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

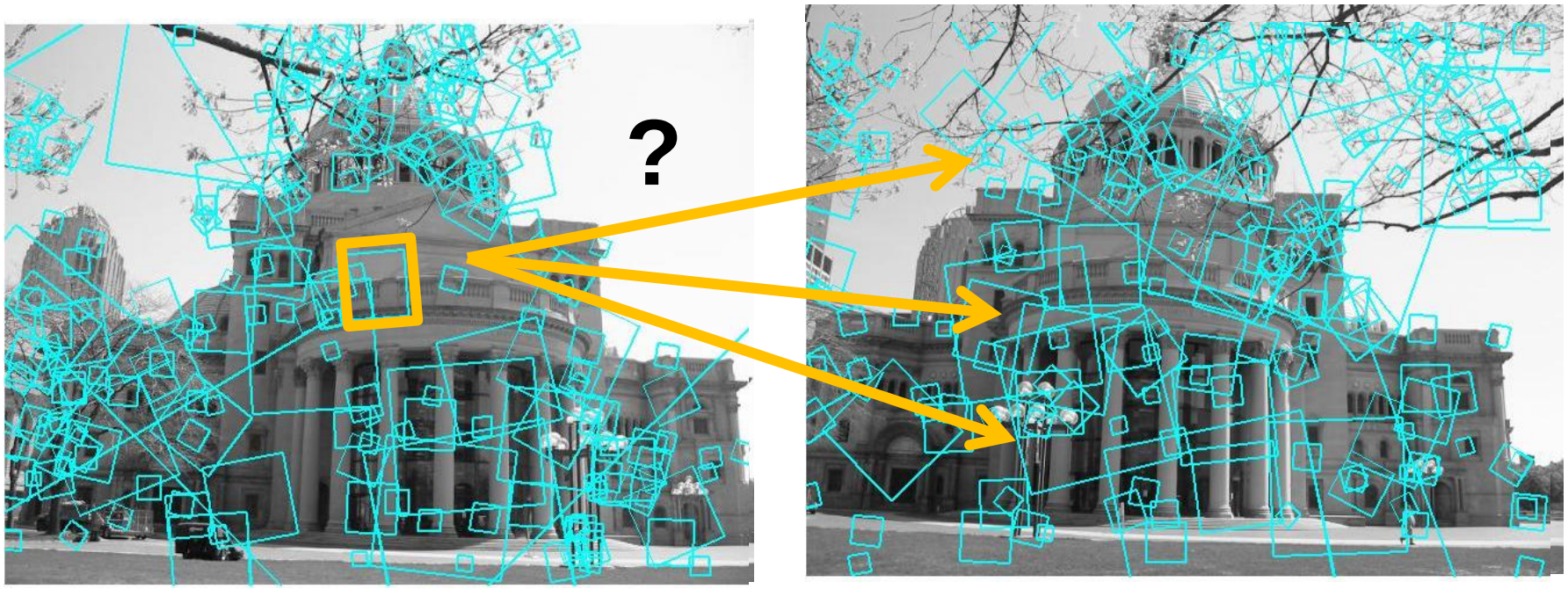


Image 1 Image 2

- 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)

K. Grauman

Robust matching

Image 1

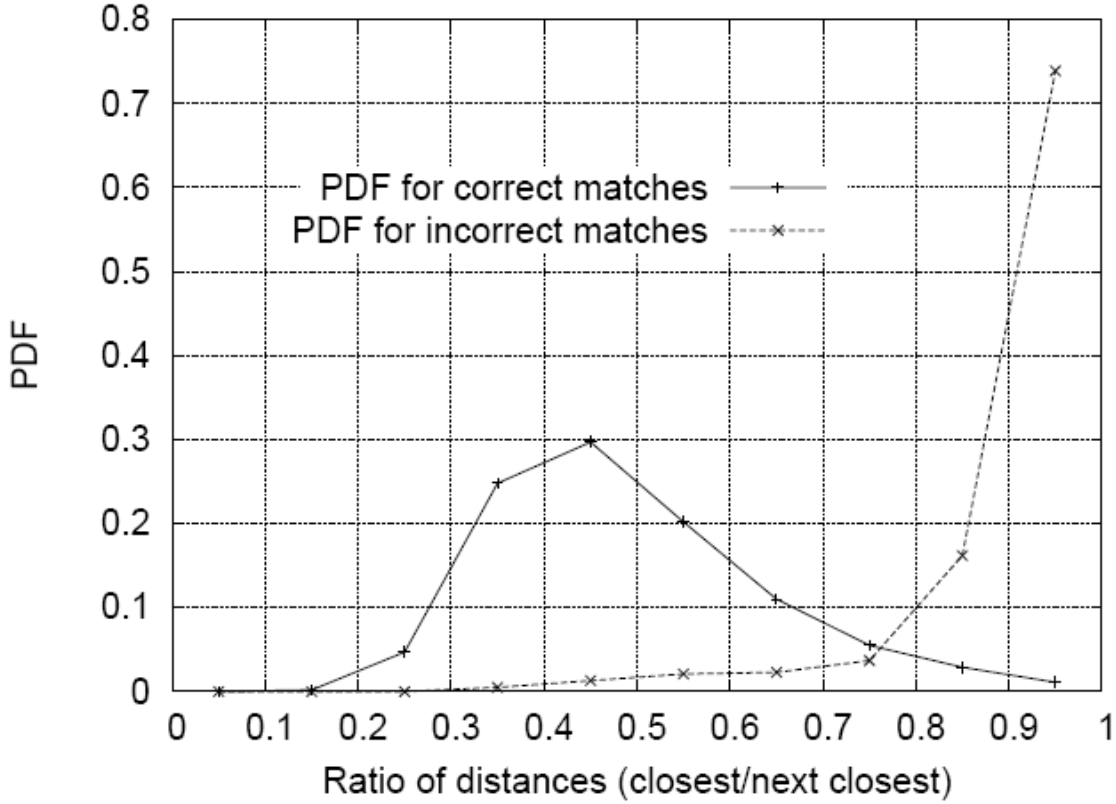
Image 2

- 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.

K. Grauman

Matching SIFT descriptors

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



Lowe IJCV 2004

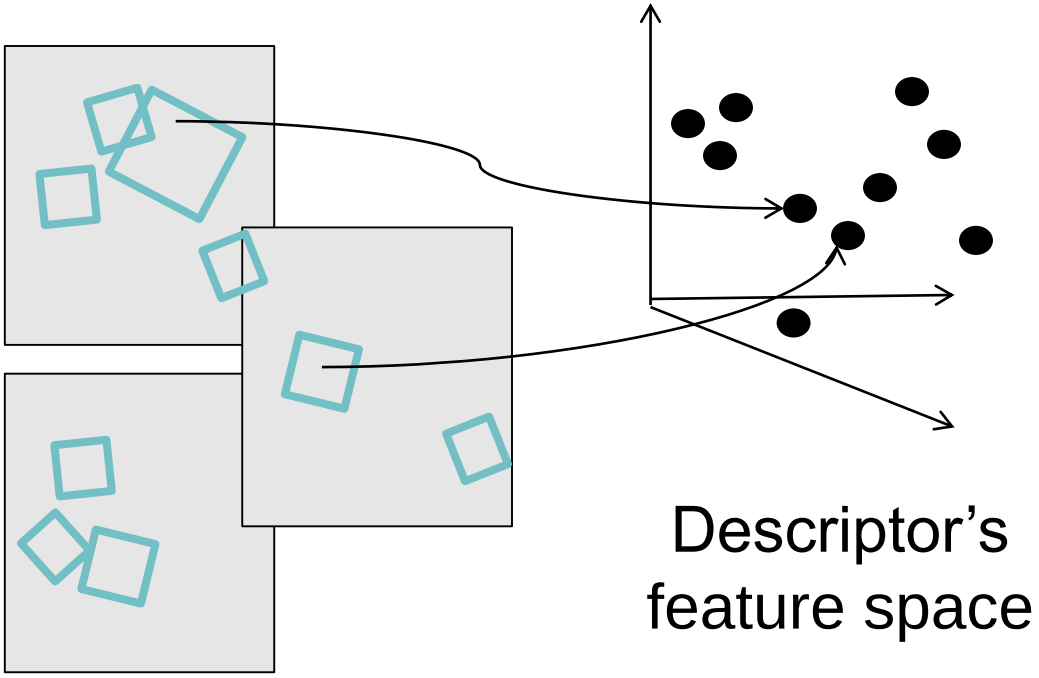
Efficient matching

- 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?

