Indexing with local features, Bag of words models

Thursday, Oct 29

Kristen Grauman
UT-Austin

Last time

- Interest point detection
  - Harris corner detector
  - Laplacian of Gaussian, automatic scale selection

Local features: main components

1) Detection: Identify the interest points

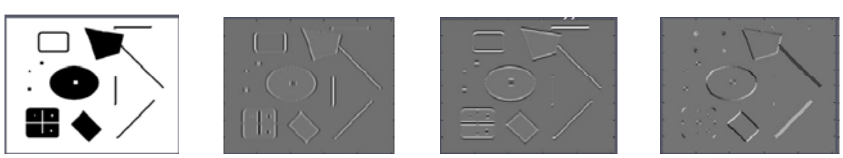
2) Description: Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views

Corners as distinctive interest points

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).



Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

Harris corners example

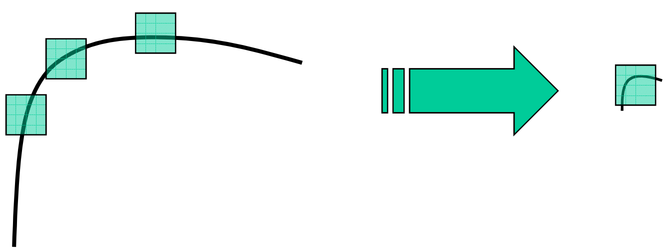
Any local max in 3 x 3 window from the R map

Only local maxes exceeding average R (thresholded)

Properties of the Harris corner detector

Rotation invariant? Yes

Scale invariant? No

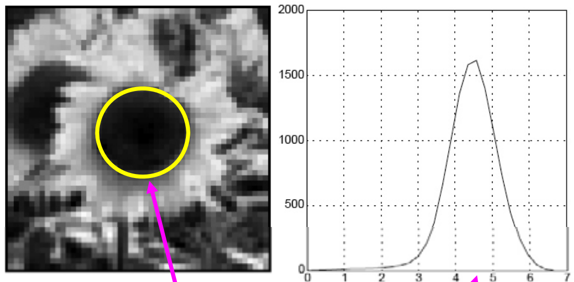


All points will be classified as edges

Corner !

Automatic scale selection

We define the *characteristic scale* as the scale that produces peak of Laplacian response



