MLRF Lecture 04

J. Chazalon, LRE/EPITA, 2025

Content based image retrieval

Lecture 04 part 02

Two strategies using local descriptors

Keep all local descriptors

+ Enables geometric validation
+ Better part detection in theory

- Huge memory requirements
- Complex voting scheme

Like what we did in practice session 3 to match parts of an image (useful to validate geometric constraints and classify an image at the same time).

Build a global descriptor using local ones

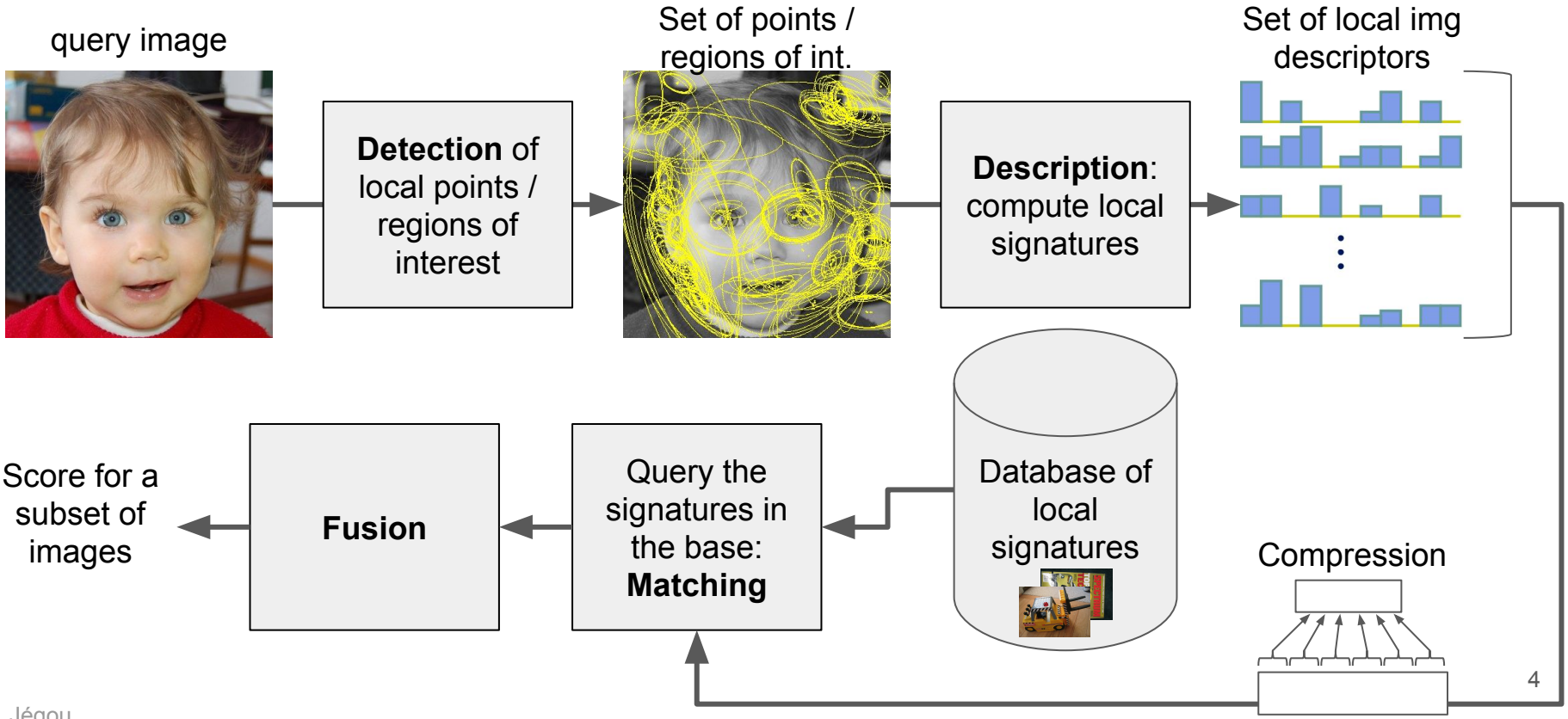
+ Inspired by text retrieval
+ Compact representation
+ Tricks to embed spatial information
+ Limited memory requirements

- Requires pooling/embedding

Like what we did in practice session 2 with the color histogram, at the bubble level!

