Multimedia Indexing and Retrieval

Visual content representation and retrieval

Georges Quénot

Multimedia Information Modeling and Retrieval Group

Laboratory of Informatics of Grenoble

Outline

- Introduction
- Query by example versus search
- Descriptors
- Classification, fusion, post-processing ...
- Conclusion

Introduction

Multimedia Retrieval

- User need → retrieved documents
- Images, audio, video
- Retrieval of full documents or passages (e.g. shots)

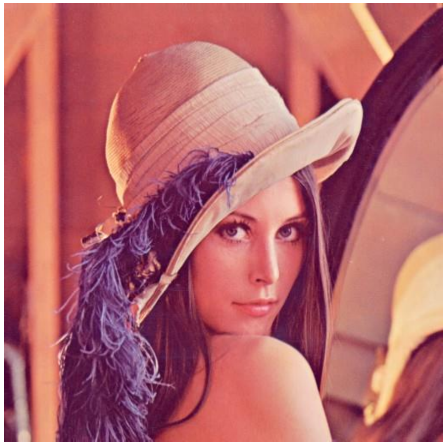
- Search paradigms:
  - Surrounding text → may be missing, inaccurate or incomplete
  - Query by example → need for what you are precisely looking for
  - Content based search (using keywords or concepts) → need for *content-based indexing* → “semantic gap problem”
  - Combinations including feedback

- Need for specific interfaces

The “semantic gap”

“... the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation” [Smeulders et al., 2002].

The “semantic gap” problem



Face

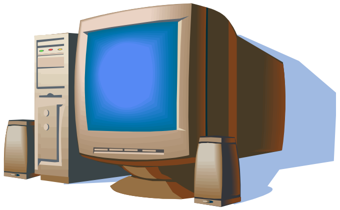
Woman

Hat

Lena

…

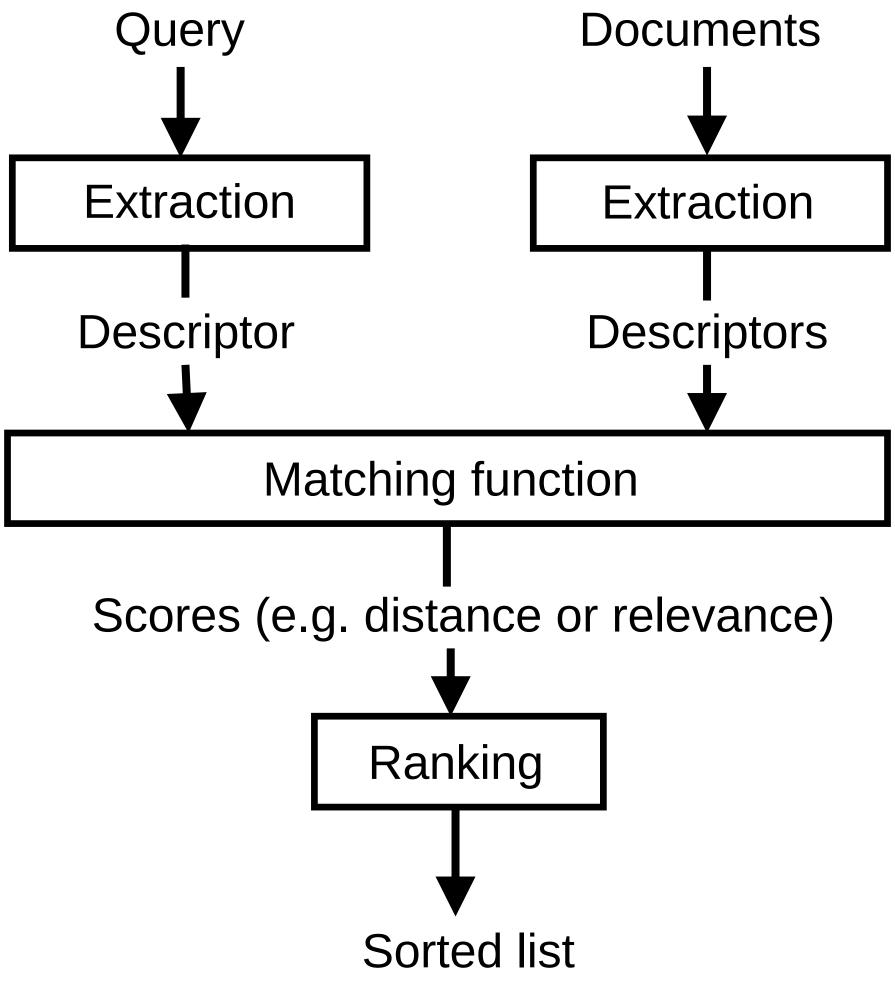
122	112	98	85	…
126	116	102	89	…
131	121	106	95	…
134	125	110	99	…
…	…	…	…	…



?

Retrieval (query by examples) versus indexing (for enabling query by key words / concepts)

Query BY Example (QBE)



