

# Computer Vision - Lecture 13

## Indexing and Visual Vocabularies

12.12.2016

Bastian Leibe

RWTH Aachen

<http://www.vision.rwth-aachen.de>

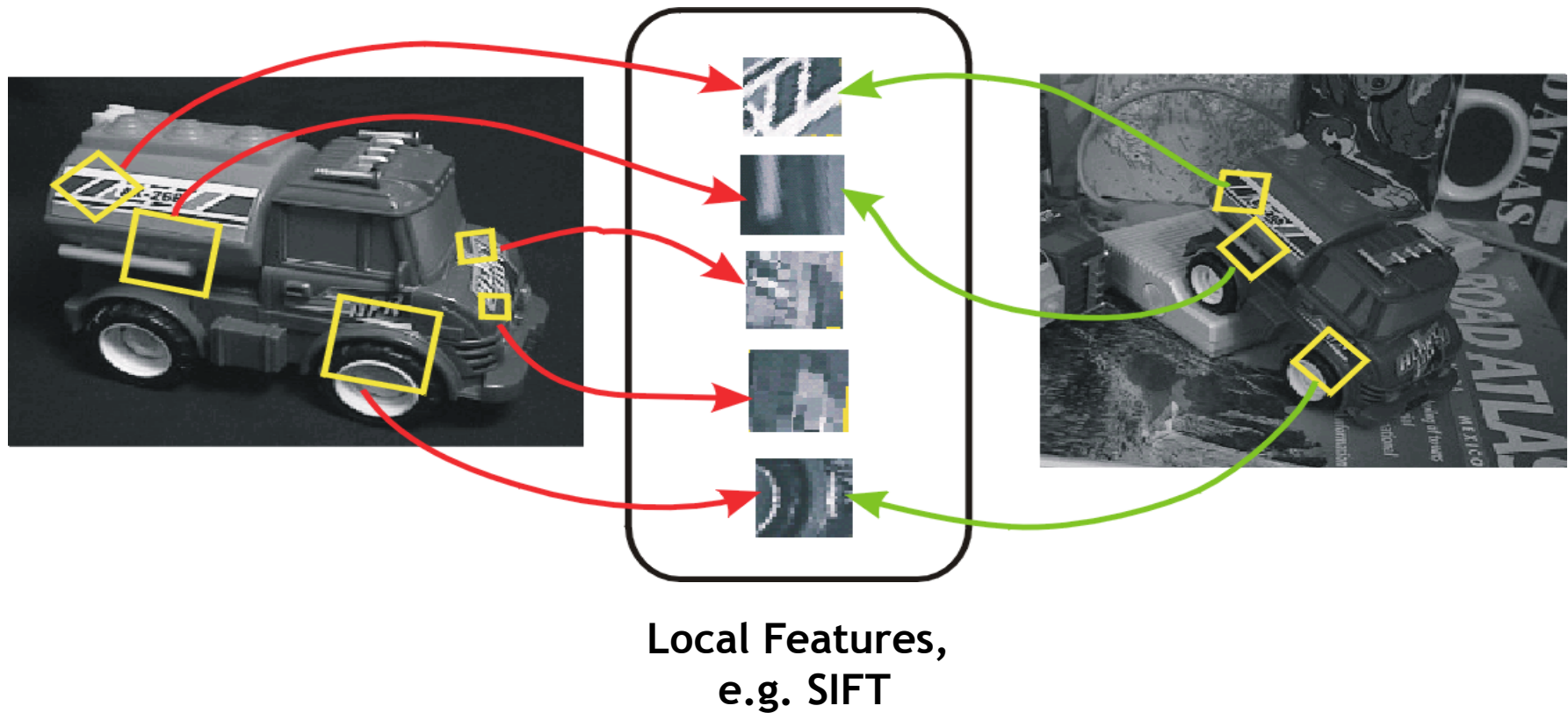
[leibe@vision.rwth-aachen.de](mailto:leibe@vision.rwth-aachen.de)

# Course Outline

- **Image Processing Basics**
- **Segmentation & Grouping**
- **Object Recognition**
- **Object Categorization I**
  - Sliding Window based Object Detection
- **Local Features & Matching**
  - Local Features - Detection and Description
  - Recognition with Local Features
  - **Indexing & Visual Vocabularies**
- **Object Categorization II**
  - Bag-of-Words Approaches & Part-based Approaches
- **3D Reconstruction**

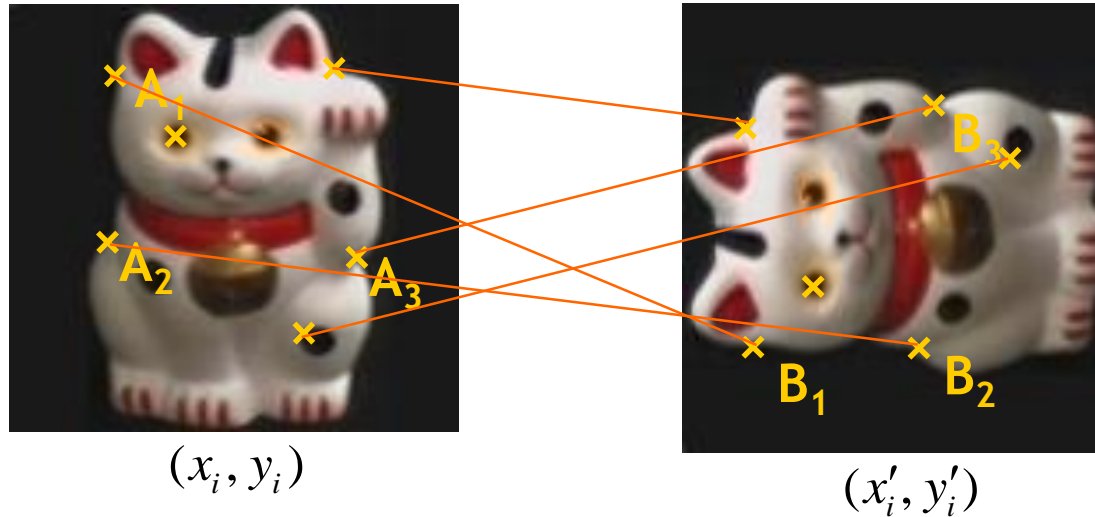
# Recap: Recognition with Local Features

- Image content is transformed into local features that are invariant to translation, rotation, and scale
- Goal: Verify if they belong to a consistent configuration



# Recap: Fitting an Affine Transformation

- Assuming we know the correspondences, how do we get the transformation?

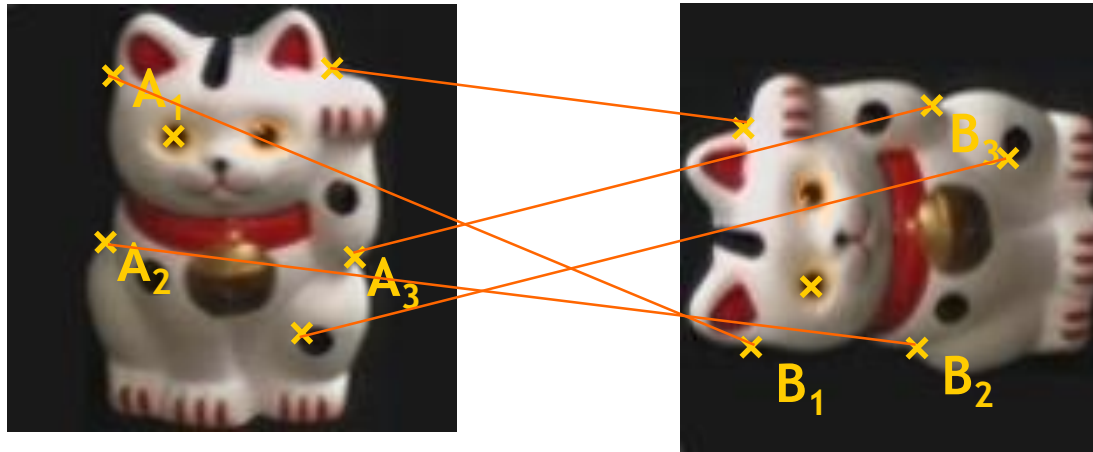


$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

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

# Recap: Fitting a Homography

- Estimating the transformation



Homogenous coordinates

Image coordinates

$$\begin{aligned} \mathbf{x}_{A_1} &\leftrightarrow \mathbf{x}_{B_1} \\ \mathbf{x}_{A_2} &\leftrightarrow \mathbf{x}_{B_2} \\ \mathbf{x}_{A_3} &\leftrightarrow \mathbf{x}_{B_3} \\ &\vdots \end{aligned}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ \frac{1}{z'} y' \\ z' \end{bmatrix}$$

Matrix notation

$$x' = Hx$$

$$x'' = \frac{1}{z'} x'$$

$$x_{A_1} = \frac{h_{11} x_{B_1} + h_{12} y_{B_1} + h_{13}}{h_{31} x_{B_1} + h_{32} y_{B_1} + 1}$$

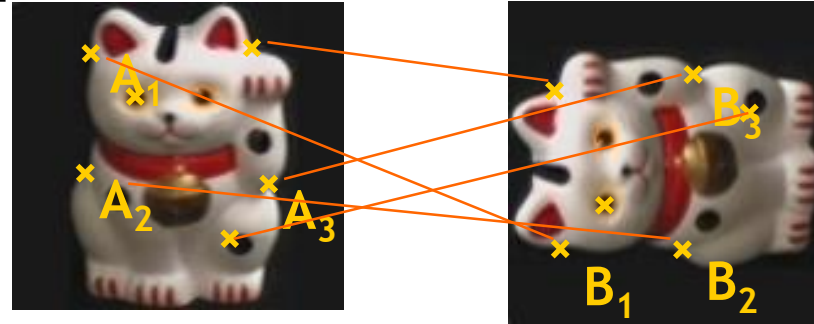
$$y_{A_1} = \frac{h_{21} x_{B_1} + h_{22} y_{B_1} + h_{23}}{h_{31} x_{B_1} + h_{32} y_{B_1} + 1}$$

B. Leibe

# Recap: Fitting a Homography

- Estimating the transformation

$$\begin{aligned}
 h_{11} x_{B_1} + h_{12} y_{B_1} + h_{13} - x_{A_1} h_{31} x_{B_1} - x_{A_1} h_{32} y_{B_1} - x_{A_1} &= 0 \\
 h_{21} x_{B_1} + h_{22} y_{B_1} + h_{23} - y_{A_1} h_{31} x_{B_1} - y_{A_1} h_{32} y_{B_1} - y_{A_1} &= 0
 \end{aligned}$$

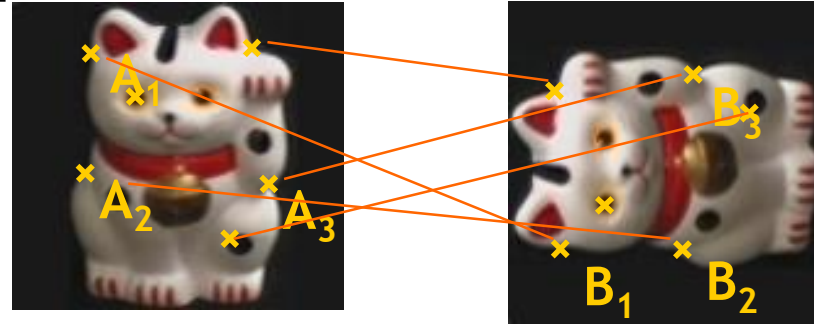


$$\begin{aligned}
 \mathbf{x}_{A_1} &\leftrightarrow \mathbf{x}_{B_1} \\
 \mathbf{x}_{A_2} &\leftrightarrow \mathbf{x}_{B_2} \\
 \mathbf{x}_{A_3} &\leftrightarrow \mathbf{x}_{B_3} \\
 &\vdots \\
 &\vdots
 \end{aligned}
 \quad
 \begin{bmatrix}
 x_{B_1} & y_{B_1} & 1 & 0 & 0 & 0 & -x_{A_1} x_{B_1} & -x_{A_1} y_{B_1} & -x_{A_1} \\
 0 & 0 & 0 & x_{B_1} & y_{B_1} & 1 & -y_{A_1} x_{B_1} & -y_{A_1} y_{B_1} & -y_{A_1} \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
 \end{bmatrix}
 \cdot
 \begin{bmatrix}
 h_{11} \\
 h_{12} \\
 h_{13} \\
 h_{21} \\
 h_{22} \\
 h_{23} \\
 h_{31} \\
 h_{32} \\
 1
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot
 \end{bmatrix}$$

$$Ah = 0$$

# Recap: Fitting a Homography

- Estimating the transformation
- Solution:
  - Null-space vector of A
  - Corresponds to smallest eigenvector



$$\begin{aligned} \mathbf{x}_{A_1} &\leftrightarrow \mathbf{x}_{B_1} \\ \mathbf{x}_{A_2} &\leftrightarrow \mathbf{x}_{B_2} \\ \mathbf{x}_{A_3} &\leftrightarrow \mathbf{x}_{B_3} \\ &\vdots \end{aligned}$$

$$\begin{aligned} &\text{SVD} \\ &\downarrow \\ \mathbf{A} &= \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U} \begin{bmatrix} d_{11} & \cdots & d_{19} \\ \vdots & \ddots & \vdots \\ d_{91} & \cdots & d_{99} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{19} \\ \vdots & \ddots & \vdots \\ v_{91} & \cdots & v_{99} \end{bmatrix}^T \end{aligned}$$

$$Ah = 0$$

$$\mathbf{h} = \frac{[v_{19}, \dots, v_{99}]}{v_{99}}$$

Minimizes least square error

# Recap: Object Recognition by Alignment

- Assumption
  - Known object, rigid transformation compared to model image
  - ⇒ *If we can find evidence for such a transformation, we have recognized the object.*

- You learned methods for
  - Fitting an *affine transformation* from  $\geq 3$  correspondences
  - Fitting a *homography* from  $\geq 4$  correspondences

Affine: solve a system

$$At = b$$

Homography: solve a system

$$Ah = 0$$

- Correspondences may be noisy and may contain outliers
  - ⇒ Need to use robust methods that can filter out outliers



# Recap: Robust Estimation with RANSAC

## RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
  2. Compute transformation from seed group
  3. Find *inliers* to this transformation
  4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
- Keep the transformation with the largest number of inliers

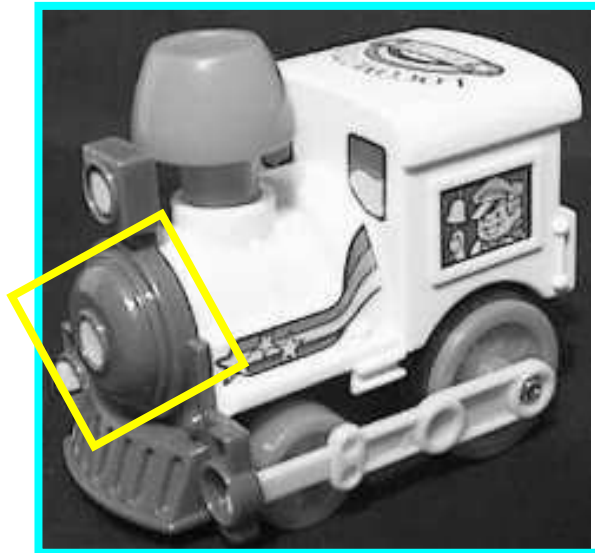
# Problem with RANSAC

- In many practical situations, the percentage of outliers (incorrect putative matches) is often very high (90% or above).
- Alternative strategy: Generalized Hough Transform

# Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).

