

MLRF Lecture 04

J. Chazalon, LRDE/EPITA, 2020

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

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).

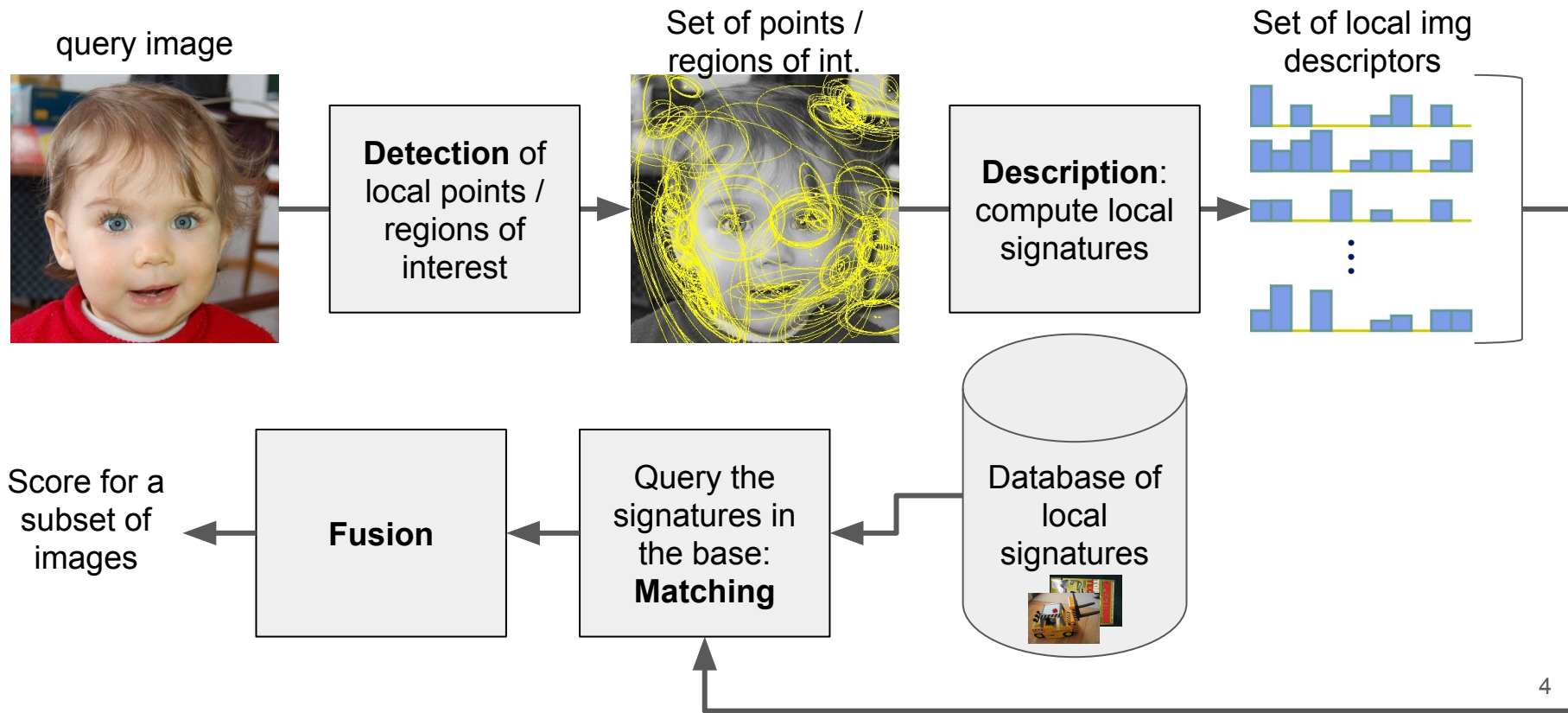
Build a global descriptor using local ones