Query

Documents

Extraction

Extraction

Descriptor

Descriptors

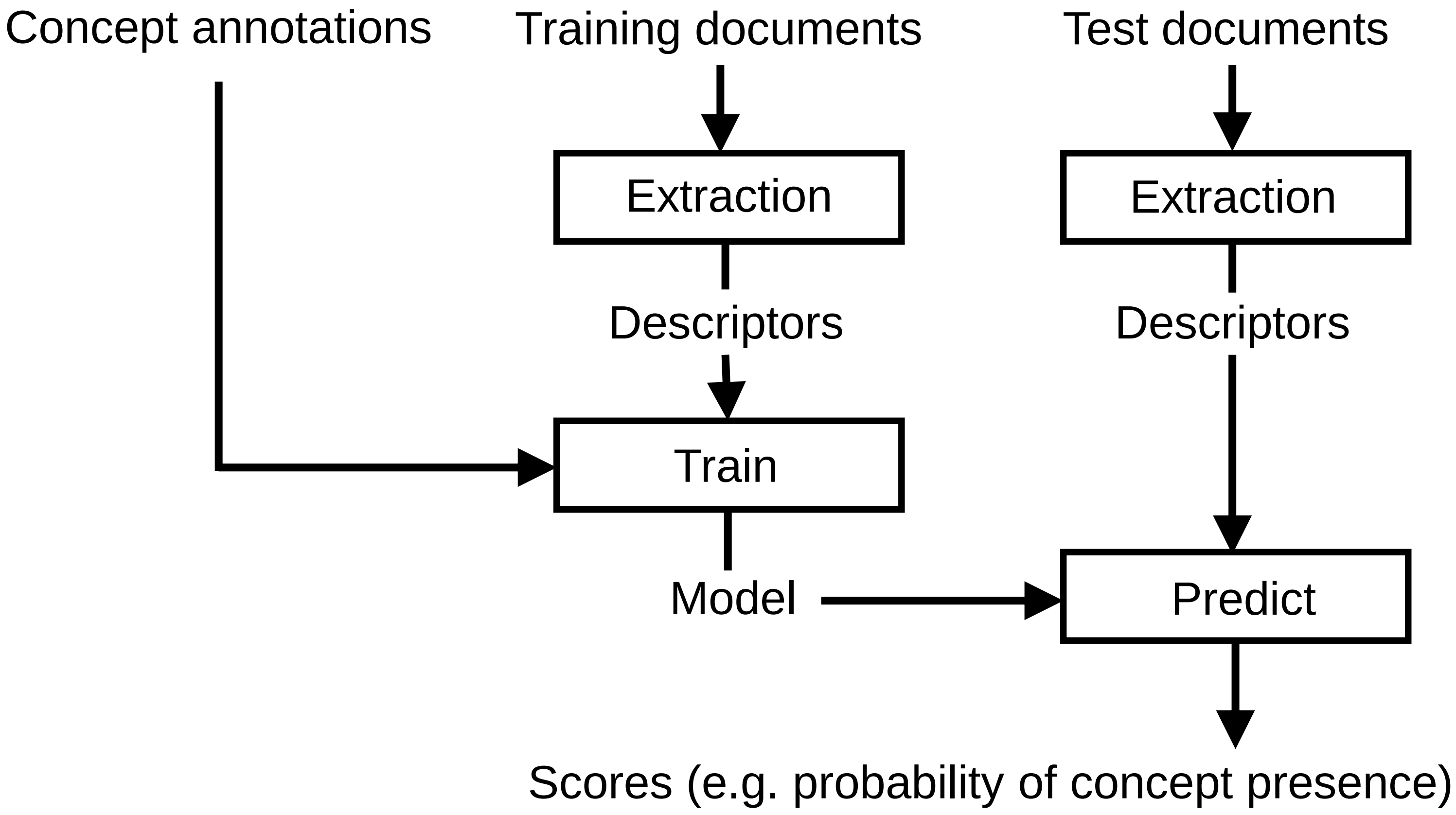
Matching function

Scores (e.g. distance or relevance)

Ranking

Sorted list

Content based indexing by supervised learning



Concept annotations

Training documents

Test documents

Extraction

Extraction

Descriptors

Descriptors

Train

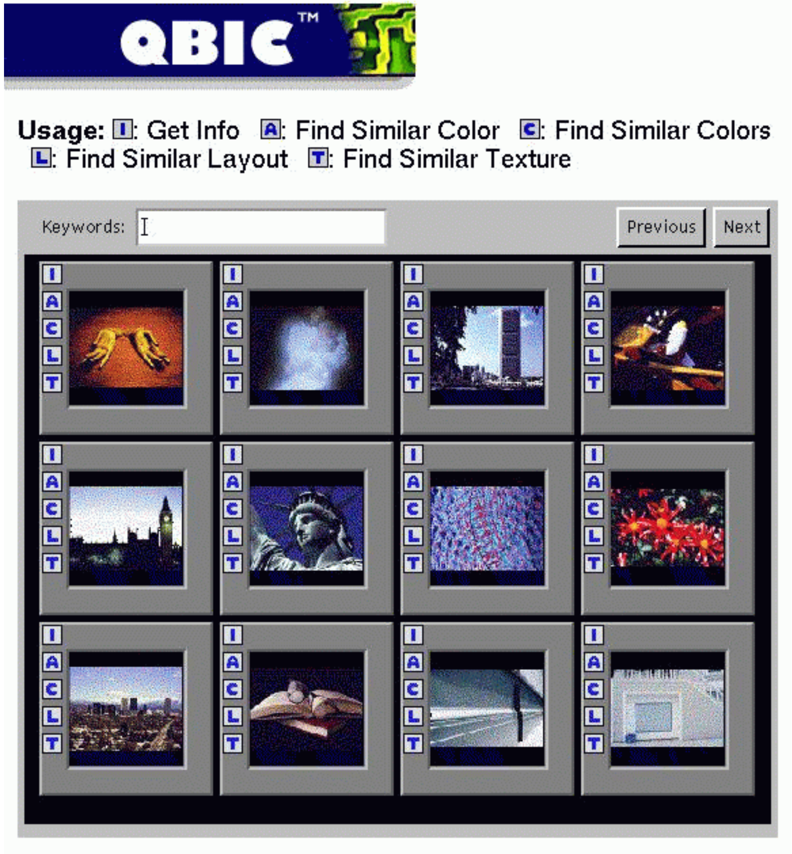
Model

Predict

Scores (e.g. probability of concept presence)

Example : the QBIC system

- Query By Image Content, IBM (stopped demo)
  `http://wwwqbic.almaden.ibm.com/cgi-bin/photo-demo`



Usage: I: Get Info A: Find Similar Color C: Find Similar Colors L: Find Similar Layout T: Find Similar Texture

Descriptors

Descriptors

- Engineered descriptors
  - Color
  - Texture
  - Shape
  - Points of interest
  - Motion
  - Semantic
  - Local versus global
  - …
- Learned descriptors
  - Deep learning
  - Auto encoders
  - …

Histograms - general form

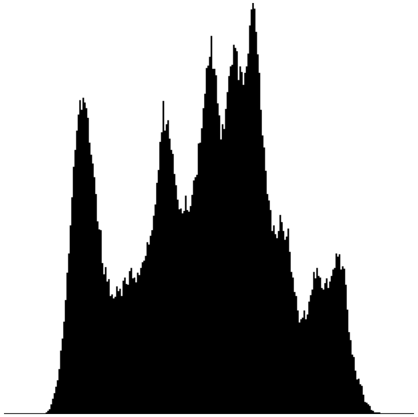
- A fixed set of *disjoint categories* (or *bins*), numbered from 1 to *K*.
- A set of *observations* that fall into these categories
- The histogram is the vector of *K* values *h*[*k*] with *h*[*k*] corresponding to the number of observations that fell into the category *k*.
- By default, the *h*[*k*] are integer values but they can also be turned into real numbers and normalized so that the *h* vector length is equal to 1 considering either the L_1 or L_2 norm
- Histograms can be computed for several sets of observations using the same set of categories producing one vector of values for each input set

Histograms – text example

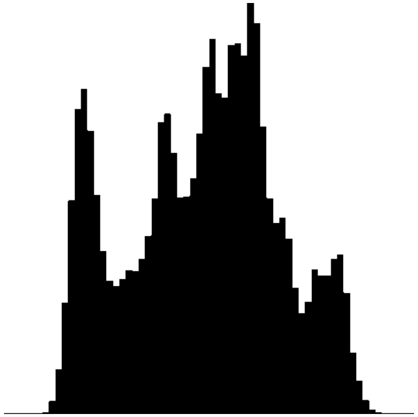
- A vector of term frequencies (tf) is an histogram
- The categories are the index terms
- The observations are the terms in the documents that are also in the index
- A tf.idf representation corresponds to a weighting of the bins, less relevant in multimedia since histograms bins are more symmetrical by construction (e.g. built by K-means partitioning)

Image intensity histogram

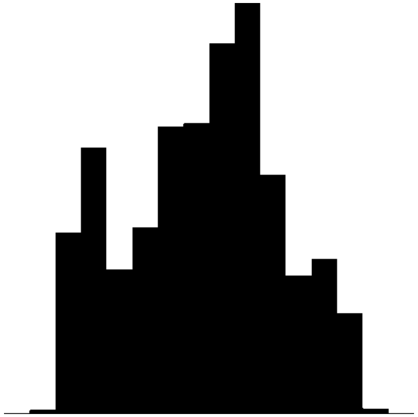
- The set of categories are the possible intensity values with 8-bit coding, ranging from 0 (black) to 255 (white) or ranges of these intensity values



