Modeling and Processing of Multimedia Data

International Second cycle degree programme (LM) in
Digital Humanities and Digital Knowledge (DHDK)
University of Bologna

Multimedia Information Retrieval – Part II

Home page: http://www-db.disi.unibo.it/courses/DMMMDB/
Electronic version: 2.02.MultimediaInformationRetrieval-II.pdf
Electronic version: 2.02.MultimediaInformationRetrieval-II-2p.pdf

Outline

- Description models for MM data retrieval
- Low-level features for MM data content representation
- Similarity measures for MM data content comparison
- Region-based image retrieval
  - The Windsurf system

MM data retrieval

- From the previous lesson we know that *features are a smarter way to represent MM data content than their original format*
  - e.g., color and texture for an image
- Today we focus on which are the *most suitable models* for
  - representing, interpreting, describing and
  - comparing such features
    - E.g., *color histograms for images by using the Euclidean distance as similarity measure*

- …with the *final goal to be able to retrieve from MM collections those objects which are most interesting for us*!! ☺

Content-based search

- First approach to search for MM objects relies on standard *text-based techniques*, provided objects come with a precise textual description of what they represent/describe, i.e., of their semantics
- However, the "annotation" of MM objects is a *subjective, time consuming, and tedious* process (completely manual!!)
- A more convenient approach, suitable to manage large DBs, is to

  automatically extract from MM objects
  a set of (low-level) relevant numerical ***features***

  that, at least partially, convey some of the semantics of the objects
- Clearly, which are *the "best" features to extract depend on the specific medium and on the application at hand* (i.e., what we are looking for)

Look for cheetahs?

This is fine; but, how to find it?

Content-based similarity search

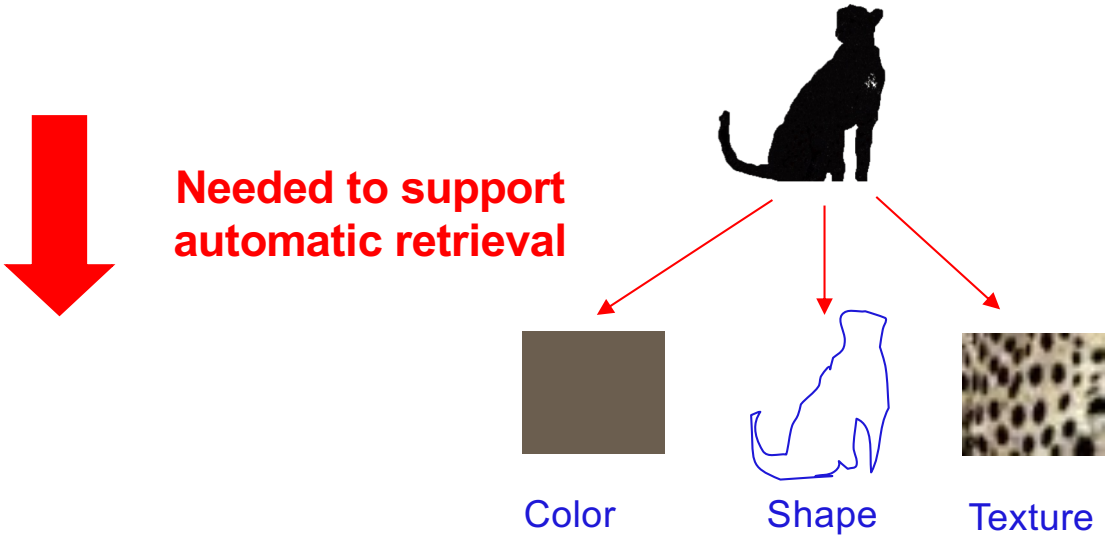
- Once we have feature values, we can search objects by using them
- Assume a database (DB) with *N* MM objects (e.g., images) and, for each of the *N* objects, we have extracted the "relevant features"
  - E.g., we could extract some color information from images
- We can now search for objects whose feature values are "***similar***" (in some sense to be defined) to the feature values of our query [SWS+00, LSD+06, LZL+07, DJL+08]
- In general, this approach, much *alike as it happens in text-retrieval*, *cannot guarantee that all and only relevant results are returned as result of a query*

Look for cheetahs?

Oops! Not really a cheetah ;-)

The general scenario

- In general, we have a 2-levels scenario:

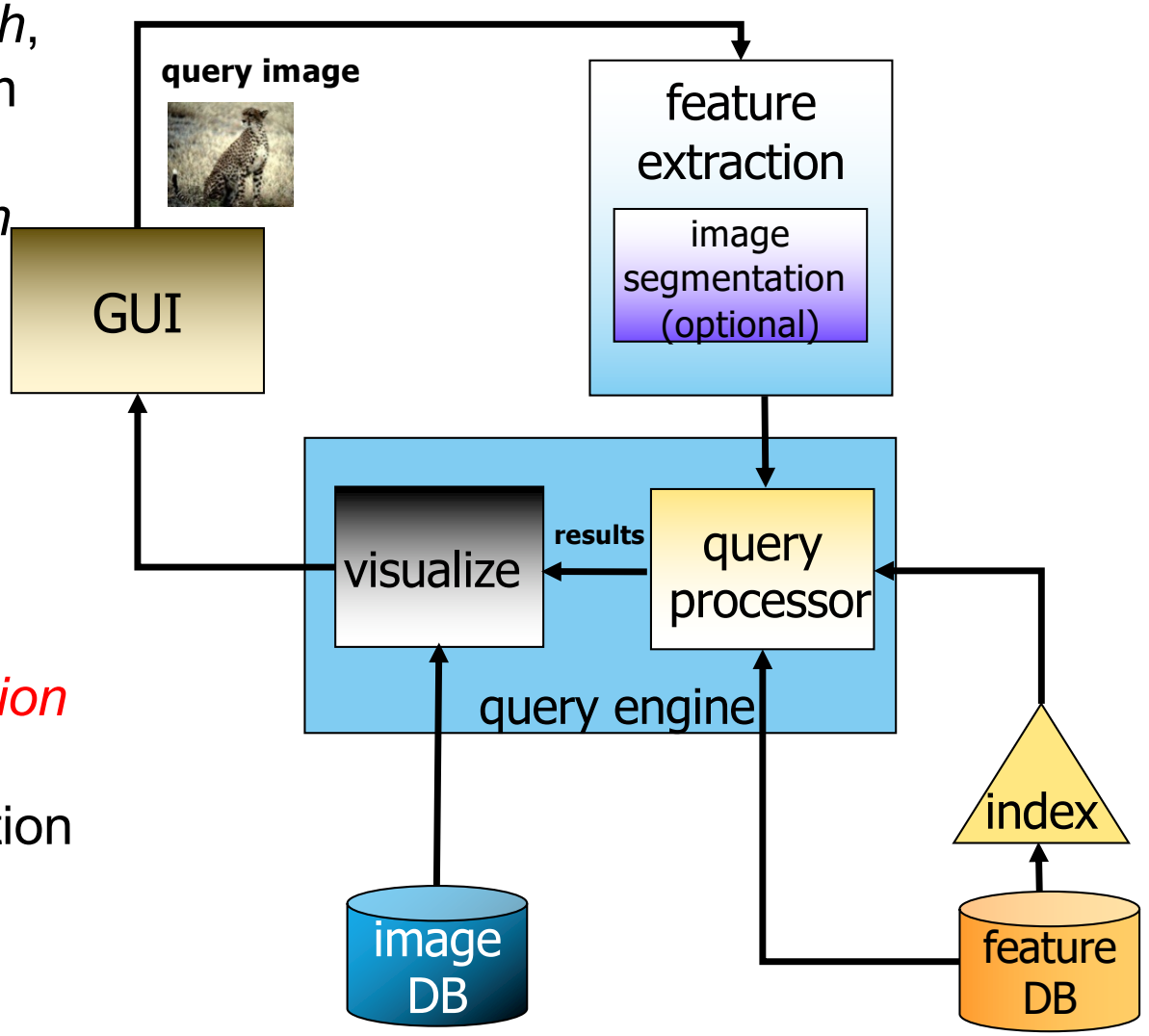


The objects level
Ultimately, we want to *find relevant objects*

The features level
We extract features from the objects, and use them for querying the DB

The reference architecture

- For the *text-based approach*, the image querying problem can be simply transformed into a *traditional information retrieval problem*
  - as we will see speaking about *MM data annotation*…



- For *content-based information retrieval* (CBIR) more sophisticated query evaluation techniques are required

Variables of the CBIR problem

- How the *set of relevant results* is determined depends
  - on *which low-level features are used* to characterize the MM data content
  - on the *similarity criterion* (distance function) used to *compare* such features
  - on how DB objects are *ranked with respect to the query*
  - on whether the user is interested in the whole MM data query or only in a part of it

