

CS 1674: Intro to Computer Vision
Feature Matching and Indexing

Prof. Adriana Kovashka
University of Pittsburgh
September 26, 2016

HW3P post-mortem

- Matlab: 21% of you reviewed 0-33% of it
 - Please review the entire tutorial ASAP
- How long did HW3P take? (Answer on Socrative)
- What did you learn from it?
- What took the most time?

Plan for Today

- Feature detection (wrap-up)
- Matching features
- Indexing features
 - Visual words
- Application to image retrieval

Matching local features



Image 1



Image 2

- To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest feature Euclidean distance)
- Simplest approach: compare them all, take the closest (or closest k , or within a thresholded distance)

Robust matching



Image 1

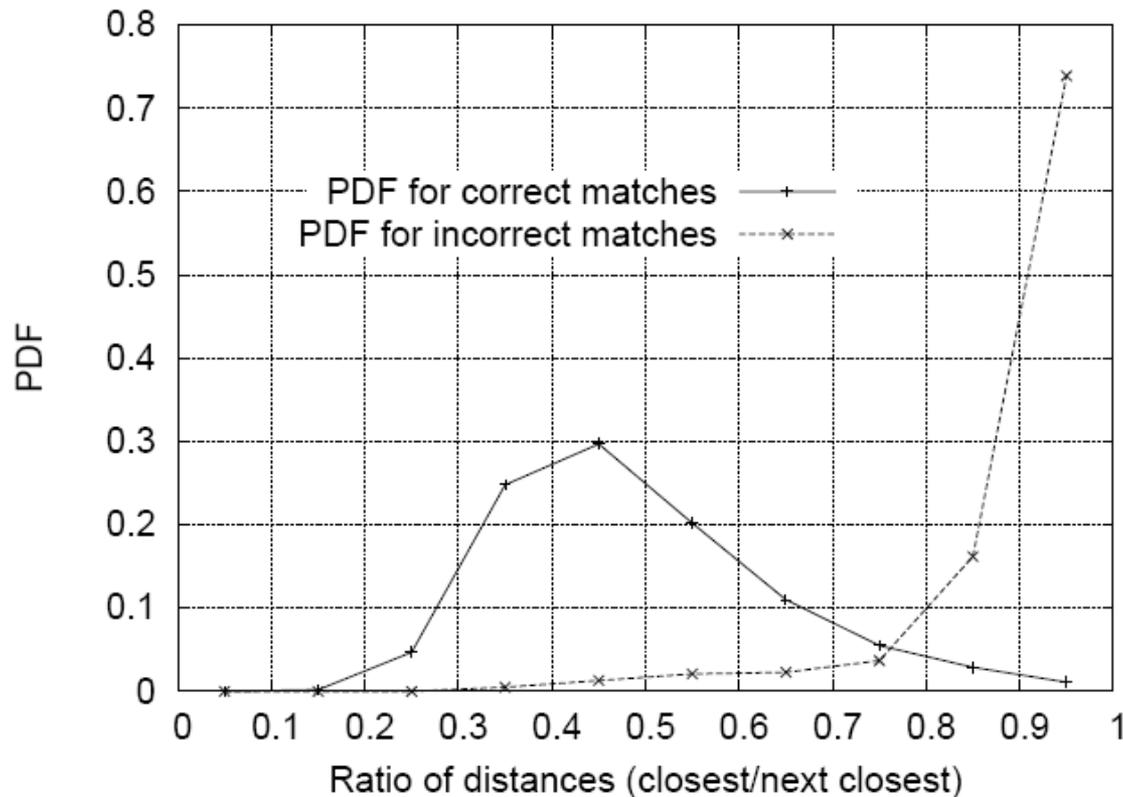


Image 2

- At what Euclidean distance value do we have a good match?
- To add robustness to matching, can consider **ratio** : distance to best match / distance to second best match
- If low, first match looks good.
- If high, could be ambiguous match.

Matching SIFT descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor

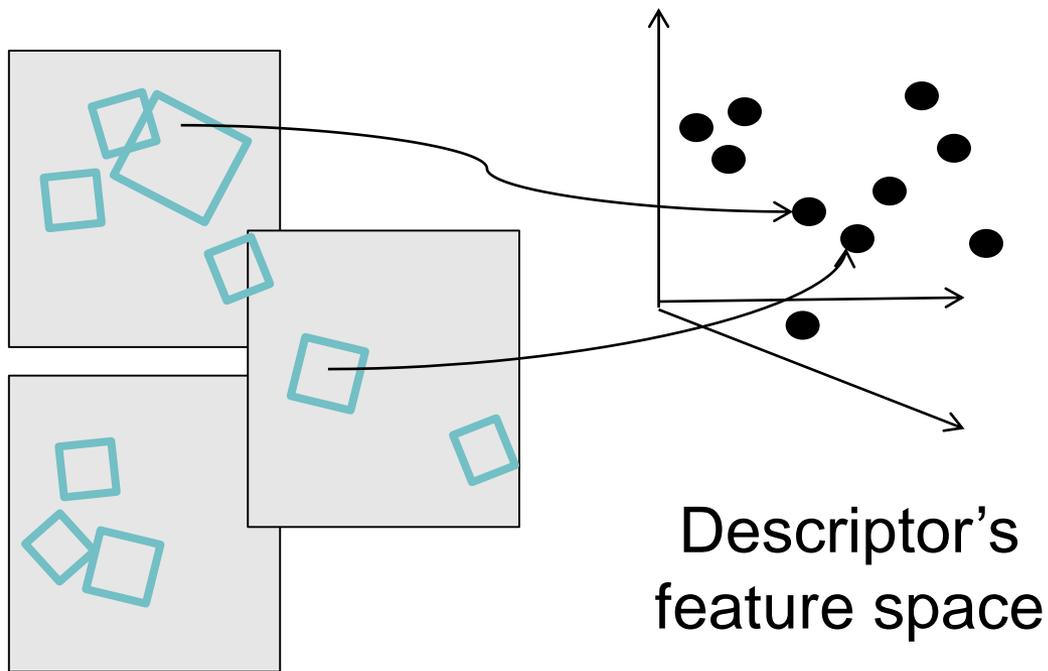


Efficient matching

- So far we discussed matching across just two images
- What if you wanted to match a query feature from one image, to all frames in a video, or to a giant database?
- 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?

