# Instance recognition and discovering patterns

Tues Nov 3

Kristen Grauman
UT Austin

## Announcements

- Change in office hours due to faculty meeting:
  - Tues 2-3 pm – for rest of semester
- Assignment 4 posted Oct 30, due Nov 13.

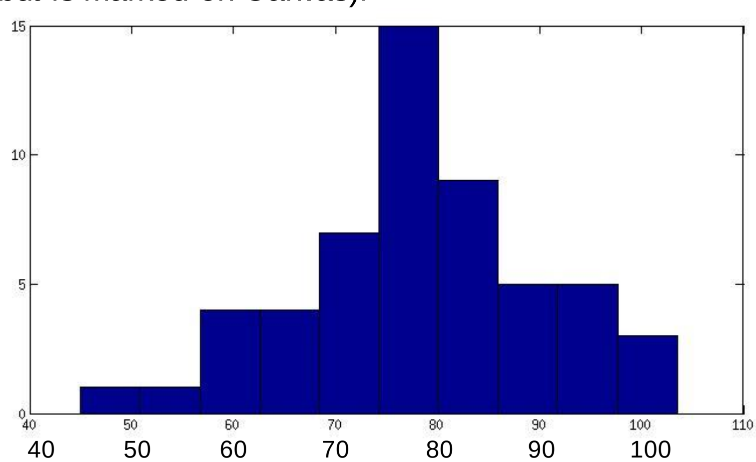
# Today

- Brief review of a few midterm questions
- Instance recognition wrap up:
  - Spatial verification
  - Sky mapping example
  - Query expansion
- Mosaics examples
- Discovering visual patterns
  - Randomized hashing algorithms
  - Mining large-scale image collections

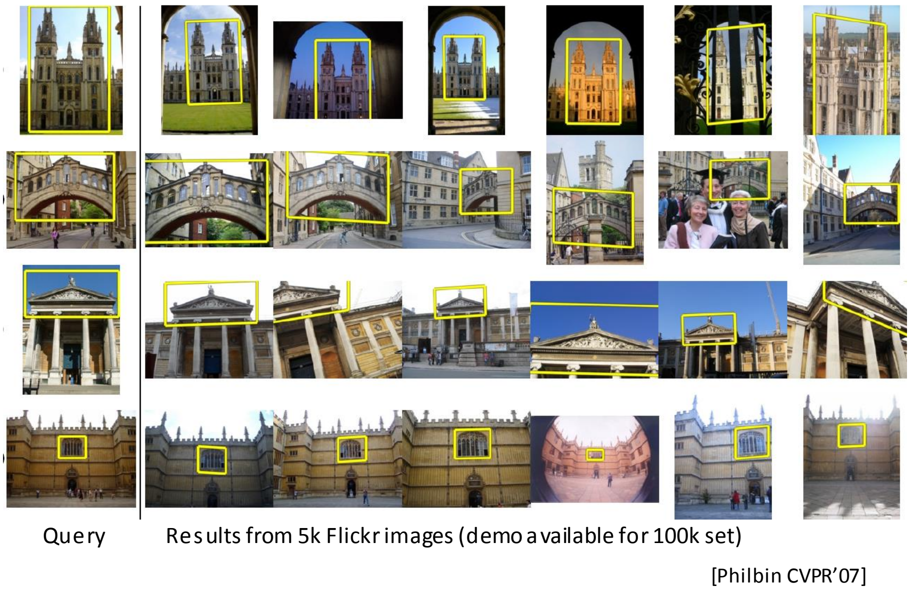
# Midterms

Mean = 78%
Std dev = 12

7 points added to all scores (not marked on your sheet, but is marked on Canvas).



# Last time: instance recognition




# Last time

- **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

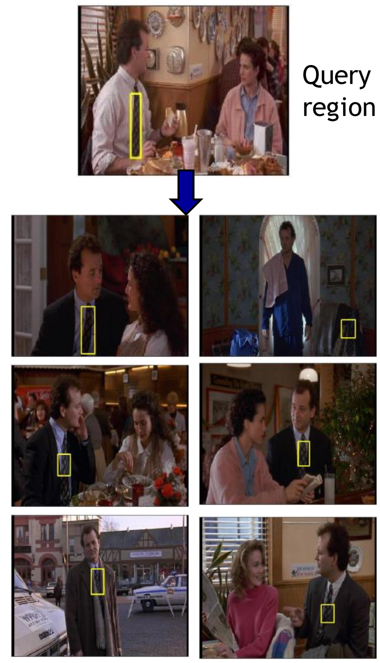
Visual Object Recognition Tutorial

# 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




K. Grauman, B. Leibe

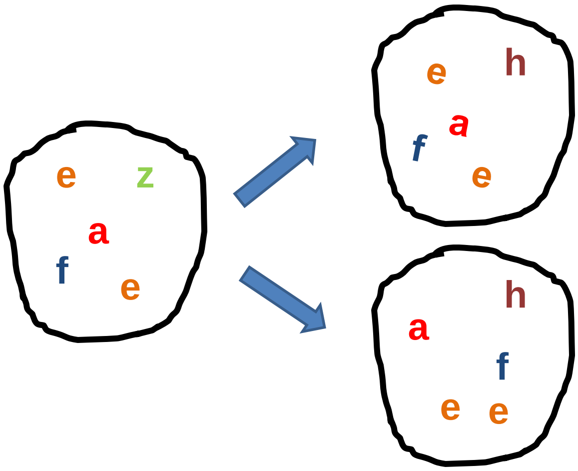
12

# 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?*

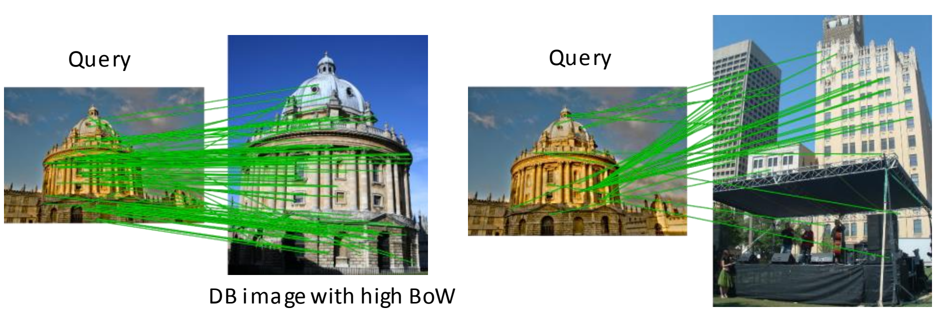



Derek Hoiem

# Spatial Verification

Query

Query



DB image with high BoW similarity

DB image with high BoW similarity

Both image pairs have many visual words in common.