- All these aspects strongly influence the *query evaluation process!*

- Simplest case: each MM data object (i.e., image) is characterized using *global low-level features* and the *result of a query consists in the set of DB objects* that "better match" the visual characteristics of the target object, according to a predefined *similarity criterion*, which is in turn based on such features
  - *This is also defined Nearest Neighbors (NN) search problem*

Representing color

- In a digital image, the *color space that encodes the color content of each pixel of the image is necessarily discretized*
  - This depends on how many bits per pixel (bpp) are used

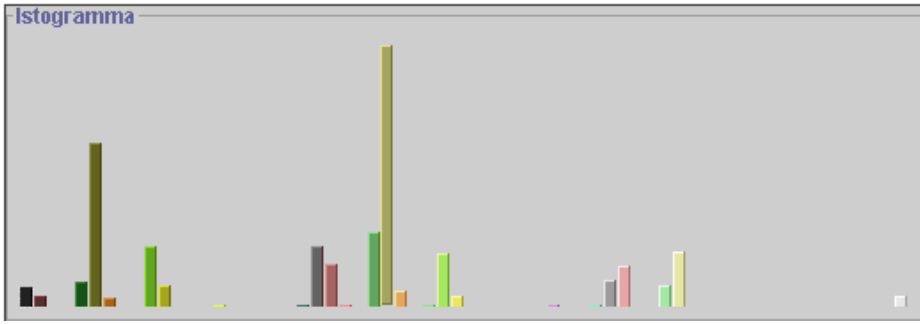
  Example:
  - if one represents images in the RGB space by using $8 \times 3 = 24$ bpp, the number of possible distinct colors is $2^{24} = 16{,}777{,}216$
  - With *8 bits per channel*, we have *256 possible values on each channel*
- Although discrete, the possible color values are still too many if one wants to compactly represent the color content of an image
  - This also aims at achieving some robustness in the matching process (e.g., the two RGB values (123,078,226) and (121,080,230) are almost indistinguishable)
- In practice, a common approach to represent color is to make use of histograms…

Color histograms

- A color histogram h is a D-dimensional vector, which is obtained by *quantizing the color space into D distinct colors*
  - Typical values of D are 32, 64, 256, 1024, …

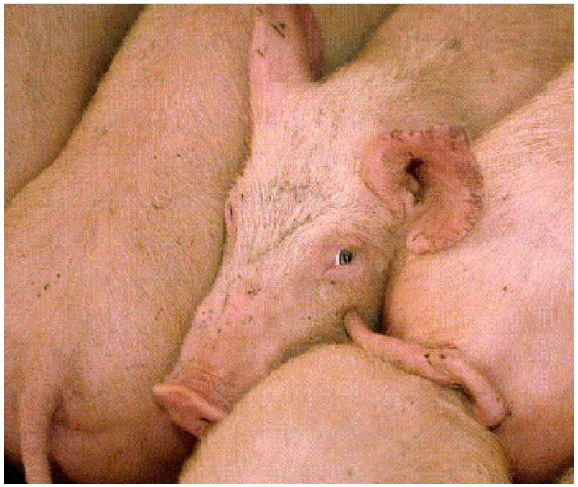
  Example: the HSV color space can be quantized into D=32 colors:
  H is divided into *8* intervals, and *S* into *4*
  V = *0* guarantees invariance to light intensity
- The *i-th* component (also called *bin*) of *h stores the percentage (number) of pixels in the image whose color is mapped to the i-th color*
- Although conceptually simple, *color histograms are widely* used since they are relatively invariant to translation, rotation, scale changes and partial occlusions

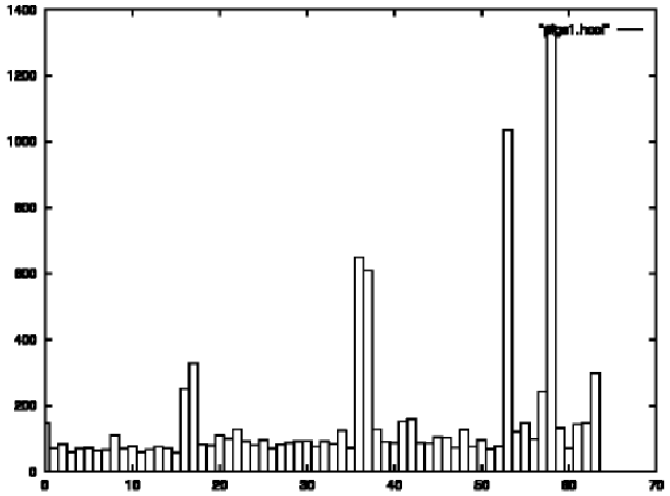
D = 64

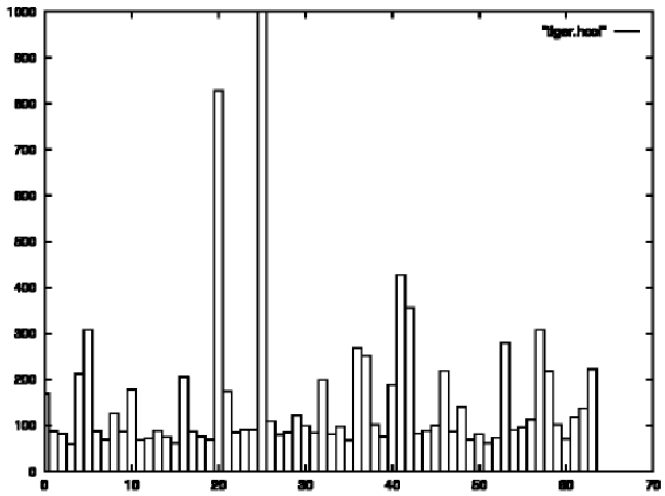


Further examples

- Two D=64 color histograms



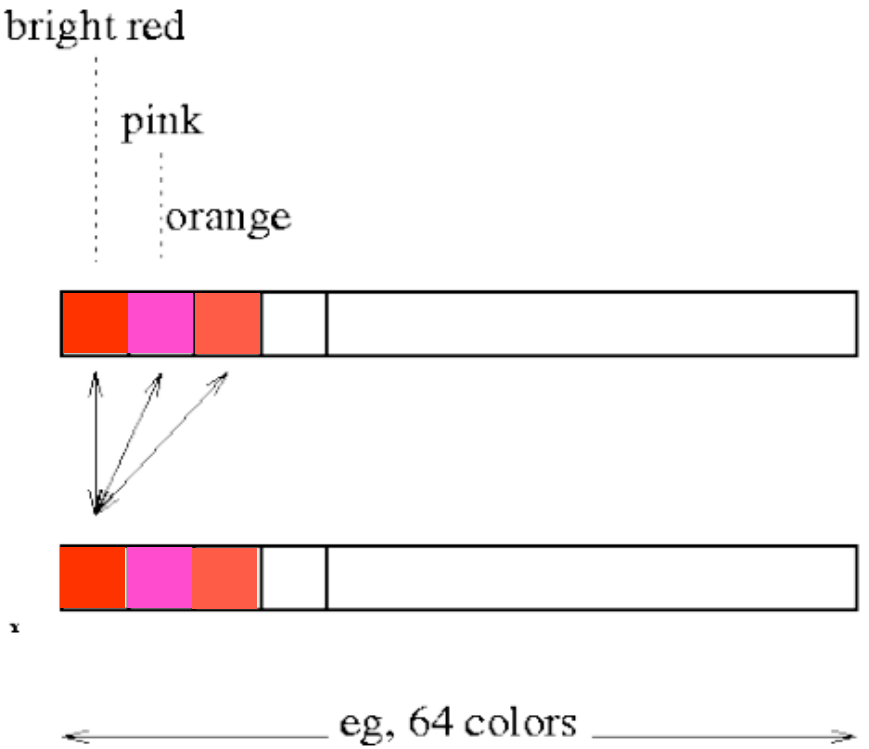




Comparing color histograms

- Since histograms are vectors, we can use any Lp-norm to measure the *distance* (*dissimilarity*) of two color histograms
- However, doing so we are not taking into account colors' correlation
  - Depending on the query and the dataset, we might therefore obtain low-quality results
  - Weighted Lp-norms and relevance feedback can partially alleviate the problem…

- The *problem is that Lp-norms just consider the difference of corresponding bins, i.e., they perform a 1-1 comparison*
- With color histograms, our "coordinates" are not unrelated ("cross-talk" effect)



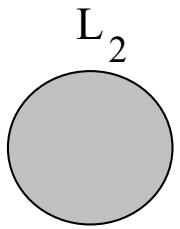
Recall on Lp-norms

- Given two *D*-dimensional vectors *p* and *q*, their distance in the reference *D*-dimensional space based on Lp-norm is:

$$L_p(p,q) = \left(\sum_{i=1}^{D} (|p_i - q_i|)^p \right)^{1/p} \qquad 1 \le p < \infty$$

