GBGI9U07: multimedia document: description and automatic retrieval

1. Introduction, descriptors and correspondence

Georges Quénot

Multimedia Information Indexing and Retrieval Group

Laboratory of Informatics of Grenoble

January 2024

Outline

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

Introduction

Multimedia Retrieval

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

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

- Need for specific interfaces

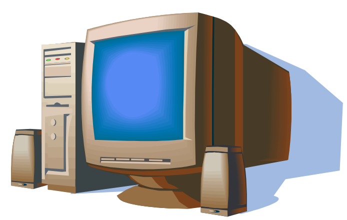
The “semantic gap”

“... 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	…
…	…	…	…	…



?

"Signal" level

- Signal :
  - Variable in time, in space and/or in other physical dimensions,
  - Analog : physical phenomenon (pressure of an acoustic wave or distribution of light intensity) or its modeling by another one (electronic or chemical for example),
  - Digital : same content but "discretized"
    - of the value,
    - of time,
    - of space,
    - and/or others (light frequency for example).

“Signal” level

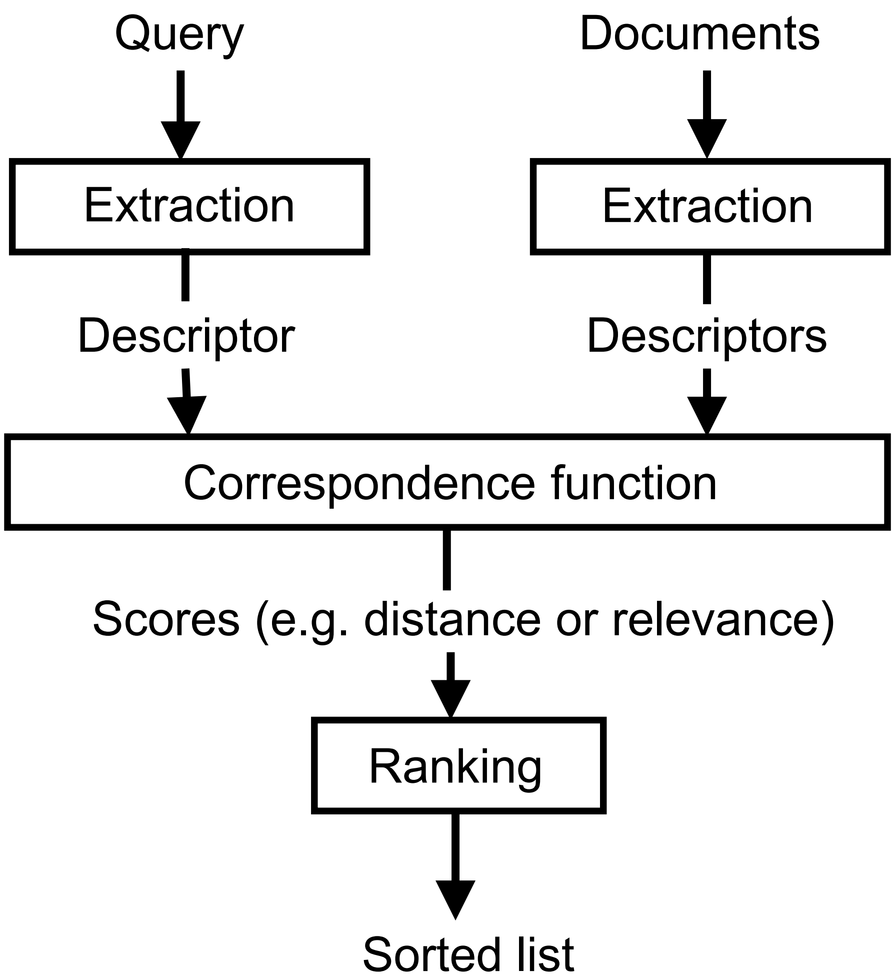
- Signal, examples :
  – Sound (monophonic) : values sampled at 16 kHz on 16 bits (one temporal dimension, zero spatial dimensions),
  – Still image (monochrome) : values sampled on a 2D grid on 8 bits (zero temporal dimension, two spatial dimensions; the spatial sampling frequency depends upon the sensor),
  – Stereo sound, color image: multiplication of the channels (additional dimension),
  – Video (image sequence): like still image fixe but additionally sampled in time (24-30 Hz; one temporal dimension, two spatial dimensions, one chromatic dimension),
  – Images 3D (scanners), 3D sequences, …

“Signal” and “semantic” levels

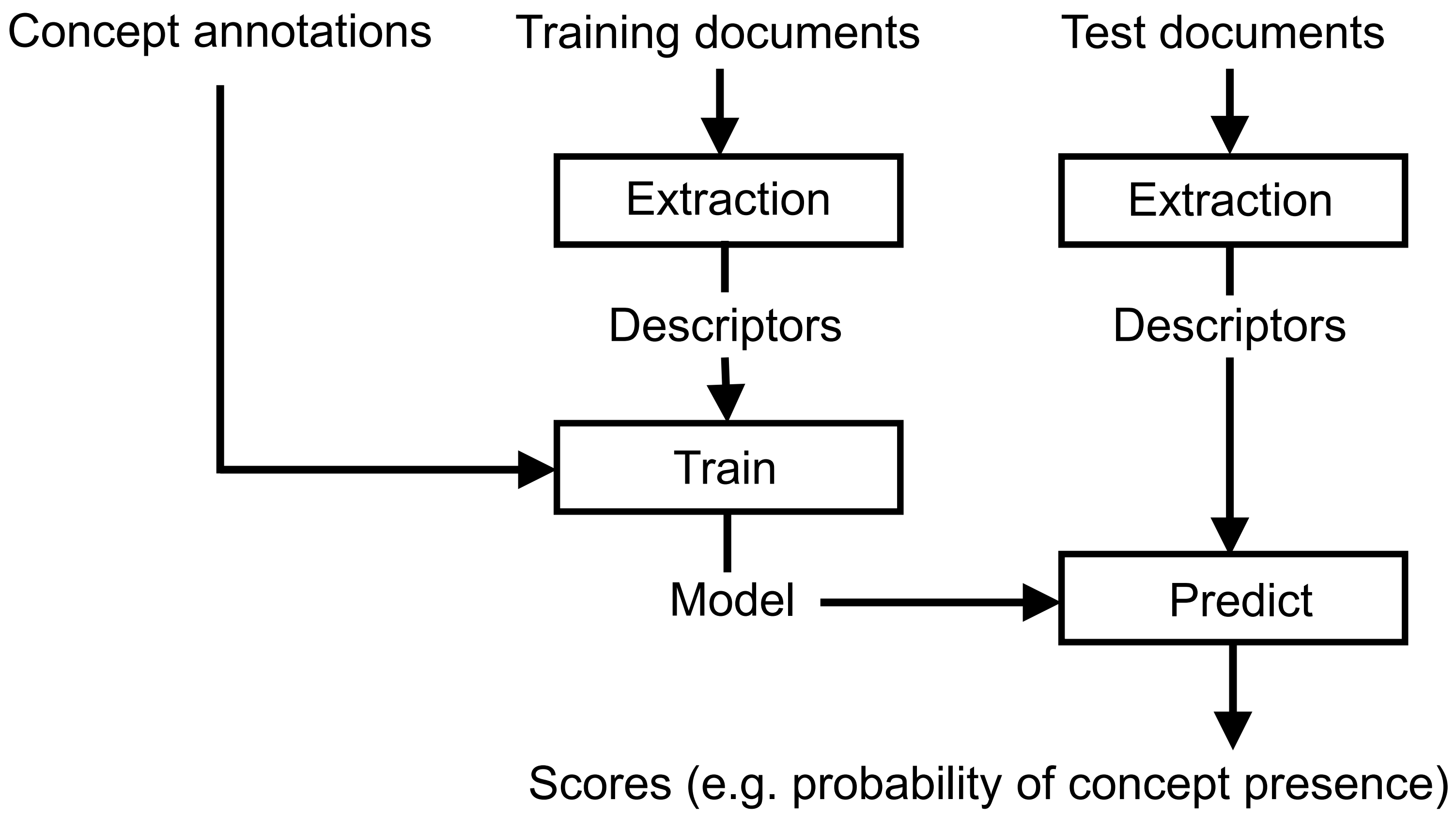
- Semantics (opposed to signal) :
  - “Abstract” concepts and relations,
  - Symbolic representations (also signal),
  - Successive levels of abstraction from the “signal / physical / concrete / objective” level to the “semantic / conceptual / symbolic / abstract / subjective” level,
  - Gap between the signal and semantic levels (“red” versus “700-600 nm”),
  - Somewhat artificial distinction,
  - Intermediate levels difficult to understand,
  - Search at the signal level, at the semantic level or with a combination of both.

Query by example versus search

Query BY Example (QBE)



Content based indexing by supervised learning



Example : the QBIC system

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

Content-based search

- Aspects :
  - Signal : arrays of numbers (“low level”),
  - Semantic : concepts or keywords (“high level”).
- Search :
  - Semantic $\rightarrow$ semantic : classical for text,
  - Semantic $\rightarrow$ signal : images corresponding to a concept ?
  - Signal $\rightarrow$ signal : image containing a part of another image ?
  - Signal $\rightarrow$ semantic : concepts associated to an image ?
- Approaches :
  - Bottom-up : signal $\rightarrow$ semantic,
  - Top-down : semantic $\rightarrow$ signal,
  - Combination of both.

Document representation

- Compression : encoding and decoding
- Indexing : characterization of the contents

JPEG
GIF
PNG
MJPEG
DV
MPEG-1
MPEG-2
MPEG-4

Documents
↓
Compression
↓
Representation
↓
Decompression
↓
Documents*

Documents
↓
Indexing
↓
Representation
↓
Retrieval
↓
Reference

MPEG-7

Problems

- Choice of a representation model,
- Indexing method and index organization,
- Choice and implementation of the search engine,
- Very high data volume,
- Need for manual intervention.

Representation models

- Semantic level:
  - keywords, word groups, concepts (thesaurus),
  - Conceptual graphs (concepts and relations),
- Signal level:
  - Feature vectors,
  - Sets of interest points,
- Intermediate level:
  - Transcription of the audio track,
  - Sets of key frames,
  - Mixed and structured representations, levels of detail,
  - Application domain specificities,
- Standards (MPEG 7).

Indexing methods and index organization

- Build representations from document contents,
- Extract features for each document or document part:
  - Signal level: automatic processing,
  - Semantic level : more complex, manual to automatic.
- Globally organize the features fo the search:
  - Sort, classify, weight, tabulate, format, …
- Application domain specificities,
- Problem of the quality versus cost compromise.

Choice and implementation of the search engine

- Search for the “best correspondence” between a query and the documents,
- Semantic → semantic:
  - Logical, vector space and probabilistic models,
  - Keywords, word groups, concepts, conceptual graphs, …
- Signal → signal :
  - Color, texture, points of interest, …
  - Images, imagettes, pieces of image, sketches, …
- Semantic → signal :
  - Correspondence evaluated during the indexing phase (in general).
- Search with mixed queries.

Descriptors

Descriptors

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

Histograms - general form

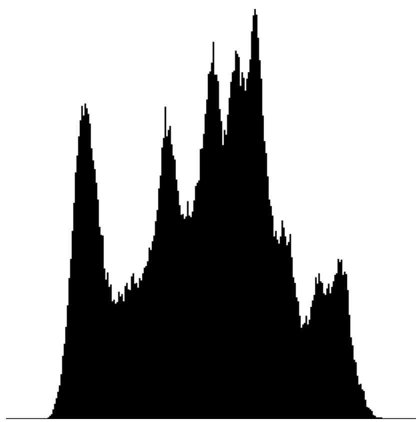
- A fixed set of *disjoint categories* (or *bins*), numbered from 1 to K.
- A set of *observations* that fall into these categories
- The histogram is the vector of K values $h[k]$ with $h[k]$ corresponding to the number of observations that fell into the category k.
- By default, the $h[k]$ are integer 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

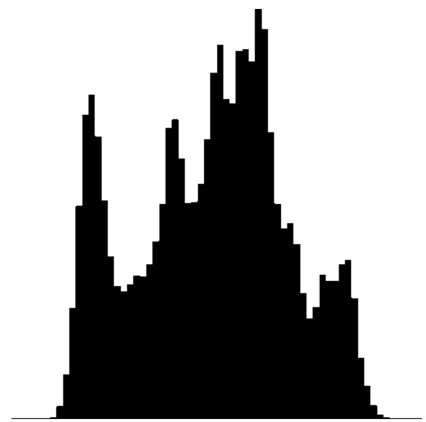
- A vector of term frequencies (tf) is an histogram
- The categories are the index terms
- The observations are the terms in the documents that are also in the index
- A tf.idf representation corresponds to a weighting of the bins, less relevant in multimedia since histograms bins are more symmetrical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

Image intensity histogram

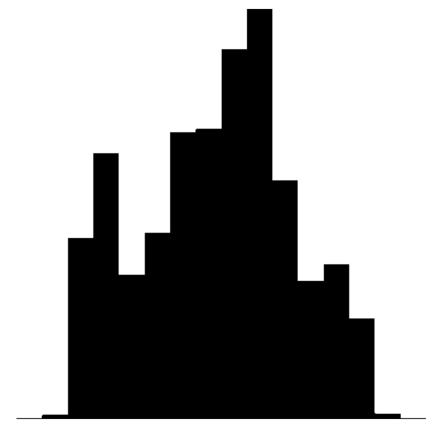
- The set of categories are the possible intensity values with 8-bit coding, ranging from 0 (black) to 255 (white) or ranges of these intensity values



