Object of Interest-based visual navigation, retrieval and semantic content identification system

Khalid Idrissi, Guillaume Lavoué, Julien Ricard, Atilla Baskurt

▶ **To cite this version:**

Khalid Idrissi, Guillaume Lavoué, Julien Ricard, Atilla Baskurt. Object of Interest-based visual navigation, retrieval and semantic content identification system. Computer Vision and Image Understanding, 2004, 94 (1-3), pp.271-294. 10.1016/j.cviu.2003.10.014 . hal-01541465

HAL Id: hal-01541465

https://hal.archives-ouvertes.fr/hal-01541465

Submitted on 28 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial| 4.0 International License

Object of interest-based visual navigation, retrieval, and semantic content identification system

Khalid Idrissi,* Guillaume Lavoué, Julien Ricard, and Atilla Baskurt

LIRIS, FRE 2672 CNRS, INSA Lyon/UCB Lyon 1/UL Lyon 2/EC Lyon, Bâtiment Nautibus, 8 Boulevard Niels Bohr, 69622 Villeurbanne Cedex, France

This study presents a content-based image retrieval system IMALBUM based on local region of interest called object of interest (OOI). Each segmented or user-selected OOI is indexed with new local adapted descriptors associated to color, texture, and shape features. This local approach is an efficient way to associate the local semantic content with low-level descriptors (color, texture, shape, etc.) computed on regions selected by the user. So the user actively takes part in the indexing process (offline) and can use a selected OOI as a query for the retrieval system (online). The IMALBUM system proposes original functionalities. A visual navigation tool allows to surf in the image database when the user has no precise idea of what he is really searching for in the database. Furthermore, when an OOI is selected as a query for retrieval, a semantic content identification tool indicates to the user the probable class of this unknown object. The performance of these different tools are evaluated on different databases.

Keywords: Image indexing; Image retrieval; Visual navigation in image database; Semantic content analysis; Semantic content identification

* Corresponding author. Fax: +33-4-72-43-13-12.
E-mail address: kidrissi@liris.cnrs.fr (K. Idrissi).

1. Introduction

Rapid growth in the technology for multimedia acquisition, storage, transmission, visualization, and interaction, has contributed to an amazing growth in the amount of multimedia. As content generation and diffusion increase, the need for efficient tools to filter, search, and retrieve this content becomes even more acute. In recent years, the problem of content-based image retrieval (CBIR) has attracted the interest of scientists, especially on media descriptors, similarity measures, and search optimization through databases.

The architecture of a CBIR system is presented in Fig. 1. A subsystem extracts the descriptors from the images of the database (offline) and from the key image used as the query (online). The other subsystem is the search engine, which generates the images similar to this query (online). In the last few years many CBIR systems have been developed. Most of them (QBIC [1], Photobook [2], VisualSEEk [3], SWIM [4], ImageScape [5], NeTra [6], etc.) use color, texture, and shape as attributes for image description by evaluating them on the whole image. Some systems include the user in a search loop with a relevant feedback mechanism in order to adapt continuously the search parameters according to the user's choices [7]. The question that one can ask is what correlation exists between image descriptors and its semantic content, knowing that generally the user queries are based on semantic and not on low-level image features. In SIMPLIcity system [8] the authors use a semantic classification of the image database (indoor/outdoor, city/landscape, textured/non-textured, etc.) combined with a region-based image decomposition. Then the descriptors are chosen according to the semantic sense detected in the key image.

Traditional CBIR systems describe the image with global properties, calculated on the whole image. A complementary approach consists in considering only some objects of interest (OOI), which carry the most information about the image. In this perspective, we propose two approaches:

- A global one, without user interaction, where global descriptors are extracted from the whole image.

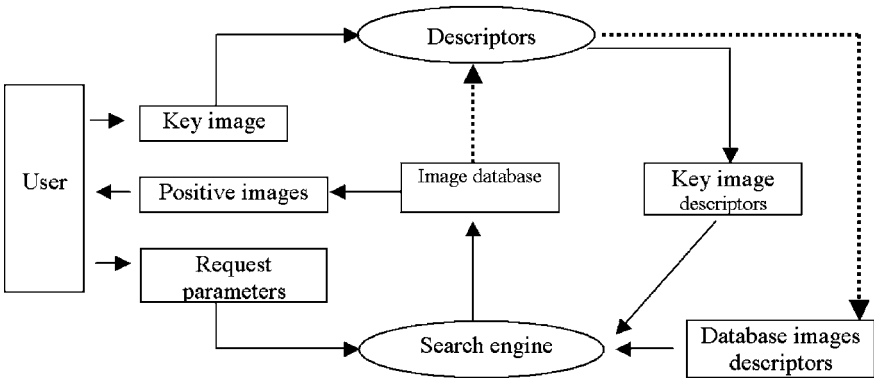

Fig. 1. CBIR generic architecture.

- A local one, where the user will have to specify some OOI from which local descriptors will be automatically extracted.

This paper focuses on the local approach assuming that the user starts a query with an OOI selected in an image. We think that this is an efficient way to enrich the CBIR system with new functionalities. In addition to the indexing and retrieval engine, the system IMALBUM presented in this study proposes an original tool for visual navigation in the image database and a semantic content analysis and identification tool.

Section 2 details the color image segmentation method. Section 3 deals with the descriptors developed for color, texture, and shape features. The similarity measures are detailed in Section 4. The IMALBUM system and its functions are presented in Section 5. Experiments and results are presented and discussed in the last section.

2. Color image segmentation

Image segmentation, which is one of the most important steps in image analysis, aims to divide the given image into uniform and homogeneous regions with respect to characteristic features. For this purpose, we introduce a new approach for color image segmentation based on cross entropy minimization [9]. In an image retrieval context, the query image segmentation as well as the extraction of key OOI descriptors are real time procedures. That is why, we privilege the use of a fast and simple segmentation. The proposed method is composed of two stages: A color reduction in the color space (quantization), and a spatial regularization which merges adjacent regions in the image space. The parameters that this method uses are evaluated automatically by taking into account factors of visual perception. Thus the number of color classes is not fixed by the user but determined by the method according to the colorimetric content of the image. The stopping criteria are the homogeneity of the color classes and their number of pixels.

2.1. Color quantization

The proposed color quantization is based on minimum cross entropy (MCE). The cross entropy [10] is a measure between two probability distributions $P = p_1, \ldots, p_n$ and $Q = q_1, \ldots, q_n$, which allows to evaluate the information theoretical distance, therefore the degree of likeness, between the two distributions:

$$E(Q,P) = \sum_{k=1}^{N} q_k \log_2 \left(\frac{q_k}{p_k} \right). \tag{1}$$

In image processing, the image is considered as a probability distribution in which the p_i probability of the ith pixel is given by the ratio between its gray level and the sum of the gray levels of all the image's pixels. Then the MCE technique is used to threshold a gray level image because it permits to obtain an optimal threshold which minimizes the dissimilarity between original and threshold images. For a given threshold value, the cross entropy between the two images is proportional to:

$$\eta(t) = \sum_{j=1}^{t-1} jh_j \ln\left(\frac{j}{\mu_1(t)}\right) + \sum_{j=t}^{L} jh_j \ln\left(\frac{j}{\mu_2(t)}\right), \tag{2}$$

where L is the number of gray levels in the image, h_j is the number of pixels with gray level j; and μ_1 and μ_2 are the mean values of the pixels belonging, respectively, to class 1 and class 2 of the segmentation map [11]. This method allows us to find the optimal threshold t_{opt}, which splits the original image histogram into two classes (Fig. 2).

An extension of this method to the use of two variables is possible [12]. Indeed, starting with a gray level image, it is possible to deduce from it an image relating to a given characteristic (for example local entropy). The image resulting from the normalized sum of the two preceding images represents a new variable M, defined as a combination of their respective variables x_1 (gray level) and x_2 (e.g., local entropy).

