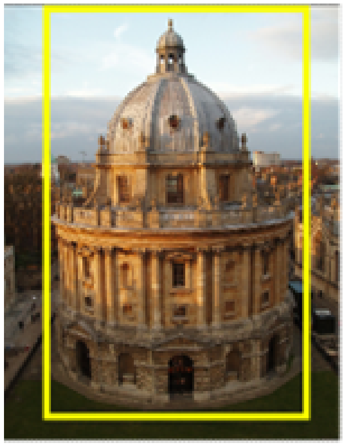
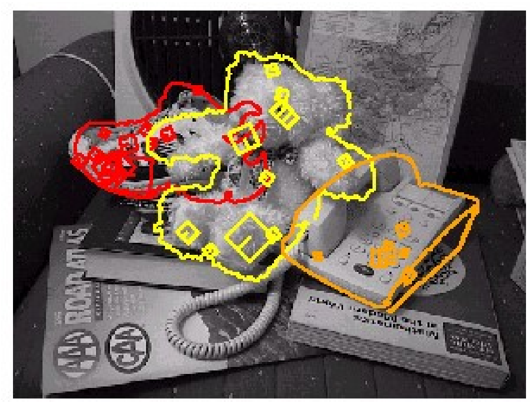


Matching objects in images

Kristen Grauman



Recognizing or retrieving specific objects

Example I: Visual search in feature films

Visually defined query

"Groundhog Day" [Rammis, 1993]

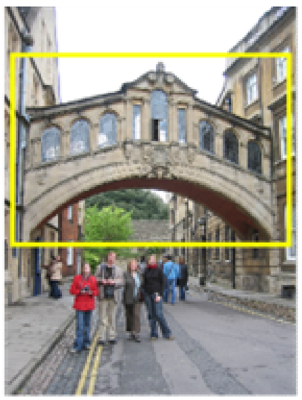
"Find this clock"

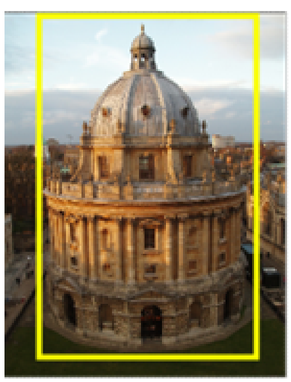
"Find this place"

Slide credit: J. Sivic

Recognizing or retrieving specific objects

Example II: Search photos on the web for particular places

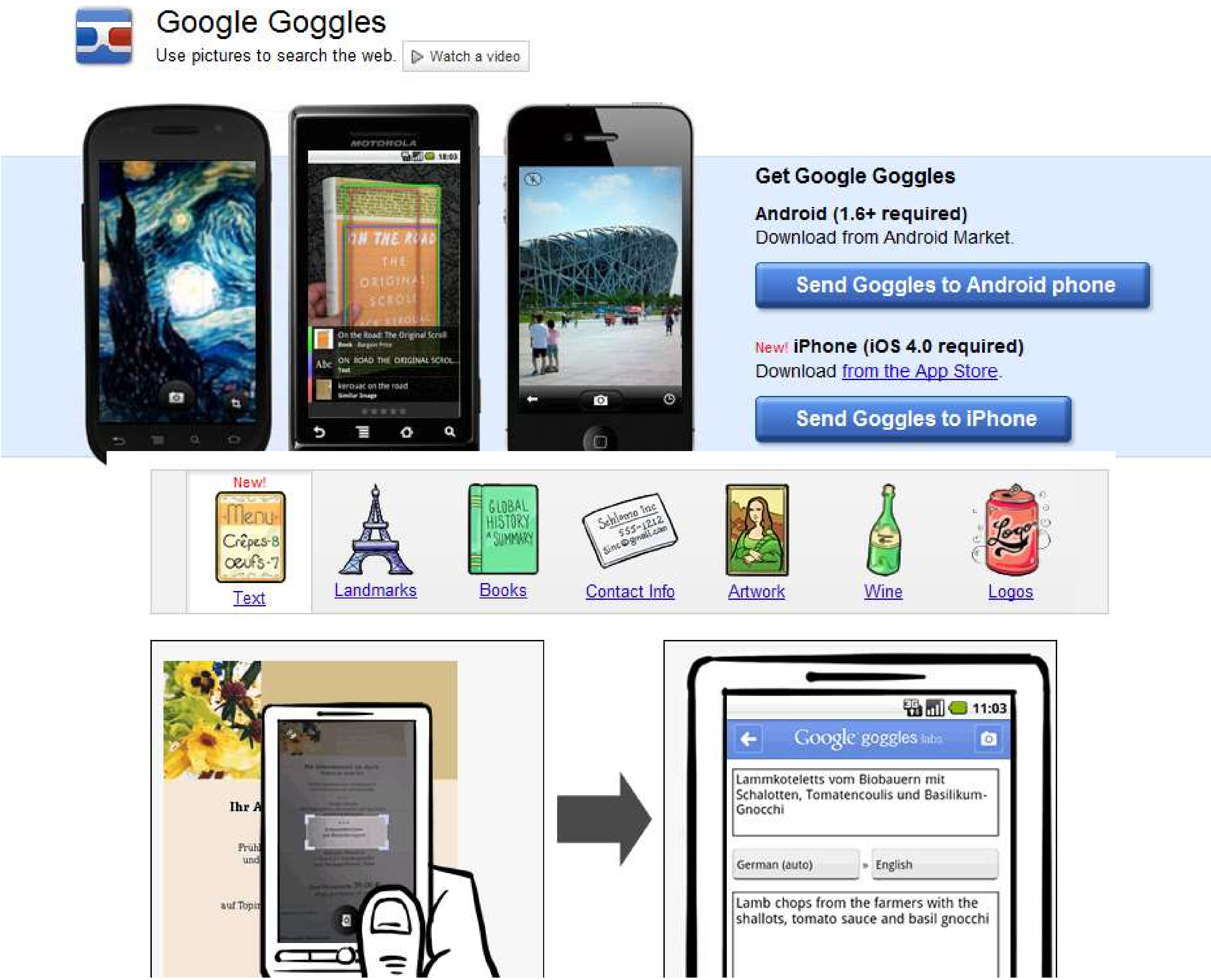




Find these landmarks

...in these images and 1M more

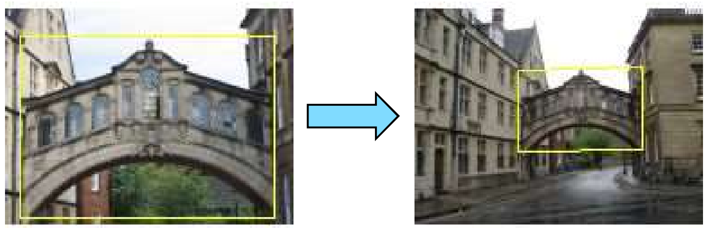
Slide credit: J. Sivic



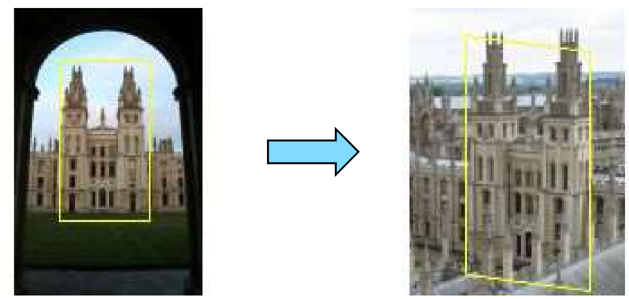
Google Goggles
Use pictures to search the web.
Watch a video
MOTOROLA
Get Google Goggles
Android (1.6+ required)
Download from Android Market.
Send Goggles to Android phone
New! iPhone (iOS 4.0 required)
Download from the App Store.
Send Goggles to iPhone
New!
Text
Landmarks
Books
Contact Info
Artwork
Wine
Logos
11:03
Google goggles labs
Lammkoteletts vom Biobauern mit Schalotten, Tomatencoulis und Basilikum-Gnocchi
German (auto)
English
Lamb chops from the farmers with the shallots, tomato sauce and basil gnocchi

Why is it difficult?

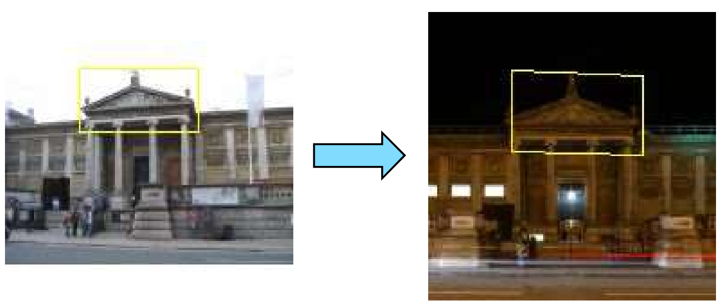
Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion



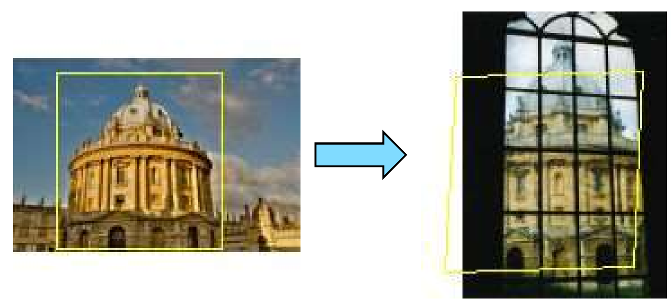
Scale



Viewpoint



Lighting



Occlusion

We can't expect to match such varied instances with a single global template...

Slide credit: J. Sivic

Text retrieval vs. image search

- What makes the problems similar, different?

Matching objects in images

- Motivation – visual search
- Visual words
  - quantization, index, bags of words
- Spatial verification
  - affine; RANSAC, Hough
- Other text retrieval tools
  - tf-idf, query expansion
- Example applications

Invariant local features

Image 1

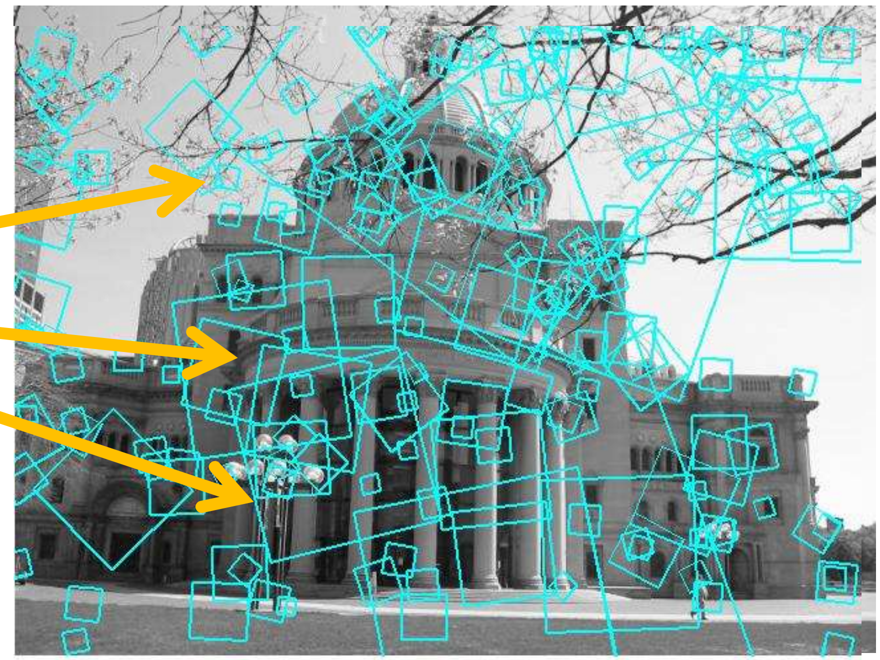


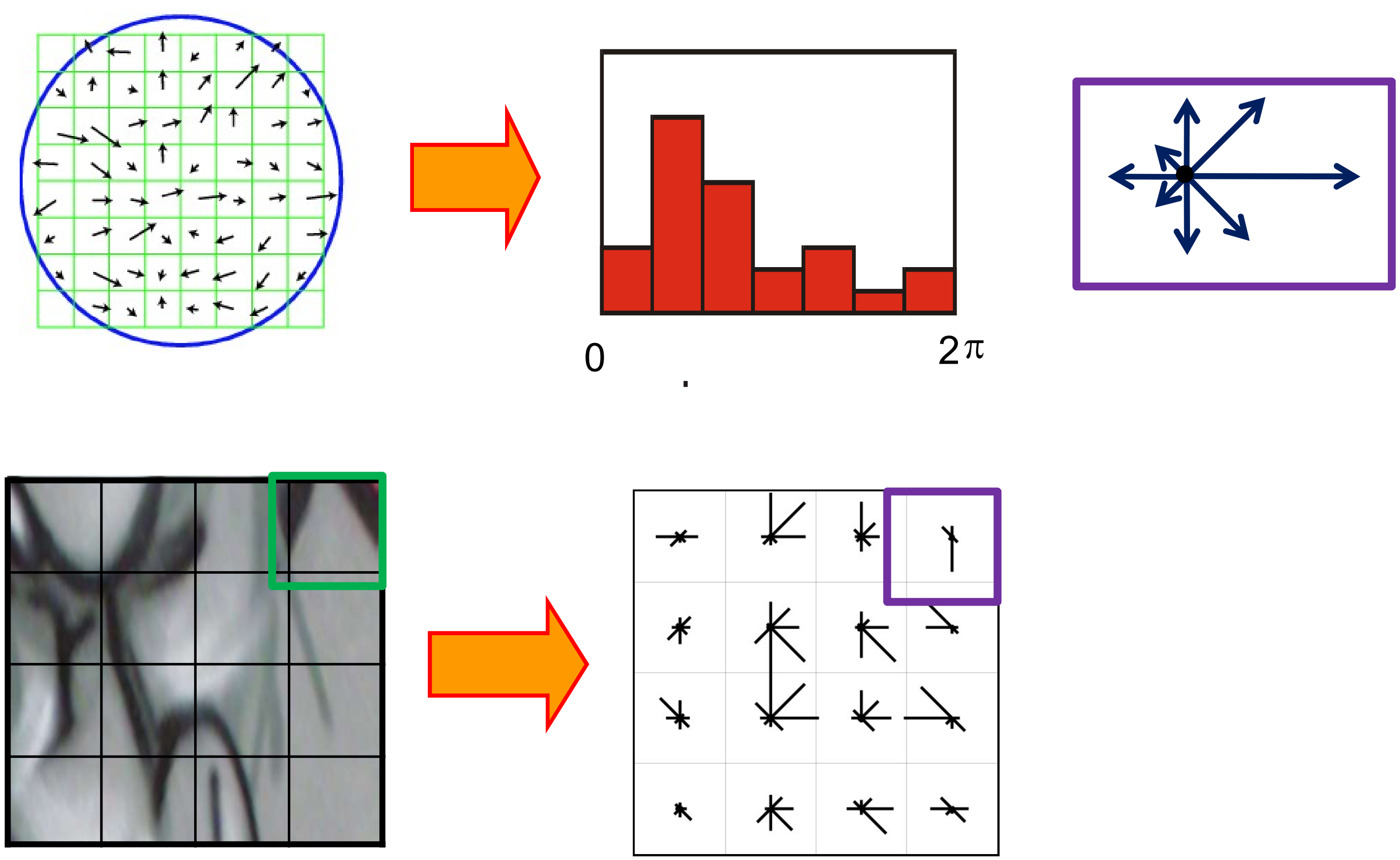
Image 2

We can detect features *independently* per image that are scale, translation, and rotation *invariant*.

Kristen Grauman

Local feature descriptors

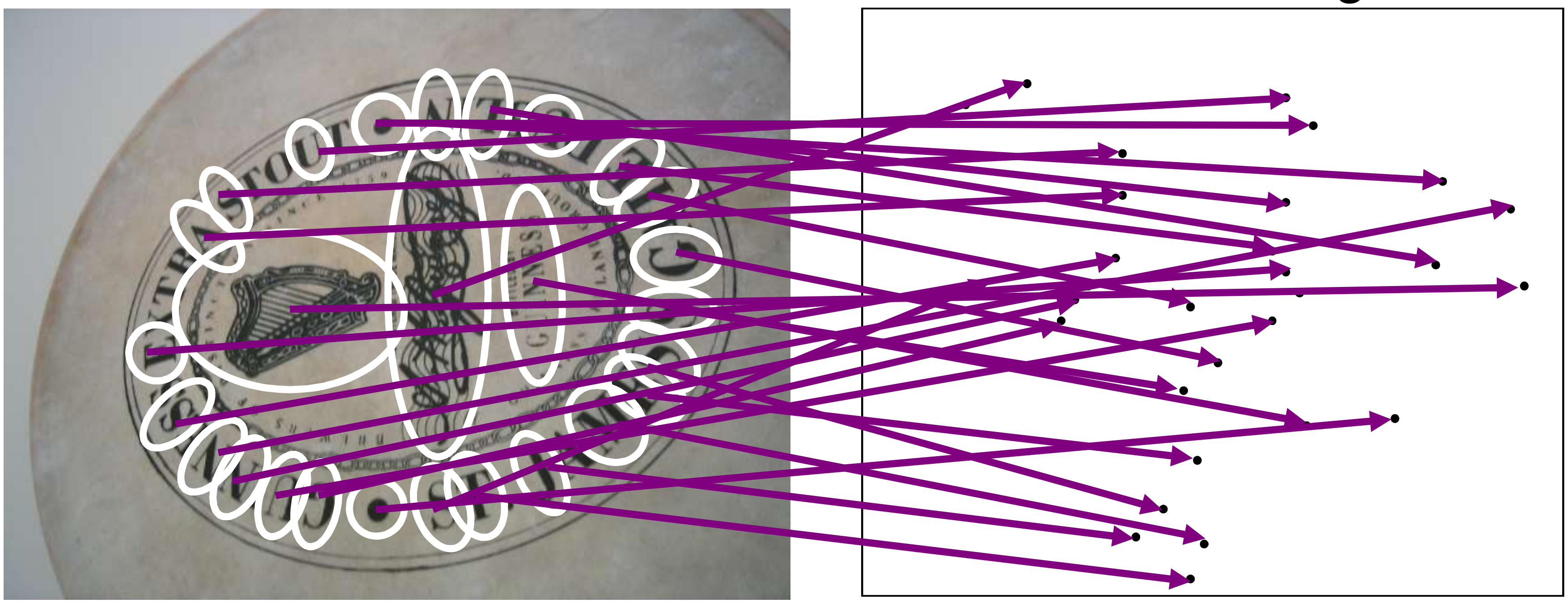
- SIFT [Lowe 2004] : Use histograms to bin pixels within sub-patches according to their orientation.