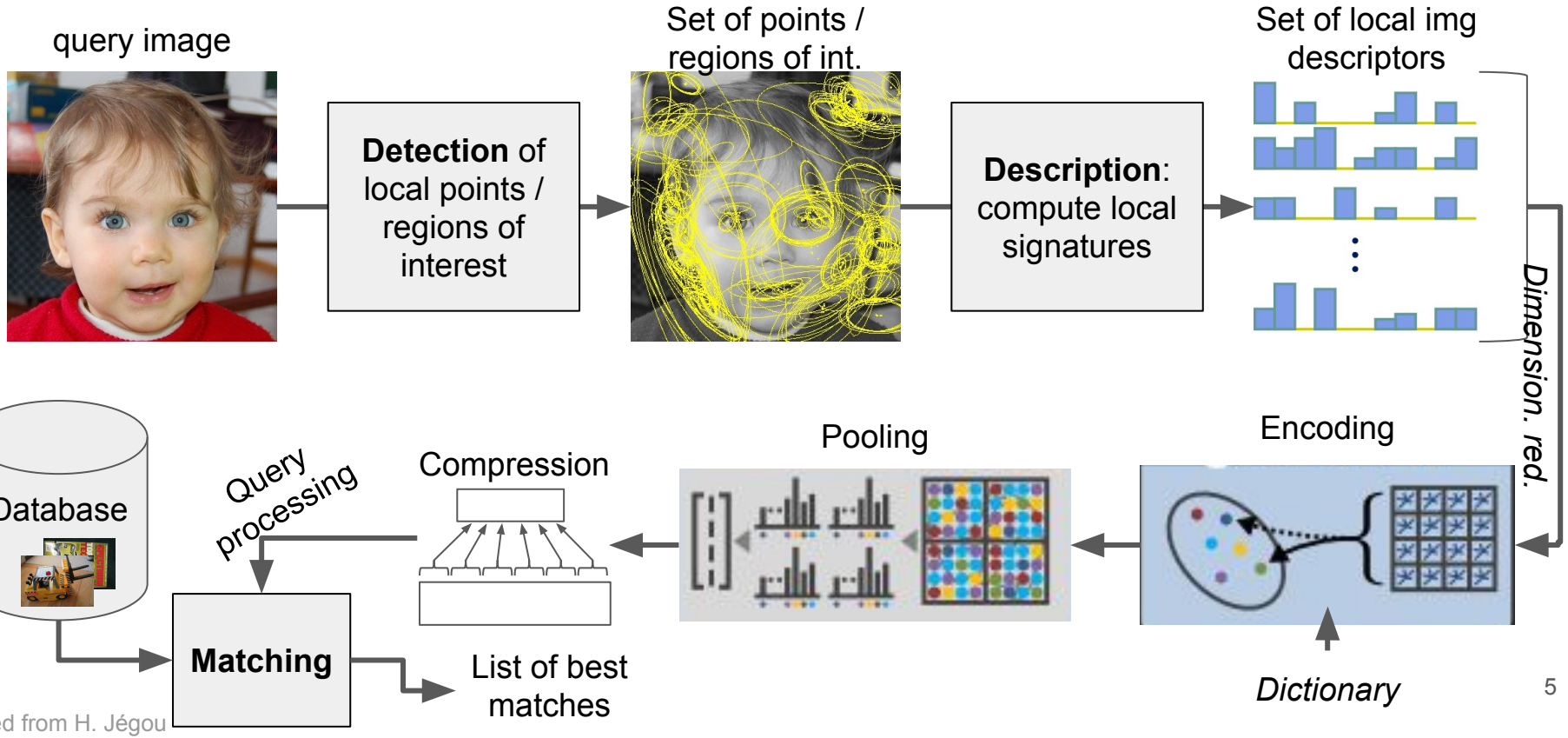
***Bag of Features* approach**

Pipeline with local descriptors (prev. lecture)



from H. Jégou

Pipeline with bag of features (current lecture)

