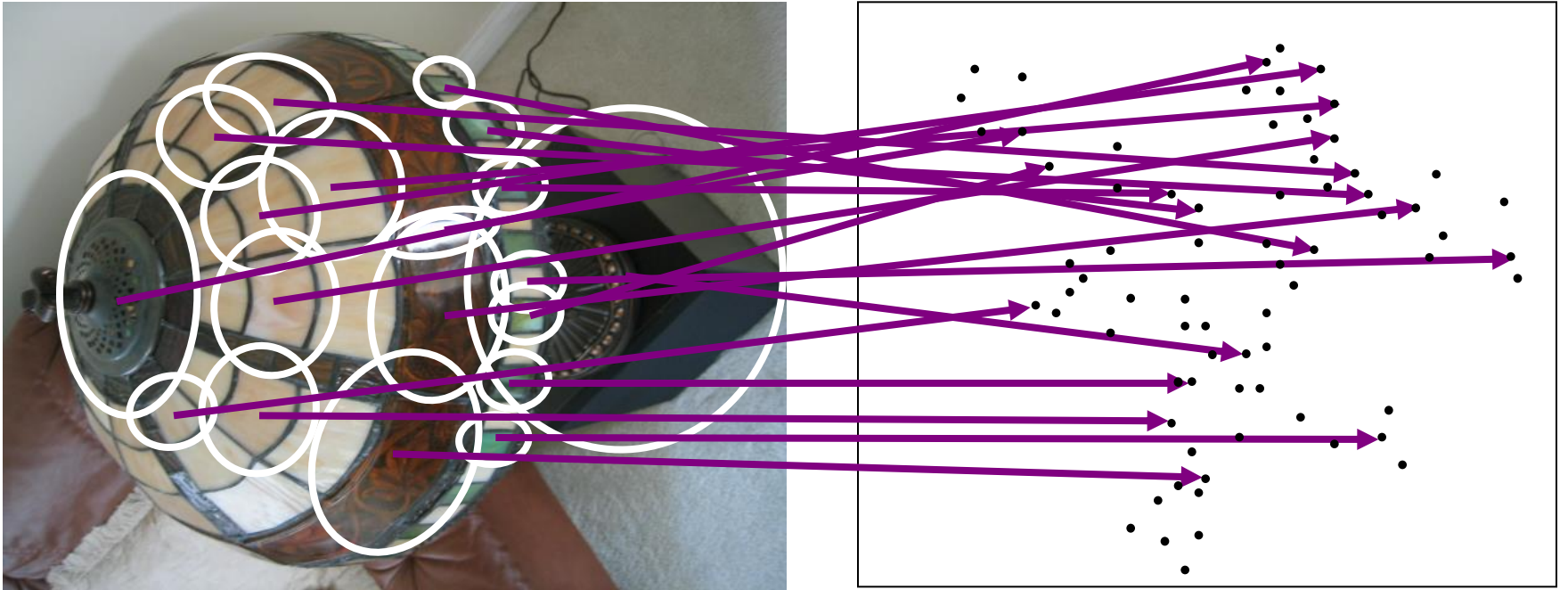


e.g., SIFT descriptor space: each point is 128-dimensional

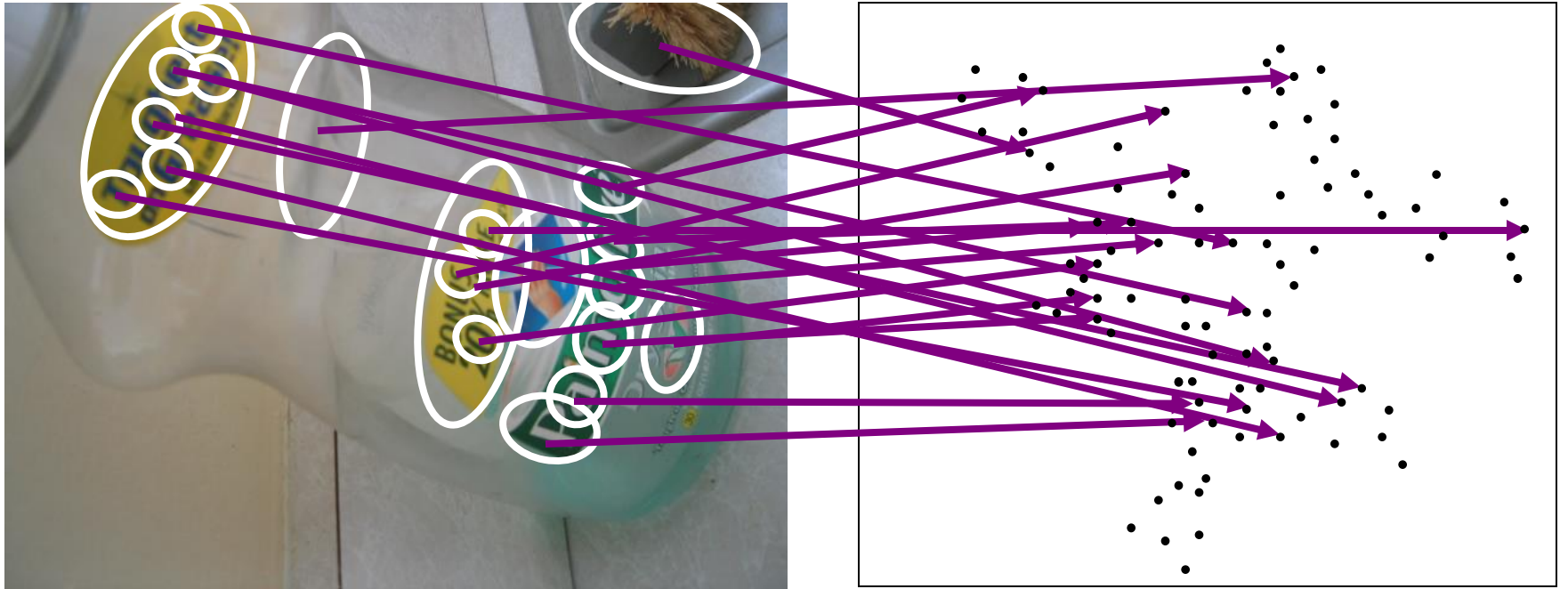
# Visual words: main idea

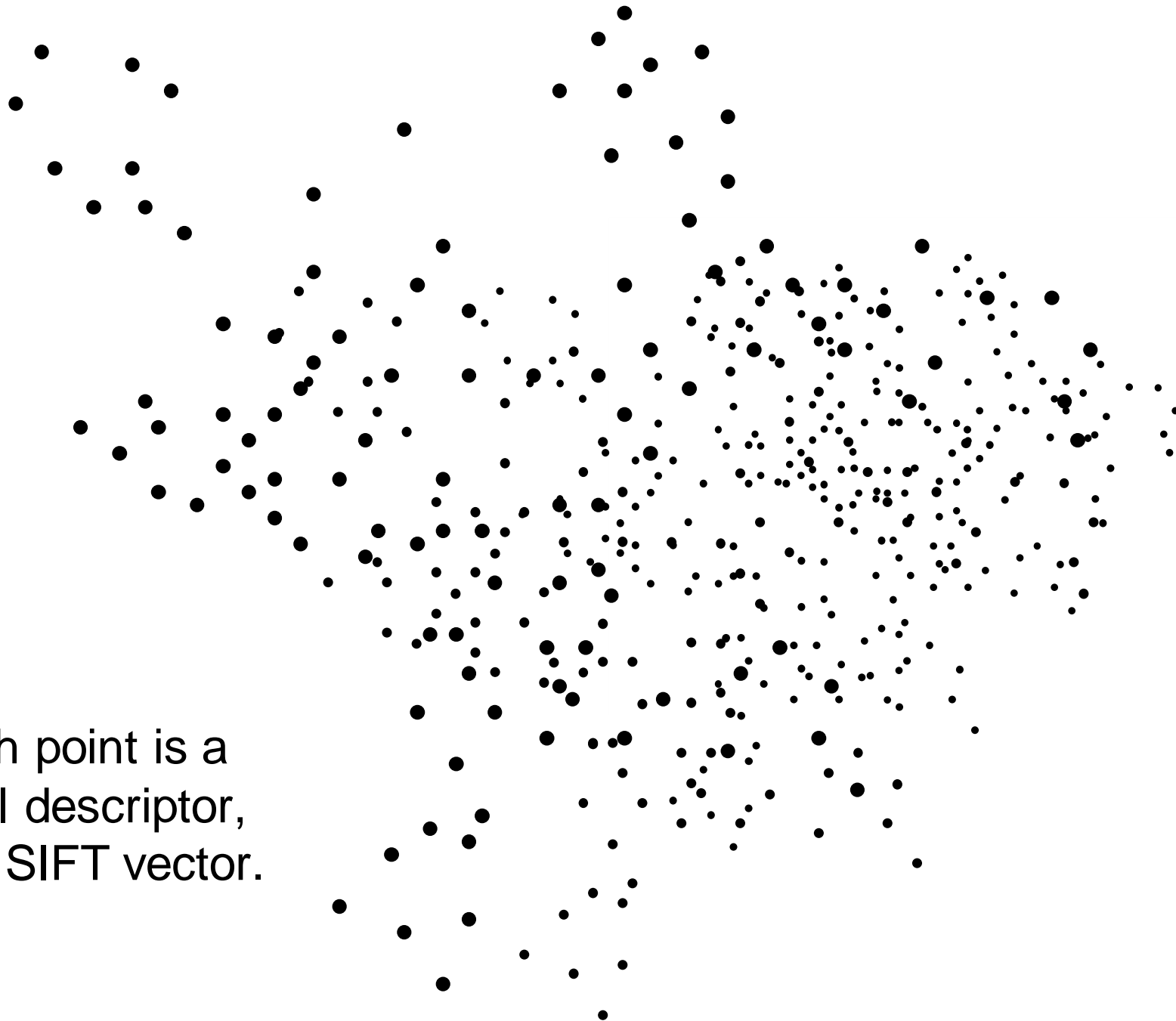