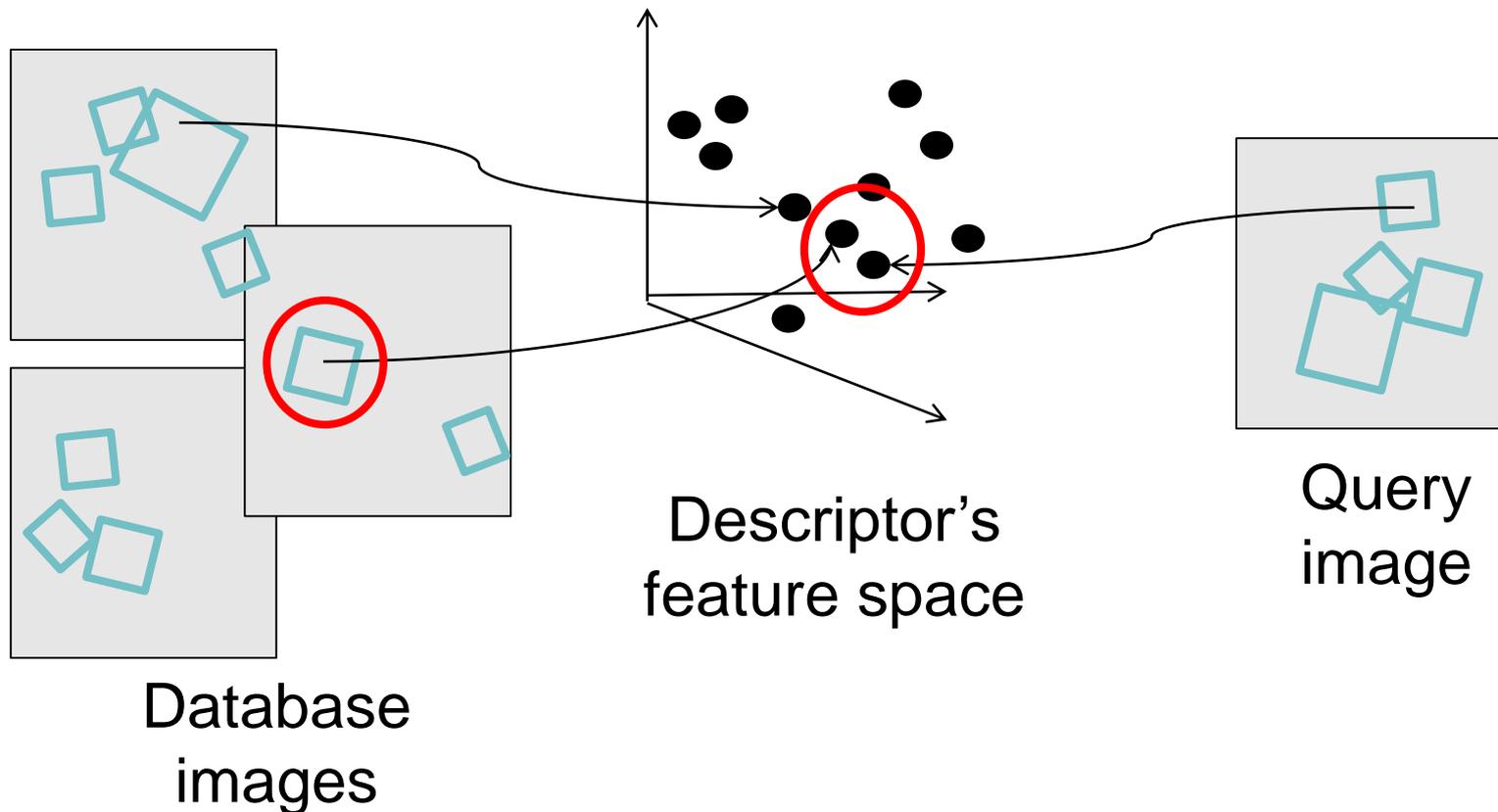
Indexing local features: Setup

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



Indexing local features: Setup

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



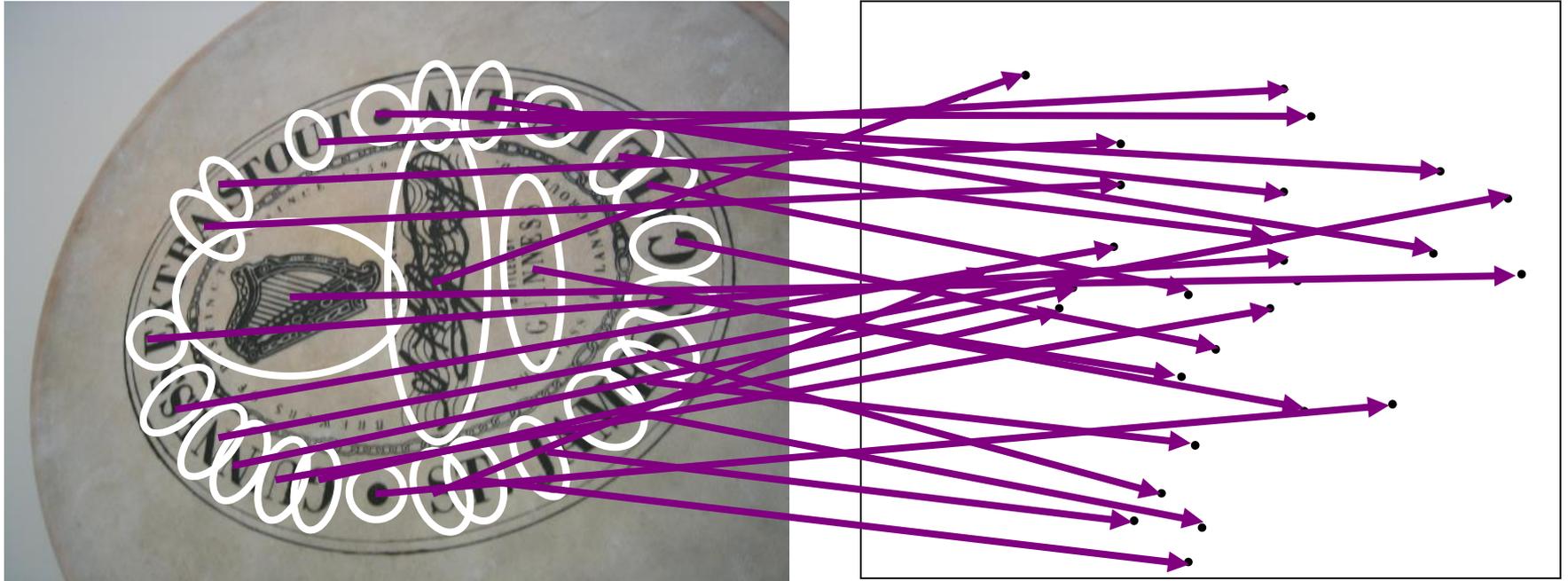
Indexing local features: Inverted file index

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; 85
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; 168
	Fires, Prescribed ; 148
	Fisherman's Village; 151
	Flagler County; 171
	Flagler, Henry; 97,165,167,171
	Florida Aquarium; 186
	Florida,
	12,000 years ago; 187
	Cavern SP; 114
	Map of all Expressways; 2-3
	Mus of Natural History; 134
	National Cemetery ; 141
	Part of Africa; 177
	Platform; 187
	Sheriff's Boys Camp; 126
	Sports Hall of Fame; 130
	Sun 'n Fun Museum; 97
	Supreme Court; 107
	Florida's Turnpike (FTP), 178,189
	25 mile Strip Maps; 66
	Administration; 189
	Coin System; 190
	Exit Services; 189
	HEFT; 76,161,190
	History; 189
	Names; 189
	Service Plazas; 190
	Spur SR91; 76
	Ticket System; 190
	Toll Plazas; 190
	Ford, Henry; 152