Adapted from K. Grauman

Indexing local features: Setup

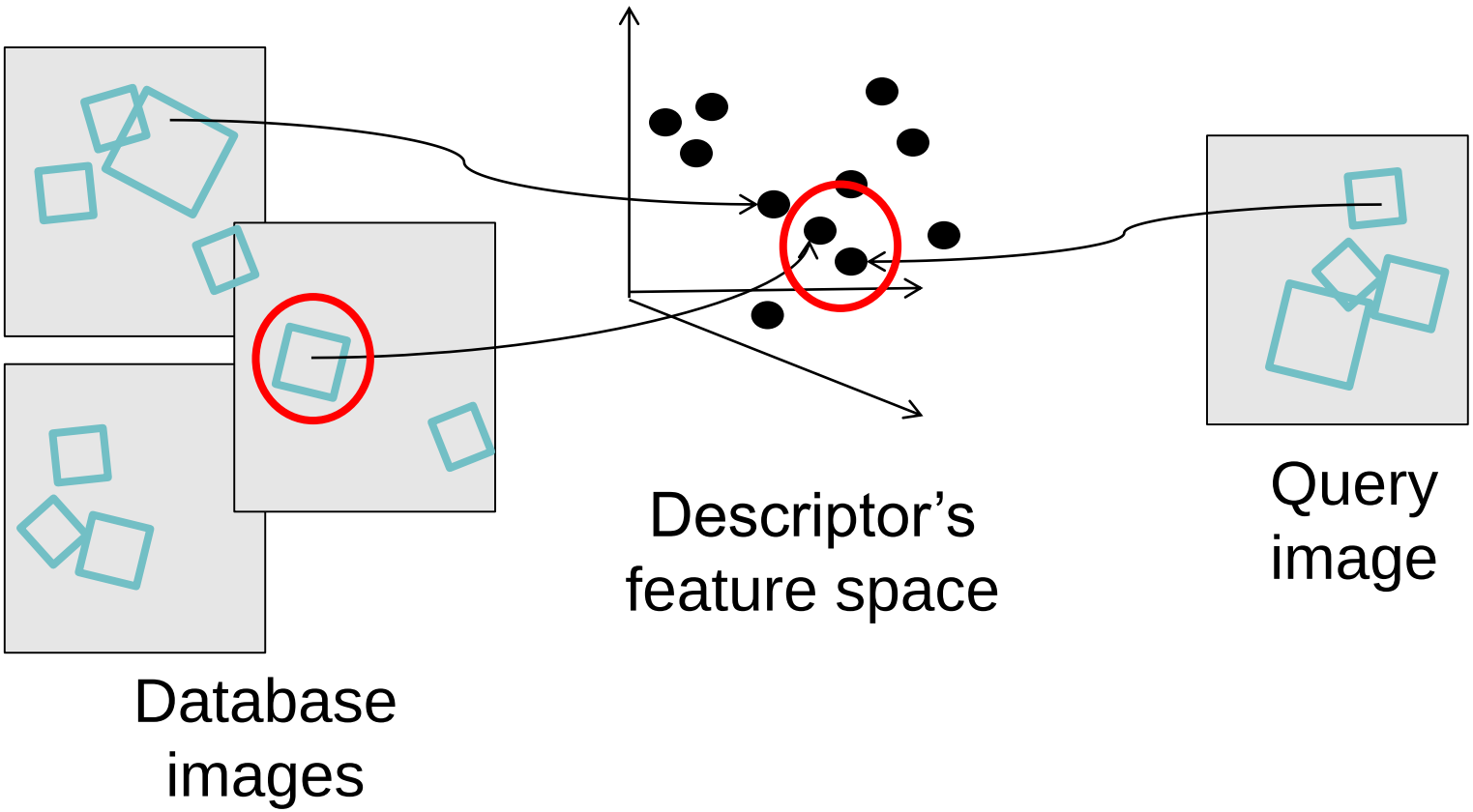
- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