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

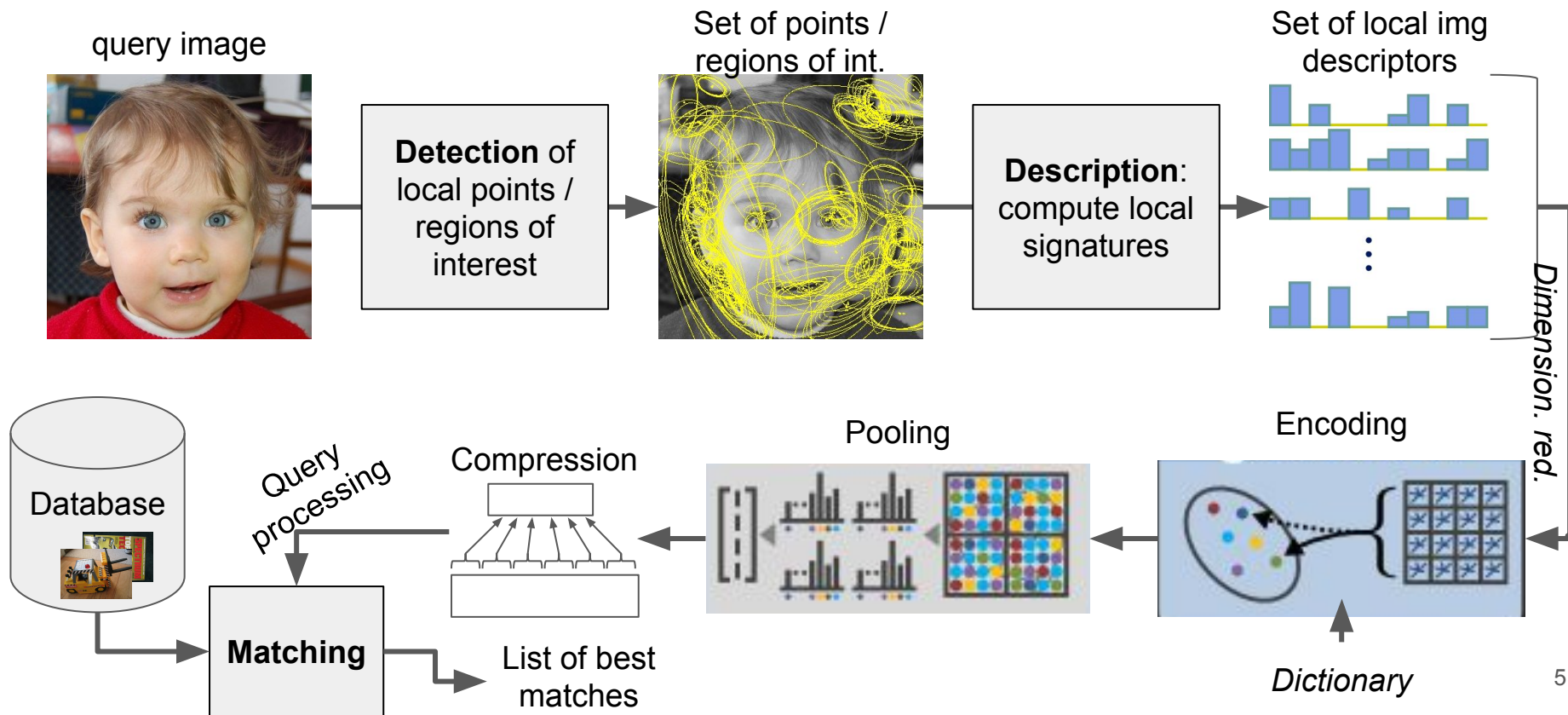
Like what we did in practice session 2 with the color histogram, at the bubble level!

Bag of Feature approach

Pipeline with local descriptors (prev. lecture)



Pipeline with bag of features (current lecture)