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

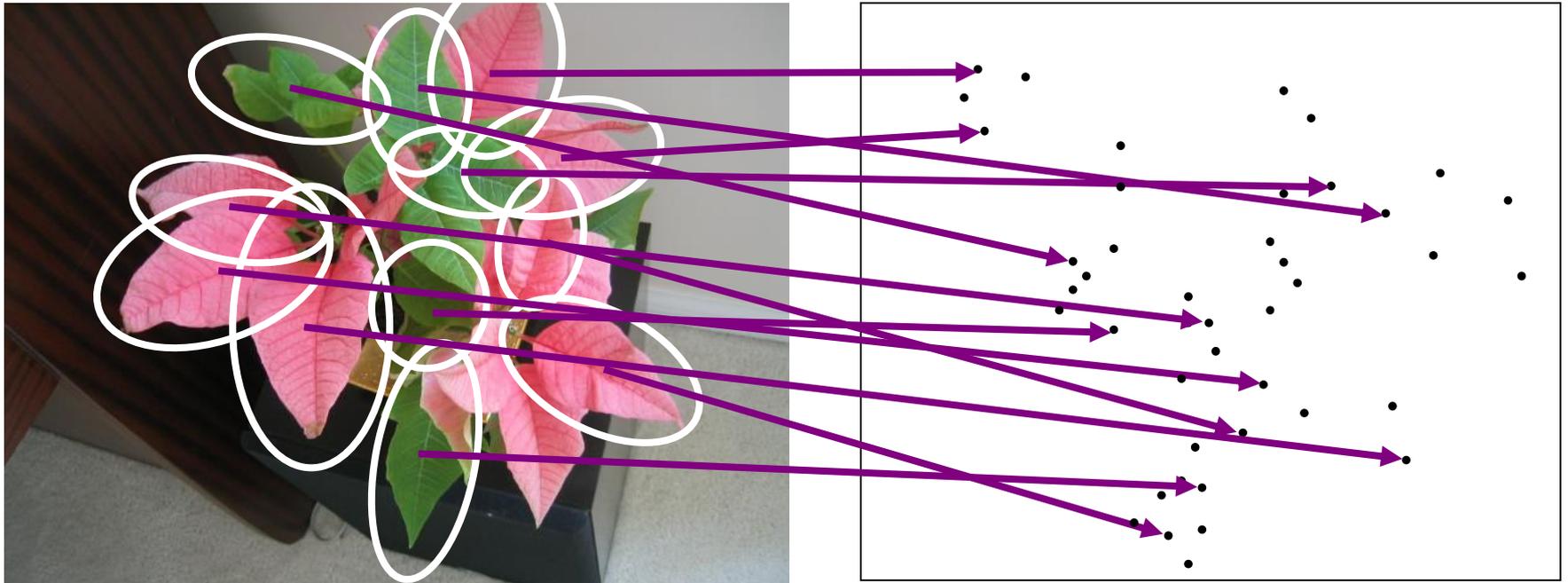
Visual words: main idea

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

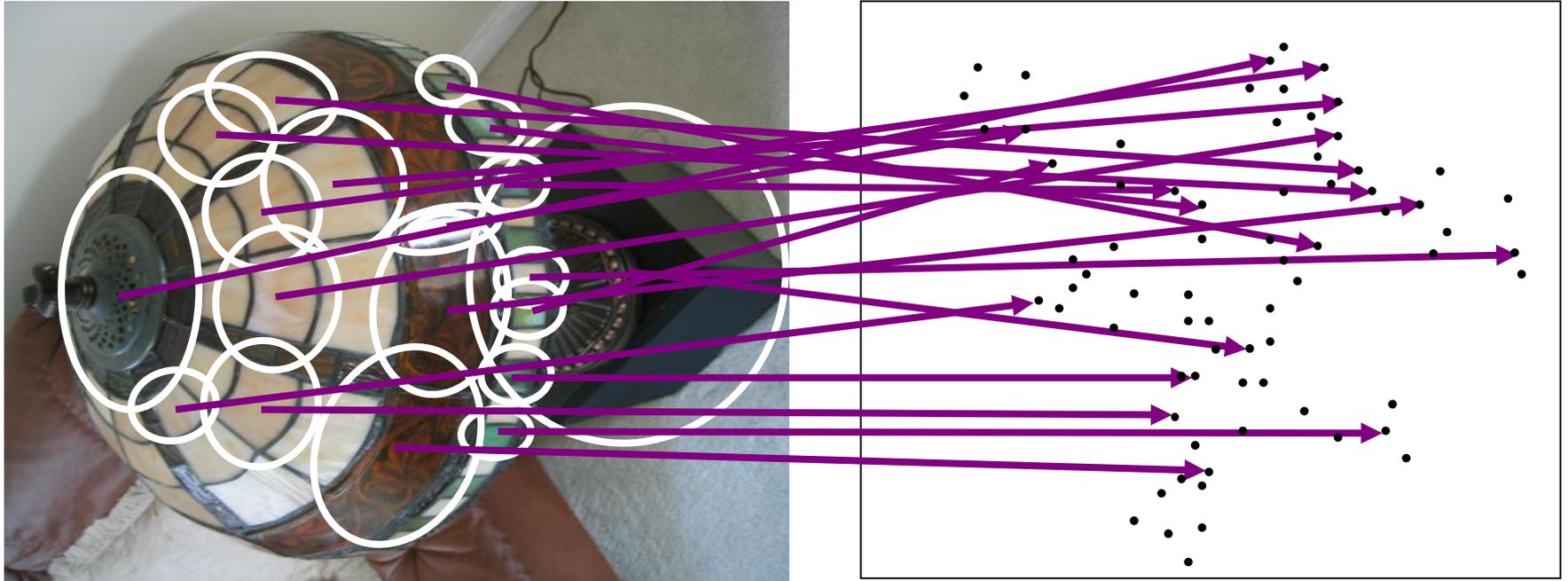


e.g., SIFT descriptor space: each point is 128-dimensional

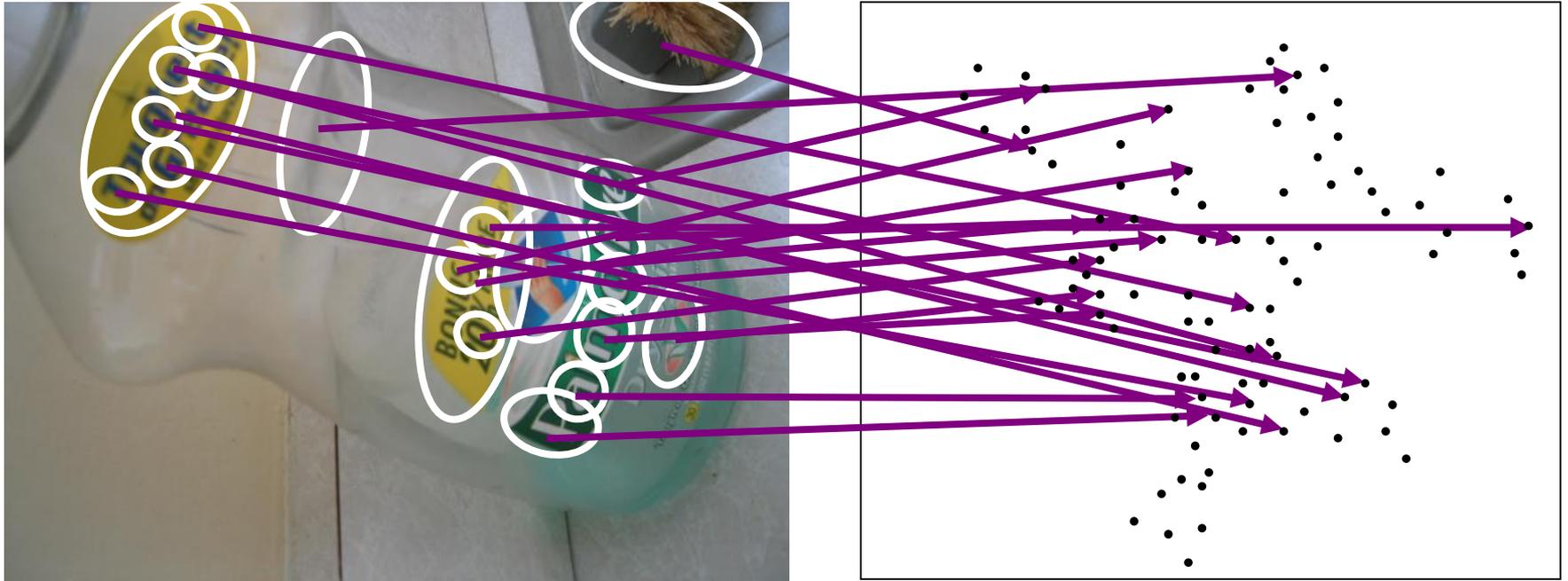
