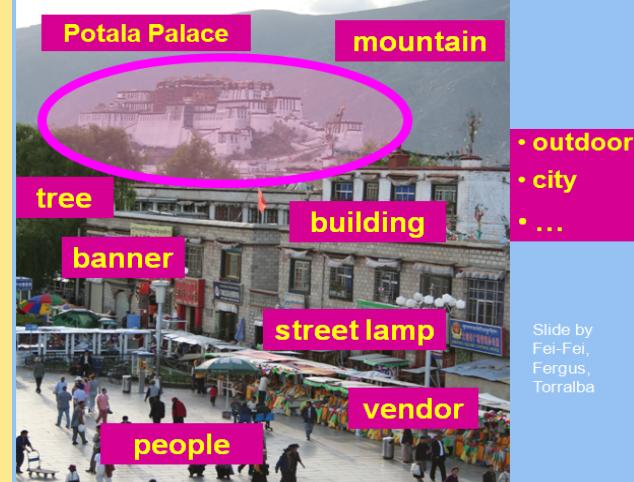


From Bag of features to Bag of Words



1. **Introduction to Bag of Words**
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline

Bag of Feature (BoF) Model

Image



(features)

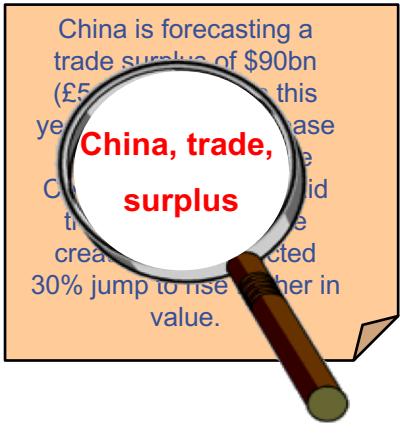


Bag of Words representation

- BoF
 - Local signatures: not a scalable representation
 - Not a *semantic* representation
- Model to represent images for categorization: « **Bag of Words BoW** »
- BoW model computed from BoF (Bag of features)



Bag of Words (BoW) model: basic explication with textual representation and color indexing



Comparing 2 docs using visual/color/word occurrences

Bag of Visual Words (BoW)

(features)



BoW : histogram on
visual dictionary



Questions:

1. Which dictionary ?
2. How to project the BoF onto the dico
3. How to compute the histogram?

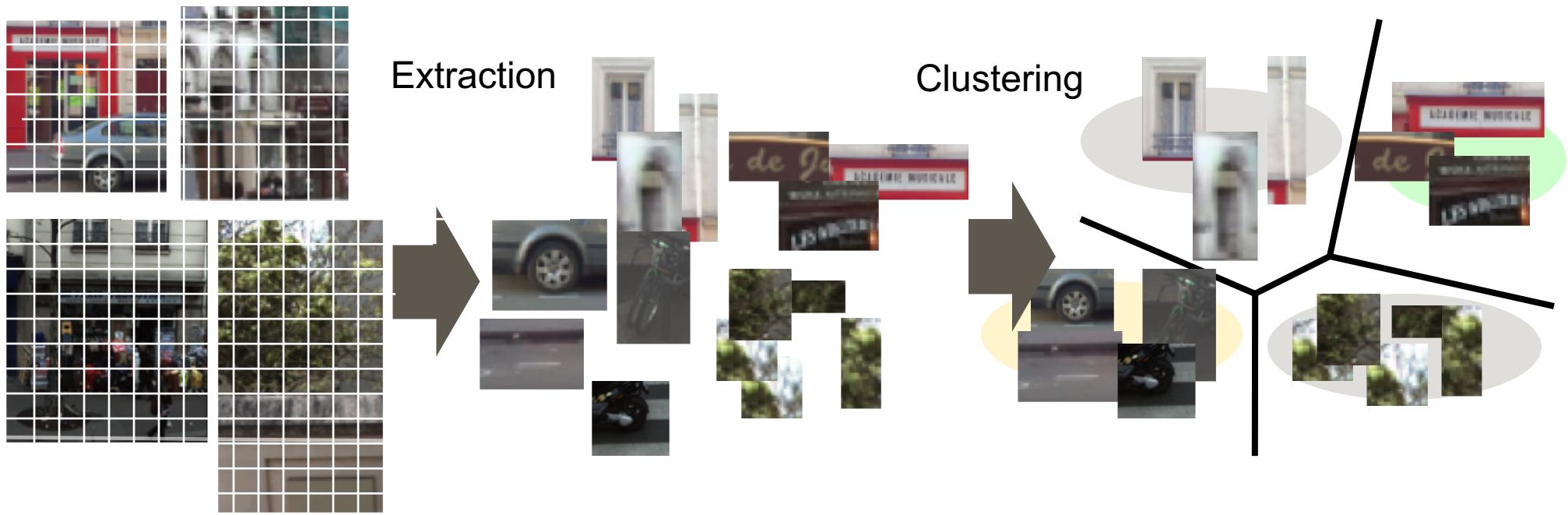
Outline

1. Introduction to Bag of Words
- 2. Dictionary computation**
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline

3 steps
for
BoWs

Step 1 : dico computation

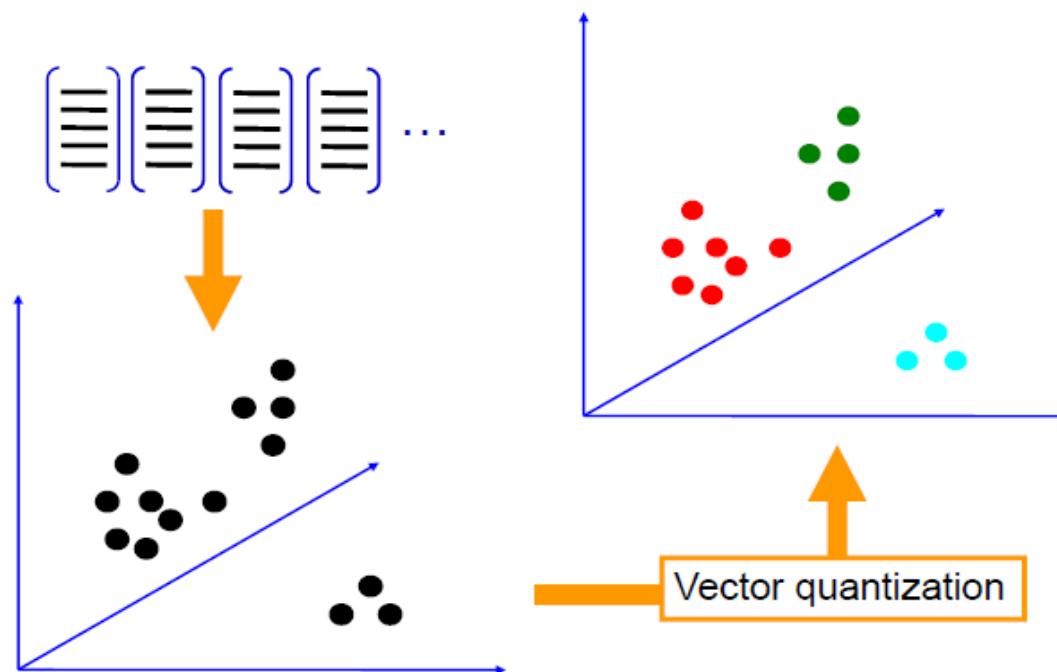
1. Extraction of local features (pattern/visual words) in images
 - Training dataset in classification
 - Image dataset in retrieval
2. Clustering of feature space



Training set but no labels => UNSUPERVISED Learning

Step 1 : dico computation

- Many algorithms for clustering :
 - K-Means
 - Vectorial Quantization
 - Gaussian Mixture Models
 - ...



Clustering with K clusters

Input: set of n points $\{x_j\}_n$ in R^d

Goal: find a set of K ($K < n$) points $w = \{w_k\}_K$
that give an approximation of the n input points,
ie. minimizing mean square error $C(w)$:

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2$$

At k fixed, complexity is $O(n^{(Kd+1)} \log n)$

A lot of strategies to approximate the global optimization problem

Clustering with K clusters

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2$$

K-means Algorithm:

Init K centers (c_k) by sampling K points w_k in R^d

1. (Re)assign each point x_i to the cluster s_i with the center w_{s_i} so that $\text{dist}(x_i, w_{s_i})$ is less than dist from x_i to any other clusters
2. Move all w_k inside each cluster as the new barycenter from all the points assigned to the cluster k (equ. to minimize the corresponding mean square error)
3. Go to step 1 if some points changed clusters during the last iteration