256-bin



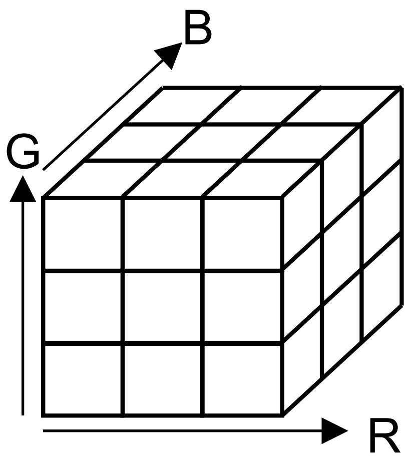
64-bin



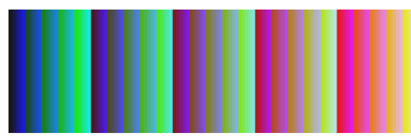
16-bin

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

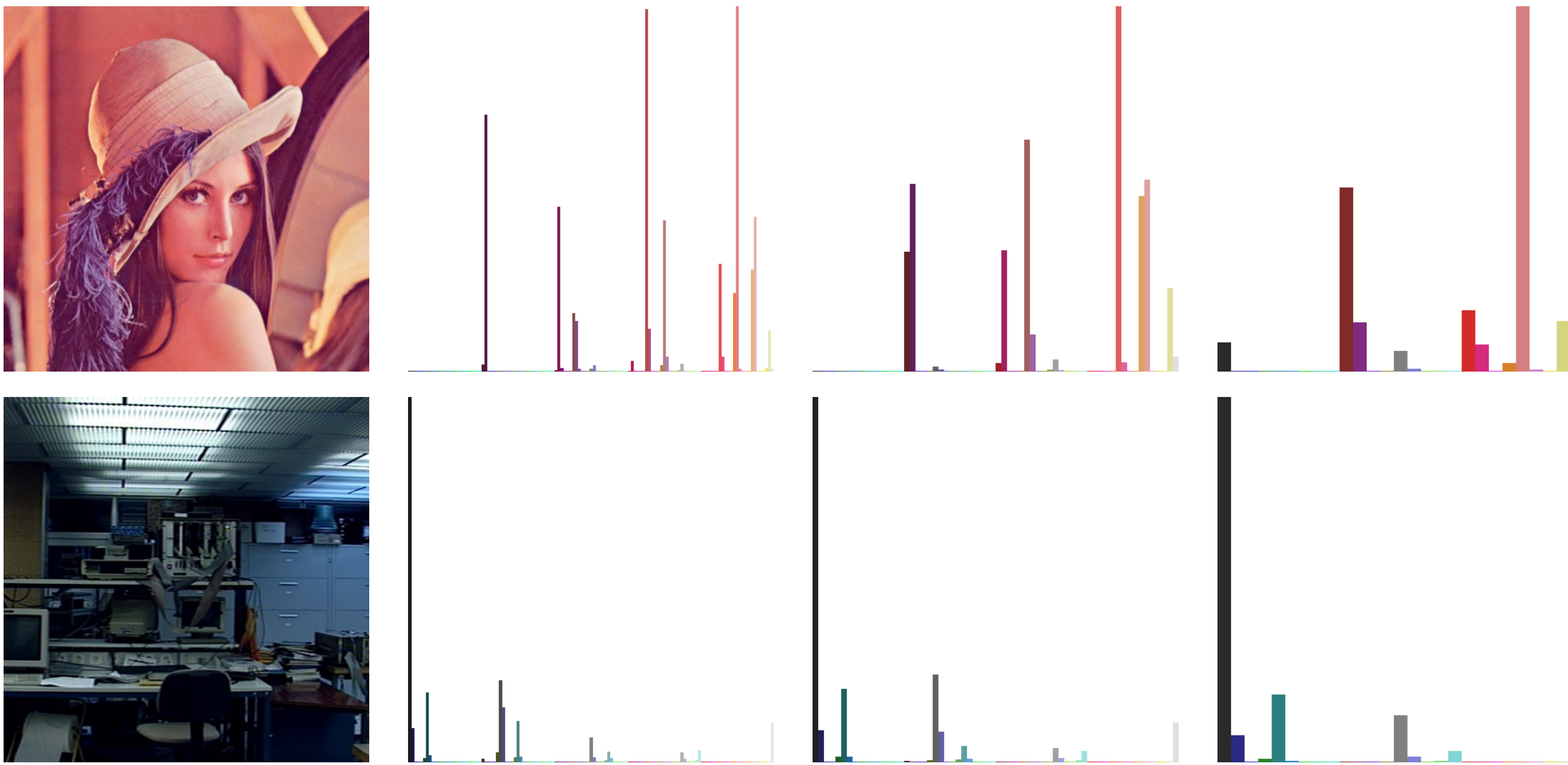
4×4×4-bin
64-bin

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



5×5×5-bin
125-bin

4×4×4-bin
64-bin

3×3×3-bin
27-bin

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 spaces

- Linear:
  - RGB: Red, green, blue
  - YUV: Luminance, chrominance (L – red, L – blue)
- Non linear:
  - HSV: Hue, Saturation, Value
  - LAB: Luminance, “blue – yellow”, “green – red”

Color moments

- Moments (color distribution global statistics)
  - Means
  - Covariances
  - Third order moments
  - Can be combined with image coordinates
  - Fast and easy to compute and compact representation but not very accurate

Color moments

- Means:
  $mR = (\Sigma R)/N$, $mG = (\Sigma G)/N$, $mB = (\Sigma B)/N)$
- Means + variances: + covariances:
  $mRR = (\Sigma(R-mR)^2)/N$, $mGB = (\Sigma(G-mG)(B-mB))/N$,
  …
- Higher order moments:
  $mRGB = (\Sigma(R-mR)(G-mG)(B-mB))/N$, mRRR, mRGG, …

- Moments associated to spatial components :
  $mRX = (\Sigma(R-mR)(X-mX))/N$, mRGX, mBXY, …

Image normalization

- Objective : to become more robust against 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.

Correspondence functions for color

- Vectors of moments:
  - Euclidean distance : search for exact similarity,
  - Angle between vectors : search for similarity with robustness to illumination changes,
- Histograms:
  - Euclidean or χ^2 distance: search for exact similarity,
  - Robustness to illumination changes can only be obtained by an intensity normalization pre-processing,
  - Earth-mover distance: compute the cost for transforming one histogram into another by giving a flat penalty for passing from one bin to another
  - Histograms by blocks : sum of the smaller block to block distances only (typically 8 out of 16): permits a search with only a portion of an image,
- Correlograms:
  - Euclidean or χ^2 distance, with or without intensity normalization.

Texture descriptors

- Computed on the luminance component only
- Rather fuzzy concept,
- Frequential composition or local variability,
- Fourier transforms,
- Gabor filters,
- Neuronal filters,
- Cooccurrence matrices,
- Many possible combination,
- Feature vector,
- Associated correspondence functions,
- Normalization possible.

1D discrete convolution

Mathematical definition:

- f and g : functions from $\mathbb{Z}$ to $\mathbb{R}$ (or to $\mathbb{C}$)

$$(f * g)(n) = \sum_{m \in \mathbb{Z}} f(m)g(n-m) = \sum_{m \in \mathbb{Z}} f(n-m)g(m)$$

- Infinite sums must be convergent
- In practice: f and g are defined on bounded regions $\rightarrow$ padding (usually with zeroes) $\rightarrow$ "side effects"

1D discrete convolution

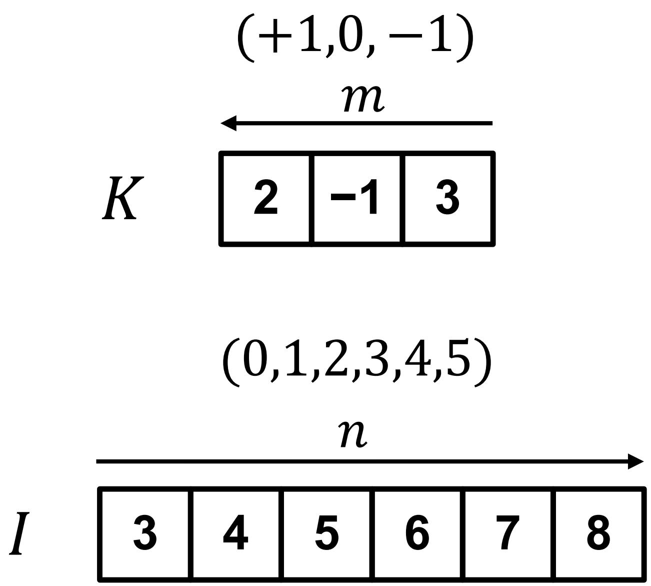
Signal processing:

- Application of a 1D convolution kernel K to a 1D input signal I for producing an output signal $O = K * I$ with $O(n) = \sum_{m \in W} K(m) I(n-m)$
- m : within a finite (and usually centered) window W around the current location (n)
- Properties: linear (relatively to I), “local” and translation invariant
- The convolution product is commutative and associative: the sequential application of several kernels is the same as a single application of their product

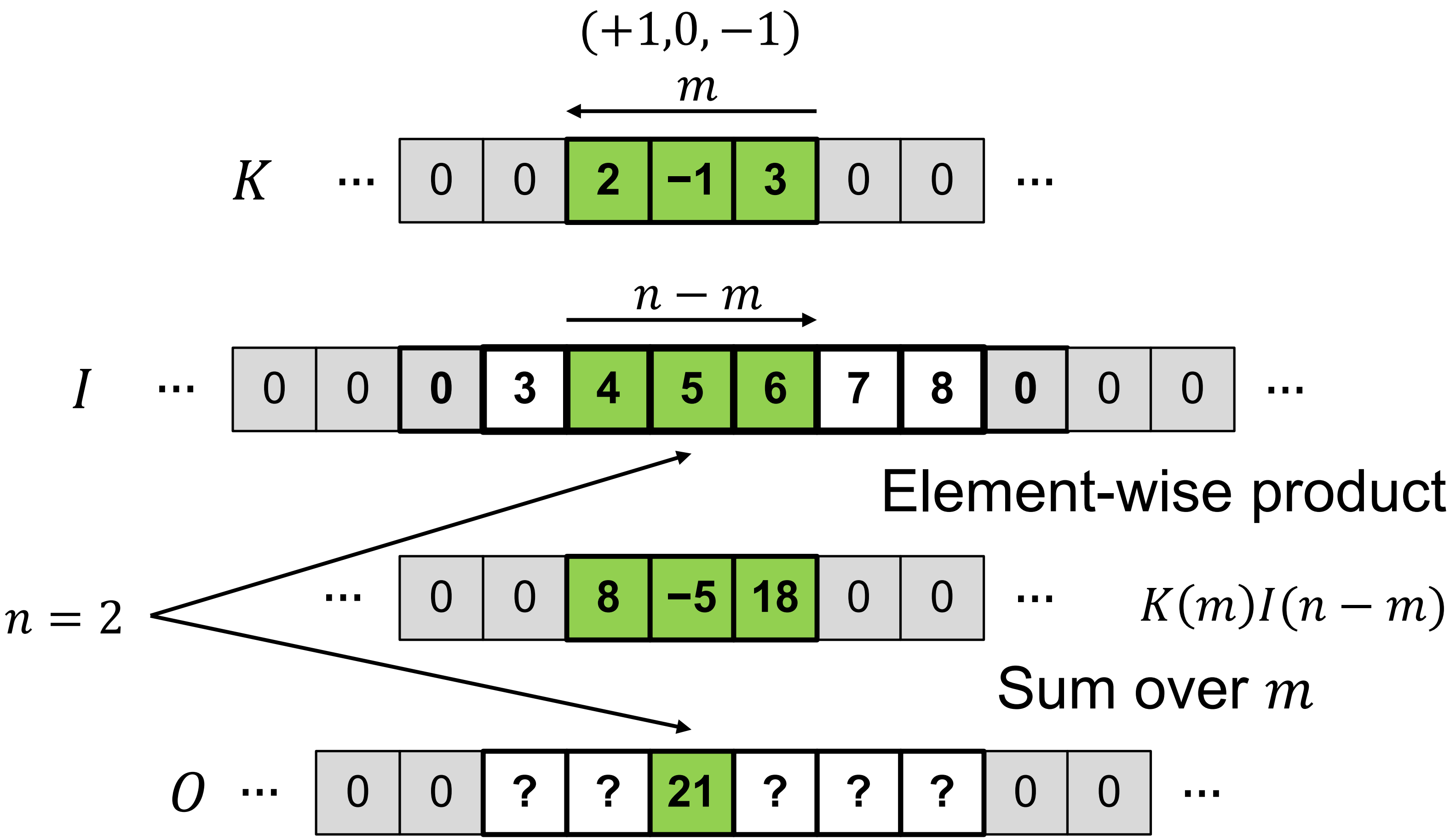
1D discrete convolution

Signal processing:

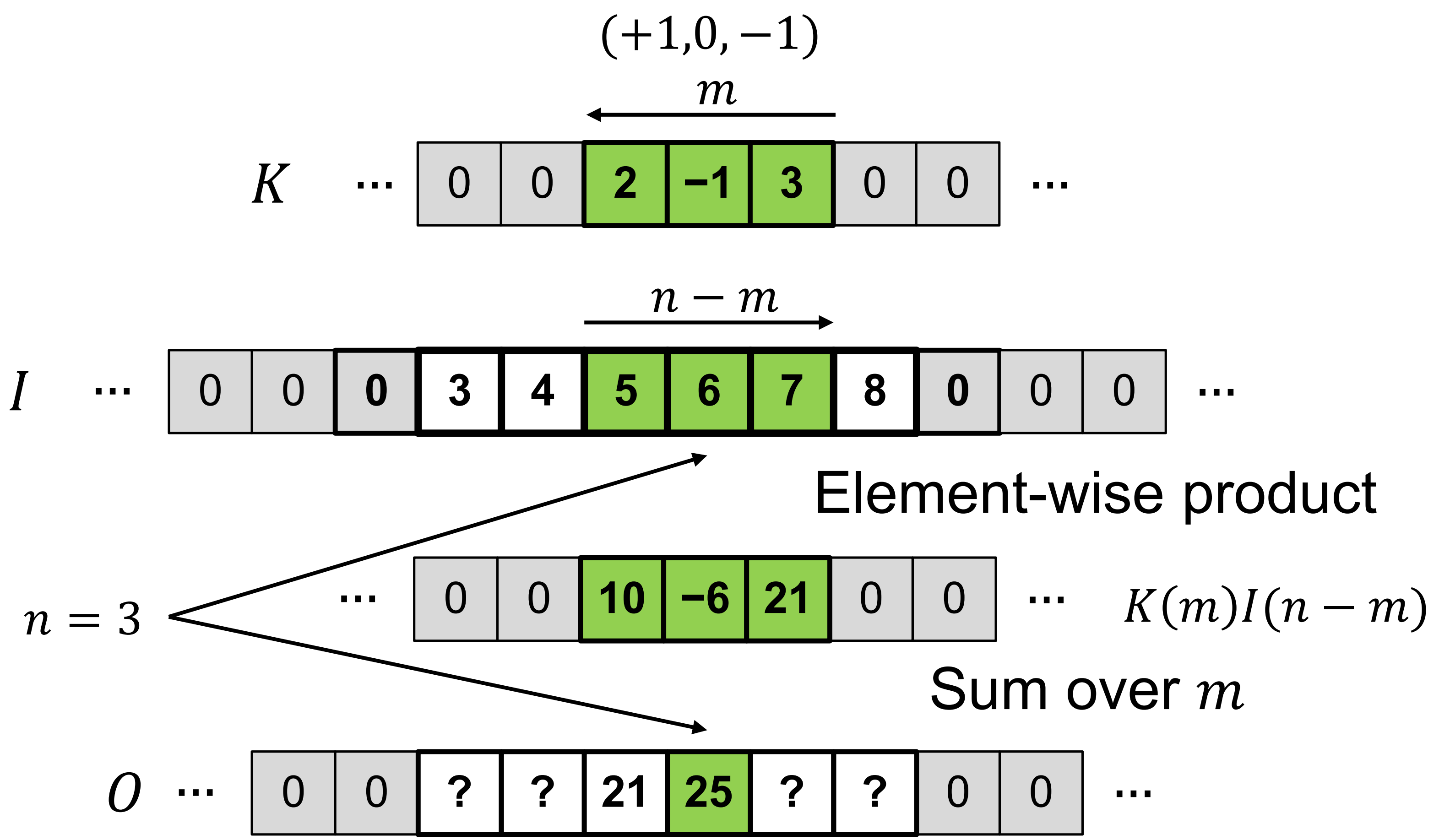
- Application of a1D convolution kernel K to a 1D input signal I