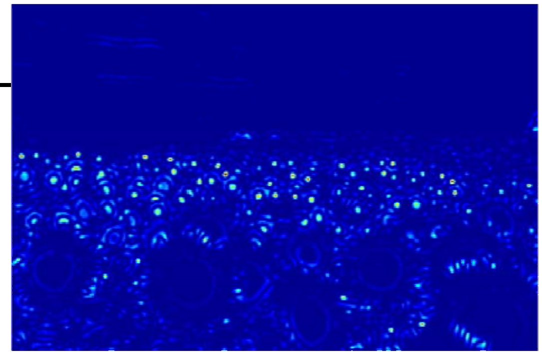
characteristic scale

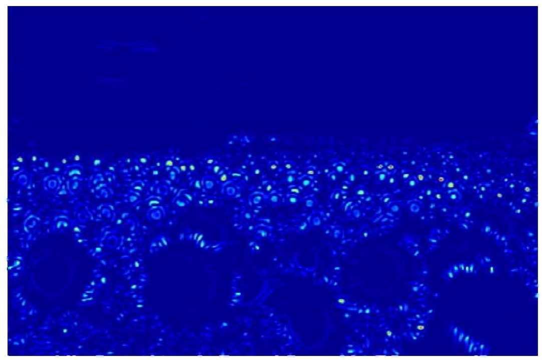
Slide credit: Lana Lazebnik

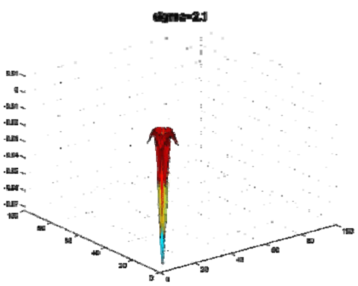
Example

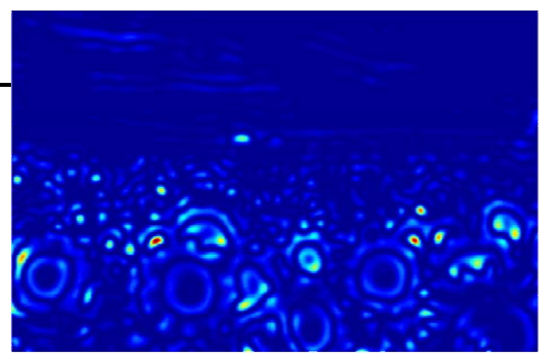
Original image at ¾ the size

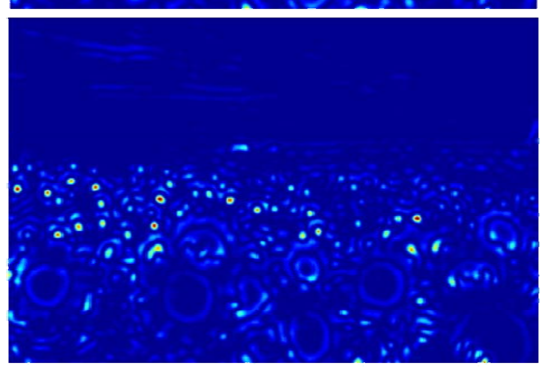
Original image
at ¾ the size

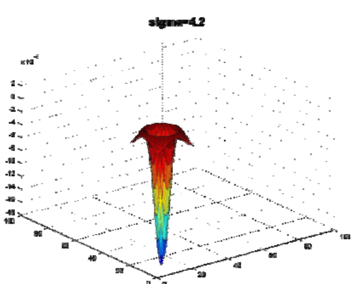


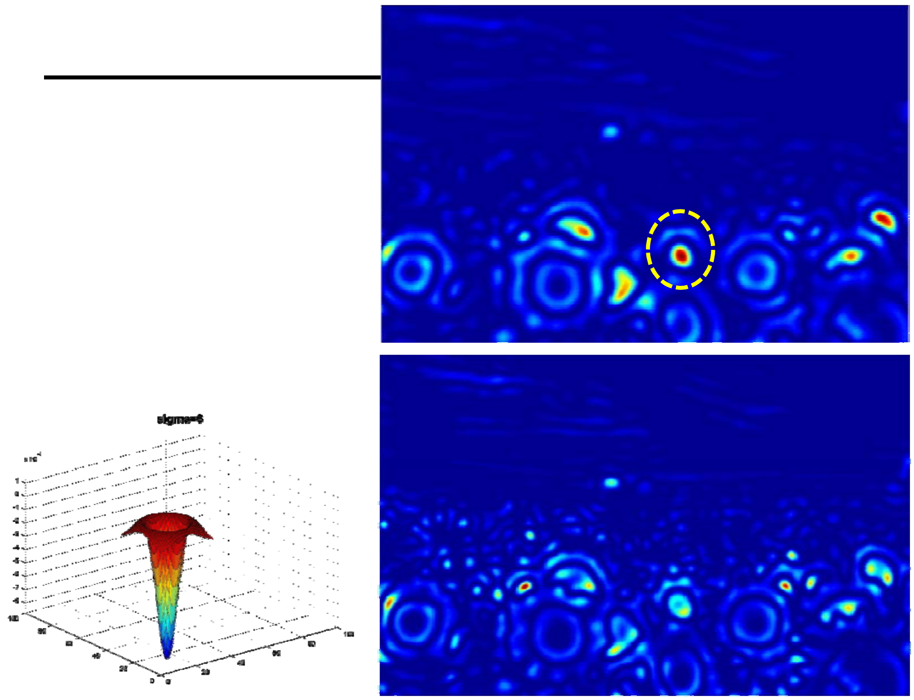


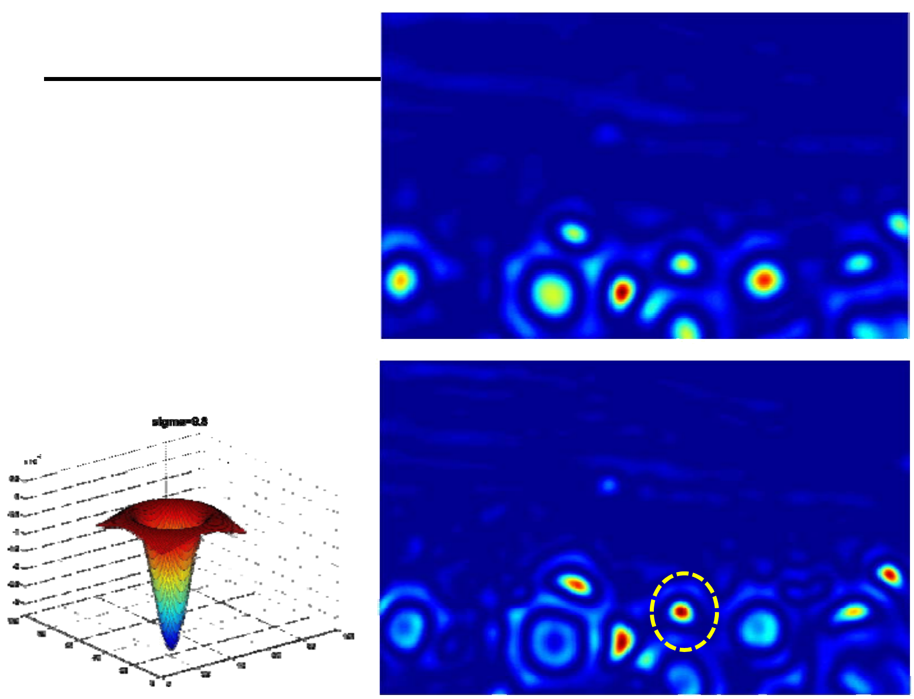


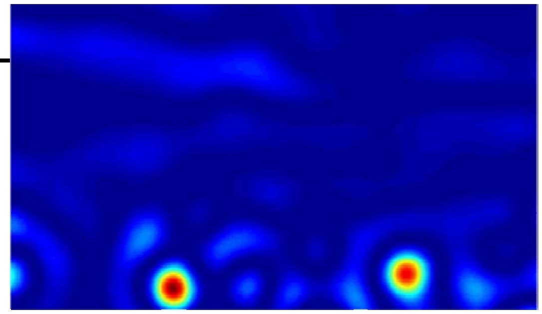


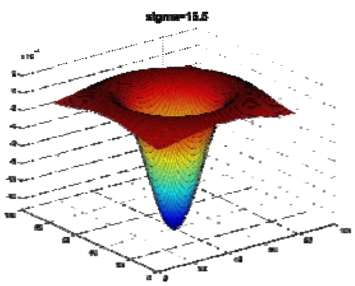


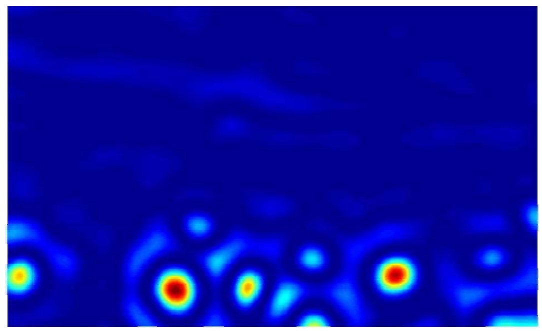


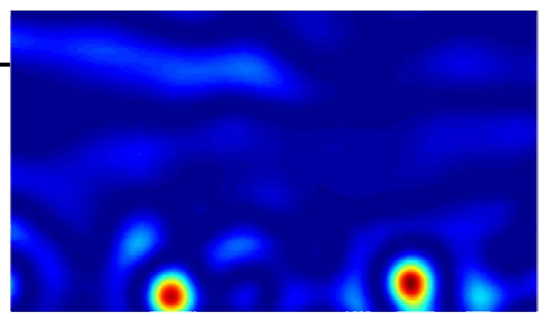

sigma=6

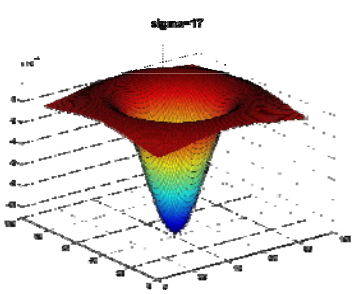

sigma=9.8

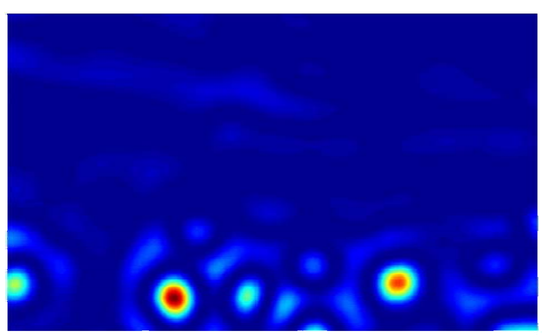



sigma=15.5



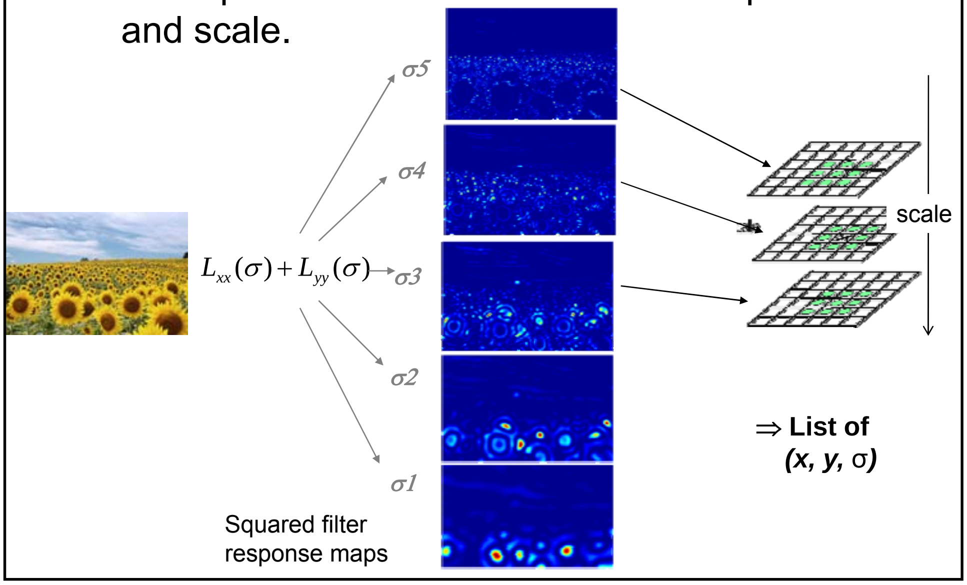



sigma=17



Scale invariant interest points

Interest points are local maxima in both position and scale.



Today

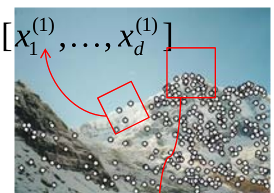
- Matching local features
- Indexing features
- Bag of words model

Local features: main components

1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

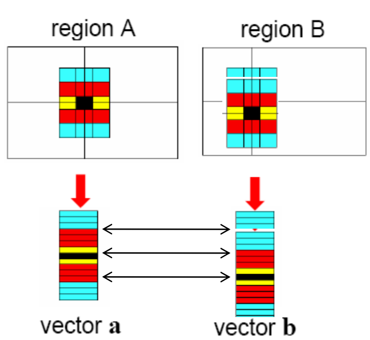
$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views