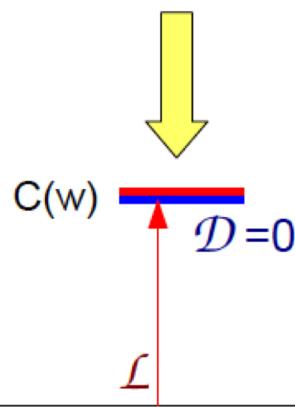
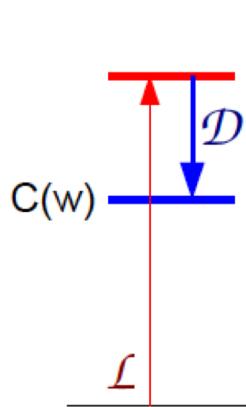
Output: the set of the final K cluster centers $\{c_k = w_k\}$

K-means : why it is successful ?

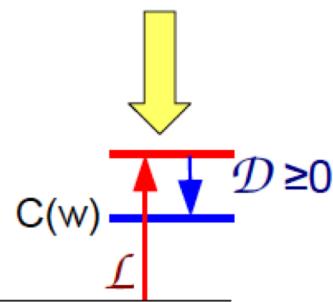
Consider an arbitrary cluster assignment s_i .

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2 = \underbrace{\sum_{i=1}^n \|x_i - w_{s_i}\|^2}_{\mathcal{L}(s,w)} - \underbrace{\sum_{i=1}^n \|x_i - w_{s_i}\|^2 - \min_k \|x_i - w_k\|^2}_{\mathcal{D}(s,w) \geq 0}$$

1. Change s_i to minimize \mathcal{D} leaving $C(w)$ unchanged.

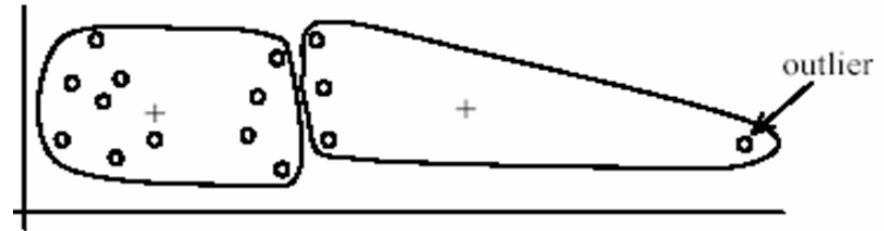


2. Change w_k to minimize \mathcal{L} . Meanwhile \mathcal{D} can only increase.



Clustering

- K-means :
 - Pros
 - Simplicity
 - Convergence (local min)
 - Cons
 - Memory-intensive
 - Depending on K
 - Sensitive to initialization
 - Sensitive to artifacts
 - Limited to spherical clusters
 - Concentration of clusters to areas with high densities of points (Alternatives : radial based methods)
- K-Means deeply used in practice



