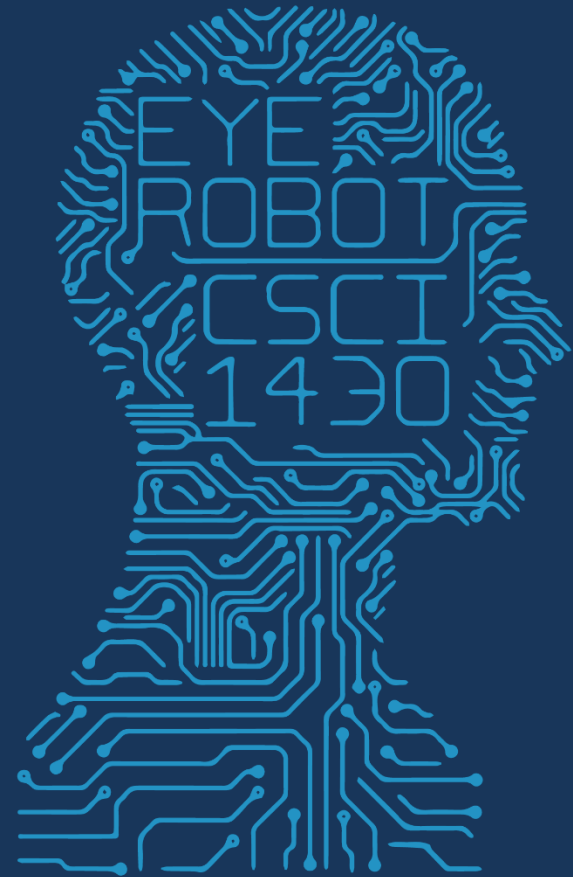




1950

FUTURE VISION



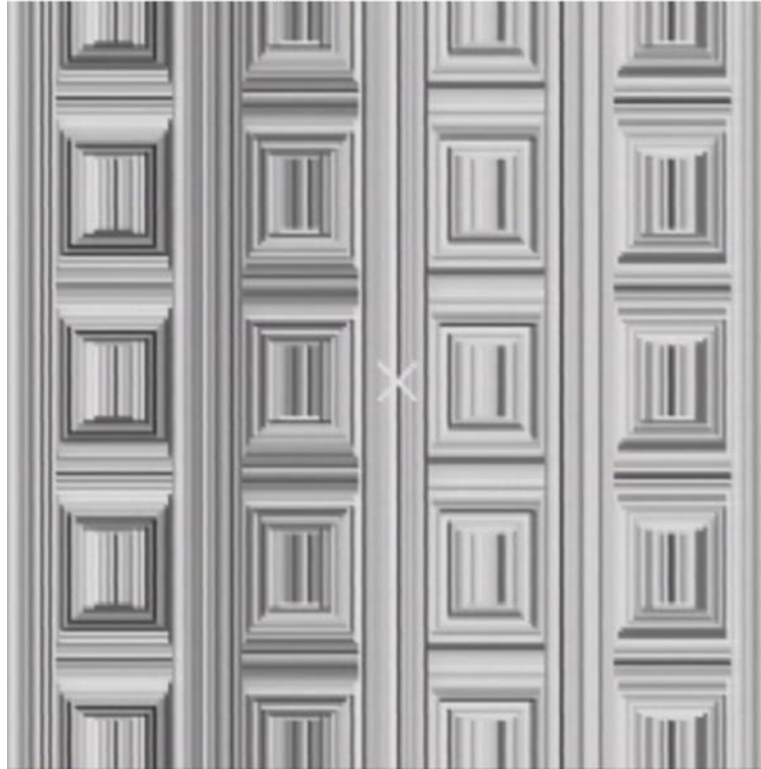
2020

COMPUTER VISION



Coffer Illusion

How many circles do you see?



# Machine Learning Problems

*Supervised Learning*

*Unsupervised Learning*

*Discrete*

classification or  
categorization

clustering

*Continuous*

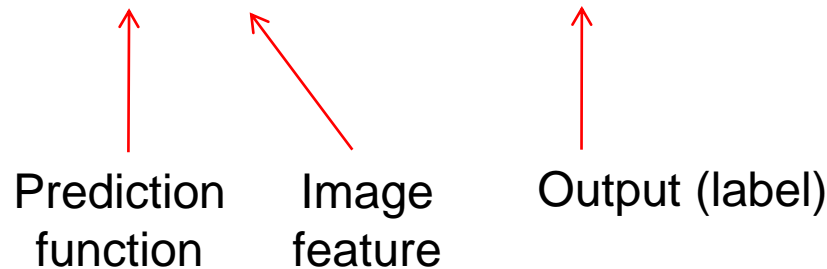
regression

dimensionality  
reduction



# Supervised learning

$$f(\mathbf{x}) = y$$



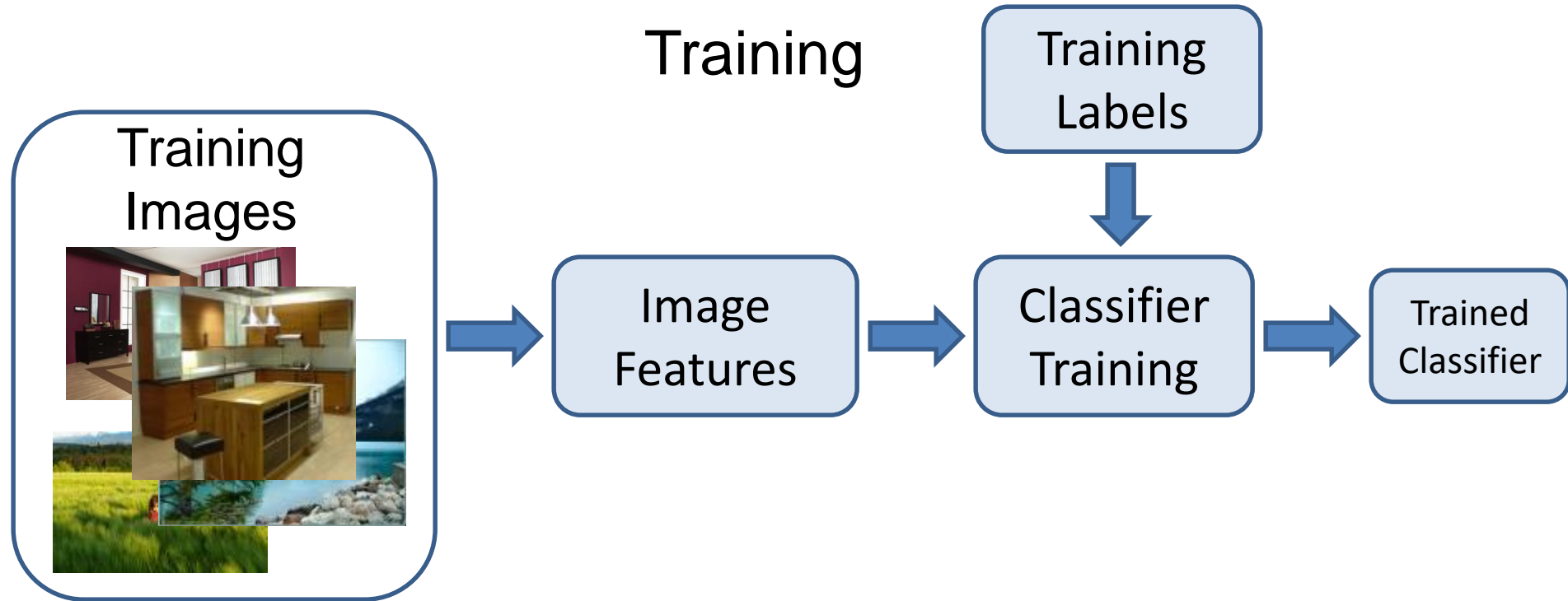
**Training:** Given a *training set* of labeled examples:

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$$

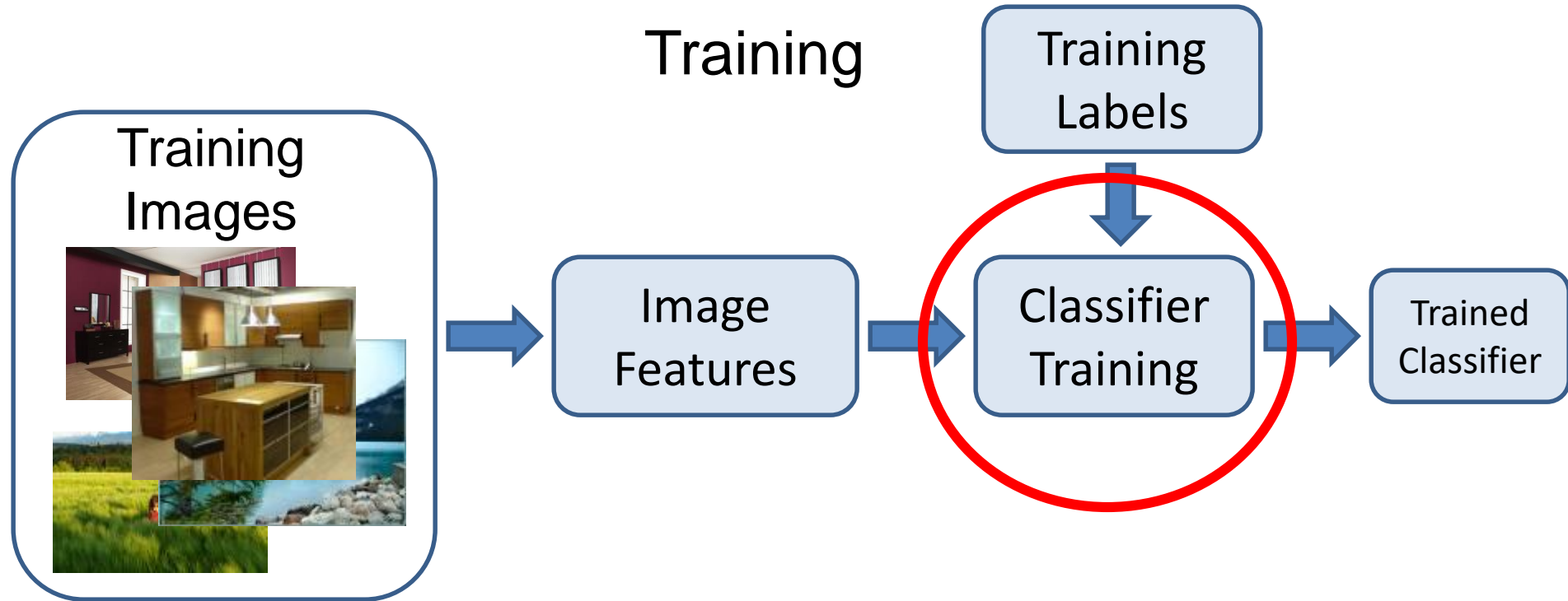
Estimate the prediction function  $f$  by minimizing the prediction error on the training set.

**Testing:** Apply  $f$  to a unseen *test example*  $\mathbf{x}$  and output the predicted value  $y = f(\mathbf{x})$  to *classify*  $\mathbf{x}$ .

# Image Categorization

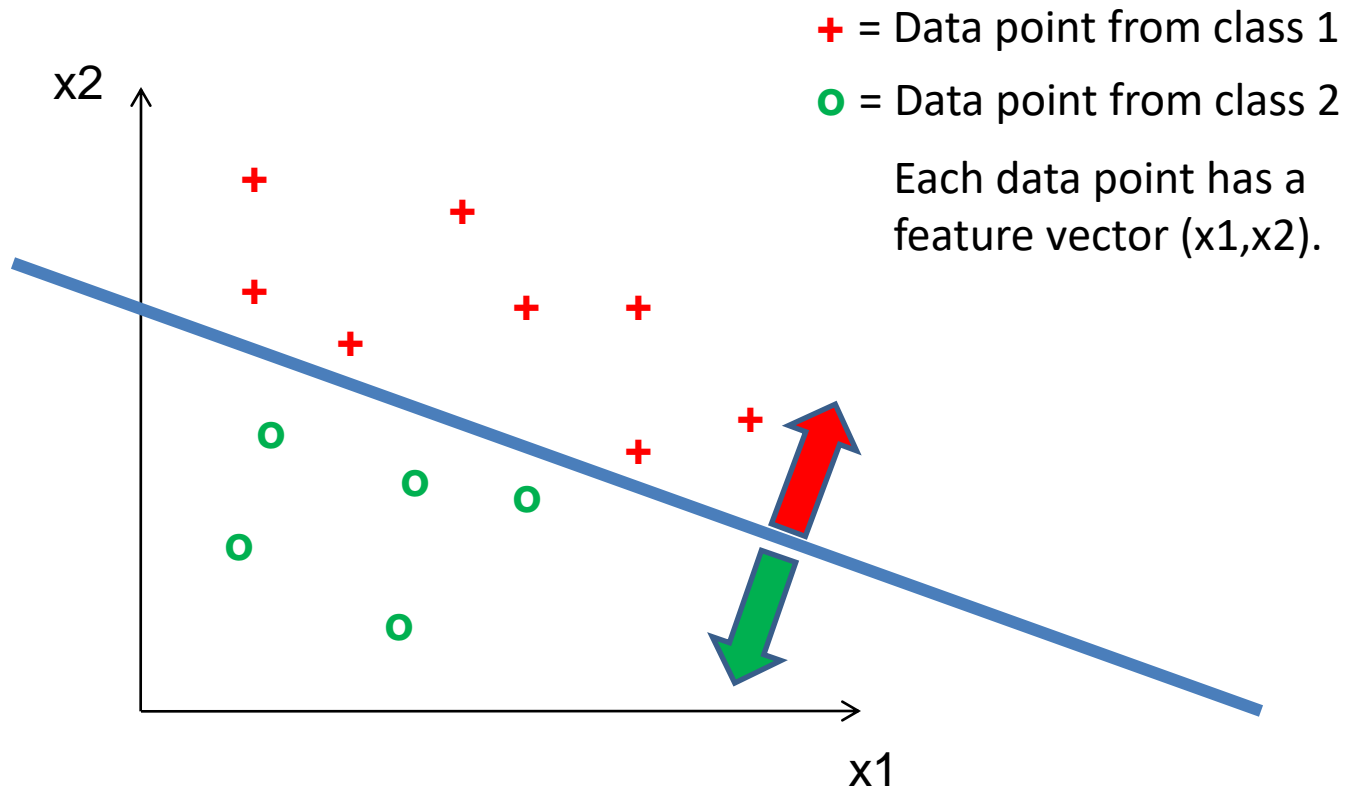


# Classifiers

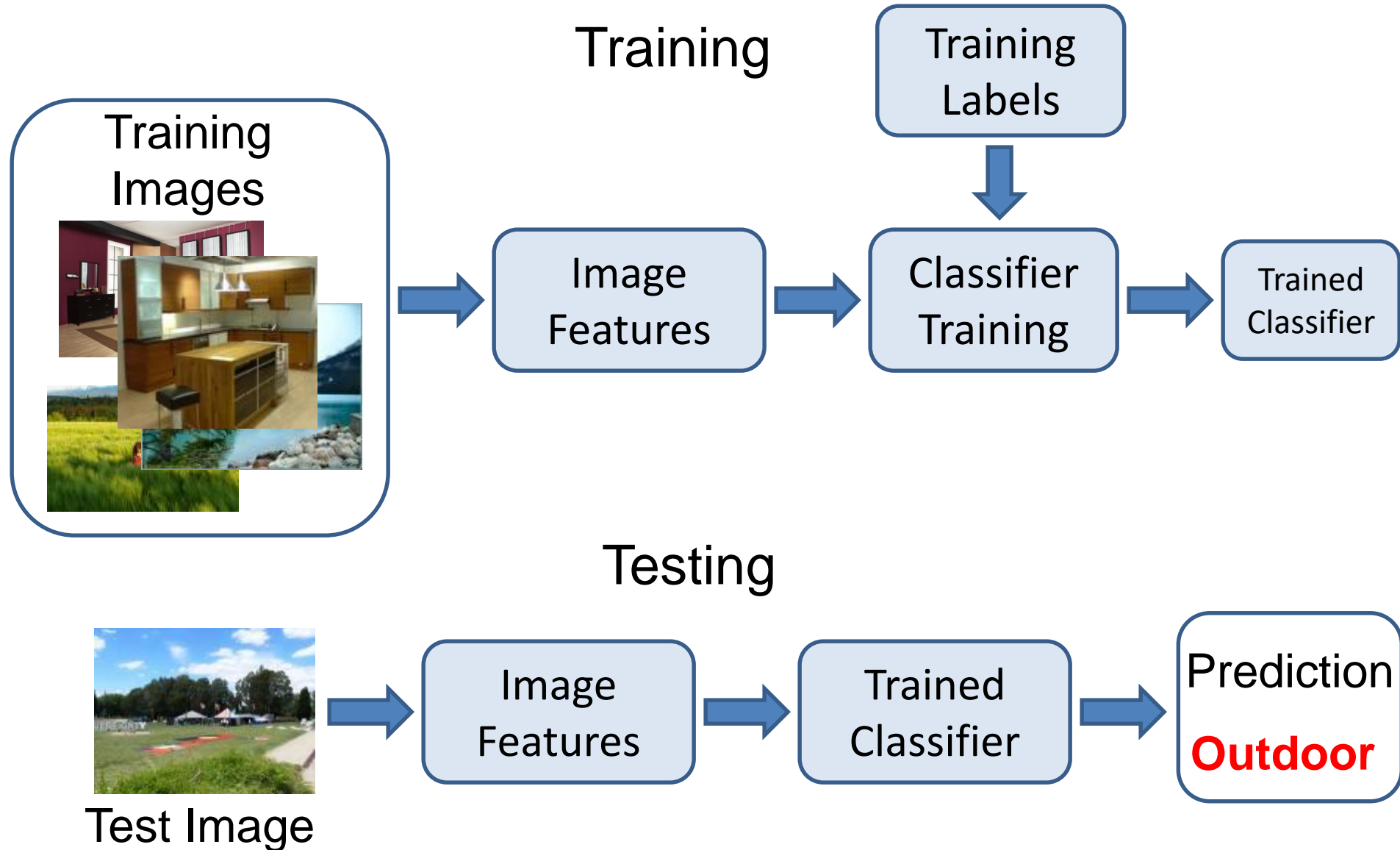


# Learning a classifier

Given a set of features with corresponding labels, learn a function to predict the labels from the features.



# Image Categorization



# Example: Scene Categorization

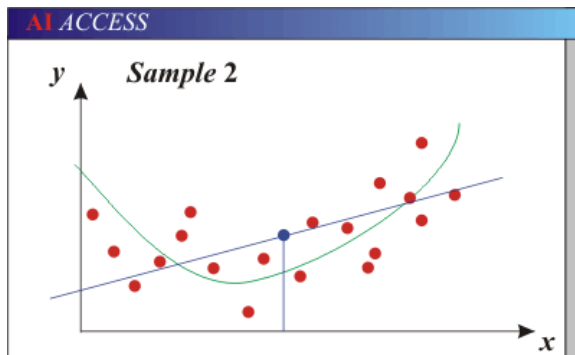
- Is this a kitchen?



# Bias-Variance Trade-off

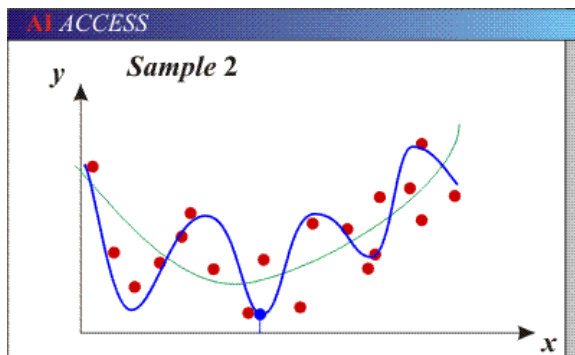
**Bias:** *error in model assumptions*; how much the average model over all training sets differs from the true model.

**Variance:** how much models estimated from different training sets differ from each other.



Models with too few parameters are inaccurate because of a large bias.

- Not enough flexibility!



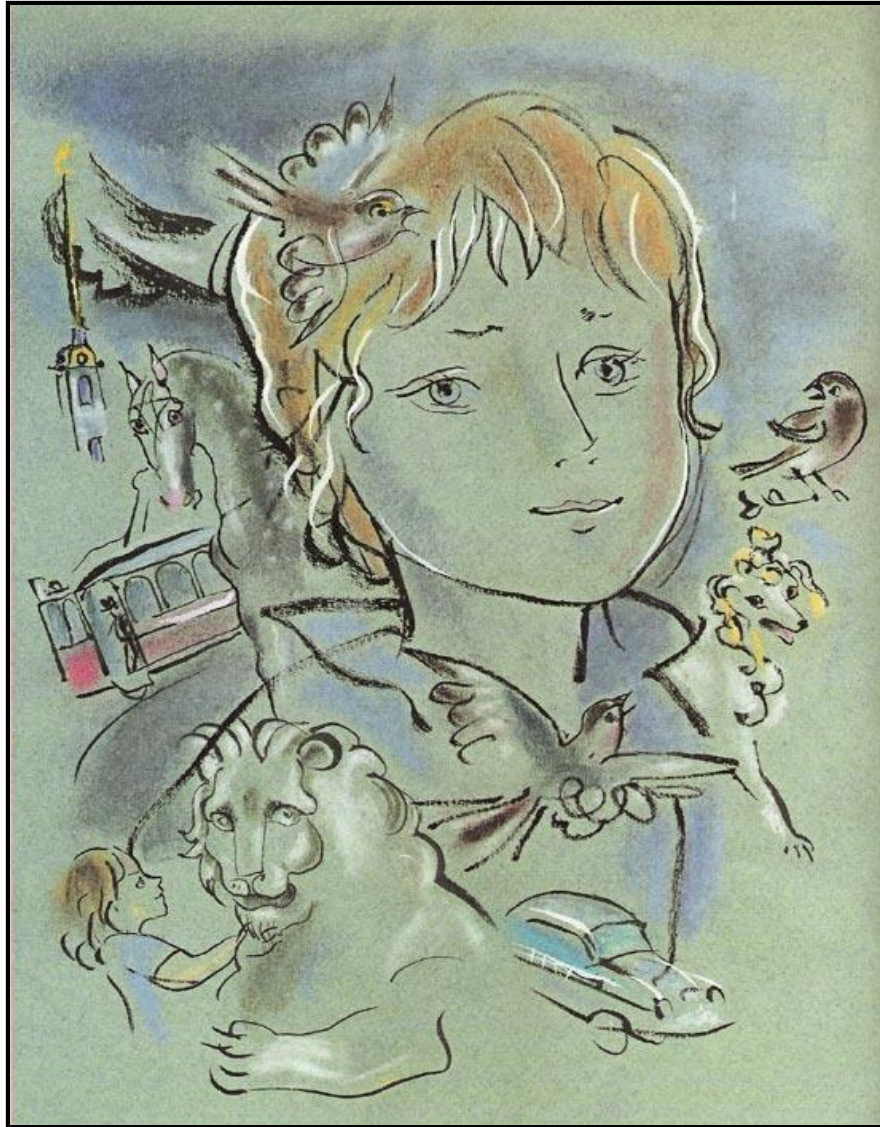
Models with too many parameters are inaccurate because of a large variance.

- Too much sensitivity to the sample.



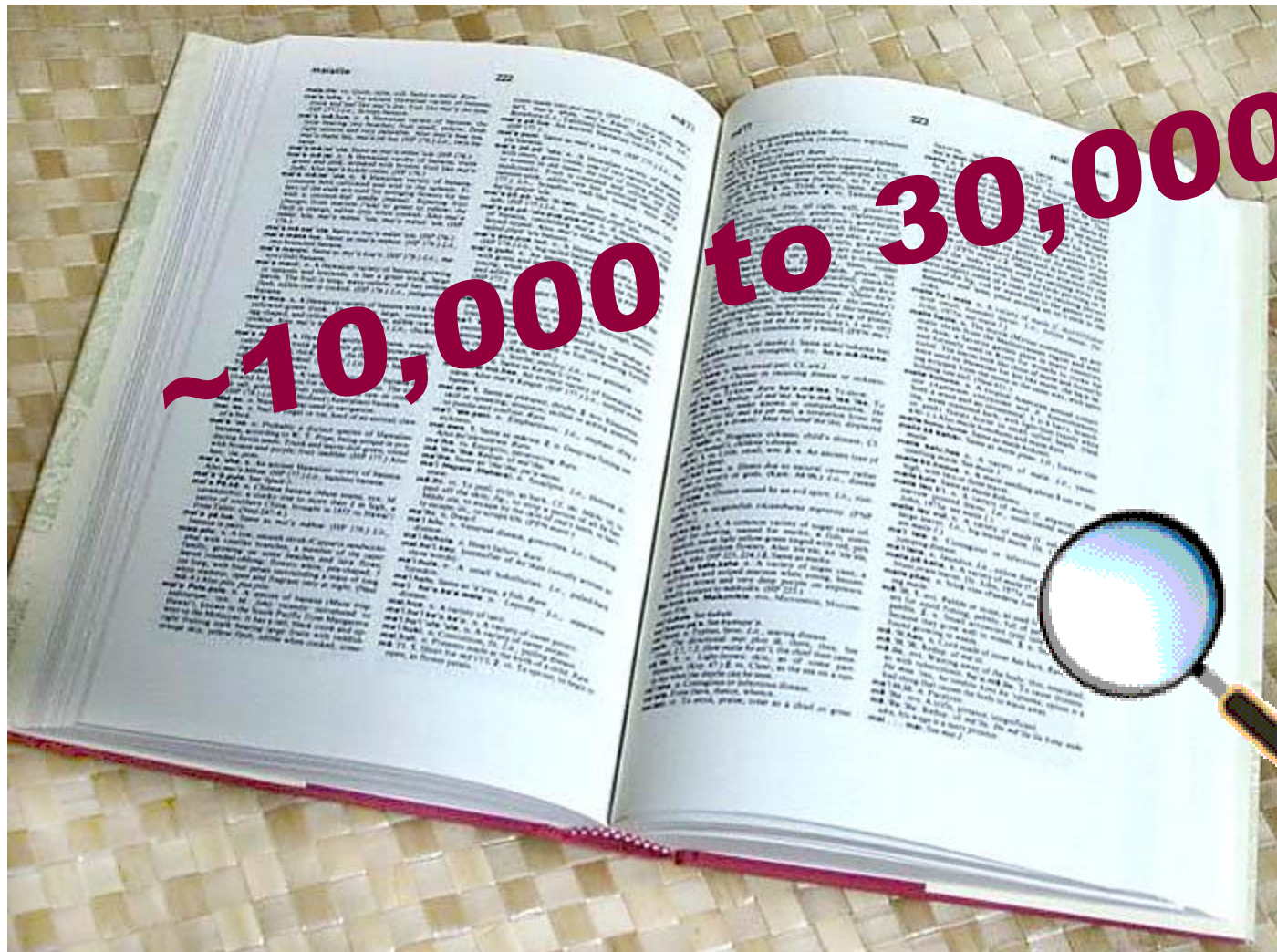
# Recognition: Overview and History

---





# How many visual object categories are there?





~10,000 to 30,000

# OBJECTS

ANIMALS

PLANTS

INANIMATE

.....

VERTEBRATE

NATURAL

MAN-MADE

MAMMALS

BIRDS

TAPIR

BOAR

GROUSE

CAMERA





# Specific recognition tasks



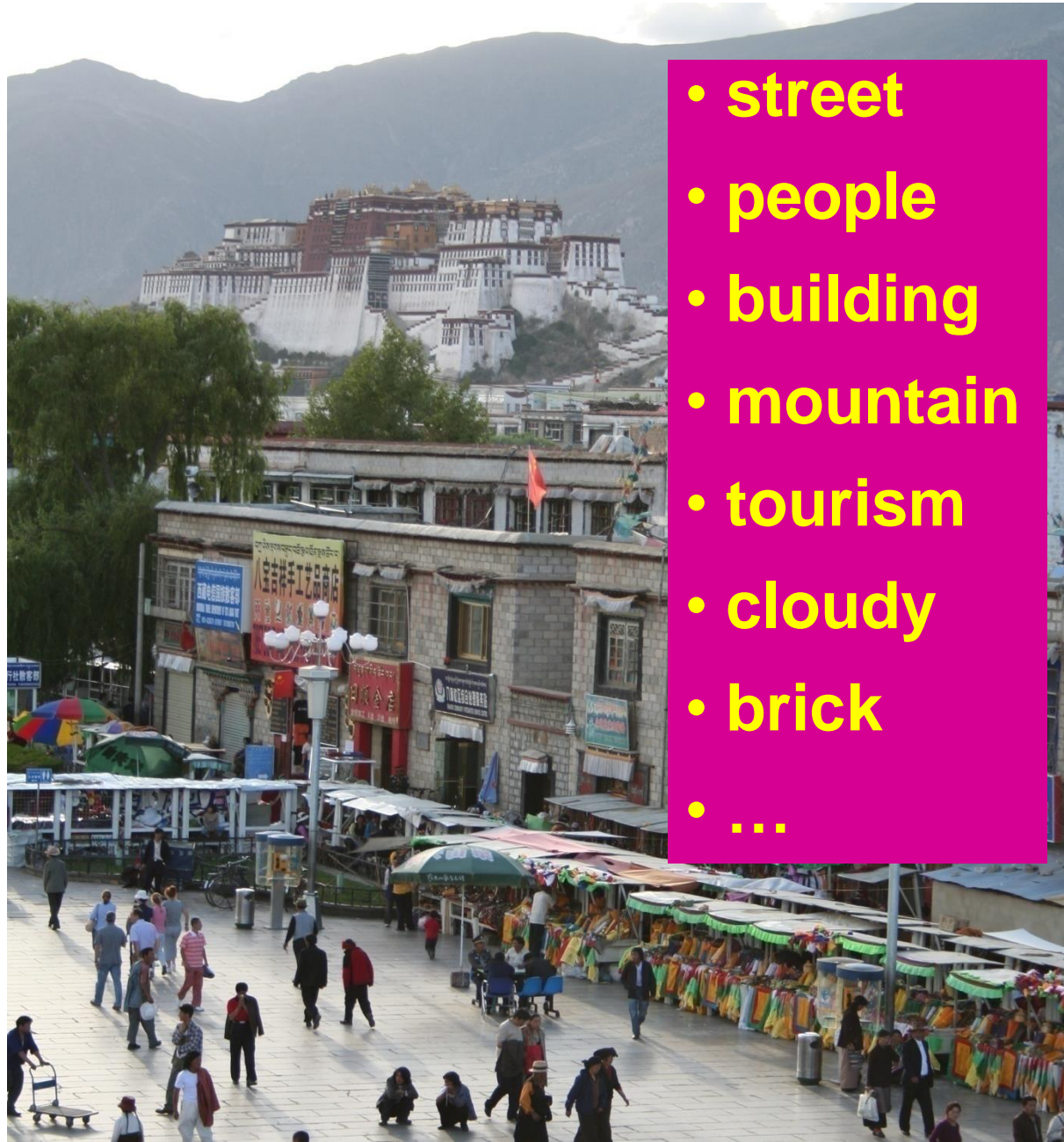
# Scene categorization or classification

- outdoor/indoor
- city/forest/factory/etc.





# Image annotation / tagging / attributes



- street
- people
- building
- mountain
- tourism
- cloudy
- brick
- ...