The application of the CEM to this new image will subdivide its 2D histogram into two classes (Fig. 3). The form of the subdivision depends on the selected combination ($M = x_1 + x_2$ in Fig. 3A and $M = \sqrt{(x_1)^2 + (x_2)^2}$ in Fig. 3B).

The segmentation of a color image is thus possible by extension of the preceding approach by considering images of the three color components and by using this process in an iterative way. The selected combination is given by:

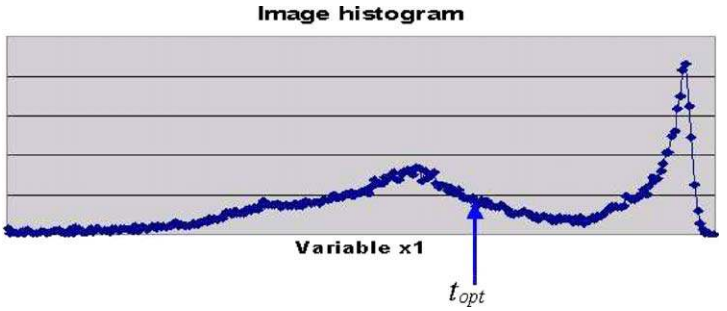

Fig. 2. The division of 1D histogram with two classes.

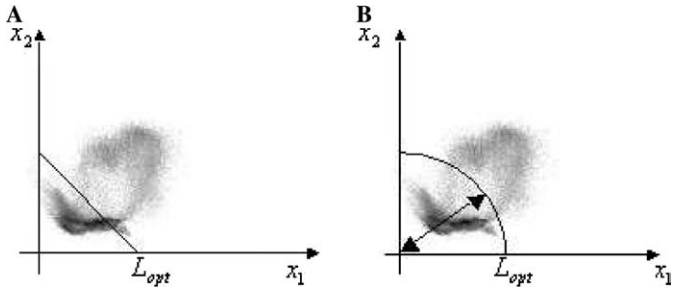

Fig. 3. Partitioning of the x_1, x_2 plane by cross entropy minimization.

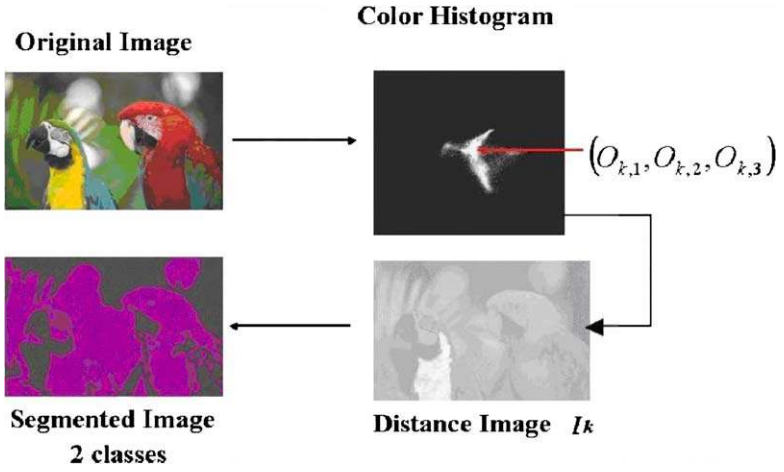

Fig. 4. Segmentation procedure for $k = 1$.

$$I_k(m,n) = \sqrt{\sum_{i=1}^{3}(C_i(m,n) - O_{k,i})^2}, \tag{3}$$

where the origin was relocated towards the centroid $CD_k = (O_{k,1}, O_{k,2}, O_{k,3})$ which represents the dominant color of the image at the kth iteration and corresponds to the maximum of the 3D histogram at this iteration. In Eq. (3), $I_k(m,n)$ represents the Euclidean color distance between the color of the pixel (m,n) belonging to the class being subdivided and the dominant color of this class, and where C_i is the ith color component of the pixel (m,n). The image I_k corresponds to the distance image between the color of each pixel and the dominant color at the kth iteration. We choose the *Lab* color space for which the Euclidean distance is a good approximation of the color distance. In the first iteration, the whole image is considered as the class to split (Fig. 4).

The thresholding of I_k with the MCE method provides the optimal threshold which splits the "distance" image into two classes: $C_{k,i}$ whose pixels are near CD_k and $\overline{C_{k,i}}$ whose pixels are far from it. This method can be repeated on each found class, splitting it into two new classes (Fig. 5). Resulting from this phase, we have a quantized image, that is to say a class-map of the original image with the label of the associated color class of each pixel. In spatial domain each class is formed with a set of similar separated regions of the original image (Fig. 5(a.*i*)), while in color domain each class is characterized by its dominant color and variance (Fig. 5(b.*i*)).

2.2. Region merging

The quantized image is composed of numerous spatial regions. The purpose of the merging step is to reduce this over-segmented image, in order to extract significant regions, corresponding to the real objects of the scene. Our approach has the advantage of selecting exactly the number of final regions.

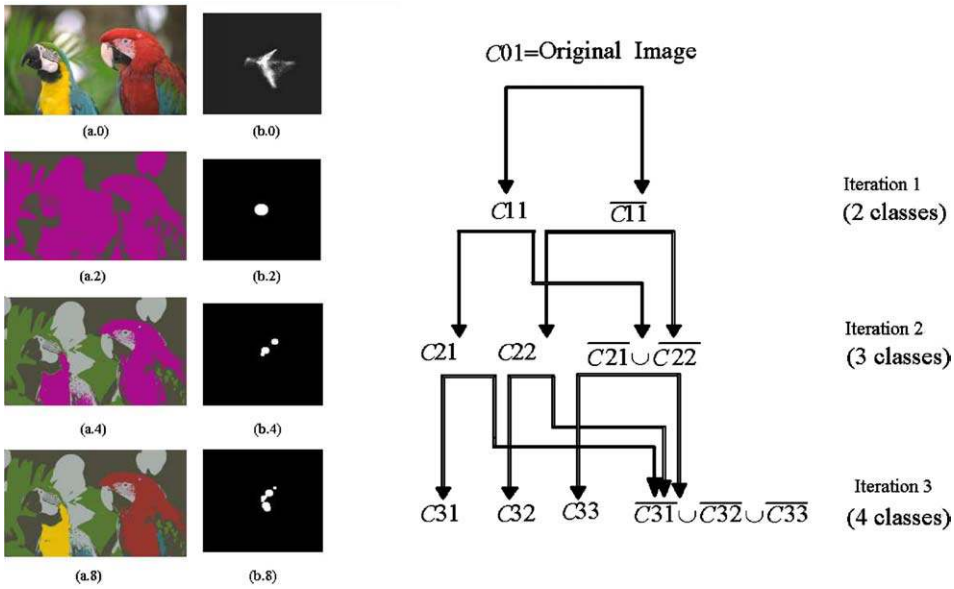

Fig. 5. Class determination with the proposed approach. (a.*i*) represents the segmented image at the *i*th iteration with *i* dominant colors, and (b.*i*) represents the histogram classes at the *i*th iteration (projection in the plan).

2.2.1. The region adjacency graph

The efficiency of an algorithm depends on the data schema used. The purpose of the algorithm is to merge adjacent similar regions. Thus a good representation to operate is a region adjacency graph [13], which is an algebraic structure that contains a set of nodes and a set of edges. Each node represents a connected region, and each edge represents an adjacency between two regions. Edges are evaluated by the color distance between the two corresponding regions.

2.2.2. General algorithm

Once the image has been quantized, we have its class-map, a gray level image with labels of the classes. First, this resulting gray levels image is spatially regularized, i.e., alone pixels are replaced by the most represented class in a V8 neighborhood.

Second, connected regions are extracted and labeled. The result is a region-map of the image. Thanks to the region-map, the RAG is processed, and distances between adjacent regions are calculated. The reduction of the graph can then be processed: at each iteration the smallest edge of the graph is eliminated, thus the corresponding regions are merged; then the graph is updated. When two regions are merged, the color of the resulting region is that of the largest region. Colors are not merged in order to keep original colors from the quantization, and also in order to prevent the merging of too many regions by a chaining phenomenon.