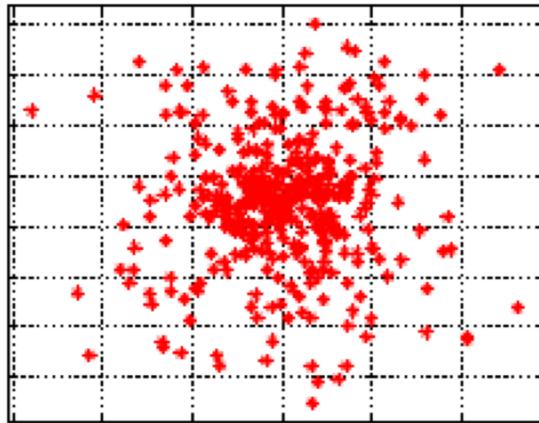
(A): Undesirable clusters



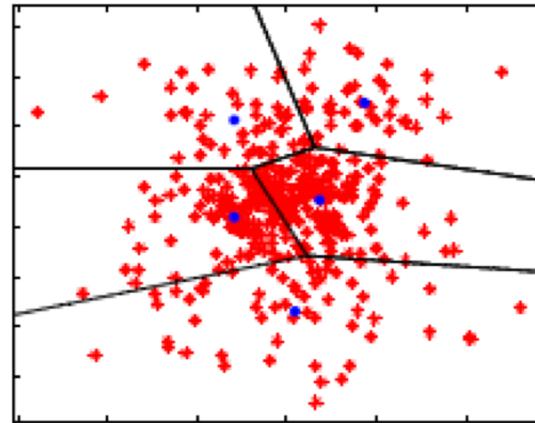
(B): Ideal clusters

Clustering

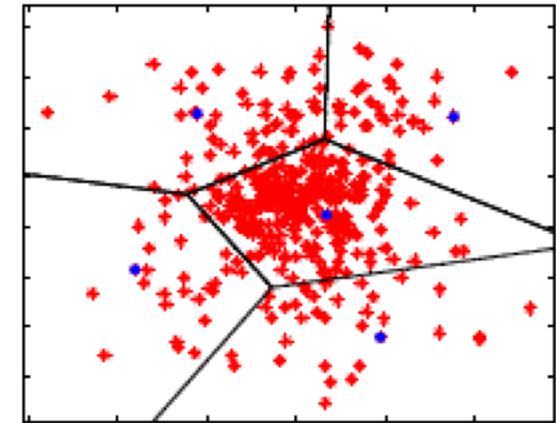
- Uniform / K-means / radius-based :



(a) Histogram



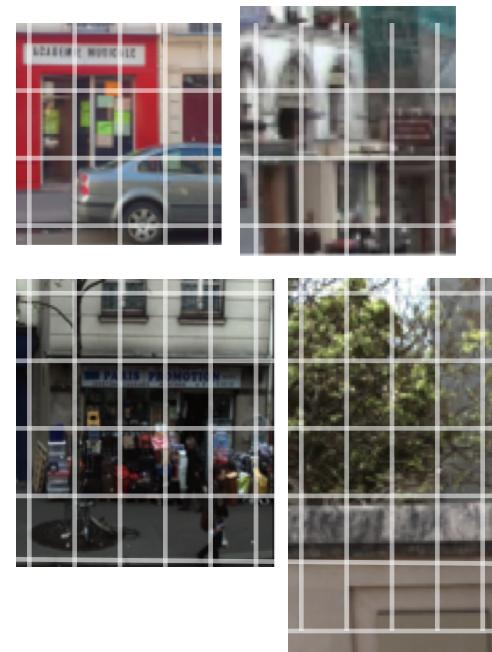
(b) K -means



(c) Radius-based

- *Radius-based clustering assigns all features within a fixed radius of similarity r to one cluster.*

