



# Indexing with local features, Bag of words models

Thursday, Oct 30

Kristen Grauman

UT-Austin

# Today

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

# Main questions

- Where will the interest points come from?
  - What are salient features that we'll *detect* in multiple views?
- How to *describe* a local region?
- How to establish *correspondences*, i.e., compute matches?

# Last time: Local invariant features

- Problem 1:
  - Detect the *same* point *independently* in both images



no chance to match!

We need a repeatable detector

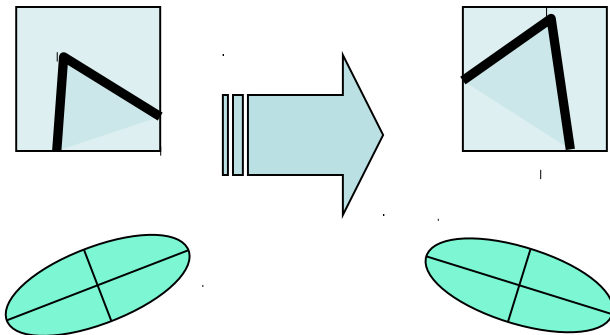
# Harris corner detector: rotation invariant detection

- Algorithm steps:
  - Compute  $M$  matrix within all image windows to get their  $R$  scores
  - Find points with large corner response  $R >$  threshold)
  - Take the points of local maxima of  $R$



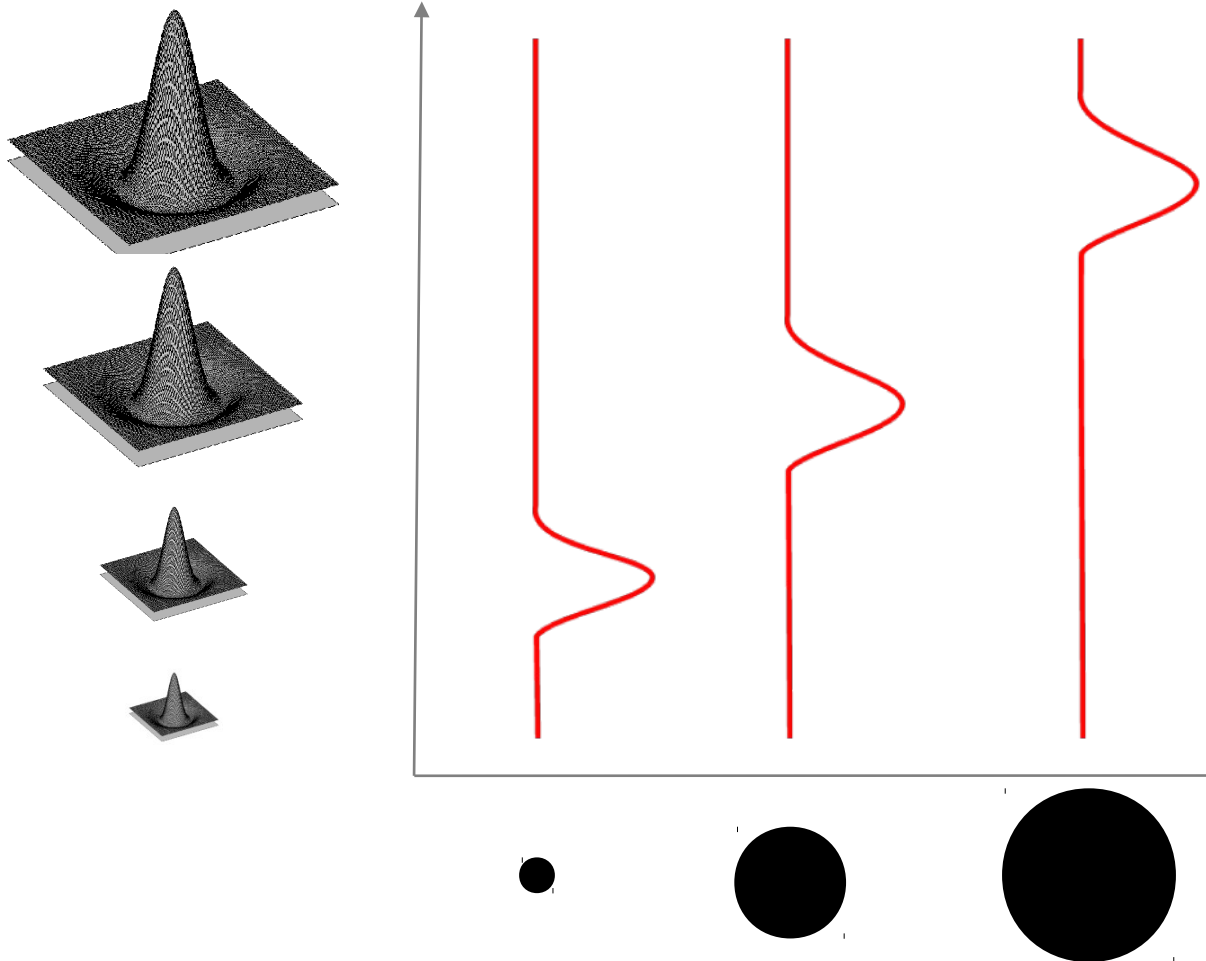
Corner response  $R$  is invariant to image rotation.

Ellipse rotates but its shape (i.e. eigenvalues) remains the same.



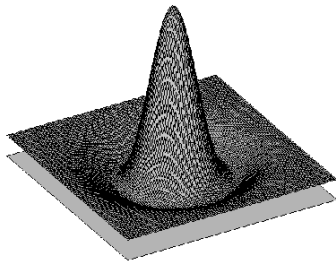
# Laplacian of Gaussian: scale invariant detection

- Laplacian-of-Gaussian = “blob” detector

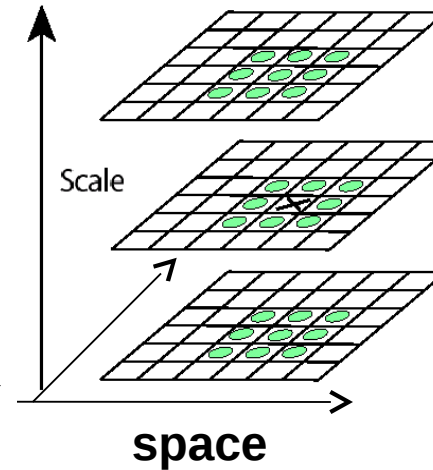
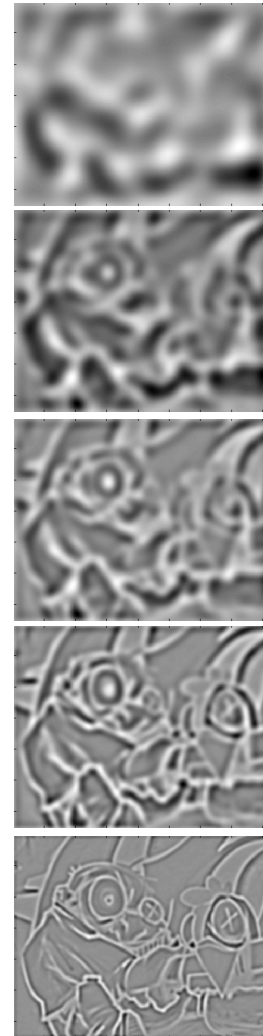


# Laplacian of Gaussian: scale invariant detection

- Interest points:  
Local maxima in scale space of Laplacian-of-Gaussian

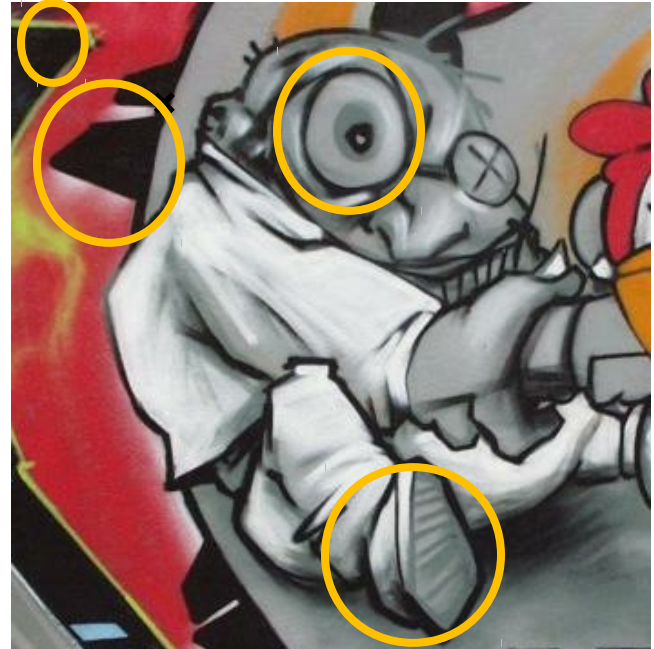
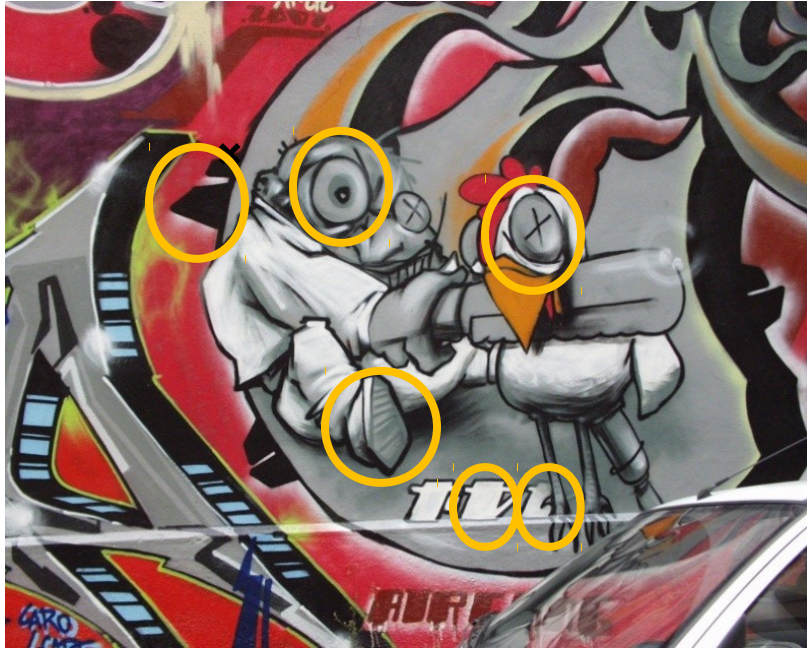


$$L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow \begin{matrix} \nearrow \sigma^5 \\ \nearrow \sigma^4 \\ \rightarrow \sigma^3 \\ \searrow \sigma^2 \\ \searrow \sigma \end{matrix}$$



**$\Rightarrow$  List of  
 $(x, y, \sigma)$**

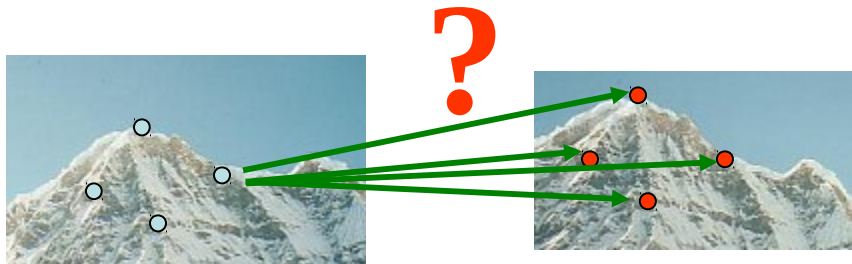
# Laplacian of Gaussian: scale invariant detection





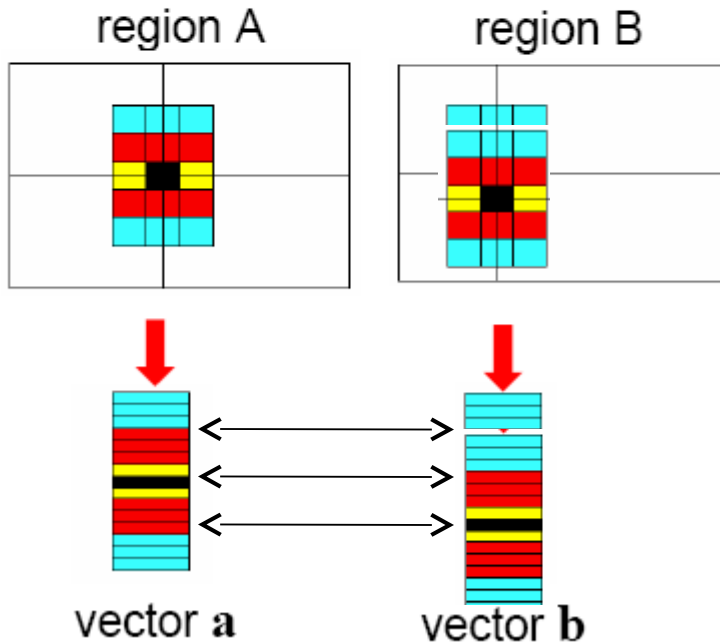
# Last time: Local invariant features

- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

# Raw patches as local descriptors

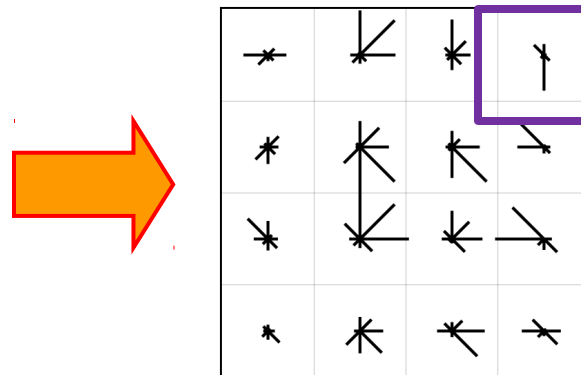
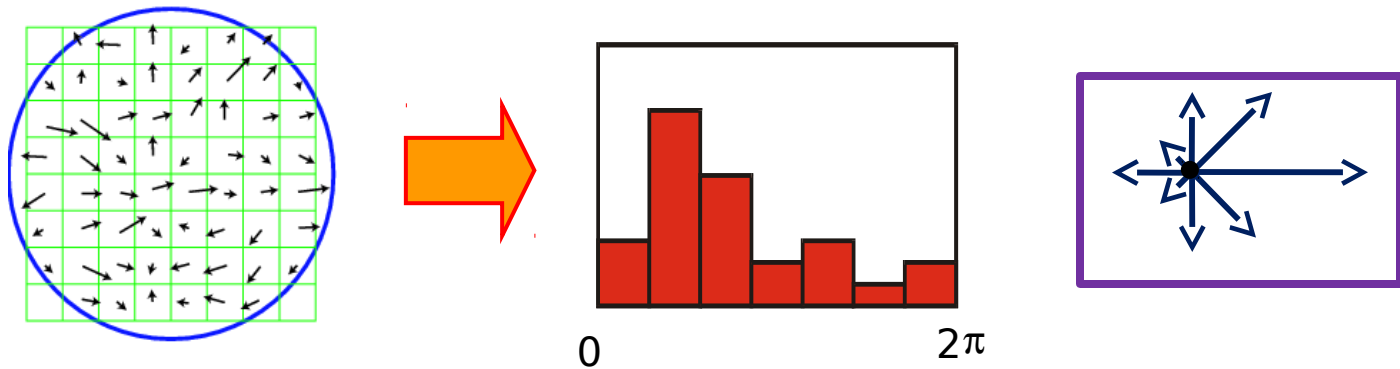


The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

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

# SIFT descriptors [Lowe 2004]

- More robust way to describe the neighborhood: use histograms to bin pixels within sub-patches according to their orientation.