This graph reduction stops when the number of regions reaches the queried number chosen by the user.

2.2.3. Color difference measurement

The measure of color differences is one of the most difficult problems that exist in color distribution analysis. It depends on the color space and also on the color metric used. We use the *Lab* color space which is perceptually uniform, that is to say the Euclidean distance between two colors is proportional to the perceived difference between them. The distance D_{ij} used in our method is equal to the Euclidean distance in *Lab* space ED_{ij}, weighted by the N_{ij} coefficient introduced in [14]. N_{ij} measures the nesting between the two corresponding regions. The color distance is also weighted by a A_{ij} coefficient in order to eliminate the smallest regions:

$$D_{ij} = ED_{ij} * N_{ij} * A_{ij}, \tag{4}$$

where $N_{ij} = \min(P_i, P_j)/4P_{ij}$, P_i is the perimeter of the ith region, P_{ij} is the size of the common border between ith and jth regions, and A_{ij} is given by:

$$A_{ij} = \begin{cases} \epsilon & \text{if } (N_i < NbPixMin \text{ or } N_j < NbPixMin), \\ 1 & \text{else,} \end{cases}$$

where N_i is the number of pixels of the ith region, *NbPixMin* is a minimum number of pixels fixed by the user, and ϵ is a positive number near 0. The A_{ij} factor can be considerate as a filtering factor. When a region's area is smaller than *NbPixMin* pixels, it is considered too small, thus its distance with its adjacent regions is reduced by the coefficient, equal to ϵ. The considered region will be more easily merged with another. It is a method to eliminate the smallest regions. The value of *NbPixMin* depends on the queried size (or number) of final regions. The value of ϵ has been heuristically fixed to 0.01. This value makes it possible to considerably accelerate the fusion of the smallest regions, while keeping the merging order. The aim of the N_{ij} factor is to consider the spatial disposition of the regions in the merging decision. Regions with a large common border are more likely to belong to the same object, thus their color distance is reduced (see Fig. 6).

Criteria of speed, simplicity and adaptability to the image content were essential in the choice of the segmentation method. The parameters that this method uses are evaluated automatically by taking into account visual perception factors. Thus the number of color classes is not fixed by the user but determined by the method according to the colorimetric content of the image. The stopping criteria are the homogeneity of the color classes and their number of pixels. The only parameter to be

Fig. 6. The complete segmentation process. (A) Original image, (B) quantized image (10 classes, 2850 regions), and (C) segmented image (5 classes, 32 regions).

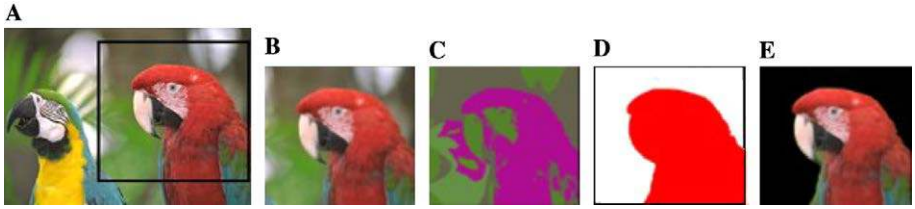

Fig. 7. OOI selection. Original image (A), selected area (B), segmented area (C), OOI defined by the user by clicking in several segmented regions (D), and the resulting OOI (E).

fixed is the number of regions, but it has a rather relative importance because, once the segmentation processed, the user defines his object of interest by clicking on regions which constitute it. In practice we have not fixed the number of regions, but a threshold for the color distance used during the region merging. Thus the number of regions is fixed, also, according to the content of the image.

Another reason for the choice of this method is the fact that the principle of the cross entropy minimization is based on the search of the threshold which maximizes the similarity between the original image and the thresholded image, which appears relevant to us for a retrieval system based on similarity and therefore by resemblance with the key image. Lastly, another advantage of our method is that it is coupled with the color feature. Indeed, the majority of the used color descriptors are determined during the segmentation step.

2.3. Object of interest selection

Fig. 7 illustrates the OOI extraction which is processed manually, with the help of segmentation. First, the user chooses the part of the image which contains what he is interested by (Fig. 7A). Then the segmentation process is performed in the selected window (Fig. 7C). Finally the user clicks on a set of regions (10 in Fig. 7C example) in order to define his OOI (Fig. 7D). The OOI is thus extracted (Fig. 7E).

Of course, the OOI definition is very dependent of the segmentation results. If the segmentation fails to distinguish the object from the background, the OOI extraction will be very complicated. This indexing method requires the intervention of an operator to select the OOI, however, it presents the benefit of not freezing the description of an image with key words, sometimes subjective, but rather to describe the image according to its contents (color, texture, and shape). We currently work on the integration of key words in the image description. Note that several OOI per image can be selected by the user.

3. Object of interest description

Color, texture, and shape are used as low-level attributes for OOI description. The high discrimination power of the color makes it one of the most widely used visual

features in image retrieval. We fixed our choice on OOI dominant color descriptor. The texture and shape features are relevant in our local approach as well as the color one. We thus used entropy and inverse moment issued from co-occurrence matrix for texture description and ART coefficients for shape description.

We think that the choice of descriptors invariance is related to the user request. In certain situations, the invariance of descriptors will be desirable (same objects on various scales, independence to brightness, etc.), whereas sometimes, the variance of some descriptors can be requested. The proposed local approach concerns OOI. We have thus supposed that the request generally relate to a given object, primarily characterized by its shape, and possibly by its texture and its color. It seemed to us, in this context, that shape descriptor invariance is important. The description step gives to each image as many descriptors as its number of selected OOI.

3.1. Color

The object of interest color descriptors are the dominant colors CD_i in the OOI, and their variance σ_i and percentage p_i [15]. The values of CD_i are supplied directly by the segmentation, and the corresponding variance and percentage are computed on the OOI. We assume that the number of dominant colors is up to 4. Tests carried out on a heterogeneous base of more than 2000 images enabled us to validate this assumption. It is thus possible to describe an object by 1, 2, 3 or 4 colors. Therefore, each OOI selected by the user is described by:

$$(\underbrace{CD_1, p_1, \sigma_1}_{\substack{\text{dominant}\\ \text{color 1}}}, \underbrace{CD_2, p_2, \sigma_2}_{\substack{\text{dominant}\\ \text{color 2}}}, \underbrace{CD_3, p_3, \sigma_3}_{\substack{\text{dominant}\\ \text{color 3}}}, \underbrace{CD_4, p_4, \sigma_4}_{\substack{\text{dominant}\\ \text{color 4}}}), \tag{5}$$

where CD_i is the color vector (L_i, a_i, b_i) corresponding to the ith dominant color, p_i is the percentage vector (p_i^L, p_i^a, p_i^b) corresponding to the ith dominant color, and σ_i is the variance vector $(\sigma_i^L, \sigma_i^a, \sigma_i^b)$ corresponding to the variances of the three color components.

3.2. Texture

The texture characterizes the relationship between adjacent pixels. Many approaches exist for its description and extraction. As we consider a local region, the texture attribute is significant and has to be used in an image retrieval scheme.

Motivated by the human perception of the texture, Tamura and al. [16] defined six parameters to describe the most important texture properties for humans. Therefore many CBIR systems used those parameters for texture description (QBIC [1], MARS [17], MetaSEEk [18], etc.). The frequency transformations are also an attractive tool in texture representation. VisualSEEk [3] and NeTra [6] systems used statistics (means and variance) extracted from wavelet sub-bands. Some authors combined different approaches, statistics and wavelet [19,20], neural networks and wavelet [21]. Comparative studies are also performed. Ma and Manjunath [22] found out that the Gabor features give better results than orthogonal, bi-orthogonal, and tree

structured wavelet transforms. In [23], Ohanian and Dubes established that co-occurrence matrix performs better than Markov Random Field, multi-channel filtering, and fractal-based representations.

