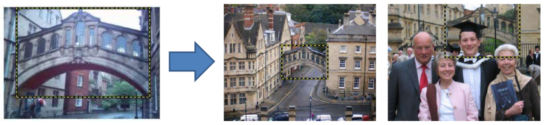
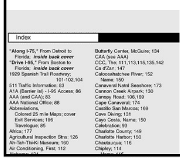
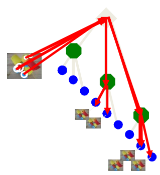


Instance recognition

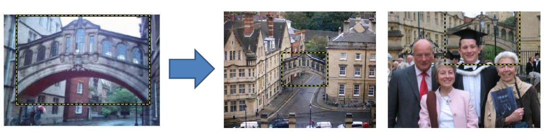
Tues April 3
Kristen Grauman
UT Austin





Instance recognition

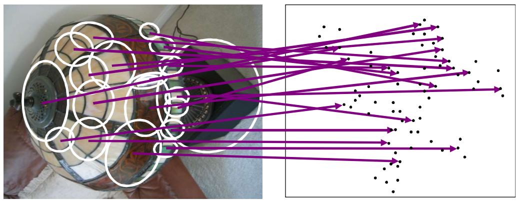
- Indexing local features efficiently (last time)
- Spatial verification models



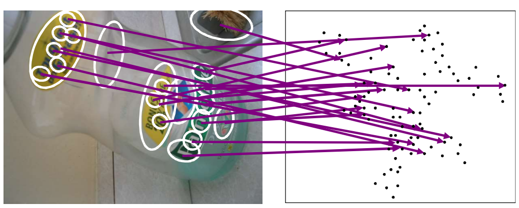
Picking up from last time

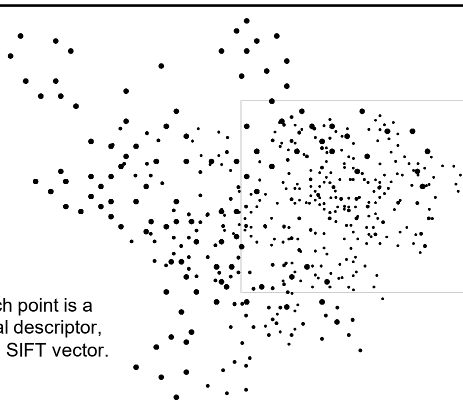
- Instance recognition wrap up:
  - Impact of vocabulary tree
  - Spatial verification
  - Sky mapping example
  - Query expansion

Visual words: main idea

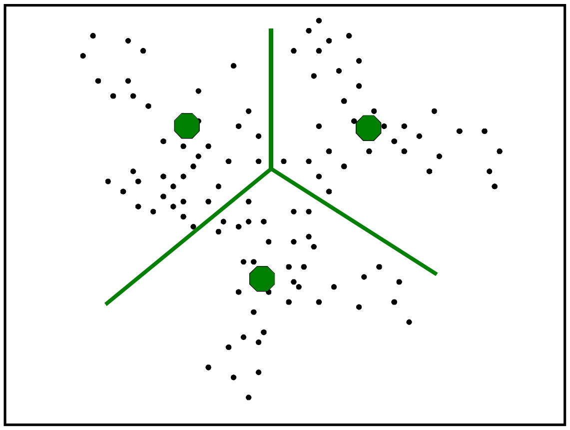


Visual words: main idea



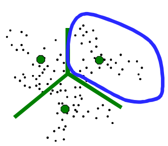


Each point is a local descriptor, e.g. SIFT vector.



Visual words

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



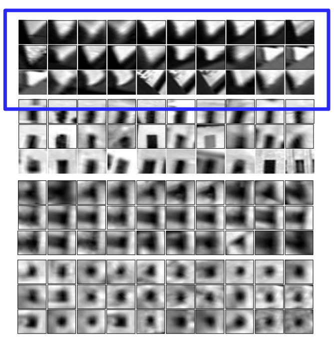
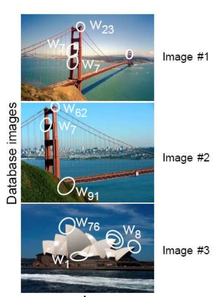


Figure from Sivic & Zisserman, ICCV 2003

Inverted file index



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

Slide credit: Kristen Grauman

Inverted file index

New query image

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

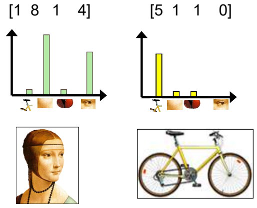
- New query image is mapped to indices of database images that share a word.

