

Instance Recognition



Jia-Bin Huang

Virginia Tech

ECE 6554 Advanced Computer Vision

Administrative stuffs

- Paper review submitted?
- Topic presentation
- Experiment presentation
- “For”/ “Against” discussion lead
- Questions?

Today's class

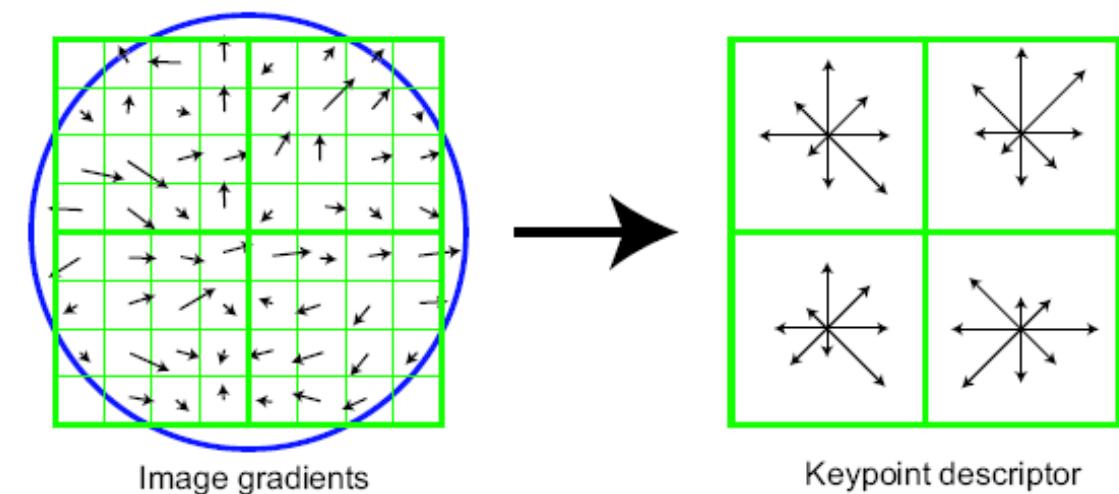
- Review keypoint detection and descriptors
- Review SIFT features
- Indexing features
- Fast image search

Discussion – Think-pair-share

- Find a person you don't know
- Discuss
 - strength,
 - weakness, and
 - potential extension
- Share with class

Keypoint detection and descriptors

- **Keypoint detection:** repeatable and distinctive
 - Corners, blobs, stable regions
 - Harris, DoG
- **Descriptors:** robust and selective
 - spatial histograms of orientation
 - SIFT



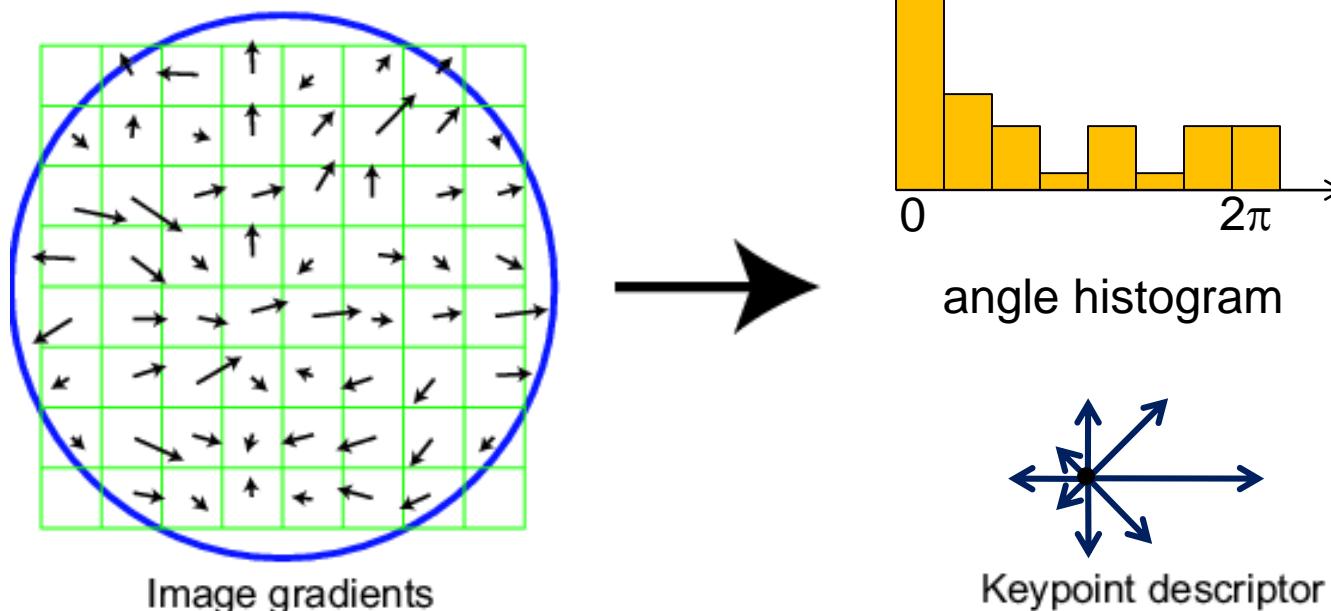
Local Descriptors

- The ideal descriptor should be
 - Robust
 - Distinctive
 - Compact
 - Efficient
- Most available descriptors focus on edge/gradient information
 - Capture texture information
 - Color rarely used

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

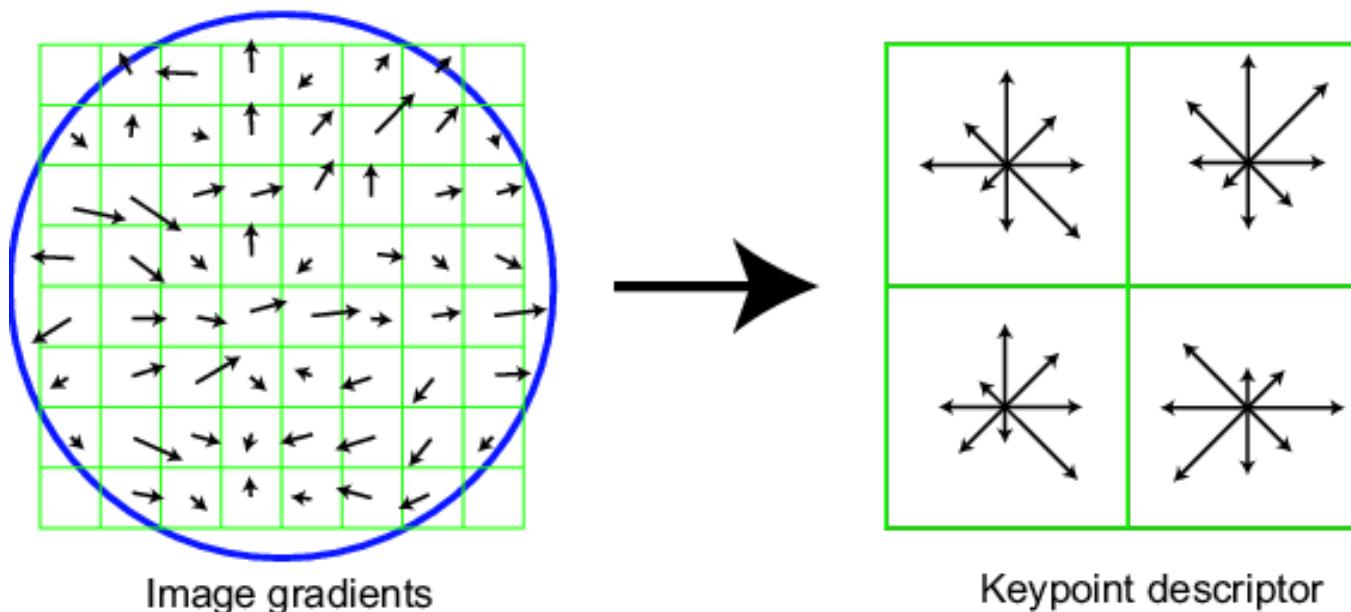


Adapted from slide by David Lowe

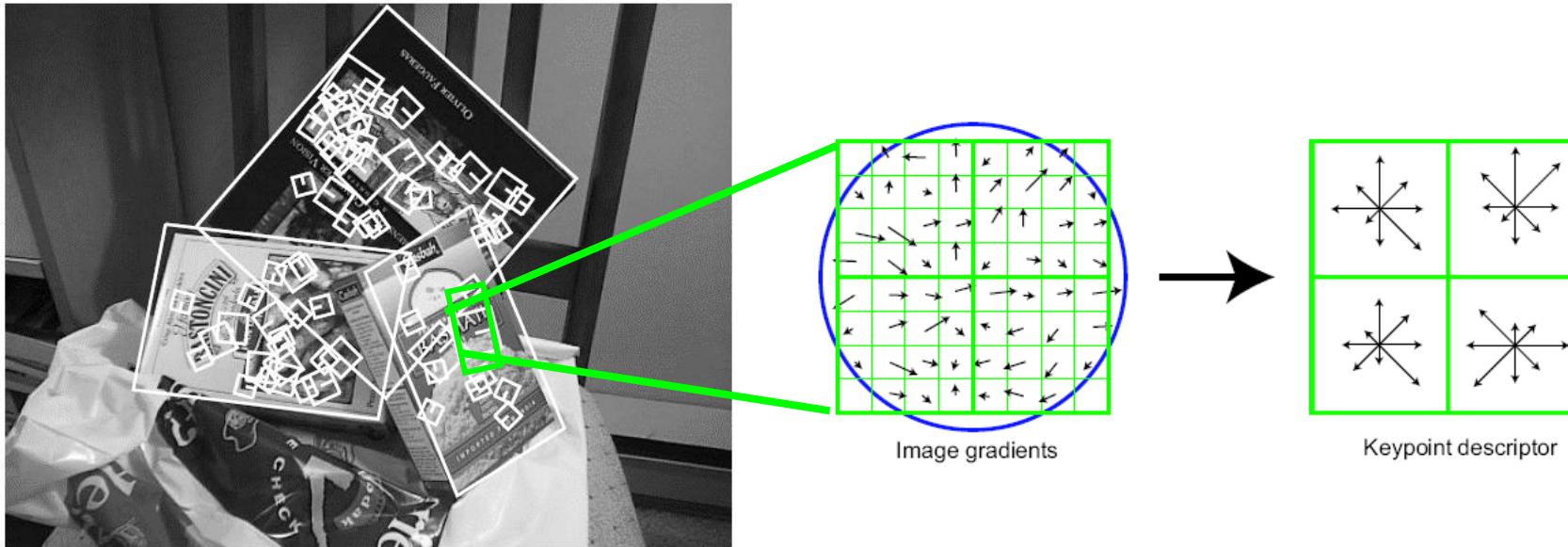
SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Local Descriptors: SIFT Descriptor



Histogram of oriented gradients

- Captures important texture information
- Robust to small translations / affine deformations

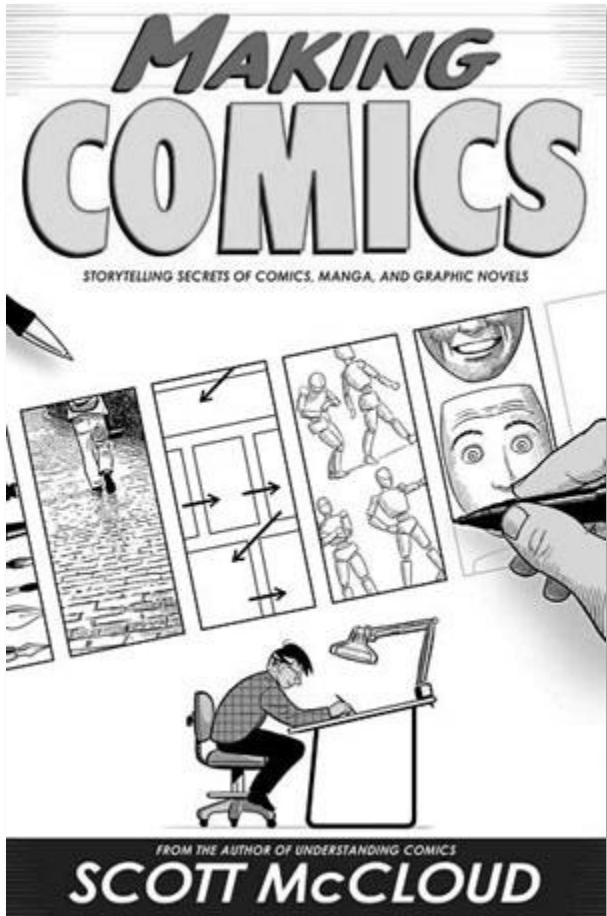
Details of Lowe's SIFT algorithm

- Run DoG detector
 - Find maxima in location/scale space
 - Remove edge points
- Find all major orientations
 - Bin orientations into 36 bin histogram
 - Weight by gradient magnitude
 - Weight by distance to center (Gaussian-weighted mean)
 - Return orientations within 0.8 of peak
 - Use parabola for better orientation fit
- For each (x,y,scale,orientation), create descriptor:
 - Sample 16x16 gradient mag. and rel. orientation
 - Bin 4x4 samples into 4x4 histograms
 - Threshold values to max of 0.2, divide by L2 norm
 - Final descriptor: 4x4x8 normalized histograms

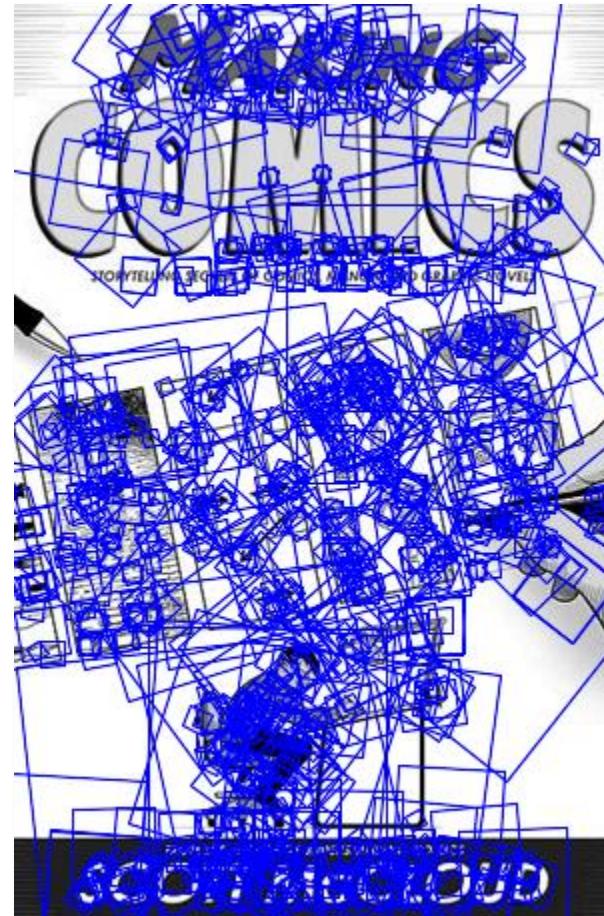
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

SIFT Example



sift



868 SIFT features

Feature matching

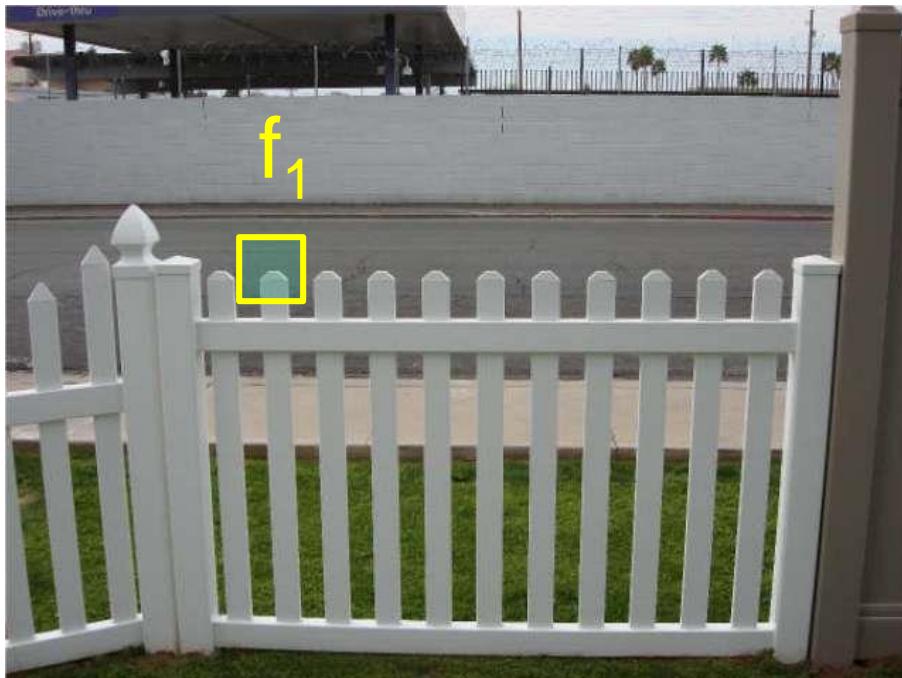
Given a feature in I_1 , how to find the best match in I_2 ?

1. Define distance function that compares two descriptors
2. Test all the features in I_2 , find the one with min distance

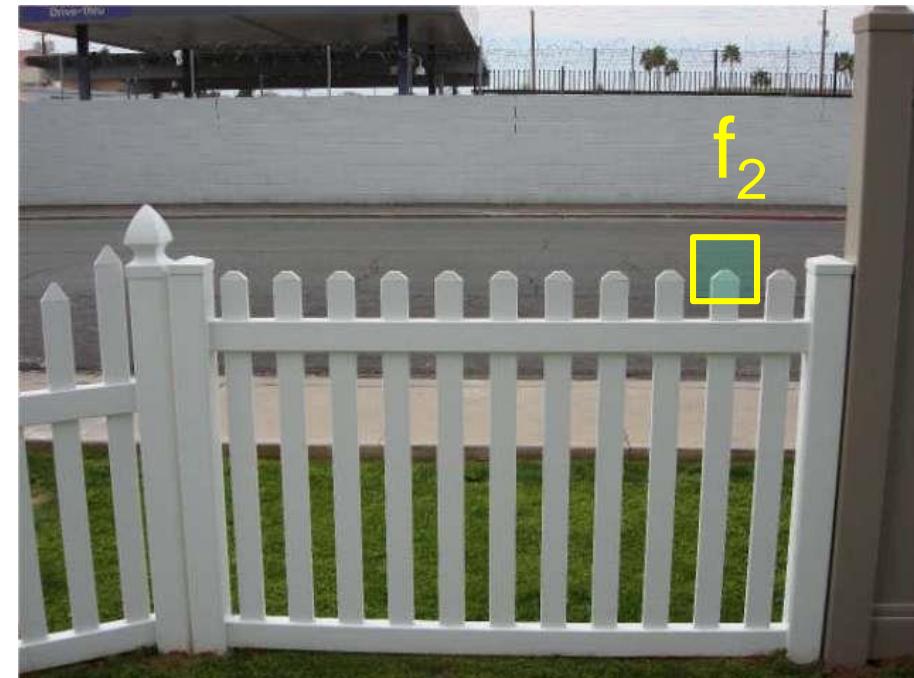
Feature distance

How to define the difference between two features f_1, f_2 ?

- Simple approach: L_2 distance, $\|f_1 - f_2\|$
- can give good scores to ambiguous (incorrect) matches



I_1

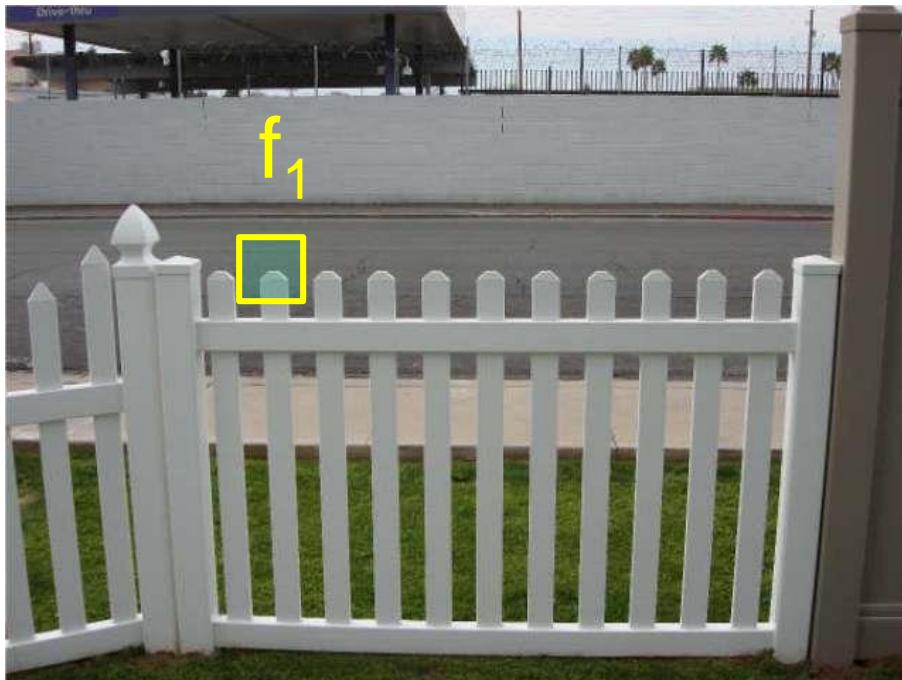


I_2

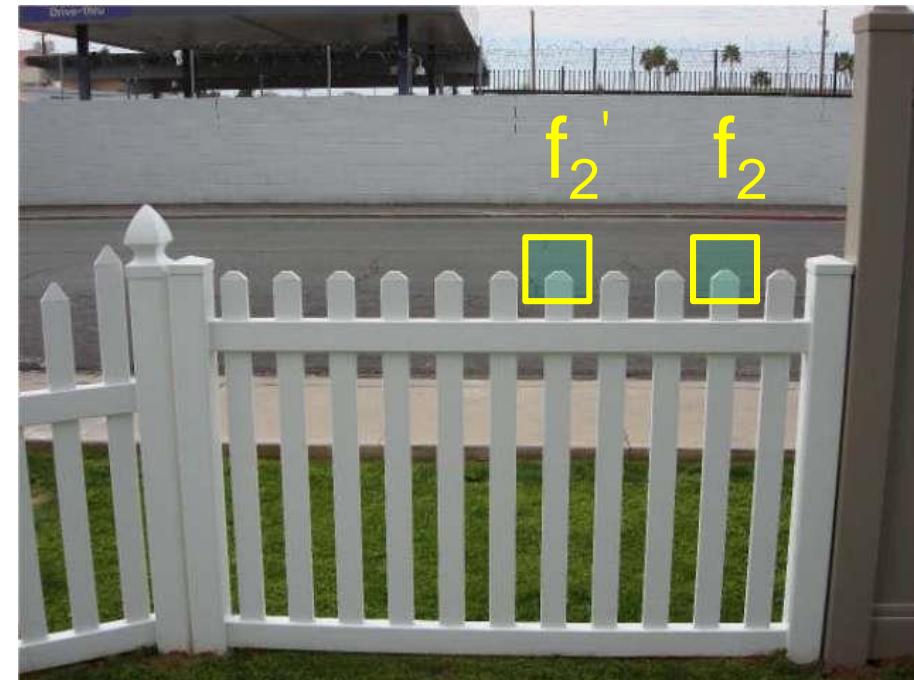
Feature distance

How to define the difference between two features f_1, f_2 ?