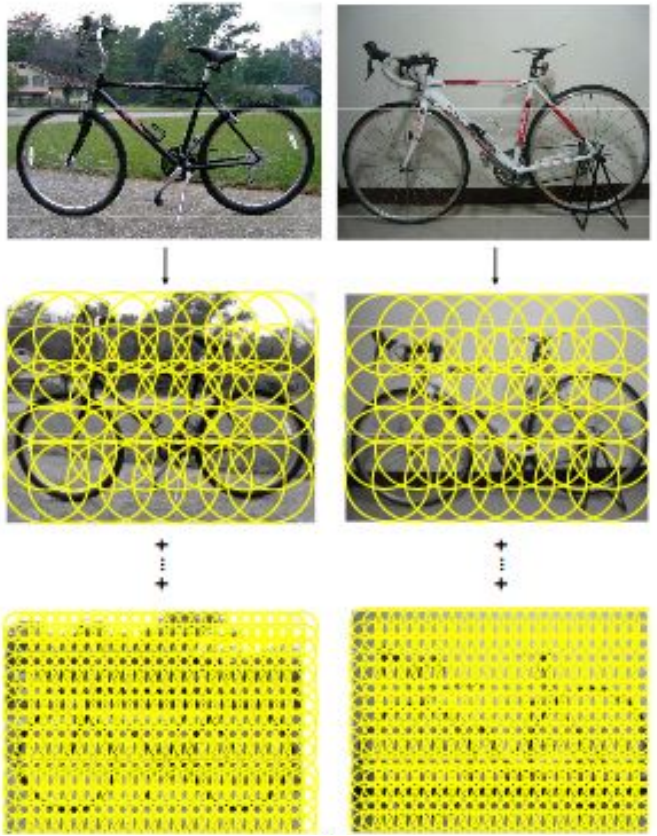


Adapted from H. Jégou

Feature extraction

Sparse vs Dense detection

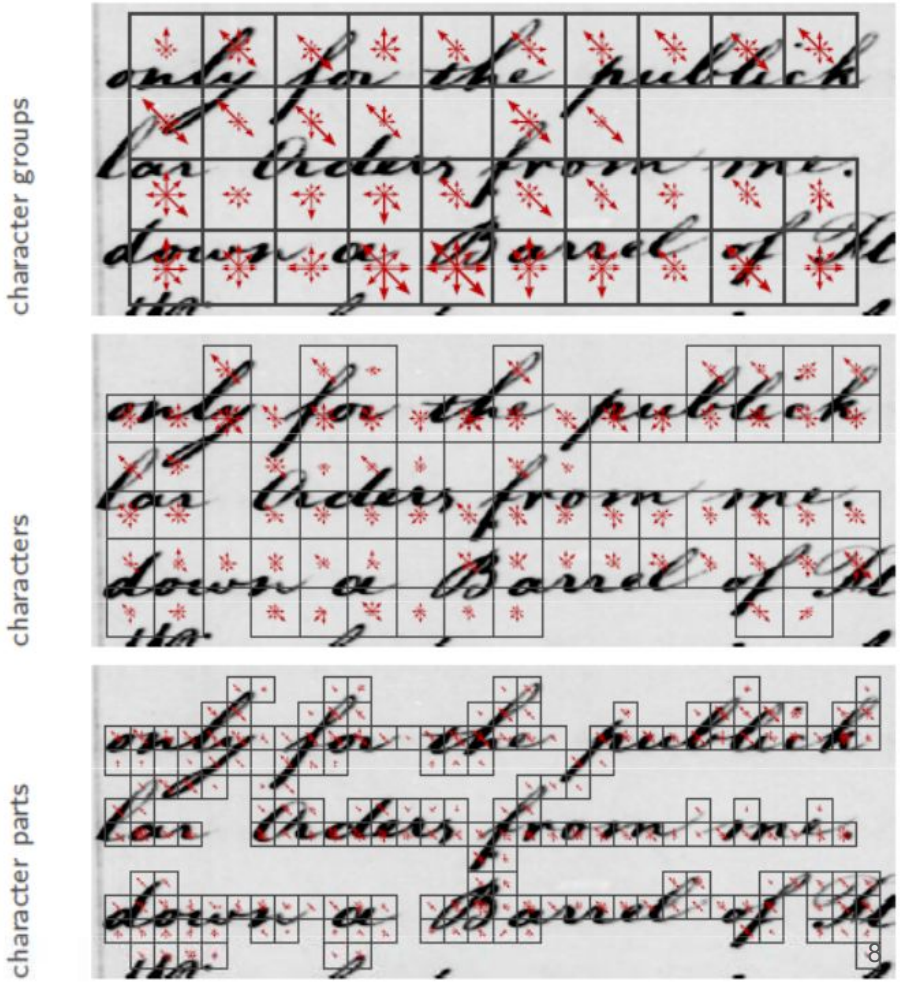
Sparse



Dense

Sparse vs Dense detection

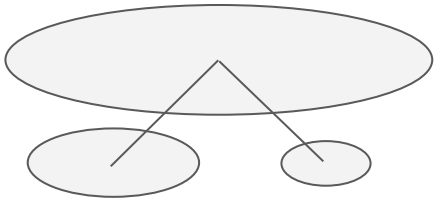
For dense detections,
we usually filter regions
with low variance



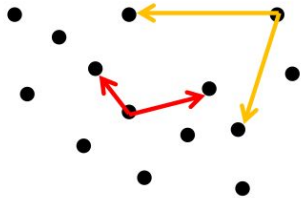
from M. Rusiñol

Making descriptor more discriminative

Build pairs from included regions



Or triplets from neighbor points



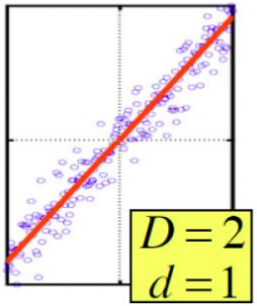
Dimensionality reduction

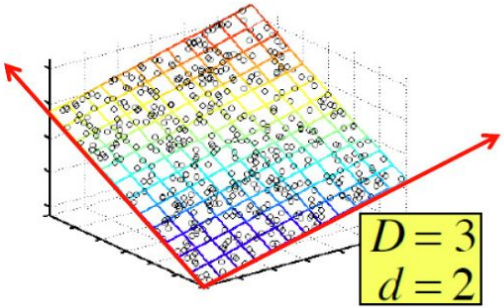
See J. Leskovec, A. Rajaraman, J. Ullman, Mining of Massive Datasets, chapter 11 at http://mmds.org

Dimensionality reduction

Often used before encoding to:

- limit dictionary sizes
- facilitate quantization





Several techniques :

- Principal-Component Analysis (PCA):
  - New basis for coordinates (translation + rotation)
  - Widely used
- Singular-Value Decomposition (SVD) and CUR Decomposition
  - More powerful
  - Latent modeling of “topics” or “concepts”

Encoding

Bag of Visual Words

Modern approaches are derived from this one.

Reuses the ideas of text / web search to images.

From a set of descriptor, build a histogram of quantized descriptors.
Much alike a color histogram!

General idea: summarize the distribution of local patterns

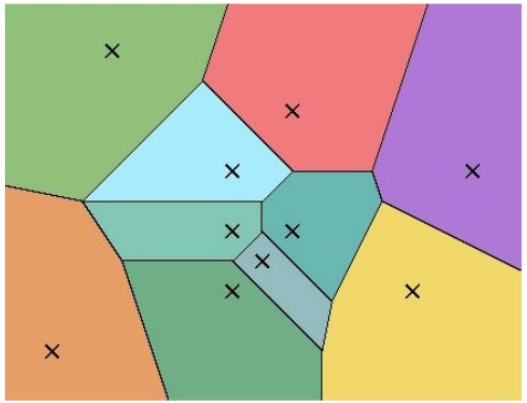
J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos”, International Conference on Computer Vision, 2003

Quantization

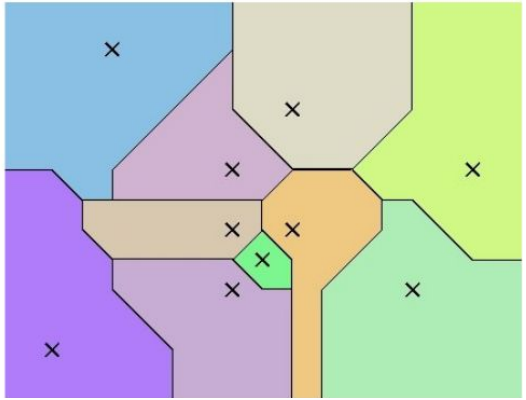
Discretization of some signal — **Lossy process!**

Vectorial formulation: f: $R^d \rightarrow F$, with F = {1, 2, …, k}

Defines a **Voronoi diagram**, ie a decomposition of a metric space determined by the distances to a discrete set of point.