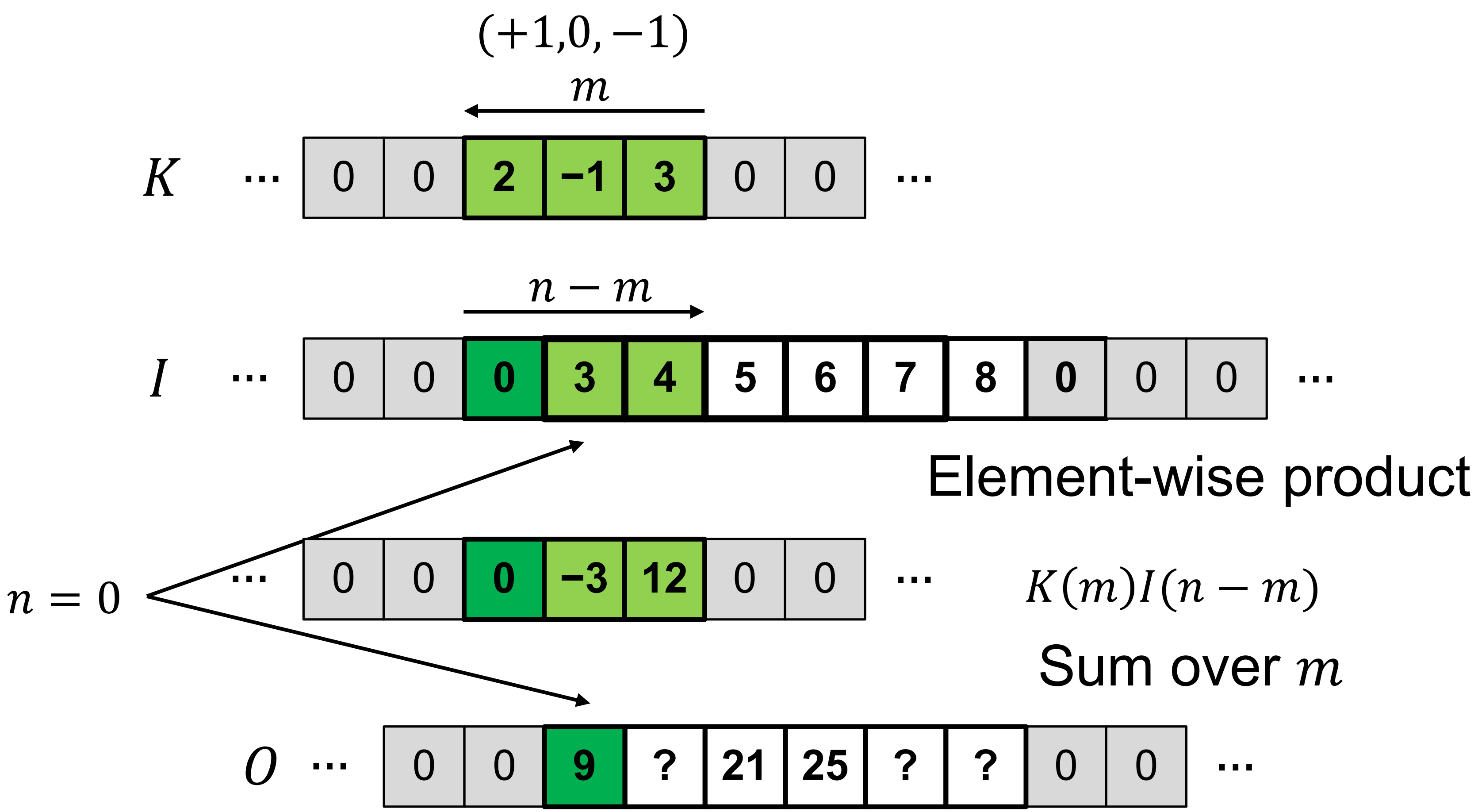
1D discrete convolution



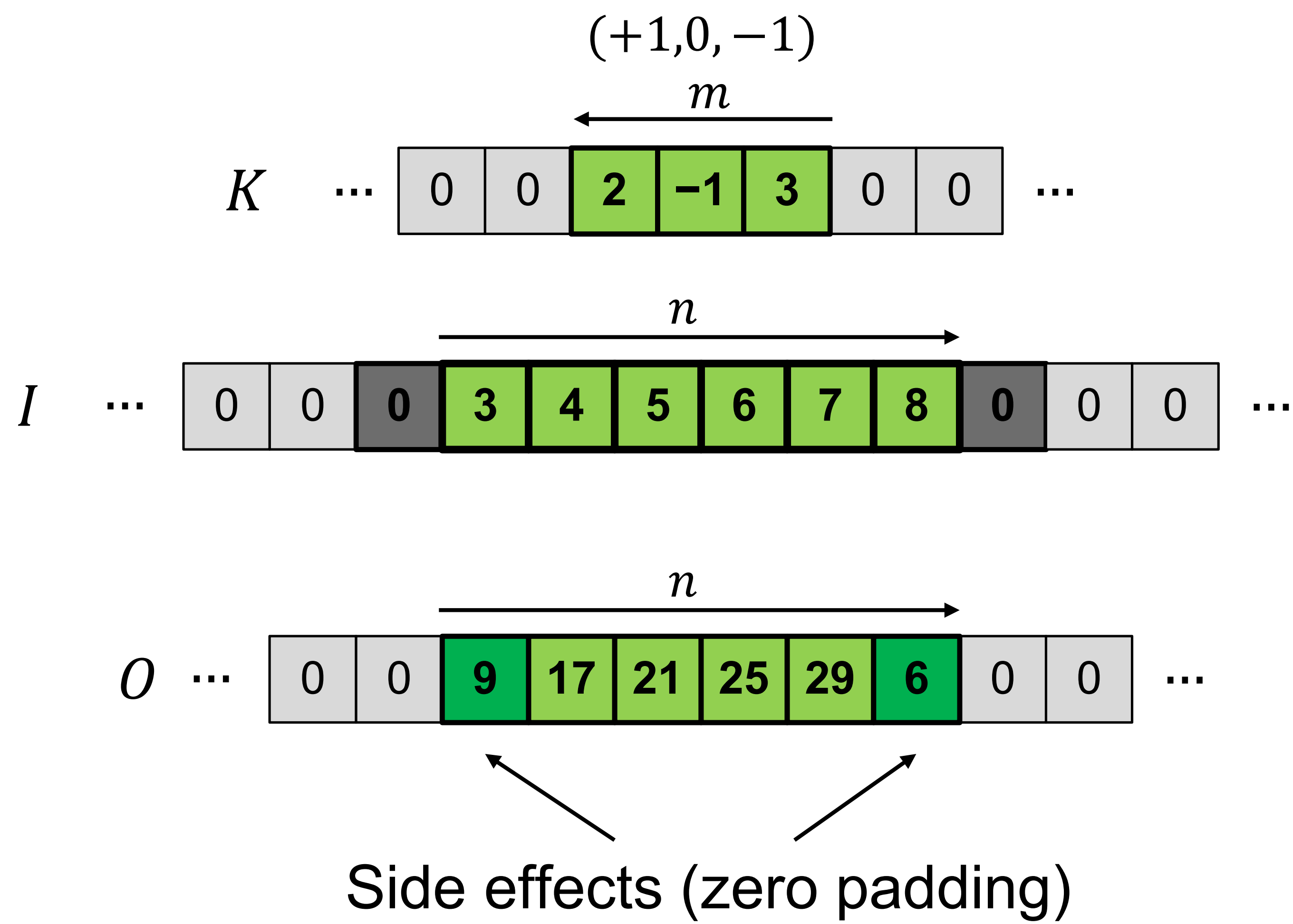
1D discrete convolution



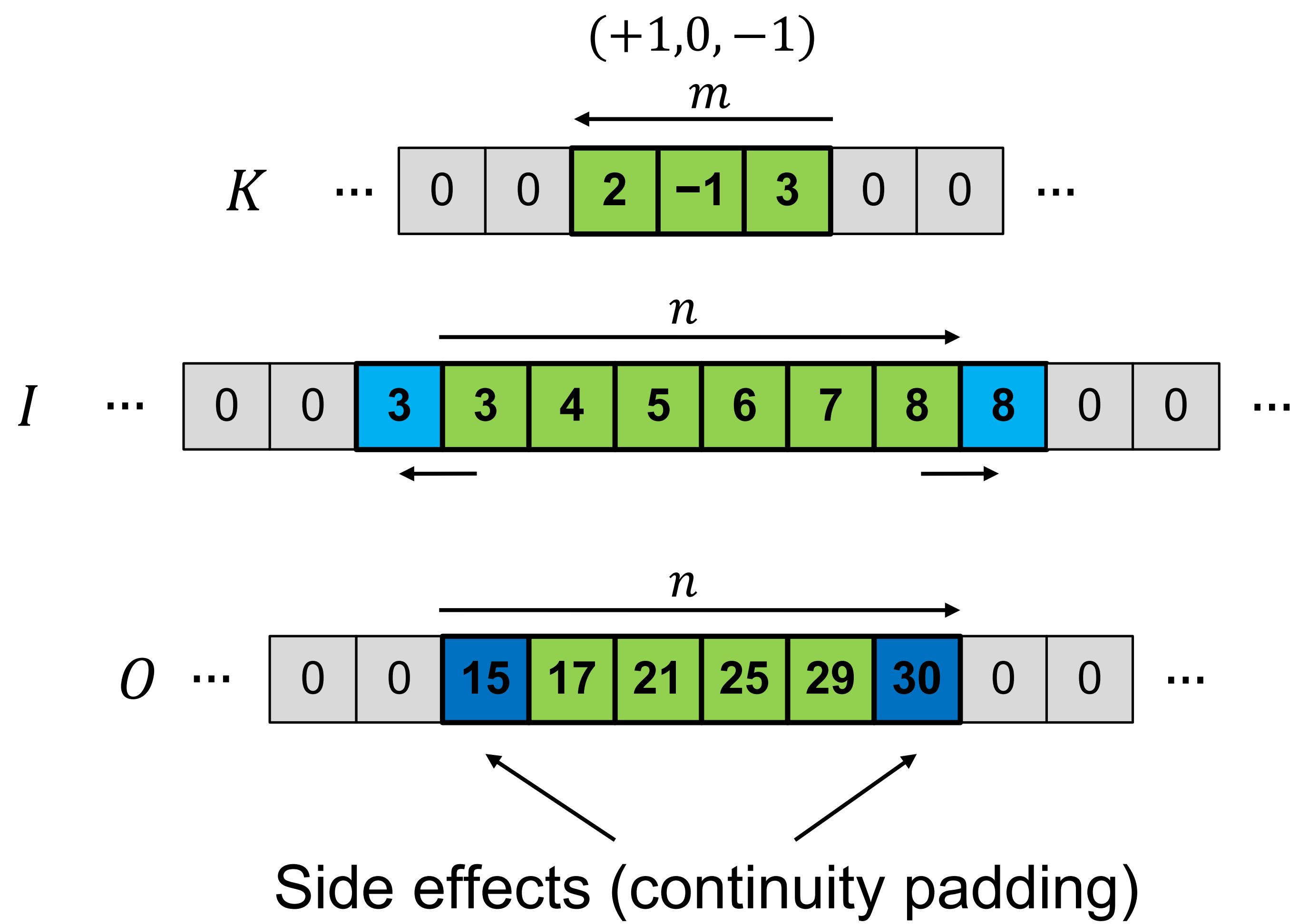
1D discrete convolution



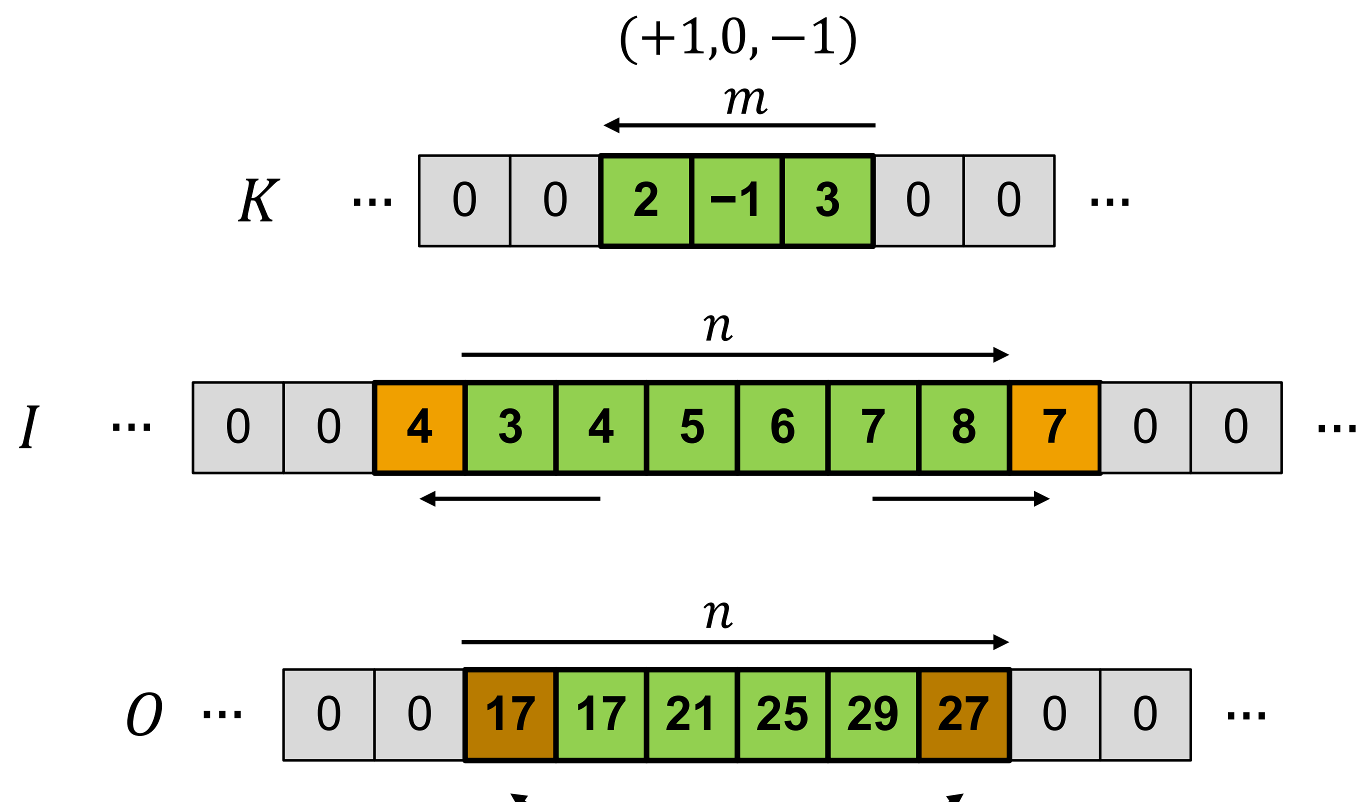
1D discrete convolution



1D discrete convolution

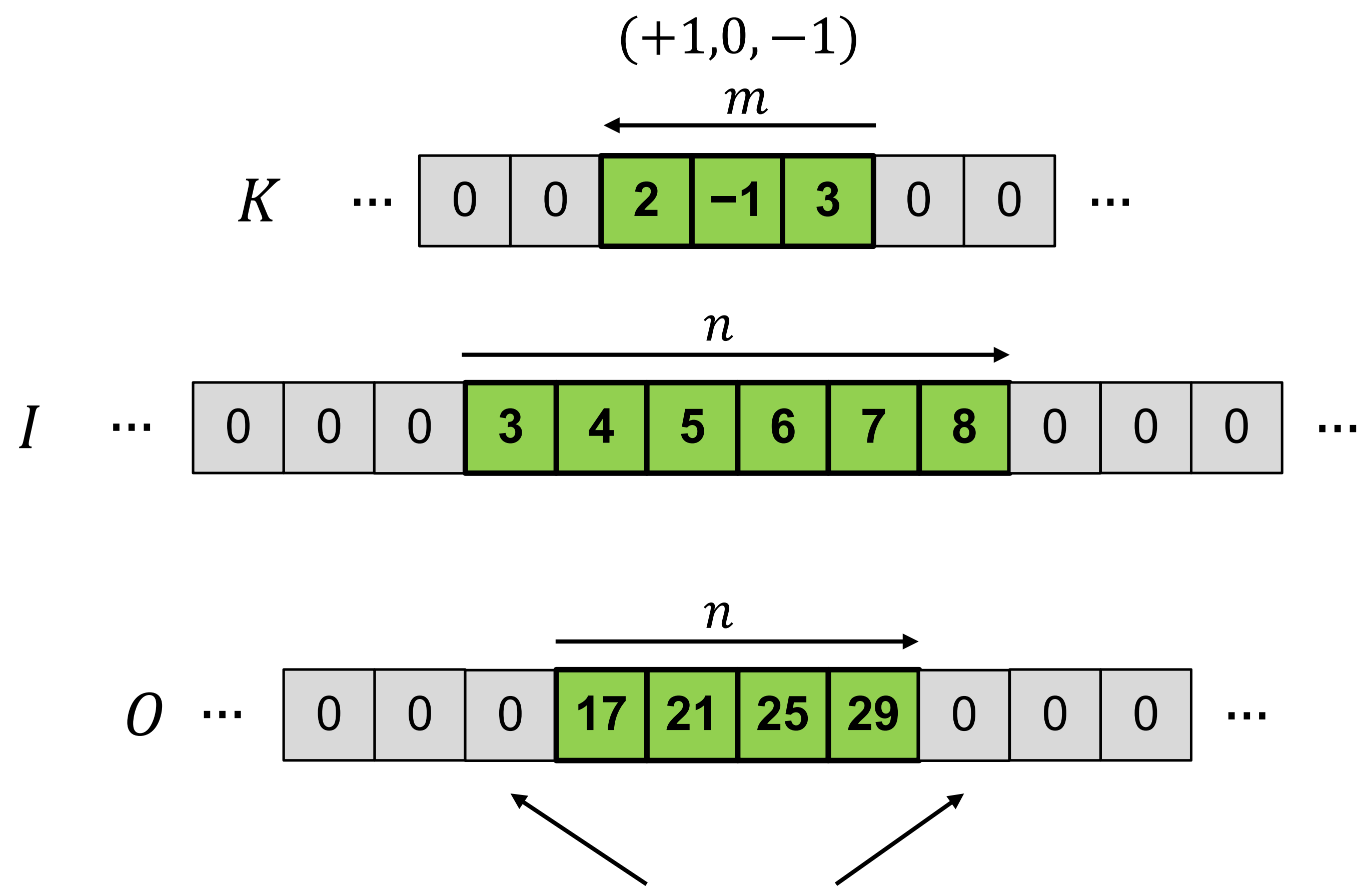


1D discrete convolution



Side effects (symmetry padding)

1D discrete convolution



No side effects (cropping)

1D discrete convolution

Examples:

- Derivative: $D(m) = \big(\delta\,(m+1) - \delta\,(m-1)\big)/2$
  ($D(m) = +1$ if $m = -1$, -1 if $m = +1$, 0 otherwise)
  i.e.: $O(n) = (I(n+1) - I(n-1))/2$

- Average on a sliding window (basic smoothing):
  $A(m) = \frac{1}{2w+1}$ if $|m| \leq w$, 0 otherwise.
  Window size is $2w + 1$.

- Gaussian smoothing: $G_\sigma(m) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{m^2}{2\sigma^2}}$
  practical extension: 3-4σ

1D discrete convolution

Examples (all kernels are centered):

- Derivative: $D = \frac{1}{2} \times \boxed{1}\boxed{0}\boxed{-1}$
- Sliding average: $A_2 = \frac{1}{5} \times \boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1}$
- Binomial filter (discrete and bounded Gaussian filter)

$$B_w(m) = \frac{C_{2w}^{m+w}}{2^{2w}} = \frac{(2w)!}{2^{2w}(w-m)!(w+m)!} \qquad \sigma = \frac{\sqrt{w}}{2}$$

$$B_1 = \frac{1}{4} \times \boxed{1}\boxed{2}\boxed{1}$$

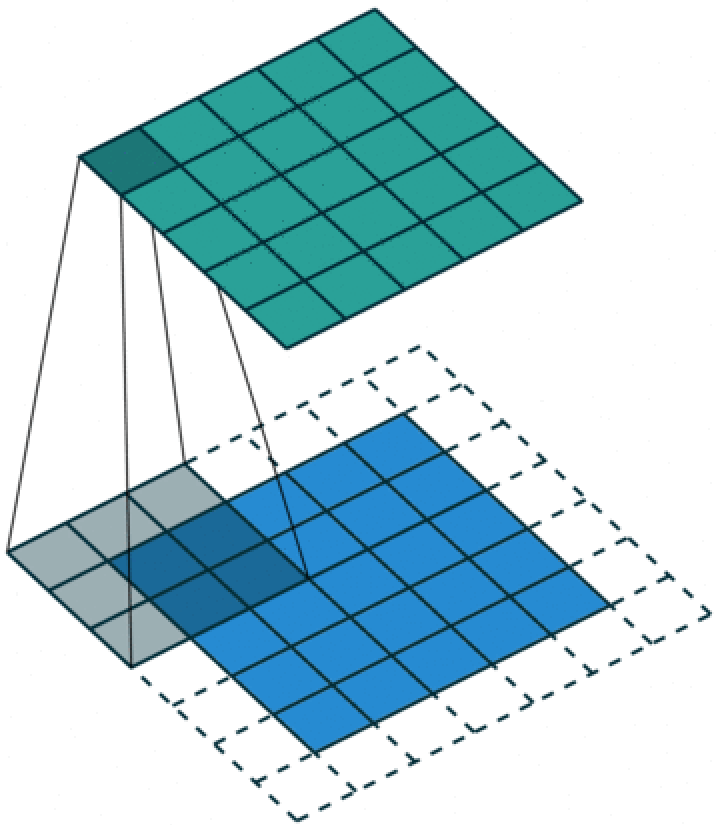
$$B_2 = \frac{1}{16} \times \boxed{1}\boxed{4}\boxed{6}\boxed{4}\boxed{1}$$

$$B_3 = \frac{1}{64} \times \boxed{1}\boxed{6}\boxed{15}\boxed{20}\boxed{15}\boxed{6}\boxed{1}$$

2D (image) convolution

- $O(i,j) = (K * I)(i,j) = \sum_{(m,n)} K(m,n) I(i-m, j-n)$
- m and n : within a window around the current location, corresponding to the filter size
- $K(m,n)$: convolution kernel, usually bounded
- Linear, “local” and translation invariant
- Side effects

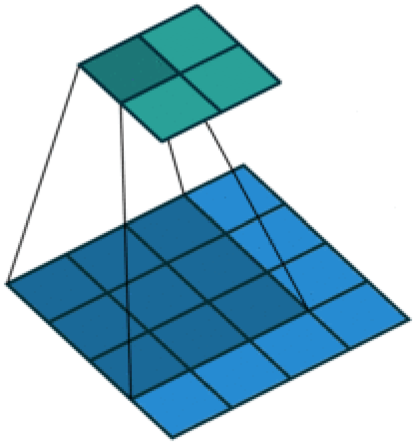
Classical image convolution (2D to 2D)