We have chosen two texture parameters among the 14 parameters proposed by Haralick [24], calculated using the co-occurrence matrix relating to the image of brightness L, in the horizontal, vertical, and diagonal directions. Certain comparative studies about texture descriptors [25,26] showed that the two chosen parameters, "ent: entropy" and "idm: inverse moment," have the strongest discrimination capacity. This approach for the texture description also presents the benefit to be applied to an OOI of unspecified shape and not necessarily rectangular.

3.3. Shape

3.3.1. Region description with ART

The shape feature is performed with a generalized version of the angular radial transform (ART) [27,28] descriptor proposed by MPEG-7. ART is a region-based shape descriptor which measures the pixel distribution within a 2D object or a region. Since it is based on both boundary and internal pixels, it can describe complex objects containing multiple disconnected regions as well as simple objects with or without holes. This region-based shape descriptor belongs to the broad class of shape analysis techniques based on the moment [27]. Conceptually, the descriptor works by decomposing the shape using a number of orthogonal 2D basis functions (complex-values), defined by ART. The normalized and quantized coefficients obtained are used to describe the shape. Typically, the radial variation is 3 and the angular variation is 12 to obtain 35 coefficients.

3.3.2. ART drawbacks

Two drawbacks of ART have to be underlined. First, ART recommended by the MPEG-7 standard is limited to binary image only. ART is not adapted to color objects. In order to index an object or a segmented region in a color or a gray level image, we propose an extension which combines the shape information and the spatial distribution of the dominant colors of the region: color angular radial transform (CART) [29].

The second drawback of ART transform concerns its invariance to rotations. As the basis functions are symmetrical in the angular direction, the invariance is inherent for planar rotations. However, non-planar rotations are not taken into account. Such a rotation induces a real deformation of the original shape due to the perspective projection on the image plan. In this study, we generalize in the next section the basis functions in order to ensure invariance to non-planar rotations (generalized CART).

3.3.3. Color ART

The basis functions of the color ART are the same as those of the ART transform. The color object is first represented in the perceptually uniform (L^*, a^*, b^*) color space. As we already index the spatial distribution of the dominant colors of the

object (Section 3.1), the chrominance part of the information is not projected on the basis functions.

Only the luminance component is considered to compute the ART coefficients. Note that MPEG-7 suggests the ART transform to be applied on binary objects. Like many systems [8,30], we used the luminance for the calculation of the parameters. The application of ART on the luminance component allows to take into account the internal variations of the objects (contours, holes, texture, etc.).

3.3.4. Generalization of CART to non-planar rotations (GCART)

The goal of this generalization is to make the CART invariant to non-planar transformations that an object undergoes within an image. An object in a natural scene does not have any chance to be parallel to the plan of the camera. This highly probable situation will disturb its shape in the image and will prevent the identification. In Fig. 8, a plane object (stamp) is seen with three angles of acquisition and corresponds to three different shapes projected on the same image plane. To obtain CART descriptor invariant to non-planar rotations, it is necessary to generalize the CART transformation with new basis functions (GCART) (Fig. 8).

In order to define the transformations undergone by an object during non-planar rotations and projection on the image plane, we discretize the disturbance space according to three parameters: radial direction ς, rotation angle ψ, and perspective coefficient p. A partitioning of the space of the deformations in $K = k_\varsigma * k_\psi * k_p$ planes of projection is considered. The basis functions are deformed in a same way to obtain K sets. Each object is indexed with these K sets of projected basis functions. The number of projections is limited to have a reasonable computation complexity. The values, $k_\varsigma = 12$, $k_\psi = 3$, and $k_p = 3$, are chosen for the examples presented in Section 6. In other words, we dispose of $K = 108$ sets of coefficients to describe a shape. This means 108 distances to measure the similarity between a query and a given referenced object. This similarity measure is detailed in Section 4.1.3.

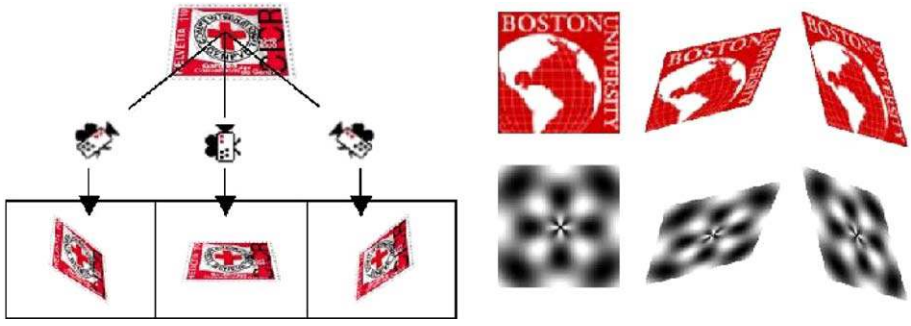

Fig. 8. Example of plane object seen according to various angles and example of basic functions projected on the support plane of the object to be identified.

4. Similarity measures

4.1. Marginal matching

4.1.1. Color similarity

We propose a new distance for color image similarity measure, in order to match the visual query to the targets [31]. The difficulty lies in the coupling of the different dominant colors of the two images we try to fit. In our case, images are described by their color descriptors (Dominant color, Percentage, and Spread out). Assuming that each dominant color can be modeled by a Gaussian distribution, a new image histogram is generated from the color descriptor. In other words, we model the 3D *Lab* distribution of the quantized images (with up to four dominant colors in this study) using a mixture of Gaussian distributions (Fig. 9) centered on each dominant color.

The histogram contribution of every dominant color CD_i is given by:

$$H(x) = \frac{p_i}{\sigma_i\sqrt{2\pi}} \exp\left\{ -\frac{(x - CD_i)^2}{2\sigma_i^2} \right\}, \tag{6}$$

where p_i is the dominant color percentage, σ_i is the variance, while the value x of the color corresponds to the bin number in the histogram. The resulting histogram is the sum of the dominant color contributions (Fig. 9). This approach allows us to integrate all the color descriptors in the modeled histogram. A Kullback distance [10] is thus performed in its symmetric form to measure the similarity between two generated distributions H_Q and H_I then the color distance between the query images Q and a database image I:

$$d_c(Q, I) = \sum_{n=1}^{N} \sum_{m=1}^{3} (q_{n,m} - i_{n,m}) \log_2 \left(\frac{q_{n,m} + 1}{i_{n,m} + 1} \right), \tag{7}$$

where N is the number of histogram bins (256), M is the number of color components ($M = 3$ for *Lab* space), $q_{n,m}$ is the percentage of the mth component of the nth color in Q, and $i_{n,m}$ is the percentage of the mth component of the nth color in I.

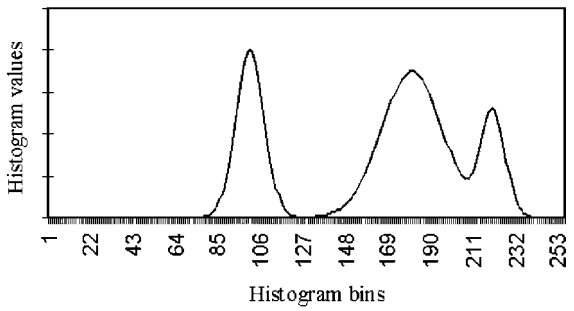

Fig. 9. Example of the generation of 1D histogram for one of *Lab* components considering three dominant colors.