Raw patches as local descriptors

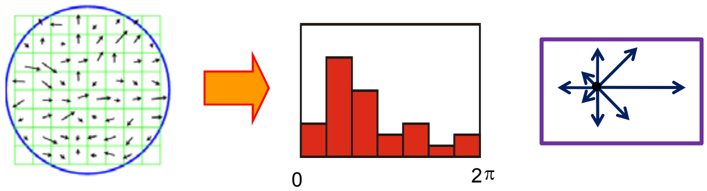


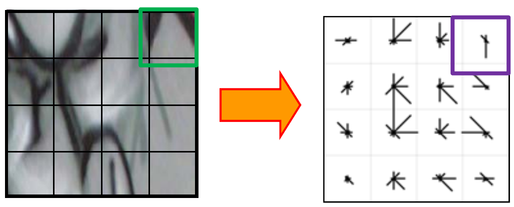
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.

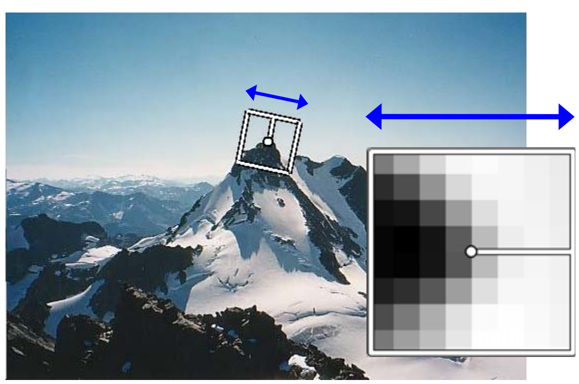




Why subpatches?

Why does SIFT have some illumination invariance?

Making the descriptor rotation invariant

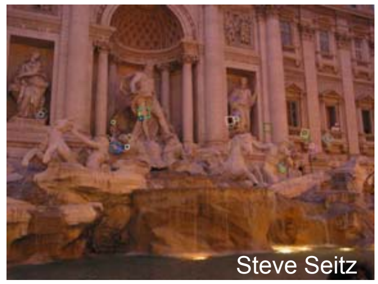


- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

Image from Matthew Brown

SIFT descriptor [Lowe 2004]

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

Steve Seitz

Local features: main components

1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views

Matching local features

Matching local features

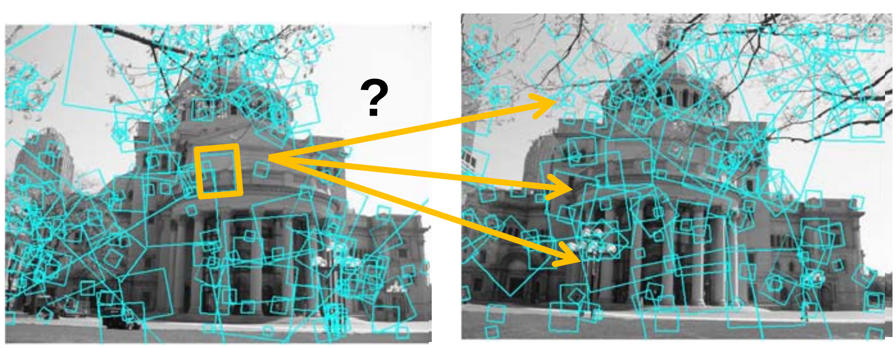


Image 1 Image 2

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

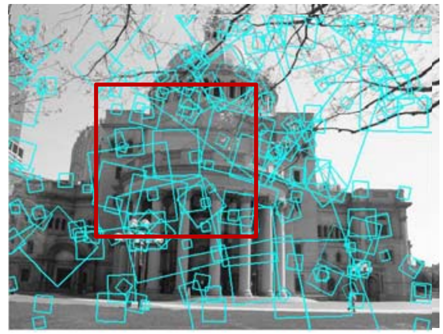


Image 2

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

Ambiguous matches

Image 1

Image 2

At what SSD value do we have a good match?

To add robustness to matching, can consider **ratio** : distance to best match / distance to second best match

If high, could be ambiguous match.