Kristen Grauman

Visual words: main idea

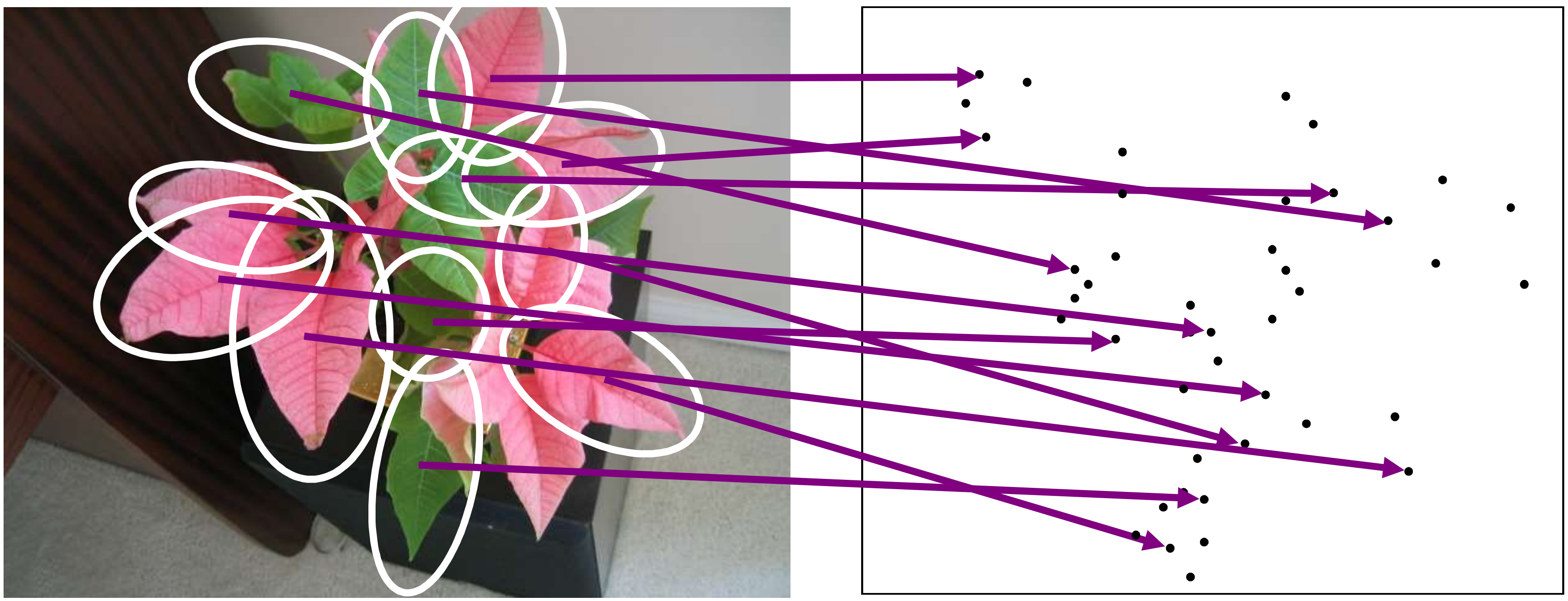
- Extract some local features from a number of images …



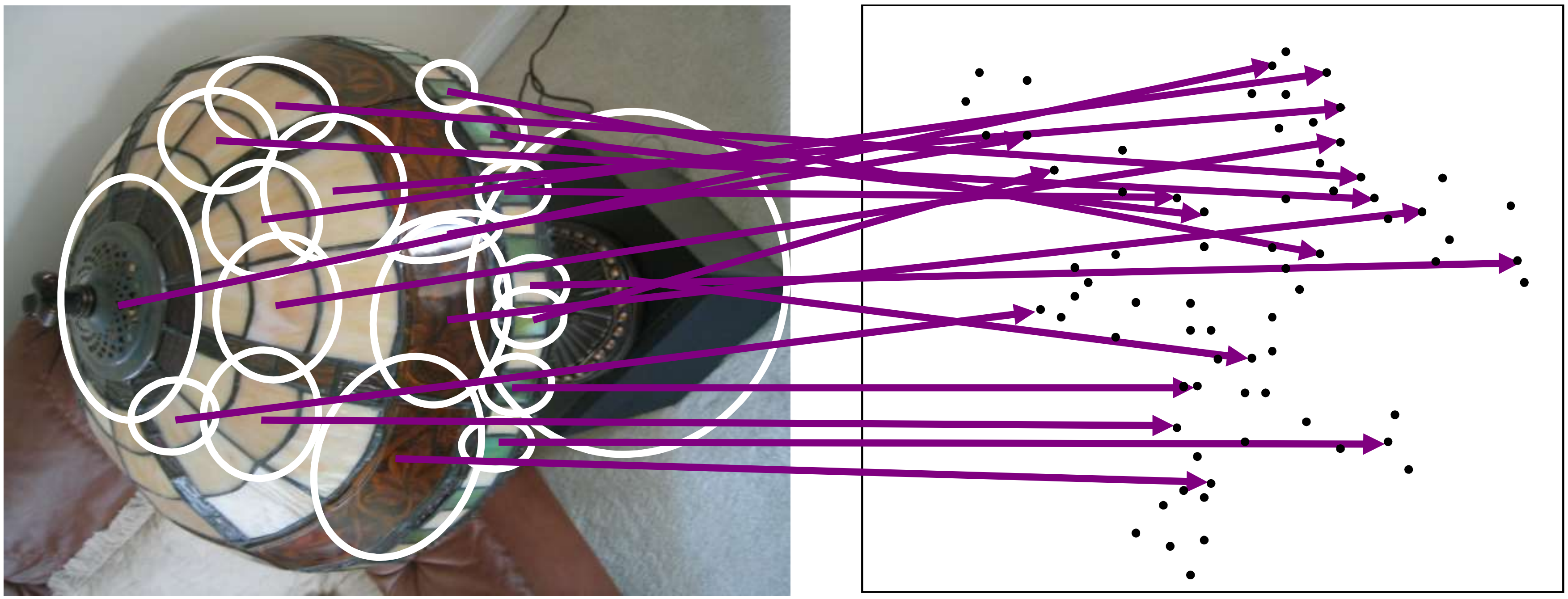
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

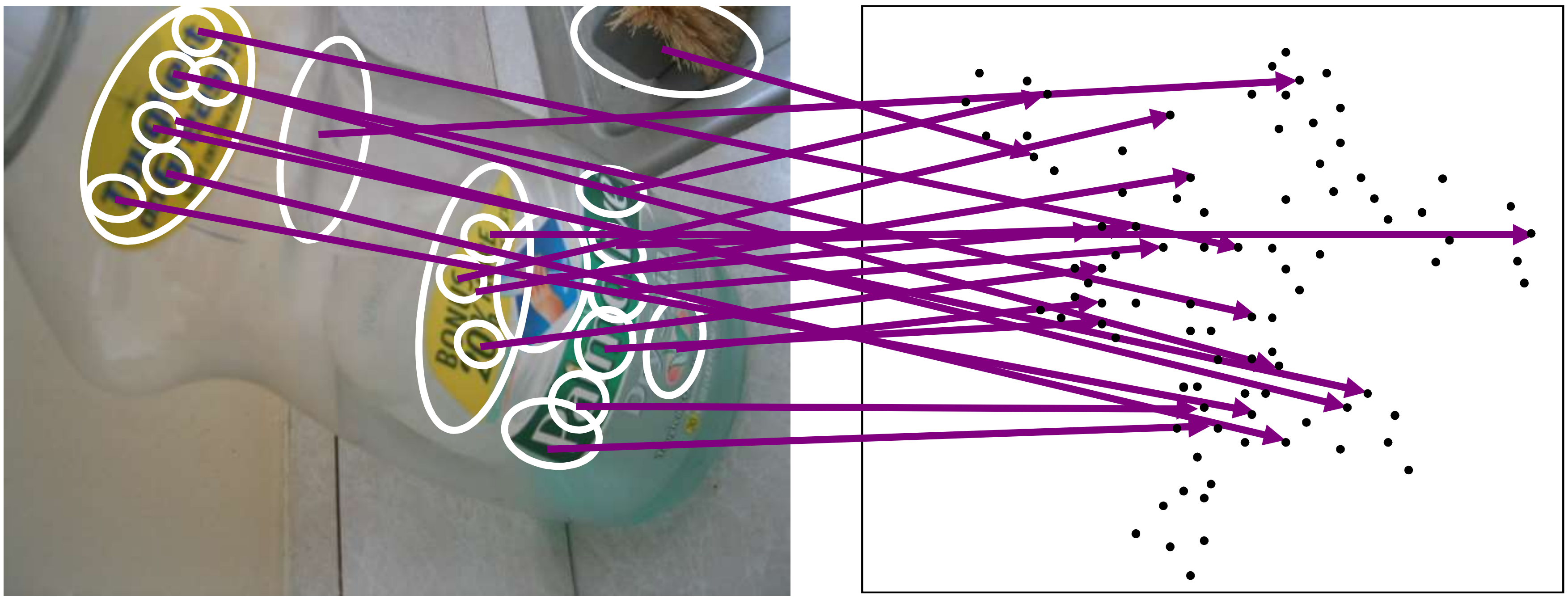
Visual words: main idea

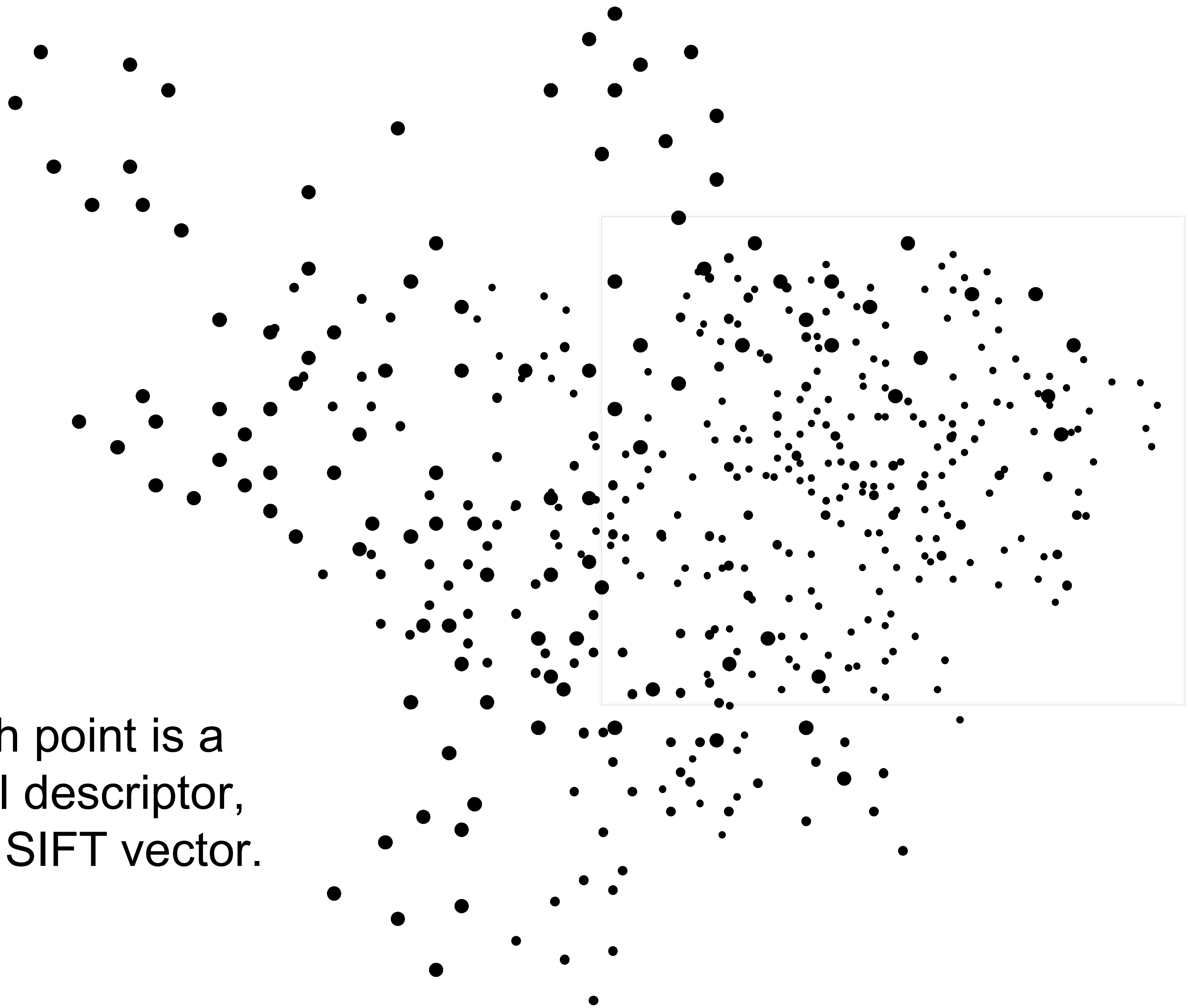


Visual words: main idea

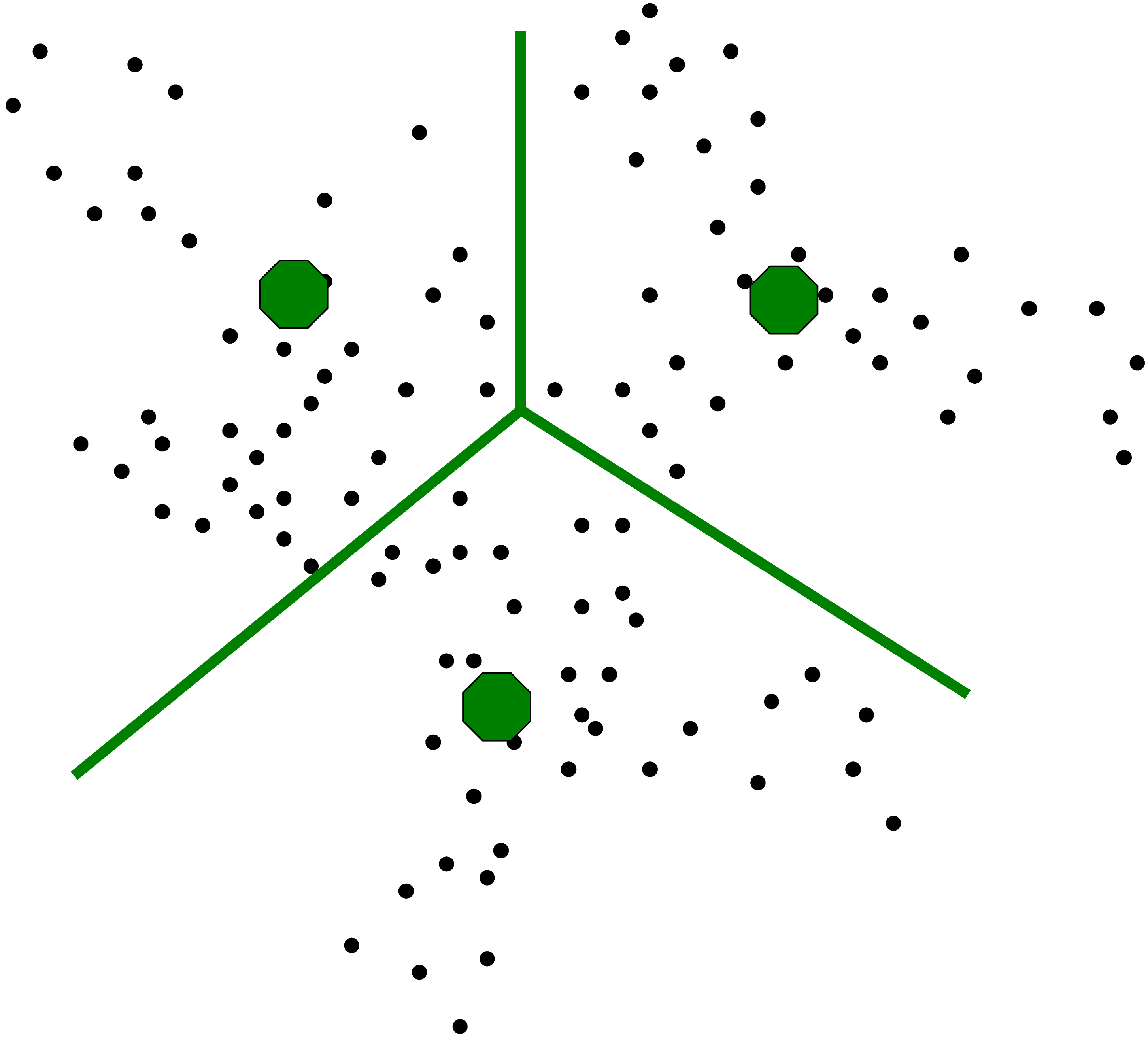


Visual words: main idea



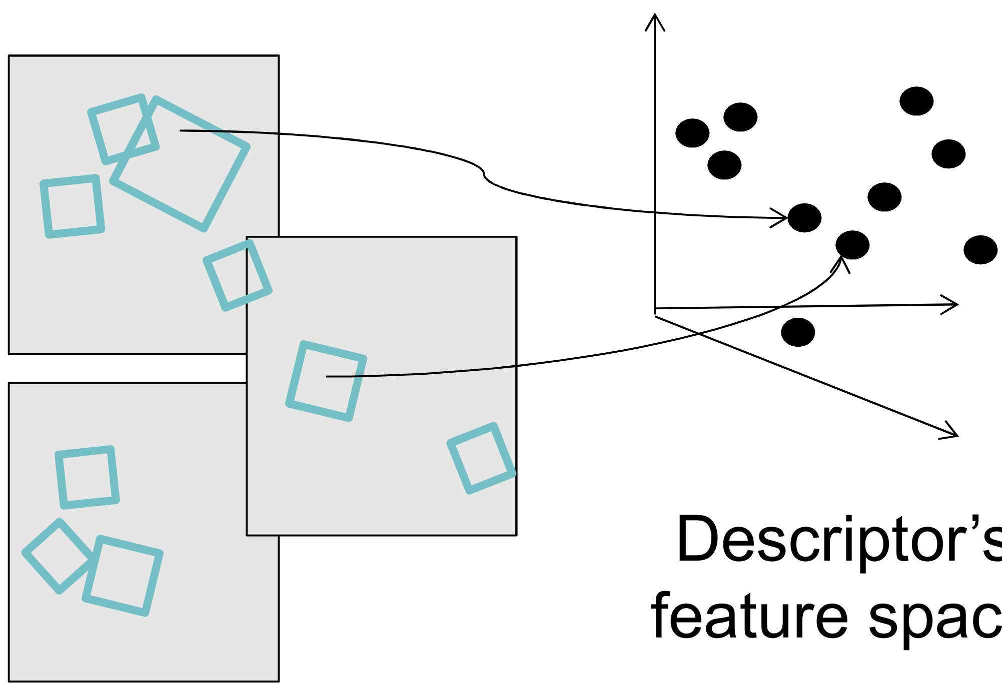


Each point is a local descriptor, e.g. SIFT vector.