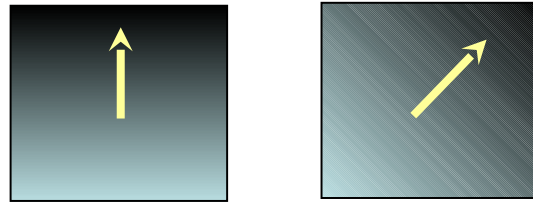
*Why subpatches?*

*Why does SIFT have some illumination invariance?*

# Rotation invariant descriptors

- Find local orientation

Dominant direction of gradient for the image patch



- Rotate patch according to this angle

This puts the patches into a canonical orientation.

# Feature descriptors: SIFT

---

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](http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT)



# Interest points + descriptors

- So far we have methods to find interest points and describe the surrounding image neighborhood.
- This will map each image to a list of local descriptors.



$\left( \begin{array}{c} X_1, \\ X_2, \\ \dots \\ X_{128} \end{array} \right)$

- *How many detections will an image have?*

# Many Existing Detectors Available

- **Hessian & Harris** [Harris '88] [Beaudet '78],
- **Laplacian, DoG** [Lowe 1999] [Lindeberg '98],
- **Harris-/Hessian-Laplace** Schmid '01] [Mikolajczyk &
- **Harris-/Hessian-Affine** Schmid '04] [Mikolajczyk &
- **EBR and IBR** '04] [Tuytelaars & Van Gool
- **MSER** [Matas '02]
- **Salient Regions** [Kadir & Brady '01]
- **Others...**

# You Can Try It At Home...

- **For most local feature detectors, executables are available online:**
- **<http://robots.ox.ac.uk/~vgg/research/affine>**
- **<http://www.cs.ubc.ca/~lowe/keypoints/>**
- **<http://www.vision.ee.ethz.ch/~surf>**



# Affine Covariant Features



KATHOLIEKE UNIVERSITEIT  
**LEUVEN**

**INRIA**  
RHÔNE-ALPES



Collaborative work between: the Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception.

## Affine Covariant Region Detectors

Input image



Region detector

Detector output

```
format:
-----
1.0
m
u1 v1 a1 b1 c1
:
um vm am bm cm

output example:
img1.haraff
```



display\_features.m

Image with displayed regions



### Parameters defining an affine region

$u, v, a, b, c$  in  $a(x-u)^2 + 2b(x-u)(y-v) + c(y-v)^2 = 1$   
with  $(0, 0)$  at image top left corner

### Code

- provided by the authors, see [publications](#) for details and links to authors web sites.

#### Linux binaries

[Harris-Affine & Hessian-Affine](#)

[MSER](#) - Maximally stable extremal regions (also Windows)

[IBR](#) - Intensity extrema based detector

[EBR](#) - Edge based detector

[Salient](#) region detector

#### Example of use

```
prompt>./h_affine.ln -haraff -i img1.ppm -o img1.haraff -thres 1000
```

```
prompt>./h_affine.ln -hesaff -i img1.ppm -o img1.hesaff -thres 500
```

```
prompt>./mser.ln -t 2 -es 2 -i img1.ppm -o img1.mser
```

```
prompt>./ibr.ln img1.ppm img1.ibr -scalefactor 1.0
```

```
prompt> ./ebr.ln img1.ppm img1.ebr
```

```
prompt>./salient.ln img1.ppm img1.sal
```

#### Displaying i

```
matlab>> d
```

```
matlab>> d
```

```
matlab>> d
```

```
matlab>> d
```

```
matlab>> d
```

```
matlab>> d
```

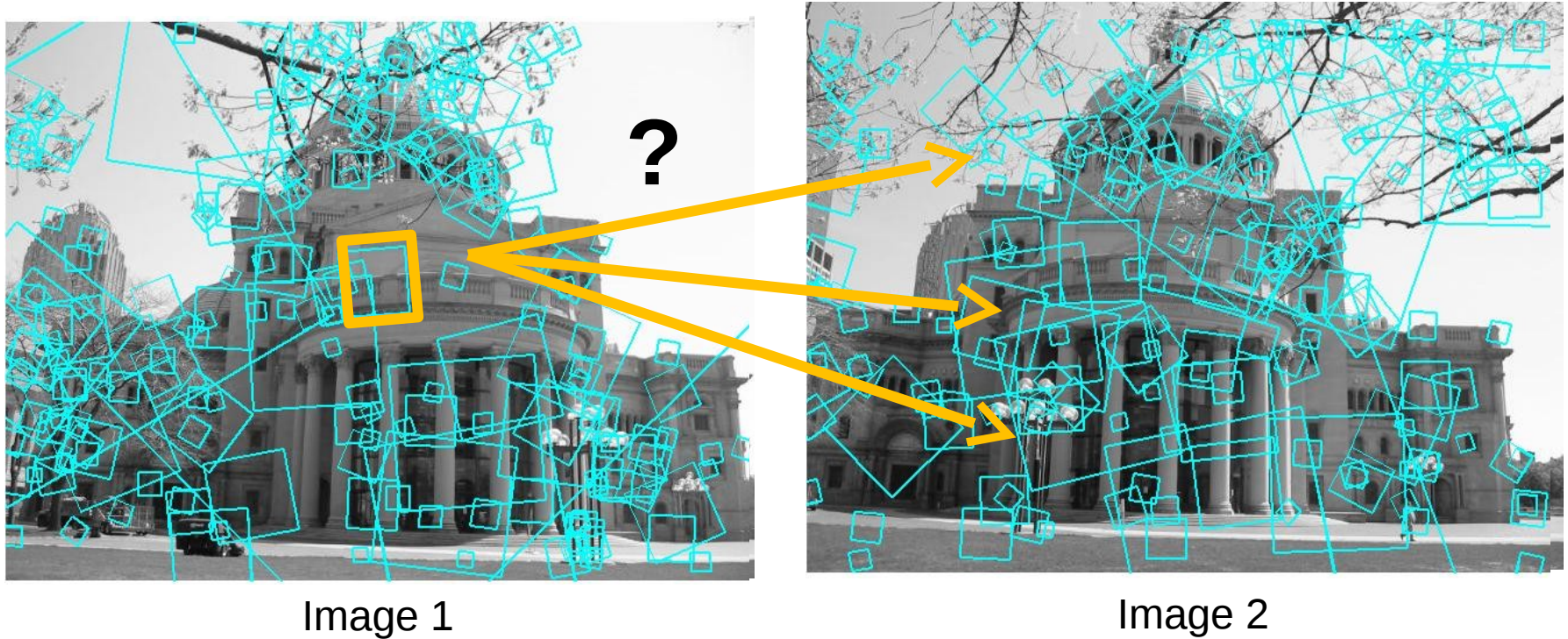
# Main questions

- Where will the interest points come from?
  - What are salient features that we'll *detect* in multiple views?
- How to *describe* a local region?
- How to establish *correspondences*, i.e., compute matches?

# Matching local features



# Matching local features



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, first match looks good.

# Applications of local invariant features & matching

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
  - Specific objects
  - Textures
  - Categories
- ...

# Wide baseline stereo



[Image from T. Tuytelaars ECCV 2006 tutorial]



# Panorama stitching



(a) Matier data set (7 images)



(b) Matier final stitch

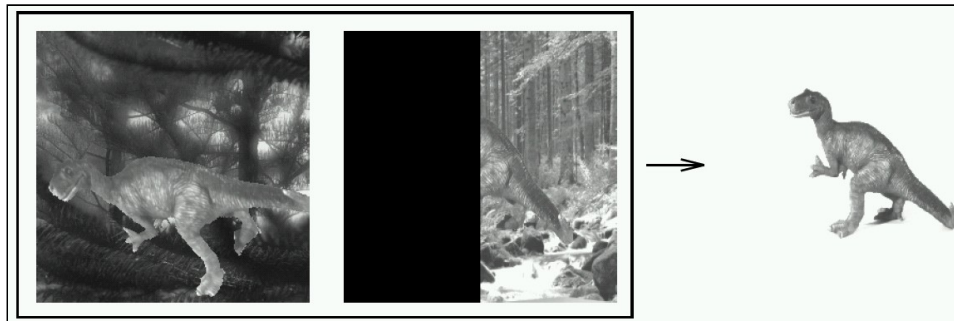
Brown, Szeliski, and Winder, 2005

# Automatic mosaicing



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

# Recognition of specific objects, scenes



Schmid and Mohr 1997



Sivic and Zisserman, 2003



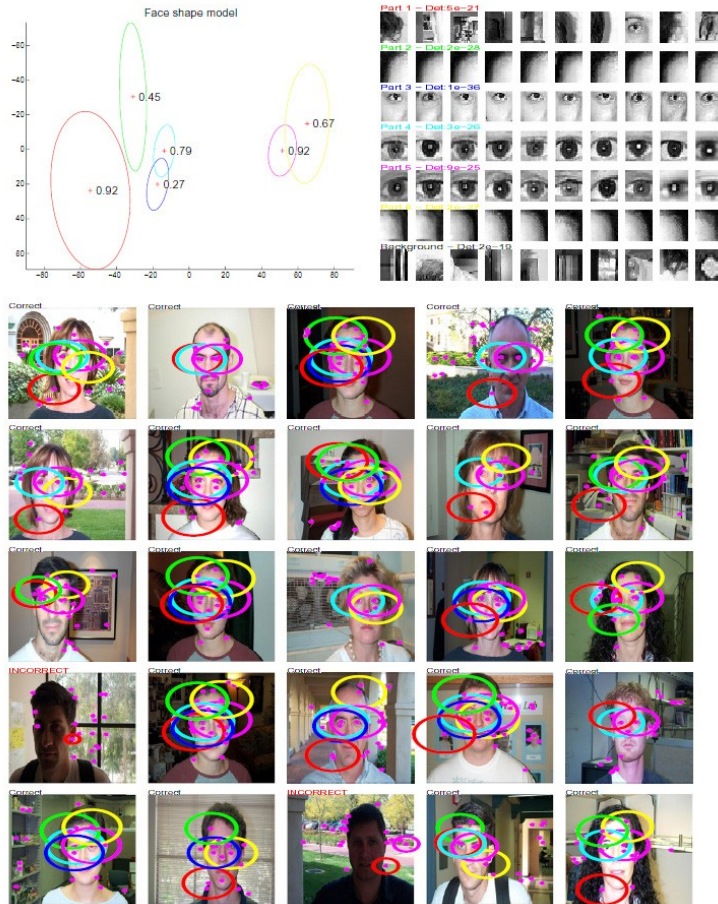
Rothganger et al. 2003



Lowe 2002

# Recognition of categories

## Constellation model



Weber et al. (2000)  
Fergus et al. (2003)

## Bags of words

Database	Sample cluster #1	Sample cluster #2
Airplanes		
Motorbikes		
Leaves		
Wild Cats		
Faces		
Bicycles		
People		

Csurka et al. (2004)  
Dorko & Schmid (2005)  
Sivic et al. (2005)  
Lazebnik et al. (2006), ...

# Value of local features

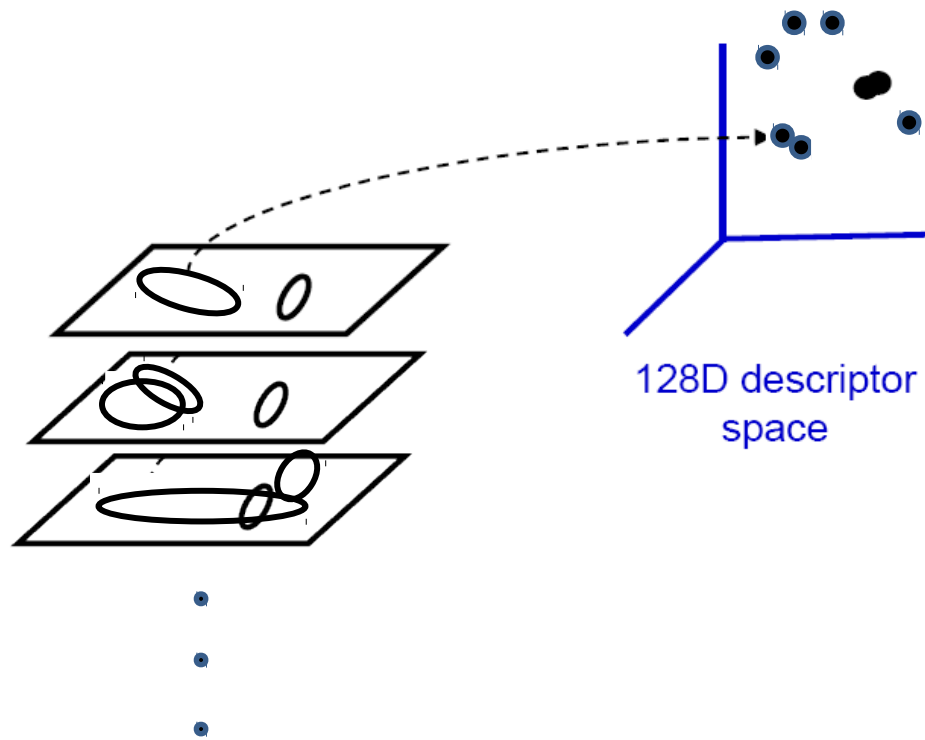
- Critical to find distinctive and repeatable local regions for multi-view matching
- Complexity reduction via selection of distinctive points
- Describe images, objects, parts without requiring segmentation; robustness to clutter & occlusion
- Robustness: similar descriptors in spite of moderate view changes, noise, blur, etc.