3x3 convolution, no stride, half padding

Animation from https://github.com/vdumoulin/conv_arithmetic/

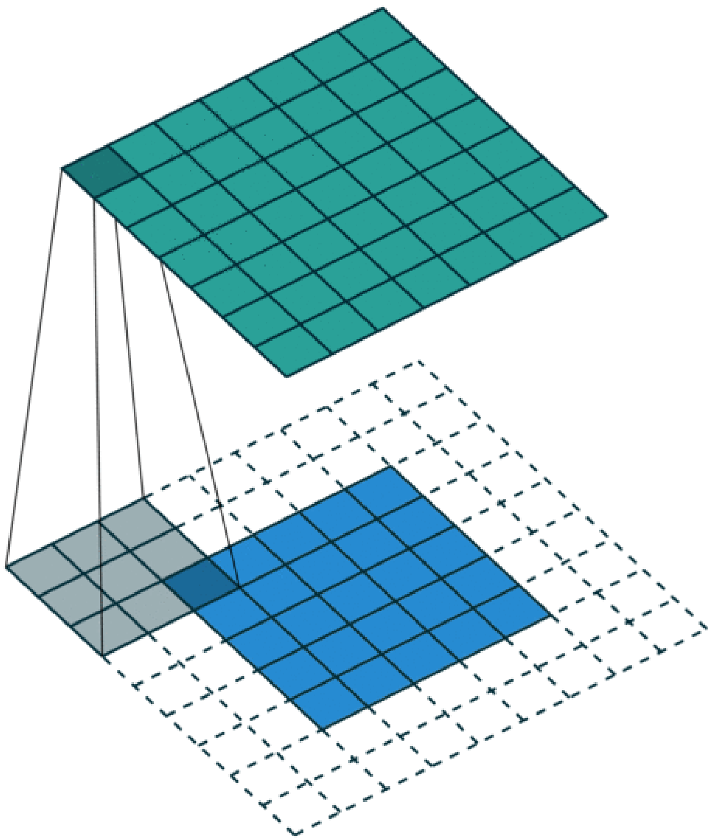
Classical image convolution (2D to 2D)



3×3 convolution, no stride, no padding

Animation from https://github.com/vdumoulin/conv_arithmetic/

Classical image convolution (2D to 2D)



3×3 convolution, no stride, full padding

Animation from https://github.com/vdumoulin/conv_arithmetic/

2D discrete convolution

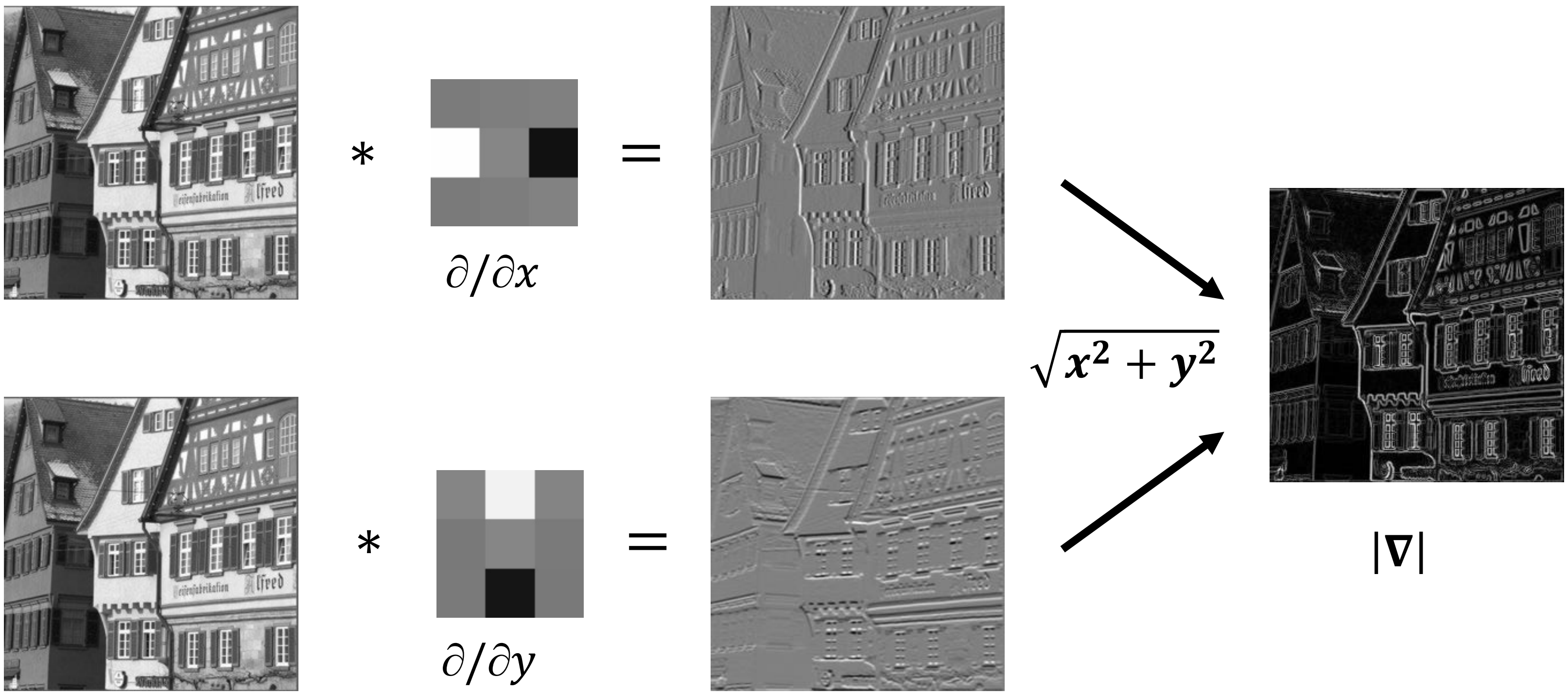
Examples, partial derivatives (all kernels are centered):

- $\partial/\partial x = \frac{1}{2} \times \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = \frac{1}{2} \times \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$

- $\partial/\partial y = \frac{1}{2} \times \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \frac{1}{2} \times \begin{array}{|c|}\hline 1 \\ \hline 0 \\ \hline -1 \\ \hline \end{array}$

2D discrete convolution

Examples, partial derivatives (all kernels are centered):



2D discrete convolution

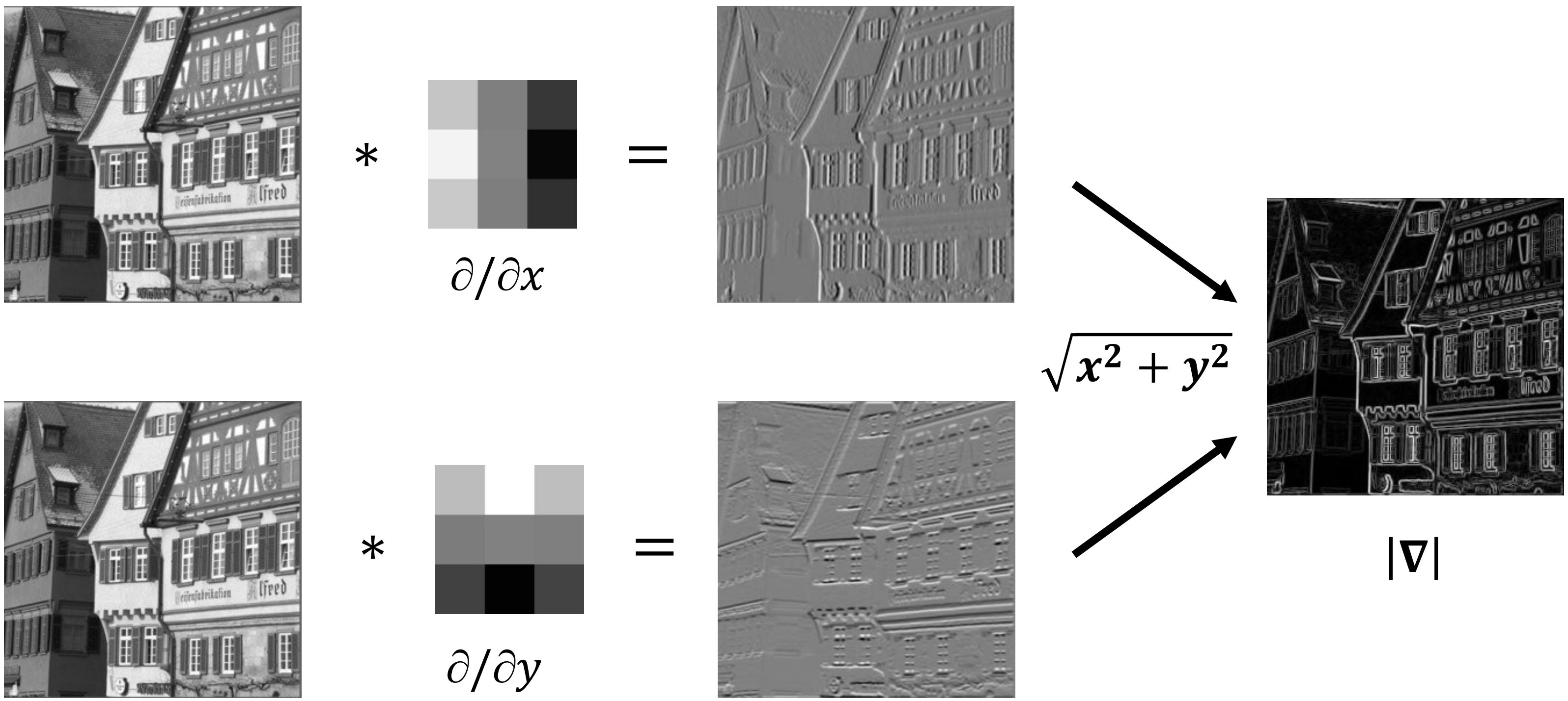
Partial derivatives, smoothed versions:

- $\partial/\partial x = \frac{1}{2} \times \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad * \quad \frac{1}{4} \times \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad = \quad \frac{1}{8} \times \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$

- $\partial/\partial y = \frac{1}{2} \times \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad * \quad \frac{1}{4} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad = \quad \frac{1}{8} \times \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

2D discrete convolution

Partial derivatives, smoothed versions:



Gabor filter

- Circular Gabor filter:

$$G_{\square\theta}\ (m,n) = \frac{1}{2\pi\sigma^2} . e^{-\frac{m^2+n^2}{2\sigma^2}} . e^{2\pi i \frac{m.\cos\theta + n.\sin\theta}{\lambda}}$$

Gaussian:
- Locality
- Side effect
- Filter width

Wave:
- Wave length
- Orientation

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:

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

Set of convolutional (linear) transform followed by a non-linear transformation (module) and a global pooling (average) : specific case of CNN layer.

Gabor transforms

"Separable" formulation:

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

$$h(\sigma,\theta,\lambda,I,i,j)=\sum_k \frac{e^{-\left(\frac{k^2}{2\sigma^2}\right)}}{\sqrt{2\pi\sigma}}.e^{2\pi\mathbf{i}\left(\frac{k.cos\theta}{\lambda}\right)}.I(i+k,j)=H(i,j)$$

$$g(\sigma,\theta,\lambda,I,i,j)=\sum_l \frac{e^{-\left(\frac{l^2}{2\sigma^2}\right)}}{\sqrt{2\pi\sigma}}e^{2\pi\mathbf{i}\left(\frac{l.sin\theta}{\lambda}\right)}.h(\sigma,\theta,\lambda,I,i,j+l)=G(i,j)$$

Gabor transforms

Linear combination coefficients:

$$c(k) = \frac{e^{-\left(\frac{k^2}{2\sigma^2}\right)}}{\sqrt{2\pi}\sigma}.\left(cos\left(\frac{2\pi k.cos\theta}{\lambda}\right) + \mathbf{i}.sin\left(\frac{2\pi k.cos\theta}{\lambda}\right)\right)$$

$$d(l) = \frac{e^{-\left(\frac{l^2}{2\sigma^2}\right)}}{\sqrt{2\pi}\sigma}.\left(cos\left(\frac{2\pi l.sin\theta}{\lambda}\right) + \mathbf{i}.sin\left(\frac{2\pi l.sin\theta}{\lambda}\right)\right)$$

Gabor transforms

Simplified expressions:

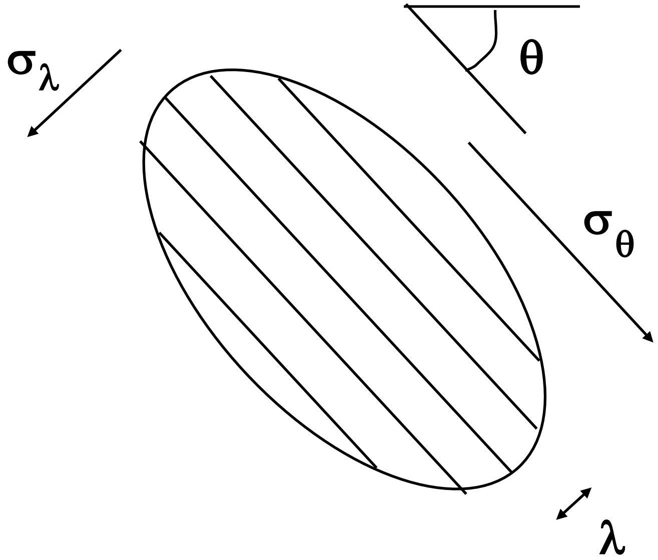
$$H(i,j) = \sum_k c(k).I(i+k,j)$$

$$G(i,j) = \sum_l d(l).H(i,j+l)$$

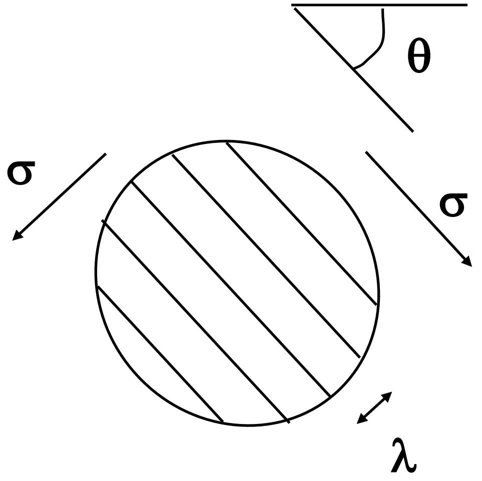
$$E^2 = \frac{1}{N} \sum_{i,j} \mid G(i,j) \mid^2$$