Slide credit: Kristen Grauman

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] [5 1 1 0]



$\vec{d}_j$ $\vec{q}$

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

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

for vocabulary of *V* words

Slide credit: Kristen Grauman

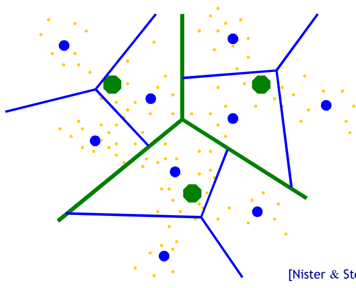
Instance recognition: remaining issues

- 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?

Slide credit: Kristen Grauman

Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:



[Nister & Stewenius, CVPR'06]

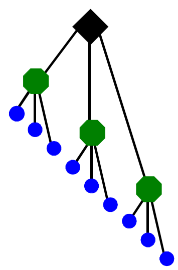
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

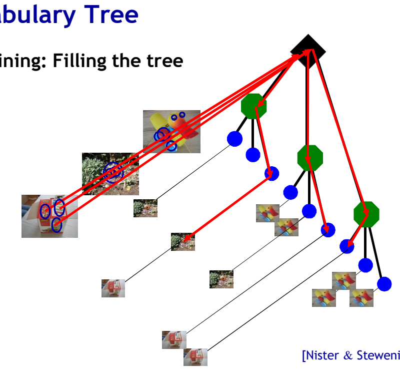
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree

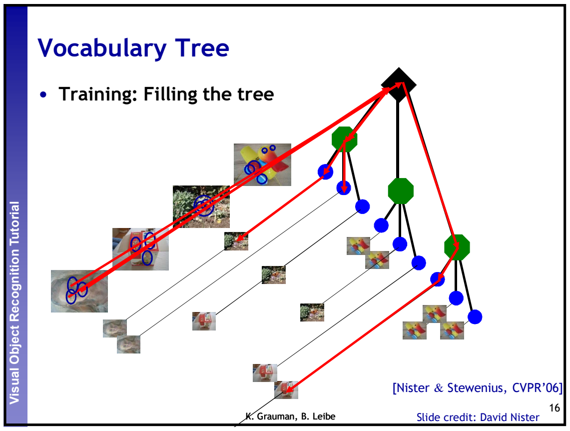


[Nister & Stewenius, CVPR'06]

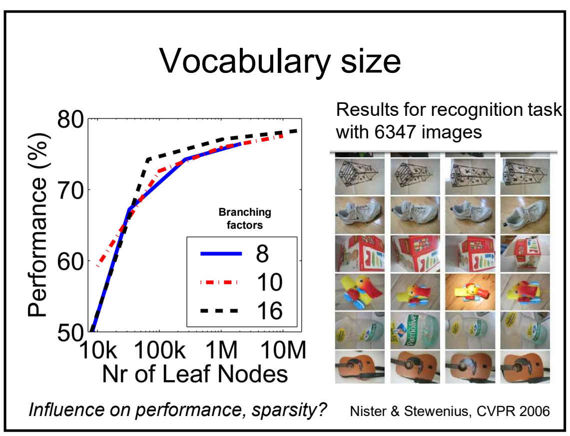
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister



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



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

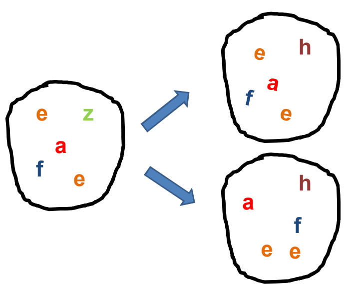
Slide credit: Kristen Grauman

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?