# Today

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

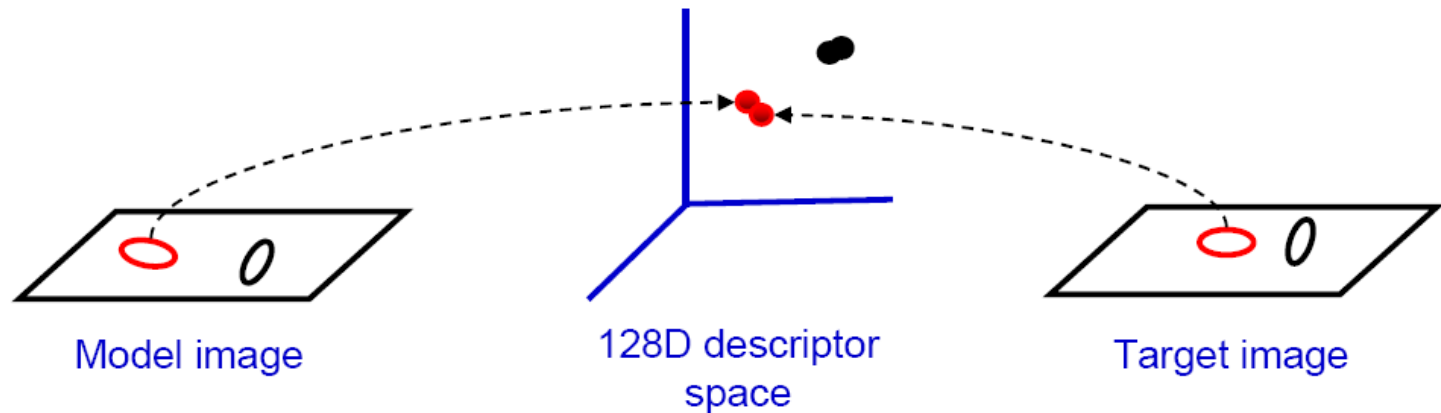
# Indexing local features

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



# Indexing local features

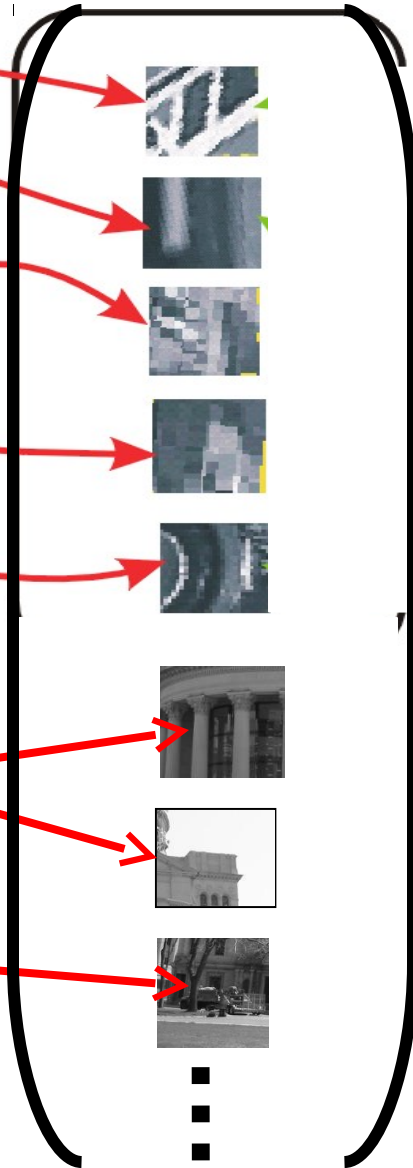
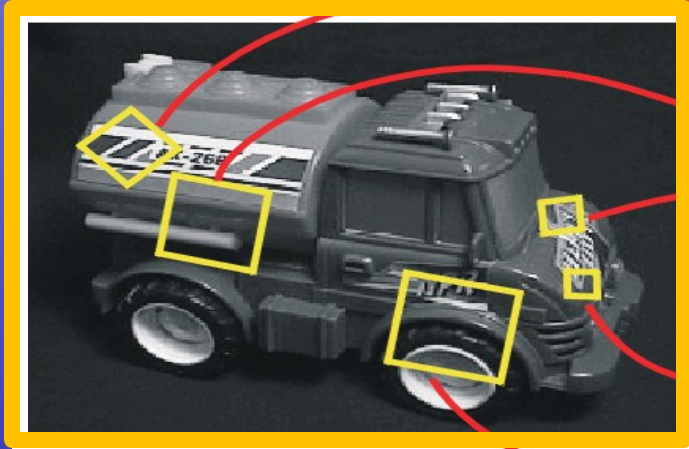
- **When we see close points in feature space, we have similar descriptors, which indicates similar local content.**



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



# 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?**
  - **Low-dimensional descriptors : can use standard efficient data structures for nearest neighbor search**
  - **High-dimensional descriptors: approximate nearest neighbor search methods more practical**
  - **Inverted file indexing schemes**

# Indexing local features: inverted file index

Index	
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142
511 Traffic Information; 83	Ca d'Zan; 147
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152
AAA (and CAA); 83	Name; 150
AAA National Office; 88	Canaveral Natnl Seashore; 173
Abbreviations,	Cannon Creek Airpark; 130
Colored 25 mile Maps; cover	Canopy Road; 106,169
Exit Services; 196	Cape Canaveral; 174
Travelogue; 85	Castillo San Marcos; 169
Africa; 177	Cave Diving; 131
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93
Air Conditioning, First; 112	Charlotte County; 149
Alabama; 124	Charlotte Harbor; 150
Alachua; 132	Chautauqua; 116
County; 131	ChIPLEY; 114
Alafia River; 143	Name; 115
Alapaha, Name; 126	Choctawatchee, Name; 115
Alfred B Maclay Gardens; 106	Circus Museum, Ringling; 147
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180
Alligator Hole (definition); 157	City Maps,
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195
Alligators; 100,135,138,147,156	Jacksonville; 163
Anastasia Island; 170	Kissimmee Expwys; 192-193
Anhaica; 108-109,146	Miami Expressways; 194-195
Apalachicola River; 112	Orlando Expressways; 192-193
Appleton Mus of Art; 136	Pensacola; 26
Aquifer; 102	Tallahassee; 191
Arabian Nights; 94	Tampa-St. Petersburg; 63
Art Museum, Ringling; 147	St. Augustine; 191
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141
Aucilla River Project; 106	Clearwater Marine Aquarium; 187
Babcock-Web WMA; 151	Collier County; 154
Bahia Mar Marina; 184	Collier, Barron; 152
Baker County; 99	Colonial Spanish Quarters; 168
Barefoot Mailmen; 182	Columbia County; 101,128
Barge Canal; 137	Coquina Building Material; 165
Bee Line Expy; 80	Corkscrew Swamp, Name; 154
Belz Outlet Mall; 89	Cowboys; 95
Bernard Castro; 136	Crab Trap II; 144
Big "I"; 165	Cracker, Florida; 88,95,132
Big Cypress; 155,158	Crosstown Expy; 11,35,98,143
Big Foot Monster; 105	Cuban Bread; 184
Billie Swamp Safari; 160	Dade Battlefield; 140
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161
Blue Angels	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; 168
	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; 76,181,190
	History; 189
	Names; 189
	Service Plazas; 190
	Spur SR91; 76
	Ticket System; 190
	Toll Plazas; 190
	Ford, Henry; 152

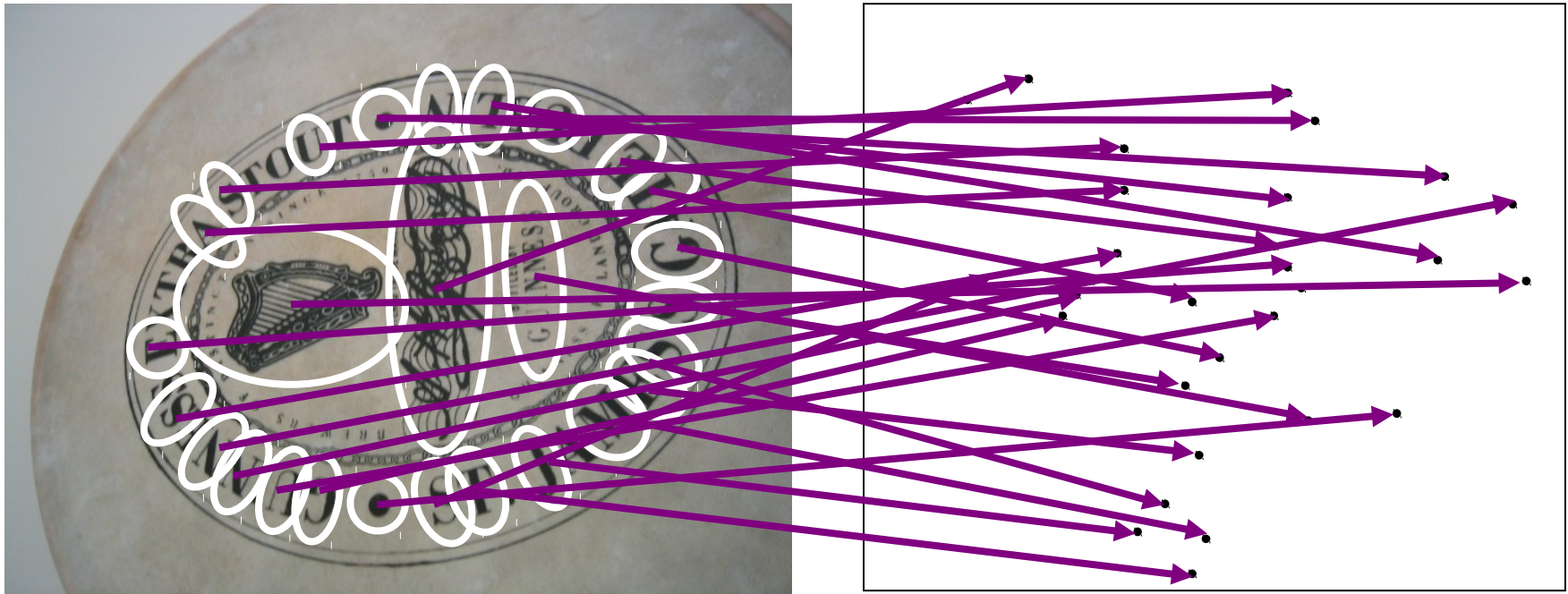
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features

# Text retrieval vs. image search

- What makes the problems similar, different?

