Bag of Visual Words

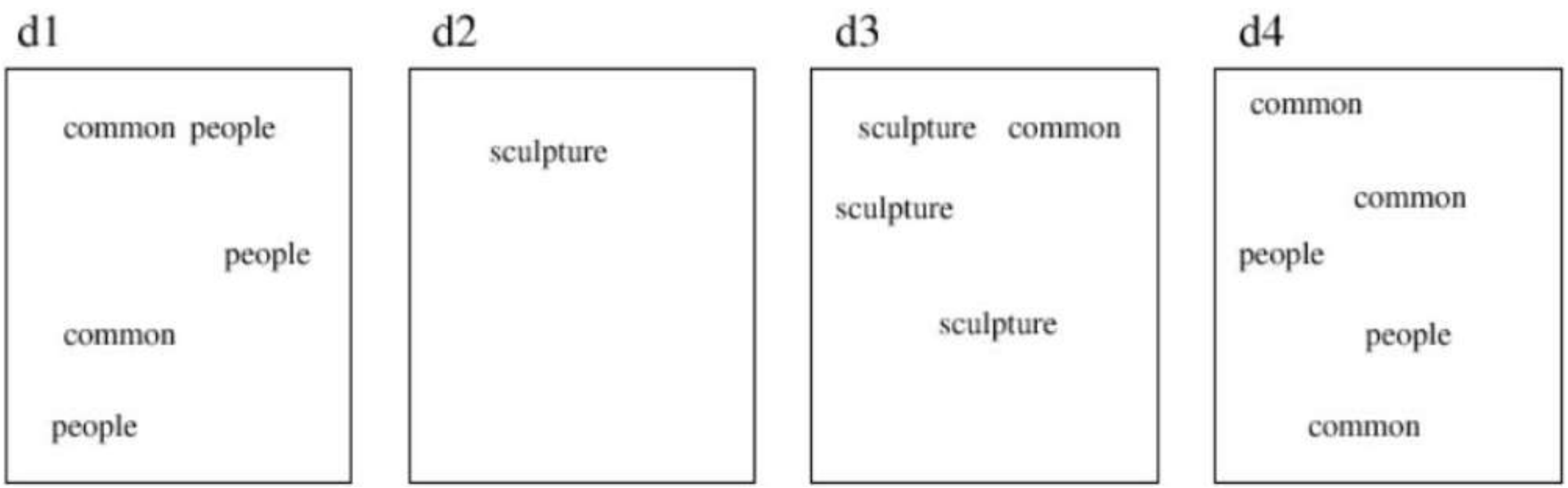
Outline

- Bag-of-features – Origin
- Bag-of-features steps
- Indexing local features

Bag-of-features

Bag-of-features – Origin: bag-of-words (text)

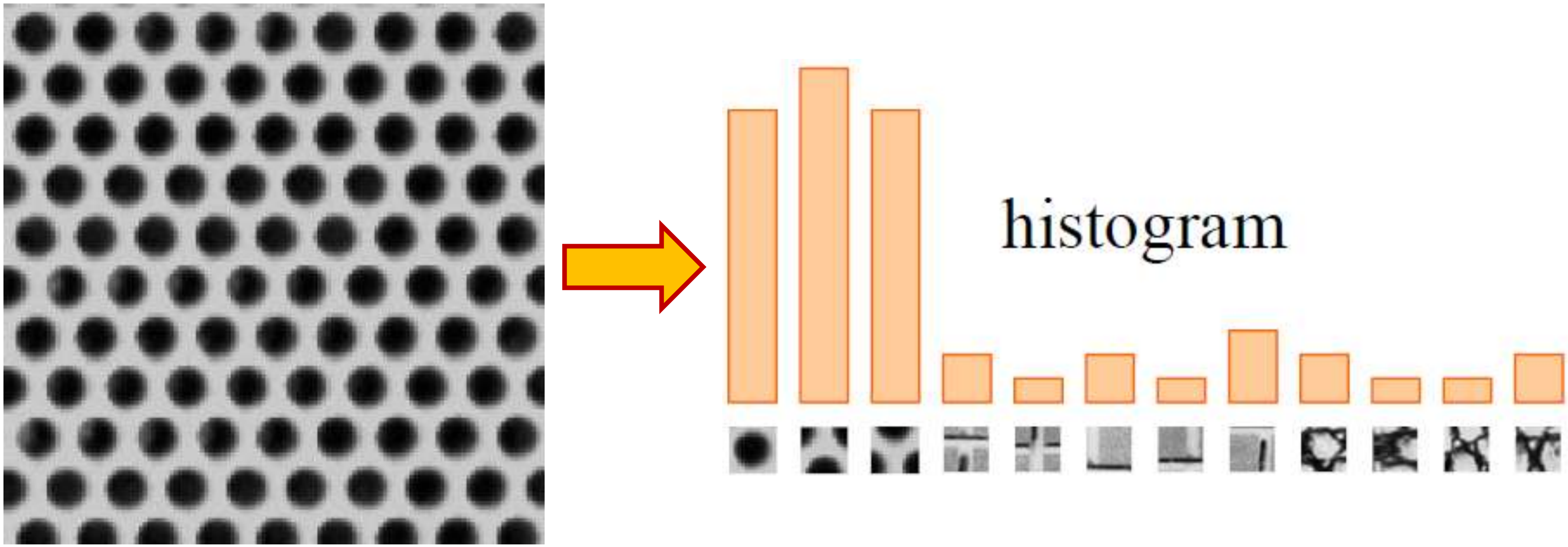
- Order less document representation: frequencies of words from a dictionary
- Classification to determine document categories



Bag-of-words:

Common	2	0	1	3
People	3	0	0	2
Sculpture	0	1	3	0

Bag-of-features – Origin: texture recognition

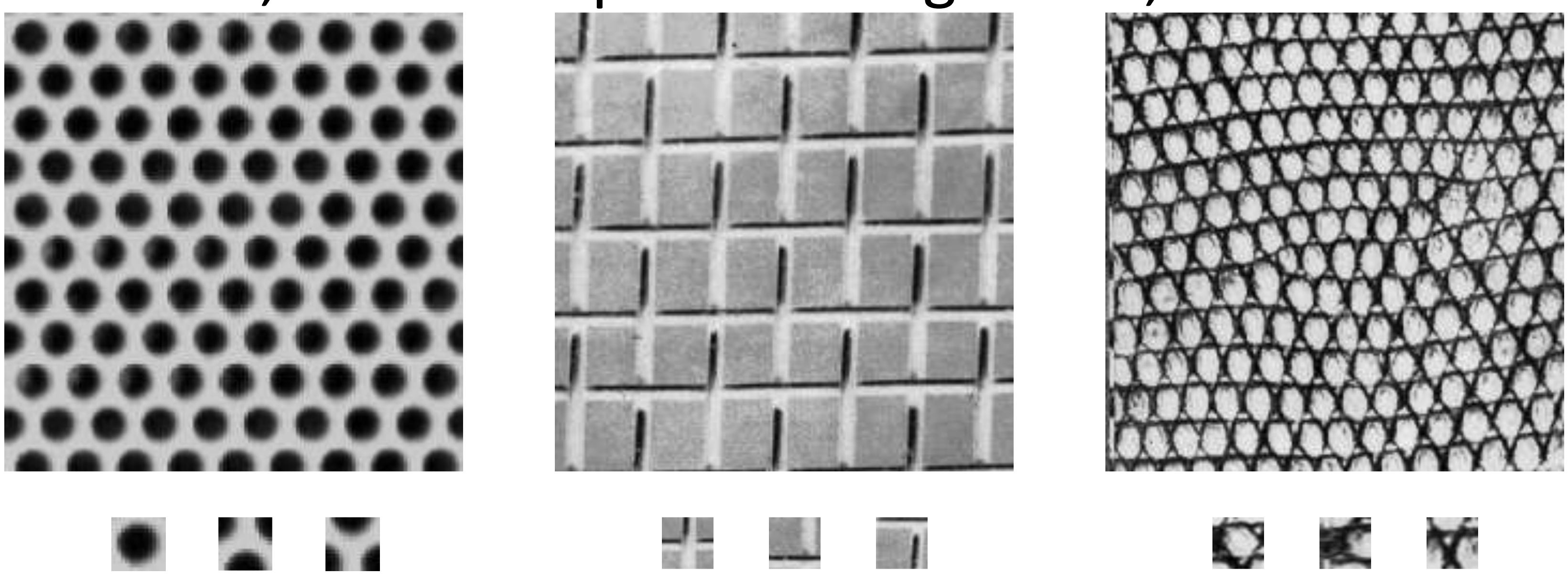


- Texture is characterized by the repetition of basic elements or *textons*

Bags of features

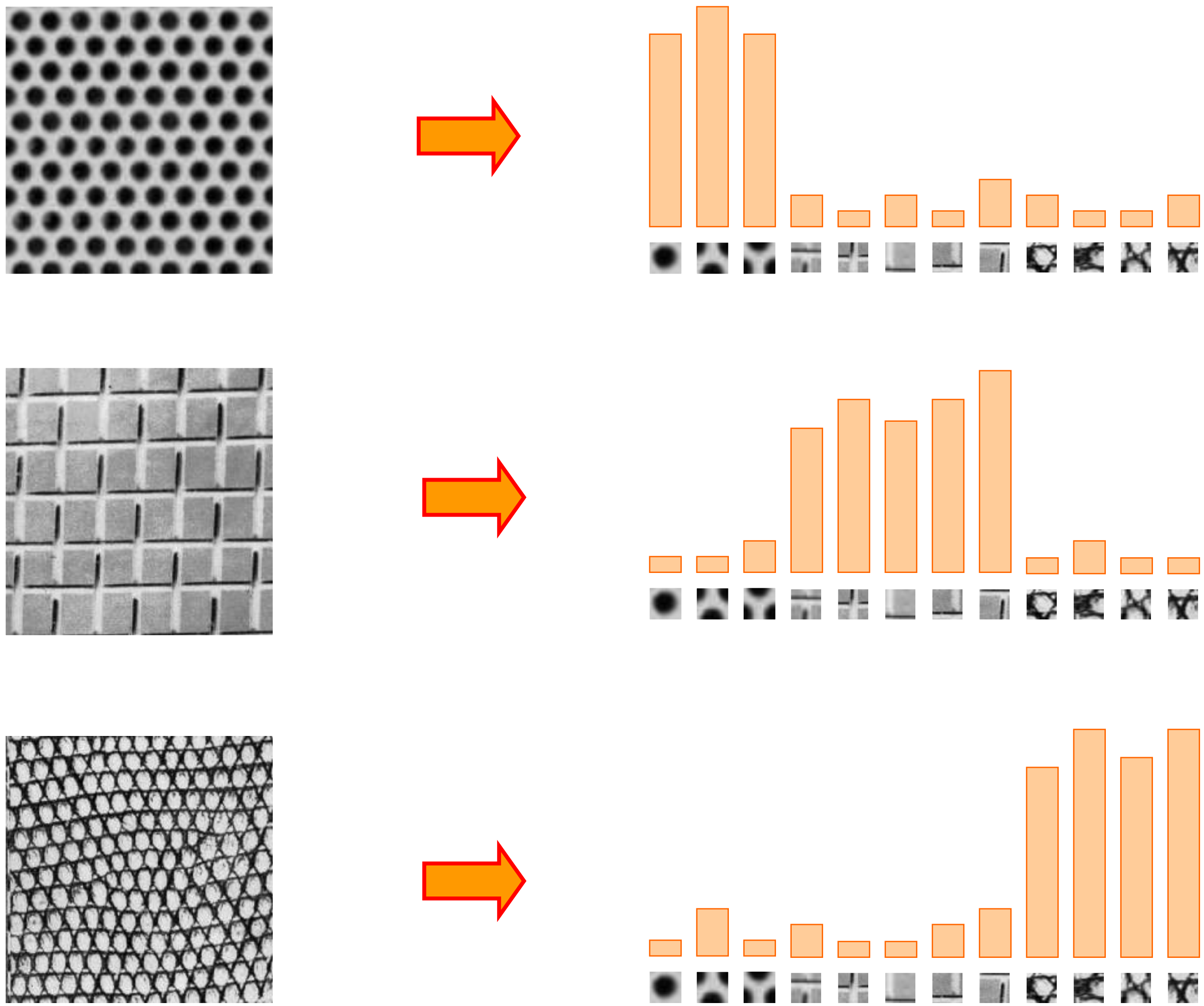
Origin 1: Texture recognition

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



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

Origin 1: Texture recognition



Origin 2: Bag-of-words models

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

Origin 2: Bag-of-words models

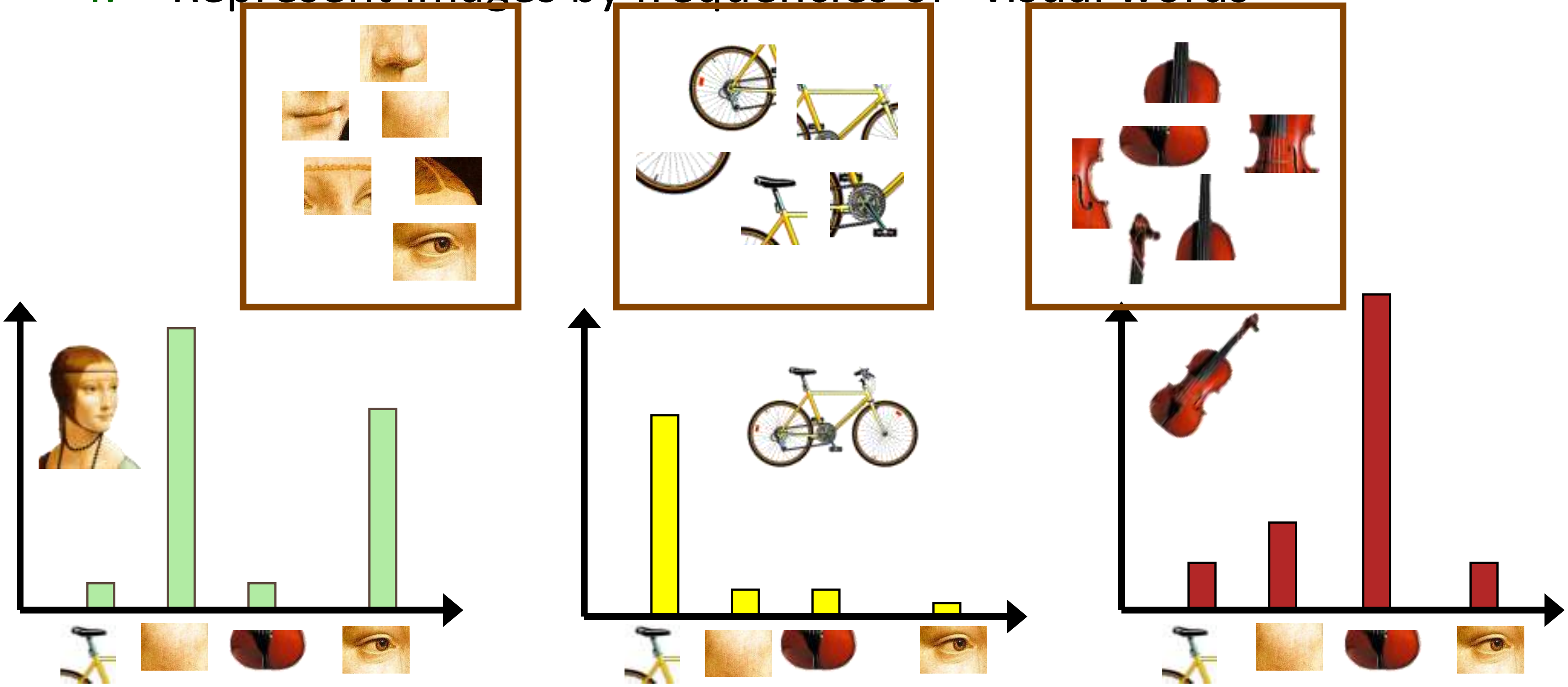
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Outline

- Bag-of-features – Origin
- Bag-of-features steps
- Indexing local features

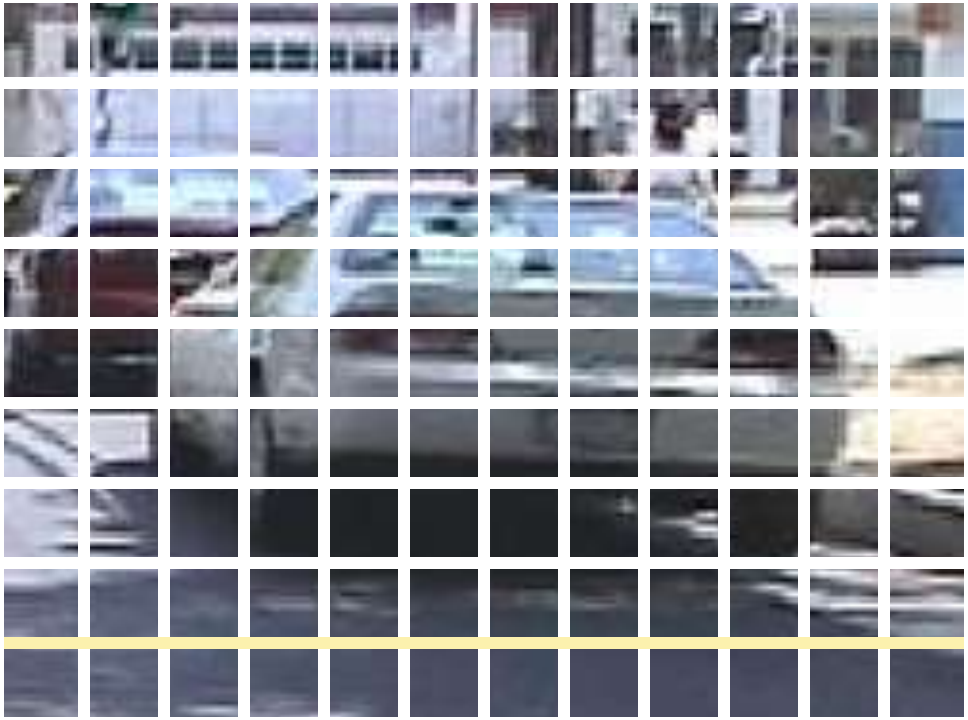
Bag-of-features steps

1. Extract local features
2. Learn "visual vocabulary"
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of "visual words"