- Better approach: ratio distance = $\|f_1 - f_2\| / \|f_1 - f_2'\|$
 - f_2 is best SSD match to f_1 in I_2
 - f_2' is 2nd best SSD match to f_1 in I_2
 - gives large values for ambiguous matches

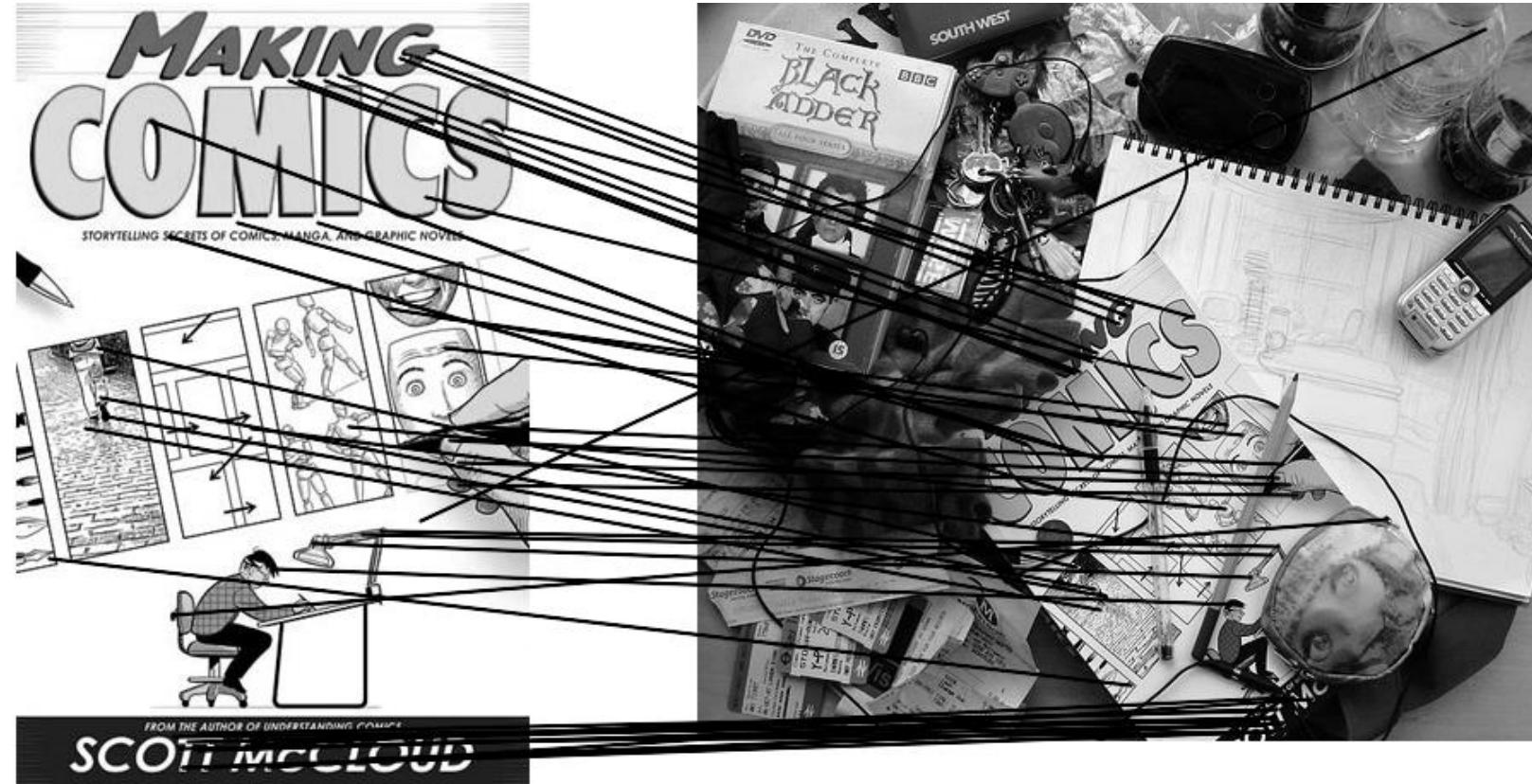


I_1



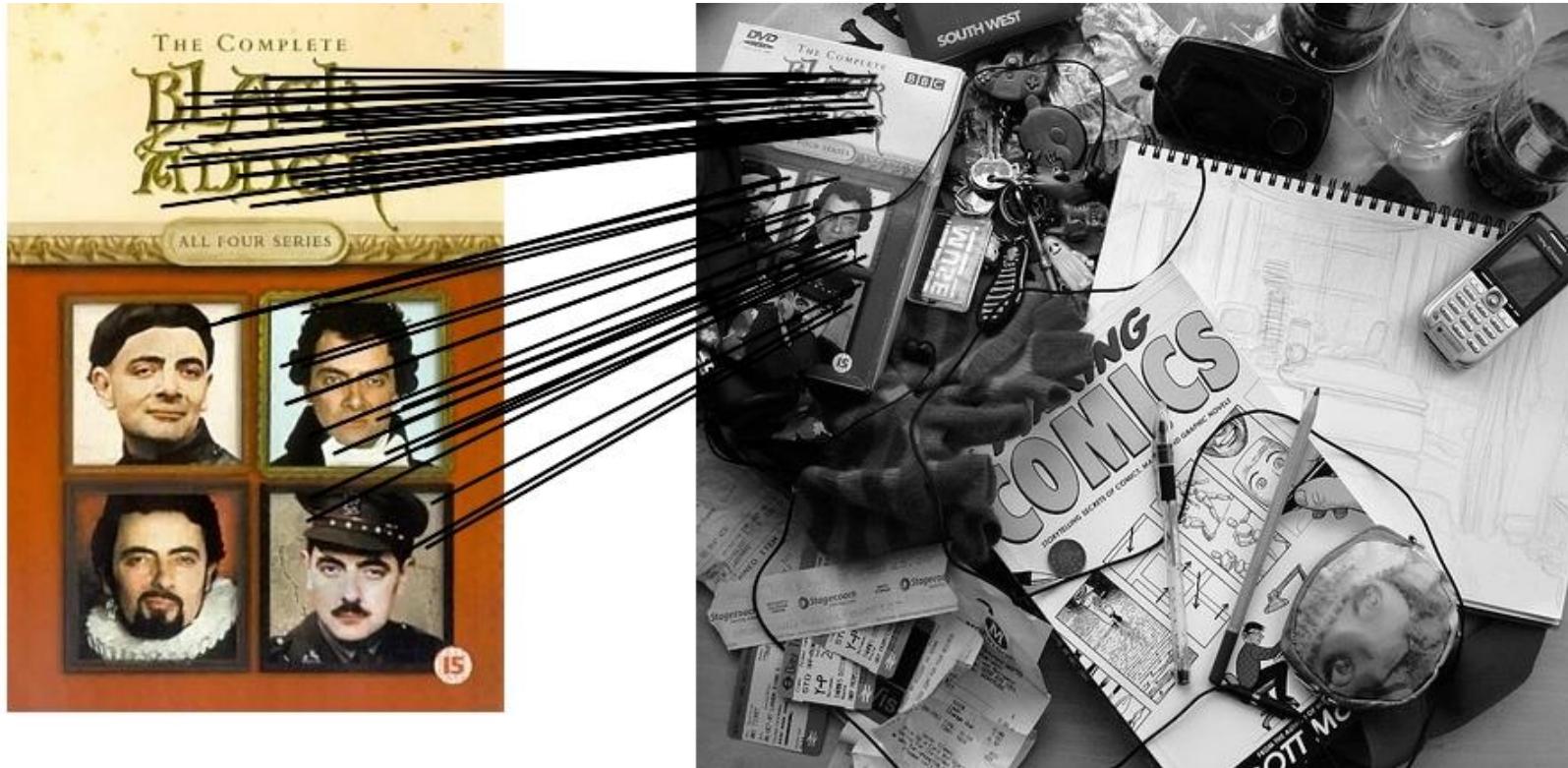
I_2

Feature matching example



51 matches

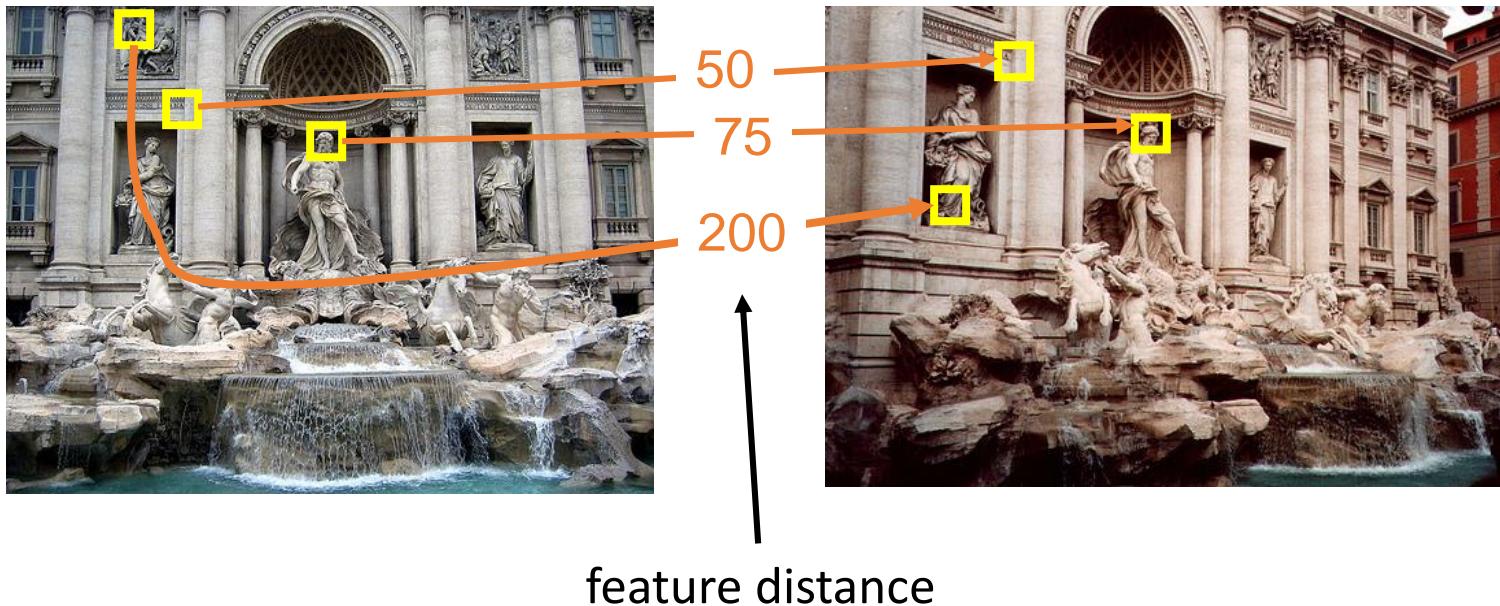
Feature matching example



58 matches

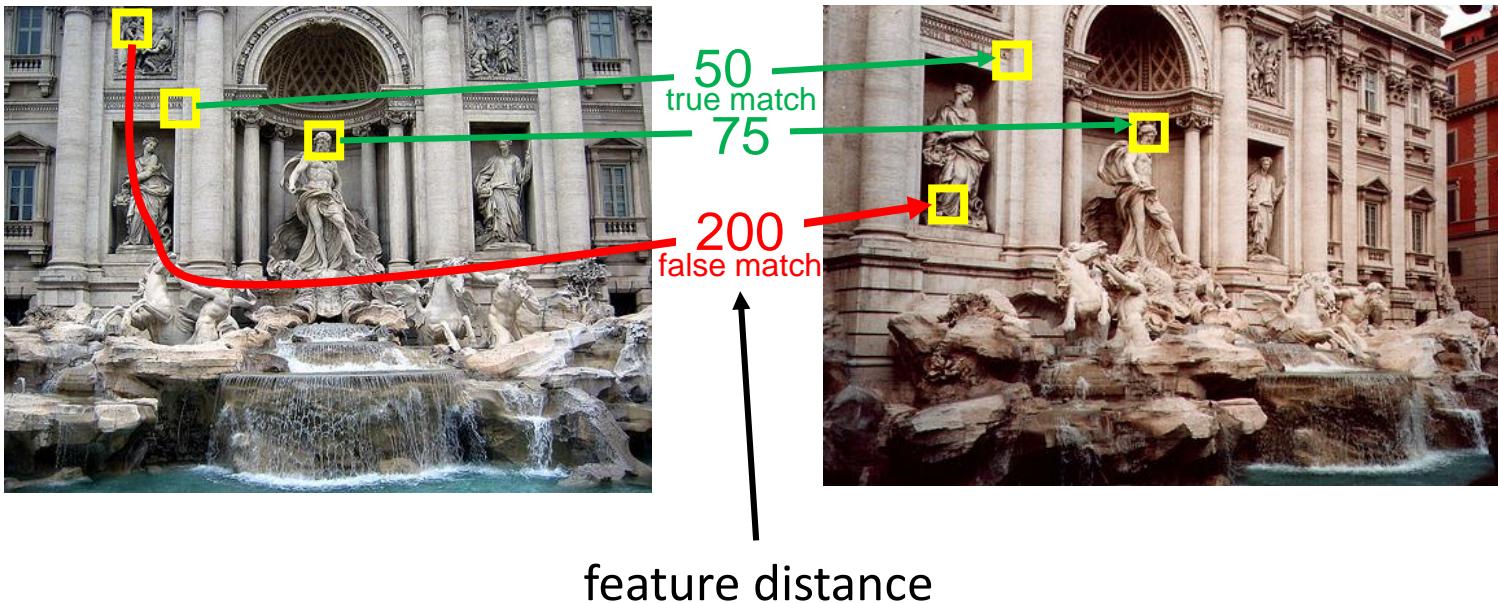
Evaluating the results

How can we measure the performance of a feature matcher?



True/false positives

How can we measure the performance of a feature matcher?



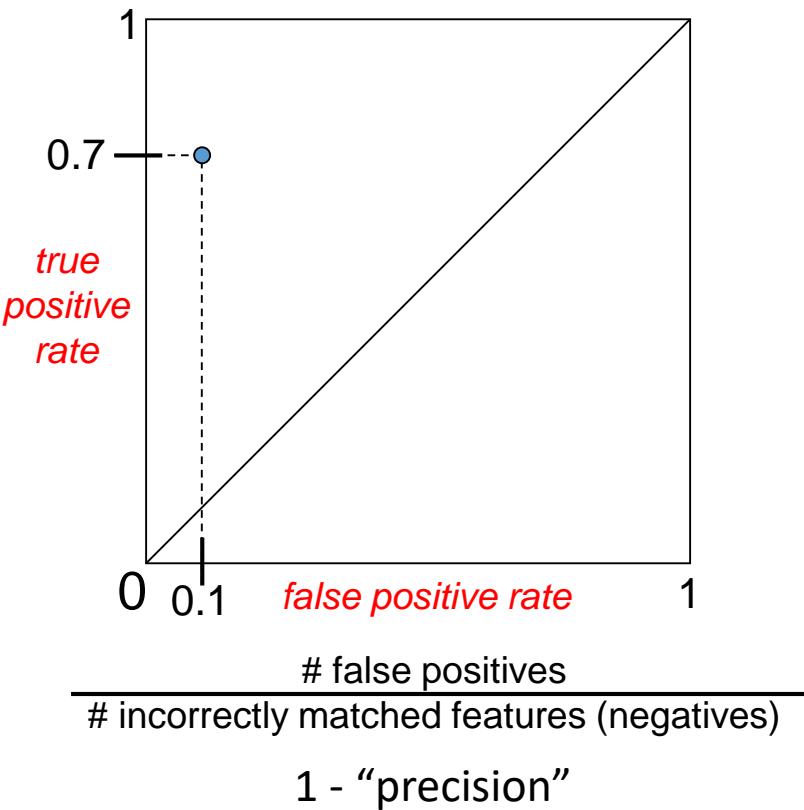
The distance threshold affects performance

- True positives = # of detected matches that are correct
 - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
 - Suppose we want to minimize these—how to choose threshold?

Evaluating the results

How can we measure the performance of a feature matcher?

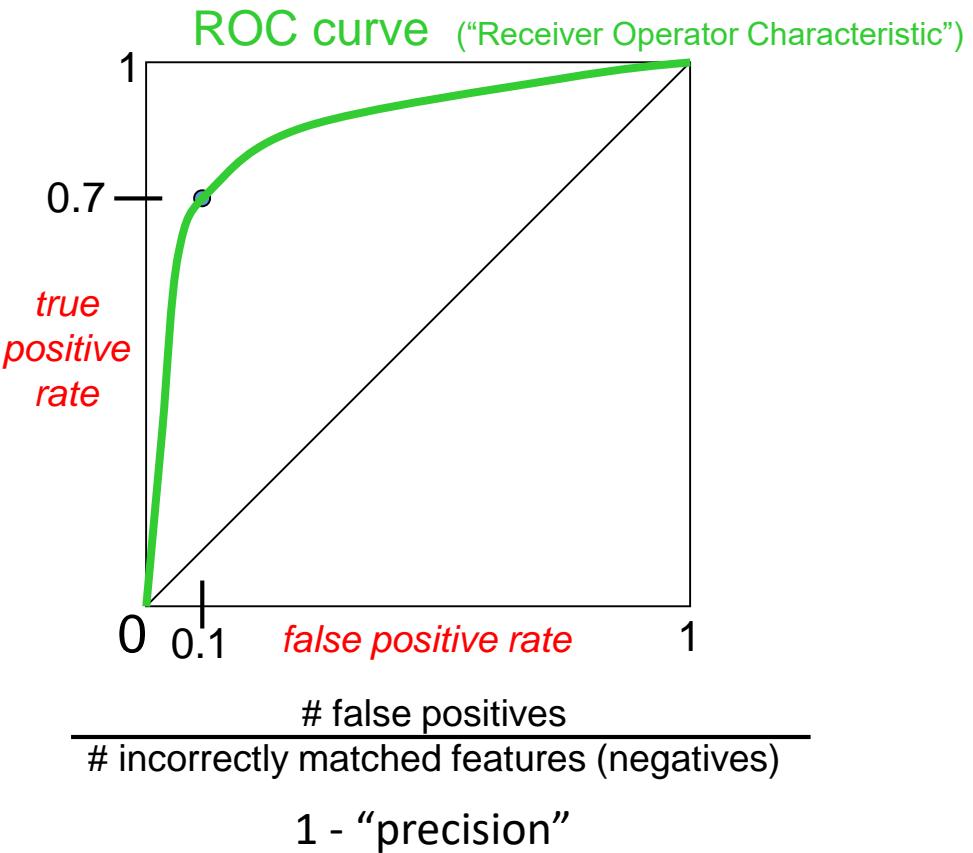
$$\frac{\# \text{ true positives}}{\# \text{ correctly matched features (positives)}} \quad \text{"recall"}$$



Evaluating the results

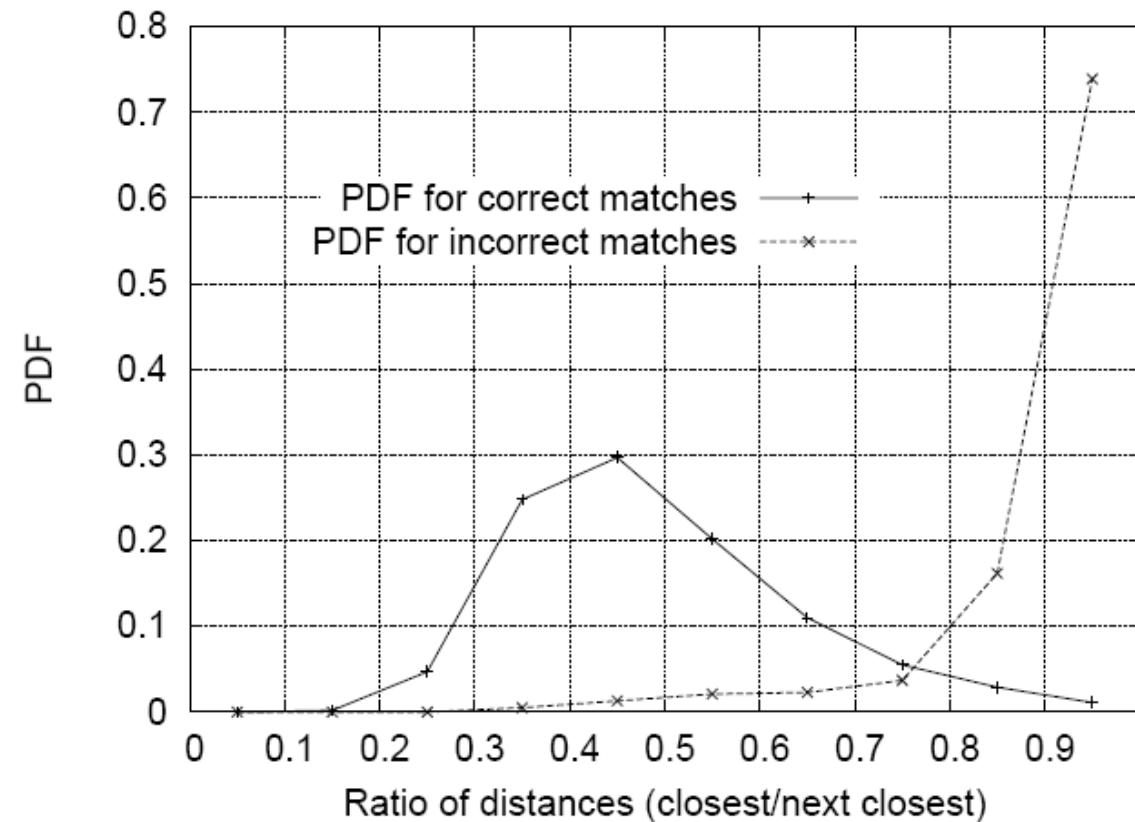
How can we measure the performance of a feature matcher?

$$\frac{\# \text{ true positives}}{\# \text{ correctly matched features (positives)}} \quad \text{"recall"}$$