# Visual words: main idea

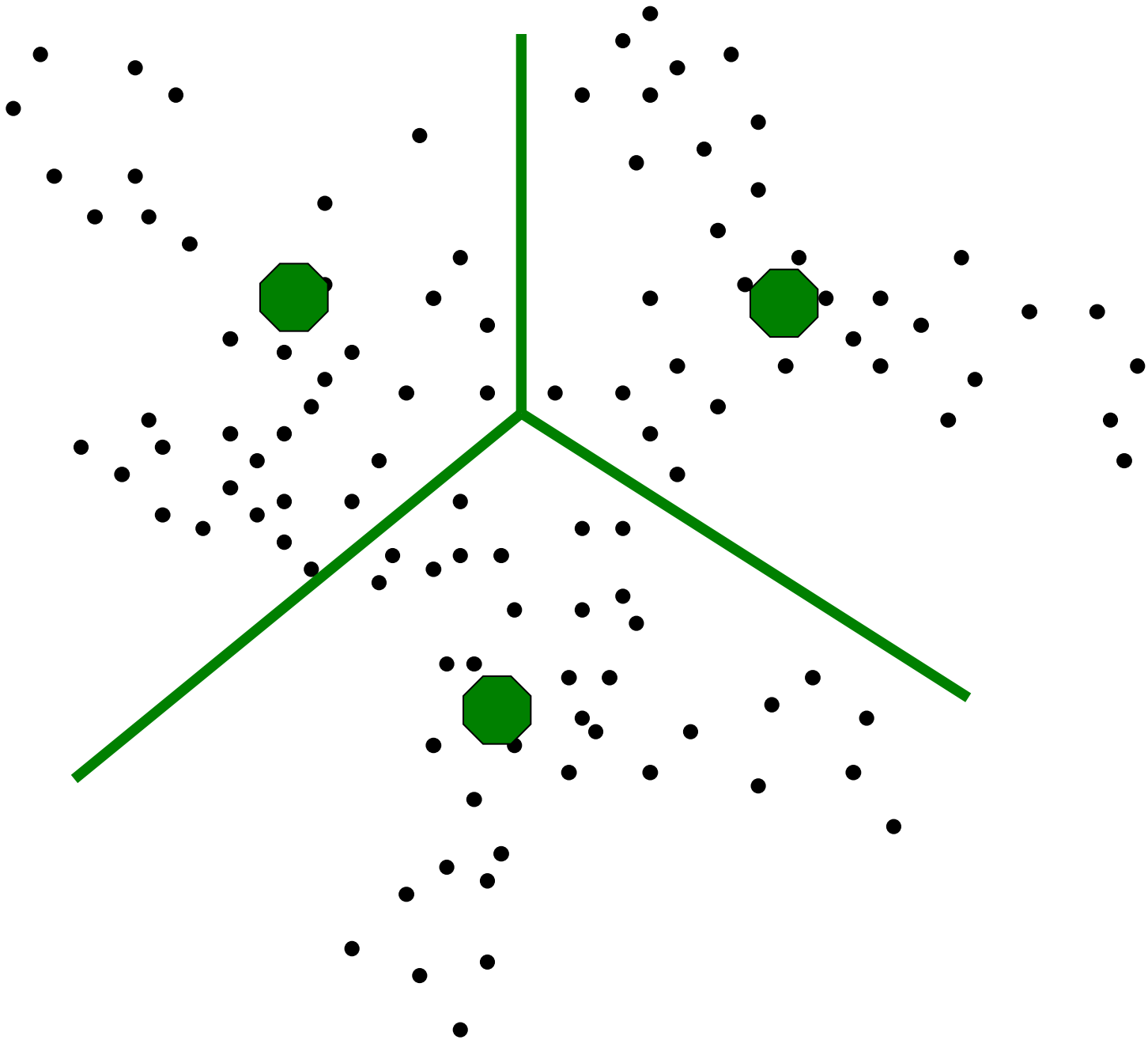


# Visual words: main idea





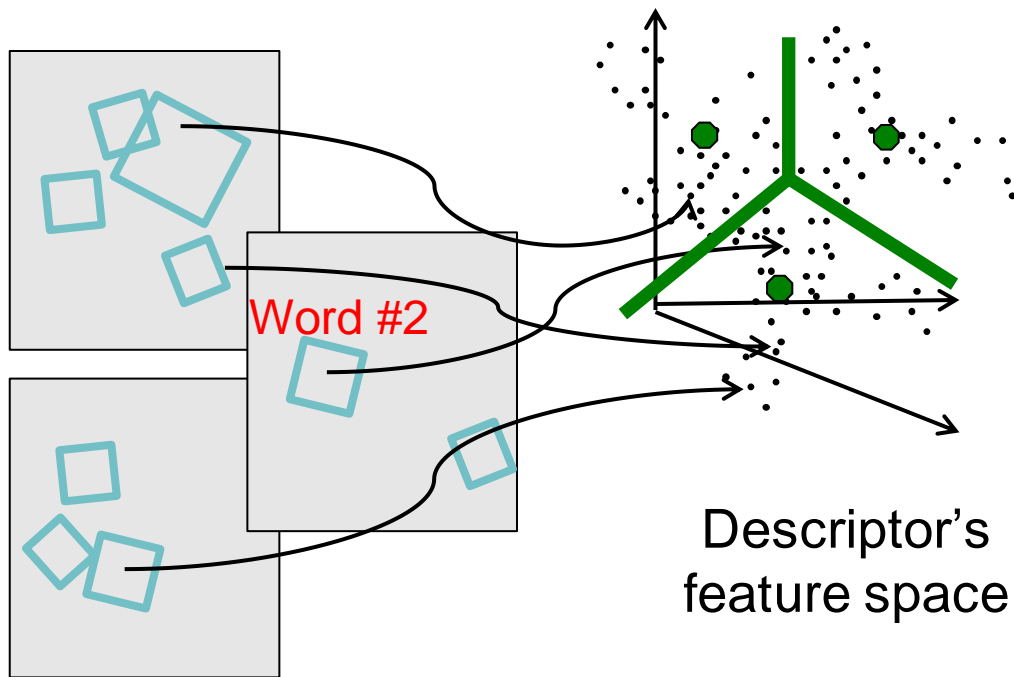
Each point is a  
local descriptor,  
e.g. SIFT vector.