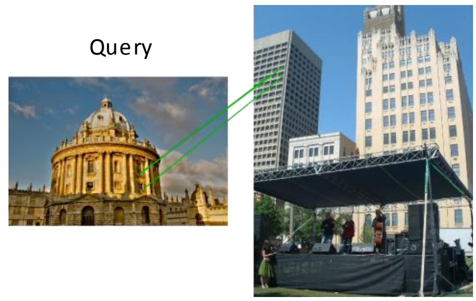
Slide credit: Ondrej Chum

# Spatial Verification

Query

DB image with high BoW similarity

Query



DB image with high BoW similarity

Only some of the matches are mutually consistent

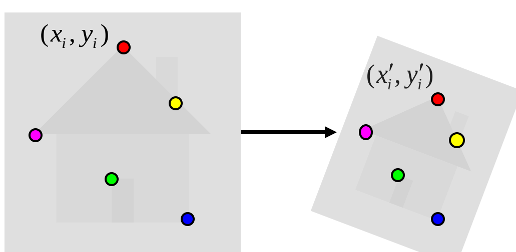
Slide credit: Ondrej Chum

# Spatial Verification: two basic strategies

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

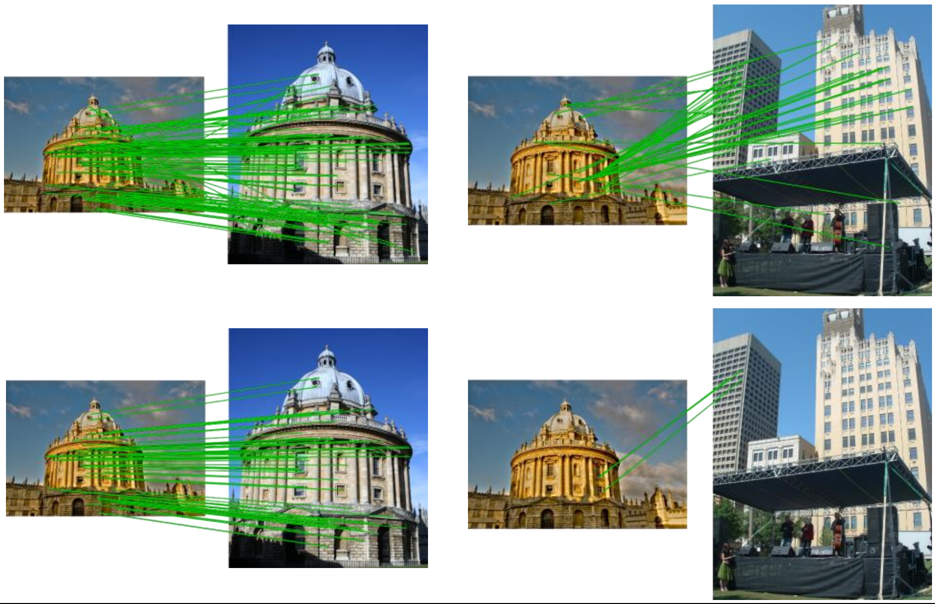



Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

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

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

# 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).

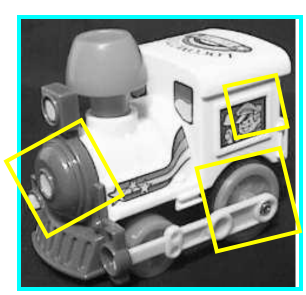
Model


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

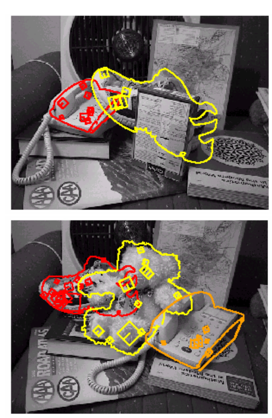
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Lana Lazebnik

## Example result

Background subtract for model boundaries

Objects recognized,



Recognition in spite of occlusion

[Lowe]

# 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 Applications


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

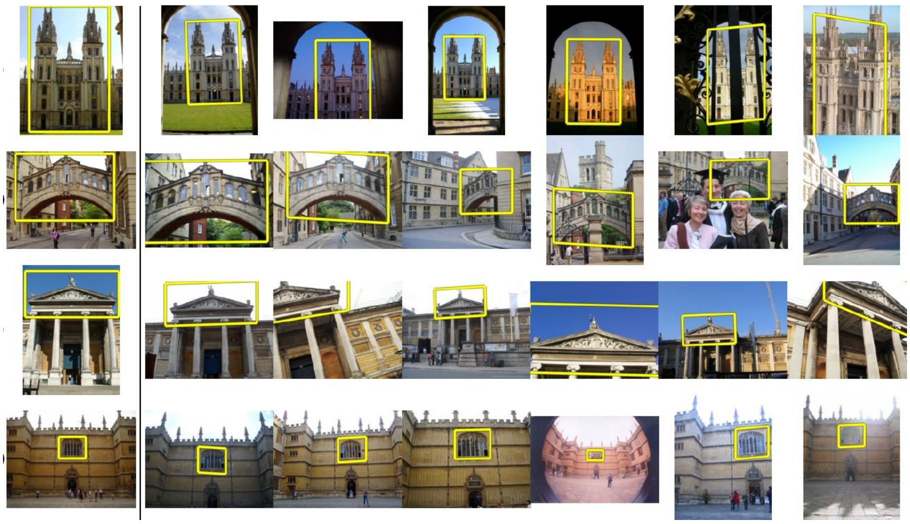
Visual Object Recognition Tutorial

B. Leibe

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

Visual Object Recognition Tutorial

# Application: Large-Scale Retrieval



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

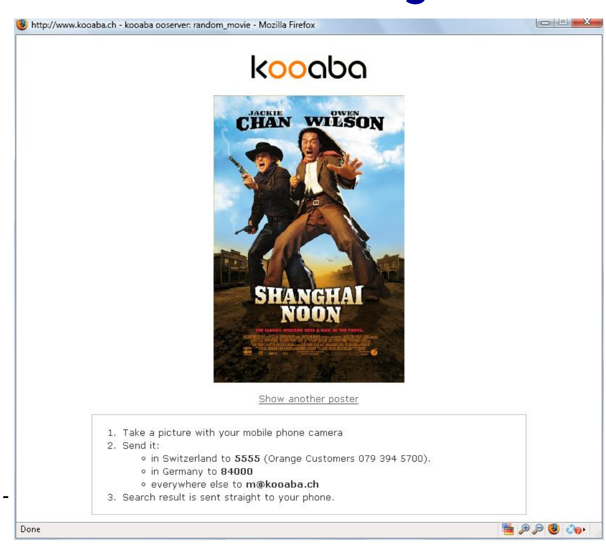
[Philbin CVPR'07]

Visual Object Recognition Tutorial

# 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#

CCPP
CENTER FOR COSMOLOGY AND PARTICLE PHYSICS

Astrometry.net

# Making the Sky Searchable: Fast Geometric Hashing for Automated Astrometry

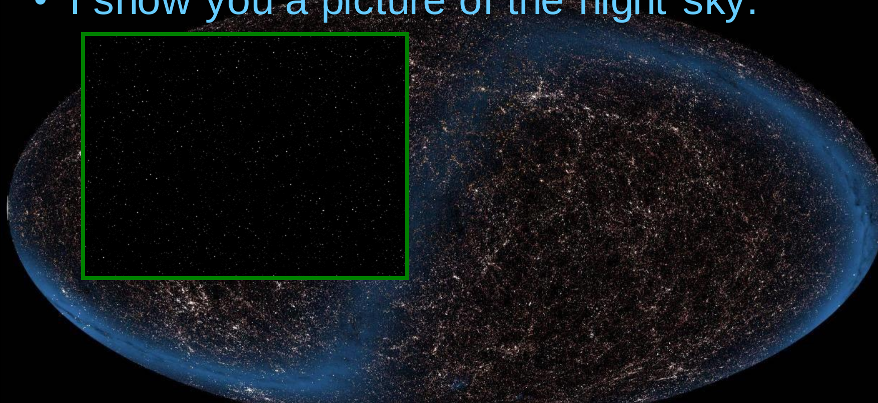
Sam Roweis, Dustin Lang & Keir Mierle
University of Toronto

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.

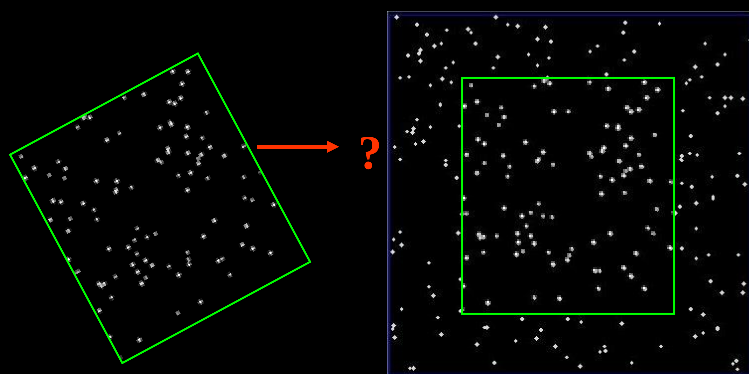


- You tell me where on the sky it came from.