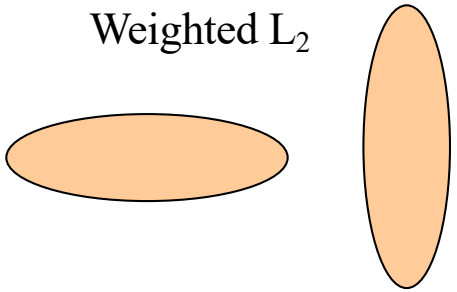
- A relevant example is the *Euclidean* distance (*p*=*2*):

$$L_2(p,q) = \left(\sum_{i=1}^{D} (|p_i - q_i|)^2 \right)^{1/2}$$



and its *weighed version*:

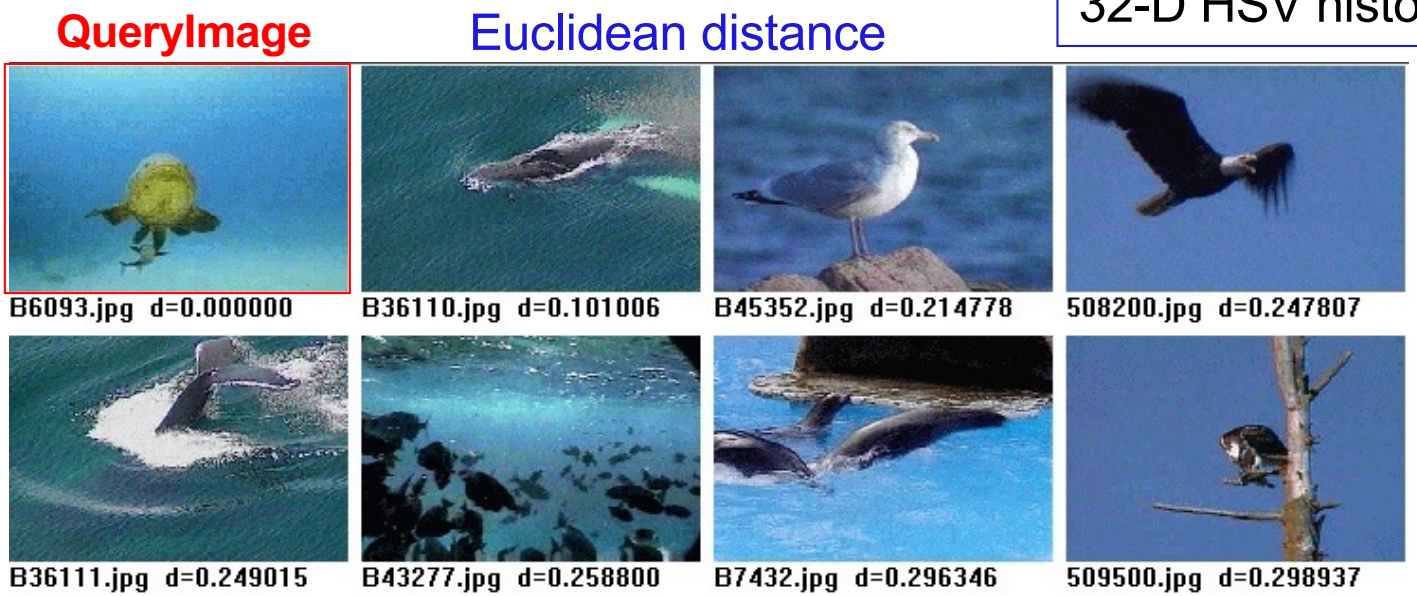
$$L_{2,W}(p,q,W) = \left(\sum_{i=1}^{D} w_i (|p_i - q_i|)^2 \right)^{1/2}$$

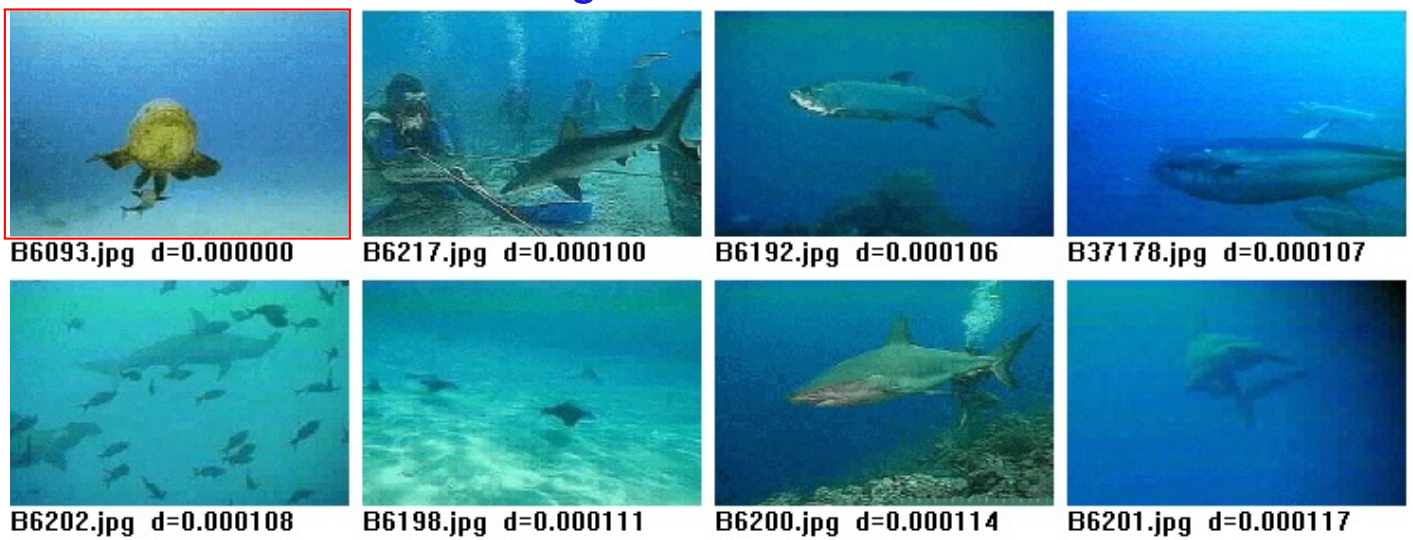


where $W(w_1, ..., w_D)$ is the *vector of weights* that reflect the importance of each coordinate of the D-dimensional space

Sample queries based on color (1)

32-D HSV histograms





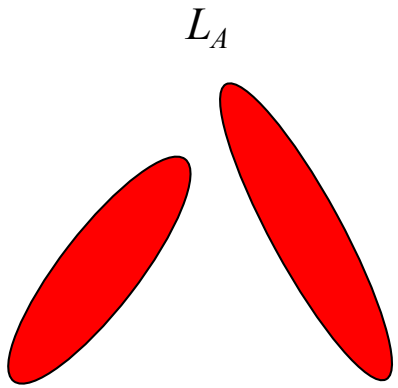
Sample queries based on color (2)

32-D HSV histograms

Quadratic distance

- Consider two histograms *h* and *q*, both with *D* bins
- Their quadratic distance is defined as:

$$L_A(h,q;A) = \sqrt{\sum_{i=1}^{D}\sum_{j=1}^{D} a_{i,j}(h_i - q_i)(h_j - q_j)}$$
$$= \sqrt{(h-q)^T \times A \times (h-q)}$$



where $A = \{a_{i,j}\}$ *is called the (color-)similarity matrix*

- The value of $a_{i,j}$ is the "*similarity*" of the i-th and the j-th colors ($a_{i,i} = 1$)
- Note that:
  - when *A is a diagonal matrix* we are back to the *weighted Euclidean distance*,
  - when *A = I (the identity matrix)* we obtain the L_2 distance

Quadratic distance vs. Euclidean distance

- As a simple example, let D = 3, with colors red, orange, and blue
- Consider 3 pure-color images and the corresponding histograms:

h1=(1,0,0) **h2=(0,1,0)** **h3=(0,0,1)**

- Using L_2, the distance between two different images is always $\sqrt{2}$
- On the other hand, let the color-similarity matrix be defined as:

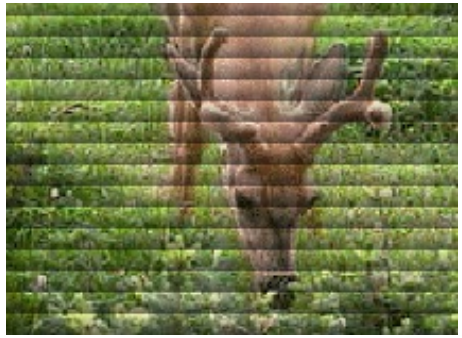
A			
	1	**0.8**	**0**
	0.8	**1**	**0**
	0	**0**	**1**

- Now we have $L_A(h1,h2) = \sqrt{0.4}$, whereas $L_A(h1,h3) = L_A(h2,h3) = \sqrt{2}$

Representing texture (1)

- Unlike color, texture is not a property of the single pixel, rather it is a collective property of a pixel and its, suitably defined, “neighborhood”