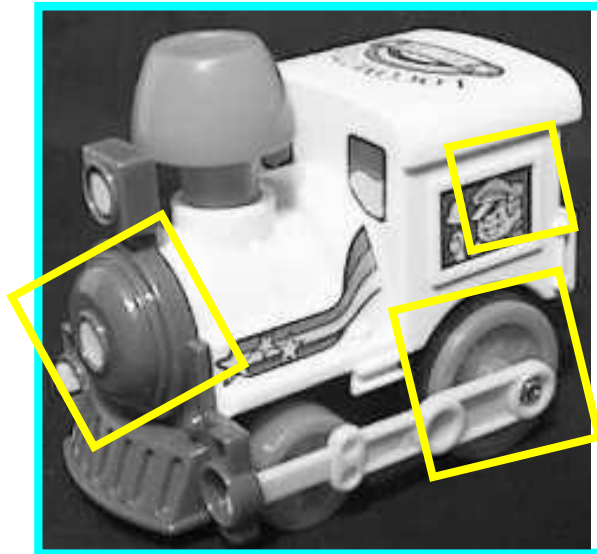
model



# Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
  - Of course, a hypothesis from a single match is unreliable.
  - Solution: let each match vote for its hypothesis in a Hough space with very coarse bins.

model



# Pose Clustering and Verification with SIFT

- To detect instances of objects from a model base:



## 1. Index descriptors

- Distinctive features narrow down possible matches



# Pose Clustering and Verification with SIFT

- To detect instances of objects from a model base:



## 1. Index descriptors

- Distinctive features narrow down possible matches

## 2. Generalized Hough transform to vote for poses

- Keypoints have record of parameters relative to model coordinate system

## 3. Affine fit to check for agreement between model and image features

- Fit and verify using features from Hough bins with 3+ votes

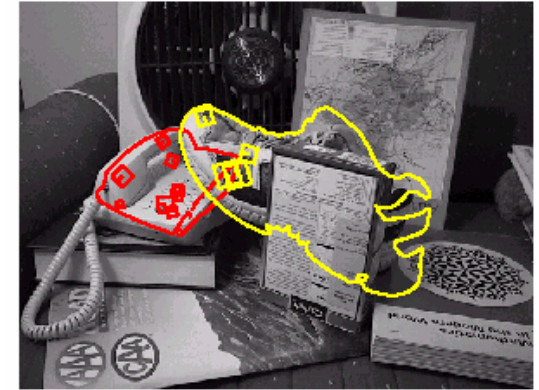
# Object Recognition Results



**Background subtract for model boundaries**



**Objects recognized**

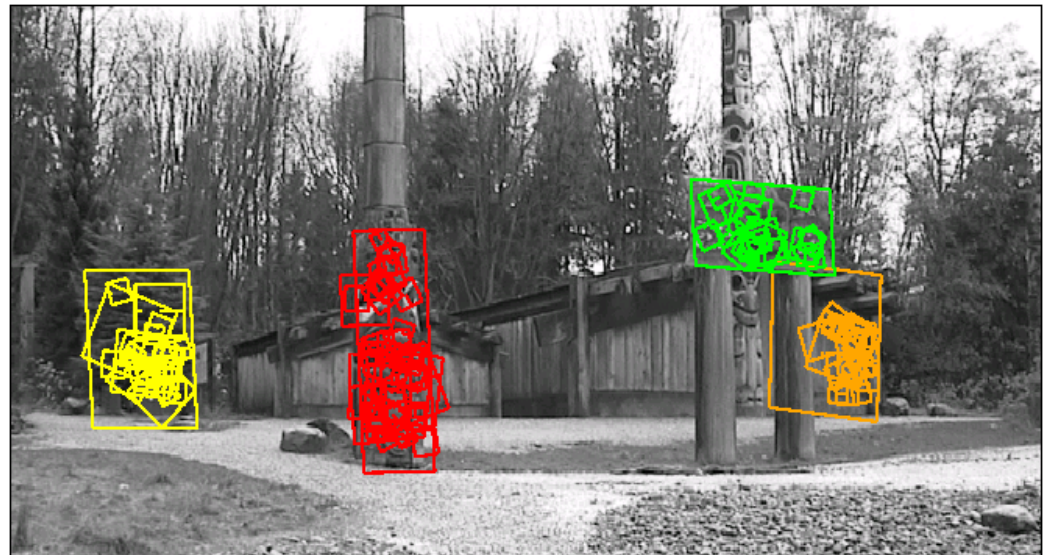


**Recognition in spite of occlusion**

# Location Recognition



Training



[Lowe, IJCV'04]

Slide credit: David Lowe



# Topics of This Lecture

- **Indexing with Local Features**
  - Inverted file index
  - Visual Words
  - Visual Vocabulary construction
  - tf-idf weighting
- **Bag-of-Words Model**
  - Use for image classification

# Application: Mobile Visual Search



## Google Goggles in Action

Click the icons below to see the different ways Google Goggles can be used.



- Take photos of objects as queries for visual search

# Large-Scale Image Matching Problem

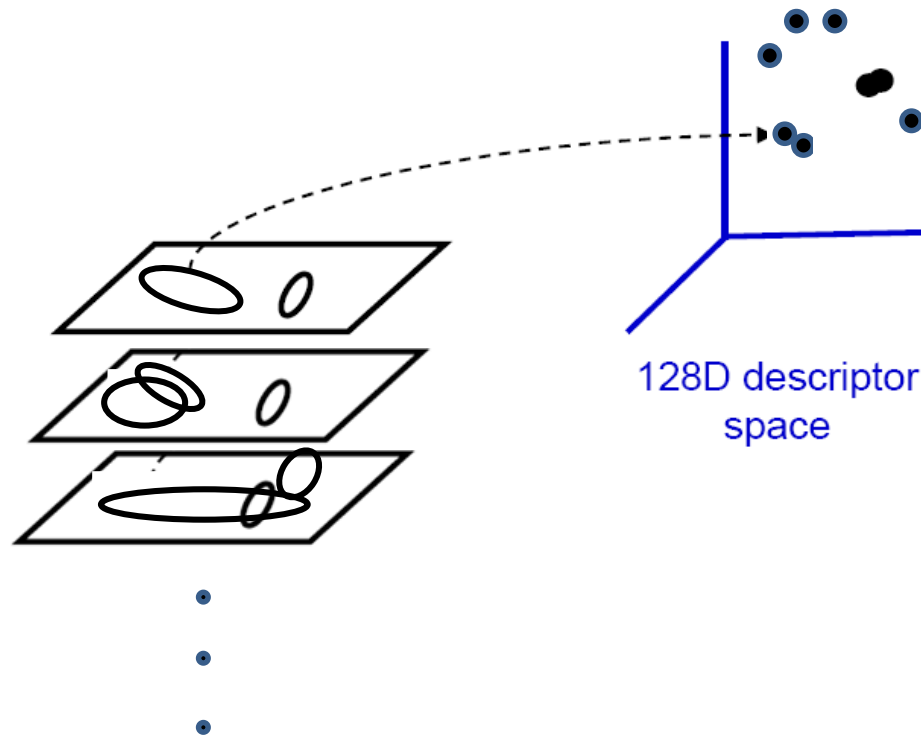


Database with thousands (millions) of images

- How can we perform this matching step efficiently?

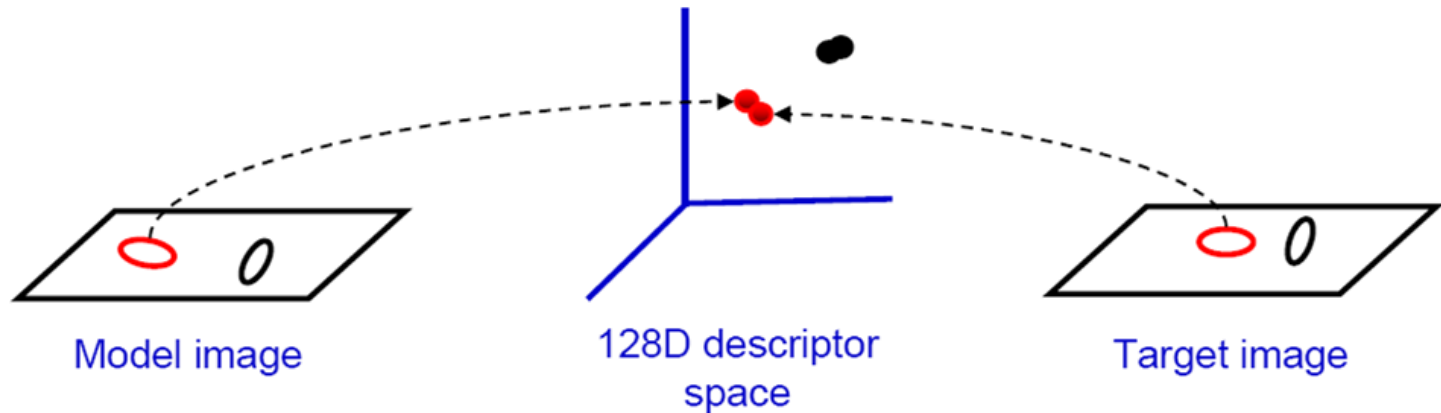
# Indexing Local Features

- 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 for many applications
  - E.g. Image matching,
  - E.g. Retrieving images of similar objects,
  - E.g. Object recognition, categorization, 3d Reconstruction,...

# 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 (e.g. through PCA):
  - 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	Fl 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; 85
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; 76,161,190
	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".

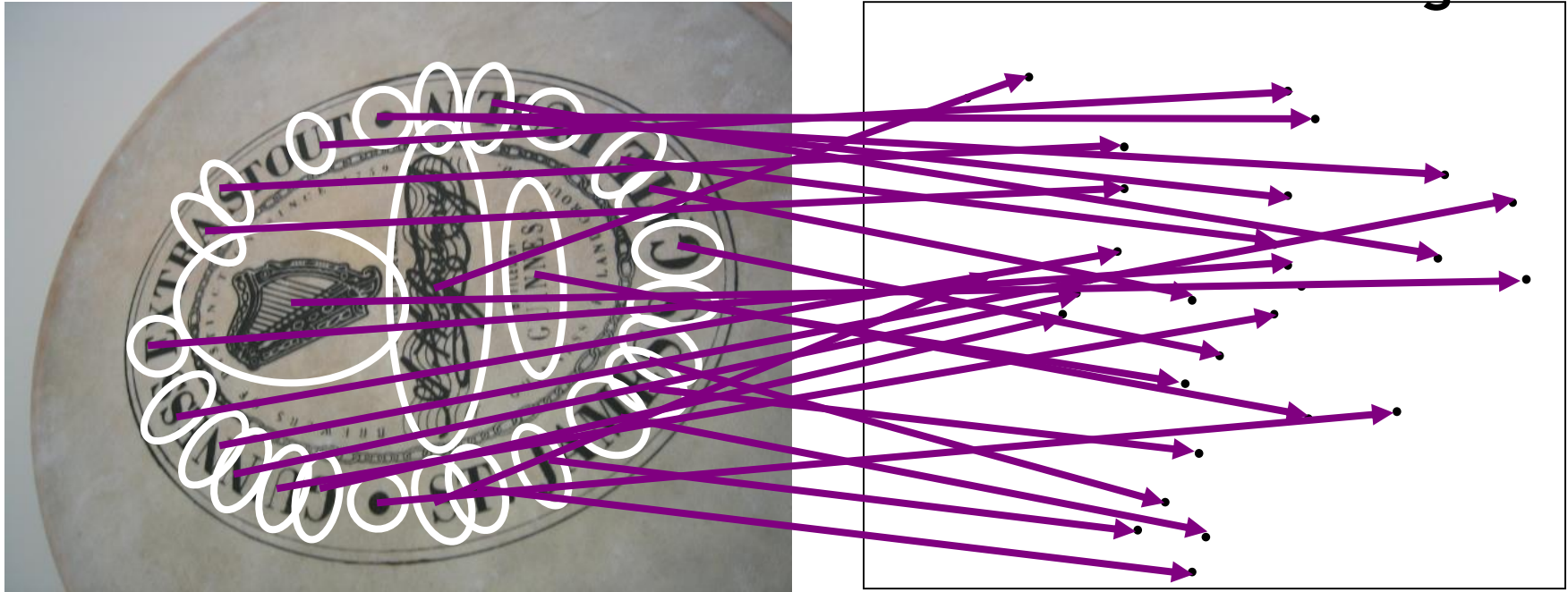
# Text Retrieval vs. Image Search

- What makes the problems similar, different?