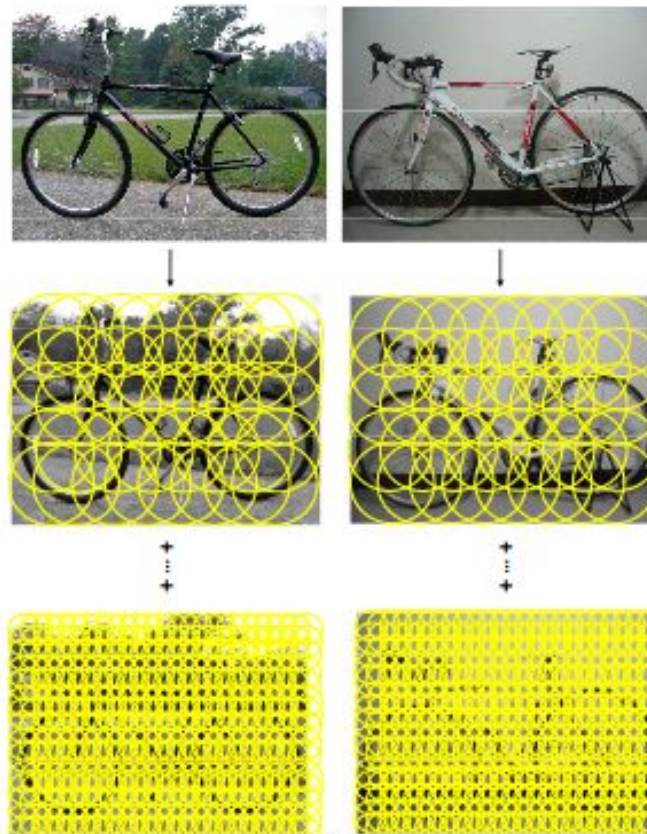


Feature extraction

Sparse vs Dense detection



Sparse

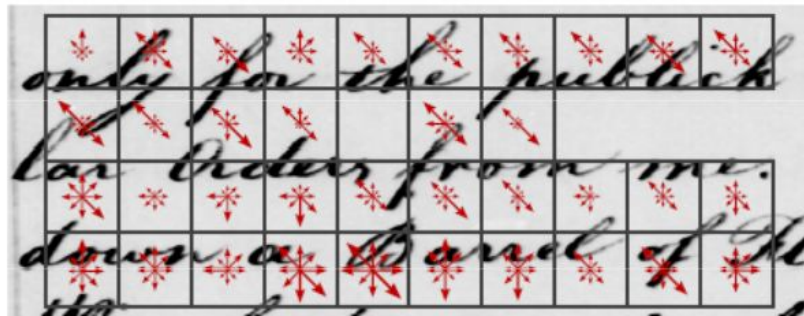


Dense

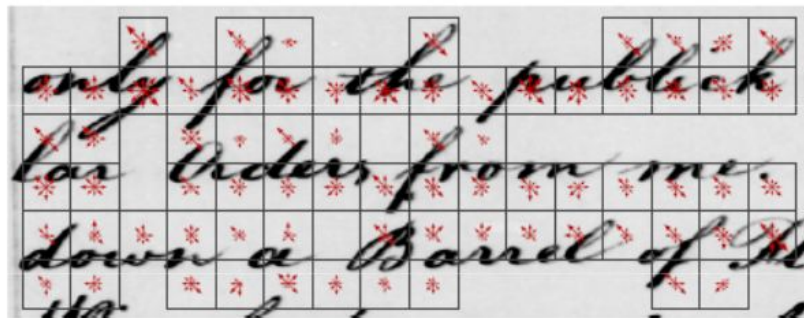
Sparse vs Dense detection

For dense detections,
we usually filter regions
with low variance

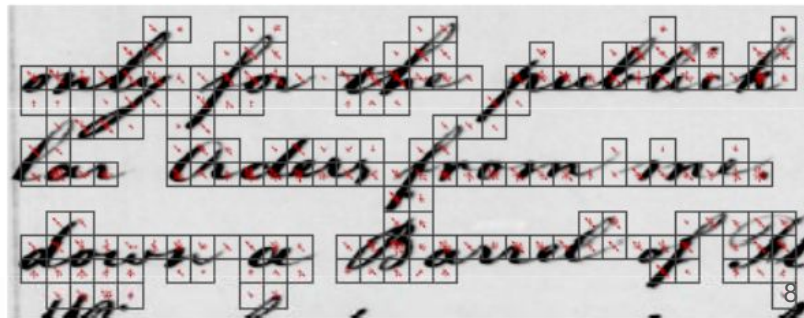
character groups



characters

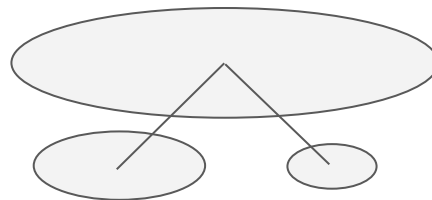


character parts

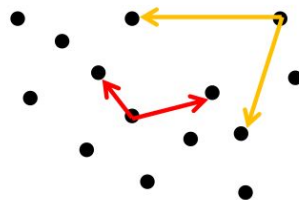


Making descriptor more discriminative

Build pairs from included regions



Or triplets from neighbor points



Dimensionality reduction

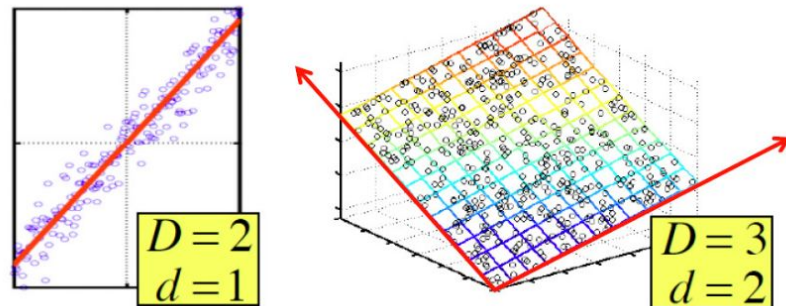
Dimensionality reduction

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.

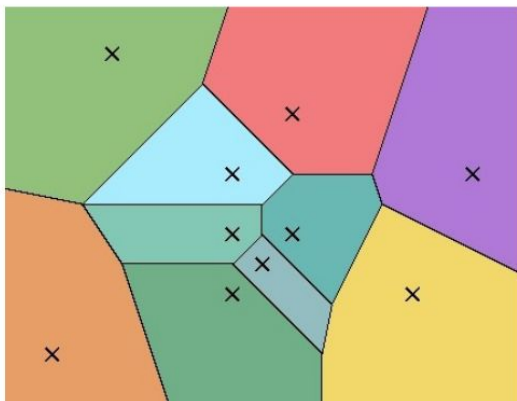
Much alike a color histogram!