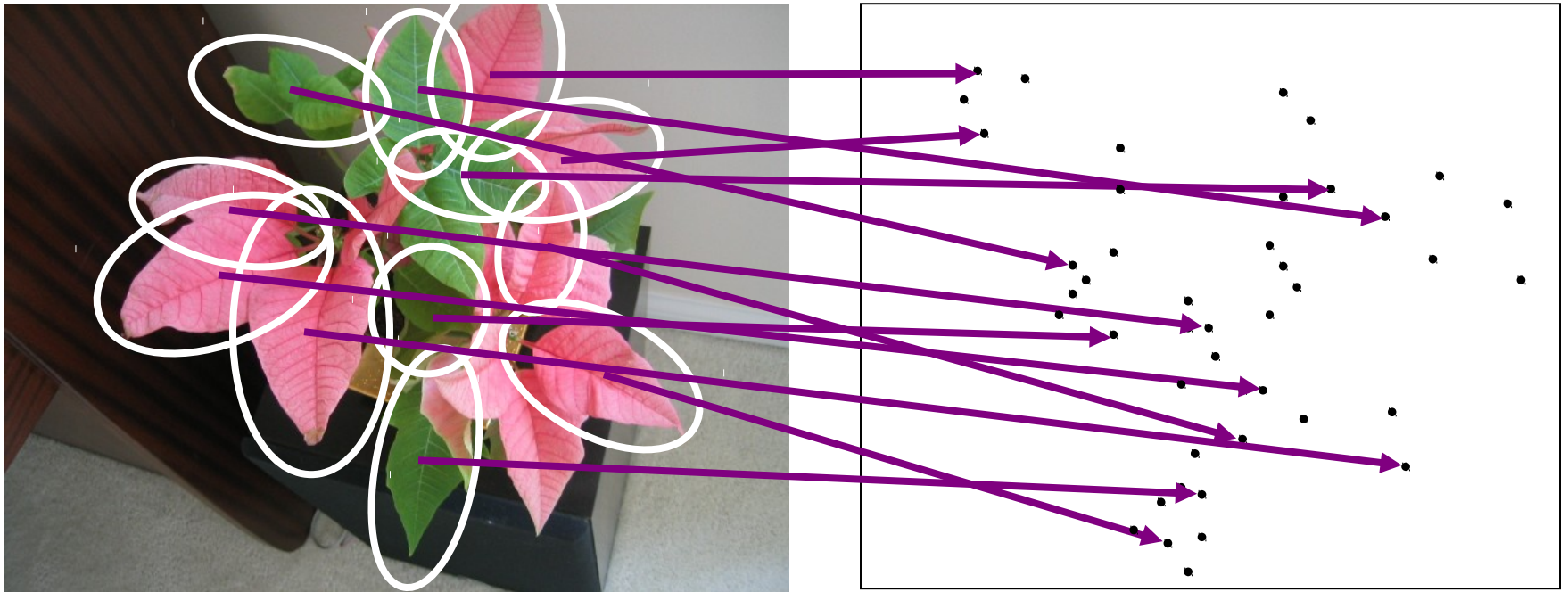
# Visual words: main idea

- Extract some local features from a number of

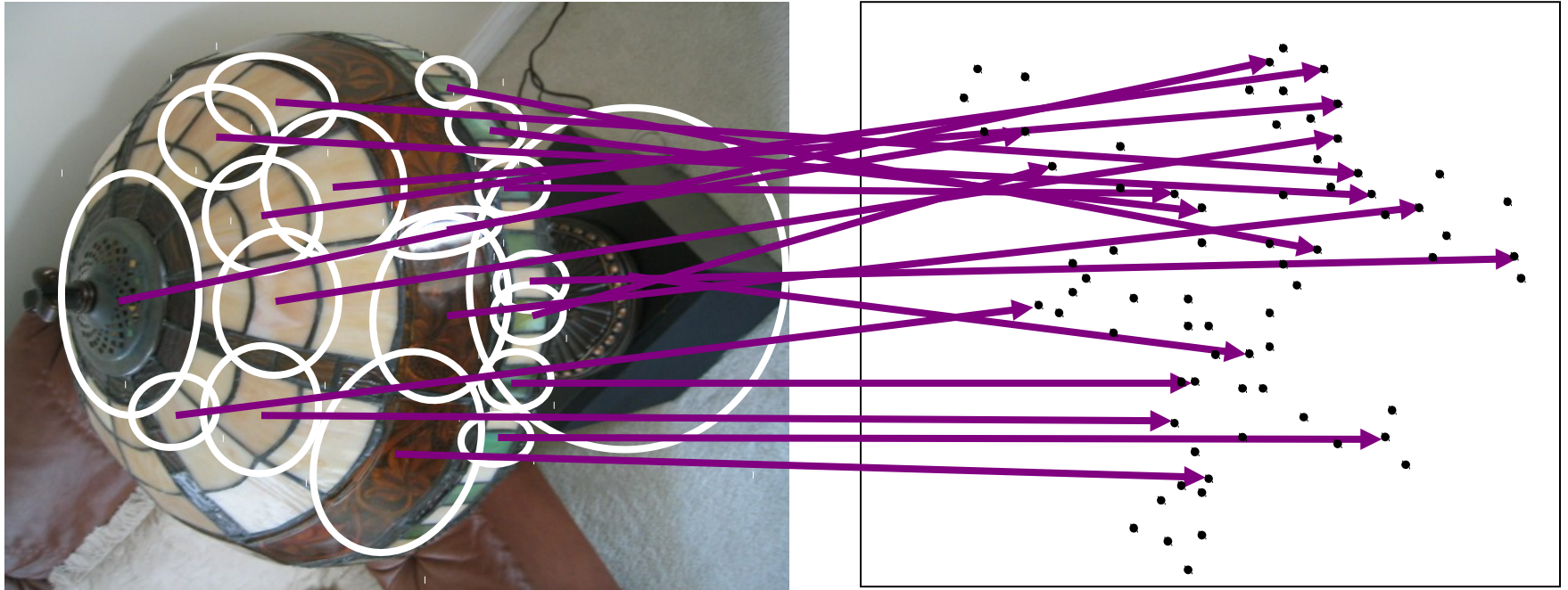


e.g., SIFT descriptor space:  
each point is 128-dimensional

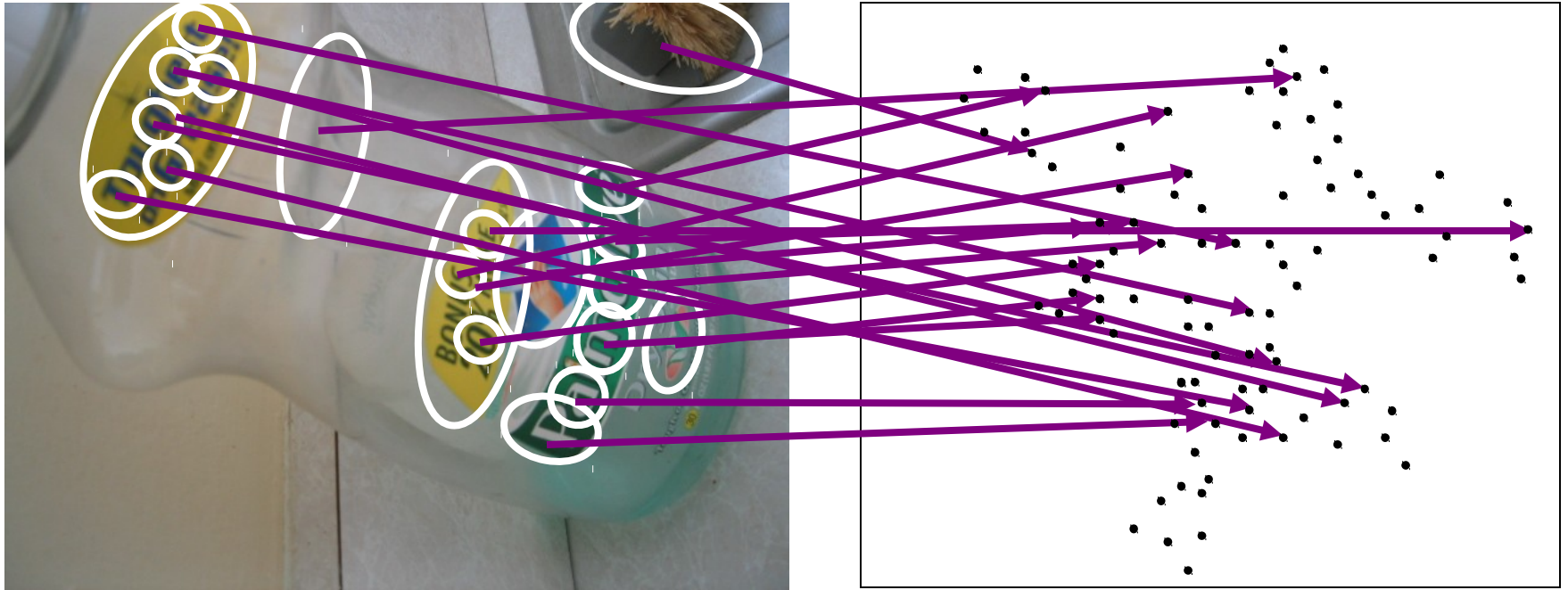
# Visual words: main idea



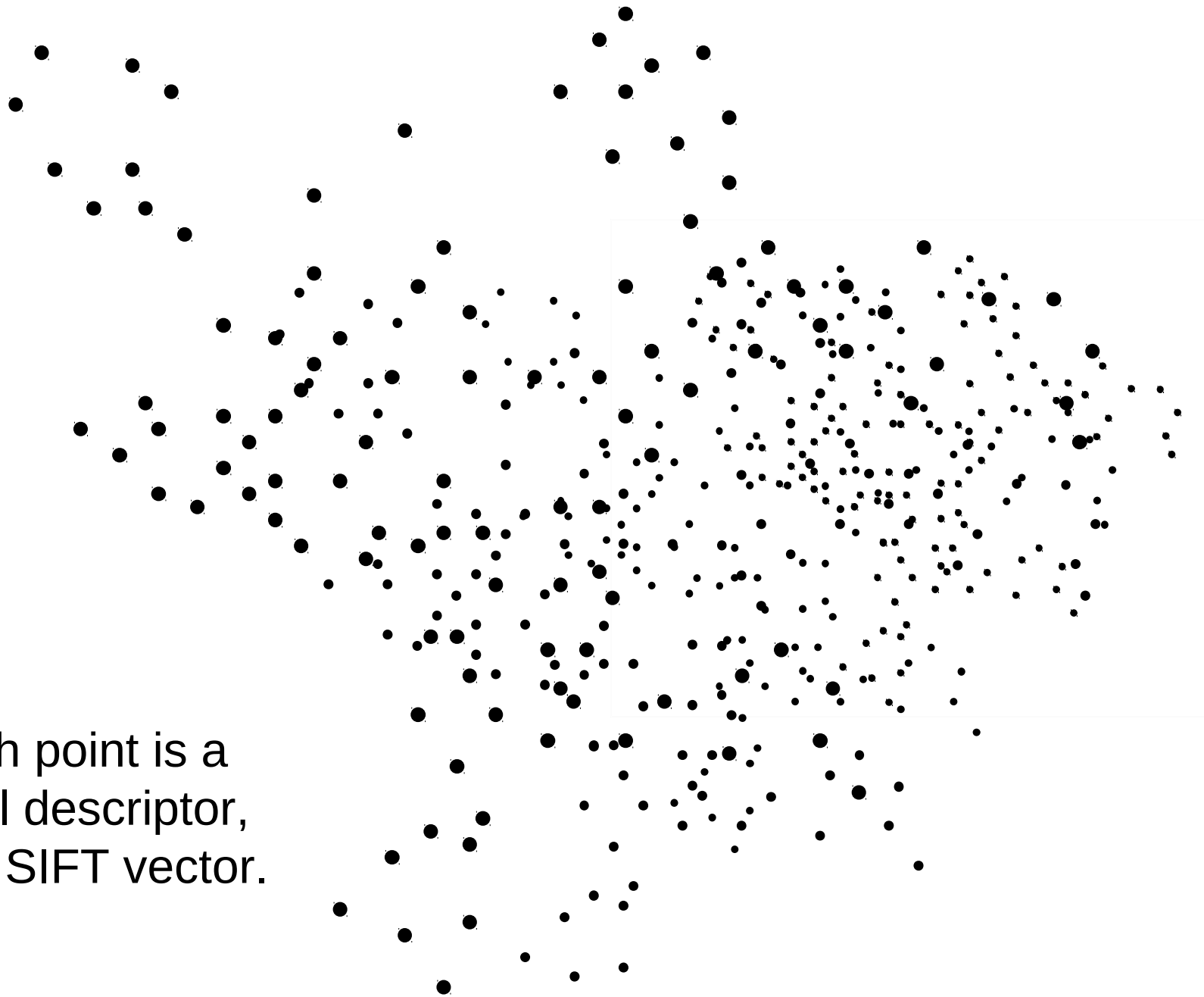
# Visual words: main idea



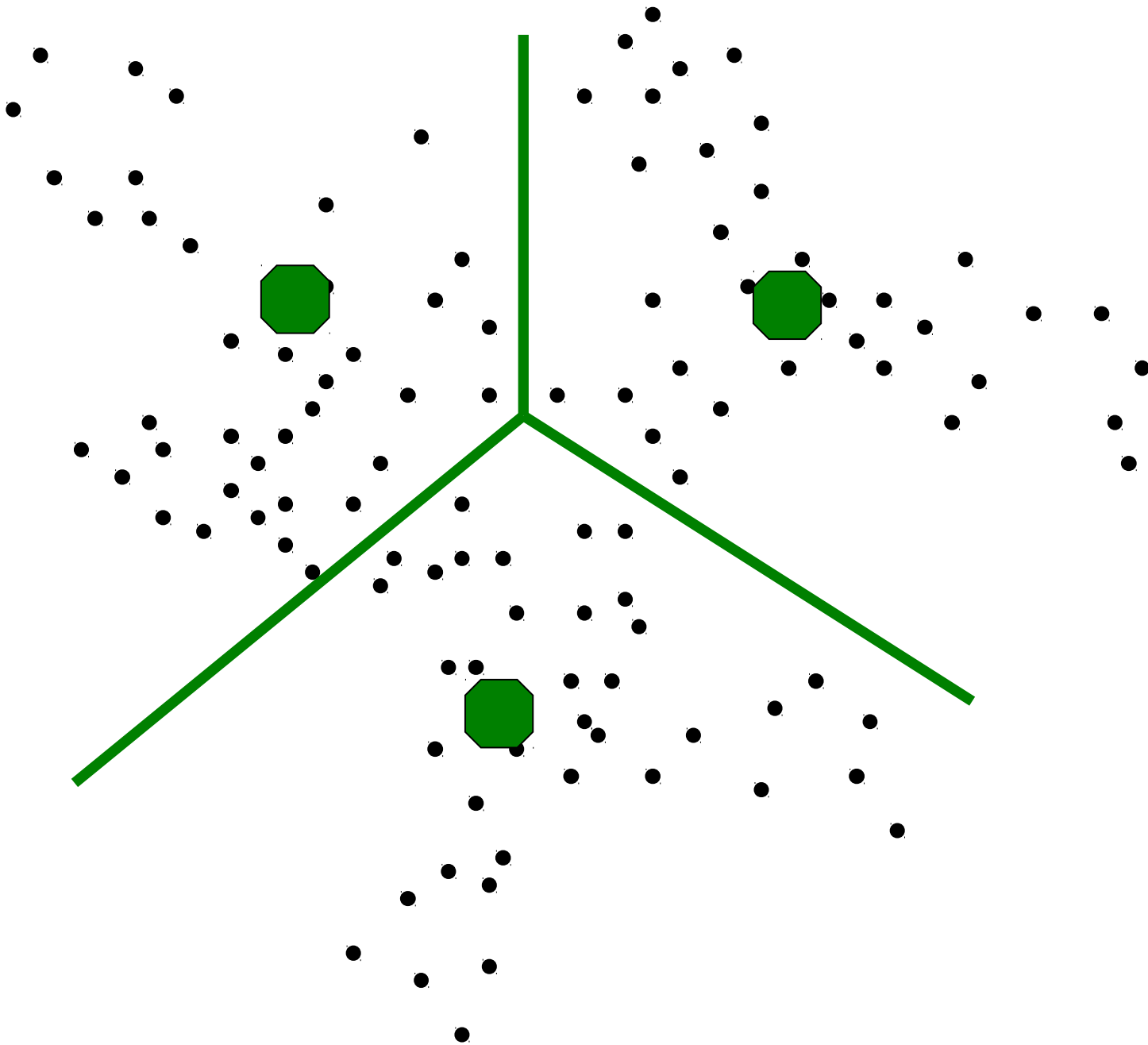
# Visual words: main idea







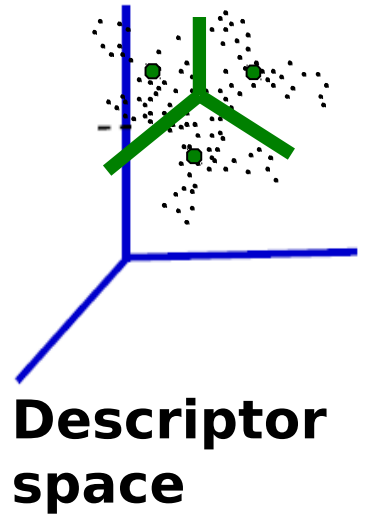
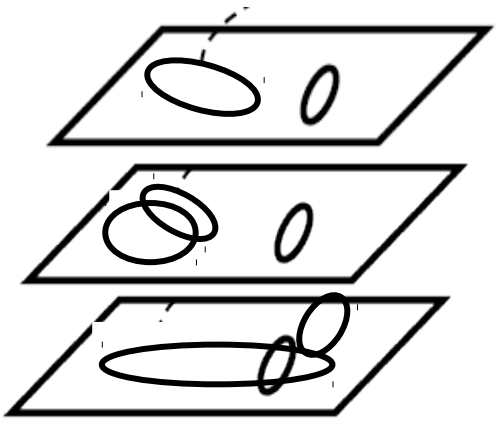
Each point is a local descriptor, e.g. SIFT vector.



# Visual words: main idea

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

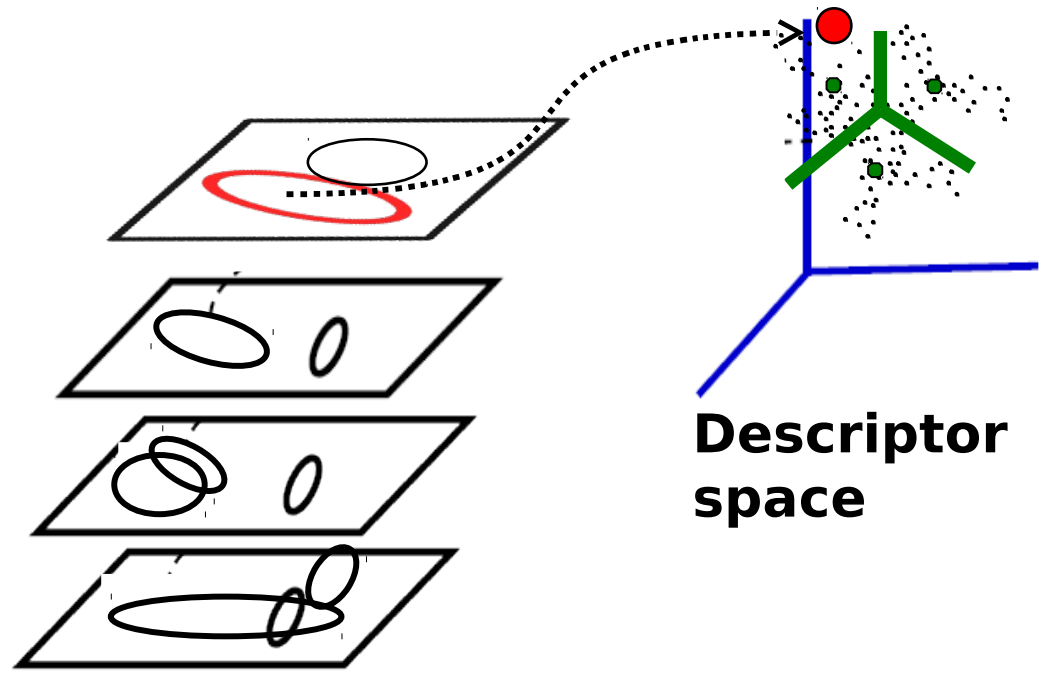
- Quantize via clustering, let cluster centers be the prototype "words"