Visual words: main idea

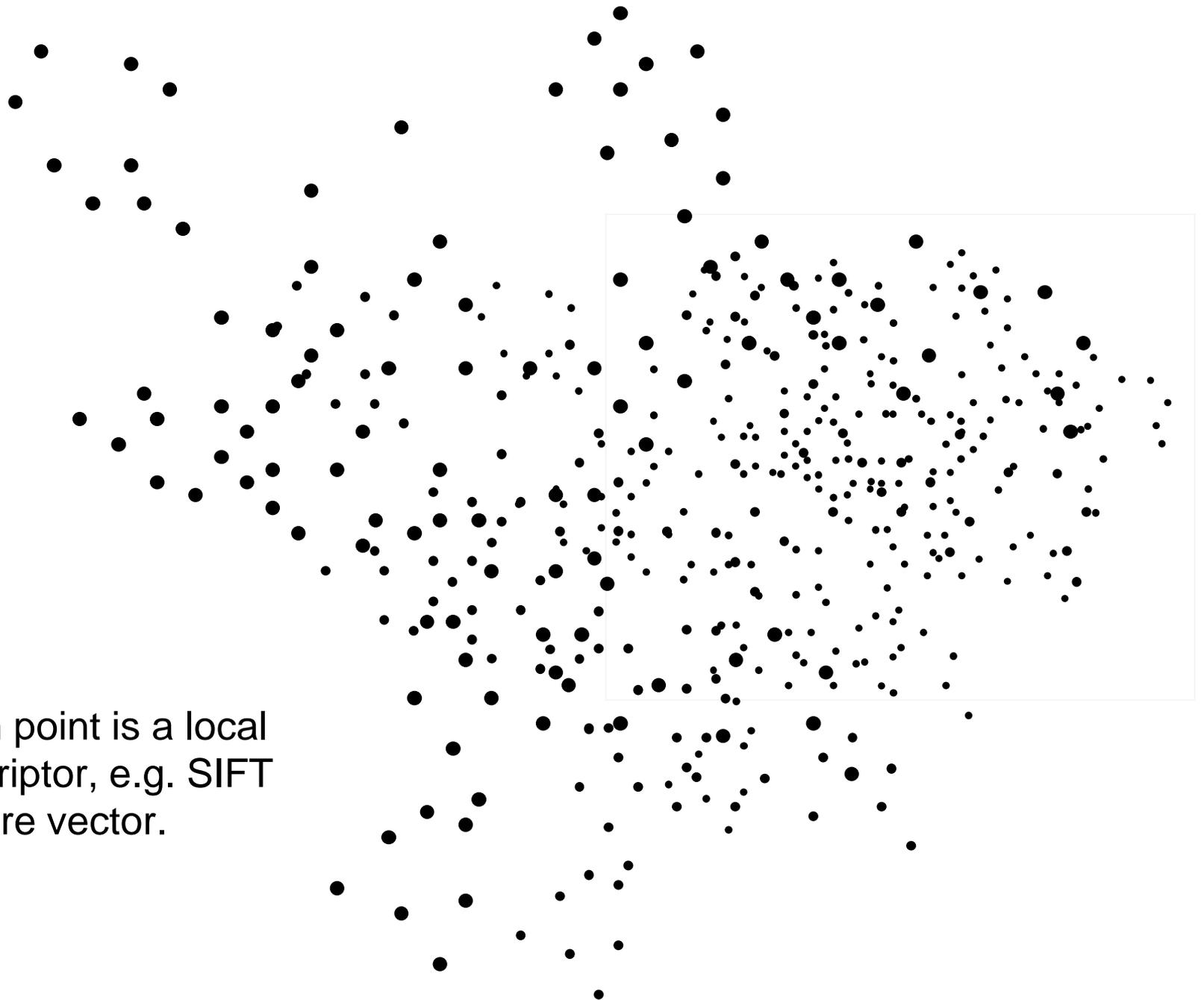


Visual words: main idea

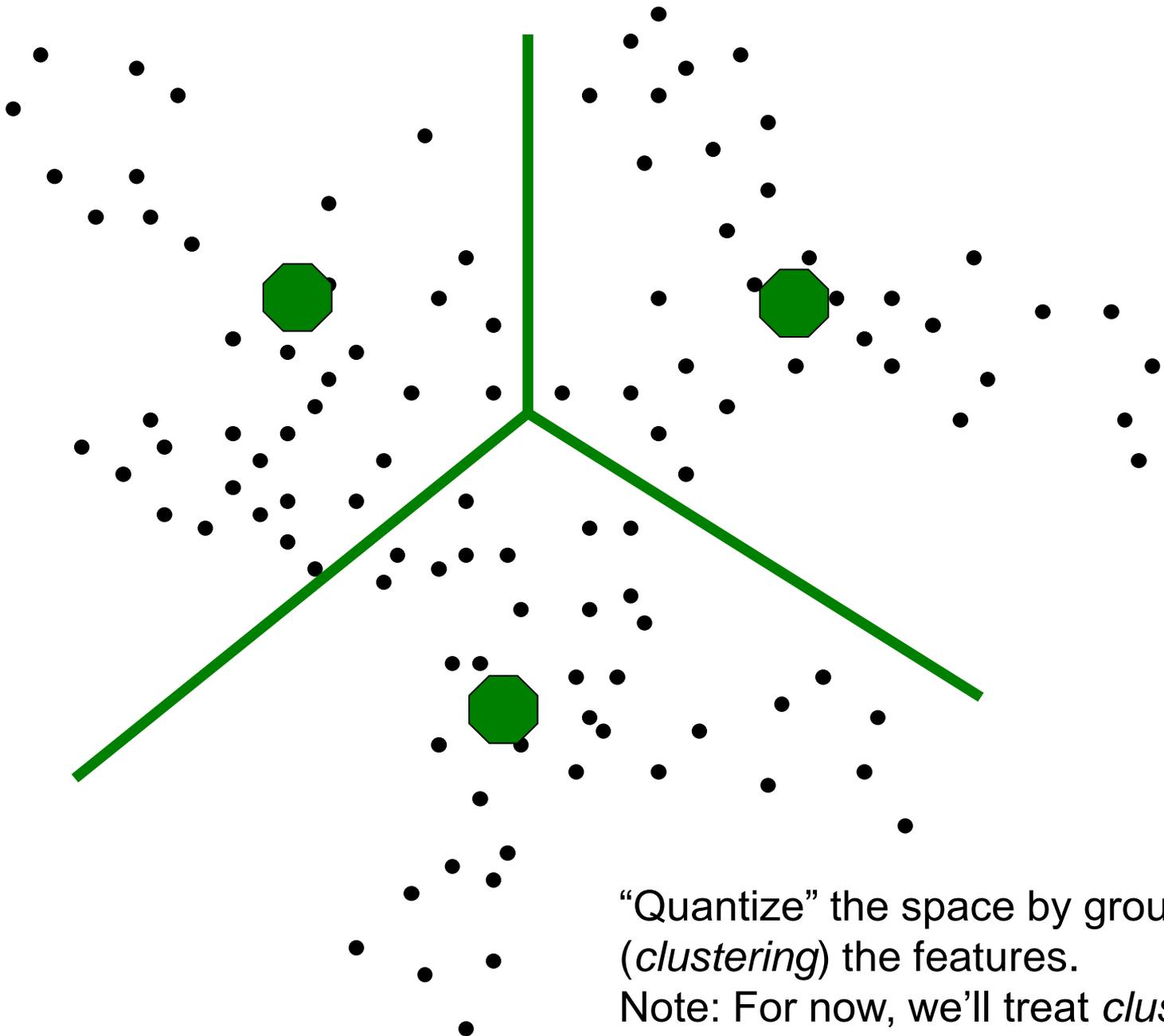


Visual words: main idea





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



“Quantize” the space by grouping (*clustering*) the features.