4.1.2. Texture similarity

The texture similarity measurement is calculated using the Euclidean distance between the normalized texture coefficients:

$$d_{\mathrm{t}}(Q,I) = \sqrt{(ent_q - ent_i)^2 + (idm_q - idm_i)^2}. \tag{8}$$

4.1.3. Shape similarity

The ART similarity between two objects is measured by the L1 distance between their series of opposite quantified ART coefficients. The distance between the objects Q and I is measured by:

$$d_{\mathrm{ART}}(Q,I) = \frac{1}{35}\sum_{i=1}^{35} \left| Art_Q[i] - Art_I[i] \right|. \tag{9}$$

For GCART, knowing that each object is described by $K = k_\varsigma * k_\psi * k_p$ series of ART coefficients created from the basis functions projected on K planes of projections, the shape similarity distance between Q and I is achieved by computing a set of distances $d_{\mathrm{ART}}(Q,I_j)$. For each value of j, the ART coefficients of Q computed on the original basis functions and those of I computed on the jth projection of the basis functions are compared using (9). Then the shape distance between Q and I is given by:

$$d_{\mathrm{s}}(Q,I) = \min_{j\in(k_\varsigma,k_\psi,k_p)} \left(\frac{1}{35}\sum_{i=1}^{35} \left| Art_Q[i] - Art_{I_j}[i] \right| \right), \tag{10}$$

where Art_Q is the set of ART coefficients of the key object and Art_{I_j} is the set of coefficients of the I object, calculated on the jth projection of the basis functions. The minimum is considered in order to take into account all the possible perspective views of the object.

4.2. Combining features for matching

To estimate the similarity between two images, we have to evaluate the marginal similarities between their descriptors corresponding to the same attributes, then to mix them together. A global similarity function D_{G} is computed as a weighted sum of the marginal similarities:

$$D_{\mathrm{G}} = w_{\mathrm{c}}.d_{\mathrm{c}} + w_{\mathrm{s}}.d_{\mathrm{s}} + w_{\mathrm{t}}.d_{\mathrm{t}}, \tag{11}$$

where d_i and w_i represent, respectively, the marginal normalized similarity and the associated weight for the attribute i. $i \in \{\mathrm{c},\ \mathrm{s},\ \mathrm{t}\}$ for color, shape, and texture. The weights can be fixed interactively by the user according to his request or evaluated automatically by the system when the image database classes are known with respect to $w_{\mathrm{c}} + w_{\mathrm{s}} + w_{\mathrm{t}} = 1$.

5. IMALBUM: visual navigation, retrieval, and semantic content identification system

In this section, the IMALBUM system is presented. IMALBUM allows the user to select an OOI in an image, then to use it as a query for retrieval. Several tools are proposed to the user:

- Image segmentation (Section 2).
- User assistance for OOI selection (Section 2).
- OOI description (Section 3).
- Similarity evaluation (Section 4).
- OOI retrieval (Section 5).
- Visual navigation (Section 5).
- OOI content identification (Section 5).

Images that we use for our tests come from two commercial bases: Corel and Goodshoot. In both cases, images are already classified semantically. Within the framework of our paper, we have considered these classifications to be "ground truth."

5.1. Retrieval tools

Fig. 10 shows the desktop of IMALBUM system. The retrieval system, is composed of three windows. The query object is displayed in Fig. 10A, and the results

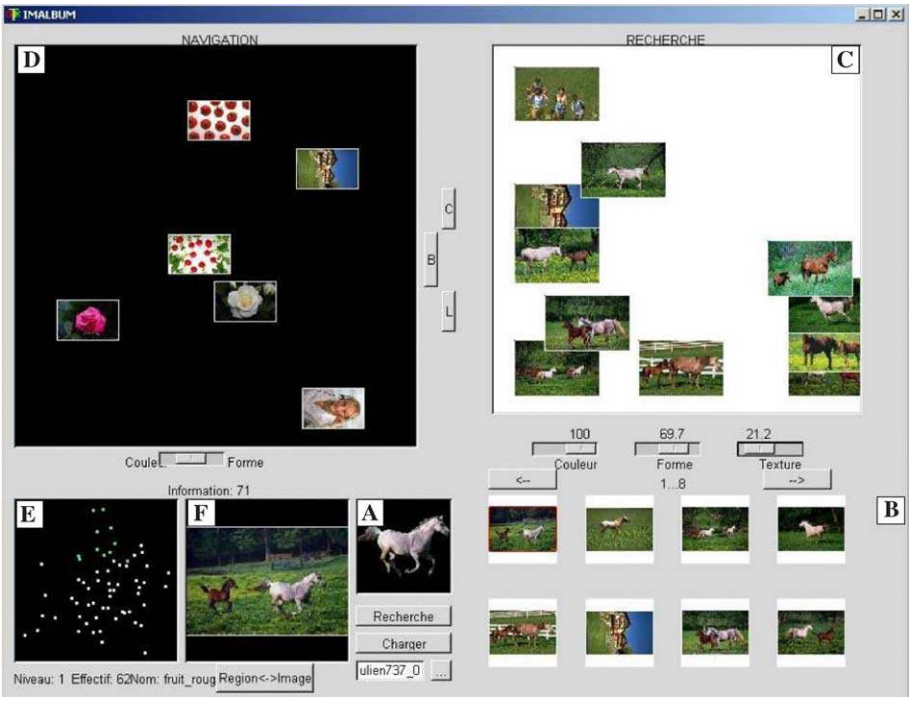

Fig. 10. The user friendly interface of IMALBUM system.

of the search are provided in Fig. 10B. Three cursors allow the user to adjust the search, according to the weights that he wants to give to the different characteristics (color, texture, and shape). In Fig. 10C, results are displayed in a parametric way, abscise and ordinate axes are descriptors chosen by the user (color and shape descriptors in Fig. 10C example). This way of displaying shows to the user how the resulting objects are distributed according to the features that he has selected as important ones.

5.2. Visual navigation tools

The direct search is particularly effective when the user searches for a specific object and knows which OOI to use as the query. If, on the contrary, he is searching for a type of object and not a precise one, the user needs another database access mechanism. The proposed system proposes to couple traditional search with a system of visual navigation. These two approaches are complementary. The navigation environment allows the user to start from an overview of the database, and iteratively zoom in on the interesting parts, until he can locate the desired object. The main problem with navigation is how to produce a visualization of the whole collection and how to provide an effective mechanism to navigate through the object database. The navigation environment can work either for OOI or for entire images. Few studies on these problems exist [32,33]. Pecenovic et al. [34] proposes to carry out a projection of Sammon [35] on the base, in order to project it on the more discriminating plane; the base is then classified in a hierarchical way, in order to allow the user to gradually zoom in towards the part which interests him. Chen et al. [36] builds, starting from the base, a hierarchical structure by agglomerating the closest objects. The proposed approach consists in classifying the database into a hierarchical tree structure, and in projecting objects into a two-dimensional space where similar objects are close to each other.

5.2.1. Browsing structure processing

Extraction of representative vectors. The aim of this stage is to extract, for each image, using the descriptors obtained during the indexing stage, a representative vector V in order to represent the database by a cloud of points in a n-dimensional space. $n = n_c + n_f + n_t$, with n_c, n_f, n_t, respective sizes of the color, shape, and texture descriptors.

$$\begin{aligned} V_i &= w_c * C_j \quad \text{for } 0 < i < n_c \text{ and } 0 < j < n_c, \\ V_i &= w_f * F_j \quad \text{for } n_c < i < n_c + n_f \text{ and } 0 < j < n_f, \\ V_i &= w_t * T_j \quad \text{for } n_c + n_f < i < n_c + n_f + n_t \text{ and } 0 < j < n_t, \end{aligned} \tag{12}$$

where C_i, F_i, and T_i represent coefficients of color, shape and texture descriptors, after normalization. w_c, w_f, and w_t are weighting factors according to the characteristics to be privileged.