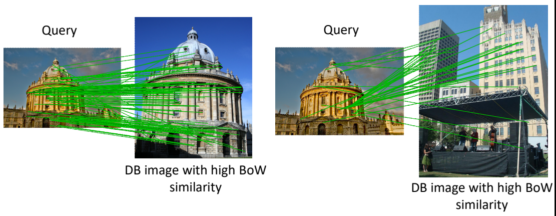
Slide credit: Kristen Grauman

Which matches better?



Derek Hoiem

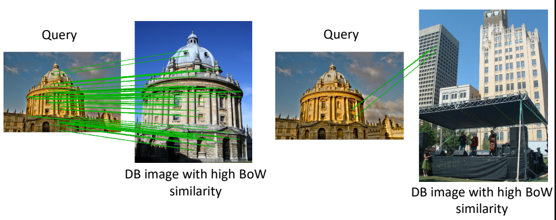
Spatial Verification



Both image pairs have many visual words in common.

Slide credit: Ondrej Chum

Spatial Verification



Only some of the matches are mutually consistent

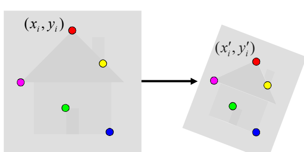
Slide credit: Ondrej Chum

Spatial Verification: two basic strategies

- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible 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

RANSAC verification

Recall: Fitting an affine transformation

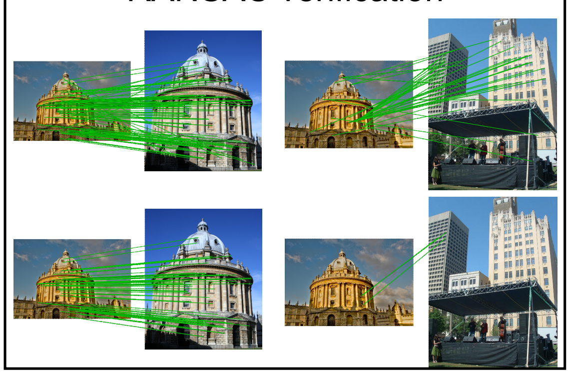


Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

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

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

RANSAC verification

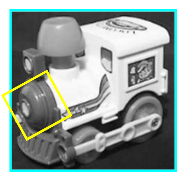


Spatial Verification: two basic strategies

- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible 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

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).

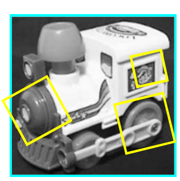

Model

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

Gen Hough Transform details (Lowe's system)

- **Training phase:** For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)
- **Test phase:** Let each match btwn a test SIFT feature and a model feature vote in a 4D Hough space
  - Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
  - Vote for two closest bins in each dimension
- Find all bins with at least three votes and perform geometric verification
  - Estimate least squares *affine* transformation
  - Search for additional features that agree with the alignment

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Lana Lazebnik

Recall: difficulties of voting

- Noise/clutter can lead to as many votes as true target
- Bin size for the accumulator array must be chosen carefully
- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.

Example result

Background subtract for model boundaries

Objects recognized,

Recognition in spite of occlusion

[Lowe]

Example applications

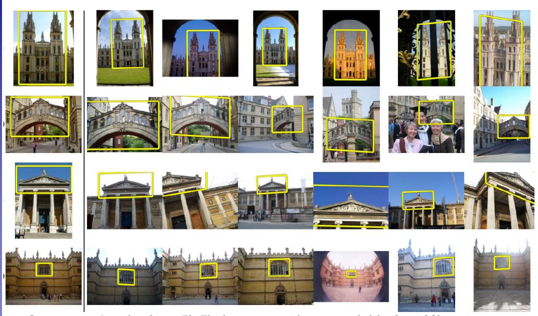
- Snap, pick, pay

- https://www.usatoday.com/videos/tech/2014/10/31/18261641/

Slide credit: Kristen Grauman

Application: Large-Scale Retrieval

Visual Object Recognition Tutorial



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

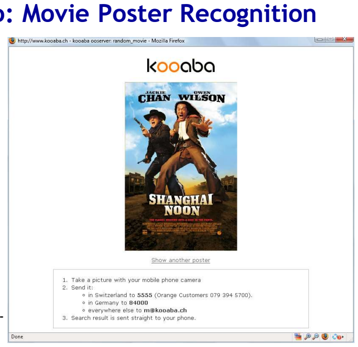
[Philbin CVPR'07]

Web Demo: Movie Poster Recognition

Visual Object Recognition Tutorial

50'000 movie posters indexed

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



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

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?