K. Grauman

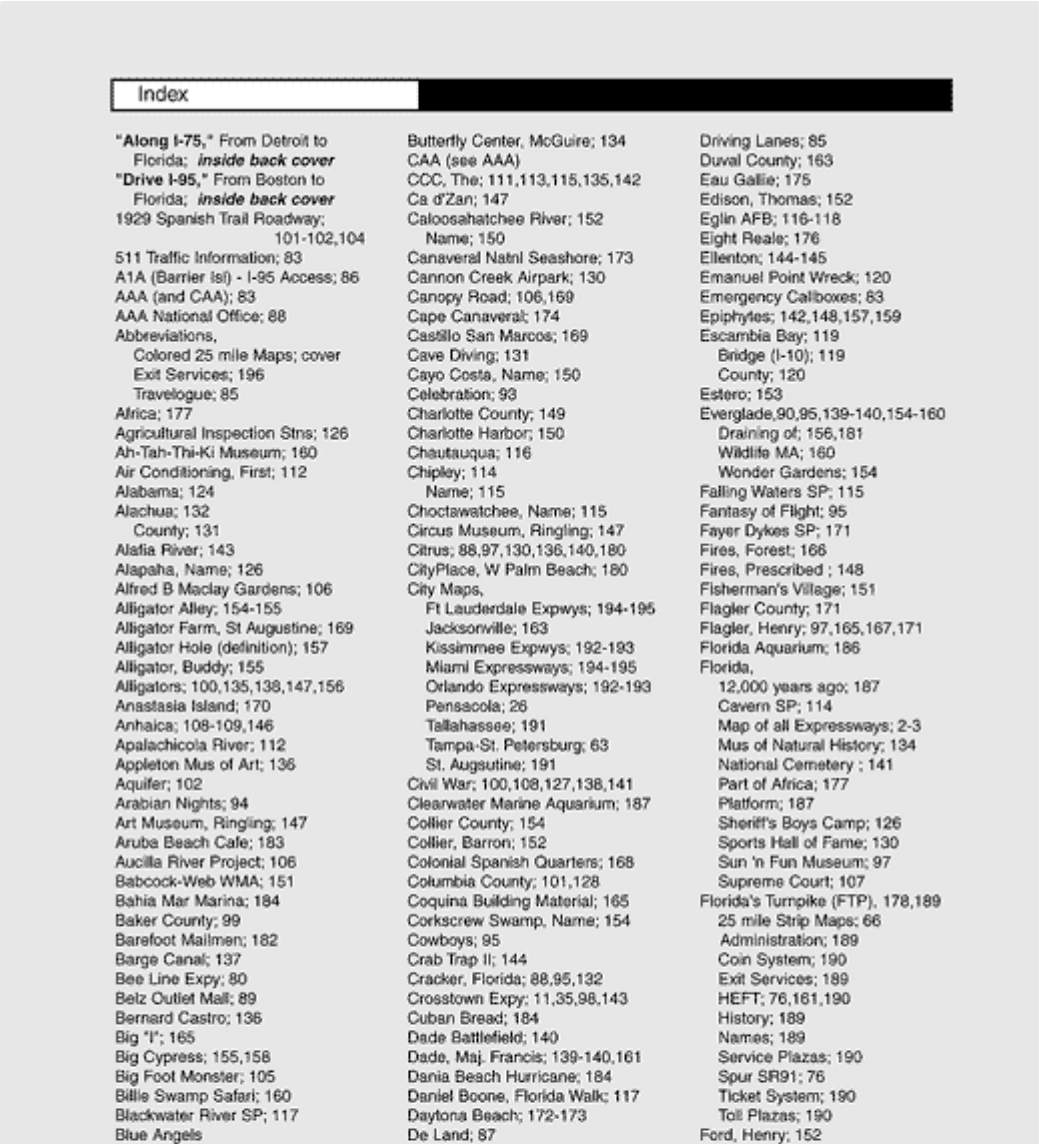
Indexing local features: Setup

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



K. Grauman

Indexing local features: Inverted file index



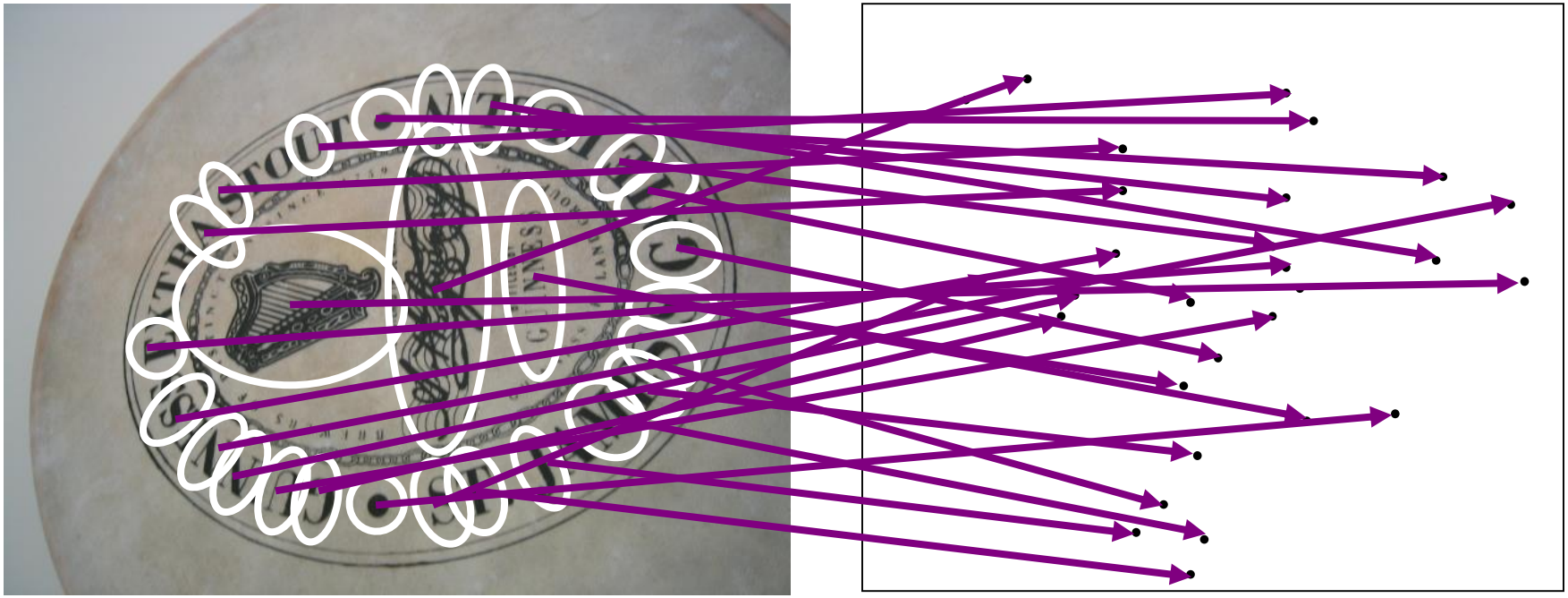
Index

- 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".

K. Grauman

Visual words: main idea

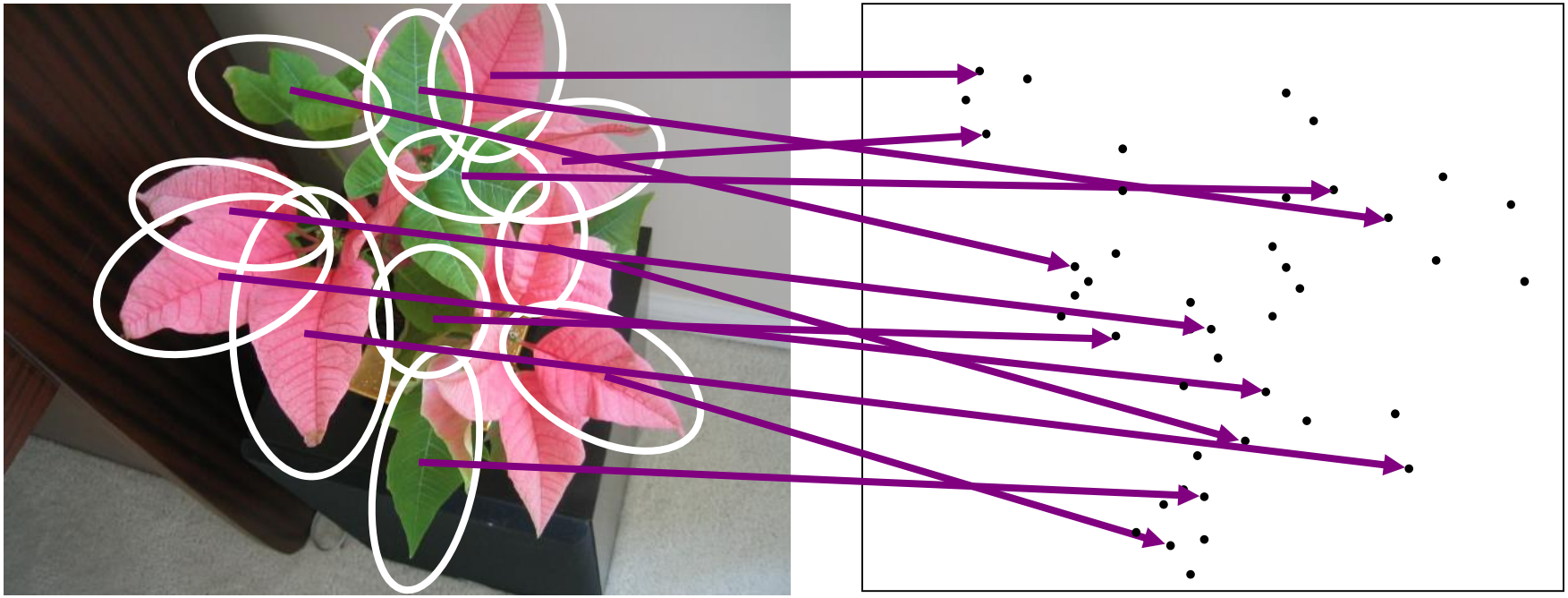
- Extract some local features from a number of images …