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

# Rules of the game

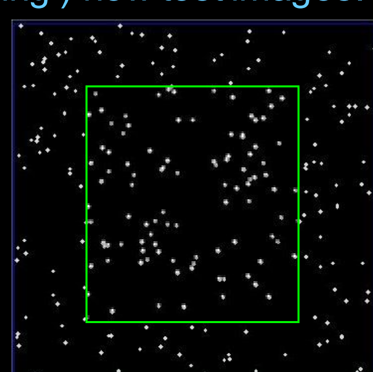
- We start with a catalogue of stars in the sky, and from it build an index which is used to assist us in locating (‘solving’) new test images.



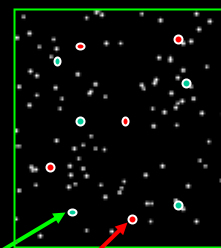



# Rules of the game

- 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.



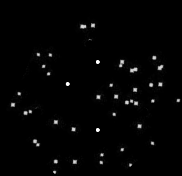
# Distractors and Dropouts

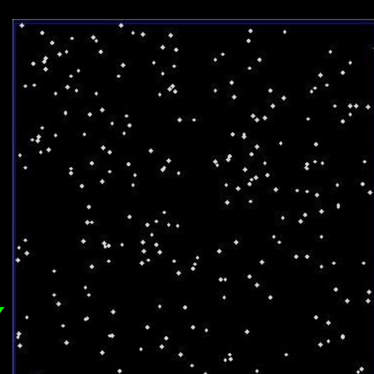


- Bad news:
  Query images may contain some extra stars that are not in your index catalogue, and some catalogue stars may be missing from the image.
- These “distractors” & “dropouts” mean that naïve matching techniques will not work.



# You try

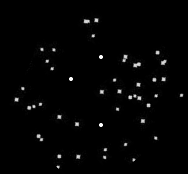


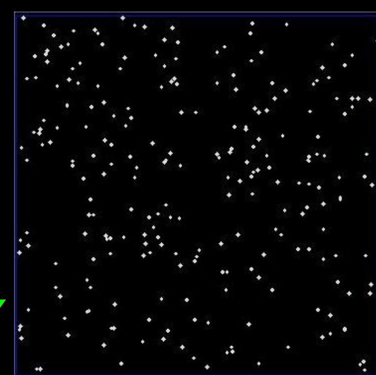


Find this “field” on this “sky”.

# You try

Hint #1: Missing stars.



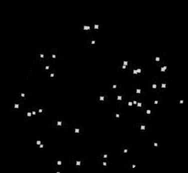


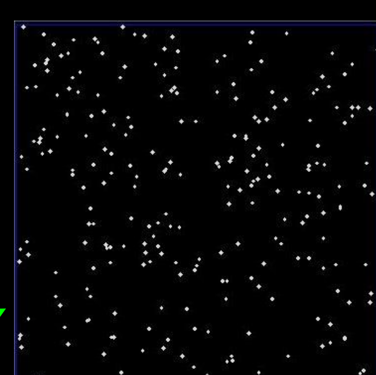
Find this "field" on this "sky".



# You try

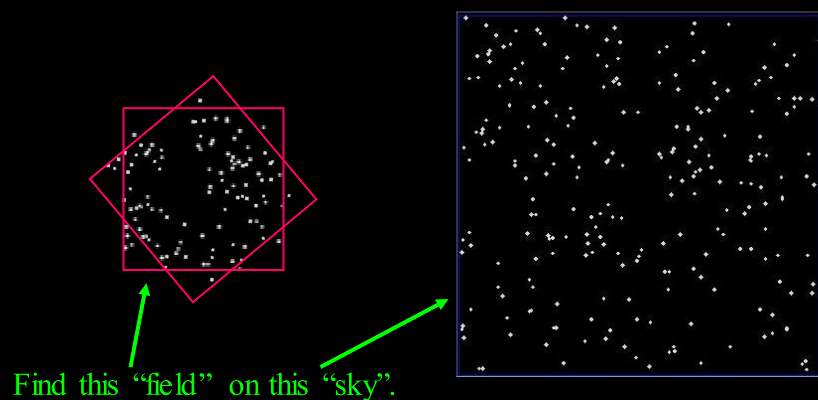
Hint #1: Missing stars.
Hint #2: Extra stars.





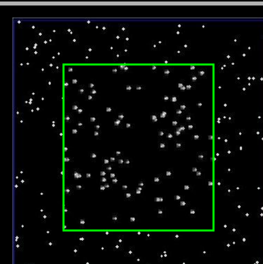
Find this "field" on this "sky".

# 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™

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

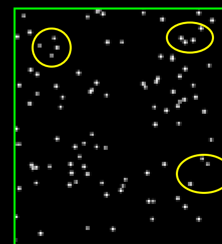
# (Inverted) Index of Features

- 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"?

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

## 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.

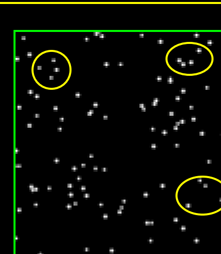




## Robust Features for Geometric Hashing

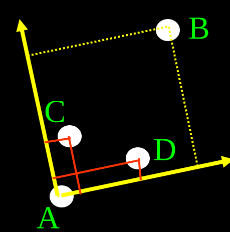
- In our star matching task, the features we chose must be invariant to scale, rotation and translation.

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.

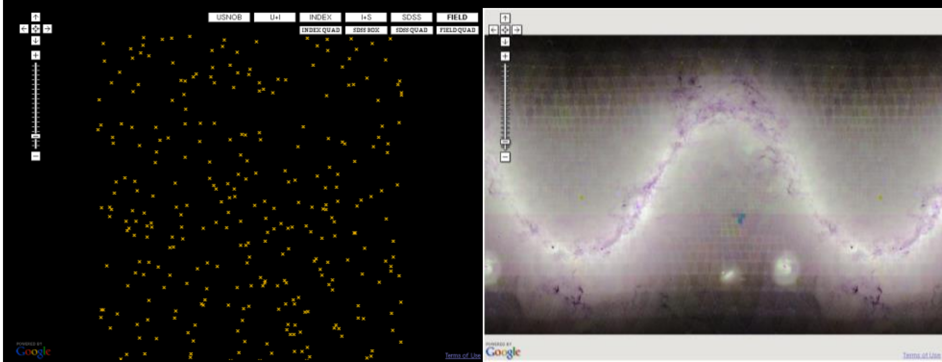





# Solving a new test image

- Identify objects (stars+galaxies) in the image bitmap and create a list of their 2D positions.
- Cycle through all possible valid* 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.

# A Real Example from SDSS

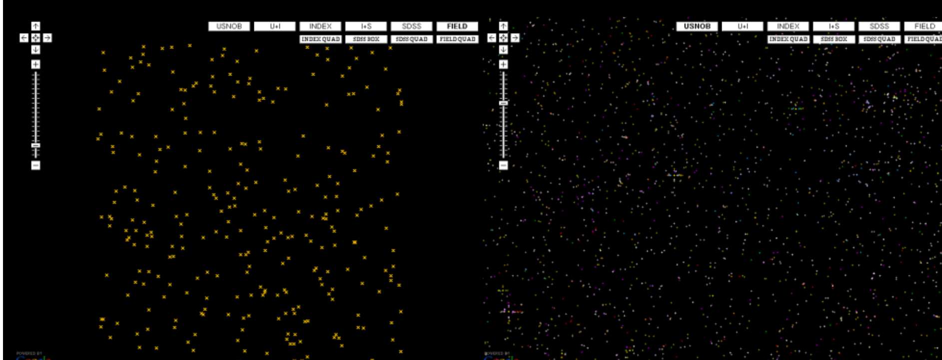


Query image
(after object detection).