# Visual words: main idea

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**

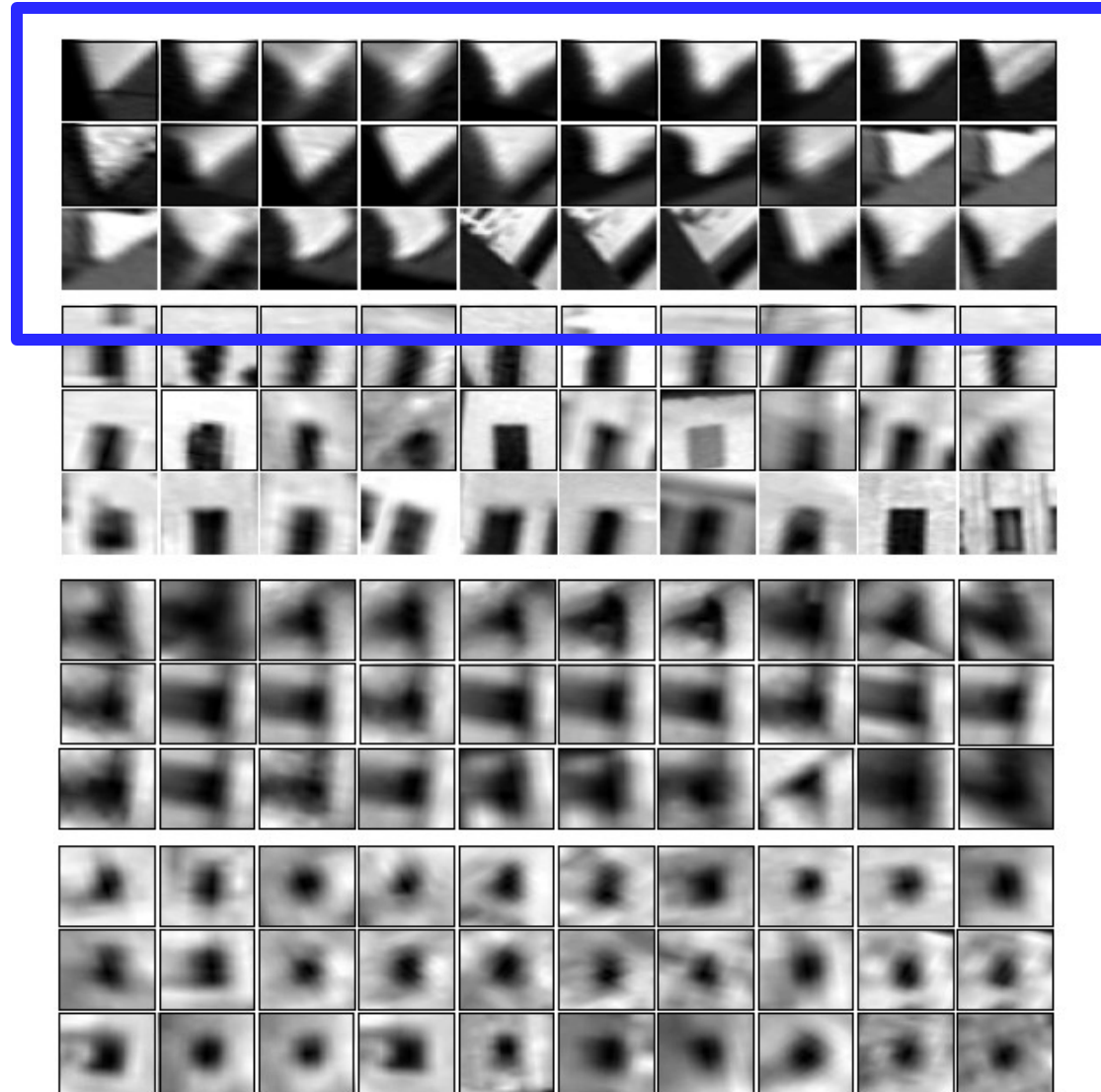
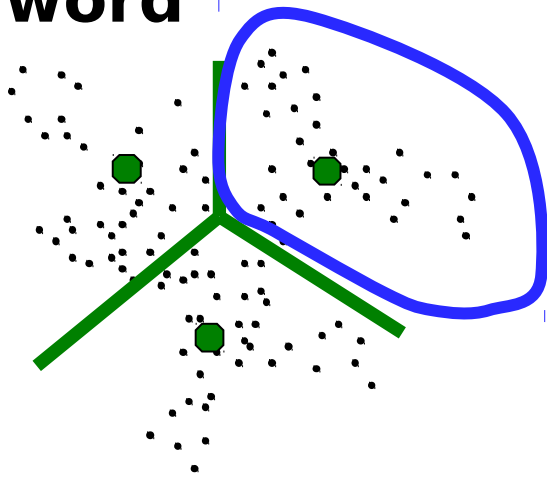
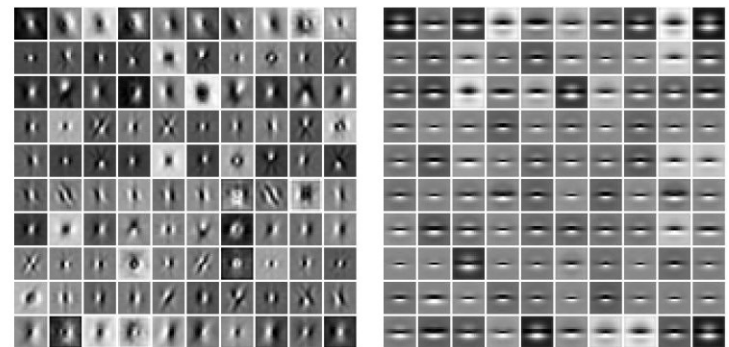
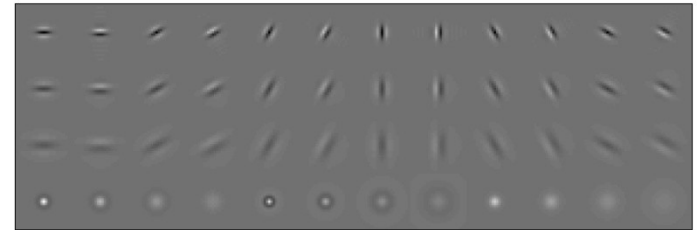
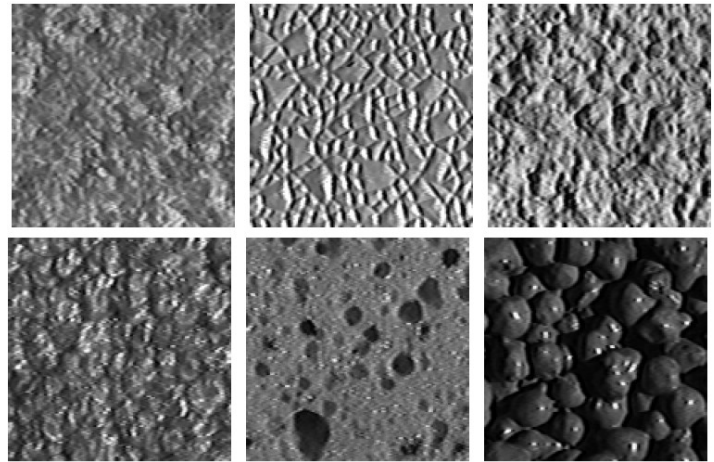


Figure from Sivic & Zisserman, ICCV 2003

# Visual words: texture representation

- **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.**

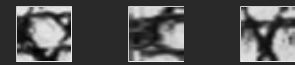
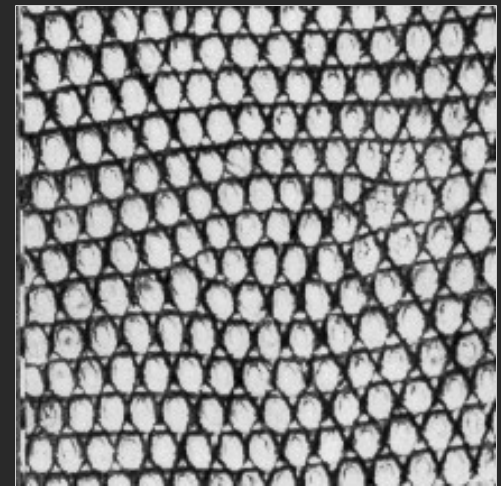
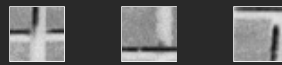
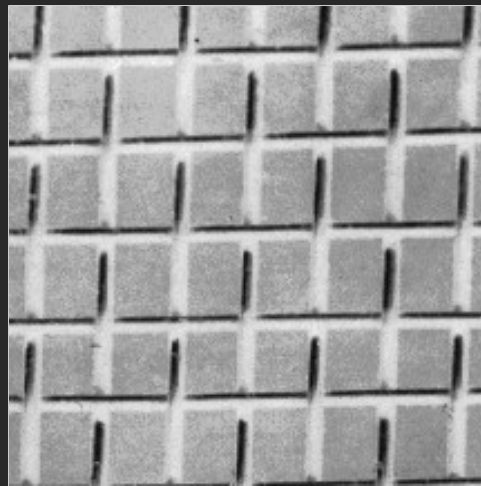
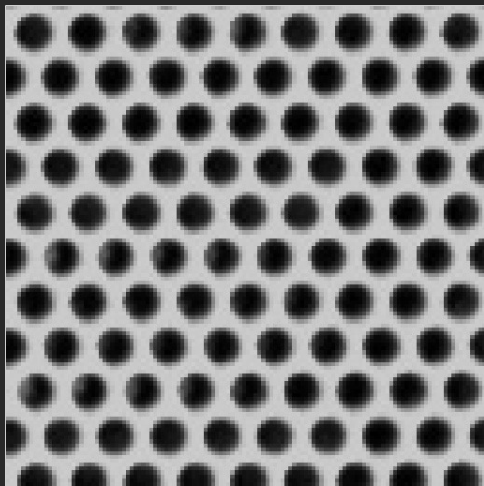
Leung & Malik 1999;  
Varma & Zisserman, 2002;  
Lazebnik, Schmid & Ponce,  
2003;





# Visual words: texture representation

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

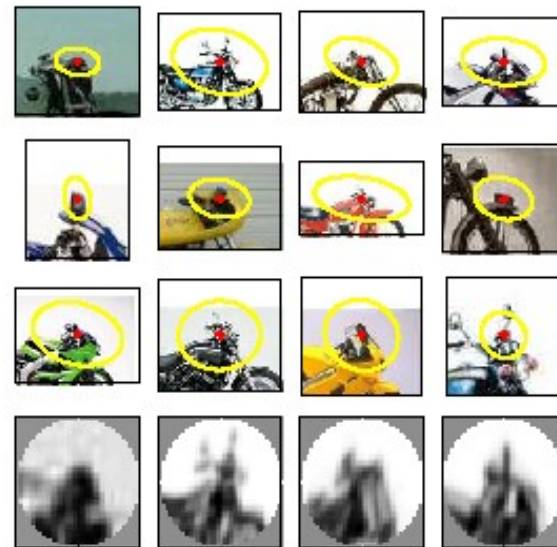
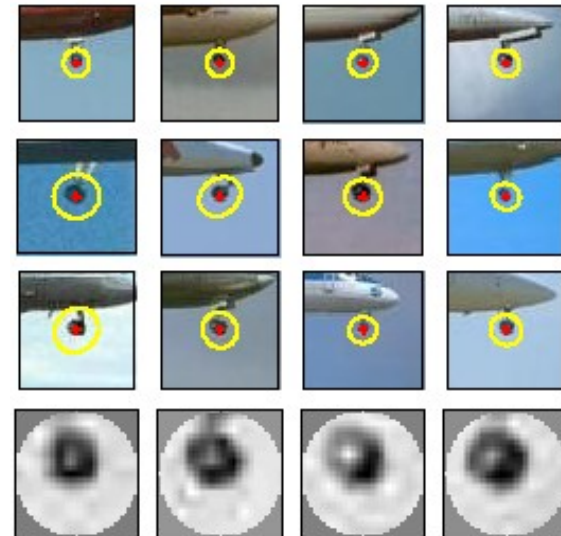




# Visual words

- **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.



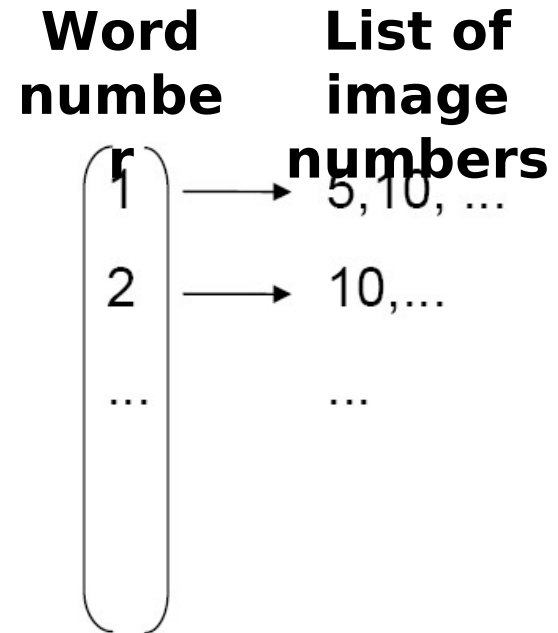
# Inverted file index for images comprised of visual words



frame #5



frame #10



*When will this give us a significant gain in efficiency?*

- 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 on the messages that reach our eyes. For a long time, the retinal image was considered as a movie image. It was discovered that the perception is more complex following the path to the various centers of the cortex, Hubel and Wiesel have demonstrated that the *message about image falling on the retina undergoes a point-by-point analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for 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 created by a predicted 30% increase in exports to \$750bn, compared with \$575bn in 2004. Imports are expected to reach \$660bn. The increase in the trade surplus will annoy the US. China's government has deliberately agreed to a trade deal with the US. The yuan is expected to rise against the dollar. The government also needs to increase the demand for the yuan in the country. China's government has permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.

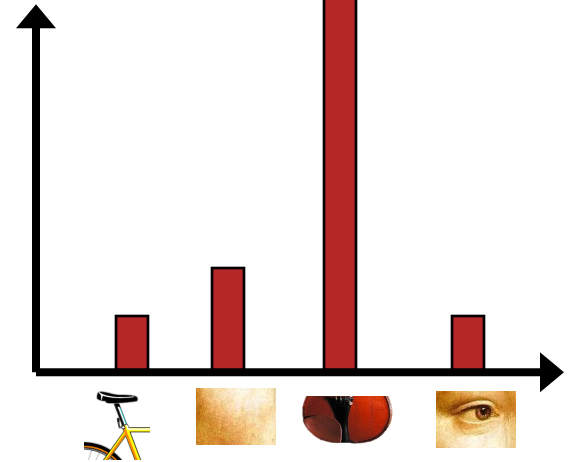
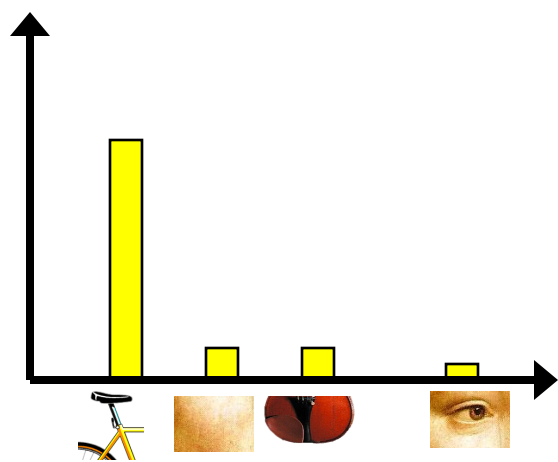
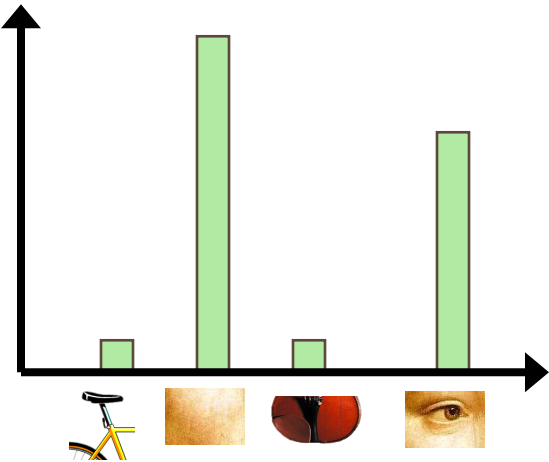
**China, trade,  
surplus, commerce,  
exports, imports, US,  
yuan, bank, domestic,  
foreign, increase,  
trade, value**

**Object**



**Bag of 'words'**

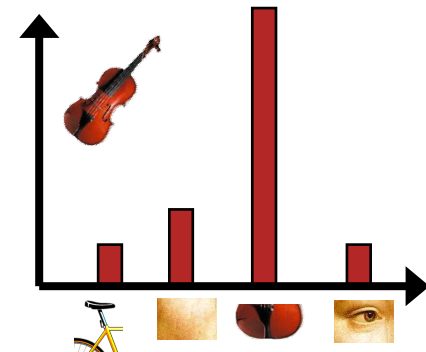
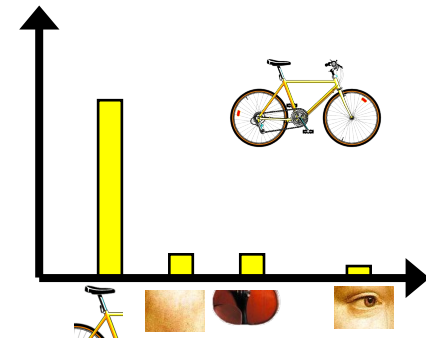
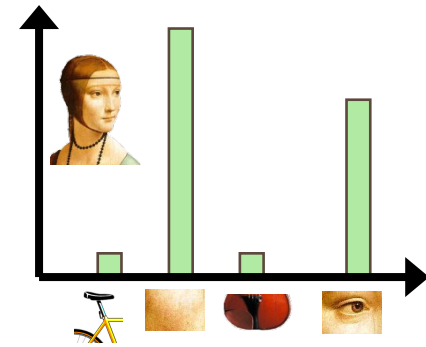