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

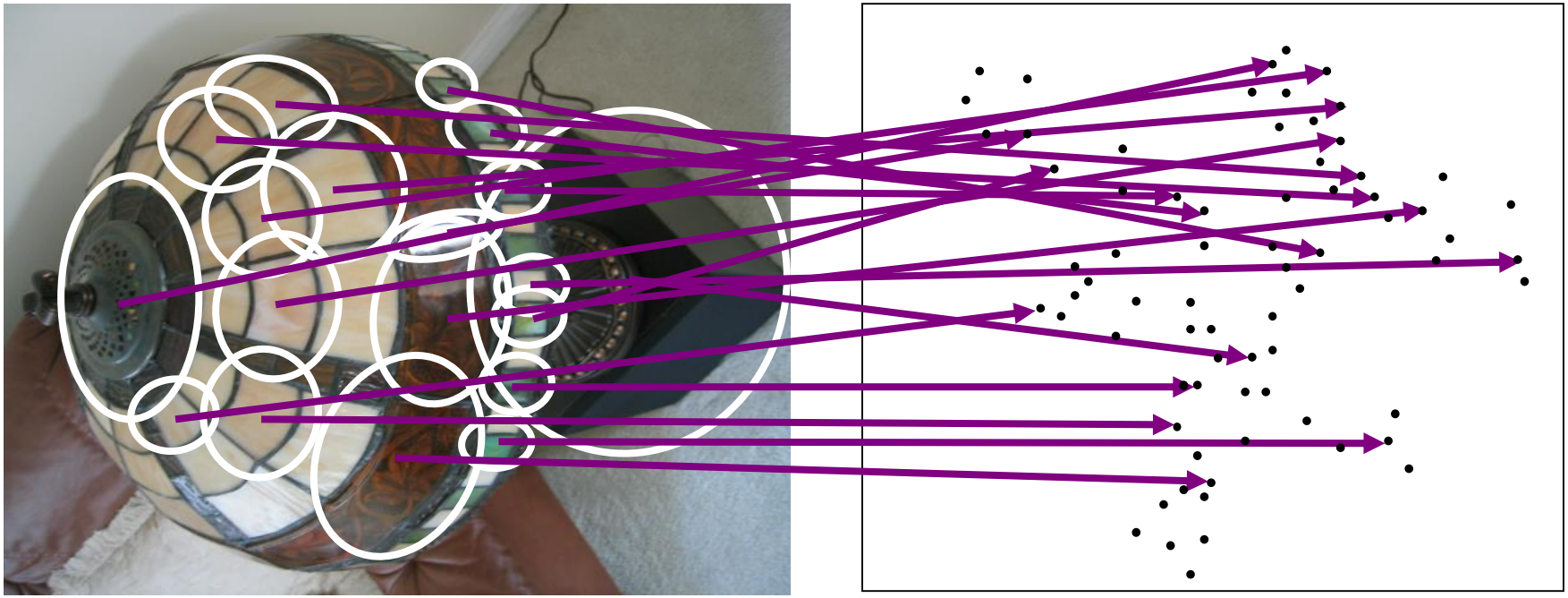
D. Nister, CVPR 2006

Visual words: main idea



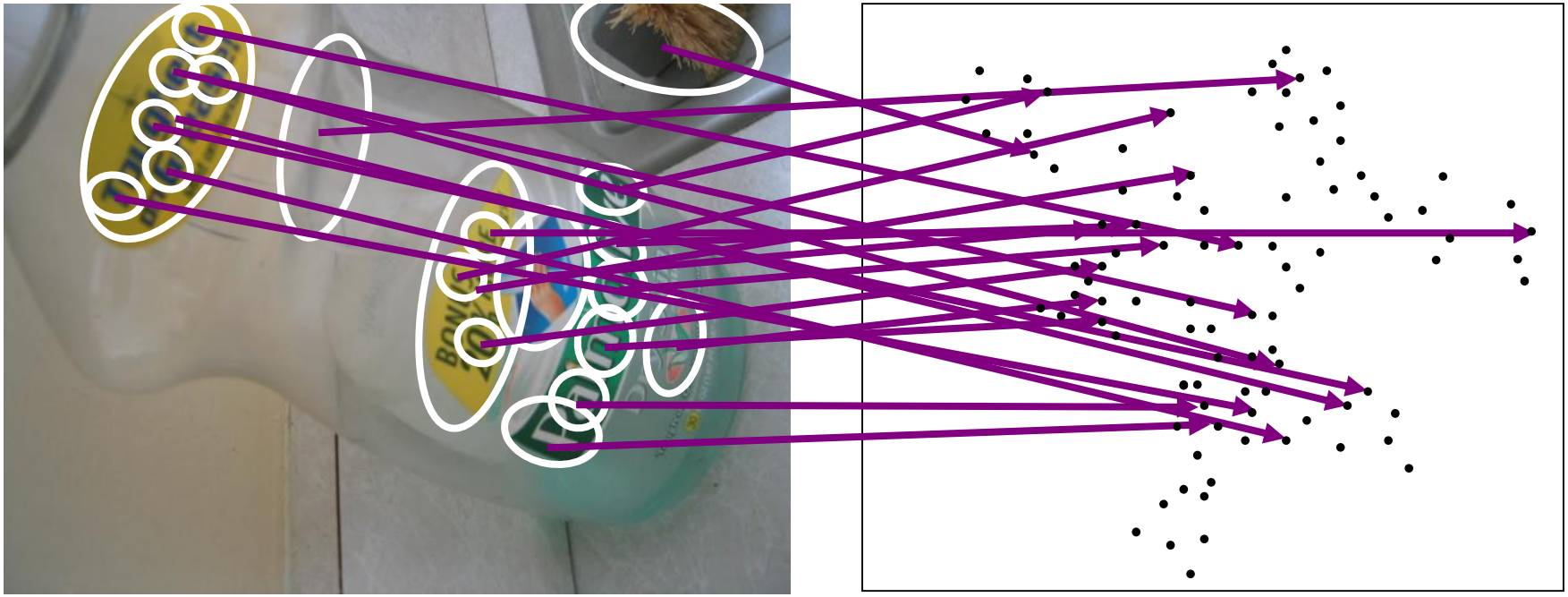
D. Nister, CVPR 2006

Visual words: main idea

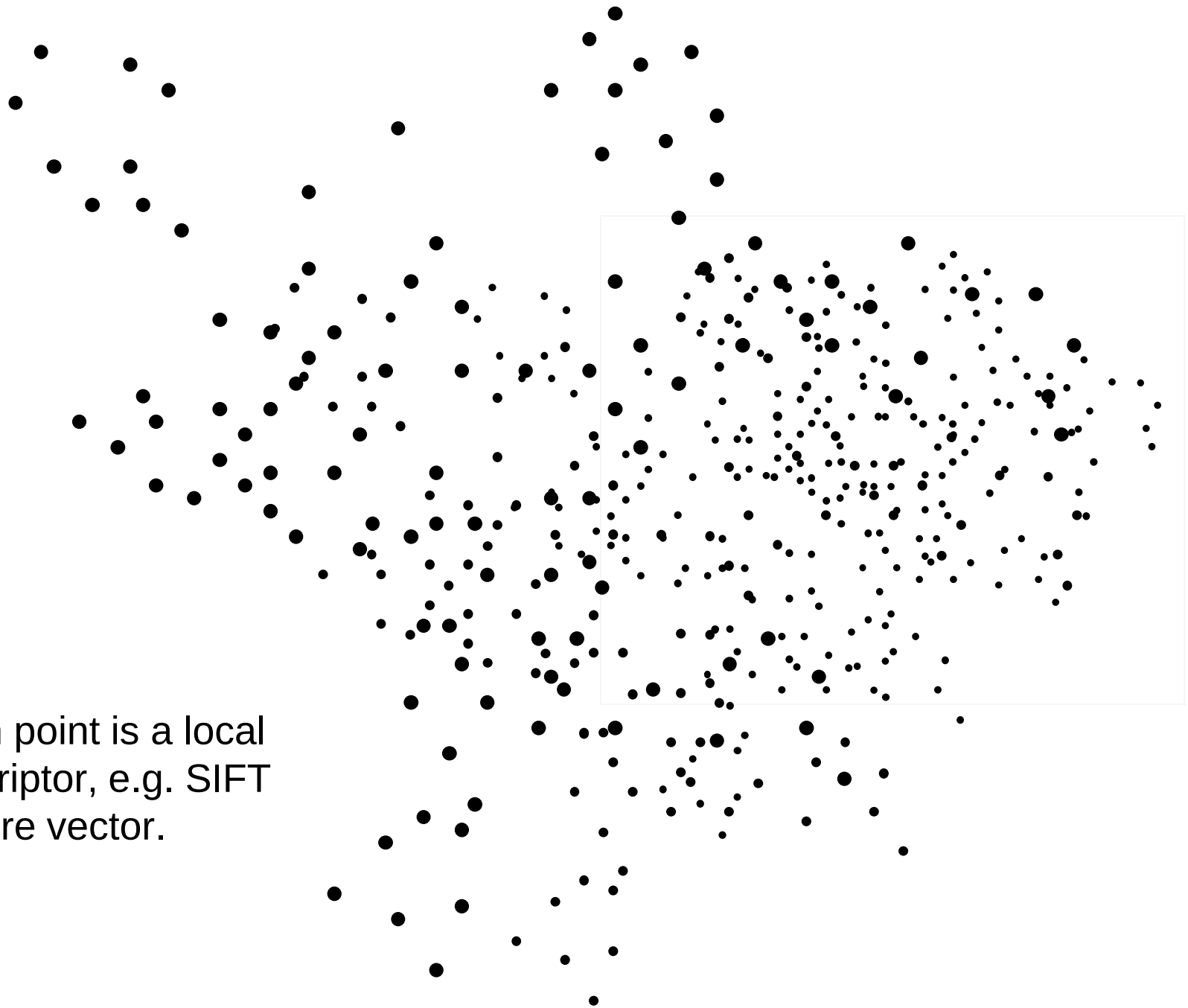


D. Nister, CVPR 2006

Visual words: main idea