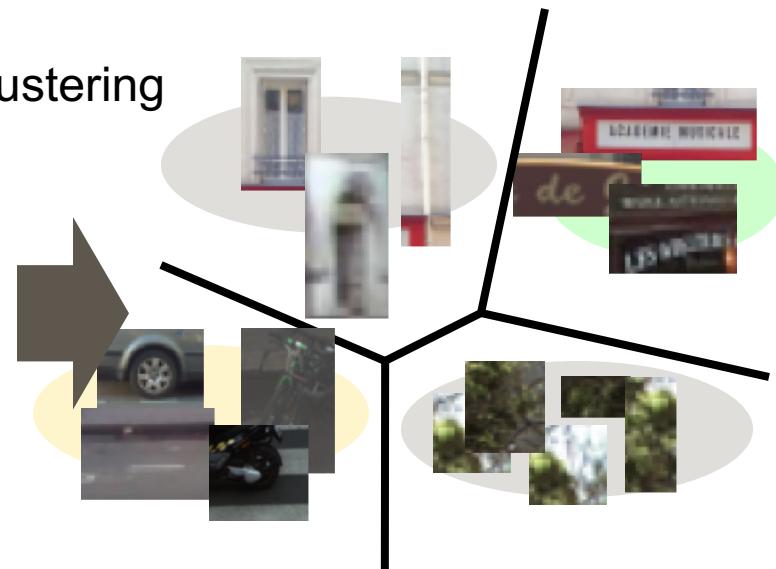
Visual words



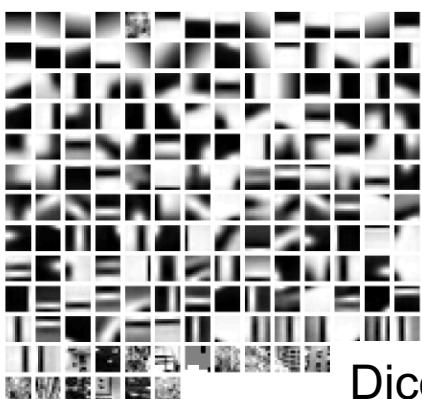
Extraction



Clustering



Formation du dico



Centers = dico. Visual words



Dico examples

Plan

1. Introduction to Bag of Words
2. Dictionary computation
- 3. Coding of local descriptors**
4. Image signature computation: pooling
5. Whole recognition pipeline

Step 2 : BoW image signature

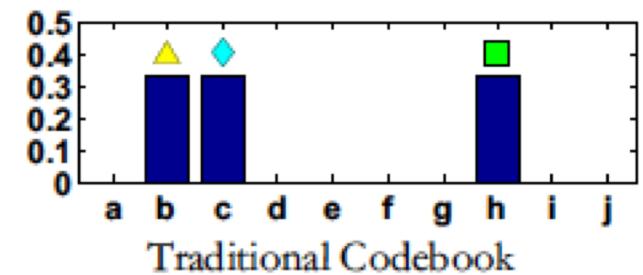
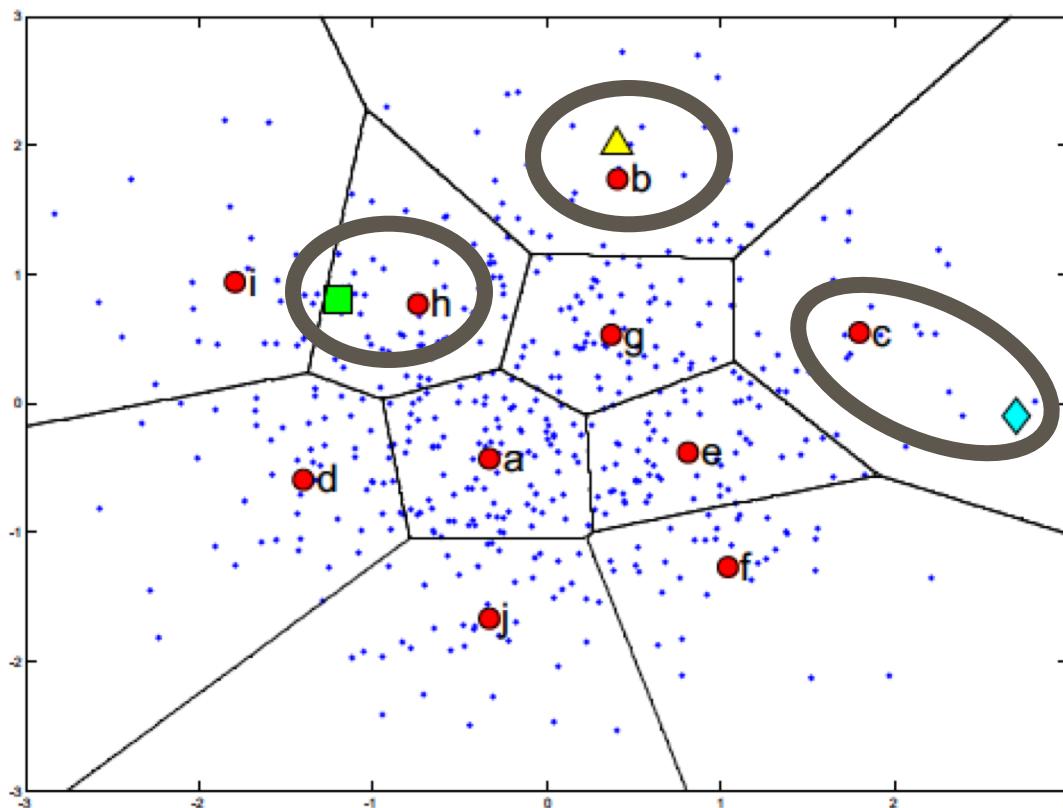
- For each image:
 - For each local feature: find the closest visual word
 - Increase the corresponding bin in histogram