Hierarchical clustering. The database is organized into a hierarchical tree structure, in order to allow the user to zoom in on any particular region. To accomplish

this we recursively cluster images in the higher dimensional space (n dimensions) into regions that contain similar images. For each found region, we pick a representative image that is closest to the centroid of the cluster. This tree structure helps users to navigate efficiently through the image collection. At each level, the clustering step is done via a k-means algorithm (a usual unsupervised fast classification method). Then clusters are regularized (elimination of smallest clusters, merging of nearest clusters). We create K clusters G_k characterized by their centroid G_k^0. Images are grouped according to their low-level descriptors (color, spatial coherency) and not according to semantic criteria. Therefore, for the user certain images seem not to be in the correct group. That is the reason why a fuzzy classification (fuzzy k-means algorithm) [37] is processed after the k-means algorithm. The goal is to allow a given image to belong to several clusters if necessary. In this algorithm, each point P_i is associated with a percentage μ_{ik} of membership to each fuzzy cluster GF_k.

- First, μ_{ik} coefficients are initialized according to the results of the k-mean algorithm

$$\begin{aligned} &\mu_{ik} = 1 \quad \text{if } P_i \in G_k, \\ &\mu_{ik} = 0 \quad \text{if } P_i \notin G_k. \end{aligned} \tag{13}$$

- Then fuzzy classes centroids G_k^0 are processed:

$$GF_k^0 = \frac{\sum_{i=0}^{N} (\mu_{ik})^m P_i}{\sum_{i=0}^{N} (\mu_{ik})^m}, \quad k = 1, \ldots, K, \tag{14}$$

where m is a weighting factor such as $m \in]0, \ldots, +\infty[$. The larger m is, the fuzzier the classification is, and the closer m is to 1, the more strict is the quantification.

- The percentages μ_{ik} of membership are updated:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{K} (d_{ik}/d_{ij})^{2/(m-1)}}, \tag{15}$$

where d_{ik} is the Euclidean distance between GF_k^0 and p_i.

- Lastly, we compare the matrices of membership U before and after the iteration:

$$\|U^P - U^{P-1}\| < \epsilon, \tag{16}$$

where P is the number of the current iteration, ϵ is the threshold, and U is such as: $U[i][j] = \mu_{ij}$. The two stages (14) and (15) are repeated until the condition (16) is checked. According to its percentages, an image can belong to several clusters.

Multivariate data projection. The purpose of this step is to construct, for each cluster, a visualizable map of images in which similar images are adjacent. The images are represented by their vectors in a high dimensional space. This task amounts to a multivariate data projection. There are several methods to accomplish this, including Principal Component Analysis [38,39], multidimensional scaling [40,41], and Sammons projection [35]. The chosen method is the PCA. It is a fast linear transformation that achieves the maximum distance preservation from the original high dimensional feature space to a 2D plane for our case. For each cluster, only

representative images of its sub-clusters are displayed, thus the PCA projection is performed only on these representative images, in order to project them on the two principal axes which carry most information (i.e., variance). Experiments show that the quantity of information (the inertia percentage) carried by the two principal axis is more than 80%.

5.2.2. The browsing mechanism

Thanks to the pre-computed results of the hierarchical structuring and PCA projection steps, the database is presented as a real time browsable space, via the navigation window (Fig. 10D). A cursor allows to choose between various hierarchical structures, according to the weights which the user wants to give to shape, color, and texture features. When entering a cluster of the hierarchical tree (A1 in Fig. 11), the user can see the representative images of each sub-clusters (B1, B2, B3, and B4), or the whole cluster if this one is final. When clicking on a representative image (B1) situated at the $(N-1)$ level, the user enters the sub-cluster situated at the (N) level. Fig. 12 shows an example of navigation through a database.

The window (Fig. 10E), presents the images of the current cluster as a cloud of points, in order to be able to judge distribution of the images and the size of sub-clusters. The user has also the possibility to select an image (Fig. 10F) and to place

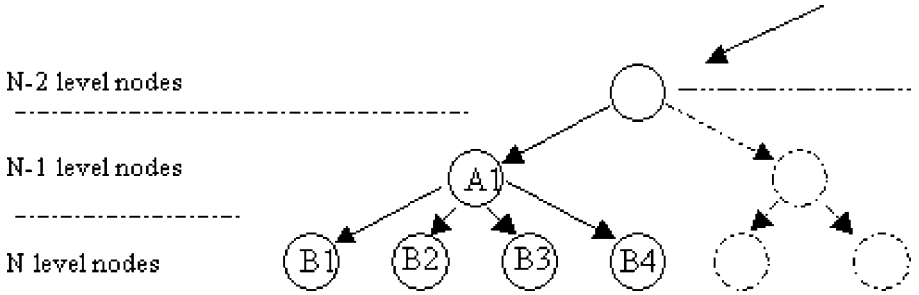

Fig. 11. Hierarchical database structure. Each node situated at the N level corresponds to a cluster of images, itself constituted of sub-clusters from the $(N+1)$ level.

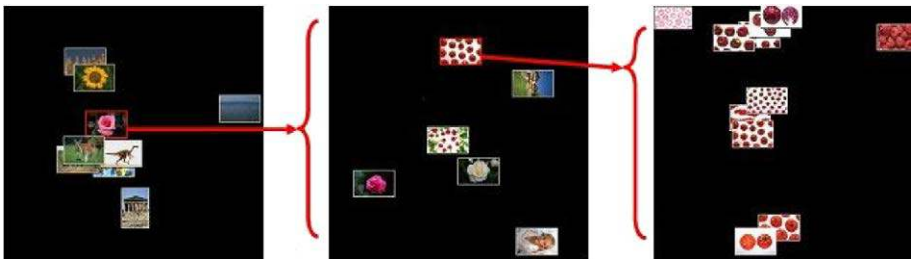

Fig. 12. An example of navigation.

himself automatically in all the final groups where this image is present, which allows a transverse navigation, in addition to in-depth navigation. Navigation and direct search are complementary tools in our system. At any level of navigation, the user can select an image in the navigation window and uses it as a key-image to launch a search. Reciprocally, following a search request, the user can select one of the returned images and visualize in the navigation window the final groups containing the selected image.

5.3. Semantic content identification tool

This tool allows to associate a semantic meaning to the query. In order to highlight the interest of such a tool, a well-referenced image database is to be considered. As detailed in Section 6.3, an image database of leaves is constituted with a priori known classes. This tool gives the degree of belonging to a probable class of objects for the unknown object used as a query.

6. Experiments

6.1. Contribution of the local approach

To evaluate the benefits of the local approach in the searching process, and the contribution of navigation as an alternative for image databases access, we carried out a set of tests on a database of 396 heterogeneous images, containing 12 classes of 33 images, indexed with the constraint to keep only one OOI per image. The class of each image corresponds to that of the identified OOI (e.g., horse, palm tree, etc.) (Fig. 13).

The recall curve is plotted using two types of visual query: each image of the database, and images not belonging to the base but containing similar OOI to those identified in the database. We have compared the marginal and combined use of color, shape, and texture features. Fig. 14 represents the mean value of the Recall parameter obtained for 396 requests.

For each request, the considered key OOI is an object of interest belonging to one of the twelve classes. The curve shows that when using *color* + *shape* + *texture* combination, the results are better than the use of only one attribute (gain of 15% for rank 66 which is twice the cardinal of each class). When considered alone, texture gives the lowest percentage because it is not as discriminate as the other features for this database.

6.2. Comparison between local and global approaches

On the same database, the comparison of Recall values for local and global approaches gives a real advantage to the local one. Indeed, if we look at the rank 66 (twice the cardinal of each class), the local approach increases the Recall values by about 16% (Fig. 15).

Fig. 13. The different classes of the database.

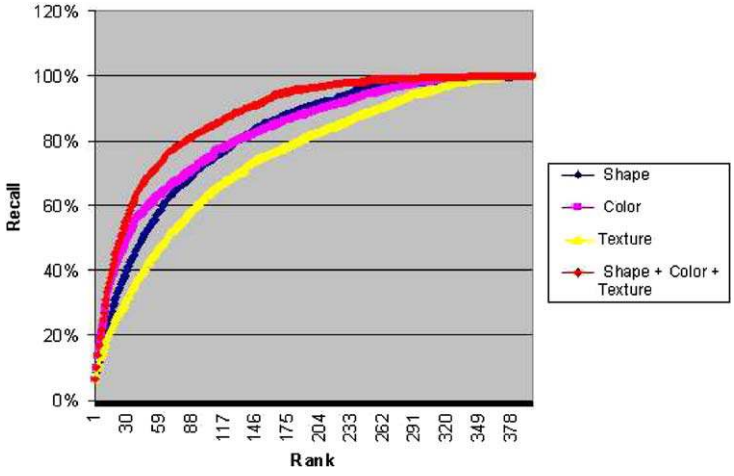

Fig. 14. Recall values obtained for different configurations.