Gabor transforms

Elliptic:

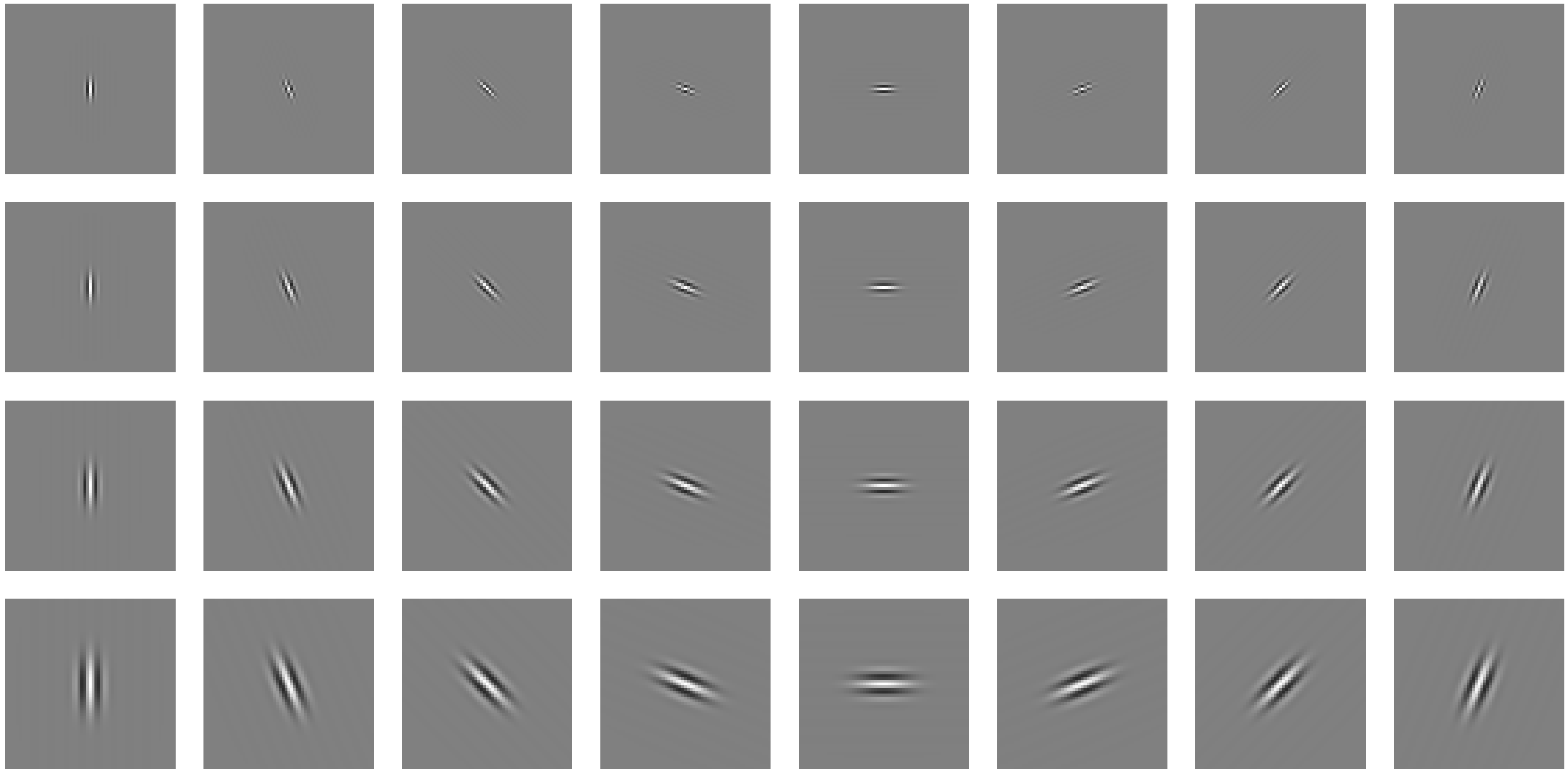


Circular:



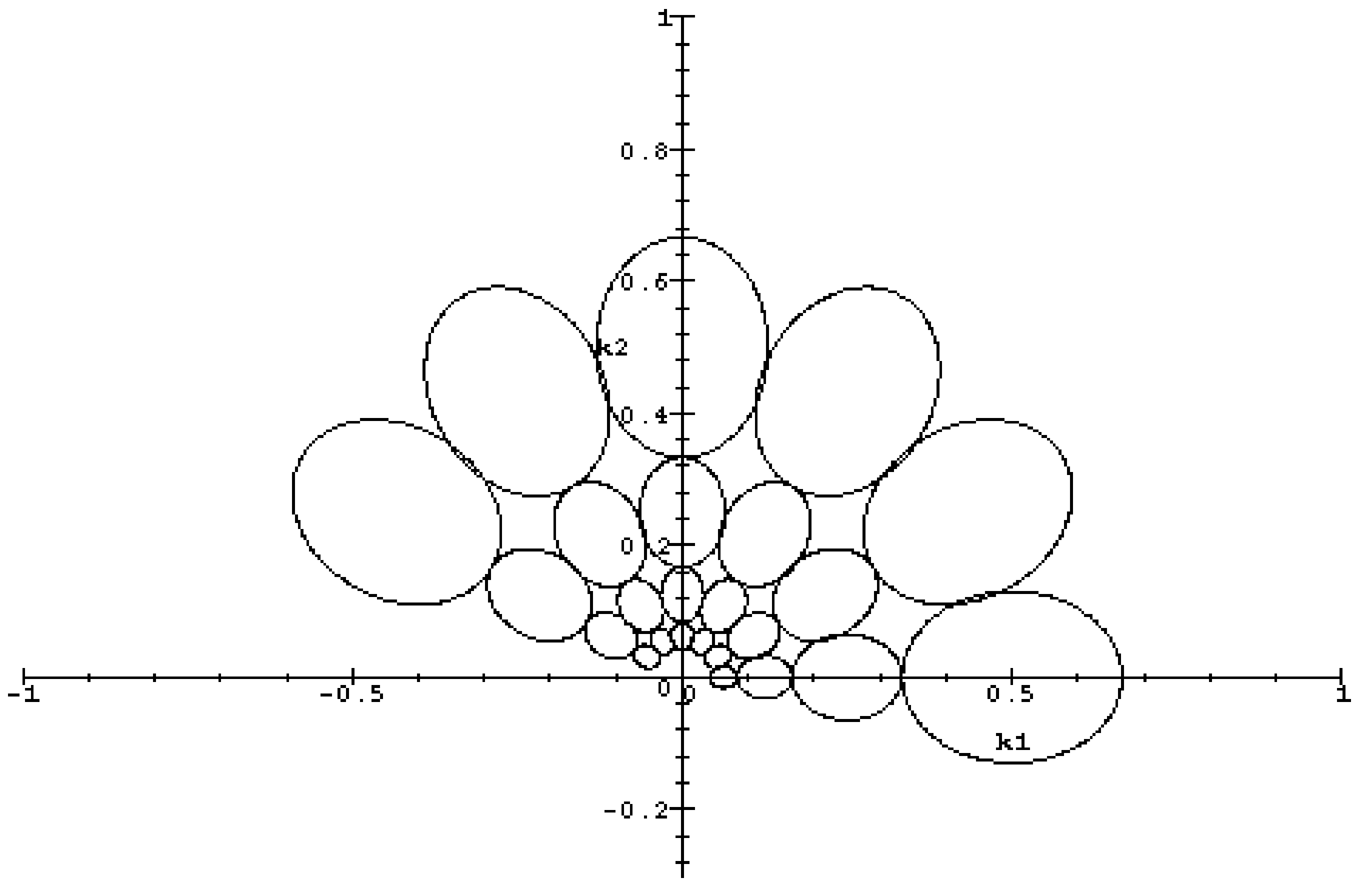
Filtres de Gabor

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,

Correspondence Functions for Gabor transforms

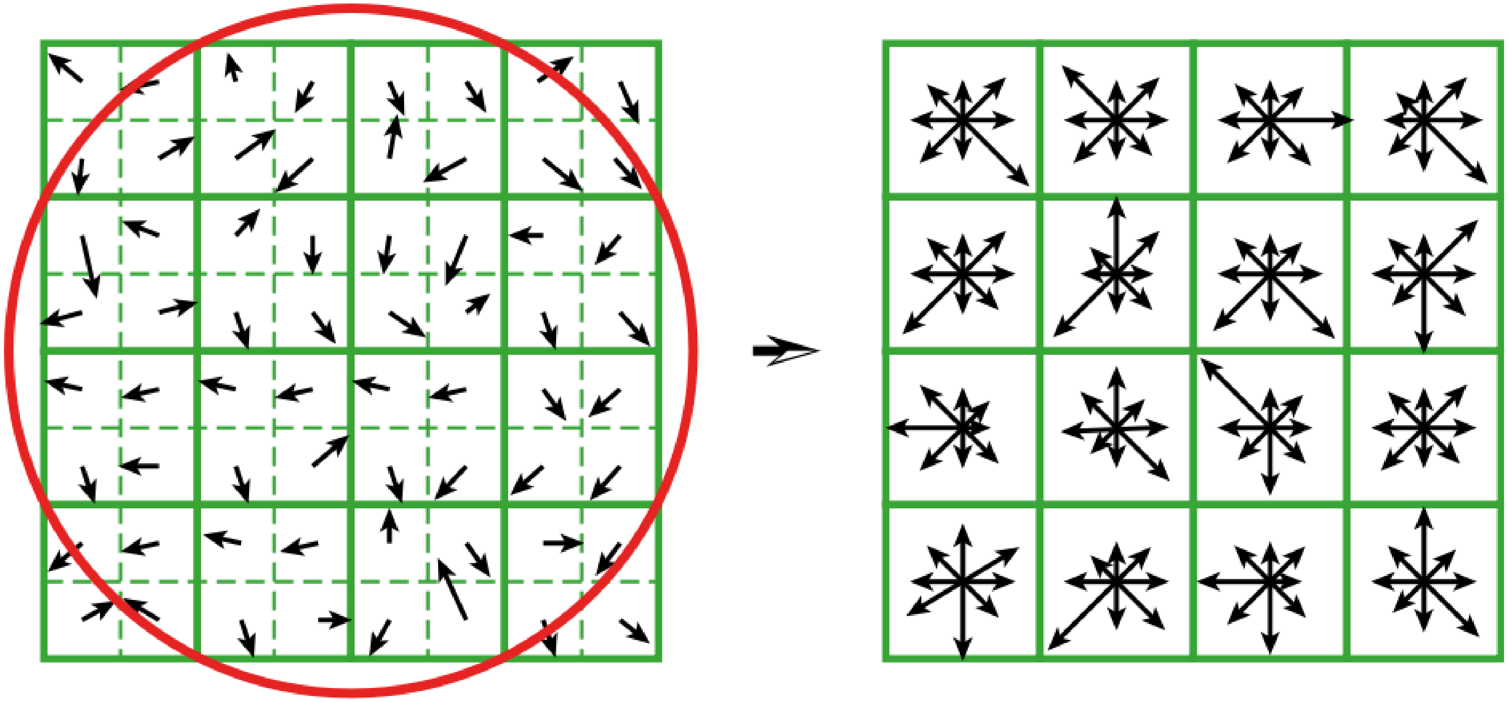
- Euclidean Distance : searching for identities,
- Angle between vectors : searching for similarities robust to illumination changes,

Descriptors 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 a given scale,
  - Convolution with derivatives of Gaussians,
  - Harris-Laplace detector.
- Construction of invariants by an appropriate combination of these various derivatives,
- Each point is selected and then represented by the set of values of these invariants,
- The set of selected points of interest is topologically organized (relations between neighbor points),
- The structure is irregular and the size of the description depends upon the image contents,
- Descriptions are large.

Descriptors of points of interest

- SIFT descritptor: Histogram of gradient direction:
  8 bins times 4 x 4 blocks in a neighborhood of the point.



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
- 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], Supervectors

Aggregation of local descriptors

- Histogramming according to clusters in the local descriptor space:
  - Clustering: partitioning of the descriptor space according to training data:
    - k-means or equivalent method
    - each cluster is represented by its centroid
  - Mapping: associating a local descriptor to a cluster:
    - getting a cluster number for each local descriptor
    - number of the nearest centroid vector
  - Histogramming: counting the local descriptors in each cluster for a given image:
    - one histogram per image

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, $i = 1 \dots N$:
    - find the nearest center $c_j \; : \; j = \arg\min d(x_i, c_k)$
    - assign the x_i data point to the cluster j $\quad x_i \rightarrow c_j$
  - for each cluster, $j = 1 \dots K$:
    - 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 $\quad c_j = \dfrac{\sum_{x_i \rightarrow c_j} x_i}{\sum_{x_i \rightarrow c_j} 1}$
- 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
- 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 ($\rightarrow$ regular n-ary tree)
  - Recursively split in two parts only the biggest cluster into sub-clusters ($\rightarrow$ irregular binary tree)
- Hierarchical mapping: recursive search of the closest center from the coarsest to the finest grain.