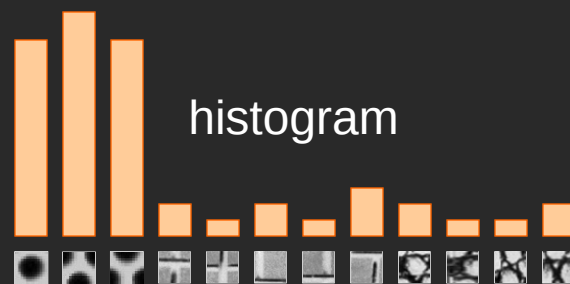
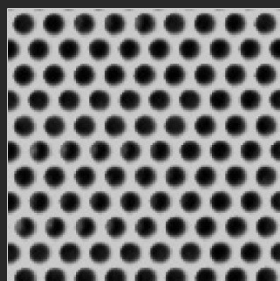


# Bags of visual words

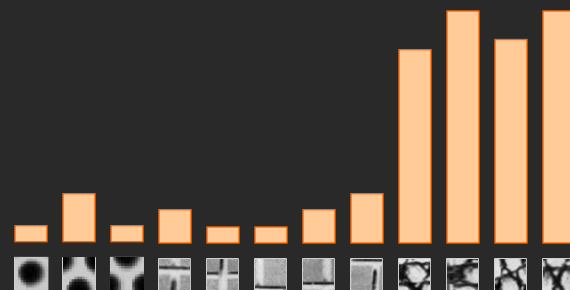
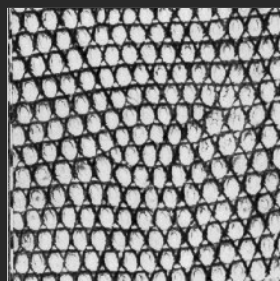
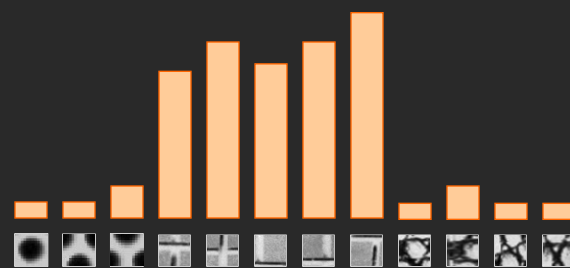
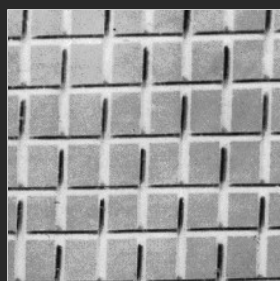
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



# Similarly, bags of textons for texture representation



Universal texton dictionary



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

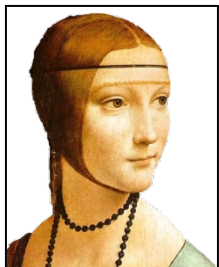
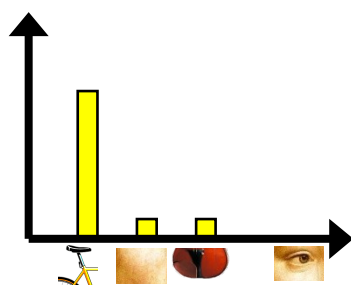
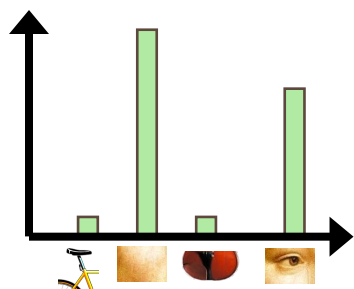
Source: Lana Lazebnik



# Comparing bags of words

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

$[1 \ 8 \ 1 \ 4]'$     $[5 \ 1 \ 1 \ 0]$



$d_j$

$q$

$$\begin{aligned} \text{sim}(d_j, q) &= \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} \\ &= \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \end{aligned}$$

# *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)

The diagram shows the formula  $t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$  with four arrows pointing to its components from descriptive text labels:

- An arrow from the label "Number of occurrences of word i in document d" points to the numerator  $n_{id}$ .
- An arrow from the label "Number of words in document d" points to the denominator  $n_d$ .
- An arrow from the label "Total number of documents in database" points to the numerator  $N$  of the logarithm.
- An arrow from the label "Number of occurrences of word i in whole database" points to the denominator  $n_i$  of the logarithm.

Number of occurrences of word i in document d

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

Number of words in document d

Total number of documents in database

Number of occurrences of word i 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”



# Example



## retrieved shots



Start frame 52907



Key frame 53026



End frame 53028



Start frame 54342



Key frame 54376



End frame 54644



Start frame 51770



Key frame 52251



End frame 52348



Start frame 54079



Key frame 54201



End frame 54201



Start frame 38909



Key frame 39126



End frame 39300



Start frame 40760



Key frame 40826



End frame 41049



Start frame 39301



Key frame 39676



End frame 39730

# 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

- Collecting words within a query region



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

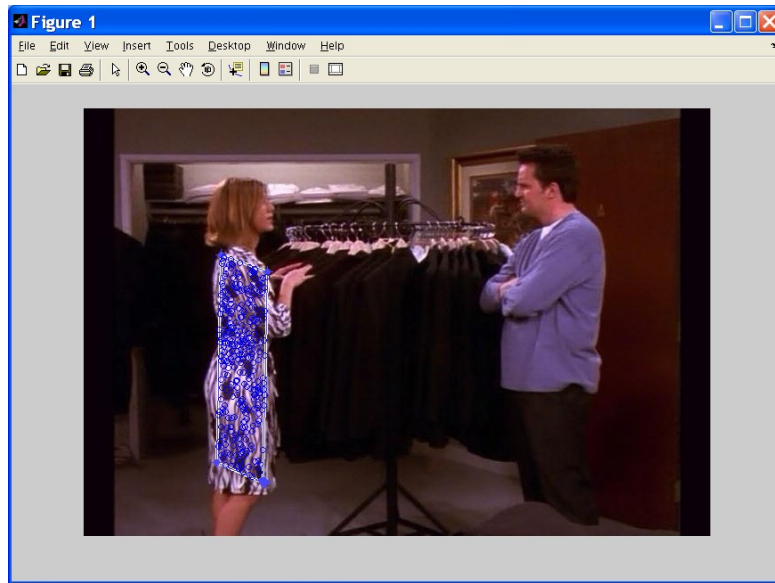


raw nn 1sim=0.56697

raw nn 2sim=0.56163

raw nn 5sim=0.54917





raw nn 1sim=0.67618



raw nn 2sim=0.66144



raw nn 3sim=0.66023



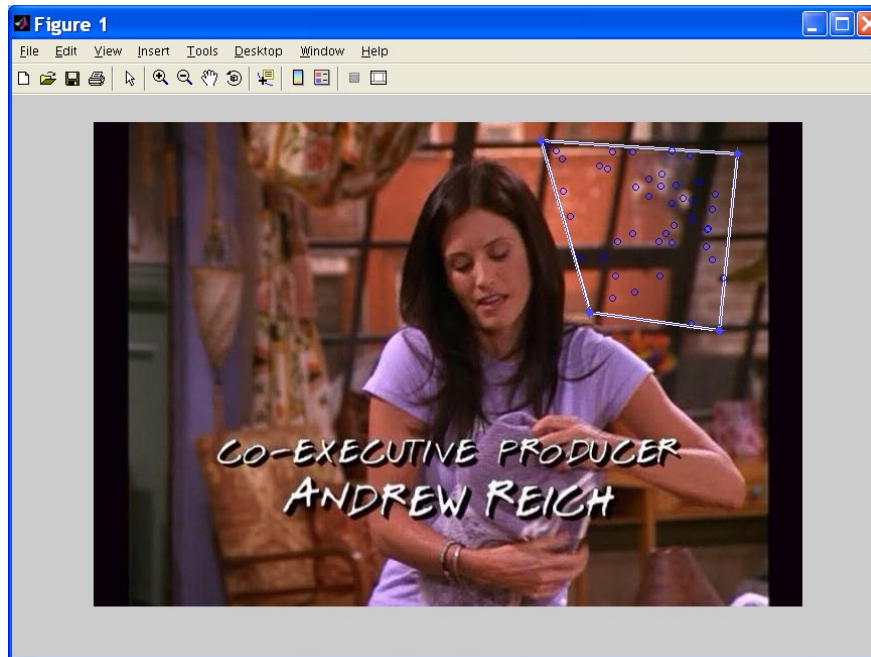
raw nn 4sim=0.65774



raw nn 5sim=0.6546







wtd nn 1sim=0.51966



wtd nn 2sim=0.50849



wtd nn 3sim=0.47587



wtd nn 4sim=0.46849



wtd nn 5sim=0.45963



# Bag of words representation: spatial info

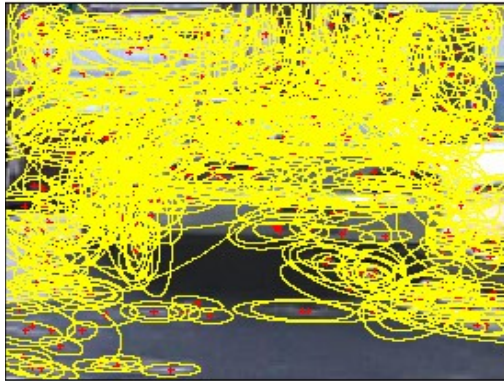
- A bag of words is an orderless representation: throwing out spatial relationships between features
- Middle ground:
  - Visual “phrases” : frequently co-occurring words
  - Semi-local features : describe configuration, neighborhood
  - Let position be part of each feature
  - Count bags of words only 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?**
- **Clustering / quantization algorithm**
- **Unsupervised vs. supervised**
- **What corpus provides features (universal vocabulary?)**
- **Vocabulary size, number of words**

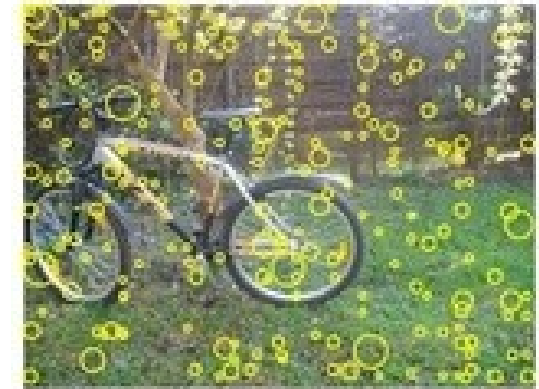
# Sampling strategies



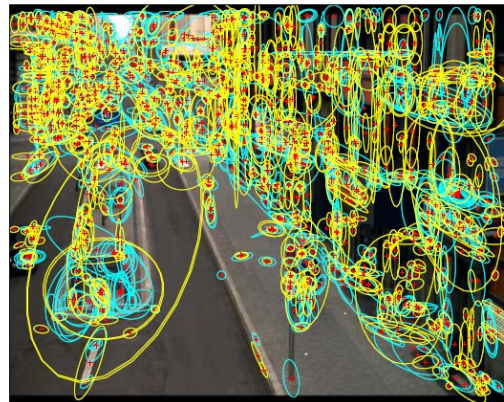
Sparse, at  
interest points



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.

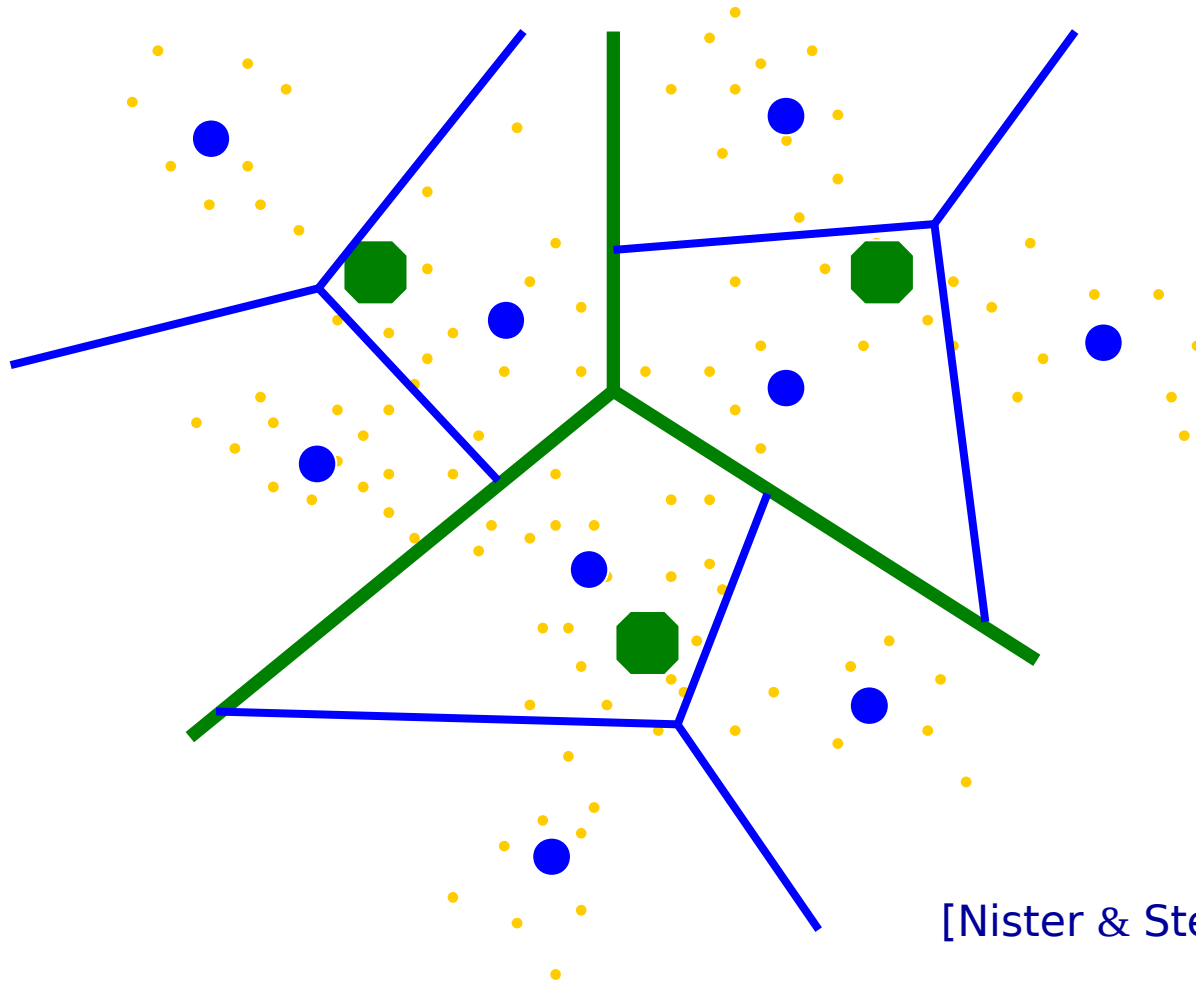
[See Nowak, Jurie & Triggs, ECCV 2006]

# Clustering / quantization methods

- **k-means (typical choice), agglomerative clustering, mean-shift,...**
- **Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies**
  - **Vocabulary tree [Nister & Stewenius, CVPR 2006]**