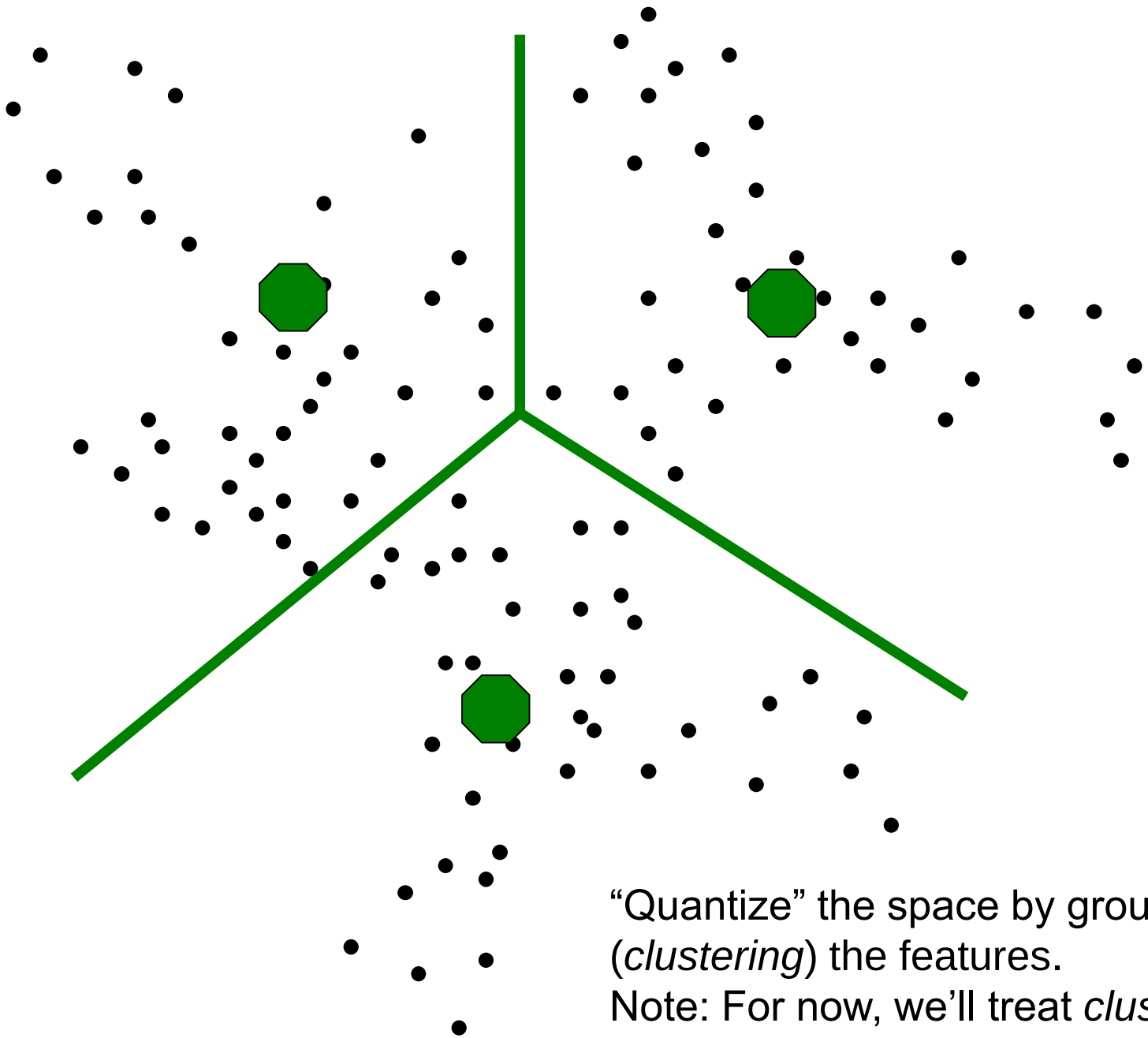


D. Nister, CVPR 2006



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

D. Nister, CVPR 2006

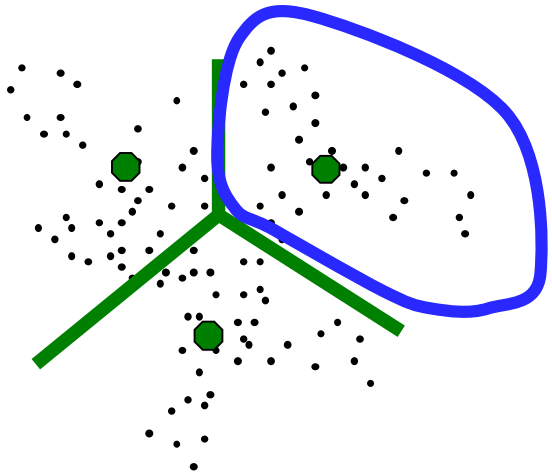


“Quantize” the space by grouping (*clustering*) the features.
Note: For now, we’ll treat *clustering* as a black box.