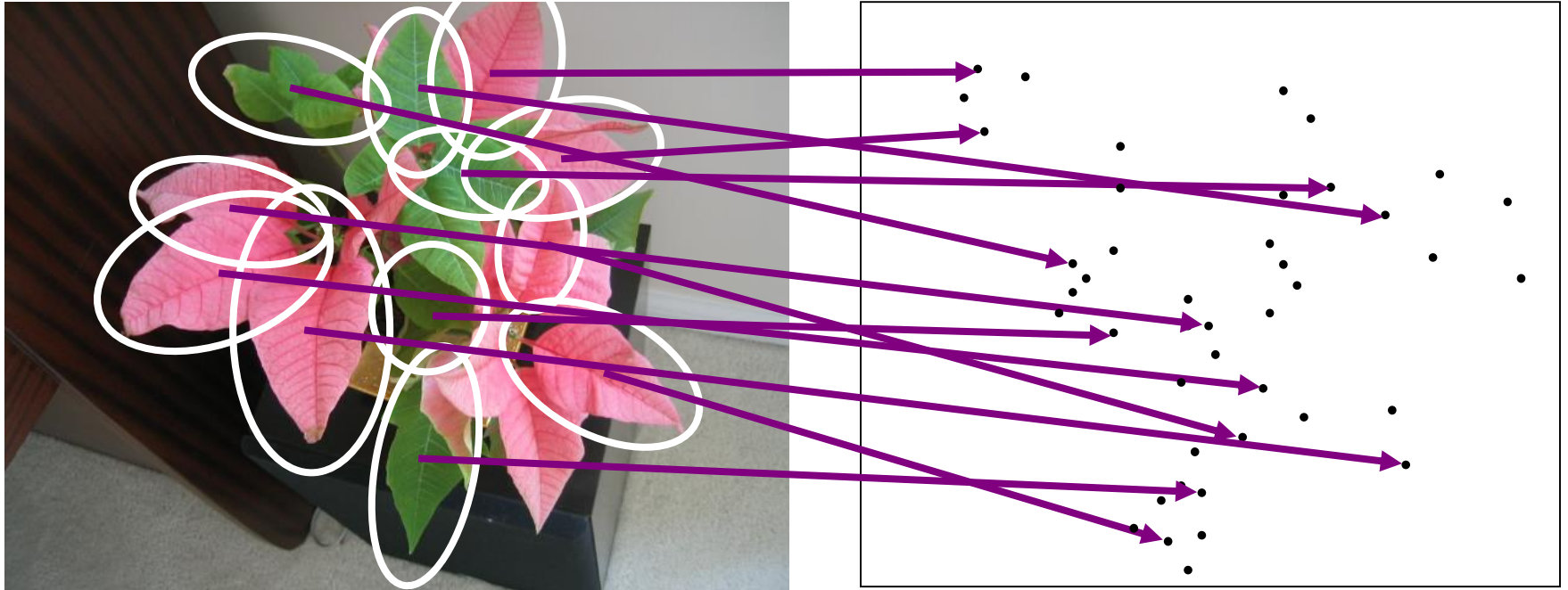


# Visual Words: Main Idea

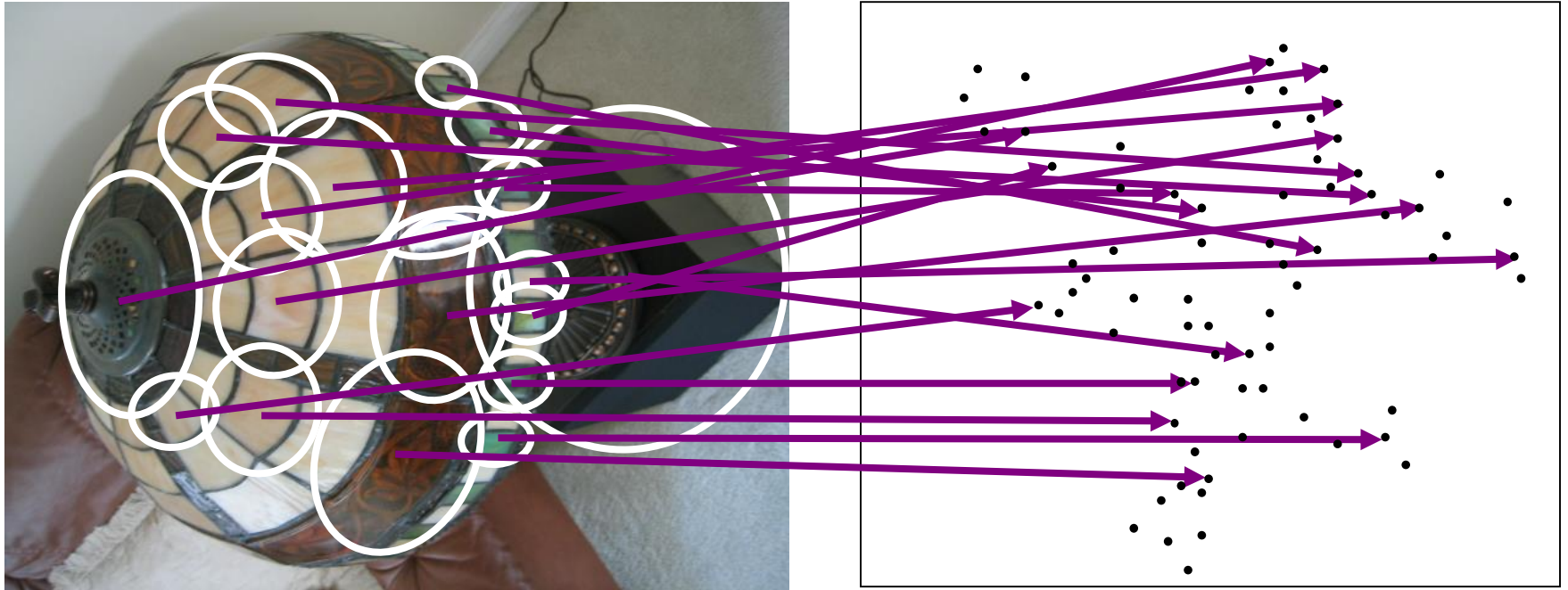
- Extract some local features from a number of images ...



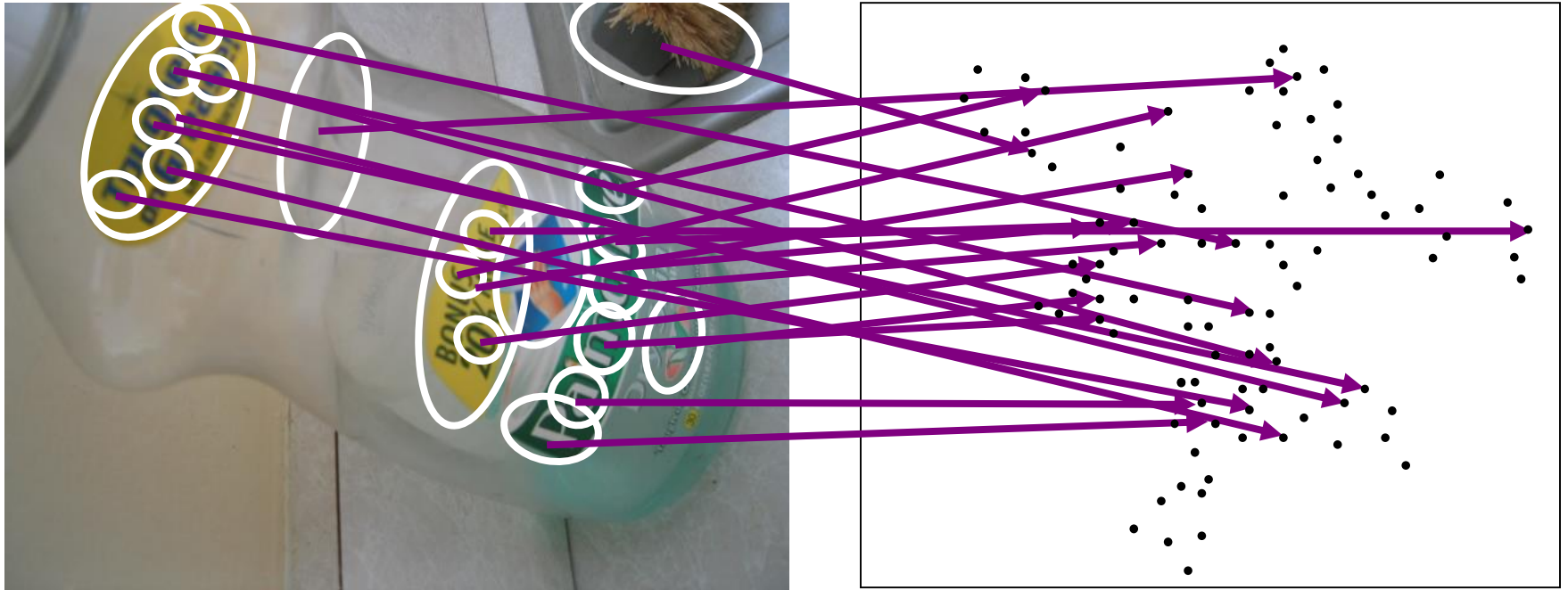
# Visual Words: Main Idea



# Visual Words: Main Idea

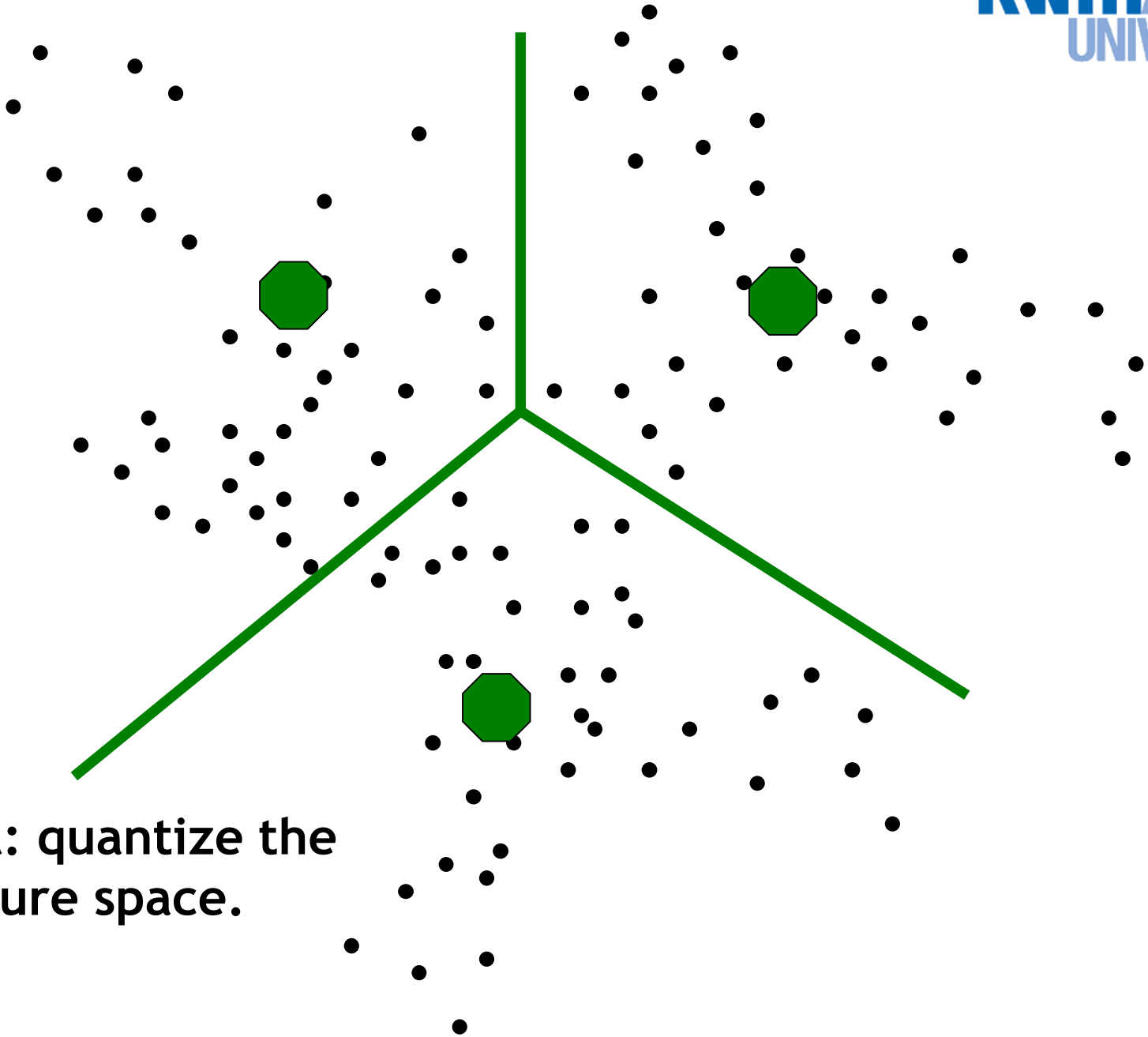


# Visual Words: Main Idea





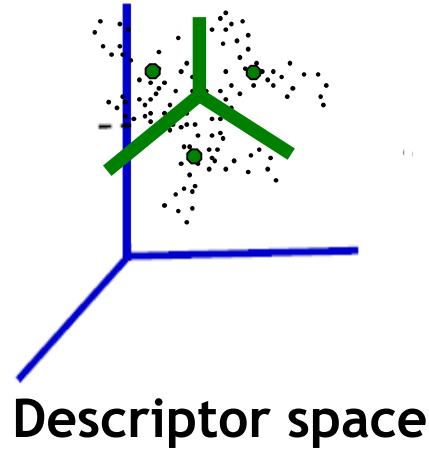
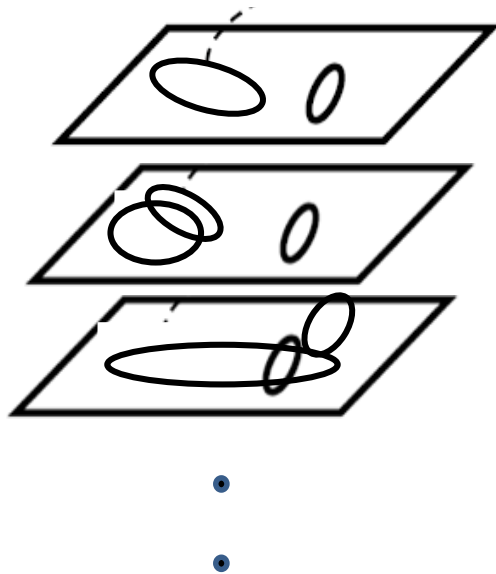
Each point is a  
local descriptor,  
e.g. SIFT vector.