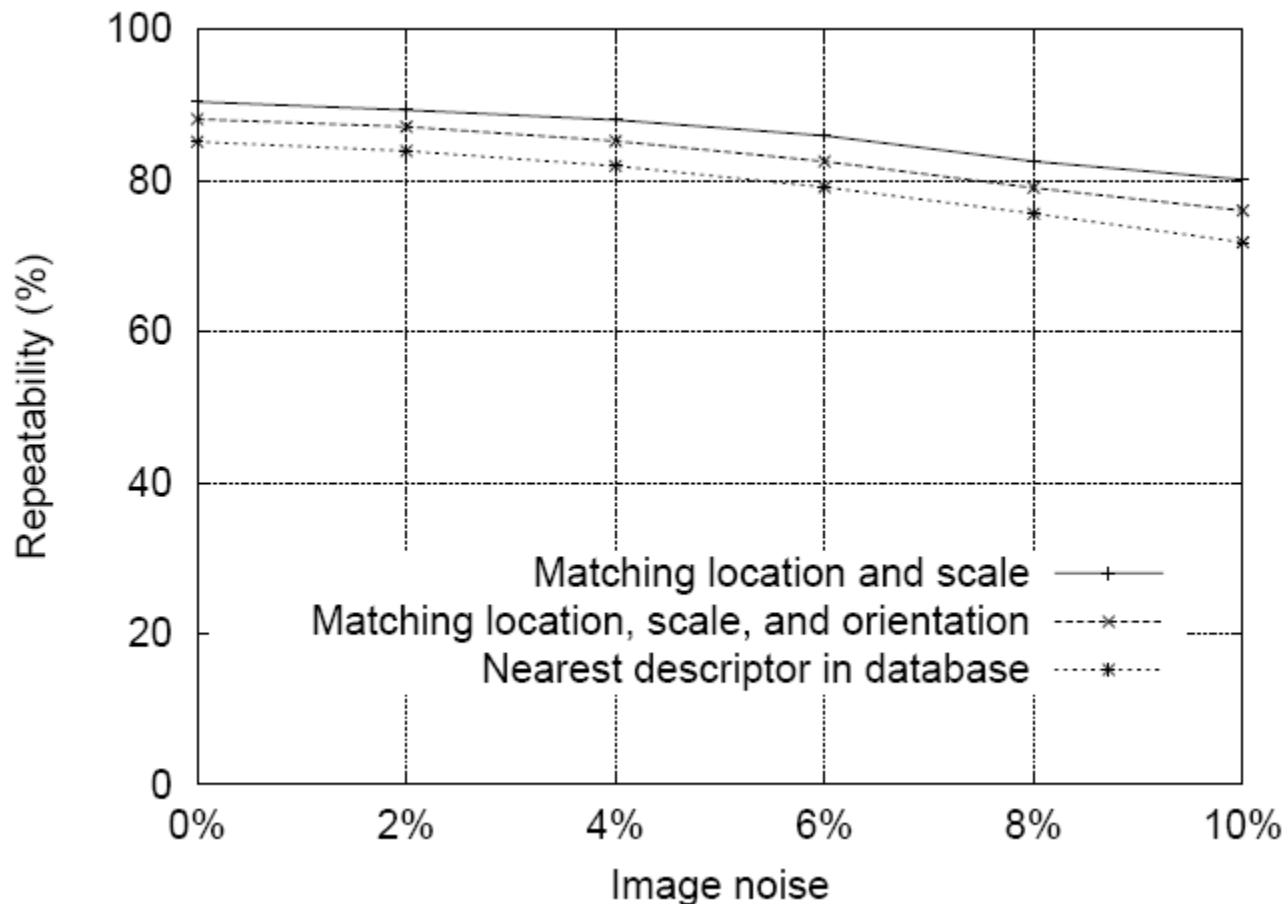


Matching SIFT Descriptors

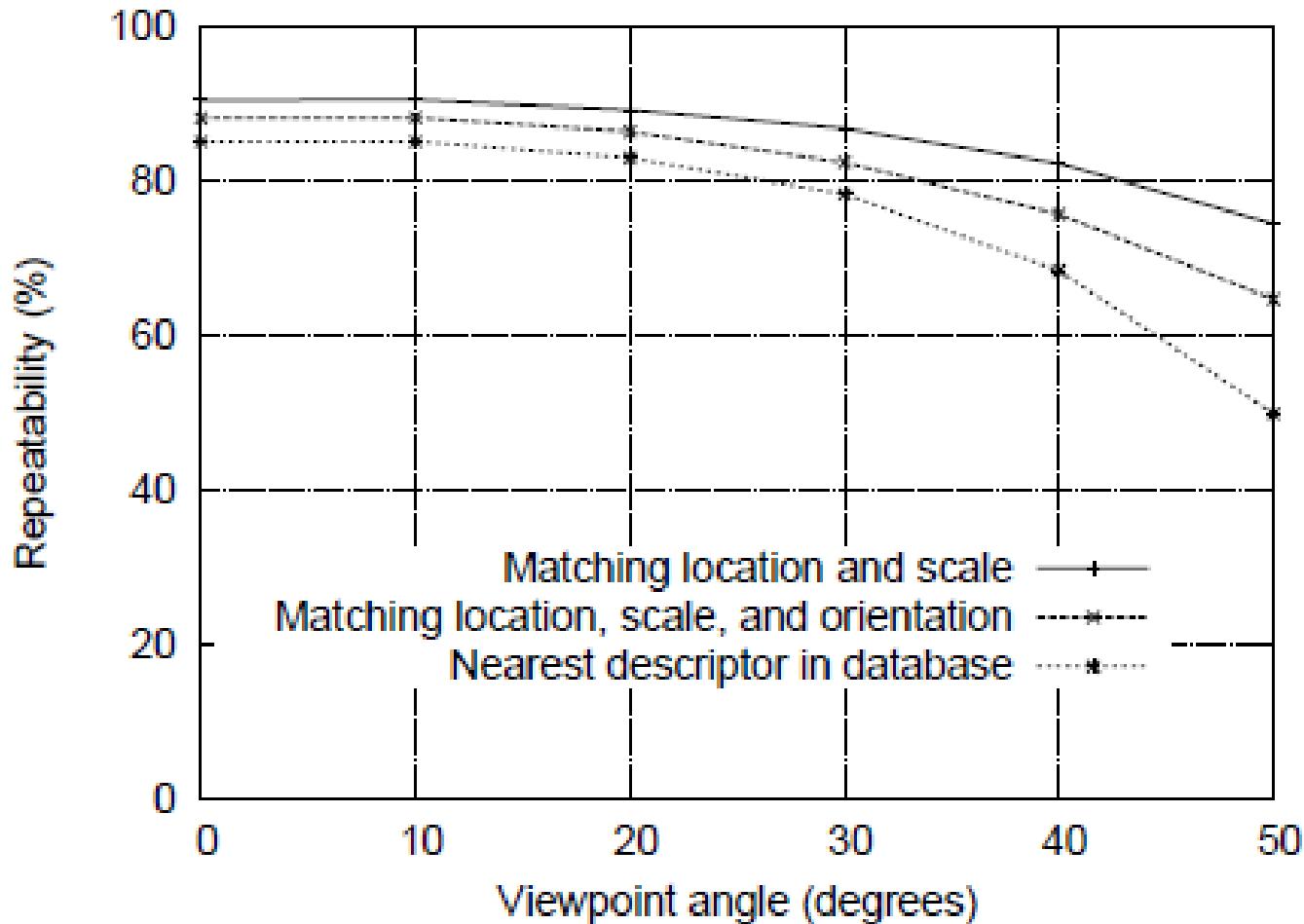
- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor



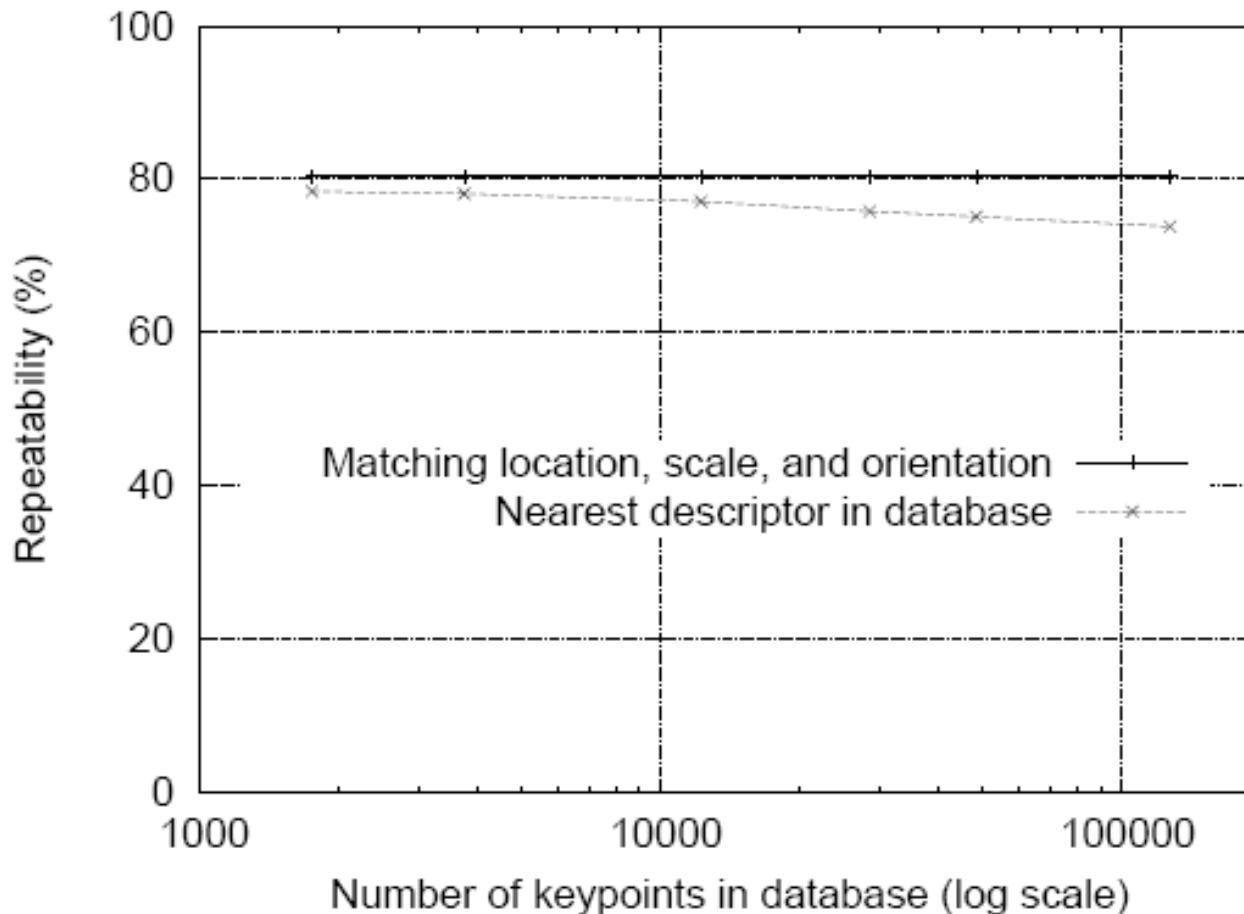
SIFT Repeatability



SIFT Repeatability



SIFT Repeatability



Matching local features



Matching local features

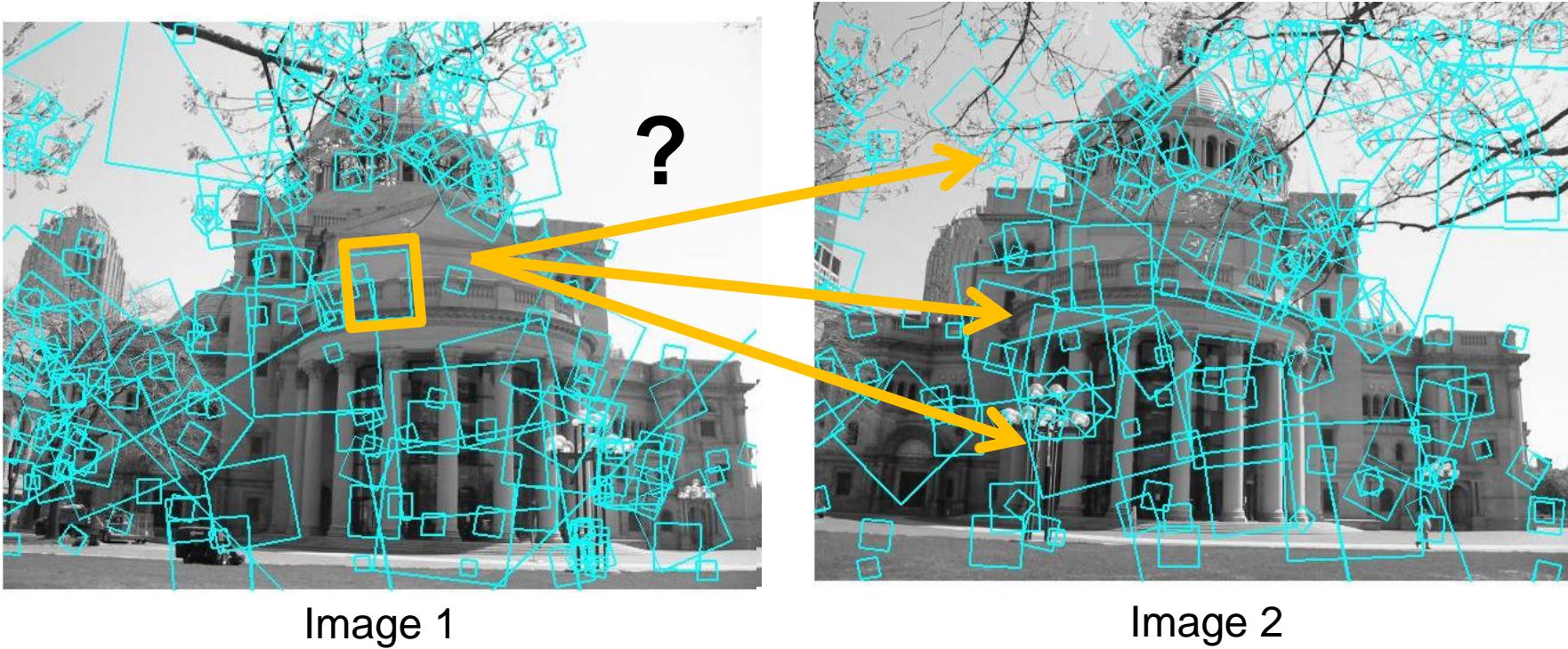


Image 1

Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Matching local features



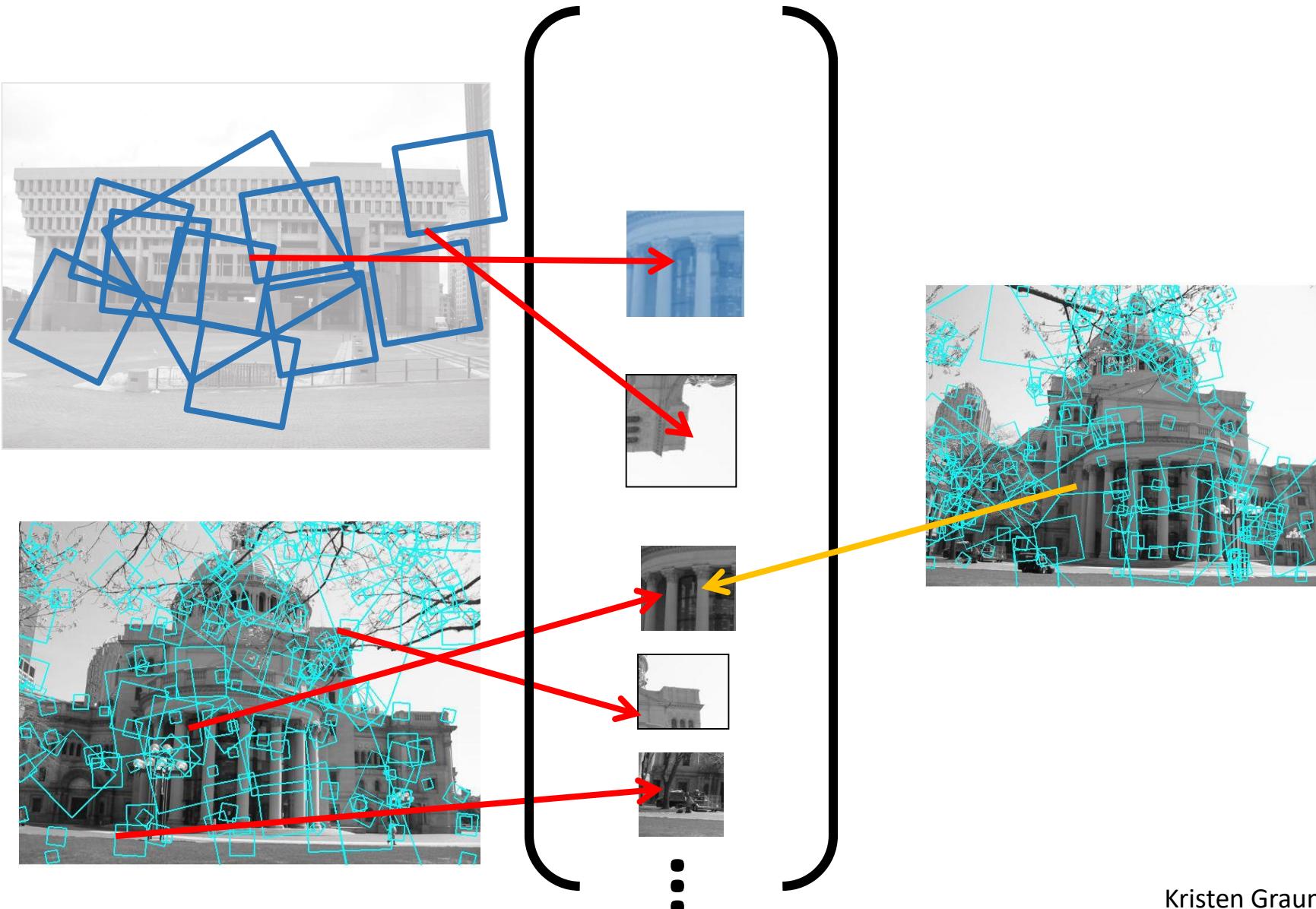
Image 1



Image 2

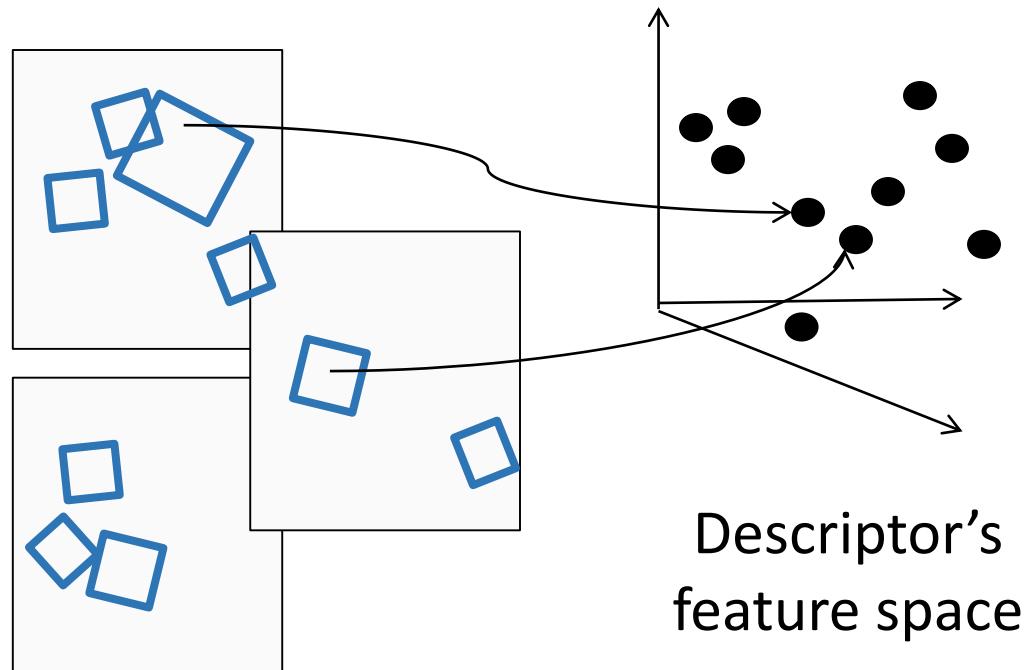
In stereo case, may constrain by proximity if we make assumptions on max disparities.

Indexing local features



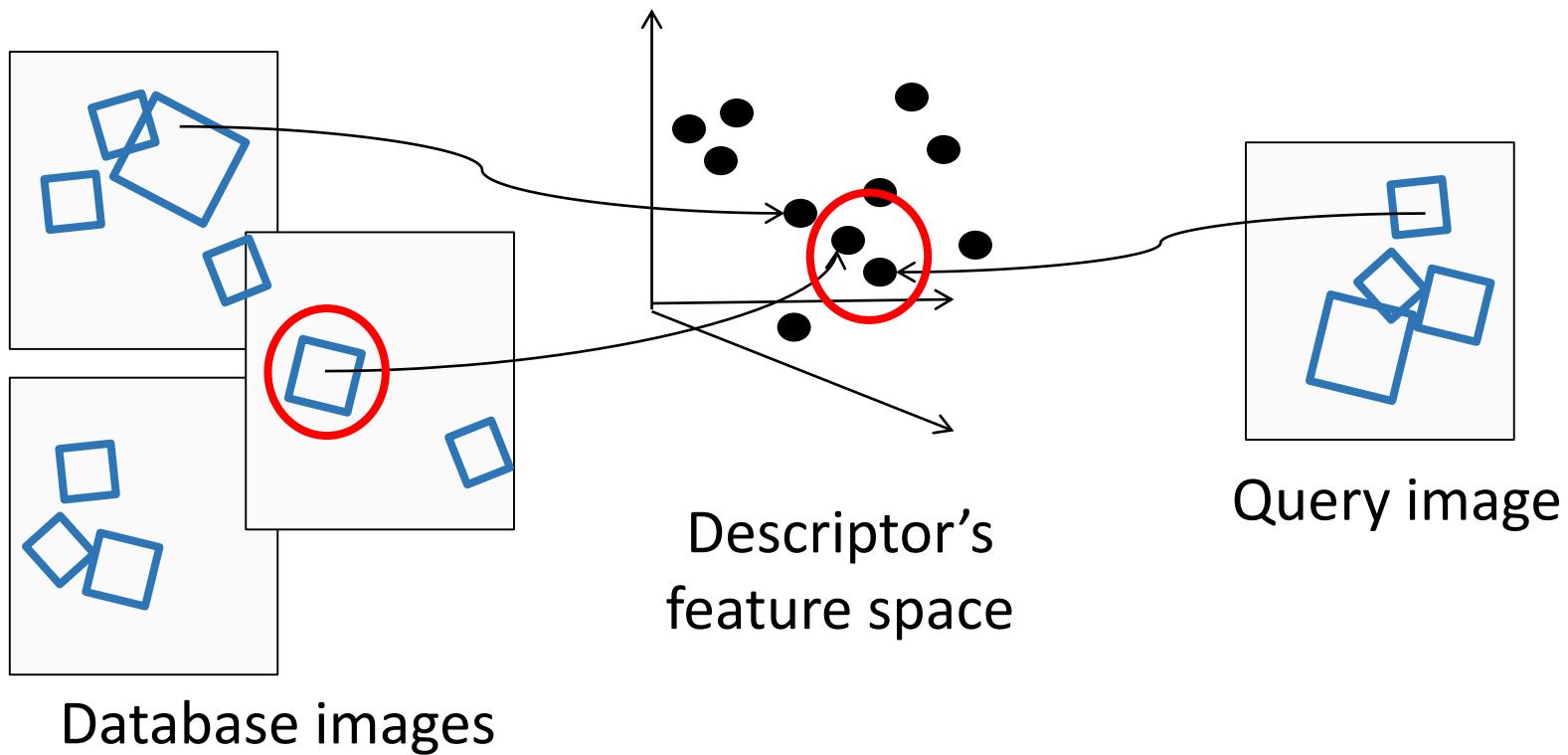
Indexing local features

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



Indexing local features

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



Indexing local features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

Indexing local features: inverted file index

Index
"Along I-75," From Detroit to Florida; <i>inside back cover</i> "Drive I-95," From Boston to Florida; <i>inside back cover</i> 1929 Spanish Trail Roadway; 101-102,104 511 Traffic Information; 83 A1A (Barrier Isl) - I-95 Access; 86 AAA (and CAA); 83 AAA National Office; 88 Abbreviations, Colored 25 mile Maps; cover Exit Services; 196 Travelogue; 85 Africa; 177 Agricultural Inspection Stns; 126 Ah-Tah-Thi-Ki Museum; 160 Air Conditioning, First; 112 Alabama; 124 Alachua; 132 County; 131 Alafia River; 143 Alapaha, Name; 126 Alfred B Maclay Gardens; 106 Alligator Alley; 154-155 Alligator Farm, St Augustine; 169 Alligator Hole (definition); 157 Alligator, Buddy; 155 Alligators; 100,135,138,147,156 Anastasia Island; 170 Anhinga; 109-109,146 Apalachicola River; 112 Appleton Mus of Art; 136 Aquifer; 102 Arabian Nights; 94 Art Museum, Ringling; 147 Aruba Beach Cafe; 183 Aucilla River Project; 106 Babcock-Web WMA; 151 Bahia Mar Marina; 184 Baker County; 99 Barefoot Mallmen; 182 Barge Canal; 137 Bee Line Expy; 80 Belz Outlet Mall; 89 Bernard Castro; 136 Big "I"; 165 Big Cypress; 155,158 Big Foot Monster; 105 Billie Swamp Safari; 160 Blackwater River SP; 117 Blue Angels

Butterfly Center, McGuire; 134 CAA (see AAA) CCC, The; 111,113,115,135,142 Ca d'Zan; 147 Caloosahatchee River; 152 Name; 150 Canaveral Natnl Seashore; 173 Cannon Creek Apark; 130 Canopy Road; 106,160 Cape Canaveral; 174 Castillo San Marcos; 169 Cave Diving; 131 Cayo Costa, Name; 150 Celebration; 93 Charlotte County; 149 Charlotte Harbor; 150 Chautauqua; 116 Chipley; 114 Name; 115 Choctawhatchee, Name; 115 Circus Museum, Ringling; 147 Citrus; 88,97,130,136,140,180 CityPlace, W Palm Beach; 180 City Maps, Ft Lauderdale Expyws; 194-195 Jacksonville; 163 Kissimmee Expyws; 192-193 Miami Expressways; 194-195 Orlando Expressways; 192-193 Pensacola; 26 Tallahassee; 191 Tampa-St. Petersburg; 63 St. Augustine; 191 Civil War; 100,108,127,138,141 Clearwater Marine Aquarium; 187 Collier County; 154 Collier, Barron; 152 Colonial Spanish Quarters; 168 Columbia County; 101,128 Coquina Building Material; 165 Corkscrew Swamp, Name; 154 Cowboys; 85 Crab Trap II; 144 Cracker, Florida; 88,95,132 Crosstown Expy; 11,35,98,143 Cuban Bread; 184 Dade Battlefield; 140 Dade, Maj. Francis; 139-140,161 Danis Beach Hurricane; 184 Daniel Boone, Florida Walk; 117 Daytona Beach; 172-173 De Land; 87 Ford, Henry; 152

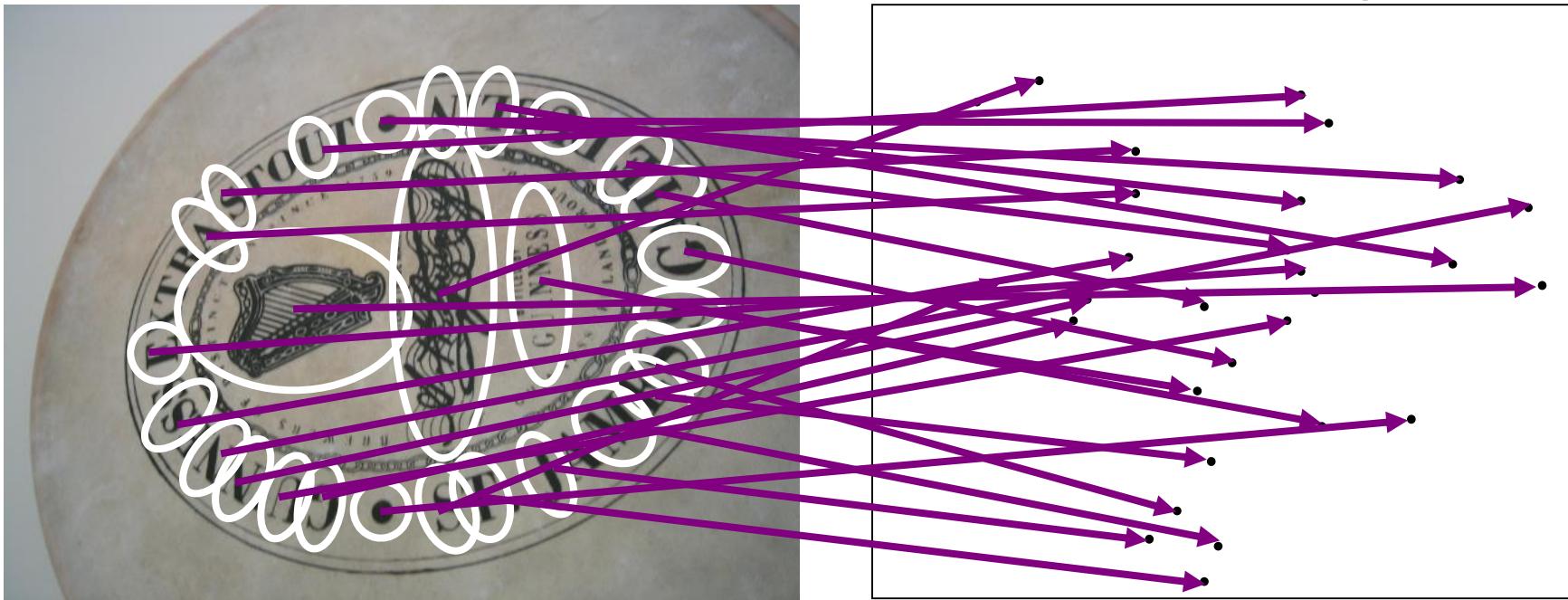
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to “visual words”.