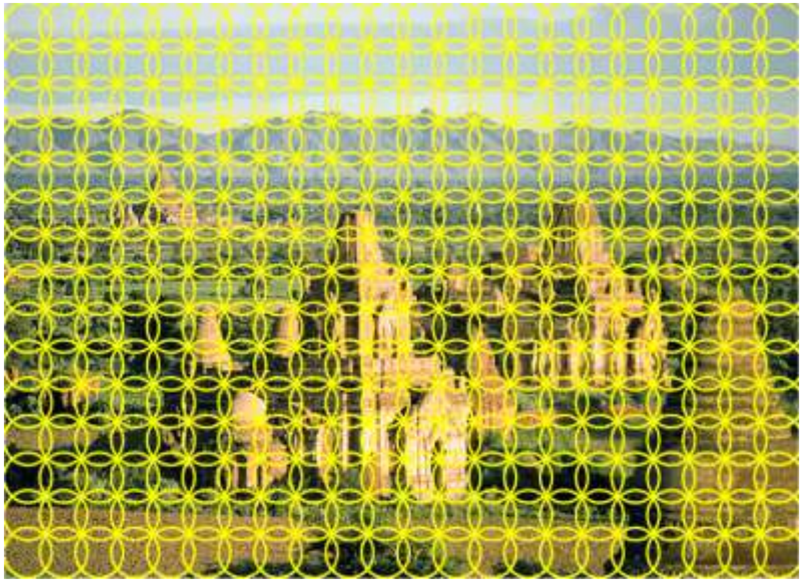
1. Local feature extraction

- Regular grid

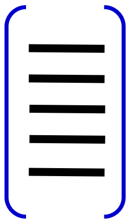


1. Local feature extraction

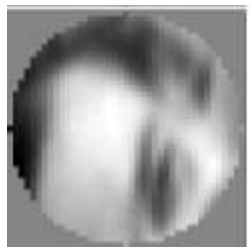
- Regular grid or interest regions



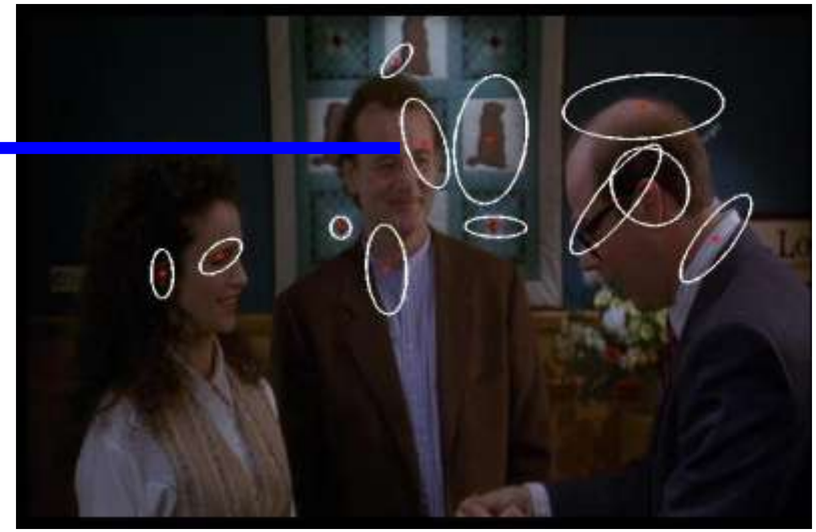
1. Local feature extraction



Compute descriptor



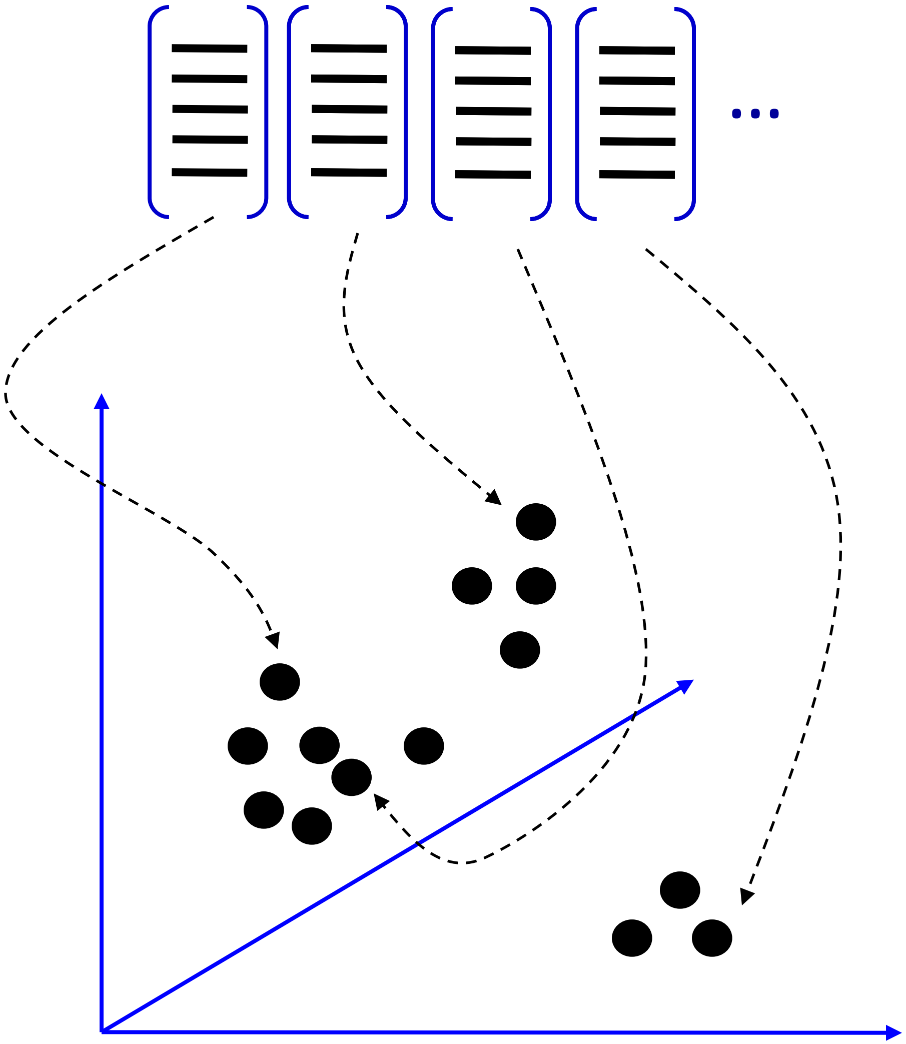
Normalize patch



Detect patches

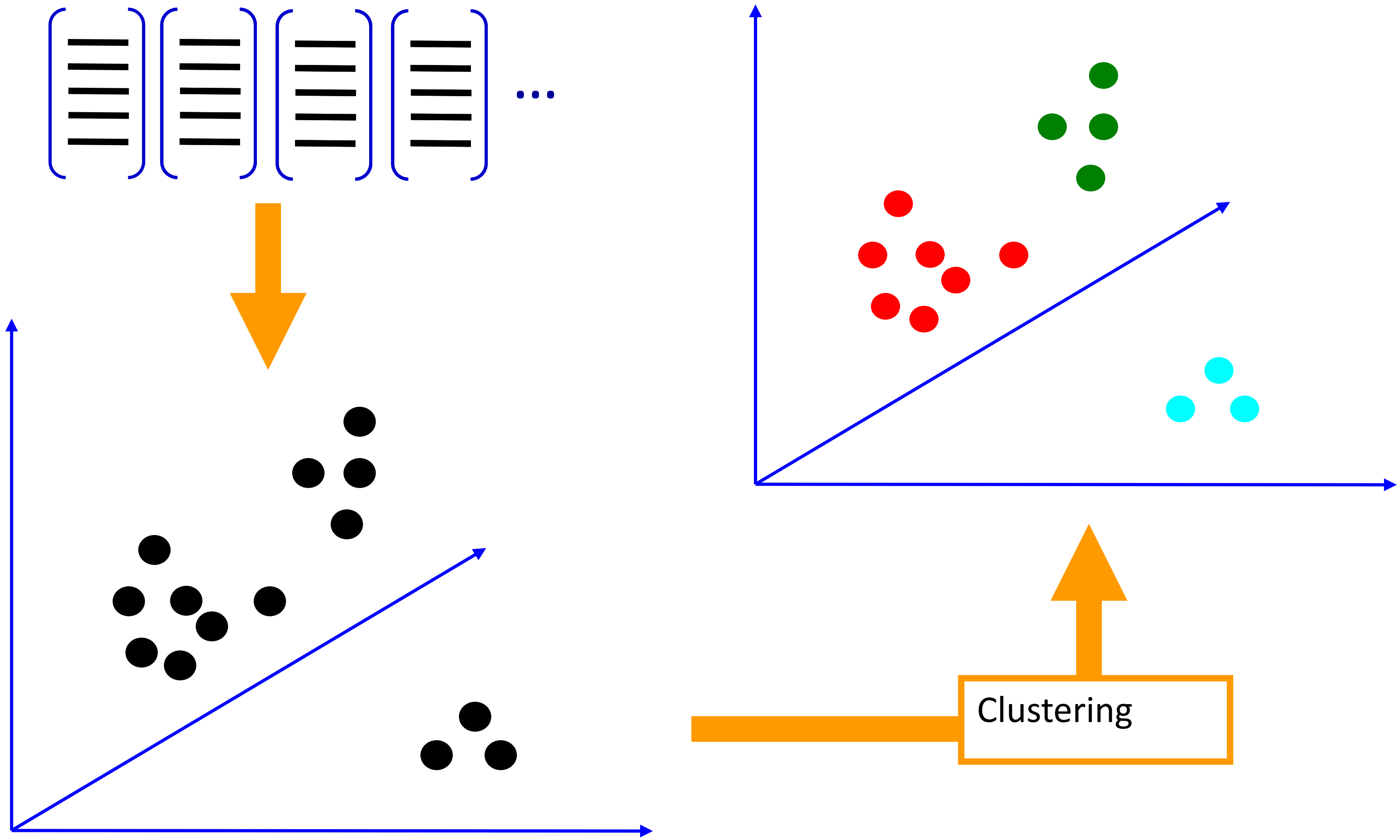
Slide credit: Josef Sivic

2. Learning the visual vocabulary



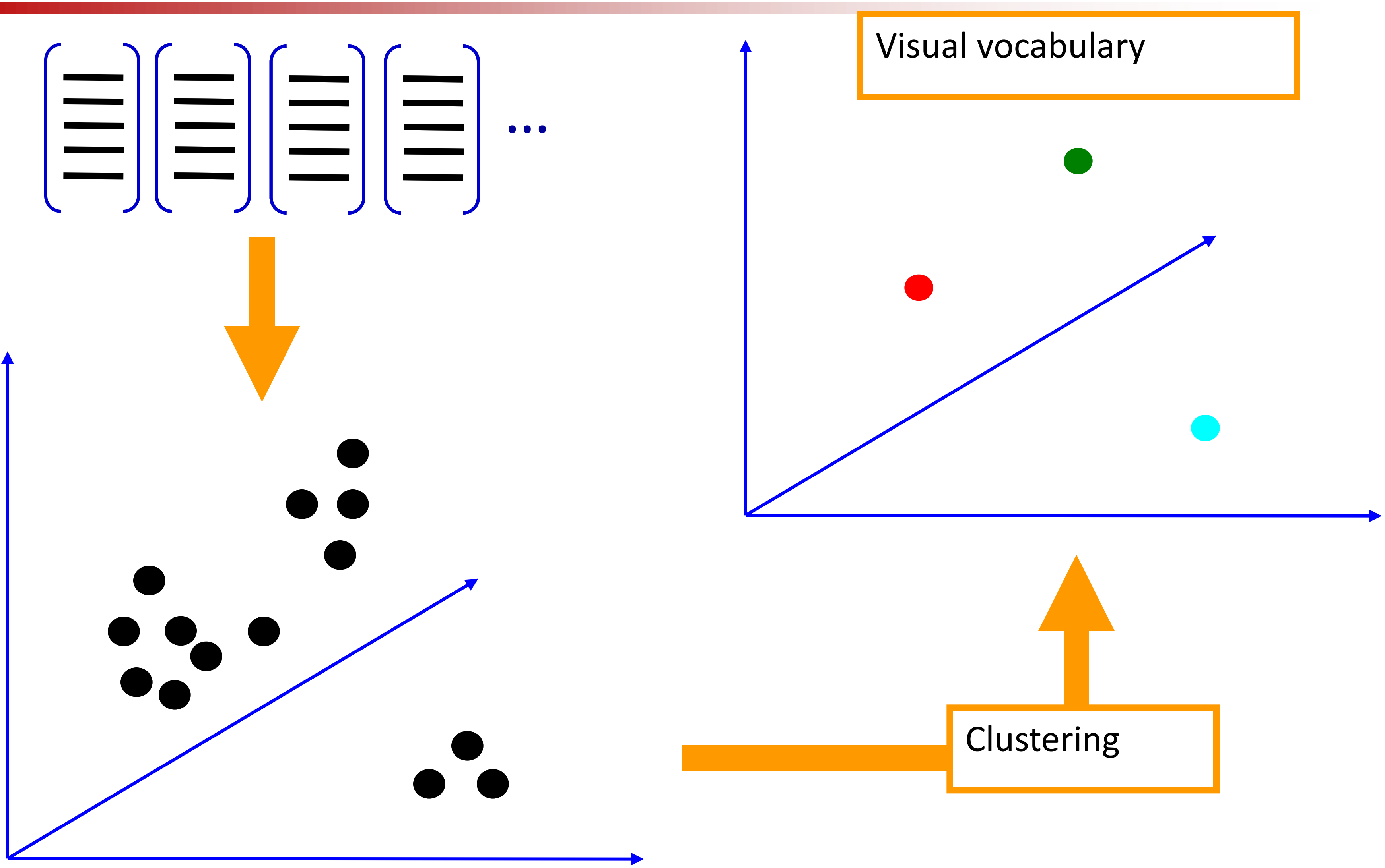
Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