- Image signature (global Index):
 - Vector (histogram)
 - $M=K$ dimension = dico size
 - Each term represents a Likelihood to get this visual word

Projection to dictionary

- Original BoW strategy: **hard assignment/coding**
 - Find the closest cluster for each feature
 - Assign a fix weight (e.g. 1)



Notations:

- Image data :
- Centers :
- Coding :

$$f : \mathbb{R}^d \longrightarrow \mathbb{R}^M$$

$$x_j \longrightarrow f(x_j) = \alpha_j = \{\alpha_{m,j}\}, \quad m \in \{1; M\}$$

Hard coding: $f = f_Q$ assigns a constant weight to its closest center:

$$f_Q(x_j)[m] = \begin{cases} 1 & \text{if } m = \underset{k \in \{1; M\}}{\operatorname{argmin}} \|x_j - c_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{X} = \left\{ x_j \in \mathbb{R}^d \right\}, j \in \{1; N\}$$
$$\mathbf{C} = \{C_m\}, m \in \{1; M\}$$

$$\begin{array}{c}
 & & x_1 & & x_j & & x_N \\
 & c_1 & \left[\begin{array}{cccc} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{array} \right] & \Rightarrow g: \text{pooling} \\
 \mathbf{H} = c_m & & & & & & \\
 & c_M & & & & & \\
 & & & \Downarrow & & & \\
 & & f: \text{cooding} & & & &
 \end{array}$$

Plan

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
- 4. Image signature computation: pooling**
5. Whole recognition pipeline

Aggregating projections => global image index

- Global Index: image likelihood to get each visual word
- Several strategies to aggregate the projections: **pooling**

$$g : \mathbb{R}^N \longrightarrow \mathbb{R}$$

$$\alpha_{\mathbf{m}} = \{\alpha_{m,j}\}, j \in \{1; N\} \longrightarrow g(\alpha_m) = z_m$$



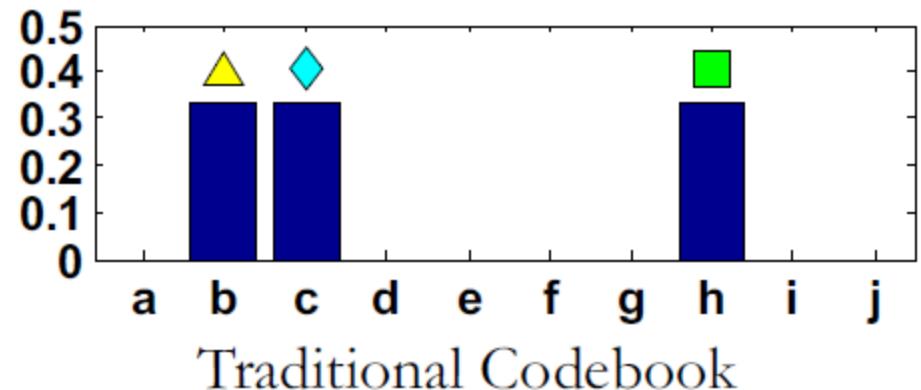
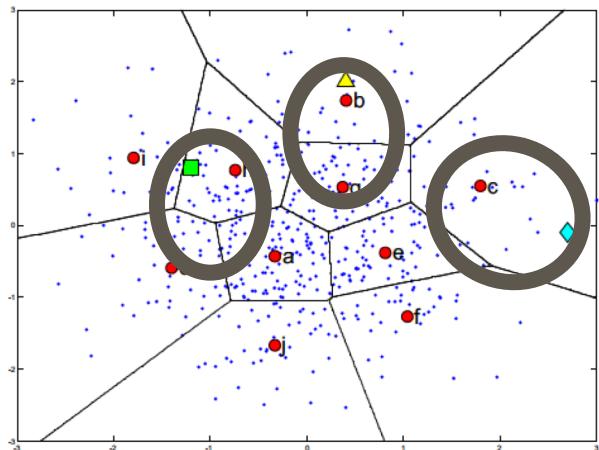
$$\begin{array}{c}
 & x_1 & x_j & x_N \\
 \\
 \mathbf{H} = c_1 & \left[\begin{array}{cccc} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{array} \right] & c_m & \Rightarrow g: \text{pooling} \\
 \\
 & \Downarrow & & f: \text{cooding} \\
 \end{array}$$