Indexing local features

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

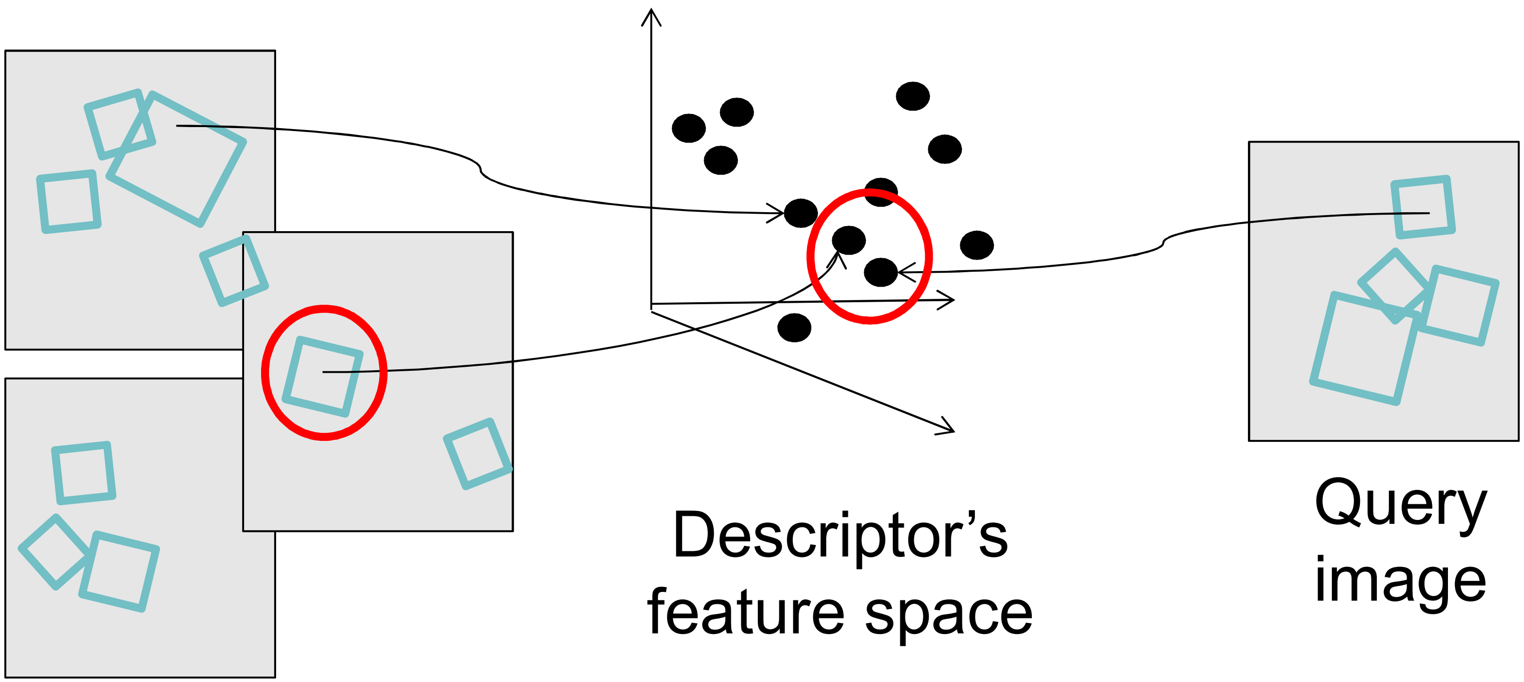


Descriptor's feature space

Kristen Grauman

Indexing local features

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



Database images

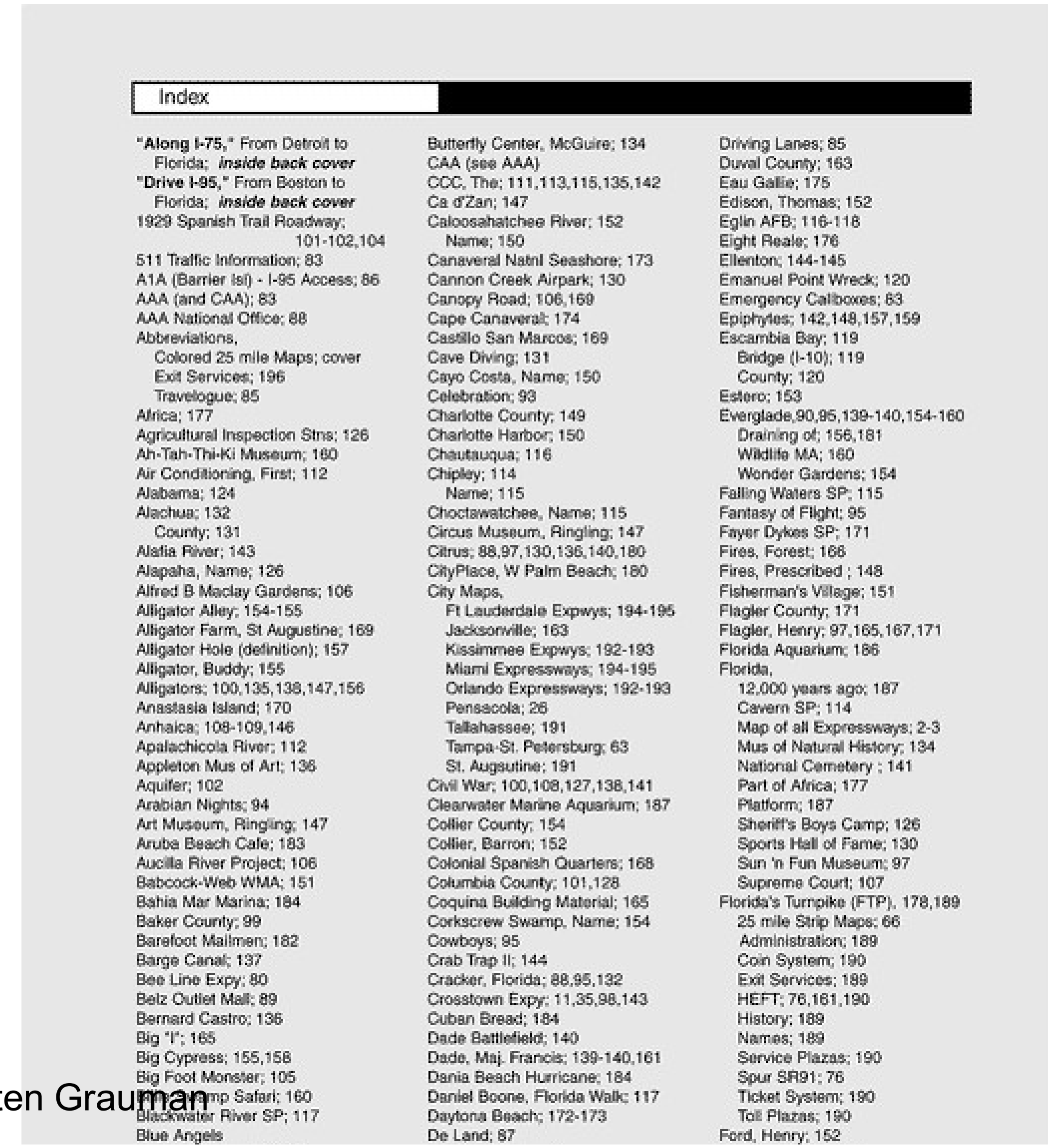
Descriptor's feature space

Query image

Easily can have millions of features to search!

Kristen Grauman

Indexing local features: inverted file index

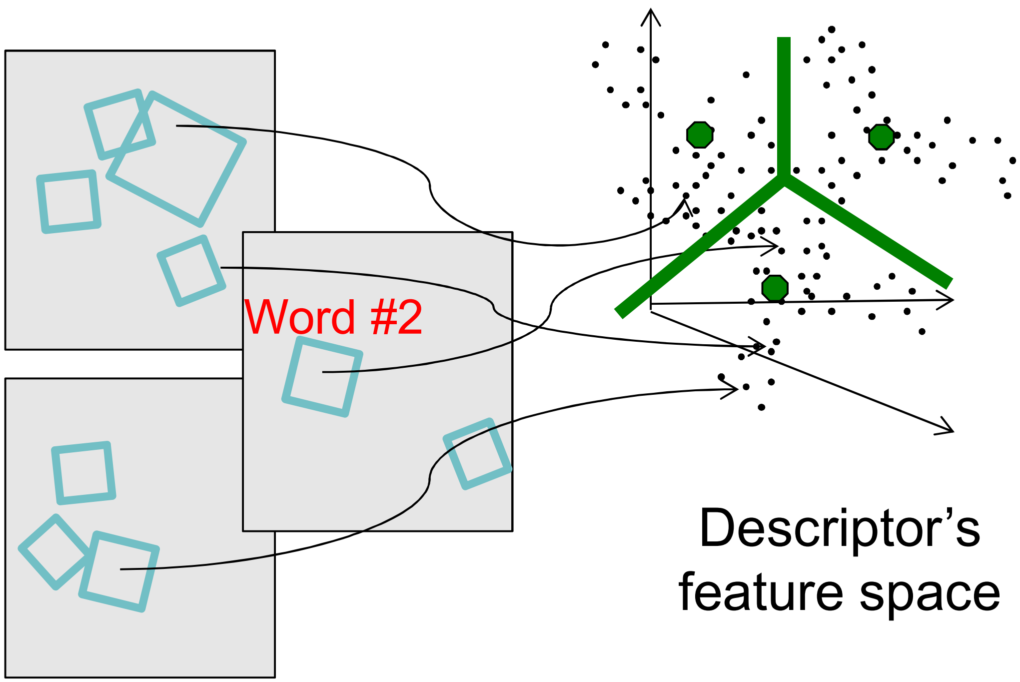


- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

Kristen Grauman

Visual words

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



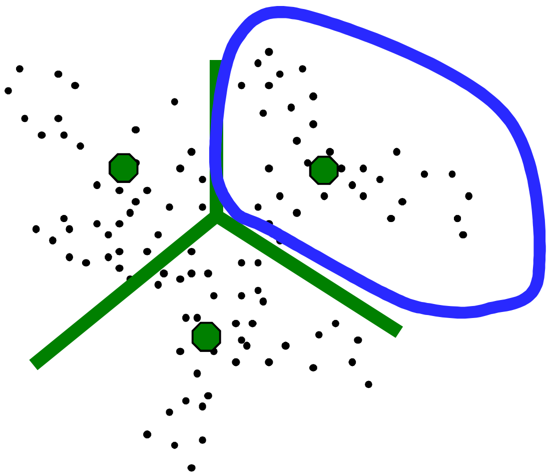
Descriptor's feature space

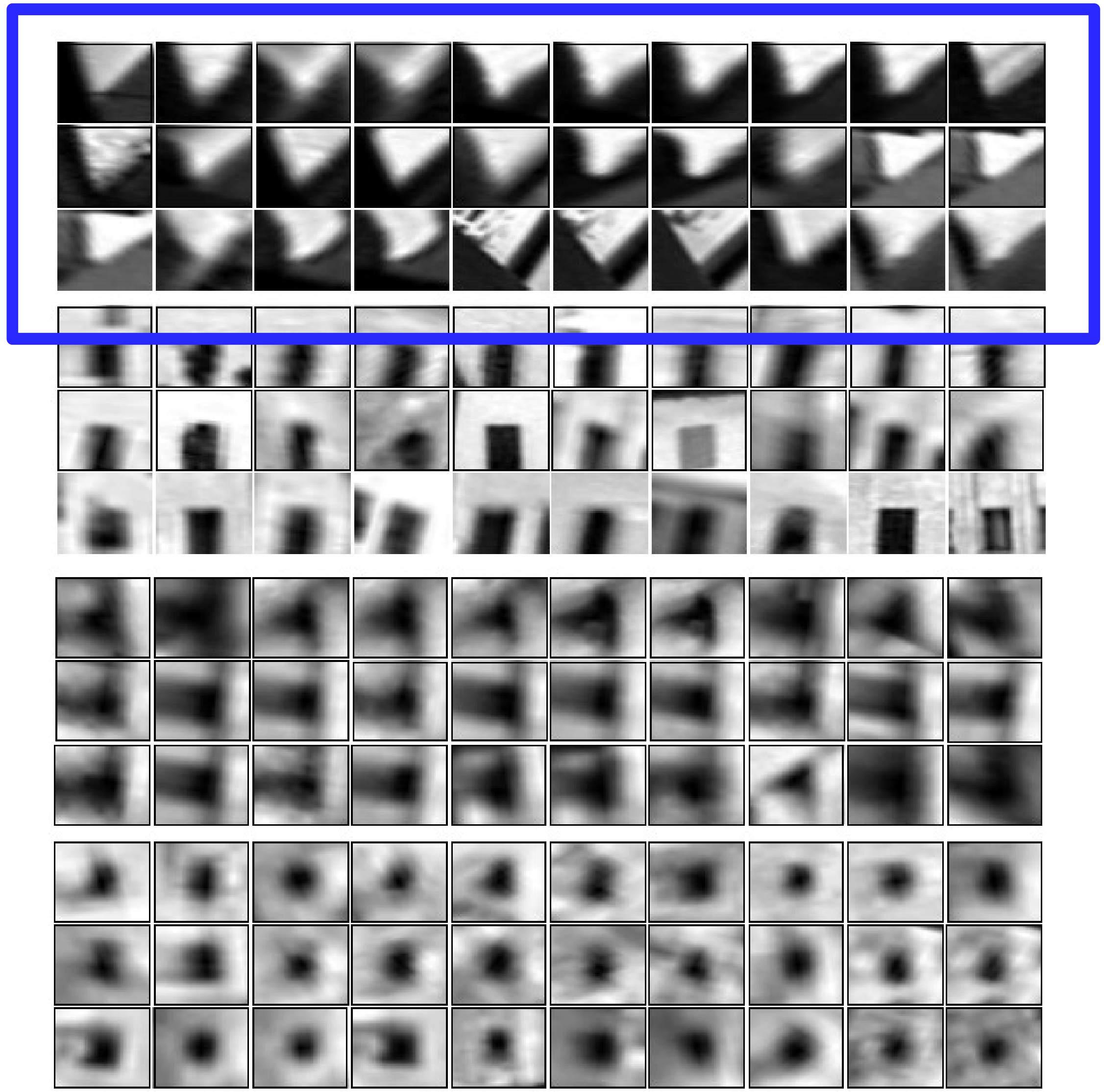
- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

Kristen Grauman

Visual words

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





Kristen Grauman

Figure from Sivic & Zisserman, ICCV 2003

Visual vocabulary formation

Issues:

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

Kristen Grauman

Inverted file index

Image #1

Image #2

Image #3

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

…

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

Kristen Grauman

Inverted file index

New query image

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

…

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

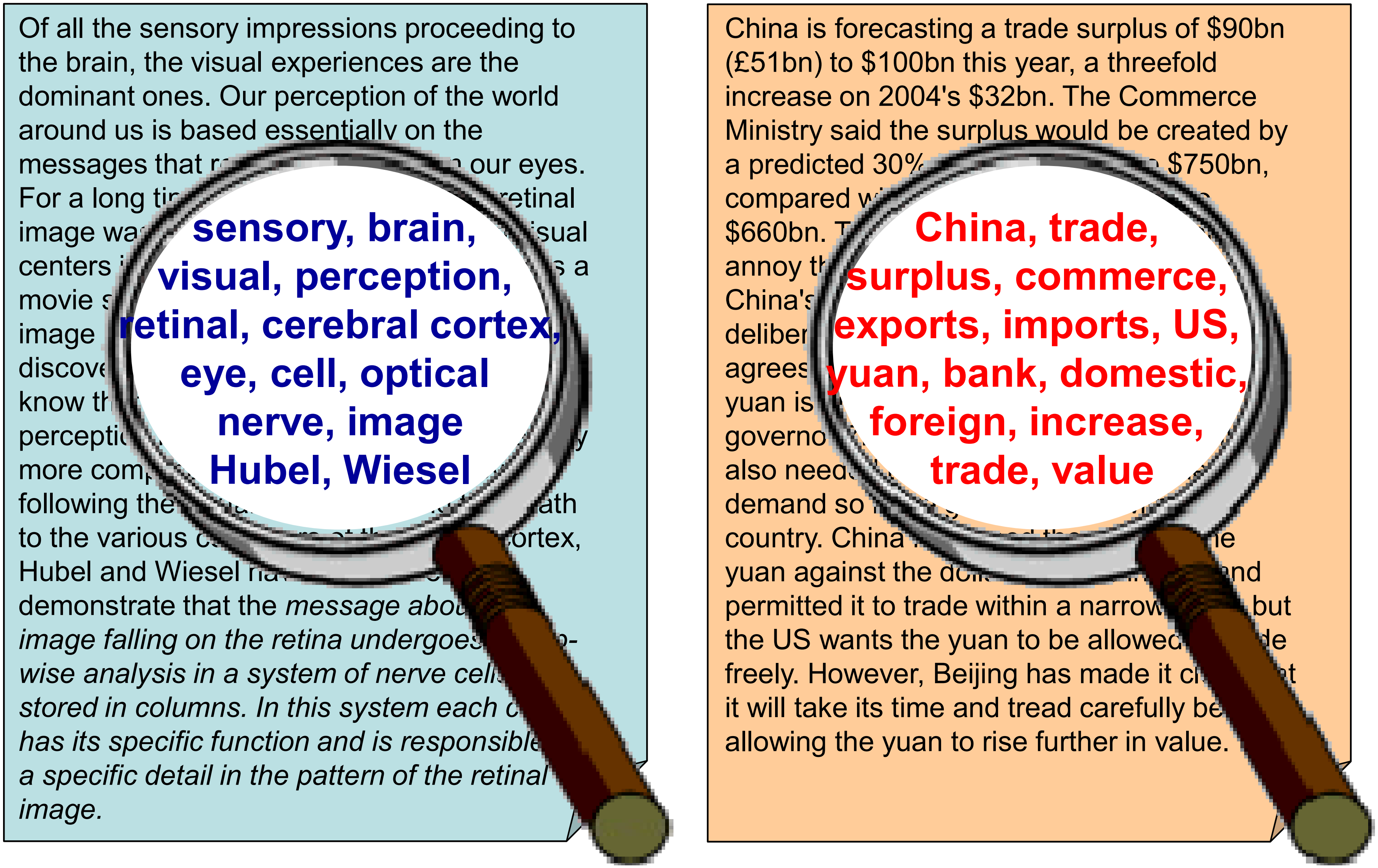
Kristen Grauman

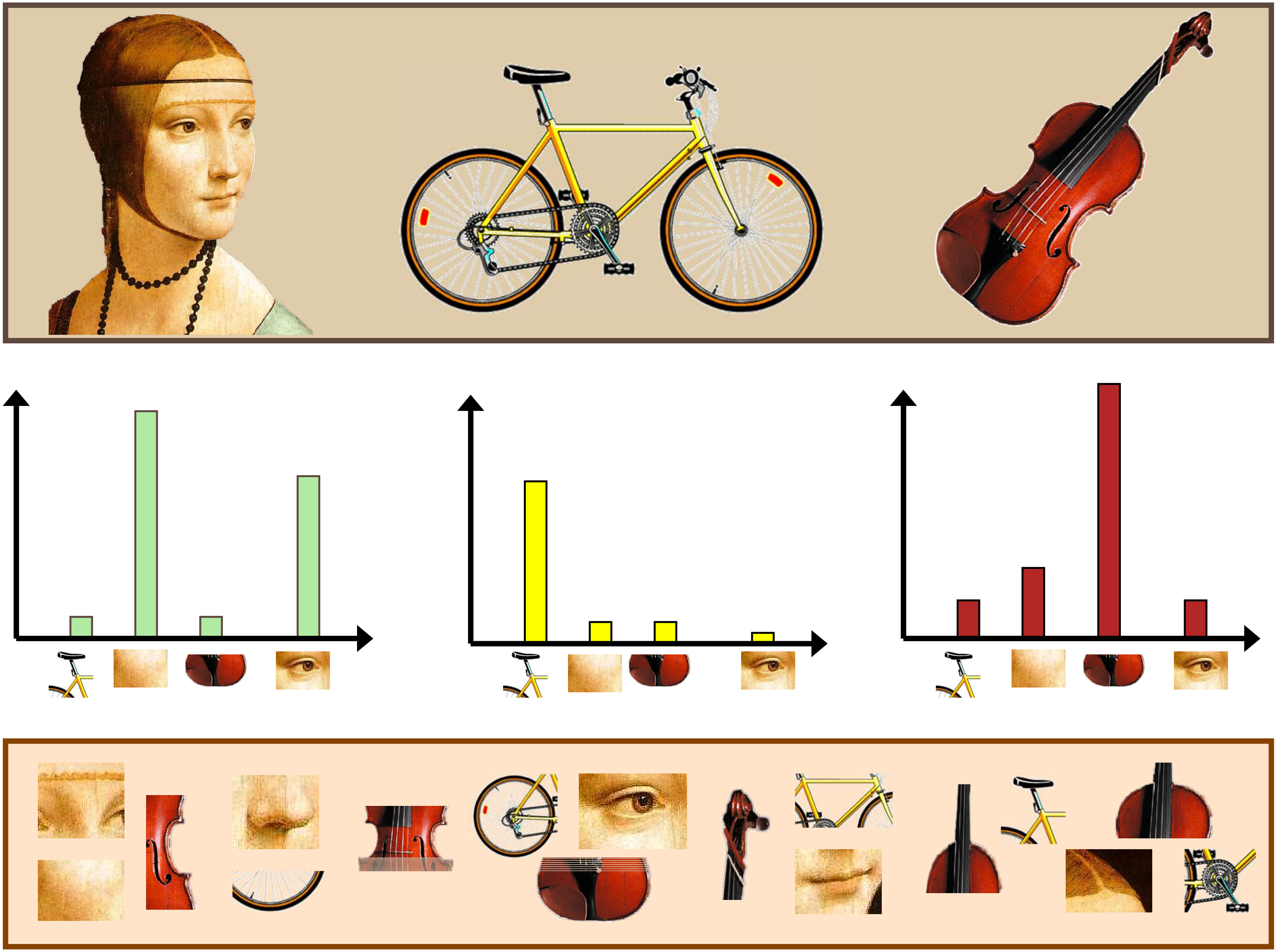
Instance recognition: remaining issues

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

Kristen Grauman

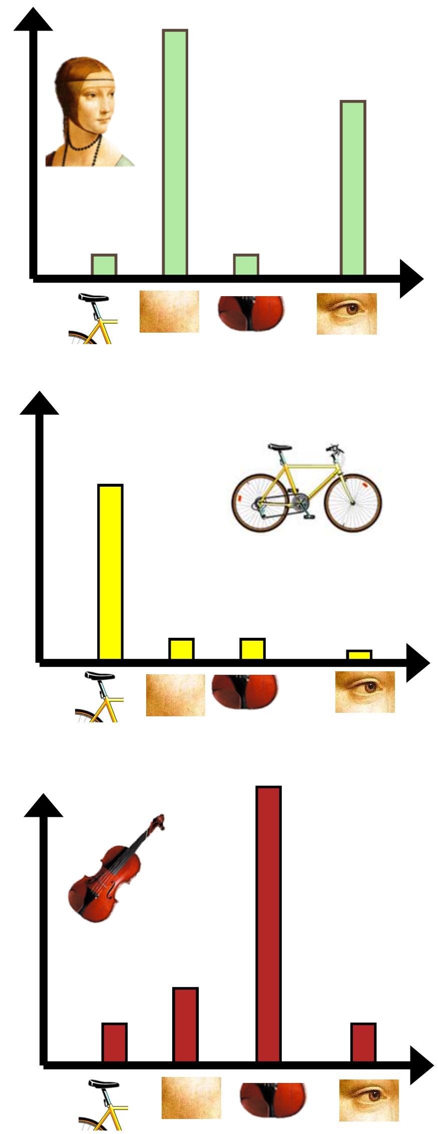
Analogy to documents





Bags of visual words

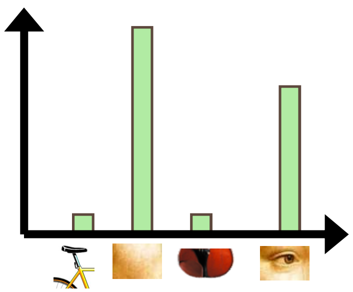
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



Comparing bags of words

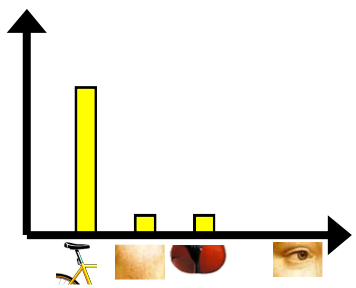
- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.

[1 8 1 4]



$\vec{d}_j$

[5 1 1 0]



$\vec{q}$

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

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

for vocabulary of *V* words

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- Describe frame by frequency of each word within it, 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

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

Total number of documents in database

Number of documents word i occurs in, in whole database

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

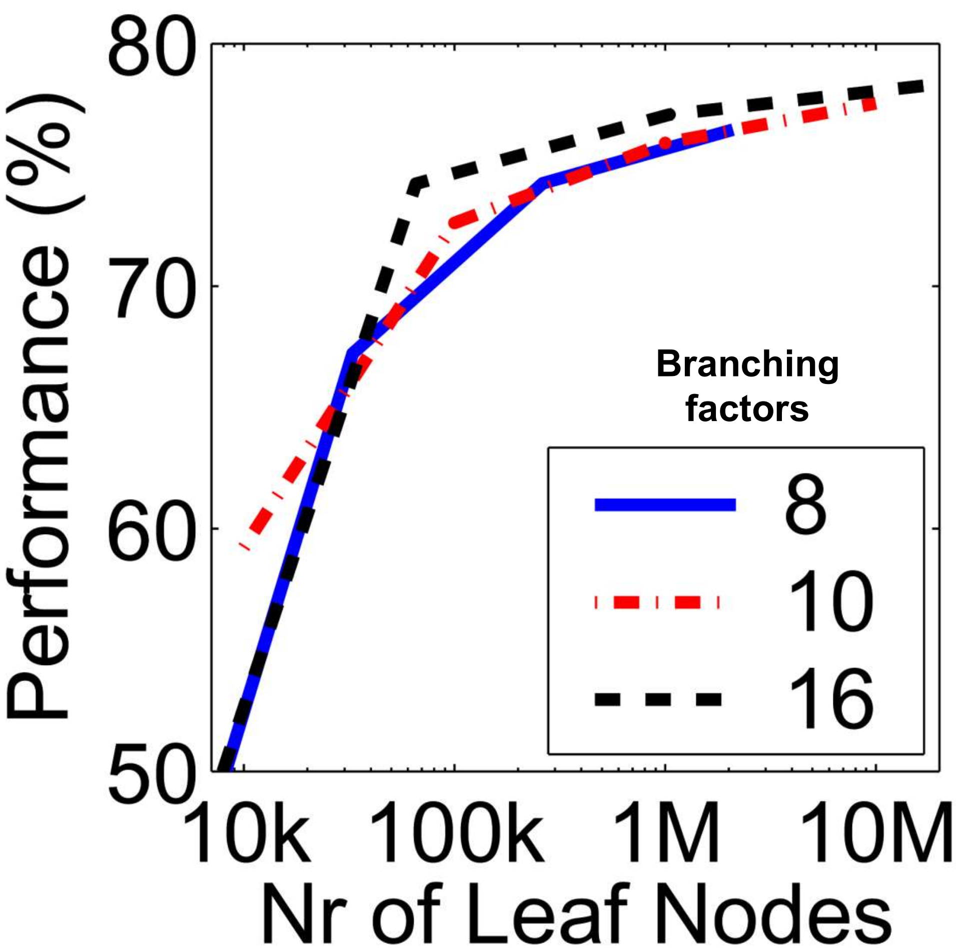
Kristen Grauman

Instance recognition: remaining issues

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

Kristen Grauman

Vocabulary size



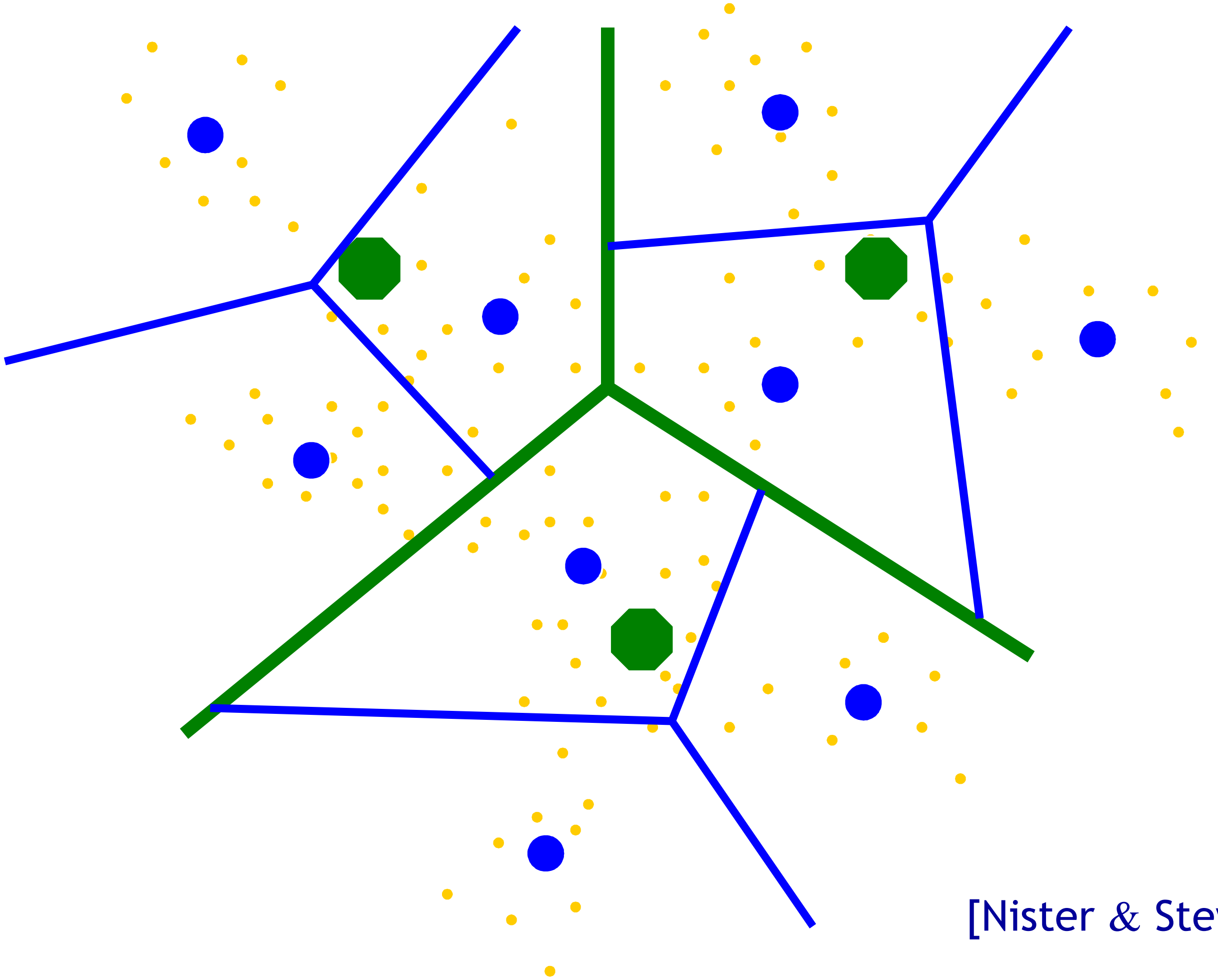
Results for recognition task with 6347 images

Influence on performance, sparsity?