Review: K-means clustering

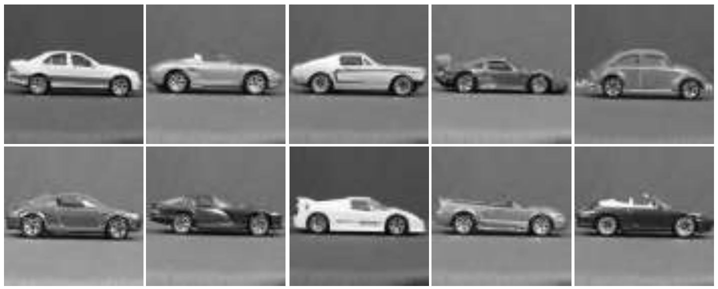
- Want to minimize sum of squared Euclidean distances between features $\mathbf{x}_i$ and their nearest cluster centers $\mathbf{m}_k$

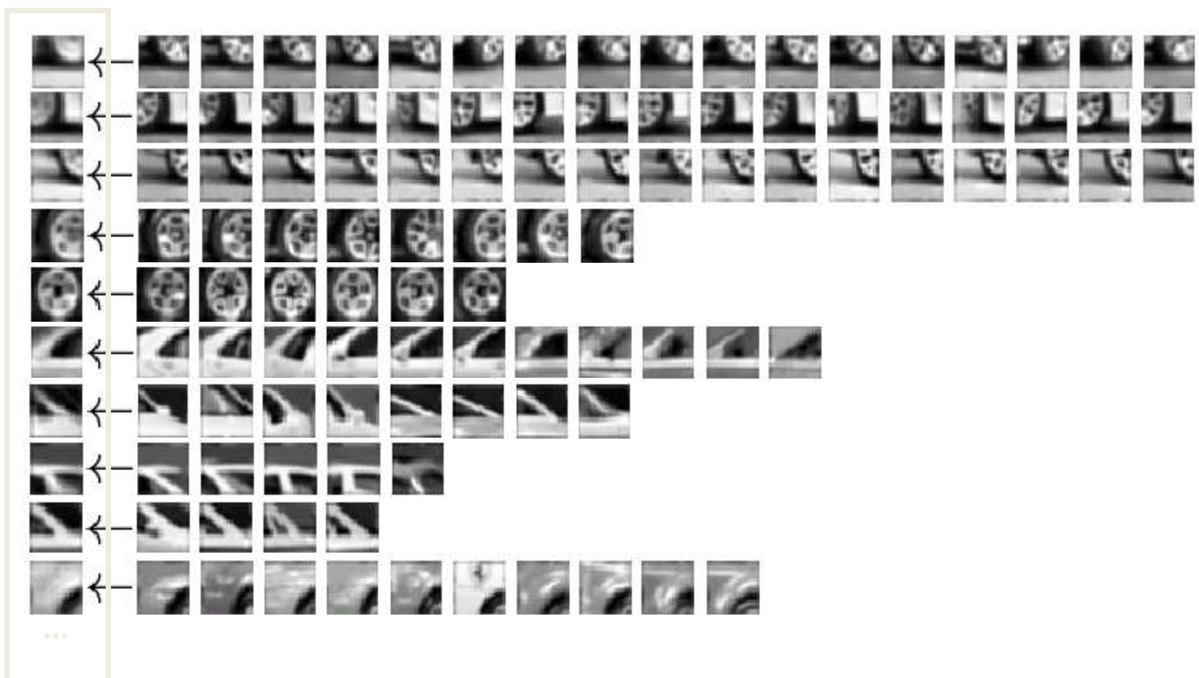
$$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$

Algorithm:

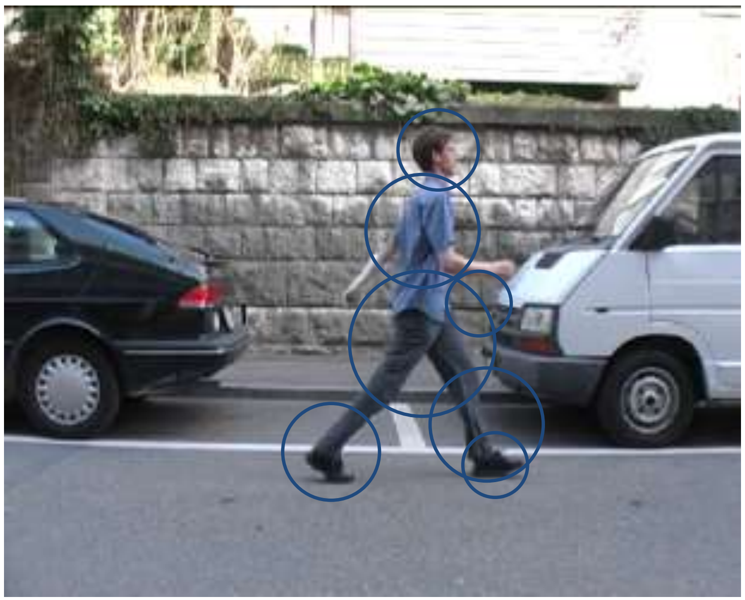
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each feature to the nearest center
  - Recompute each cluster center as the mean of all features assigned to it