256-bin



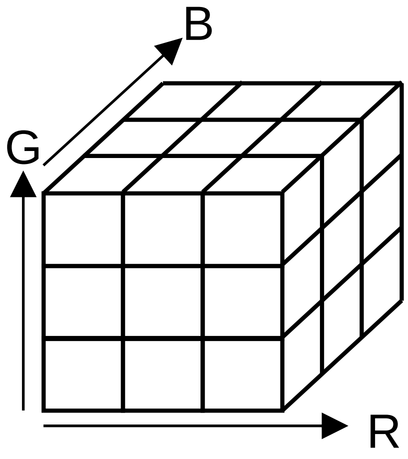
64-bin



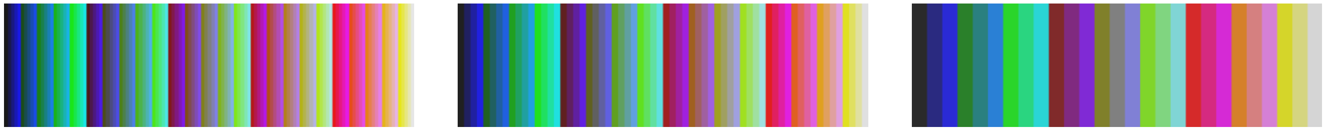
16-bin

Image color histogram

- The set of categories are ranges of possible color values
- A common choice is a per component decomposition resulting in a set of parallelepipeds



Representations with the parallelepipeds' center colors:



5×5×5-bin
125-bin

4×4×4-bin
64-bin

3×3×3-bin
27-bin

- Any color space can be chosen (YUV, HSV, LAB …)
- Any number of bins can be chosen for each dimension
- The partition does not need to be in parallelepipeds

Image color histogram

- The set of categories are ranges of possible color values

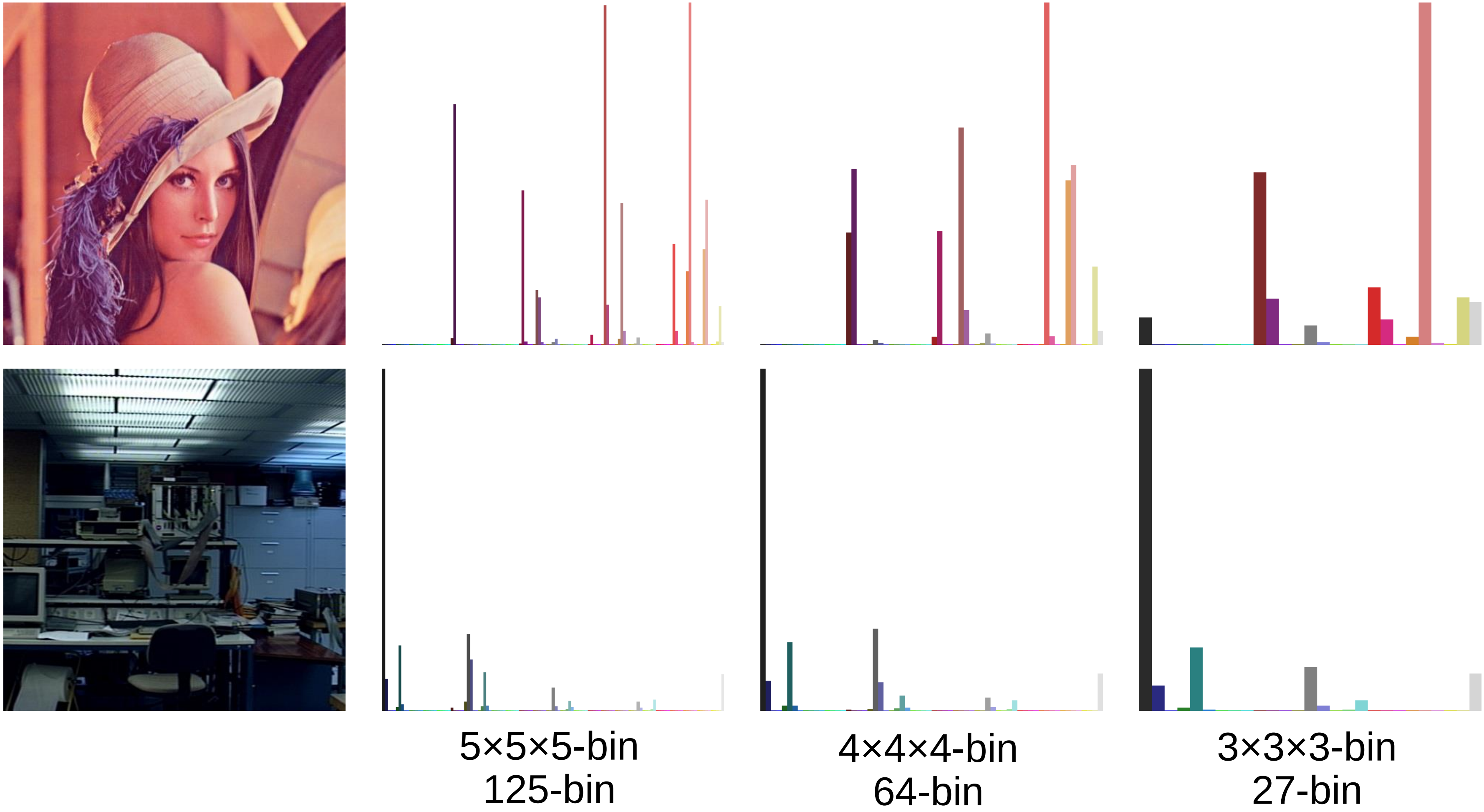


Image histograms

- Rather invariant to image size if normalized to unit vector length with L_1 or L_2 norm
- Rather invariant to content displacements or symmetries
- NOT invariant to illuminations changes, gain and offset normalization may be needed
- Histograms are distributions, better compared using a $\chi 2$ distance that Euclidean one:

$$d(x,y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$$

- Earth Mover Distance (EMD) can be even better
- Alternatively, taking the square root of the histogram elements can make the Euclidean distance suitable

Image histograms

- Can be computed on the whole image,
- Can be computed by blocks:
  - One (mono or multidimensional) histogram per image block,
  - The descriptor is the concatenation of the histograms of the different blocks.
  - Typically : 4×4 complementary blocks but non symmetrical and/or non complementary choices are also possible. For instance: 2×2 + 1×3 + 1×1
- Size problem → only a few bins per dimension or a lot of bins in total

Fuzzy histograms

- Objective: smooth the quantization effect associated to the large size of bins (typically 4×4×4 for RGB).
- Principle: split the accumulated value into two adjacent bins according to the distance to the bin centers.

Color correlograms

- Parallelepipeds/bins are taken in the Cartesian product of the color space by itself : six components H(r1,g1,b1,r2,g2,b2) (or only four components if the color space is projected on only two dimensions: H(u1,v1,u2,v2)). [Huang et al, 1997]
- Bi-color values are taken according to a distribution of the image point couples:
  - At a given distance one from the other,
  - And/or in one or more given direction.
- Allows for representing *relative spatial relationships between colors*,
- Large data volumes and computations

Image normalization

- Objective : to become more robust again illumination changes before extracting the descriptors.
- Gain and offset normalization: enforce a mean and a variance value by applying the same affine transform to all the color components, non-linear variants.
- Histogram equalization: enforce an as flat as possible histogram for the luminance component by applying the same increasing and continuous function to all the color components.
- Color normalization: enforce a normalization which is similar to the one performed by the human visual: “global” and highly non linear.

Texture descriptors

- Computed on the luminance component only
- Frequential composition or local variability
- Fourier transforms
- Gabor filters
- Neuronal filters
- Co-occurrence matrices
- Normalization possible.