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

Automatic mosaicing

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

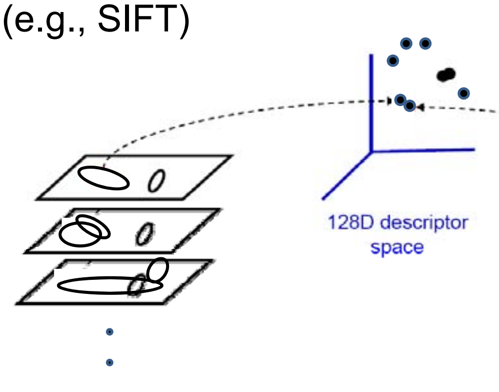
Lowe 2002

Today

- Matching local features
- Indexing features
- Bag of words model

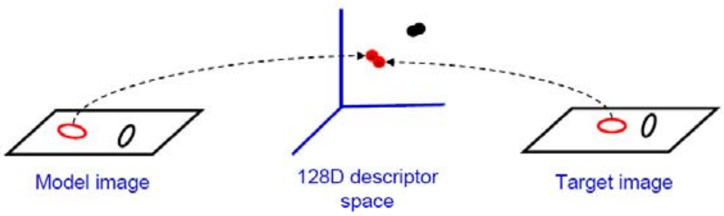
Indexing local features

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



Indexing local features

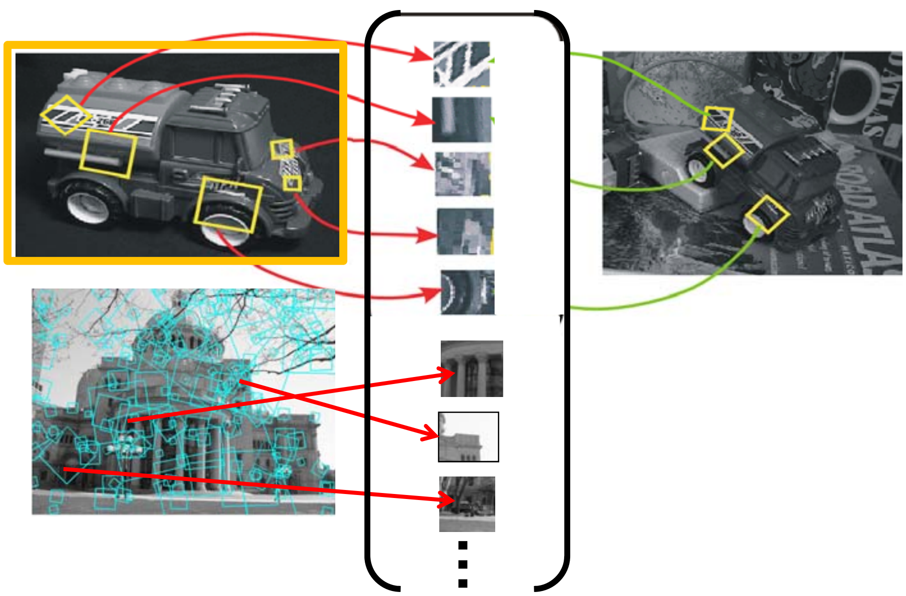
- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



- This is of interest not only for 3d reconstruction, but also for retrieving images of similar objects.

Figure credit: A. Zisserman

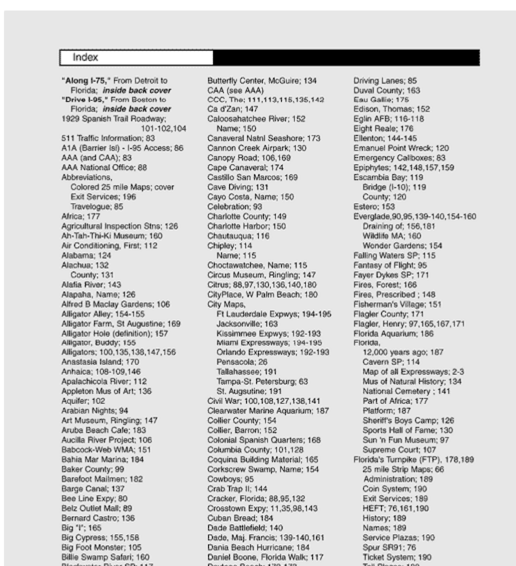
Indexing local features



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; ***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 Mailmen; 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 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
Chipley; 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 Expwys; 194-195
  Jacksonville; 163
  Kissimmee Expwys; 192-193
  Miami Expressways; 194-195
  Orlando Expressways; 192-193
  Pensacola; 26
  Tallahassee; 191
  Tampa-St. Petersburg; 63
  St. Augsutine; 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; 11,35,98,143
Cuban Bread; 184
Dade Battlefield; 140
Dade, Maj. Francis; 139-140,161
Dania Beach Hurricane; 184
Daniel Boone, Florida Walk; 117
Daytona Beach; 172-173
De Land; 87
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 of; 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
  Ticket System; 190
  Toll Plazas; 190
Ford, Henry; 152

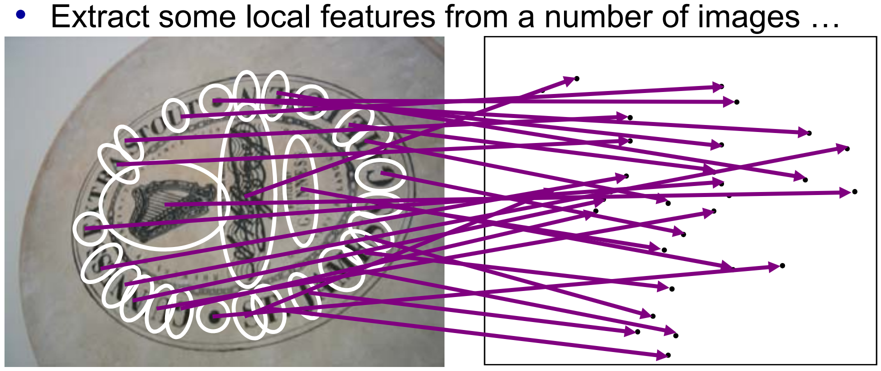
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

Text retrieval vs. image search

- What makes the problems similar, different?

Visual words: main idea

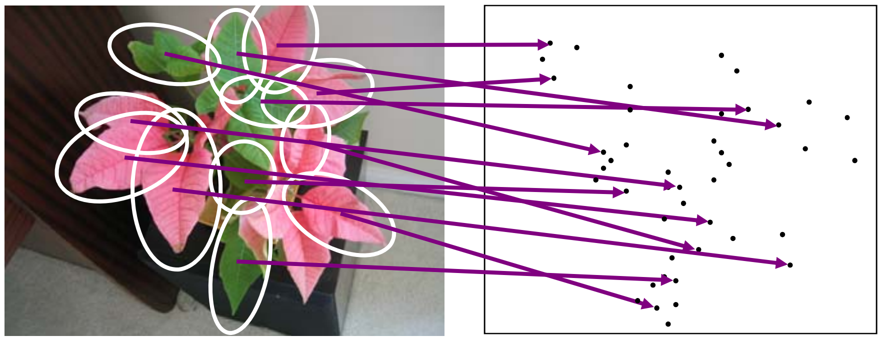
- Extract some local features from a number of images …



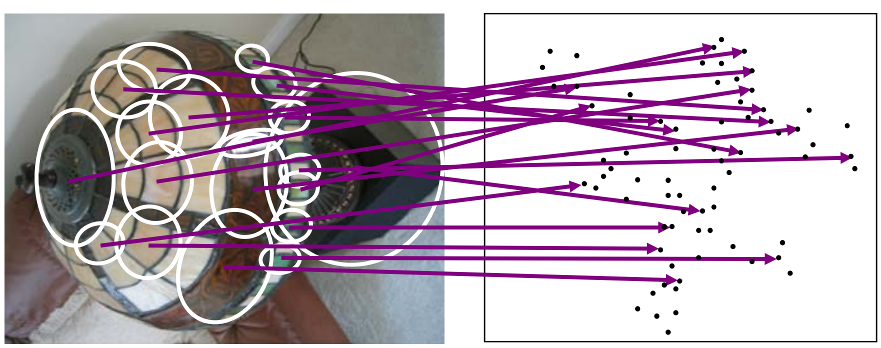
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