Example codebook



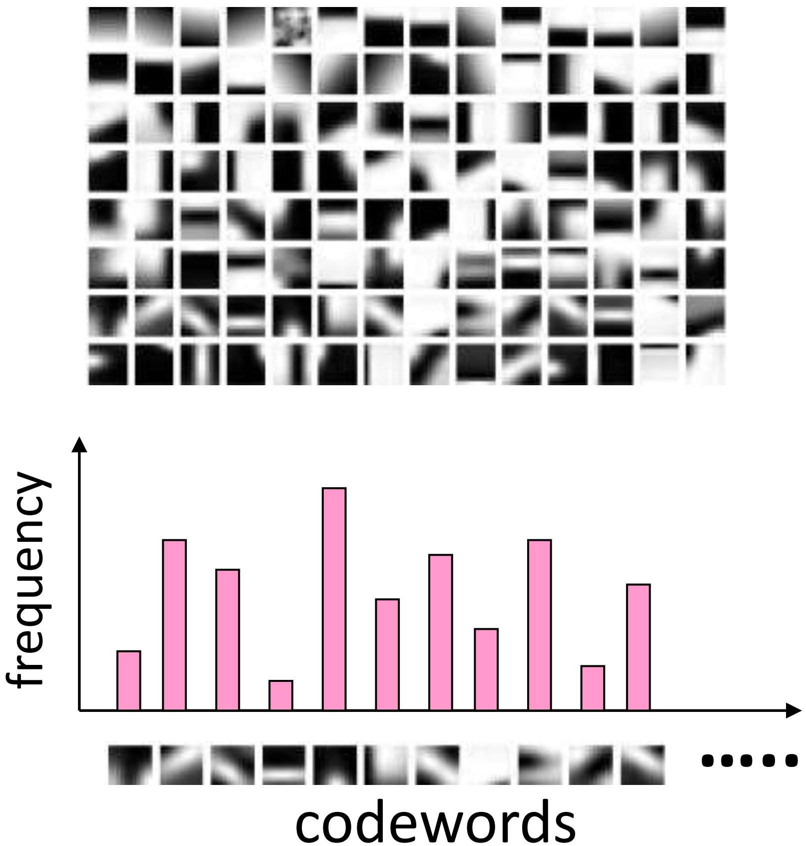


Another codebook

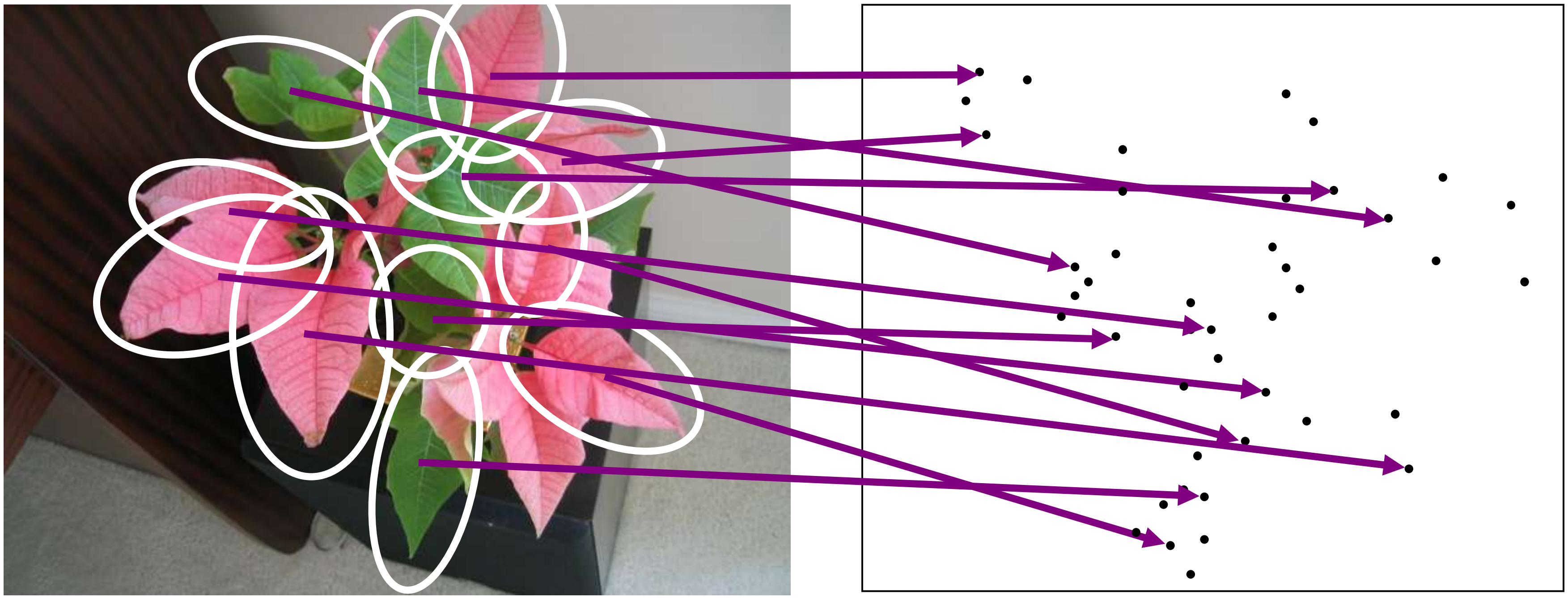


Bag-of-words

visual vocabulary

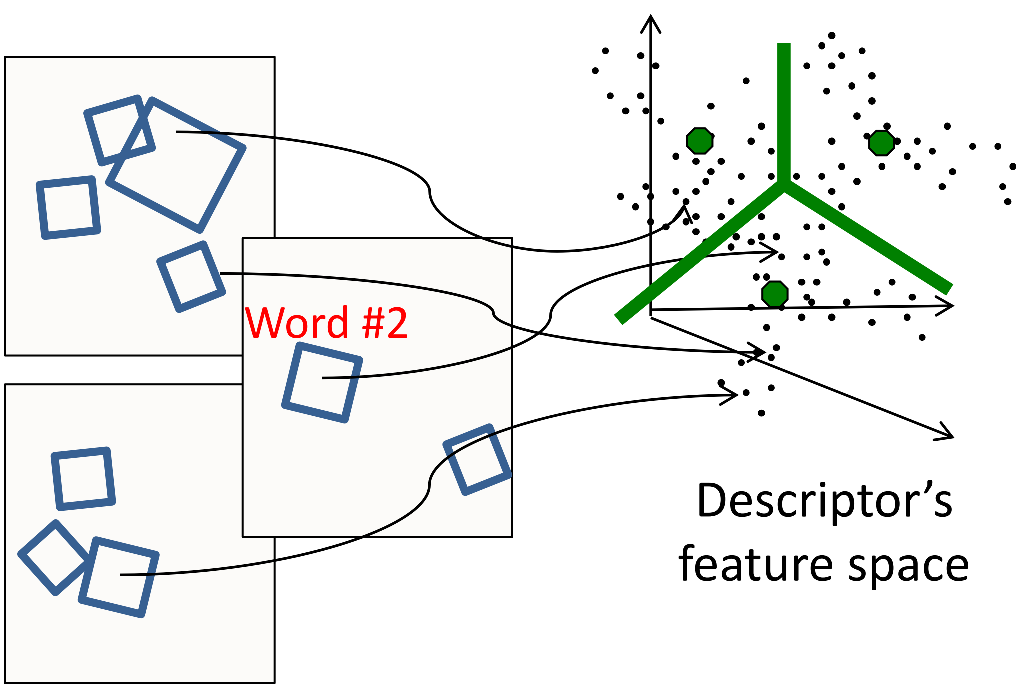


Visual words



Visual words (quantizing the feature space)

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

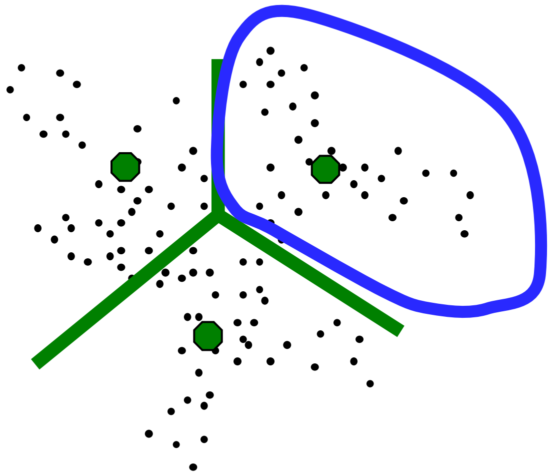


- Quantize via clustering, let cluster centers be the prototype "words"

- Determine which word to assign to each new image region by finding the closest cluster center.

Visual words

- Each group of patches belongs to the same visual word!