Idea: quantize the  
feature space.

# Indexing with Visual Words

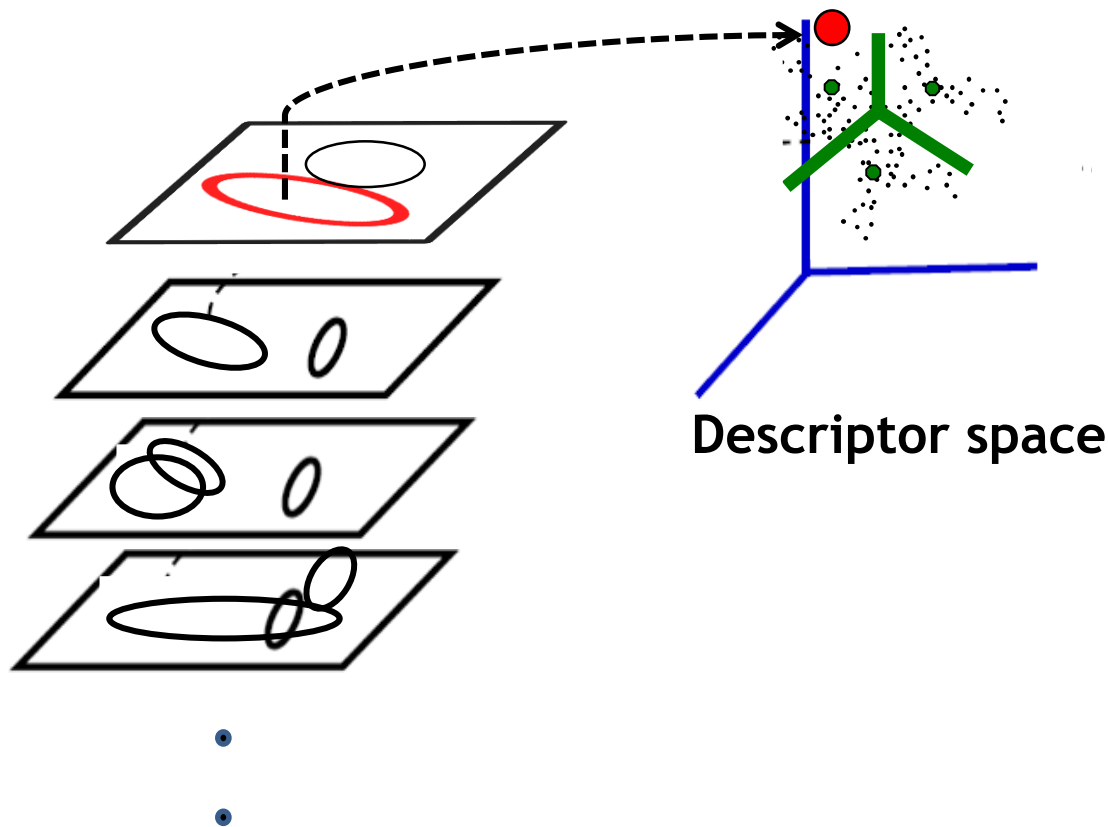
Map high-dimensional descriptors to tokens/words by quantizing the feature space



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

# Indexing with 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

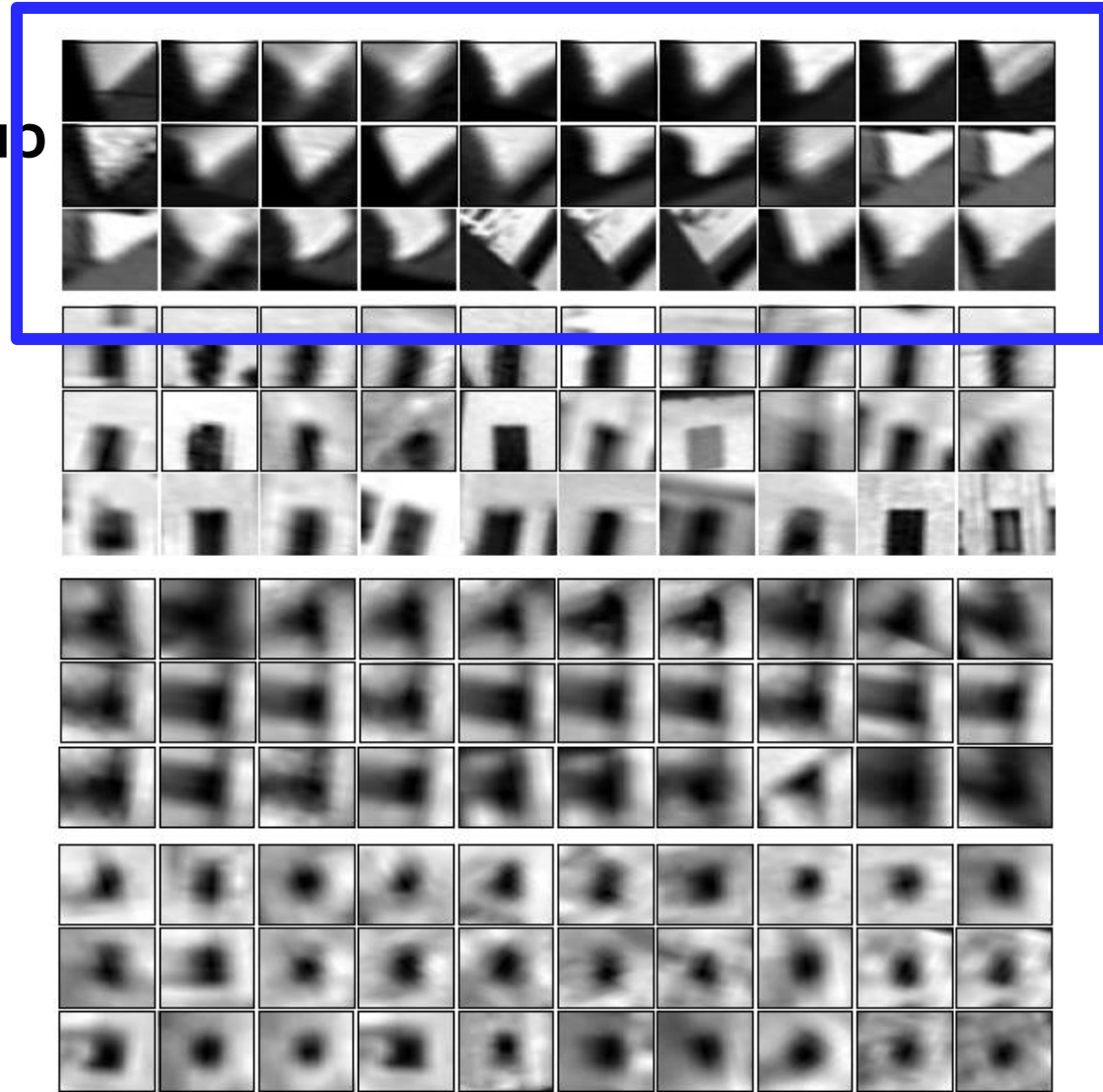
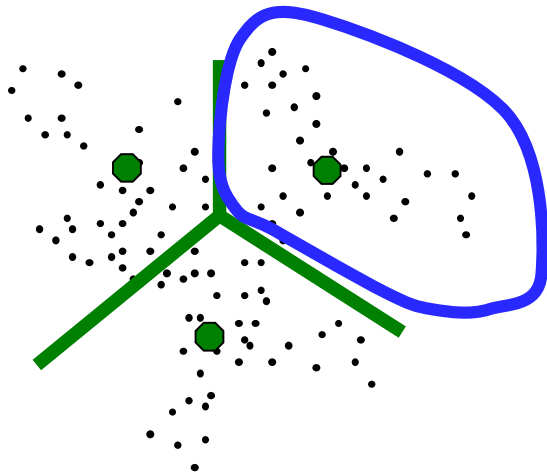
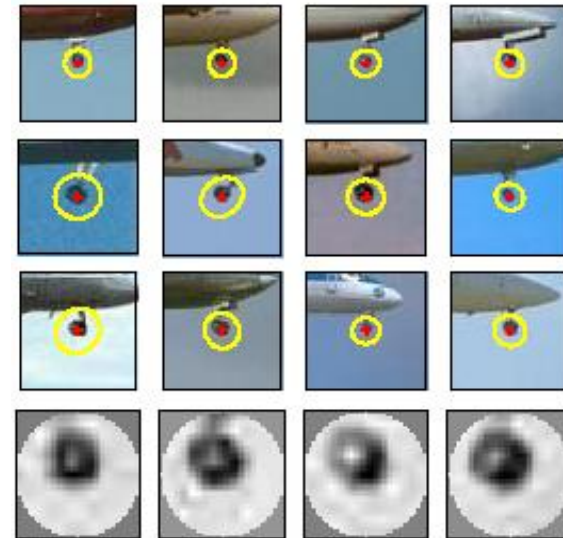


Figure from Sivic & Zisserman, ICCV 2003

# Visual Words

- Often 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 for Images of Visual Words

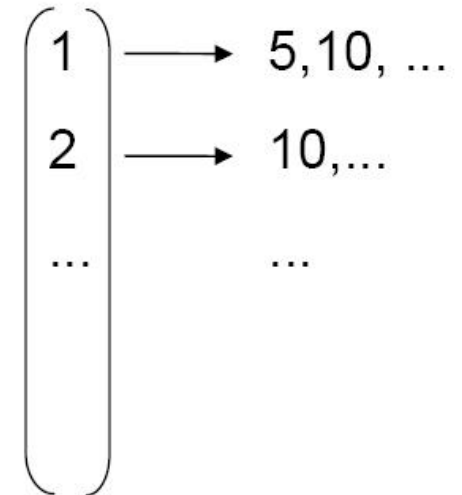


frame #5



frame #10

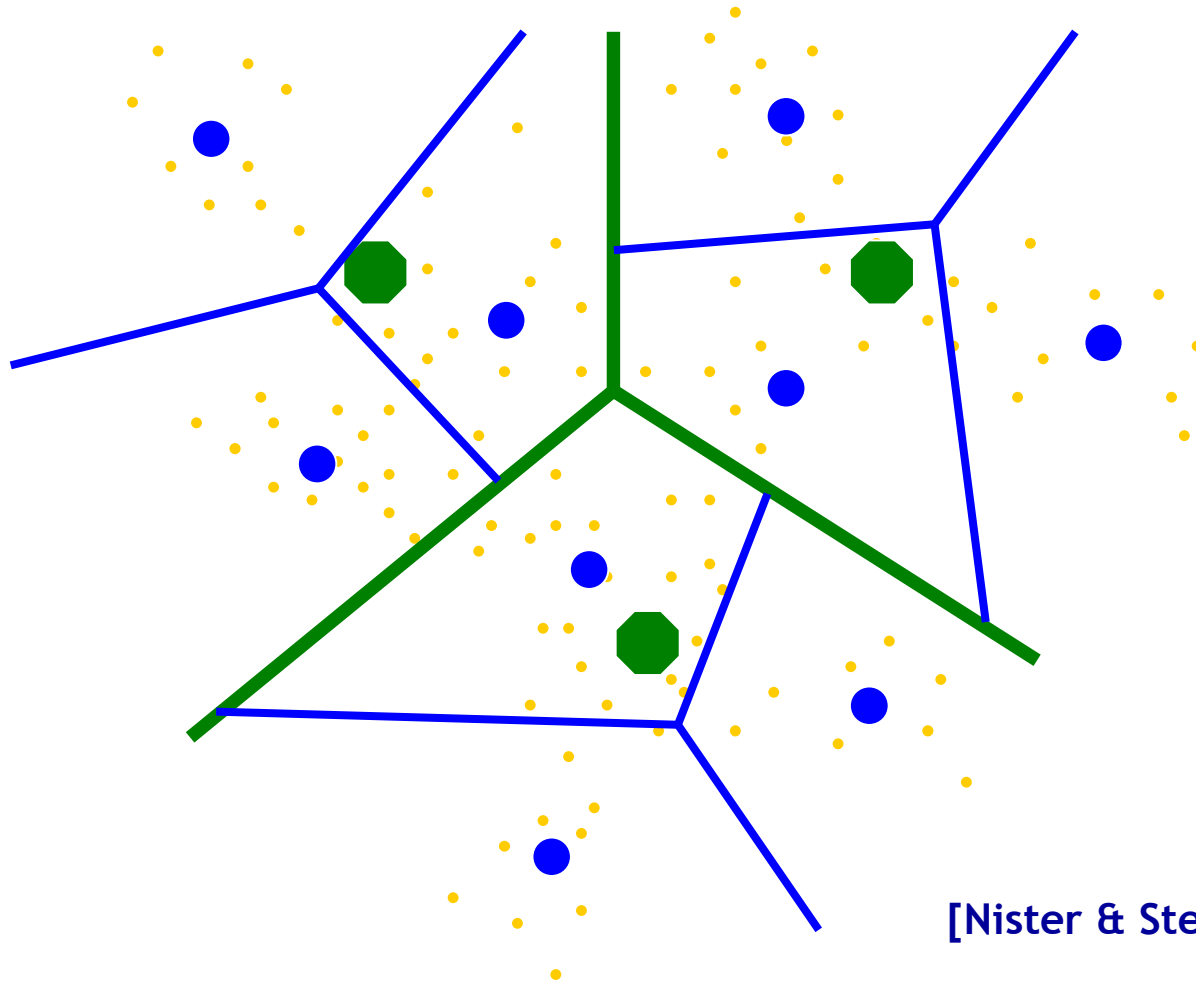
Word number	List of image numbers
1	5, 10, ...
2	10, ...
...	...