Text retrieval vs. image search

- What makes the problems similar, different?

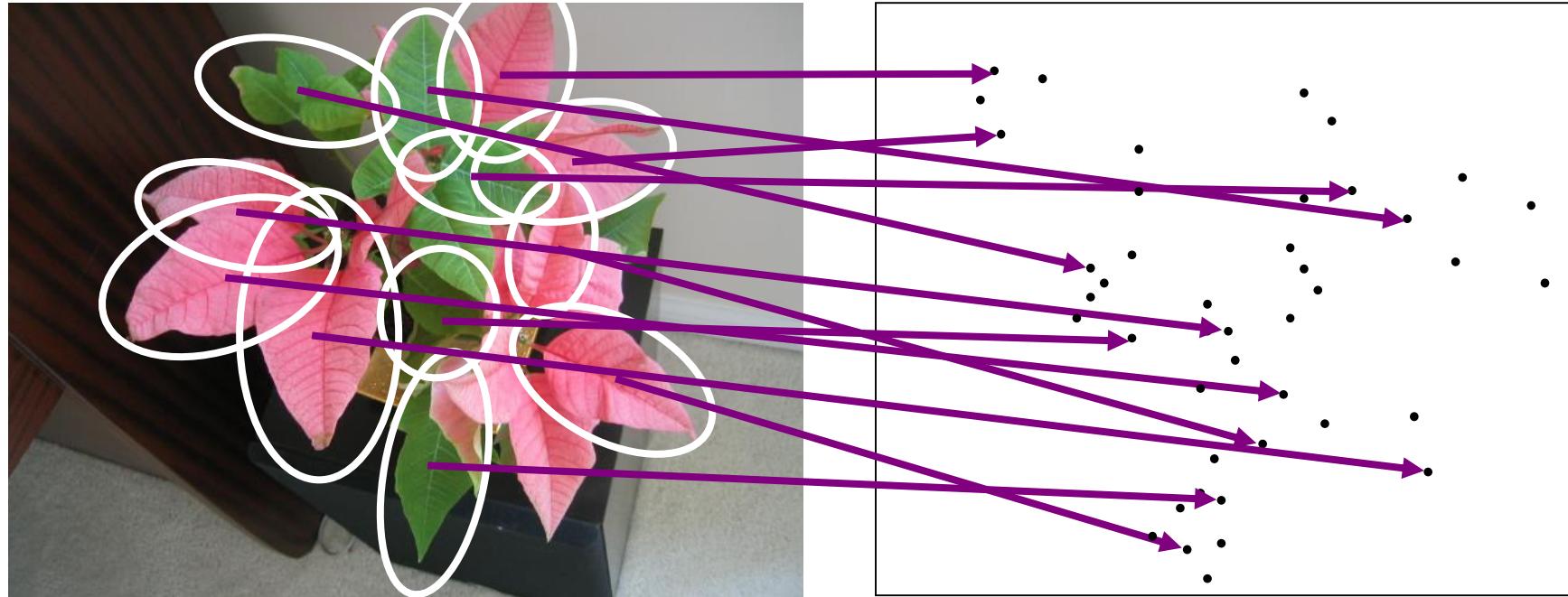
Visual words: main idea

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

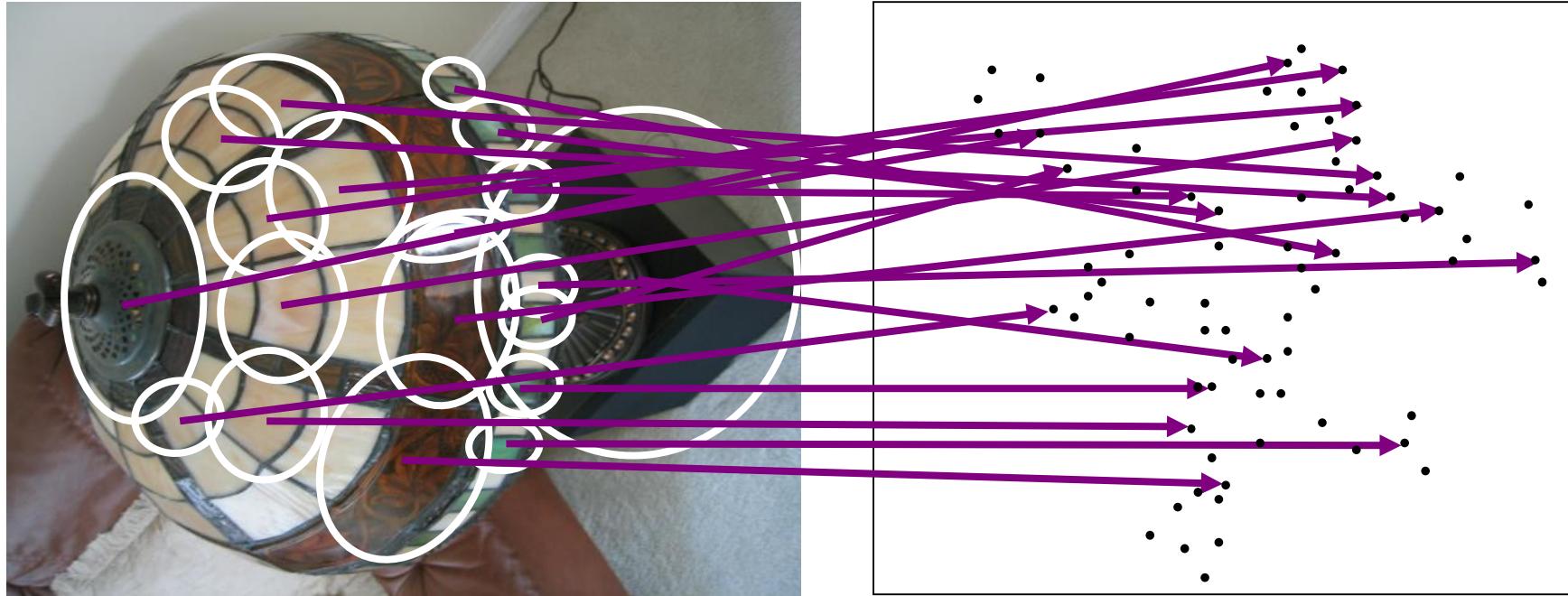


e.g., SIFT descriptor space: each point is 128-dimensional

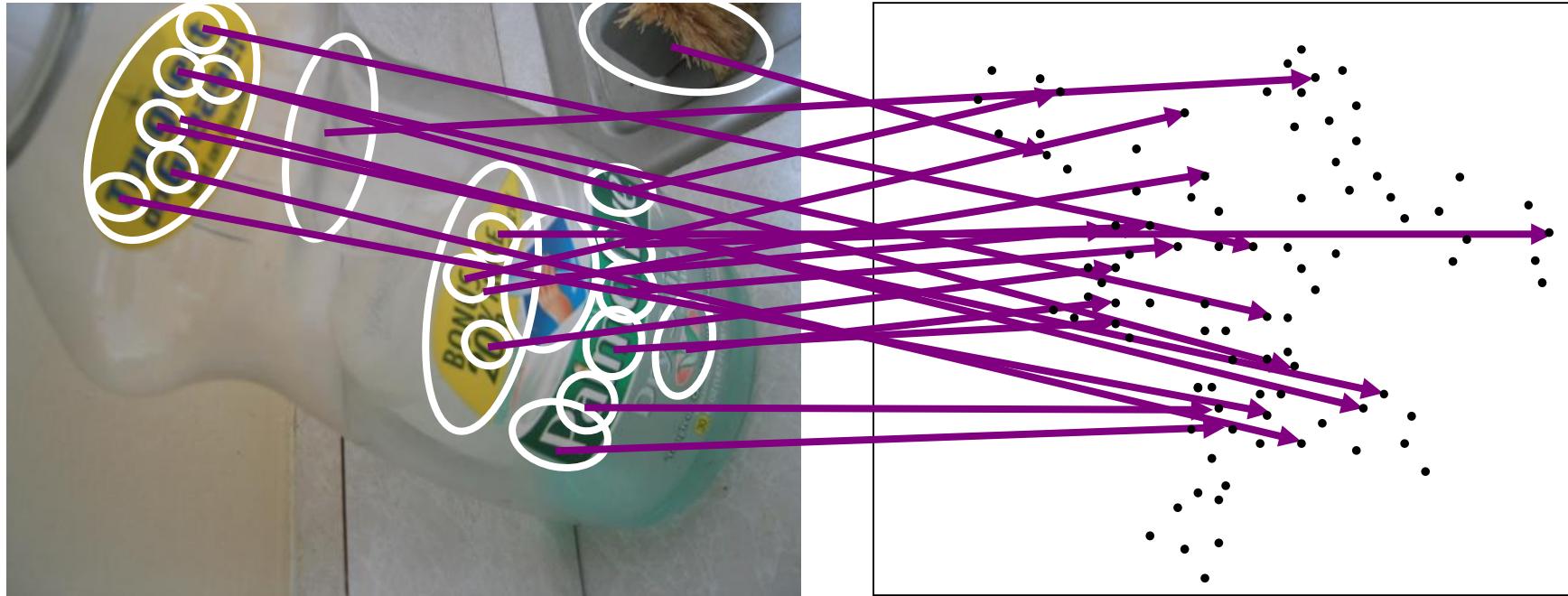
Visual words: main idea



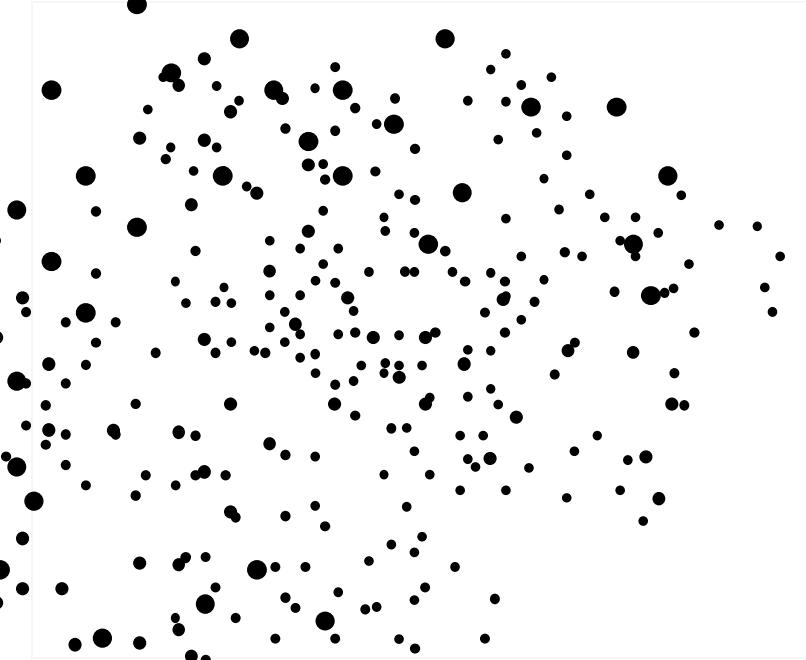
Visual words: main idea

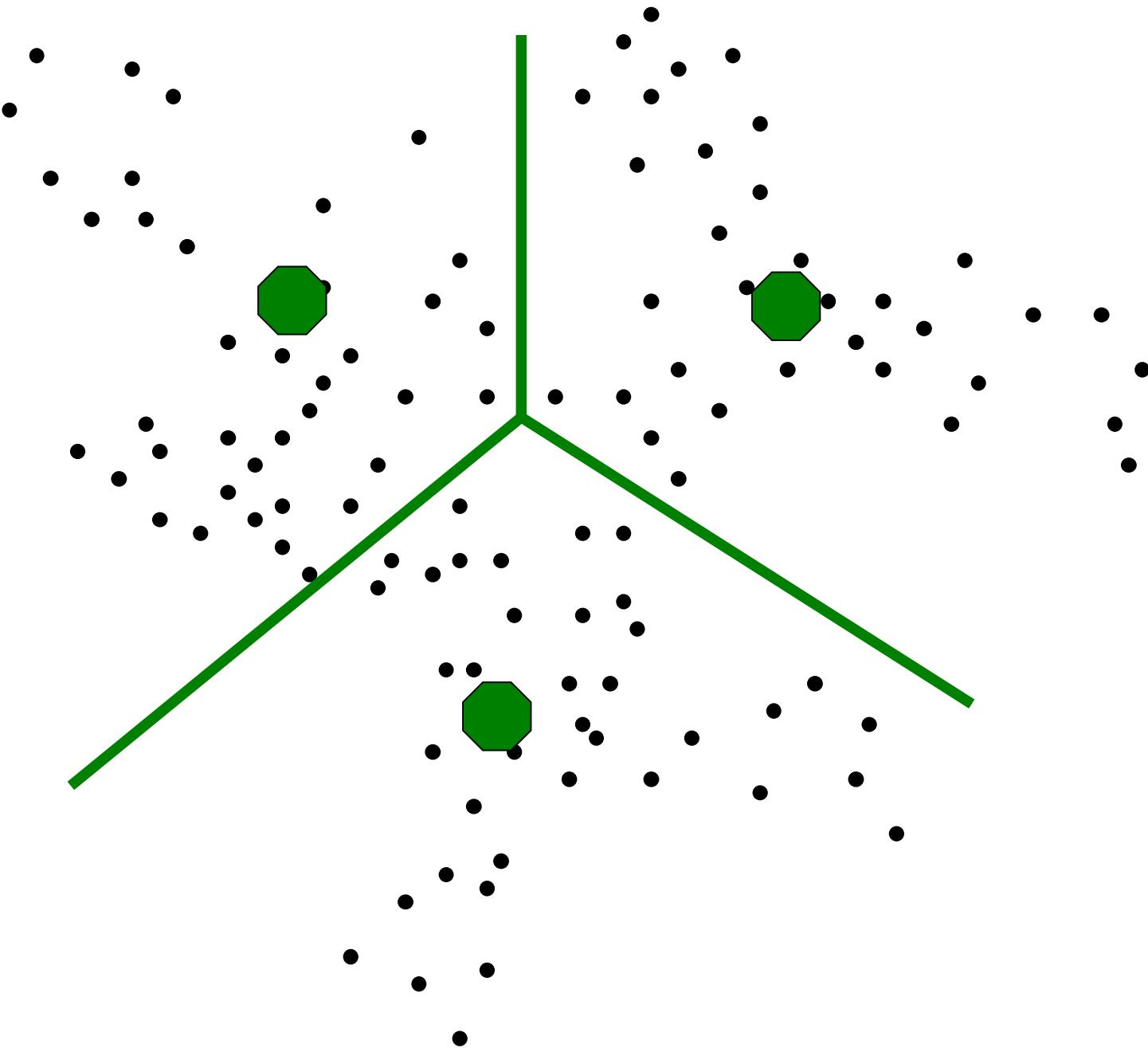


Visual words: main idea



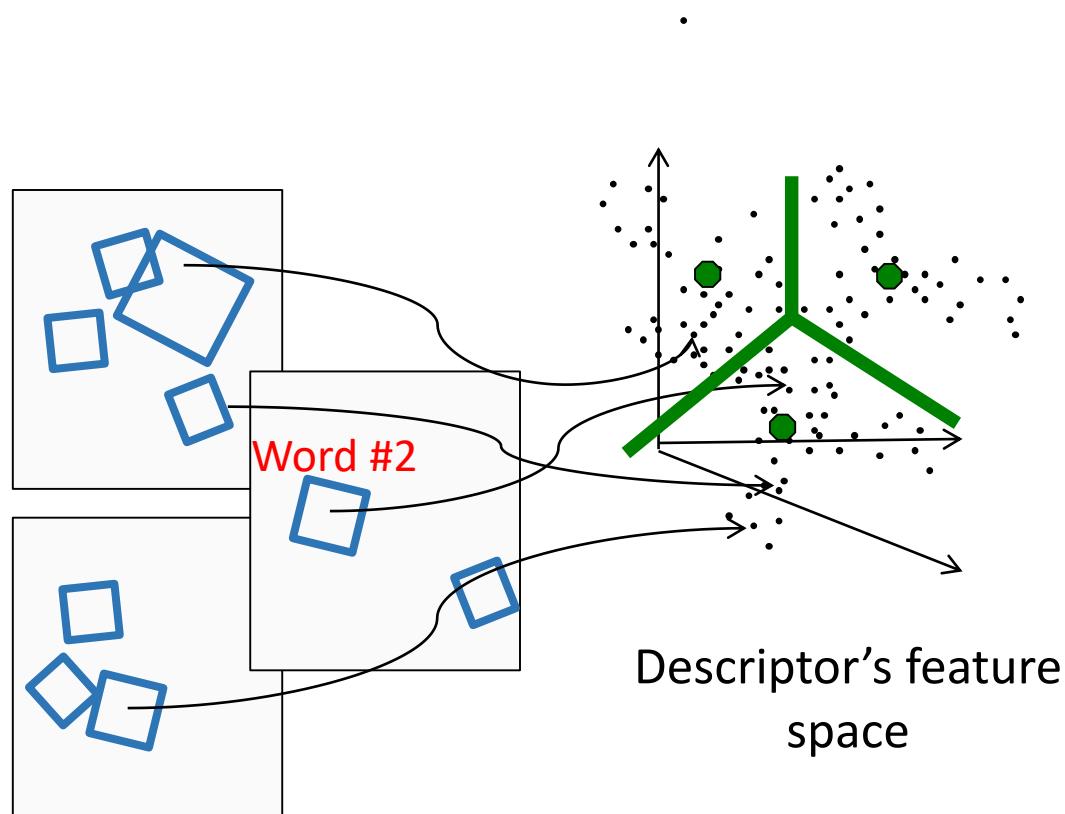
Each point is a
local descriptor,
e.g. SIFT vector.