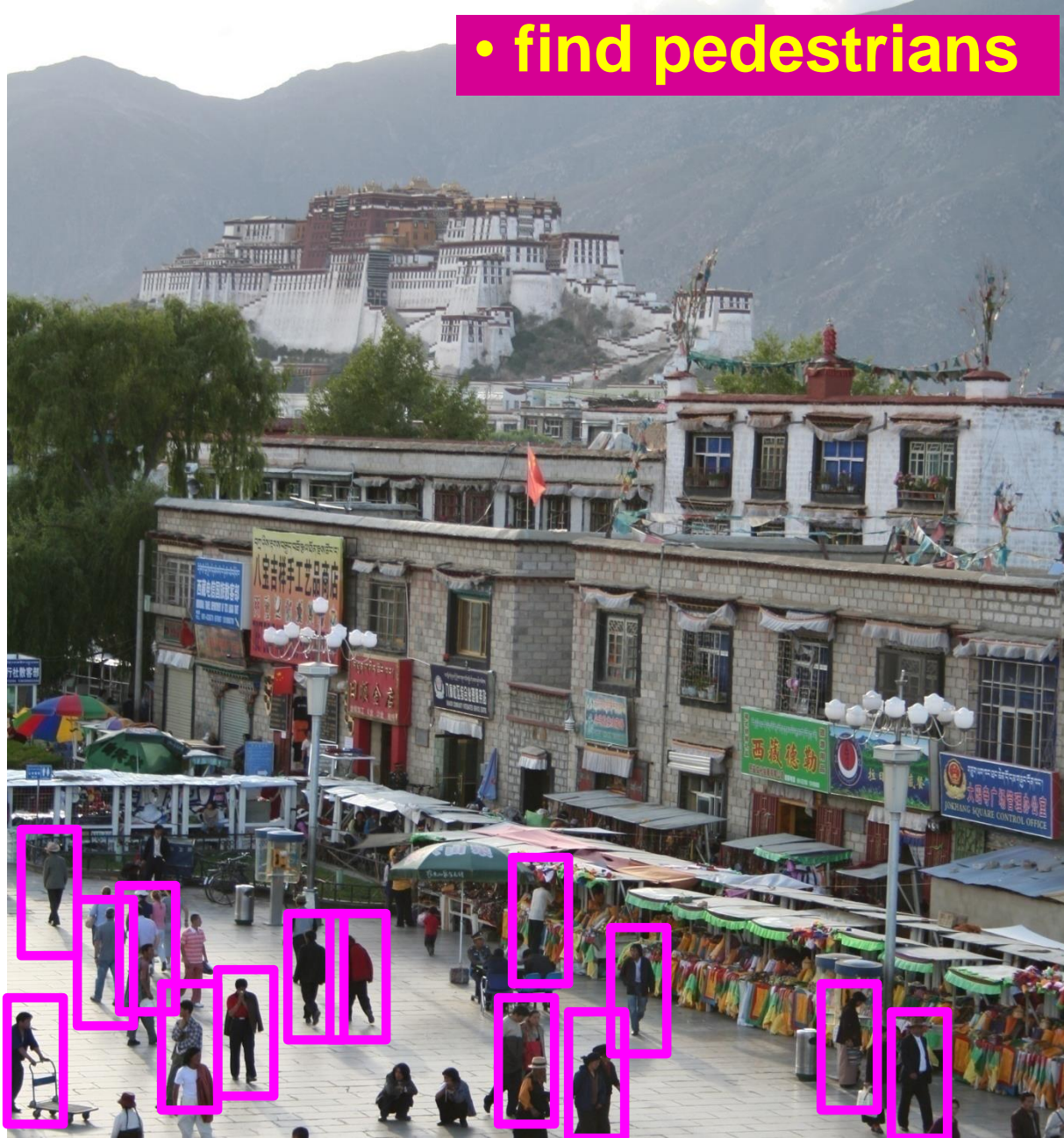
# Image parsing / semantic segmentation





# Object detection

- find pedestrians





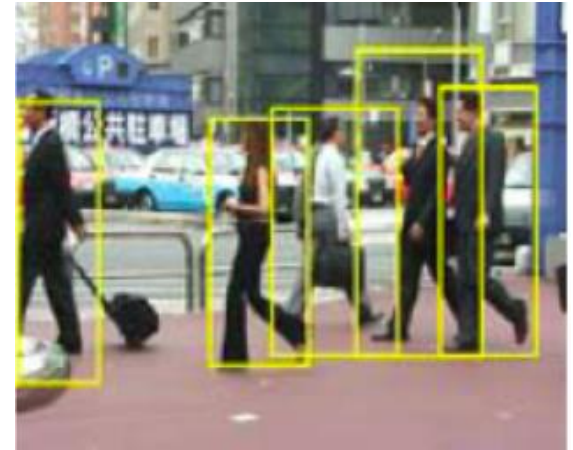
# Scene understanding?



# Category vs. instance recognition

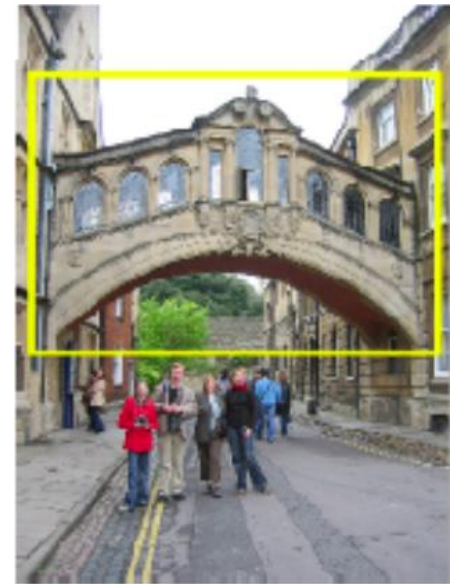
## Category:

- Find all the people
- Find all the buildings
- Often within a single image
- Often ‘sliding window’

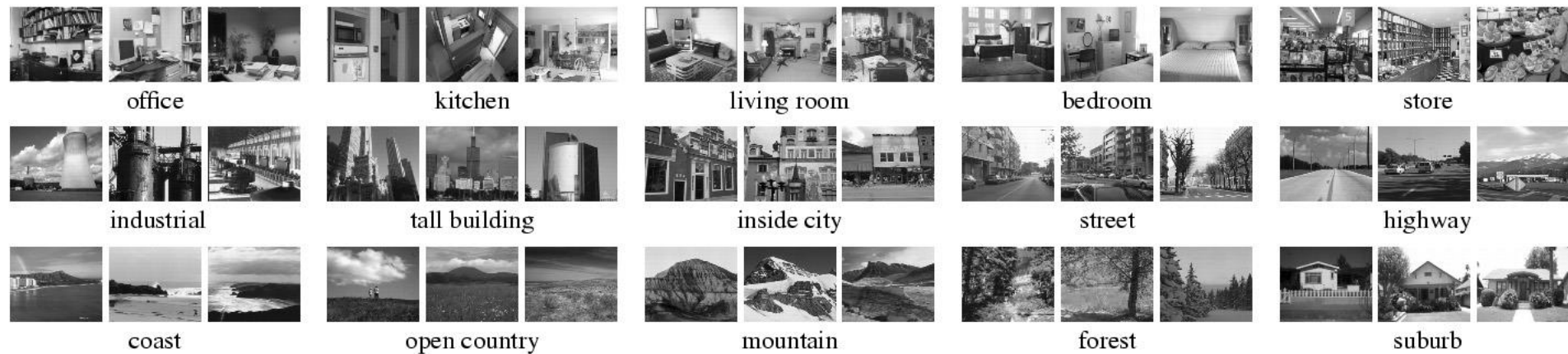


## Instance:

- Is this face James?
- Find this specific famous building
- Often within a database of images

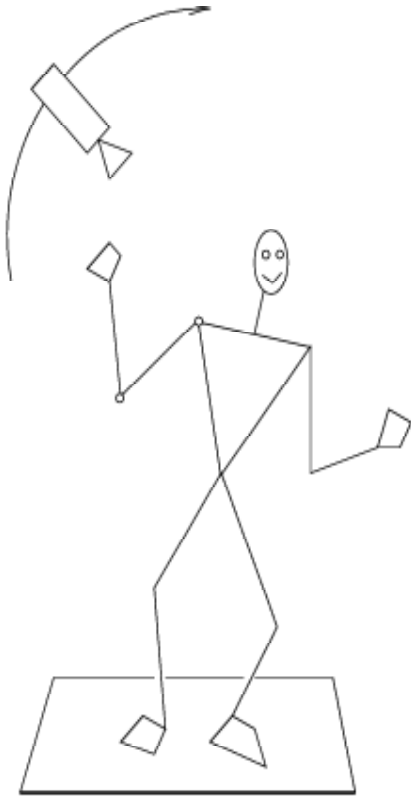


# Scene recognition dataset



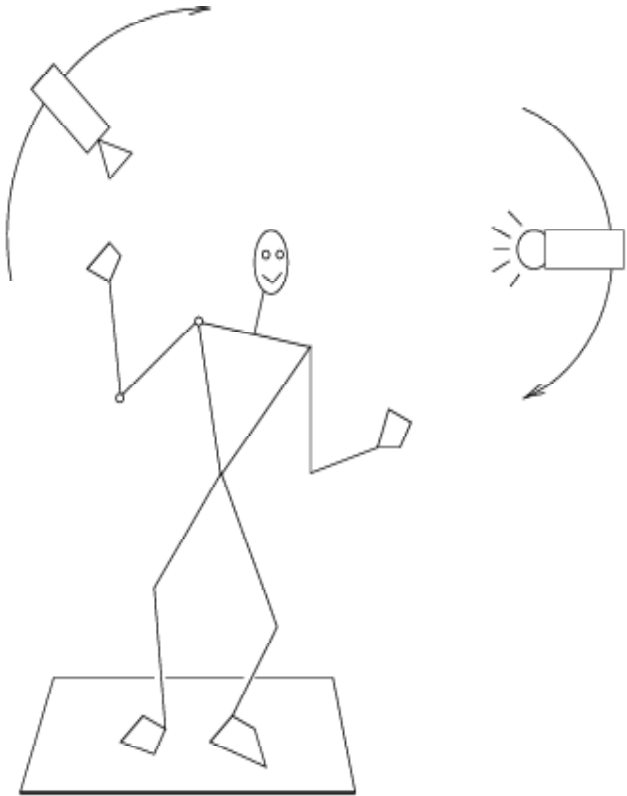
Instance or category?

# Recognition is all about modeling variability



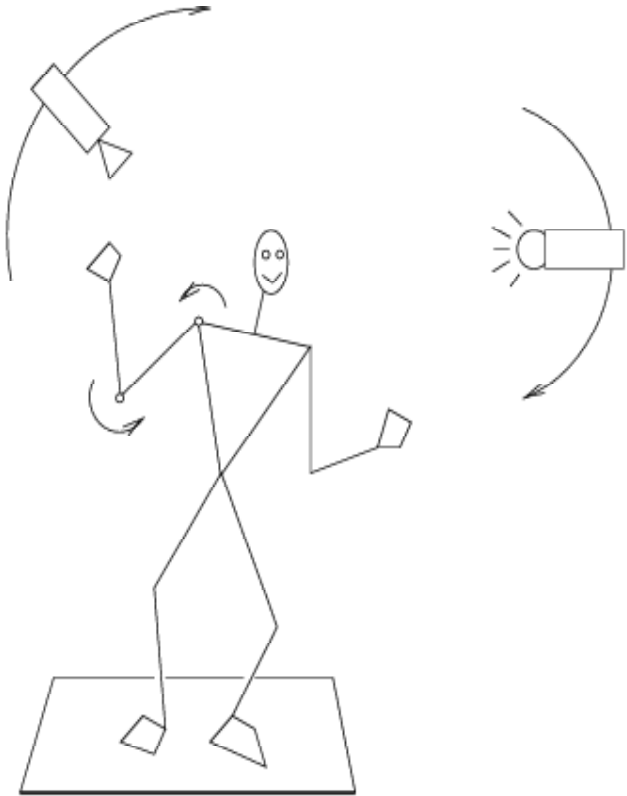
Variability: Camera position

# Recognition is all about modeling variability



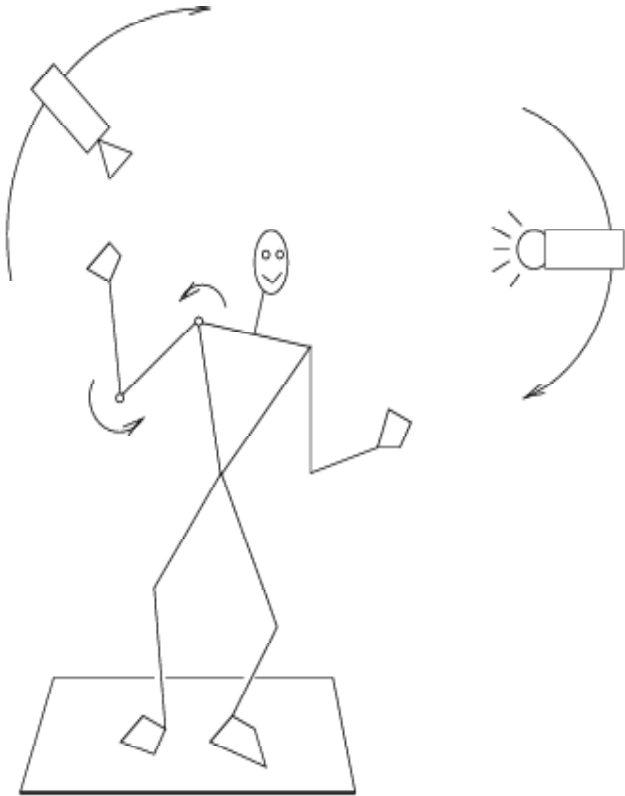
Variability: Camera position  
Illumination

# Recognition is all about modeling variability




Variability: Camera position  
Illumination  
Pose/shape parameters

# Recognition is all about modeling variability

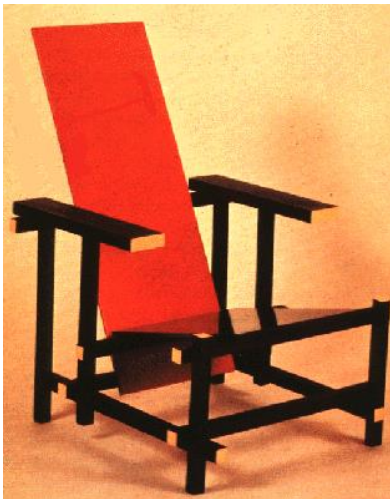


**Variability:** Camera position  
Illumination  
Pose/shape parameters

 Within-class variations?

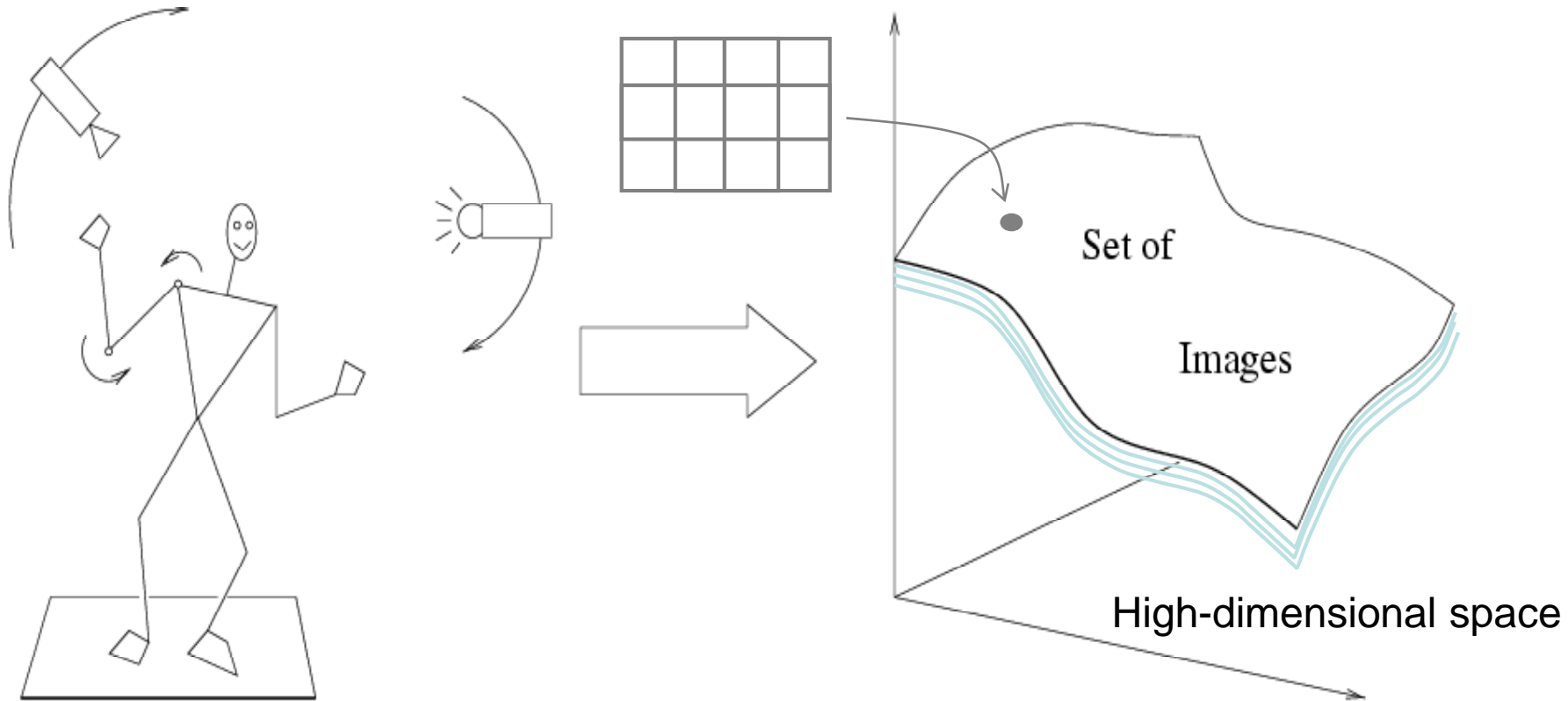


# Within-class variations





# Recognition is all about modeling variability



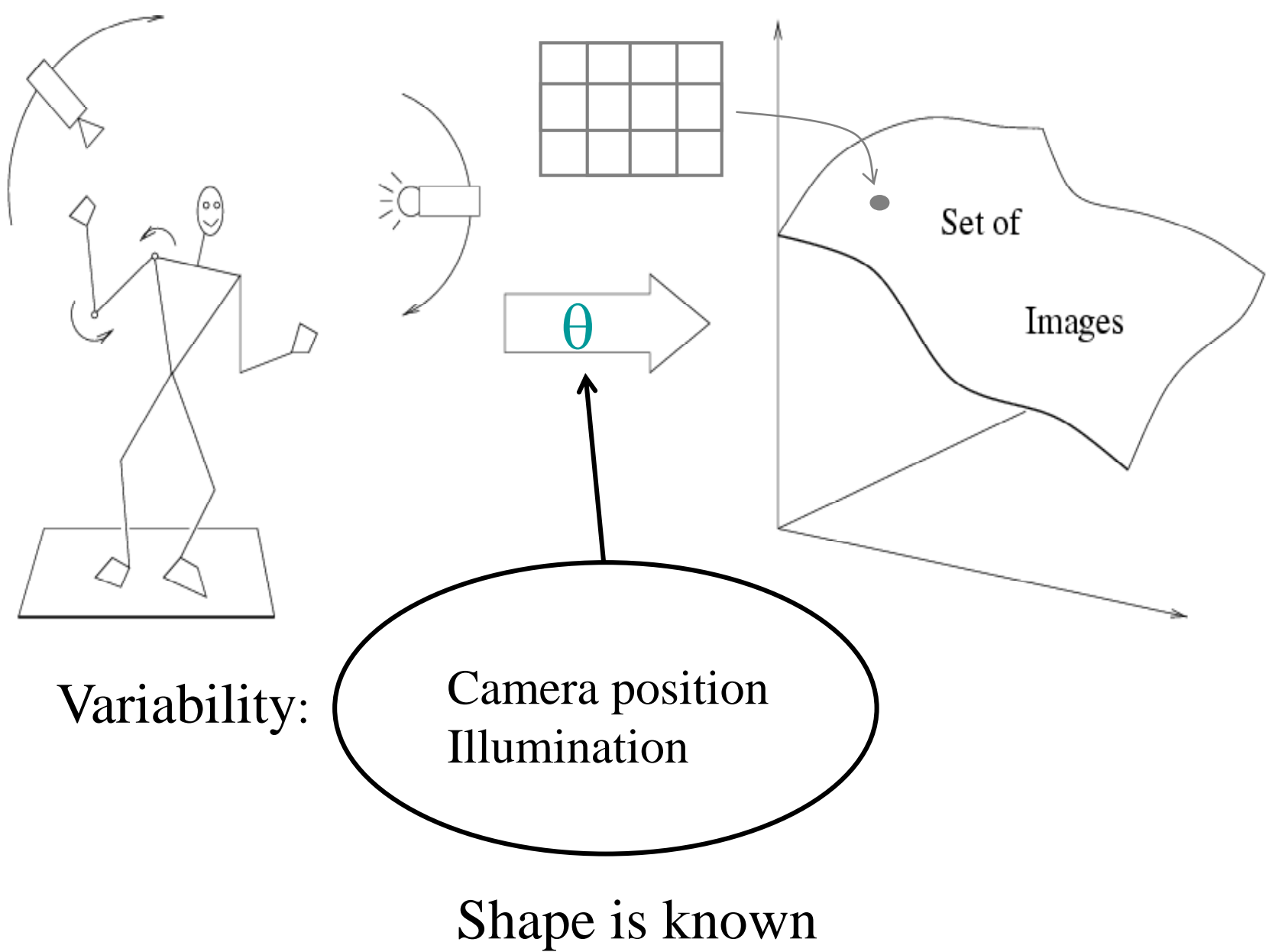
Variability:

- Camera position
- Illumination
- Pose/shape parameters
- Within-class variation

# History of ideas in recognition

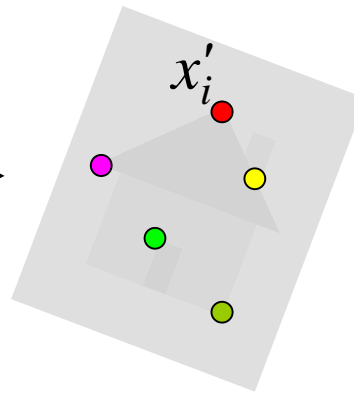
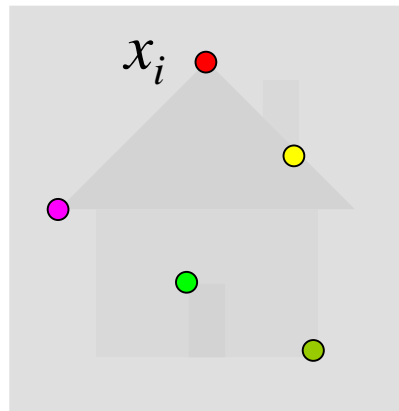
- 1960s – early 1990s: the geometric era

No digital cameras!  
Slow compute!



# Alignment

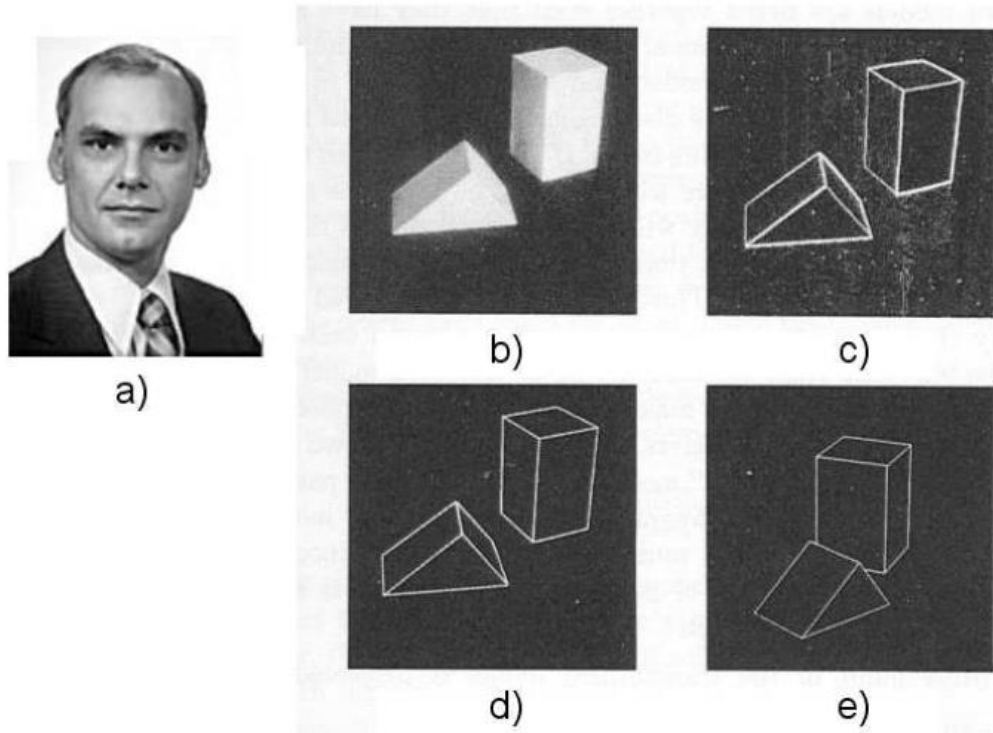
- Alignment: fitting a model to a transformation between pairs of features (*matches*) in two images



Find transformation  $T$   
that minimizes

$$\sum_i \text{residual}(T(x_i), x'_i)$$

# Recognition as an alignment problem: Block world



L. G. Roberts

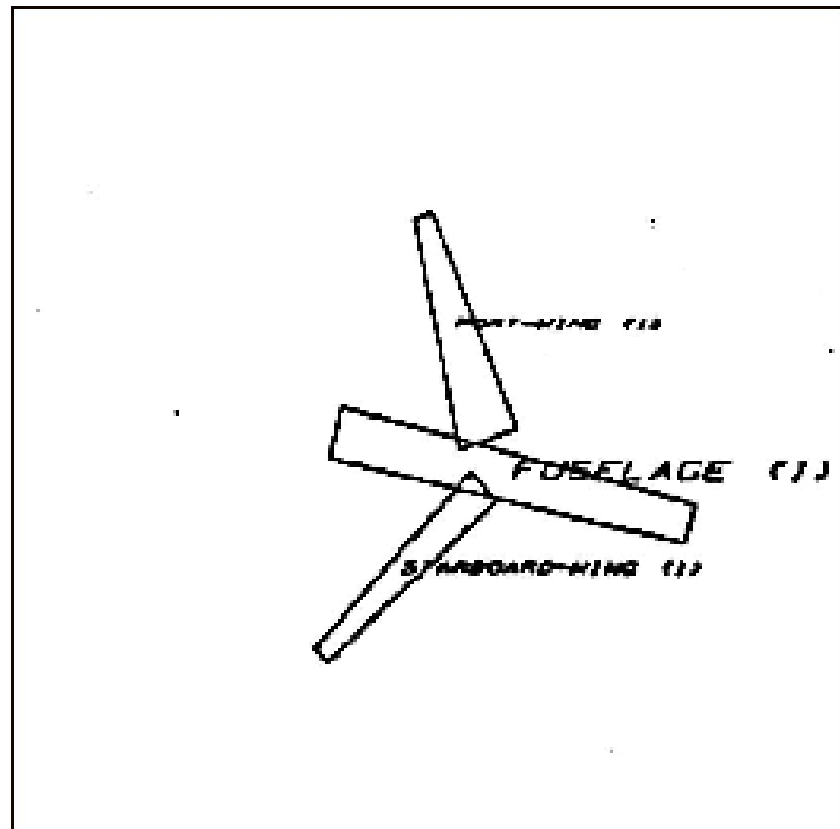
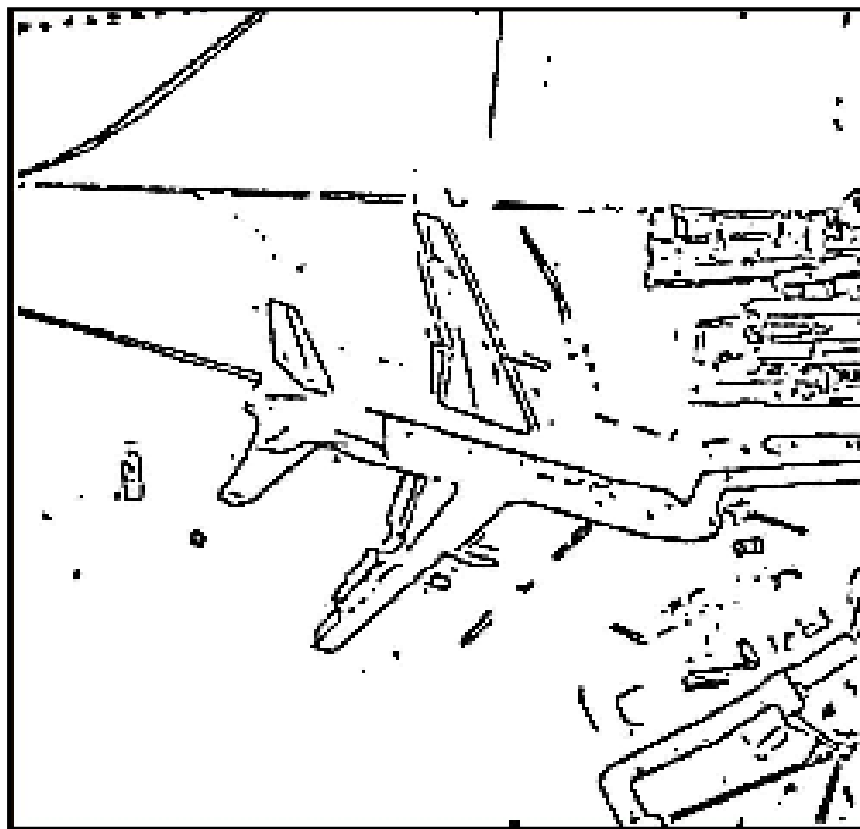
[Machine Perception of  
Three Dimensional Solids](#),

Ph.D. thesis, MIT

Department of Electrical  
Engineering, 1963.

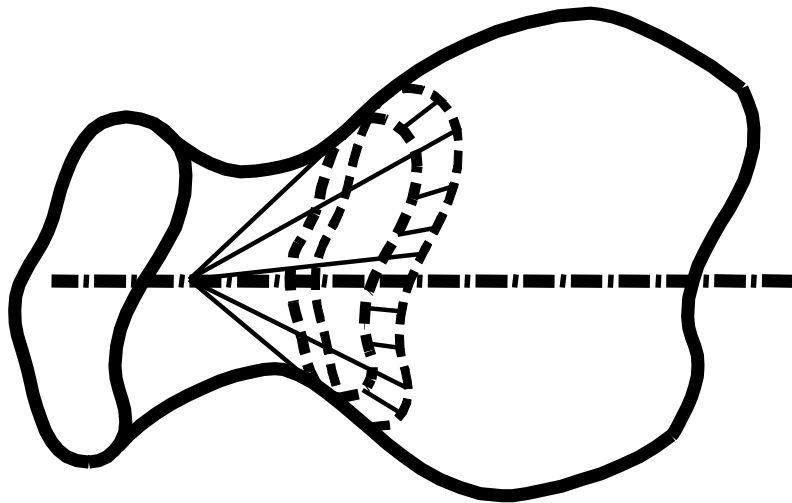
**Fig. 1.** A system for recognizing 3-d polyhedral scenes. a) L.G. Roberts. b) A blocks world scene. c) Detected edges using a 2x2 gradient operator. d) A 3-d polyhedral description of the scene, formed automatically from the single image. e) The 3-d scene displayed with a viewpoint different from the original image to demonstrate its accuracy and completeness. (b) - e) are taken from [64] with permission MIT Press.)

Representing and recognizing object categories is harder...