Note: For now, we’ll treat *clustering* as a black box.

Visual words

- Patches on the right = regions used to compute SIFT
- If I group these, each group of patches will belong to the same “visual word”

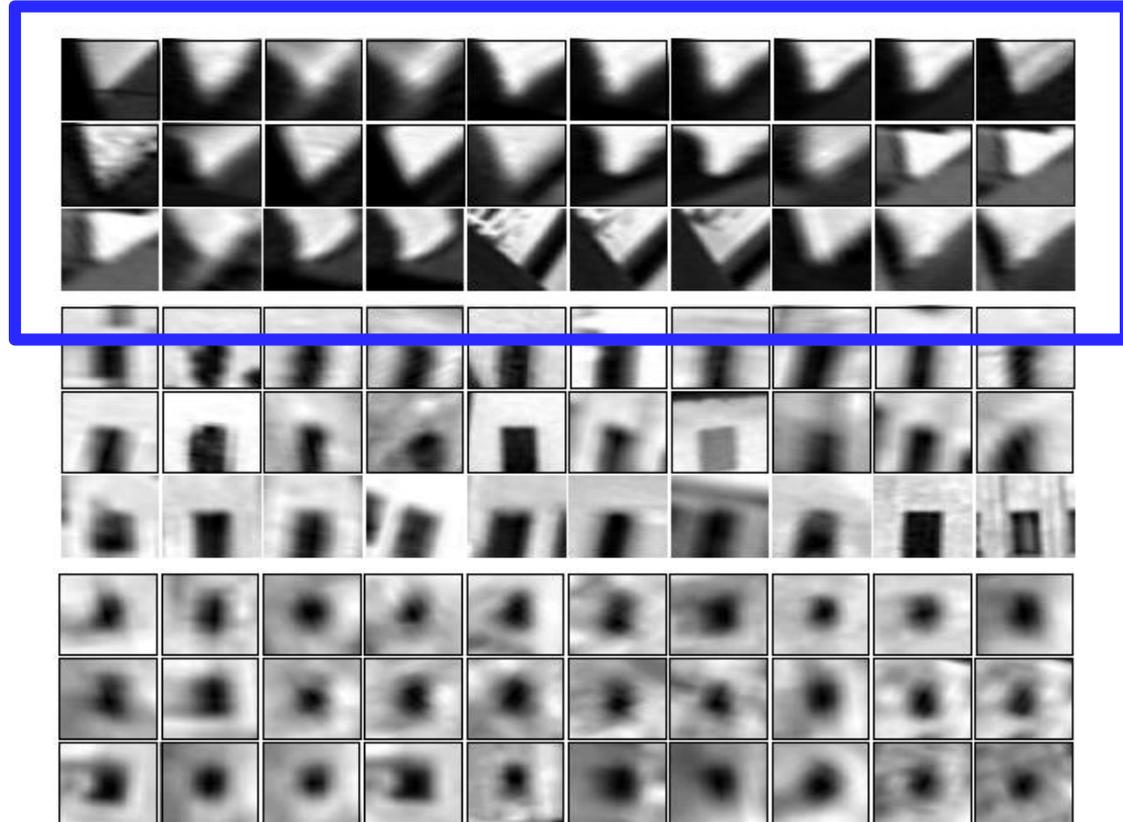
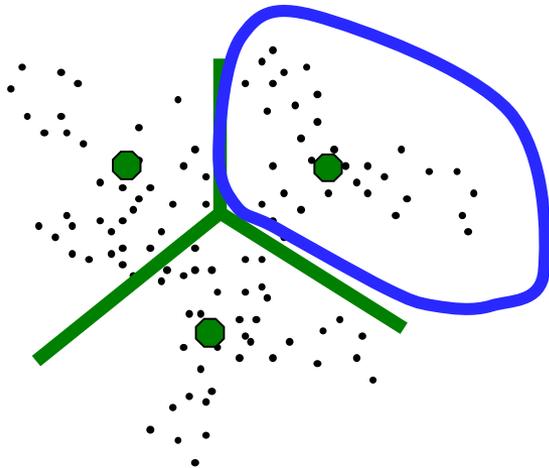
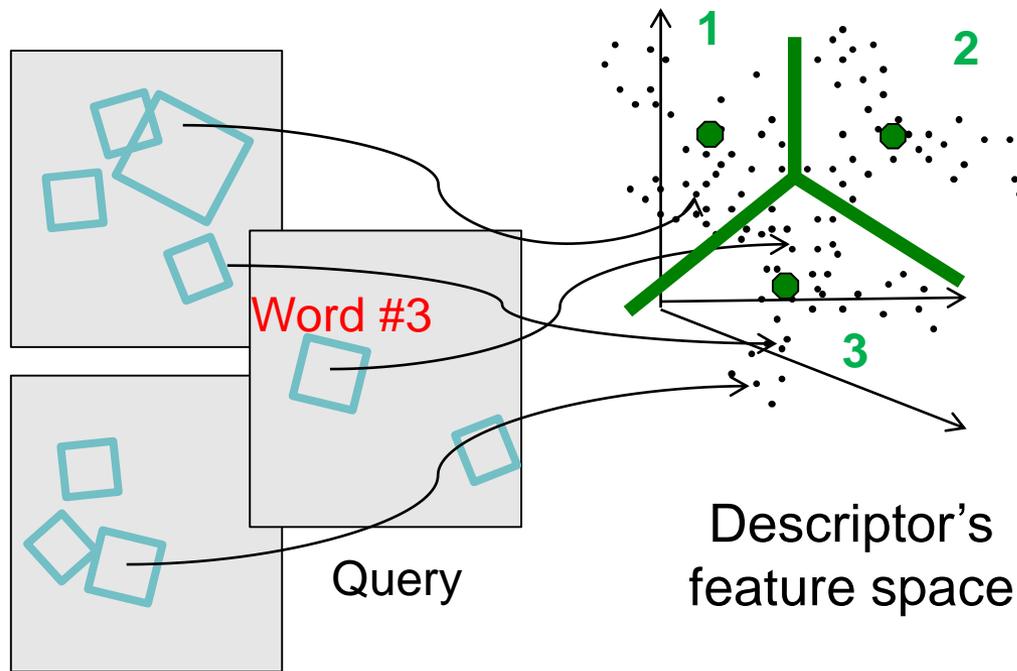