Visual words

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

Visual words

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

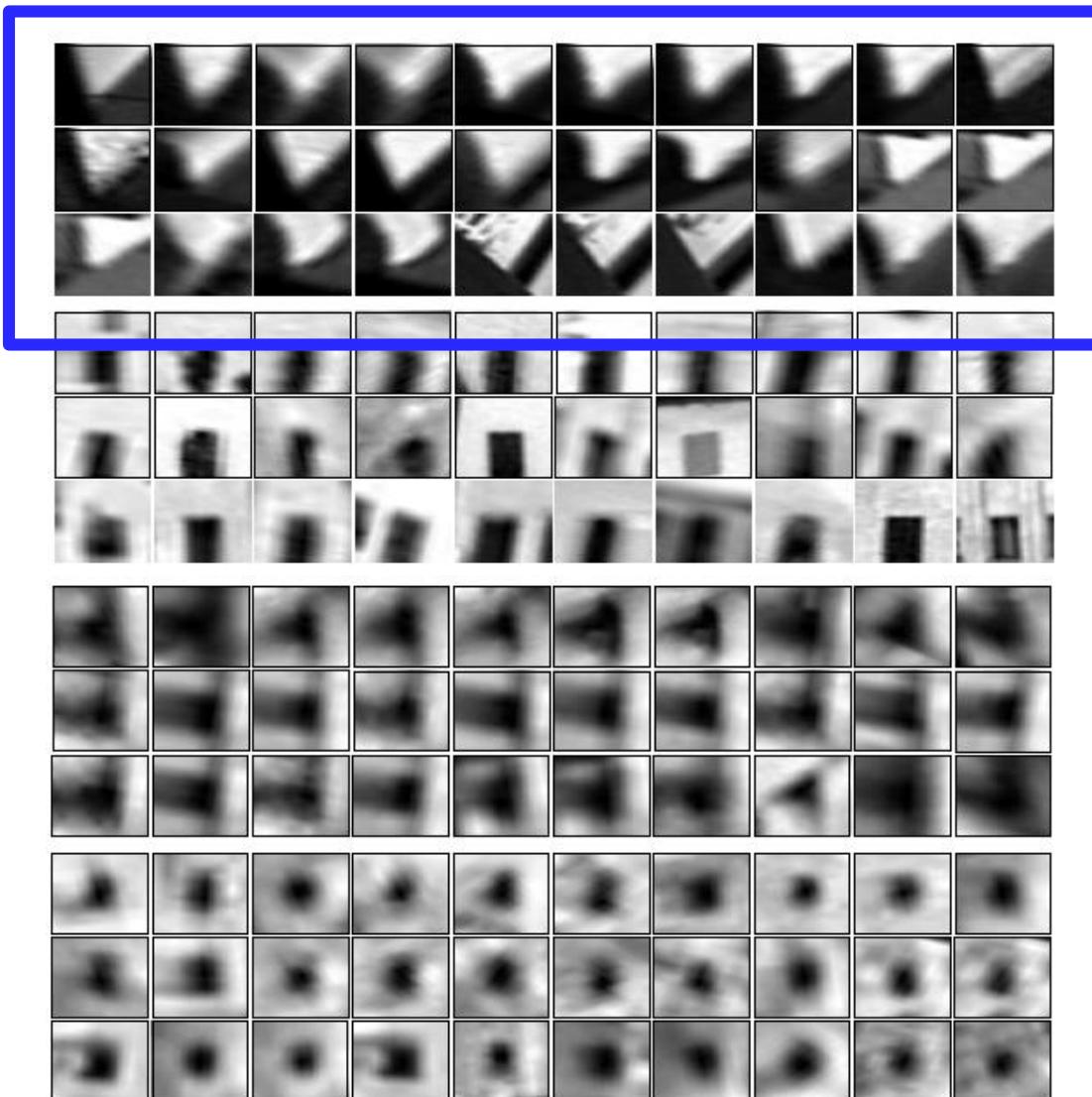
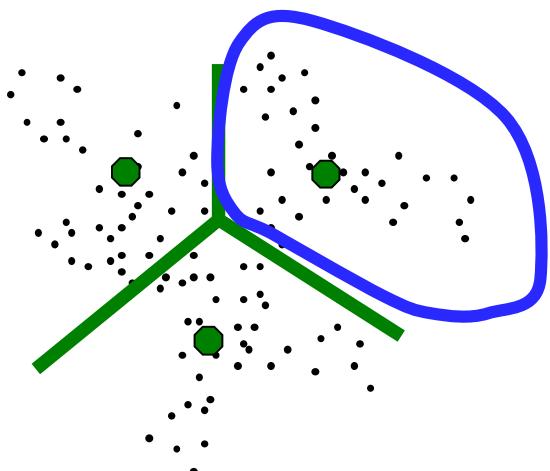
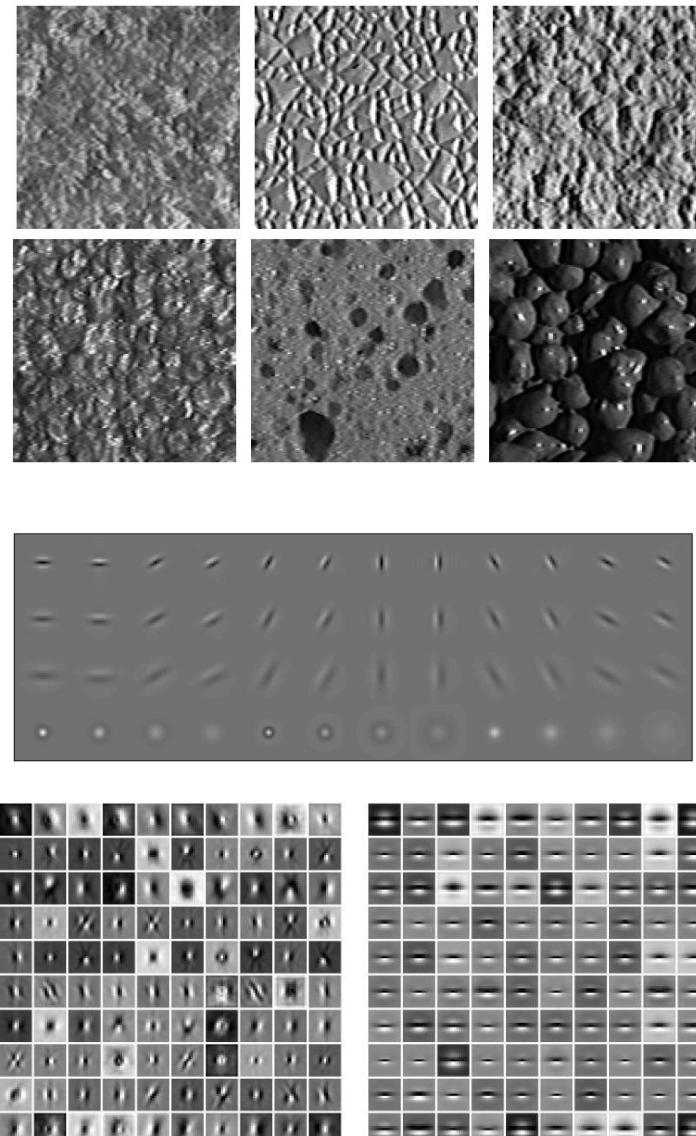


Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman

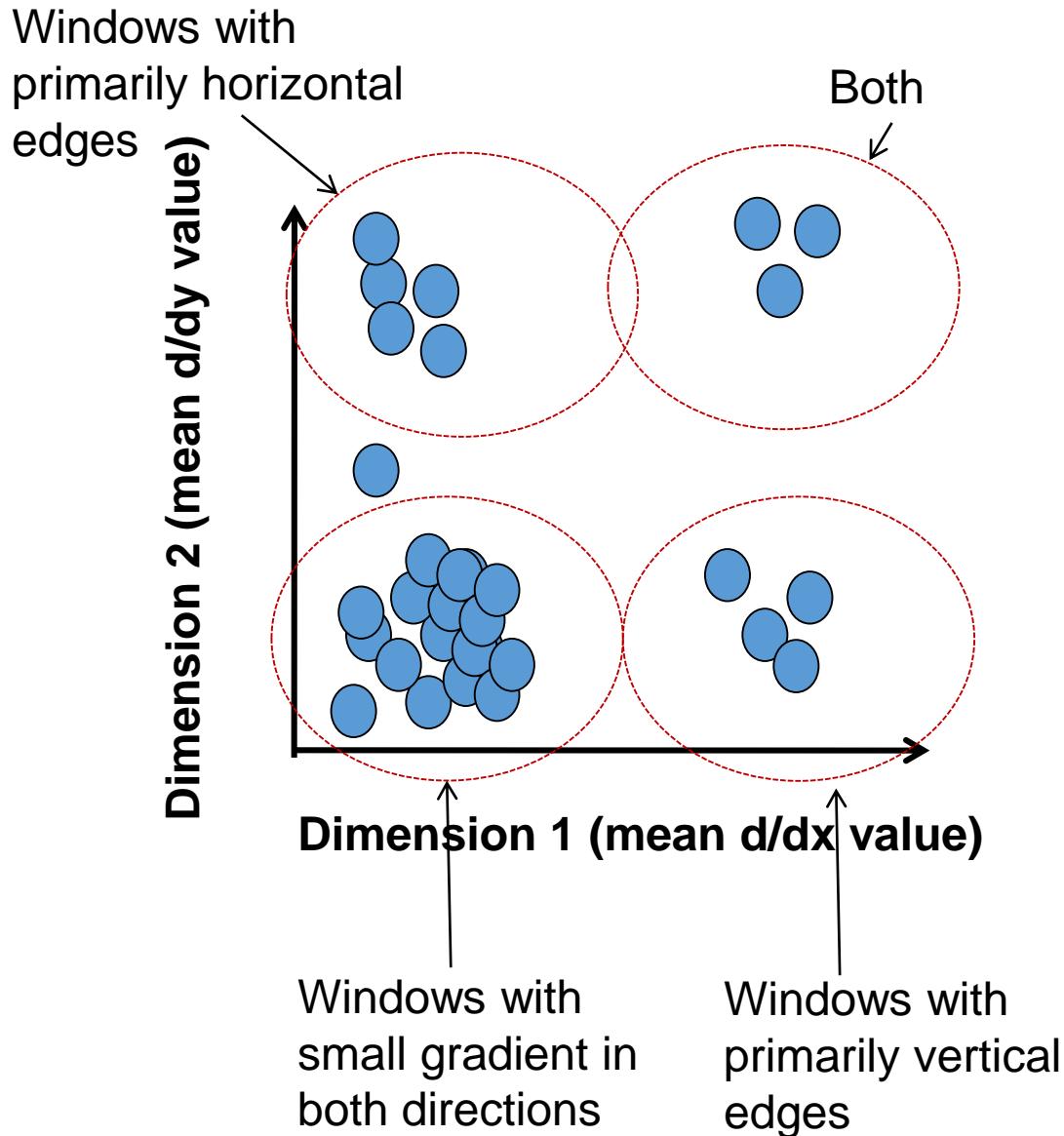
Visual words and textons

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.



Leung & Malik 1999; Varma &
Zisserman, 2002

Recall: Texture representation example



	<u>mean d/dx value</u>	<u>mean d/dy value</u>
Win. #1	4	10
Win.#2	18	7
:	:	
Win.#9	20	20

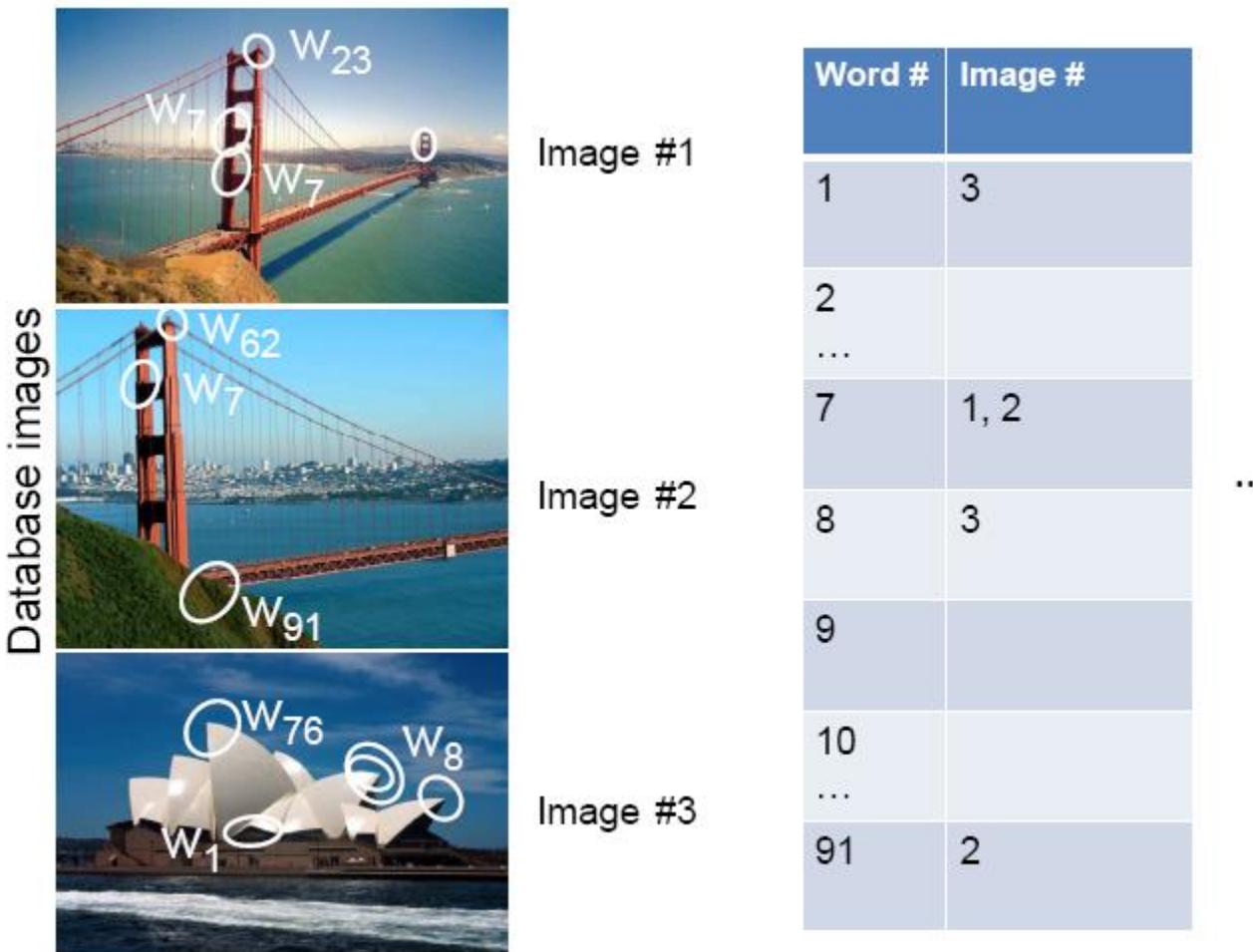
**statistics to
summarize patterns
in small windows**

Visual vocabulary formation

Issues:

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

Inverted file index



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

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.

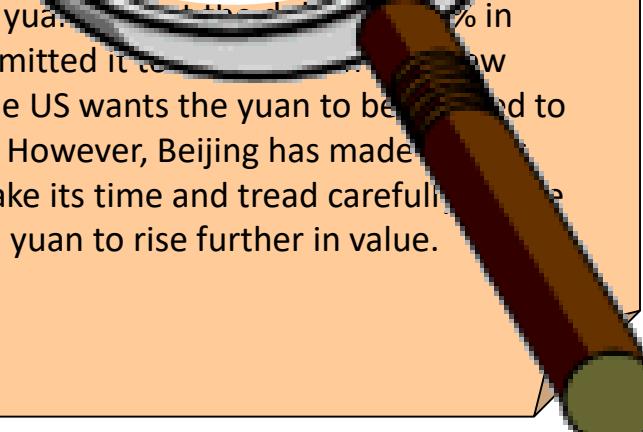
- If a local image region is a visual word, how can we summarize an image (the document)?

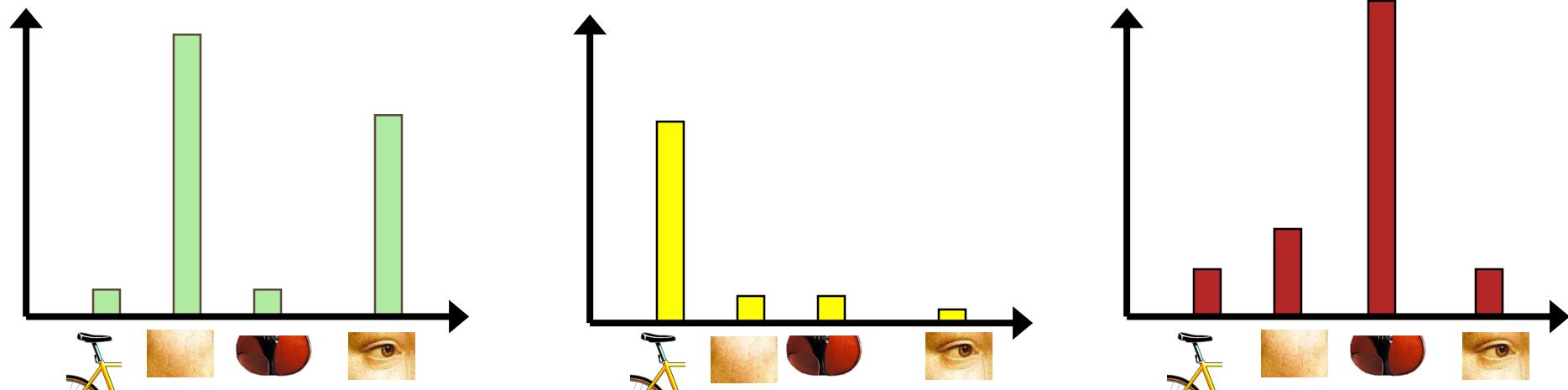
Analogy to documents

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 upon what we see. Light does not reach the brain from our eyes directly. We have thought that the optic nerve carried the point by which the eye influenced the cerebral cortex. But it is now known that upon what we see depends the way in which we perceive it. Through the work of Hubel and Wiesel, now known as the "two-dimensional theory of perception", we have learned more completely how the brain processes the visual impressions. By studying the various cell layers of the visual cortex, Hubel and Wiesel have been able to determine the message about the image falling on the retina. This message undergoes a step-wise analysis in a systematic way. The information is stored in nerve cells stored in columns. In this system, each column of nerve cells has its specific function and is responsible for extracting a specific detail in the pattern of the retinal image.

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

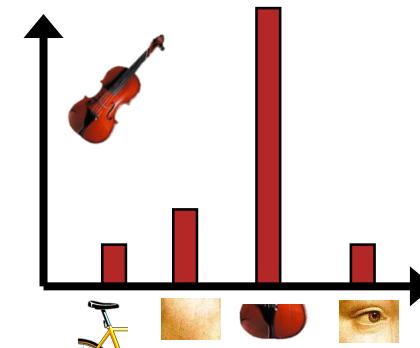
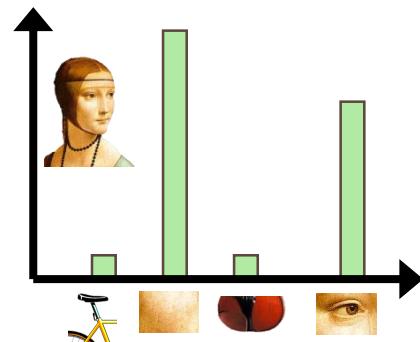
China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be driven by a predicted 30% jump in exports, with a 18% rise in imports. Further analysis showed China's deliberate policy to keep the value of the yuan, one factor in the surplus, at a low level. Xiaochuan, the central bank, has more to be done to stay within the range of the value of the yuan. It has cut the interest rate by 0.25% in July and permitted it to fluctuate more in the band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made clear that it will take its time and tread carefully, allowing the yuan to rise further in value.





Bags of visual words

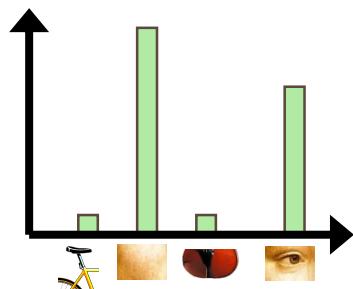
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



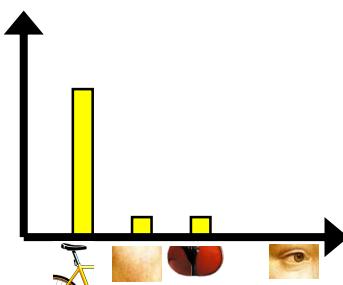
Comparing bags of words

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

[1 8 1 4]



[5 1 1 0]



\vec{d}_j

\vec{q}

Bags of words for content-based image retrieval

Visually defined query

“Find this
clock”



“Groundhog Day” [Rammis, 1993]



“Find this
place”



retrieved shots

Example



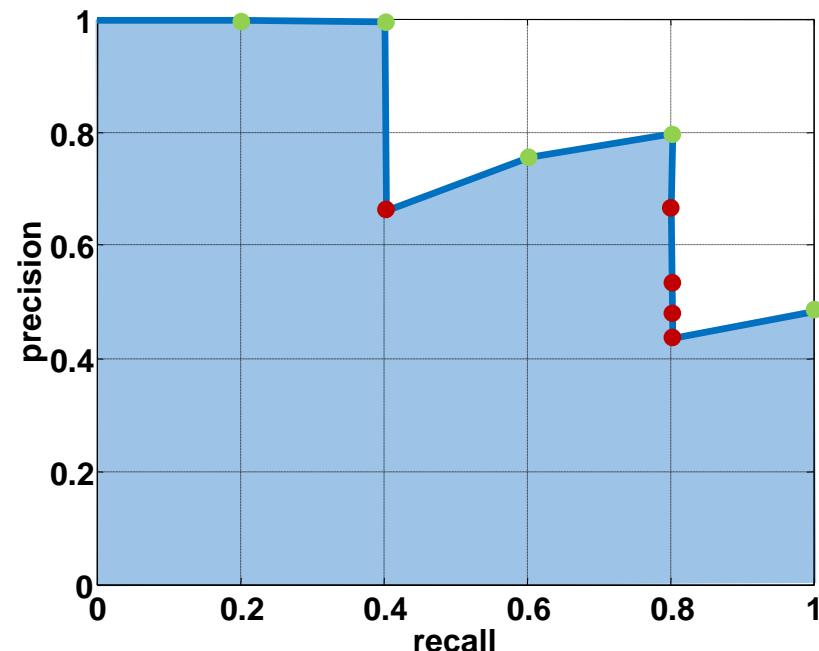
Scoring retrieval quality



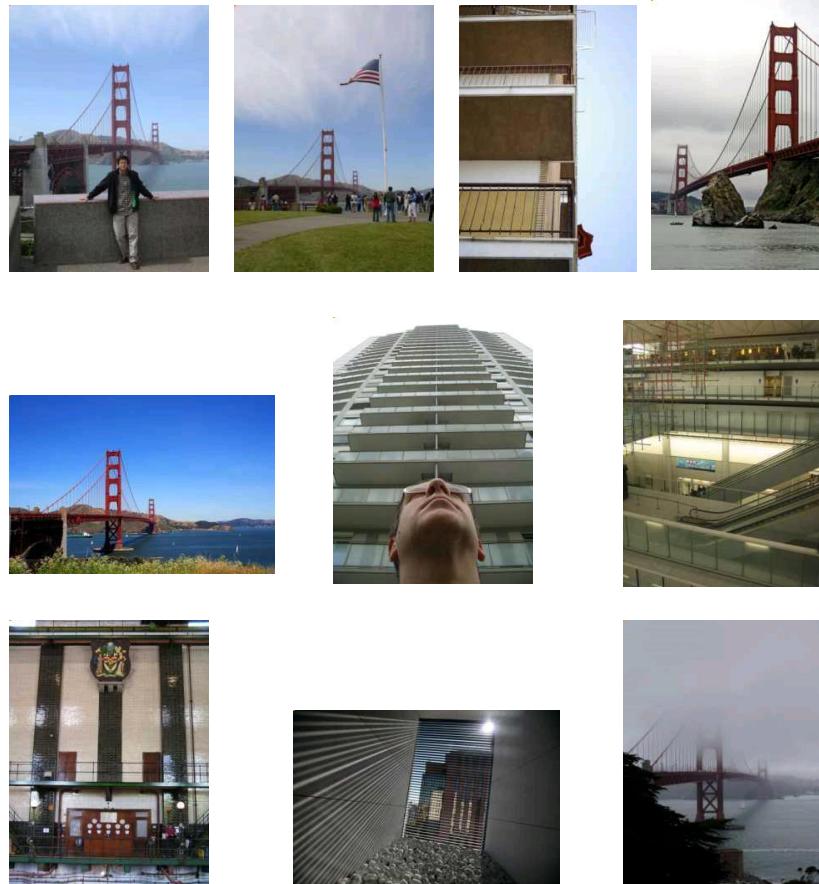
Query

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

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

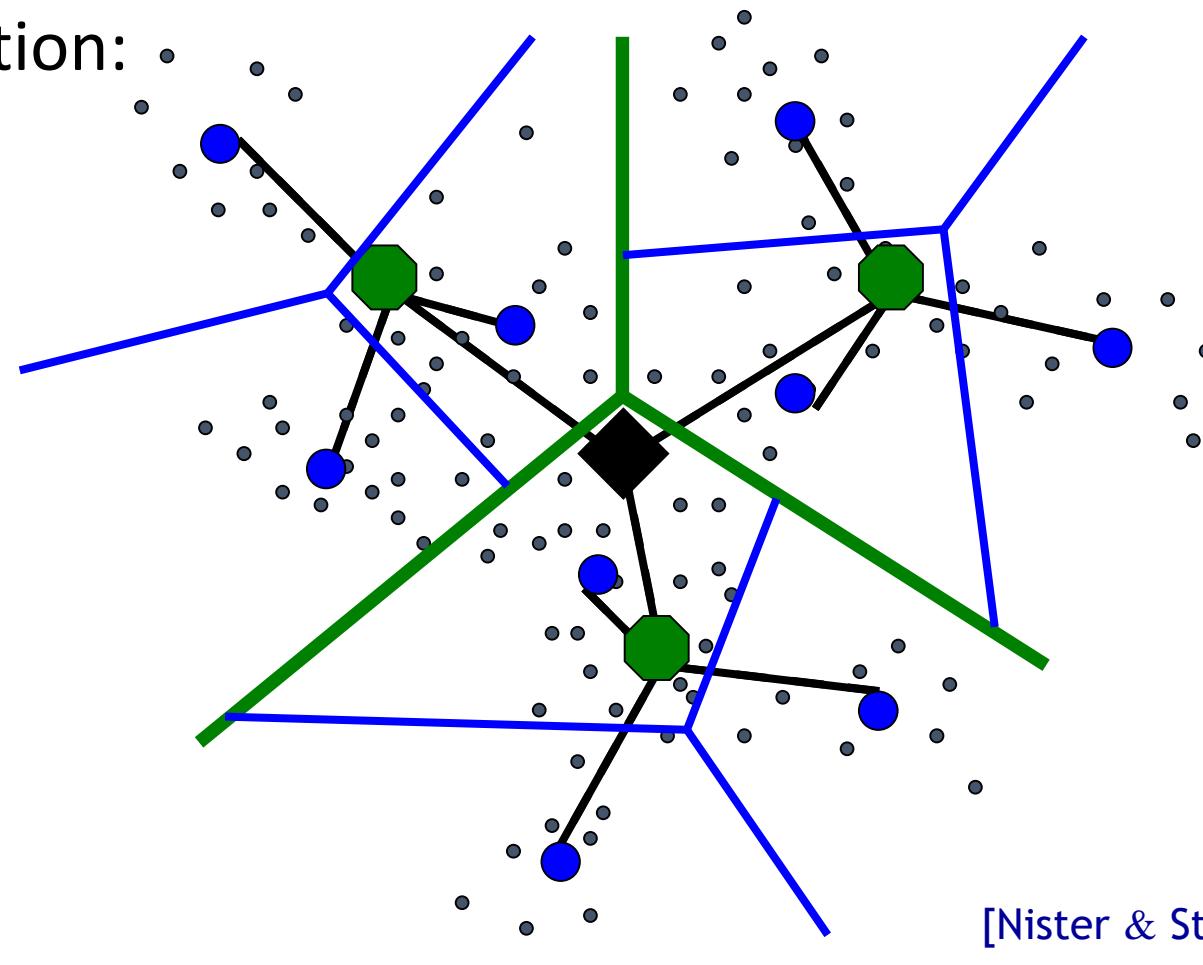


Results (ordered):



Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:

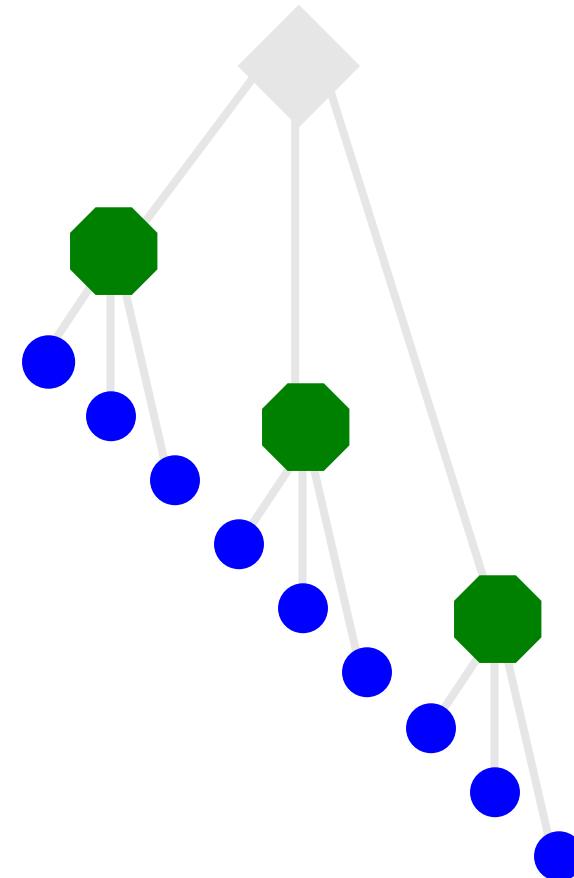


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

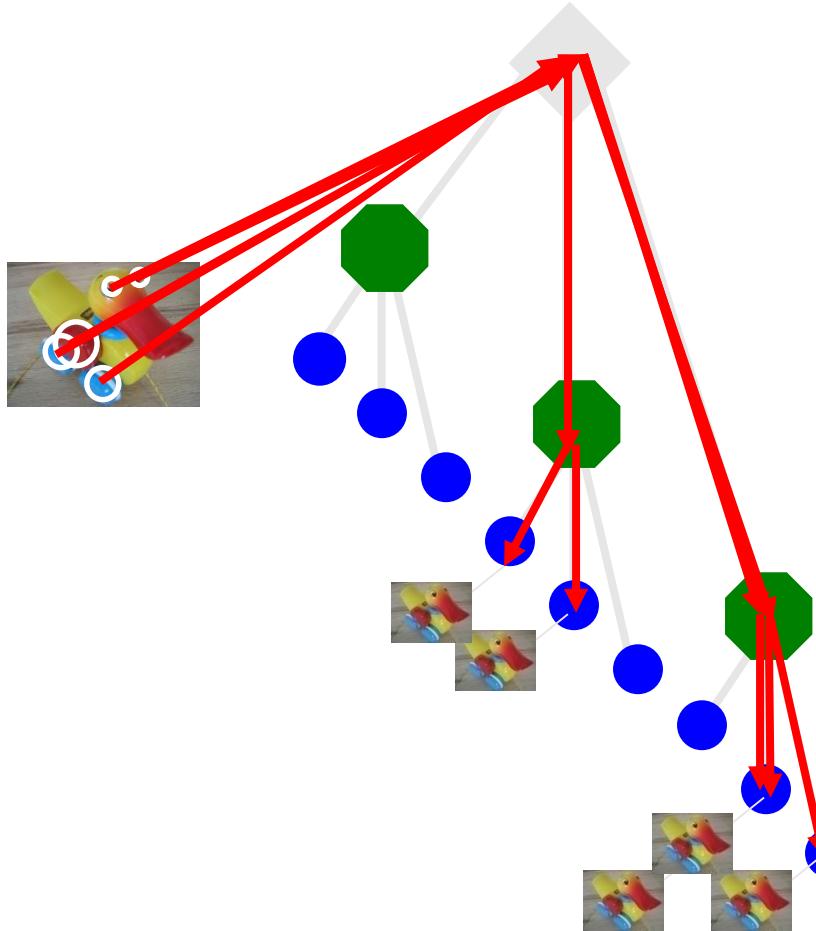
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Vocabulary Tree

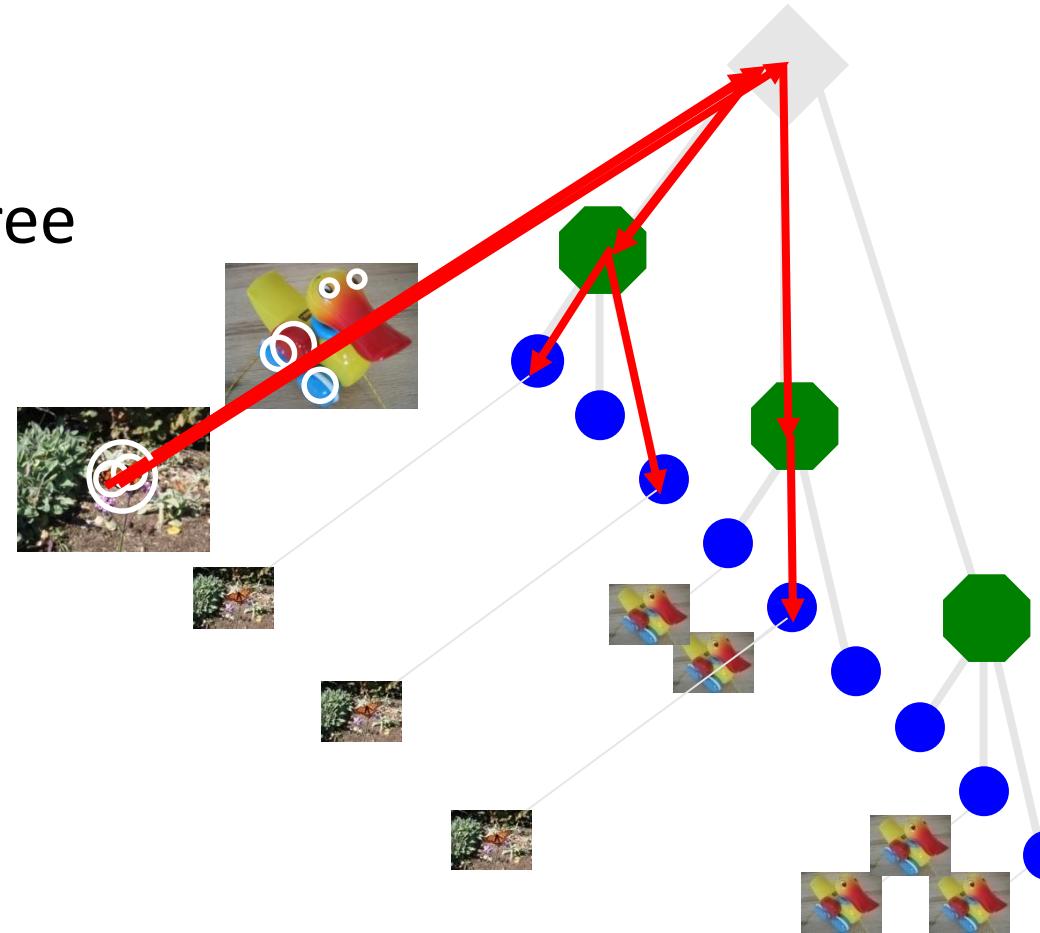
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Vocabulary Tree

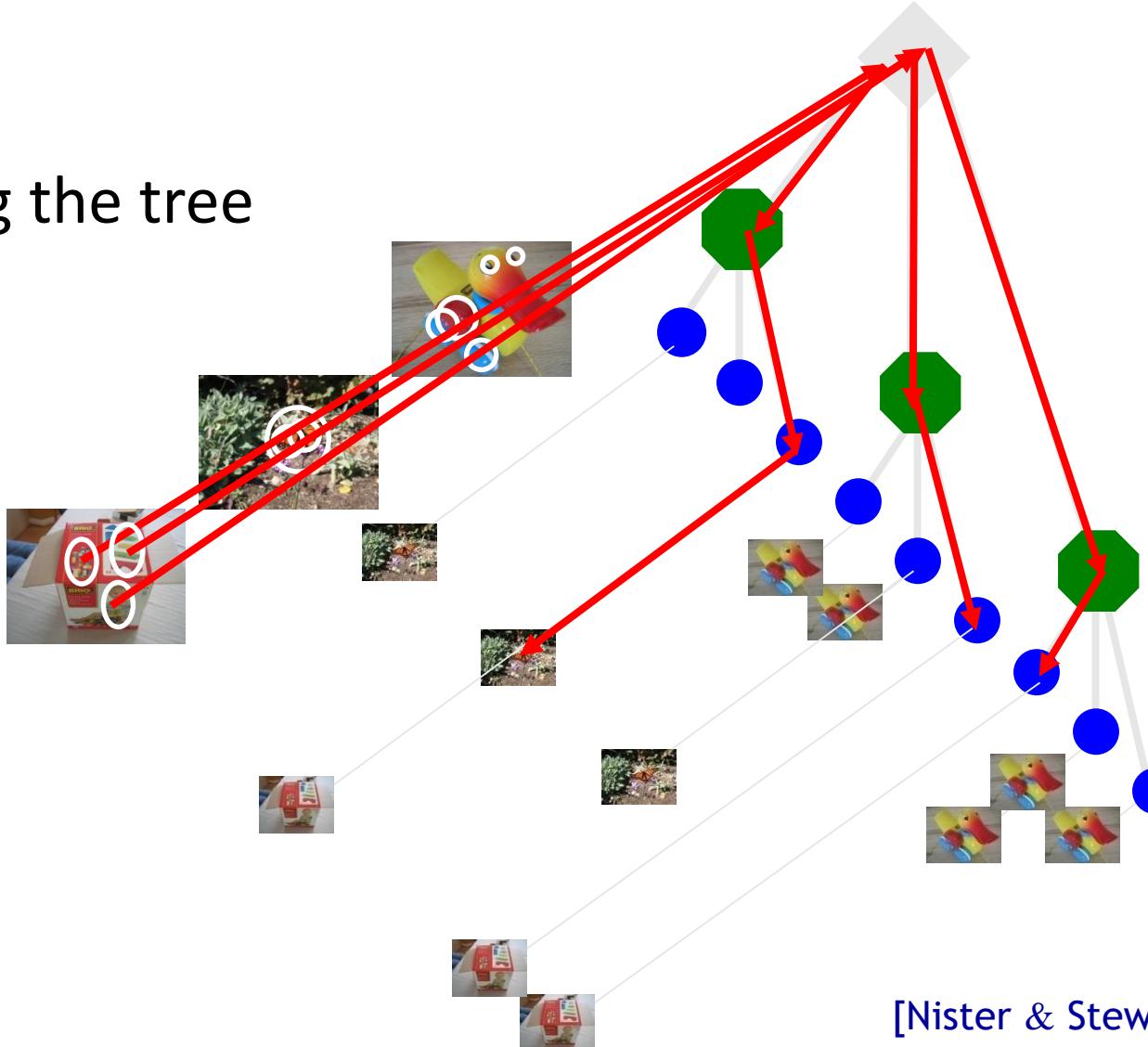
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Vocabulary Tree

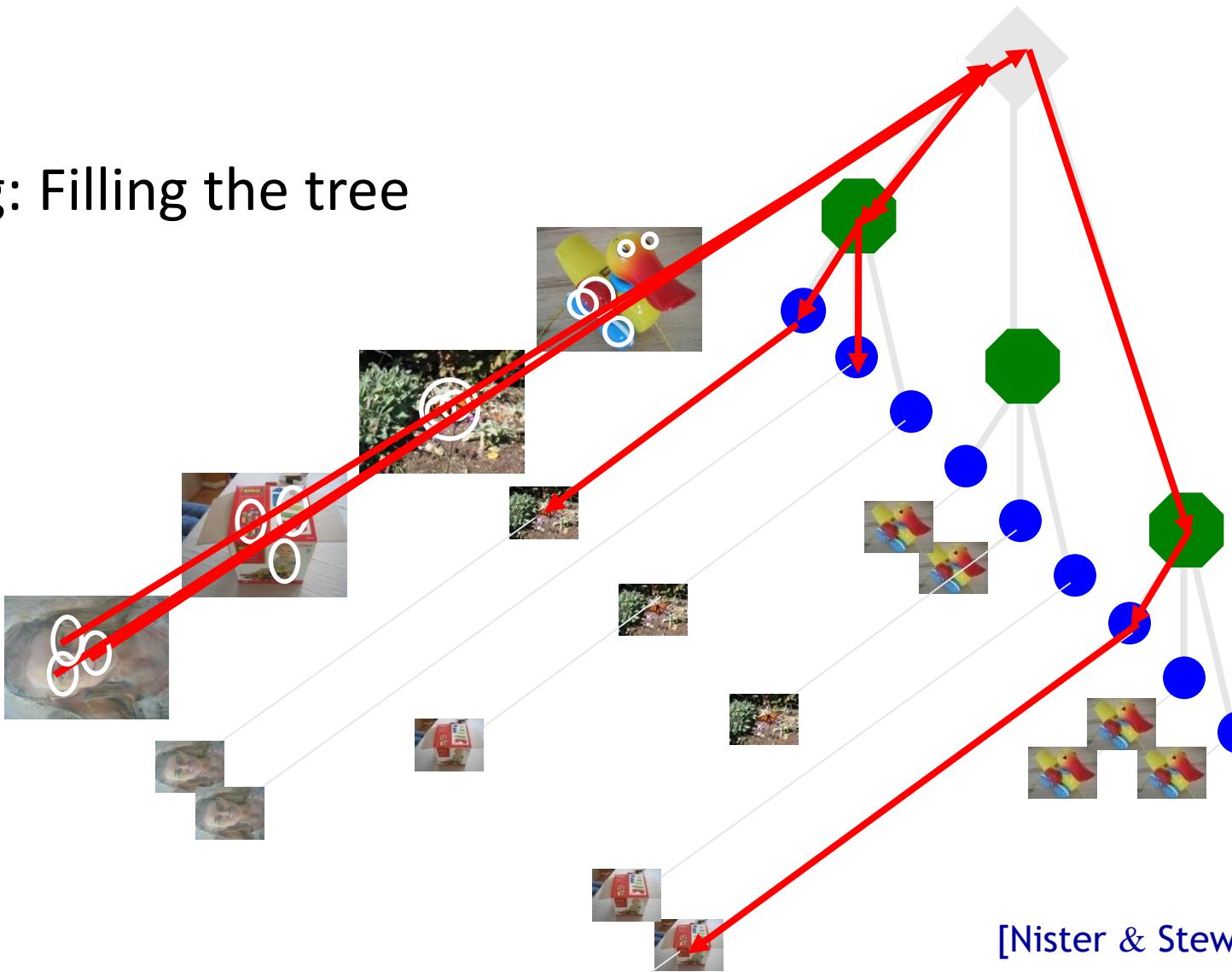
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Vocabulary Tree

- Training: Filling the tree



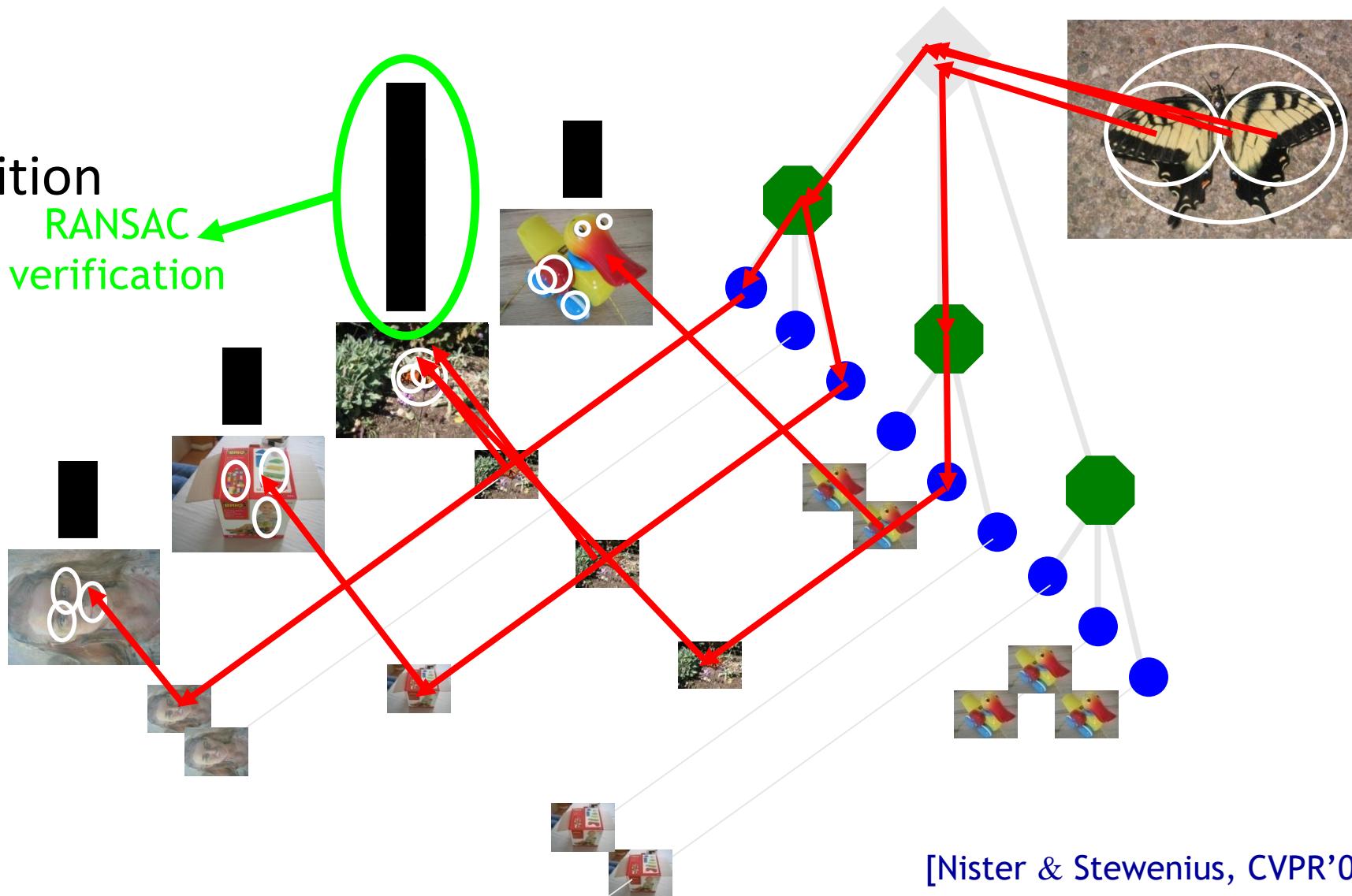
[Nister & Stewenius, CVPR'06]

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

Vocabulary Tree

- Recognition

RANSAC
verification



[Nister & Stewenius, CVPR'06]

Bags of words: pros and cons

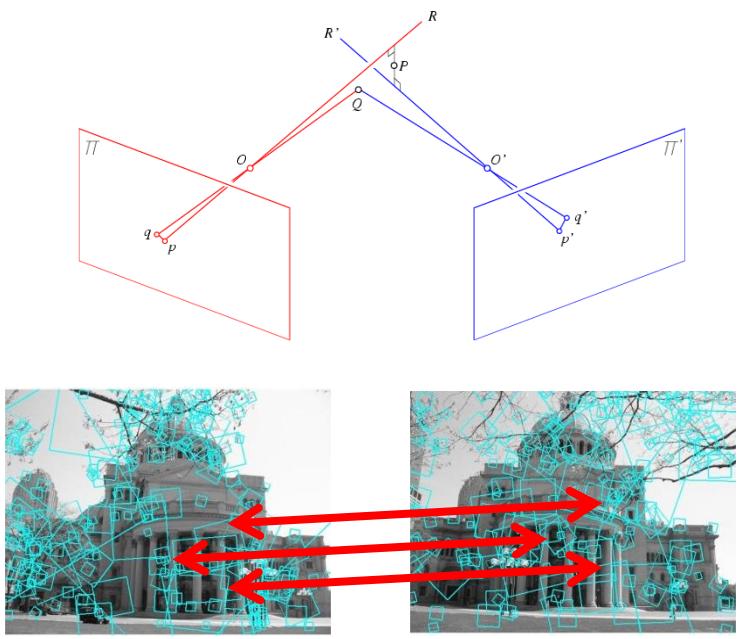
- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides vector representation for sets
- + very good results in practice

- 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

Summary So Far

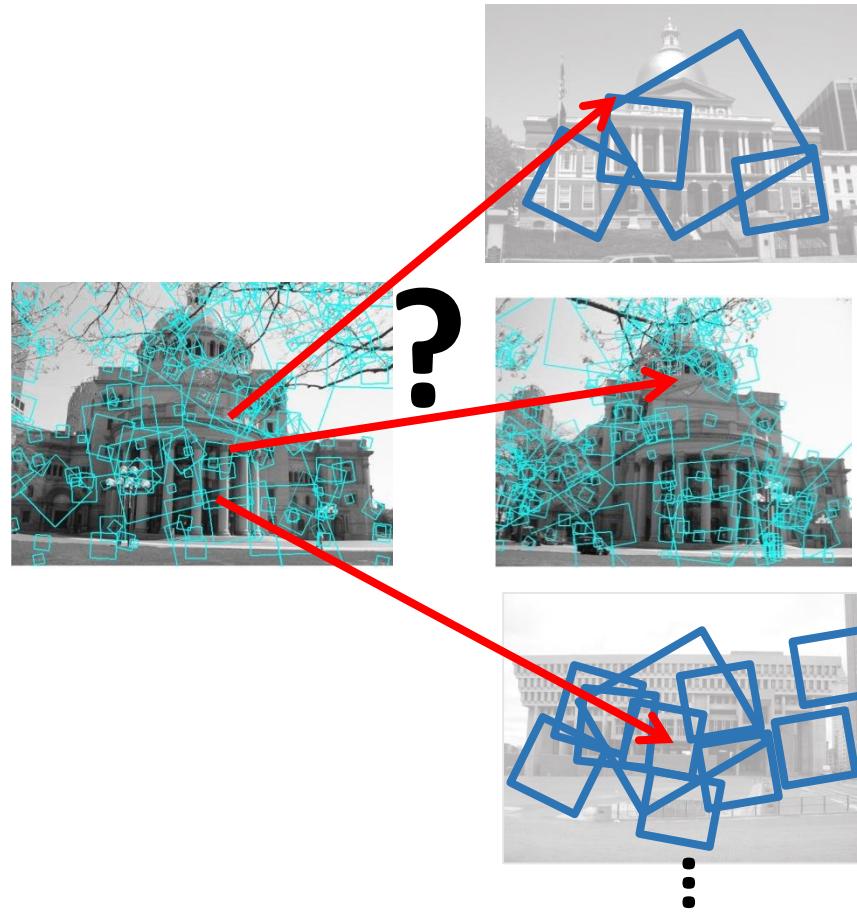
- **Matching local invariant features:** useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
 - Summarize image by distribution of words
 - Index individual words
- **Inverted index:** pre-compute index to enable faster search at query time

Multi-view matching



Matching two given
views for depth

vs



Search for a matching view
for recognition

Instance recognition

- 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

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?