D. Nister, CVPR 2006

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"



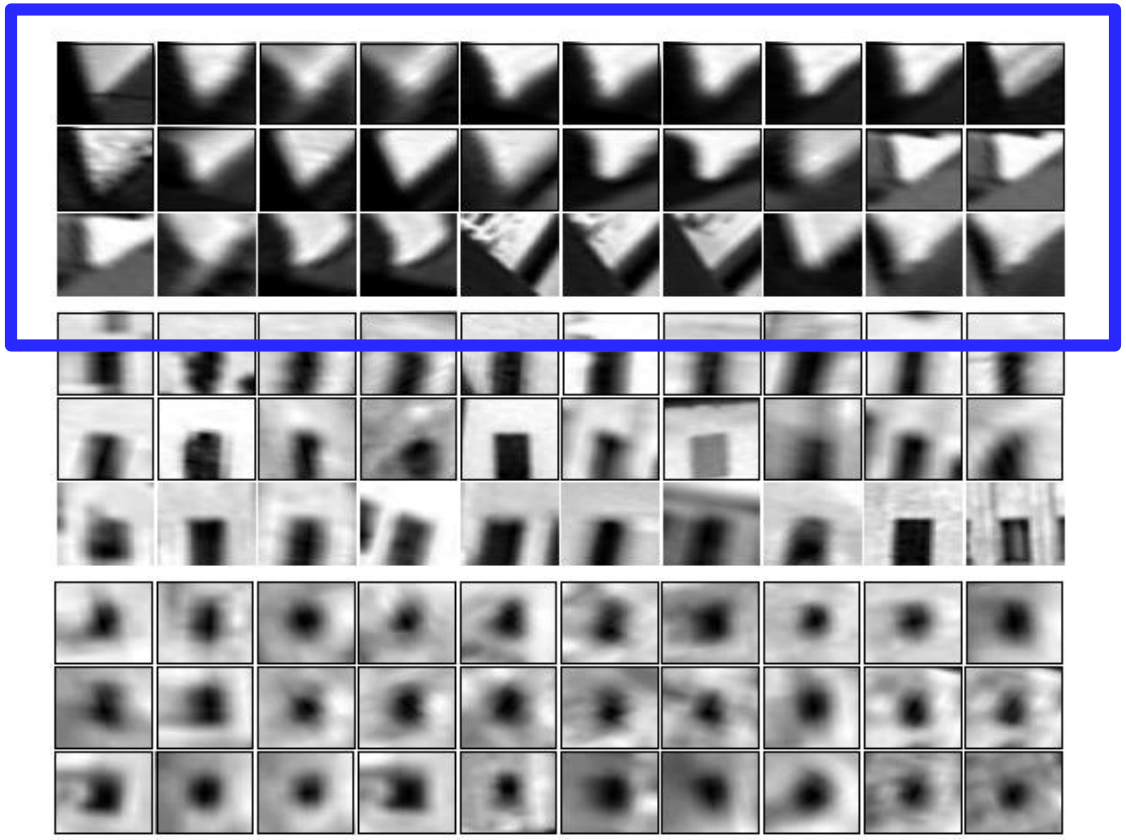
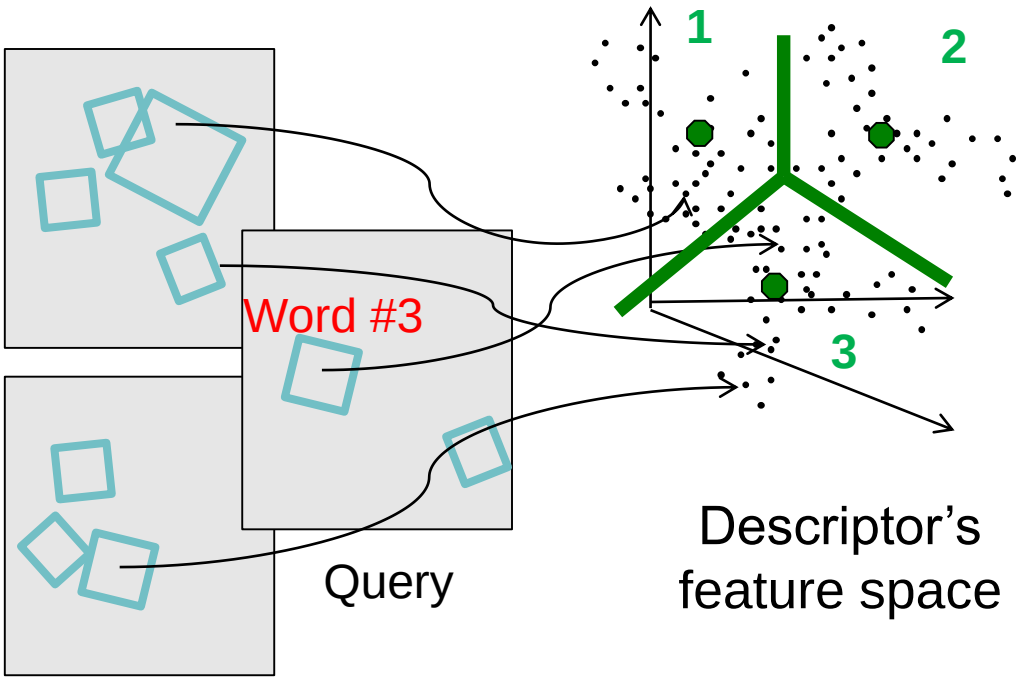


Figure from Sivic & Zisserman, ICCV 2003

Adapted from K. Grauman

Visual words for indexing

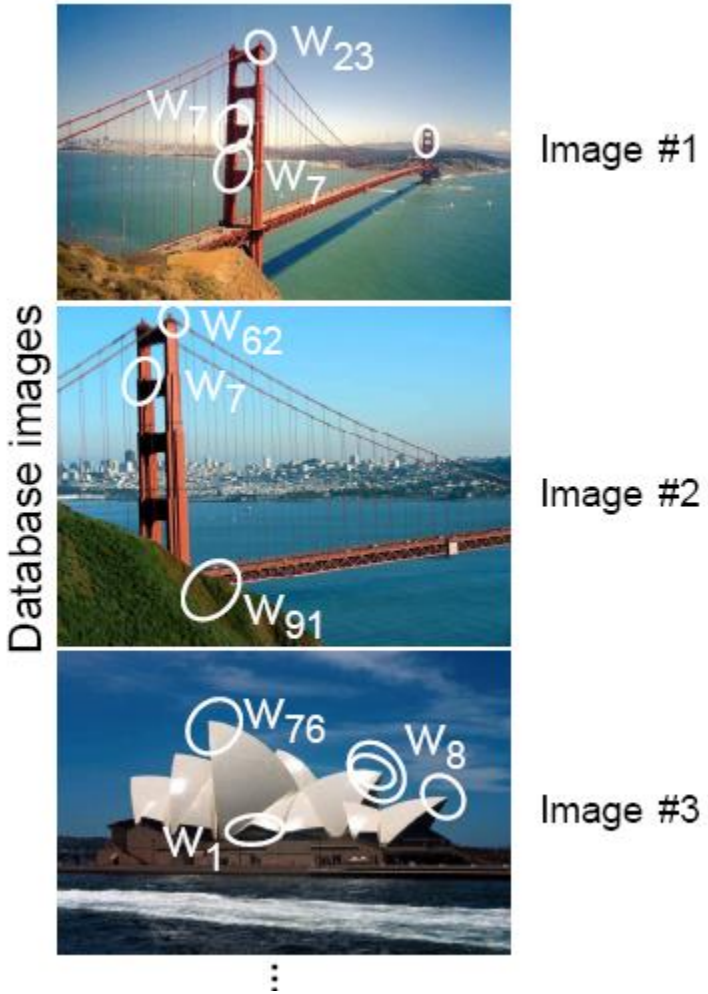
- Map high-dimensional descriptors to tokens/words by quantizing the feature space



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

Adapted from K. Grauman

Inverted file index



Word #	Image #
1	3
2 …	
7	
8	
9	
10 …	
91	

…

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

K. Grauman

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.*

Adapted from K. Grauman

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 r[illegible] our eyes. For a long ti[illegible] retinal image wa[illegible] isual centers i[illegible] s a movie s[illegible] image [illegible] discove[illegible] know th[illegible] perceptio[illegible] more com[illegible] following the [illegible] ath to the various c[illegible] ortex, Hubel and Wiesel ha[illegible] demonstrate that the *message abou[illegible] image falling on the retina undergoes [illegible]-wise analysis in a system of nerve cell[illegible] stored in columns. In this system each c[illegible] has its specific function and is responsible [illegible] a specific detail in the pattern of the retinal image.*