Figure from Sivic & Zisserman, ICCV 2003

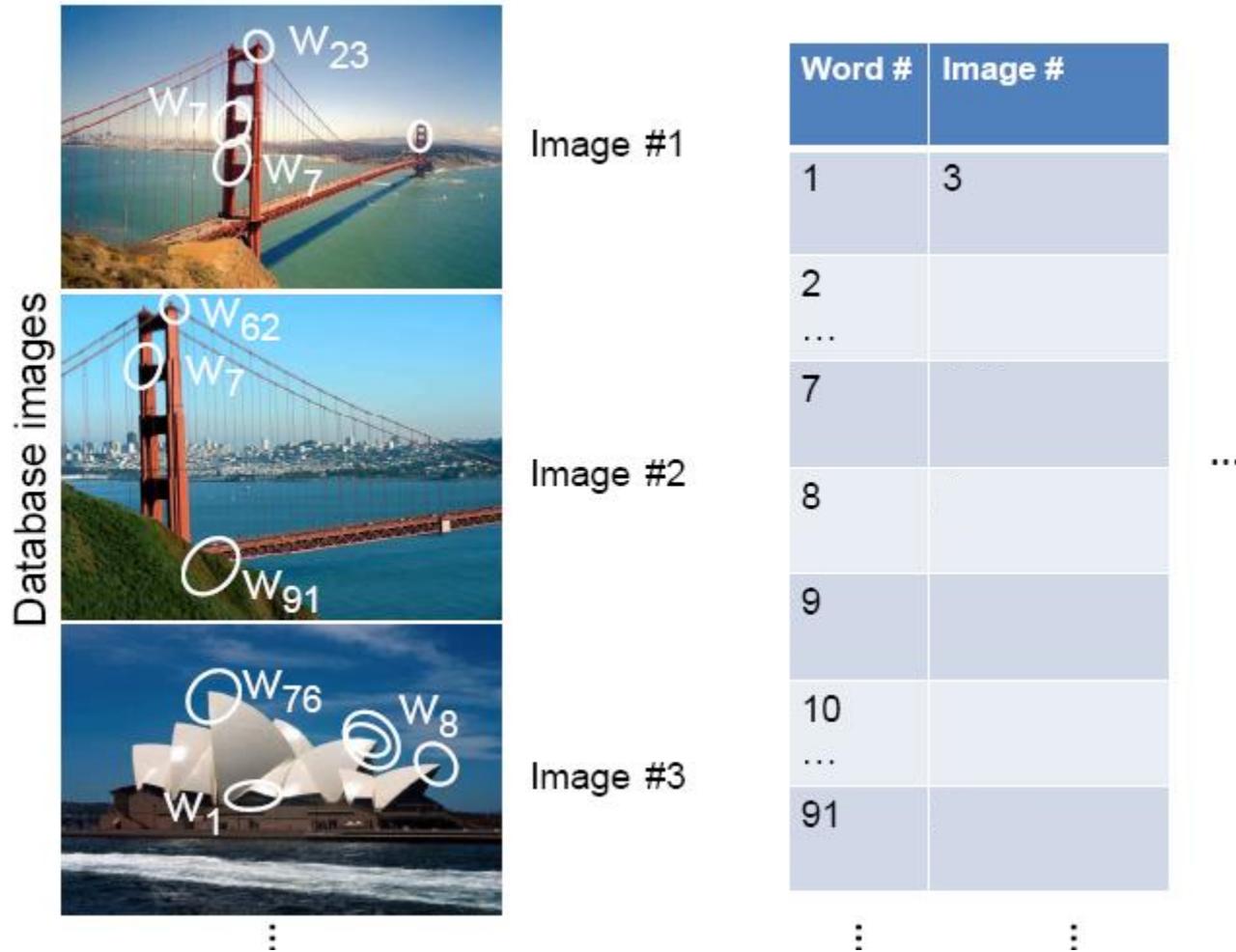
Visual words for indexing

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



- Each cluster has a center.
- Determine which word to assign to each new image region by finding the closest cluster center.

Inverted file index



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

Inverted file index

When will this indexing process give us a gain in efficiency?



New query image

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

- For a new query image, we can figure out which database images share a word with it, and retrieve those images as matches.
- We can call this retrieval process *instance recognition*.

How to describe entire document?

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 discovered that the image is analyzed in a more complex way following the path to the various centers of the cortex, Hubel and Wiesel have demonstrated that the *message about the image falling on the retina undergoes a point-by-point 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**

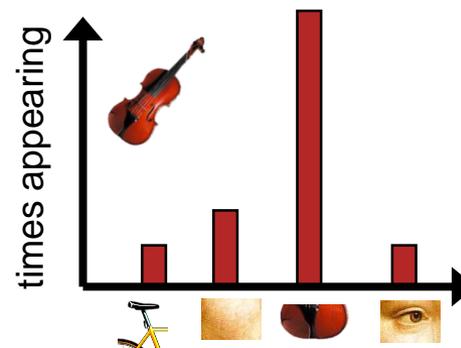
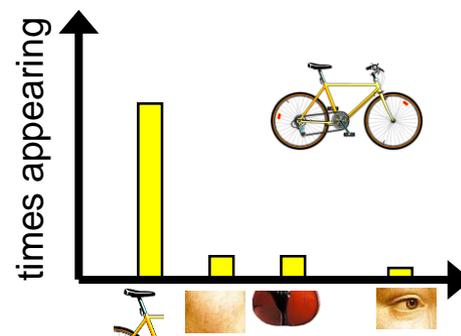
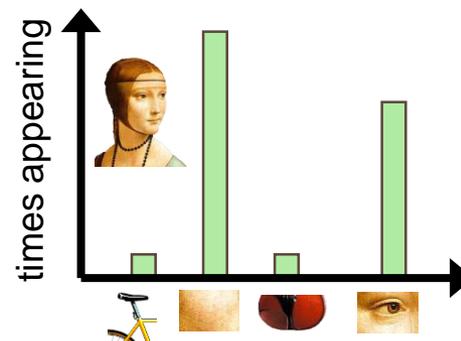
China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$575bn in 2004. The surplus of \$660bn. The surplus will annoy the US. China's government has deliberately agreed to keep the yuan is pegged to the US dollar. The government also needs to keep the demand so high for the country. China has kept the yuan against the dollar 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**

Describing images w/ visual words

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

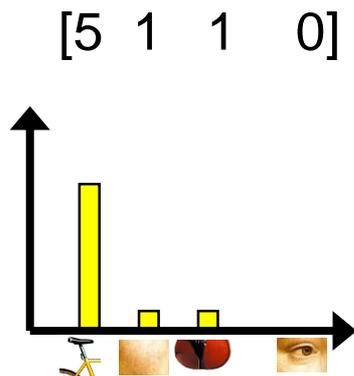
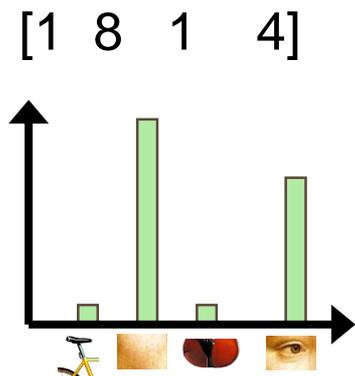
Feature patches:



Visual words

Comparing bags of words

- Rank images by normalized scalar product between their 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