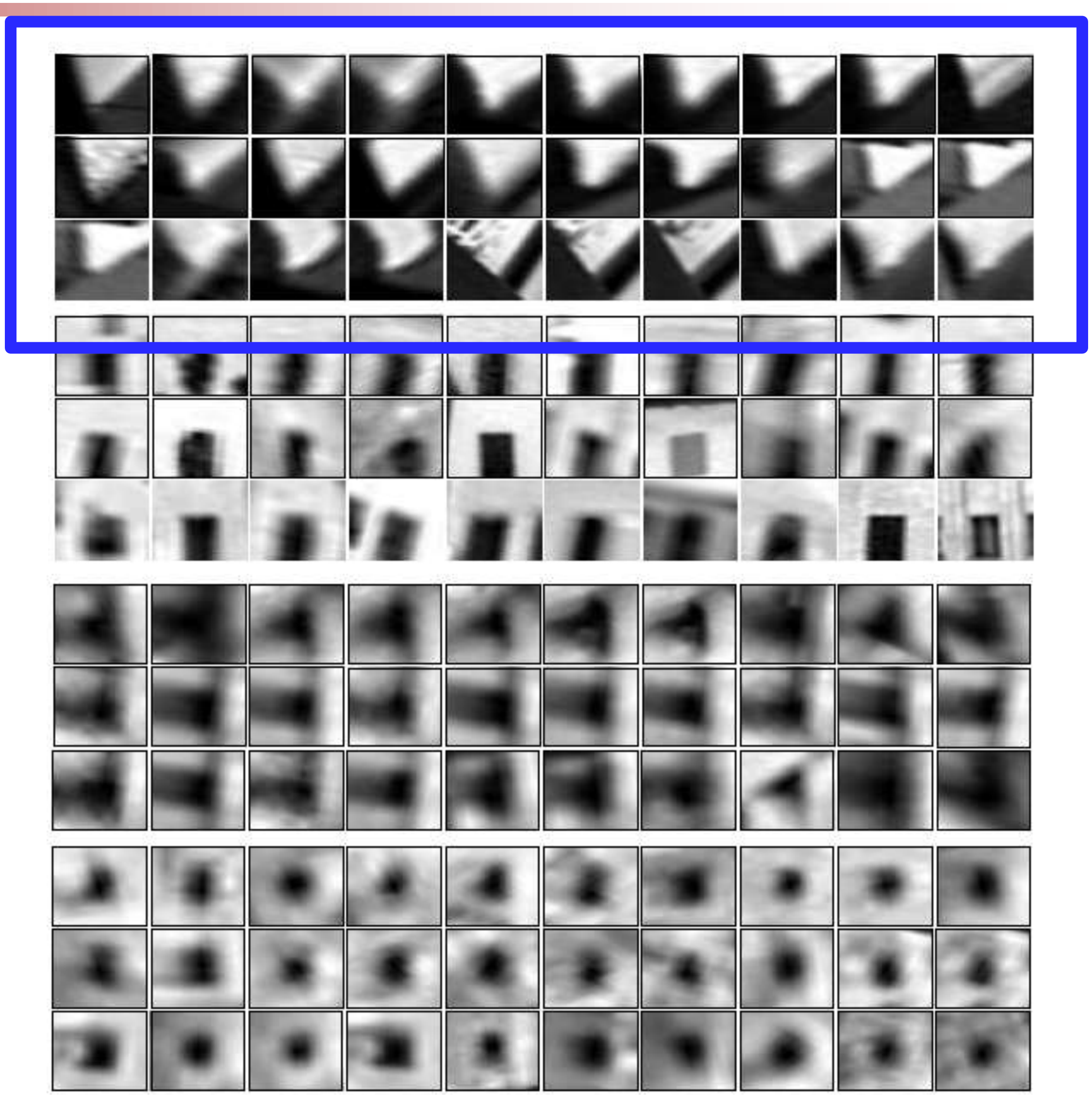


Figure from Sivic & Zisserman, ICCV 2003

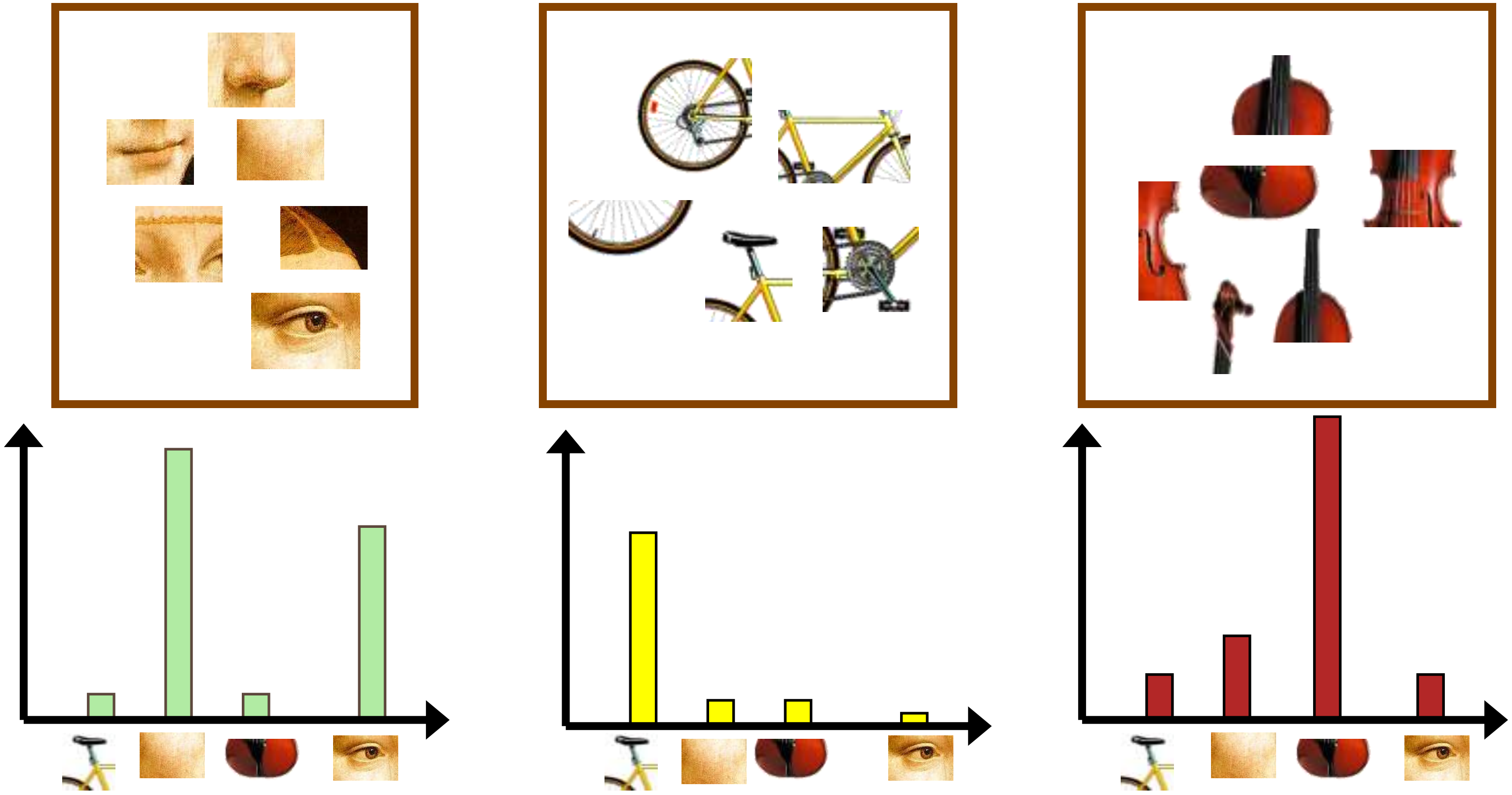
Visual vocabulary formation

Issues:

- Sampling strategy: where to extract features? Fixed locations or interest points?
- Clustering / quantization algorithm
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words
- Weight of each word?

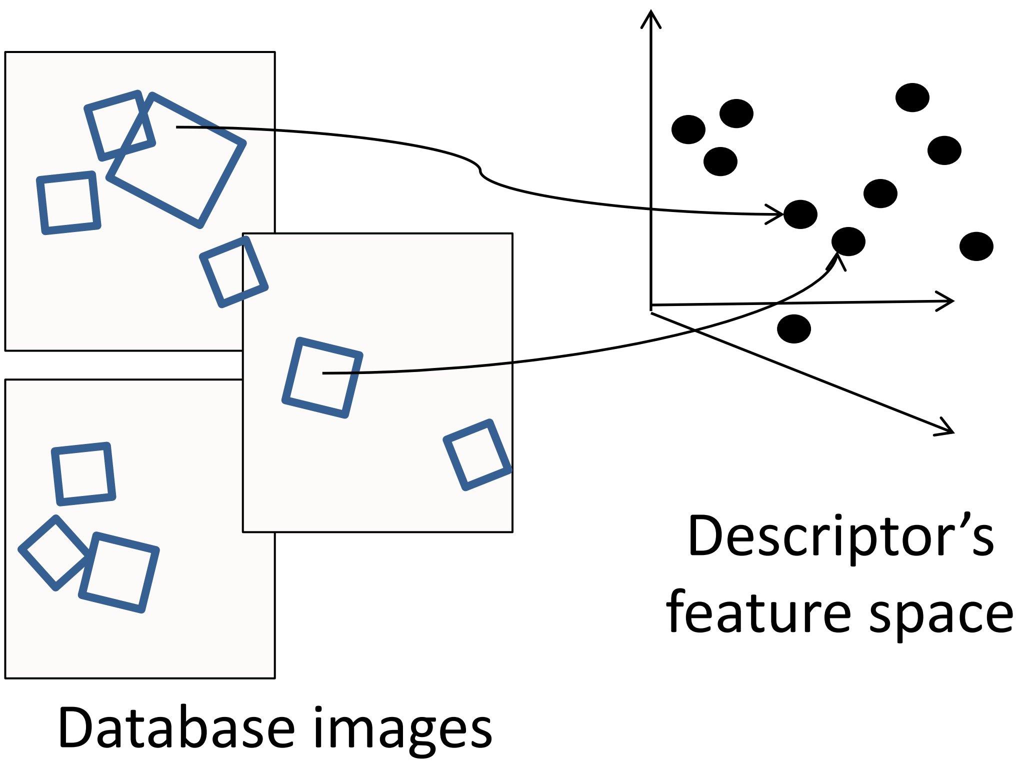
Bag-of-features steps

1. Extract local features
2. Learn "visual vocabulary"
3. **Quantize local features using visual vocabulary**
4. **Represent images by frequencies of "visual words"**