“mosaic” effect

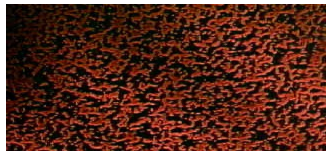


“blinds” effect

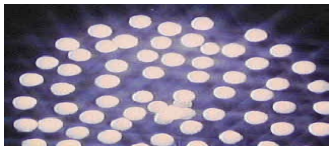
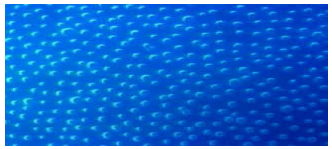
- Intuitively, texture provides information about the uniformity, granularity and regularity of the image surface
- It is usually computed just considering the gray-scale values of pixels (i.e., the V channel in HSV)

Representing texture (2)

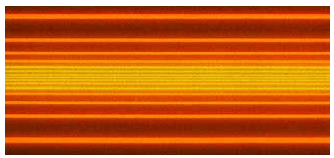
- *Tamura features* correspond to properties of a texture which are *readily perceived*, that is *coarseness, contrast and directionality* (*3-D* feature vector)
  - Coarseness - coarse vs. fine: it provides information about the "granularity" of the pattern

  - Contrast - high vs. low contrast: it measures the amount of local changes in brightness

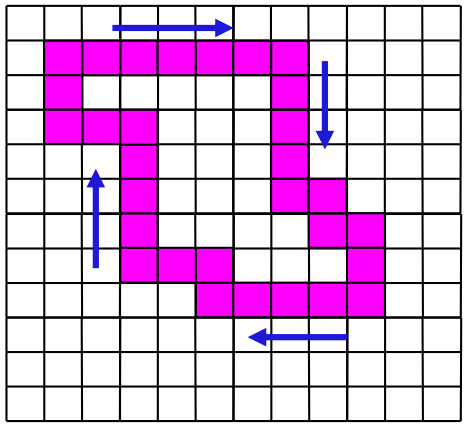
  - Directionality - directional vs. non-directional: it's a global property of the image




Representing shape

- Once one has succeeded in extracting an object's contour, the next step is how to represent/encode it
- A common approach is to *navigate* the contour, which leads to an ordering of the pixels in the contour:

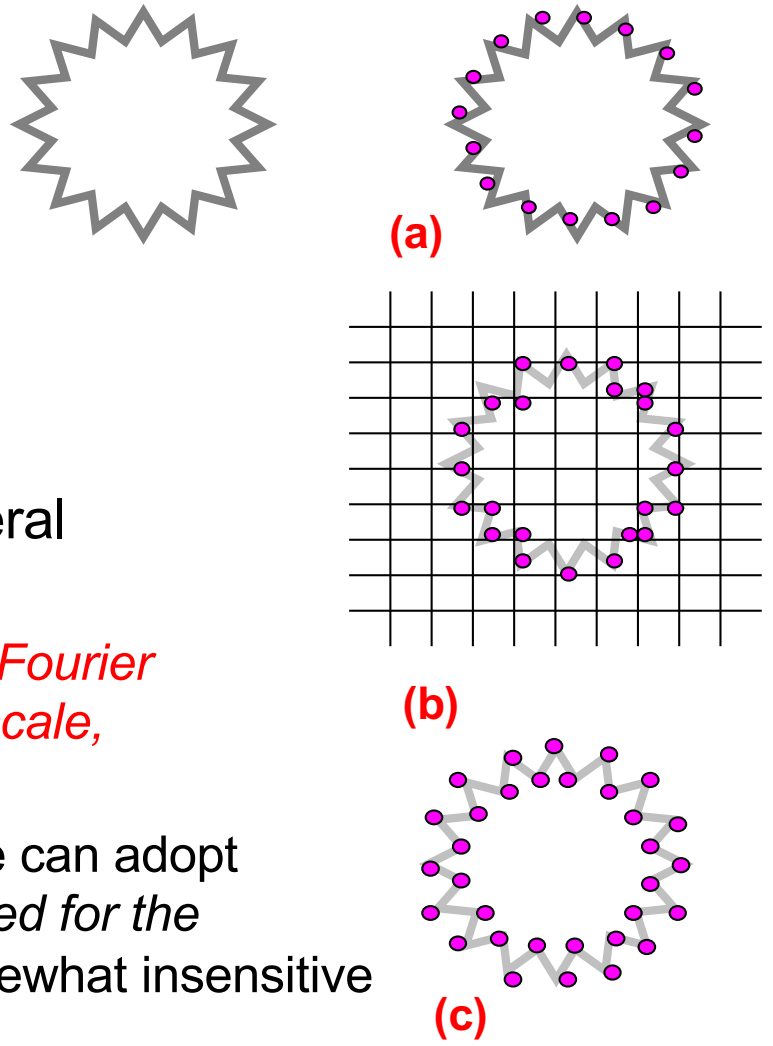
  { (x(t),y(t)) : t = 1…,M }



- A 2nd step is to represent the resulting curve in a parametric form
- For instance, a possibility is to resort to complex values, by setting $z(t) = x(t) + j\, y(t)$
- Thus, now we have vectors of complex values…
- The problem is that each vector has a different length (i.e., *M depends on the specific image*)

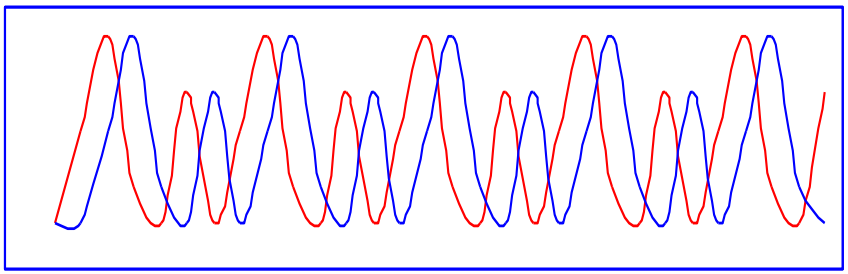
Representative points

- The *idea is to keep only the D most "interesting" points*
- Some methods are:
  - *Equally-spaced sampling* (a)
  - *Grid-based sampling* (b)
  - *Maximum curvature points* (c)
  - *Fourier-based methods*, which first compute the DFT of the contour, and then keep only the first D coefficients
- Working in the frequency domain has several advantages:
  - It can be proved that *by properly modifying Fourier coefficients one can achieve invariance to scale, translation and rotation*
  - Further, by viewing shape as a "signal", one can adopt *distance measures that have been developed for the comparison of time series* and that are somewhat insensitive to signals' modifications



Comparing shapes

- The commonest way to measure the (dis-)similarity of two shape vectors of equal length D is based on Euclidean distance (L_2)
- However, with Euclidean distance we have to face a basic problem
  - Sensitivity to "alignment of values"



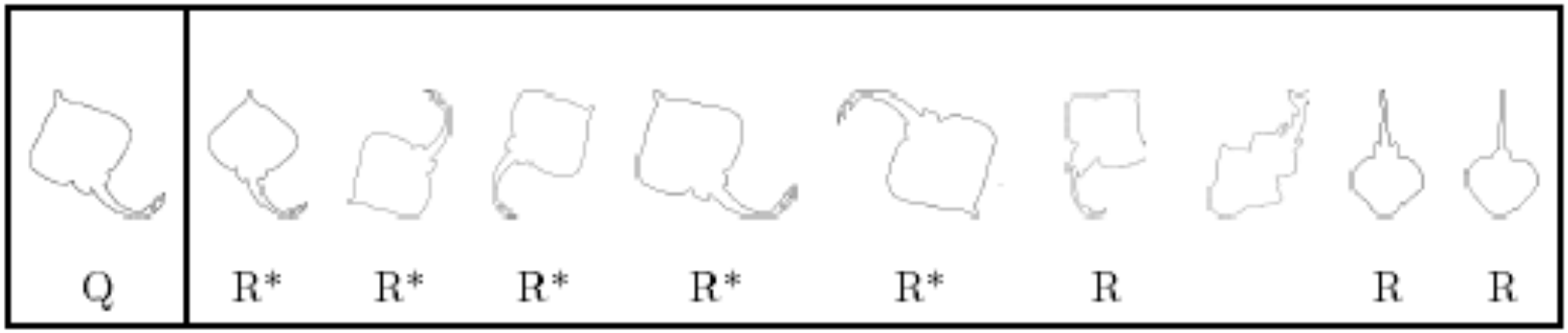
- Intuitively, we would need a distance measure that is able to "match" a point of time series *s* even with "surrounding" points of time series *q*
  - Alternatively, we may view the time axis as a "stretchable" one
- A distance like this exists, and is called "Dynamic Time Warping" (DTW)