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

# Example: Recognition with Vocabulary Tree

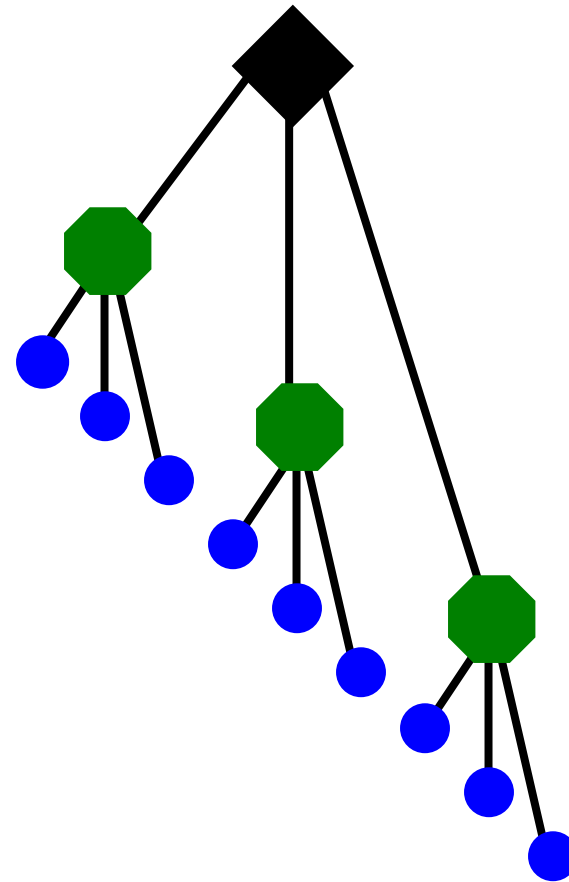
- Tree construction:



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

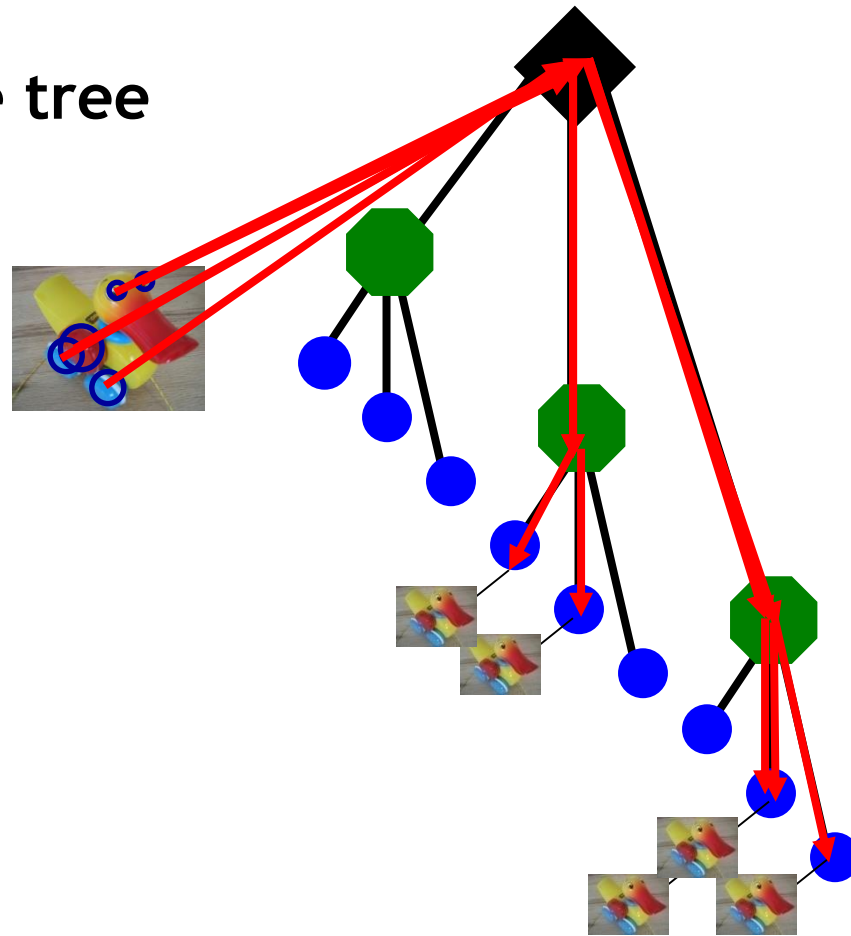
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

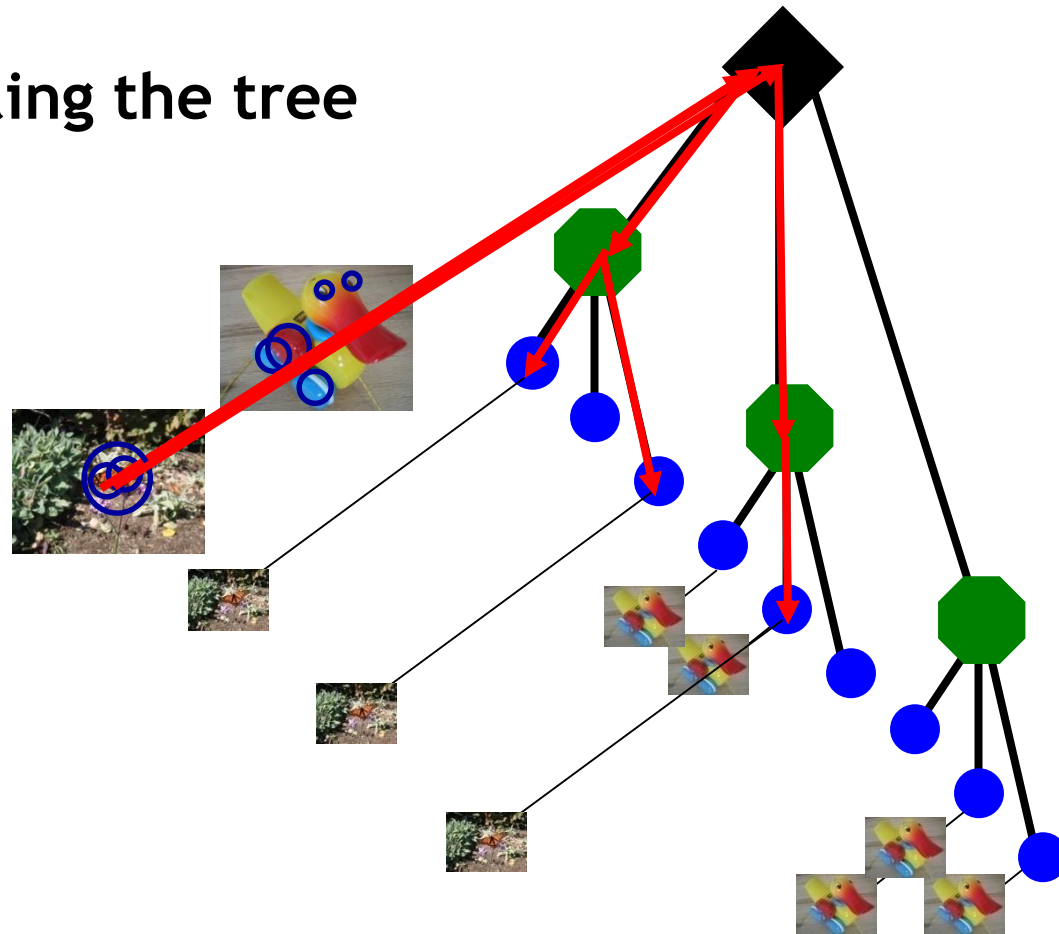
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

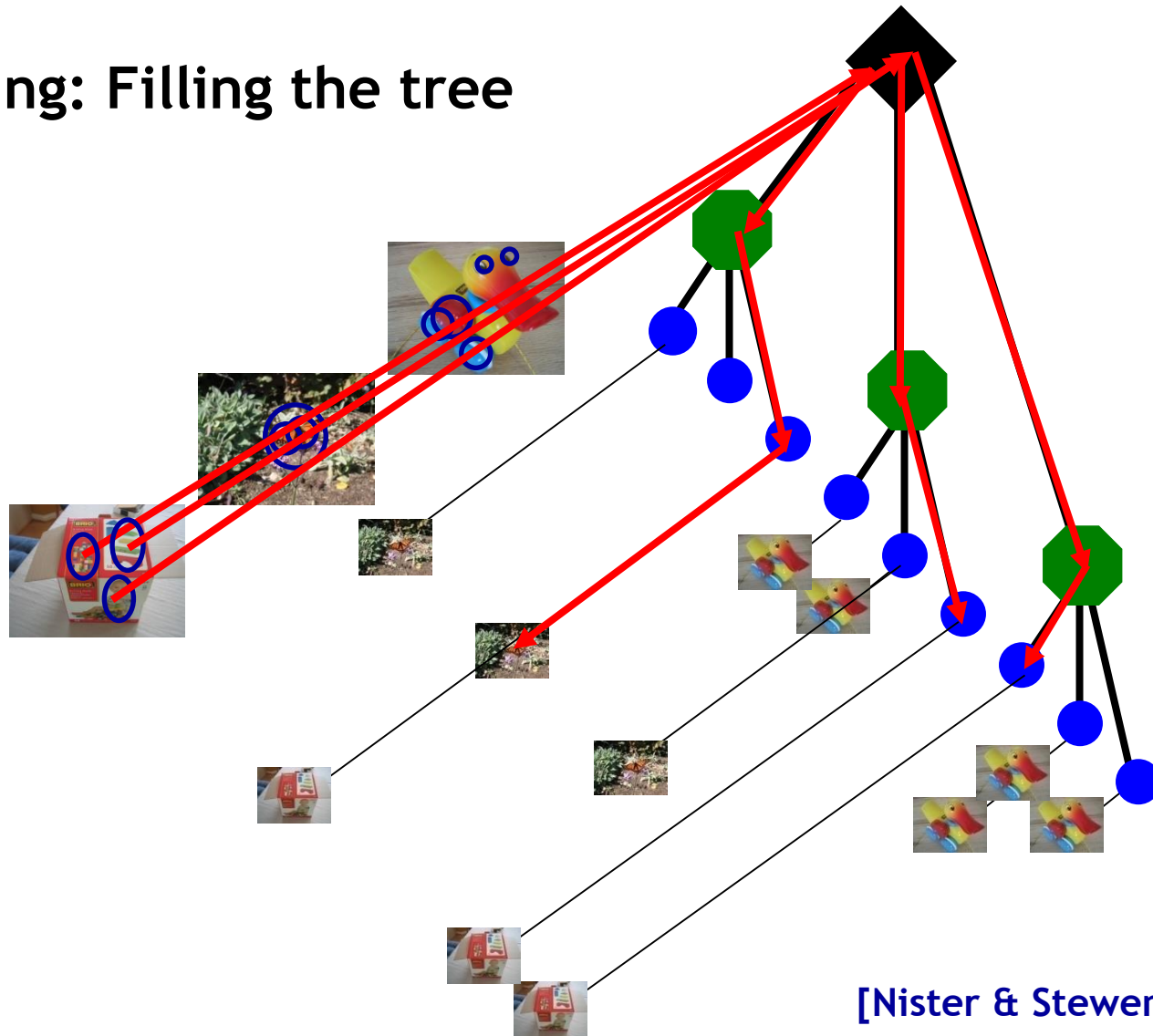
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