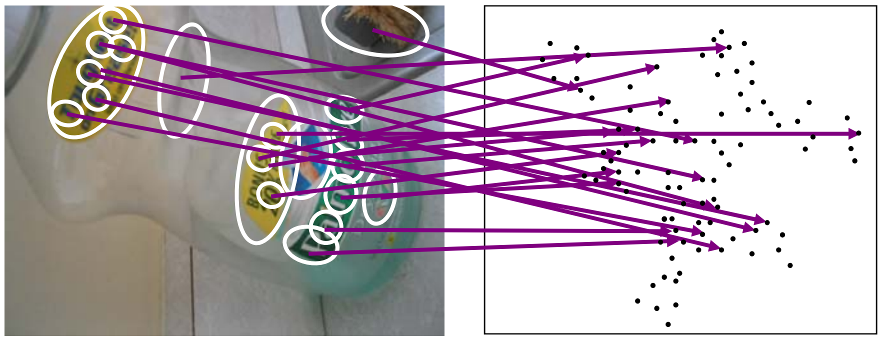
Visual words: main idea

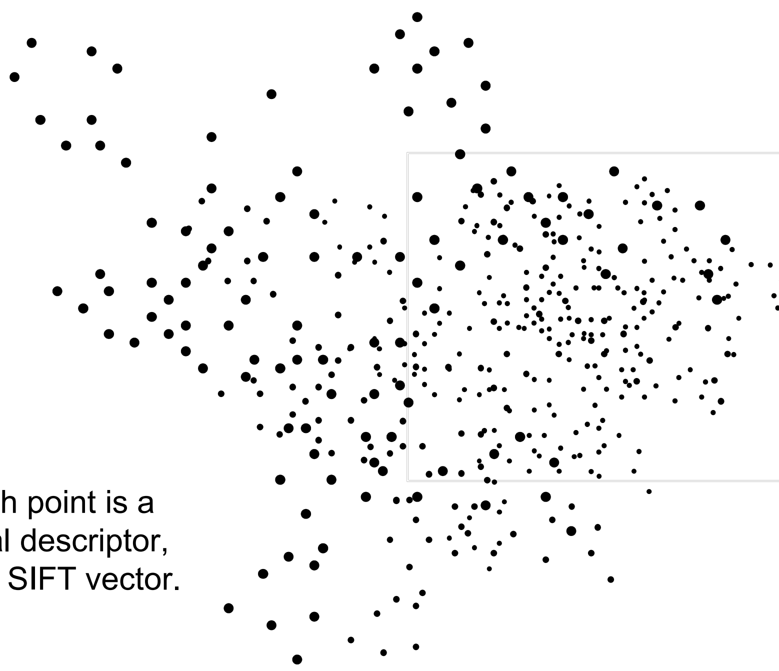


Visual words: main idea

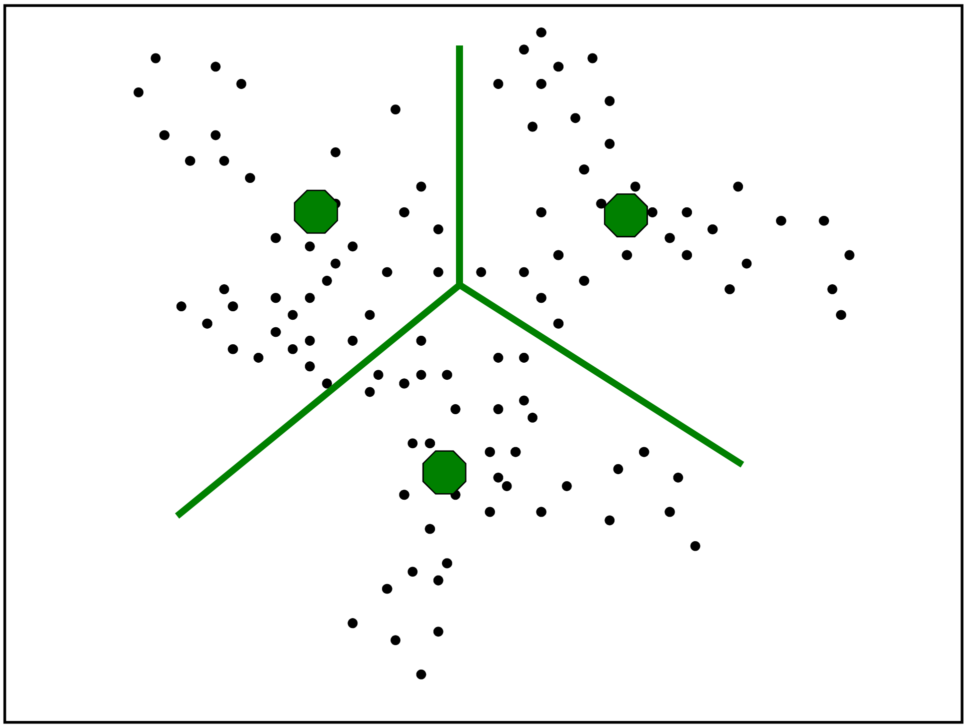


Visual words: main idea



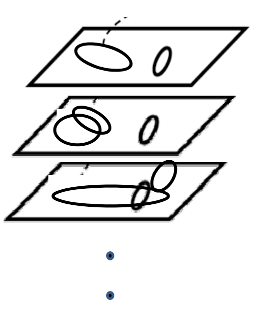


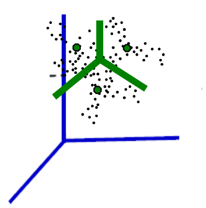
Each point is a local descriptor, e.g. SIFT vector.



Visual words

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

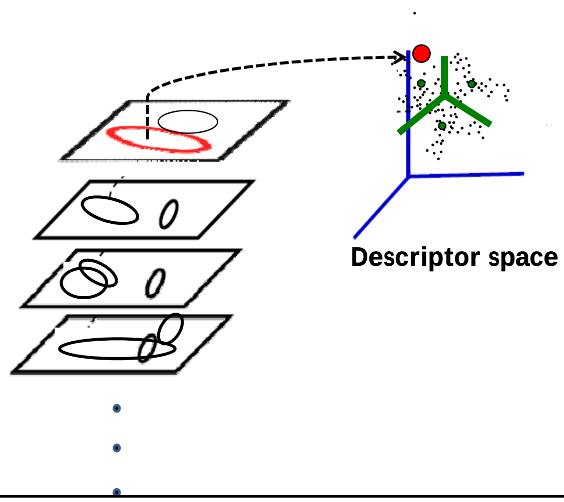




- Quantize via clustering, let cluster centers be the prototype "words"