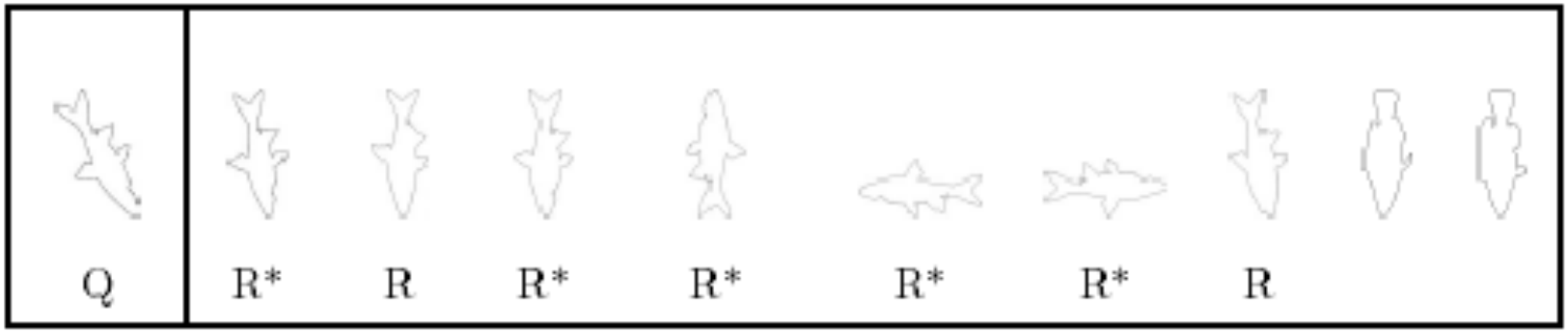
Sample queries based on shape [BCP02]

R = relevant
(same type of fish)

1100 objects' contours

QueryImage





This is not the whole story…

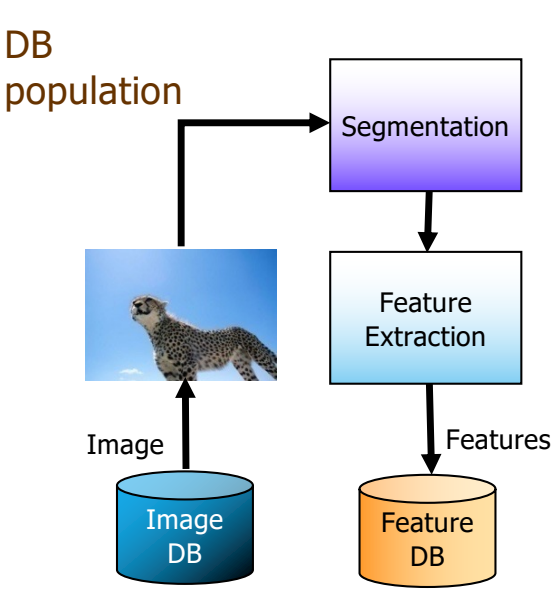
- …of course, many other features models (and correspondent distance functions) have been defined for MM data
- This was just a way to provide some concrete examples of features and modalities to comparing them!

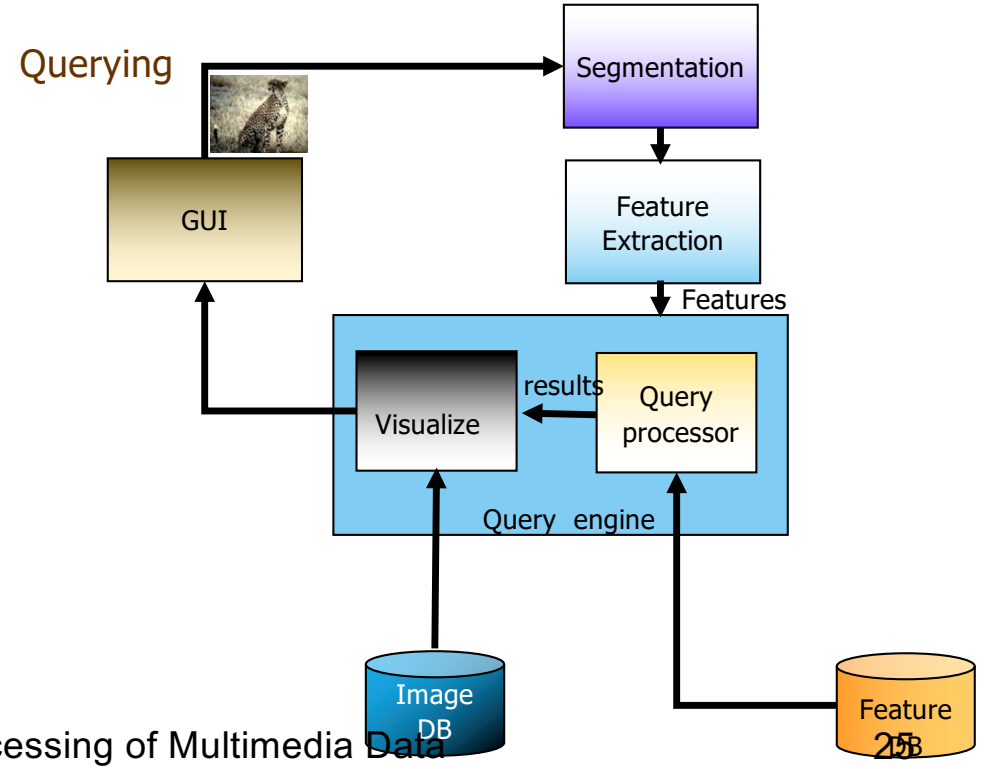
- Note that, besides "generic" features, any specific image domain/application needs to extract and manage specific features, which in general require much more sophisticated tools than the one we have seen
  - E.g., face/fingerprints recognition
- Nonetheless, what is important to stress is that *the problem of how to search in large image DB's remains (almost) the same*! ☺

- Let's go now into the details of what happens and how things can become complex in a "real" image retrieval system…

The region-based image retrieval approach

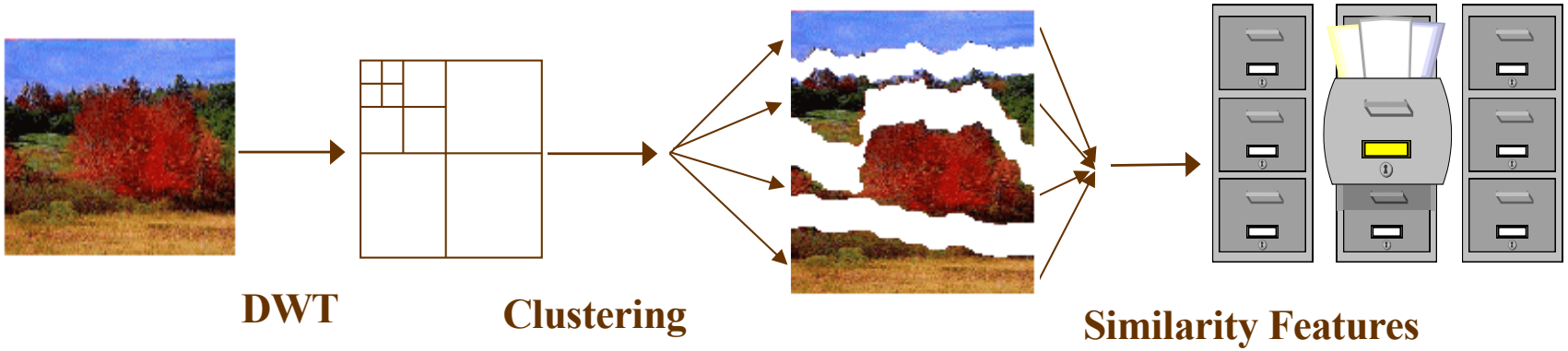
- DB population time:
  - Preprocess images to segment them into regions
  - Represent regions as vectors of features
- Query time:
  - Compare query regions to DB regions
  - Assess similarity between images by *combining* similarity between regions





Windsurf case study [ABP99, BCP00, BP00, BC03, Bar09a, BCP+09, BCP10]

- Windsurf: Wavelet-Based Indexing of Images Using Regions Fragmentation
  - Discrete Wavelet Transform (DWT): extracts a set of features representing the image in the color-texture space
  - Clustering: fragments the image into a set of regions using wavelet coefficients
  - Similarity Features: used to compare regions