Euclidean (L2) distance



Manhattan (L1) distance

Bag of Visual Words (continued)

Cluster centers are determined using k-Means (once for all on a training set).

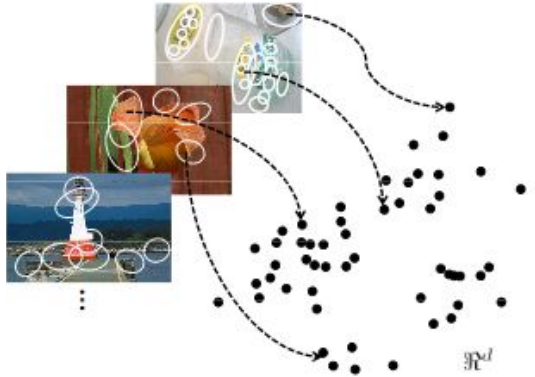
Each descriptor is quantized: store the code of the closest centroid.

Build a histogram of descriptors count for each cluster.

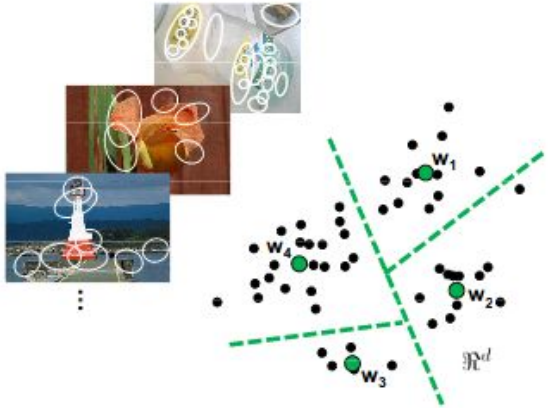
The set of cluster centers is called the dictionary, the codebook or also the visual vocabulary.

We can choose the number of words!

Codebook creation

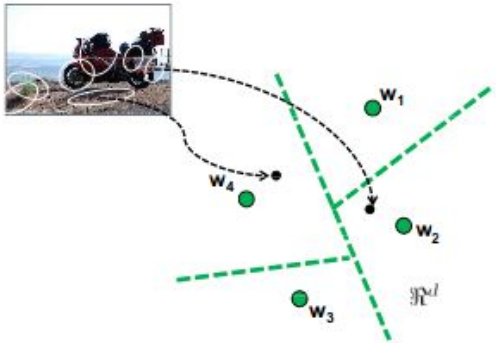


(a)

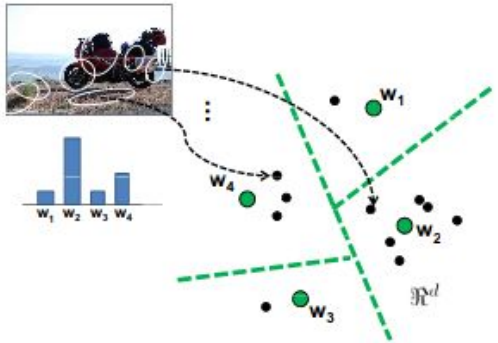


(b)

Indexing process

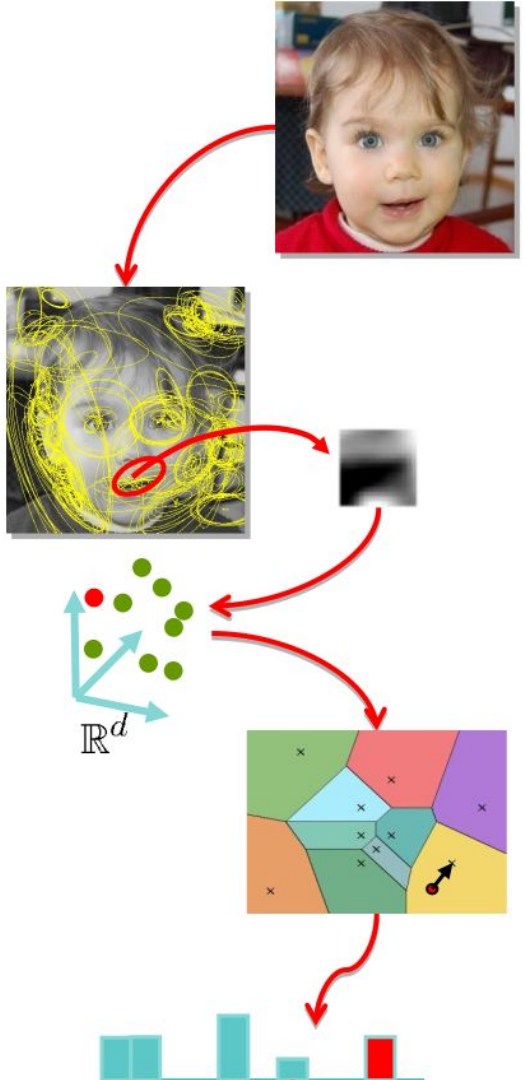


(c)



(d)

from H. Jégou, C. Schmid



from H. Jégou

Bag of Visual Words: Vector size

The resulting vector size for a given image is given by:

D = vocabulary size

Usually, the bigger the vocabulary the better the results (up to some limits).

Several thousands of words are common.