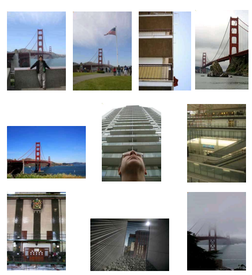
Kristen Grauman

Scoring retrieval quality

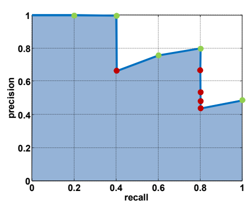
Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):



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



Slide credit: Ondrej Chum

Recognition via alignment

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.

Summary so far

- **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

Making the Sky Searchable: Fast Geometric Hashing for Automated Astrometry

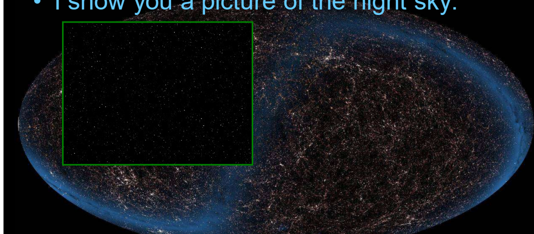
Sam Roweis, Dustin Lang & Keir Mierle
University of Toronto

David Hogg & Michael Blanton
New York University

http://astrometry.net roweis@cs.toronto.edu

Basic Problem

- I show you a picture of the night sky.

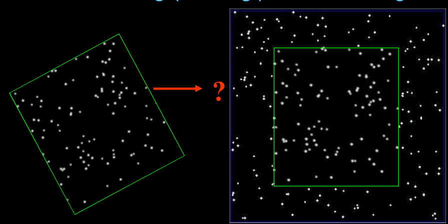


- You tell me where on the sky it came from.

http://astrometry.net roweis@cs.toronto.edu

Rules of the game

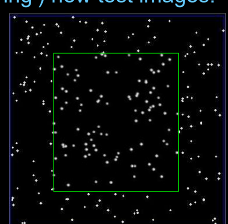
- We start with a catalogue of stars in the sky, and from it build an index which is used to assist us in locating (‘solving’) new test images.



http://astrometry.net roweis@cs.toronto.edu

Rules of the game

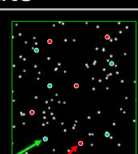
- We start with a catalogue of stars in the sky, and from it build an index which is used to assist us in locating (‘solving’) new test images.
- We can spend as much time as we want building the index but solving should be fast.
- Challenges:
  1) The sky is big.
  2) Both catalogues and pictures are noisy.



http://astrometry.net roweis@cs.toronto.edu

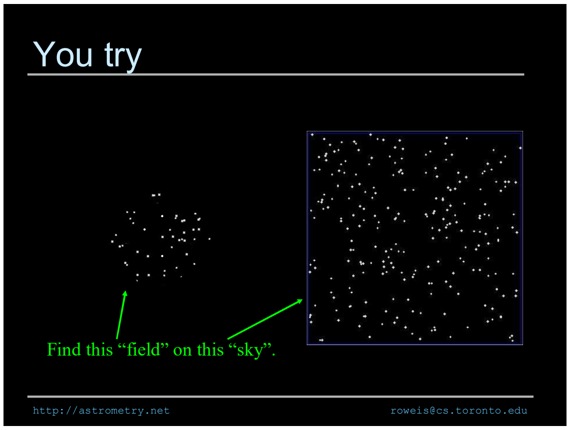
Distractors and Dropouts

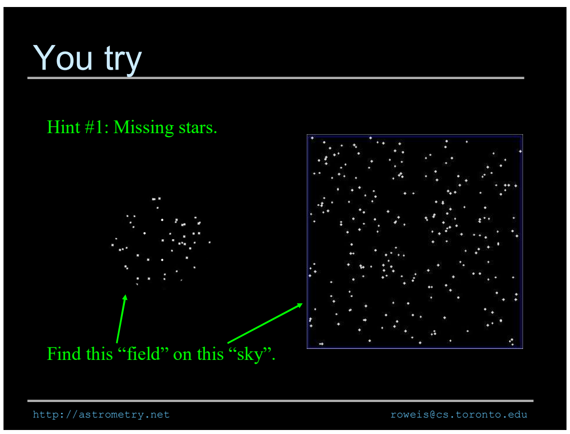
- Bad news:
  Query images may contain some extra stars that are not in your index catalogue, and some catalogue stars may be missing from the image.