# Visual words

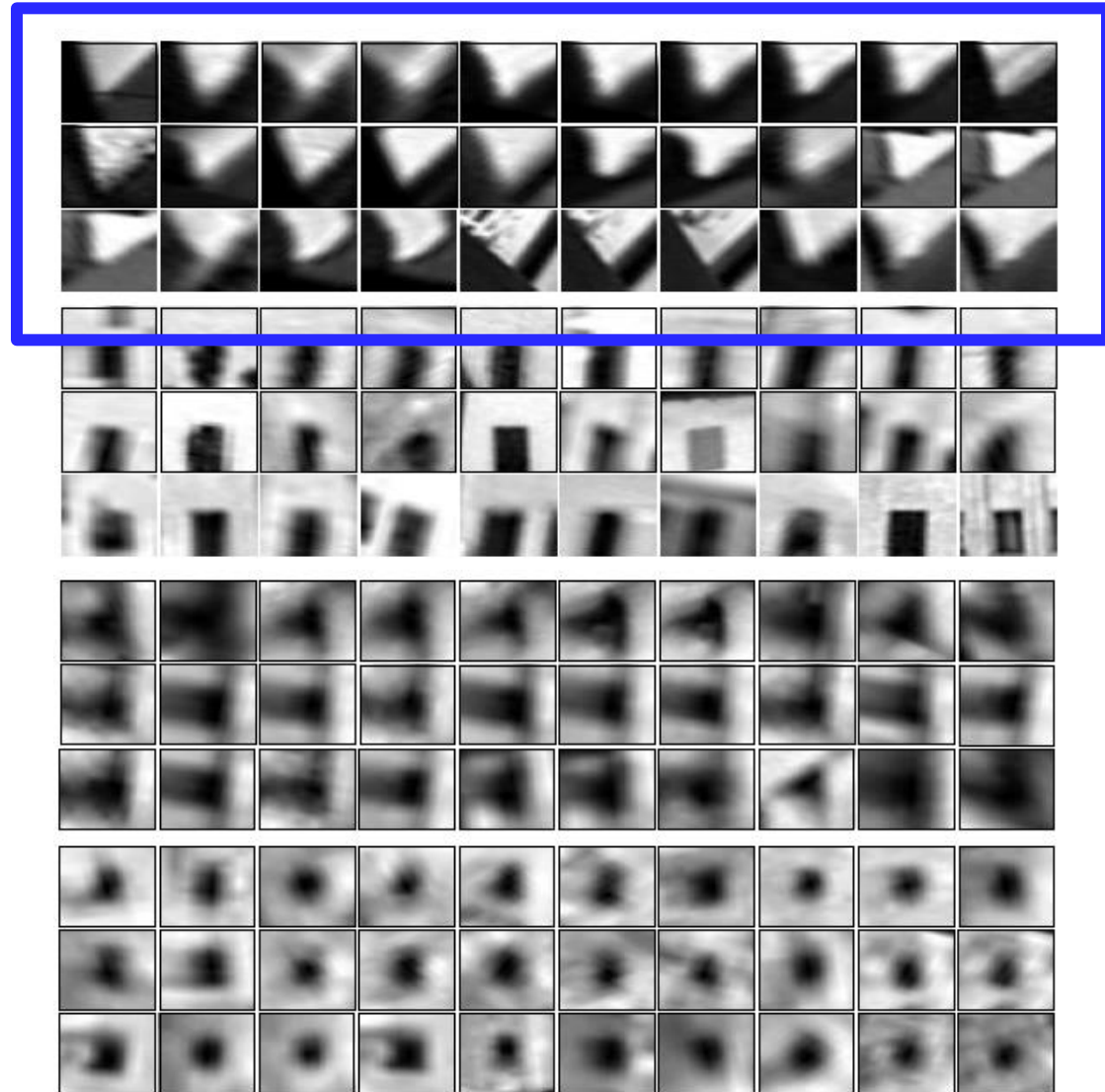
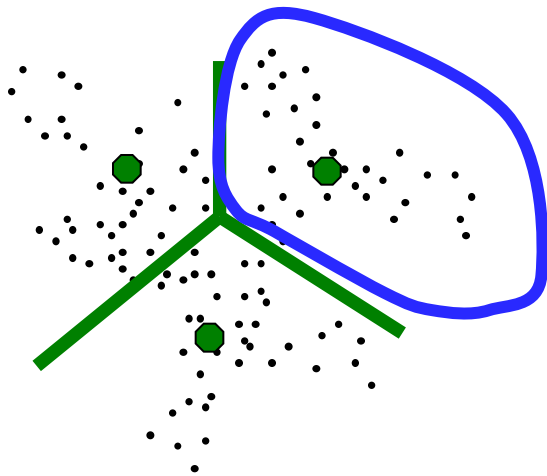
- Map high-dimensional descriptors to tokens/words by quantizing the feature space



- 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.

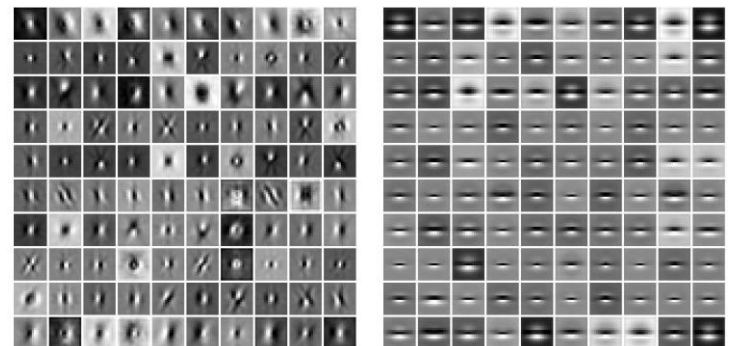
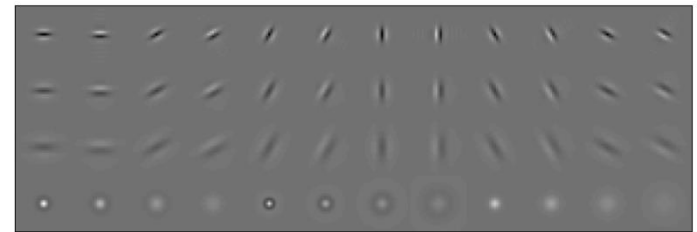
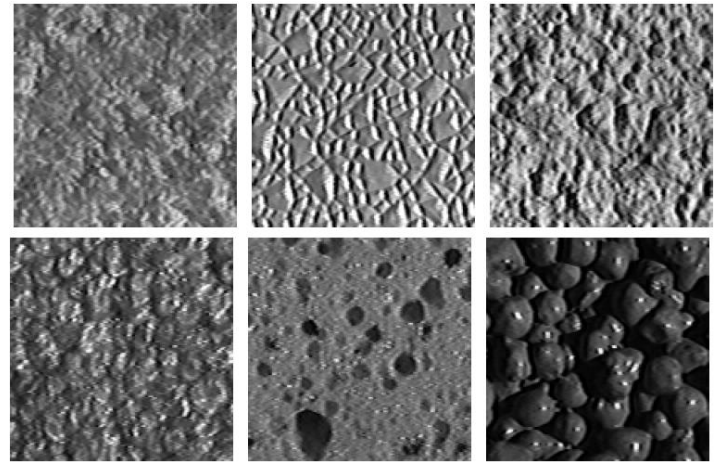
# Visual words

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



# Visual words and textons

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.