An all-sky catalogue.

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

# A Real Example from SDSS

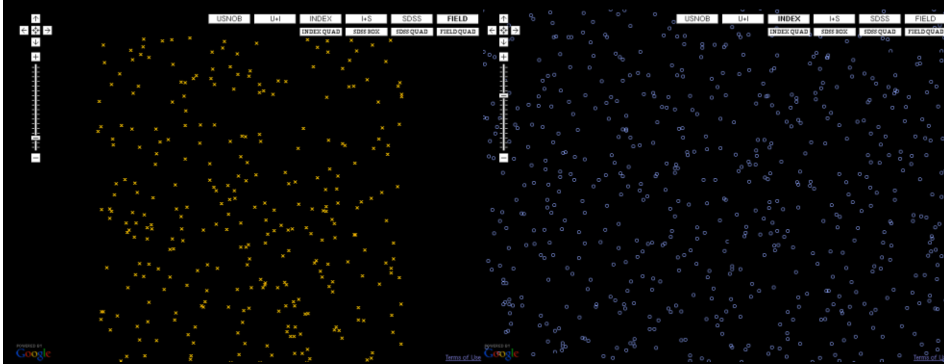


Query image
(after object detection).

Zoomed in by a
factor of ~ 1 million.

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

# A Real Example from SDSS

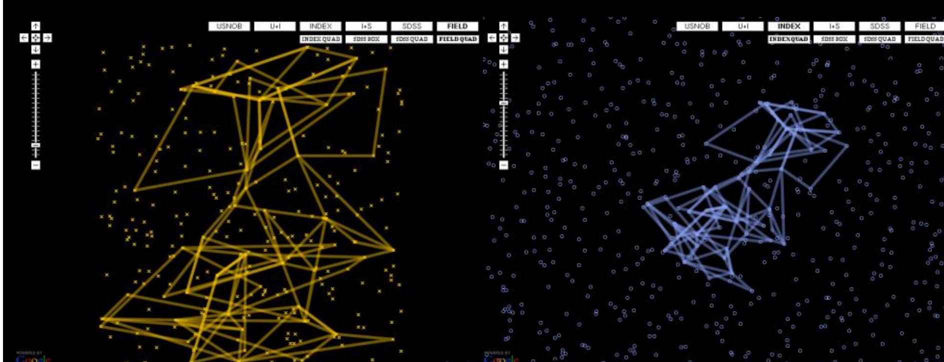


Query image
(after object detection).

The objects in our index.

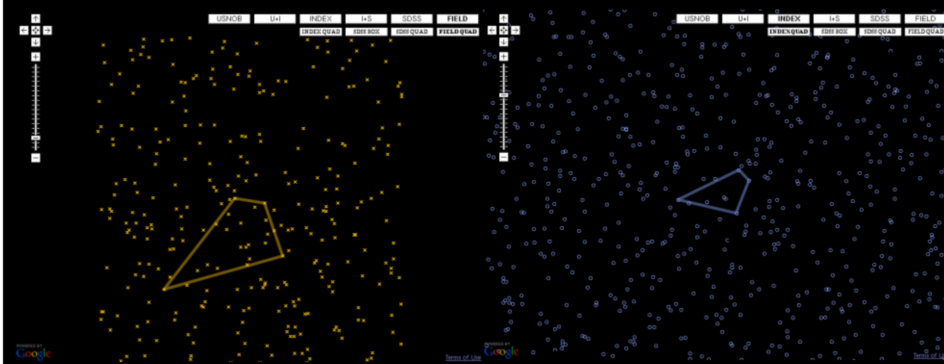


# A Real Example from SDSS



All the quads in our index which
are present in the query image.

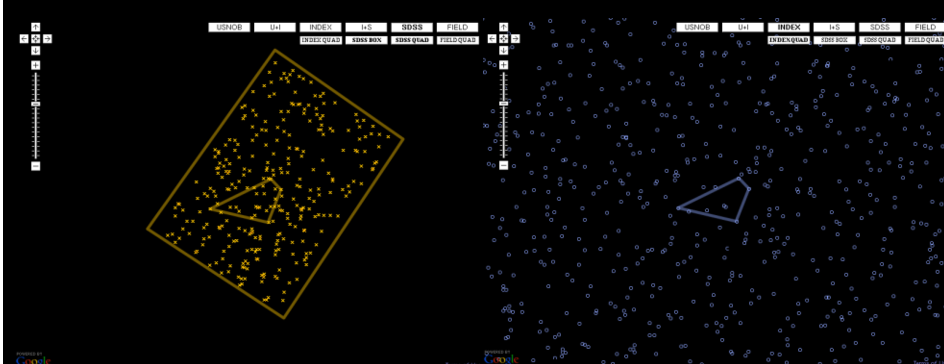
# A Real Example from SDSS



A single quad which we happened to try.

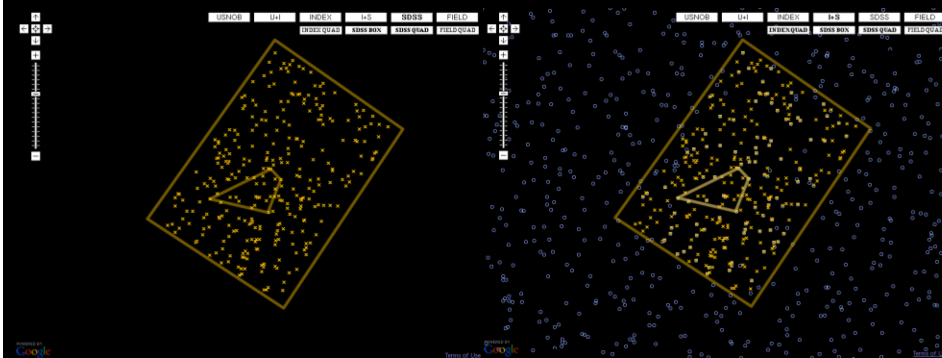


# A Real Example from SDSS



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

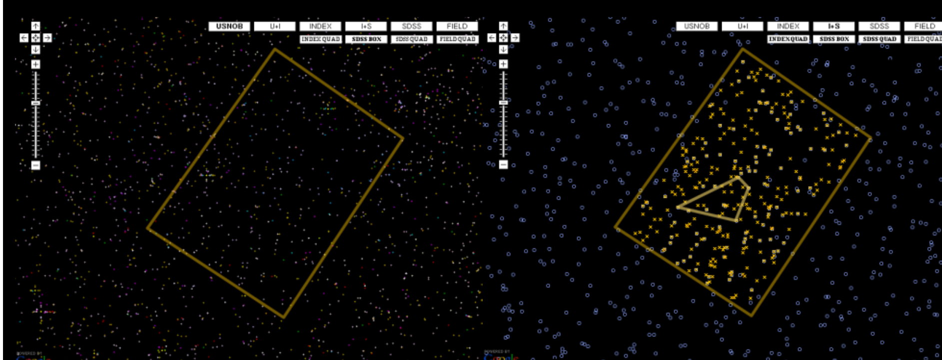
# A Real Example from SDSS




The proposed match, on which we run verification.