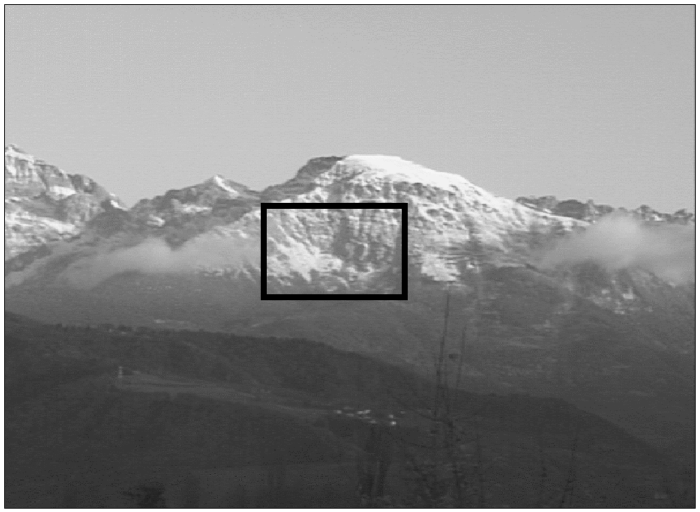
Correspondence functions for points of interest

- Generally very complex functions,
- Relaxation methods:
  - Randomly choose a point in the description of the query image,
  - Compare the neighborhood of this point to all the neighborhoods of all the points of the candidate document,
  - Amongst those that are “close” in the sense of the spatial relations and the values of the associated attributes, do a complementary search to see if the neighbor points are also “close” in the same sense,
  - Propagate the correspondence between “close” points by following the point topologies in the query and candidate images,
  - Find the best possible global correspondence respecting these topologies et preserving close characteristics for the in correspondence,
  - Globally evaluate (quantify) the quality of the correspondence.

Correspondence functions for points of interest

- Very costly method both for representation volume and computation time for the correspondence function,
- But very accurate and selective,
- Allows for retrieving an image from a portion of it by searching for a partial correspondence,
- Can be made robust to rotations by choosing appropriate invariants,
- Can be made robust to scale transforms by using multi-scale representations (even more costly)
- Usable only on small to medium image collections (~1000-10,000 images)
- Recent progress: up to millions of images.

Correspondence functions for points of interest

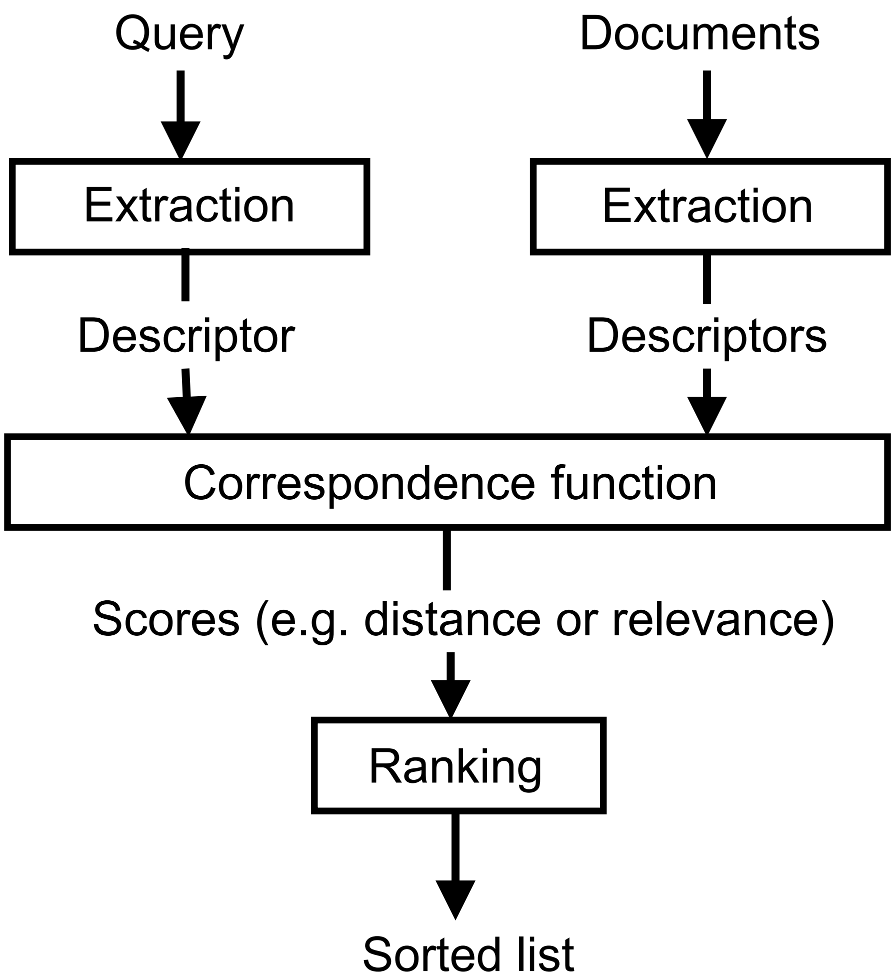


Example of an image pair involving a large scale change due to the use of a zoom. The scale factor between the images is 6. The common portion represents less than 17% of the image.

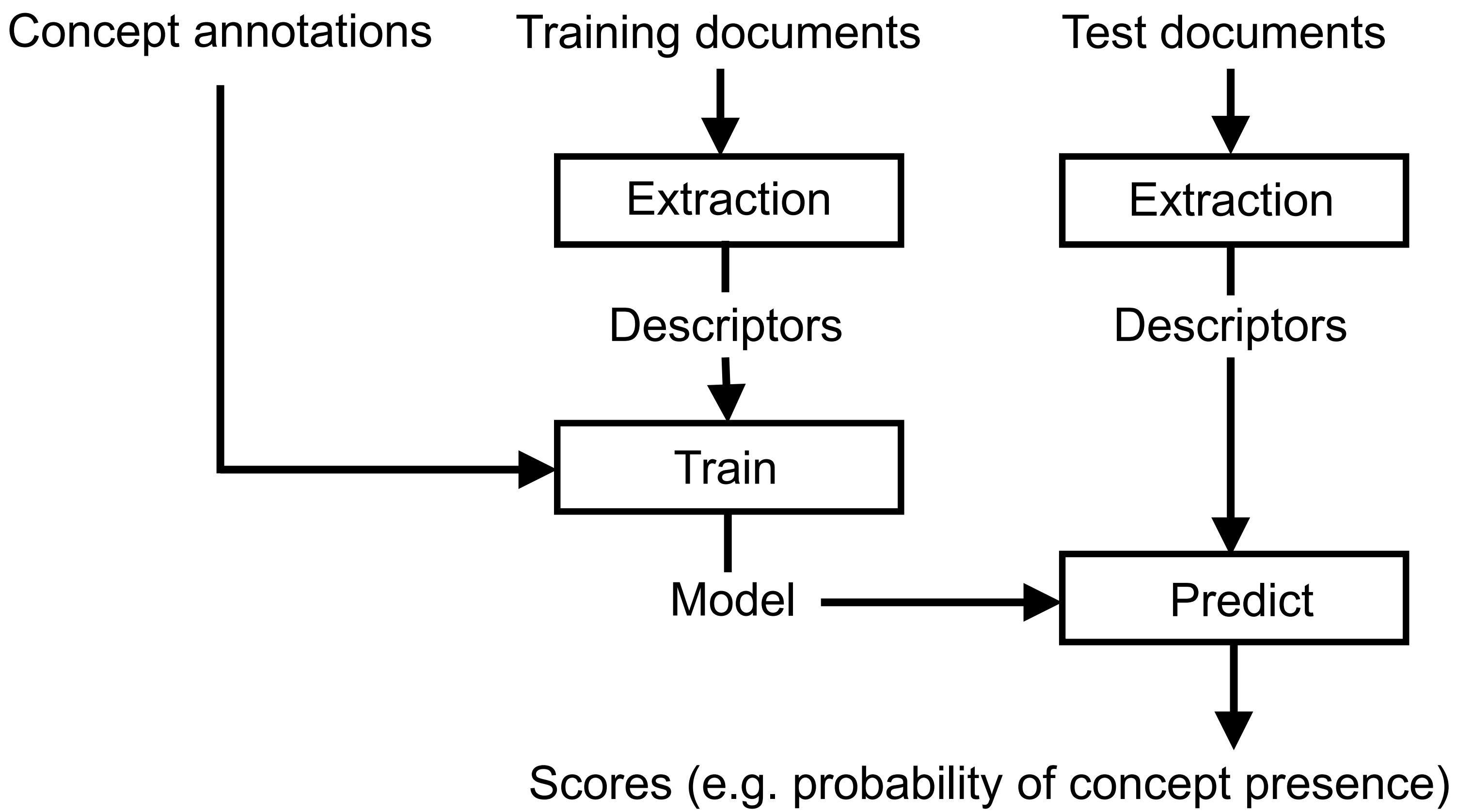
Use of several types of descriptors

- Several types of descriptors : choice according to the target application or to the query type,
- Several correspondence function for each type of descriptor : choice according to the target application or to the target query type (invariances that are desired or not for instance),
- Combination of the descriptions,
- Combination of the correspondence functions,
- Combination with descriptions from the semantic level.

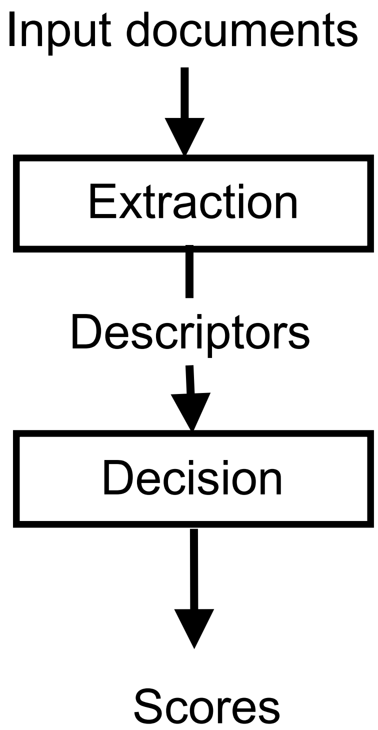
Query BY Example (QBE)



Content based indexing by supervised learning



Common processing, single descriptor



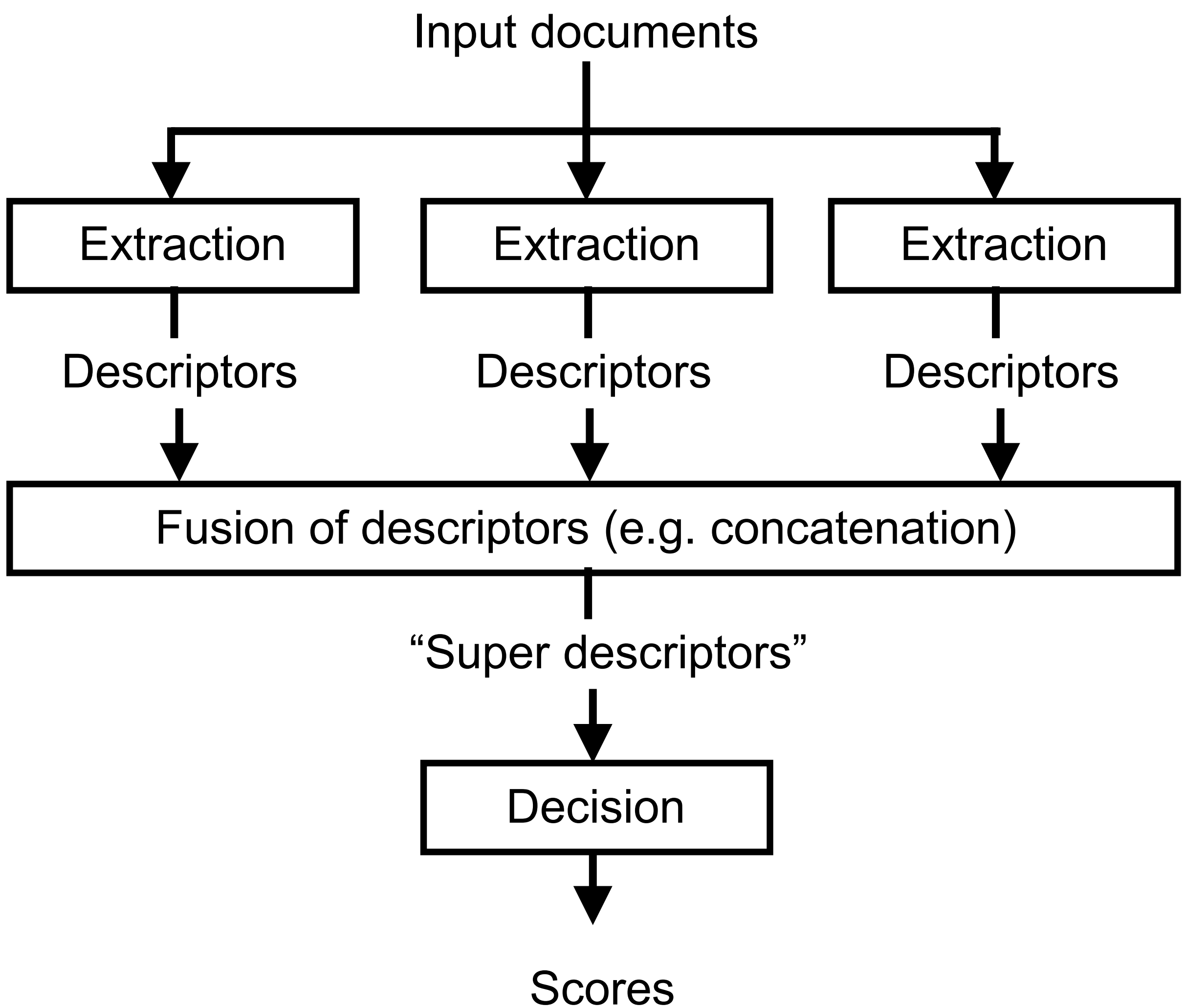
Query,
search collection,
training collection,
test collection …