Nister & Stewenius, CVPR 2006

Vocabulary Trees: hierarchical clustering for large vocabularies

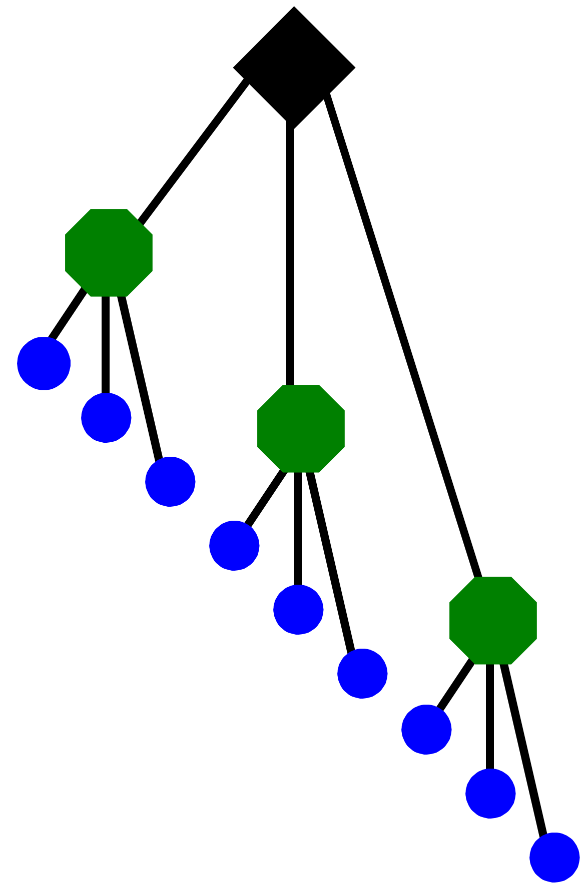
- Tree construction:



[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

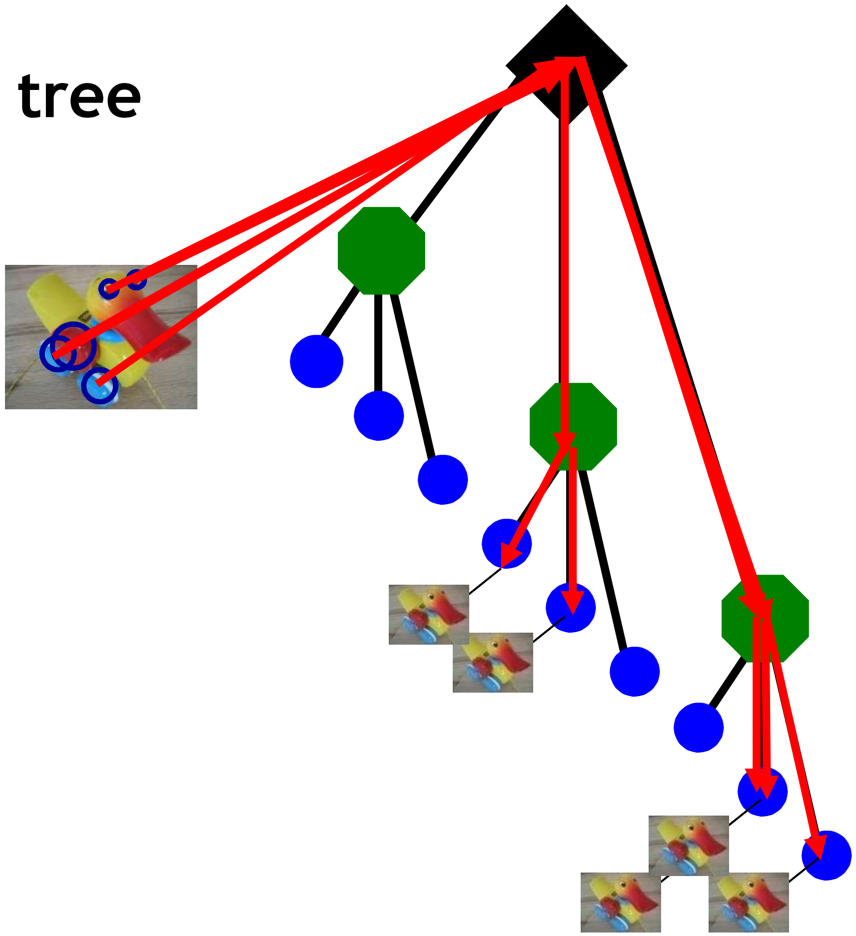


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**

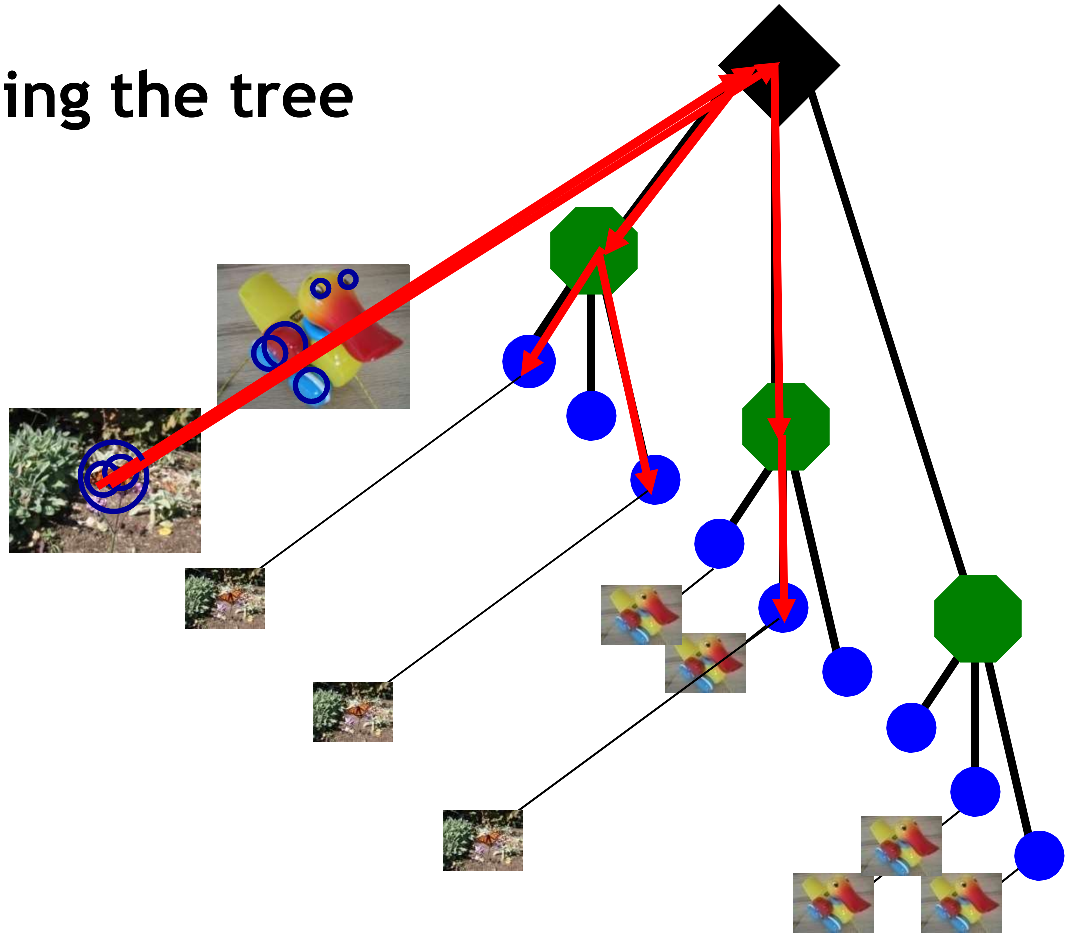


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**

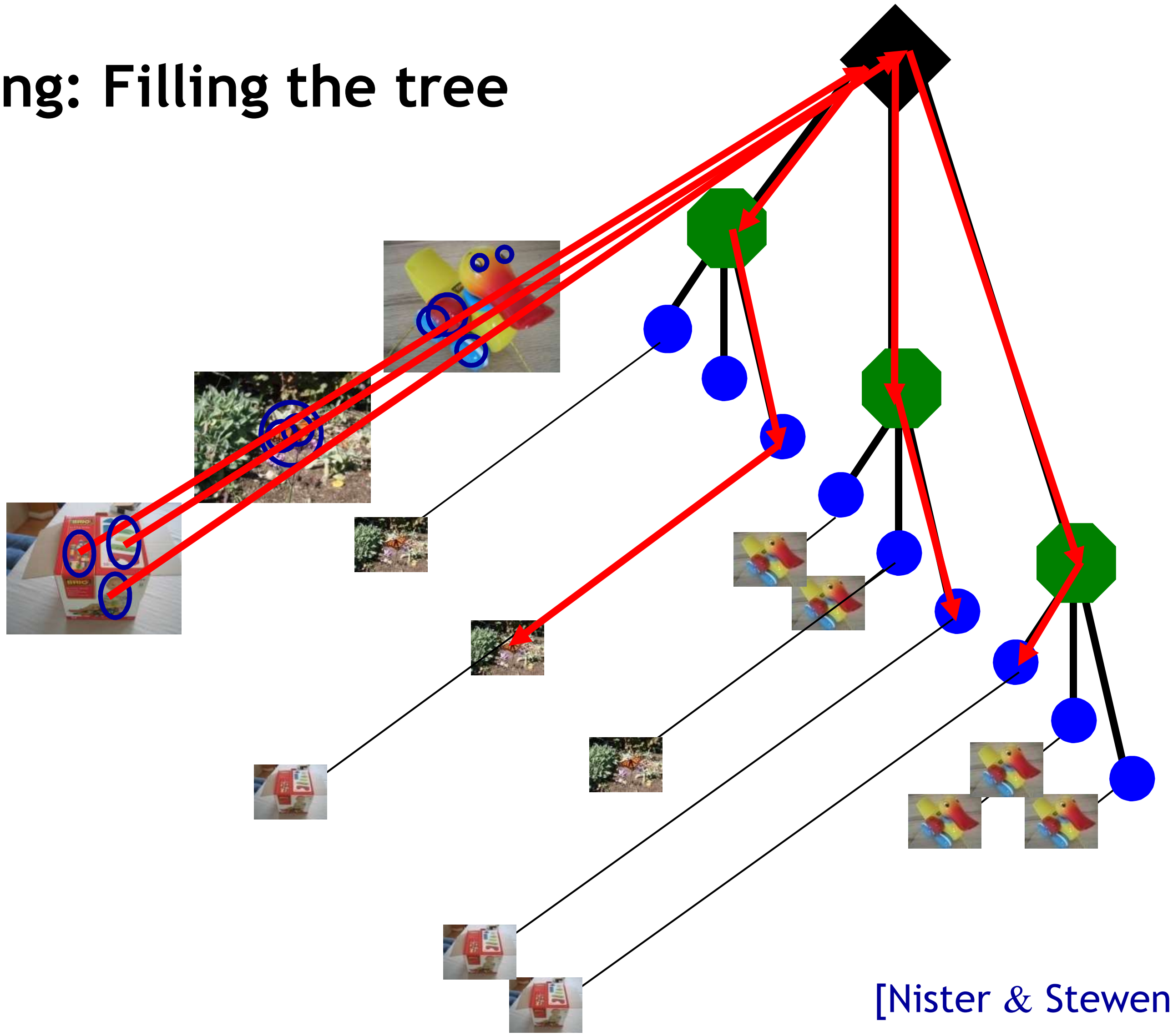


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**

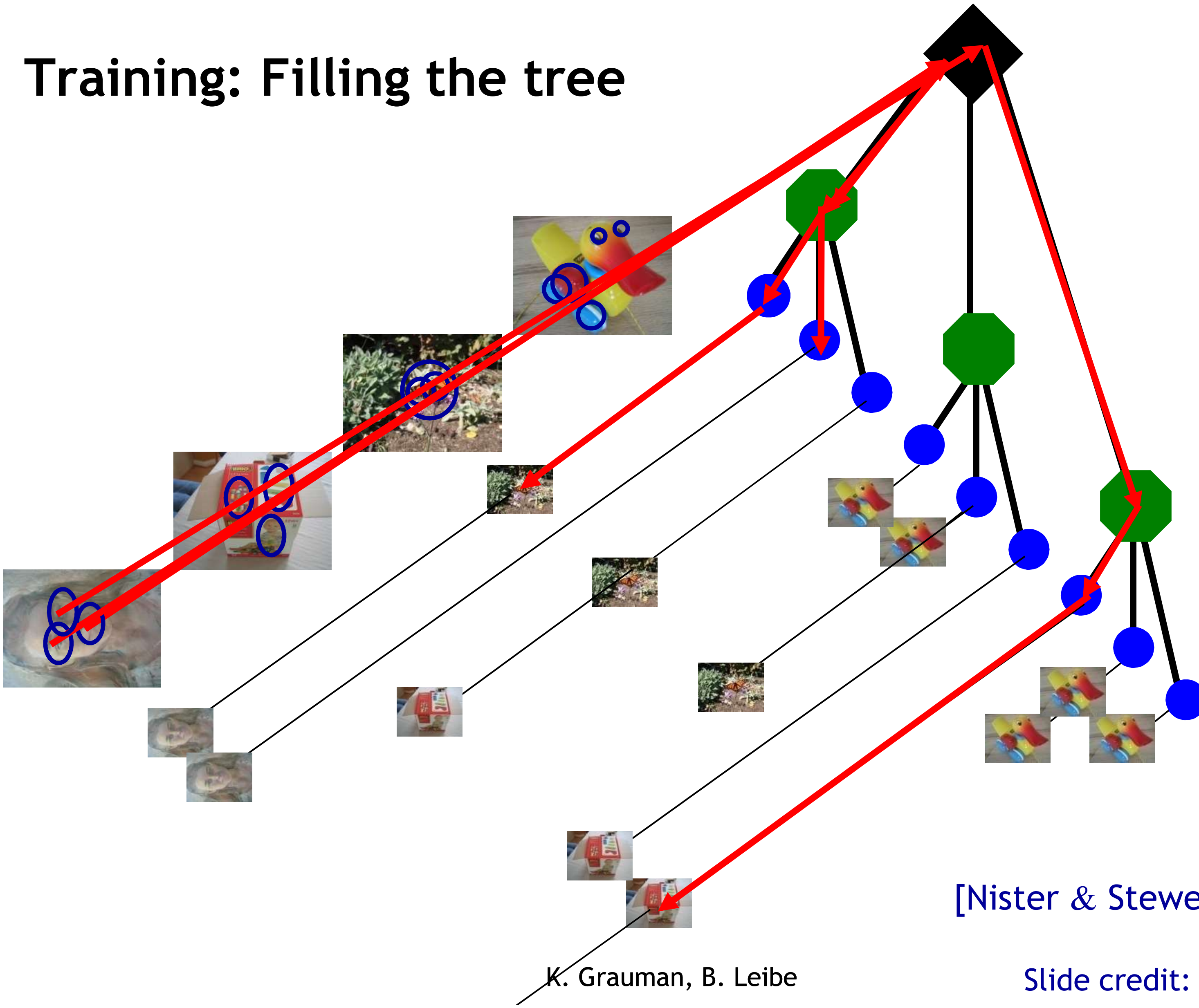


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**

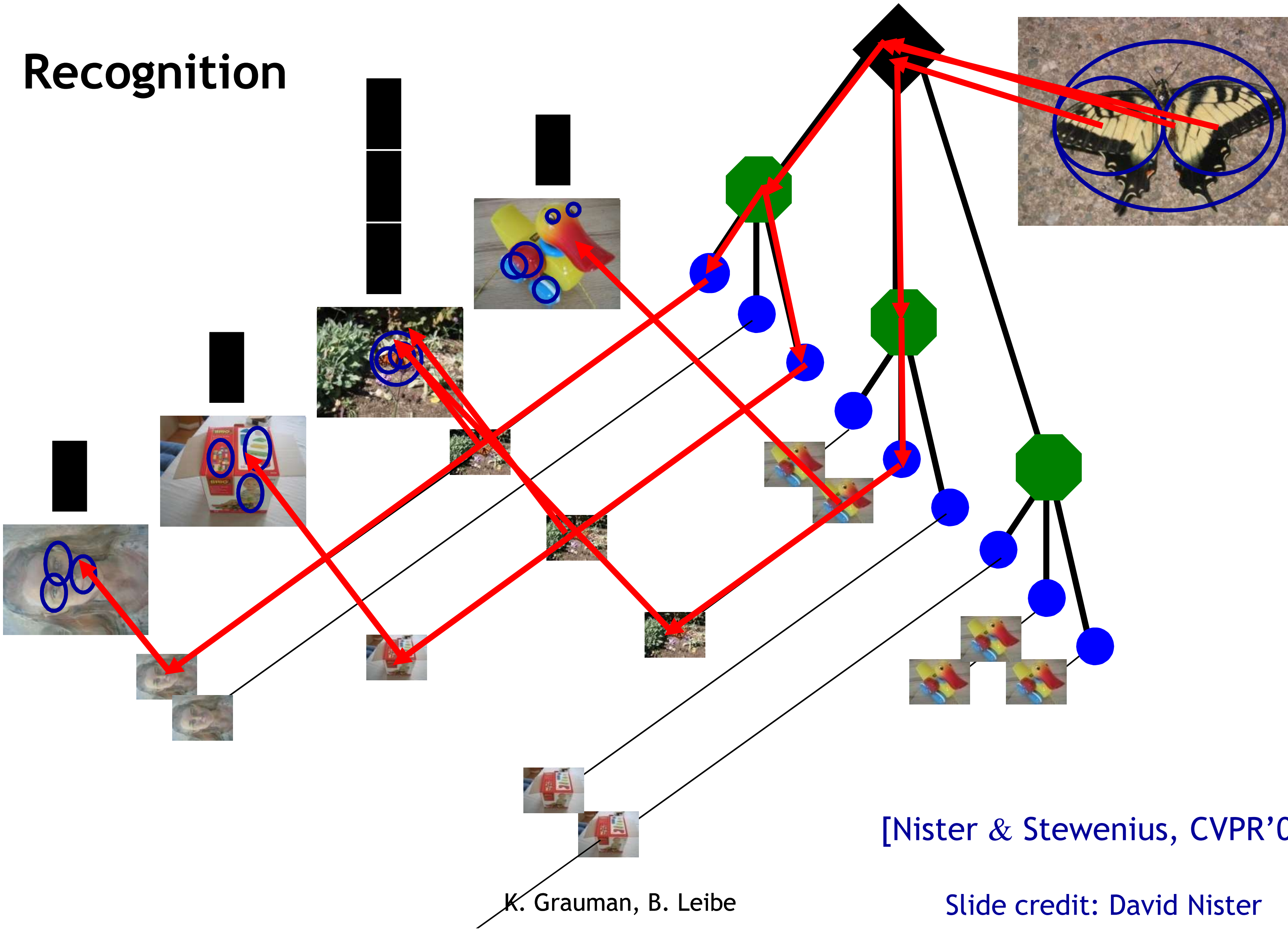


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Recognition**



[Nister & Stewenius, CVPR'06]

Vocabulary trees: complexity

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

Word assignment cost vs. flat vocabulary

Visual words/bags of words

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides vector representation for sets
+ very good results in practice