# Example: Recognition with Vocabulary Tree

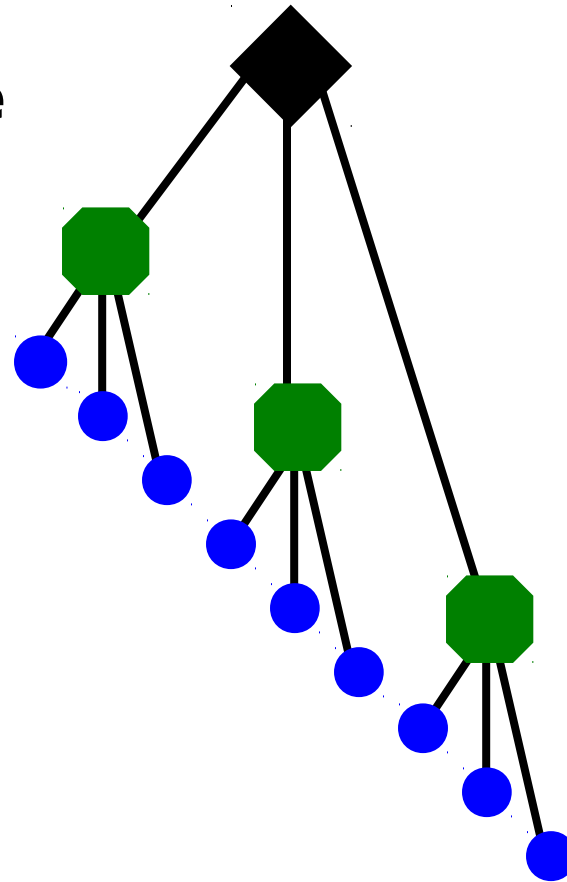
- **Tree construction:**



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

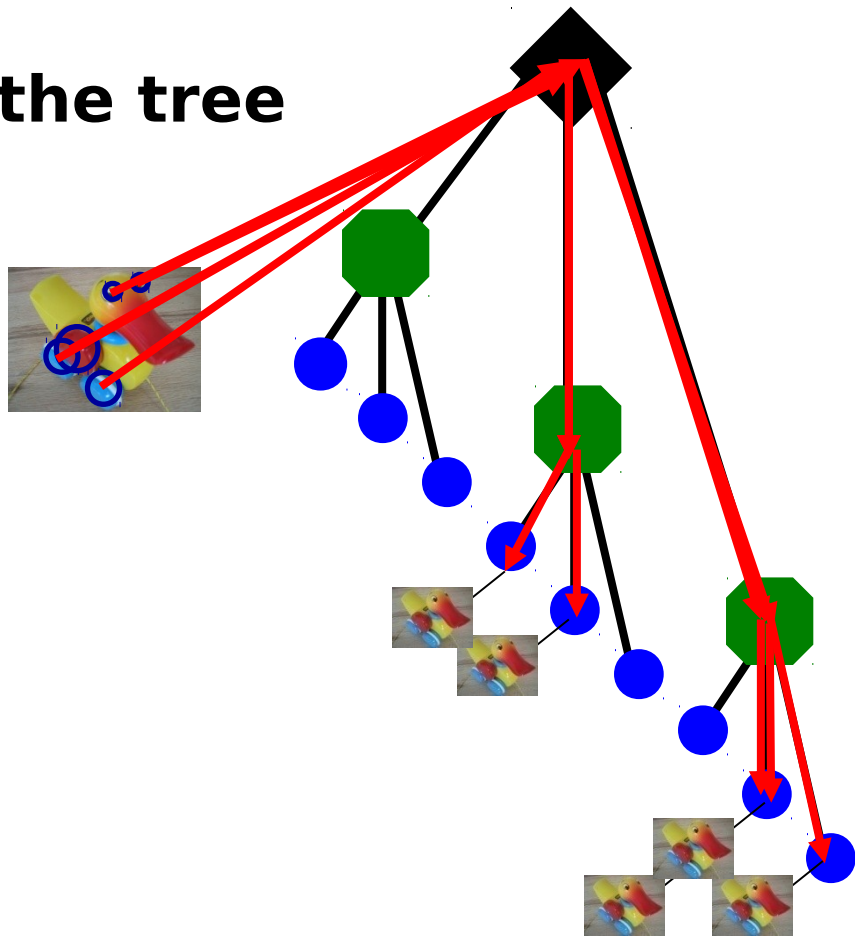
- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