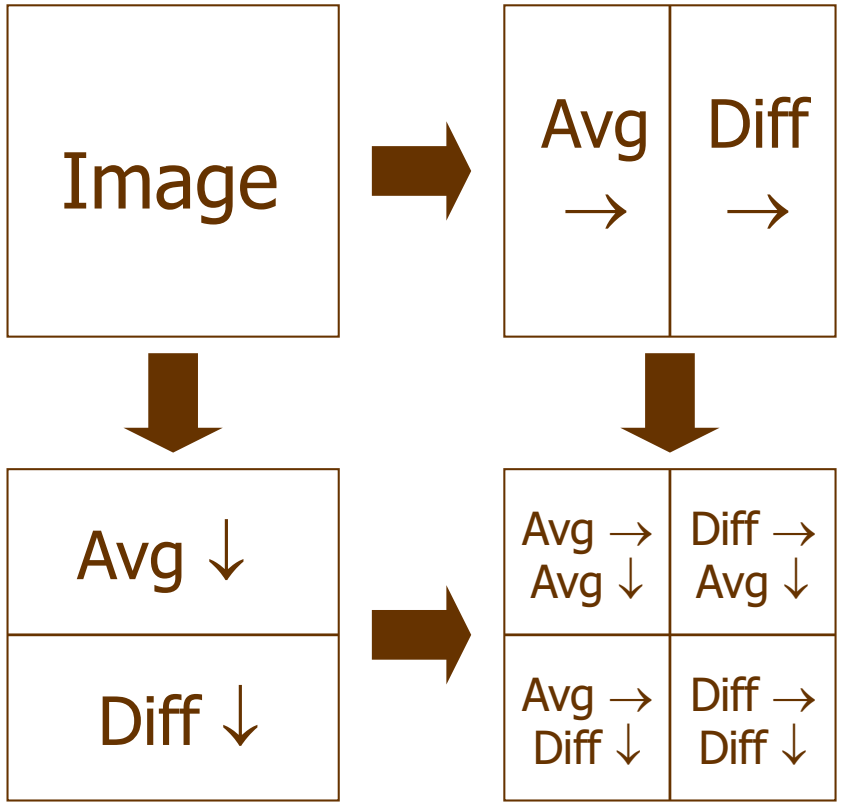


Discrete Wavelet Transform (DWT)

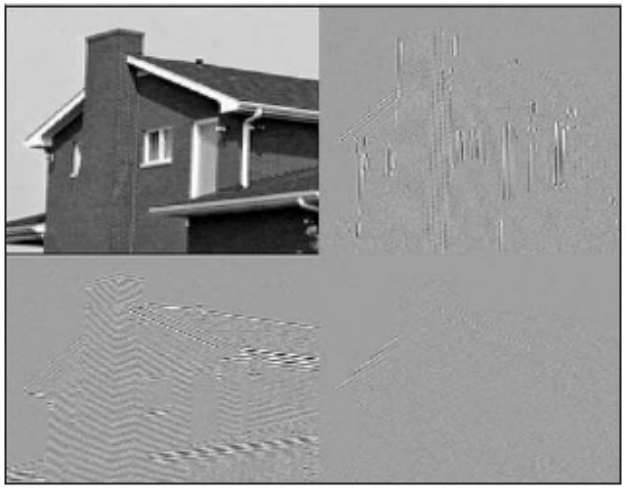
- Haar wavelet:
  simple and quick
- Each coefficient is defined by:
  - level DWT (l)
  - frequency sub-band (B)
  - color channels (H, S, V)

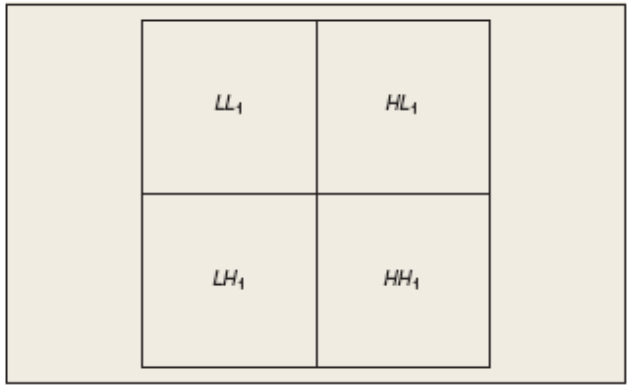
$$w_j^{l;B} = \left(w_0^{l;B}, w_1^{l;B}, w_2^{l;B} \right)$$

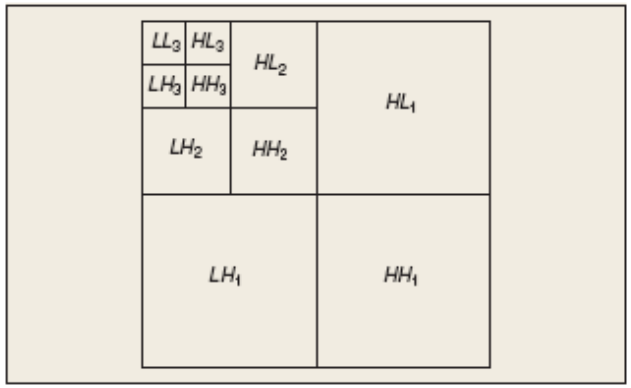
$$B \in \{LL, LH, HL, HH\}$$



DWT: practical example







Clustering (1)

- *K*-means algorithm (3rd level and low frequence info)
  - Choose *k* initial centroids;
  - Associate each point to its nearest centroid;
  - Recompute centroids and repeat previous step;
  - Stop when solution does not change.
- Mahalanobis distance:

$$\delta\left(w_i^{3,LL}, w_j^{3,LL}\right)^2 = \left(w_i^{3,LL} - w_j^{3,LL}\right)^T . \left(C^{3;LL}\right)^{-1} \cdot \left(w_i^{3,LL} - w_j^{3,LL}\right)$$

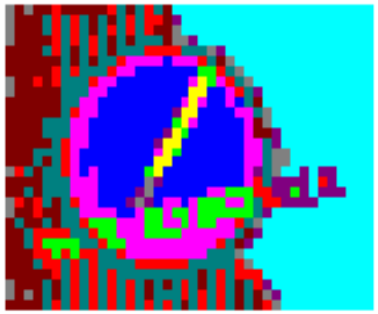
  - Correlation between wavelet coefficients takes into account variations in color, i.e. texture

Clustering (2)

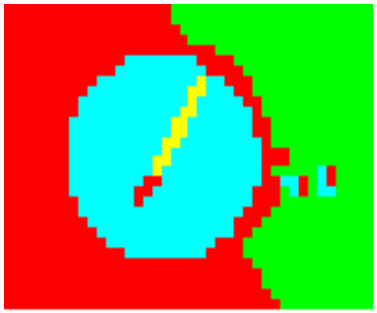
- Optimal value for *k*?
- Minimization of a validity function
  - Intra-cluster distance
  - Clusters' size
  - Inter-cluster distance

Input image

Clusters for *k*=2



Clusters for *k*=10



Clusters for *k*=4 (Optimal solution)

Similarity features

- Region similarity with Bhattacharyya distance
  - Regions are ellipsoids in 37-D feature space (all frequencies info is used)
    - (3-D centroid + 6-D covariance matrix + 1-D region size)
  - Distance between regions' centroids (color info)
  - Covariance matrices (texture info)

$$d_B(R_i, R_j)^2 = \frac{1}{2}\ln\left(\frac{\left|\frac{C_{R_i}^{3;B} + C_{R_j}^{3;B}}{2}\right|}{\left|C_{R_i}^{3;B}\right|^{\frac{1}{2}} \cdot \left|C_{R_j}^{3;B}\right|^{\frac{1}{2}}}\right) + \frac{1}{8}\left[\left(\mu_{R_i}^{B} - \mu_{R_j}^{B}\right)^T \cdot \left(\frac{C_{R_i}^{3;B} + C_{R_j}^{3;B}}{2}\right)^{-1} \cdot \left(\mu_{R_i}^{B} - \mu_{R_j}^{B}\right)\right]$$

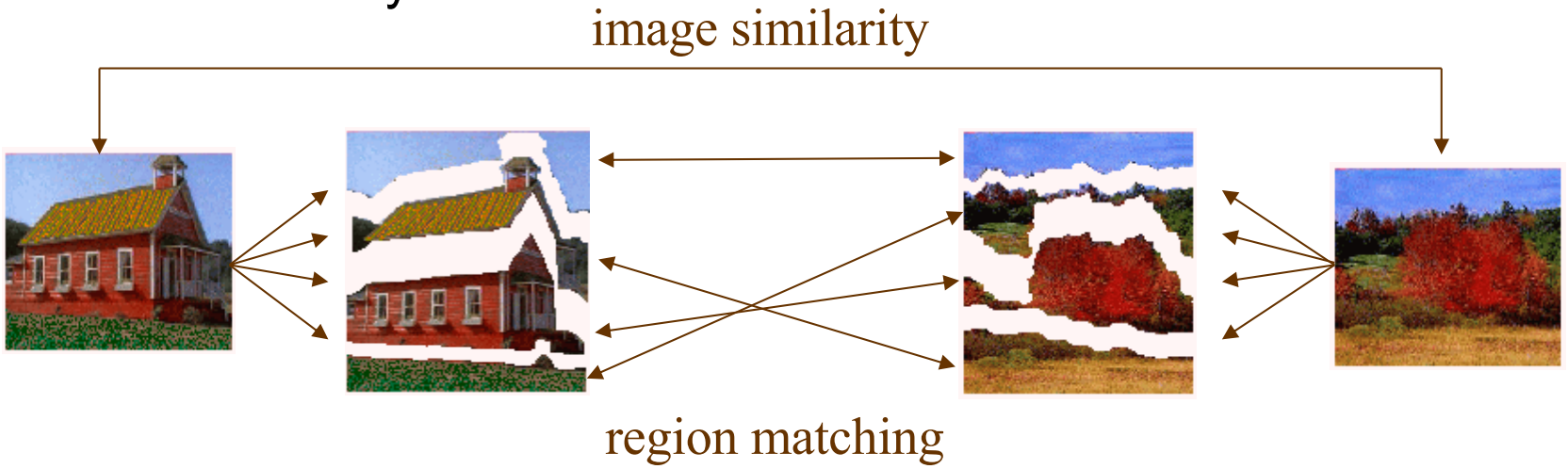
Image similarity

- Similarity between images is a function of similarities among “matched” regions
- How regions are "matched" can therefore strongly influence the result of a query