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

# A Real Example from SDSS

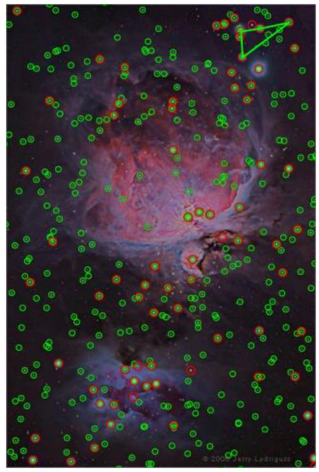
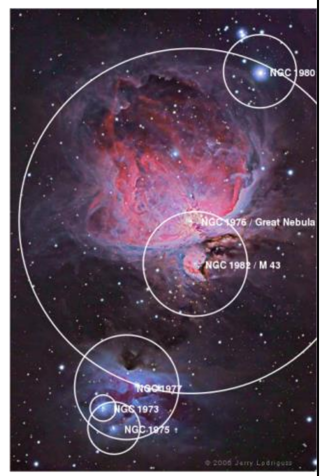



The verified answer, overlaid on the original catalogue.

The proposed match, on which we run verification.

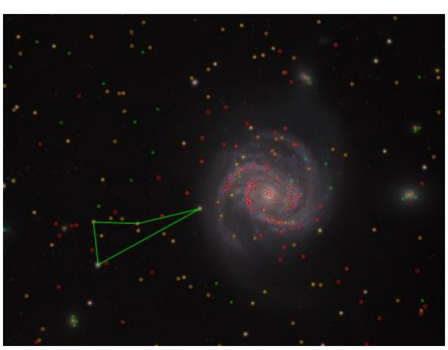
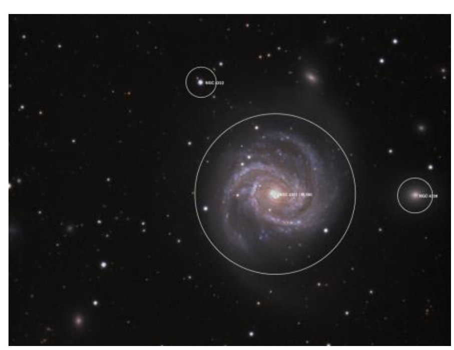
http://astrometry.net roweis@cs.toronto.edu

# Example

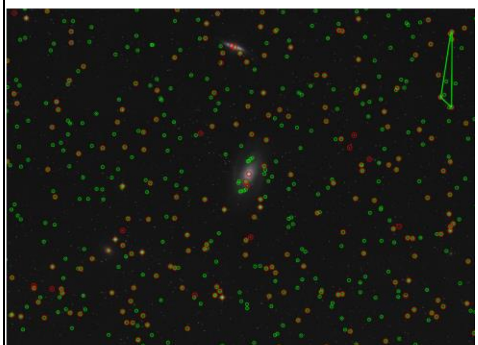

A shot of the Great Nebula, by Jerry Lodriguss (c.2006), from astropix.com
http://astrometry.net/gallery.html

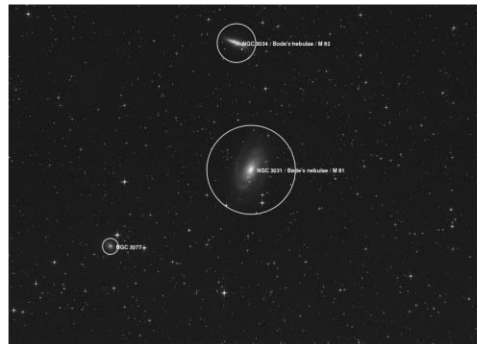
# Example

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

# 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?

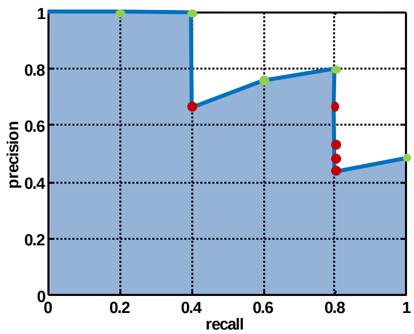
Kristen Grauman

# Scoring retrieval quality

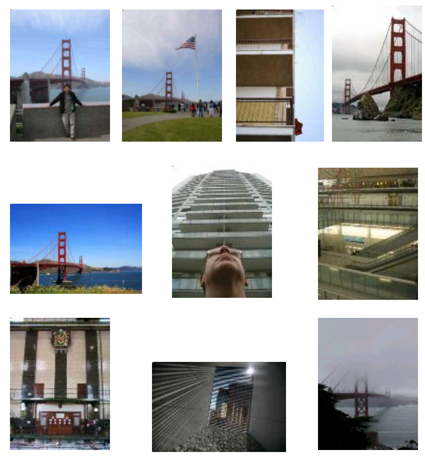
Query

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

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




Results (ordered):



Slide credit: Ondrej Chum

# Recognition via alignment

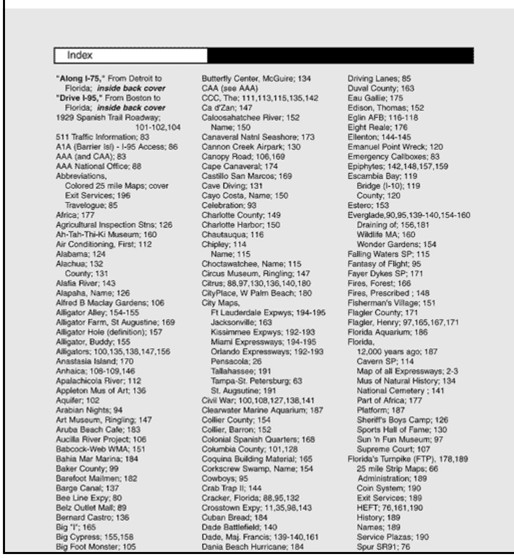
**Pros**:

- 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.

# 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 ***greens*** 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