Bag of Visual Words: Normalization (1/2)

Problem:

- The values in the histogram are **absolute**: each bin count the number of occurrence of each *visual word*.
- This make the descriptor **sensitive to the variation of number of descriptors** extracted in each image (more descriptors ⇒ higher peaks)

Solution:

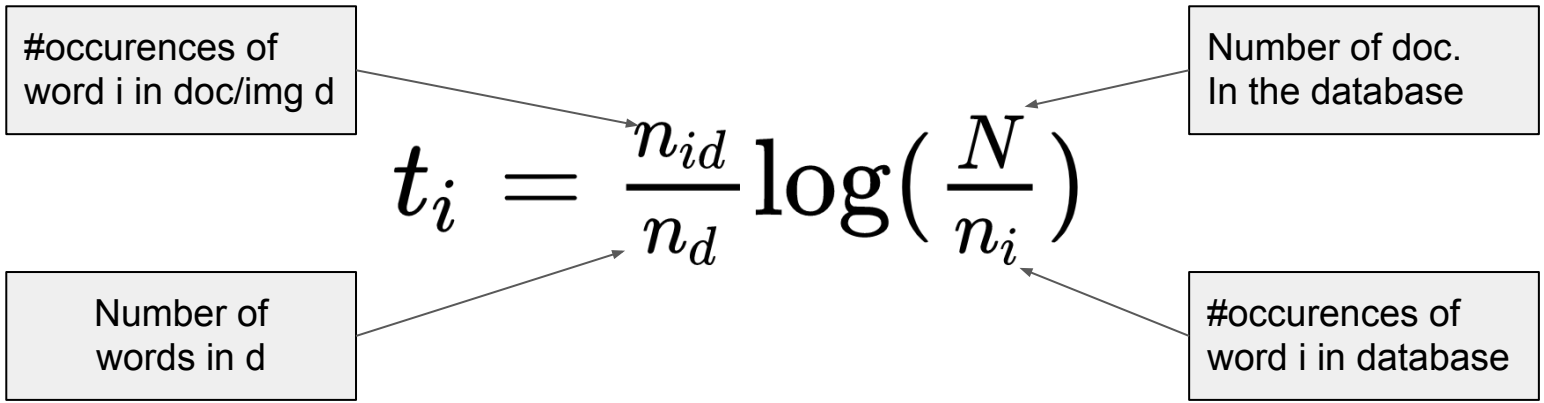
- **Normalize** (divide) histogram value by the number of descriptors extracted
  - *Warning: we get values in [0,1] range: we need float vectors!*
- This is a **L1 normalization**: $||v||_1 = \Sigma_i |v_i|$

Bag of Visual Words: Normalization (2/2)

Like for text retrieval, it is common to **reweight the BoVW vectors** using the **TFIDF** technique.

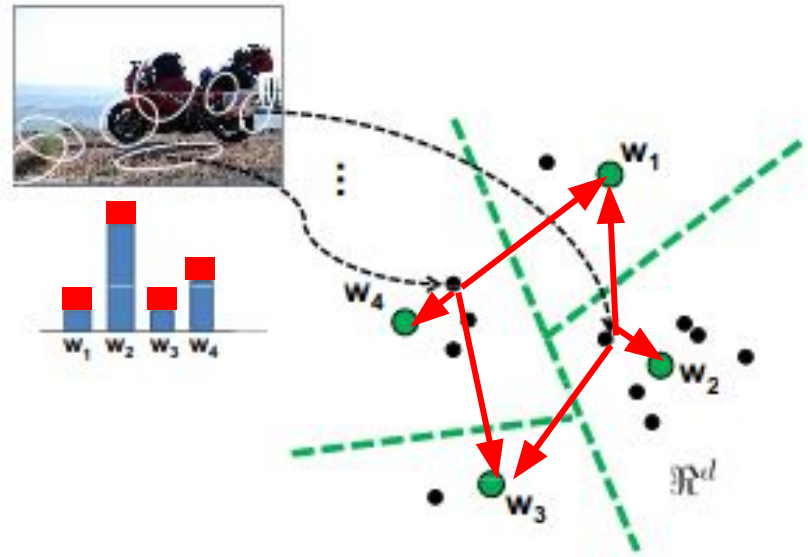
Goal: give more importance to rare words than to frequent ones.

For each dimension of the histogram, compute a new value t_i:



Variant: Soft BoVW

Use soft assignment to clusters, add counts to neighbor bins

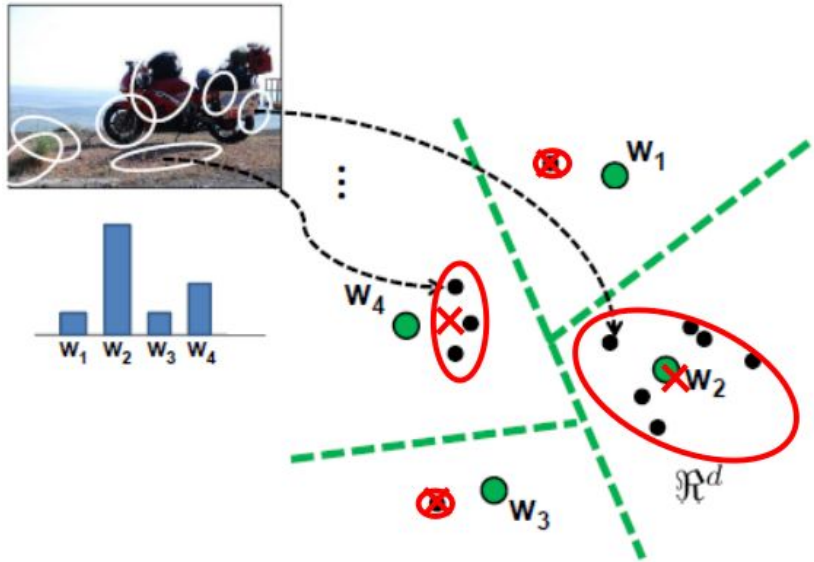


Other variants

BoVW is only about counting the number of local descriptors assigned to each Voronoi region.