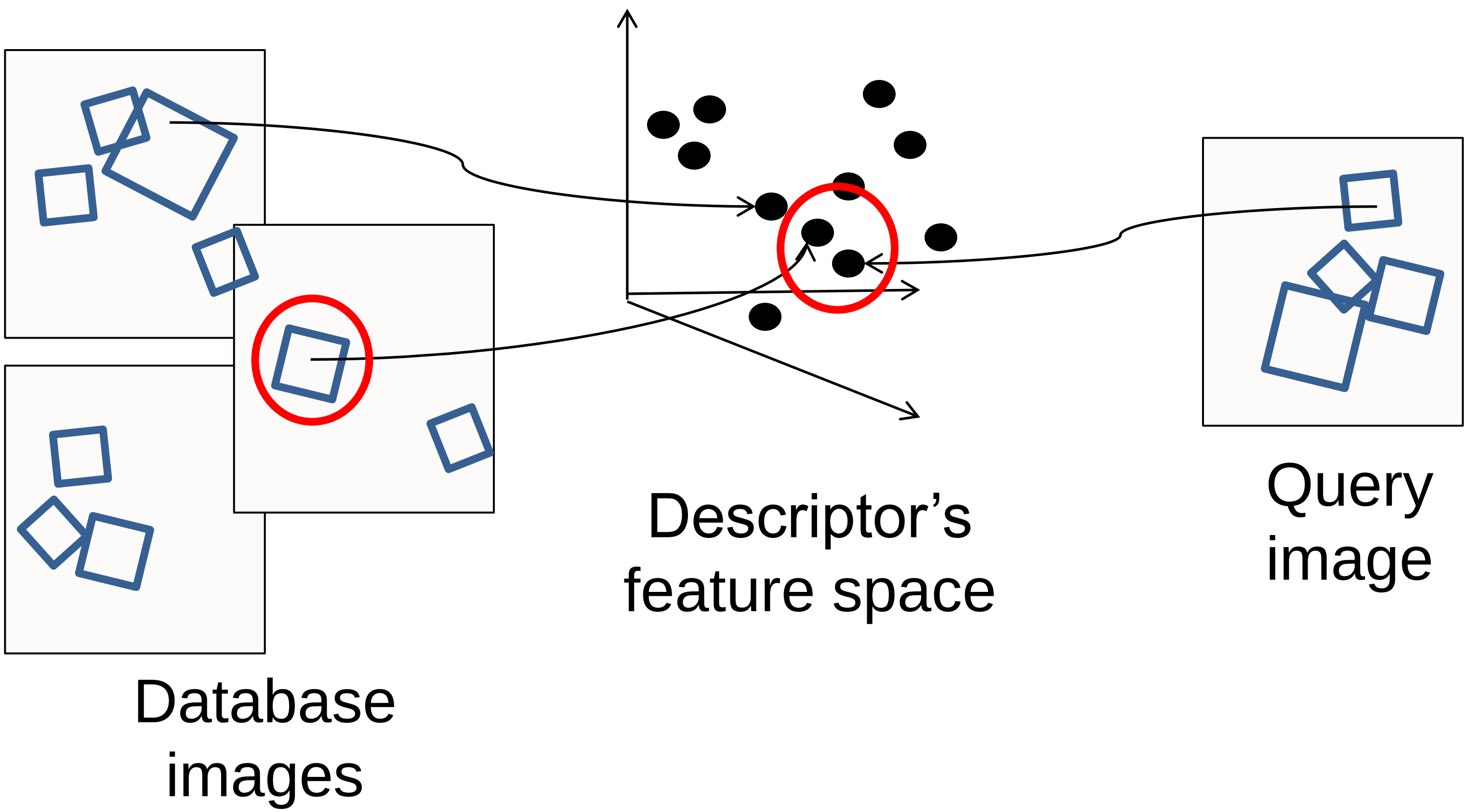
Indexing local features

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



Indexing local features

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



Indexing local features: **inverted file**

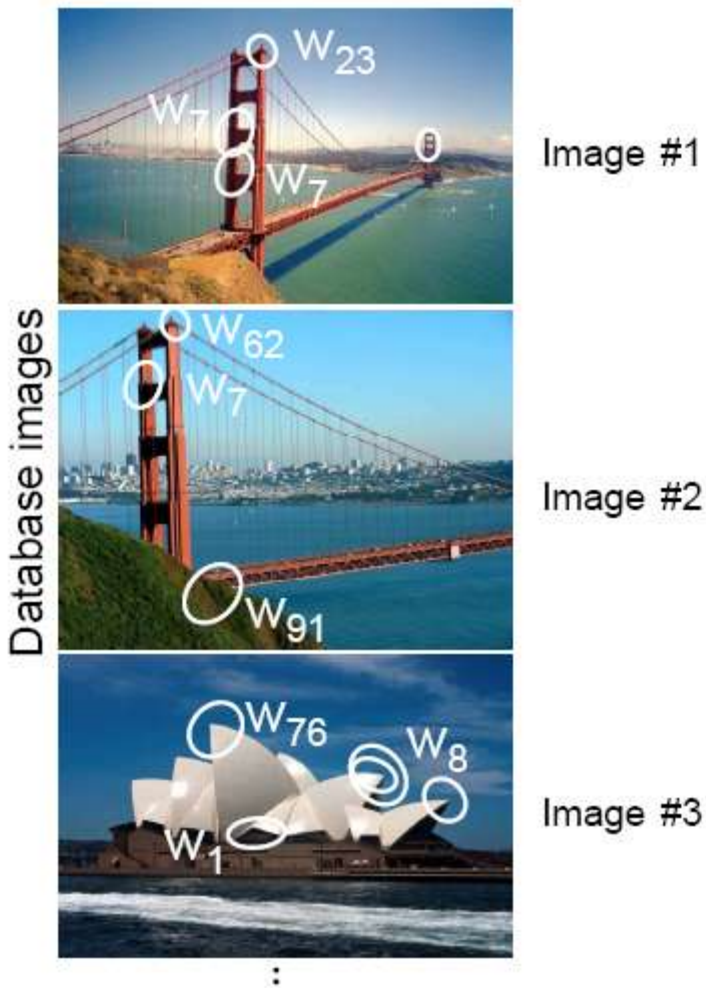
Index

"Along I-75," From Detroit to
Florida; ***inside back cover***
"Drive I-95," From Boston to
Florida; ***inside back cover***
1929 Spanish Trail Roadway; 101-102,104
511 Traffic Information; 83
A1A (Barrier Isl) - I-95 Access; 86
AAA (and CAA); 83
AAA National Office; 88
Abbreviations,
Colored 25 mile Maps; cover
Exit Services; 196
Travelogue; 85
Africa; 177
Agricultural Inspection Stns; 126
Ah-Tah-Thi-Ki Museum; 160
Air Conditioning, First; 112
Alabama; 124
Alachua; 132
County; 131
Alafia River; 143
Alapaha, Name; 126
Alfred B Maclay Gardens; 106
Alligator Alley; 154-155
Alligator Farm, St Augustine; 169
Alligator Hole (definition); 157
Alligator, Buddy; 155
Alligators; 100,135,138,147,156
Anastasia Island; 170
Anhaica; 108-109,146
Apalachicola River; 112
Appleton Mus of Art; 136
Aquifer; 102
Arabian Nights; 94
Art Museum, Ringling; 147
Aruba Beach Cafe; 183
Aucilla River Project; 106
Babcock-Web WMA; 151
Bahia Mar Marina; 184
Baker County; 99
Barefoot Mailmen; 182
Barge Canal; 137
Bee Line Expy; 80
Belz Outlet Mall; 89
Bernard Castro; 136
Big "I"; 165
Big Cypress; 155,158
Big Foot Monster; 105
Billie Swamp Safari; 160
Blackwater River SP; 117
Blue Angels
A4-C Skyhawk; 117
Atrium; 121
Blue Springs SP; 87
Blue Star Memorial Highway; 125
Boca Ciega; 189
Boca Grande; 150
Boca Raton; 182
Bonnie Blue Flag; 124
Boyd Hill Nature Trail; 188
Bradenton; 145-147
Breakers, The, Palm Beach; 181
Brickell Point, Miami; 185
Britton Hill; 116
Brogan Museum; 107
Bromeliads (see Epiphytes)
Broward County; 159,181
Broward, Gov. Napoleon; 156
Bulow Plantation Ruins; 171
Bush, Gov. Jeb; 100
Butterfly Center, McGuire; 134
CAA (see AAA)
CCC, The; 111,113,115,135,142
Ca d'Zan; 147
Caloosahatchee River; 152
Name; 150
Canaveral Natnl Seashore; 173
Cannon Creek Airpark; 130
Canopy Road; 106,169
Cape Canaveral; 174
Castillo San Marcos; 169
Cave Diving; 131
Cayo Costa, Name; 150
Celebration; 93
Charlotte County; 149
Charlotte Harbor; 150
Chautauqua; 116
Chipley; 114
Name; 115
Choctawatchee, Name; 115
Circus Museum, Ringling; 147
Citrus; 88,97,130,136,140,180
CityPlace, W Palm Beach; 180
City Maps,
Ft Lauderdale Expwys; 194-195
Jacksonville; 163
Kissimmee Expwys; 192-193
Miami Expressways; 194-195
Orlando Expressways; 192-193
Pensacola; 26
Tallahassee; 191
Tampa-St. Petersburg; 63
St. Augsutine; 191
Civil War; 100,108,127,138,141
Clearwater Marine Aquarium; 187
Collier County; 154
Collier, Barron; 152
Colonial Spanish Quarters; 168
Columbia County; 101,128
Coquina Building Material; 165
Corkscrew Swamp, Name; 154
Cowboys; 95
Crab Trap II; 144
Cracker, Florida; 88,95,132
Crosstown Expy; 11,35,98,143
Cuban Bread; 184
Dade Battlefield; 140
Dade, Maj. Francis; 139-140,161
Dania Beach Hurricane; 184
Daniel Boone, Florida Walk; 117
Daytona Beach; 172-173
De Land; 87
De Soto, Hernando,
Anhaica; 108-109,146
County; 149
Explorer; 146
Landing; 146
Napitaca; 103
National Park; 147
Tallahassee; 108
DeFuniak Springs; 116
Name; 115
Delnor-Wiggins Pass SP; 155
Denoël Cafe, St Augustine; 169
Devil's Millhopper; 132
Dickson Azalea Park; 89
Dinosaur World; 98
Discovery Cove; 90
Dixie Highway; 186
Don Garlits Drag Racing Mus; 138
Douglas, Marjory Stoneham ; 159
Driving Lanes; 85
Duval County; 163
Eau Gallie; 175
Edison, Thomas; 152
Eglin AFB; 116-118
Eight Reale; 176
Ellenton; 144-145
Emanuel Point Wreck; 120
Emergency Callboxes; 83
Epiphytes; 142,148,157,159
Escambia Bay; 119
Bridge (I-10); 119
County; 120
Estero; 153
Everglade,90,95,139-140,154-160
Draining of; 156,181
Wildlife MA; 160
Wonder Gardens; 154
Falling Waters SP; 115
Fantasy of Flight; 95
Fayer Dykes SP; 171
Fires, Forest; 166
Fires, Prescribed ; 148
Fisherman's Village; 151
Flagler County; 171
Flagler, Henry; 97,165,167,171
Florida Aquarium; 186
Florida,
12,000 years ago; 187
Cavern SP; 114
Map of all Expressways; 2-3
Mus of Natural History; 134
National Cemetery ; 141
Part of Africa; 177
Platform; 187
Sheriff's Boys Camp; 126
Sports Hall of Fame; 130
Sun 'n Fun Museum; 97
Supreme Court; 107
Florida's Turnpike (FTP), 178,189
25 mile Strip Maps; 66
Administration; 189
Coin System; 190
Exit Services; 189
HEFT; 76,161,190
History; 189
Names; 189
Service Plazas; 190
Spur SR91; 76
Ticket System; 190
Toll Plazas; 190
Ford, Henry; 152
Fort Barrancas; 122
Buried Alive; 123
Fort Caroline; 164
Fort Clinch SP; 161
Fort De Soto & Egmont Key; 188
Fort Lauderdale; 161,182-184
Fort Myers; 152-153
Fort Pierce; 177-178
Farmers Market; 178
Fountain of Youth; 170
Frank Lloyd Wright Center; 97
Gadsden County; 110
Gainesville; 99,104,131-135,146
Gamble Plantation; 145
Garden of Eden; 112
Gasparilla, Pirate; 150
Gatorade; 134
Gaylord Palms; 90
Geology;102-103,110,131-132

Indexing local features: Inverted index

- 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.
  - page $\sim$ image
  - word $\sim$ feature
- To use this idea, we'll need to map our features to "visual words".

Inverted file index



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

…

Why the index give us a significant gain in efficiency?

database images are loaded into the index mapping words to image numbers

Inverted file index

New query image

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

...

A query image is matched to database images that share visual words.

tf-idf weighting

- ***t**erm **f**requency – **i**nverse **d**ocument **f**requency*
- Describe the *frequency* of each word within an image, decrease the weights of the words that appear often in the database
  - economic, trade, … $w \nearrow$ discriminative regions
  - the, most, we, … $w \searrow$ common regions

tf-idf weighting

- ***t**erm **f**requency – **i**nverse **d**ocument **f**requency*
- Describe the *frequency* of each word within an image, decrease the weights of the words that appear often in the database