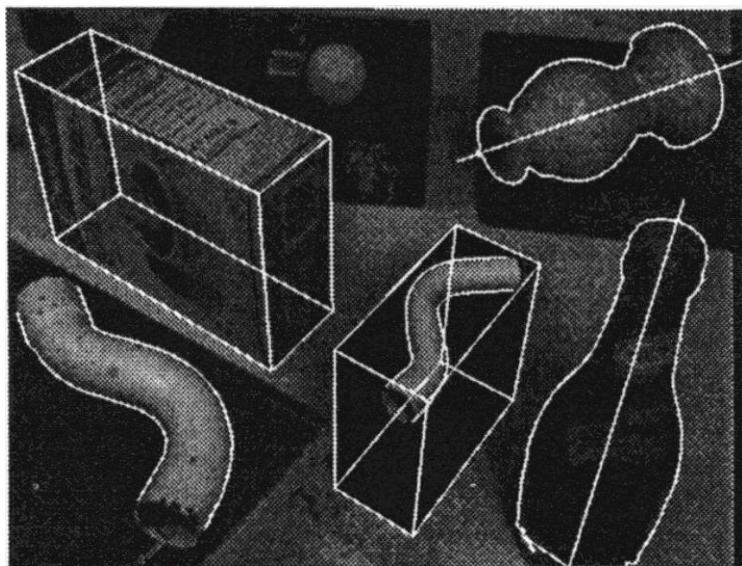
ACRONYM (Brooks and Binford, 1981)

Binford (1971), Nevatia & Binford (1972), Marr & Nishihara (1978)

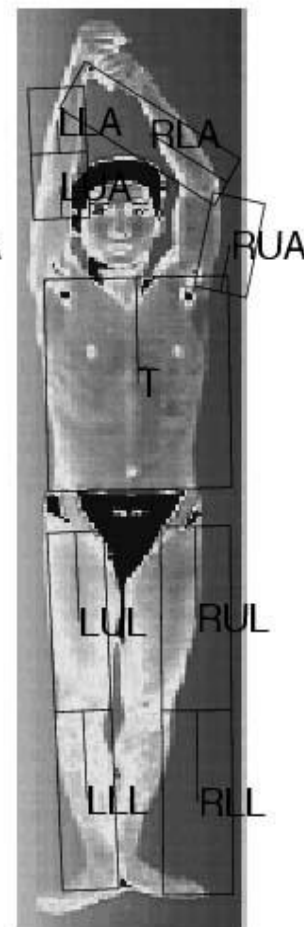
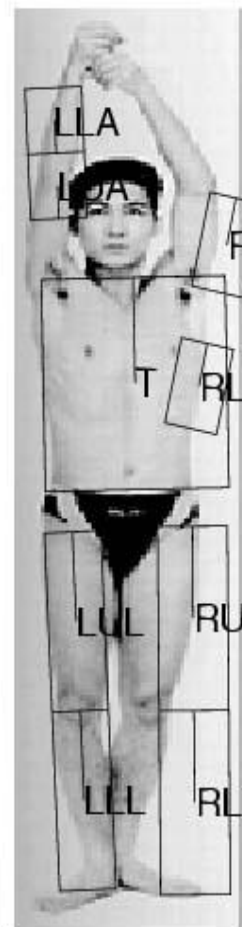
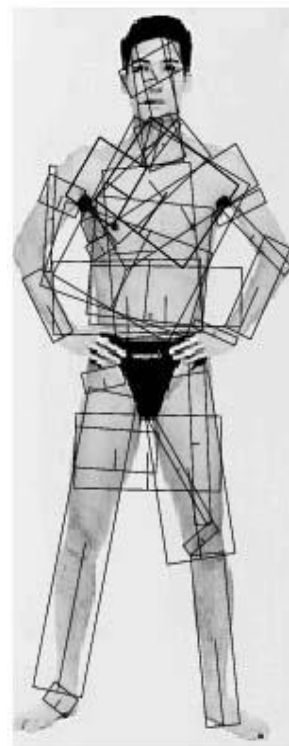


Generalized cylinders  
Ponce et al. (1989)

## General shape primitives?



Zisserman et al. (1995)



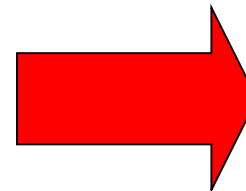
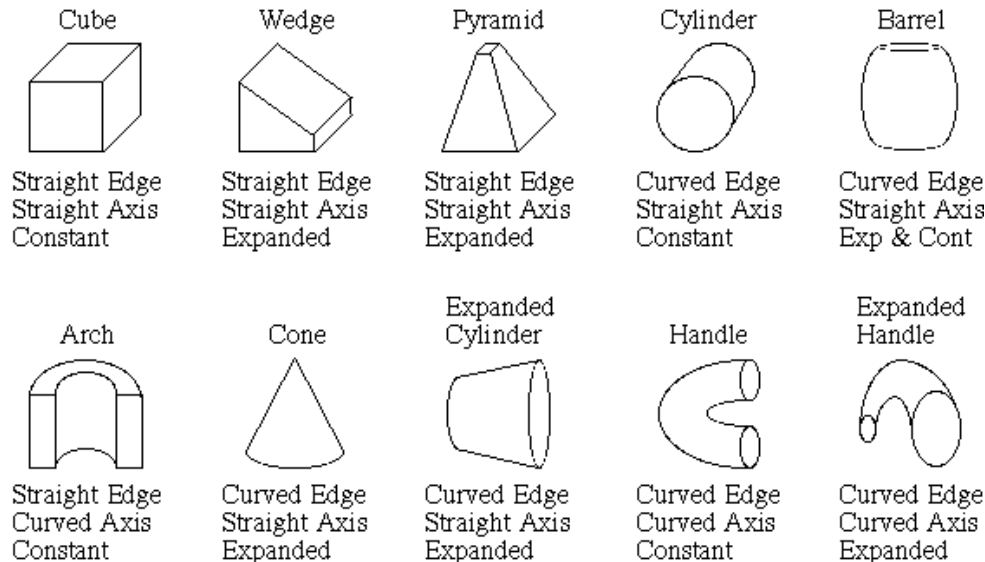
Forsyth (2000)



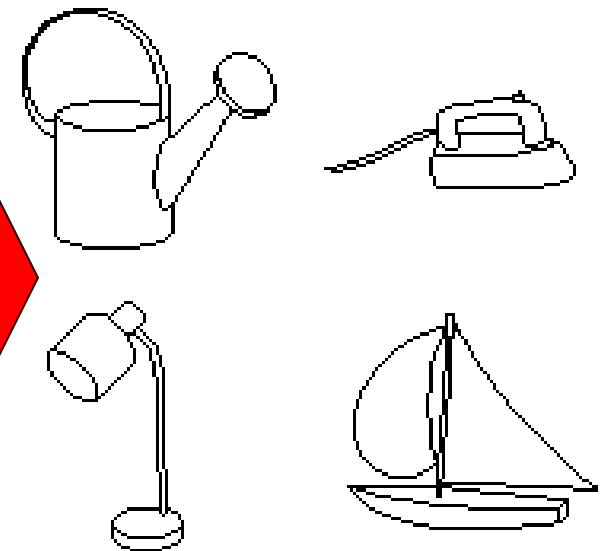
# Recognition by components

Biederman (1987)

## Primitives (geons)



## Objects



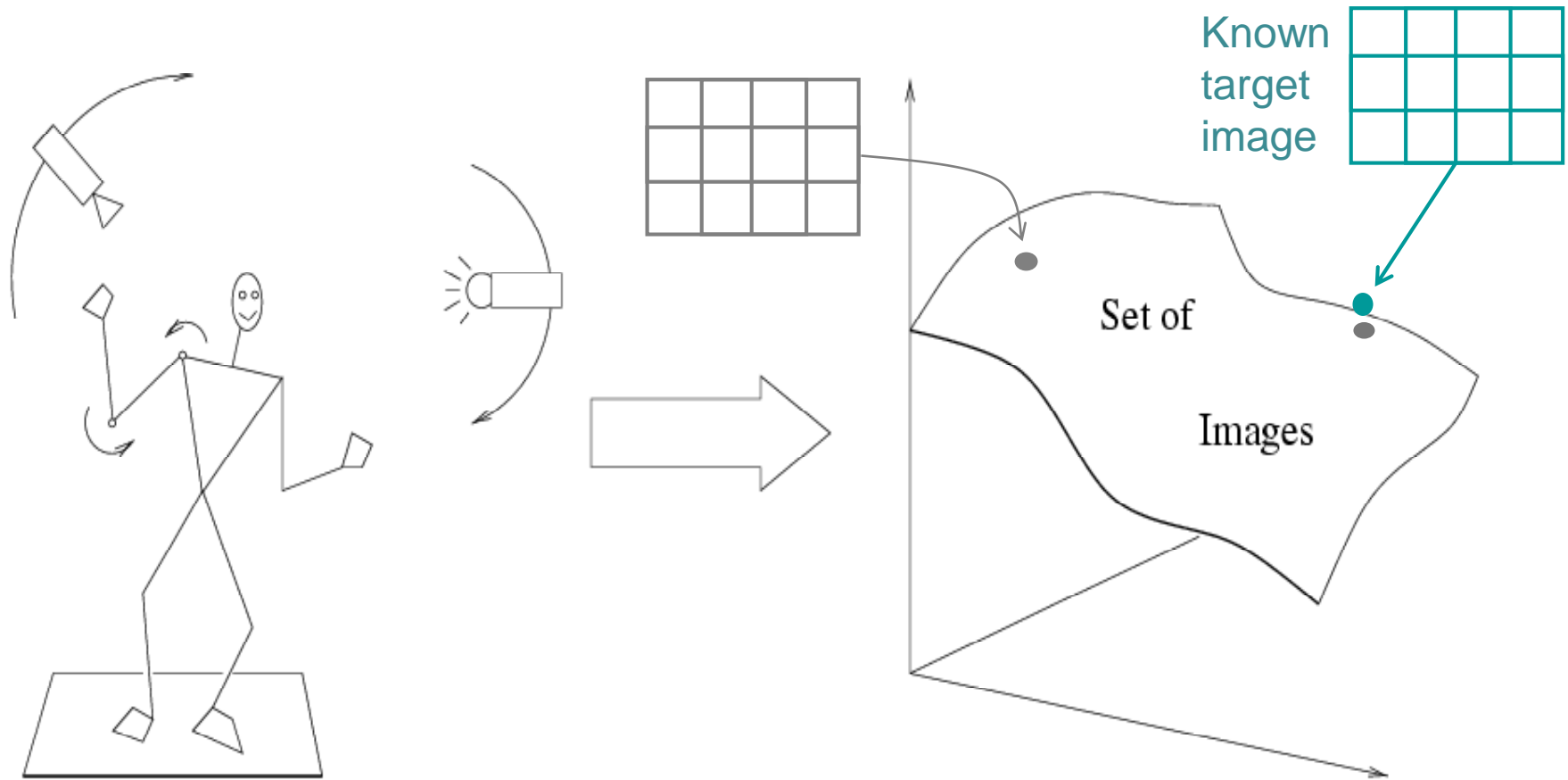
[http://en.wikipedia.org/wiki/Recognition\\_by\\_Components\\_Theory](http://en.wikipedia.org/wiki/Recognition_by_Components_Theory)

# History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models

No digital cameras!  
Slow compute!

Slow compute!

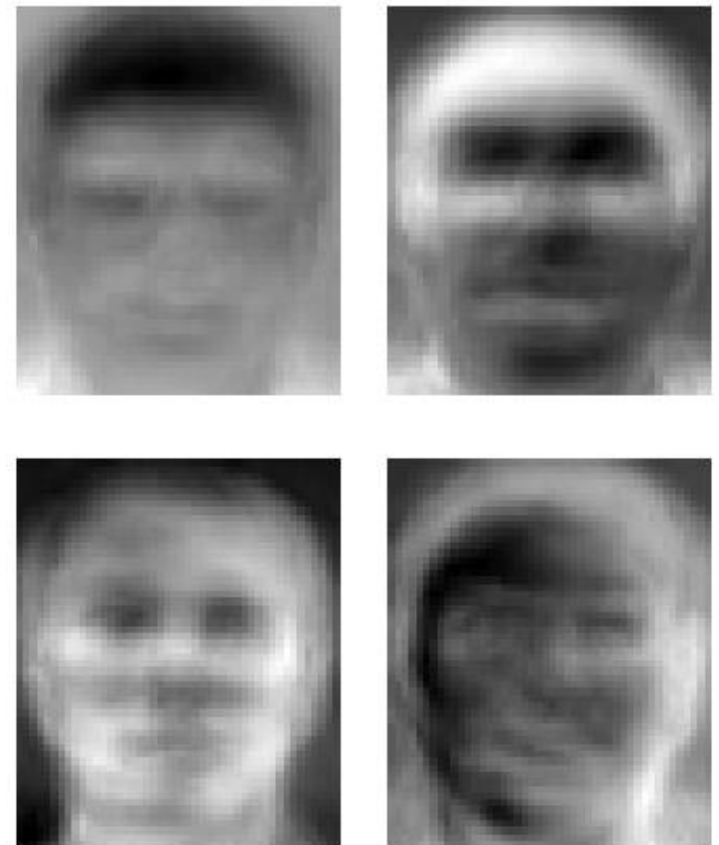


Empirical models of image variability

## Appearance-based techniques

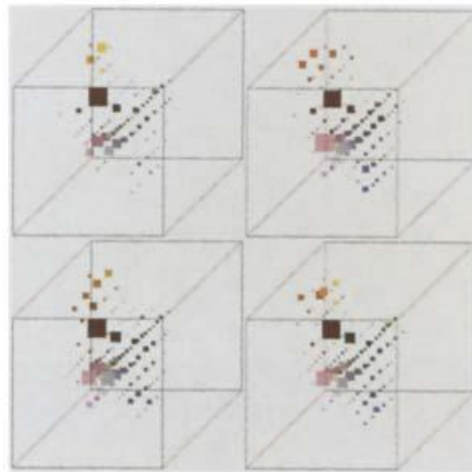
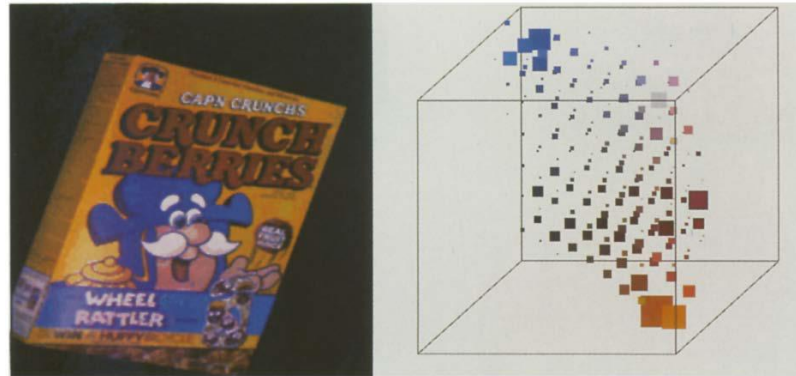
Turk & Pentland (1991); Murase & Nayar (1995); etc.

# Eigenfaces (Turk & Pentland, 1991)



Experimental Condition	Correct/Unknown Recognition Percentage		
	Lighting	Orientation	Scale
Forced classification	96/0	85/0	64/0
Forced 100% accuracy	100/19	100/39	100/60
Forced 20% unknown rate	100/20	94/20	74/20

# Color Histograms



Swain and Ballard, [Color Indexing](#), IJCV 1991.



# History of ideas in recognition

- 1960s – early 1990s: the geometric era No digital cameras!  
Slow compute!
- 1990s: appearance-based models Slow compute!
- 1990s – present: sliding window approaches

# Sliding window approaches



# Sliding window approaches



- Turk and Pentland, 1991
- Belhumeur, Hespanha, & Kriegman, 1997
- Schneiderman & Kanade 2004
- Viola and Jones, 2000



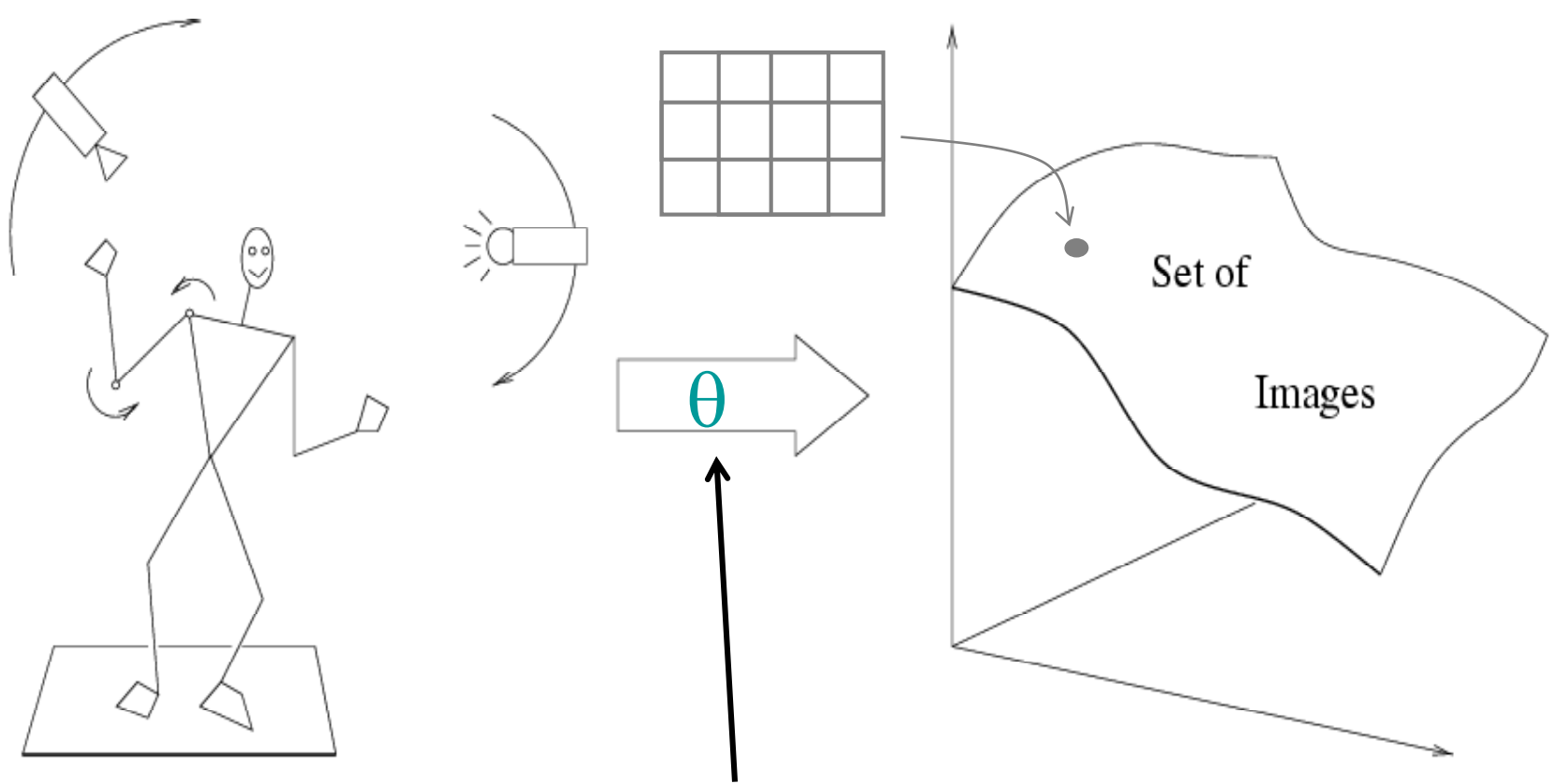
- Schneiderman & Kanade, 2004
- Argawal and Roth, 2002
- Poggio et al. 1993

# History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features

No digital cameras!  
Slow compute!

Slow compute!

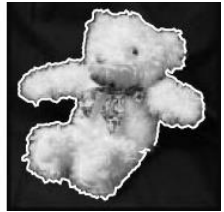


Variability:

Camera position  
Illumination  
Shape is partially known



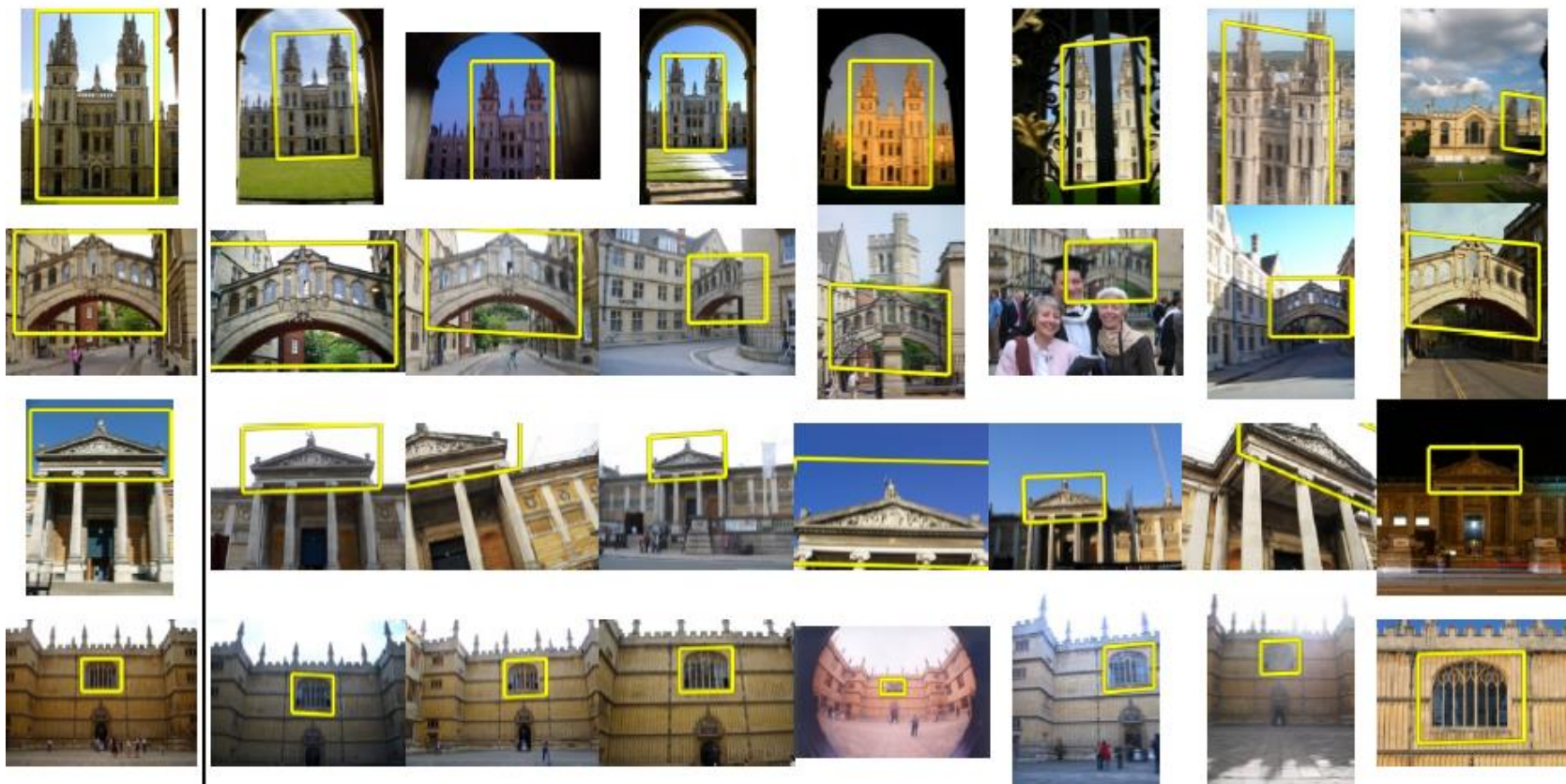
# Local features for object instance recognition



D. Lowe (1999, 2004)

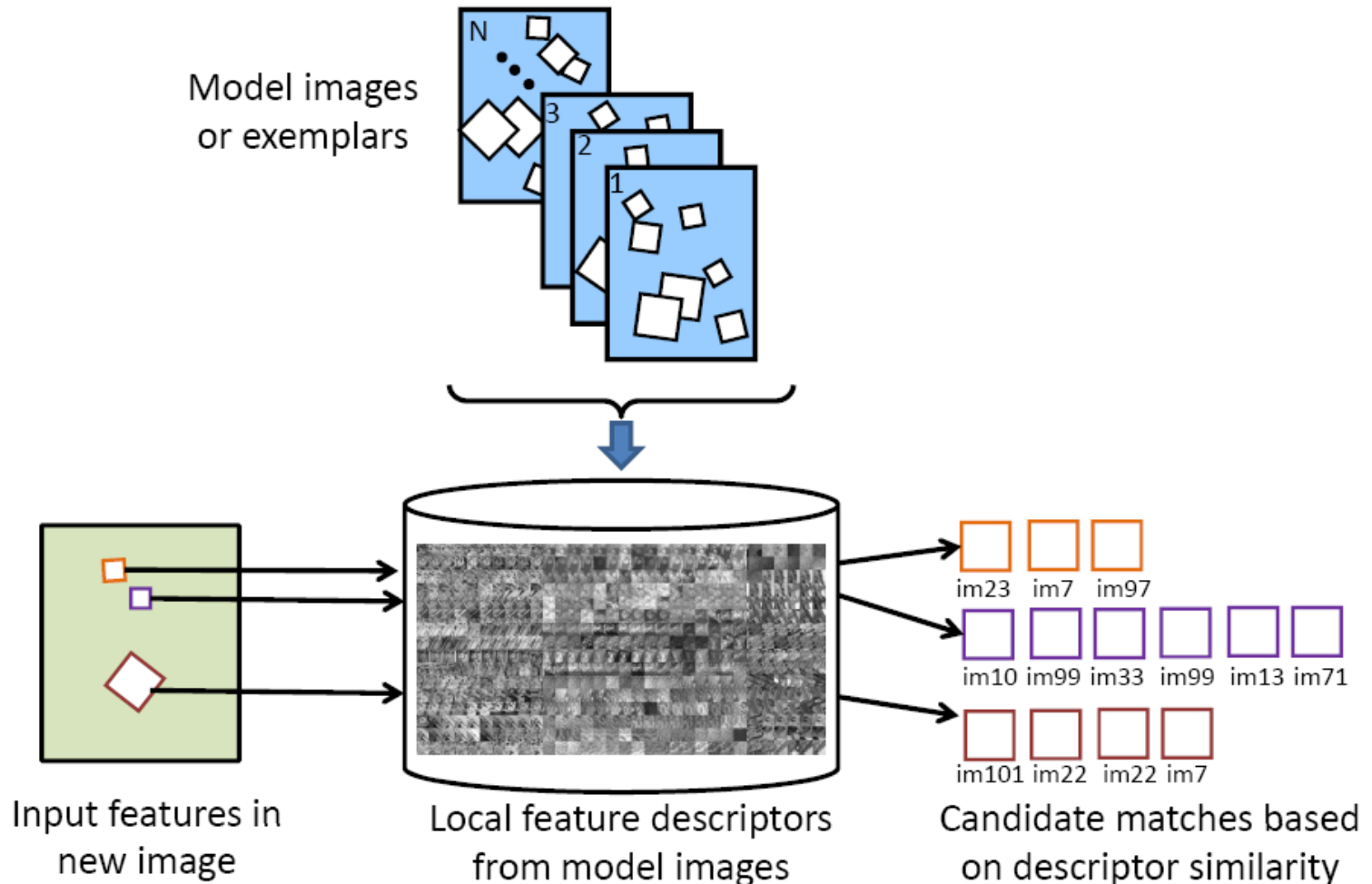
# Large-scale image search

Combining local features, indexing, and spatial constraints



# Large-scale image search

Combining local features, indexing, and spatial constraints



# Large-scale image search

Combining local features, indexing, and spatial constraints

## Google Goggles in Action

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



*Available on phones that run Android 1.6+ (i.e. Donut or Eclair)*

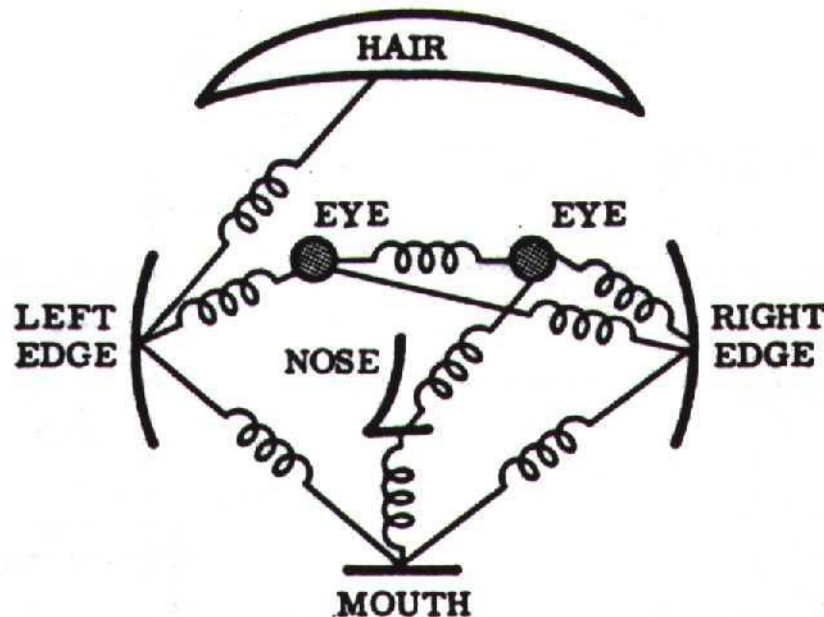
# History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models

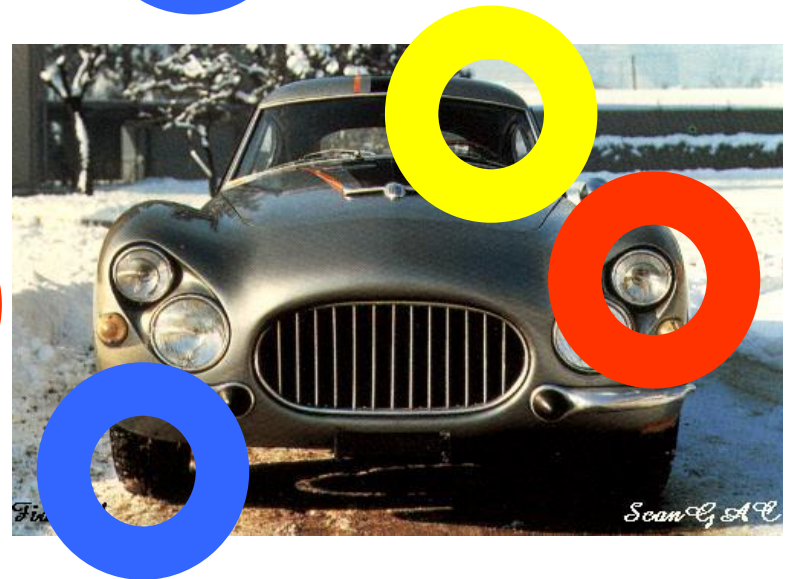


# Parts-and-shape models

- Model:
  - Object as a set of parts
  - Relative locations between parts
  - Appearance of part



# Constellation models



Weber, Welling & Perona (2000), Fergus, Perona & Zisserman (2003)

# History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features (next!)

No digital cameras!  
Slow compute!

Slow compute!

Early GPU compute.

# History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features (next!)
- *Present trends:*  
Combined local and global methods,  
context, deep learning

No digital cameras!  
Slow compute!

Slow compute!

Early GPU compute.

GPU/cloud compute.

# Recognition Issues

How to summarize the content of an entire image?

How to gauge overall similarity?

How large should the vocabulary be?

How to perform quantization efficiently?

How to score the retrieval results?

How might we add more spatial verification?



# Recognition Issues

**How to summarize the content of an entire image?**

How to gauge overall similarity?

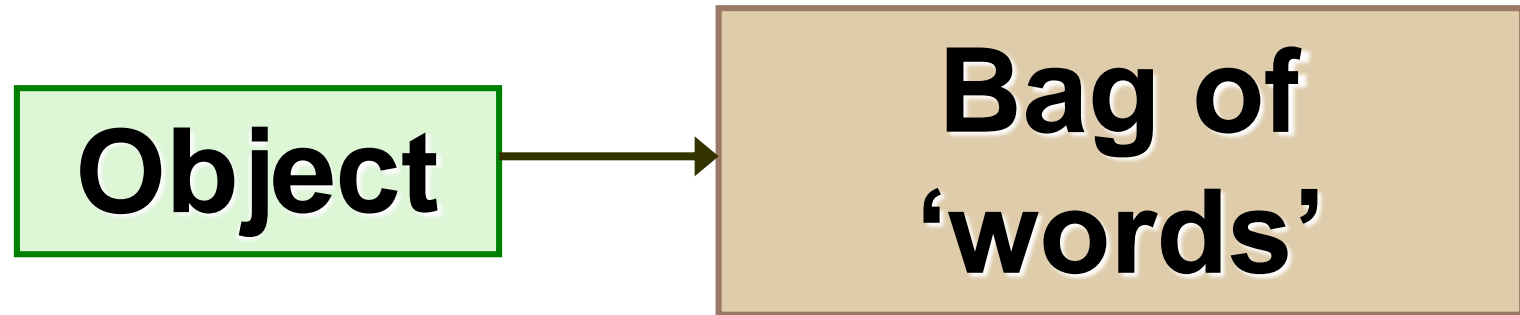
How large should the vocabulary be?

How to perform quantization efficiently?

How to score the retrieval results?

How might we add more spatial verification?