It is possible to includes other statistics:

- Mean of local descriptor ✖
- Distribution of descriptors within a cluster

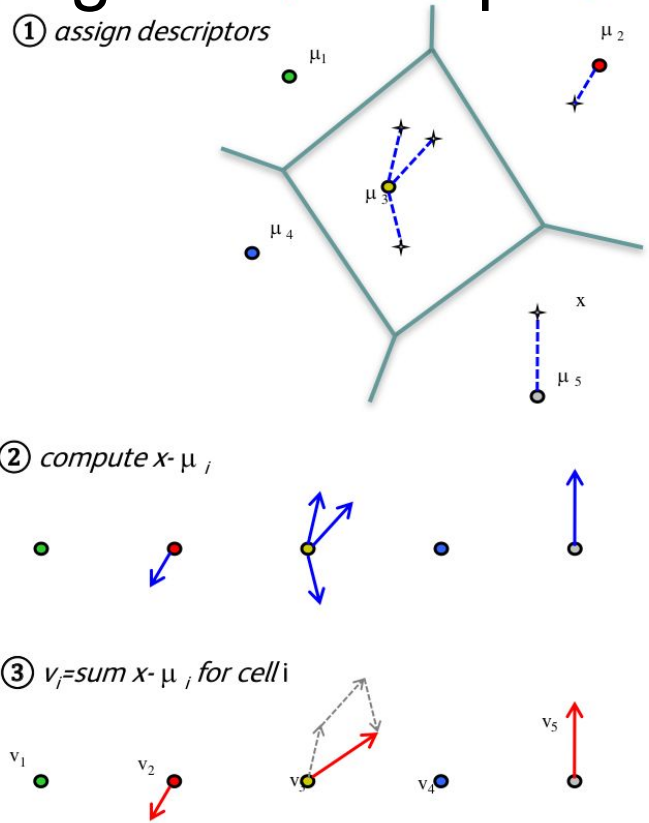


VLAD: vector of locally aggregated descriptors

Given a codebook $\{\mu_i, i = 1 \dots N\}$,
e.g. learned with K-means, and a set of
local descriptors $X = \{x_t, t = 1 \dots T\}$:

- ① assign: $\mathrm{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$
- ②③ compute: $v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$
- concatenate v_i's + ℓ_2 normalize

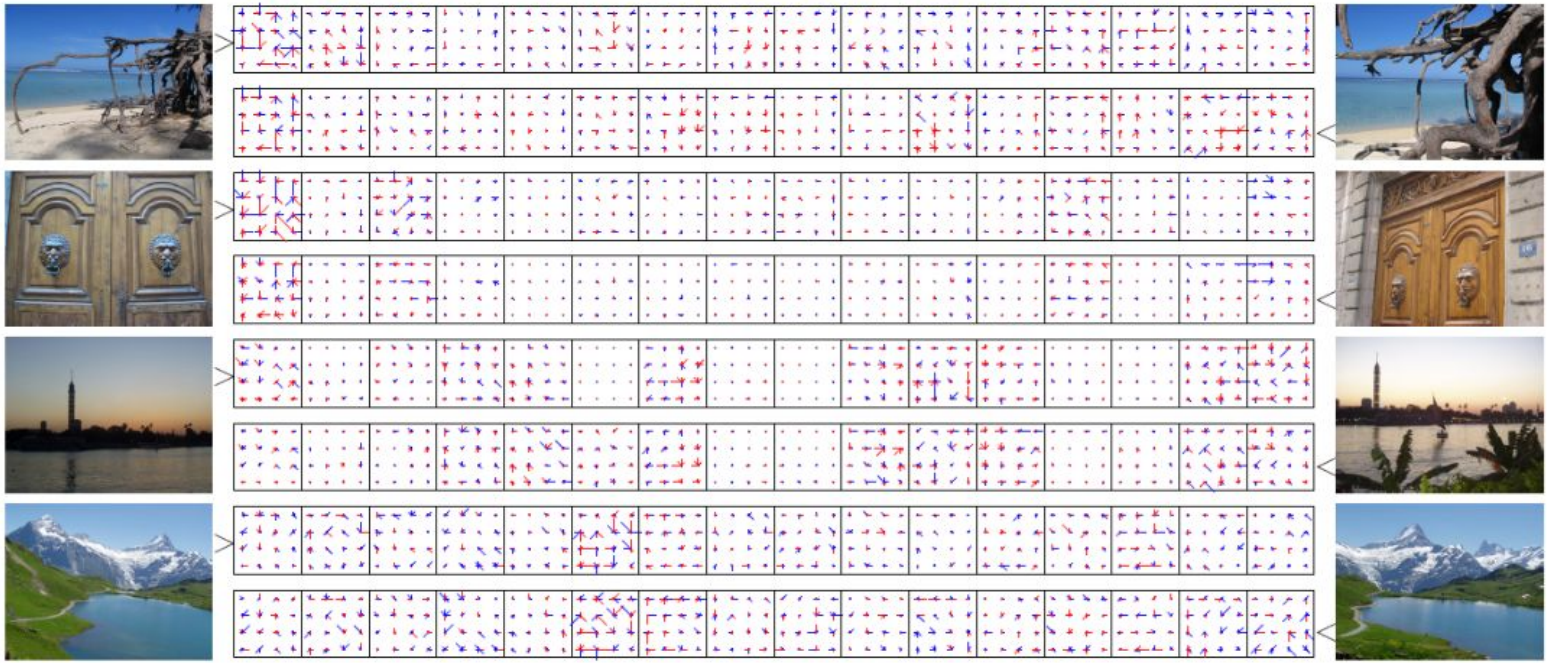
→ the VLAD **is DxN dimensional**



H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 3304–3311.