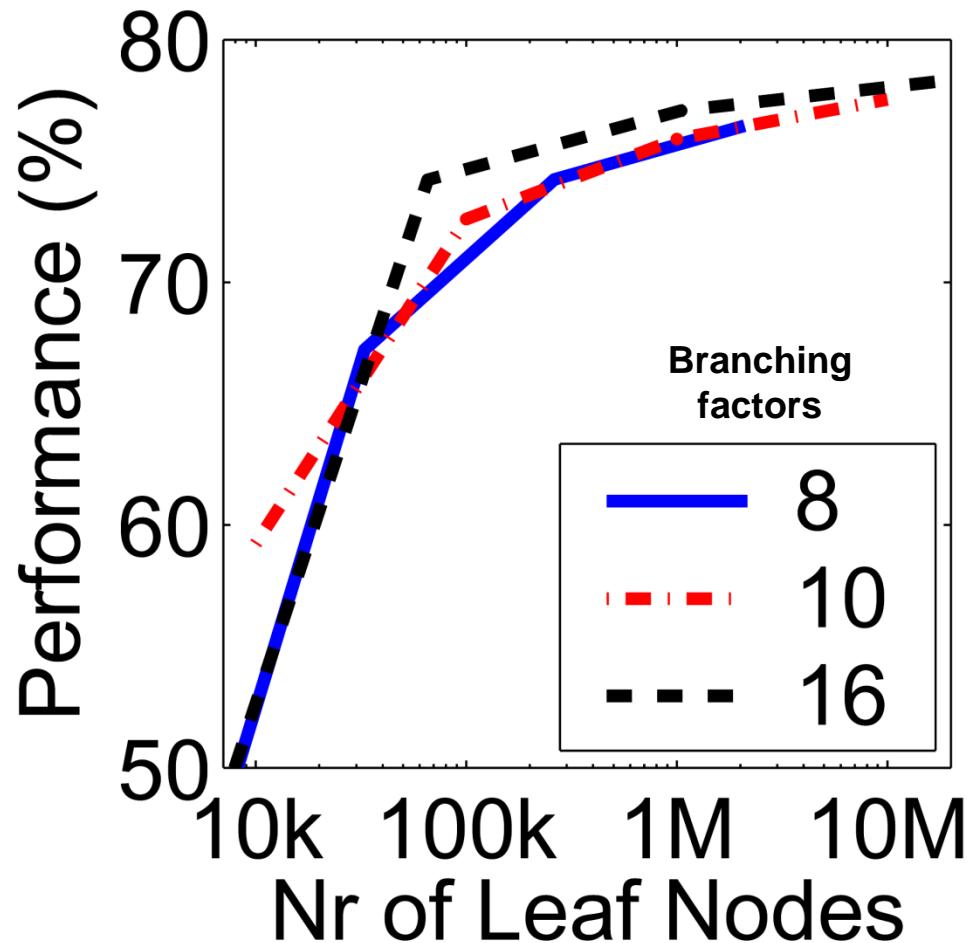
Instance recognition: remaining issues

- 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?

Instance recognition: remaining issues

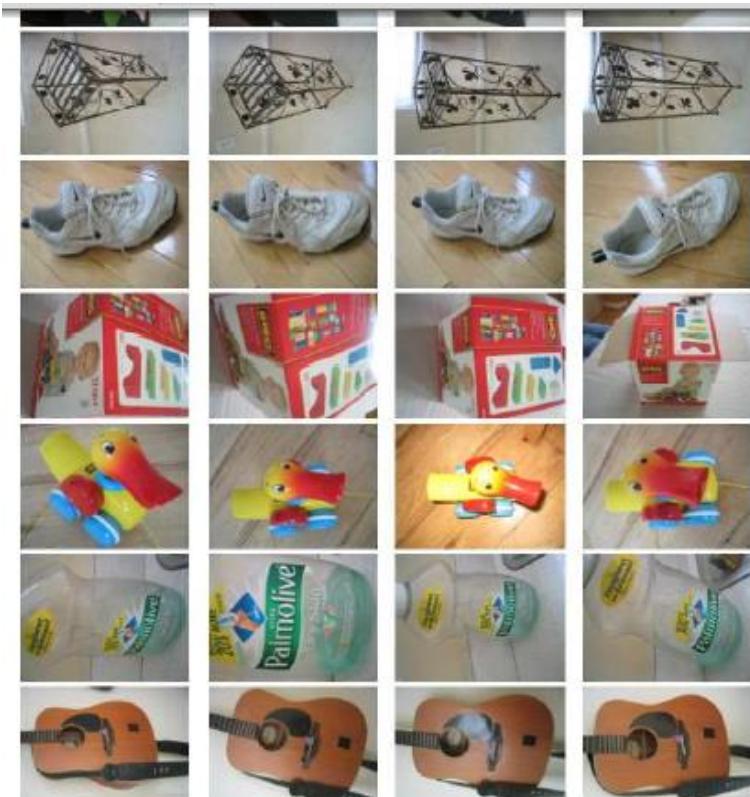
- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size



Influence on performance, sparsity

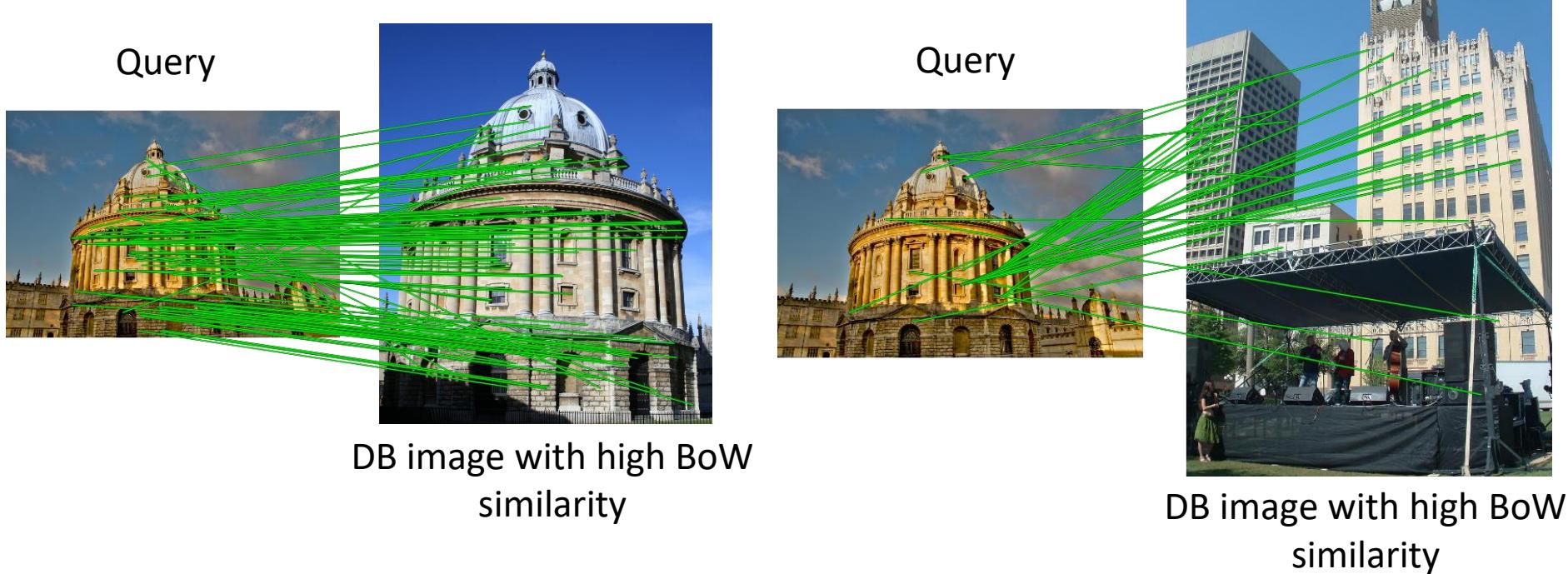
Results for recognition task
with 6347 images



Instance recognition: remaining issues

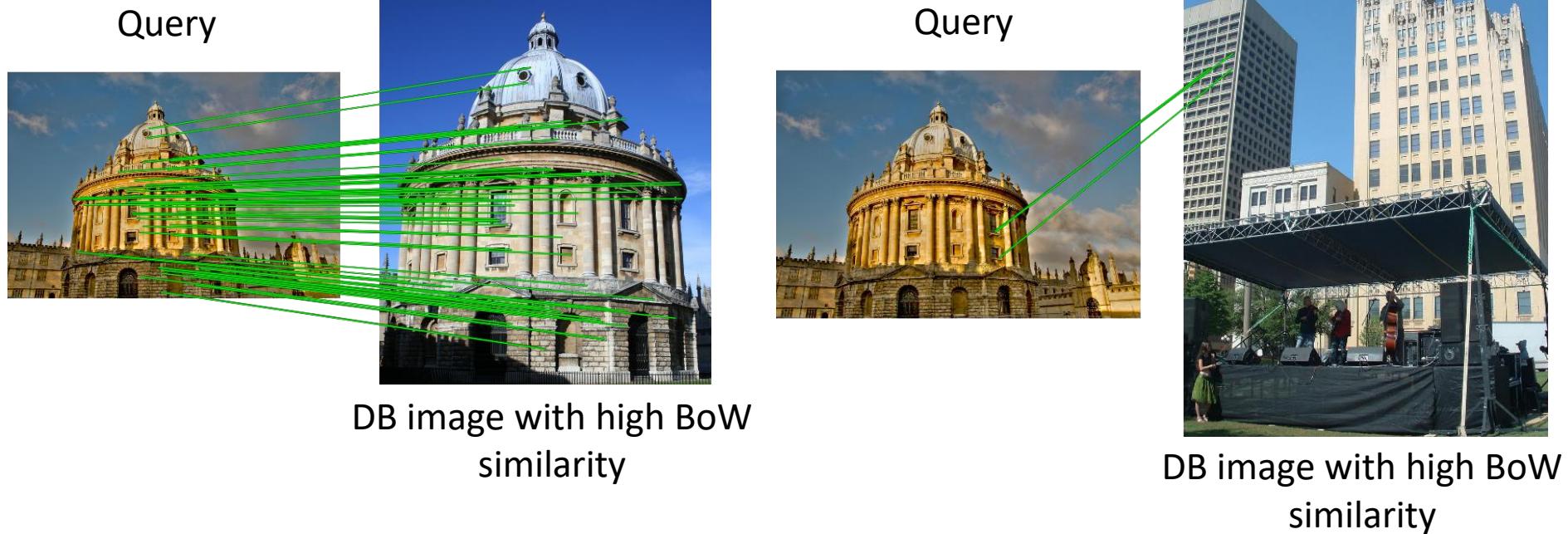
- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Spatial Verification



Both image pairs have many visual words in common.

Spatial Verification

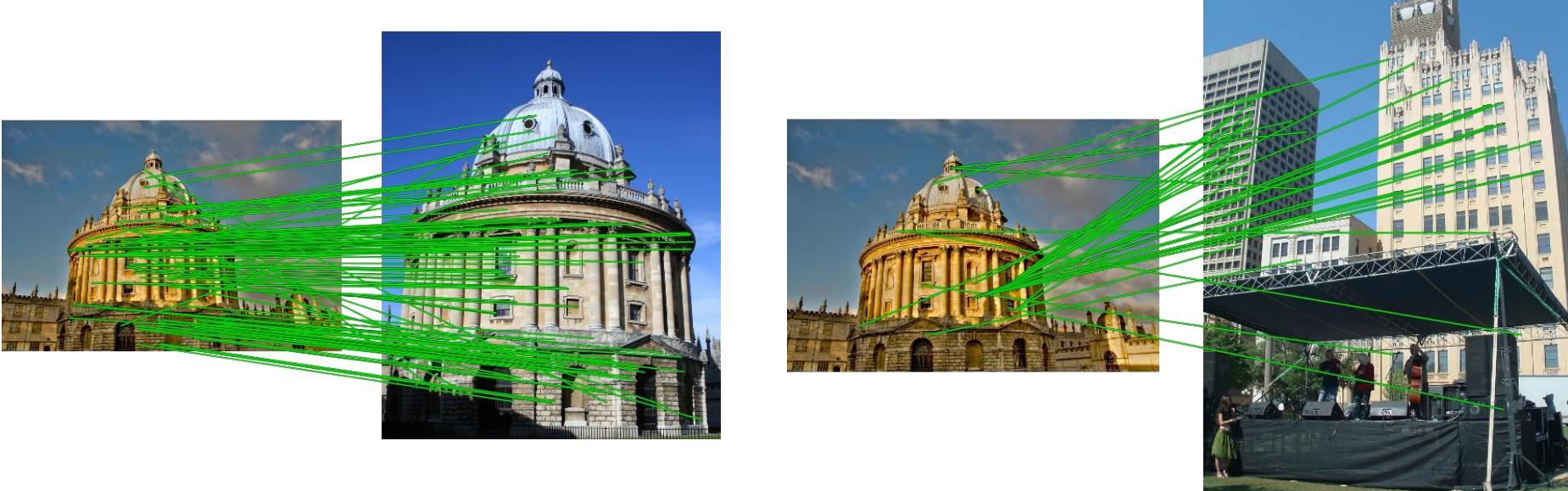


Only some of the matches are mutually consistent

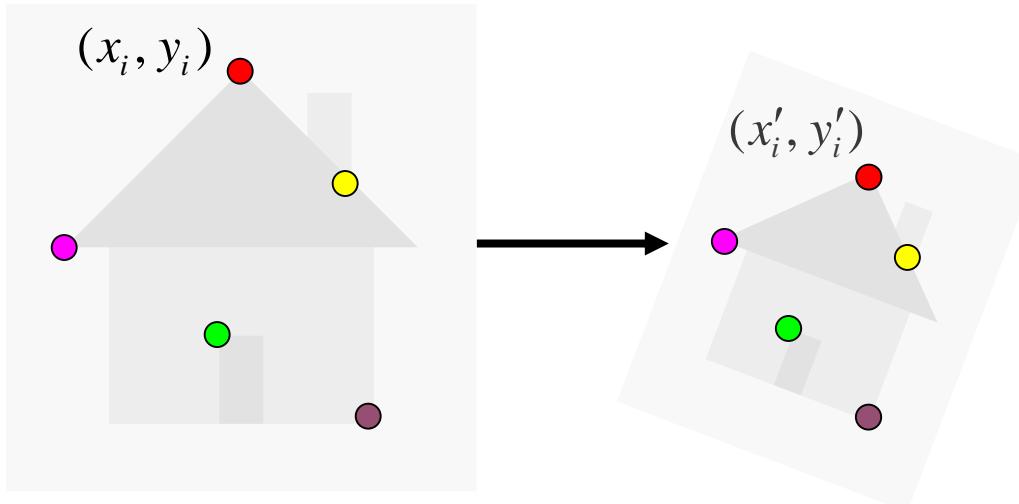
Spatial Verification: two basic strategies

- 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



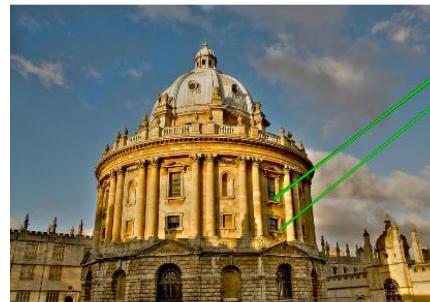
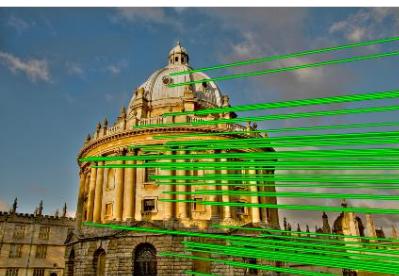
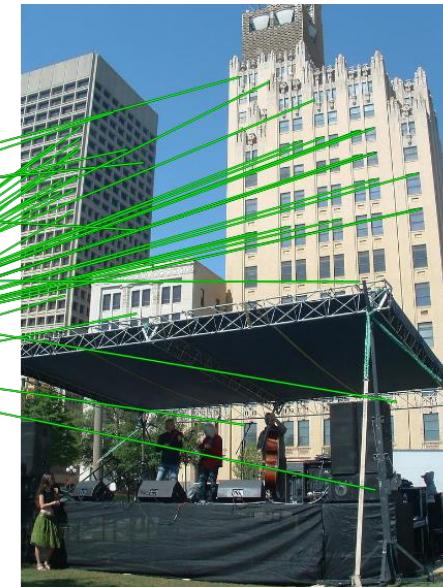
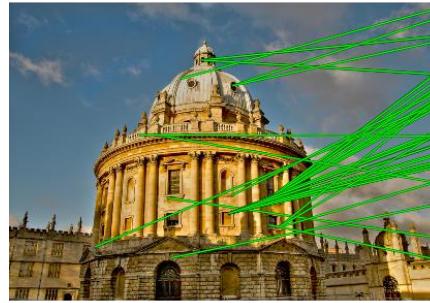
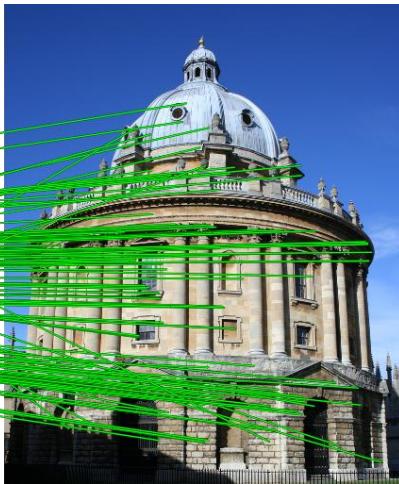
Recall: Fitting an affine transformation



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

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$
$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \dots & & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ \dots \end{bmatrix}$$

RANSAC verification



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>



Query
region

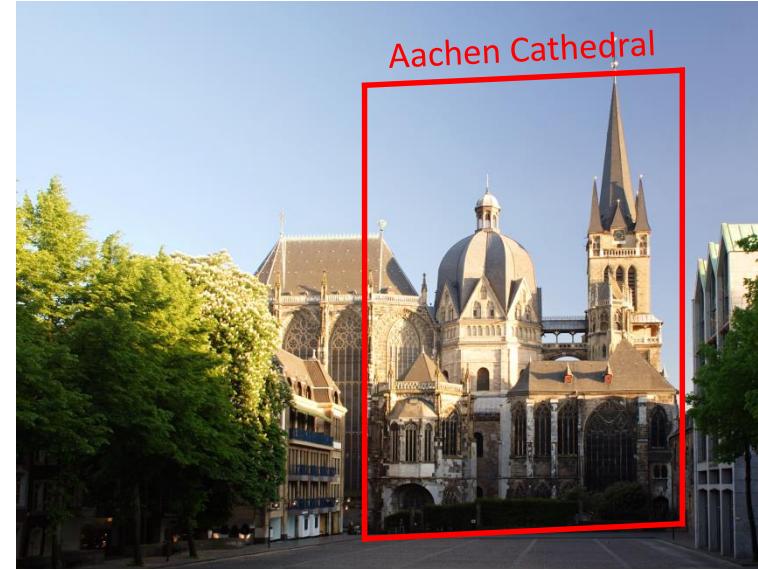


Retrieved frames



Kristen Grauman

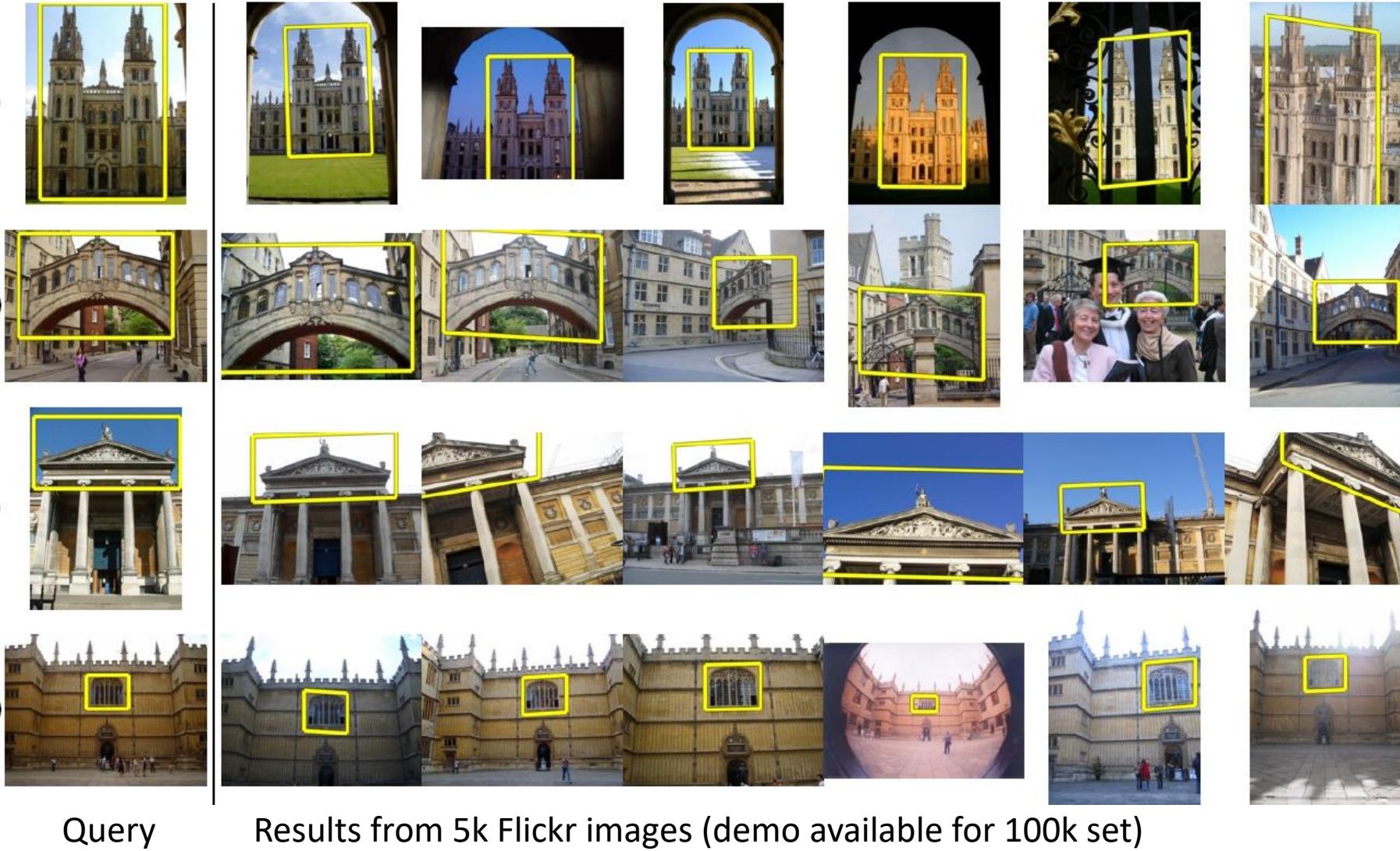
Example Applications



Mobile tourist guide

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

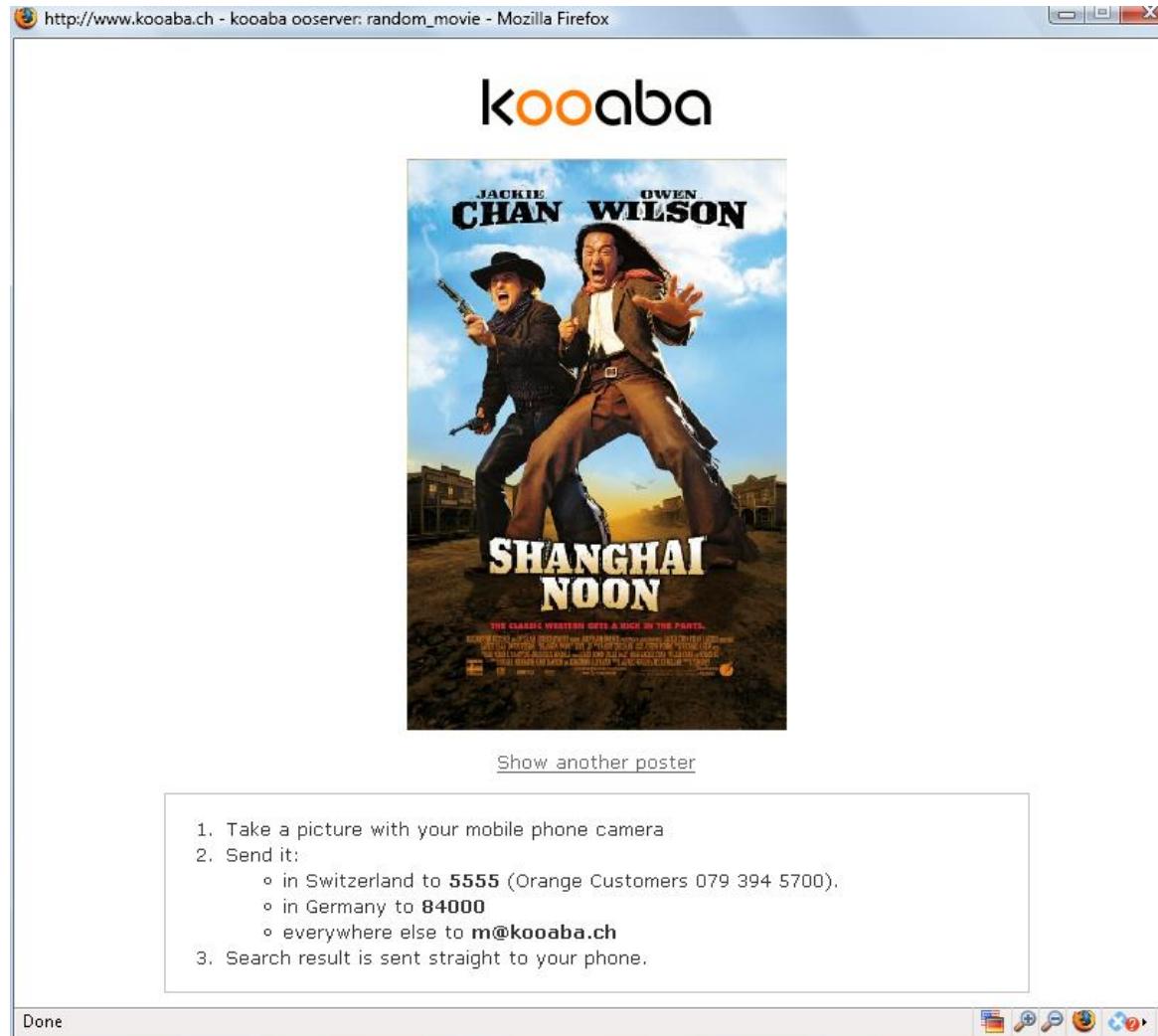
Application: Large-Scale Retrieval



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](#)