Color, texture,
bag of SIFTs …

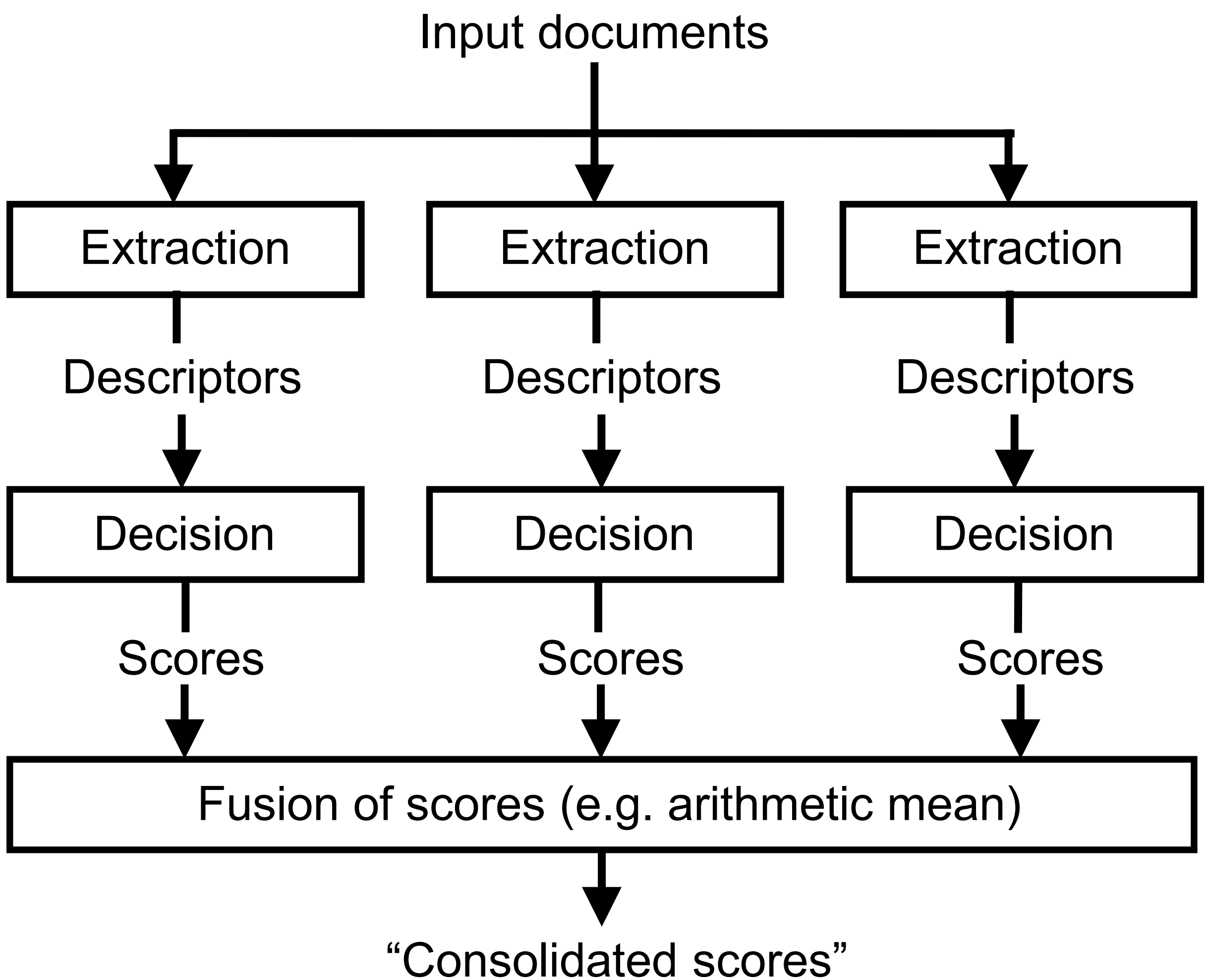
Correspondence function,
train / predict

Similarity measure,
probability of presence

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



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



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,
  - Learning is needed (representative data and process).
- Less information, faster once learning is done,
- Euclidean distance on the shortened vector.

Fusion of the correspondence functions (late)

- Each correspondence function generally produces a quantitative value that estimate a similarity,
- 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 logarithms),
  - The minimum value,
  - A classifier (SVM, neural network, …)
- Problem for the choice of the weights and/or for the classifier training.

Computation of the relevance

- Euclidean distance, angle between vectors,
- Comparison between a query vector to all the vectors in the database (no pre-selection),
- “Small” number of dimensions (< 10) : clustering techniques hierarchical search,
- “Medium” number of dimensions (~ 10+) : methods based on space partitioning,
- “Large” number of dimensions(>> 10) : no known method faster that a full linear scan,
- Reduction of the number of dimensions by Principal Component Analysis.

Principal Component Analysis 1

- "Natural" data contain redundancies:
  - Neighbor pixels' values are correlated
  - Political opinions and age of people are correlated
  - Weight and size of objects are correlated
  - …
- Principal Component Analysis aims at
  - Identify and characterize redundancies in data
  - Transform data for removing and reducing redundancies and possibly noise
  - "Ordinary or classical" PCA operates in the context of linear algebra (non linear variants also exist)

Principal Component Analysis 2

- Redundancies are identified as correlations
- Correlation is measured by covariance
  - Considering a set of samples $\{(x_i, y_i), i \in \{1 \dots N\}\}$, covariance is defined as:

$$\mathbf{cov}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{i}=\boldsymbol{N}} (x_i - \overline{\boldsymbol{x}})\,(y_i - \overline{\boldsymbol{y}}) \quad \text{with:} \quad \overline{\boldsymbol{x}} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{i}=\boldsymbol{N}} x_i$$

  - Correlation is defined as:

$$\mathbf{r} = \frac{\mathbf{cov}(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{\mathbf{cov}(\boldsymbol{x}, \boldsymbol{x})\mathbf{cov}(\boldsymbol{y}, \boldsymbol{y})}}$$

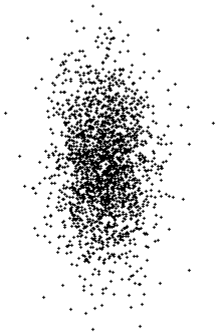
Principal Component Analysis 3

- Examples: no correlation (normal distributions)

cov(x,x) = 2500
cov(x,y) = 0
cov(y,y) = 2500
r = 0

cov(x,x) = 2500
cov(x,y) = 0
cov(y,y) = 225
r = 0

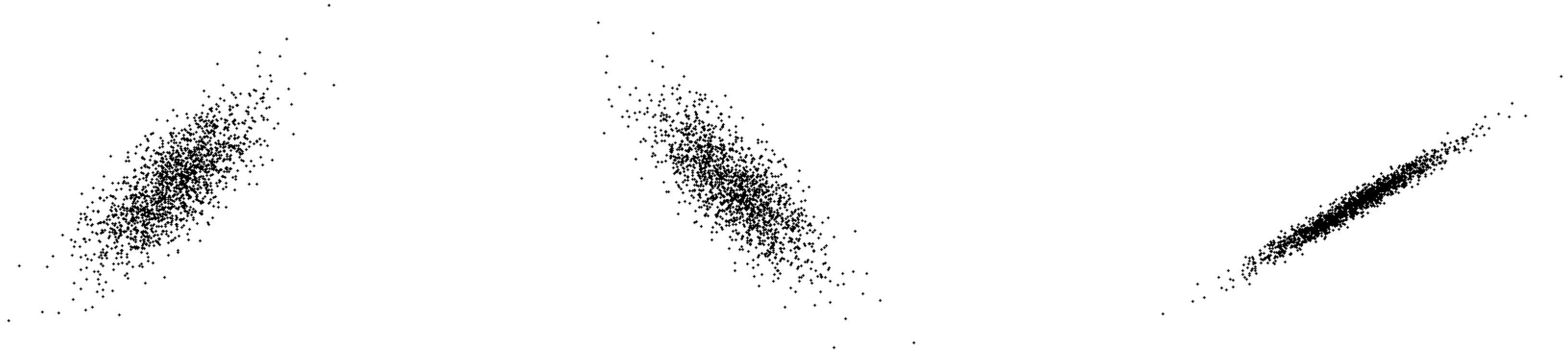
cov(x,x) = 625
cov(x,y) = 0
cov(y,y) = 2500
r = 0



Principal Component Analysis 4

- Examples: correlation (normal distributions)

cov(x,x) = 1800	cov(x,x) = 1800	cov(x,x) = 2500
cov(x,y) = 1350	cov(x,y) = –1350	cov(x,y) = 1470
cov(y,y) = 1800	cov(y,y) = 1800	cov(y,y) = 900
r = +0.75	r = –0.75	r = 0.98



Principal Component Analysis 5

- Covariance matrix:

$$\Sigma = \begin{pmatrix} \mathbf{cov}(\boldsymbol{x},\boldsymbol{x}) & \mathbf{cov}(\boldsymbol{x},\boldsymbol{y}) \\ \mathbf{cov}(\boldsymbol{y},\boldsymbol{x}) & \mathbf{cov}(\boldsymbol{y},\boldsymbol{y}) \end{pmatrix}$$

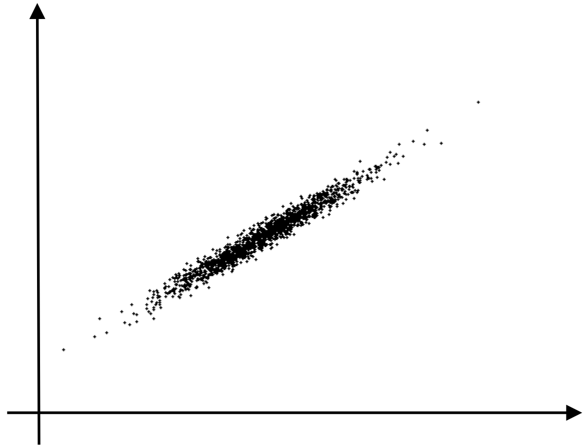
- Properties:
  - Σ is symmetric and positive $\rightarrow$ diagonalizable
  - $\exists$ rotation matrix R so that $R^{-1}\Sigma R$ is diagonal
  - If the rotation R is applied to the data:
    - Σ becomes diagonal
    - r becomes 0
    - the x and y components becomes decorrelated
    - redundancy is removed
    - Independent components can be sorted according to their variance (square root of the diagonal term)

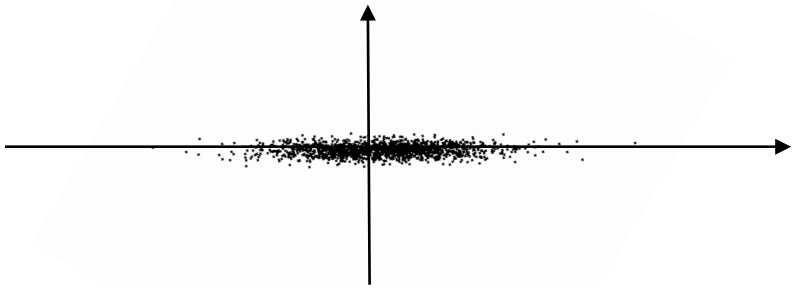
Principal Component Analysis 6

- Rotation (and translation) of the data

cov(x,x) = 2500
cov(x,y) = 1470
cov(y,y) = 900
r = 0.98

cov(x,x) = 3364
cov(x,y) = 0
cov(y,y) = 49
r = 0





Principal Component Analysis 7

- Generalization from sets of two-dimensional samples $\{(x_i, y_i), i \in \{1 \dots N\}\}$
  to sets of D-dimensional samples
  $\{(x_{i1}, x_{i2} \quad \dots x_{iD}), i \in \{1 \dots N\}\}$

$$\Sigma_{\boldsymbol{jk}} = \mathbf{cov}(\boldsymbol{x}_{.\boldsymbol{j}}, \boldsymbol{x}_{.\boldsymbol{k}}) = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{i}=\boldsymbol{N}} (x_{ij} - \overline{\boldsymbol{x}_{.\boldsymbol{j}}})\,(x_{ik} - \overline{\boldsymbol{x}_{.\boldsymbol{k}}})$$

- Σ is a $D \times D$ symmetric and positive matrix that can be diagonalized as $R^{-1}\Sigma R$
- Data can be rotated and centered accordingly into decorrelated components of decreasing variance

Principal Component Analysis 8

- With real high-dimensional sets of samples, the variance of the decorrelated components decreases very rapidly
- If correlation is high in the data, many of the last components have very small variances
- Dropping the components with very small variance does not significantly change the results
- Dropping components whose variance is smaller than the level of noise even improve performance
- Dropping components is a linear projection

Principal Component Analysis 9

- PCA summary:
  - Translation to center of data (removing mean vector)
  - Rotation to the principal axes (from covariance matrix)
  - Projection on the “big variance” axes (dropping of small variance components)
- PCA (almost) preserve the Euclidean distance
  - Translation and rotation are isometries: they preserve Euclidean distance
  - Projection dropping only small variance axes is close to an isometry: Euclidean distance is almost preserved
- Real data do not follow normal distributions but do exhibit significant correlations anyway

User interface

- Classical interface for the part of the query given at the semantic level (e.g. text input for keywords),
- Plus possibility to define a query at the signal level:
  - Query by example : one or several images or video segments, initially given or selected during relevance feedback,
  - Library of signal elements : colors, textures, shapes (that could be entered as sketches),
  - Possibility to define a relative importance for the various signal (or semantic) features available,
  - Possibility to define a fusion method for the correspondence functions (sum, product, min, …),
  - The system can also make these choices by analysis of the relevance feedback,
  - Link between signal and semantics.

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.