6.3. Contribution of the navigation

The first contribution is the fact that the user does not need a key image. Without navigation tool, the user must analyze the whole database until he finds an image being able to be used as a key image. With our navigator, a global and synthetic vision of the database is offered to the user. The evaluation protocol is as follows:

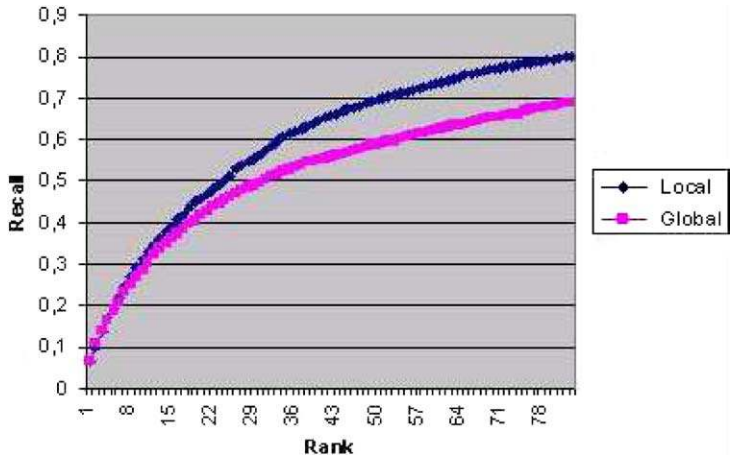

Fig. 15. Recall values obtained with local and global approaches on the database of 396 images (only the first ranks are presented).

(a) 10 novice users, without knowledge on the image database, are asked to use the system. Each user is provided with a set of 10 images, randomly selected from the database.
(b) For each image, we count the number of clicks allowing the user to find it in the database using the navigator. A way to evaluate the system consists in counting the number of images displayed to the user, before finding the target one, as it is proposed by Cox et al. [7]. We have chosen the number of clicks because the number of displayed images for each click is limited to a maximum value (15). Considering the test database (396 images), without navigation system (random browser), one would have to view 198 images (50%) before finding the target image. For 15 images per iteration, it represents 13.2 clicks on average. Thus we can, on this basis, evaluate the contribution of such a navigation system, compared with a random browser.
(c) We compute, for each user, three histograms representing the numbers of images found with less than 5 clicks, between 5 and 15 clicks, and with more than 15 clicks.
(d) The user can change the database structure (different weights for color, shape, and texture) during navigation, according to the OOI which is selected in each test image.

Note that before the evaluation, each user tries the navigator during about 5 min in order to learn in details, how to use it.

The evaluation was done with 10 users. Table 1 presents the number of clicks needed by each of them to reach to the target image. Note that the rows of this table correspond to 10 target images which were different for each user (random selection). The search was ended with 15 clicks even if the target image was not found.

When we analyze these very first results in terms of three classes of clicks (<5, between 5 and 15, and >15), two extreme classes stand out: 60% of the images are

Table 1
Number of clicks for each user to retrieve the target image

User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
2	11	15	12	7	4	9	12	6	2
4	3	15	5	4	3	12	14	2	2
15	2	15	15	9	1	2	4	2	15
9	15	4	2	2	5	2	2	15	5
3	5	3	4	6	9	15	15	15	15
2	3	3	15	7	5	15	15	4	4
15	7	4	15	4	2	5	4	15	15
8	15	2	3	3	4	15	5	3	4
4	4	2	2	3	6	3	3	3	11
2	4	3	3	15	2	2	15	2	6

found with less than 5 clicks (average of 3.14 corresponding to 45 images viewed before finding the target image), this result is very encouraging, this means that the user reaches a target image between 396 heterogeneous images with about 3 clicks in average, instead of 13.2 clicks for the whole database with the random browser. The complexity, in the number of clicks needed, is logarithmic therefore results should be even better with a larger database.

About 20% of them cannot be found with less than 15 clicks. For these images, two remarks can be done:

- Either the annotation (choice of the OOI) is not really adapted to the content.
- Or during the navigation, the user has real difficulties to decide which database structure he has to privilege (in terms of weights for color, shape, and texture). Generally, the users choose the color as the dominant feature and 4–5 clicks are necessary to go through the navigator before they understand that they have, in fact, to privilege an other feature like the shape for instance. This wrong choice makes them loose about 5–7 clicks before they can restart a navigation into the same database differently and correctly structured.

Note that less than 20% of the target images are found between 5 and 10 clicks.

6.4. Identification of the probable class of leaf

A database with 147 different tree-leaf images is considered, and an unknown real leaf that we digitized thanks to a digital camera is used as request with $w_c = 0.20$, $w_s = 0.75$, and $w_t = 0.05$. Good identification results are obtained. An example is shown in Fig. 15. The coefficients $D \in [0, 1]$ represent similarity distances of the resulted objects with the key one. The aim of this identification system is to provide the user with the most probable tree types to which the unknown leaf must belong (Fig. 16).

According to the search, the two most similar leaves are poplar leaves. The unknown leaf was indeed recognized by an expert as a Black poplar one, therefore our test is validated. This test is generalized to 50 other unknown leaves. The most similar leaf obtained by the identification tool for these test images, corresponded to the exact tree type in 48 cases.

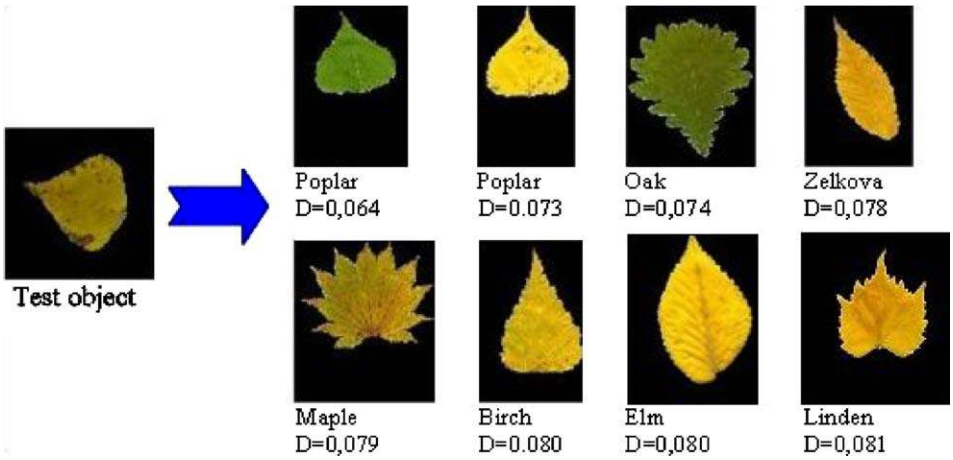

Fig. 16. Similarity distance of nearest database objects.

7. Conclusion

The content-based image retrieval system IMALBUM was presented. This color image indexing and retrieval scheme is based on objects of interest segmentation and indexing. The proposed system provides several new functionalities to the user:

- The indexing is done with a user-friendly interface. The user has the possibility to select one or several key regions issued from the segmentation for local content indexing.
- For the retrieval, the image is automatically segmented and the user can select an object of interest (OOI) (defined as the union of several segmented regions) to define precisely his query. The retrieval process then finds all the images of the database which contain one or several similar OOI.
- A visual navigation tool allows to go through the image database (''surfing'') when the user does not have any given object of interest to use as a query. The experiment presented in this study, shows that more than 60% of target images are found with 3 clicks using the navigator through an experimental heterogeneous database.
- When the image database is composed of well-defined classes of objects as is the case for the different databases, the semantic content analysis and identification tool proposes a percentage of belonging to one of these classes for an unknown object launched as a query. So we have the possibility of associating a semantic meaning to the query.