Gabor transforms

(Circular) Gabor filter of direction θ, of wavelength λ and of extension σ :

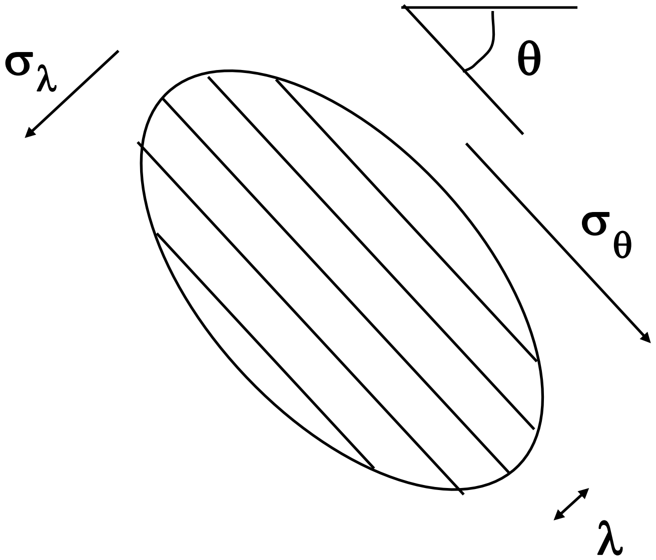
$$g(\sigma,\theta,\lambda,I,i,j) = \frac{1}{2\pi\sigma^2} \sum_{k,l} e^{-\left(\frac{k^2+l^2}{2\sigma^2}\right)}.e^{2\pi\mathbf{i}\left(\frac{k.cos\theta+l.sin\theta}{\lambda}\right)}.I(i+k,j+l)$$

Energy of the image through this filter:

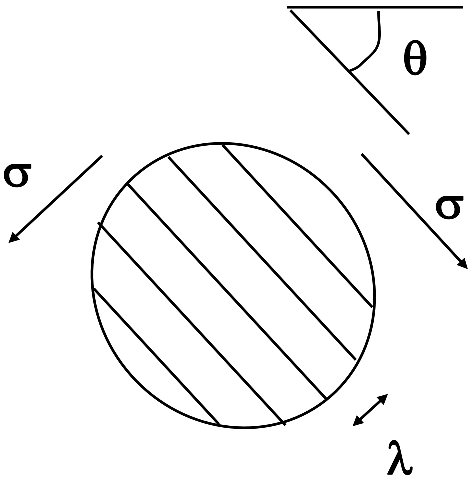
$$E_g(\sigma,\theta,\lambda,I)^2 = \frac{1}{N}\sum_{i,j} \mid g(\sigma,\theta,\lambda,I,i,j) \mid^2$$

Gabor transforms

Elliptic:

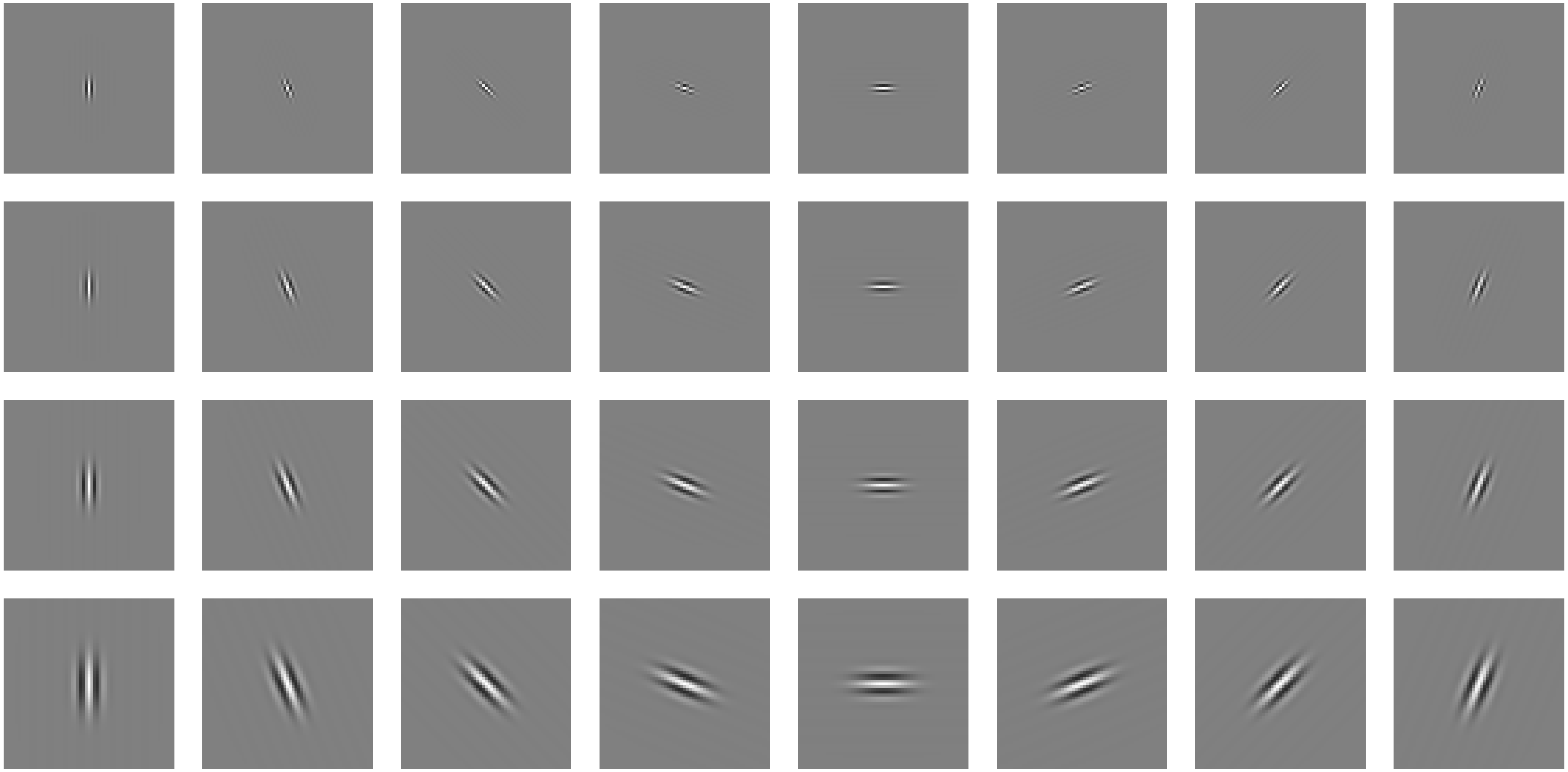


Circular:



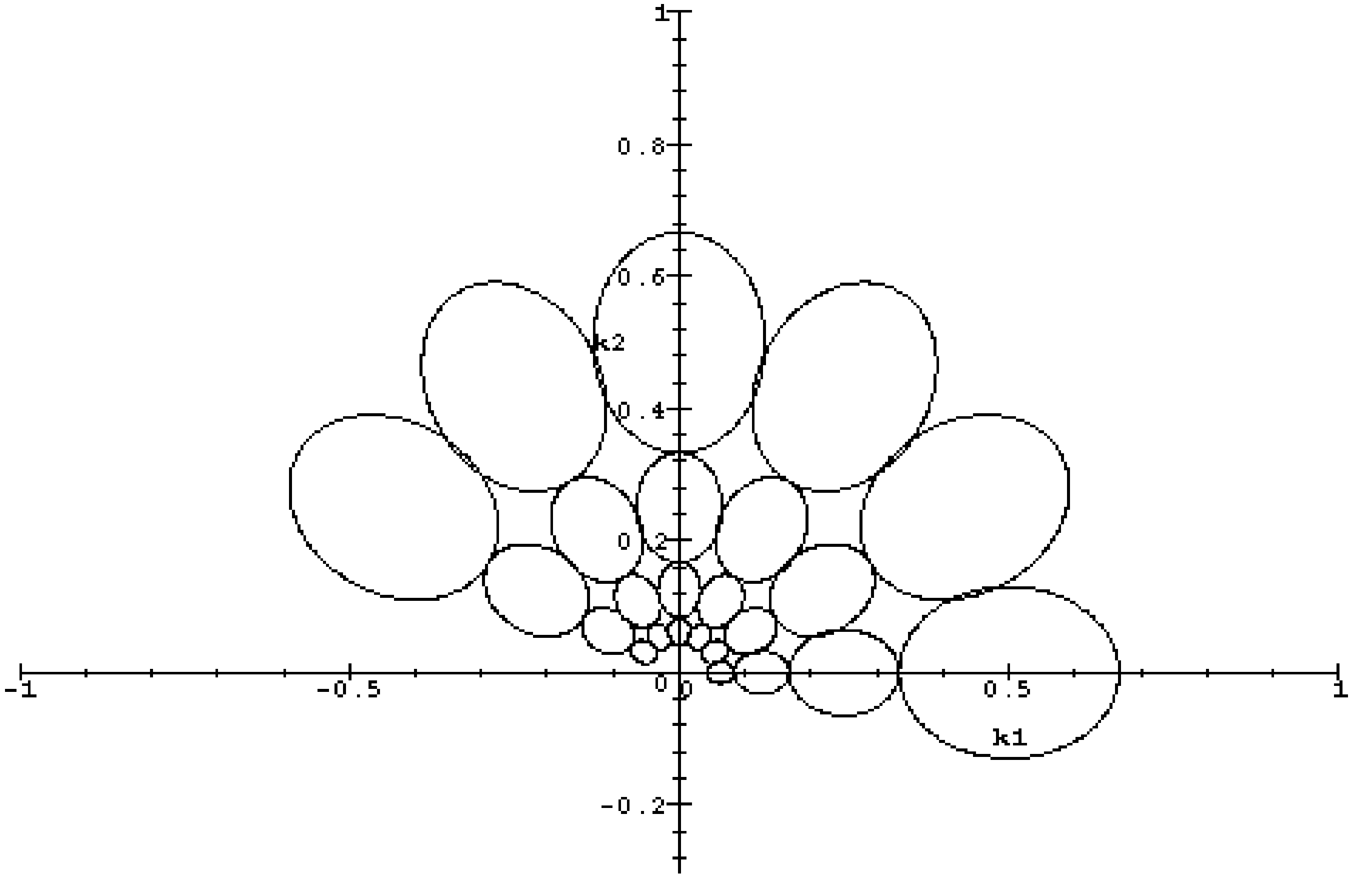
Gabor Filters

Example of elliptic filters with 8 orientations and 4 scales



Gabor filters in Fourier space

Elliptic filters with 6 orientations and 4 scales in the frequential domain (Fourier space)



Gabor transforms

- Circular:
  - scale λ, angle θ, variance σ,
  - σ multiple of λ, typically : $\sigma = 1.25\ \lambda$,

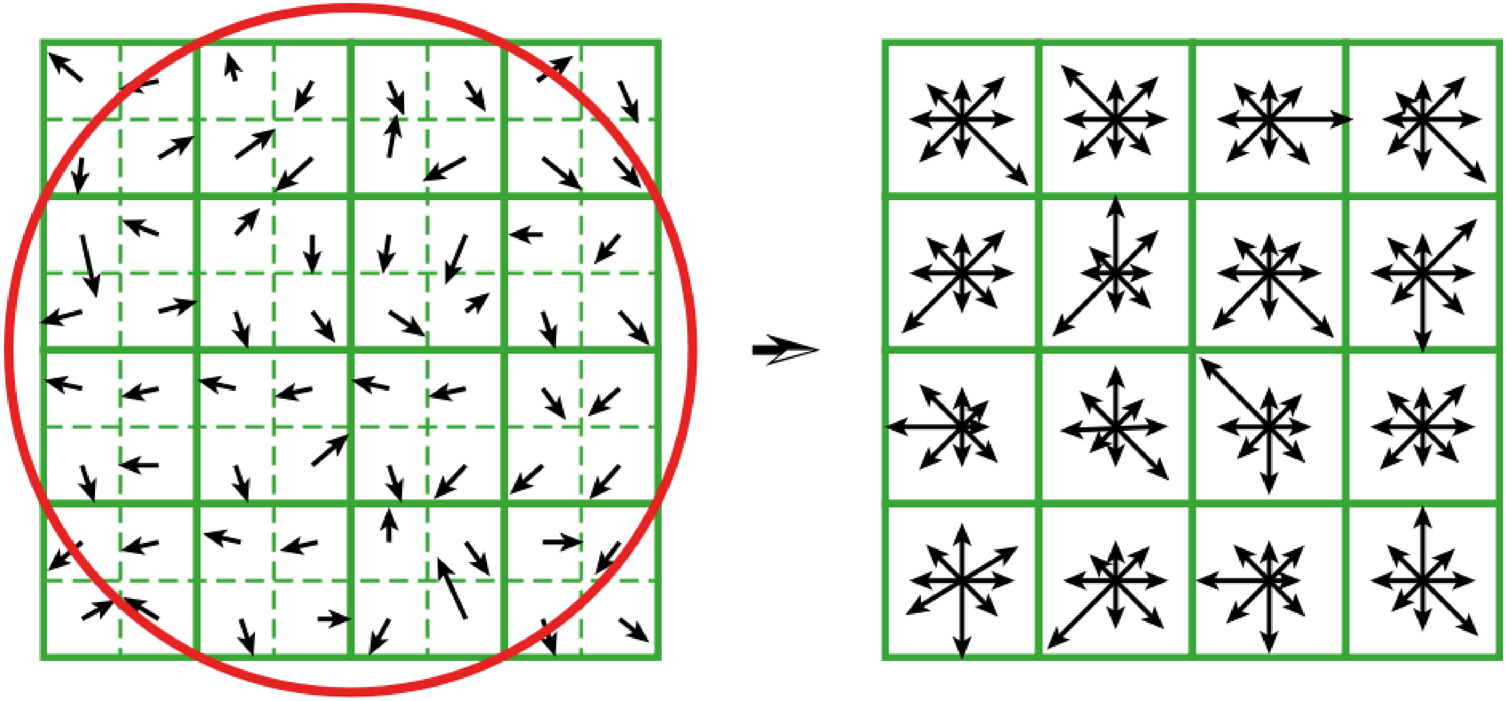
    ("same number" of wavelength whatever the λ value)
- Elliptic:
  - scale λ, angle θ, variances σ_λ and $\sigma_{\theta,}$
  - σ_λ and σ_θ multiples of λ, typically : $\sigma_\lambda = 0.8\ \lambda$ et $\sigma_\theta = 1.6\ \lambda$,
- 2 independent variables:
  - scale λ : N values (typically 4 to 8) on a logarithmic scale (typical ratio of $\sqrt{2}$ to 2)
  - angle θ : P values (typically 8),
  - $N.P$ elements in the descriptor,

Selection of points of interest

- “High curvature” points or “corners”,
- “Singular” points of the I[i][j] surface,
- Extracted using various filters:
  - Computation of the spatial derivatives at several scales,
  - Convolution with derivatives of Gaussians,
  - Harris-Laplace detector.
- Interest points are selected, filtered and described
- 2D (image): Scale Invariant Feature Transform (SIFT) [Lowe, 2004]
- 3D (video): Space-Time Interest Points (STIP) [Laptev, 2005]
- Variable number of points per image or per video shot → need for aggregation

Descriptors of points of interest

- SIFT descriptor: Histogram of gradient direction:
  8 bins times 4 x 4 blocks in a neighborhood of the point.
- Neighborhoods are scaled according to the detector output