- Training: Filling the tree

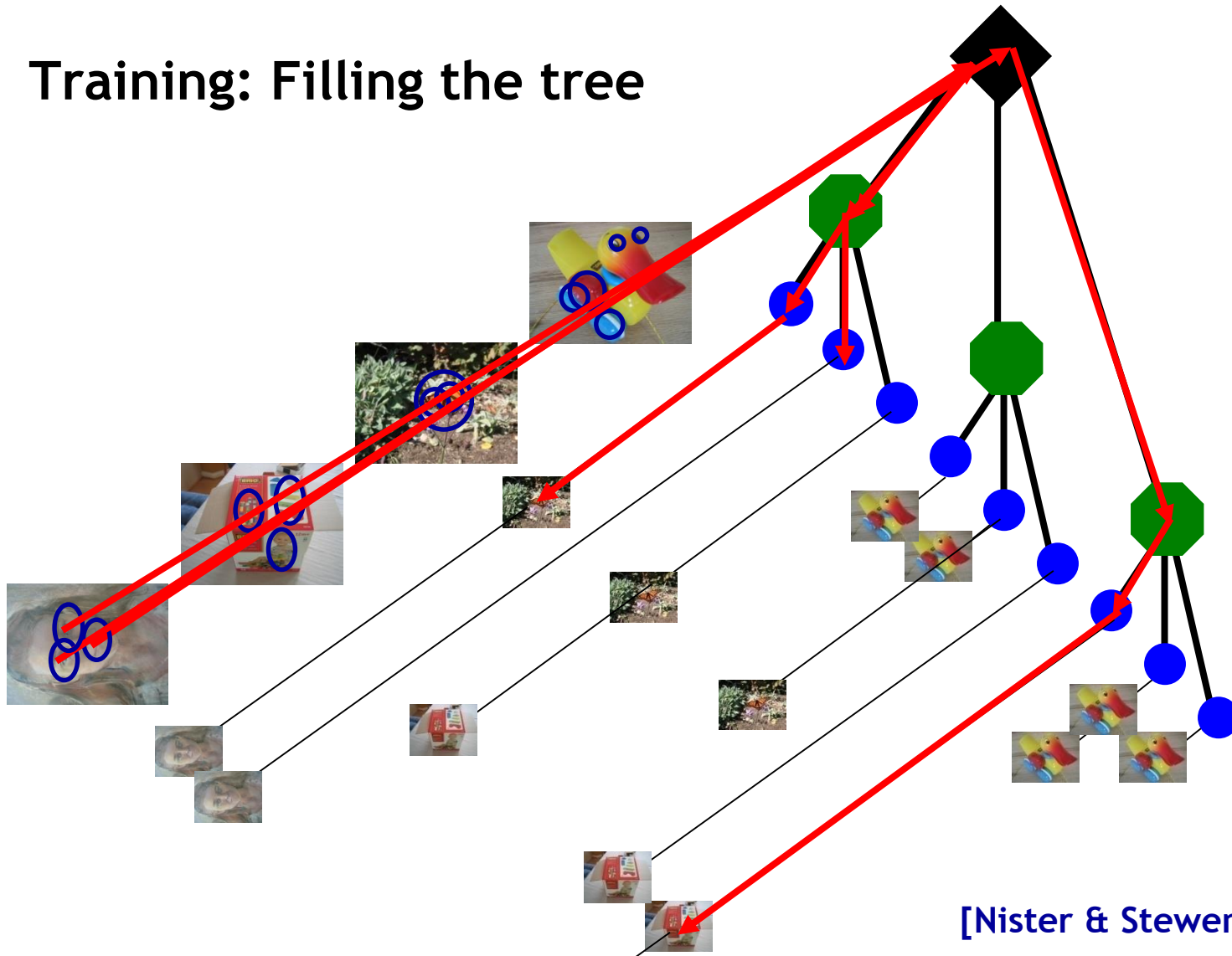


[Nister & Stewenius, CVPR'06]



# Vocabulary Tree

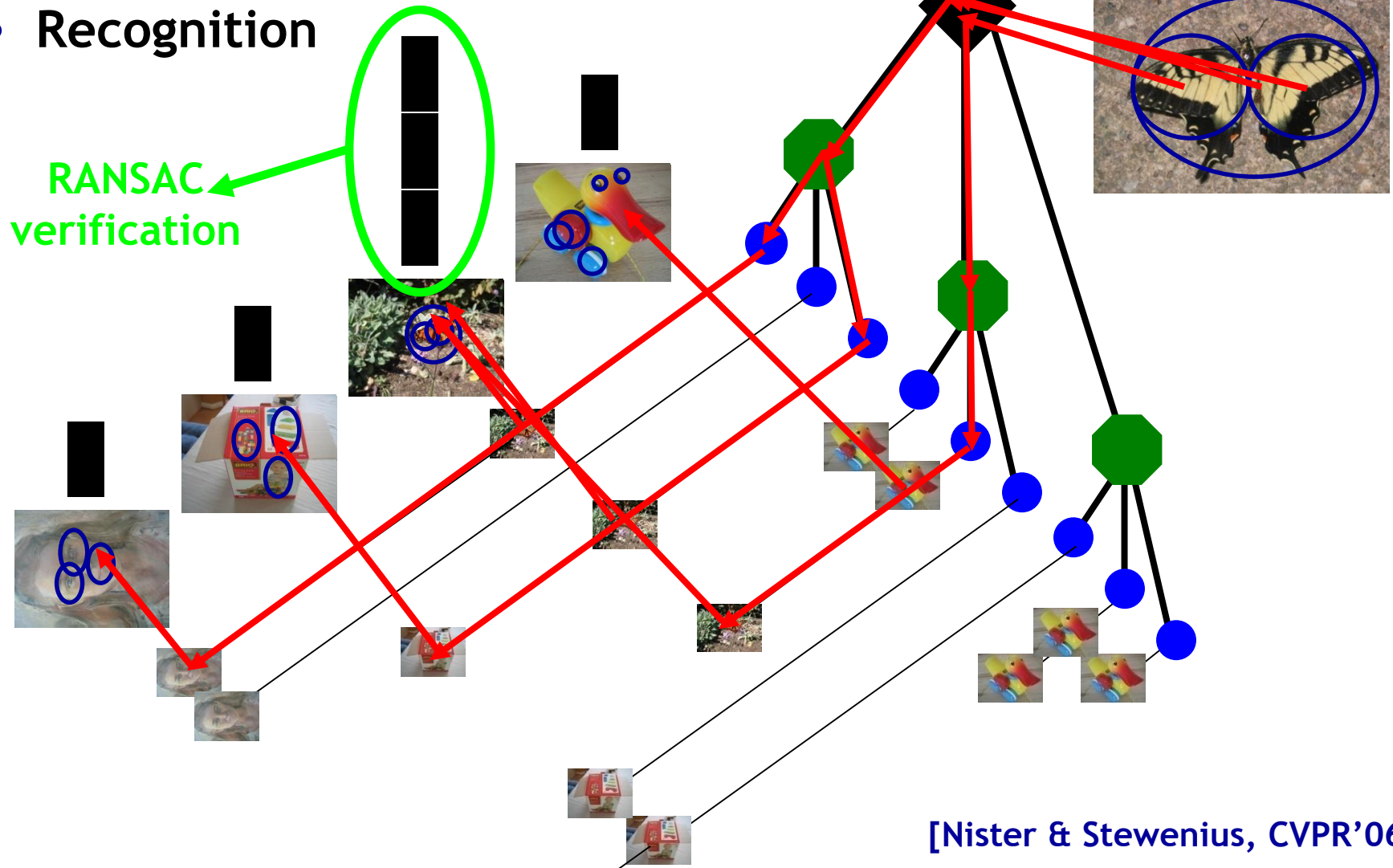
- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

# Vocabulary Tree

- Recognition



[Nister & Stewenius, CVPR'06]

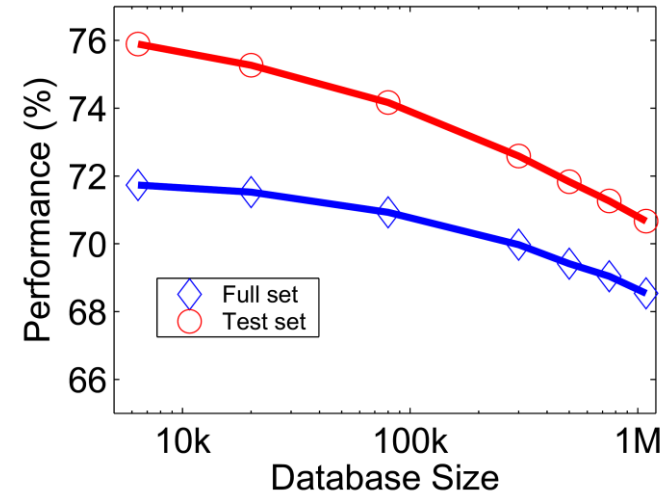
# Quiz Questions

- What is the computational advantage of the hierarchical representation vs. a flat vocabulary?
- What dangers does such a representation carry?

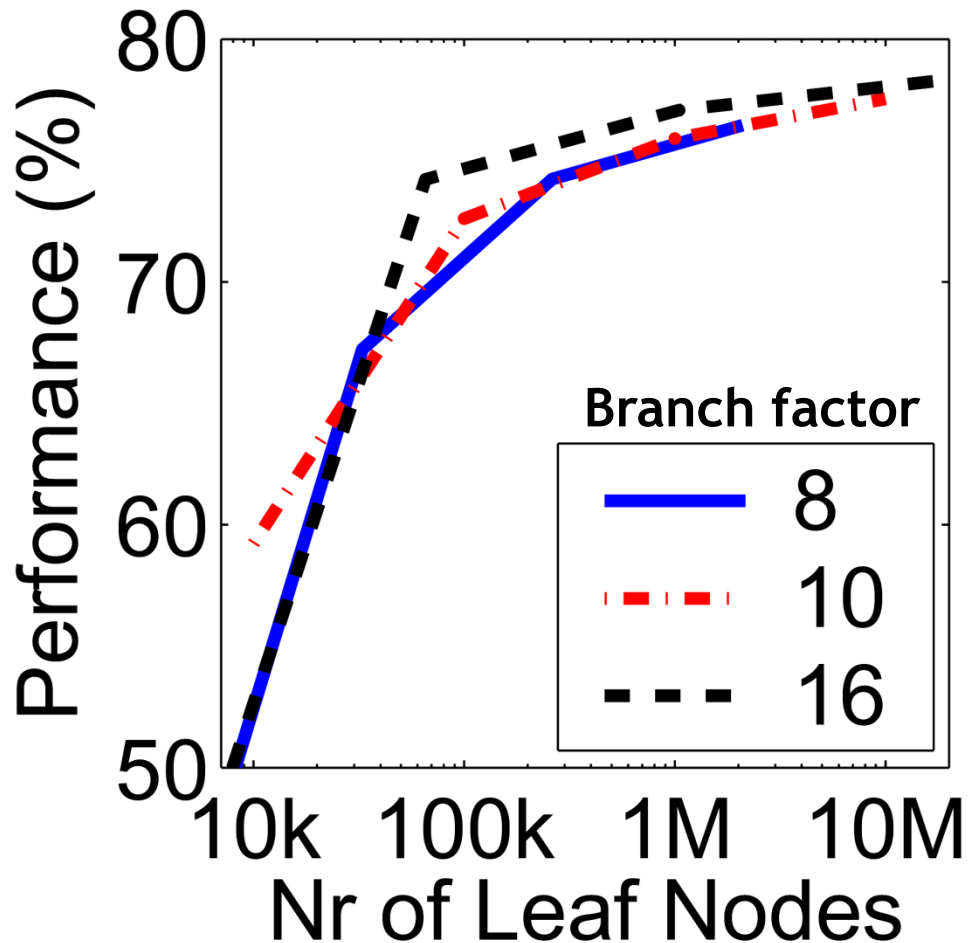
# Vocabulary Tree: Performance

- Evaluated on large databases
  - Indexing with up to 1M images
- Online recognition for database of 50,000 CD covers
  - Retrieval in ~1s (in 2006)
- Experimental finding that large vocabularies can be beneficial for recognition

[Nister & Stewenius, CVPR'06]



# Vocabulary Size



- Larger vocabularies can be advantageous...
- But what happens when the vocabulary gets too large?
  - Efficiency?
  - Robustness?

# 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)

Number of occurrences of word  $i$  in document  $d$

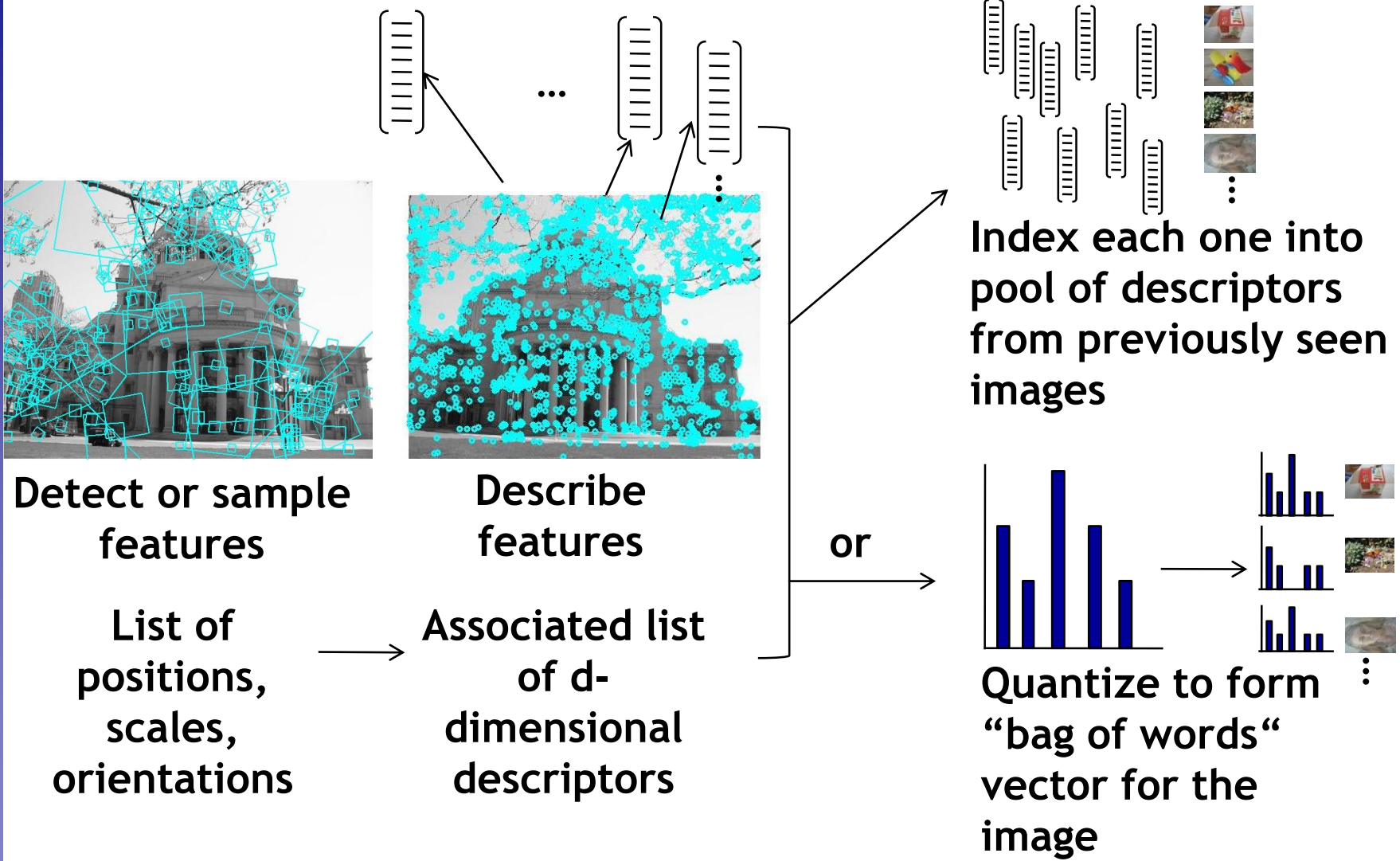
Number of words in document  $d$

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

Total number of documents in database

Number of occurrences of word  $i$  in whole database

# Summary: Indexing features



# Application for Content Based Img Retrieval

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

Visually defined query

“Find this  
clock”