We are investigating the possibility of analyzing the semantic meaning of the union of several objects of interest selected by the user for the same query. We are also working on the integration of relevance feedback in our process, in order to dynamically and visually determine the weights for color, texture, and shape features. The history of visual navigation path will also lead us to modelize a profile for the user.

References

[1] W. Niblack, R. Barber, W. Equitz, M. Flickher, E. Glasman, D. Petkovic, P. Yanker, The QBIC project: querying images by content using color, texture, and shape, in: Conf. on Storage and Retrieval for Image and Video Databases, vol. 25, no. 8, 1993, pp. 173–187.

[2] A. Pentland, R. Picard, S. Sclaroff, Photobook: tools for content-based manipulation of image databases, Internat. J. Comput. Vision 18 (3) (1996) 233–254.

[3] J. Smith, S. Chang, Querying by color regions using the VisualSEEk contentbased visual query system, in: M.T. Maybury (Ed.), Intelligent Multimedia Information Retrieval, AAAI Press, 1997.

[4] H. Zhang, C. Low, S. Smoliar, J. Wu, Video parsing, retrieval and browsing: an integrated and content-based solution, in: The Third ACM Internat. Conf. on Multimedia, 1995, pp. 15–24.

[5] M.S. Lew, K. Lempinen, N. Huijsmans, Webcrawling using sketches, in: The Second Internat. Conf. on Visual Information Systems, 1997, pp. 77–84.

[6] W. Ma, B. Manjunath, NeTra: A toolbox for navigating large image databases, Multimedia Syst. 7 (3) (1999) 184–198.

[7] I. Cox, M. Miller, T. Minka, T. Papathormas, P. Yianilos, The bayesian image retrieval system, PicHunter: theory, implementation and psychophysical experiments, IEEE Trans. on Image Process. 9 (1) (2000) 20–37.

[8] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (9) (2001) 947–963.

[9] K. Idrissi, J. Ricard, A. Baskurt, Multi-component cross entropy segmentation for color image retrieval, in: The Second Internat. Symp. on Image and Signal Processing and Analysis, 2001, pp. 132–137.

[10] S. Kullback, Information Theory and Statistics, Wiley, New York, 1959.

[11] C. Li, C. Lee, Minimum cross entropy thresholding, Pattern Recogn. 26 (4) (1993) 617–625.

[12] Y. Zimmer, R. Tepper, S. Akselrod, A two-dimensional extension of minimum cross entropy thresholding for the segmentation of ultrasound images, Ultrasound Med. Biol. 29 (9) (1996) 1183–1190.

[13] K. Saarinen, Color image segmentation by a watershed algorithm and region adjacency graph processing, in: Internat. Conf. on Image Processing, 1994, pp. 1021–1024.

[14] R. Schettini, A segmentation algorithm for color images, Pattern Recogn. Lett. 14 (6) (1993) 499–506.

[15] K. Idrissi, J. Ricard, A. Anwander, A. Baskurt, An image retrieval system based on local and global color descriptors, in: The Second IEEE Pacific-Rim Conf. on Multimedia, Beijing, 2001, pp. 55–62.

[16] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Trans. Syst. Man Cybernet. 8 (6) (1978) 460–473.

[17] T. Huang, S. Mehrotra, K. Ramchandran, Multimedia analysis and retrieval system (MARS) project, in: The 33rd Annual Clinic on Library Application of Data Processing—Digital Image Access and Retrieval, 1996, pp. 0.

[18] M. Beigi, A.B. Benitez, S.-F. Chang, MetaSEEk: a content-based metasearch engine for images, in: Storage and Retrieval for Image and Video Databases (SPIE), 1998, pp. 118–128.

[19] K. Thyagarajan, J. Nguyen, C. Persons, A maximum likelihood approach to texture classification using wavelet transform, in: Internat. Conf. on Image Processing (ICIP), vol. 94, 1994, pp. 640–644.

[20] A. Kundu, J. Chen, Texture classification using QMF bank-based subband decomposition, Graphical Models Image Process. (CVGIP) 54 (1992) 369–384.

[21] C. Busch, M. Gross, Interactive neural network texture analysis and visualization for surface reconstruction in medical imaging, EUROGRAPHICS '93, Comput. Graph. Forum 12 (3) (1993) 49–60.

[22] W. Ma, B. Manjunath, A comparison of wavelet transform features for texture image annotation, in: Internat. Conf. on Image Processing (ICIP), vol. 2, 1995, pp. 256–259.

[23] P.P. Ohanian, R.C. Dubes, Performance evaluation for four classes of texture features, Pattern Recogn. 25 (2) (1992) 819–833.

[24] R. Haralick, Statistical and structural approaches to texture, in: IEEE, vol. 67, no. 5, 1979, pp. 786–804.

[25] C. Gotlieb, H. Kreyszig, Texture descriptors based on co-occurrence matrices, in: Computer Vision, Graphics and Image Processing, vol. 51, 1990, pp. 70–86.
[26] R. Conners, C. Harlow, A theoretical comparison of texture algorithms, IEEE Trans. PAMI 2 (3) (1980) 204–222.
[27] S. Jeannin, Mpeg-7 visual part of experimentation model version 9.0, in: ISO/IEC JTC1/SC29/WG11/N3914, 55th Mpeg Meeting, Pisa, 2001.
[28] W.-Y. Kim, Y.-S. Kim, A new region-based shape descriptor, in: ISO/IEC MPEG99/M5472, TR 15-01, Maui, Hawaii, 1999.
[29] M. Akcay, A. Baskurt, B. Sankur, Measuring similarity between color image regions, in: EUSIPCO'02, vol. 1, Toulouse, France, 2002, pp. 115–118.
[30] J. Laaksonen, J. Koskela, S.P. Laakso, E. Oja, PicSOM: Content-based image retrieval with self-organizing maps, Pattern Recogn. Lett. 21 (13–14) (2000) 1199–1207.
[31] K. Idrissi, J. Ricard, A. Baskurt, An objective performance evaluation tool for color based image retrieval systems, in: Internat. Conf. on Image Processing (ICIP02), vol. 2, 2002, pp. 389–392.
[32] A. Hiroike, Y. Musha, A. Sugimoto, Y. Mori, Visualization of information spaces to retrieve and browse image data, in: Third Internat. Conf. on Visual Information Systems, 1999.
[33] T.S.H. Munehiro Nakazato, 3d mars: immersive virtual reality for content based image retrieval, in: IEEE Internat. Conf. on Multimedia and Expo (ICME2001), 2001.
[34] Z.P. Pecenovic, M. Do, M. Vetterli, P. Pu, Integrated browsing and searching of large image collections, in: The Fourth Internat. Conf. on Visual Information and Information Systems (VISUAL), 2000, pp. 279–289.
[35] J. Sammon, A nonlinear mapping for data structure analysis, IEEE Trans. Comput. 18 (5) (1969) 401–409.
[36] J. Chen, C. Bouman, J. Dalton, Similarity pyramids for browsing and organization of large image databases, in: SPIE/IS&T Conf. on Human Vision and Electronic Imaging III, vol. 3299, 1998, pp. 563–575.
[37] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[38] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
[39] B. Moghaddam, Q. Tian, T.S. Huang, Spatial visualization for content based image retrieval, in: IEEE Internat. Conf. on Multimedia and Expo (ICME'01), Waseda University, Tokyo, Japan, 2001.
[40] W.S. Torgeson, Theory and Methods of Scaling, Wiley, NewYork, 1958.
[41] Y. Rubner, Perceptual metrics for image database navigation, Ph.D. thesis, Stanford University, Stanford University, CA, USA, 1999.