# Bag-of-features models



# Origin 1: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

# Origin 1: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

2007-01-23: State of the Union Address

George W. Bush (2001-)

abandon accountable affordable afghanistan africa aided ally anbar armed army baghdad bless challenges chamber chaos  
choices civilians coalition commanders commitment confident confront congressman constitution corps debates deduction  
deficit deliver democratic deploy dikembe diplomacy disruptions earmarks economy einstein elections eliminates  
expand extremists failing faithful families freedom fuel funding god haven ideology immigration impose  
insurgents iran **iraq** islam julie lebanon love madam marine math medicare moderation neighborhoods nuclear offensive  
palestinian payroll province pursuing **qaeda** radical regimes resolve retreat rieman sacrifices science sectarian senate  
september shia stays strength students succeed sunni tax territories **terrorists** threats uphold victory  
violence violent **war** washington weapons wesley

US Presidential Speeches Tag Cloud

<http://chir.ag/phernalia/preztags/>

# Origin 1: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



# Origin 1: Bag-of-words models

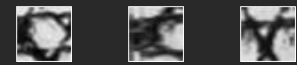
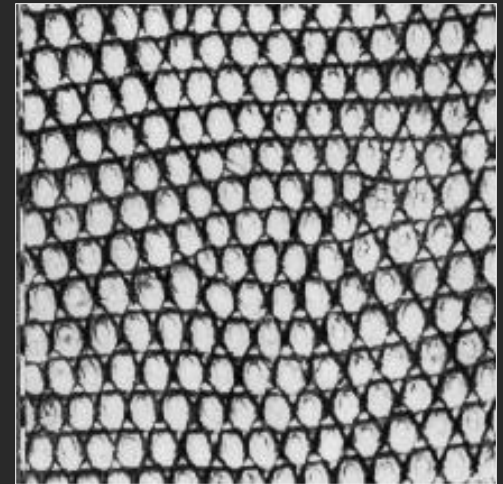
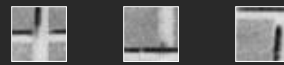
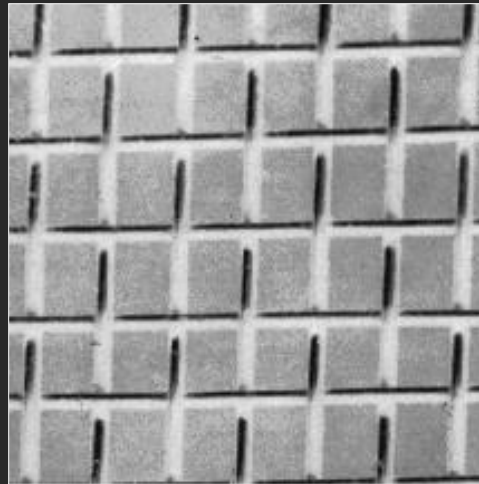
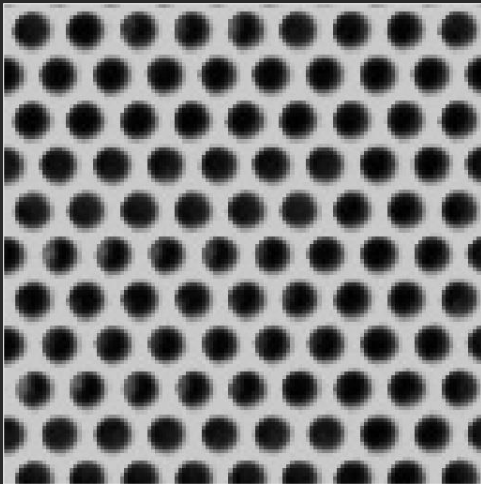
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)





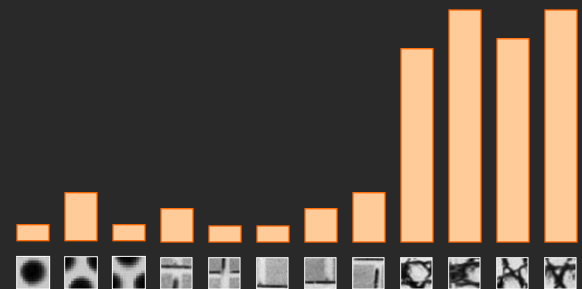
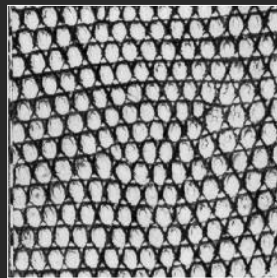
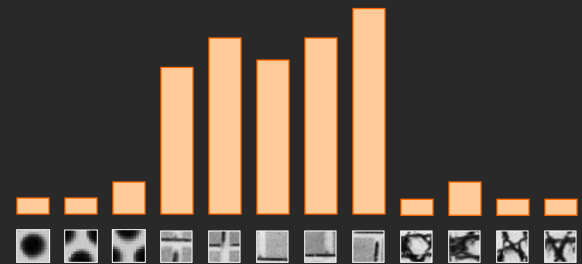
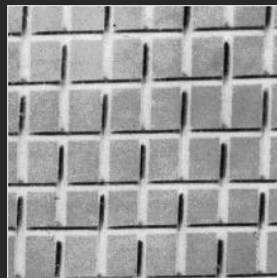
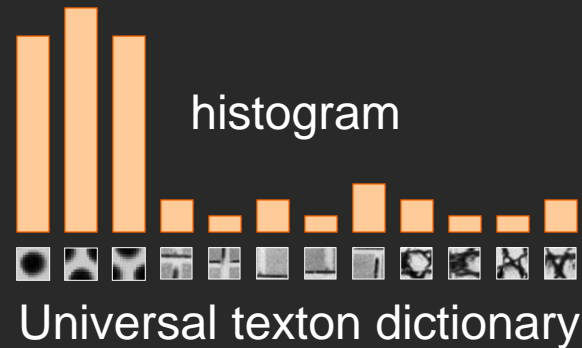
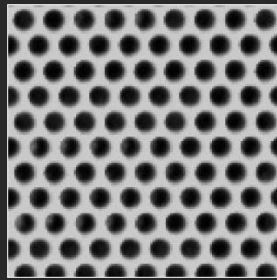
# Origin 2: Texture recognition

- Characterized by repetition of basic elements or *textons*
- For stochastic textures, the identity of textons matters, not their spatial arrangement



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

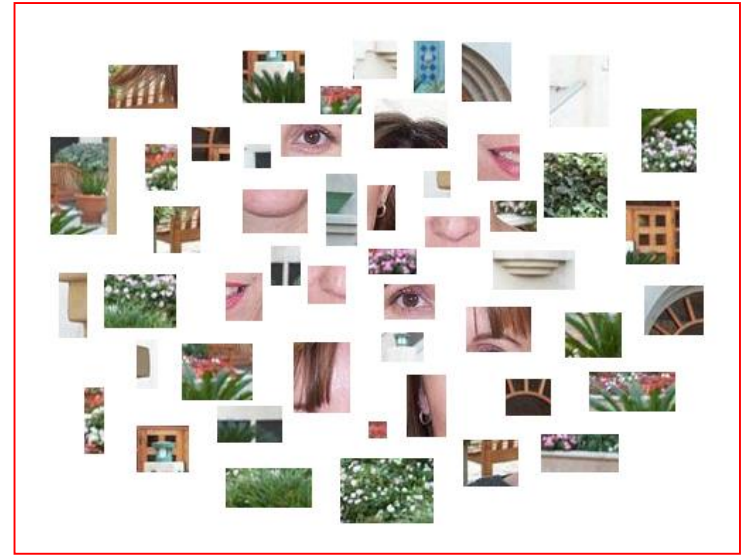
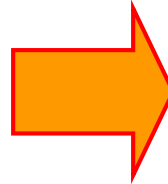
# Origin 2: Texture recognition



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

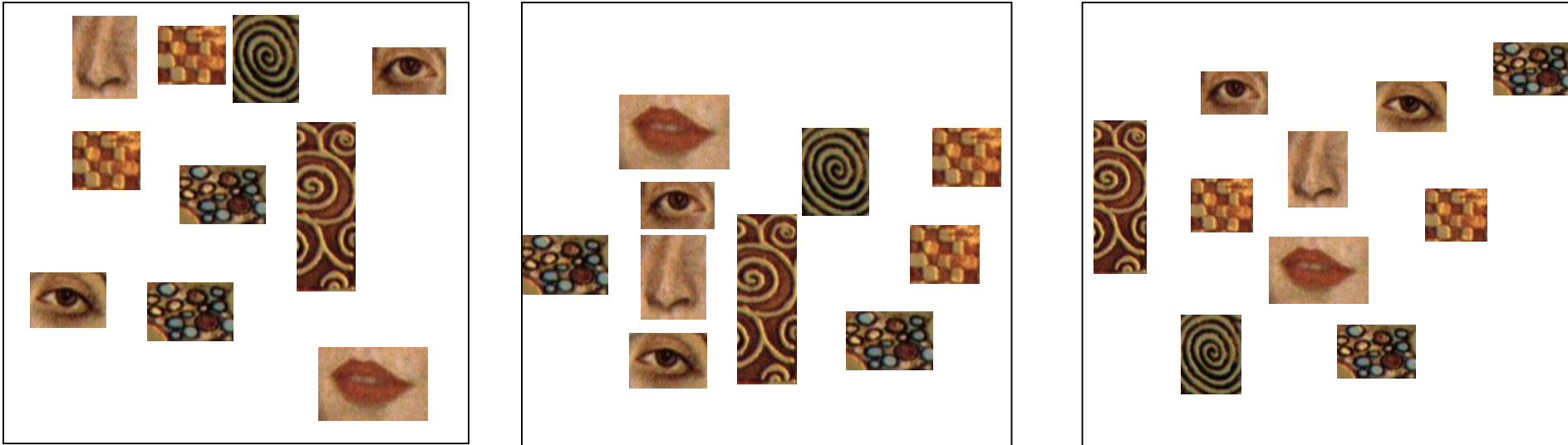
# Bag-of-features models

---



# Objects as texture

- All of these are treated as being the same

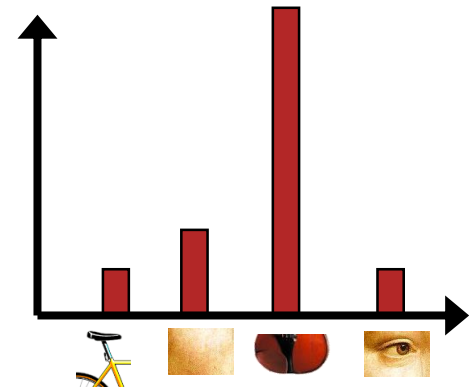
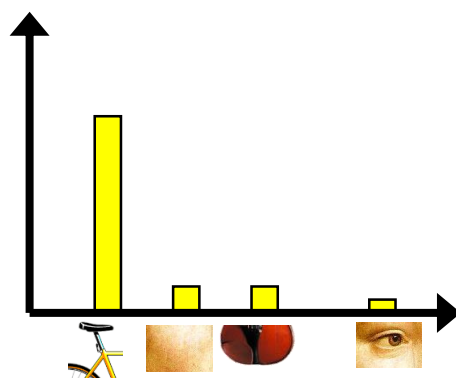
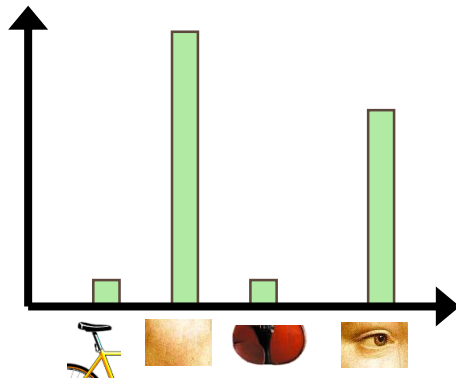
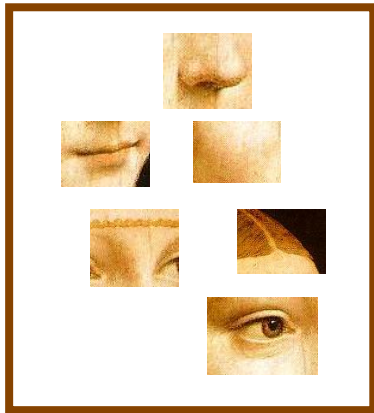


- No distinction between foreground and background: scene recognition?

# Bag-of-features steps

---

1. Feature extraction
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”





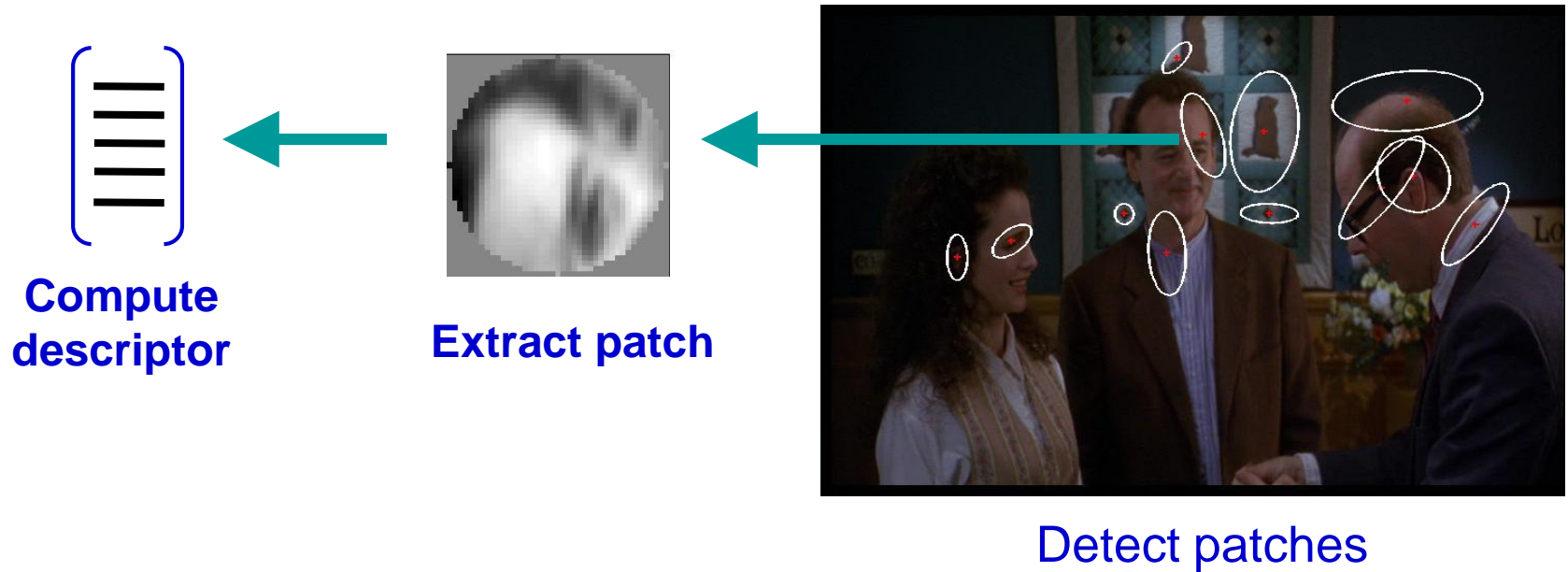
# 1. Feature extraction

- Regular grid or interest regions

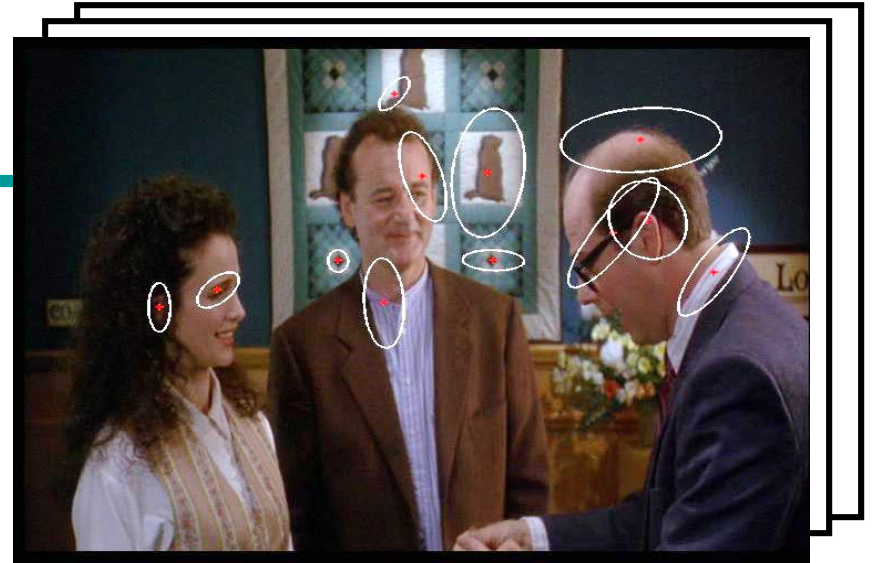
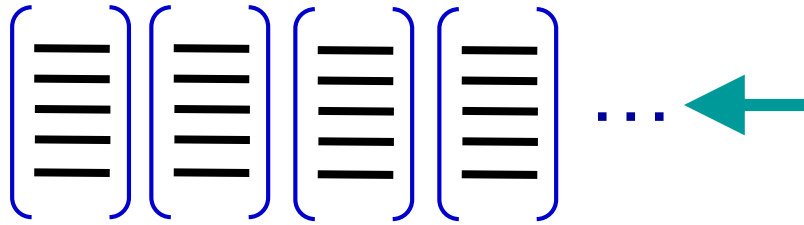




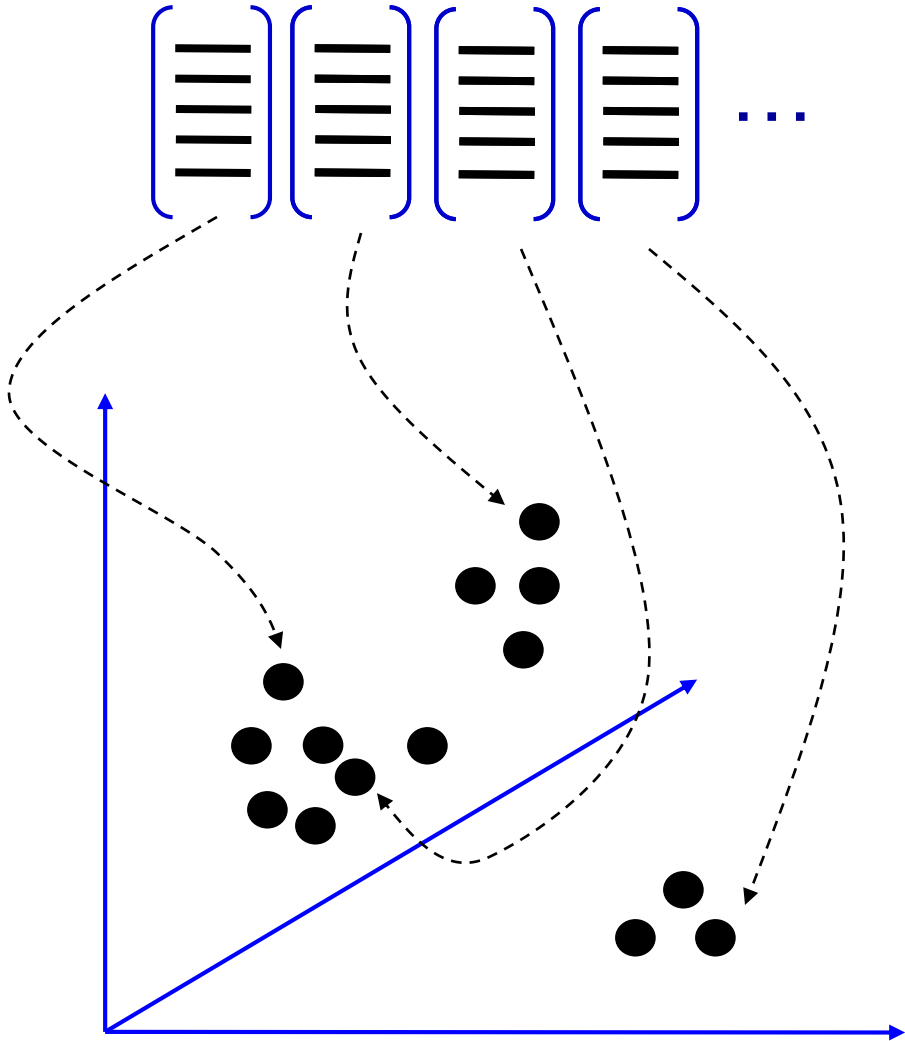
# 1. Feature extraction



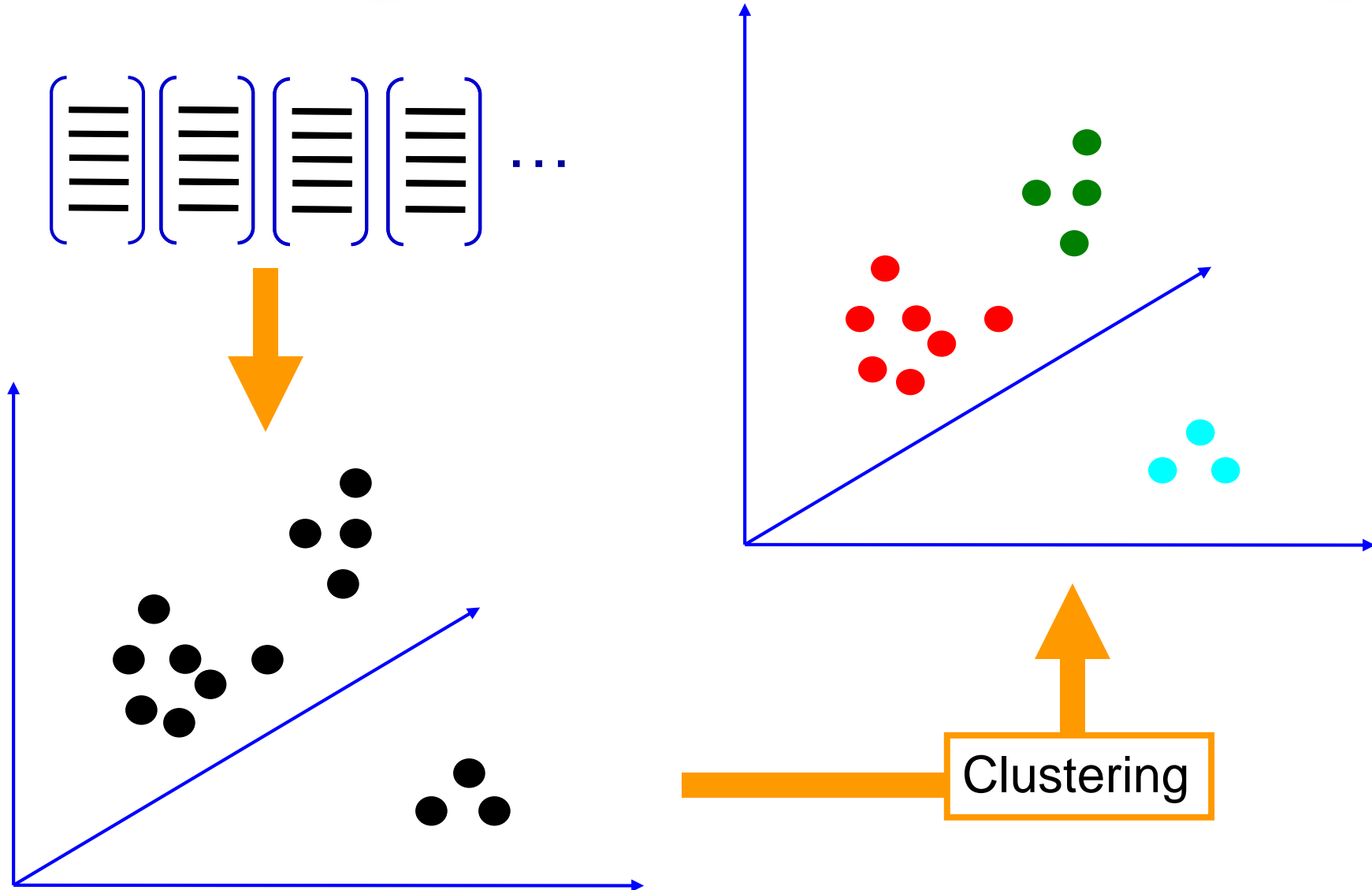
# 1. Feature extraction



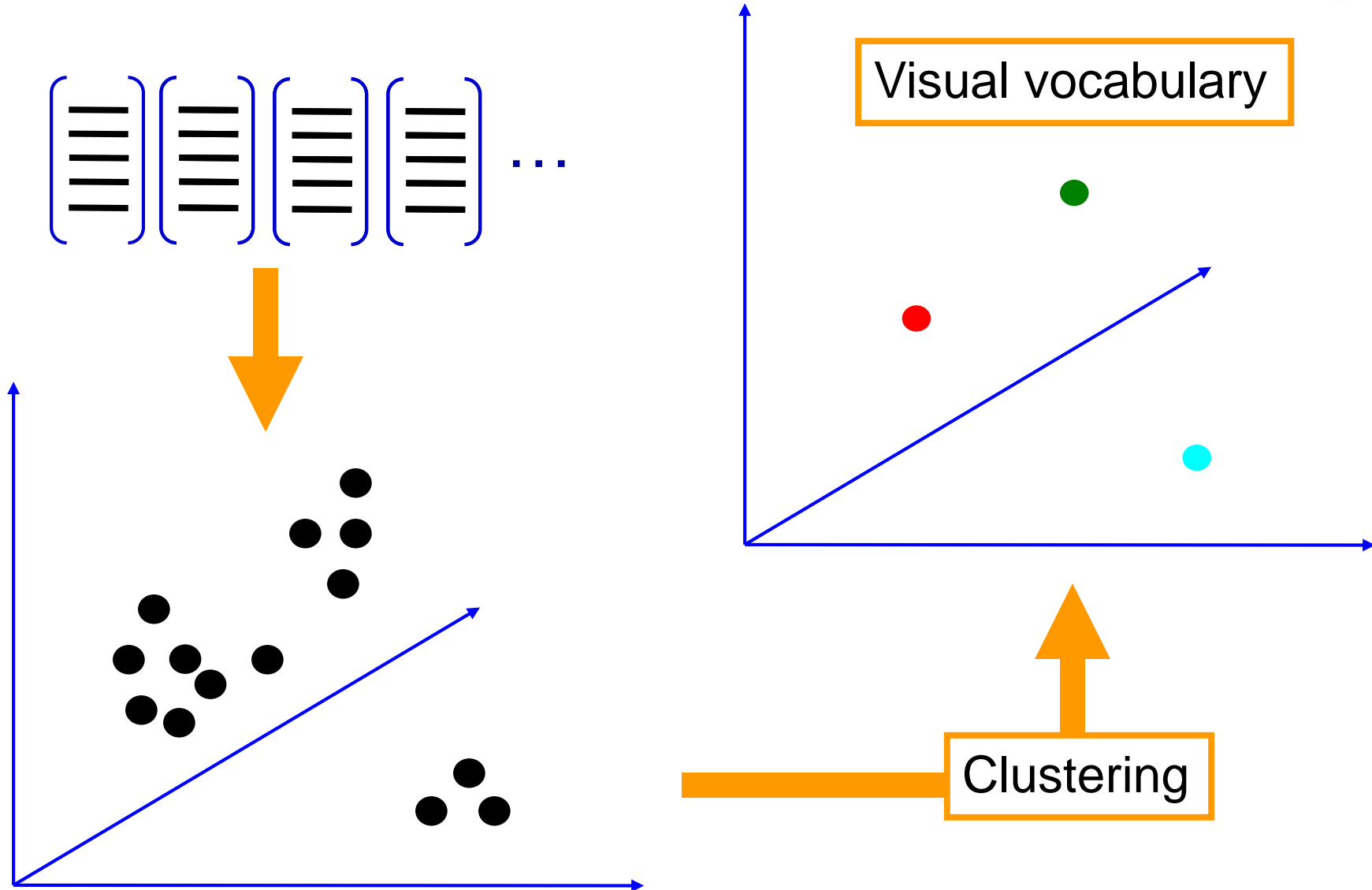
## 2. Learning the visual vocabulary



## 2. Learning the visual vocabulary



# 3. Quantize the visual vocabulary

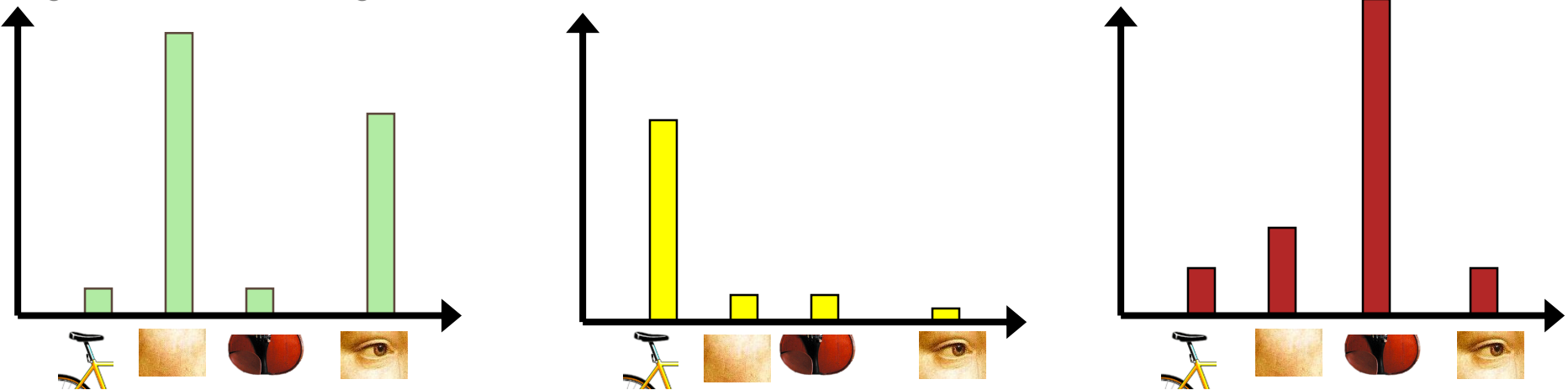




Visual words

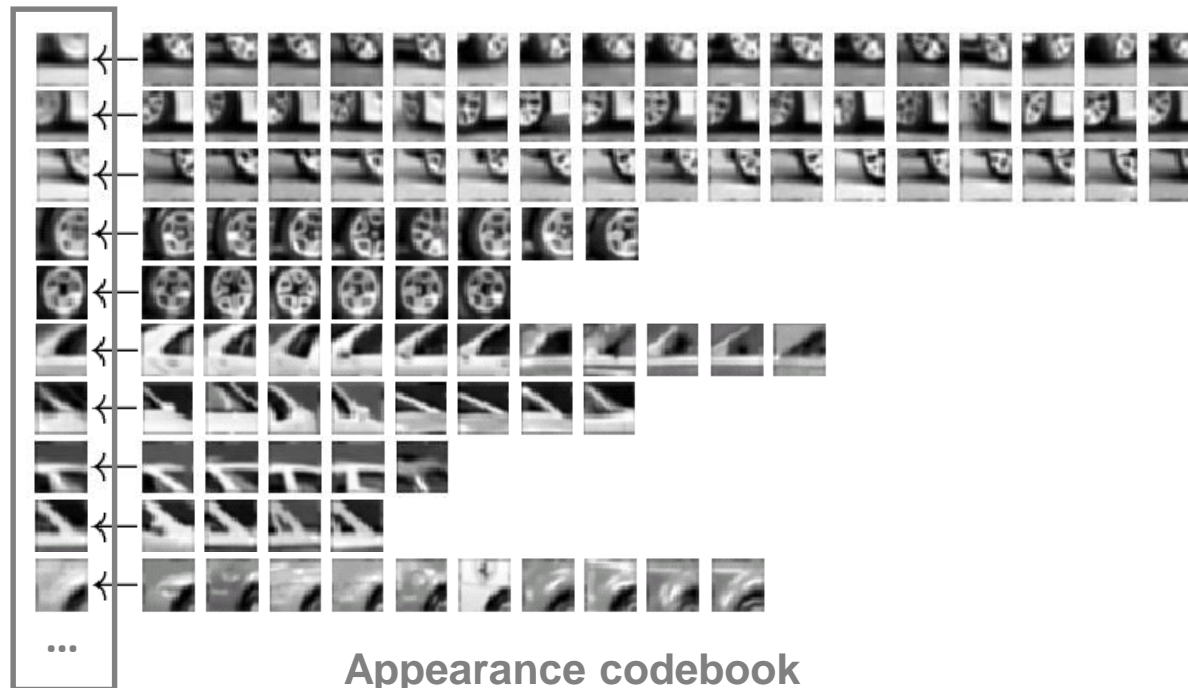
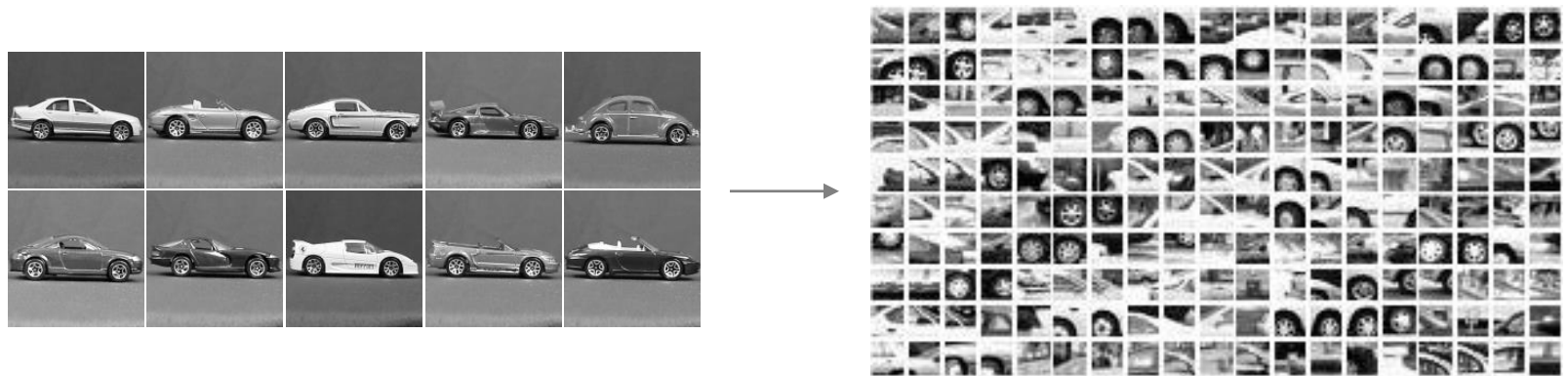


Bag of visual words histograms



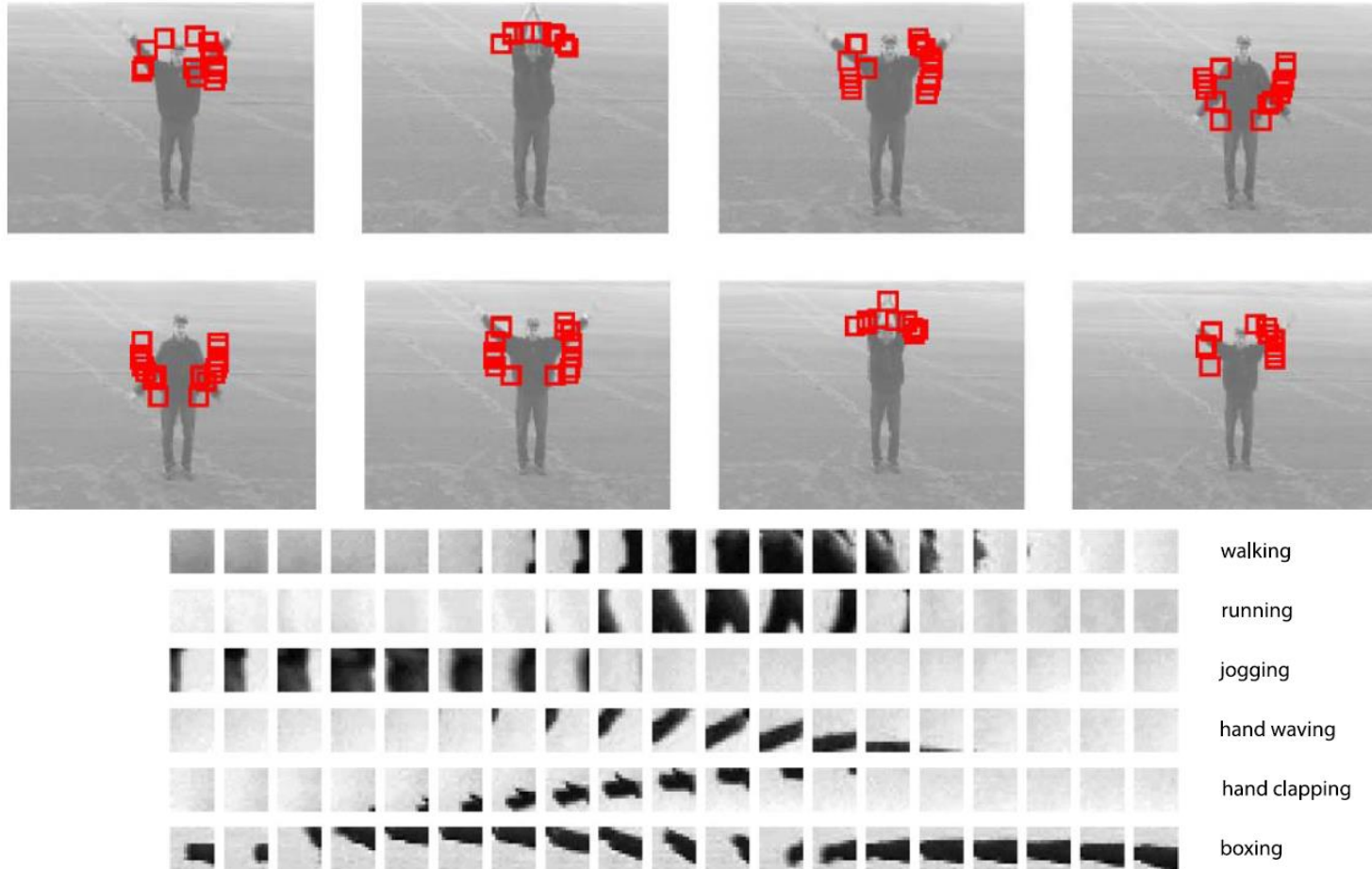


# Example real codebook



# Bags of features for action recognition

## Space-time interest points

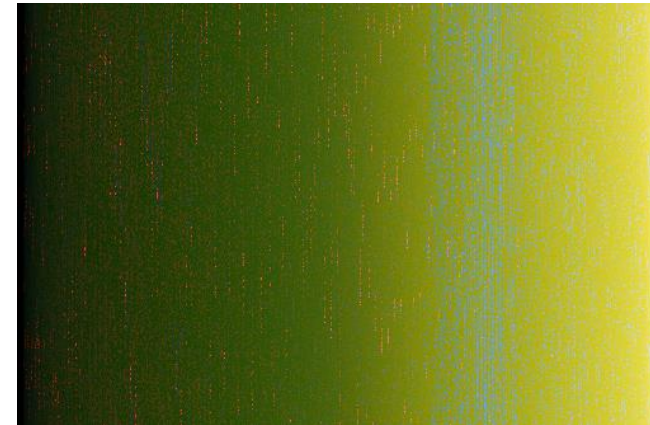
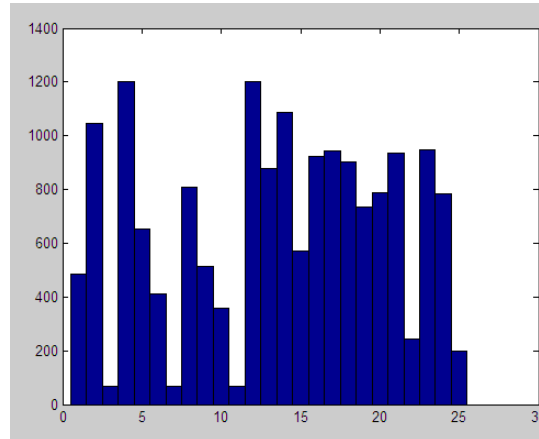


Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, [Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words](#), IJCV 2008.