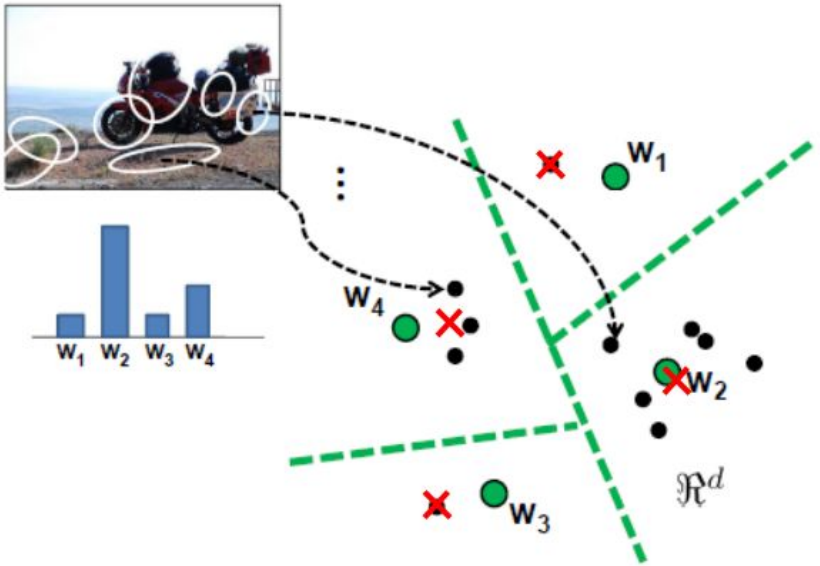
VLAD: vector of locally aggregated descriptors

Visualisation of $v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$



VLAD: vector of locally aggregated descriptors

The distribution of samples in each cell is encoded by its mean (minus the centroid mean)



Fisher vector

$X = \{x_t, t = 1 \dots T\}$ is the set of T i.i.d. D-dim local descriptors (e.g. SIFT) extracted from an image:

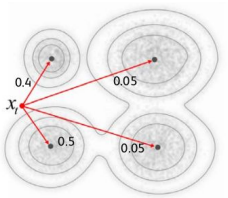
$$G_\lambda^X = \frac{1}{T} \sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t)$$

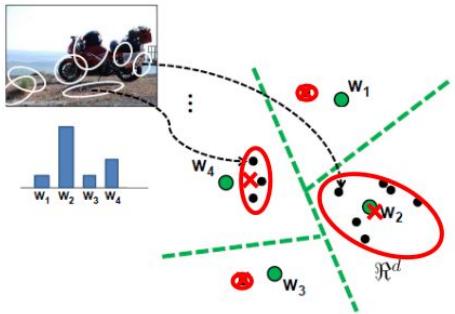
$u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x)$ is a Gaussian Mixture Model (GMM) with parameters $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots N\}$

The FV is typically 2xDxN dimensional

With respect to the VLAD:

- add 2nd order moments
- soft-assignment of local descriptors
- per-dimension whitening





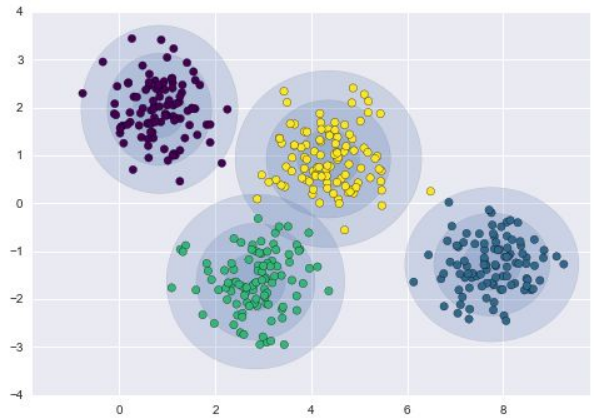
F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” CVPR 2007

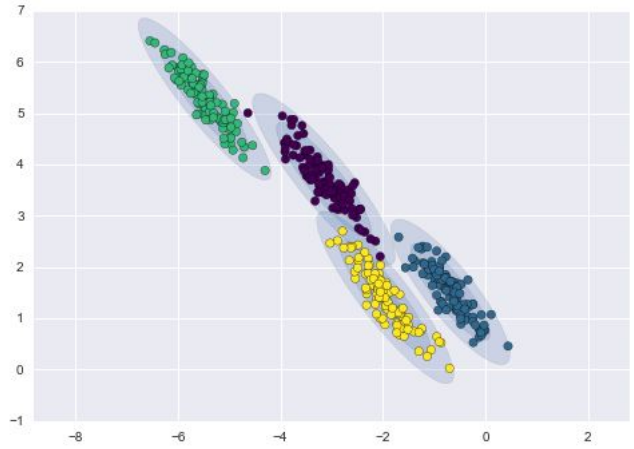
GMM

Probability that x is generated by the k-th Gaussian:

$$p(\mathbf{x}|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma_k}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_k)^\top \Sigma_k^{-1}(\mathbf{x} - \mu_k)\right].$$

Here μ_k and Σ_k are respectively the mean and covariance of the distribution.





Fisher vector

Requires many normalization tricks:

- PCA on the local descriptors is necessary because of the GMM diagonal approximation
- L2 normalization to make the FV more compliant with the dot-product assumption
- Power-normalization $f(z) = \text{sign}(z)|z|^{\alpha} \text{ with } 0 \leq \alpha \leq 1$

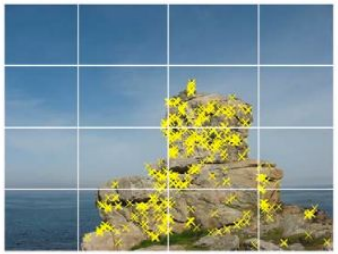
For a detailed analysis see: Sánchez, Perronnin, Mensink, Verbeek, “Image Classification with the Fisher Vector: Theory and Practice”, IJCV’13

Pooling

Spatial pooling

Used to keep some spatial info

Then, perform pooling over each region.