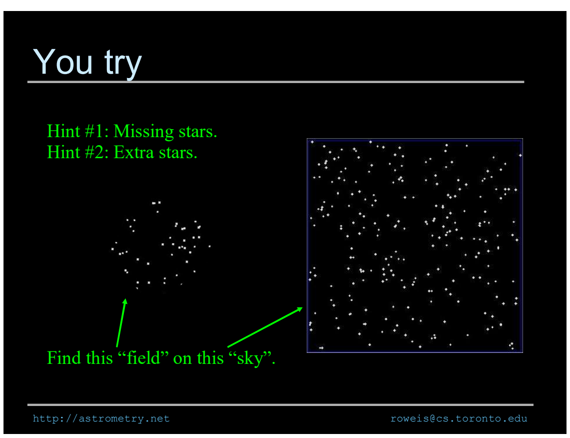


- These “distractors” & “dropouts” mean that naïve matching techniques will not work.

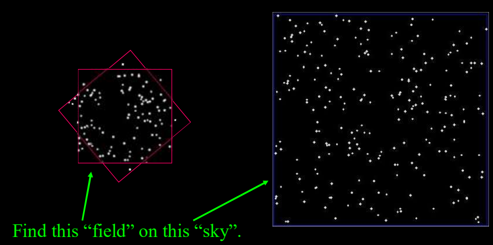
http://astrometry.net roweis@cs.toronto.edu


You try
Find this "field" on this "sky".
http://astrometry.net
roweis@cs.toronto.edu

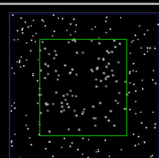

You try
Hint #1: Missing stars.
Find this "field" on this "sky".
http://astrometry.net
roweis@cs.toronto.edu


You try
Hint #1: Missing stars.
Hint #2: Extra stars.
Find this "field" on this "sky".
http://astrometry.net
roweis@cs.toronto.edu

You try



Robust Matching

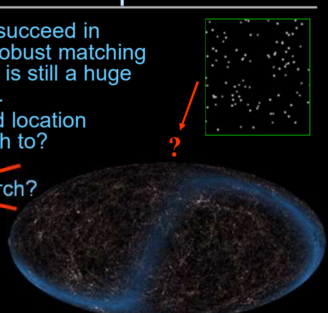


- We need to do some sort of robust matching of the test image to any proposed location on the sky.
- Intuitively, we need to ask: *"Is there an alignment of the test image and the catalogue so that (almost*) every catalogue star in the field of view of the test image lies (almost*) exactly on top of an observed star?"*

Solving the search problem

- Even if we can succeed in finding a good robust matching algorithm, there is still a huge search problem.
- Which proposed location should we match to?
- ~~Exhaustive search?~~

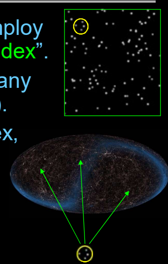
  too expensive!



The Sky is Big™

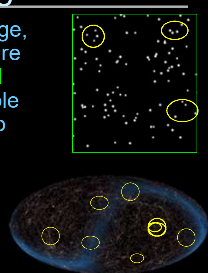
(Inverted) Index of Features

- To solve this problem, we will employ the classic idea of an "inverted index".
- We define a set of "features" for any particular view of the sky (image).
- Then we make an (inverted) index, telling us which views on the sky exhibit certain (combinations of) feature values.
- This is like the question: Which web pages contain the words "machine learning"?



Matching a test image

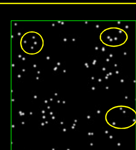
- When we see a new test image, we compute which features are present, and use our inverted index to look up which possible views from the catalogue also have those feature values.
- Each feature generates a candidate list in this way, and by intersecting the lists we can zero in on the true matching view.