Visual words

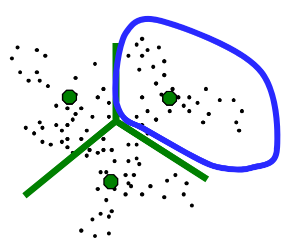
Map high-dimensional descriptors to tokens/words by quantizing the feature space



- 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



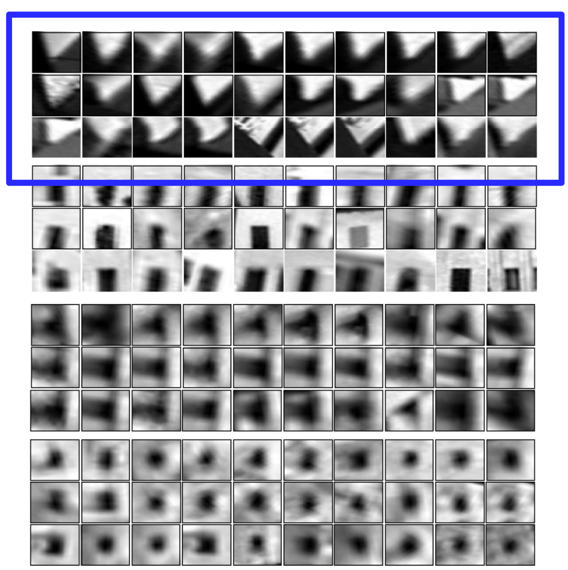
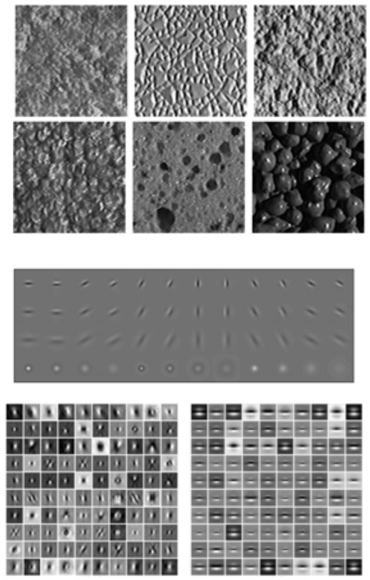


Figure from Sivic & Zisserman, ICCV 2003

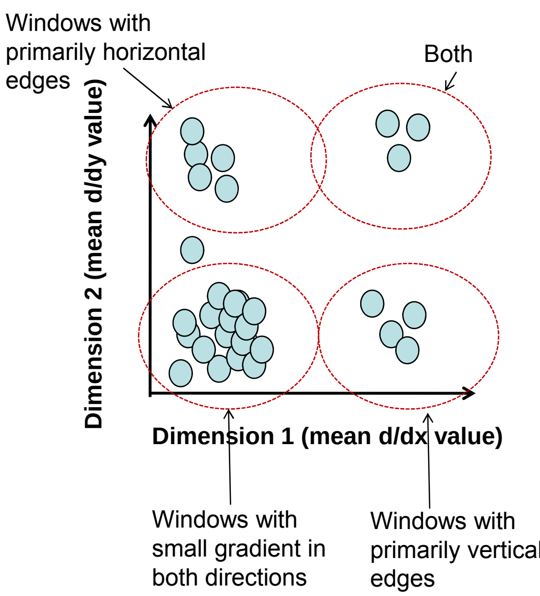
Visual words and textons

- 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; Lazebnik, Schmid & Ponce, 2003;



Recall: Texture representation example



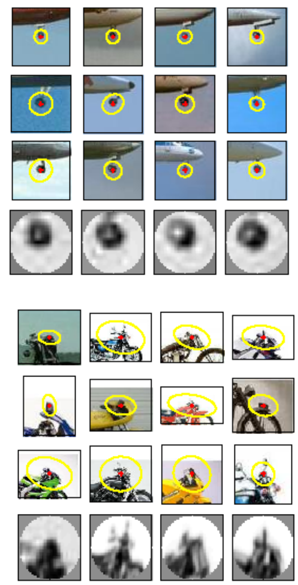
	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

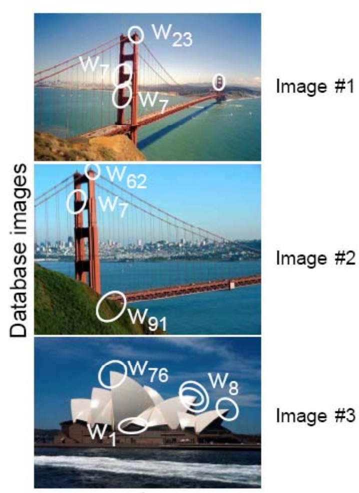
Visual words

- More recently used for describing scenes and objects for the sake of indexing or classification.



Sivic & Zisserman 2003; Csurka, Bray, Dance, & Fan 2004; many others.

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

Inverted file index

New query image

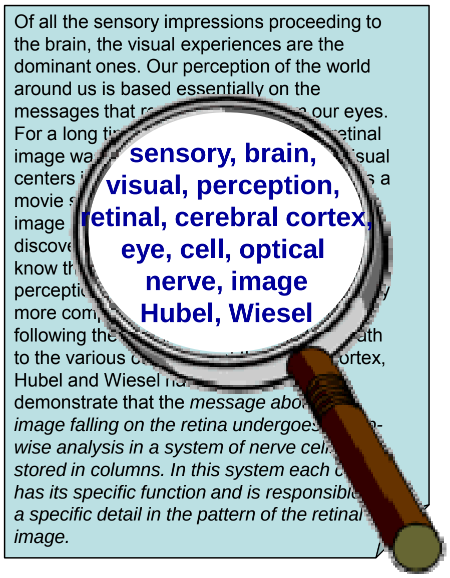
Word #	Image #
1	3
2	
7	1, 2
8	3
9	
10 …	
91	2

…

- 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



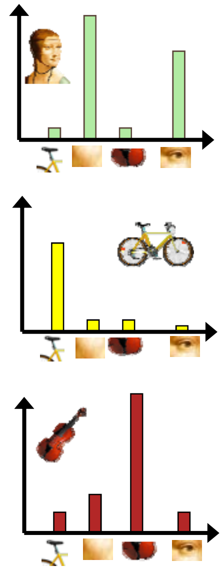
ICCV 2005 short course, L. Fei-Fei

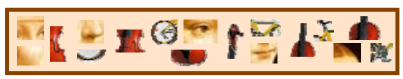
Object → **Bag of ‘words’**

ICCV 2005 short course, L. Fei-Fei

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.

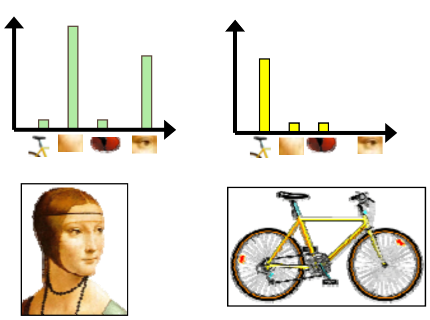




Comparing bags of words

- 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]



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

$$sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$$

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

Number of occurrences of word i in document d

Number of words in document d

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

Total number of documents in database

Number of documents word i occurs in, in whole database

Bags of words for content-based image retrieval

What if query of interest is a portion of a frame?