- **Training: Filling the tree**

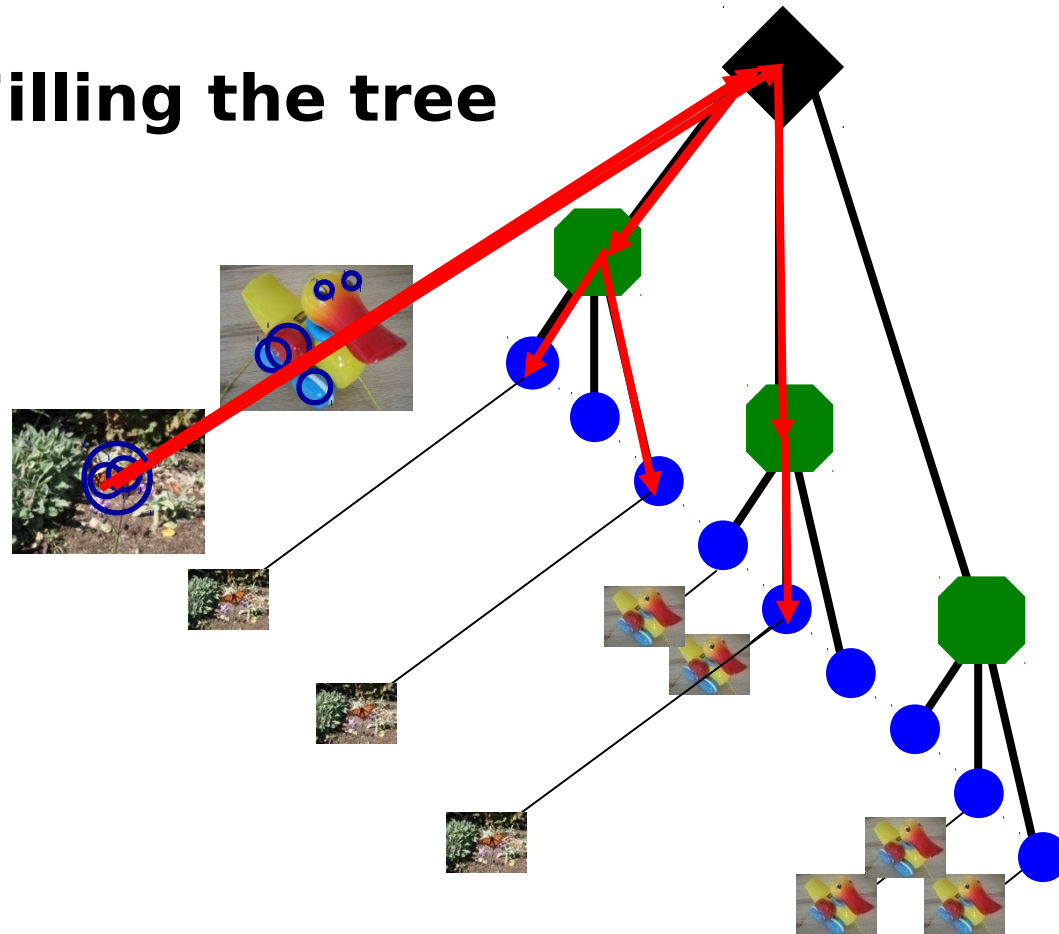


[Nister & Stewenius, CVPR'06



# Vocabulary Tree

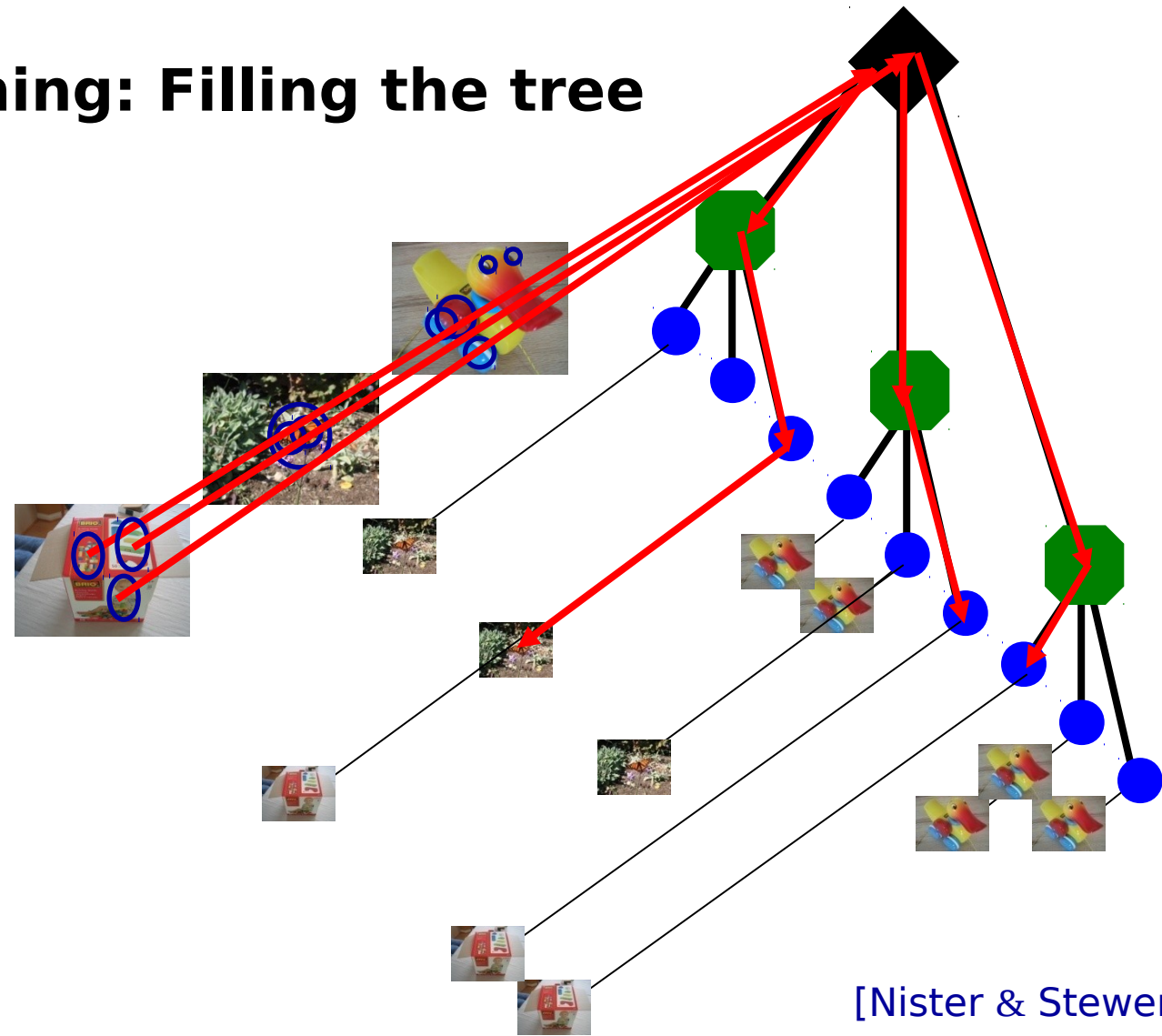
- **Training: Filling the tree**



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

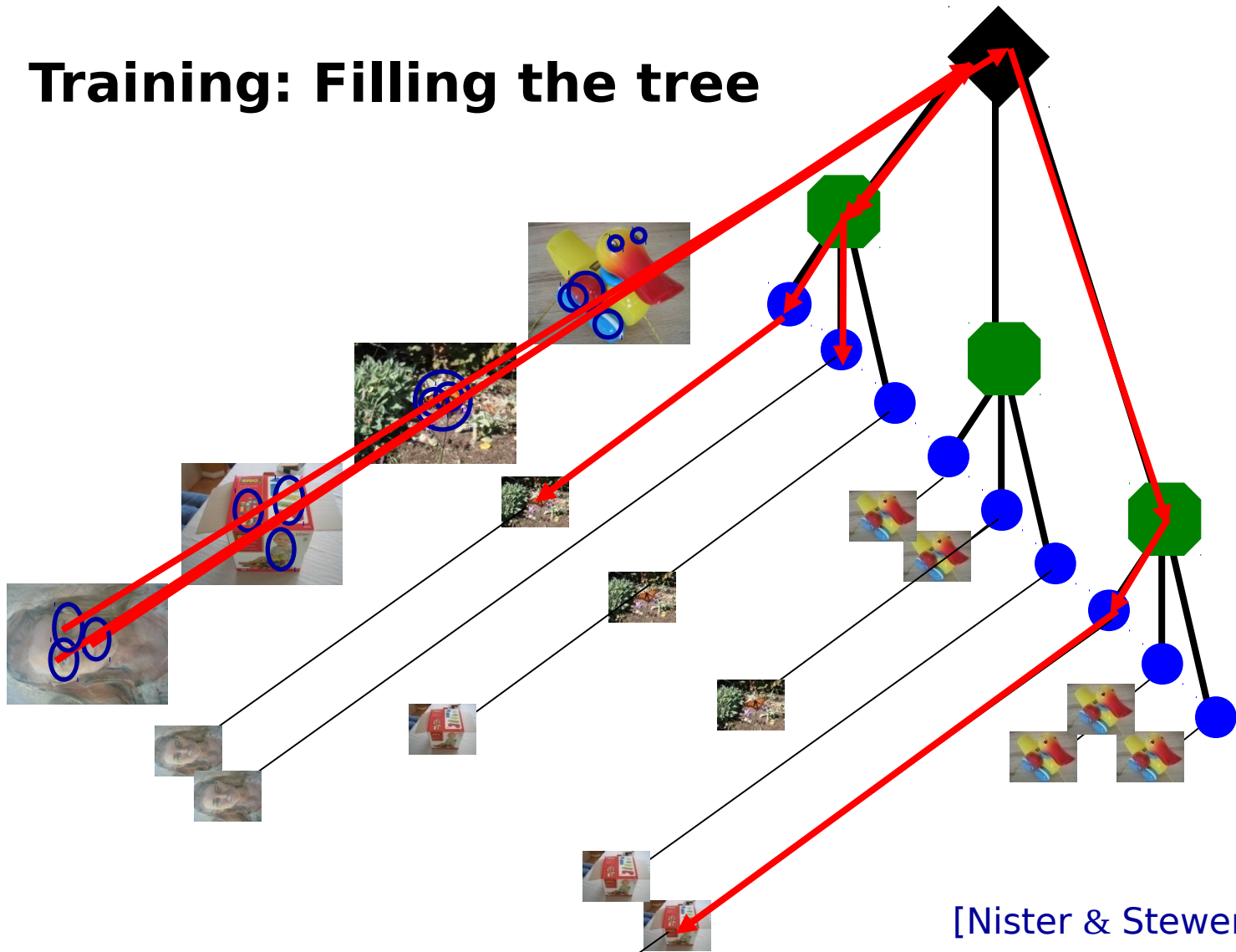
- Training: Filling the tree



[Nister & Stewenius, CVPR'06

# Vocabulary Tree

- **Training: Filling the tree**



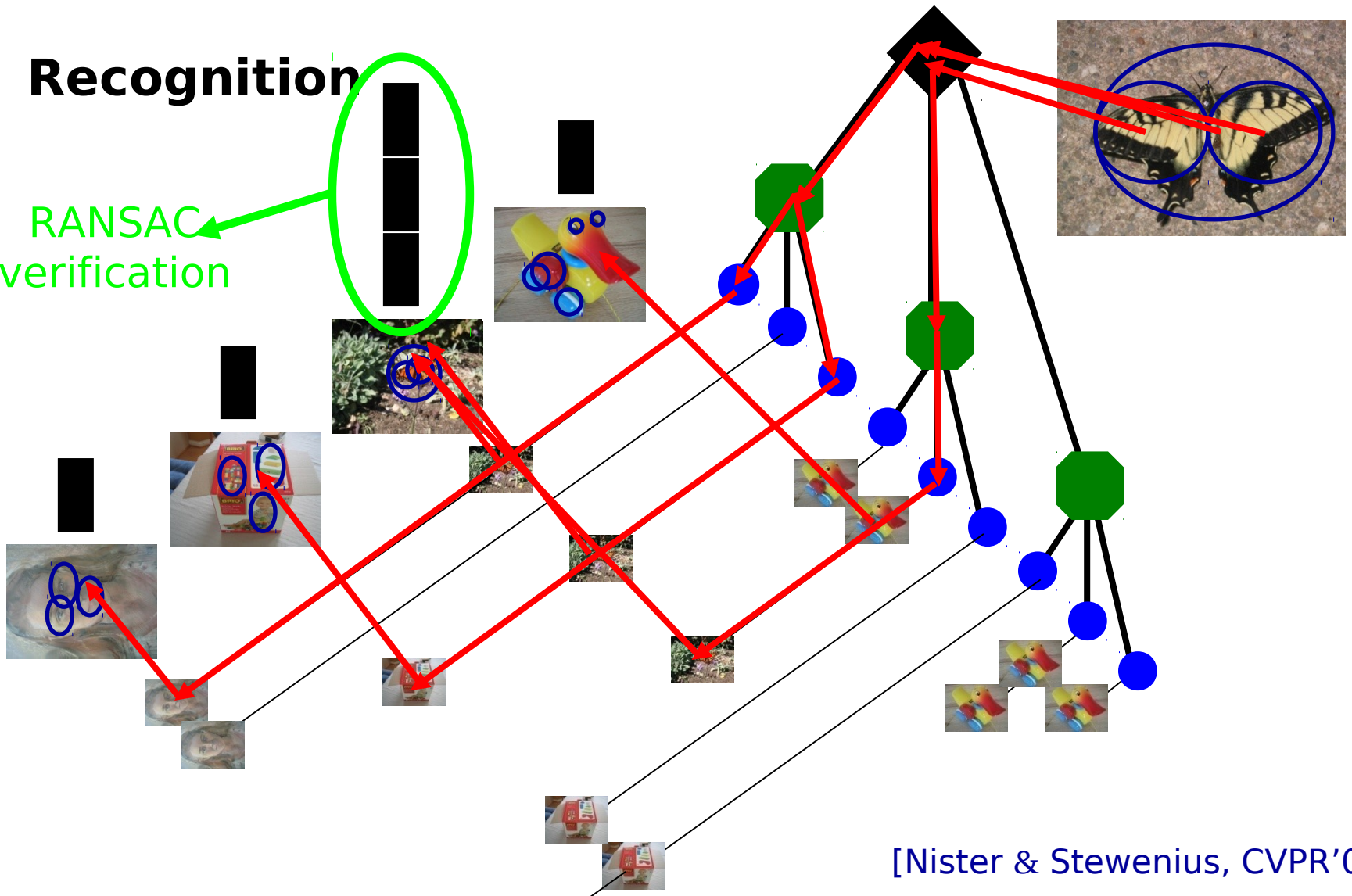
[Nister & Stewenius, CVPR'06

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

# Vocabulary Tree

- **Recognition**

RANSAC verification

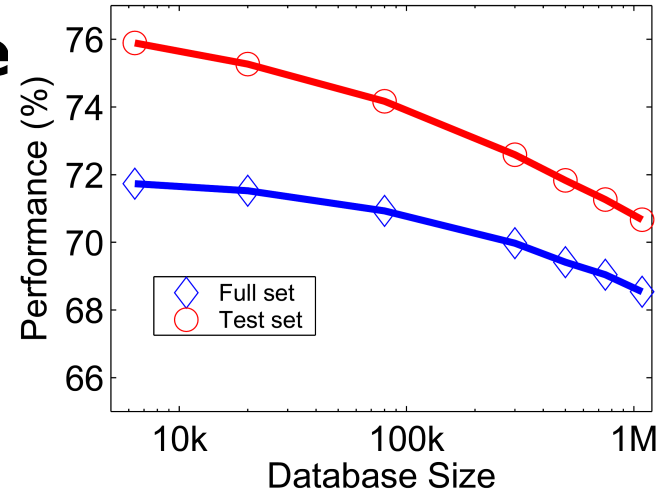


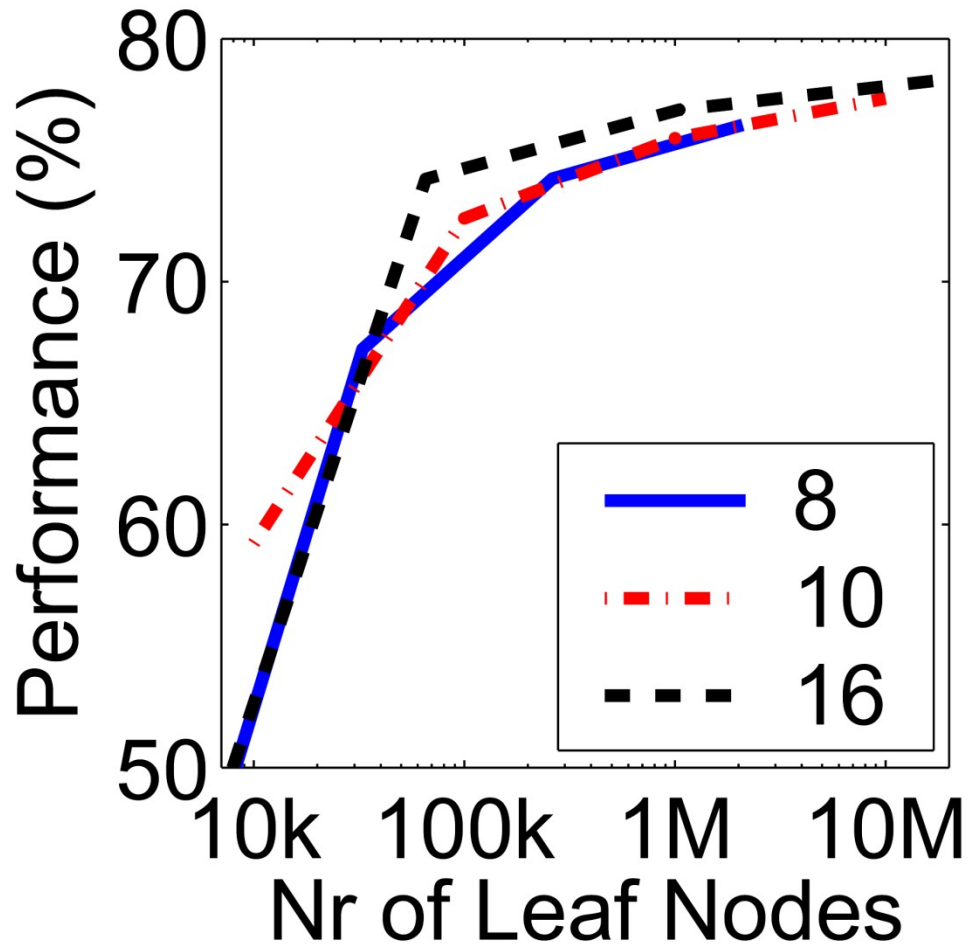
[Nister & Stewenius, CVPR'06

# Vocabulary Tree: Performance

- **Evaluated on large database**
  - Indexing with up to 1M images
- **Online recognition for database of 50,000 CD covers**
  - Retrieval in ~1s
- **Find experimentally that large vocabularies can be beneficial for recognition**

[Nister & Stewenius, CVPR'06]

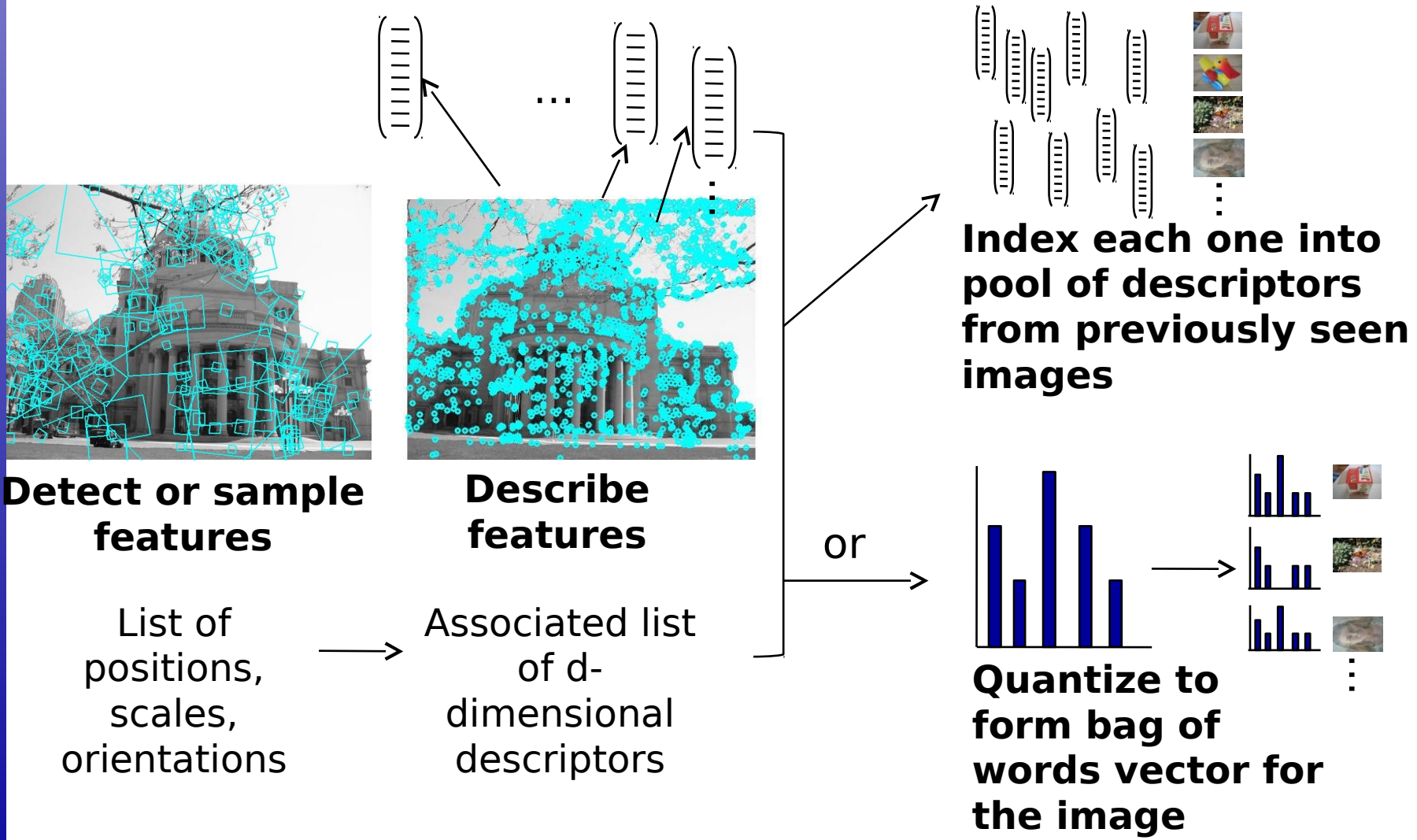




Larger vocabularies  
can be  
advantageous...

But what happens if  
it is too large?

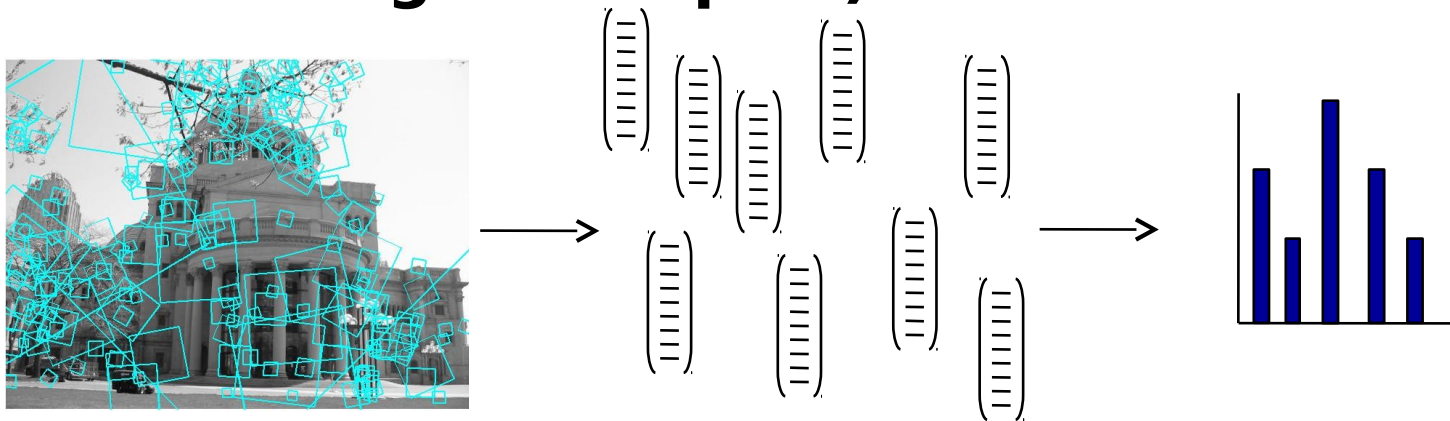
# Recap: indexing features





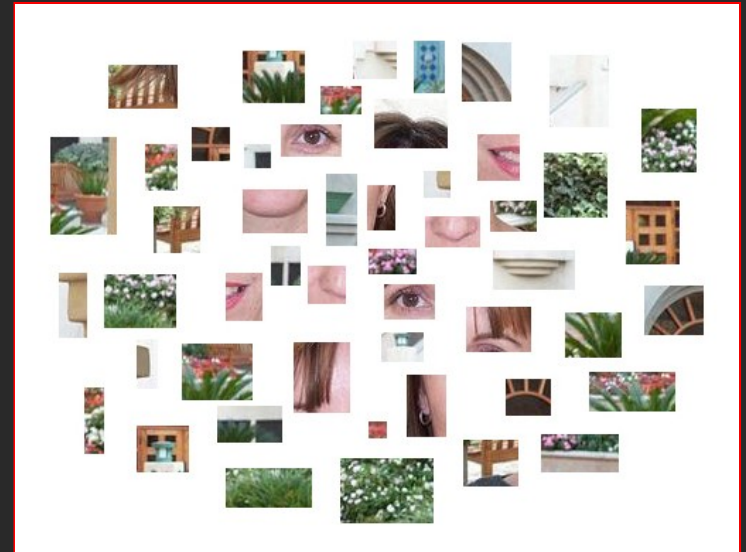
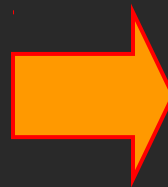
# Learning and recognition with bag of words histograms

- **Bag of words representation makes it possible to describe the unordered point set with a single vector (of fixed dimension across image examples)**



- **Provides easy way to use distribution of feature types with various learning algorithms requiring vector input.**

# Bags of features for object recognition



**face, flowers, building**

- Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

# Bags of features for object recognition

## Caltech6 dataset



class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	<b>98.8</b>	97.1	90.2
cars (rear)	98.3	<b>98.6</b>	90.3
cars (side)	<b>95.0</b>	87.3	88.5
faces	<b>100</b>	99.3	96.4
motorbikes	<b>98.5</b>	98.0	92.5
spotted cats	<b>97.0</b>	—	90.0

# 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

# Next

- Next week : Object recognition