Quantization

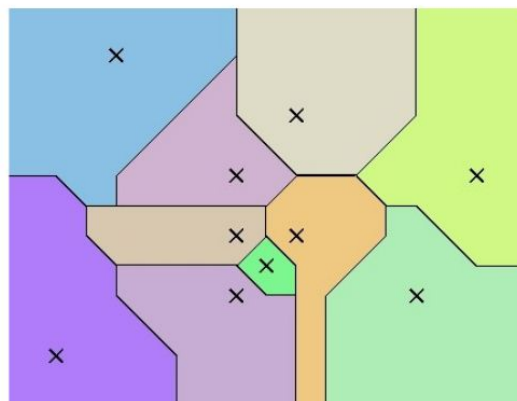
Discretization of some signal — **Lossy process!**

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

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.

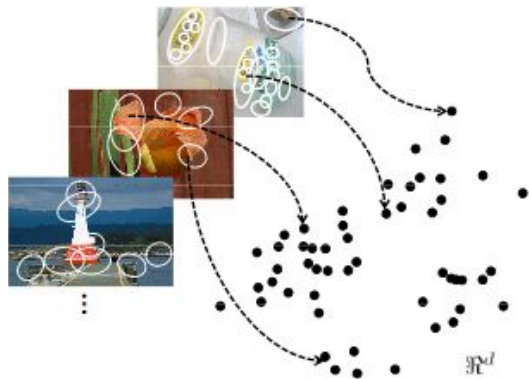
Build a histogram of descriptor count for each cluster.

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

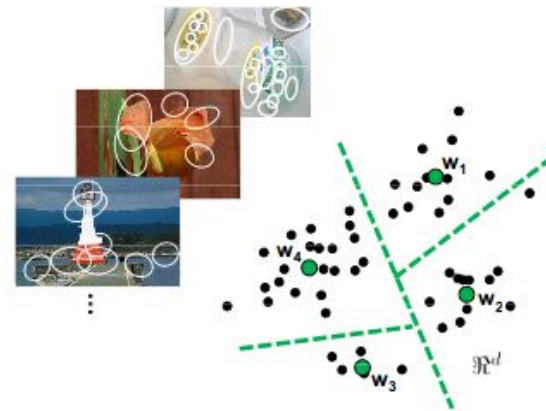
We can choose the number of words!



Codebook
creation

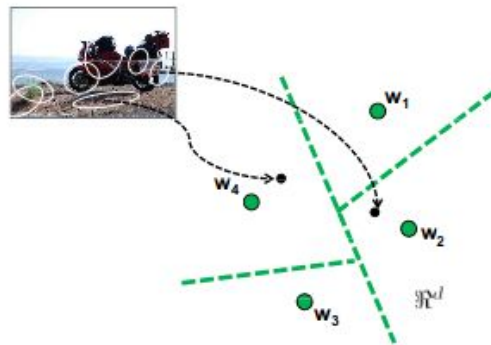


(a)

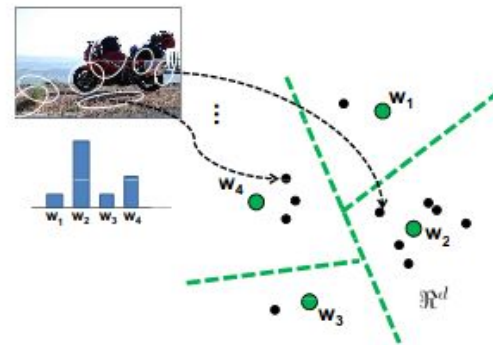


(b)