# Visual words/bags of words

- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides fixed dimensional vector representation for sets
- + very good results in practice
- background and foreground mixed when bag covers whole image -> *is it really instance recognition?*
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

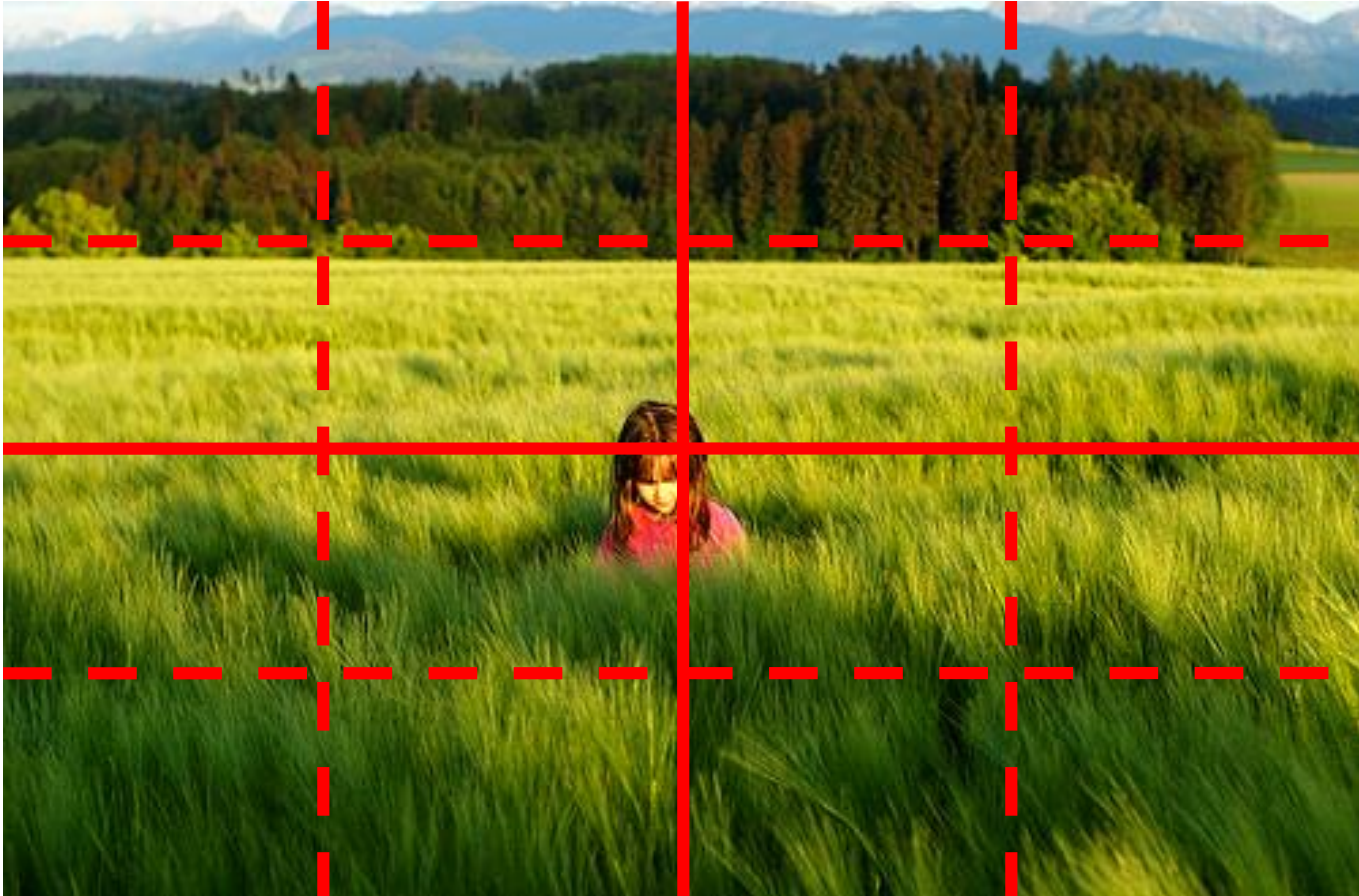
# But what about layout?



All of these images have the same color histogram.  
How to extend bag of words?



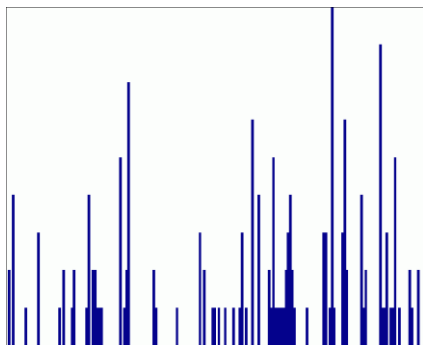
# Spatial pyramid



Compute histogram in each spatial bin

# Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

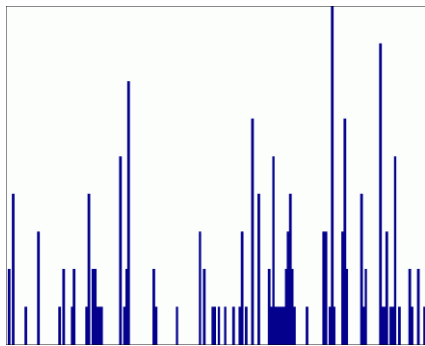


level 0

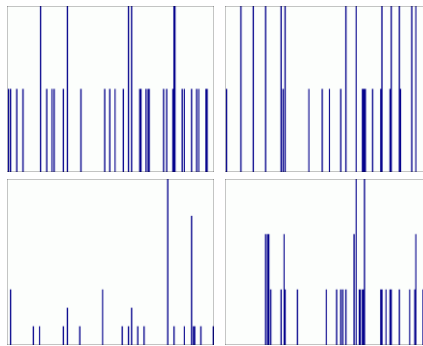


# Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



level 0



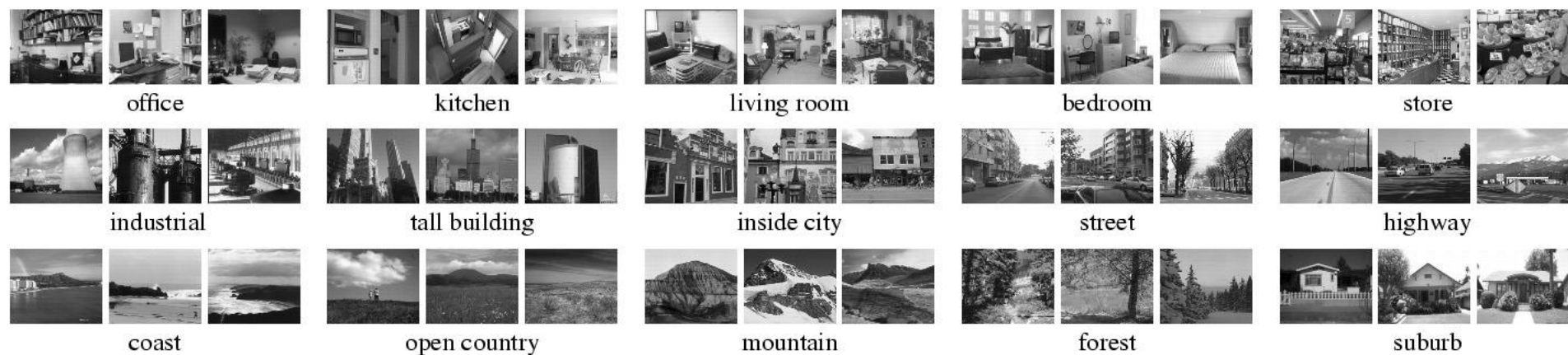
level 1

# Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



# Scene category dataset



## Multi-class classification results (100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 ( $1 \times 1$ )	45.3 $\pm$ 0.5		72.2 $\pm$ 0.6	
1 ( $2 \times 2$ )	53.6 $\pm$ 0.3	56.2 $\pm$ 0.6	77.9 $\pm$ 0.6	79.0 $\pm$ 0.5
2 ( $4 \times 4$ )	61.7 $\pm$ 0.6	64.7 $\pm$ 0.7	79.4 $\pm$ 0.3	<b>81.1</b> $\pm$ 0.3
3 ( $8 \times 8$ )	63.3 $\pm$ 0.8	<b>66.8</b> $\pm$ 0.6	77.2 $\pm$ 0.4	80.7 $\pm$ 0.3

# Recognition Issues

How to summarize the content of an entire image?

**How to gauge overall similarity?**

How large should the vocabulary be?

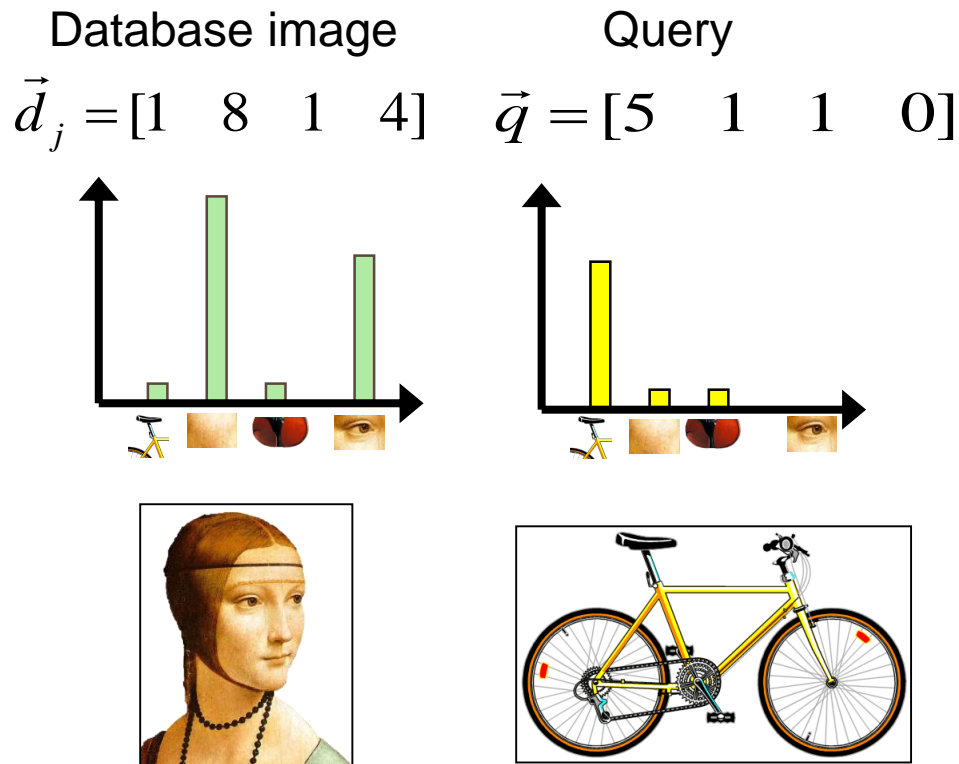
How to perform quantization efficiently?

How to score the retrieval results?

How might we add more spatial verification?

# Comparing bags of words

Compute cosine similarity (normalized scalar (dot) product) between their occurrence counts, then rank and pick smallest. *Nearest neighbor* search for similar images.



$$\text{sim}(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^V d_j(i) \times q(i)}{\sqrt{\sum_{i=1}^V d_j(i)^2} \times \sqrt{\sum_{i=1}^V q(i)^2}}$$

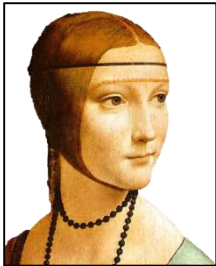
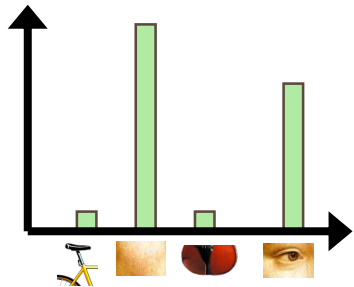
for vocabulary of  $V$  words

# Comparing bags of words

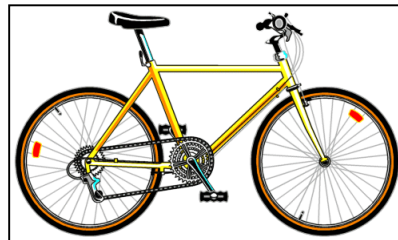
Why might we use cosine similarity here?

What 'intuitive' effect does this provide?

Database image  
 $\vec{d}_j = [1 \quad 8 \quad 1 \quad 4]$



Query  
 $\vec{q} = [5 \quad 1 \quad 1 \quad 0]$



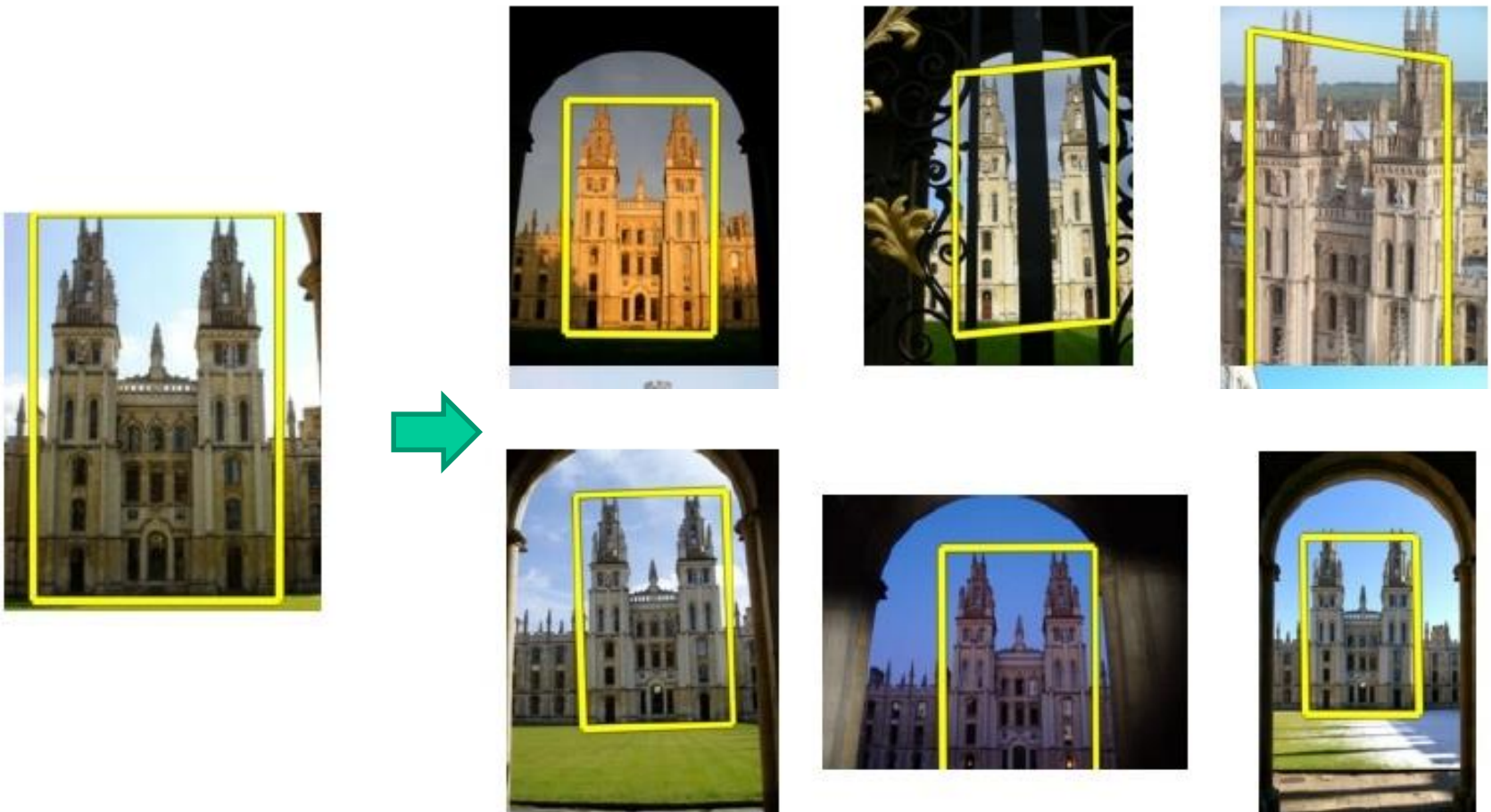
$$\text{sim}(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^V d_j(i) \times q(i)}{\sqrt{\sum_{i=1}^V d_j(i)^2} \times \sqrt{\sum_{i=1}^V q(i)^2}}$$

for vocabulary of  $V$  words

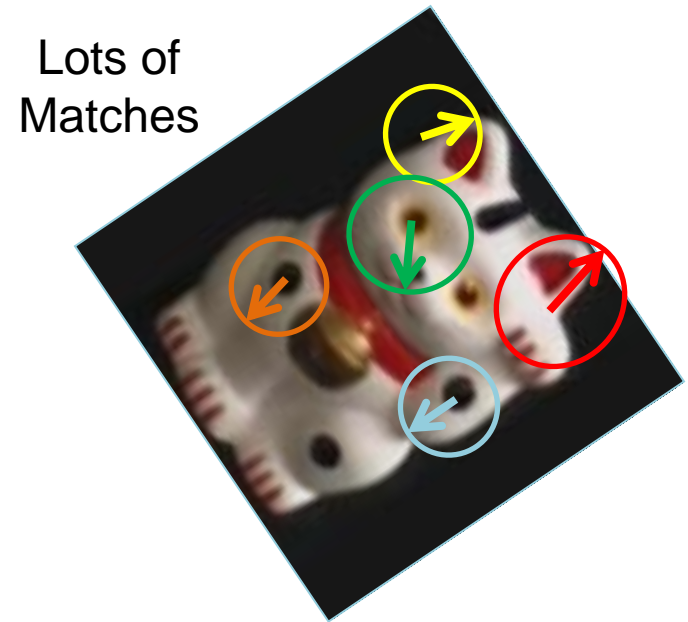


How can we quickly find images in a large database that match a given image region?



# Simple idea

See how many keypoints are close to keypoints in each other image



But this will be really, really slow!

# Fast lookup: inverted index

Index		
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142	Eau Gallie; 175
511 Traffic Information; 83	Ca d'Zan; 147	Edison, Thomas; 152
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Reale; 176
AAA National Office; 88	Canaveral Natnl Seashore; 173	Ellenton; 144-145
Abbreviations,	Cannon Creek Airpark; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106,169	Emergency Callboxes; 83
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142,148,157,159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge (I-10); 119
Agricultural Inspection Strns; 126	Cayo Costa, Name; 150	County; 120
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93	Estero; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglade,90,95,139-140,154-160
Alabama; 124	Charlotte Harbor; 150	Draining of; 156,181
Alachua; 132	Chautauqua; 116	Wildlife MA; 160
County; 131	Chieflay; 114	Wonder Gardens; 154
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawatchee, Name; 115	Fantasy of Flight; 95
Alfred B Maclay Gardens; 106	Circus Museum, Ringling; 147	Fayer Dykes SP; 171
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180	Fires, Forest; 166
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180	Fires, Prescribed ; 148
Alligator Hole (definition); 157	City Maps,	Fisherman's Village; 151
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195	Flagler County; 171
Alligators; 100,135,138,147,156	Jacksonville; 163	Flagler, Henry; 97,165,167,171
Anastasia Island; 170	Kissimmee Expwys; 192-193	Florida Aquarium; 186
Anhaica; 108-109,146	Miami Expressways; 194-195	Florida,
Apalachicola River; 112	Orlando Expressways; 192-193	12,000 years ago; 187
Appleton Mus of Art; 136	Pensacola; 26	Cavern SP; 114
Aquifer; 102	Tallahassee; 191	Map of all Expressways; 2-3
Arabian Nights; 94	Tampa-St. Petersburg; 63	Mus of Natural History; 134
Art Museum, Ringling; 147	St. Augustine; 191	National Cemetery ; 141
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141	Part of Africa; 177
Aucilla River Project; 106	Clearwater Marine Aquarium; 187	Platform; 187
Babcock-Web WMA; 151	Collier County; 154	Sheriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 130
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Barefoot Mailmen; 182	Columbia County; 101,128	Supreme Court; 107
Barge Canal; 137	Coquina Building Material; 165	Florida's Turnpike (FTP); 178,189
Bee Line Expy; 80	Corkscrew Swamp, Name; 154	25 mile Strip Maps; 66
Belz Outlet Mall; 89	Cowboys; 85	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big "I"; 165	Cracker, Florida; 88,95,132	Exit Services; 189
Big Cypress; 155,158	Crosstown Expy; 11,35,98,143	HEFT; 76,161,190
Big Foot Monster; 105	Cuban Bread; 184	History; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Names; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161	Service Plazas; 190
Blue Angels	Dania Beach Hurricane; 184	Spur SR91; 76
	Daniel Boone, Florida Walk; 117	Ticket System; 190
	Daytona Beach; 172-173	Toll Plazas; 190
	De Land; 87	Ford, Henry; 152

- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.

# Build Inverted Index from Database



Image #1

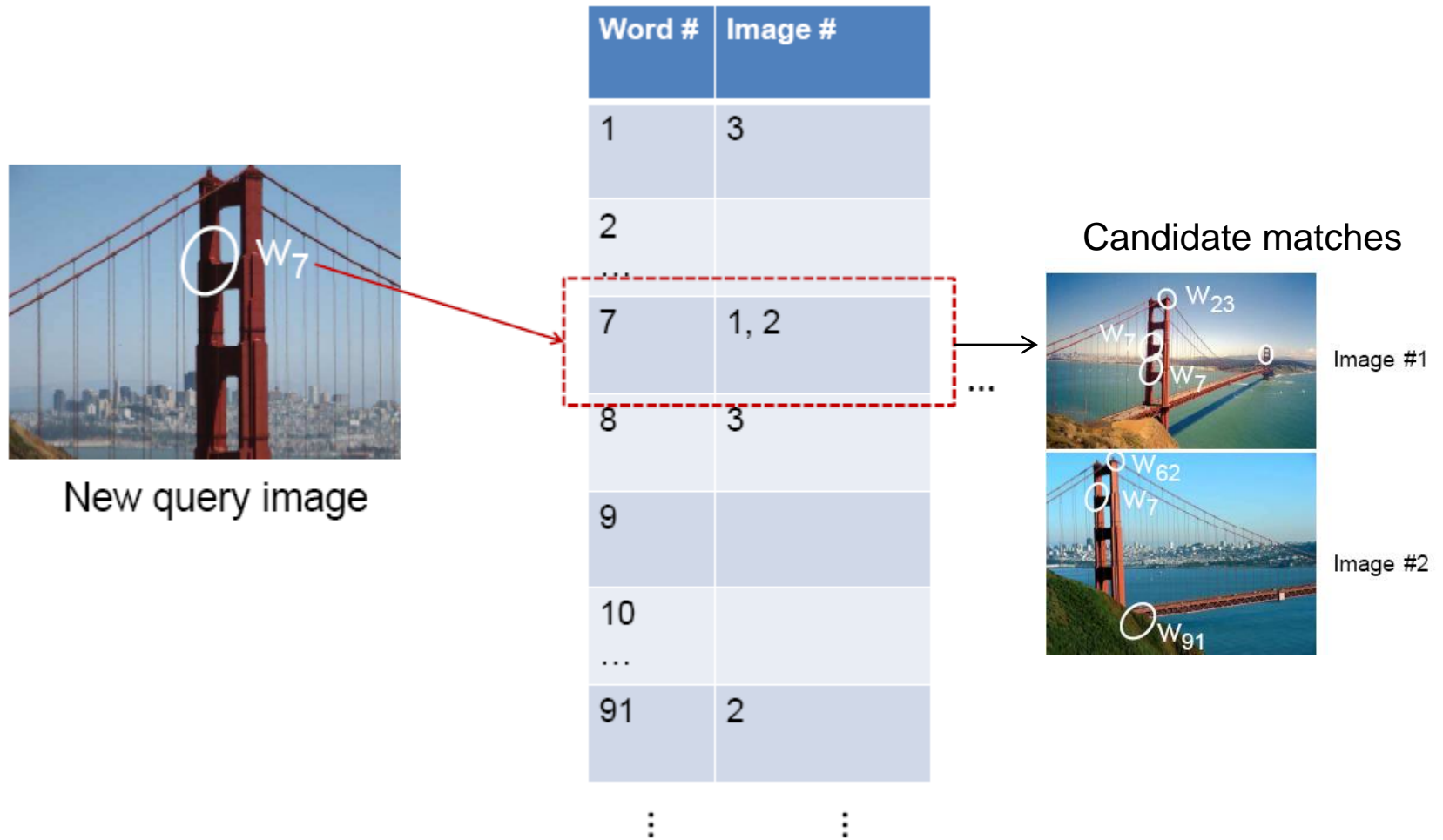
Image #2

Image #3

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

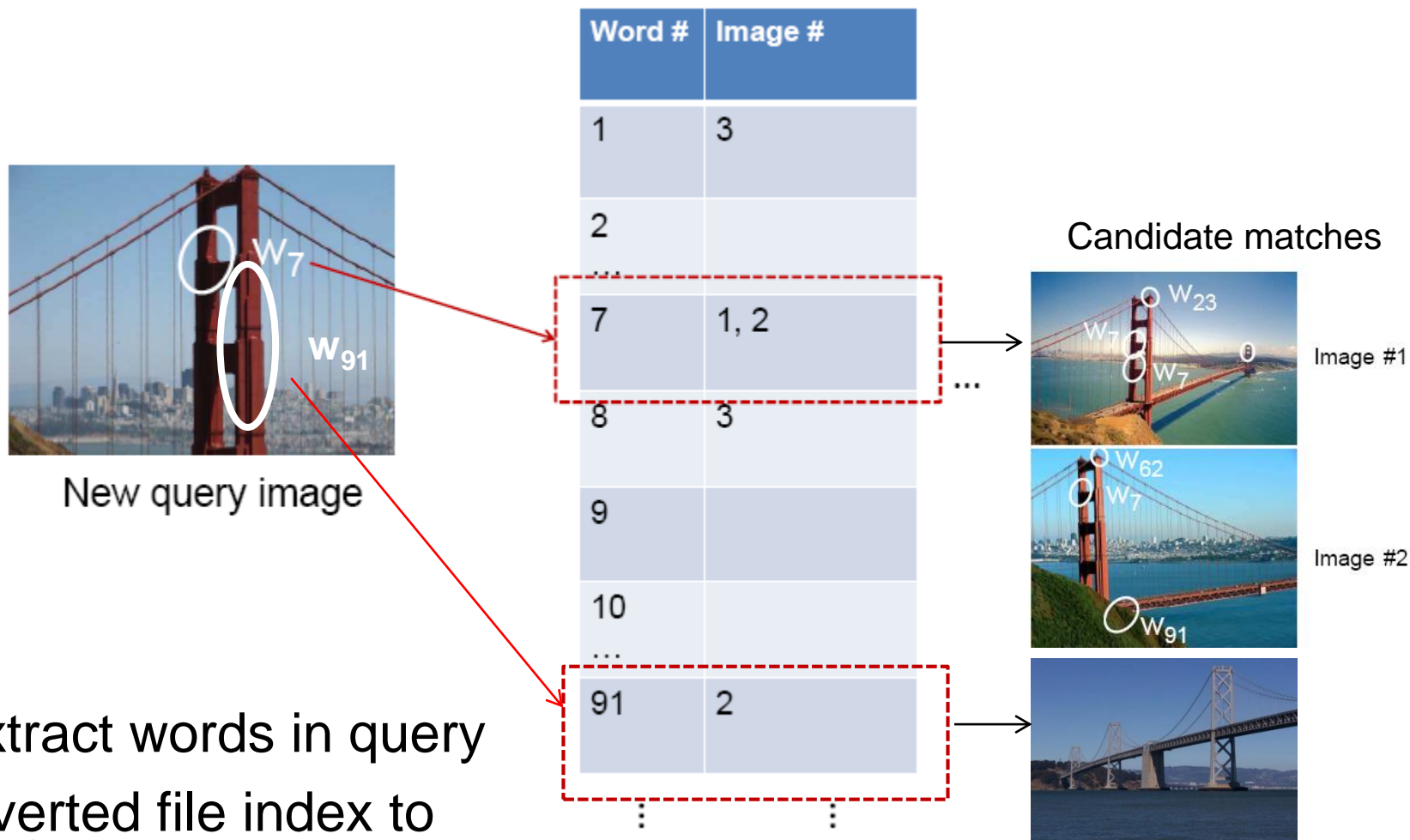
...

# Query Inverted Index





# Query Inverted Index



1. Extract words in query
2. Inverted file index to find relevant frames
3. Compare/sort word counts



# Inverted index

Key requirement: *sparsity*.

If most images contain most words, then we're not better off than exhaustive search.

- Exhaustive search would mean comparing the visual word distribution of a query versus every page.

# Recognition Issues

How to summarize the content of an entire image?  
And gauge overall similarity?