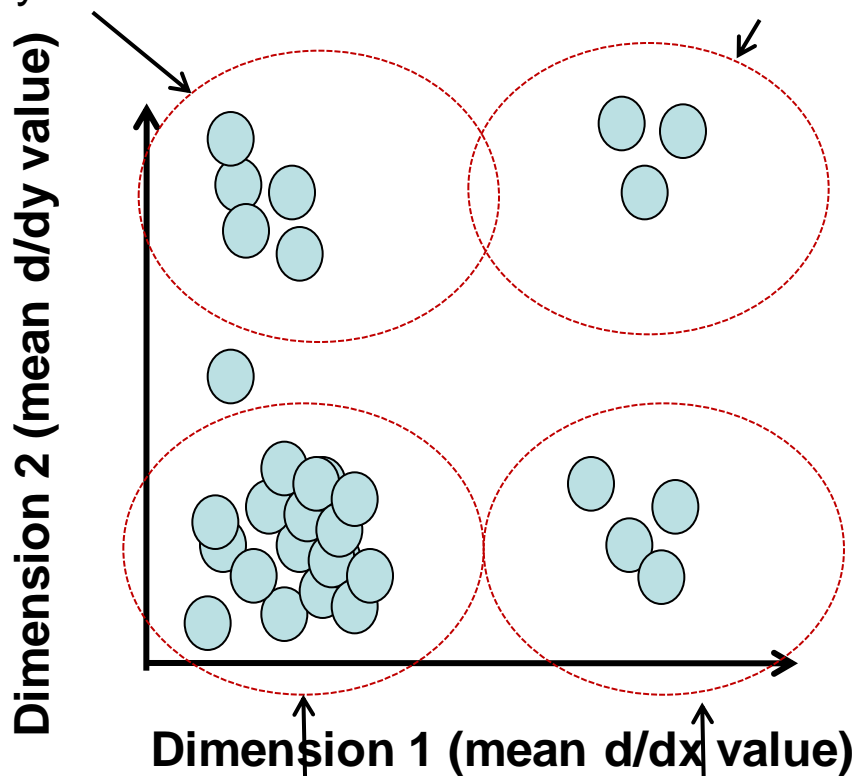


Leung & Malik 1999; Varma & Zisserman, 2002

# Recall: Texture representation example

Windows with primarily horizontal edges

Both



Windows with small gradient in both directions

Windows with primarily vertical edges

	<u>mean d/dx value</u>	<u>mean d/dy value</u>
Win. #1	4	10
Win. #2	18	7
⋮		
Win. #9	20	20

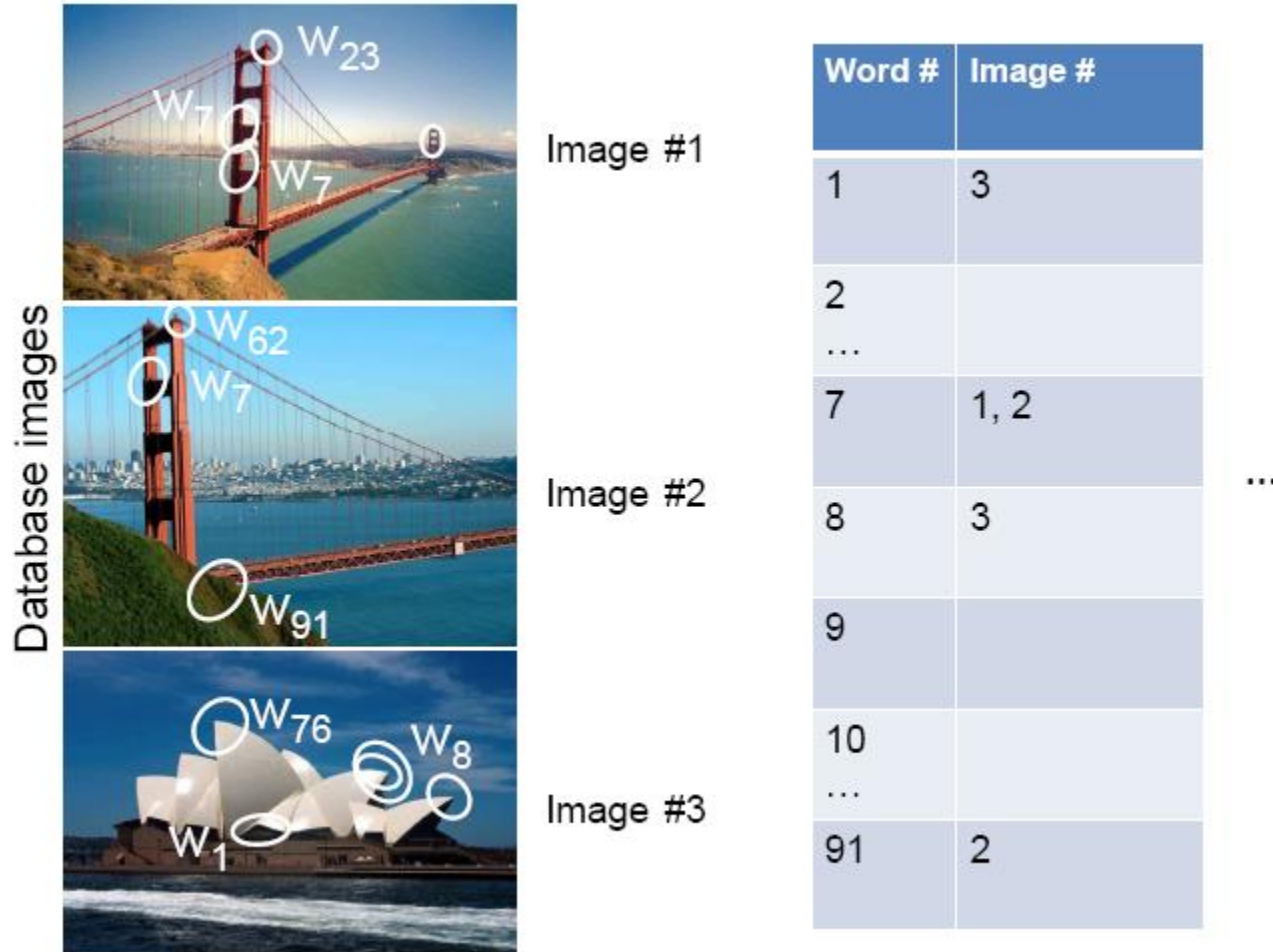
⋮  
statistics to summarize patterns in small windows

# Visual vocabulary formation

## Issues:

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

# Inverted file index



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

# Inverted file index

*When will this give us a significant gain in efficiency?*



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.

- If a local image region is a visual word, how can we summarize an image (the document)?



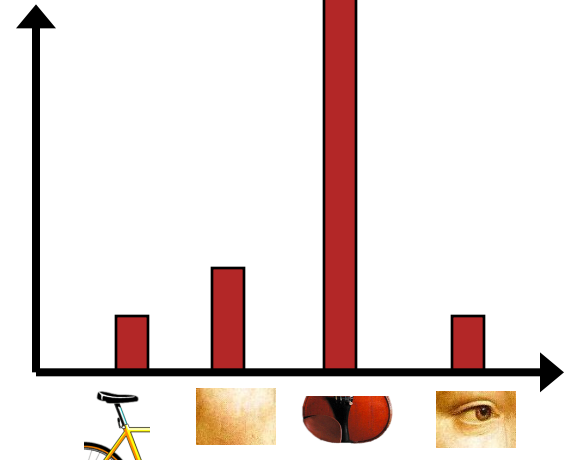
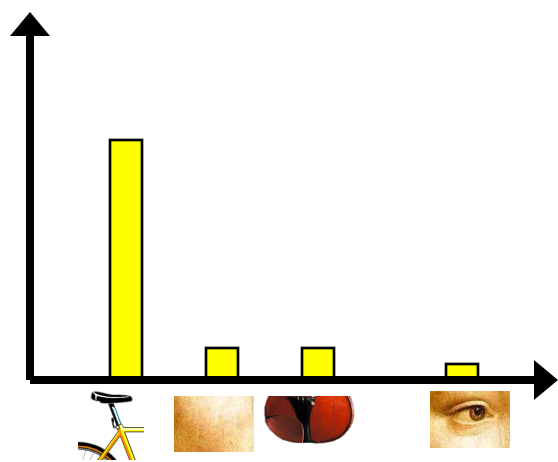
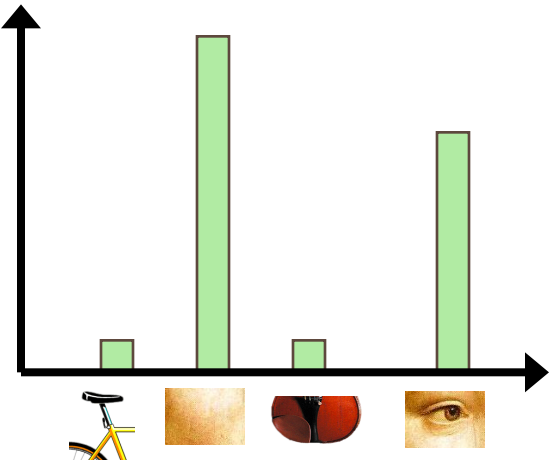
# Analogy to documents

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 our eyes. For a long time, the retinal image was considered as a movie screen. It is now known that the image undergoes a complex analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.