Visually defined query

"Groundhog Day" [Rammis, 1993]

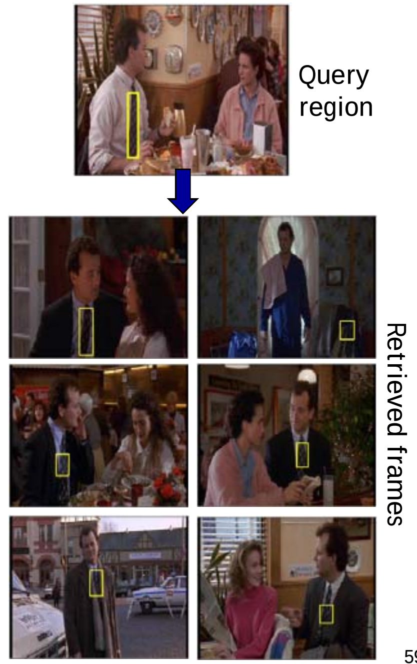
"Find this clock"

"Find this place"

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Video Google System

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



Sivic & Zisserman, ICCV 2003

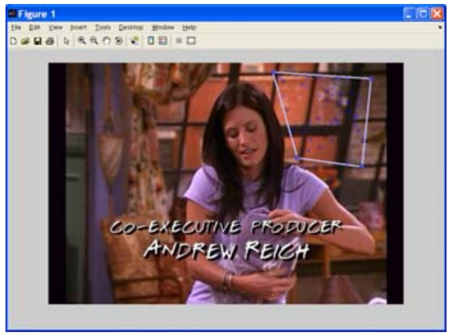
- **Demo online at :** http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html

- Collecting words within a query region

Query region:
pull out only the SIFT descriptors whose positions are within the polygon


raw nn 1sim=0.56697
raw nn 2sim=0.56163
raw nn 5sim=0.54917


raw nn 1sim=0.67818
raw nn 2sim=0.66144
raw nn 3sim=0.66023
raw nn 4sim=0.65774



Bag of words and spatial info

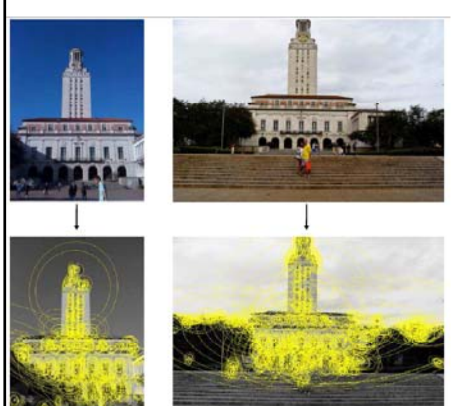
- A bag of words is an orderless representation: throwing out spatial relationships between features
- Middle ground:
  - Visual “phrases” : frequently co-occurring words
  - Semi-local features : describe configuration, neighborhood
  - Let position be part of each feature
  - Count bags of words only within sub-grids of an image
  - After matching, verify spatial consistency (e.g., look at neighbors – are they the same too?)

Visual vocabulary formation

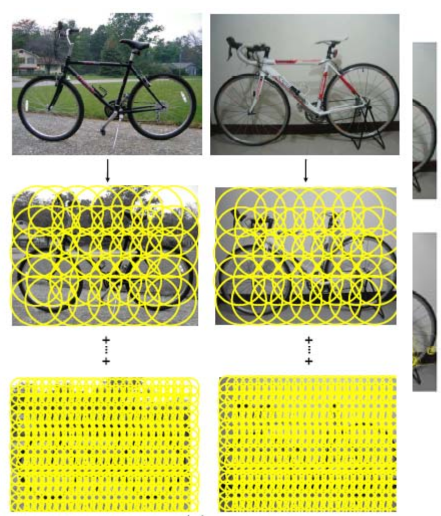
Issues:

- Sampling strategy: where to extract features?

Sampling strategies

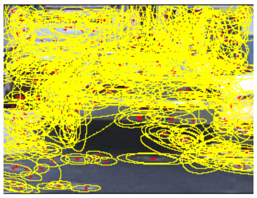


Specific object

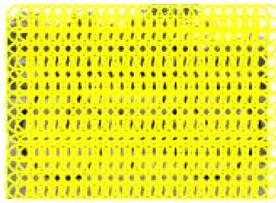


Category

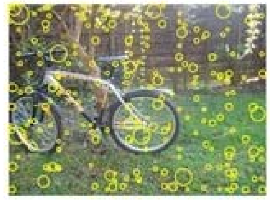
Sampling strategies



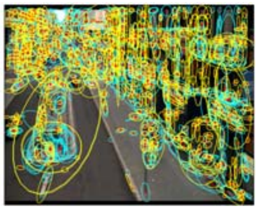
Sparse, at interest points



Dense, uniformly



Randomly



Multiple interest operators

- To find specific, textured objects, sparse sampling from interest points often more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

Image credits: F-F. Li, E. Nowak, J. Sivic

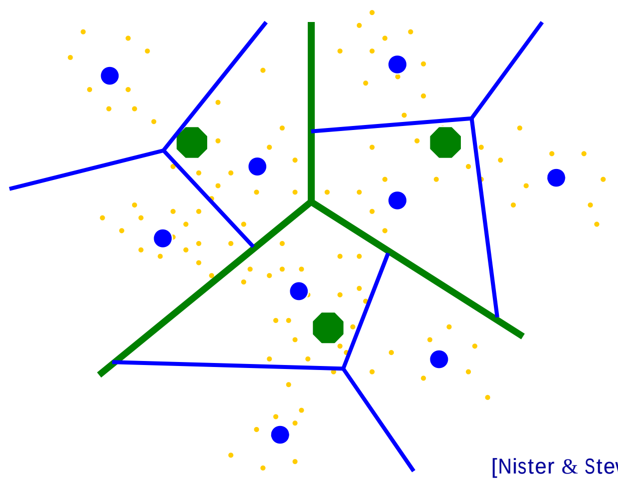
Visual vocabulary formation

Issues:

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

Vocabulary Trees: hierarchical clustering for large vocabularies

- **Tree construction:**



[Nister & Stewenius, CVPR'06]

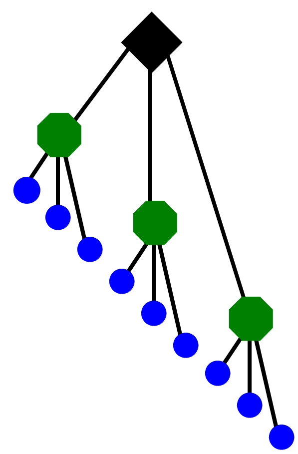
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