**How large should the vocabulary be? How to perform quantization (clustering) efficiently?**

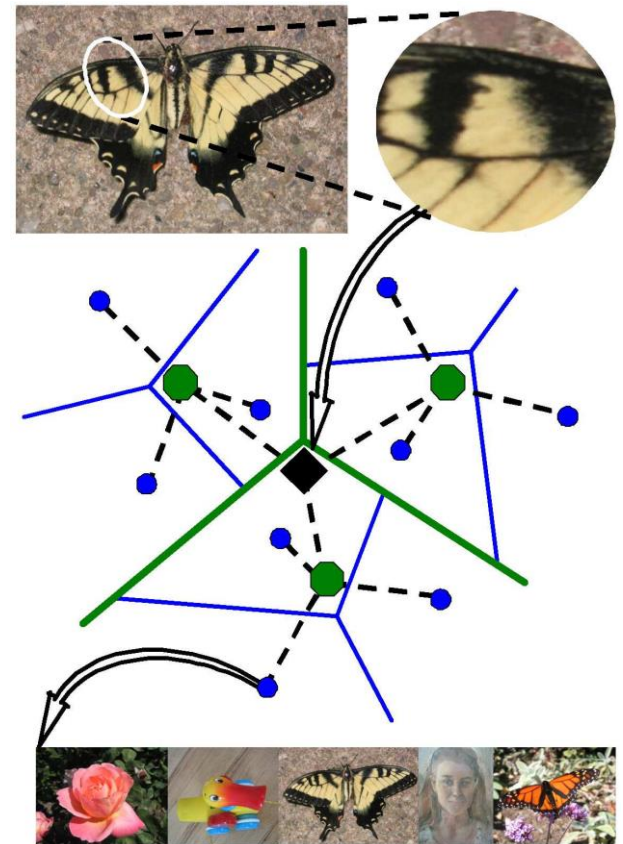
How to score the retrieval results?

How might we add more spatial verification?

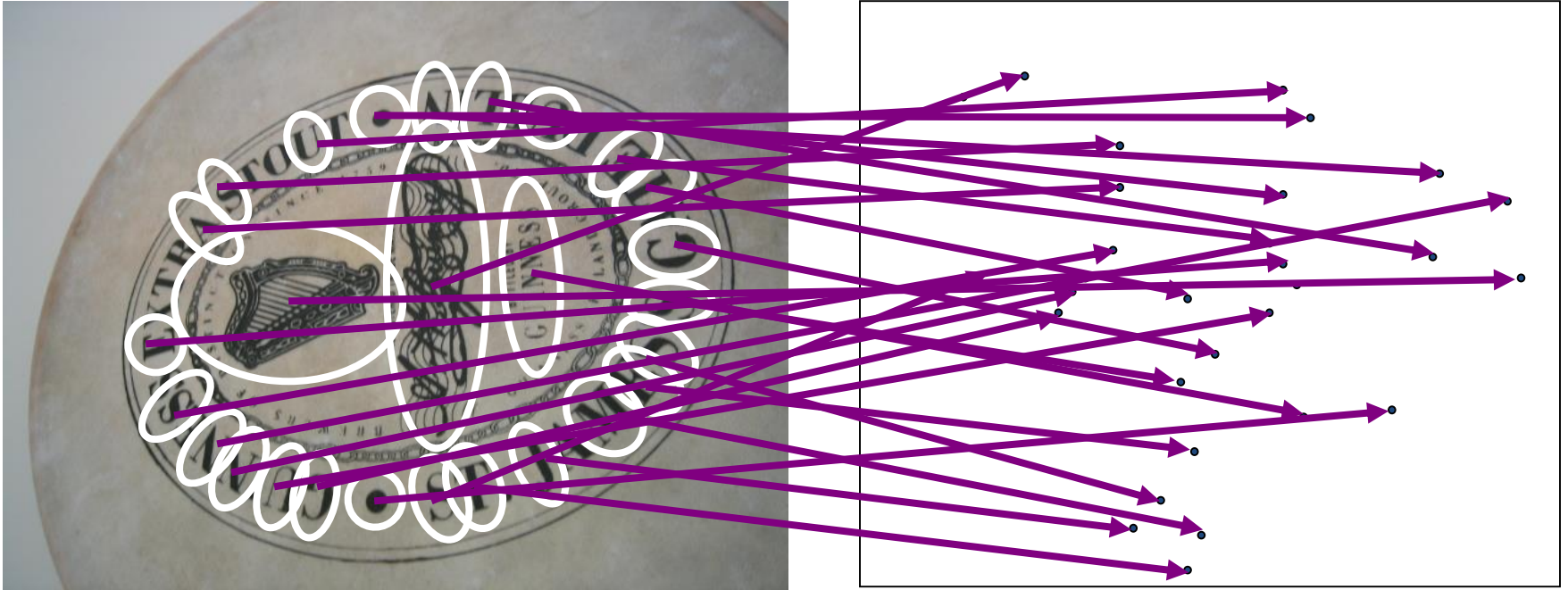
# Visual vocabularies: Issues

---

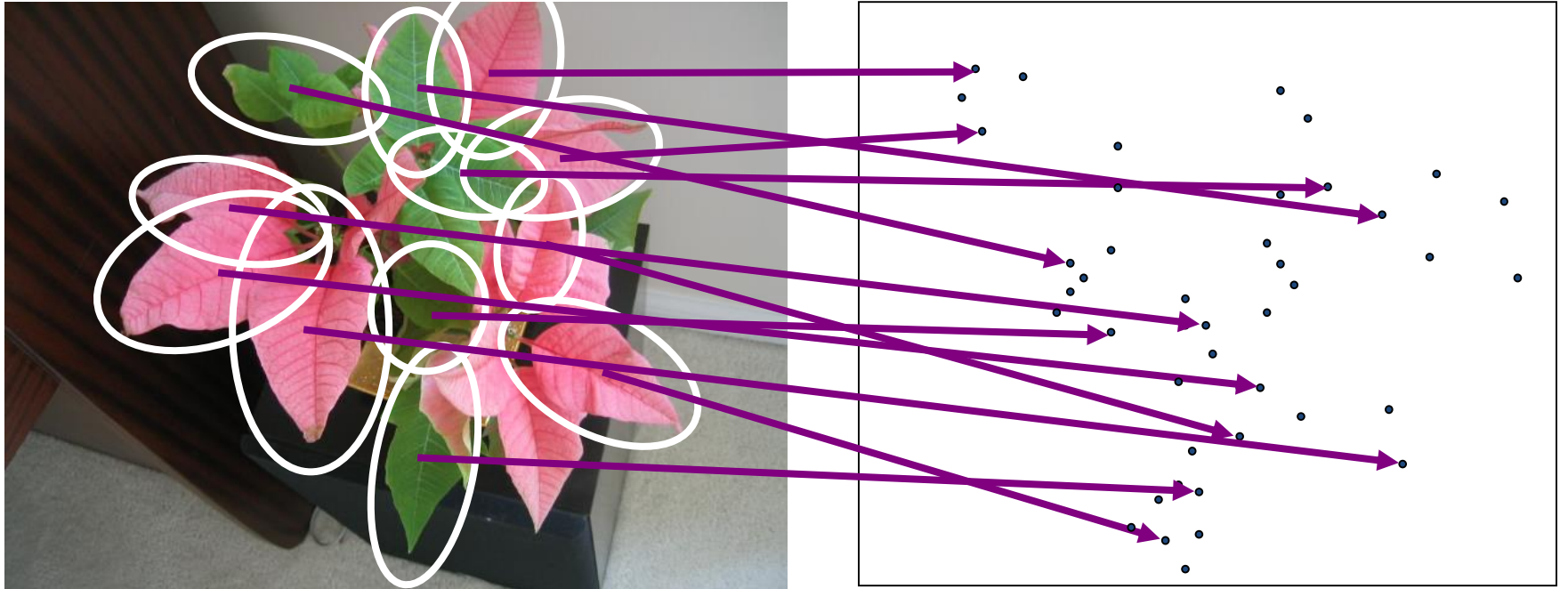
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees  
(Nister & Stewenius, 2006)



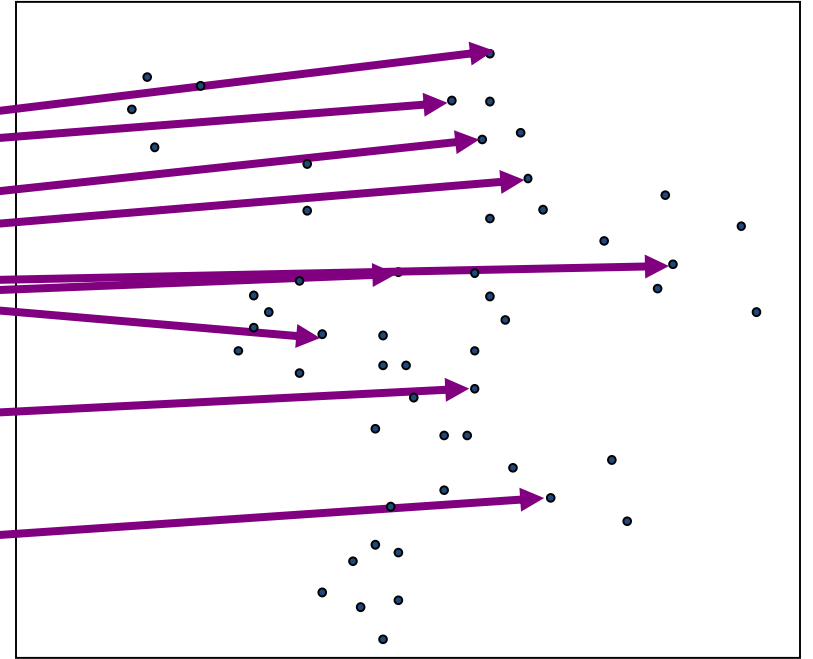
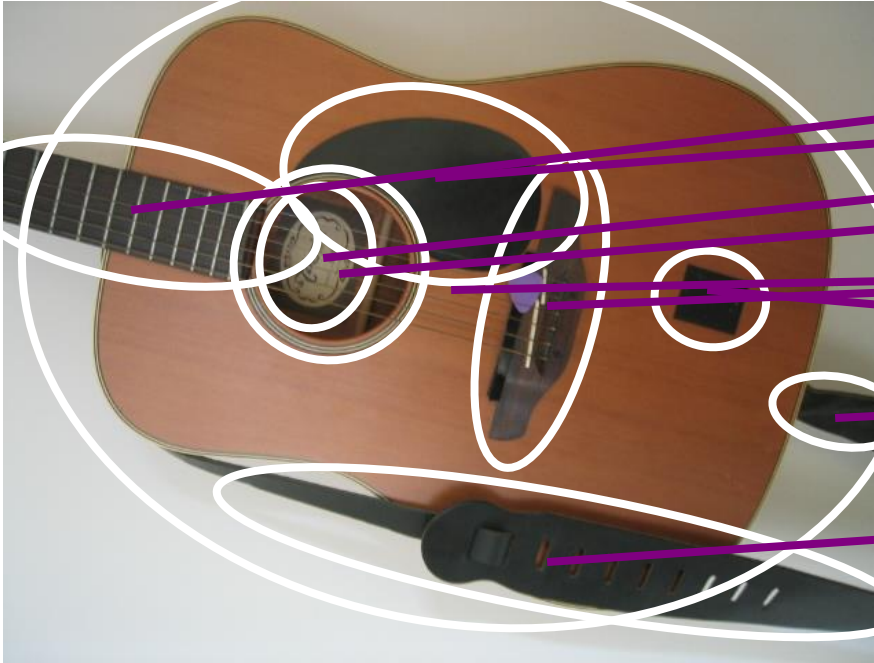
# Training the vocabulary tree