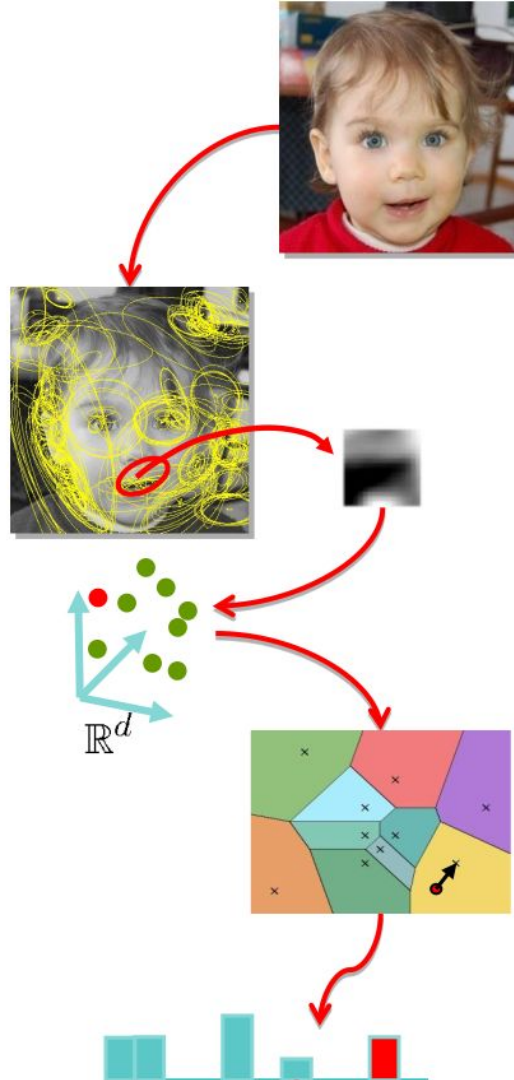
Indexing
process



(c)



(d)



Bag of Visual Words: Normalization

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 :

$$t_i = \frac{n_{id}}{n_d} \log\left(\frac{N}{n_i}\right)$$

The diagram illustrates the components of the TFIDF formula. Four boxes are connected to the formula by arrows:

- Top-left box: #occurences of word i in doc/img d (points to n_{id})
- Top-right box: Number of doc. In the database (points to N)
- Bottom-left box: Number of words in d (points to n_d)
- Bottom-right box: #occurences of word i in database (points to n_i)

Bag of Visual Words: Vector size

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

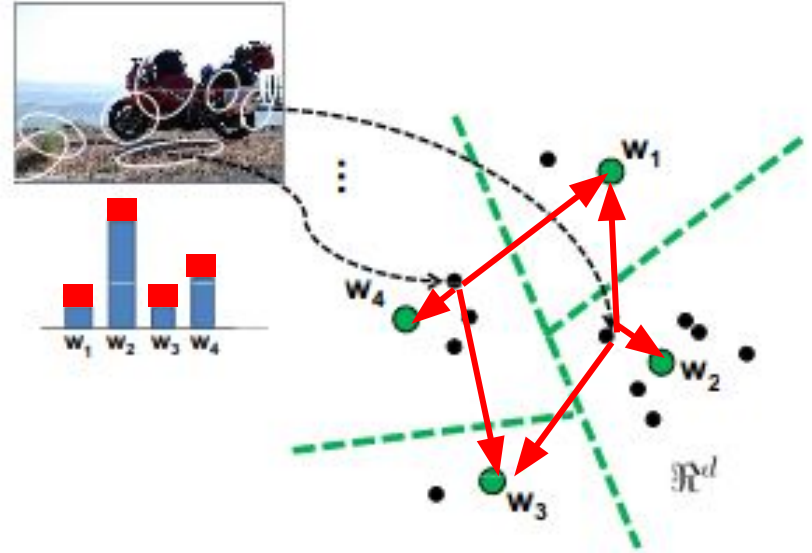
$$D = \text{vocabulary size}$$

Usually, the bigger the vocabulary the better the results.

Several thousands of words are common.

Variant: Soft BoVW

Use soft assignment to clusters,
add counts to neighbor bins

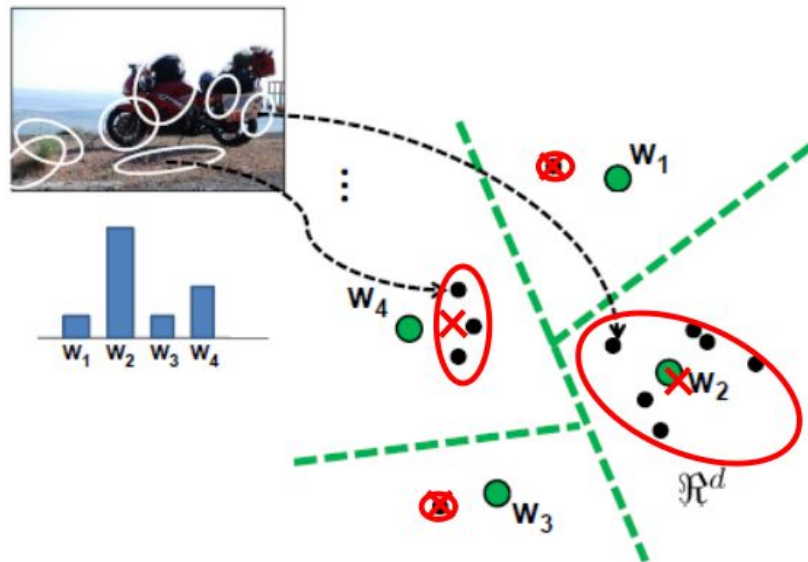


Other variants

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 $X = \{x_t, t = 1 \dots T\}$:

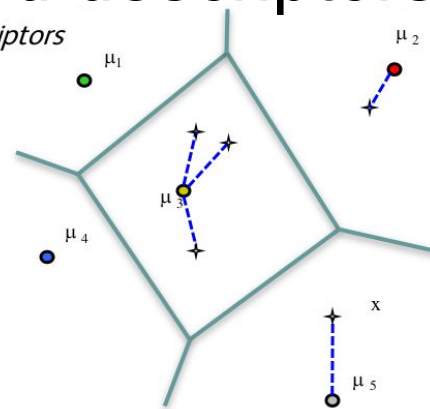
- ① assign: $\text{NN}(x_t) = \arg \min_{\mu_i} \|x_t - \mu_i\|$

- ②③ compute: $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$

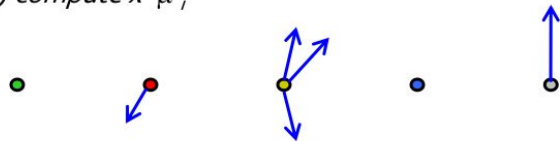
- concatenate v_i 's + ℓ_2 normalize

→ the VLAD is $D \times N$ dimensional

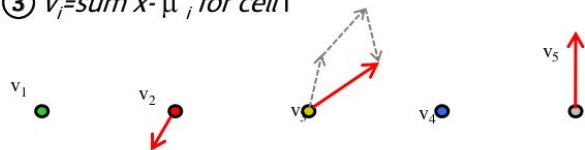
① assign descriptors



② compute $x - \mu_i$

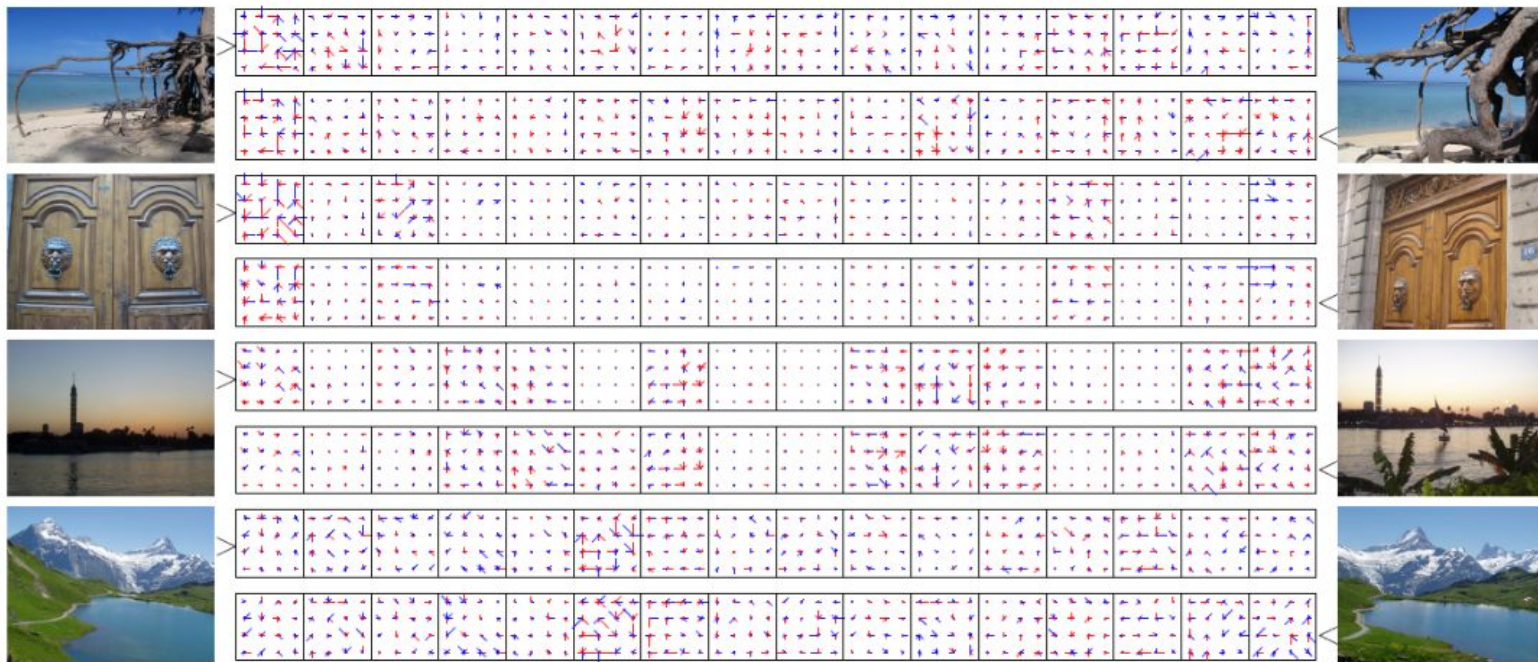


③ $v_i = \sum x - \mu_i$ for cell i



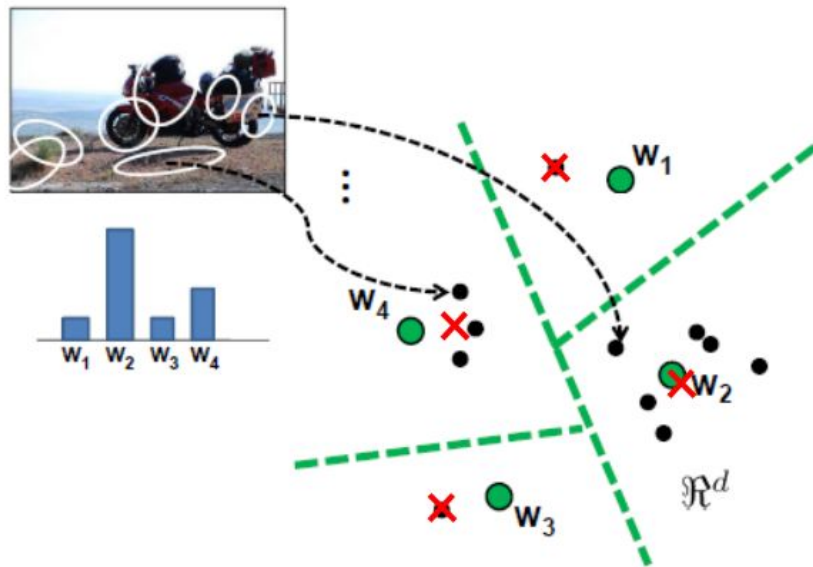
VLAD: vector of locally aggregated descriptors

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