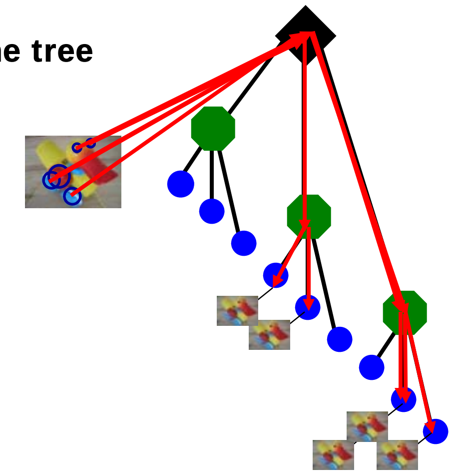
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

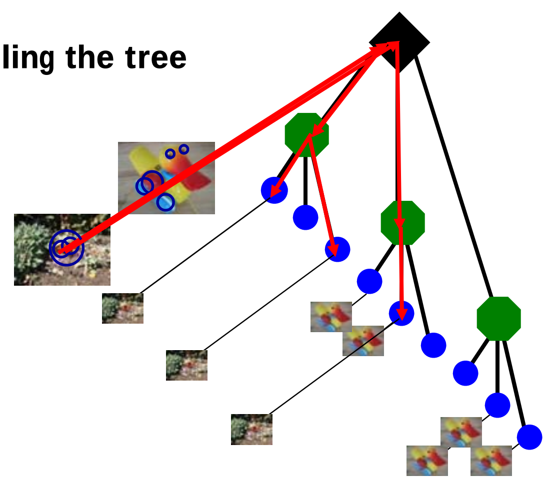
K. Grauman, B. Leibe

Slide credit: David Nister

Visual Object Recognition Tutorial

Vocabulary Tree

- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

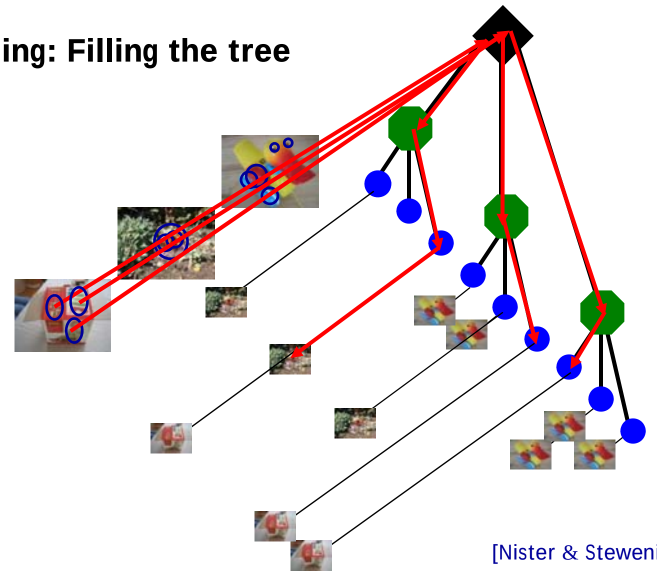
Slide credit: David Nister

Visual Object Recognition Tutorial

Visual Object Recognition Tutorial

Vocabulary Tree

- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

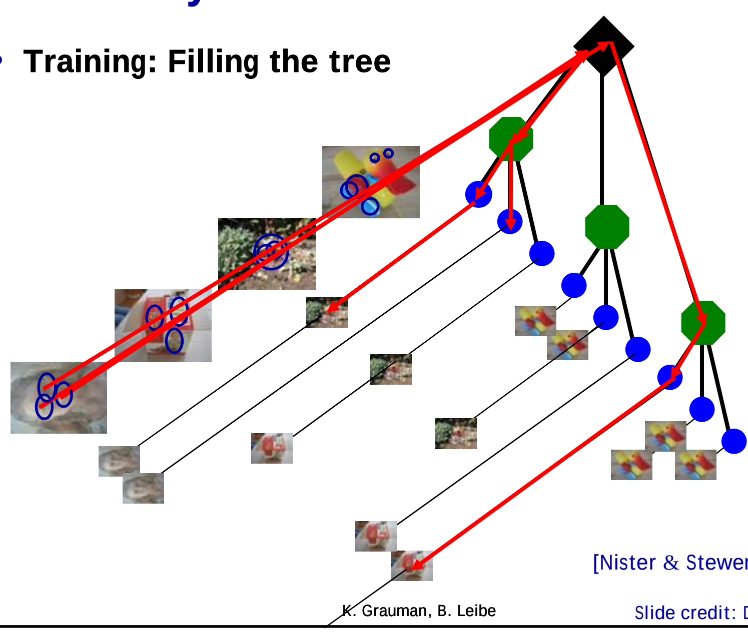
K. Grauman, B. Leibe

Slide credit: David Nister

Visual Object Recognition Tutorial

Vocabulary Tree

- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

74

K. Grauman, B. Leibe

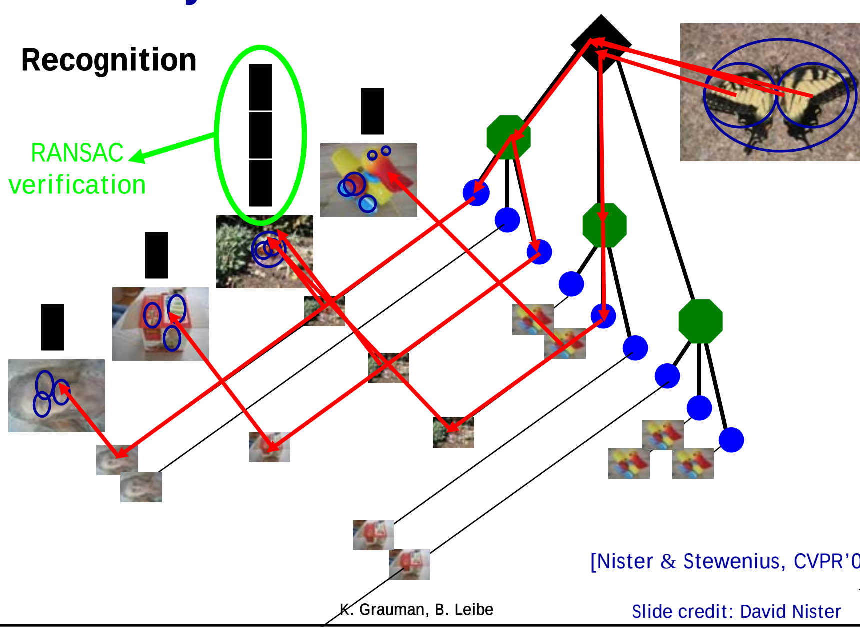
Slide credit: David Nister

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

Visual Object Recognition Tutorial

Vocabulary Tree

- **Recognition**



[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

Slide credit: David Nister

76

Bags of words: pros and cons

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

- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- interest points or sampling: no guarantee to capture object-level parts
- optimal vocabulary formation remains unclear

Summary

- **Local invariant features**: distinctive matches possible in spite of significant view change, useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- To find **correspondences** among detected features, measure distance between descriptors, and look for most similar patches.
- **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