# Training the vocabulary tree

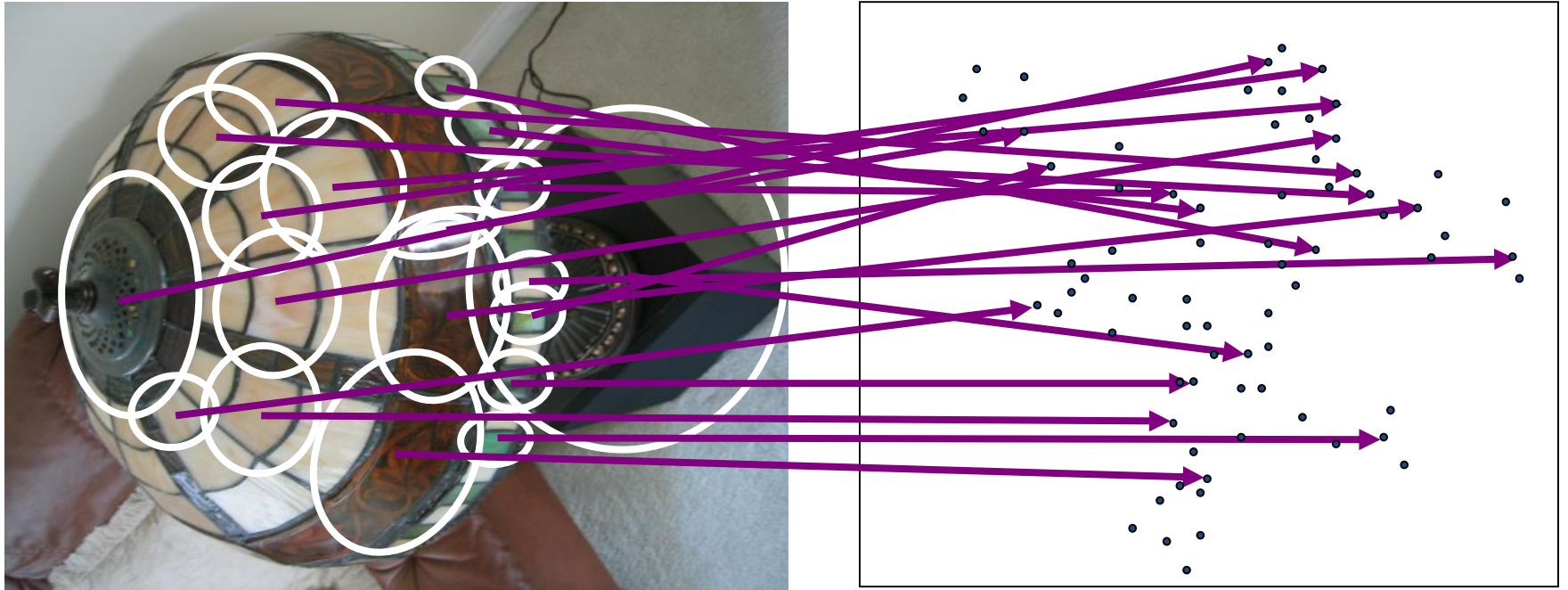


# Training the vocabulary tree

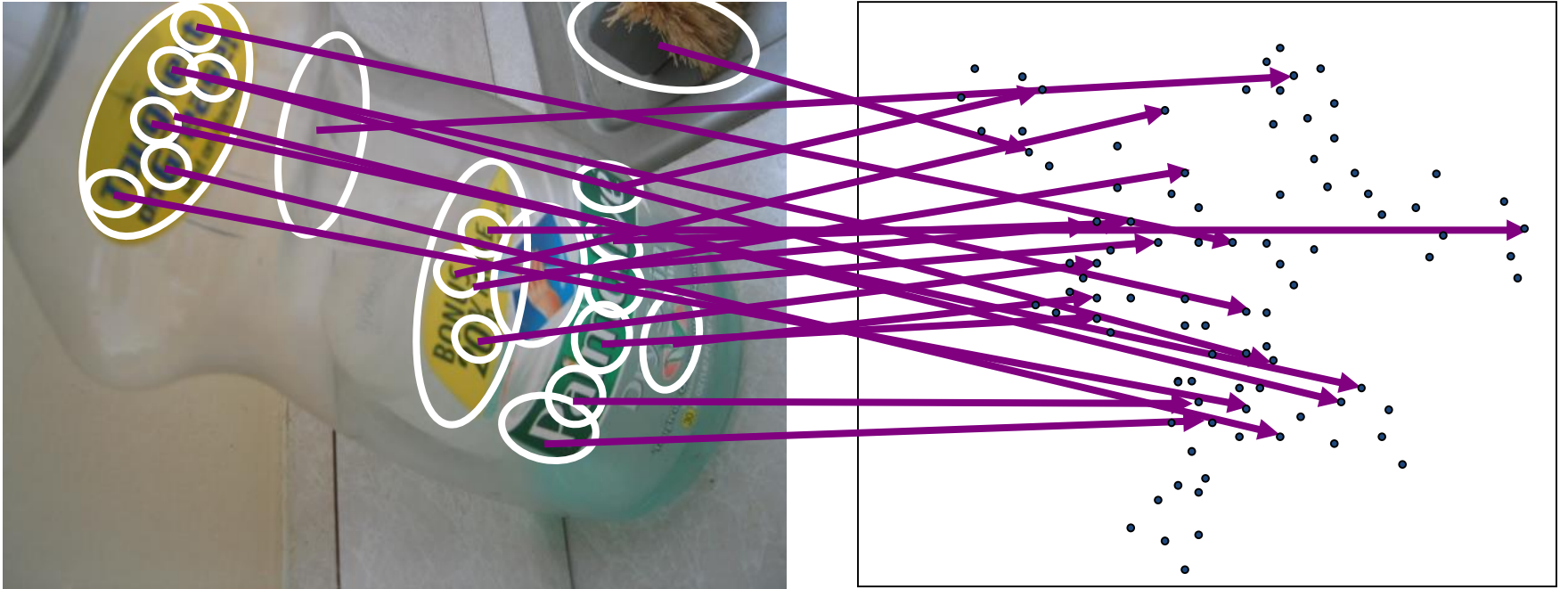




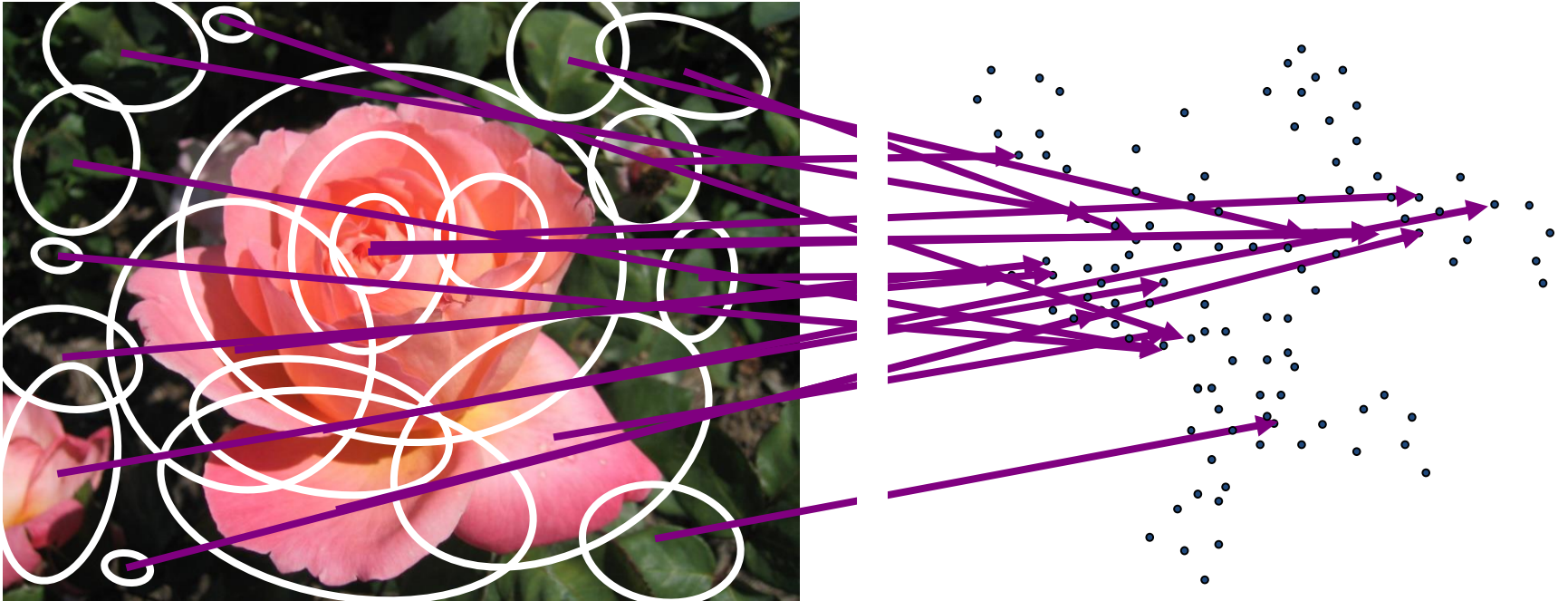
# Training the vocabulary tree

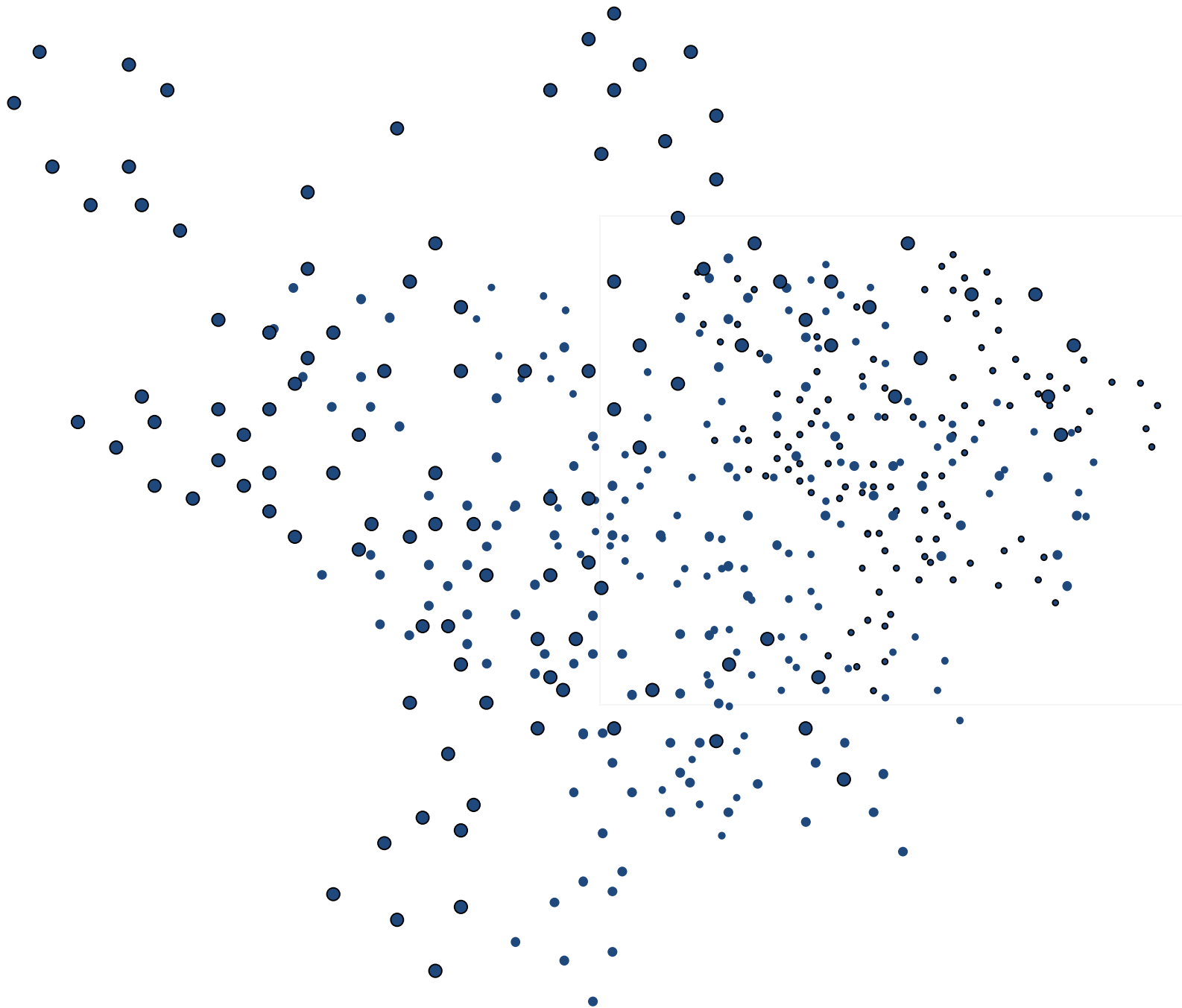


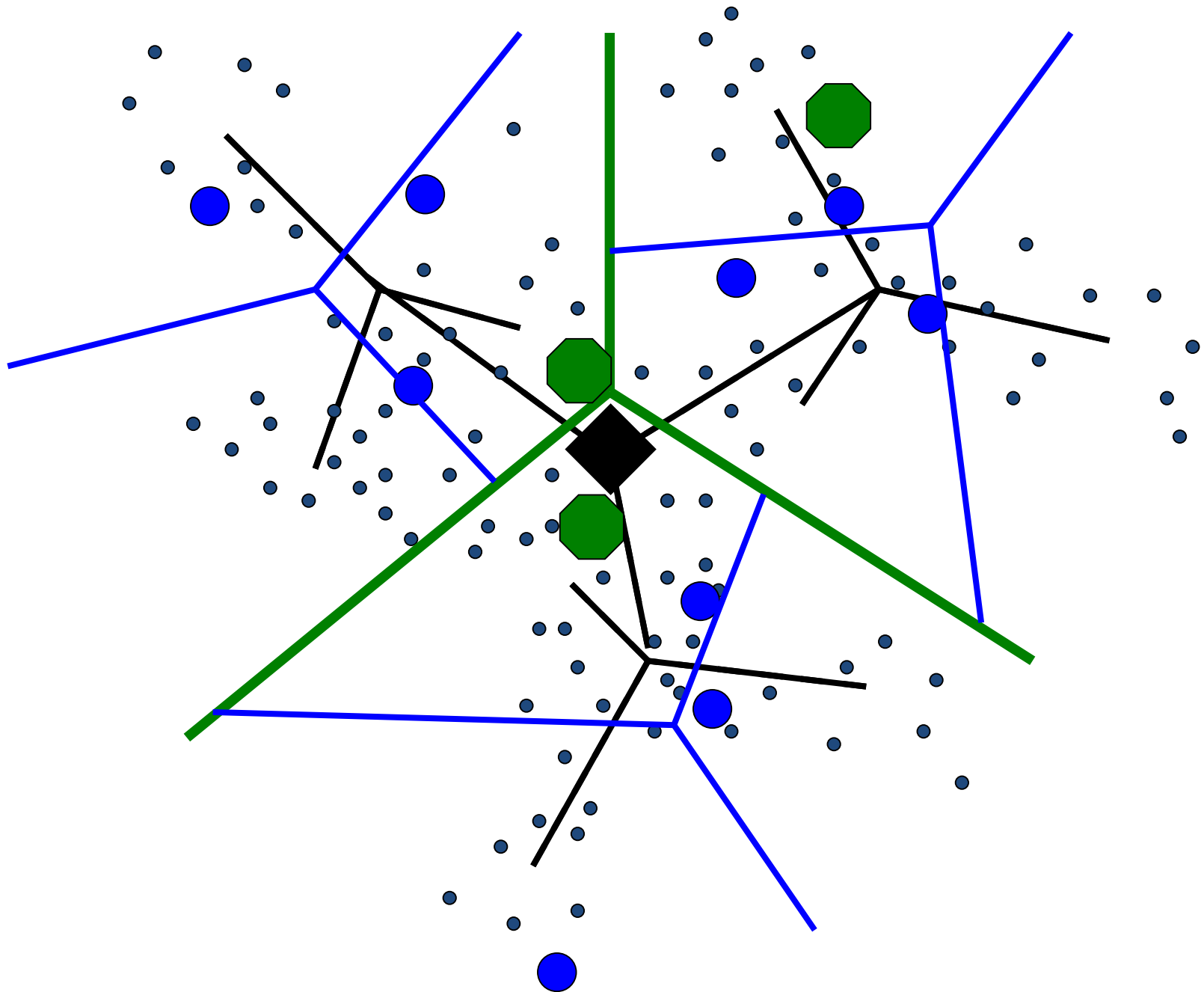
## Training the vocabulary tree

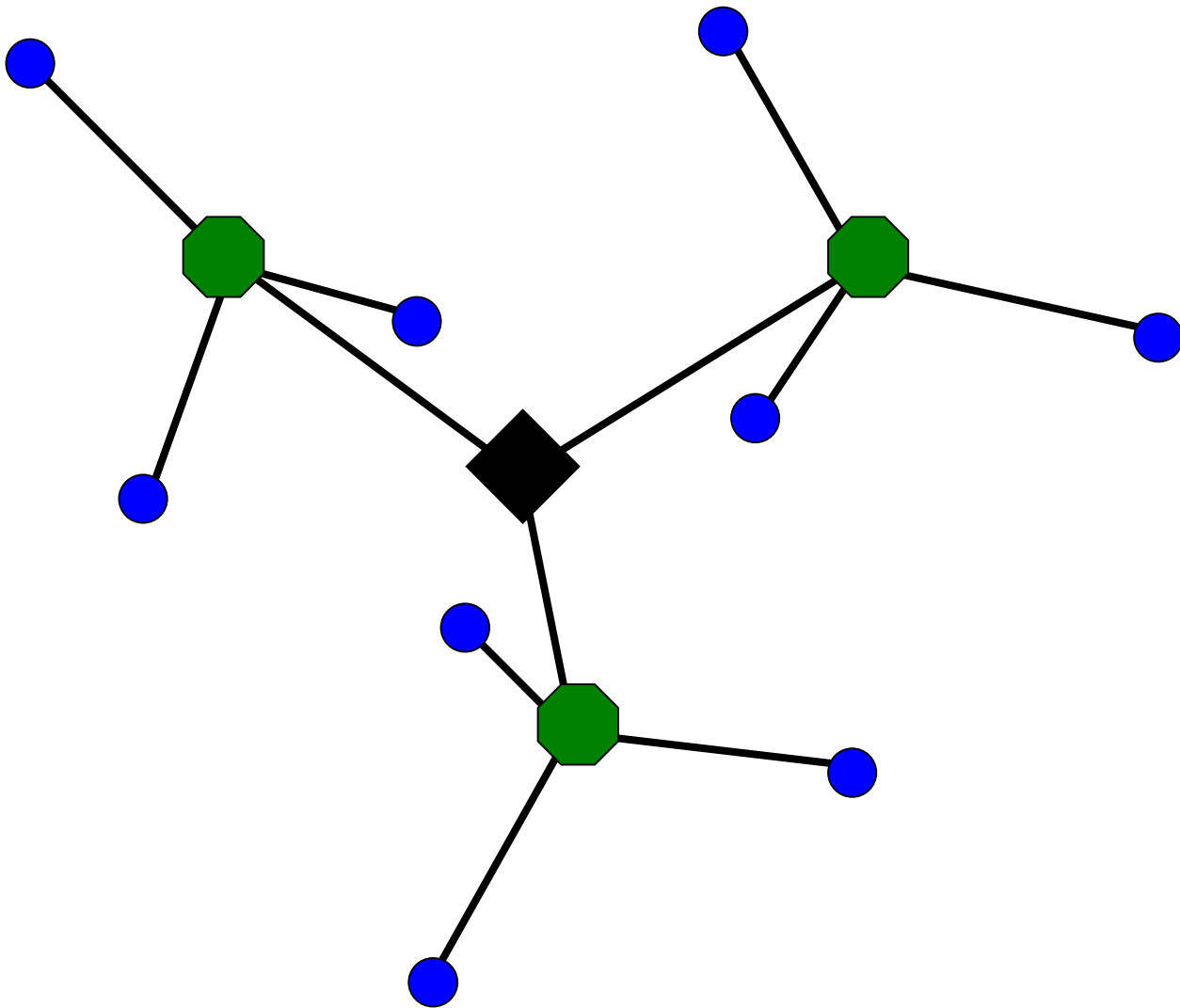


## Training the vocabulary tree

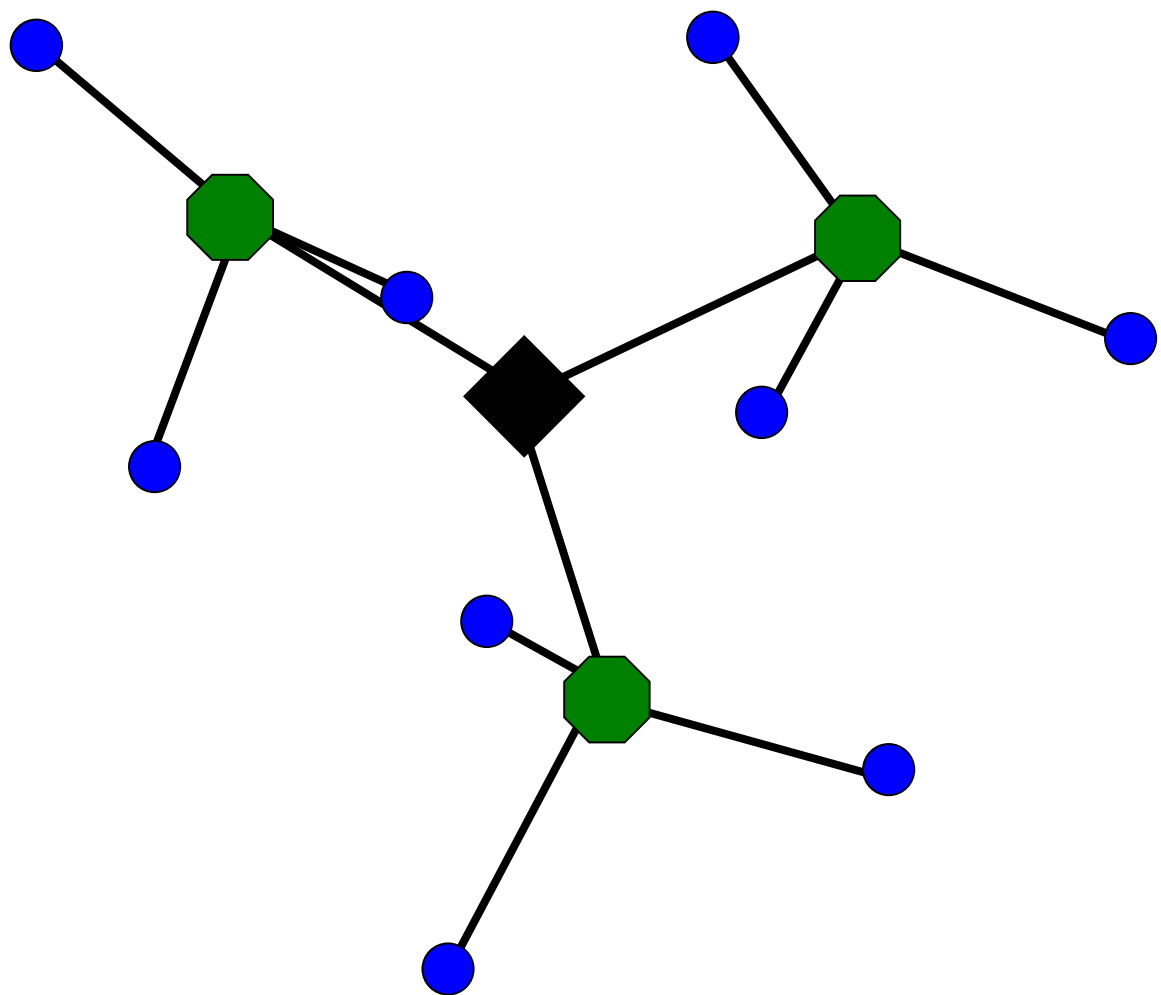


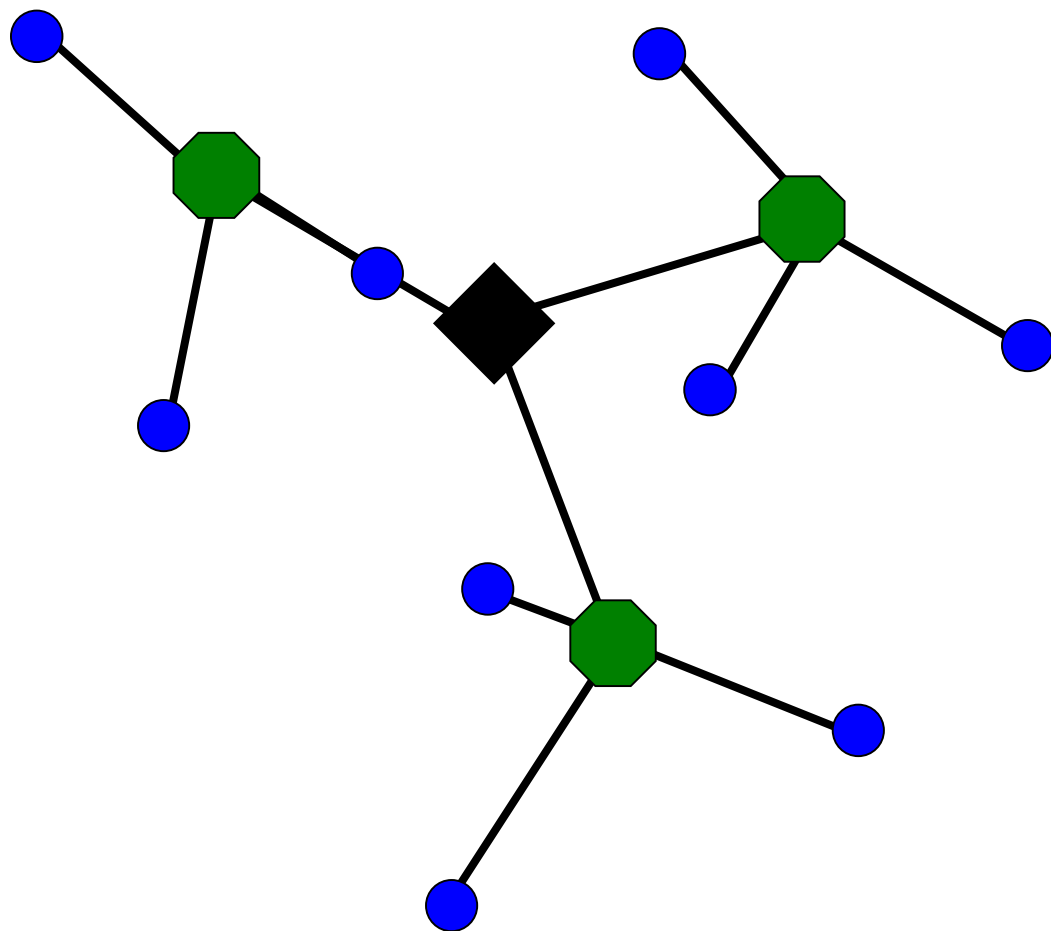


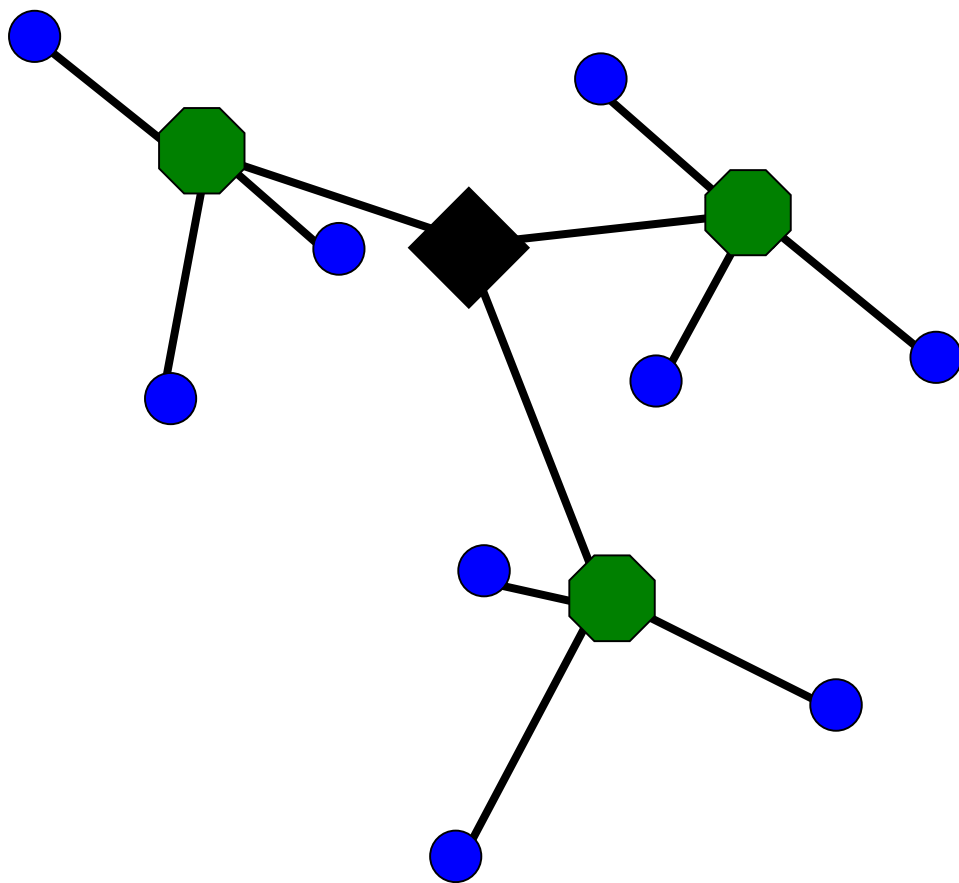


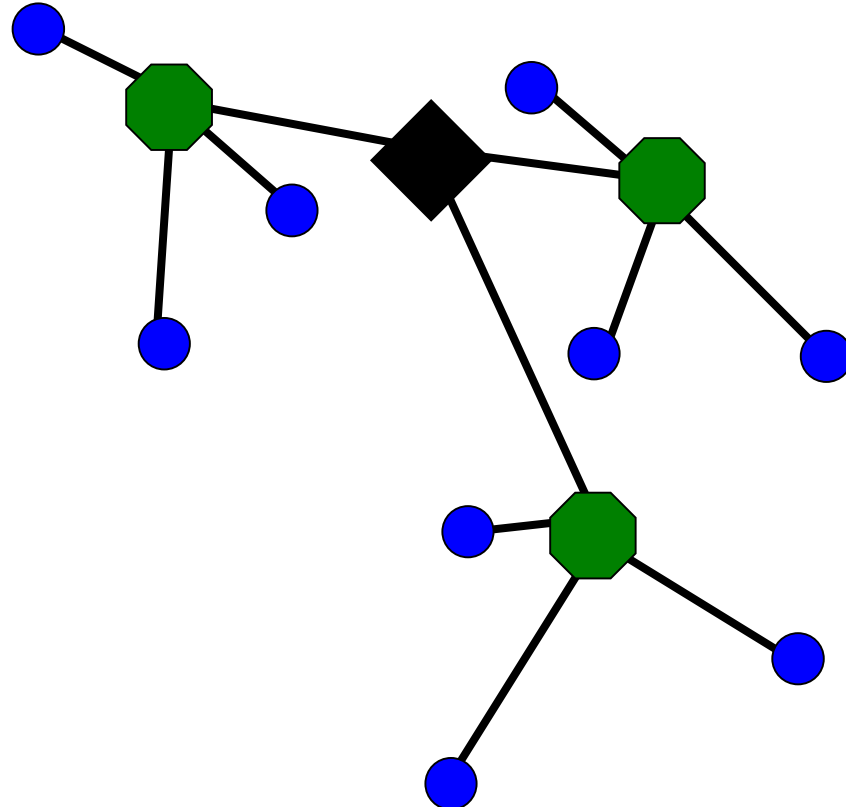


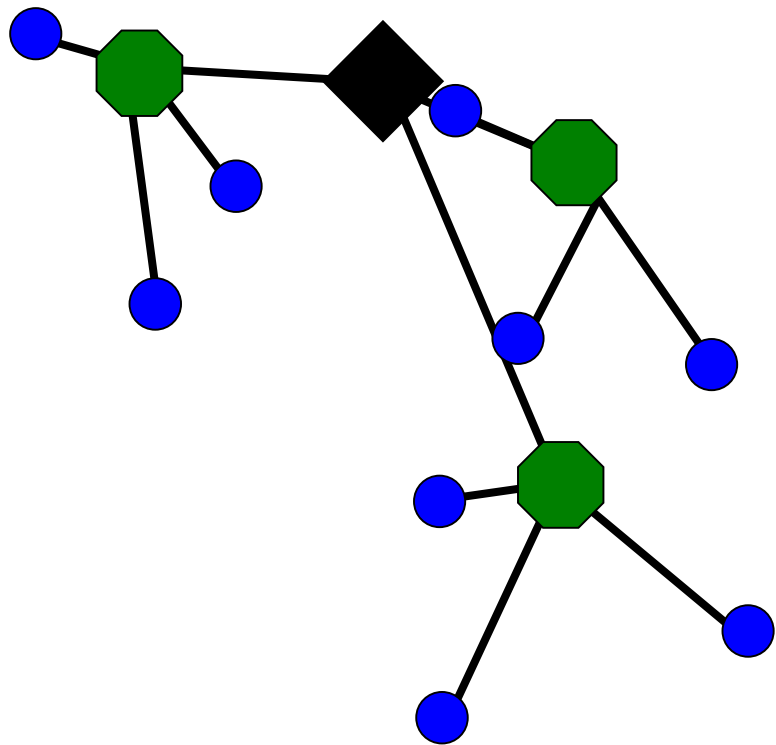


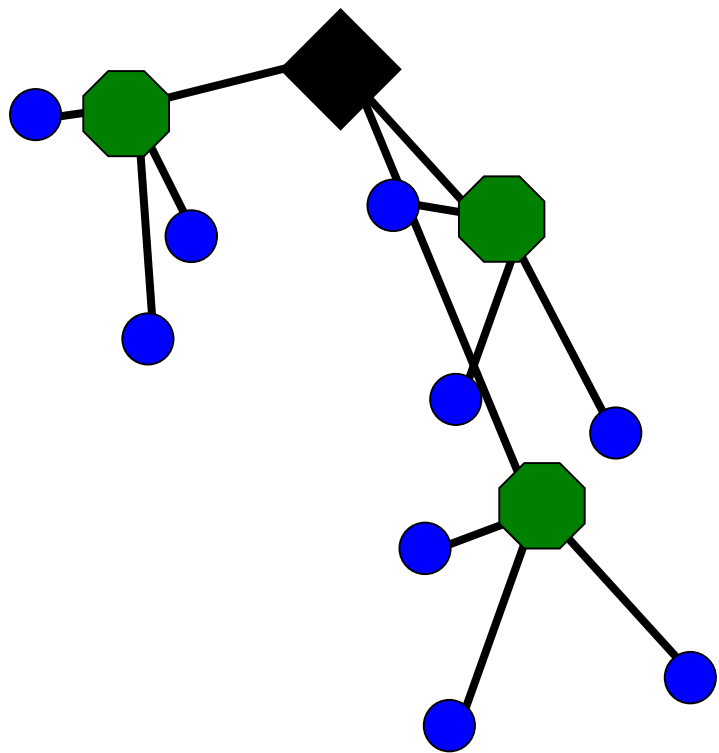




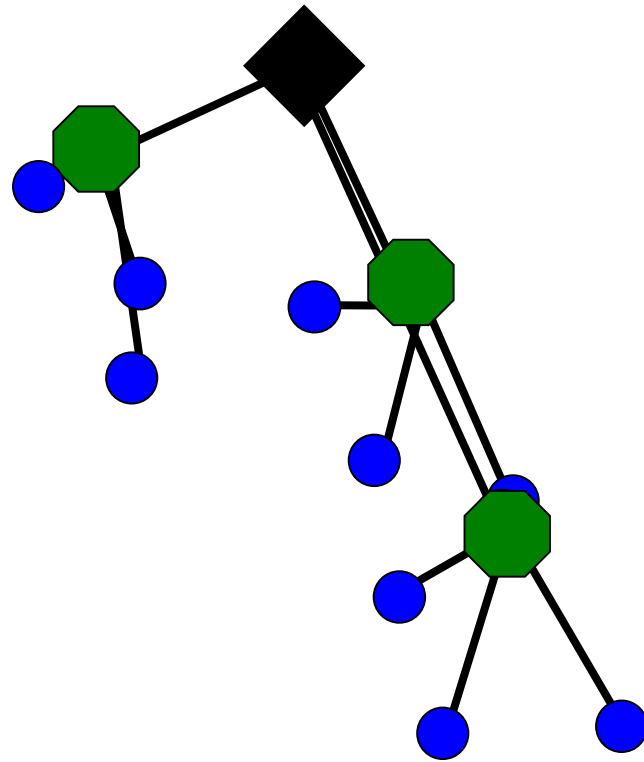


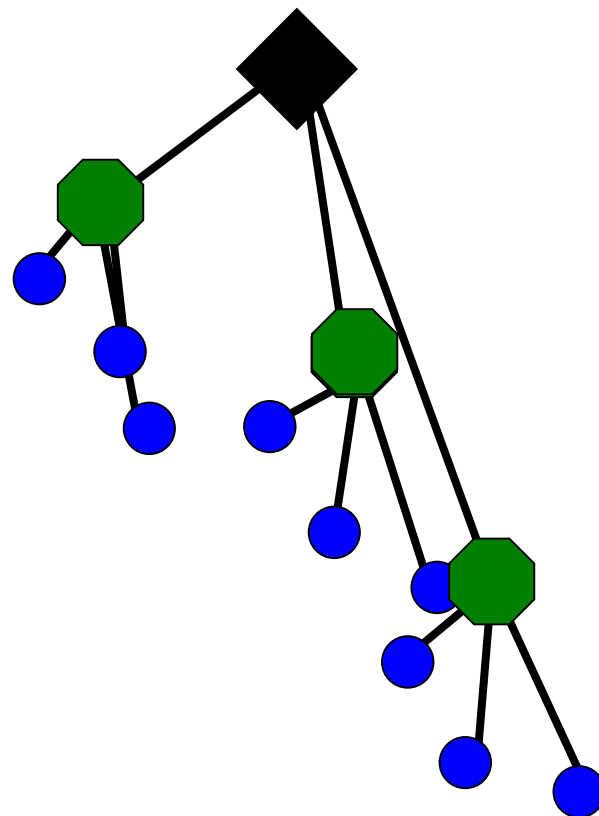




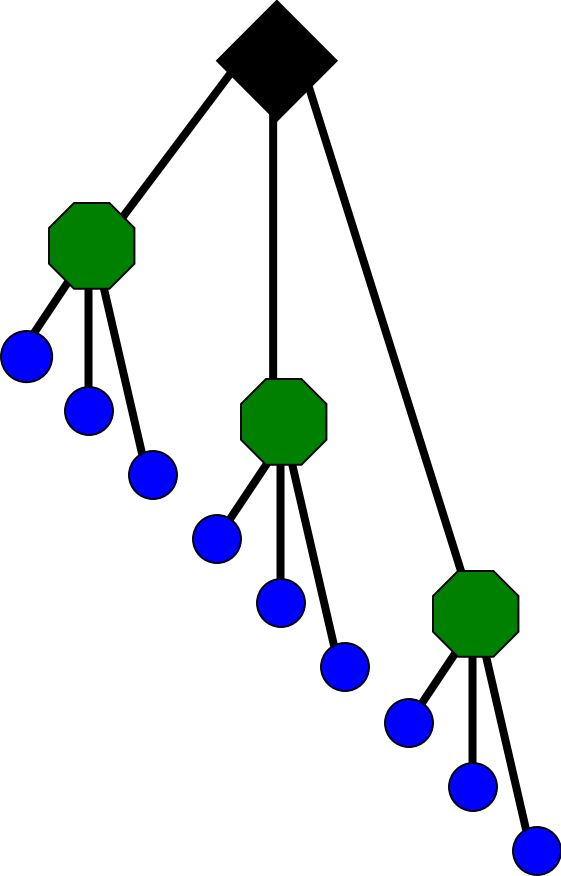




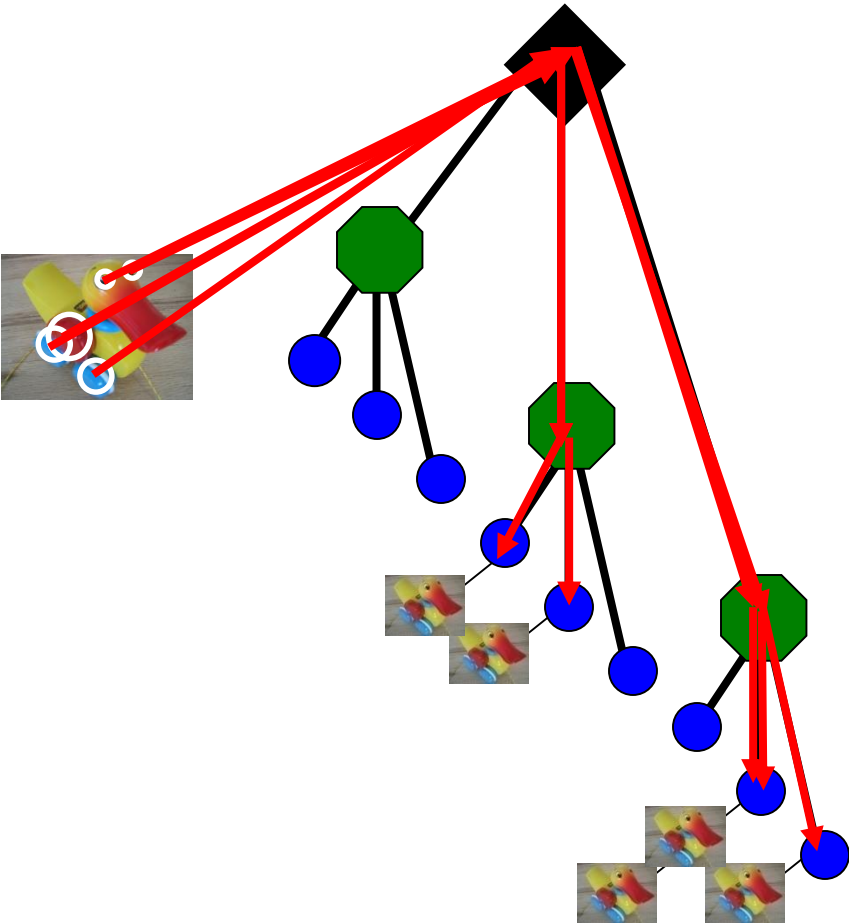


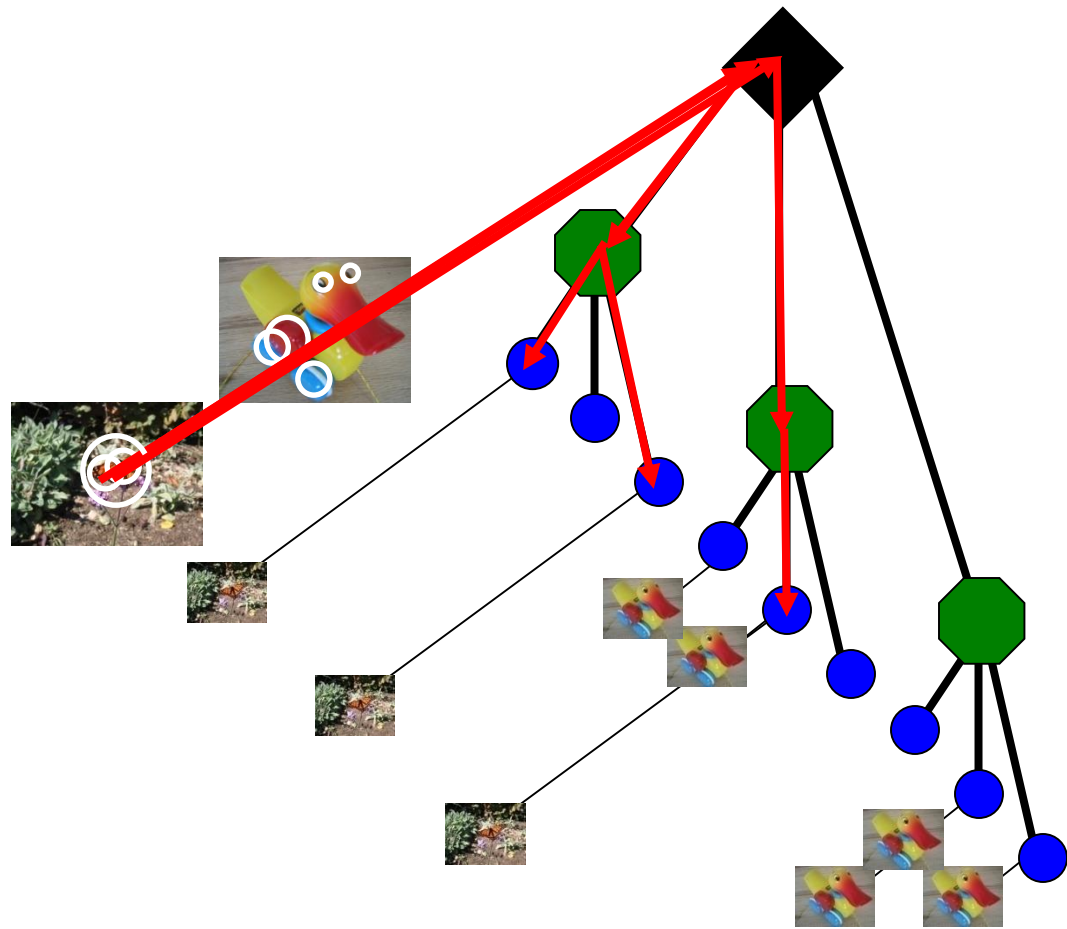


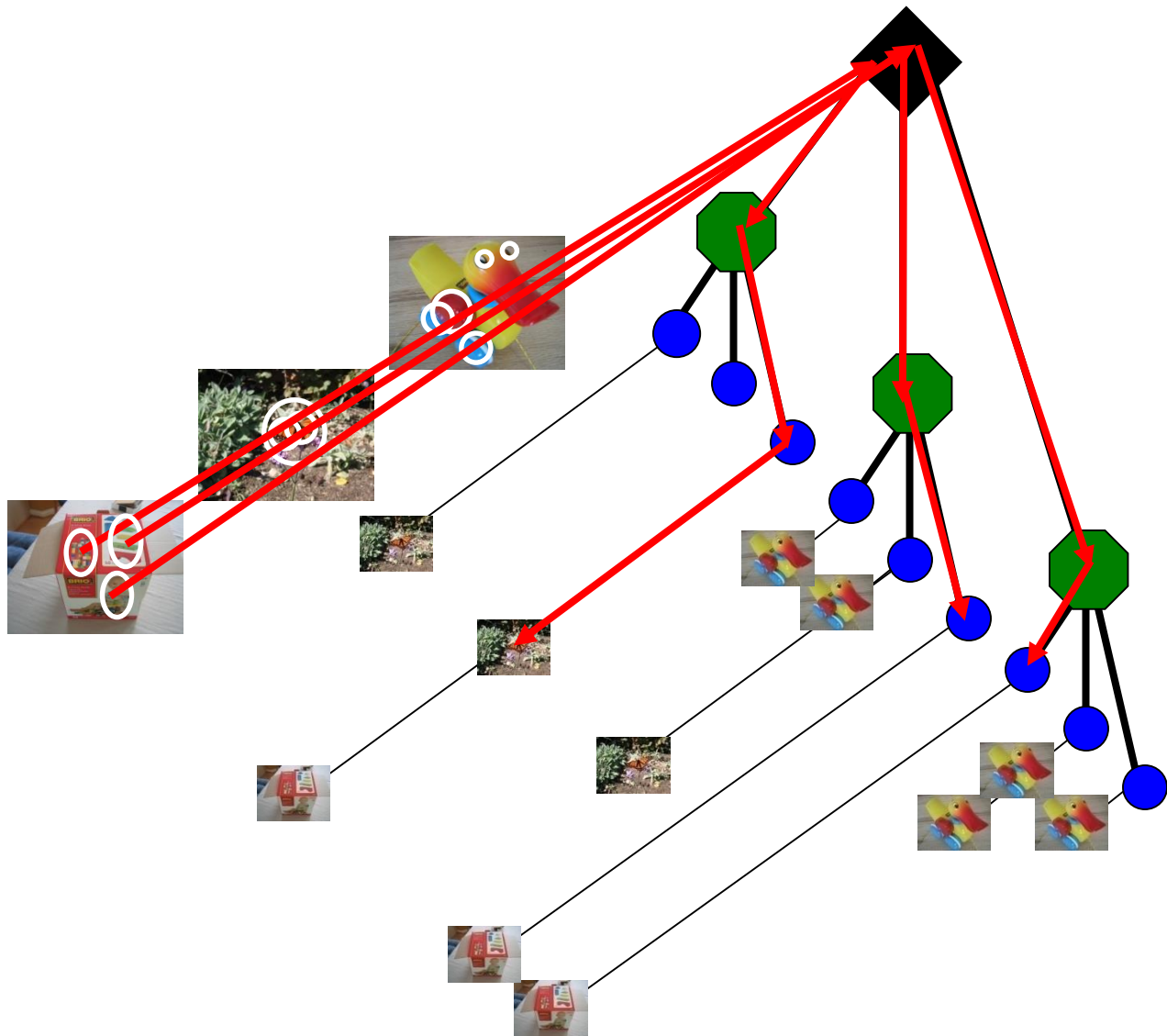
Vocabulary tree built recursively



Each leaf has inverted index

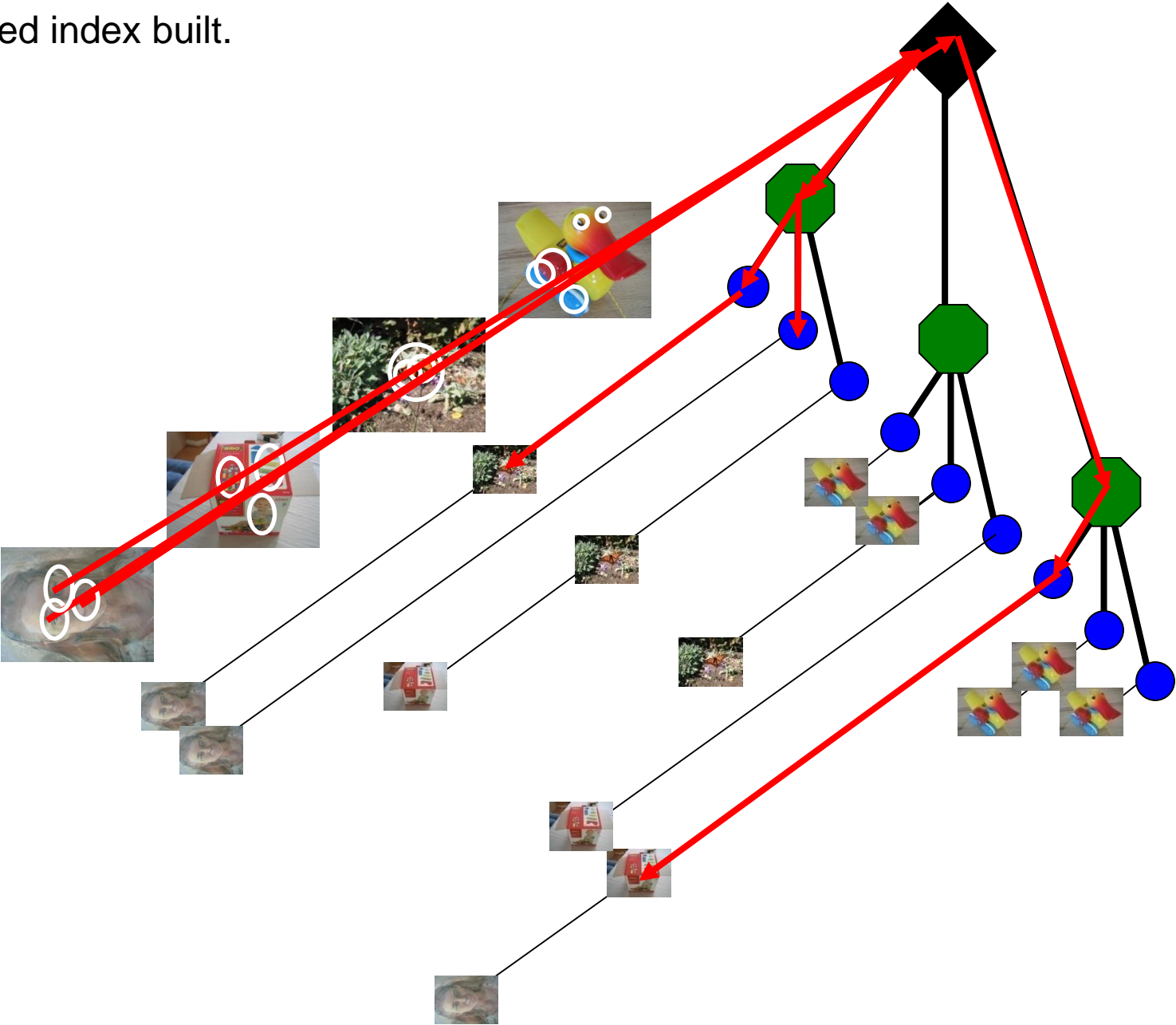


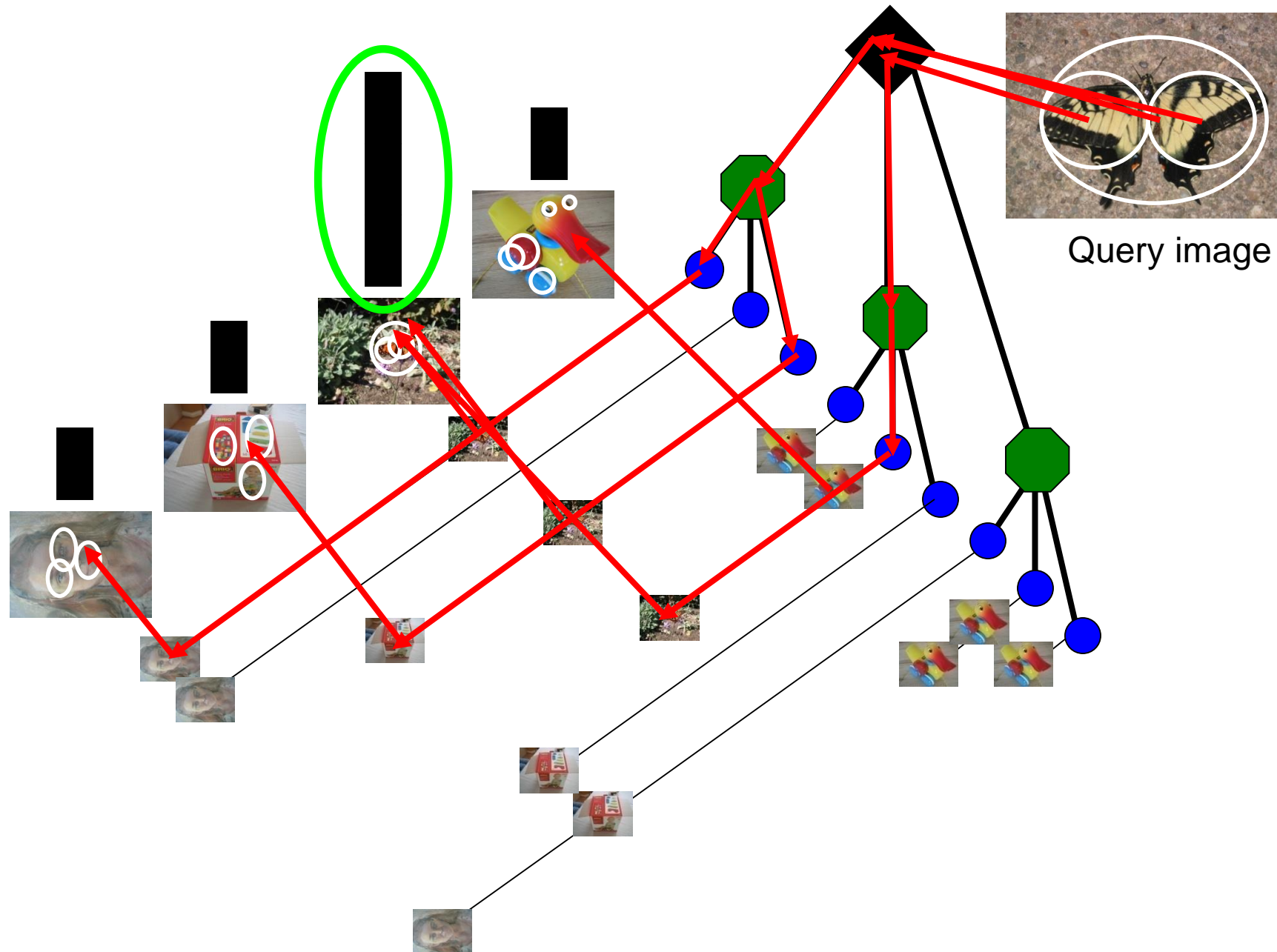




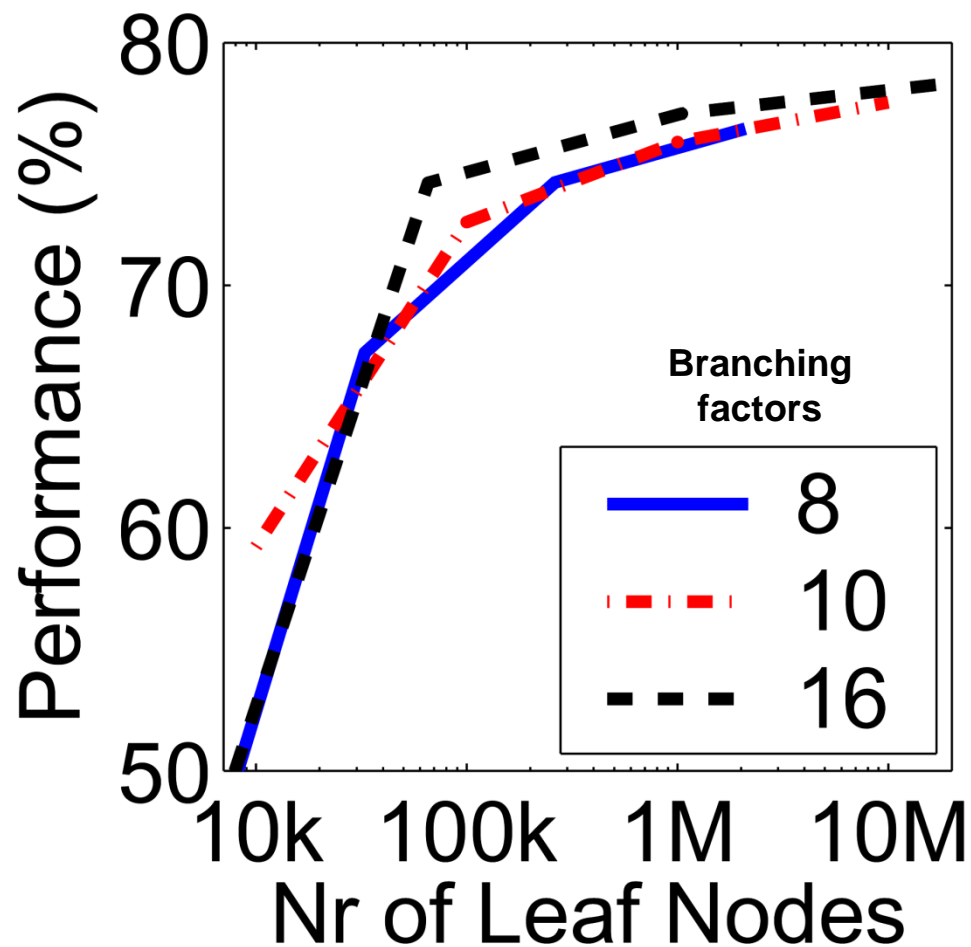


Inverted index built.