Ihr A

Früh und auf Topin

→

11:03

Google goggles labs

Lammkoteletts vom Biobauern mit Schalotten, Tomatencoulis und Basilikum-Gnocchi

German (auto) » English

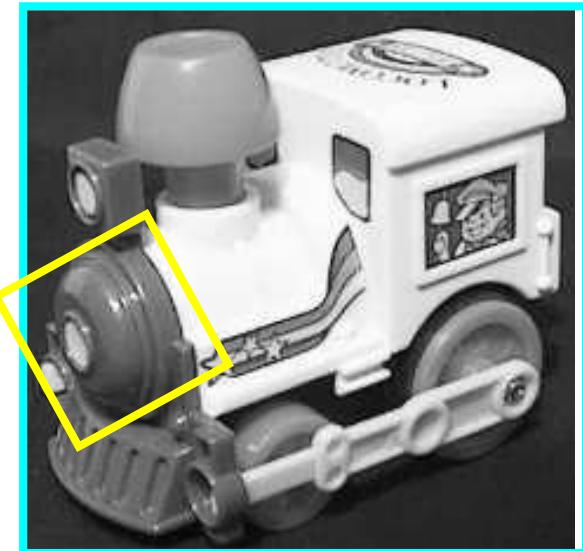
Lamb chops from the farmers with the shallots, tomato sauce and basil gnocchi

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).



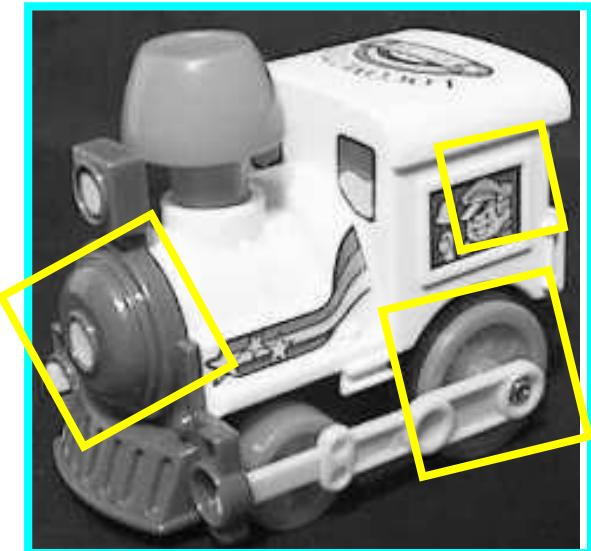
Model



Novel image

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

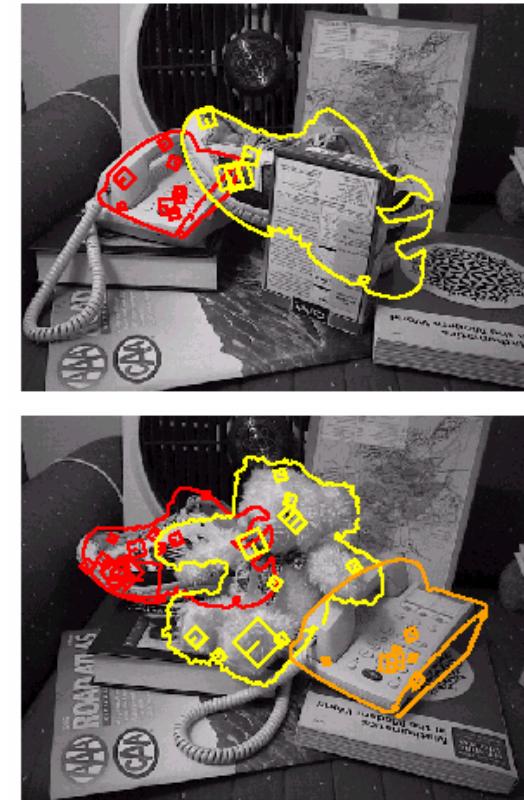
Example result



Background subtract for
model boundaries



Objects recognized,



Recognition in spite
of occlusion

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.

Gen Hough vs RANSAC

GHT

- Single correspondence -> vote for all consistent parameters
- Represents uncertainty in the model parameter space
- Linear complexity in number of correspondences and number of voting cells; beyond 4D vote space impractical
- Can handle high outlier ratio

RANSAC

- Minimal subset of correspondences to estimate model -> count inliers
- Represents uncertainty in image space
- Must search all data points to check for inliers each iteration
- Scales better to high-d parameter spaces

What else can we borrow from text retrieval?

Index

"Along I-75," From Detroit to Florida; *inside back cover*
"Drive I-95," From Boston to Florida; *inside back cover*
1929 Spanish Trail Roadway; 101-102,104
511 Traffic Information; 83
A1A (Barrier Isl) - I-95 Access; 86
AAA (and CAA); 83
AAA National Office; 88
Abbreviations,
Colored 25 mile Maps; cover
Exit Services; 196
Travelogue; 85
Africa; 177
Agricultural Inspection Stns; 126
Ah-Tah-Thi-Ki Museum; 160
Air Conditioning, First; 112
Alabama; 124
Alachua; 132
County; 131
Alafia River; 143
Alapaha, Name; 126
Alfred B Maclay Gardens; 106
Alligator Alley; 154-155
Alligator Farm, St Augustine; 169
Alligator Hole (definition); 157
Alligator, Buddy; 155
Alligators; 100,135,138,147,156
Anastasia Island; 170
Anhaica; 108-109,146
Apalachicola River; 112
Appleton Mus of Art; 136
Aquifer; 102
Arabian Nights; 94
Art Museum, Ringling; 147
Aruba Beach Cafe; 183
Aucilla River Project; 106
Babcock-Web WMA; 151
Bahia Mar Marina; 184
Baker County; 99
Barefoot Mallmen; 182
Barge Canal; 137
Bee Line Expy; 80
Belz Outlet Mall; 89
Bernard Castro; 136
Big "I"; 165
Big Cypress; 155,158
Big Foot Monster; 105
Butterfly Center, McGuire; 134
CAA (see AAA)
CCC, The; 111,113,115,135,142
Ca d'Zan; 147
Caloosahatchee River; 152
Name; 150
Canaveral Natnl Seashore; 173
Cannon Creek Airpark; 130
Canopy Road; 106,169
Cape Canaveral; 174
Castillo San Marcos; 169
Cave Diving; 131
Cayo Costa, Name; 150
Celebration; 93
Charlotte County; 149
Charlotte Harbor; 150
Chautauqua; 116
Chiplay; 114
Name; 115
Choctawatchee, Name; 115
Circus Museum, Ringling; 147
Citrus; 88,97,130,136,140,180
CityPlace, W Palm Beach; 180
City Maps,
Ft Lauderdale Expwy; 194-195
Jacksonville; 163
Kissimmee Expwy; 192-193
Miami Expressways; 194-195
Orlando Expressways; 192-193
Pensacola; 26
Tallahassee; 191
Tampa-St. Petersburg; 63
St. Augustine; 191
Civil War; 100,108,127,138,141
Clearwater Marine Aquarium; 187
Collier County; 154
Collier, Barron; 152
Colonial Spanish Quarters; 168
Columbia County; 101,128
Coquina Building Material; 165
Corkscrew Swamp, Name; 154
Cowboys; 95
Crab Trap II; 144
Cracker, Florida; 88,95,132
Crosstown Expy; 113,98,143
Cuban Bread; 184
Dade Battlefield; 140
Dade, Maj. Francis; 139-140,161
Dania Beach Hurricane; 184
Driving Lanes; 85
Duval County; 163
Eau Gallie; 175
Edison, Thomas; 152
Eglin AFB; 116-118
Eight Reale; 176
Ellenton; 144-145
Emanuel Point Wreck; 120
Emergency Callboxes; 83
Epiphytes; 142,148,157,159
Escambia Bay; 119
Bridge (I-10); 119
County; 120
Estero; 153
Everglade,90,95,139-140,154-160
Draining off; 156,181
Wildlife MA; 160
Wonder Gardens; 154
Falling Waters SP; 115
Fantasy of Flight; 95
Fayer Dykes SP; 171
Fires, Forest; 166
Fires, Prescribed; 148
Fisherman's Village; 151
Flagler County; 171
Flagler, Henry; 97,165,167,171
Florida Aquarium; 186
Florida,
12,000 years ago; 187
Cavern SP; 114
Map of all Expressways; 2-3
Mus of Natural History; 134
National Cemetery ; 141
Part of Africa; 177
Platform; 187
Sheriff's Boys Camp; 126
Sports Hall of Fame; 130
Sun 'n Fun Museum; 97
Supreme Court; 107
Florida's Turnpike (FTP); 178,189
25 mile Strip Maps; 66
Administration; 189
Coin System; 190
Exit Services; 189
HEFT; 76,161,190
History; 189
Names; 189
Service Plazas; 190
Spur SR91; 76

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 reach \$100bn by 2008, up from its predicted 30% jump in exports. The ministry also predicted a 18% rise in imports. The ministry said that China's trade surplus was due to a lack of market access and that the surplus was not sustainable. The ministry also said that the surplus was due to a lack of market access and that the surplus was not sustainable.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value



91

tf-idf weighting

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

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

Number of occurrences of word i in document d → n_{id}

Number of words in document d → n_d

Total number of documents in database → N

Number of documents word i occurs in, in whole database → n_i

Query expansion

Query: *golf green*

Results:

- How can the grass on the *greens* at a *golf* course be so perfect?
- For example, a skilled *golfer* 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

Query Expansion



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.

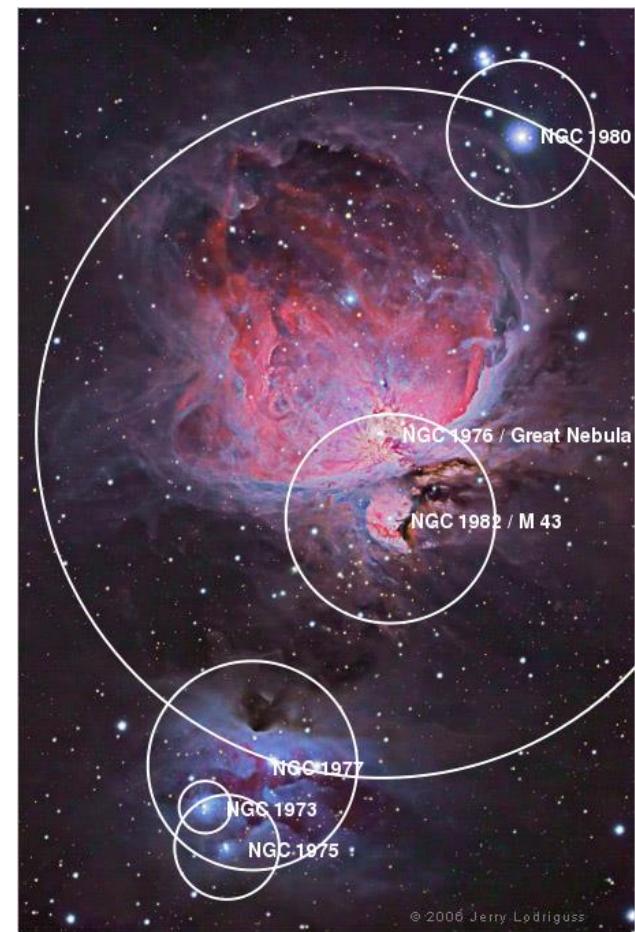
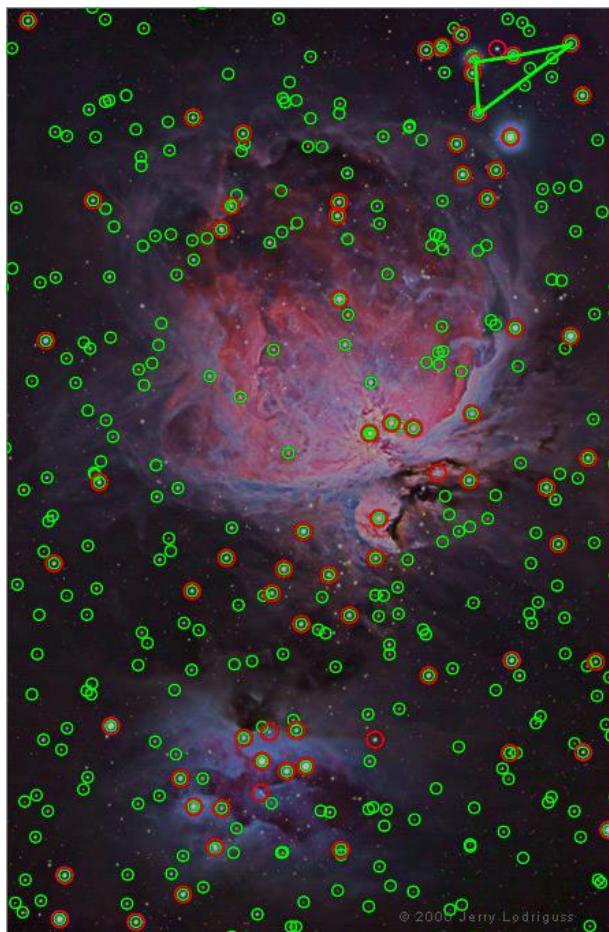


Making the Sky Searchable: Fast Geometric Hashing for Automated Astrometry

Sam Roweis, Dustin Lang & Keir Mierle
University of Toronto

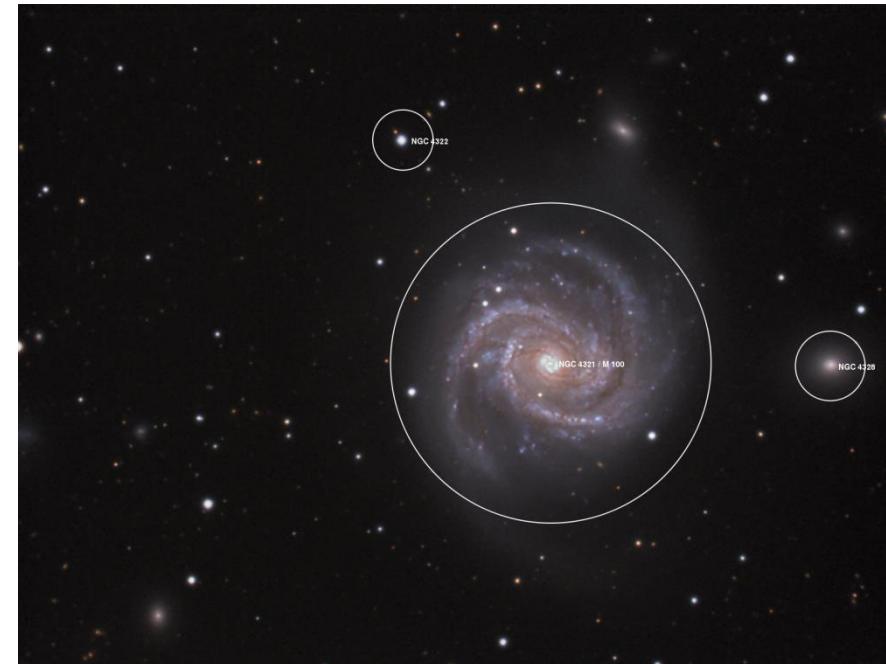
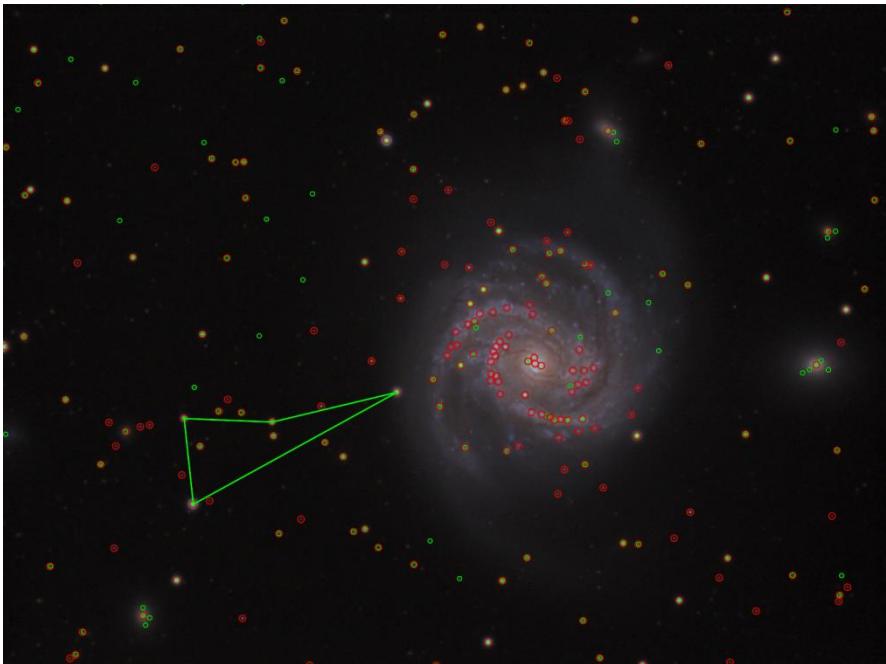
David Hogg & Michael Blanton
New York University

Example



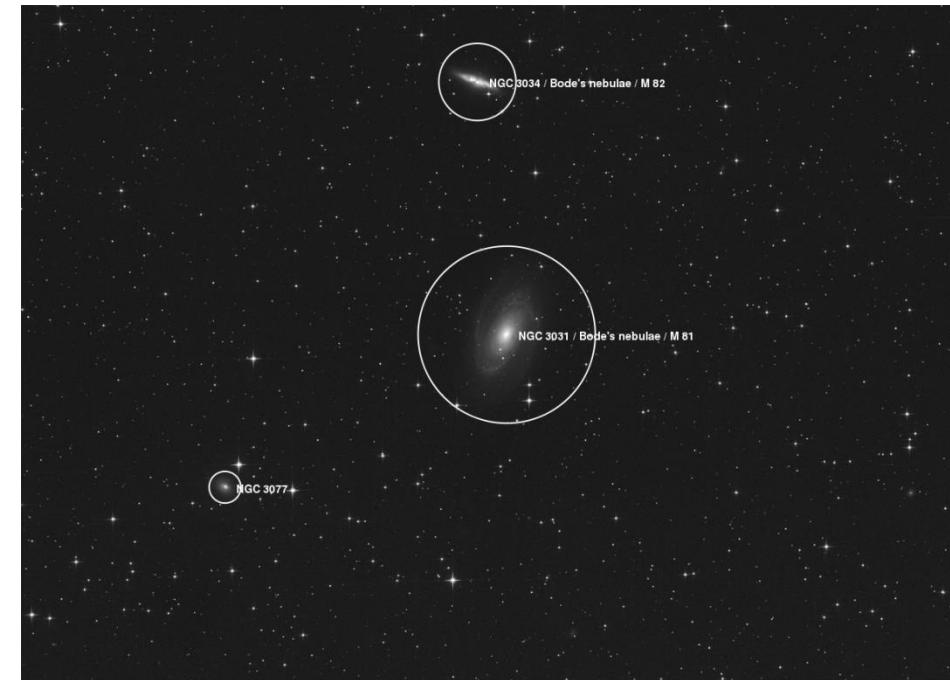
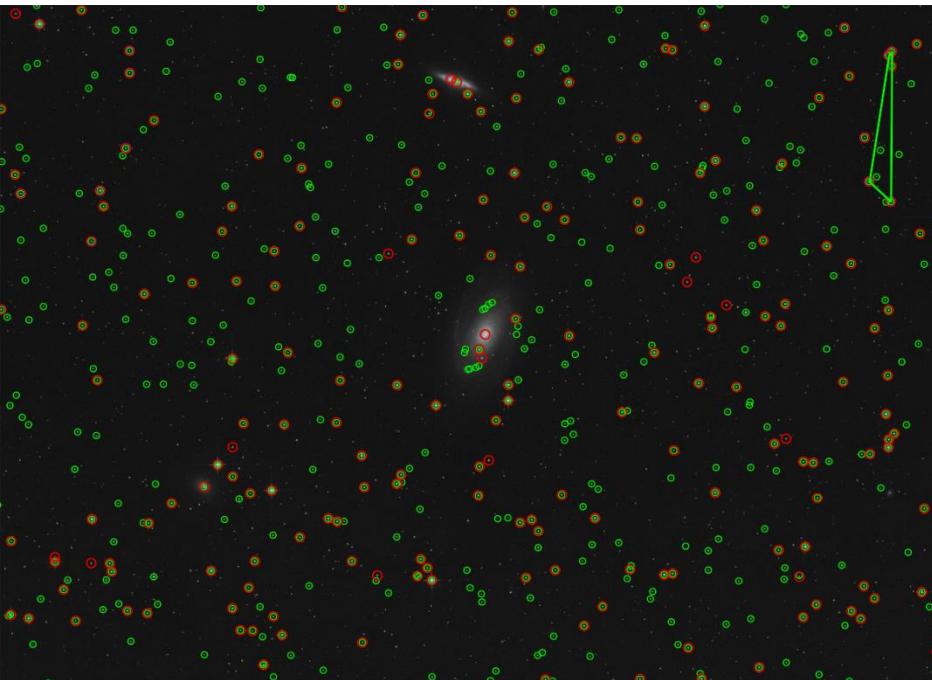
A shot of the Great Nebula, by Jerry Lodriguss (c.2006), from astropix.com
<http://astrometry.net/gallery.html>

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>

Things to remember

- **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