Robust Features for Geometric Hashing

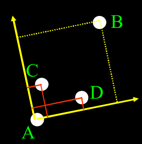
- In our star matching task, the features we chose must be invariant to scale, rotation and translation.

The features we use are the relative positions of nearby quadruples of stars.



Quads as Robust Features

- We encode the relative positions of nearby quadruples of stars (ABCD) using a coordinate system defined by the most widely separated pair (AB).
- Within this coordinate system, the positions of the remaining two stars form a 4-dimensional code for the shape of the quad.



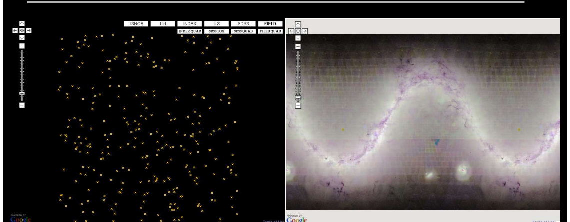
http://astrometry.net roweis@cs.toronto.edu

Solving a new test image

- Identify objects (stars+galaxies) in the image bitmap and create a list of their 2D positions.
- Cycle through all possible quads (brightest first) and compute their corresponding codes.
- Look up the codes in the code KD-tree to find matches within some tolerance; this stage incurs some false positive and false negative matches.
- Each code match returns a candidate position & rotation on the sky. As soon as 2 quads agree on a candidate, we proceed to verify that candidate against all objects in the image.

http://astrometry.net roweis@cs.toronto.edu

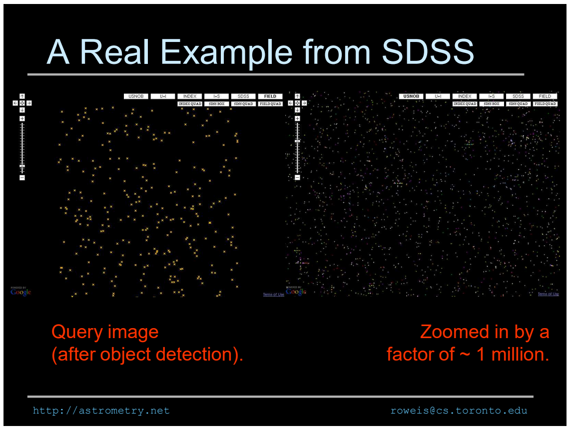
A Real Example from SDSS

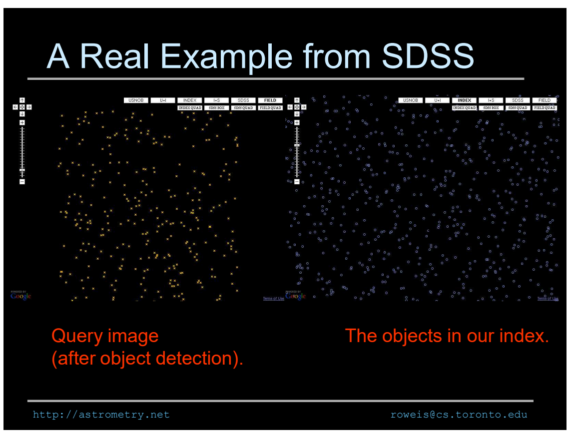


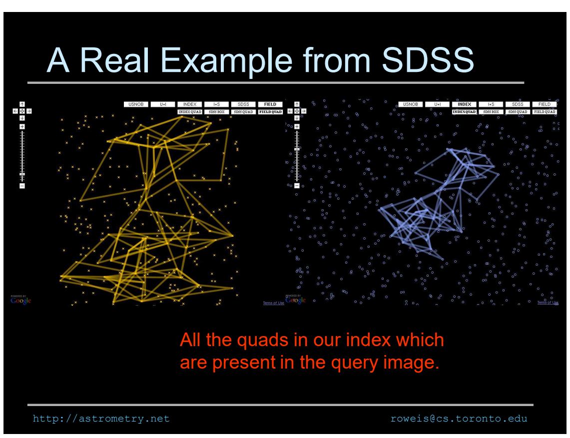
Query image (after object detection).

An all-sky catalogue.