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

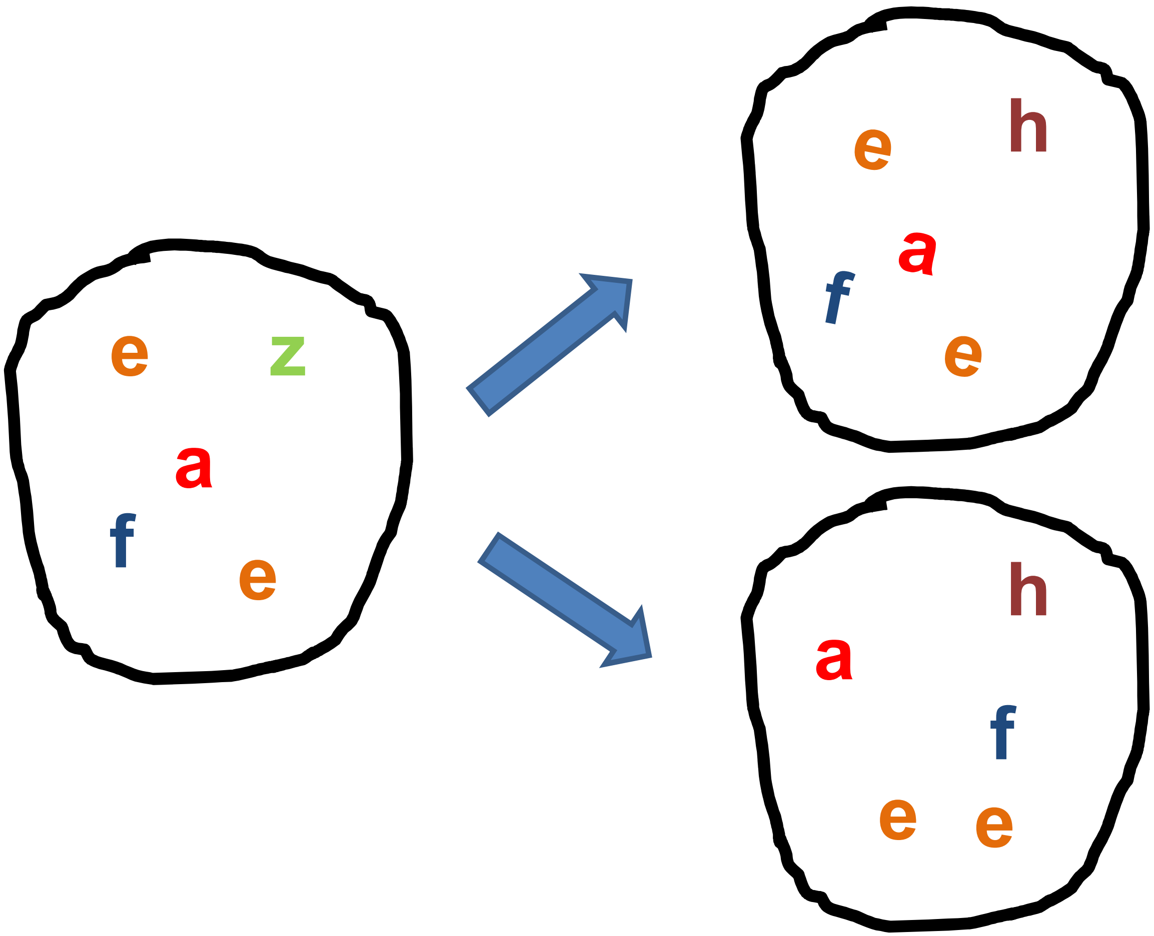
Kristen Grauman

Instance recognition: remaining issues

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

Kristen Grauman

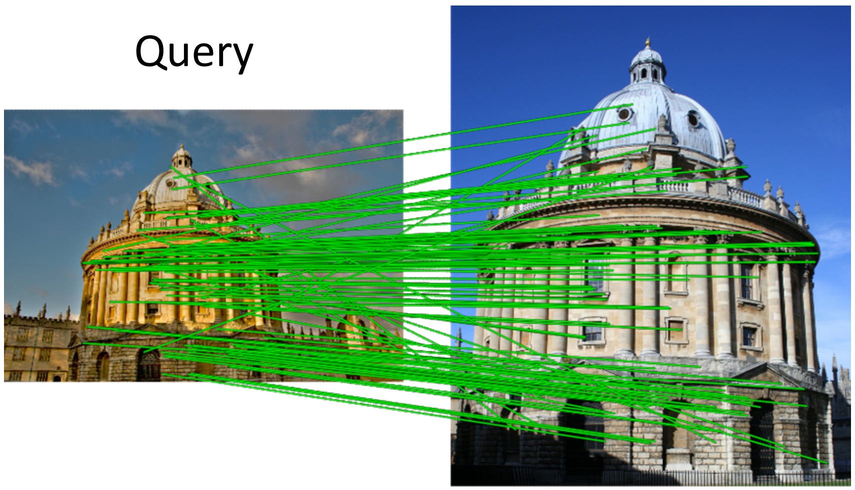
Which matches better?



Derek Hoiem

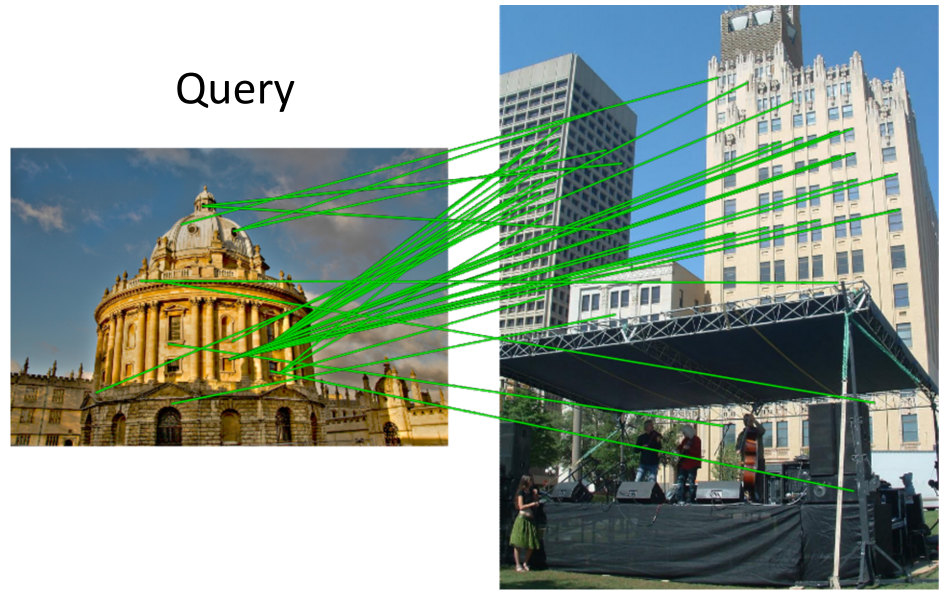
Spatial Verification

Query



DB image with high BoW similarity

Query



DB image with high BoW similarity

Both image pairs have many visual words in common.

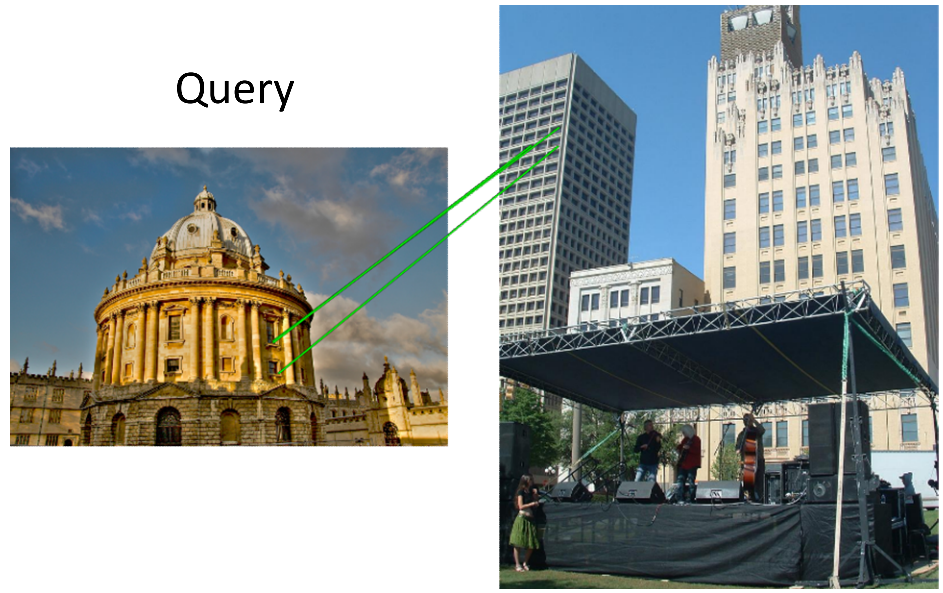
Slide credit: Ondrej Chum

Spatial Verification

Query

DB image with high BoW similarity

Query



DB image with high BoW similarity

Only some of the matches are mutually consistent

Slide credit: Ondrej Chum

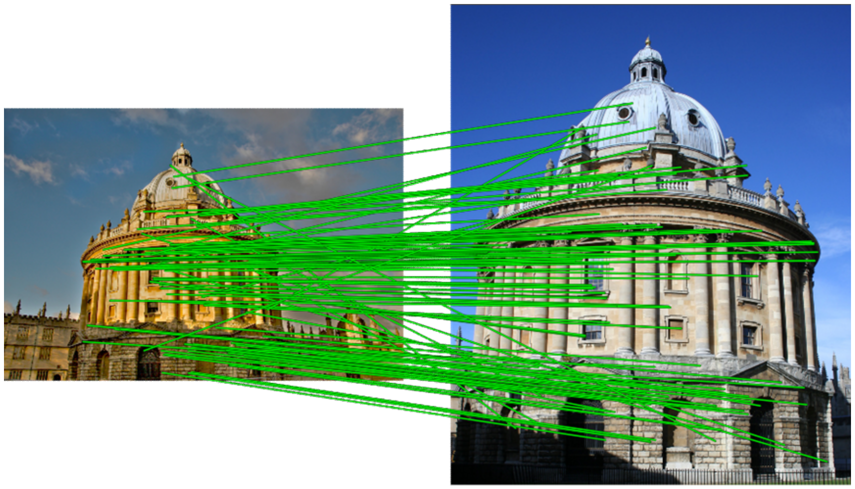
Spatial Verification: two basic strategies

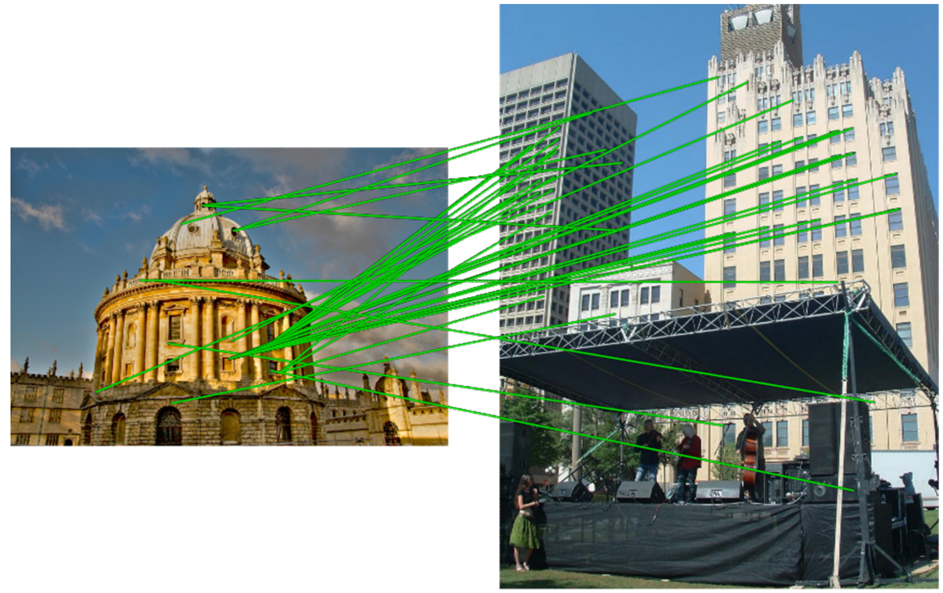
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible affine transformations
    - e.g., “success” if find an affine transformation with > N inlier correspondences

- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

Kristen Grauman

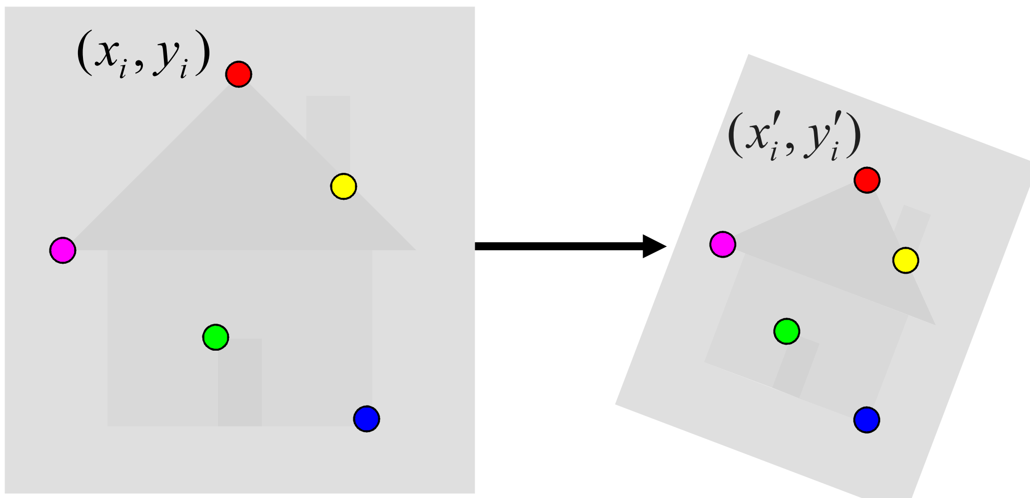
RANSAC verification





For matching specific scenes/objects, common to use an **affine transformation** for spatial verification

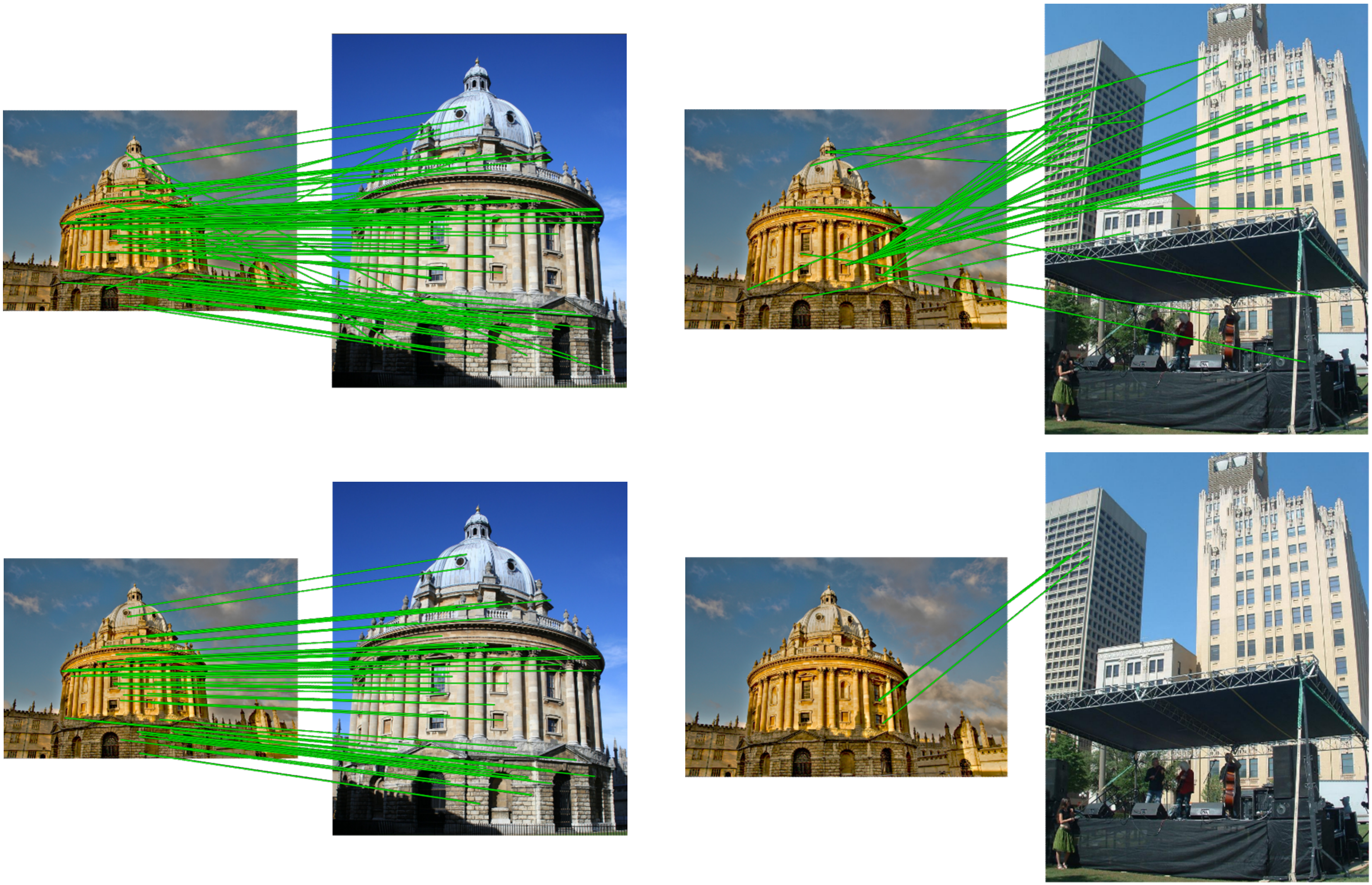
Fitting an affine transformation



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

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

RANSAC verification



Spatial Verification: two basic strategies

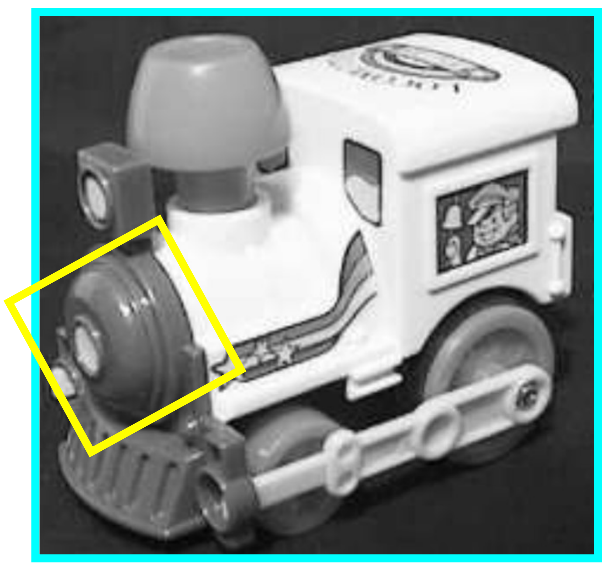
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible affine transformations
    - e.g., “success” if find an affine transformation with > N inlier correspondences

- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

Kristen Grauman

Voting: Generalized Hough Transform

- If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).