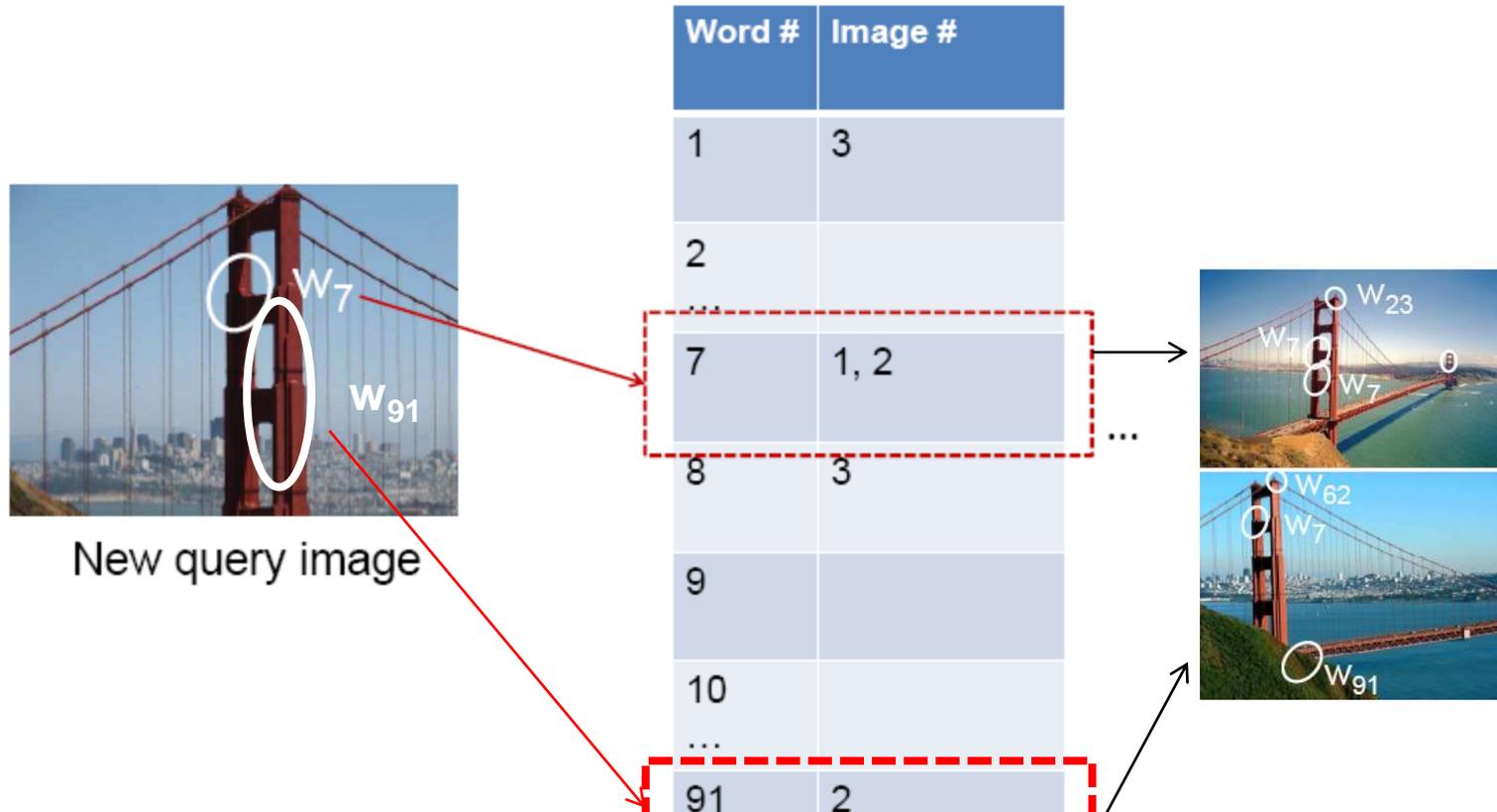
$$\text{sim}(d_j, q) = \text{dot}(d_j, q) / (\text{norm}(d_j, 2) * \text{norm}(q, 2))$$

Bags of words: pros and cons

- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + 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

Inverted file index and bags of words similarity



1. (offline) Extract features in database images, cluster them to find words, make index
2. Extract words in query (extract features and map each to closest cluster center)
3. Use inverted file index to find frames relevant to query
4. For each relevant frame, rank them by comparing word counts of query and frame

One more trick: *tf-idf* weighting

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

Number of occurrences of word i in document d

Number of words in document d

Total number of documents in database

Number of documents in which word i occurs

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

Normalized bag-of-words

Bags of words for content-based image retrieval

Visually defined query

“Groundhog Day” [Rammis, 1993]

“Find this clock”



“Find this place”



Example



retrieved shots



Video Google System

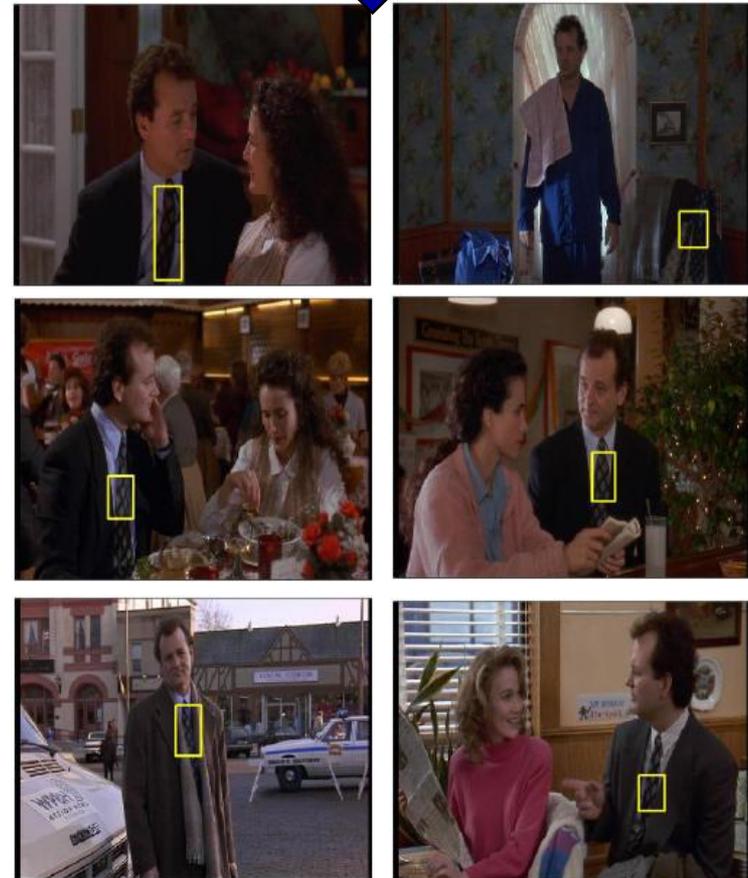
1. Collect all words within query region
2. Inverted file index to find relevant frames (skip for HW5P)
3. Compare word counts (BOW)
4. **Spatial verification** (skip)