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:

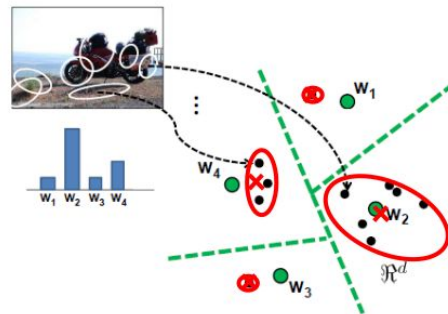
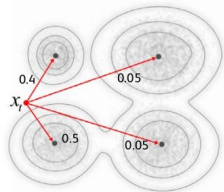
$$G_{\lambda}^X = \frac{1}{T} \sum_{t=1}^T \nabla_{\lambda} \log u_{\lambda}(x_t)$$

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

The FV is typically $2xD \times N$ dimensional

With respect to the VLAD:

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

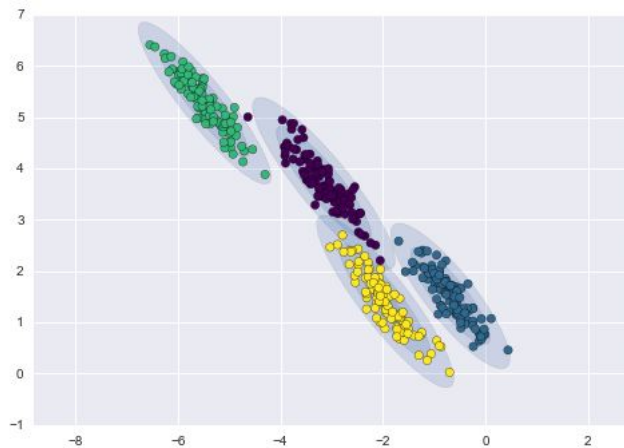
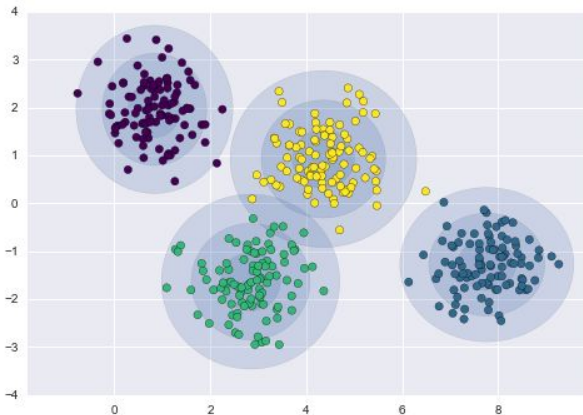


GMM

Probability that \mathbf{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 $f(z) = \text{sign}(z)|z|^\alpha$ 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.