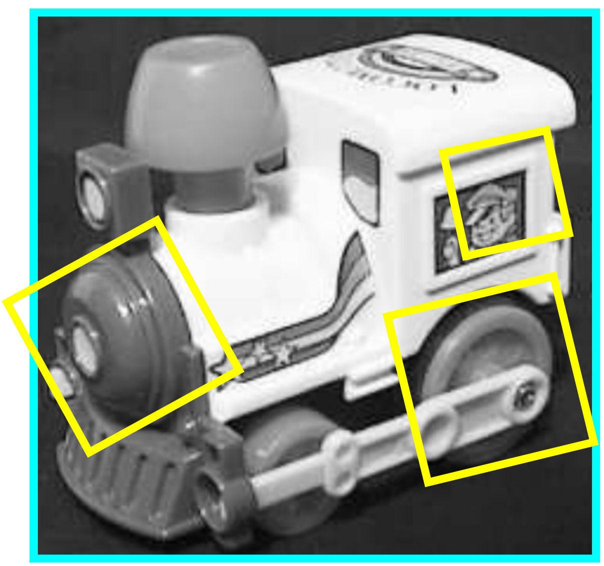
Model

Novel image

Adapted from Lana Lazebnik

Voting: Generalized Hough Transform

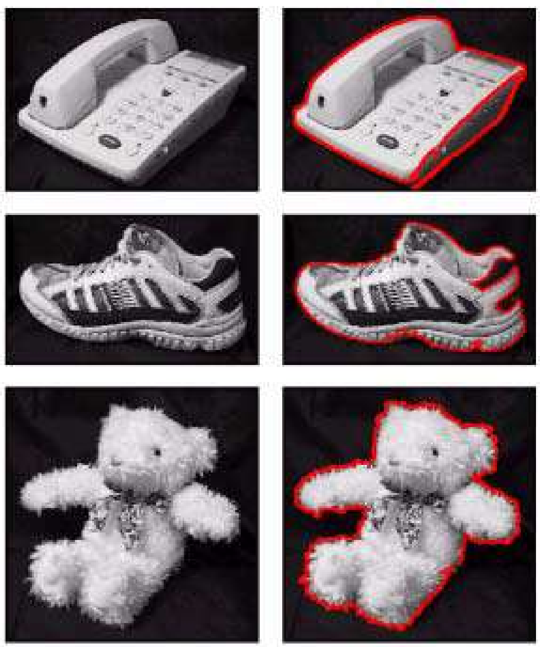
- A hypothesis generated by a single match may be unreliable,
- So let each match **vote** for a hypothesis in Hough space



Model

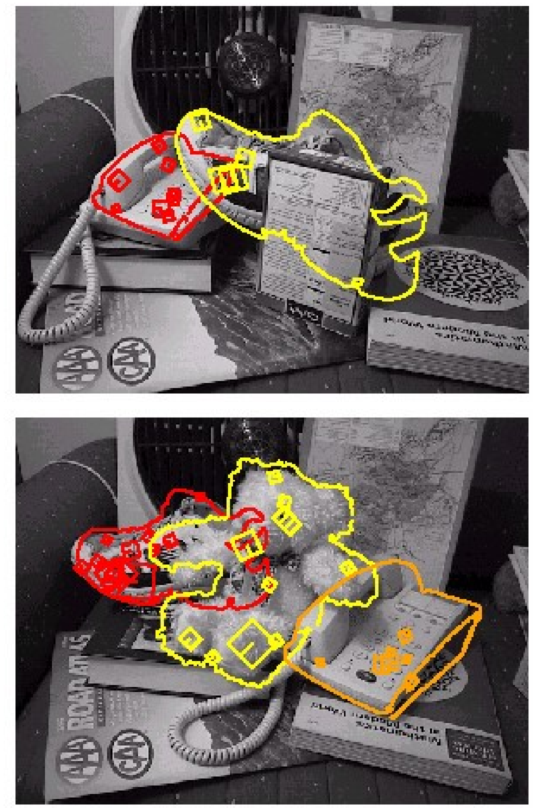
Novel image

Example result



Background subtract
for model boundaries

Objects recognized,



Recognition in
spite of occlusion

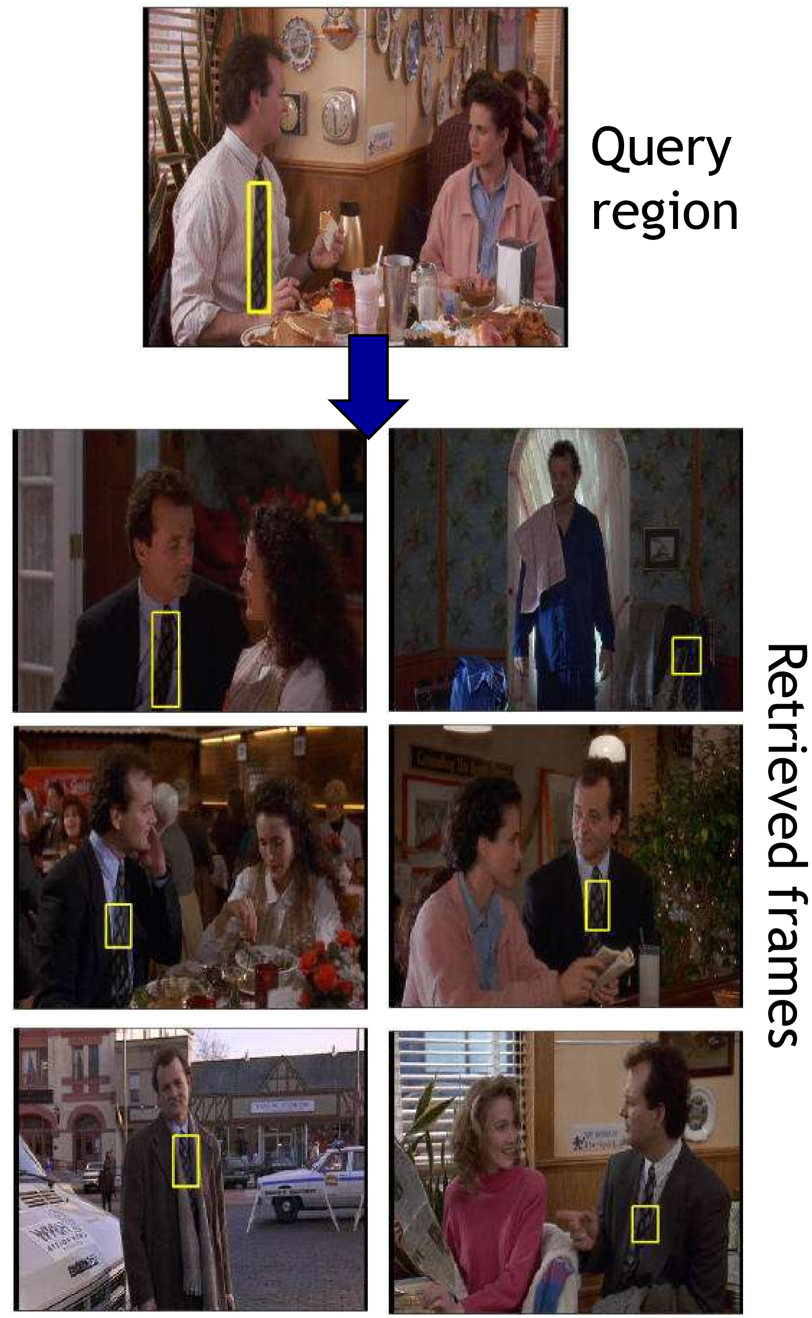
[Lowe]

Video Google System

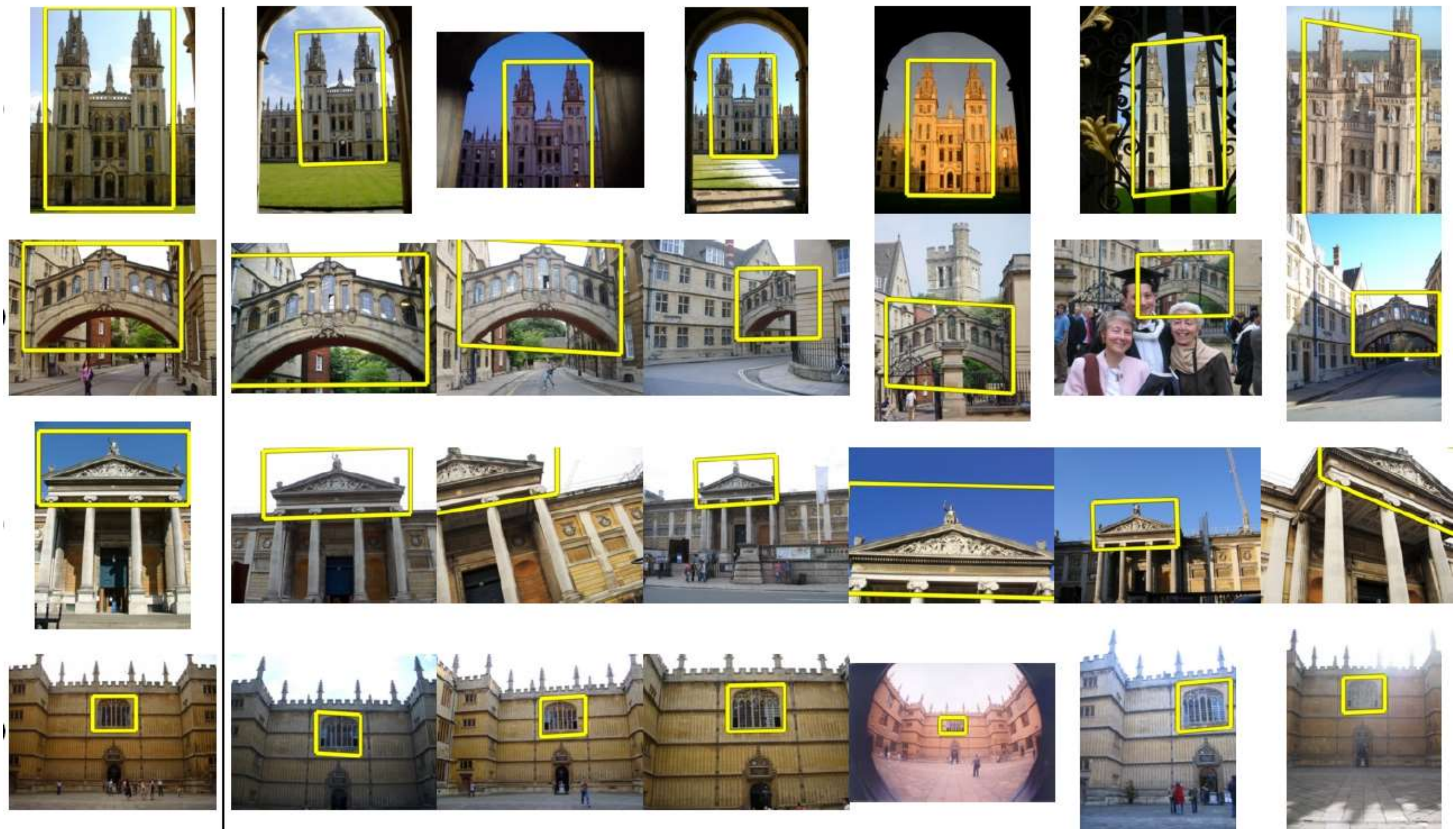
1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

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



Object retrieval with large vocabularies and fast spatial matching, Philbin et al., CVPR 2007



Query Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR'07]

World-scale mining of objects and events from community photo collections, Quack et al., CIVR 2008

Auto-annotate by connecting to content on Wikipedia!

Example Applications

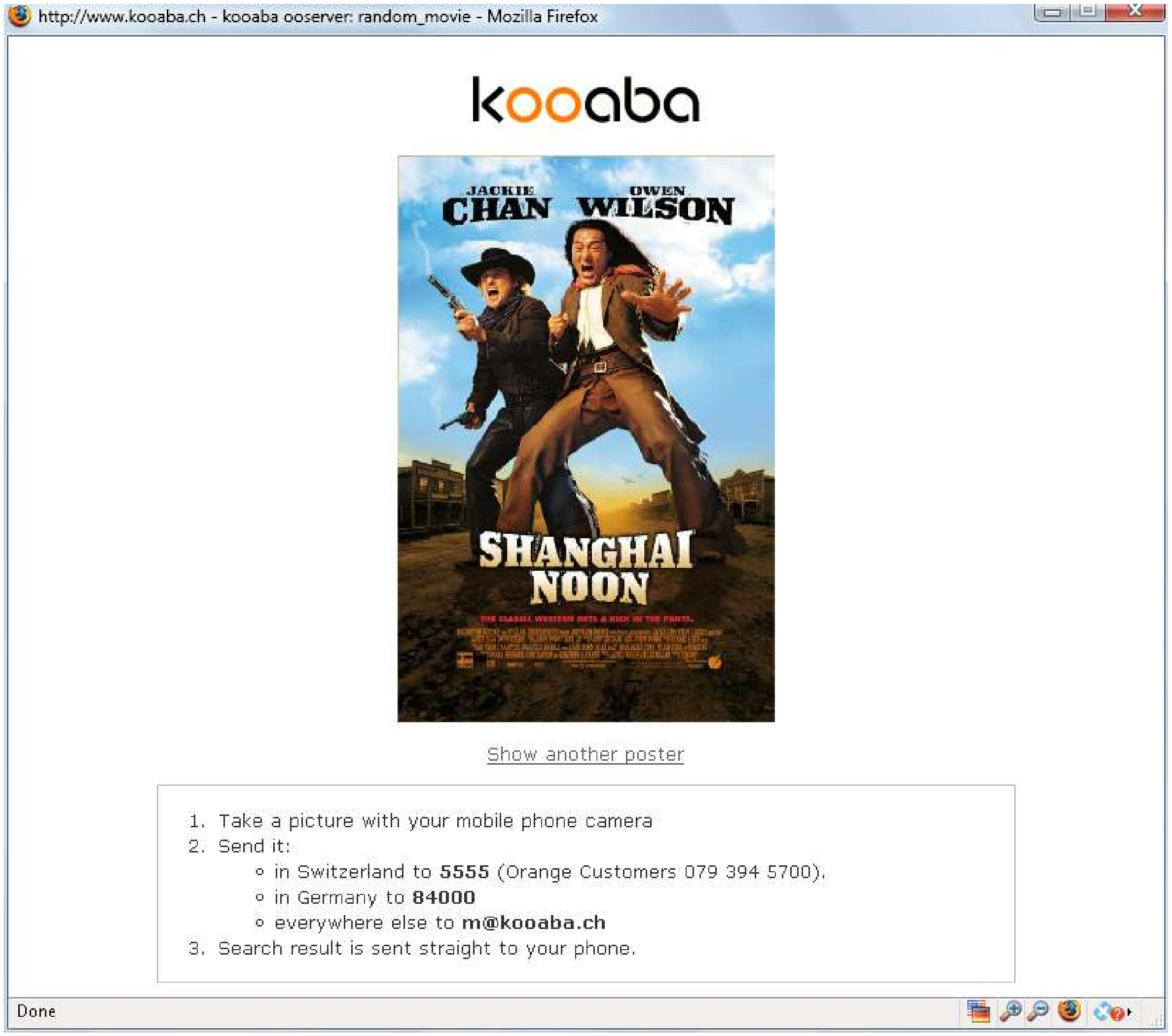
Mobile tourist guide
- Self-localization
- Object/building recognition
- Photo/video augmentation

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

Web Demo: Movie Poster Recognition

50'000 movie posters indexed

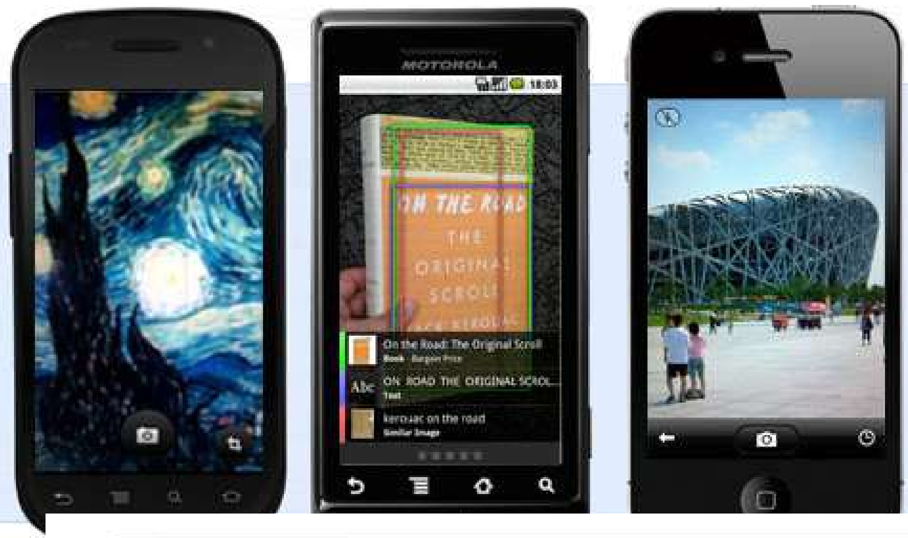
Query-by-image from mobile phone available in Switzer-land



http://www.kooaba.com/en/products_engine.html#

Google Goggles

Use pictures to search the web. ▷ Watch a video



Get Google Goggles

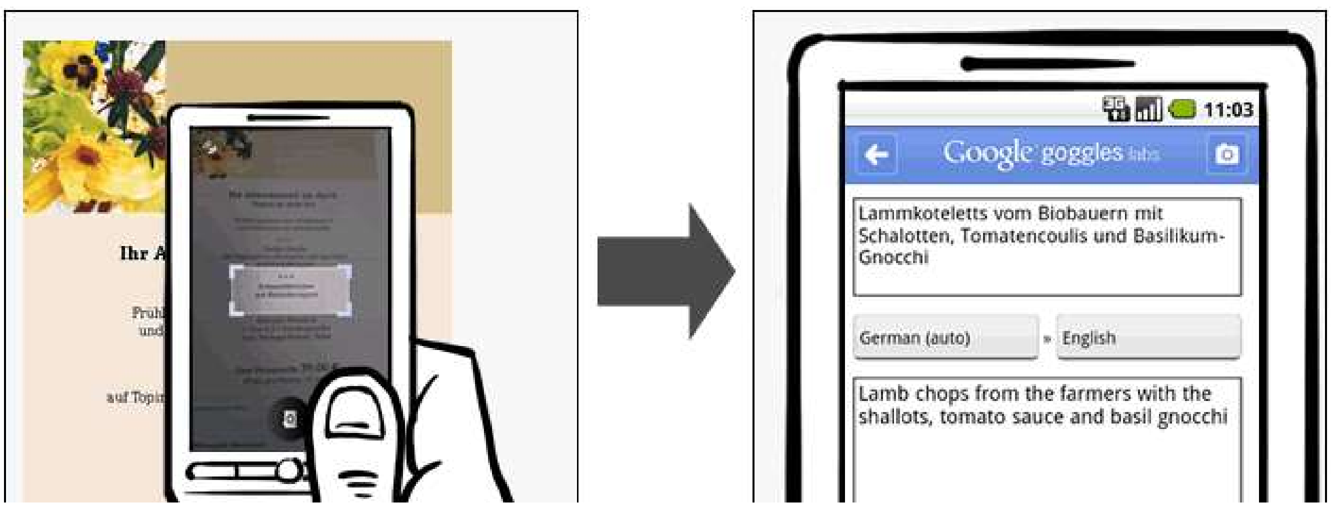
Android (1.6+ required)
Download from Android Market.