- Example: “one-to-one” match (formulated as Assignment Problem)

Assignment problem

- Goal: “Find the optimal match where unit elements of fixed size are matched individually”



- Implemented with the Hungarian algorithm, maximizing a function that is monotonic in the similarity scores (e.g. average)

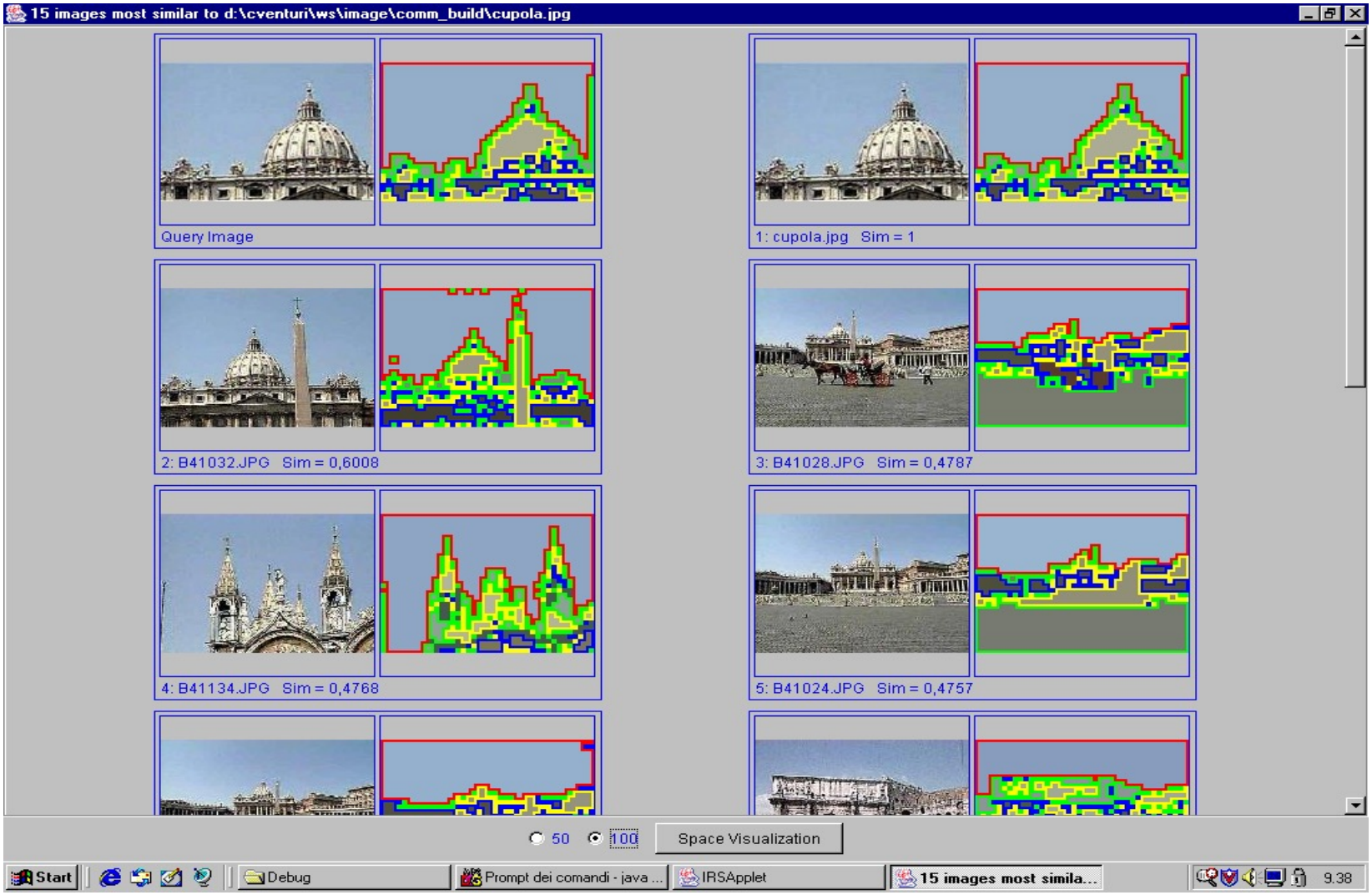
	r_1	r_2	r_3	r_4	r_5
q_1	.52	.17	.41	.16	.29
q_2	.27	.19	.81	.35	.49
q_3	1.0	.11	.27	.24	.29

(.52+.81+1.0)/3=.77

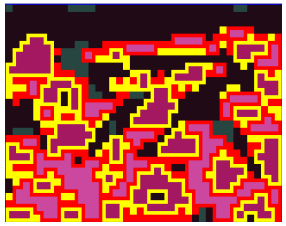
	r_1	r_2	r_3	r_4	r_5
q_1	.52	.17	.41	.16	.29
q_2	.27	.19	.81	.35	.49
q_3	1.0	.11	.27	.24	.29