**sensory, brain,  
visual, perception,  
retinal, cerebral cortex,  
eye, cell, optical  
nerve, image  
Hubel, Wiesel**

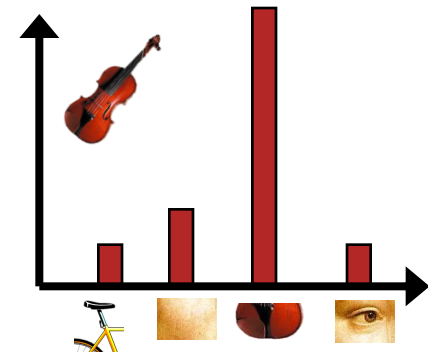
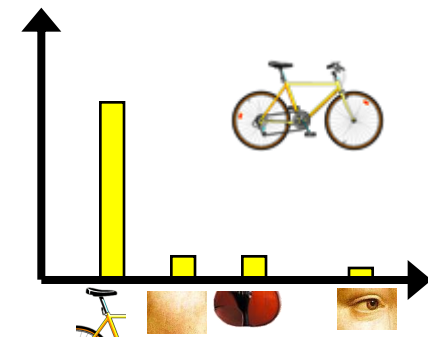
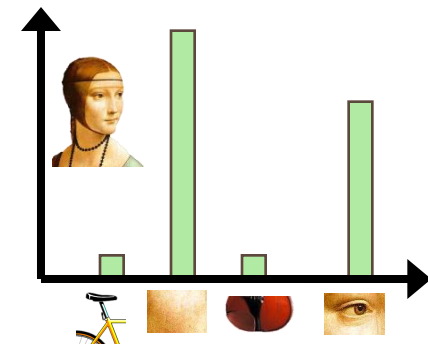
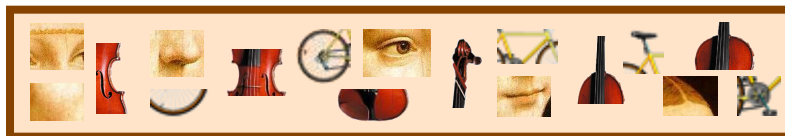
China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$570bn in 2004. The surplus of \$660bn. The surplus will annoy the US. China's government has deliberately agreed to keep the yuan is fixed against the dollar. The government also needs to control the demand so that it does not become a country. China has kept the yuan against the dollar fixed 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.

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



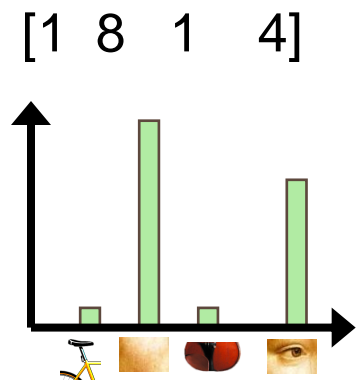
# Bags of visual words

- 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.



$\vec{d}_j$



$\vec{q}$

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

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

for vocabulary of  $V$  words

# *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)

The diagram shows the formula  $t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$  with arrows pointing from descriptive text to its components. The text 'Number of occurrences of word i in document d' points to the numerator  $n_{id}$ . The text 'Number of words in document d' points to the denominator  $n_d$ . The text 'Total number of documents in database' points to the numerator  $N$  of the log term. The text 'Number of documents word i occurs in, in whole database' points to the denominator  $n_i$  of the log term.

Number of occurrences of word  $i$  in document  $d$

Number of words in document  $d$

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

Total number of documents in database

Number of documents word  $i$  occurs in, in whole database

# Bags of words for content-based image retrieval

Visually defined query

“Find this clock”



“Find this place”



“Groundhog Day” [Rammis, 1993]



# Example



## retrieved shots



# Video Google System

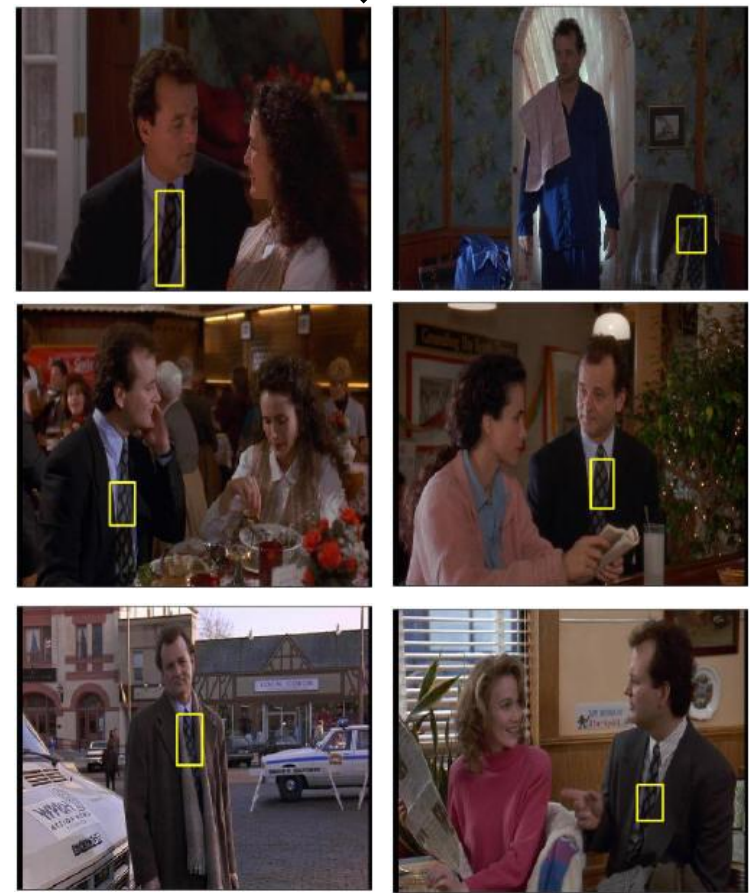
1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- Demo online at : <http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html>