“Find this  
place”



“Groundhog Day” [Rammis, 1993]





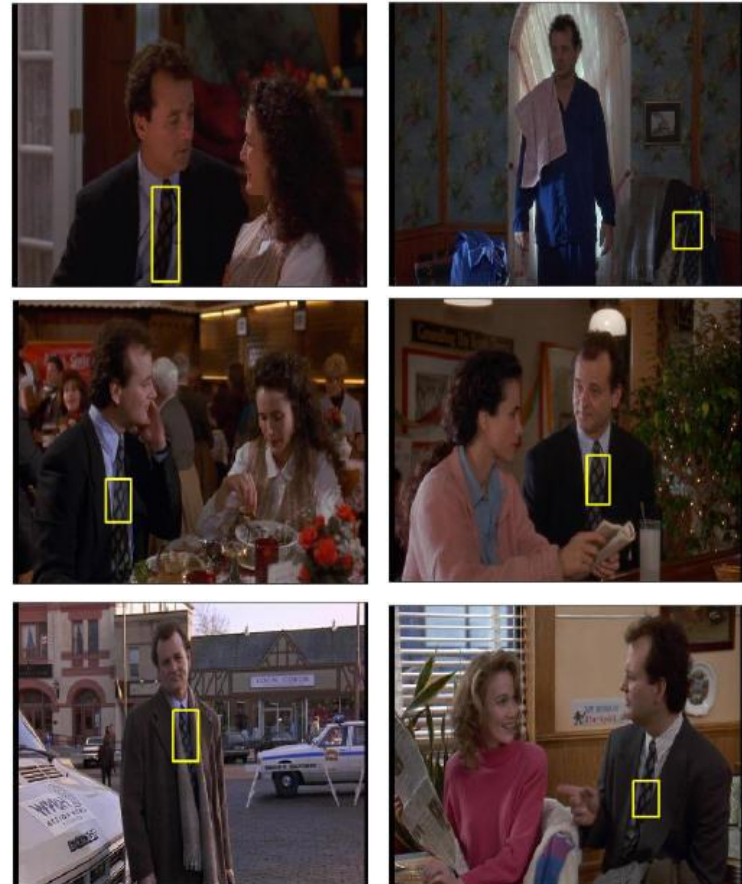
# Video Google System

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



Query  
region



Retrieved frames

# Collecting Words Within a Query Region

- Example: Friends



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

# Example Results



Query

raw nn 1sim=0.56697

raw nn 2sim=0.56163

raw nn 5sim=0.54917



# More Results



Query

raw nn 1sim=0.67818

raw nn 2sim=0.66144

raw nn 3sim=0.66023

raw nn 4sim=0.65774

raw nn 5sim=0.65463

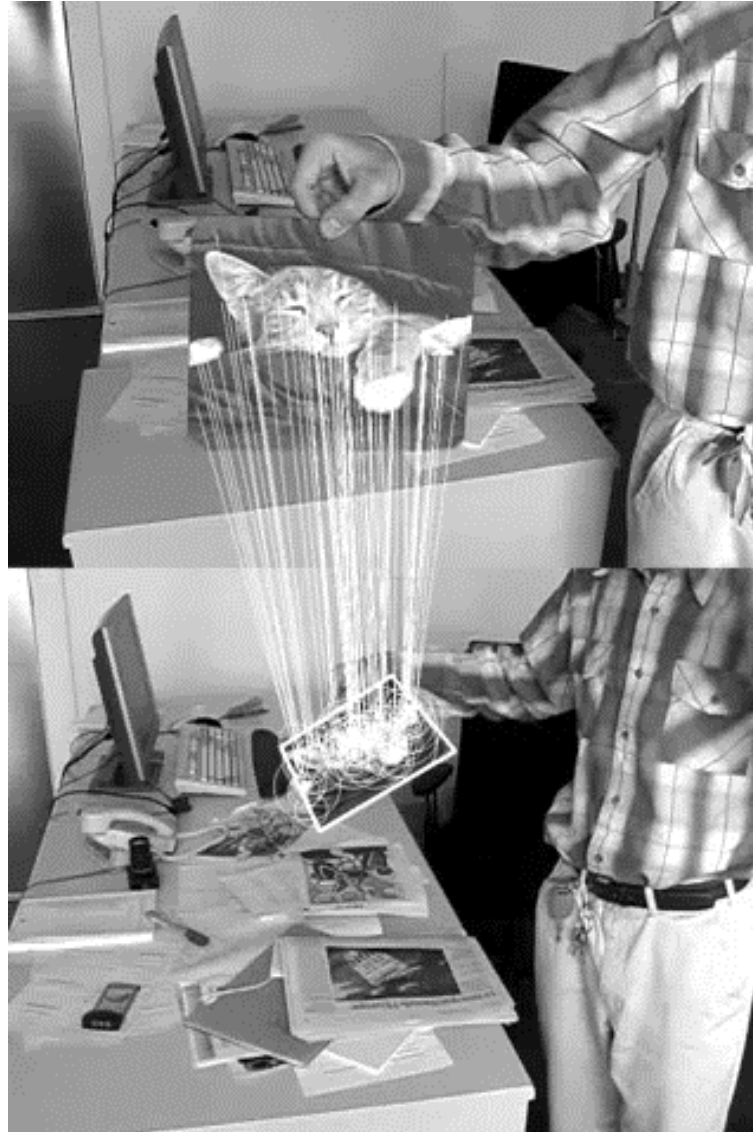


Retrieved shots

# Applications: Aachen Tourist Guide



# Applications: Fast Image Registration



# Applications: Mobile Augmented Reality

Mobile Phone  
Augmented Reality  
at  
30 Frames per Second  
using  
Natural Feature Tracking  
(all processing and rendering done in software)

D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg,  
[Pose Tracking from Natural Features on Mobile Phones](#). In *ISMAR 2008*.

# Topics of This Lecture

- Indexing with Local Features
  - Inverted file index
  - Visual Words
  - Visual Vocabulary construction
  - tf-idf weighting
- **Bag-of-Words Model**
  - Use for image classification



# 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 the brain receives from our eyes. For many years it was thought that the retina was the point by which the visual information enters the brain; the discovery of the screen in the eye led to the discovery of the eye, cell, optical nerve, image Hubel, Wiesel

considerable amount of information about the visual world is processed along their path through the layers of the optical cortex, Hubel and Wiesel have been able to demonstrate that the message about the image falling on the retina undergoes a step-wise analysis by 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.

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 30% jump in exports to \$100bn, with a 18% rise in imports. The figures are likely to be revised upwards as China has long complained that the US has long had an unfair trade policy. The trade surplus is expected to reach \$100bn only on the assumption that the yuan, bank, domestic, foreign, increase, trade, value

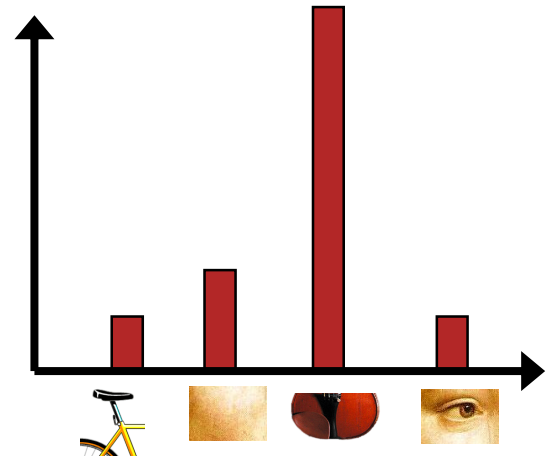
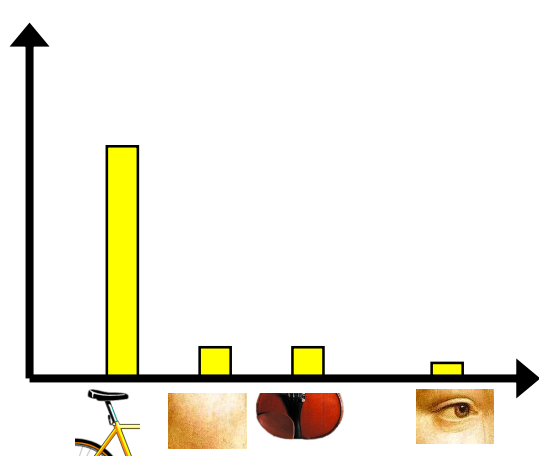
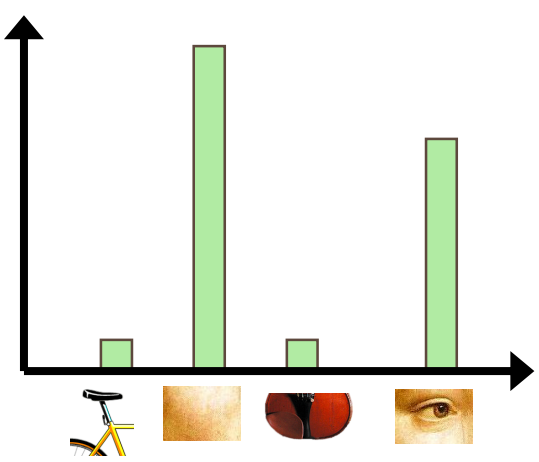
Zhou Xiaochuan, governor of the People's Bank of China, said the government needed to control the yuan's value to meet the demand so many firms are struggling in the country. China increased the value of the yuan against the dollar by 2.1% in 2005 and permitted it to trade within a narrow band, but the 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.

Object



Bag of 'words'

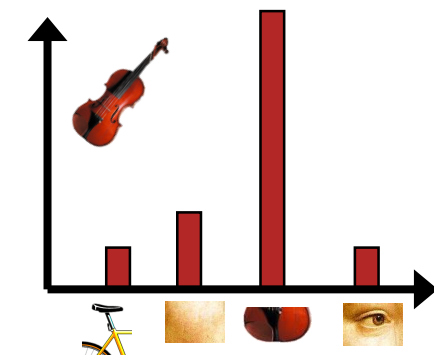
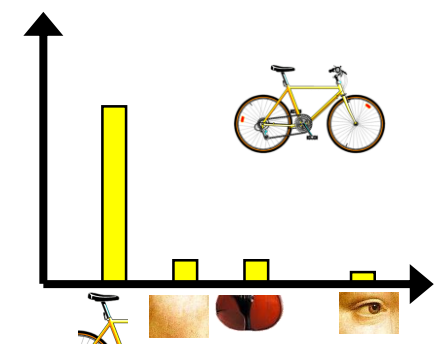
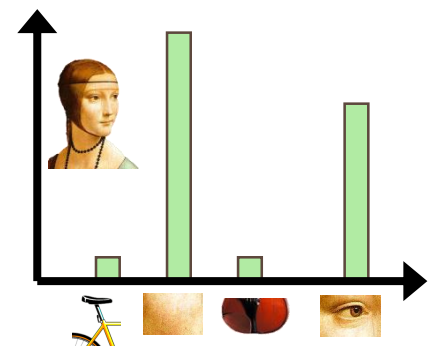




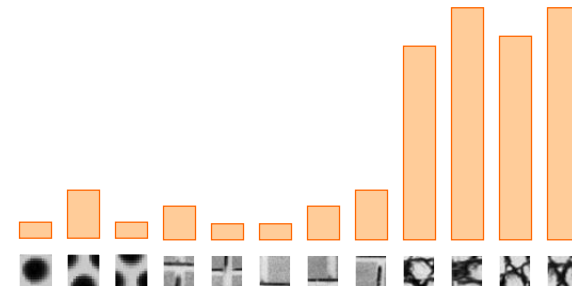
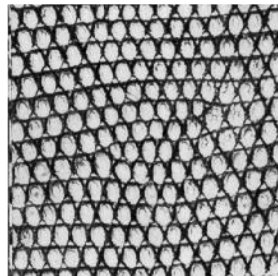
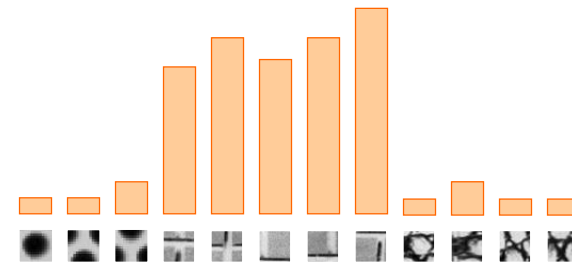
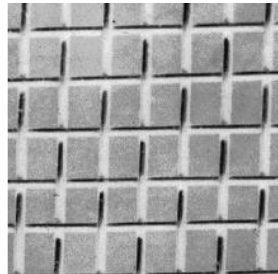
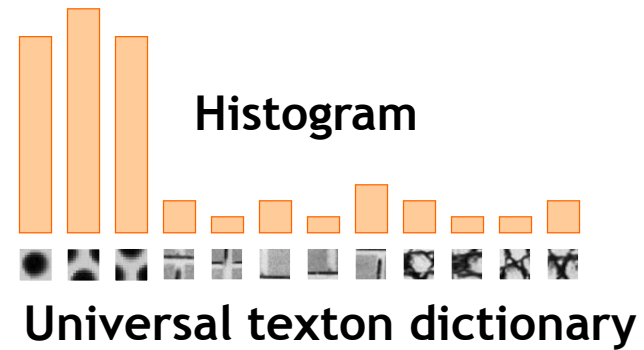
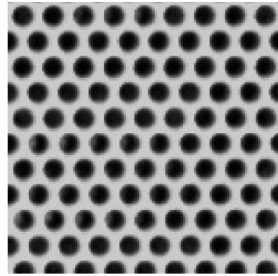


# Bags of Visual Words

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



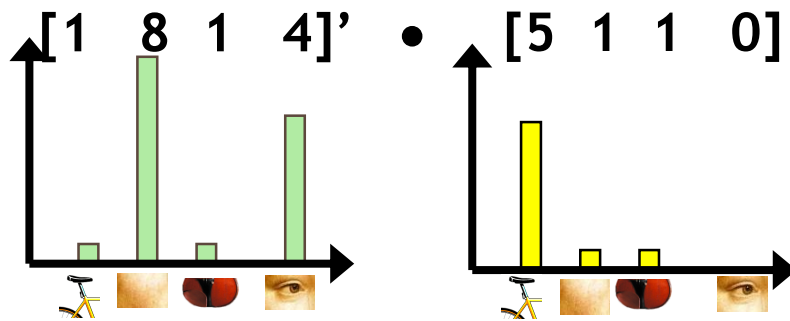
# Similarly, Bags-of-Textons for Texture Repr.