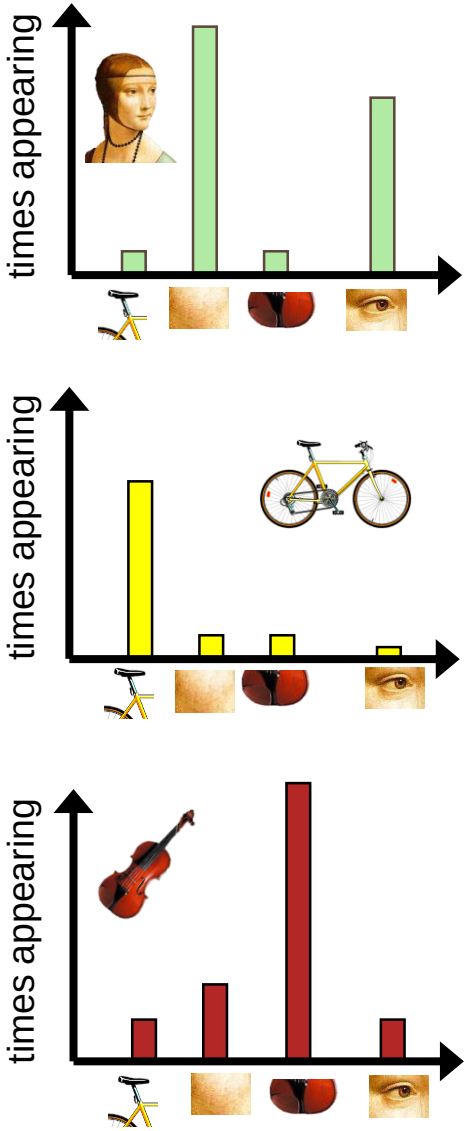
sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

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% [illegible] $750bn, compared w[illegible] $660bn. T[illegible] annoy th[illegible] China's [illegible] deliber[illegible] agrees [illegible] yuan is [illegible] governo[illegible] also need[illegible] demand so [illegible] country. China [illegible] e yuan against the do[illegible] nd permitted it to trade within a narrow [illegible] but the US wants the yuan to be allowed [illegible] de freely. However, Beijing has made it c[illegible] t it will take its time and tread carefully be[illegible] 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.



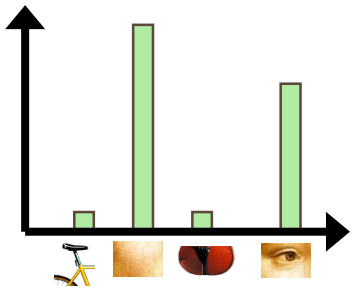
Feature patches:

K. Grauman

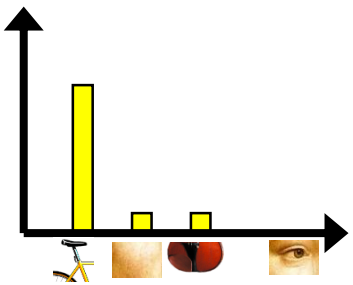
Comparing bags of words

- Rank images by normalized scalar product between their occurrence counts---*nearest neighbor* search for similar images.

[1 8 1 4]



[5 1 1 0]

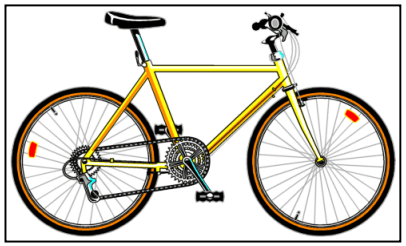


$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

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

for vocabulary of *V* words

$\vec{d}_j$



$\vec{q}$

```
sim(d_j, q) = dot(d_j, q) / (norm(d_j, 2) * norm(q, 2))
```

K. Grauman

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

Adapted from K. Grauman

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. (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

Adapted from K. Grauman

One more trick: *tf-idf* weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- 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

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

Total number of documents in database

Number of documents in which word i occurs

Normalized bag-of-words

Adapted from K. Grauman

Bags of words for content-based image retrieval

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Example

retrieved shots

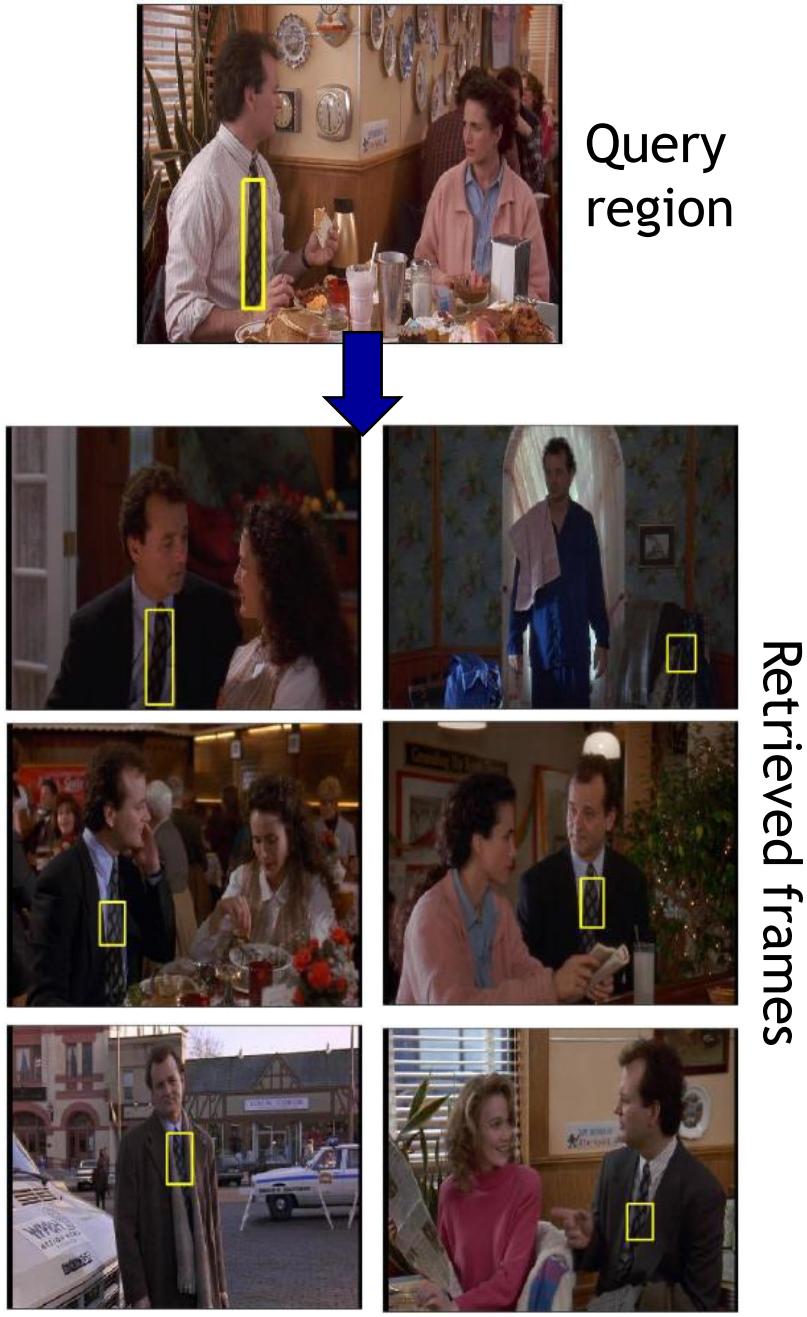
Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