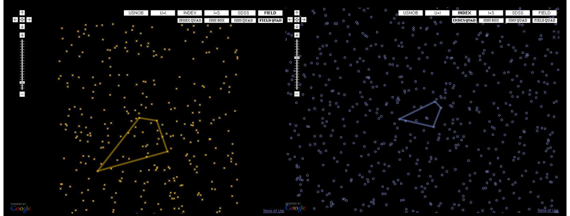
http://astrometry.net roweis@cs.toronto.edu


A Real Example from SDSS
Query image
(after object detection).
Zoomed in by a
factor of ~ 1 million.
http://astrometry.net
roweis@cs.toronto.edu


A Real Example from SDSS
Query image
(after object detection).
The objects in our index.
http://astrometry.net
roweis@cs.toronto.edu


A Real Example from SDSS
All the quads in our index which
are present in the query image.
http://astrometry.net
roweis@cs.toronto.edu

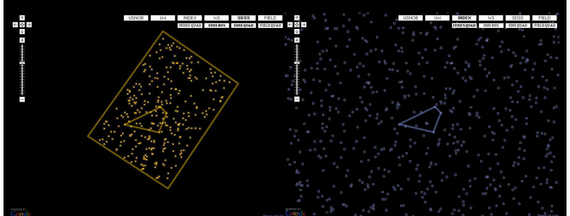
A Real Example from SDSS



A single quad which we happened to try.

http://astrometry.net roweis@cs.toronto.edu

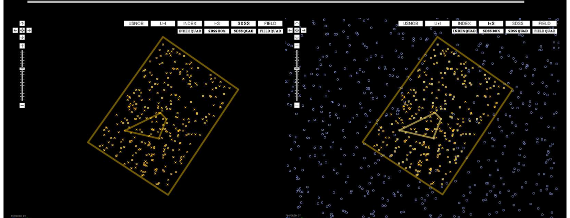
A Real Example from SDSS



The query image scaled, translated & rotated
as specified by the quad.

http://astrometry.net roweis@cs.toronto.edu

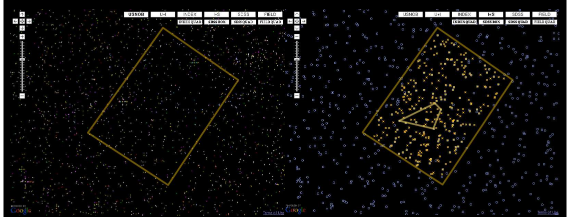
A Real Example from SDSS



The proposed match, on
which we run verification.

http://astrometry.net roweis@cs.toronto.edu

A Real Example from SDSS

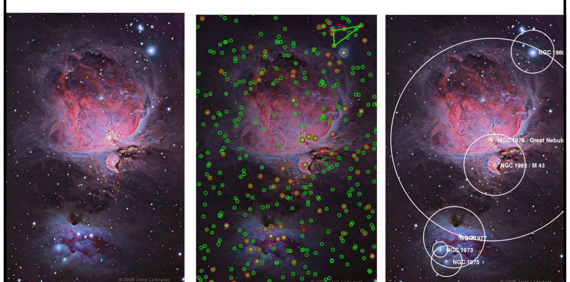


The verified answer, overlaid on the original catalogue.

The proposed match, on which we run verification.

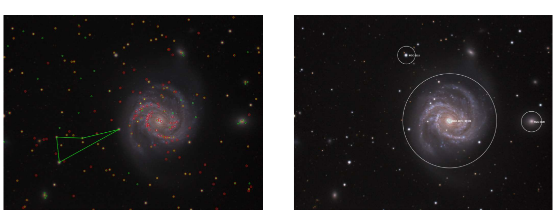
http://astrometry.net roweis@cs.toronto.edu

Example



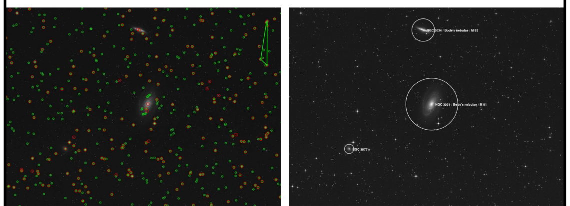
A shot of the Great Nebula, by Jerry Lodriguss (c.2006), from astropix.com
http://astrometry.net/gallery.html