Local versus global descriptors

- Global descriptors: single vector for a whole image
- Local descriptors: one vector for each pixel, image patch, image block shot 3D patch … e.g. SIFT or STIP
- Need for a single vector of fixed length far any image and with comparable components across images
- *Aggregation* of local descriptors → global descriptor

Aggregation of local descriptors

- Building of a single global descriptor
- Homogeneous with the local descriptor:
  - max or average pooling
- Heterogeneous with the local descriptor:
  - Histogramming according to clusters in the local descriptor space [Sivic, 2003][Cusrka, 2004]
  - Gaussian Mixture Models (GMM)
  - Fisher Vectors (FV) [Perronnin, 2006],Vectors of Locally Aggregated Descriptors (VLAD) [Jégou, 2010] or Tensors (VLAT) [Gosselin, 2011], Supervector

Clustering

- Given a set (x_i) of N data points in a metric space
- Find a set (c_j) of K centers
- Minimizing the representation square error:

$$E = \sum_i \left(\min_j \left(d(x_i, c_j)^2 \right) \right)$$

- Direct search not possible
- Use heuristics for finding good local minima
- Cluster j = subset (part) of the data space which is closest to center c_j than to any other center
- The set of clusters is a partition of the data space
- This partition is *adapted* to the training data

K-means Clustering

- Given a set (x_i) of N data points in a metric space
- Randomly select a set (c_j) of K centers
- Repeat until convergence (no change in centers):
  - for each x_i data point, $i = 1 \ldots N$:
    - find the nearest center $c_j \; : \; j = \arg\min d(x_i, c_k)$
    - assign the x_i data point to the cluster j $x_i \rightarrow c_j$
  - for each cluster, $j = 1 \ldots K$:
    - set the new center c_j as the mean of all x_i data point previously assigned to the cluster j :
or to a random value if no data point is assigned

$$c_j = \frac{\sum_{x_i \rightarrow c_j} x_i}{\sum_{x_i \rightarrow c_j} 1}$$

- Complexity: O(#iterations × #clusters × #points × #dimensions)

K-means Clustering

- K-means is relatively fast and efficient compared to alternate and more complex methods
- The final result depends upon the choice of the initial centers; it is always possible to run it many times with different initial conditions and select the one obtaining the smallest representation error
- Tends do produce clusters of comparable size
- Convergence is guaranteed but it may take a large number of iterations and only a local minimum is guaranteed
- For practical applications, a full convergence is not necessary and does not make a big difference

Hierarchical K-means Clustering

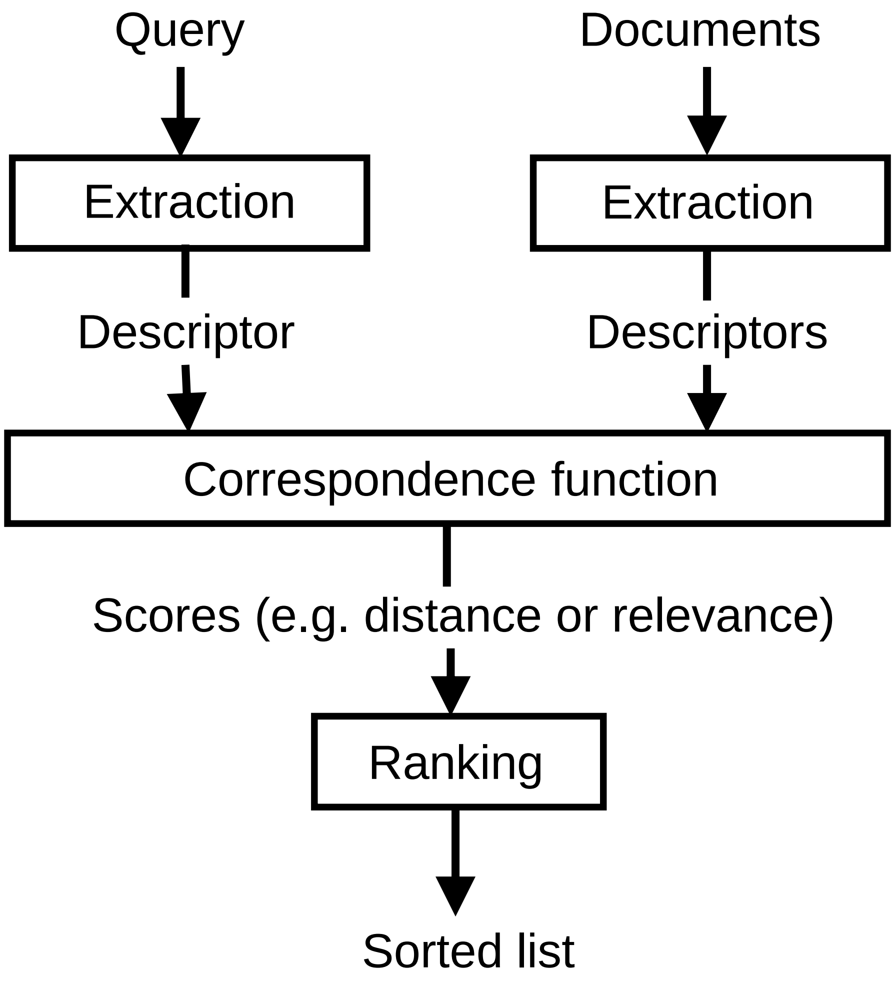
- Hierarchical K means may be faster (both for the clustering and the mapping) but less accurate
- The hierarchical structure of the set of clusters may be useful for some applications
- Two main strategies:
  - Recursively split all the clusters into a (small) fixed number of sub-clusters (e.g. recursive dichotomy) starting with a single cluster (→ regular n-ary tree)
  - Recursively split in two parts only the biggest cluster into sub-clusters (→ irregular binary tree)
- Hierarchical mapping: recursive search of the closest center from the coarsest to the finest grain.

Semantic or intermediate descriptors

- Use of classifiers trained on other data and for other target concepts [Ayache, 2007]
- Vectors of scores of the other target concepts can be used as intermediate or high level descriptors (opposed to low-level ones that are “close to the signal”)
- Semantic descriptors can be either global or local (e.g. on pixels or patches)
- Semantic descriptors carry different information than low-level one and of higher semantic value
- The target concepts composing the semantic descriptors does not need to be related to the final target ones
- They do not need either to be recognized very accurately
- Semantic descriptors are often as good as or better than state of the art low-level ones and boost performance when combined with them