Sivic & Zisserman, ICCV 2003

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



Query region



Retrieved frames

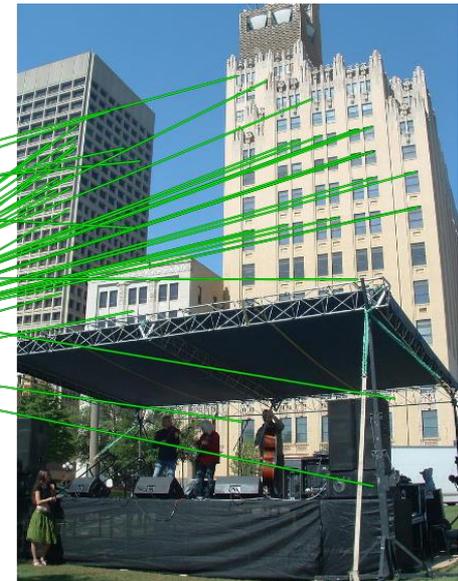
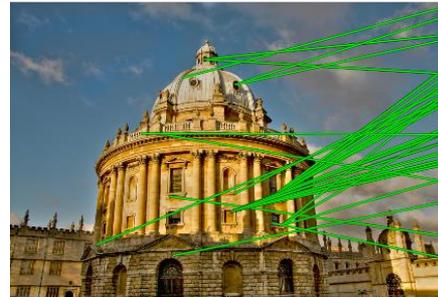
Preview: Spatial Verification

Query



DB image with high BoW
similarity

Query

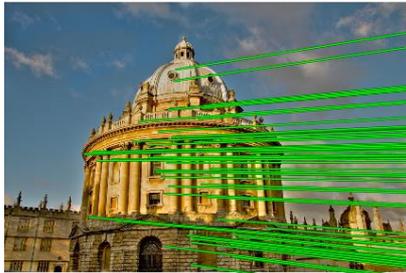


DB image with high BoW
similarity

Both image pairs have many visual words in common.

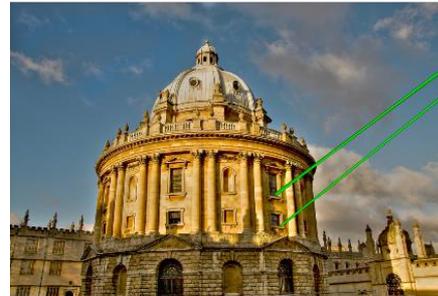
Preview: Spatial Verification

Query



DB image with high BoW similarity

Query



DB image with high BoW similarity

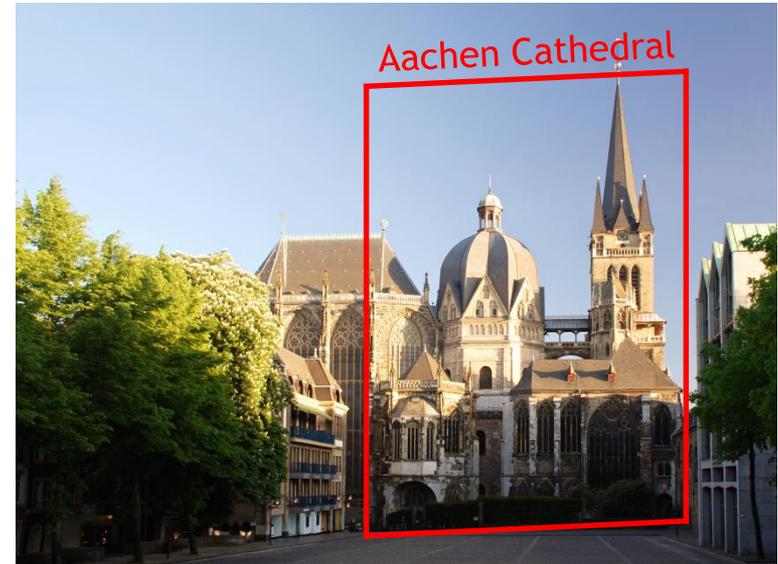
Only some of the matches are mutually consistent.

Example Applications



Mobile tourist guide

- Object/building recognition
- Self-localization
- Photo/video augmentation



Scoring retrieval quality



Query

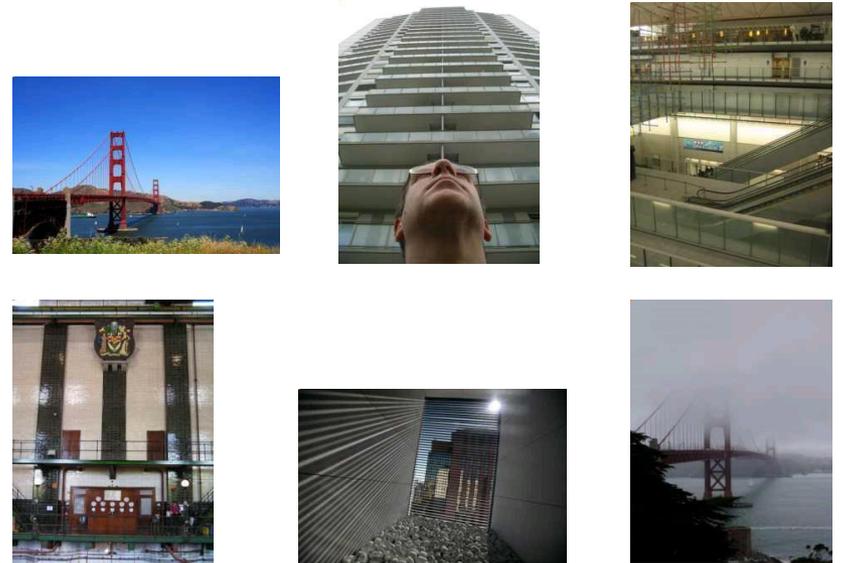
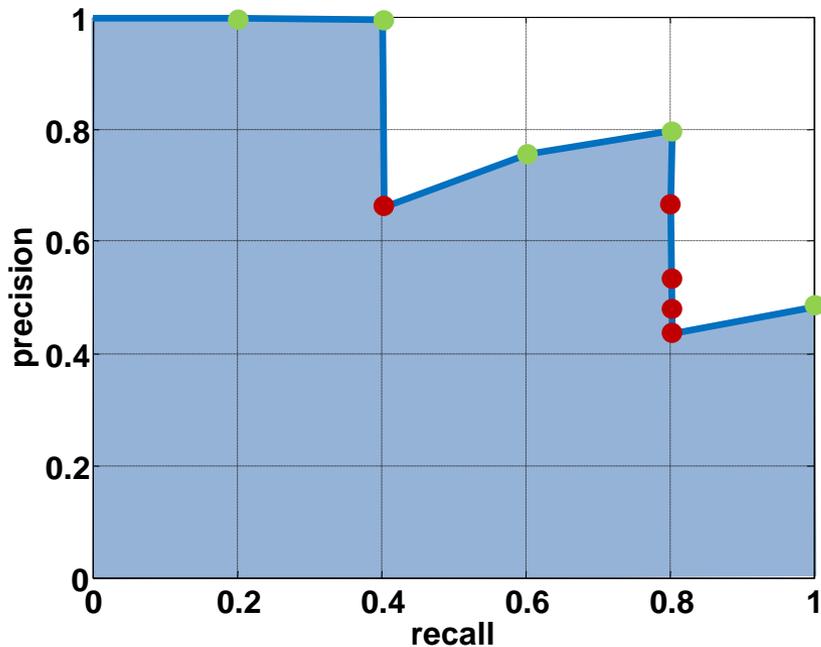
Database size: 10 images

Relevant (total): 5 images
(e.g. images of Golden Gate)

Results (ordered):



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



Indexing and Retrieval: Summary

- **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**: match local features
 - Optionally, perform spatial verification