Example



An amateur shot of M100, by Filippo Ciferri (c.2007) from flickr.com
http://astrometry.net/gallery.html

Example



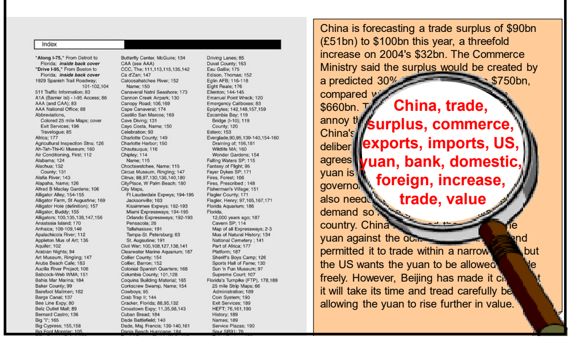
A beautiful image of Bode's nebula (c.2007) by Peter Bresseler, from starlightfriend.de
http://astrometry.net/gallery.html

Recognition via alignment

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.

What else can we borrow from text retrieval?



Query expansion

Query: ***golf green***

Results:

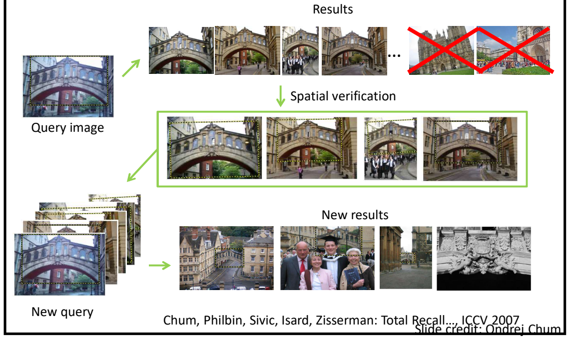
- How can the grass on the ***greens*** at a ***golf*** course be so perfect?
- For example, a skilled ***golf***er expects to reach the ***green*** on a par-four hole in **...**
- Manufactures and sells synthetic ***golf*** putting ***green***s and mats.

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

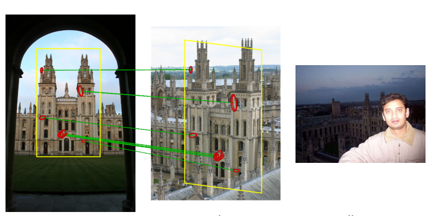
Query Expansion



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

Slide credit: Ondrej Chum

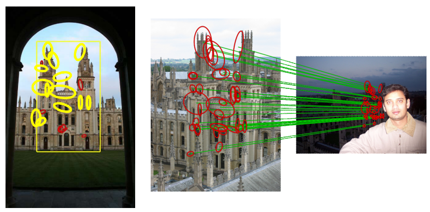
Query Expansion Step by Step



Query Image Retrieved image Originally not retrieved

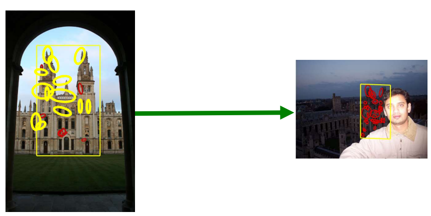
Slide credit: Ondrej Chum

Query Expansion Step by Step



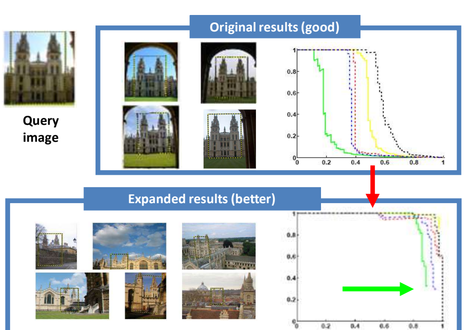
Slide credit: Ondrej Chum

Query Expansion Step by Step



Slide credit: Ondrej Chum

Query Expansion Results



Slide credit: Ondrej Chum

Coming up

- Guest lecture Thurs: Prof. Ray Mooney
  - Video captioning
- Next week:
  - Mining and visual pattern discovery
  - Category recognition / supervised learning