Query region



Retrieved frames



# Scoring retrieval quality



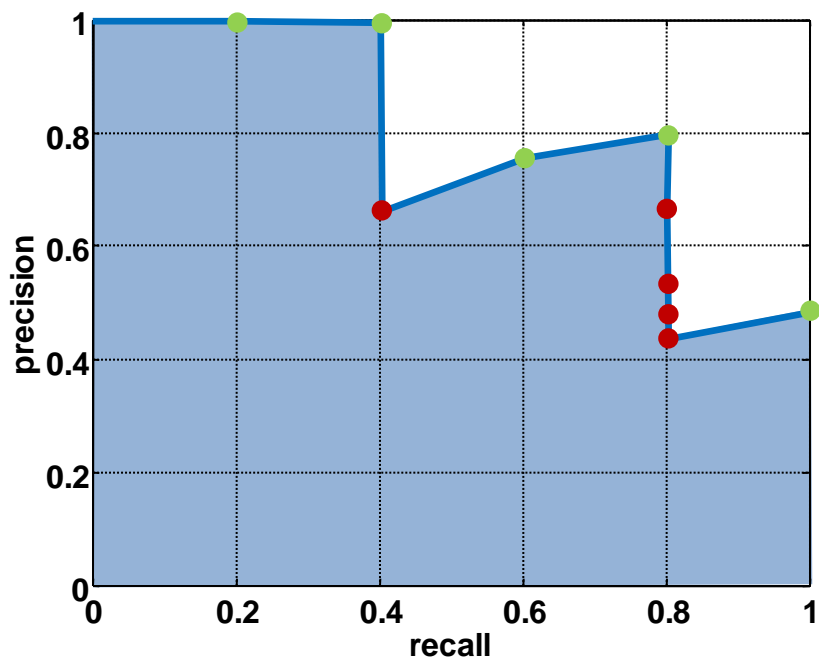
Query

Database size: 10 images

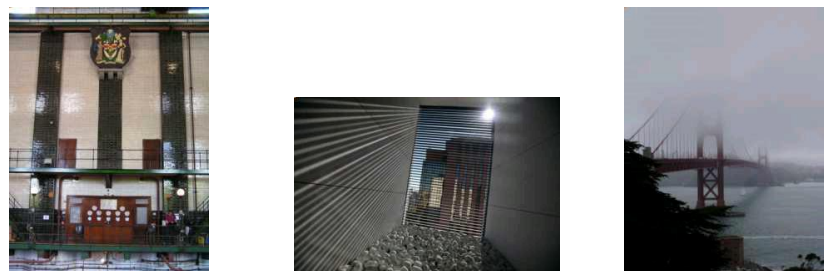
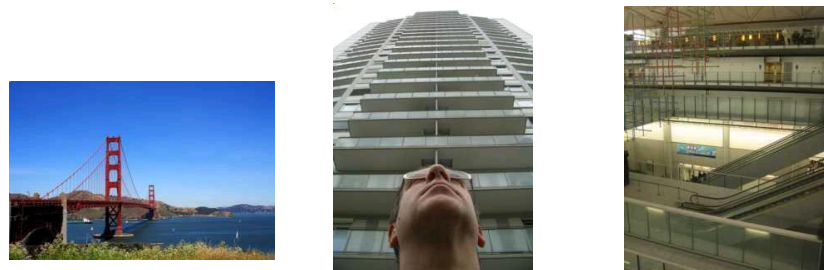
Relevant (total): 5 images

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

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

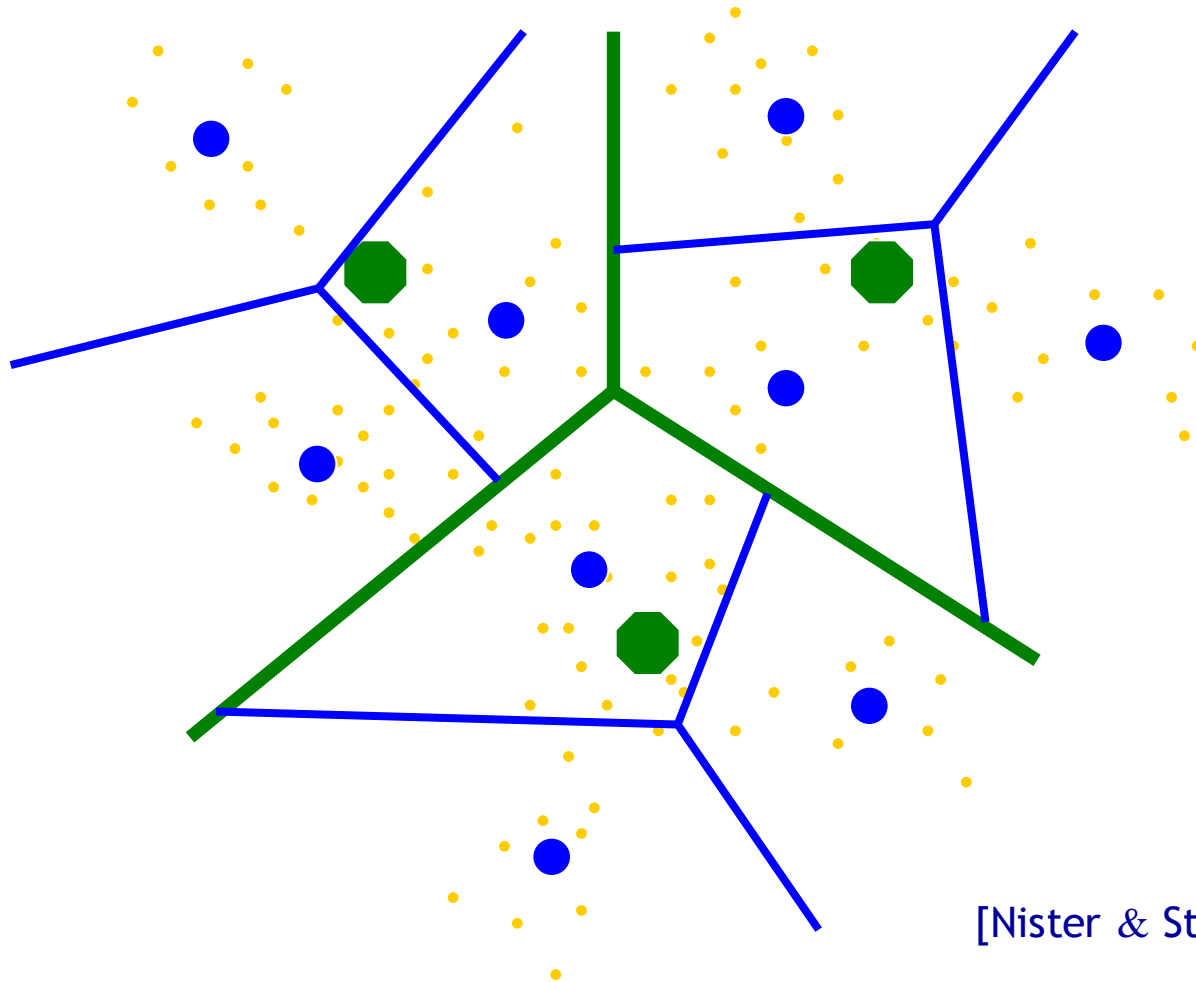


Results (ordered):



# Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:

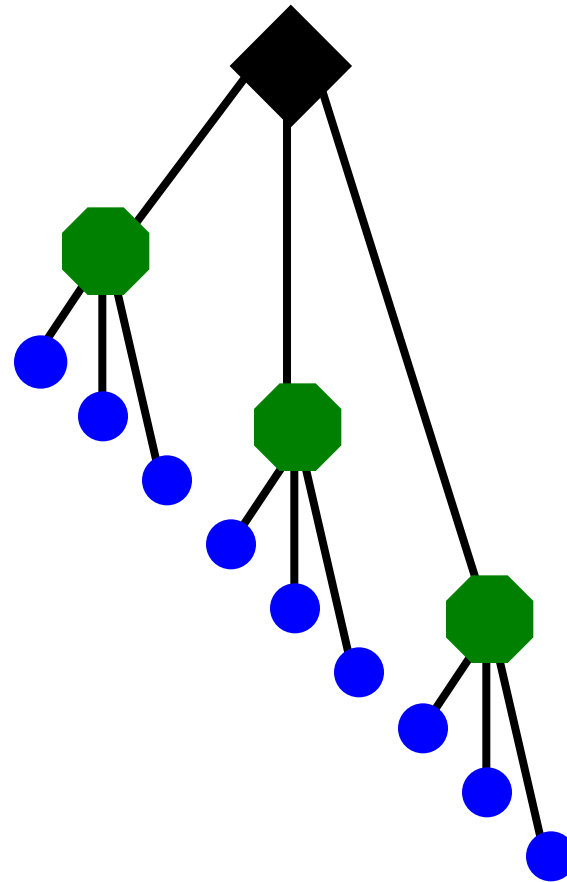


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

# Vocabulary Tree

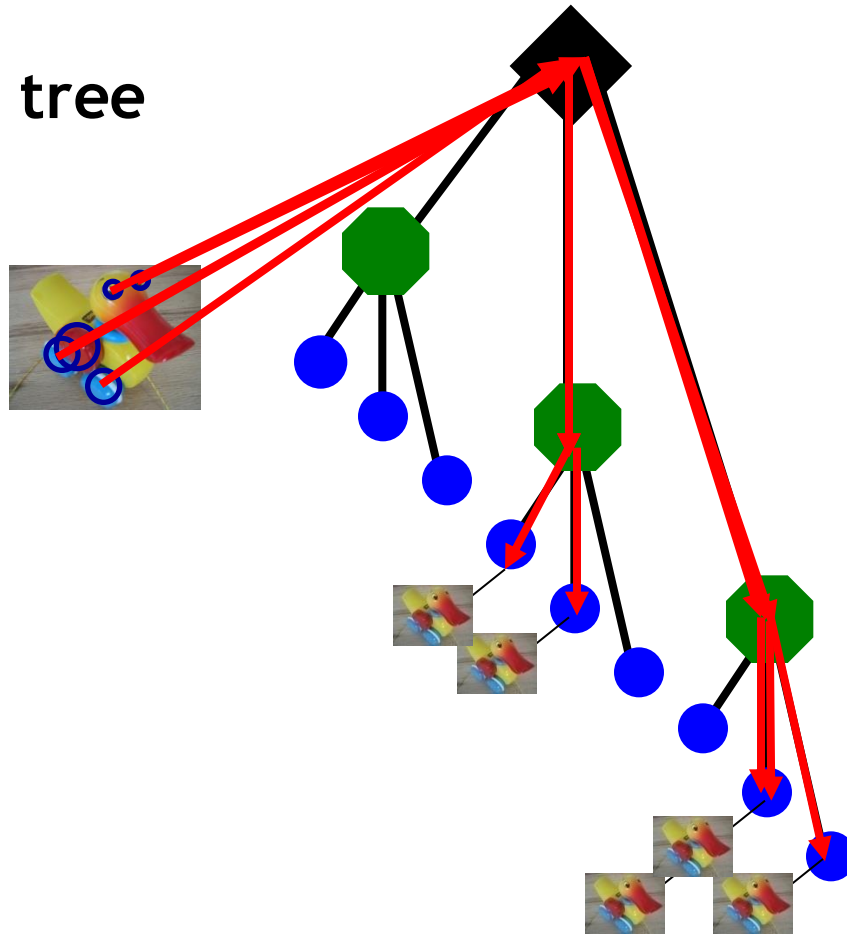
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

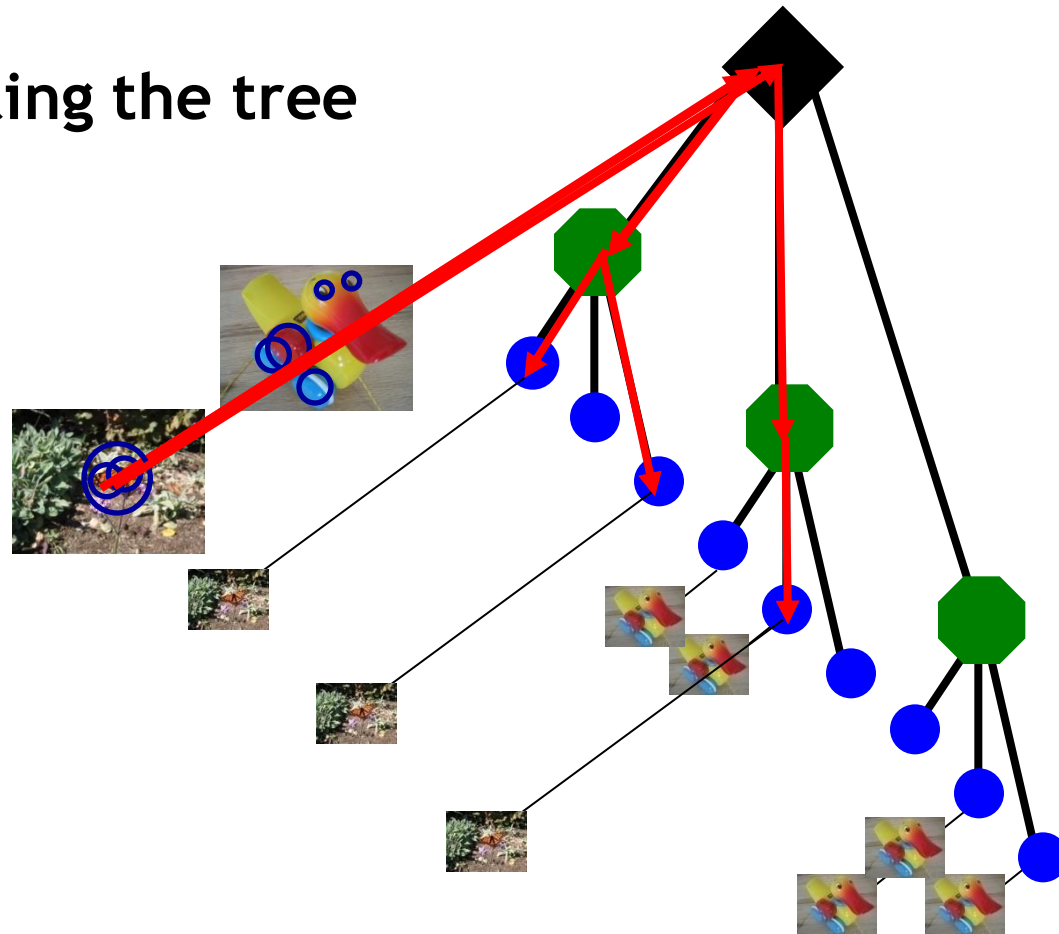
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

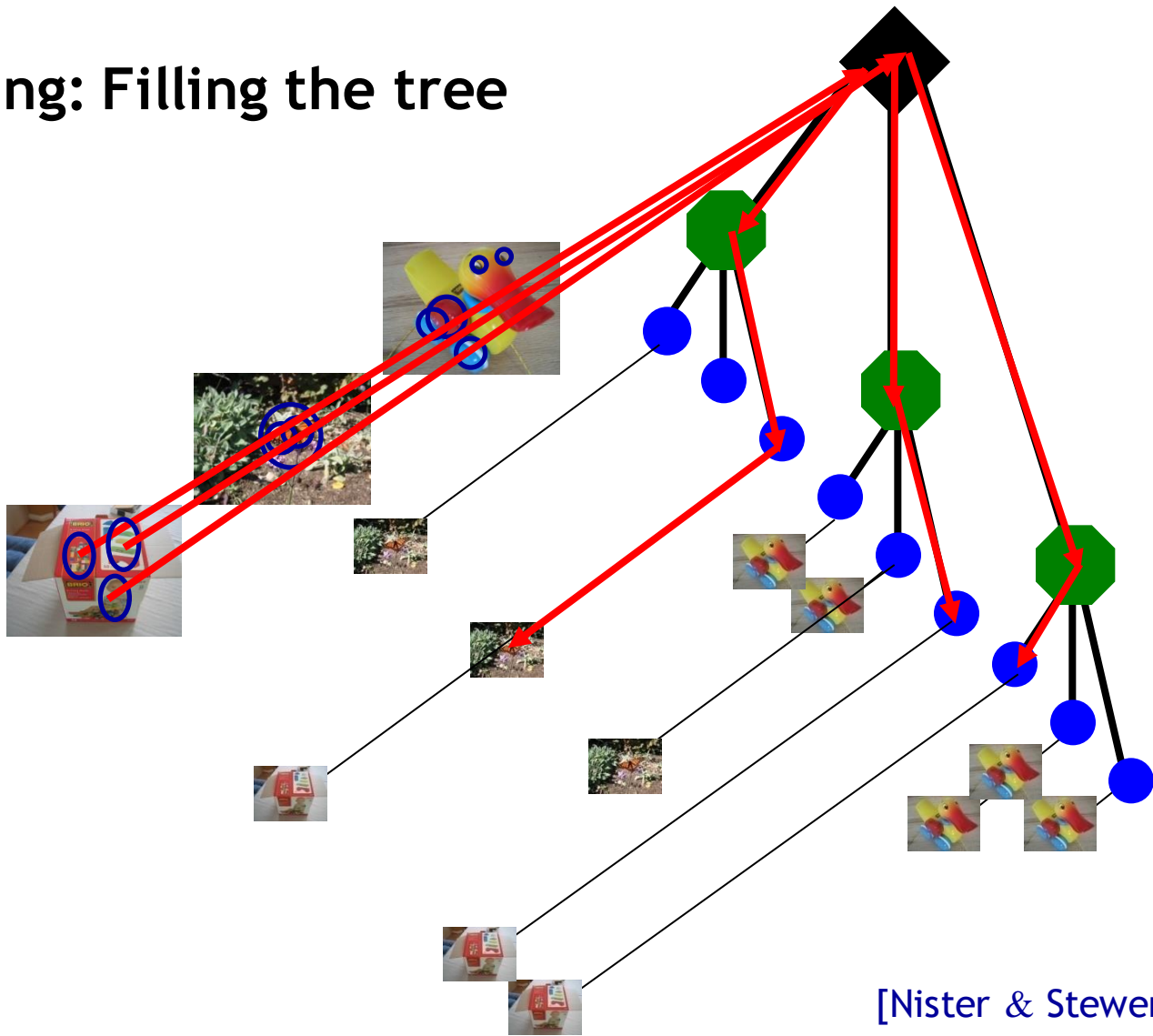
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