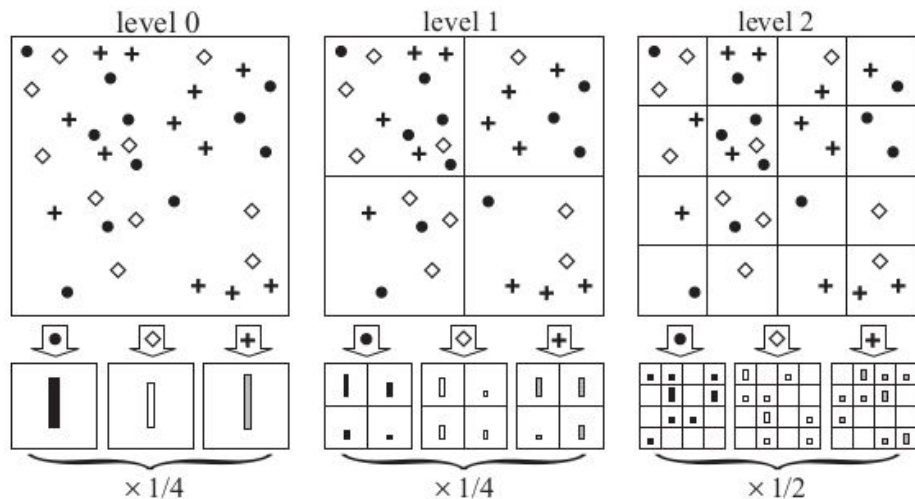
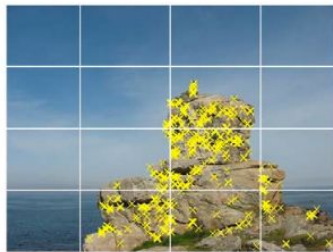


Figure 1. Toy example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall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

Compact the descriptors

Use LSH for binarization

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

Compute a binary code $\mathbf{b}(\mathbf{x})$ from vector \mathbf{x} as

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

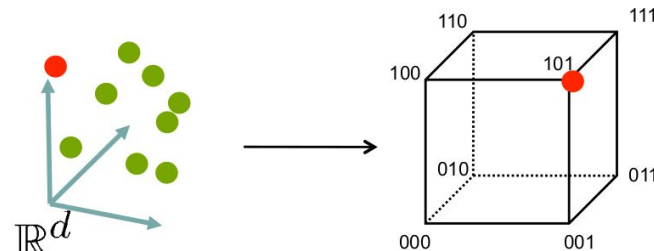
$$b(x) = (b_1(x), \dots, b_B(x))$$

⇒ very compact

⇒ very fast comparison (Hamming distance, CPU accel.)

⇒ suitable for very high-dimensional vectors

BUT linear search



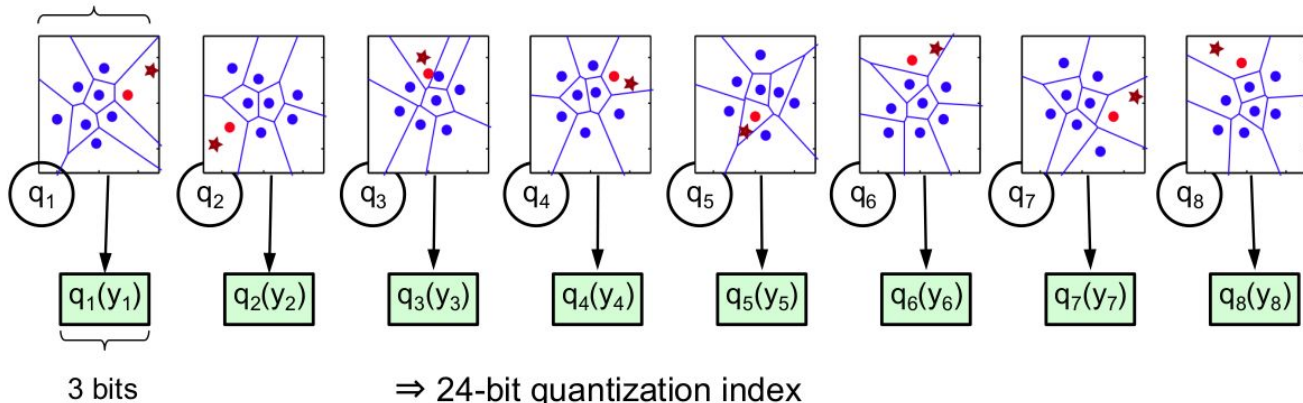
Product Quantizer

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

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

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

y_1 : 2 components



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



Limitations

Limitations of the BoVW approach

The process organizes the representation based on descriptor density: some features / dimensions are meaningful and some are not.

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 triplet loss in particular).