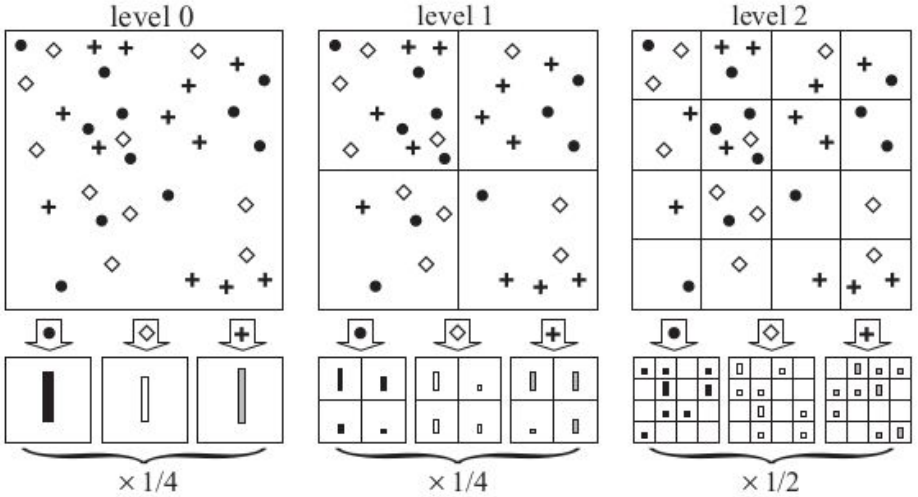
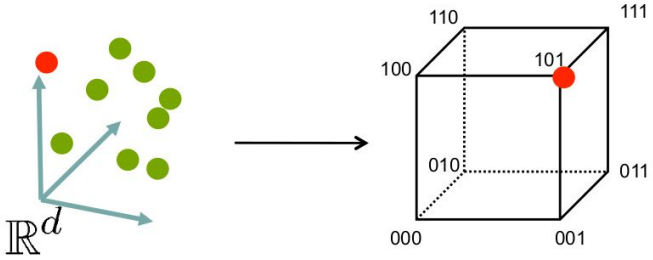


Figure 1. Toy example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin. Finally, we weight each spatial histogram according to eq. (3).

S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 2, pp. 2169–2178.

Compact the descriptors

Use LSH for binarization



Given **B** random projections with direction $\mathbf{a}_i$

Compute a binary code **b(x)** from vector **x** as

$$b_i(x) = \text{sign } a_i^\top x$$

$$b(x) = (b_1(x), \ldots, b_B(x))$$

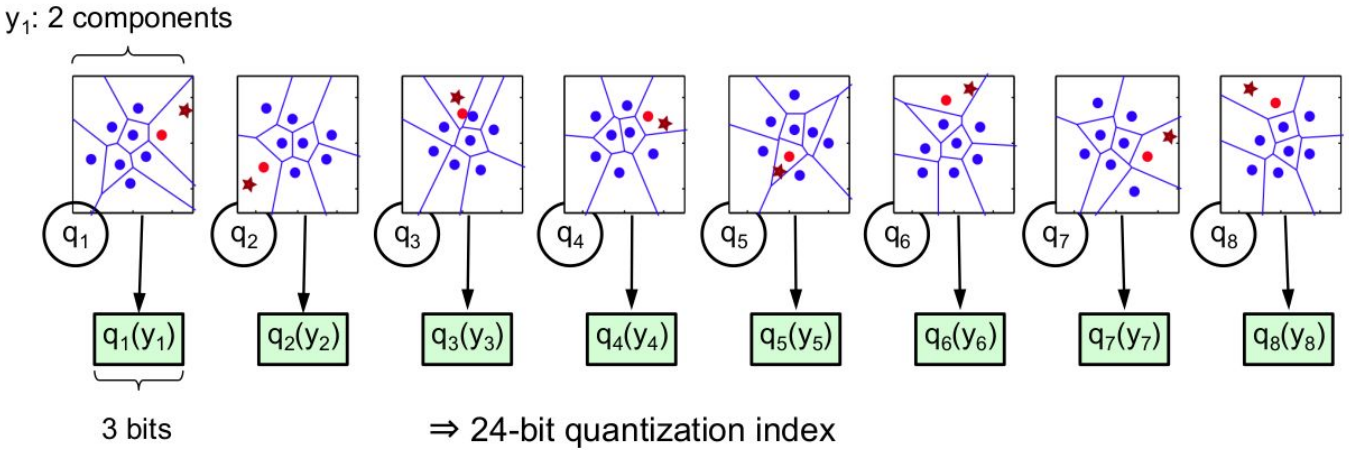
⇒ very compact
⇒ very fast comparison (Hamming distance, CPU accel.)
⇒ suitable for very high-dimensional vectors
(then use LSH again for binary indexing)

Product Quantizer

Split a vector into m subvectors $y \rightarrow [y_1|\ldots|y_m]$

Subvectors are quantized separately (with ex. k-Means)

Example: y= 16-dim vector split in 8 subvectors of dimension 2



Indexing & Matching

Approximate Nearest Neighbor problem

See lecture 3 part 3: Descriptors Matching and Indexing

Many options, like KD-Trees, Hierarchical K-Means, regular LSH…

Query

Query expansion

Recover more relevant results by reissuing a query with top results

O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In Computer Vision, International Conference on, pages 1–8, Rio de Janeiro, Brazil, October 2007.

Limitations

Limitations of the BoVW approach

The process organizes the representation based on descriptor density: some features / dimensions are meaningful and some are not. This is a self-supervised process without task-specific semantics.

Distances may not have the same meaning over the complete space: we would need to “warp” the representation space using supervised learning. Deep learning makes it possible (with contrastive loss in particular).