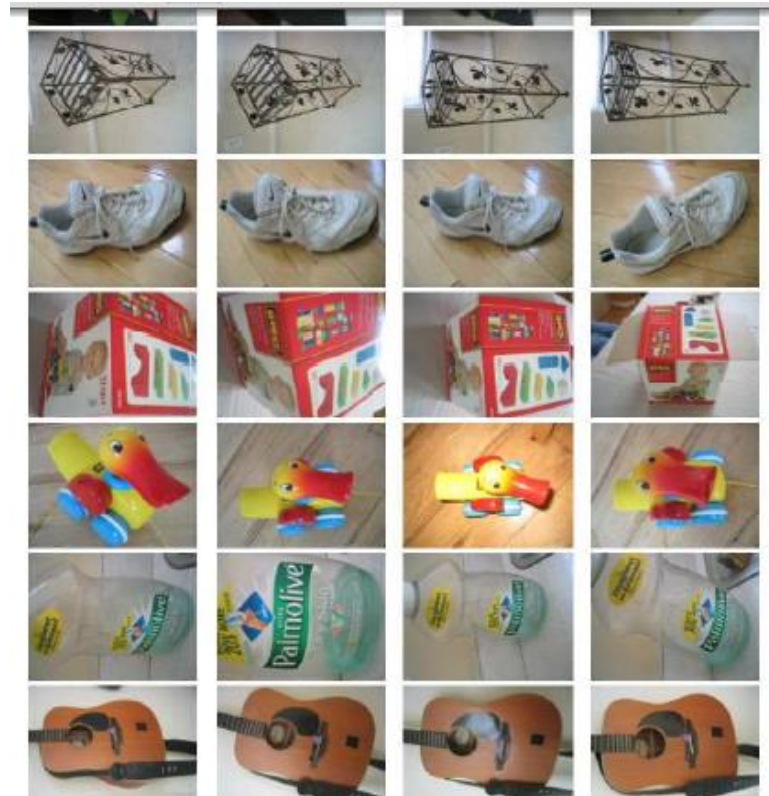


# Vocabulary size



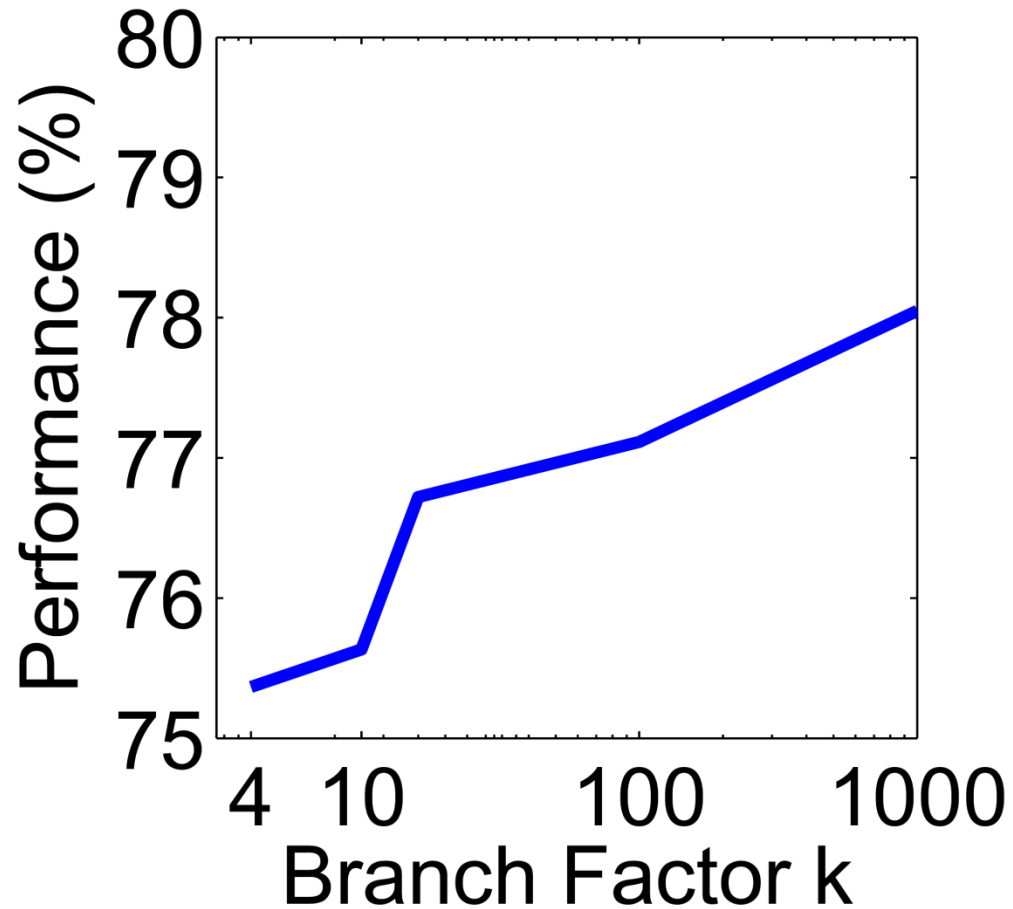
*Influence on performance, sparsity*

Recognition with 6347 images



Nister & Stewenius, CVPR 2006

Higher branch factor works better  
(but slower)

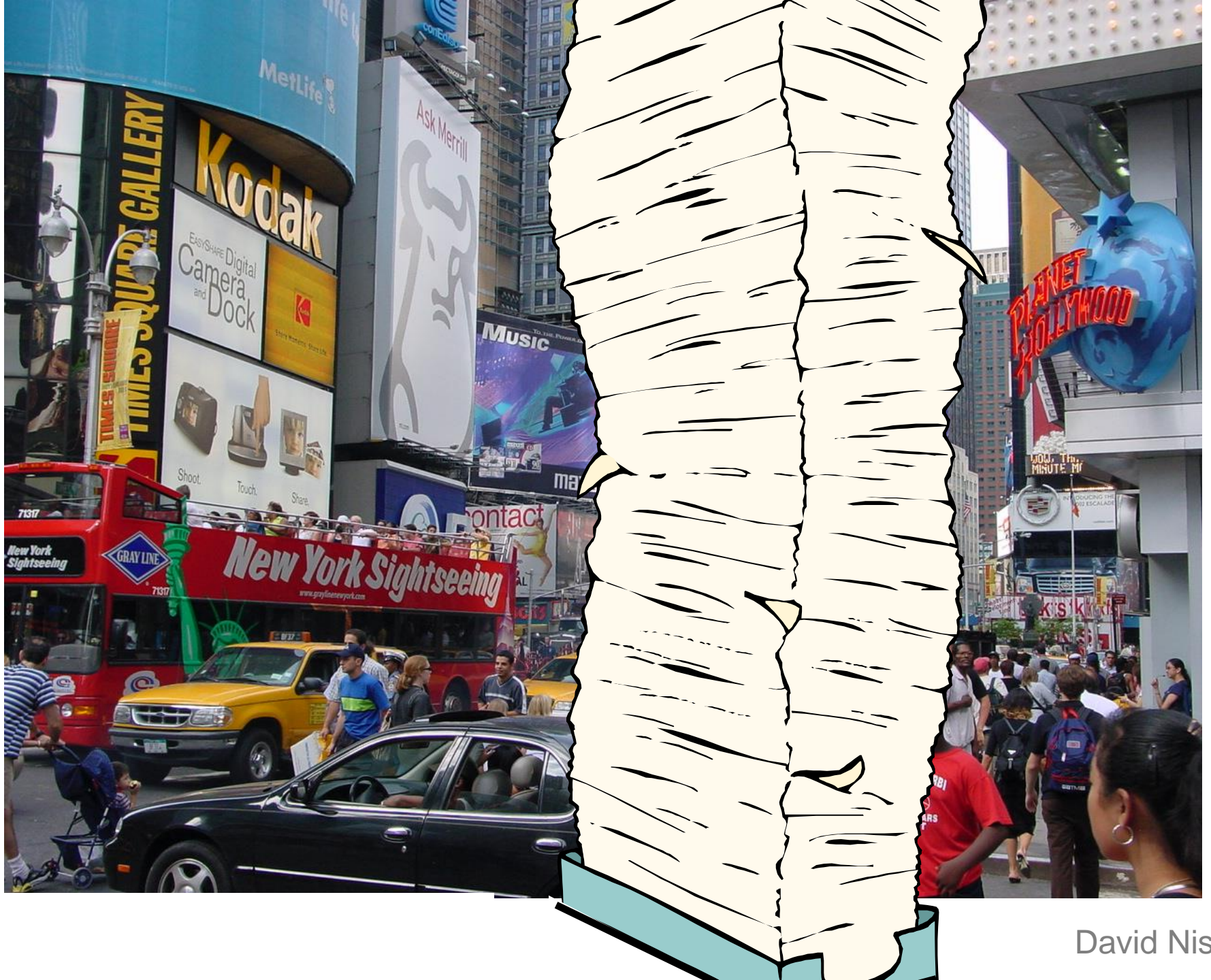


# (2006) 110,000,000 images in 5.8 Seconds

On a 50k image index













# Recognition Issues

How to summarize the content of an entire image?  
And gauge overall similarity?

How large should the vocabulary be? How to  
perform quantization efficiently?

**How to score the retrieval results?**

How might we add more spatial verification?

# Precision and Recall

True positive (tp) – correct attribution

True negative (tn) – correct rejection

False positive (fp) – incorrect attribution

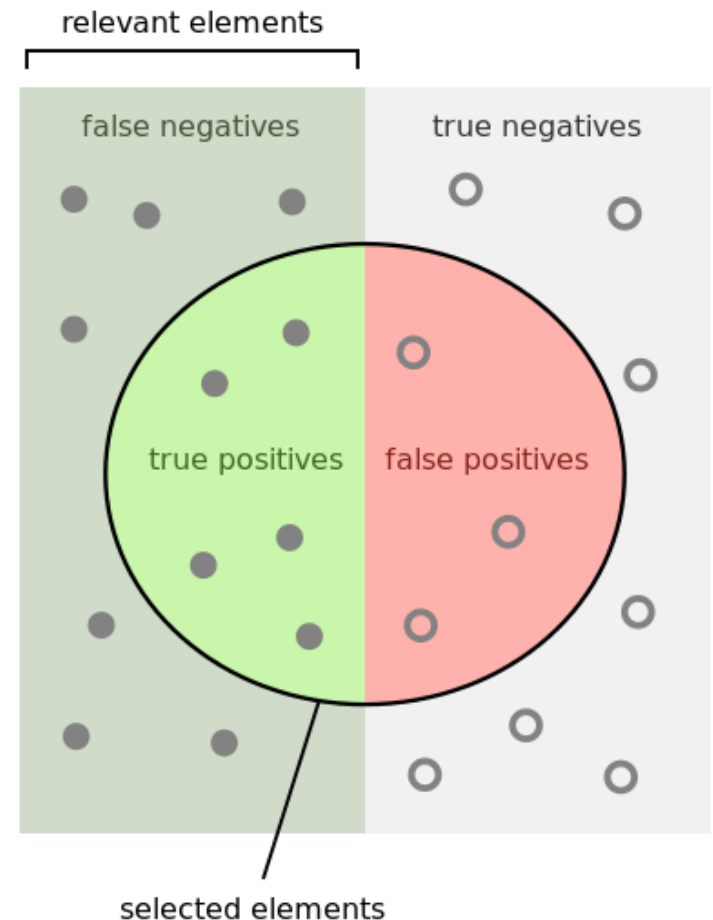
False negative (fn) – incorrect rejection

$$\text{Precision} = \frac{tp}{tp + fp}$$

Precision = #relevant / #returned

$$\text{Recall} = \frac{tp}{tp + fn}$$

Recall = #relevant / #total relevant



How many selected items are relevant?

Precision =  $\frac{\text{green semi-circle}}{\text{green semi-circle} + \text{red semi-circle}}$

How many relevant items are selected?

Recall =  $\frac{\text{green semi-circle}}{\text{green semi-circle} + \text{green rectangle}}$

# Scoring retrieval quality



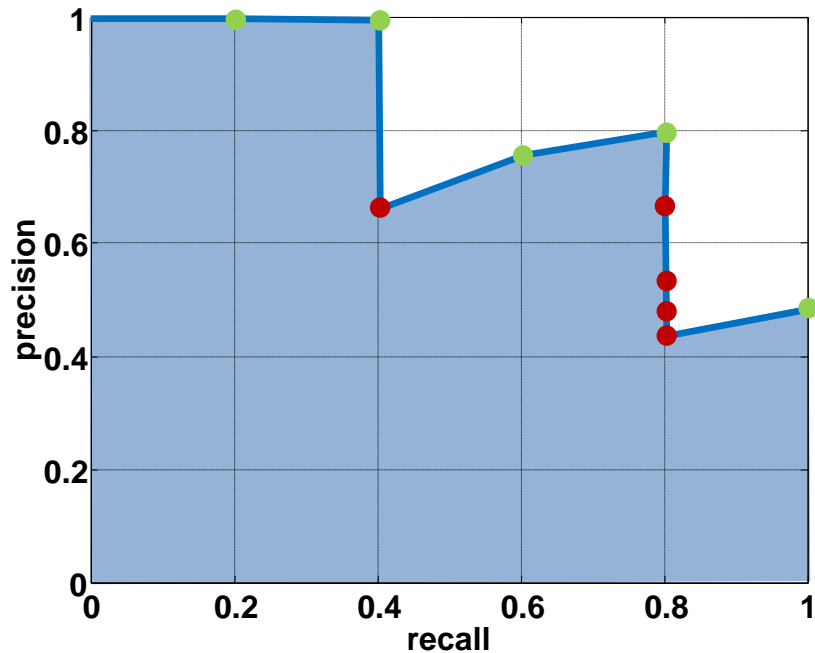
Query

Database size: 10 images

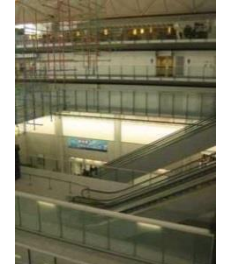
Relevant (total): 5 images

precision =  $\# \text{relevant} / \# \text{returned}$

recall =  $\# \text{relevant} / \# \text{total relevant}$



Results (ordered):





# What else can we borrow from text retrieval?

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 Natl Seashore; 173
Abbreviations,	Cannon Creek Airport; 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 Strs; 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	Choctawhatchee, 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	Flt 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 Mallmen; 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
	Dade Battlefield; 140
	Dade, Maj. Francis; 139-140,161
	Dania Beach Hurricane; 184
	Driving Lanes; 85
	Duval County; 163
	Eau Gallie; 175
	Edison, Thomas; 152
	Eglin AFB; 116-118
	Eight Reale; 176
	Ellenton; 144-145
	Emanuel Point Wreck; 120
	Emergency Callboxes; 83
	Epiphytes; 142,148,157,159
	Escambia Bay; 119
	Bridge (I-10); 119
	County; 120
	Esterio; 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

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 \$660bn. The surplus will annoy the US because China's deliberate policy is to agree to a yuan is also needed to govern the demand so that the country. China's yuan against the dollar has been permitted it to trade within a narrow band but the US wants the yuan to be allowed to move 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**

# *tf-idf* weighting

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

The diagram illustrates the tf-idf formula with annotations for each variable:

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

Annotations:

- Number of occurrences of word  $i$  in document  $d$  (points to  $n_{id}$ )
- Number of words in document  $d$  (points to  $n_d$ )
- Total number of documents in database (points to  $N$ )
- Number of documents word  $i$  occurs in, in whole database (points to  $n_i$ )



# Query expansion

Use good retrieved results as new inputs.  
Increase recall possibly at the expense of precision.

Example query: ***golf green***

Results:

- How can the grass on the ***greens*** at a ***golf*** course be so perfect?
- For example, a skilled ***golfer*** expects to reach the ***green*** on a par-four hole in ...
- Manufactures and sells synthetic ***golf*** putting ***greens*** and mats.

*Good new queries: grass, golf course, par-four hole, putting, etc.*

Irrelevant result can cause a `topic drift`:

- Volkswagen ***Golf***, 1999, ***Green***, 2000cc, petrol, manual, hatchback, 94000miles, 2.0 GTi, 2 Registered Keepers, HPI Checked, Air-Conditioning, Front and Rear Parking Sensors, ABS, Alarm, Alloy

*Bad new queries: petrol, hatchback, ABS, etc.*

# Query expansion

Results



Query image

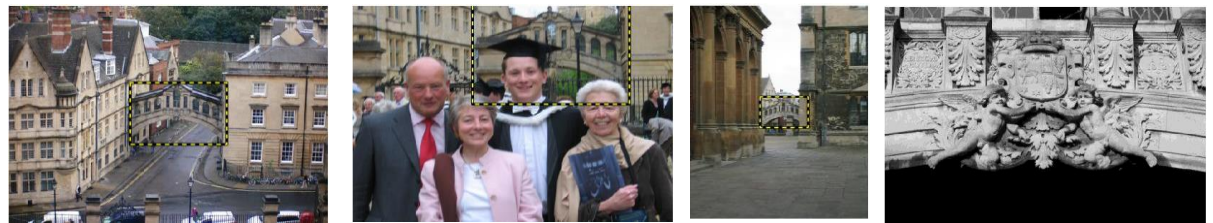


Spatial verification



New query

New results



# Recognition Issues

How to summarize the content of an entire image?  
And gauge overall similarity?

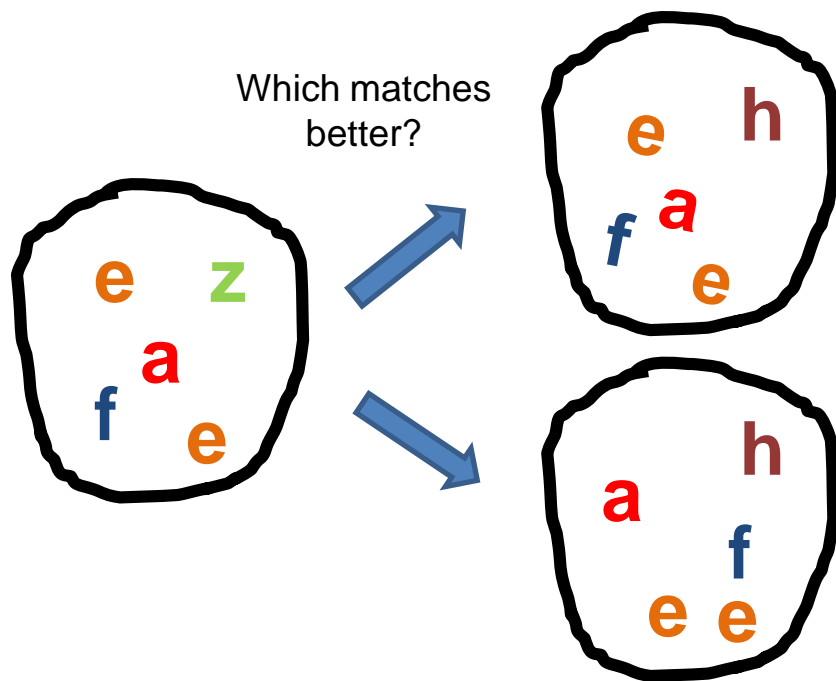
How large should the vocabulary be? How to  
perform quantization efficiently?

How to score the retrieval results?

**How might we add more spatial verification?**

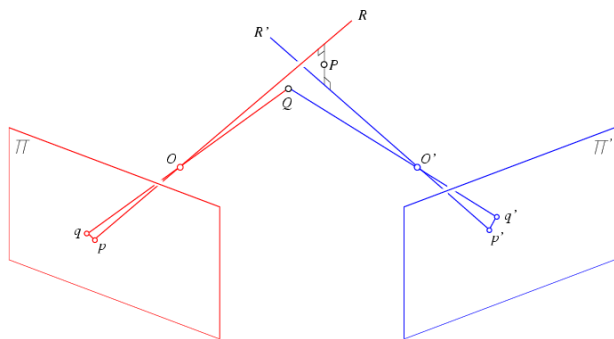
# Can we be more accurate?

So far, we treat each image as containing a “bag of words”, with no spatial information

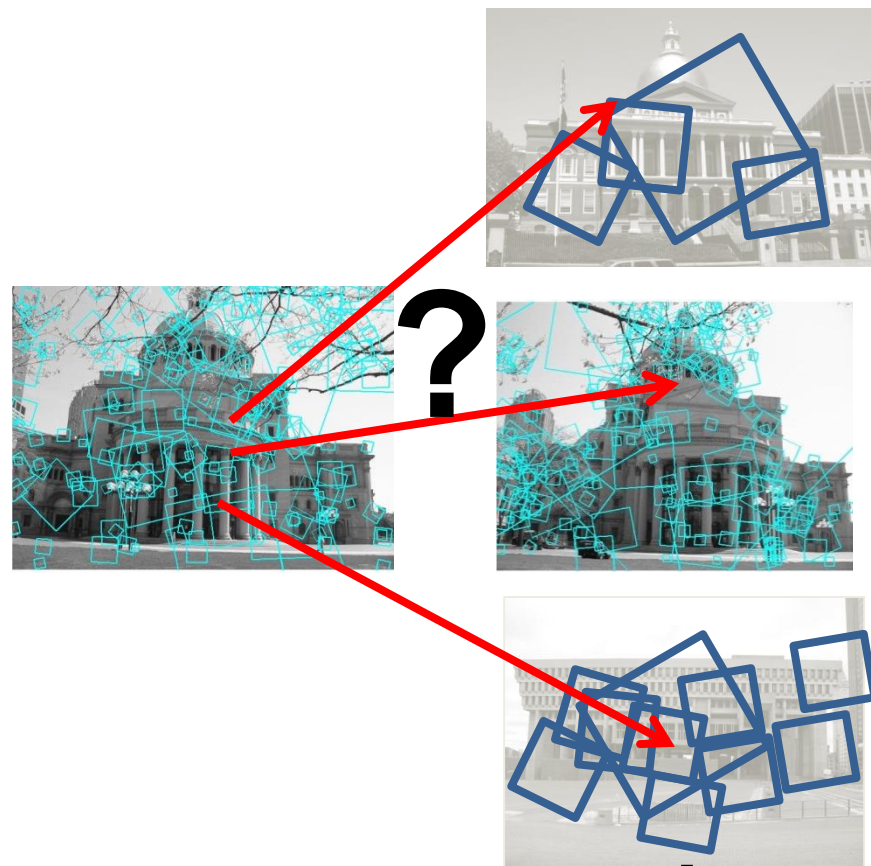


Real objects have consistent geometry

# Multi-view matching



vs



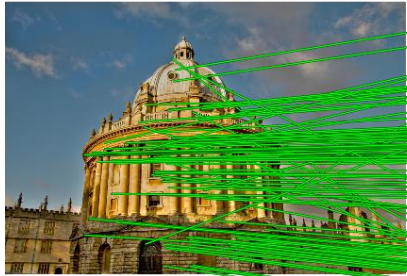
Matching two given  
views for depth

Search for a matching  
view for recognition



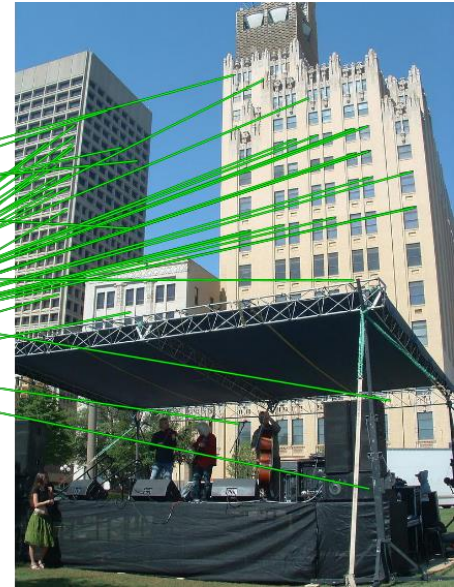
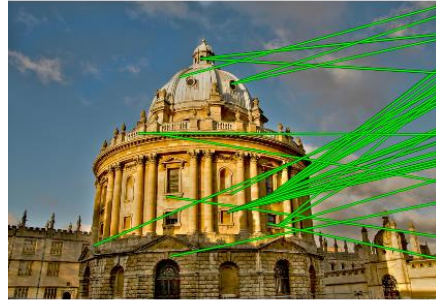
# Spatial Verification

Query



DB image with high BoW  
similarity

Query



DB image with high BoW  
similarity

Both image pairs have many visual words in common.



# Spatial Verification

Query



DB image with high BoW  
similarity

Query



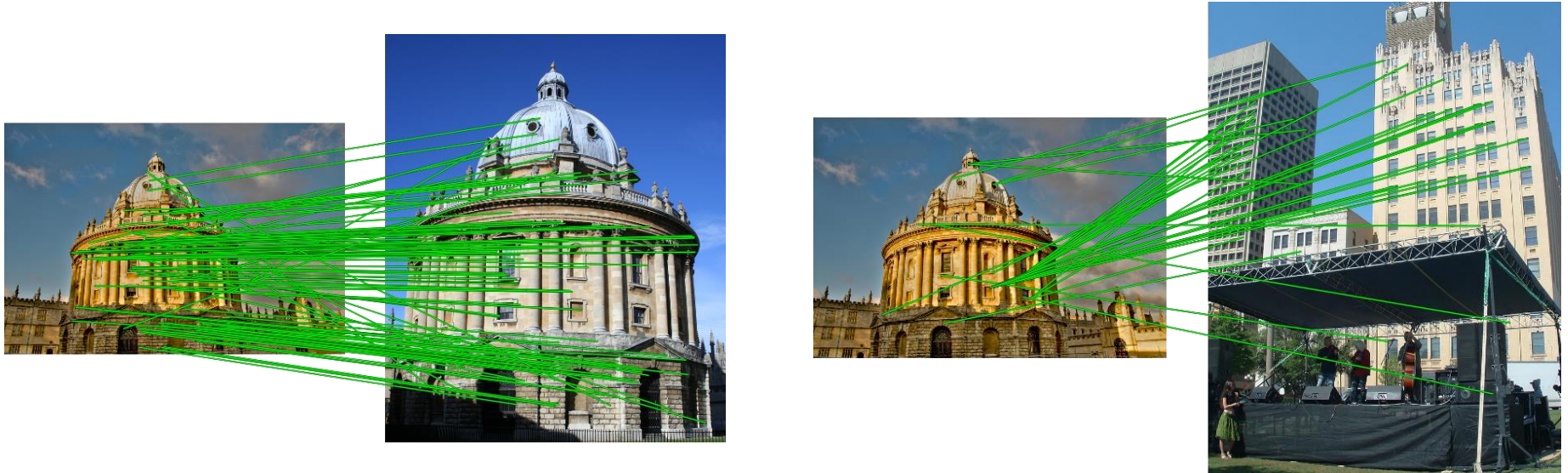
DB image with high BoW  
similarity

Only some of the matches are mutually consistent with real-world geometry imaged by a camera.

# Spatial Verification: two basic strategies

- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., “success” if find a transformation with  $> N$  inlier correspondences
- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

# No verification



# RANSAC verification

Fails to meet threshold  
on # inliers! Good!



# Recognition via alignment

## Pros:

- Effective for reliable features within clutter
- Great for matching specific instances

## Cons:

- Expensive post-process (how long for proj3?!)
- Not suited for category recognition

# Summary

- **Bag of words:** quantize feature space into discrete visual words
  - Summarize image by distribution of words
- **Inverted index:** visual word index for faster query time
- **Evaluation:**
- **Additional spatial verification alignment:**
  - Robust fitting : RANSAC, Generalized Hough Transform
  - We will do this in detail later on in the course

# Lessons from a decade later

For *Category* recognition (project 3)

- Bag of Feature models remained the state of the art until Deep Learning.
- Spatial layout either isn't that important or its too difficult to encode.
- Quantization error is, in fact, the bigger problem. Advanced feature encoding methods address this.
- Bag of feature models are nearly obsolete. At best they seem to be inspiring tweaks to deep models e.g., NetVLAD.



# Lessons from a decade later

For *instance* retrieval (this lecture):

- deep learning is taking over.
- learn better local features (replace SIFT)  
e.g., MatchNet 2015
- learn better image embeddings (replace visual word histograms)  
e.g., Vo and Hays 2016.
- learn spatial verification  
e.g., DeTone, Malisiewicz, and Rabinovich 2016.
- learn a monolithic deep network to recognition all locations  
e.g., Google's PlaNet 2016.