(.29+.81+1.0)/3=.7

Sample query

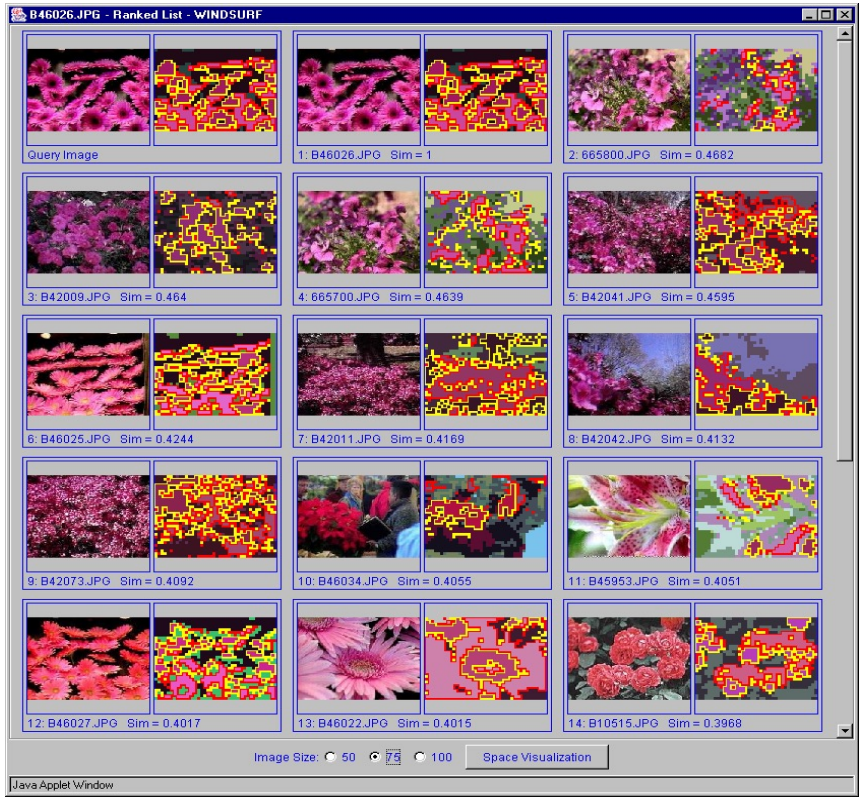


Effectiveness comparison example

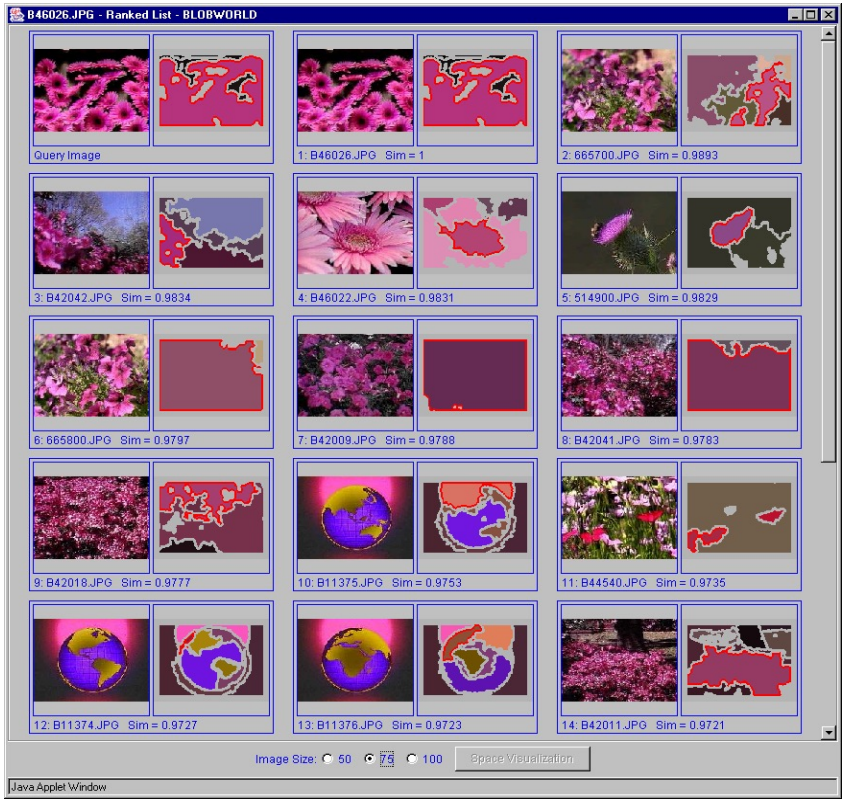




"**Flowers**" query **Winsdurf** clusters **Blobworld** [CTB+99] clusters

Windsurf



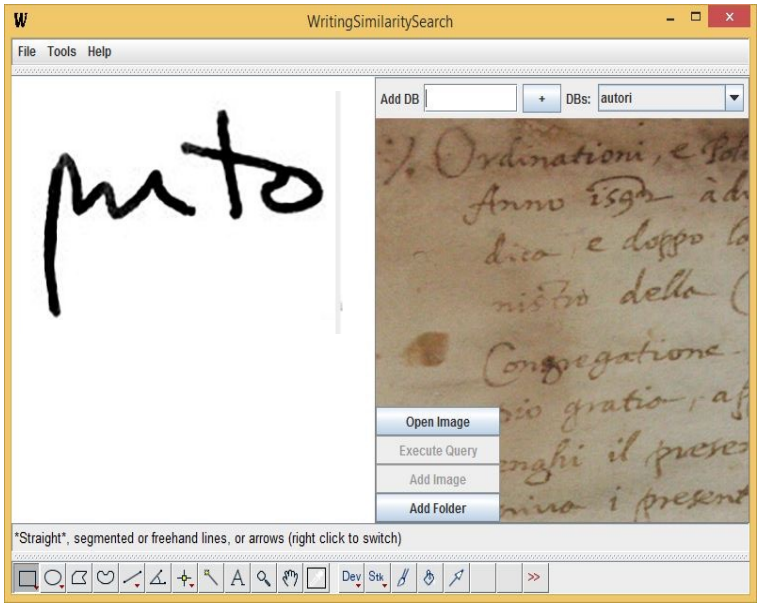
Blobworld



Windsurf in a specialized context: handwritings

- To provide an example of the *generality* of the *Windsurf framework*, here we show a pic of the *WritingSimilaritySearch system* built on top of the Windsurf system

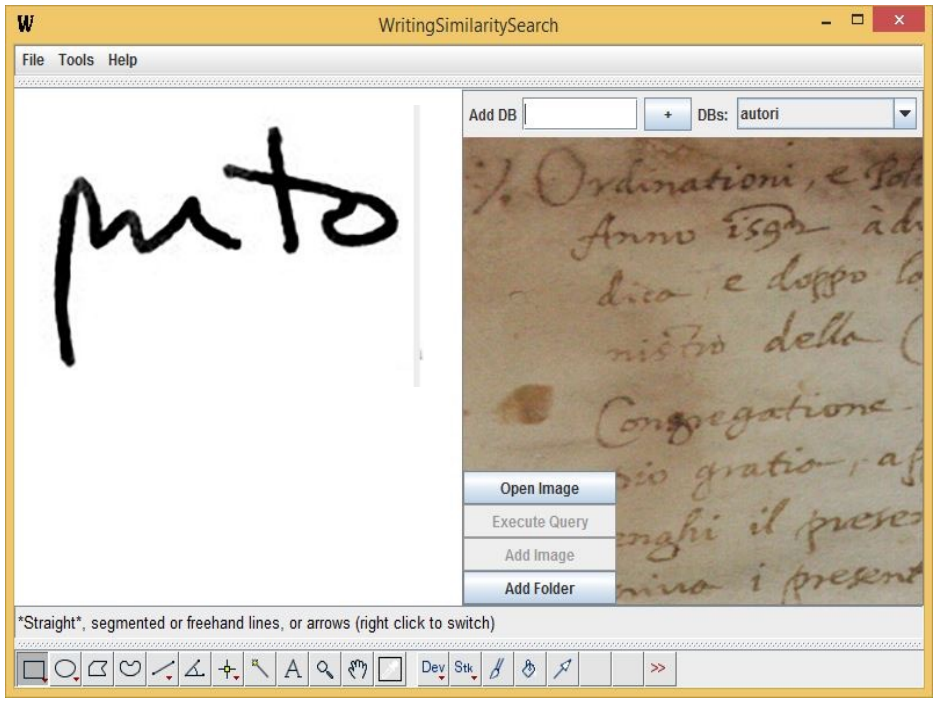
- Let's instantiate the "*key points" of the Windsurf* within the new context:
  - Regions correspond to *local features* (i.e., key points of SURF)
  - Region distance function is the *Euclidean* distance
  - The matching problem is solved by means of an "*approximation*" of the 1-1 matching
    - *"best bin first" match*

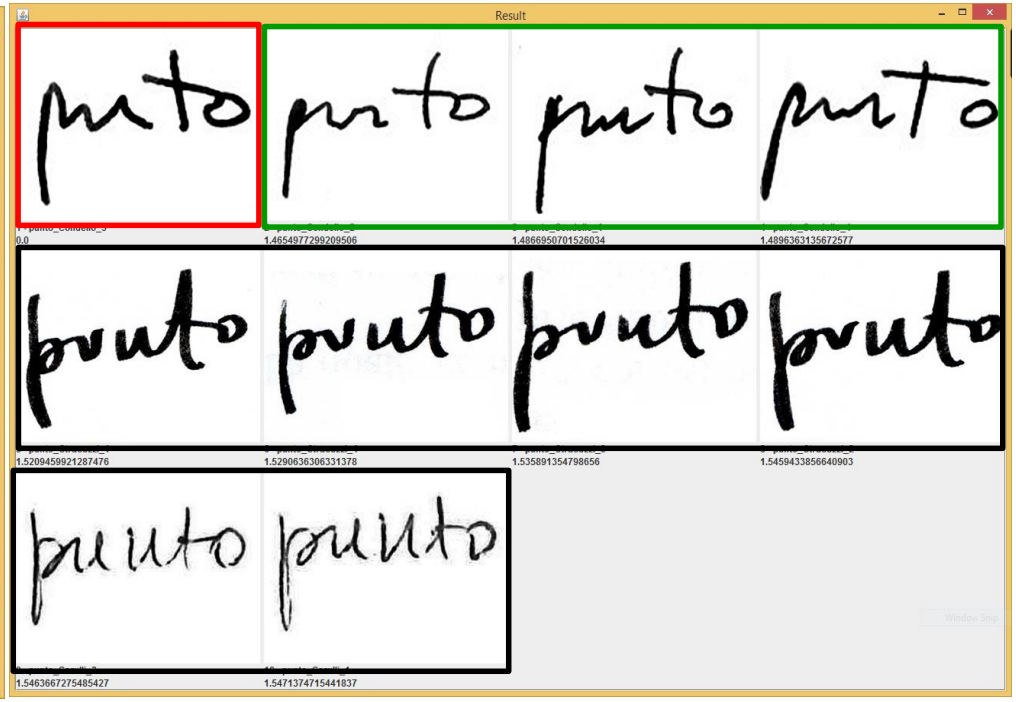


WritingSimilaritySearch: an example

query

Top-*k* results (k=9)





Exercise 1.D

- Let's complete our Exercise 1, in its last part "D"…
- Starting from the **definitions of the low-level features** you selected for describing the **"content" of unstructured data** involved in your MM applications, provide a **concrete representation/comparison modality** of them with visual examples
- Among features possibilities:
  - global features vs. local features (region-based approach) and/or
  - local salient points

E.g., "*global color distribution for an image*" (definition) vs. "*color histograms*" by using the "*weighted Euclidean distance*" as similarity measure (representation/comparison modality)

Exercise 1.D: students to do list

- Starting from the solution you proposed in Exercise 1.C (i.e., the file "**Secondname.Firstname.ESE1.C.ppt**"), enrich the description of your MM Applications by adding
  - **concrete representation/comparison modality** of involved **low-level features** with **visual examples**

- **In doing the exercise, let's keep in mind the final goal: retrieve relevant MM content!** ☺