Retrieval, indexing and fusion

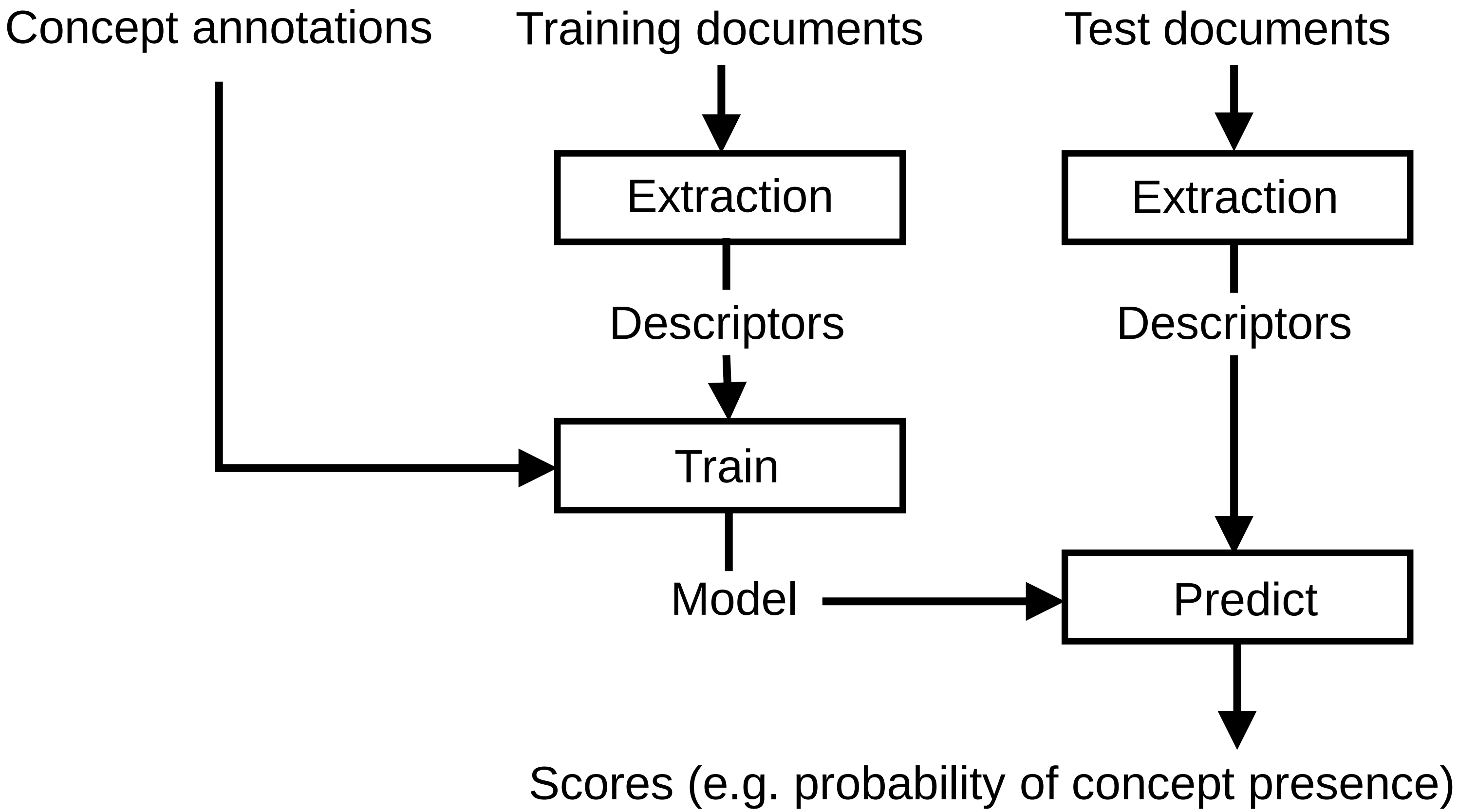
Query BY Example



Query by example

- Single query sample:
  - χ^2, EMD or histogram intersection for histograms
  - Euclidian Distance : searching for identities
  - Angle between vectors : searching for similarities robust to illumination changes (for some other descriptors, e.g. Gabor transforms)
- Multiple queries or relevance feedback:
  - Linear combination of distances with different weights for positively and negatively marked samples [Rocchio, 1971]
  - Supervised learning from the marked samples (active learning)
  - Rely also on the choice of a distance between global descriptions
- Direct matching and scoring between sets of local descriptors:
  - Costly but good for searching specific instances rather than general categories

Content based indexing by supervised learning



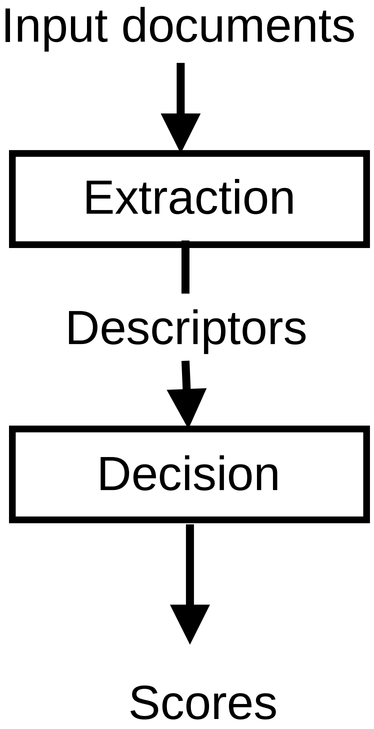
Content based indexing by supervised learning

- Training from annotated collections:
  - LSCOM-TRECVid for videos
  - Pascal VOC or ImageNet for still images
  - Many others, e.g. Hollywood2 for actions in movies
- Use of supervised learning methods:
  - Support Vector Machines (SVM), linear or RBF
  - K nearest neighbors (KNN)
  - Neural Networks (NN), Multi-Layer Perceptrons (MLP)
  - Many others again
  - Adaptations for highly imbalanced data sets
- Fusion if several descriptors and/or several learning methods are simultaneously used.

Fusion

- Several possible descriptors
- Several possible classifiers or correspondence functions
- Early versus late fusion [Snoek, 2005]
  - Early: concatenation of normalized descriptors
  - Late: combination of classification scores
- Kernel fusion [Ayache, 2007]
  - Fusion of kernels in RBF-based (e.g. SVM) learning methods
- These fusion methods are also applicable to query by example

Common processing, single descriptor



Input documents

Extraction

Descriptors

Decision

Scores

Query,
search collection,
training collection,
test collection …

Color, texture,
bag of SIFTs …

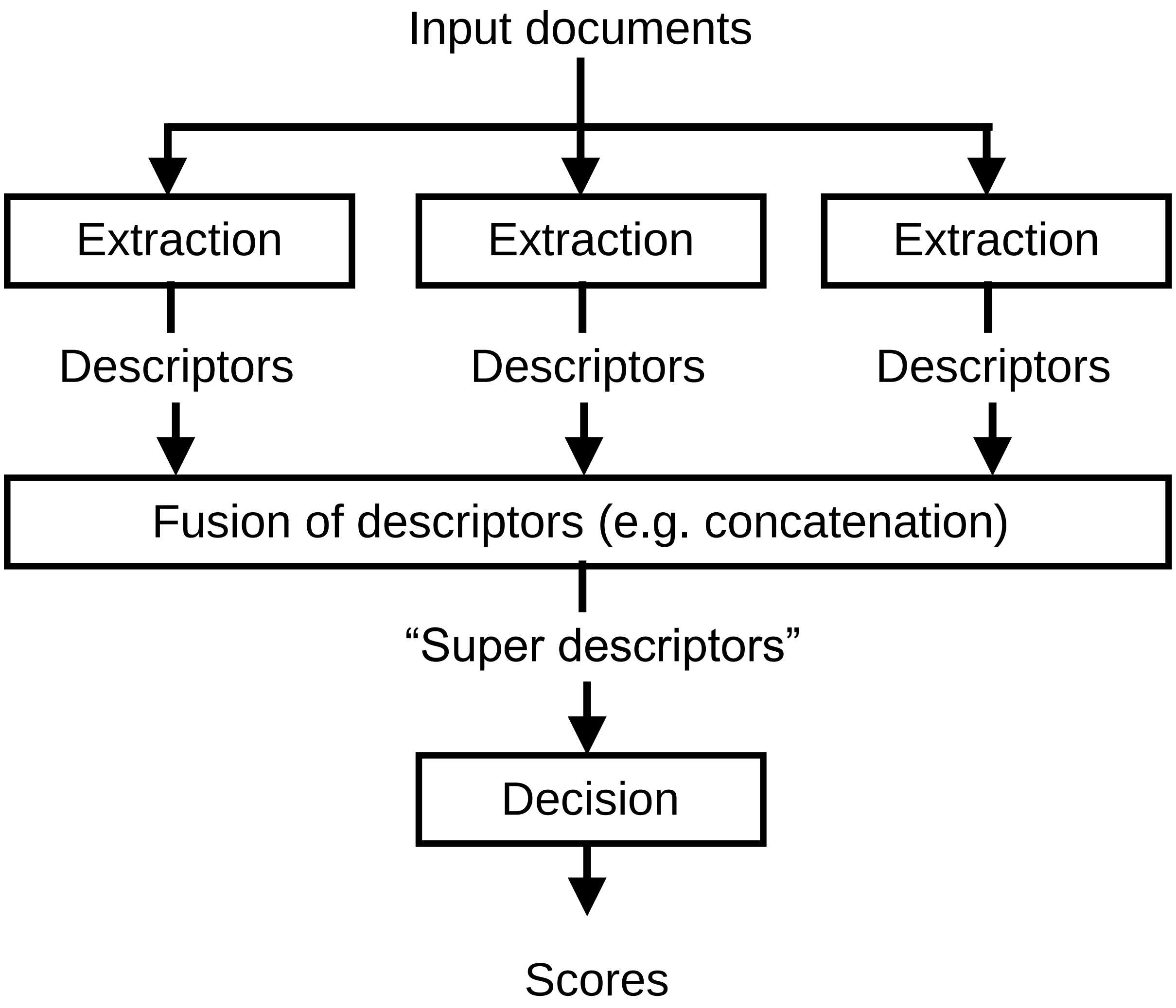
Correspondence function,
train / predict