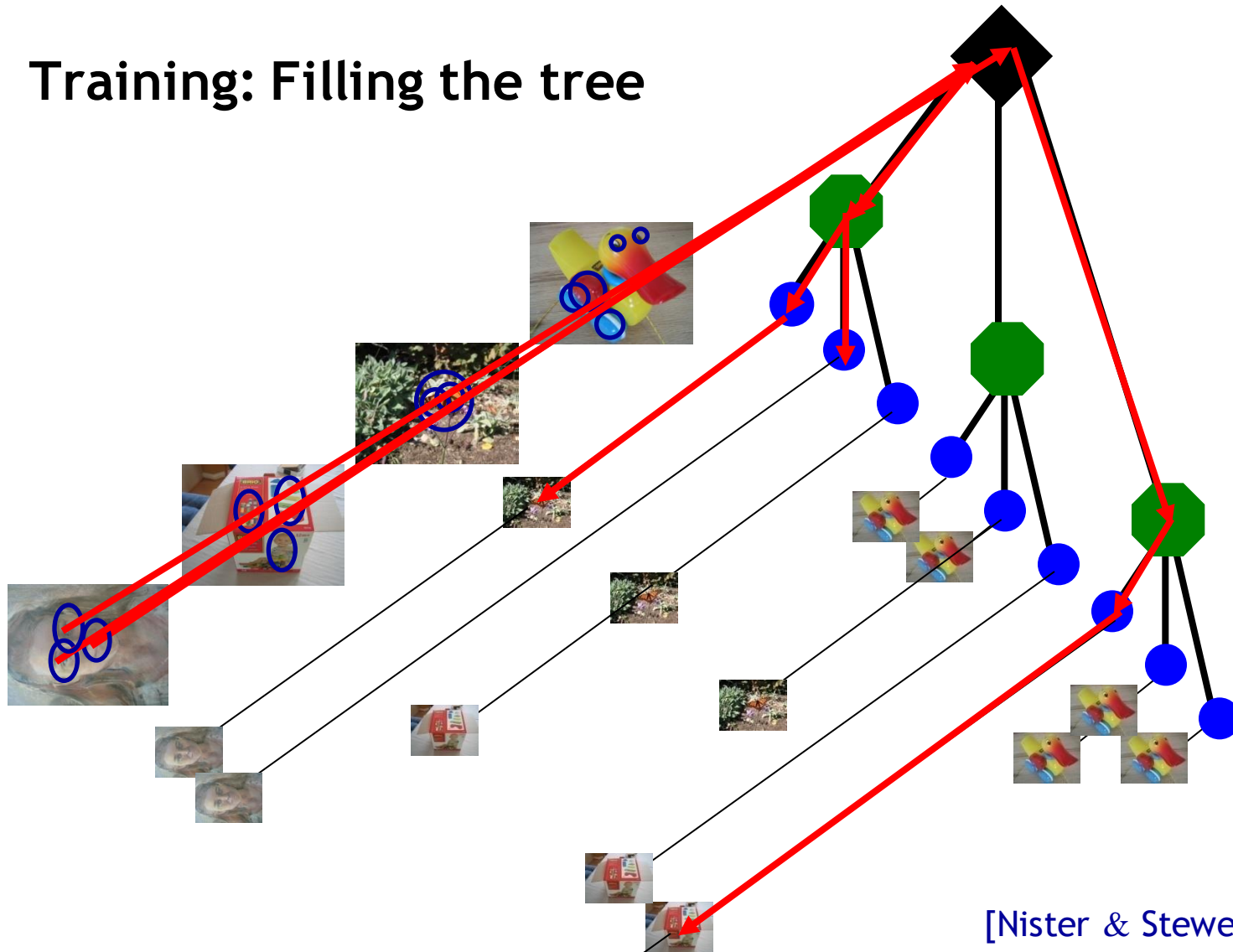
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

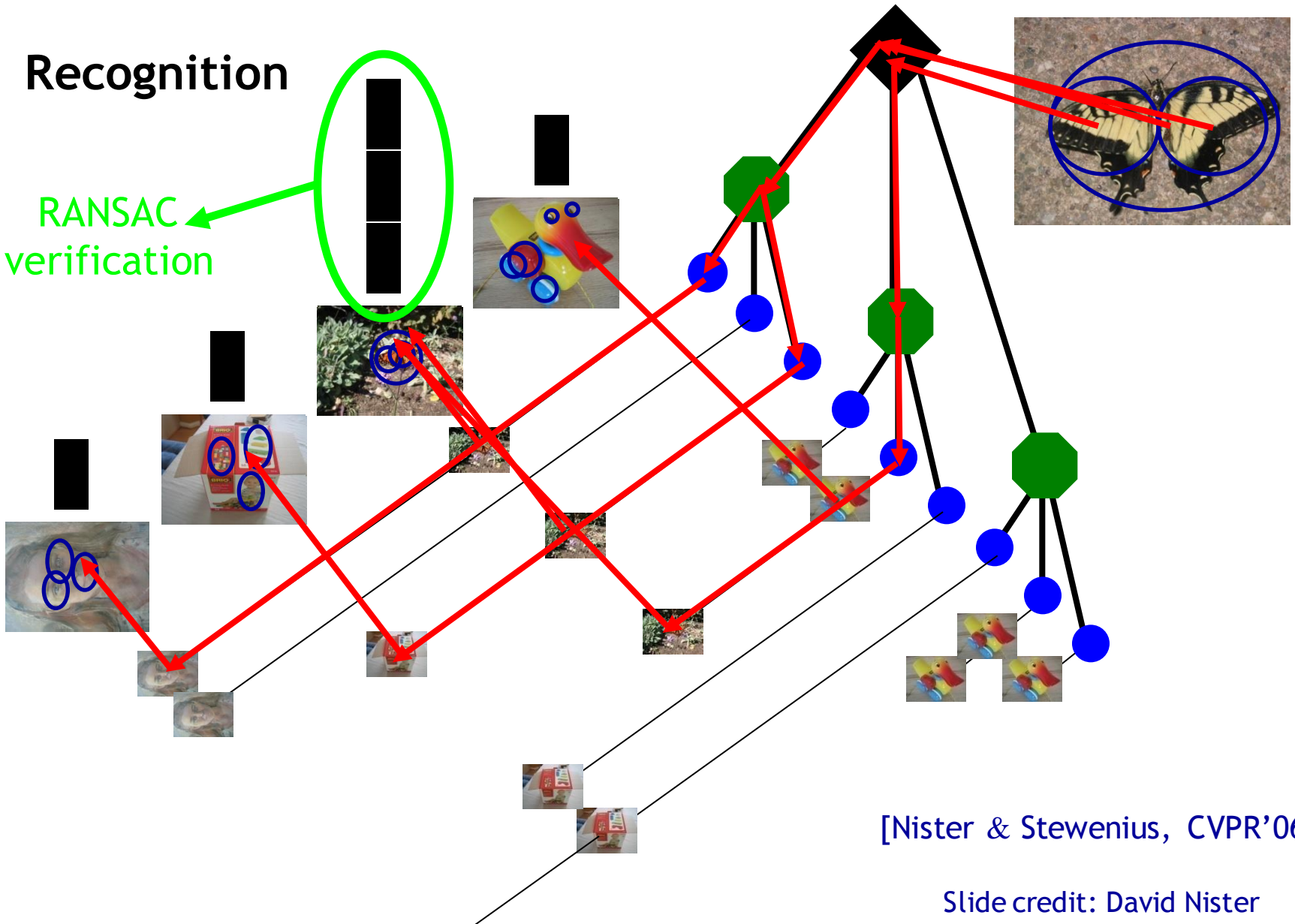
What is the computational advantage of the hierarchical representation bag of words, vs. a flat vocabulary?



# Vocabulary Tree

- Recognition

RANSAC  
verification



[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

# Bags of words: pros and cons

- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides vector representation for sets
- + very good results in practice
  
- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear

# 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