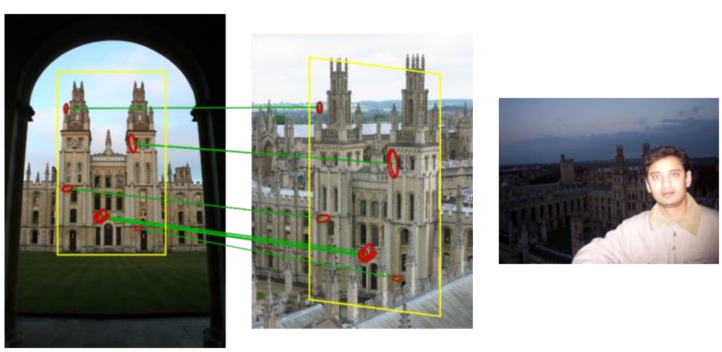
Slide credit: Ondrej Chum

# Query Expansion


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

Slide credit: Ondrej Chum

# Query Expansion Step by Step



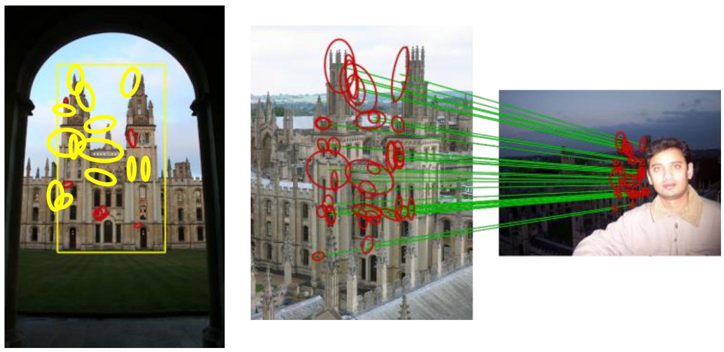
Query Image

Retrieved image

Originally not retrieved

Slide credit: Ondrej Chum

# Query Expansion Step by Step



Slide credit: Ondrej Chum

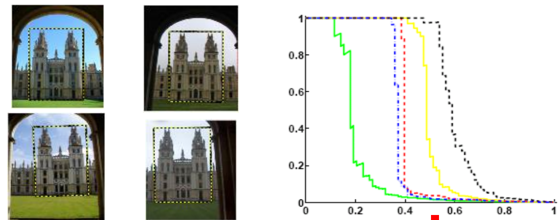
# Query Expansion Step by Step

Slide credit: Ondrej Chum

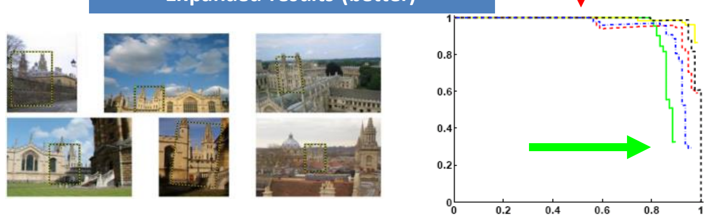
# Query Expansion Results

**Query image**

**Original results (good)**



**Expanded results (better)**



Slide credit: Ondrej Chum

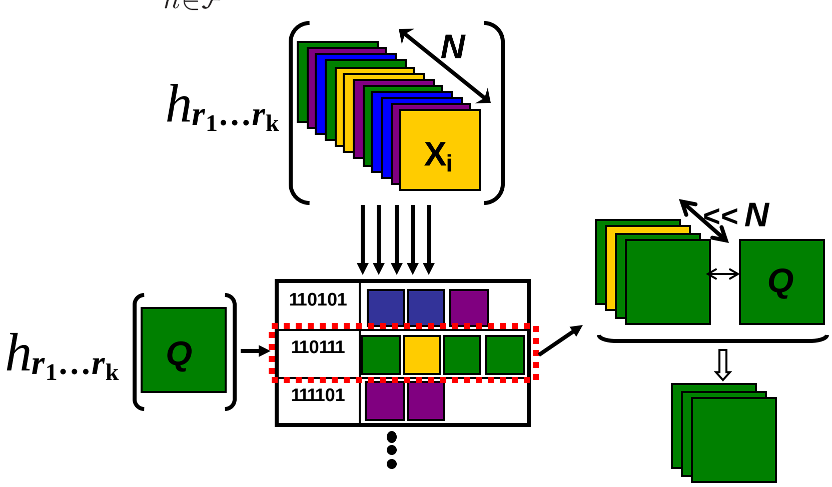
# Today

- Brief review of a few midterm questions
- Instance recognition wrap up:
  - Spatial verification
  - Sky mapping example
  - Query expansion
- Mosaics examples
- Discovering visual patterns
  - Randomized hashing algorithms
  - Mining large-scale image collections

# Locality Sensitive Hashing (LSH)

*[Indyk and Motwani '98, Gionis et al.'99, Charikar '02, Andoni et al. '04]*

$$\Pr_{h \in \mathcal{F}} [h(x) = h(y)] = sim(x, y)$$

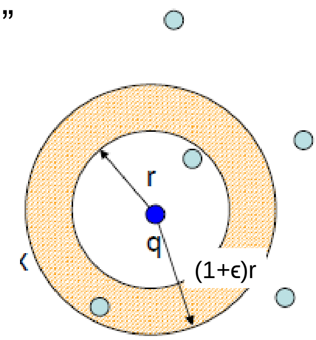



Kristen Grauman

# Locality Sensitive Hashing (LSH)

*[Indyk and Motwani '98, Gionis et al.'99, Charikar '02, Andoni et al. '04]*

- Formally, ensures "approximate" nearest neighbor search
  - With high probability, return a neighbor within radius (1+ϵ)r, if there is one.
  - Guarantee to search only $O\left(N^{\frac{1}{1+\epsilon}}\right)$ of the database
- LSH functions originally for Hamming metric, Lp norms, inner product.

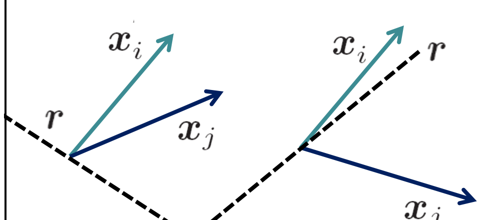



Kristen Grauman

# LSH function example: inner product similarity

The probability that a *random hyperplane* separates two unit vectors depends on the angle between them:

$$\Pr[\text{sign}(\boldsymbol{x}_i^T\boldsymbol{r}) = \text{sign}(\boldsymbol{x}_j^T\boldsymbol{r})] = 1 - \frac{1}{\pi}\cos^{-1}(\boldsymbol{x}_i^T\boldsymbol{x}_j)$$




High dot product: unlikely to split

Lower dot product: likely to split

Corresponding hash function:

$$h_{\boldsymbol{r}}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{r}^T\boldsymbol{x} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

for $\vec{r}_i \sim N(\mu = 0, \sigma^2 = 1)$

*[Goemans and Williamson 1995, Charikar 2004]*

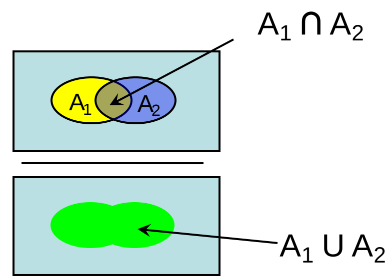
Kristen Grauman

# LSH function example: Min-hash for set overlap similarity

*[Broder, 1999]*

$$\Pr_{h \in \mathcal{F}}[h(x) = h(y)] = sim(x, y)$$

$$\mathrm{sim}(\mathcal{A}_1, \mathcal{A}_2) = \frac{|\mathcal{A}_1 \cap \mathcal{A}_2|}{|\mathcal{A}_1 \cup \mathcal{A}_2|} \in \langle 0, 1 \rangle$$




Kristen Grauman

# LSH function example: Min-hash for set overlap similarity

| Vocabulary | | | | | | Set ***A*** | Set ***B*** | Set ***C*** |
|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | A B C | B C D | A E F |

**Random orderings** / **min-Hash**

| | A | B | C | D | E | F | | Set A | Set B | Set C |
|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$: | 0.41 (3) | 0.90 (6) | 0.22 (2) | 0.59 (5) | 0.75 (4) | 0.07 (1) | ~ Un (0,1) | C | C | F |
| $f_2$: | 0.19 (1) | 0.31 (2) | 0.94 (6) | 0.35 (3) | 0.88 (5) | 0.63 (4) | ~ Un (0,1) | A | B | A |
| $f_3$: | 3 | 2 | 1 | 6 | 4 | 5 | | C | C | A |
| $f_4$: | 4 | 3 | 5 | 6 | 1 | 2 | | B | B | E |

overlap (***A***,***B***) = 3/4 (1/2)    overlap (***A***,***C***) = 1/4 (1/5)    overlap (***B***,***C***) = 0 (0)

Slide credit: Ondrej Chum

*[Broder, 1999]*

## LSH function example: Min-hash for set overlap similarity

*A*: A E J Q R V Y

*B*: A C E Q V Z

Ordering by $f_2$

*A* ∪ *B*: A C E J Q R V Y Z

h1(*A*) A    h1(*B*) A

h2(*A*) Q    h2(*B*) C

$$P(h(A) = h(B)) = \frac{|A \cap B|}{|A \cup B|}$$

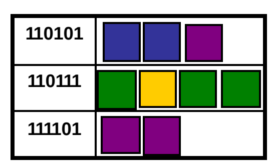
Slide credit: Ondrej Chum

*[Broder, 1999]*

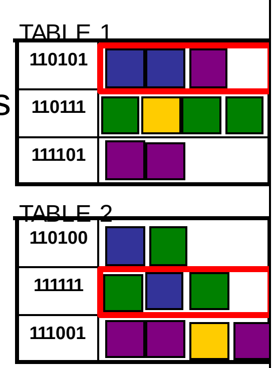
## Multiple hash functions and tables

- Generate *k* such hash functions, concatenate outputs into hash key:

$$P\left(h_{1,\dots,k}(x) = h_{1,\dots,k}(y)\right) =$$




- To increase recall, search multiple independently generated hash tables
  - Search/rank the union of collisions in each table, or
  - Require that two examples in at least *T* of the tables to consider them similar.