Aggregating projections => global image index

- **Sum pooling** : initial BoW strategy (just counting occurrences of words in the document)

Classical BoW = **hard coding + sum pooling**

1. Find the closest cluster for each feature
2. Assign a fix weight (e.g. 1) to this cluster



Aggregating projections => global image index

- BoW Sum pooling:

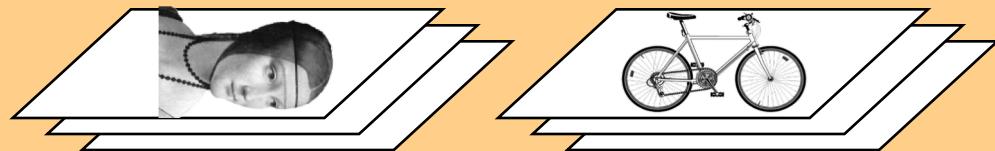
$$z_m = g(\alpha_m) = \sum_{j=1}^N \alpha_{m,j} = \sum_{j=1}^N f_Q(x_j)[m]$$

$$z_m = \sum_{j=1}^N \begin{cases} 1 & \text{if } m = \underset{k \in \{1; M\}}{\operatorname{argmin}} \|x_j - c_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

Plan

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
- 5. Whole recognition pipeline**

Representation



1. feature detection
& representation

2. codewords dictionary

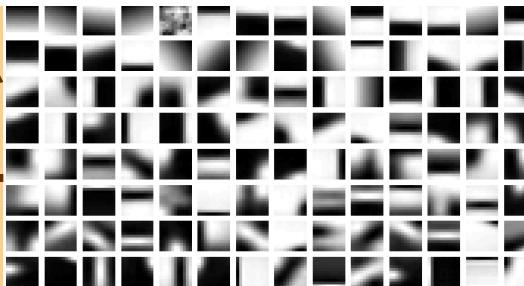
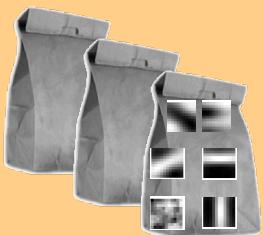


image representation

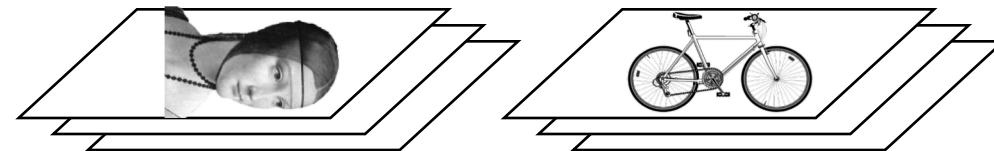
3.



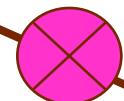
$\left[\begin{array}{c} \\ \\ \end{array} \right] \left[\begin{array}{c} \\ \\ \end{array} \right] \left[\begin{array}{c} \\ \\ \end{array} \right] \dots$

Representation

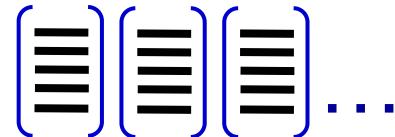
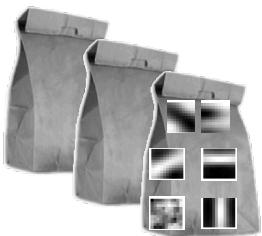
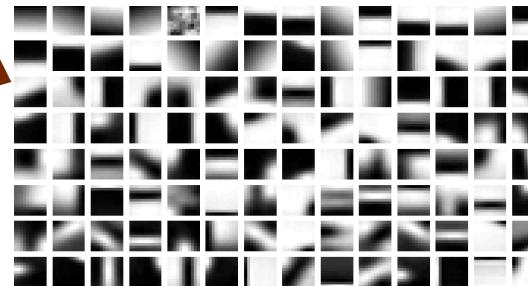
Learning and Recognition



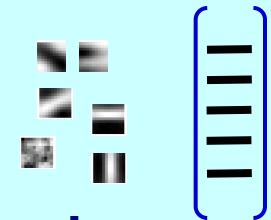
1. feature detection & representation



2. codewords dictionary



- 3.



- 4.

category models
(and/or) classifiers



category
decision