Similarity measure,
probability of presence

Fusion of representations (early)

- For all vector description (of fixed size), whatever their origin,
- Possibility to concatenate the various descriptors in a unique mixed descriptor → normalization problem,
- Possibility to reduce la dimension of the resulting vector (and/or of each original vector) in order to keep only the most relevant information:
  - Principal Component Analysis,
  - Neural networks (auto-encoders),
  - Learning is needed (representative data and process).
- Less information, faster once learning is done,
- Euclidean distance on the shortened vector.

Common processing, multiple descriptors, single decision (early fusion)



Input documents

Extraction

Extraction

Extraction

Descriptors

Descriptors

Descriptors

Fusion of descriptors (e.g. concatenation)

“Super descriptors”

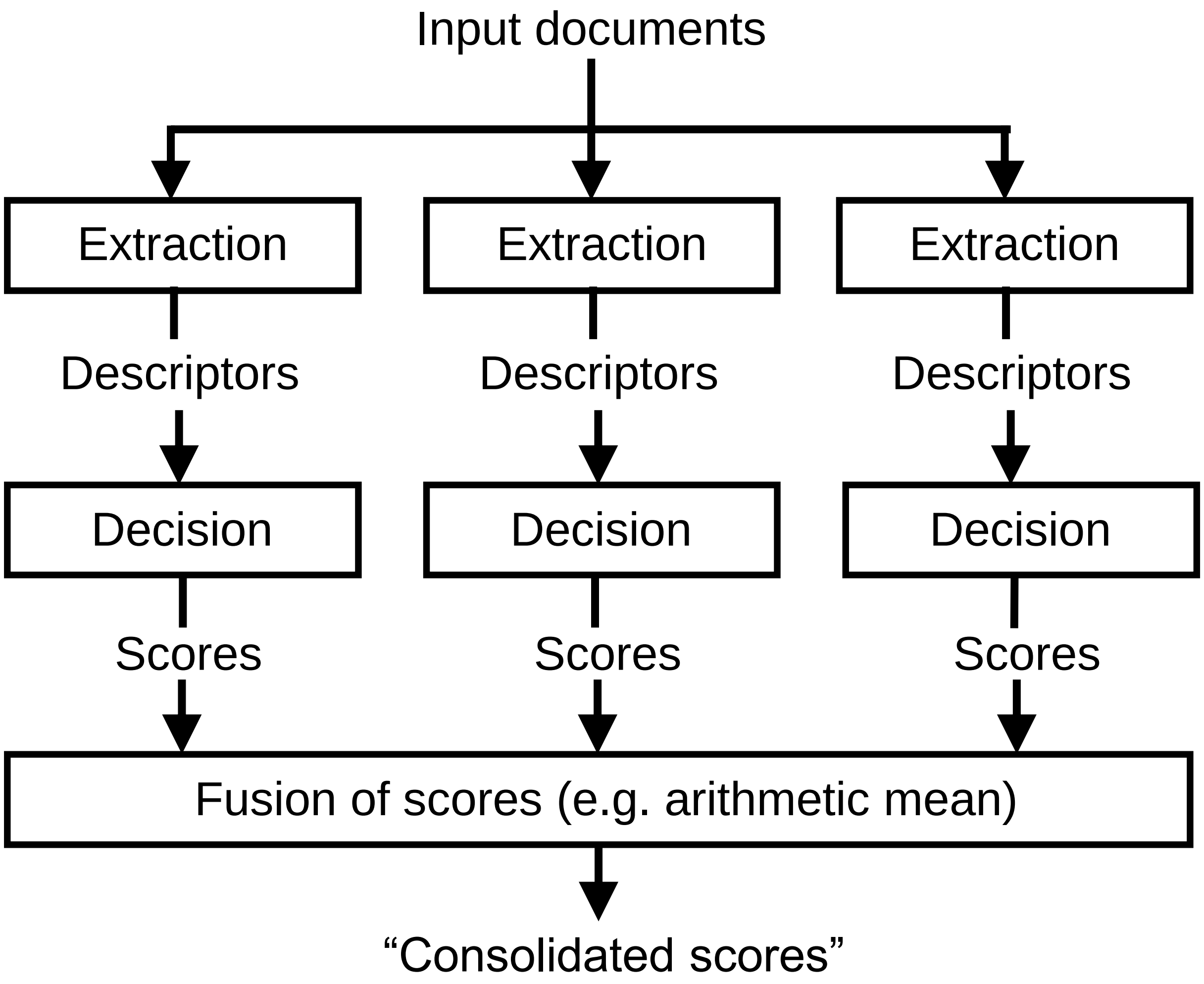
Decision

Scores

Fusion of scores (late)

- Each correspondence function generally produces a quantitative value that estimate a similarity (QBE case)
- It is always possible to come to the case in which the values are between 0 and 1 and represent a relevance
- In order to fuse the results from several functions, we may use :
  - A weighted sum,
  - A weighted product (weighted sum on the ogarithms),
  - The minimum value,
  - A classifier (SVM, neural network, …)
- Problem for the choice of the weights and/or for the classifier training.

Common processing, multiple descriptors, multiple decision (late fusion)


Input documents

Extraction

Extraction

Extraction

Descriptors

Descriptors

Descriptors

Decision

Decision

Decision

Scores

Scores

Scores

Fusion of scores (e.g. arithmetic mean)

“Consolidated scores”

Re-ranking for concept classification

- Re-ranking (or re-scoring): use of detections scores for other concepts of for other samples for improving the detection of a given concept for a given sample
- Temporal re-scoring [Safadi, 2010]
  - Re-score shots in a video with the hypothesis of a global or a local homogeneity of the contents
- Conceptual re-scoring [Hamadi, 2013]
  - Re-score an image or video sample for several concepts using implicit (co-occurrences) or explicit (ontologies) between them
- Combination of both

Conclusion

Search at the signal level: conclusion

- Representation by different types of descriptors and evaluation of relevance by various functions,
- A single type: results from poor to average,
- Several types simultaneously: results from average to good with possible domain adaptation
- Possibility to adjust the compromise quality - performance - general - size of the database
- Performance limited by the "analog" (not symbolic) aspect of representations
- High-level (semantic) descriptors greatly helps