Send Goggles to Android phone

New! **iPhone (iOS 4.0 required)**
Download [from the App Store](#).

Send Goggles to iPhone

New!
Text | Landmarks | Books | Contact Info | Artwork | Wine | Logos



Recognition via feature matching+spatial verification

Pros:

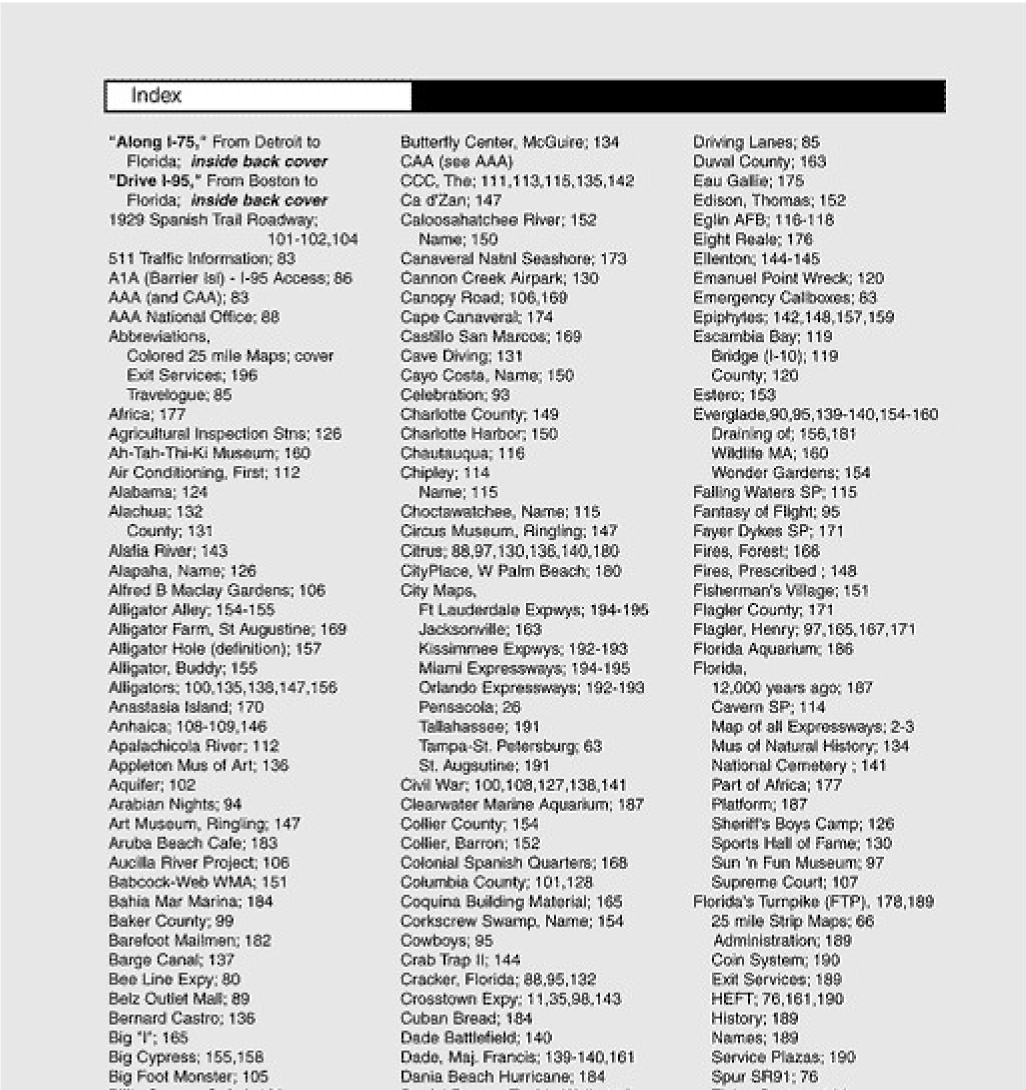
- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

Cons:

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

Kristen Grauman

What else can we borrow from text retrieval?


Index

Query expansion

Query: ***golf green***

Results:

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

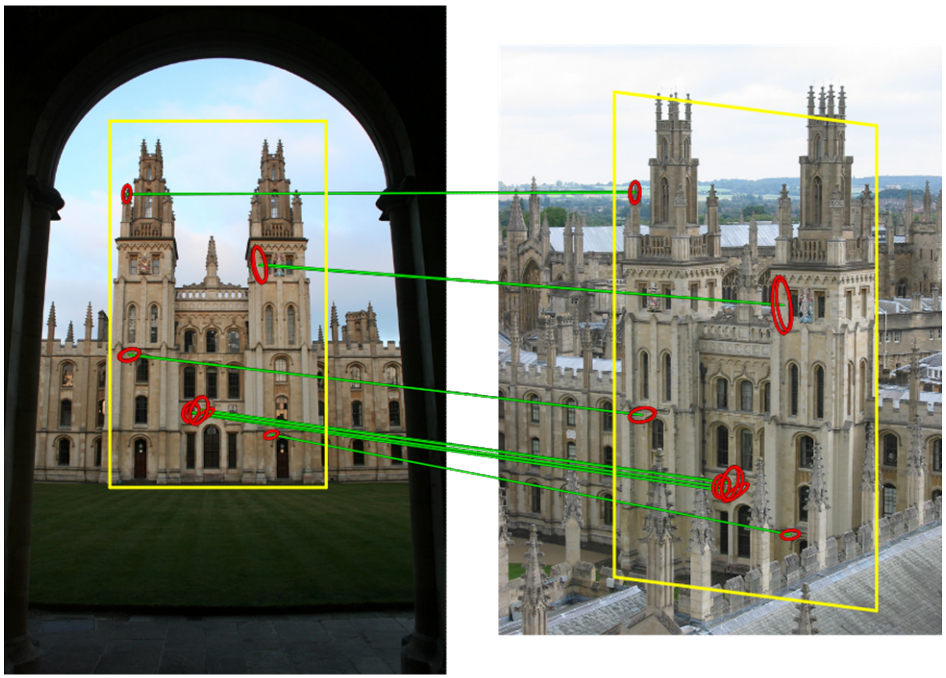
Slide credit: Ondrej Chum

Query Expansion

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

Slide credit: Ondrej Chum

Query Expansion Step by Step

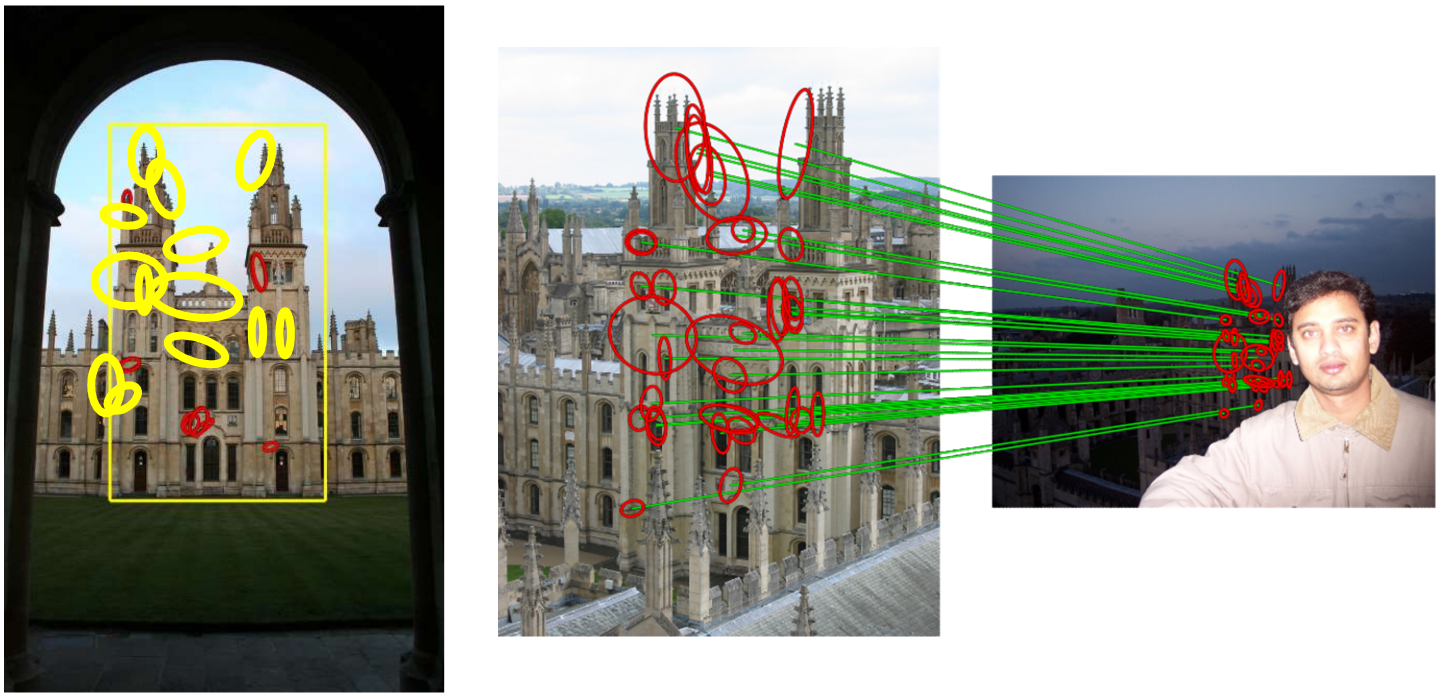


Query Image   Retrieved image

Originally not retrieved

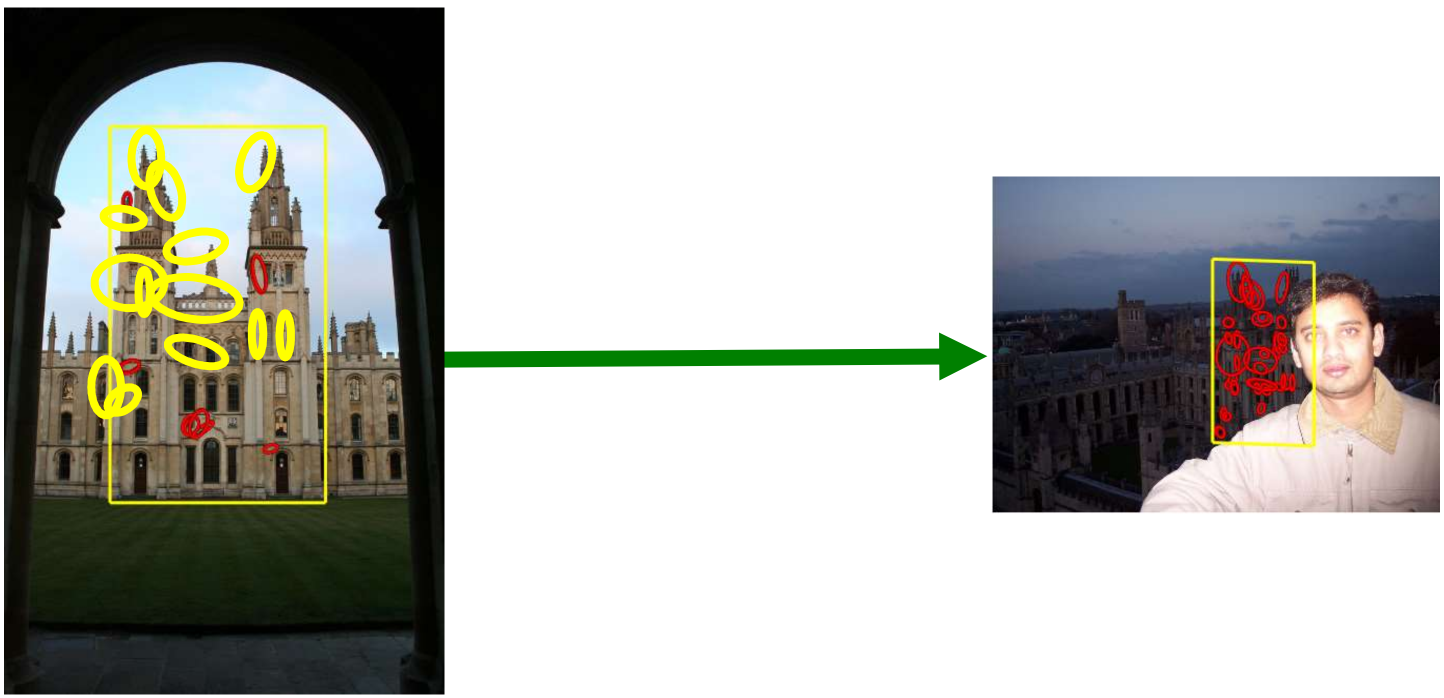
Slide credit: Ondrej Chum

Query Expansion Step by Step



Slide credit: Ondrej Chum

Query Expansion Step by Step

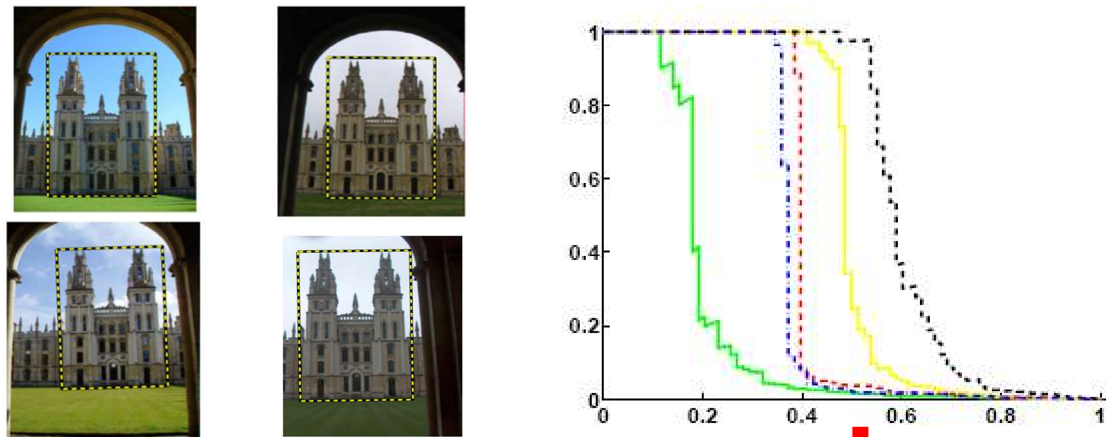


Slide credit: Ondrej Chum

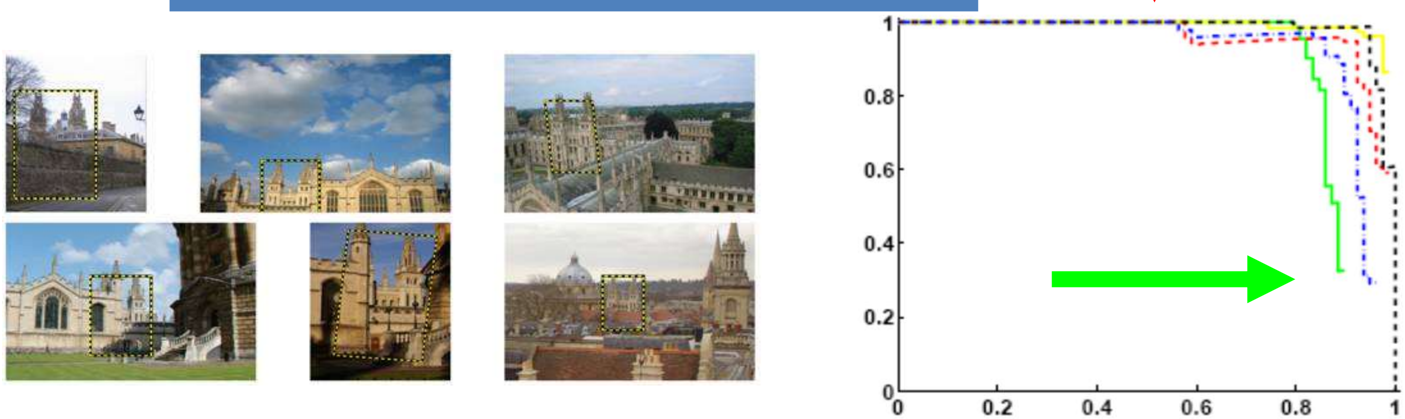
Query Expansion Results

Query image

Original results (good)



Expanded results (better)



Slide credit: Ondrej Chum

Summary

- **Matching local invariant features**
  - Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
  - Summarize image by distribution of words
  - Index individual words
- **Inverted index**: pre-compute index to enable faster search at query time
- **Recognition of instances via alignment:** matching local features followed by spatial verification
  - Robust fitting : RANSAC, GHT

Kristen Grauman