Kristen Grauman

# Mining for common visual patterns

In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

- What is common?
- What is unusual?
- What co-occurs?
- Which exemplars are most representative?

Kristen Grauman

# Mining for common visual patterns

In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

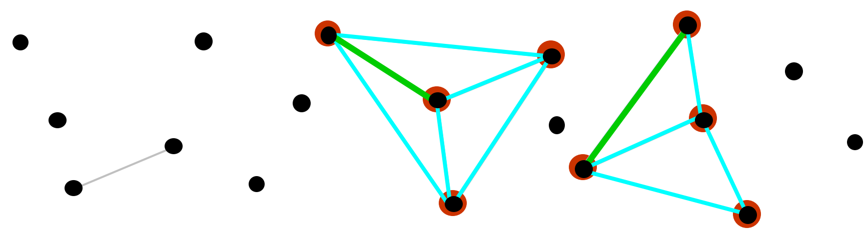
We'll look at a few examples:

- **Connected component clustering** via hashing [Geometric Min-hash, Chum et al. 2009]
- **Visual Rank** to choose "image authorities" [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns [Quack et al., 2007]

Kristen Grauman

# Connected component clustering with hashing

1. Detect seed pairs via hash collisions
2. Hash to related images
3. Compute connected components of the graph



*Contrast with frequently used quadratic-time clustering algorithms*

Slide credit: Ondrej Chum

# Geometric Min-hash

*[Chum, Perdoch, Matas, CVPR 2009]*


- Main idea: build spatial relationships into the hash key construction:
  - Select first hash output according to min hash ("central word")
  - Then append subsequent hash outputs from within its neighborhood

Figure from Ondrej Chum

# Results: Geometric Min-hash clustering

*[Chum, Perdoch, Matas, CVPR 2009]*

All Soul's
Ashmolean
Balliol
Bodleian
Christ Church
Cornmarket

Hertford
Keble
Magdalen
Pitt Rivers
Radcliffe Camera

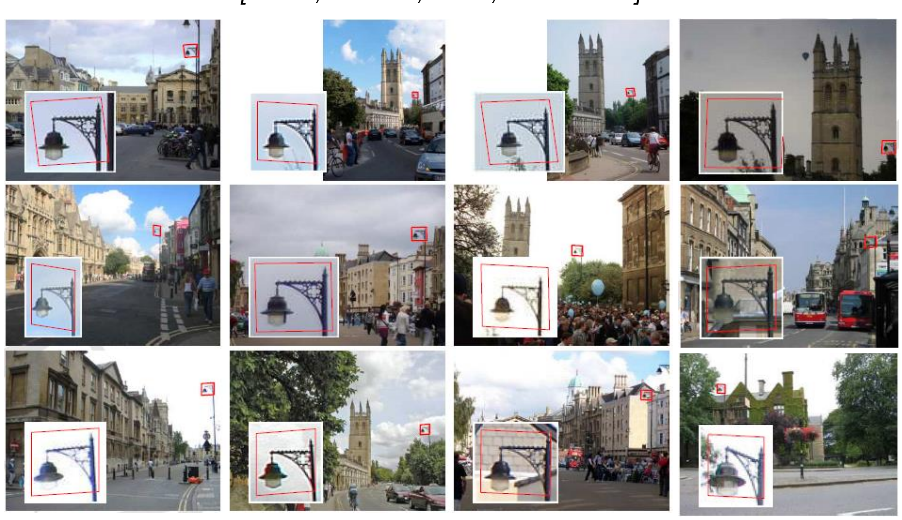
100 000 Images downloaded from FLICKR

Includes 11 Oxford Landmarks with manually labeled ground truth

Slide credit: Ondrej Chum

# Results: Geometric Min-hash clustering

*[Chum, Perdoch, Matas, CVPR 2009]*



Discovering small objects

Slide credit: Ondrej Chum

# Results: Geometric Min-hash clustering

*[Chum, Perdoch, Matas, CVPR 2009]*

Discovering small objects

Slide credit: Ondrej Chum

# Mining for common visual patterns

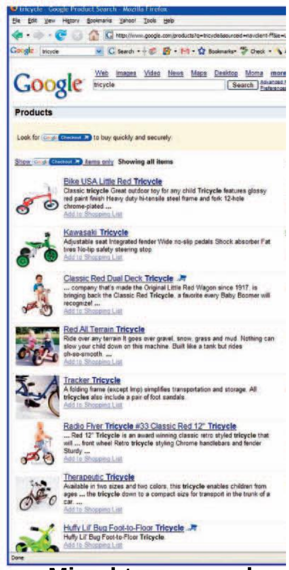
In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

We'll look briefly at a few recent examples:

- **Connected component clustering** via hashing [Geometric Min-hash, Chum et al. 2009]
- **Visual Rank** to choose "image authorities" [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns [Quack et al., 2007]

## Visual Rank: motivation

Product search



Mixed-type search

- **Goal:** select small set of "best" images to display among millions of candidates

Kristen Grauman

## Visual Rank

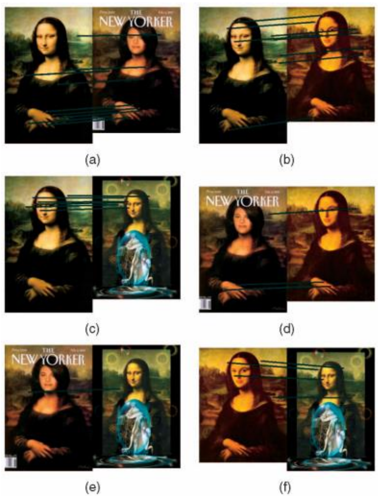
*[Jing and Baluja, PAMI 2008]*

- Compute relative "authority" of an image based on random walk principle.
  - Application of PageRank to visual data
- **Main ideas**:
  - Graph weights = number of matched local features between two images
  - Exploit text search to narrow scope of each graph
  - Use LSH to make similarity computations efficient

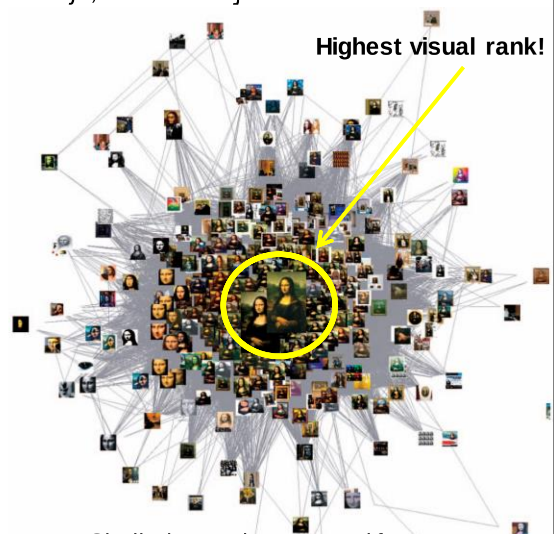
Kristen Grauman

# Results: Visual Rank

*[Jing and Baluja, PAMI 2008]*




Original has more matches to rest

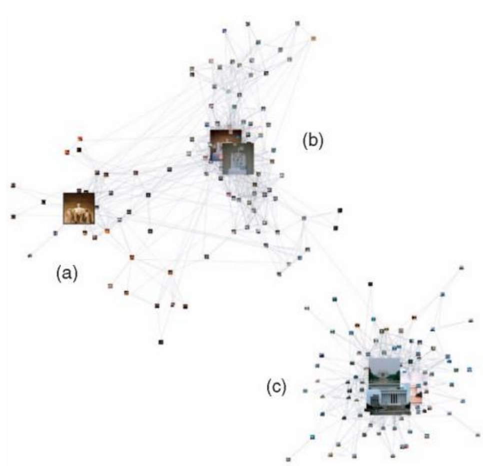



Similarity graph generated from top 1,000 text search results of "Mona-Lisa"

Kristen Grauman

# Results: Visual Rank

*[Jing and Baluja, PAMI 2008]*




Similarity graph generated from top 1,000 text search results of "Lincoln Memorial".
Note the diversity of the high-ranked images.