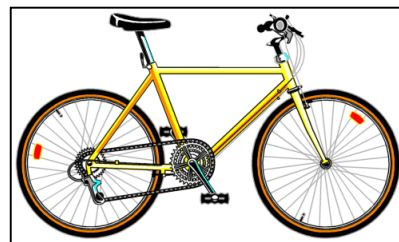
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

# Comparing Bags of Words

- We build up histograms of word activations, so any histogram comparison measure can be used here.
- E.g. we can rank frames by normalized scalar product between their (possibly weighted) occurrence counts
  - *Nearest neighbor* search for similar images.



$\vec{d}_j$

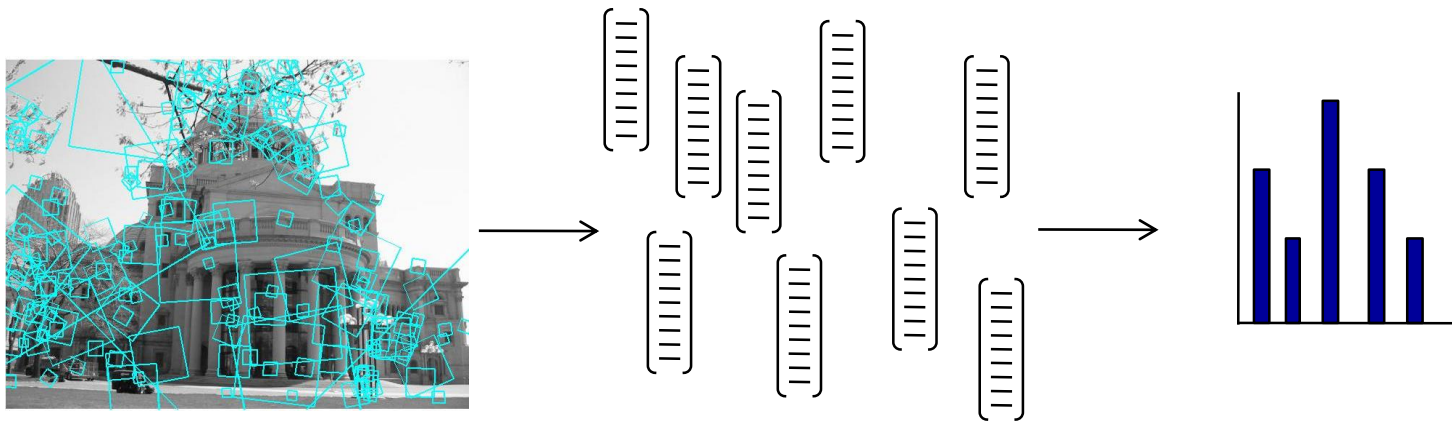


$\vec{q}$

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

# Learning/Recognition with BoW Histograms

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



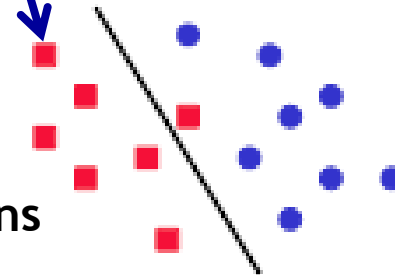
- Provides easy way to use distribution of feature types with various learning algorithms requiring vector input.

# Bags-of-Words for Classification



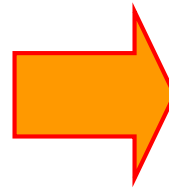
- Compute the word activation histogram for each image.
- Let each such BoW histogram be a feature vector.
- Use images from each class to train a classifier (e.g., an SVM).

Violins





# BoW for Object Categorization



{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)

# BoW for Object Categorization

## Caltech6 dataset

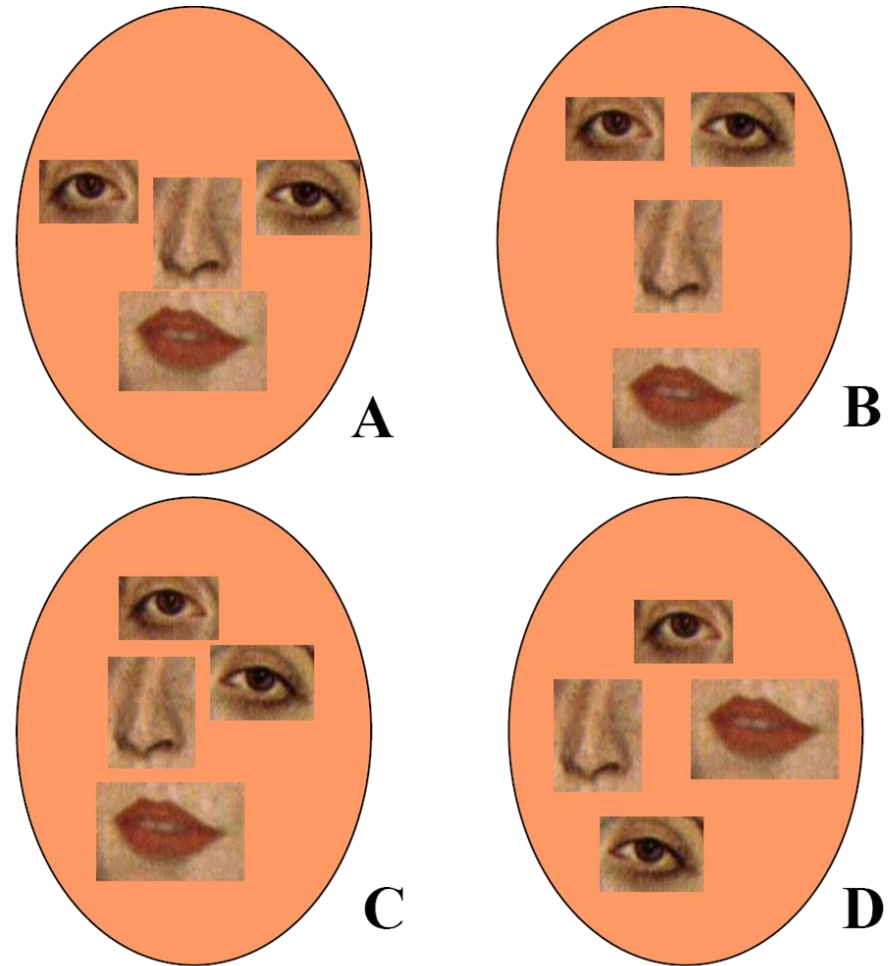


class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	<b>98.8</b>	97.1	90.2
cars (rear)	98.3	<b>98.6</b>	90.3
cars (side)	<b>95.0</b>	87.3	88.5
faces	<b>100</b>	99.3	96.4
motorbikes	<b>98.5</b>	98.0	92.5
spotted cats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

- **Good performance for pure classification (object present/absent)**
  - Better than more elaborate part-based models with spatial constraints...
  - What could be possible reasons why?

# Limitations of BoW Representations

- The bag of words removes spatial layout.
- This is both a strength and a weakness.
- *Why a strength?*
- *Why a weakness?*

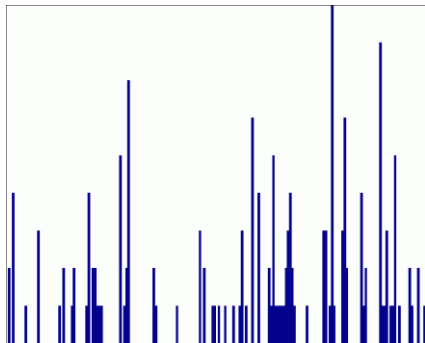


# BoW Representation: Spatial Information

- 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?)

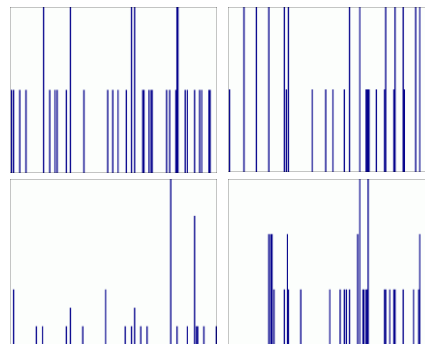
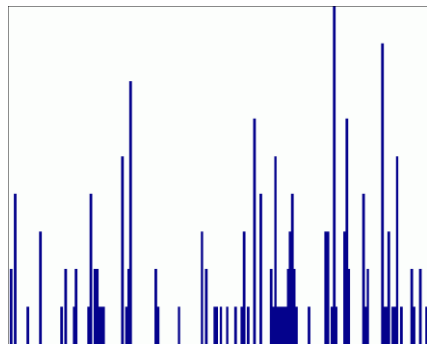
# Spatial Pyramid Representation

- Representation in-between orderless BoW and global appearance



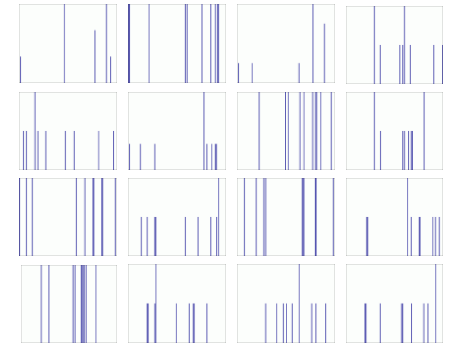
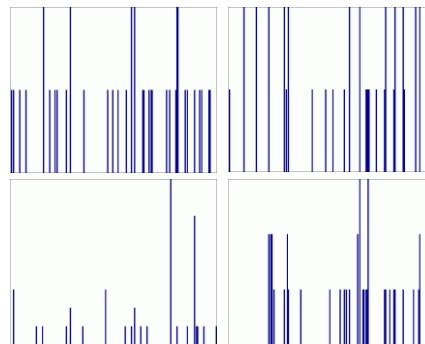
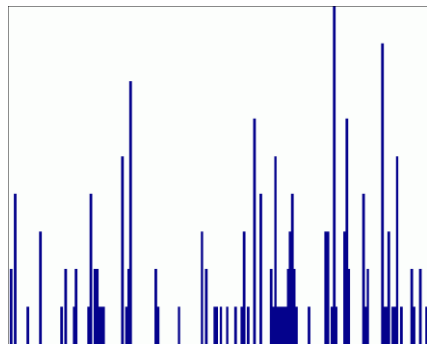
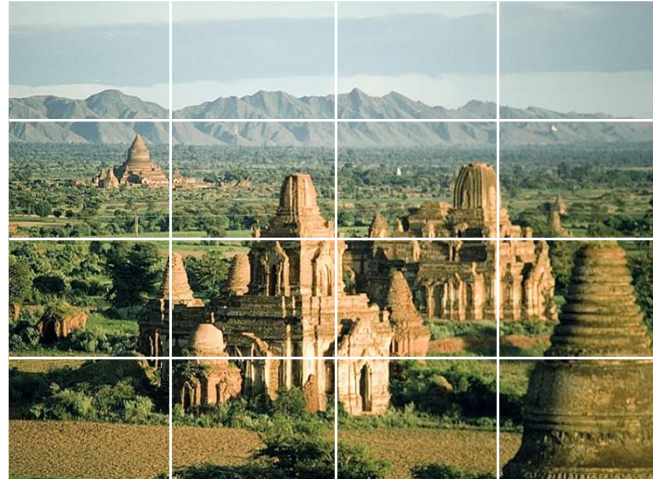
# Spatial Pyramid Representation

- Representation in-between orderless BoW and global appearance



# Spatial Pyramid Representation

- Representation in-between orderless BoW and global appearance



# Summary: Bag-of-Words

- Pros:

- Flexible to geometry / deformations / viewpoint
- Compact summary of image content
- Provides vector representation for sets
- Empirically good recognition results in practice

- Cons:

- 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.



# References and Further Reading

- More details on RANSAC can be found in Chapter 4.7 of
  - R. Hartley, A. Zisserman  
Multiple View Geometry in Computer Vision  
2nd Ed., Cambridge Univ. Press, 2004
- Details about the Hough transform for object recognition can be found in
  - D. Lowe, [Distinctive image features from scale-invariant keypoints](#),  
*IJCV* 60(2), pp. 91-110, 2004
- Details about the Video Google system can be found in
  - *J. Sivic, A. Zisserman,*  
[Video Google: A Text Retrieval Approach to Object Matching in Videos](#), ICCV'03, 2003.