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

Number of words in document *d*

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

Total number of documents in database

Number of documents containing word *i*, in whole database

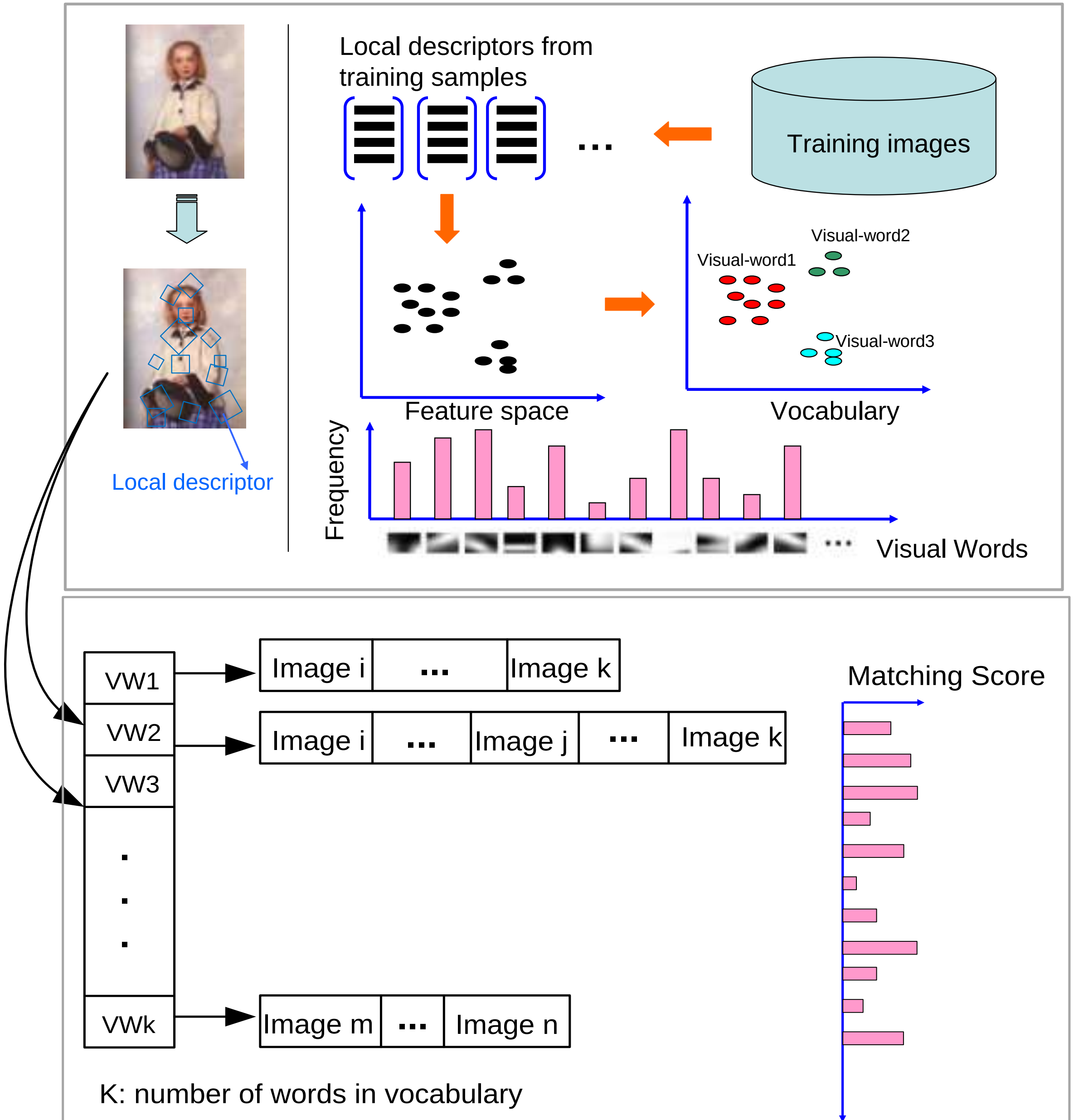
tf-idf weighting: matching

- At match time, each document (or query region) is represented by its tf-idf vector,

  $\boldsymbol{t} = (t_1, \ldots, t_i, \ldots t_m)$.

- The similarity between two documents is measured by the dot product between their corresponding normalized vectors $\hat{\mathbf{t}} = \mathbf{t}/\|\mathbf{t}\|$

Bag-of-Words + Inverted file



Bag-of-words representation

Inverted file

Visual vocabularies: Details

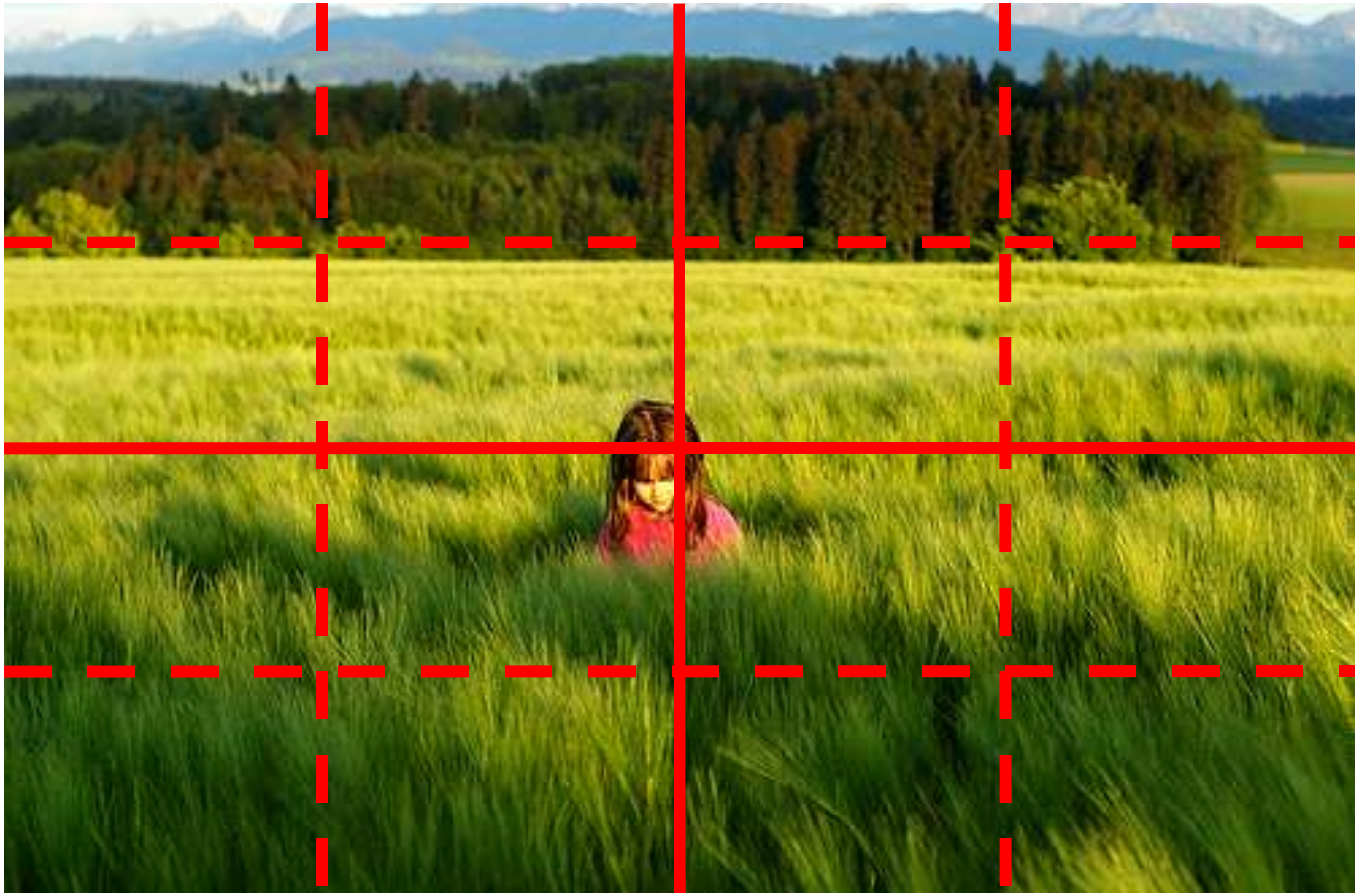
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
  - Right size is application-dependent

- Improving vocabulary quality
  - Discriminative/supervised training of codebooks
  - Sparse coding, non-exclusive assignment to codewords

The lack of geometry in BoW (spatial information)

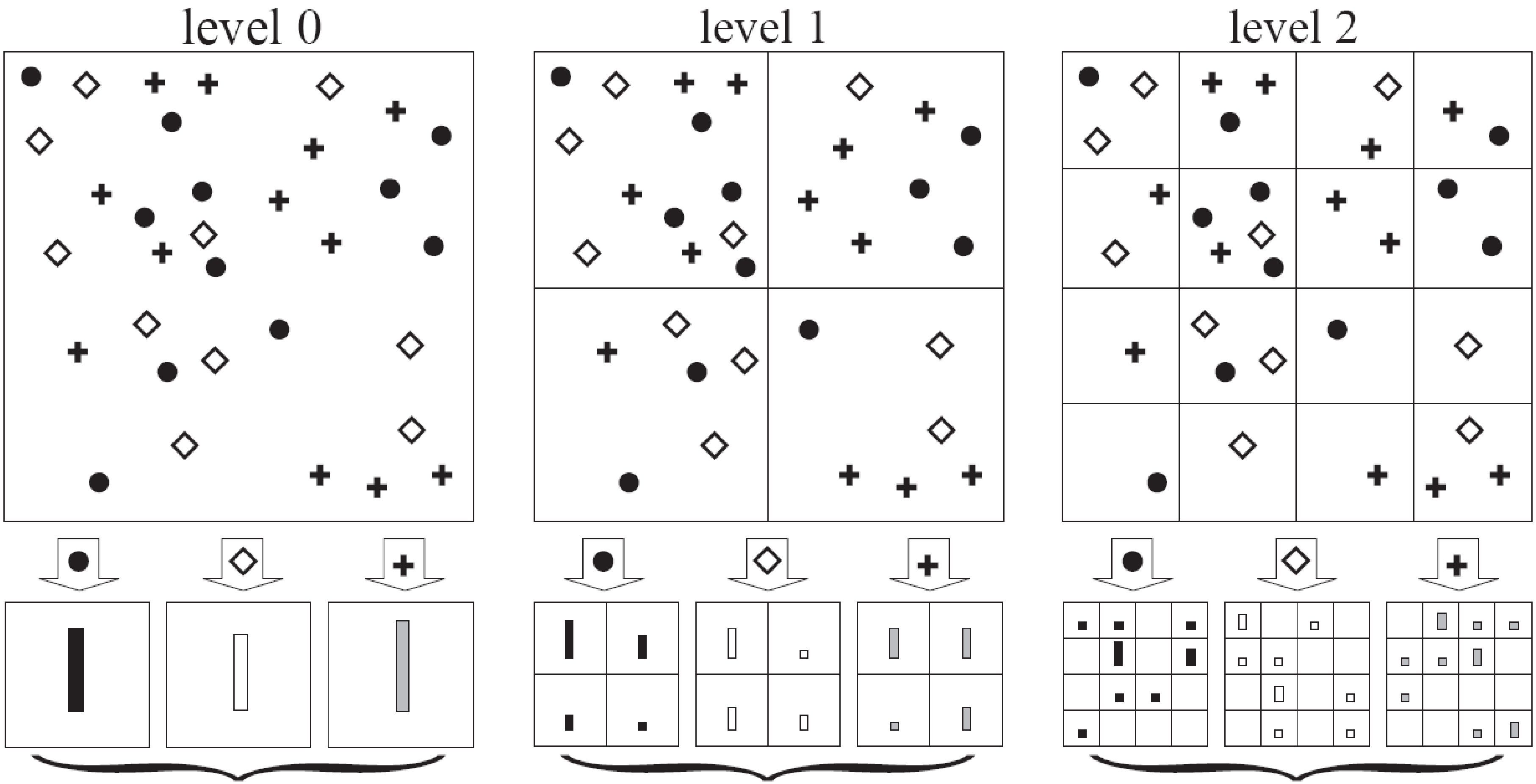
- The lack of geometry in the bag-of-words representation (BoW) can be either an advantage or a disadvantage.
- By encoding only the occurrence of the appearance of the local patches, not their relative geometry, we get significant flexibility to viewpoint and pose changes.
- Geometry between features can itself be an important discriminating factor, which a BoW will miss.

Spatial pyramid



- Compute histogram in each spatial bin

Spatial pyramid



- One critism high dimensionality and you can deal with by doing PCA dimension reduction