Kristen Grauman

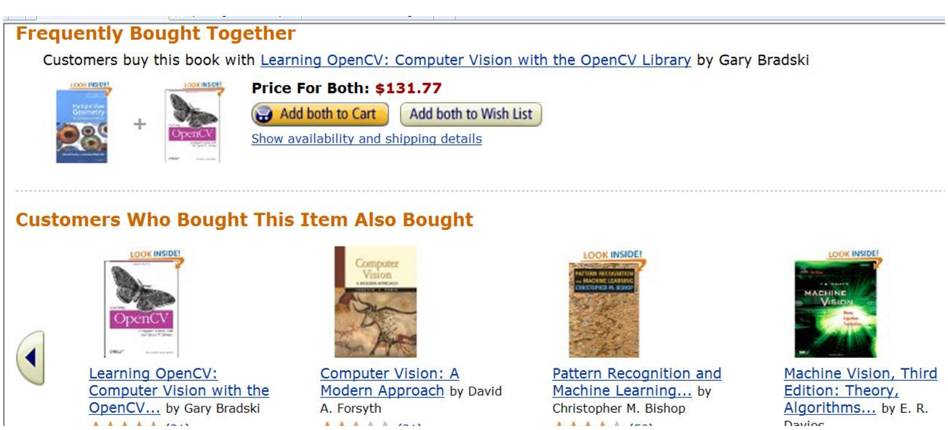
# Mining for common visual patterns

In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

We'll look briefly at a few recent examples:

- **Connected component clustering** via hashing [Geometric Min-hash, Chum et al. 2009]
- **Visual Rank** to choose "image authorities" [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns [Quack et al., 2007]

# Frequent item-sets

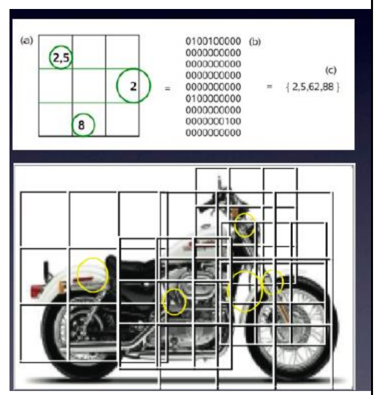



Kristen Grauman

# Frequent item-set mining for spatial visual patterns

*[Quack, Ferrari, Leibe, Van Gool, CIVR 2006, ICCV 2007]*

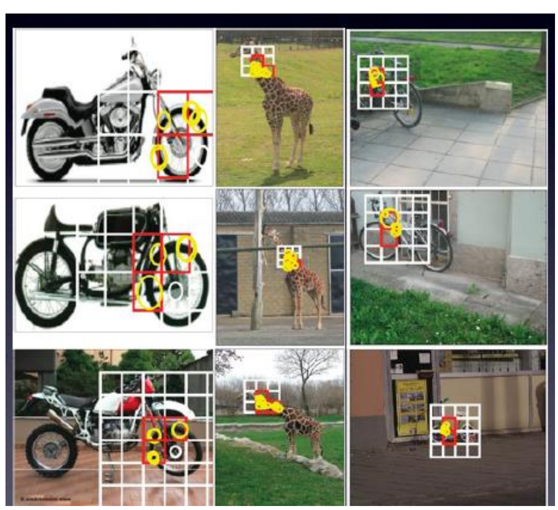
- What configurations of local features frequently occur in large collection?
- **Main idea**: Identify *item-sets* (visual word layouts) that often occur in *transactions* (images)
- Efficient algorithms from data mining (e.g., Apriori algorithm, Agrawal 1993)




Kristen Grauman

# Frequent item-set mining for spatial visual patterns

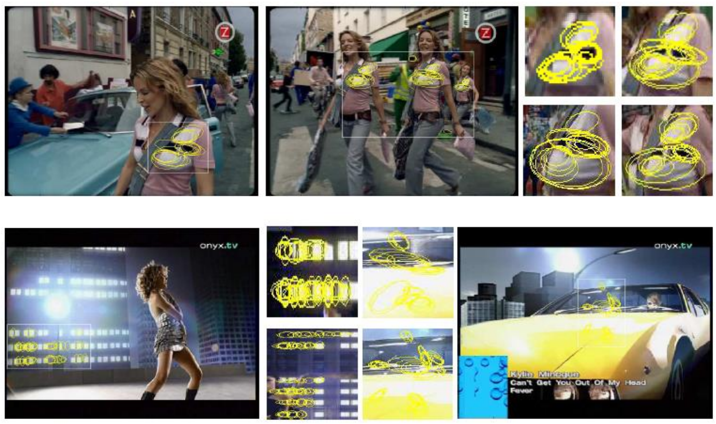
*[Quack, Ferrari, Leibe, Van Gool, CIVR 2006, ICCV 2007]*



Kristen Grauman

# Frequent item-set mining for spatial visual patterns

*[Quack, Ferrari, Leibe, Van Gool, CIVR 2006, ICCV 2007]*

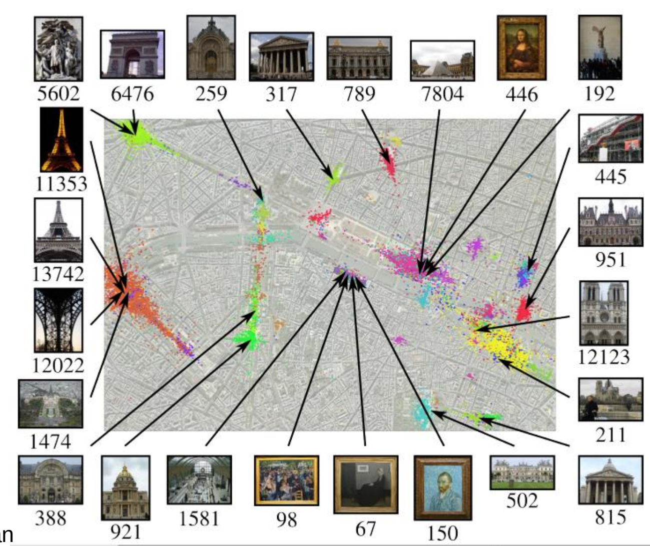


Two example itemset clusters

Kristen Grauman

# Discovering favorite views

*Discovering Favorite Views of Popular Places with Iconoid Shift. T. Weyand and B. Leibe. ICCV 2011.*




Kristen Grauman

# Today

- Brief review of a few midterm questions
- Instance recognition wrap up:
  - Spatial verification
  - Sky mapping example
  - Query expansion
- Mosaics examples
- Discovering visual patterns
  - Randomized hashing algorithms
  - Mining large-scale image collections

# Coming up

- Category recognition
- Supervised learning
- Sliding window object detection (Faces!)