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



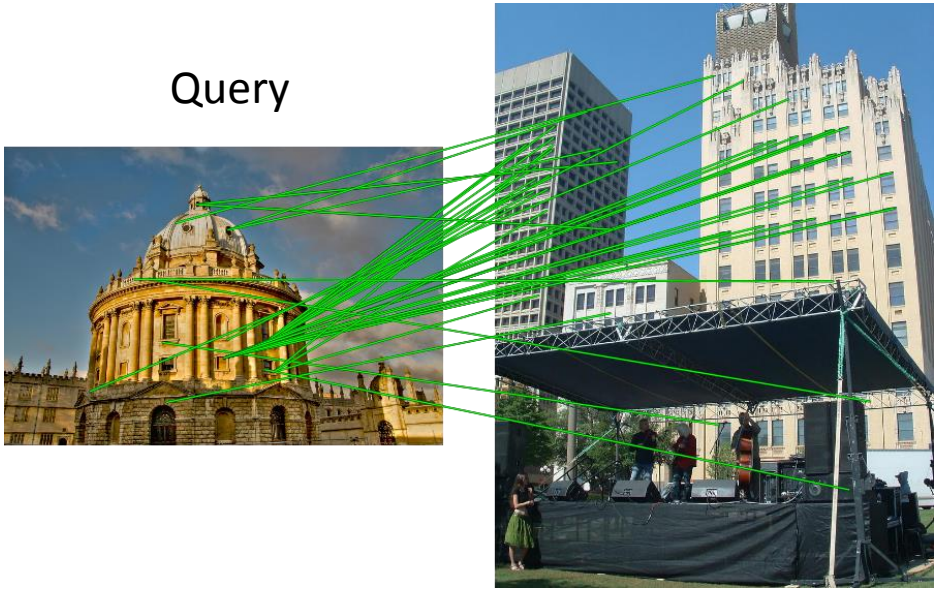
K. Grauman

Preview: Spatial Verification

Query

DB image with high BoW similarity

Query



DB image with high BoW similarity

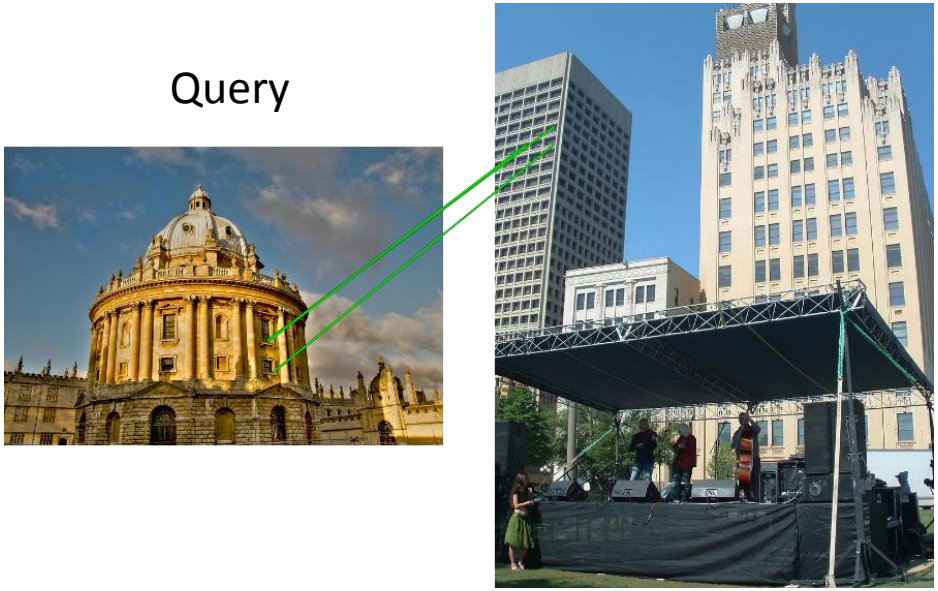
Both image pairs have many visual words in common.

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

[Quack, Leibe, Van Gool, CIVR'08]

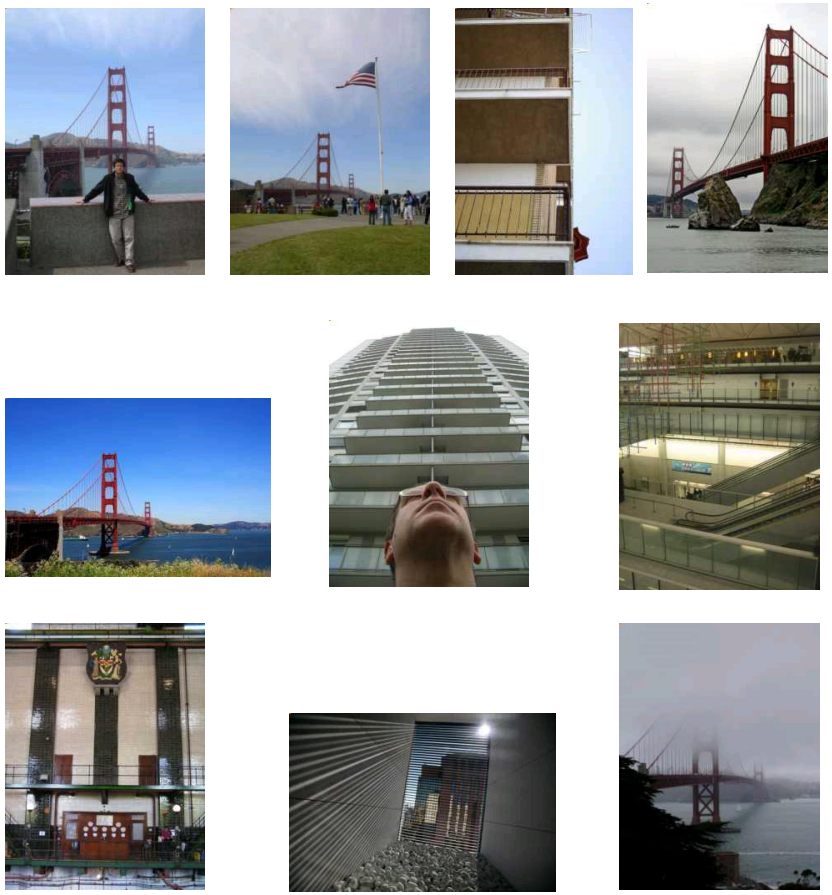
Scoring retrieval quality

Query

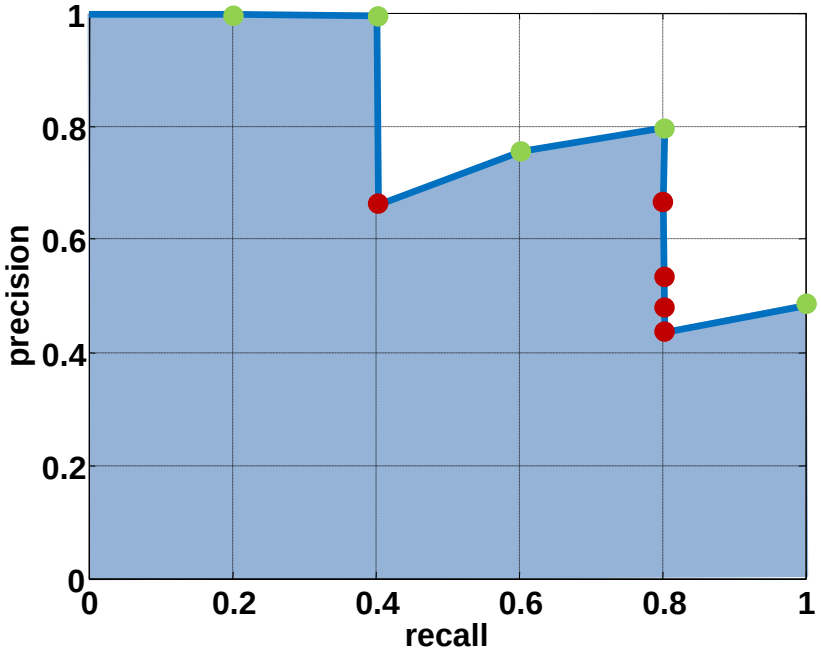
Database size: 10 images

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

Results (ordered):



precision = #relevant / #returned
recall = #relevant / #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

Adapted from K. Grauman